Kernel nonnegative matrix factorization : application to hyperspectral imagery

Fei Zhu

▶ **To cite this version:**

Fei Zhu. Kernel nonnegative matrix factorization : application to hyperspectral imagery. Machine Learning [cs.LG]. Université de Technologie de Troyes, 2016. English. NNT : 2016TROY0024 . tel-03361933

HAL Id: tel-03361933
https://theses.hal.science/tel-03361933

Submitted on 1 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse
de doctorat
de l'UTT

Fei ZHU

Kernel Nonnegative Matrix Factorization: Application to Hyperspectral Imagery

Spécialité :
Optimisation et Sûreté des Systèmes

2016TROY0024

Année 2016

utt
université de technologie
Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Fei ZHU

le 19 septembre 2016

Kernel Nonnegative Matrix Factorization: Application to Hyperspectral Imagery

JURY

M. C. JUTTEN	PROFESSEUR DES UNIVERSITES	Président
M. D. BRIE	PROFESSEUR DES UNIVERSITES	Rapporteur
M. X. BRIOTTET	DIRECTEUR DE RECHERCHE ONERA	Examinateur
M. P. HONEINE	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. C. RICHARD	PROFESSEUR DES UNIVERSITES	Rapporteur
M. H. SNOUSSI	PROFESSEUR DES UNIVERSITES	Examinateur

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor and mentor Prof. Paul Honeine. I am truly grateful for his continuous guidance, motivation and support, without which this work would not have been possible. Thanks to his guidance in both my master's internship and this thesis, I have been introduced to the domain of hyperspectral imagery processing and become a problem solver. During my research work, he keeps immense interest on my topic of research and always been available to offer valuable comments and advises.

I gratefully acknowledge the members of my Ph.D defense committee. Sincere thanks to Prof. Cédric Richard and Prof. David Brie for devoting their time to reading this manuscript and providing insightful comments. I thank also Prof. Christian Jutten, Prof. Xavier Briottet and Prof. Hichem Snoussi, who have kindly accepted to serve on the committee.

A special thank goes to Abderrahim Halimi. Working with him helped me with a deeper understanding on the topic of nonlinear hyperspectral unmixing. The work on correntropy maximization via ADMM would not have been realized without the numerous discussions we have made. I thank also Jie Chen for his kindness of sharing research experience, as well as many hyperspectral data and Matlab codes.

I gratefully acknowledge all the members of Systems Modeling and Dependability Laboratory (LM2S) of Université de Technologie de Troyes (UTT), for their enormous help and friendship from my master period to Ph.D period. I thank every lecturer, who have given excellent courses and guided my experimental works. I thank all the colleagues for sharing a comfortable research environment with me. Although I cannot name everyone, I have felt the kindness and friendship from them over coffee-talks, discussions and even smiles we exchanged when crossed in the corridor.

I would like to express my gratitude to all the secretaries who have kindly helped me: Mesdames Véronique Banse and Bernadette André in LM2S, and Mesdames Pascale Denis, Isabelle Leclercq and Thérèse Kazarian in doctoral school. Thank for their availability and help during my study at UTT.

My time at UTT was made enjoyable and memorable in large part due to my friends. Particular thanks go to Heping, Yanyan, Yaofu, Xiaowei, Wenjin, Sandy, Patrick and

Nisrine, for accompanying me all the way and letting me not give up in front of stress and pressure. Words cannot express my gratitude to them.

I would like to thank China Scholarship Council (CSC) for its generous financial support during the 60 months, which made possible my study in France.

Last but not least, I owe my deepest gratitude to my parents for their endless understanding, support and love. They have always been the source of courage in my life.

To my parents,

Abstract

This thesis aims to propose new nonlinear unmixing models within the framework of kernel methods and to develop associated algorithms, in order to address the hyperspectral unmixing problem. First, we investigate a novel kernel-based nonnegative matrix factorization (NMF) model, that circumvents the pre-image problem inherited from the kernel machines. Within the proposed framework, several extensions are developed to incorporate common constraints raised in hypersepctral images analysis. In order to tackle large-scale and streaming data, we next extend the kernel-based NMF to an online fashion, by keeping a fixed and tractable complexity. Moreover, we propose a bi-objective NMF model as an attempt to combine the linear and nonlinear unmixing models. The decompositions of both the conventional NMF and the kernel-based NMF are performed simultaneously. The last part of this thesis studies a supervised unmixing model, based on the correntropy maximization principle. This model is shown robust to outlier bands. Two correntropy-based unmixing problems are addressed, considering different constraints in hyperspectral unmixing problem. The alternating direction method of multipliers (ADMM) is investigated to solve the related optimization problems.

Keywords:

- Hyperspectral imagery
- Machine learning
- Nonlinear models
- Factorization (Mathematics)
- Non-negative matrices

Résumé

Cette thèse vise à proposer de nouveaux modèles pour la séparation de sources dans le cadre non linéaire des méthodes à noyaux en apprentissage statistique, et à développer des algorithmes associés. Le domaine d'application privilégié est le démélange en imagerie hyperspectrale. Tout d'abord, nous décrivons un modèle original de la factorisation en matrices non négatives (NMF), en se basant sur les méthodes à noyaux. Le modèle proposé surmonte la malédiction de préimage, un problème inverse hérité des méthodes à noyaux. Dans le même cadre proposé, plusieurs extensions sont développées pour intégrer les principales contraintes soulevées par les images hyperspectrales. Pour traiter des masses de données, des algorithmes de traitement en ligne sont développés afin d'assurer une complexité calculatoire fixée. Également, nous proposons une approche de factorisation bi-objective qui permet de combiner les modèles de démélange linéaire et non linéaire, où les décompositions de NMF conventionnelle et à noyaux sont réalisées simultanément. La dernière partie se concentre sur le démélange robuste aux bandes spectrales aberrantes. En décrivant le démélange selon le principe de la maximisation de la correntropie, deux problèmes de démélange robuste sont traités sous différentes contraintes soulevées par le problème de démélange hyperspectral. Des algorithmes de type directions alternées sont utilisés pour résoudre les problèmes d'optimisation associés.

Mots-clés :

- Imagerie hyperspectrale
- Apprentissage automatique
- Modèles non linéaires (statistique)
- Factorisation
- Matrices nonnégatives

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

Data processing and analysis is an increasingly active research field with great opportunities in plenty of real-world applications. The main objective is to extract valuable information and to infer hidden patterns from data of high dimension and/or of massive volume. Of particular interest is blind source separation with the unmixing of hyperspectral images in remote sensing for earth observation. These images have a spectrum for each pixel, with high spectral resolution but limited spatial resolution. The former enables the capacity of performing spectroscopic analysis to identify materials and describe processes, while the latter leads to the fact that each pixel in the image is a mixture of several pure materials, termed endmembers. The spectral unmixing consists in extracting these endmembers and estimating their contributions, termed abundances, at each pixel. It is an ill-posed inverse problem, and nonlinear unmixing remains an open and challenging issue. This chapter first introduces the concepts of hyperspectral images and spectral unmixing. Strategies to tackle separately or jointly the endmember extraction and abundance estimation are discussed. Next, the prevalent linear mixing model and several nonlinear models are presented. Finally, the structure of this manuscript and the main contributions of the thesis are outlined.

1.1 General Background

Recent years have witnessed a great proliferation of the ability to collect sensory data, including visual images, audio recordings, linguistic data, to name a few in the era of Big Data. The processing and analysis of the captured data require innovative and efficient approaches, thus presenting a great challenge to researchers. A prominent issue to address is the extraction of valuable information and the inference of hidden patterns from the raw data. These data, gathered from various sensors, are often stored in the form of matrix or tensor. In most cases, they are essentially composed of a few interrelated variables, or combined with several underlying factors or components [Cichocki et al., 2009]. In order to explore the hidden structure and extract useful information, one seeks to decompose or factorize the data into some relevant components to be determined.

Low-dimensional representations of high-dimensional data play a significant role in enhancing the data and extracting underlying components [Cichocki et al., 2009]. Of particular interest are the latent factor models based on matrix factorization (*i.e.*, decomposition). These models have been successfully applied for noise removal, model reduction, signal reconstruction, and more generally for blind source separation (BSS) [Comon and Jutten, 2010], with a wide range of real-world applications including recommendation system [Koren et al., 2009], clustering [Ding et al., 2005; Xu et al., 2003], biomedical signal processing [Wang et al., 2013a], audio signal processing [Févotte et al., 2008], and hyperspectral data analysis [Jia and Qian, 2009]. In the discipline of linear algebra, matrix factorization (or decomposition) consists in approximating a given data matrix by a product of two (or more) matrices with lower rank.

In order to deal with different constraints arising in real-word applications, the model of matrix factorization may vary by considering different assumptions with respect to the component matrices and the latent structures. A typical example is encountered when both the data under study and the unknown matrices under estimation are nonnegative, for the sake of meaningful physical interpretation such as in spectral extraction. In this case, one modifies the matrix factorization model by imposing the nonnegativity constraints on the unknown components (matrices), leading to the so-called nonnegative matrix factorization (NMF) [Lee and Seung, 1999, 2001]. Another difficulty arises when dealing with large volumes of data, such as with streaming data. In this case, the batch mode becomes inappropriate and inefficient for matrix factorization. Instead, matrix factorization techniques adapted to large-scale data processing are required, leading to online matrix factorization approaches [Mairal et al., 2010], and particularly online NMF when the nonnegativity is imposed [Cao et al., 2007; Wang et al., 2011]. Other constraints include promoting the sparseness [Kim and Park, 2007] or smoothness of the representation [Pauca et al., 2006; Jia and Qian, 2009; Qian et al., 2011]. Usually, these variations of the matrix factorization model provide more suitable models for discovering the underlying components, and help to avoid meaningless or unreasonable results.

Within the framework of blind source separation, the linear mixtures have been intensively investigated in the literature. In many real-word applications, it is however more natural to consider the general case of nonlinear mixtures, since nonlinearities may occur at different stages of the processing, including the mixing process and nonlinearities in

the used sensors (*e.g.*, post-nonlinear) [Comon and Jutten, 2010, Chap. 14]. In hyperspectral imagery, critical nonlinear effects often exist in the data under study, as reported in several studies with either physically inspired models or ground truth information (See Section 1.4) [Heylen et al., 2014]. Some efforts have been carried out to extend the linear NMF to the scope of nonlinear models, such as the quadratic NMF for bilinear models [Yang and Oja, 2012]. Some attempts have been made to provide more general nonlinear NMF variants, by investigating the framework of kernel machines [Zhang et al., 2006; Ding et al., 2010; Li and Ngom, 2012]. Providing nonlinear models and developing corresponding algorithms for NMF remain an open and challenging issue.

This thesis focuses on nonlinear spectral unmixing in hyperspectral imagery. The spectral unmixing will be introduced in the next section, before providing an overview of the linear and nonlinear models, respectively in Sections 1.3 and 1.4. Since the framework of NMF is natural to solve the spectral unmixing problem, it will be revisited in Chapters 3, 4 and 5 for nonlinear unmixing, while Chapter 6 studies a robust estimation. A more detailed state-of-the-art of the NMF and its variants is given in Section 3.1.

1.2 Spectral Unmixing of Hyperspectral Images

A hyperspectral image details the scene under scrutiny with spectral observations of electromagnetic waves emission/reflection. Typically, it corresponds to the acquisition of a ground scene from which sunlight is reflected. A hyperspectral image is a three-dimensional data cube, two of the dimensions being spatial, and the third one being the spectral. In other words, a spectrum (*i.e.,* reflectance vector) characteristic is available at each pixel. Figure 1.1 gives an example of hyperspectral image.

Usually, the spectral dimension is up to several hundreds, corresponding to the spectral bands across a continuous wavelength range. For example, the NASA *airborne visible/infrared imaging spectrometer* (AVIRIS), in operation since 1989, has 224 contiguous spectral bands, covering from 0.4 to 2.5 μm, with a ground resolution that varies from 4 to 20 m (depending on the distance of the airborne to the ground). Due to such spatial resolution, any acquired spectrum is a superposition of spectra of several distinct underlying materials.

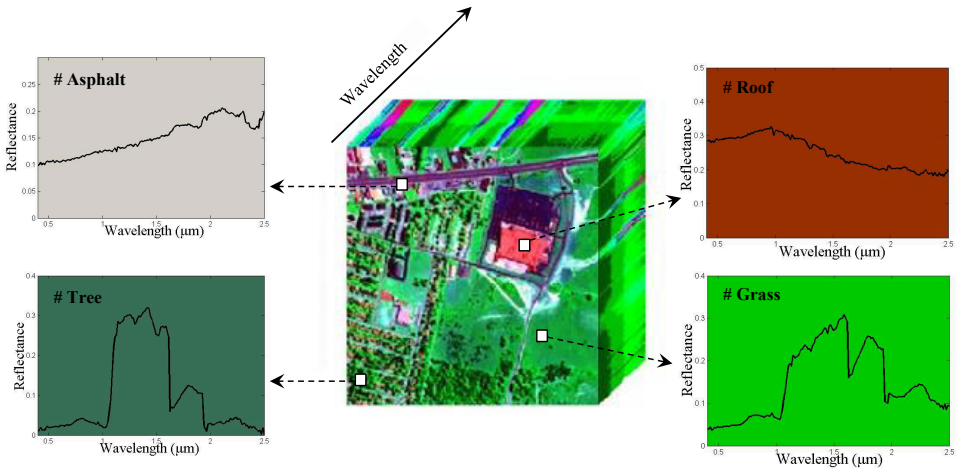


Figure 1.1: In a hyperspectral image, each pixel is a (reflectance) spectrum that deals with narrow spectral bands over a continuous spectral range. Such redundancy in the information allows to identify the material present in the scene, namely endmembers.

The unmixing of a given hyperspectral image aims to extract the spectra of these "pure" materials, called *endmembers*, and to estimate the fractional *abundances* of these endmembers in every pixel, *i.e.,* every position of the area under scrutiny. The unmixing is a challenging ill-posed inverse problem, in the same sense as blind source separation problems [Comon and Jutten, 2010]. Its ill-posedness[1] needs to be tackled by adopting in advance the mixing model, either linear or nonlinear models, and incorporating constraints such as nonnegativity, sparseness, smoothness, and spatial regularization. Once the model is fixed, the unmixing is addressed either in a divide-to-conquer scheme, by separately extracting the endmembers prior to estimating their abundances, or jointly estimating the endmembers and abundances.

1.2.1 Endmember extraction

Endmember extraction is a primary step in hyperspectral image unmixing, assuming their number is known in advance, either from ground truth or estimated using for instance [Halimi et al., 2016]. In some very rare cases, the endmembers can be known from a ground truth examination of the scene, and using some spectral library. The unmixing is then reduced to the abundance estimation step. However, the endmembers are unknown in most of the cases; they have to be identified from the observed spectra

[1] According to Hadamard, a problem is well-posed if the following three conditions are satisfied: it has a solution, the solution is unique, and the solution depends continuously on data and parameters. If any of these conditions is not fulfilled, the problem is said ill-posed.

within the hyperspectral image under study. Over the past decades, various endmember estimation methods have been proposed, as presented in the overview [Plaza et al., 2012]. Endmember extraction techniques can be split in two classes, depending if the techniques are designed based on the pure pixel assumption, namely, the endmembers are pure pixels (signatures) present in the hyperspectral image under study.

The first class brings together endmember extraction techniques that rely on the pure pixel assumption. These techniques, as a consequence, search in the image for the purest pixels in the spectral sense. Such methods include, but not limited to, the orthogonal subspace projection (OSP) [Harsanyi and Chang, 1994], the N-Findr [Winter, 1999], and the vertex component analysis (VCA) [Nascimento and Bioucas-Dias, 2005]. The N-Findr and VCA techniques seek to inflate the simplex enclosing all the spectra, and determines the endmembers as the vertices of the largest simplex. As a preprocessing step for such algorithms, a dimensionality reduction technique often needs to be applied, such as with the conventional principal component analysis (PCA).

Unfortunately, the assumption of pure pixels does not hold when the pixels are completely mixed. To this end, several recent works have been conducted to abandon this assumption, by estimating endmembers that are not necessarily present in the image, in the same spirit as estimating the abundances. This class of endmember extraction techniques rely often on the joint estimation of the endmembers and abundances as described in the upcoming Section 1.2.3. This class includes the NMF-based unmixing techniques, as well as the iterative constrained endmembers (ICE) [Berman et al., 2004].

1.2.2 Abundance estimation in supervised unmixing

While the endmember extraction is relatively easy from geometry, the abundance estimation remains an open problem, also referred to as the supervised unmixing problem. Given the identified endmembers, the abundance estimation (referred as inversion in [Bioucas-Dias et al., 2012; Dobigeon et al., 2014]) generally involves the minimization of the residual error between the observed spectra (pixels) and the inferred spectra.

As to be explained next, the abundance nonnegativity constraint (ANC) and the abundance sum-to-one constraint (ASC) are often required. Considering both constraints, the fully-constrained least-squares (FCLS) [Heinz and Chang, 2001] yields the optimal

abundances in the least-squares sense. A more recently proposed algorithm is the sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) [Bioucas-Dias and Figueiredo, 2010]. A fully-constrained variant of SUnSAL (SUnSAL-FCLS) addresses the same optimization problem as FCLS by taking advantage of the alternating direction method of multipliers (ADMM) [Boyd et al., 2011]. Besides the least-squares methods, other strategies have been proposed by employing the geometric explanation of the unmixing process such as the barycentric coordinate approach [Honeine and Richard, 2012]; or by tackling the recently-raised nonlinearity issue such as the linear-mixture/nonlinear-fluctuation model [Chen et al., 2013b], the post nonlinear model [Chen et al., 2013c] and the generalized bilinear model [Halimi et al., 2011b].

1.2.3 Joint estimation or unsupervised unmixing

In hyperspectral unmixing problems, a prior knowledge on endmembers is often unavailable. Instead of identifying endmembers and abundances sequentially, unsupervised unmixing provides an alternative by estimating the endmembers and abundances simultaneously from the observed data bulk, without much user interaction [Miao and Qi, 2007]. Assuming that the observed spectra are mixed by the linear model, the unsupervised unmixing can be considered as a blind source separation (BSS) problem, where typical BSS techniques are feasible [Comon and Jutten, 2010]. For example, independent component analysis (ICA) has been applied to hyperspectral unmixing in [Wang and Chang, 2006; Chang et al., 2002; Moussaoui et al., 2008]. However, the statistical independence assumption is not satisfied in general for spectral unmixing [Chang et al., 2002].

Among the BSS techniques, the nonnegative matrix factorization (NMF) [Lee and Seung, 1999] is a crucial one that has been widely accepted for its effectiveness in unsupervised hyperspectral unmixing [Jia and Qian, 2009; Lu et al., 2013; Huck et al., 2010; Févotte and Dobigeon, 2015]. The basic idea of NMF is to approximate an input nonnegative matrix (composed column-wise by the observed spectra) by the product of two unknown lower-rank nonnegative matrices, the first recording the endmembers and the second recording the abundances. A variant of the NMF is the semi-NMF, where the nonnegativity is imposed on a single unknown matrix. In the context of the hyperspectral unmixing problem, there are two main advantages of NMF over other

unsupervised approaches, *e.g.,* PCA and ICA. At first, the nonnegativity constraints on the input and the two unknown matrices are consistent with the nature of unmixing, where the observed pixels, the endmember spectra and the fractional abundances should be nonnegative by nature, as to be explained next. Second, NMF is able to provide a parts-based representation based on the additivity of the contributions of the bases to approximate the original data [Lee and Seung, 1999], in opposition to the holistic methods such as PCA and ICA.

These unmixing-driven NMF techniques include the minimum dispersion constrained NMF (MiniDisCo) [Huck et al., 2010]. This method includes the dispersion regularization to the conventional NMF, by integrating the sum-to-one constraint for each pixel's abundance fractions and the minimization of variance within each endmember. The problem is solved by exploiting an alternate projected gradient scheme. Another example is the robust nonnegative matrix factorization (rNMF) proposed in [Févotte and Dobigeon, 2015]. To capture the nonlinear effect (outliers), this NMF-based method introduces a group-sparse regularization term into the linear model. Accounting for both constraints, the problem is optimized by a block-coordinate descent strategy. A more detailed state-of-the-art of the NMF and its variants is given in Section 3.1.

1.3 Linear Mixing Model

Regardless of the different types of unmixing (supervised or unsupervised), the underlying mixing mechanism needs to be defined properly, namely, how the observed spectrum is generated from the endmembers. To this end, the main models and the corresponding unmixing strategies are roughly divided into two categories: the linear-based model and the nonlinear-based ones. This section describes the linear mixing model and the main associated algorithms, while next section extends this definition to nonlinear models.

The linear mixing model (LMM) assumes the mixing in the so-called macroscopic scale, namely each arriving photon interacts with only one endmember before reaching the sensor [Bioucas-Dias et al., 2012]. The light reflected from distinct endmembers is then mixed within the instrument. Consequently, each captured spectrum can be expressed as a linear combination of a set of endmembers.

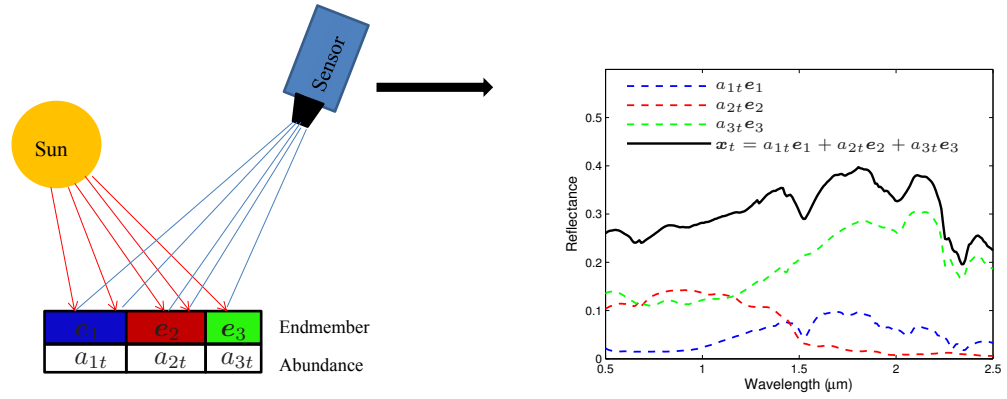


Figure 1.2: Illustration of the linear mixing. The observed spectrum $\boldsymbol{x}_t$ is a combination of the endmembers $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$, with the respective weights a_{1t}, a_{2t}, a_{3t} called abundances.

Consider a hyperspectral image of T pixels, each being a reflectance spectrum of L spectral bands in some given continuous wavelength range. Neglect for now the spatial relations of its pixels by rearranging the three-dimensional data cube as a conventional matrix where each column is an observed spectrum. Let $\boldsymbol{X} \in \mathbb{R}^{L \times T}$ denote the matrix of the T pixels/spectra of L spectral bands. Let $\boldsymbol{x}_{*t}$ be its t-th column representing the observed reflectance spectrum of t-th pixel, and $\boldsymbol{x}_{l*}$ its l-th row representing the l-th spectral band over all pixels. For notation simplicity, we denote $\boldsymbol{x}_t = \boldsymbol{x}_{*t}$, for $t = 1, \ldots, T$. The LMM can be written as

$$\begin{aligned} \boldsymbol{x}_t &= \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n + \boldsymbol{n}_t \\ &= \boldsymbol{E}\boldsymbol{a}_t + \boldsymbol{n}_t, \end{aligned} \tag{1.1}$$

where $\boldsymbol{E} = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_N] \in \mathbb{R}^{L \times R}$ is the matrix composed by the N endmembers with $\boldsymbol{e}_n = [e_{1n} \ \cdots \ e_{Ln}]^\top$, $\boldsymbol{a}_t = [a_{1t} \ \cdots \ a_{Nt}]^\top$ is the abundance vector associated with the t-th pixel, and $\boldsymbol{n}_t \in \mathbb{R}^L$ is the additive noise. In matrix form for all pixels, we have

$$\boldsymbol{X} = \boldsymbol{E}\boldsymbol{A} + \boldsymbol{N}, \tag{1.2}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_T] \in \mathbb{R}^{R \times T}$ and $\boldsymbol{N}$ is the noise matrix. The matrix factorization consists in estimating both matrices $\boldsymbol{E}$ and $\boldsymbol{A}$, where $\boldsymbol{E}$ is often termed basis matrix and $\boldsymbol{a}_t$ is termed the encoding vector.

The LMM holds for the flat scene with separated regions corresponding to distinct

endmembers (referred as checkerboarder pattern in [Heylen et al., 2014]), as illustrated in Figure 1.2. Under such assumption, the abundances represent the proportions of regional surface occupied by the corresponding endmember at the pixel. Such physical interpretation leads to two constraints that are commonly considered in unmixing: the abundance nonnegativity constraint (ANC) given by

$$a_{nt} \geq 0, \forall\ n \text{ and } t, \tag{1.3}$$

and the abundance sum-to-one constraint (ASC) given by

$$\sum_{n=1}^{N} a_{nt} = 1,\ \forall\ t. \tag{1.4}$$

The ASC is a controversy due to the considerable signature variability present in a real hyperspectral image, where at least a positive scale factor should be included at each pixel [Iordache et al., 2011]. In this manuscript, we will make precise whether an algorithm relaxes ASC or not. Representing reflectance spectra, the endmembers should be nonnegative by nature, namely $\boldsymbol{e}_n \geq 0$, for $n = 1, \ldots, N$, or equivalently in matrix form

$$\boldsymbol{E} \geq 0,$$

where the nonnegativity is element-wise.

Statistical techniques to solve the unmixing problem are often based on the minimization of the residual error, in terms of the quadratic loss function using the conventional Euclidean norm $\|\cdot\|$, with

$$\|\boldsymbol{x}_t - \boldsymbol{E}\boldsymbol{a}_t\|^2$$

for each of the observed spectra, subject to the nonnegativity of the endmembers, to the ANC (1.3) and possibly to the ASC (1.4). The above optimization problem can be written in the following matrix form

$$\arg \min_{\boldsymbol{A},\boldsymbol{E}\geq 0} \|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}\|_F^2.$$

This is the classical NMF, where the conventional algorithms alternate the optimization over each matrix while keeping the other one fixed, which is also called a two

block-coordinate descent scheme. In a supervised unmixing strategy, namely when endmembers are already known or extracted, the abundances are estimated by solving the following constrained optimization problem: $\arg\min_{\boldsymbol{a}_t} \|\boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\,\boldsymbol{e}_n\|^2$, subject to $\sum_{n=1}^{N} a_{nt} = 1$ and $a_{nt} \geq 0$, for all n and t. This is the so-called fully-constrained least-squares (FCLS). The FCLS algorithm proposed in [Heinz and Chang, 2001] yields the optimal abundances in the least-squares sense.

Besides the pixel-wise and the matrix-wise expressions, respectively in (1.1) and (1.2), it is worth noting that the LMM can be formulated using two other expressions. In an element-wise formulation, the l-th spectral band of the t-th pixel/spectrum can be decomposed as follows

$$x_{lt} = \boldsymbol{e}_{l*}\,\boldsymbol{a}_t + n_{lt}, \tag{1.5}$$

where $\boldsymbol{e}_{l*}$ is the row vector of the l-th spectral band over all the endmembers. This formulation is investigated in Section 1.4.3.2. There exists another expression for the LMM, by considering all the image pixels at each spectral band. In this case, the LMM takes the form

$$\boldsymbol{x}_{l*} = \boldsymbol{e}_{l*}\,\boldsymbol{A} + \boldsymbol{n}_{l*},$$

for the l-th spectral band, where $\boldsymbol{x}_{l*}$ is the l-th row of the data matrix $\boldsymbol{X}$ representing the l-th band of all pixels. By considering all the pixels at each spectral band, this yields the following least-squares optimization problem

$$\min_{\boldsymbol{A}} \sum_{l=1}^{L} \|\boldsymbol{x}_{l*} - \boldsymbol{e}_{l*}\boldsymbol{A}\|^2.$$

This formulation will be considered and revisited in Chapter 6 in order to derive unmixing algorithms that are robust to outlier bands.

The LMM is the most investigated model over the past decades. Besides its simplicity, the underlying light scattering mechanism approximated by LMM is often acceptable [Bioucas-Dias et al., 2012]. Under the linear assumption, various algorithms have been developed in the literature. For example, N-Findr [Winter, 1999] and VCA [Nascimento and Bioucas-Dias, 2005] for endmember extraction; FCLS [Heinz and Chang, 2001] and SUnSAL [Bioucas-Dias and Figueiredo, 2010] for abundance estimation, and the NMF-based techniques [Jia and Qian, 2009; Lu et al., 2013; Huck et al., 2010] for joint estimation. Recent works have also included the sparsity of the abundance vectors into

LMM [Bioucas-Dias and Figueiredo, 2010; Iordache et al., 2011, 2012]. In this case, each spectrum is fitted by a sparse linear mixture of endmembers, namely only the abundances with respect to a small number of endmembers are nonzero [Bioucas-Dias and Figueiredo, 2010; Iordache et al., 2014]. We refer to [Keshava and Mustard, 2002; Bioucas-Dias et al., 2012] for overviews of unmixing algorithms proposed for LMM.

1.4 Nonlinear Mixing Models

The linear mixing model is the most prevalent due to its simplicity for both the physical interpretation and algorithms design. As explained above, the linear assumption is valid only when the mixing happens in the macroscopic scale, namely when each photon interacts with only one endmember before reaching the remote sensors. However, serious nonlinear effects may exist in hyperspectral images, where the LMM becomes inappropriate [Dobigeon et al., 2014]. In such cases, more accurate and complex models are required to address the nonlinearities. The main categories of nonlinear models include the bilinear mixing models, the intimate mixing models, and the recent kernel-based ones.

1.4.1 Augmenting the linear model with a bilinear model

The bilinear mixing occurs when the photon scattered by a certain endmember reflects off other endmembers before arriving to the sensor [Dobigeon et al., 2014]. This phenomenon is obvious in particular for the hyperspectral images containing a forested field, where the interactions between the soil and the canopy happen as illustrated in Figure 1.3. By adding the second-order interactions between endmembers to the linear model, most bilinear models can be analytically expressed as

$$\boldsymbol{x}_t = \boldsymbol{E}\boldsymbol{a}_t + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \beta_{ij,t} \boldsymbol{e}_i \odot \boldsymbol{e}_j + \boldsymbol{n}_t,$$

for $t = 1, \ldots, T$, where $\odot$ represents the element-wise product (*i.e.*, Hadamard product). Various models characterize differently the interaction coefficients between endmembers, namely $\beta_{ij,t}$, as described in the following.

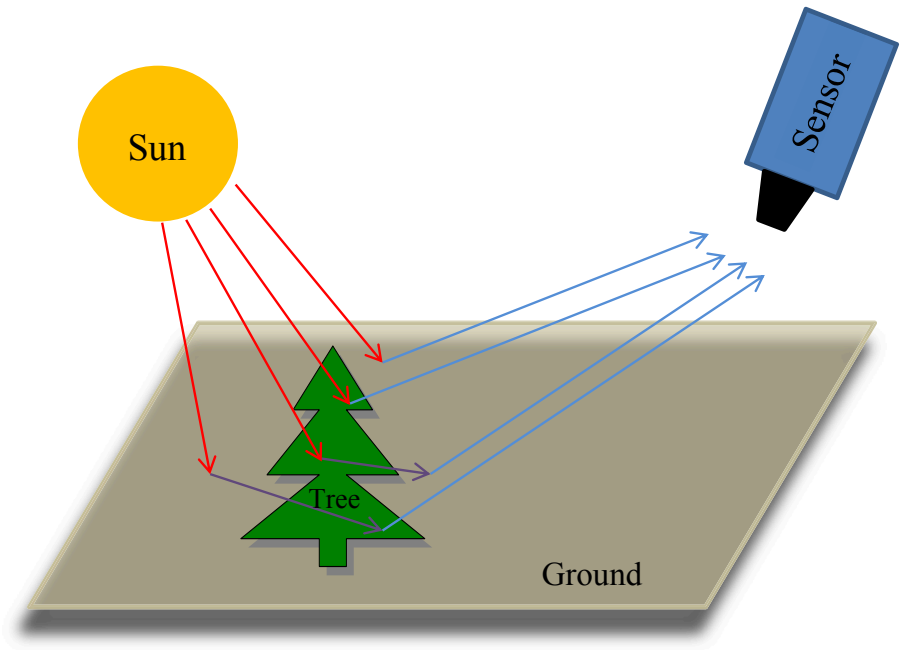


Figure 1.3: In a bilinear mixing mechanism, second-order interactions between endmembers augment the conventional linear mixture.

The Nascimento model (NM) proposed in [Nascimento and Bioucas-Dias, 2009] considers

$$\sum_{i=1}^{N} a_{nt} + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \beta_{ij,t} = 1,$$

where $a_{nt} \geq 0$ and $\beta_{ij,t} \geq 0$, for all t, n and $i \neq j$. It is obvious that the NM can be viewed as LMM with $\frac{N(N-1)}{2}$ additional endmembers composed by $\boldsymbol{e}_i \odot \boldsymbol{e}_j$, thus most abundance estimation (supervised unmixing) techniques that have been developed for LMM remain feasible with NM.

The Fan model (FM) proposed in [Fan et al., 2009] assumes that $\beta_{ij,t} = a_{it}a_{jt}$ $(i \neq j)$, resulting

$$\boldsymbol{x}_t = \boldsymbol{E}\boldsymbol{a}_t + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} a_{it}a_{jt}\boldsymbol{e}_i \odot \boldsymbol{e}_j + \boldsymbol{n}_t,$$

subject to both ANC and ASC on abundances. In FM, the nonlinear effect between endmembers is proportional to their abundance amplitudes. In particular, if $a_{it} = 0$, namely the i-th endmember does not contribute to the t-th pixel, the bilinear effects

between this endmember and other endmembers vanish. A major drawback of FM is that it is not a generalized model that includes LMM as a special case.

To overcome the limitations of FM, a generalized bilinear model (GBM) is proposed in [Halimi et al., 2011a,b], given by

$$\boldsymbol{x}_t = \boldsymbol{E}\boldsymbol{a}_t + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \gamma_{ij,t} a_{it} a_{jt} (\boldsymbol{e}_i \odot \boldsymbol{e}_j) + \boldsymbol{n}_t, \tag{1.6}$$

where $0 \leq \gamma_{ij,t} \leq 1$ controls the interactions between endmembers $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ at the t-th pixel. Assuming the endmembers are known, Bayesian algorithms are developed in [Halimi et al., 2011a,b] to estimate the abundances with GBM, considering both ANC and ASC. Appropriate prior distributions are chosen for the parameters under estimation, then the joint posterior distributions are derived. The unknown parameters are then estimated by a Markov chain Monte Carlo algorithm [Halimi et al., 2011a] or by a gradient descent algorithm [Halimi et al., 2011b], the algorithms of the latter are referred as BayGBM. In [Yokoya et al., 2014, 2012], the abundances with GBM are estimated by a semi-nonnegative matrix factorization (semi-NMF).

The polynomial post-nonlinear mixing model (PPNMM) proposed in [Altmann et al., 2012] assumes that the pixel reflectances are nonlinear functions of endmembers using

$$\boldsymbol{x}_t = \boldsymbol{E}\boldsymbol{a}_t + b_t(\boldsymbol{E}\boldsymbol{a}_t) \odot (\boldsymbol{E}\boldsymbol{a}_t) + \boldsymbol{n}_t, \tag{1.7}$$

where the nonlinear terms are characterized by the nonnegative parameter $b_t \in \mathbb{R}$, and both ANC and ASC are imposed. One good property of this model is that it boils down to LMM when $b_t = 0$. Compared with GBM, the PPNMM defines the nonlinear terms for each pixel with a single parameter b_t instead of a set of parameters $\gamma_{ij,t}$. This formulation has a less complex model than GBM which is easier for computation. In the aforementioned paper, Bayesian-based algorithms are developed to estimate the unknown abundances, referred as BayPPNMM.

1.4.2 Intimate mixing model

The intimate mixture, also termed microscopic mixture, happens when particles of different endmembers are closely adjacent within a pixel. In this case, light will interact with

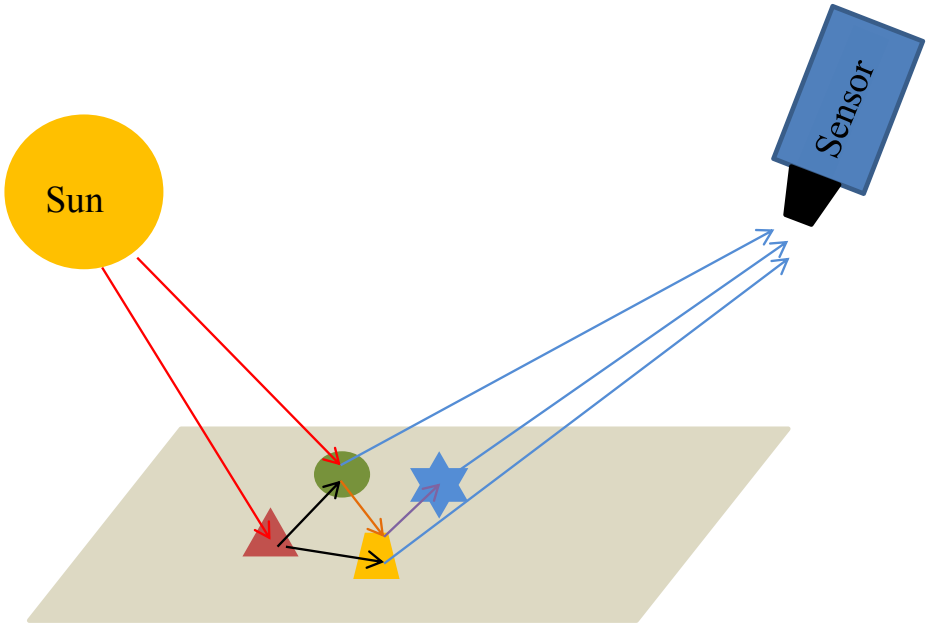


Figure 1.4: With an intimate mixing model, an observed spectrum is composed of a microscopic mixture of several endmembers.

the particles of several endmembers before reaching the sensor, as illustrated in Figure 1.4. The most well-known model to accurately characterize the intimate mixing mechanism is proposed by Hapke in [Hapke, 1981].

The average single-scattering albedo (SSA) represents the ratio of the scattered photons over the total photons influenced by the particle (scattered or absorbed). For a wavelength λ, it is defined by

$$w_\lambda = \sum_{n=1}^{N} f_n w_{n\lambda},$$

where $w_{n\lambda}$ are the material albedos and f_n the corresponding fractional proportions. Let $\boldsymbol{H}(c, w_\lambda)$ be the approximation to Chandrasekhar's function for isotropic scattering, defined as

$$\boldsymbol{H}(c, w_\lambda) = \frac{1 + 2c}{1 + 2c\sqrt{1 - w_\lambda}}.$$

The Hapke's bidirectional reflectance model is defined by

$$R_\lambda(w_\lambda) = \frac{w_\lambda}{4(c_i + c_e)} \boldsymbol{H}(c_i, w_\lambda)\boldsymbol{H}(c_e, w_\lambda), \tag{1.8}$$

where R_λ represents the reflectance at wavelength λ, and c_i and c_e denote the cosines of the angles of the incidence and emergence, respectively.

Recent works aim to combine the macroscopic mixture (LMM) and the microscopic mixture characterized by Hakpe's model [Close et al., 2012a]. In [Close et al., 2012b; Dranishnikov et al., 2013], the intimate mixture effect is taken as an additional endmember to LMM, yielding

$$\boldsymbol{x}_t = \sum_{n=1}^{N} a_{nt}\boldsymbol{e}_n + a_{N+1,t}\, R\left(\sum_{n=1}^{N} f_{nt}\boldsymbol{w}_n\right) + \boldsymbol{n}_t.$$

Here, $\boldsymbol{w}_n$ denotes the vector of SSA at all wavelengths and R is a vector function at all wavelengths, extended from the definition of R_λ in (1.8). While, $\boldsymbol{e}_n$ is the spectrum of the n-th "linear" endmember in the reflectance domain, $\boldsymbol{w}_n$ represents the spectrum of the n-th "nonlinear" endmember in the albedo domain. Their corresponding proportions at the t-th pixel are given by a_{nt} and f_{nt}, and satisfy respectively

$$\sum_{n=1}^{N+1} a_{nt} = 1, \quad \text{and} \quad a_{nt} \geq 0$$

and

$$\sum_{n=1}^{N} f_{nt} = 1, \quad \text{and} \quad f_{nt} \geq 0.$$

More recently, a macroscopic-microscopic mixing model (Mac-Mic) is proposed in [Close et al., 2014], where each pixel is assumed to be either macroscopically with LMM, or microscopically with the model $\boldsymbol{x}_t = R\big(\sum_{n=1}^{N} f_{nt}\boldsymbol{w}_n\big) + \boldsymbol{n}_t$, where f_{nt} represents the microscopic proportions. Applying a gradient-descent method, unsupervised algorithms have been developed to identify the endmembers and to estimate the mixture type and the abundance vector at each pixel.

1.4.3 Kernel-based models

Machine learning with kernel-based models allows to alleviate missing physical description of the underlying nonlinearity by defining a data-driven nonlinearity. This approach has been largely investigated in the context of the hyperspectral unmixing problem, as reviewed in the following [Broadwater et al., 2007; Chen et al., 2012, 2013b,c; Nguyen et al., 2013]. It is worth noting that most of these kernel machines operate in supervised unmixing, namely by estimating the abundances with some nonlinear model while assuming the endmembers were already identified (often with the linear model). The

methods presented in Chapters 3, 4, and 5 propose a joint estimation of the abundances and endmembers using data-driven kernel-based models.

Essentially, kernel machines operate by mapping the data (*e.g.,* the spectra in the hyperspectral image) to a so-called *feature space* $\mathcal{H}$, defined by the use of a kernel function κ as a measure of dissimilarity between data. A kernel function allows to evaluate the inner product between any pair of mapped data, without the need to explicit the nonlinear map function ϕ. Commonly-used kernels are the Gaussian kernel, the polynomial kernel and the linear kernel, with expressions given in Table 2.1. More details on kernel methods are provided in Chapter 2.

In the following, we outline the most known kernel-based unmixing methods.

1.4.3.1 Kernel fully constrained least squares

In [Broadwater et al., 2007], the kernel fully constrained least squares abundance method (KFCLS) is proposed to estimate the abundances. To this end, the optimization is performed in the feature space, by replacing the inner product with a kernel function in the FCLS. This method generalizes FCLS by estimating the abundances as follows

$$\min_{\boldsymbol{a}_t} \kappa(\boldsymbol{x}_t, \boldsymbol{x}_t) - 2\sum_{i=1}^{N} a_{it}\kappa(\boldsymbol{x}_t, \boldsymbol{e}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} a_{it}a_{jt}\kappa(\boldsymbol{e}_i, \boldsymbol{e}_j),$$

subject to both ANC and ASC. The resulting constrained optimization problem is addressed by the active set method.

In [Broadwater and Banerjee, 2009], KFCLS is further developed by employing the following physically inspired kernel

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i, \gamma)^\top \phi(\boldsymbol{x}_j, \gamma),$$

where ϕ is the nonlinear function that maps a reflectance spectrum to its SSA measurement (see Section 1.4.2) and γ represents the angle of incidence. This kernel is able to mimic the physics of the microscopic mixing mechanism, as long as the parameters are appropriately chosen. When tested on the RELAB[2] data accounting for intimate

[2]RELAB data were acquired at the NASA Reflectance Experiment Laboratory, at Brown University.

mixtures, KFCLS using the proposed kernel yields the best unmixing results, compared with the conventional Gaussian, polynomial and linear kernels.

While the above kernelized variants of FCLS operate by mapping the observed spectra with a nonlinear function, they fail to capture the interactions between the endmembers and are often criticized to have less physical interest in the context of unmixing hyperspectral images [Dobigeon et al., 2014; Chen et al., 2013b].

1.4.3.2 Nonlinear unmixing operating in a feature space

In order to characterize nonlinear interactions between the endmembers, one may define a general model of the form

$$x_{lt} = \psi(\boldsymbol{e}_{l*}) + n_{lt},$$

where ψ is a nonlinear function to be defined. While such nonlinear model is a generalization of the LMM given in (1.5), it cannot be explored as a nonlinear mixing model, since neither it does reveal the abundances, nor it allows to force constraints on them, namely the ANC and ASC. To overcome these difficulties, Chen, Richard and Honeine proposed a nonlinear model that combines a linear mixture and a nonlinear fluctuation defined in a kernel-induced feature space, as described in the following. It is worth noting that this approach follows the same idea of bilinear models, which augment the linear mixture with a nonlinear one, and extends it to other nonlinear mixing mechanisms.

The K-Hype model/algorithm introduced in [Chen et al., 2013b] explores a linear mixture with an additive nonlinear fluctuation for abundance estimation, where the nonlinearity depends exclusively on the endmembers. To be more precise, the K-Hype defines the function ψ as a partially linear function of abundance vector, combined with a nonlinear fluctuation term, with

$$x_{lt} = \boldsymbol{e}_{l*}\boldsymbol{a}_t + \psi_{\text{nlin}}(\boldsymbol{e}_{l*}),$$

where both ANC and ASC could be imposed, and ψ_{nlin} is a real-valued function belonging to a kernel-induced feature space $\mathcal{H}_{\text{nlin}}$. When a polynomial kernel is applied, the nonlinearity term is characterized by interactions between endmembers of the form $\boldsymbol{e}_1^{k_1} \odot \boldsymbol{e}_2^{k_2} ... \odot \boldsymbol{e}_N^{k_N}$, where $\odot$ is the Hadamard product.

In [Chen et al., 2012], the above additive fluctuation is relaxed by considering a convex combination between the linear model and the nonlinear one, namely by minimizing the following regularized cost function

$$\Big(\frac{1}{u}\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}}\Big) + \frac{1}{\mu}\sum_{l=1}^{L}\big(x_{lt} - \psi(\boldsymbol{e}_{l*})\big)^2,$$

where $\psi(\boldsymbol{e}_{l*}) = \psi_{\text{lin}}(\boldsymbol{e}_{l*}) + \psi_{\text{nlin}}(\boldsymbol{e}_{l*})$ and $\psi_{\text{lin}}(\boldsymbol{e}_{l*}) = \boldsymbol{e}_{l*}\boldsymbol{a}_t$ with $\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} = \|\boldsymbol{a}_t\|^2$. The parameter μ balances the regularization term and the fitting term, and u controls the tradeoff between the linear model $\psi_{\text{lin}}(\boldsymbol{e}_{l*}) = \boldsymbol{e}_{l*}\boldsymbol{a}_t$ and the nonlinear function ψ_{nlin} defined in a kernel-induced feature space $\mathcal{H}_{\text{nlin}}$. This optimization problem is solved in the same spirit as multiple kernel learning.

This combination of the linear model and an additive nonlinearity defined with kernels has been extended to other formulations. In [Chen et al., 2013c], the abundances are incorporated in the nonlinear model with a post-nonlinear model of the form $\psi(\boldsymbol{E}\boldsymbol{a}_t)$, and a Bayesian approach is used in [Altmann et al., 2014]. Based on the aforementioned framework of K-Hype, the spatial information is taken into consideration in [Chen et al., 2014] by introducing an ℓ_1-norm spatial regularization.

1.4.3.3 Nonlinear unmixing by solving the preimage problem

One may also consider the nonlinearity on the abundances, where each spectrum $\boldsymbol{x}_t$ is obtained as a nonlinear function of the corresponding abundances, namely $\phi(\boldsymbol{a}_t)$ for some nonlinear mapping function ϕ. The unmixing problem consists in estimating the inverse map, namely recovering the abundance $\boldsymbol{a}_t$ from any observed spectrum $\boldsymbol{x}_t$. In order to get the inverse map, one needs a set of available pairs of data $\{(\boldsymbol{a}_1, \boldsymbol{x}_1), ..., (\boldsymbol{a}_T, \boldsymbol{x}_T)\}$.

This approach of learning the inverse map is proposed in [Nguyen et al., 2013], and operates by solving the so-called preimage problem. Section 2.4 is devoted to describe this hard problem in kernel-based methods, and provides solutions to tackle this problem. Basically, the inverse of the mapping function ϕ, such that $\phi\colon \boldsymbol{a}_t \mapsto \boldsymbol{x}_t$, can be viewed as a dimensionality reduction operation from the spectral space to the lower-dimension space of abundances. The conformal map approach, initially introduced in [Honeine and Richard, 2011] and described in Section 2.4.3.4, was successfully applied in [Nguyen et al., 2013] to solve the unmixing problem.

The idea of mapping the abundance vectors to the space of observed spectra is also considered in [Altmann et al., 2013], where a kernel-based method for nonlinear unmixing is investigated, with a particular interest to the bilinear mixing model. To this end, a Gaussian process latent variable model is used to establish a smooth mapping from the space of abundance vectors to the space of spectra, by preserving the dissimilarities in both spaces.

1.5 Main Contributions

This thesis brings several original contributions, essentially to nonlinear NMF with kernels, and focuses on the task of nonlinear hyperspectral unmixing. First, we introduce a novel kernel-based model for nonlinear NMF that estimates jointly the endmembers and abundances (as opposed to other techniques which suffer from the preimage problem; see Section 3.2.2). Second, we extend the proposed kernel-based nonlinear NMF to an online fashion, in order to tackle large volumes of data, such as with streaming data. We also revisit the NMF as a bi-objective problem which combines the linear and kernel-based NMF models. Finally, independent from the kernel-based NMF, we propose a supervised spectral unmixing approach that is robust to outlier bands, by writing the unmixing problem as the maximization of the correntropy criterion. The main contributions are outlined next.

1.5.1 Structure of the manuscript

The rest of the manuscript is organized as follows:

The second chapter reviews kernel-based methods in machine learning, and presents the preimage problem as well as the most known techniques to solve it.

The third chapter introduces a novel kernel-based model for the NMF (referred to "**KNMF**" throughout this manuscript) that does not suffer from the preimage problem, by investigating the estimation of the factorization matrices directly in the input space. For different kernel functions, we describe two schemes for iterative algorithms: an additive update rule based on a gradient descent scheme and a multiplicative update rule. Within the proposed framework, we develop several extensions to incorporate

constraints, including sparseness, smoothness, and spatial regularization with a total-variation-like penalty. The effectiveness of the proposed method is demonstrated on the problem of unmixing hyperspectral images using well-known real images.

The fourth chapter is dedicated to extend the proposed nonlinear KNMF to an online fashion, which is necessary when dealing with streaming data. By exploring recent advances in the stochastic gradient descent and the mini-batch strategies, the proposed methods have a fixed – tractable – complexity independent of the increasing number of samples. We derive several general update rules, in both additive and multiplicative strategies, and present the case of the Gaussian kernel in detail. The performance of the proposed framework is validated on unmixing synthetic and real hyperspectral images.

The fifth chapter revisits the NMF as a multi-objective problem, in particular a bi-objective one, where the objective functions defined in both input and feature spaces are taken into account. By taking the advantage of the sum-weighted method from the literature of multi-objective optimization, the proposed bi-objective KNMF determines a set of nondominated, Pareto optimal, solutions. Moreover, the corresponding Pareto front is approximated and studied. Experimental results on unmixing synthetic and real hyperspectral images confirm the efficiency of the proposed bi-objective KNMF.

The sixth chapter develops a robust spectral unmixing approach for hyperspectral images, in the context of supervised learning, *i.e.*, the endmembers are known in prior. The robustness is achieved by writing the unmixing problem as the maximization of the correntropy criterion subject to the most commonly used constraints. Two unmixing problems are investigated: the first problem considers the fully-constrained unmixing, with both the nonnegativity and sum-to-one constraints, while the second one deals with the nonnegativity and the sparsity-promoting of the abundances. The corresponding optimization problems are solved efficiently using an alternating direction method of multipliers (ADMM) approach. Experiments on synthetic and real hyperspectral images validate the performance of the proposed algorithms for different scenarios, demonstrating that the correntropy-based unmixing is robust to outlier bands.

Finally, chapter seven concludes the thesis. The limitations of the work are discussed, as well as several perspectives for future work.

1.5.2 Publications

The research work of this thesis resulted in the following publications.

Peer-reviewed international journals

1. **F. Zhu**, P. Honeine. "Bi-objective nonnegative matrix factorization: Linear Versus Kernel-Based Models". *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 7, pp. 4012-4022, July 2016.

2. **F. Zhu**, P. Honeine. "Online kernel nonnegative matrix factorization". *Signal Processing*, vol. 131, pp. 143-153, February 2017.

3. **F. Zhu**, A. Halimi, P. Honeine, B. Chen, N. Zheng. "Correntropy maximization via ADMM - application to robust hyperspectral unmixing". *IEEE Transactions on Geoscience and Remote Sensing*, 11 pages, in revision.

Peer-reviewed international conferences with proceedings

1. **F. Zhu**, P. Honeine, K. Maya. "Kernel non-negative matrix factorization without the pre-image problem". *In Proc. of the 24th IEEE workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Reims, France, 21–24 September 2014.

2. **F. Zhu**, P. Honeine. "Pareto front of bi-objective kernel-based nonnegative matrix factorization". *In Proc. of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 585–590, Bruges, Belgium, 22–24 April 2015.

3. **F. Zhu**, P. Honeine. "Online nonnegative matrix factorization based on kernel machines". *In Proc. of the 23rd European Conference on Signal Processing (EUSIPCO)*, pages 2381–2385, Nice, France, 31 August–4 September, 2015.

4. **F. Zhu**, A. Halimi, P. Honeine, B. Chen, N. Zheng. "ADMM for Maximum Correntropy Criterion". *In Proc. of the 28th INNS and IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Vancouver, Canada, 24–29 July, 2016.

Peer-reviewed national conference with proceedings

1. P. Honeine, **F. Zhu**. "Eviter la malédiction de pré-image : application à la factorisation en matrices non négatives à noyaux". *Dans les actes du 25-ème colloque du Groupe de Recherche et d'Etudes du Traitement du Signal et des Images (GRETSI)*, France, 8–11 September 2015.

Chapter 2

Kernel Methods in Machine Learning and the Preimage Problem

Contents

Kernel methods have received considerable popularity in machine learning due to their ability to extend the linear techniques to nonlinear ones. To this end, kernel methods consist in mapping the samples from the input space into a high dimensional feature space, implicitly defined by using a specified kernel, without explicit knowledge neither on the mapping function nor on the feature space. However, several pattern recognition tasks require the reverse mapping, from the feature space back to the input space, which is an ill-posed problem. This is the curse of the preimage problem, a major drawback inherited from kernel methods. In this chapter, we first introduce the framework of kernel methods in machine learning, with the Moore-Aronszajn theorem that connects positive definite kernels to reproducing kernels and their associated Hilbert spaces. We describe two key properties which are the kernel trick and the represented theorem. The last part of this chapter is mainly devoted to the preimage problem in kernel methods, from its formulation to the techniques proposed to solve this problem.

2.1 Introduction to Machine Learning

Within the scarcity of information on a studied system, the extraction of relevant knowledge from available data has become a key challenge for engineers and researchers in many fields. Pattern recognition, data mining, classification and regression, all fall under the aegis of machine learning, which explores many disciplines of science and mathematics, including statistics, computer science, engineering, and optimization theory [Ghahramani, 2004]. The core objective of machine learning is to infer, from a set of available samples, a function that best describes the relationship within them, and thus the underlying mechanism of the studied system. Roughly, machine learning is divided as supervised learning and unsupervised learning [Burges, 1998; Mitchell, 2006].

The supervised learning, often in classification, regression and reinforcement learning, consists in estimating the correct output of a system given some input. Let $\mathcal{X} \in \mathbb{R}^L$ be the input space and $\mathcal{Y}$ the output space, *e.g.*, $\mathcal{Y} = \{-1, +1\}$ for labels in a binary classification task. The goal is to learn a general rule (function) ψ in order to predict the output $y \in \mathcal{Y}$ of a given new input sample $\boldsymbol{x}$, often given in a pairwise notation $(\boldsymbol{x}, y)$. By considering some loss function $\mathcal{L}$ that measures the error between the correct output y and the estimated one $\psi(\boldsymbol{x})$ provided by the learning machine, the expected

risk is minimized as following

$$\arg\min_{\psi} \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}(\psi(\boldsymbol{x}), y)\, P(\boldsymbol{x}, y)\, d\boldsymbol{x}\, dy, \tag{2.1}$$

where $P(\boldsymbol{x}, y)$ denotes the probability distribution of the pair $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$. Since this distribution is unknown in general, on needs to infer the function from a training set of samples, with a finite number of independent and identically distributed samples, namely N pairs of samples $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, ..., N$. Therefore, the optimal function ψ^* is determined by the minimisation of the empirical risk, namely

$$\arg\min_{\psi} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\psi(\boldsymbol{x}_i), y_i), \tag{2.2}$$

instead of working on the optimization problem (2.1).

The second class is the unsupervised machine learning, where only observations from the input space are available. The task becomes to build relevant representations, by finding patterns in the available data, namely by inferring a function that describes the hidden structure from unlabeled data. For some arbitrary loss function $\mathcal{L}$, the minimization of the expected risk takes the form

$$\arg\min_{\psi} \int_{\mathcal{X}} \mathcal{L}(\psi(\boldsymbol{x}))\, P(\boldsymbol{x})\, d\boldsymbol{x},$$

where $P(\boldsymbol{x})$, the probability distribution for $\boldsymbol{x} \in \mathcal{X}$, is not available in general. Given a finite number of training samples $\boldsymbol{x}_i$, $i = 1, ..., N$, the optimal function is estimated by minimizing the empirical risk as follows:

$$\arg\min_{\psi} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\psi(\boldsymbol{x}_i)).$$

Classical examples of unsupervised learning are clustering, dimensionality reduction, and blind source separation. The latter includes techniques such as principal component analysis, independent component analysis, and nonnegative matrix factorization.

Both supervised and unsupervised learning are ill-posed problems, since there exists an infinite number of functions that nullify the empirical risk function. However, the minimization of the empirical risk does not always correspond to the minimization of the expected risk, namely the gap between both risks could be potentially large. This

is the so-called over-fitting problem. To overcome this problem, one should perform the minimization of the empirical risk by restricting the search of the optimal solution to an hypothesis space of regular functions. This is the regularization approach firstly proposed in [Tikhonov and Arsenin, 1977]. In the succeeding section, we introduce a particular case of the space of regular functions, namely the reproducing kernel Hilbert space [Aronszajn, 1950].

2.2 Reproducing Kernels and Associated Hilbert Spaces

Kernel methods are a class of machine learning that includes nonlinear techniques, such as support vector machine (SVM) for classification and regression [Vapnik, 1995; Burges, 1998], as well as conventional linear techniques, such as PCA. Before presenting kernel methods, we first introduce in this section the concepts of reproducing kernels and reproducing kernel Hilbert spaces, and the Moore-Aronszajn theorem which connect them to positive definite kernels.

Functional analysis notations

Before proceeding, we need to introduce several important notations from functional analysis, following the lecture notes [Lorenzo and Durrett, 2010]. Let $\mathcal{F}$ be a function space, namely the space of real-valued functions defined from an input space $\mathcal{X} \subset \mathbb{R}^L$, *i.e.,* $\mathcal{F}$ is the space of functions $f : \mathcal{X} \to \mathbb{R}$.

Definition 2.1 (Inner product)**.** Consider a function $\langle \cdot, \cdot \rangle : \mathcal{F} \times \mathcal{F} \longrightarrow \mathbb{R}$, that assigns each ordered pair $(\boldsymbol{u}, \boldsymbol{v})$ a scalar $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$. It is said to be an inner product if the following conditions are satisfied:

1. $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$, for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{F}$ (Symmetry)
2. $\langle \alpha \boldsymbol{u} + \beta \boldsymbol{v}, \boldsymbol{w} \rangle = \alpha \langle \boldsymbol{u}, \boldsymbol{w} \rangle + \beta \langle \boldsymbol{v}, \boldsymbol{w} \rangle$, for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{F}$, and $\alpha, \beta \in \mathbb{R}$ (Bilinearity)
3. $\langle \boldsymbol{u}, \boldsymbol{u} \rangle \geq 0$, for all $\boldsymbol{u} \in \mathcal{F}$, with $\langle \boldsymbol{u}, \boldsymbol{u} \rangle = 0 \Longleftrightarrow \boldsymbol{u} = 0$ (Positive definiteness)

Definition 2.2 (Norm)**.** A norm is a nonnegative function $\| \cdot \| : \mathcal{F} \longrightarrow \mathbb{R}$ satisfying the following conditions:

1. $\|f\| \geq 0$ and $\|f\| = 0 \Longleftrightarrow f = 0$, for all $f \in \mathcal{F}$
2. $\|f + g\| \leq \|f\| + \|g\|$, for all $f, g \in \mathcal{F}$
3. $\|\alpha f\| = |\alpha| \cdot \|f\|$, for all $f \in \mathcal{F}$, for all $\alpha \in \mathbb{R}$

A norm can be defined from a given inner product, as follows: $\|f\| = \sqrt{\langle f, f \rangle}$.

Definition 2.3 (Inner product space)**.** An inner product space is a vector space with an inner product structure.

Since an inner product naturally defines an associated norm, an inner product space is also a normed vector space.

Definition 2.4 (Hilbert space)**.** A Hilbert space $\mathcal{H}$ is an inner product space that is complete (every Cauchy sequence converges in $\mathcal{H}$) and separable (contains a countable dense subset) endowed with an inner product. Let $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{H}}$ be this inner product, for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{H}$, then the norm in $\mathcal{H}$ is defined with $\|\boldsymbol{u}\|_{\mathcal{H}} = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\mathcal{H}}}$.

2.2.1 Positive definite kernel, reproducing kernel and RKHS

Definition 2.5 (Positive definite kernel [Aronszajn, 1950])**.** Let κ be a symmetric similarity function from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$, *i.e.*, satisfying

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i).$$

It is said to be a positive definite kernel on $\mathcal{X}$ if, and only if,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0,$$

for every $N \in \mathbb{N}$, $\boldsymbol{x}_i \in \mathcal{X}$, $c_i \in \mathbb{R}$ and $i = 1, \ldots, N$.

Definition 2.6 (Gram matrix)**.** Given a positive definite kernel κ and N observations $\boldsymbol{x}_i \in \mathcal{X}$, $i = 1, ..., N$, the $N \times N$ Gram matrix $\boldsymbol{K}$ associated with these observations is defined by entries

$$\boldsymbol{K}_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

for $i, j = 1, ..., N$.

By definition, one can easily prove that the Gram matrix is positive definite.

Now we define the reproducing kernel and reproducing kernel Hilbert space (RKHS) using the aforementioned notations. Let $\mathcal{H}$ be a Hilbert space of functions defined on a set $\mathcal{X}$. Denote by $\langle f, g\rangle_{\mathcal{H}}$ the inner product and $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$ the norm in $\mathcal{H}$, for any $f, g \in \mathcal{H}$.

Definition 2.7 (Reproducing kernel)**.** A function $\kappa : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is called a reproducing kernel of the Hilbert space $\mathcal{H}$, if the following conditions are satisfied:

1. For all $\boldsymbol{x}_i \in \mathcal{X}$, the function $\kappa(\boldsymbol{x}_i, \cdot)$ belongs to $\mathcal{H}$,
2. For all $f \in \mathcal{H}$ and $\boldsymbol{x}_i \in \mathcal{X}$, $f(\boldsymbol{x}_i) = \langle f(\cdot), \kappa(\boldsymbol{x}_i, \cdot)\rangle_{\mathcal{H}}$,

where the second propriety is often referred to the *reproducing property* in [Aronszajn, 1950].

Definition 2.8 (Reproducing kernel Hilbert space (RKHS))**.** A Hilbert space $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), if there exists a reproducing kernel whose span is dense in $\mathcal{H}$.

Next, we demonstrate the so-called *kernel trick* from the property of reproducing kernel and RKHS.

Corollary 2.1 (Kernel trick)**.** *Any evaluation of the reproducing kernel κ of a Hilbert space $\mathcal{H}$ can be expressed as an inner product in $\mathcal{H}$, namely*

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \kappa(\boldsymbol{x}_i, \cdot), \kappa(\boldsymbol{x}_j, \cdot)\rangle_{\mathcal{H}}. \tag{2.3}$$

Proof. Replacing $f(\cdot)$ in Point 2. of Definition 2.7 with the function $\kappa(\boldsymbol{x}_j, \cdot)$, which belongs to $\mathcal{H}$ due to Point 1. of Definition 2.7, leads to

$$\kappa(\boldsymbol{x}_j, \boldsymbol{x}_i) = \langle \kappa(\boldsymbol{x}_j, \cdot), \kappa(\boldsymbol{x}_i, \cdot)\rangle_{\mathcal{H}}.$$

□

According to the kernel trick, the associated norm of the RKHS is expressed as

$$\|\kappa(\boldsymbol{x}_i, \cdot)\|_{\mathcal{H}} = \sqrt{\langle \kappa(\boldsymbol{x}_i, \cdot), \kappa(\boldsymbol{x}_i, \cdot)\rangle_{\mathcal{H}}} = \sqrt{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i)}, \quad \text{for all } \boldsymbol{x}_i \in \mathcal{X},$$

from Definition 2.4. The kernel trick (2.3) is a primer property that has been applied in almost every kernel-based methods, as described in Section 2.3.1.

2.2.2 The Moore-Aronszajn Theorem

The Moore-Aronszajn Theorem [Aronszajn, 1950] provides a one-to-one correspondence between positive definite kernels and reproducing kernels.

Theorem 2.9 (Moore-Aronszajn Theorem)**.** *For every positive definite kernel κ on a set $\mathcal{X}$, there is a unique RKHS of functions on $\mathcal{X}$ for which κ is a reproducing kernel. Conversely, every reproducing kernel is a positive definite kernel.*

Proof. We give a sketch of proof of this theorem. First, given a reproducing kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \kappa(\boldsymbol{x}_i, \cdot), \kappa(\boldsymbol{x}_j, \cdot) \rangle_{\mathcal{H}}$ of the RKHS $\mathcal{H}$, one can show that it is a positive definite kernel. The symmetry property is straightforward from the symmetry of the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. The positive definiteness is proved as follows:

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \langle \kappa(\boldsymbol{x}_i, \cdot), \kappa(\boldsymbol{x}_j, \cdot) \rangle_{\mathcal{H}} \\
&= \Big\langle \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}_i, \cdot), \sum_{j=1}^{N} \alpha_j \kappa(\boldsymbol{x}_j, \cdot) \Big\rangle_{\mathcal{H}} \\
&= \Big\| \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}_i, \cdot) \Big\|_{\mathcal{H}}^2 \\
&\geq 0.
\end{aligned}$$

Second, given a positive definite kernel κ, one can construct the corresponding RKHS. To this end, let $\mathcal{H}$ be the Hilbert functional space defined as a completion of the pre-Hilbert space spanned by the set of functions

$$\left\{ f : f = \sum_{i=1}^{N} \alpha_i \, \kappa(\boldsymbol{x}_i, \cdot), \quad \boldsymbol{x}_i \in \mathcal{X}, \alpha_i \in \mathbb{R} \right\}, \tag{2.4}$$

where $\boldsymbol{x}_i \in \mathcal{X}$, $i = 1, \ldots, N$ is a set of available samples.

Given two functions of $\mathcal{H}$, namely

$$f = \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}_i, \cdot) \quad \text{and} \quad g = \sum_{j=1}^{N} \beta_j \kappa(\boldsymbol{x}_j, \cdot),$$

one can endow $\mathcal{H}$ with the inner product defined as

$$\begin{aligned}\langle f, g \rangle_{\mathcal{H}} &= \Big\langle \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}_i, \cdot), \sum_{j=1}^{N} \beta_j \kappa(\boldsymbol{x}_j, \cdot) \Big\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \beta_j \langle \kappa(\boldsymbol{x}_i, \cdot), \kappa(\boldsymbol{x}_j, \cdot) \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \beta_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j),\end{aligned}$$

where the last equality is established using the kernel trick (2.3). One can check that the reproducing property carries over to the completion $\mathcal{H}$. As for the uniqueness of $\mathcal{H}$, it is easy to prove the isometric isomorphism to any other Hilbert space with the same reproducing kernel. □

2.2.3 Kernels: Construction and examples

According to the Moore-Aronszajn Theorem (Theorem 2.9), positive definite kernels are reproducing kernels, and vice versa. For notation simplicity, we shall refer to them as kernels in the following. In this subsection, we present the operations to construct new kernels from existing ones, and give some examples of the commonly used kernels [Shawe-Taylor and Cristianini, 2004].

Denote $\kappa_1, \kappa_2 \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ two (positive definite) kernels. The function $\kappa \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a valid kernel as well if, for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, it is defined in any of the following relations:

1. *linear combination*: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \beta_1 \kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + \beta_2 \kappa_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$, for $\beta_1, \beta_2 \in \mathbb{R}_+$.
2. *shifting*: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + \ell$, for $\ell \in \mathbb{R}_+$.
3. *product*: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_j) \kappa_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
4. *power*: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_j)^p$, for $p \in \mathbb{N}_+$.
5. *exponential*: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big(\kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)$.

Table 2.1: Some common kernels and their gradients w.r.t. the first argument.

	Name	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$	$\nabla_{\boldsymbol{x}_i}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$
Projective	Linear	$\boldsymbol{x}_i^\top \boldsymbol{x}_j$	$\boldsymbol{x}_j$
	Polynomial	$(\boldsymbol{x}_i^\top \boldsymbol{x}_j + \mathrm{c})^d$	$d\,(\boldsymbol{x}_i^\top \boldsymbol{x}_j + \mathrm{c})^{(d-1)}\boldsymbol{x}_j$
	Sigmoid	$\tanh(\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j + \mathrm{c})$	$\gamma \mathrm{sech}^2(\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j + \mathrm{c})\boldsymbol{x}_j$
RBF	Gaussian	$\exp(\frac{-1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$	$-\frac{1}{\sigma^2}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)$
	Laplacian	$\exp(\frac{-1}{\sigma}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$	$-\frac{1}{\sigma}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathrm{sgn}(\boldsymbol{x}_i - \boldsymbol{x}_j)$
	Rational quadratic	$1 - \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 + \sigma}$	$\frac{-2\sigma(\boldsymbol{x}_i - \boldsymbol{x}_j)}{(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 + \sigma)^2}$

6. *normalization*: $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{\kappa_1(\boldsymbol{x}_i, \boldsymbol{x}_i)\kappa_1(\boldsymbol{x}_j, \boldsymbol{x}_j)}}$.

It can be proven that the functions built from the above operations are valid kernels, simply by checking their positive definiteness.

Generally, kernels defined on vector spaces can be divided into two categories: projective kernels and radial basis function (RBF) kernels. The projective kernels measure the similarity between samples using an inner product, and are of the form

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = f(\boldsymbol{x}_i{}^\top \boldsymbol{x}_j). \tag{2.5}$$

The RBF kernels, using distances as measures of dissimilarity, take the form

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = g(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|). \tag{2.6}$$

Of particular interest is the Gaussian kernel with

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{-1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right),$$

where $\sigma \in \mathbb{R}^*$ is the tunable bandwidth parameter. Table 2.1 lists the most commonly used examples of projective and RBF kernels, with their expressions. The last column presents, for each kernel, the expression of its gradient with respect to (w.r.t.) its first argument, which will be required in the analysis given in this thesis.

2.3 Introduction to Kernel Methods

In this section, we describe the underlying mechanism of kernel methods to build a nonlinear model from the linear one, and explain the well-known representer theorem.

2.3.1 From linear to nonlinear models using kernels

Kernel methods provide a framework to capture the nonlinear patterns from the observed data by first mapping them into a potentially high dimensional feature space, and then applying a linear model on the transformed data [Shawe-Taylor and Cristianini, 2004]. A key property behind kernel methods is the kernel trick, which allows the formulation of nonlinear variants from linear algorithms, under the assumption that the latter can be expressed in terms of inner products only, involving pairs of observed data [Hein and Bousquet, 2004], as described in the following.

The Moore-Aronszajn Theorem (Theorem 2.9) states that any (positive definite) kernel κ on some input space $\mathcal{X}$ defines a RKHS $\mathcal{H}$ with an endowed inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \kappa(\boldsymbol{x}_i, \cdot), \kappa(\boldsymbol{x}_j, \cdot) \rangle_{\mathcal{H}}, \text{ for all } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}.$$

For any given $\boldsymbol{x} \in \mathcal{X}$, the element $\kappa(\boldsymbol{x}, \cdot)$ of $\mathcal{H}$ can be viewed as a transformation of $\boldsymbol{x}$, thus defining a mapping function Φ as follows:

$$\begin{aligned} \Phi \colon \mathcal{X} &\to \mathcal{H} \\ \boldsymbol{x} &\mapsto \Phi(\boldsymbol{x}) = \kappa(\boldsymbol{x}, \cdot). \end{aligned}$$

As a consequence, the Moore-Aronszajn Theorem (Theorem 2.9) states that the evaluation of any (positive definite) kernel κ at any pair of samples corresponds to the inner product of their images in some feature space $\mathcal{H}$, namely

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}},$$

for any $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$. Figure 2.1 represents the mapping from the input space $\mathcal{X}$ to the feature space $\mathcal{H}$, *i.e.,* the RKHS associated to the used kernel κ.

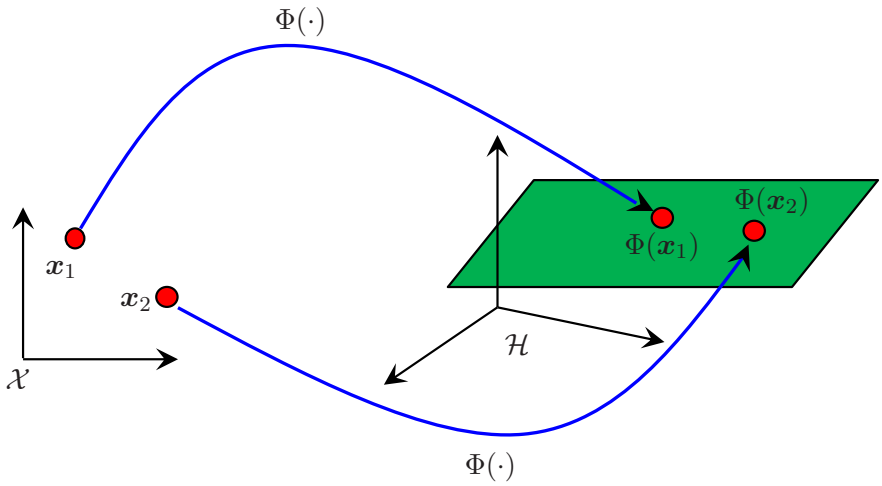


Figure 2.1: Illustration of the mapping Φ from the input space $\mathcal{X}$ to the feature space $\mathcal{H}$.

Most machine learning techniques can rely on these kernels to operate in the potentially high-dimension space, without expliciting the mapping Φ, neither computing the coordinates in that space, but rather by simply computing the inner products (via the kernel) between pairs of images of the data. This is the idea of the kernel trick. Its applicability can be easily illustrated when computing the quadratic distance between any pair of images, $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$, in the feature space, that can be evaluated using the kernel trick, with

$$\begin{aligned} \|\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)\|^2_{\mathcal{H}} &= \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_i) \rangle_{\mathcal{H}} + \langle \Phi(\boldsymbol{x}_j), \Phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}} - 2\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}} \\ &= \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) + \kappa(\boldsymbol{x}_j, \boldsymbol{x}_j) - 2\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j). \end{aligned}$$

As a consequence, any distance-based algorithm (*e.g.,* k-nearest neighbors) can be easily generalized to the feature space by evaluating the distances with kernels as above. Besides this simple illustration, it turns out that the kernel trick allows to provide non-linear functions or decision boundaries based on linear ones. To this end, it is often used in conjunction with the representer theorem.

2.3.2 The representer theorem

A wide range of machine learning problems seek the optimal function which best characterizes the relationship within training data, mainly by minimizing a regularized empirical risk function. Considering the aforementioned kernel-based framework, the optimization problem needs to be solved in the RKHS, a space of potentially infinite-dimension, *e.g.*, the RKHS associated with the Gaussian kernel. The power of the representer theorem is to provide an explicit form of the optimal solution, with a linear-in-the-parameter expression that transforms the optimization problem into the estimation of the optimal vector in a N-dimensional space, N being the number of available training data. This subsection presents the representer theorem, first proposed in [Kimeldorf and Wahba, 1971] and generalized in [Schölkopf et al., 2001] for the wide class of kernel methods.

Theorem 2.10 (The Representer Theorem)**.** *Consider a kernel κ defined on an input space $\mathcal{X}$ with its reproducing kernel Hilbert space $\mathcal{H}$, and a set of training samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathcal{X}$ with possibly their associated labels $y_1, \ldots, y_N$. For a loss function $\mathcal{L}$ and a non-decreasing function Ω on $\mathbb{R}_+$, the minimizer of a regularized empirical risk function of the form*

$$\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\psi(\boldsymbol{x}_i), y_i) + \lambda\Omega(\|\psi\|_{\mathcal{H}}^2) \tag{2.7}$$

admits the representation

$$\psi = \sum_{i=1}^{N}\alpha_i\,\Phi(\boldsymbol{x}_i), \tag{2.8}$$

where $\Phi(\boldsymbol{x}_i) = \kappa(\boldsymbol{x}_i, \cdot)$.

Proof. Let $\mathcal{H}_N$ be the subspace spanned by the functions $\{\kappa(\boldsymbol{x}_1, \cdot), \ \ldots, \ \kappa(\boldsymbol{x}_N, \cdot)\}$, namely for $\Phi(\boldsymbol{x}_i) = \kappa(\boldsymbol{x}_i, \cdot)$:

$$\mathcal{H}_N = \left\{ \psi \ : \ \psi = \sum_{i=1}^{N}\alpha_i\,\Phi(\boldsymbol{x}_i), \quad \boldsymbol{x}_i \in \mathcal{X}, \alpha_i \in \mathbb{R} \right\}.$$

Thus, every $\psi \in \mathcal{H}$ admits a unique decomposition of two parts, one belonging to $\mathcal{H}_N$ and the other belonging to its orthogonal space, *i.e.*,

$$\psi = \sum_{i=1}^{N}\alpha_i\,\Phi(\boldsymbol{x}_i) + \psi^{\perp}, \tag{2.9}$$

where the orthogonal part $\psi^{\perp}$ satisfies $\langle \psi^{\perp}, \Phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} = 0$, for $i = 1, \ldots, N$. Considering the evaluation of (2.9) at any $\boldsymbol{x}_j \in \mathcal{X}$, we have from the reproducing property

$$\begin{aligned}\langle \psi, \Phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}} &= \sum_{i=1}^{N} \alpha_i \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + \langle \psi^{\perp}, \Phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{N} \alpha_i \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j).\end{aligned}$$

It turns out that the evaluation of ψ at every observed data depends only on a set of coefficients α_i, $i = 1, \ldots, N$, thus the first term in the regularized empirical risk (2.7) is independent of $\psi^{\perp}$. Now, we consider the regularization term in (2.7). From the Pythagorean theorem, it can be expressed by

$$\Omega(\|\psi\|_{\mathcal{H}}^2) = \Omega\Big(\|\sum_{i=1}^{N} \alpha_i \, \Phi(\boldsymbol{x}_i)\|_{\mathcal{H}}^2 + \|\psi^{\perp}\|_{\mathcal{H}}^2\Big). \tag{2.10}$$

Since $\psi^{\perp}$ is orthogonal to $\sum_{i=1}^{N} \alpha_i \, \Phi(\boldsymbol{x}_i)$ and Ω is non-decreasing, reducing $\|\psi^{\perp}\|_{\mathcal{H}}^2$ will strictly reduce (2.10), while this operation has no influence on the first term of the regularized risk in (2.7). Consequently, we set $\psi^{\perp} = 0$ and obtain the optimal solution as given in (2.8). The representer theorem is then proven. □

The significance of this theorem lies in the existence of a unique solution to a regularized empirical risk; this solution can be expressed as a finite linear combination of the kernels centered on the training data. As a consequence, minimizing the aforementioned empirical risk boils down to an N-dimensional optimization problem, namely the estimation of the optimal coefficients $\alpha_1, \ldots, \alpha_N \in \mathbb{R}$. To illustrate this idea, we consider the least squares optimization problem with the Tikhonov regularization, namely with the quadratic loss $\mathcal{L}(\psi(\boldsymbol{x}_i), y_i) = (y_i - \psi(\boldsymbol{x}_i))^2$ and the identity as the regularization function Ω. By combining the resulting regularized risk with expression (2.8), we get the ridge regression problem $\min_{\boldsymbol{\alpha}} \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|^2 + \lambda \, \boldsymbol{\alpha}^{\top} \boldsymbol{K} \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_N]^{\top}$, $\boldsymbol{y} = [y_1 \ \cdots \ y_N]^{\top}$ and $\boldsymbol{K}$ is the Gram matrix of entries $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i, j = 1, \ldots, N$. Therefore, the optimal solution is given by the linear system $(\boldsymbol{K} + \lambda \, \boldsymbol{I}) \, \boldsymbol{\alpha} = \boldsymbol{y}$, where $\boldsymbol{I}$ is the identity matrix of appropriate size.

This is the essence of the representer theorem which, in conjunction with the kernel trick, constitutes the foundations of the wide class of kernel methods, including SVM for

classification and regression [Vapnik, 1995; Burges, 1998]. Indeed, only the parameters α_i and the evaluation of the kernel κ are required for a regression task where ψ is evaluated at some given sample $\boldsymbol{x}$, namely $\psi(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x})$, or for a classification task where $\psi(\boldsymbol{x})$ is compared to a threshold in order to categorize the sample.

2.4 The Preimage Problem in Kernel Methods

As described above, kernel methods for classification and regression tasks do not require the explicit definition of the $\Phi(\boldsymbol{x}_i)$, namely the images of samples under the mapping function Φ, or any other element from the feature space $\mathcal{H}$. This is not the case when dealing with kernel methods for pattern recognition, feature extraction and denoising, for instance, where one needs the *preimage* of some $\psi \in \mathcal{H}$, *i.e.,* the element in $\mathcal{X}$ whose image under Φ is ψ. It turns out that only seldom elements in the feature space have an exact preimage in the input space [Mika et al., 1999]. To overcome these difficulties, the preimage problem seeks the best approximation, namely the element $\boldsymbol{x}^*$ in the input space $\mathcal{X}$ whose image under Φ is the closest to the feature ψ. Before defining the preimage problem and describing available techniques to solve it, we illustrate the importance of the preimage through the denoising problem.

2.4.1 The curse of the preimage problem

Within the framework of kernel methods, consider a denoising task by projecting the image $\Phi(\widetilde{\boldsymbol{x}})$ of any given sample $\widetilde{\boldsymbol{x}} \in \mathcal{X}$ onto a relevant manifold, such as the one obtained from the principal axes in kernel PCA.

Consider a set of training samples $\boldsymbol{x}_i$, for $i = 1, \ldots, N$. Following the representer theorem, the principal axes take the form

$$\psi_k = \sum_{i=1}^{N} \alpha_{k,i} \Phi(\boldsymbol{x}_i),$$

where the parameters $\alpha_{k,1}, \ldots, \alpha_{k,N}$ of the k-th principal axis are obtained from an eigendecomposition of the Gram matrix of entries $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, for $i, j = 1, \ldots, N$; See [Schölkopf et al., 1998] for more details. Once the subspace defined by considering K principal axes, one can solve a denoising task [Mika et al., 1999]. To this end, the noisy

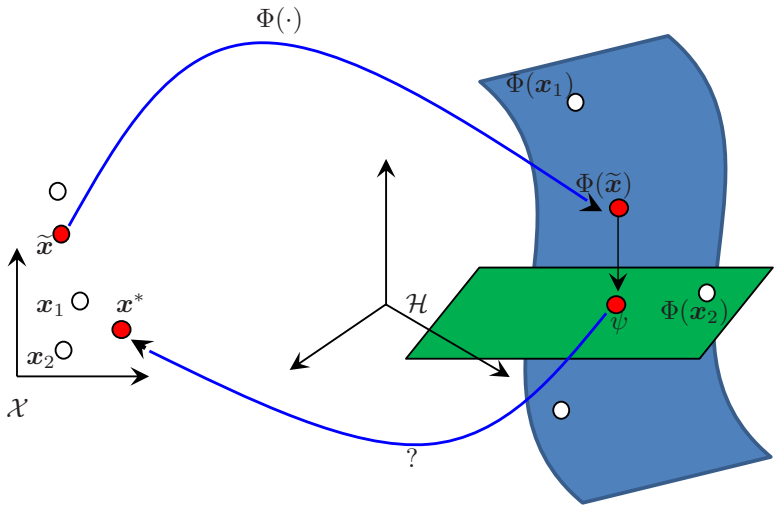


Figure 2.2: Illustration of the preimage problem, namely the estimation of the element $\boldsymbol{x}^*$ in the input space $\mathcal{X}$ whose image is the closest to ψ in the feature space $\mathcal{H}$.

sample $\widetilde{\boldsymbol{x}}$ from the input space $\mathcal{X}$ needs first to be mapped to the feature space $\mathcal{H}$. The resulting image $\Phi(\widetilde{\boldsymbol{x}})$ is denoised by projecting it onto the aforementioned subspace, which yields

$$\begin{aligned}\sum_{k=1}^{K}\langle\psi_k,\Phi(\widetilde{\boldsymbol{x}})\rangle_{\mathcal{H}}\,\psi_k &= \sum_{k=1}^{K}\Big\langle\sum_{j=1}^{N}\alpha_{k,j}\Phi(\boldsymbol{x}_j),\Phi(\widetilde{\boldsymbol{x}})\Big\rangle_{\mathcal{H}}\sum_{i=1}^{N}\alpha_{k,i}\Phi(\boldsymbol{x}_i)\\ &= \sum_{k=1}^{K}\sum_{j=1}^{N}\alpha_{k,j}\kappa(\boldsymbol{x}_j,\widetilde{\boldsymbol{x}})\sum_{i=1}^{N}\alpha_{k,i}\Phi(\boldsymbol{x}_i),\end{aligned}$$

thus, by setting $\alpha_i = \sum_{k=1}^{K}\sum_{j=1}^{N}\alpha_{k,j}\alpha_{k,i}\kappa(\boldsymbol{x}_j,\widetilde{\boldsymbol{x}})$, we get

$$\psi = \sum_{i=1}^{N}\alpha_i\,\Phi(\boldsymbol{x}_i). \tag{2.11}$$

Thus, the denoised pattern can be written as a linear expansion of the N images of the training data. It turns out that this pattern is of little interest as it belongs to the potentially infinite-dimensional RKHS. Of great interest is representing the denoised pattern in the input space, as illustrated in Figure 2.2, since the denoised sample needs to be reconstructed in the original space of input data.

Independent of the pattern recognition task (denoising, feature extraction, ...) and the used algorithm (PCA or any manifold/machine learning algorithm), the resulting pattern takes the form (2.11) due to the representer theorem in Theorem 2.10. This is

the case for instance of the algorithm introduced in [Essoloh et al., 2008] for distributed auto-localization in wireless sensor networks using a reproducing kernel Hilbert space, the kernel-based autoregressive model for prediction in time series proposed in [Kallas et al., 2013a], and the supervised nonlinear unmixing of hyperspectral images in [Nguyen et al., 2013], to name a few.

All these machine learning techniques (and more, such as kernel-based variants of the NMF; see Section 3.2.2 in Chapter 3) suffer from the so-called *curse of the preimage problem*, namely require to estimate, in the input space, the counterpart of some element of the RKHS. Most elements of the feature space RKHS, including the elements of interest, are not valid images, *i.e.*, the result of applying the map to some sample from the input space. To overcome this limitation, one needs to operate an approximation, by solving the so-called preimage problem.

2.4.2 Formulation of the preimage problem

The preimage problem is an ill-posed problem in the sense of Hadamard [Honeine and Richard, 2011] (see footnote 1 in page 5.). Indeed, the feature space is often of higher dimension than the input space, *e.g.*, the RKHS associated with the Gaussian kernel has infinite dimension. As a consequence, the exact preimages of almost all the elements in RKHS do not exist, and even when an exact preimage exists, it may not be unique. To this end, instead of identifying the exact preimage, one turns to an approximate solution, the so-called preimage, by estimating the element $\boldsymbol{x}^*$ in the input space whose image under Φ is as close as possible to ψ in the feature space, namely $\Phi(\boldsymbol{x}^*) \approx \psi$.

Mathematically, the preimage problem is formulated by minimizing the distance between $\Phi(\boldsymbol{x}^*)$ and ψ, namely

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \|\psi - \Phi(\boldsymbol{x})\|_{\mathcal{H}}^2.$$

Besides this distance-based expression, other formulations can also be considered, such as maximizing their inner product or their normalized inner product (*i.e.*, correlation or cosine), such as $\arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \psi, \Phi(\boldsymbol{x}) \rangle_{\mathcal{H}}$. In practice, all these formulations provide roughly the same results; for this reason, we consider in the following the above distance-based formulation without loss of generality.

Let $\psi = \sum_{i=1}^{N} \alpha_i \Phi(\boldsymbol{x}_i)$ ccording to the representer theorem in Theorem 2.10, leading to the following optimization problem

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \frac{1}{2} \Big\| \sum_{i=1}^{N} \alpha_i \Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}^*) \Big\|_{\mathcal{H}}^2,$$

where the factor $1/2$ is added for convenience. By expanding this expression and applying the kernel trick, the optimization problem boils down to minimizing the following cost function

$$J(\boldsymbol{x}) = -\sum_{i=1}^{N} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \frac{1}{2}\kappa(\boldsymbol{x}, \boldsymbol{x}), \tag{2.12}$$

where the terms independent of $\boldsymbol{x}$ are removed. Moreover, this cost function can be further reduced when dealing with RBF kernels such as the Gaussian and Laplacian kernels since the last term of the right-hand-side becomes a constant independent of $\boldsymbol{x}$.

2.4.3 Typical techniques for solving the preimage problem

The minimization of the cost function in (2.12) is generally nonlinear and nonconvex for nonlinear kernels. To solve this optimization problem, a variety of techniques have been developed. These techniques are essentially either based on classical optimization schemes, such as gradient descent and fixed-point techniques, or learning-based techniques that explore connections with dimensionality-reduction techniques, such as the multidimensional scaling and the conformal map approaches, to name a few [Honeine and Richard, 2011].

2.4.3.1 Gradient descent method

The gradient descent method is a first-order optimization tool that is feasible in versatile optimization problems. Over the iterations, it takes steps proportional to the opposite direction of the gradient of the objective function at current point, namely

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \nabla_{\boldsymbol{x}} J(\boldsymbol{x}^t), \tag{2.13}$$

where t is the current iteration number and η_t is the tunable step-size parameter. For the preimage problem, the gradient of the objective function $J(\boldsymbol{x})$ in (2.12) w.r.t. $\boldsymbol{x}$

takes the form

$$\nabla_{\boldsymbol{x}} J(\boldsymbol{x}) = -\sum_{i=1}^{N} \alpha_i \nabla_{\boldsymbol{x}} \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \frac{1}{2} \nabla_{\boldsymbol{x}} \kappa(\boldsymbol{x}, \boldsymbol{x}). \tag{2.14}$$

Here, $\nabla_{\boldsymbol{x}} \kappa(\cdot, \cdot)$ denotes the gradient of the kernel function w.r.t. its first argument, which can be easily derived for most valid kernels, as given in Table 2.1.

Since the objective function under study is nonlinear and nonconvex, local minima can be obtained using the gradient descent method. To alleviate this problem, several simulations with different initial points should be performed in practice. Moreover, the choice of an appropriate value of the step-size parameter, using for instance a line-search procedure, is cumbersome.

2.4.3.2 Iterative fixed-point method for particular kernels

In order to overcome the issue of tuning the step-size parameter, the principle of the fixed-point method explores an expression of the form $\boldsymbol{x}^* = f(\boldsymbol{x}^*)$ for some function f. In this case, the update rule takes the form $\boldsymbol{x}^{t+1} = f(\boldsymbol{x}^t)$. This approach can be used to solve the preimage problem, as shown next for the Gaussian kernel [Mika et al., 1999]. One can easily provide extensions to other RBF and projective kernels, as given in [Kwok and Tsang, 2004].

The Gaussian kernel takes the form

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\tfrac{-1}{2\sigma^2} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right),$$

and satisfies $\kappa(\boldsymbol{x}, \boldsymbol{x}) = 1$. After removing the constant term, the objective function (2.12) becomes

$$-\sum_{i=1}^{N} \alpha_i \exp\left(\tfrac{-1}{2\sigma^2} \|\boldsymbol{x}_i - \boldsymbol{x}\|^2\right),$$

with its gradient expressed by

$$\nabla_{\boldsymbol{x}} J(\boldsymbol{x}) = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \alpha_i \exp\left(\tfrac{-1}{2\sigma^2} \|\boldsymbol{x}_i - \boldsymbol{x}\|^2\right)(\boldsymbol{x} - \boldsymbol{x}_i).$$

By setting $\nabla_{\boldsymbol{x}} J(\boldsymbol{x})$ to zero at the optimum, we obtain the fixed-point iterative scheme

$$\boldsymbol{x}^{t+1} = \frac{\sum_{i=1}^{N} \alpha_i \exp\left(\frac{-1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}^t\|^2\right) \boldsymbol{x}_i}{\sum_{i=1}^{N} \alpha_i \exp\left(\frac{-1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}^t\|^2\right)}. \tag{2.15}$$

The disadvantage of the fixed-point method is that it suffers local minima, as well as numerical instabilities [Kwok and Tsang, 2004].

2.4.3.3 Multidimensional scaling approach

Another way to tackle the preimage problem is from the perspective of a dimensionality-reduction problem, since one needs to construct samples in the input space from elements in the larger-dimension feature space. From this connection between the two problems, the authors of [Kwok and Tsang, 2003] propose to address the preimage problem using a multidimensional scaling (MDS) approach [Williams, 2002], namely by preserving pairwise distances in input-feature spaces.

Consider the distance in the input space, namely $\|\boldsymbol{x} - \boldsymbol{x}_i\|$, and the corresponding feature-space distance between the mapped data, namely $\|\psi - \Phi(\boldsymbol{x}_i)\|_{\mathcal{H}}$. In the ideal case according to the MDS, these distances are preserved with

$$\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 = \|\psi - \Phi(\boldsymbol{x}_i)\|_{\mathcal{H}}^2, \quad \text{for all } i = 1, \ldots, N.$$

However, since the exact preimage does not exist in general, and the MDS assumption inaccurate, one considers the approximated solution obtained by minimizing the least square error between pairwise distances, namely

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \sum_{i=1}^{N} \left| \|\boldsymbol{x} - \boldsymbol{x}_i\|^2 - \|\psi - \Phi(\boldsymbol{x}_i)\|_{\mathcal{H}}^2 \right|^2.$$

By setting the gradient of the above cost function to zero, one gets an iterative fixed-point update rule with

$$\boldsymbol{x}^{t+1} = \frac{\sum_{i=1}^{N} \alpha_i (\|\boldsymbol{x}_i - \boldsymbol{x}^t\|^2 - \delta_i^2)\, \boldsymbol{x}_i}{\sum_{i=1}^{N} (\|\boldsymbol{x}_i - \boldsymbol{x}^t\|^2 - \delta_i^2)},$$

where δ_i denotes the distance in the feature space between ψ and $\Phi(\boldsymbol{x}_i)$, namely $\delta_i = \|\psi - \Phi(\boldsymbol{x}_i)\|_{\mathcal{H}}$. Under the assumption of centered data, a non-iterative approach is established using linear algebra in [Kwok and Tsang, 2003]. Compared with the iterative fixed-point method, the non-iterative method does not suffer from numerical instabilities.

2.4.3.4 Conformal map approach

A major drawback of the MDS approach is the assumption that the distances are preserved in both spaces, which is not the case in general. A way to overcome this drawback, as proposed in [Honeine and Richard, 2009] with the conformal map approach, is to consider the conservation of the inner products in both spaces, where the inner products in the feature space are computed in an appropriate basis, namely the one that allows the isometry with the input space.

To learn the preimage, the proposed approach operates in two steps. First, one constructs a basis in the feature space that is isometric with the input space basis. Second, by representing ψ in this basis and the preservation of the inner products in both input and obtained basis, one gets the inner products between the preimage and the available data in the input space, thereby estimating the preimage.

Let N be the number of basis functions to be constructed in the feature space. Following the representer theorem in Theorem 2.10, each of the basis functions in the feature space $\mathcal{H}$ can be expressed as a linear combination of the N mapped samples, namely $\Psi_k = \sum_{i=1}^{N} \alpha_{k,i}\, \Phi(\boldsymbol{x}_i)$, for $k = 1, 2, ..., N$, where $\alpha_{k,i}$ are the coefficients to be determined. Thereby, the coordinates of any image $\Phi(\boldsymbol{x}_i)$ in $\mathcal{H}$ are obtained by

$$\Psi_{\boldsymbol{x}_i} = [\langle \Psi_1, \Phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} \quad \langle \Psi_2, \Phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} \quad \cdots \quad \langle \Psi_N, \Phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}}]^\top.$$

In the ideal case, the inner products are preserved in both input and feature spaces, with

$$\Psi_{\boldsymbol{x}_i}^\top \Psi_{\boldsymbol{x}_j} = \boldsymbol{x}_i^\top \boldsymbol{x}_j, \quad \text{for all} \quad i, j = 1, 2, ..., N.$$

To solve this problem, one seeks to minimize the quadratic error given by

$$\min_{\Psi_1,\ldots,\Psi_N} \sum_{i,j=1}^{N} \left| \Psi_{\boldsymbol{x}_i}^\top \Psi_{\boldsymbol{x}_j} - \boldsymbol{x}_i^\top \boldsymbol{x}_j \right|^2 + \eta \sum_{k=1}^{N} \|\Psi_N\|_{\mathcal{H}}^2,$$

where the second term is the regularization term penalizing large norm functions. By writing the above minimization problem into the matrix form, which leads to the optimal coefficients $\alpha_{k,i}$, and considering the preservation of the inner products in both spaces, we obtain the preimage $\boldsymbol{x}^*$ of any $\sum_{i=1}^{N} \alpha_i \Phi(\boldsymbol{x}_i)$ from the expression

$$\boldsymbol{x}^* = (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} - \eta\boldsymbol{K}^{-1}) \begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}^\top,$$

where $\boldsymbol{X} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_N]$.

The conformal map approach is free from numerical instabilities or local minima as opposed to classical optimization techniques, such as gradient descent and fixed-point techniques. Compared to the MDS technique, the conformal map does not require any assumption of preserving the distances in both input and feature spaces.

2.5 Conclusion

In this chapter, we have briefly reviewed the fundamental concepts and properties behind kernel methods, including the kernel trick and the Moore-Aronszajn Theorem, as well as the definition of nonlinear models with the representer theorem. These key properties have been widely used to derive kernel methods for decisional tasks, with classification and regression. We have also described the curse of the preimage problem, formulated the preimage problem and presented the most-known techniques to tackle this problem.

The curse of the preimage problem is an inherent issue in kernel methods, due to the representer theorem and the underlying nonlinear model. Many pattern recognition tasks for unsupervised learning suffer from this issue, including auto-localisation in wireless sensor networks [Essoloh et al., 2008] and kernel-based autoregressive modeling in time series [Kallas et al., 2013a], as well as the nonlinear unmixing of hyperspectral images as given in [Nguyen et al., 2013]. In the succeeding chapter, we show that kernelizations of the nonnegative matrix factorization (NMF), as conducted in [Zhang et al., 2006; Ding et al., 2010; Li and Ngom, 2012], also suffer from this curse. We propose an original kernel-based framework for nonlinear NMF that does not suffer from the curse of the preimage problem.

Chapter 3

Kernel NMF Without the Preimage Problem

Contents

The nonnegative matrix factorization (NMF) is widely used in signal and image processing, including bio-informatics, blind source separation and hyperspectral image analysis in remote sensing. A great challenge arises when dealing with a nonlinear formulation of the NMF. Within the framework of kernel machines, the models suggested in the literature do not allow the representation of the factorization matrices, which is a fallout of the curse of the preimage. In this chapter, we propose a novel kernel-based model for the NMF that does not suffer from the preimage problem, by investigating the estimation of the factorization matrices directly in the input space. For several kernel functions, we describe two schemes for iterative algorithms: additive update rules based on a gradient descent scheme and multiplicative update rules in the same spirit as in the Lee and Seung algorithm. Within the proposed framework, we develop several extensions to incorporate constraints, including sparseness, smoothness, and spatial regularization with a total-variation-like penalty. The effectiveness of the proposed method is demonstrated with the problem of unmixing hyperspectral images, using well-known real images and results from state-of-the-art techniques.

3.1 Introduction

The nonnegative matrix factorization (NMF) has become a prominent analysis technique in many fields, owing to its power to extract sparse and tractable interpretable representations from a given data matrix. The scope of application spans feature extraction, compression and visualization, within pattern recognition, machine learning, and signal and image processing [Comon and Jutten, 2010; Gillis, 2014]. It has been popularized since Lee and Seung discovered that, when applied to an image, "NMF is able to learn the parts of objects" [Lee and Seung, 1999]. Since then, the NMF has been successfully applied in image classification [Buchsbaum and Bloch, 2002; Guillamet et al., 2001], face expression recognition [Li et al., 2001; Buciu and Pitas, 2004], audio analysis [Smaragdis, 2004; Févotte et al., 2008], objet recognition [Liu and Zheng, 2004; Wild et al., 2004], computational biology [Devarajan, 2008], gene expression data [Brunet et al., 2004; Kim and Tidor, 2003], and clustering [Young et al., 2006]. Moreover, the NMF is tightly connected to spectral clustering [Xu et al., 2003; Ding et al., 2005; Li and Ding, 2006]. See also [Cichocki et al., 2009] [Comon and Jutten, 2010, Chap. 13].

The NMF consists in approximating a nonnegative matrix with two low-rank nonnegative matrices. It allows a sparse representation with nonnegativity constraints, which often provides a physical interpretation to the factorization thanks to the resulting part-based representation, as opposed to conventional models. Typically, this idea is described with the issue of spectral unmixing in hyperspectral imagery. As already explained in Section 1.2, the spectral unmixing of a given hyperspectral image aims to extract the spectra of "pure" materials, called endmembers, and to estimate the abundance of each endmember in every pixel, *i.e.,* every position of the area under scrutiny. It is obvious that both abundances and spectra of endmembers are nonnegative. The NMF provides a decomposition suitable for such physical interpretation.

The physical interpretation of the NMF is however not for free. To illustrate this, consider the well-known singular-value-decomposition (SVD), which allows to solve efficiently the unconstrained matrix factorization problem with orthogonality constraints, under the risk of losing the physical meaning while providing a unique solution. It is known that the SVD has polynomial-time complexity. As opposed to the SVD, the NMF is unfortunately a NP-hard and an ill-posed problem, in general. In fact, it is proven in [Vavasis, 2009] that the NMF is NP-hard; see also [Gillis, 2011]. The NMF is ill-posed, as illustrated by the fact that the decomposition is not unique; see [Huang et al., 2014] and references therein. In practice, the nonuniqueness issue is alleviated by including priors other than the nonnegativity, the most known being sparseness and smoothness constraints.

First studied in 1977 in [Leggett, 1977], the NMF problem was reinvented several times, scilicet with the work of [Paatero and Tapper, 1994]. It has gained popularity thanks to the work published in *Nature* by [Lee and Seung, 1999]. Many optimization algorithms have been proposed for NMF, such as the multiple update rules [Lee and Seung, 2001] and the nonnegative least squares [Kim and Park, 2008]. Sparseness, which allows the uniqueness and enhances interpretation, is often imposed either with projections [Hoyer, 2004] or with ℓ_1-norm regularization [Kim and Park, 2007]. Smoothness also reduces the degrees of freedom, typically in the spectral unmixing problem, either by using piecewise smoothness of the estimated endmembers [Pauca et al., 2006; Jia and Qian, 2009; Qian et al., 2011], or by favoring spatial coherence with a regularization similar to the total-variation (TV) penalty [Iordache et al., 2012]. Additional constraints are the orthogonality [Ding et al., 2006; Li et al., 2007], the minimum-volume [Zhou et al., 2011a],

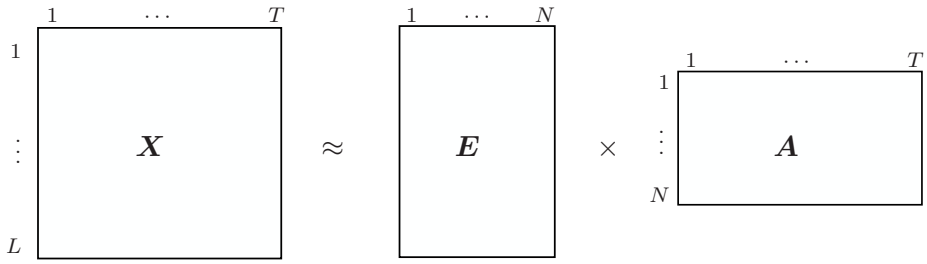


Figure 3.1: The linear NMF model.

and the sum-to-one constraint which is often imposed on the abundances [Masalmah and Veléz-Reyes, 2008]. As illustrated in all these developments with the unmixing problem in hyperspectral imagery, the NMF and most of its variants are based on a linear mixing assumption.

Providing nonlinear models for NMF is a challenging issue [Yang and Oja, 2012]. Recently, a few attempts have been made to derive kernel-based NMF, for the sake of a nonlinear variant of the conventional NMF [Zhang et al., 2006; Ding et al., 2010; Li and Ngom, 2012]. To this end, the linear model in the latter is defined by writing each column of the matrix under scrutiny as a linear combination of the columns of the first matrix to be determined, the second matrix being defined by the weights of the linear combination. By defining the input space with the columns of the studied matrix, these columns are mapped with a nonlinear transformation to some feature space where the linear model is applied. Unfortunately, the obtained results cannot be exploited, since the columns of the first unknown matrix lie in the feature space. One needs to get back from the (often infinite dimensional) feature space to the input space. This is the curse of the preimage problem, a major drawback inherited from kernel machines as described in Section 2.4. While the preimage problem was initially revealed in denoising tasks [Mika et al., 1999], this issue yields an even more difficult problem when dealing with the nonnegativity constraint, as demonstrated in [Kallas et al., 2013b].

In this chapter, we propose an original kernel-based framework for nonlinear NMF that does not suffer from the curse of the preimage problem, as opposed to other techniques derived within kernel machines (see Figure 3.2 and Figure 3.3 for a snapshot of this difference). To this end, we explore a novel model defined by the mapping of the columns of the matrices (the investigated matrix and the first unknown one), these columns lying in the input space. It turns out that the corresponding optimization problem can be efficiently tackled directly in the input space, thanks to the nature of the underlying

kernel function. We derive two iterative algorithms: an additive update rule based on a gradient descent scheme, and a multiplicative update rule in the same spirit of [Lee and Seung, 1999]. We investigate expressions associated to the polynomial and Gaussian kernels, as well as the linear one which yields the conventional linear NMF. Based on the proposed framework, we describe several extensions to incorporate constraints, including sparseness and smoothness, as well as a TV-like spatial regularization. The relevance of the proposed approach with its extensions is shown on two well-known hyperspectral images.

3.2 From the Linear NMF to its Kernelizations

This section presents the conventional linear NMF and its kernel-based counterparts, illustrating the preimage problem.

3.2.1 A primer on the NMF

The conventional NMF consists in approximating a nonnegative matrix $\boldsymbol{X}$ with the product of two low-rank nonnegative matrices $\boldsymbol{E}$ and $\boldsymbol{A}$, namely

$$\boldsymbol{X} \approx \boldsymbol{E}\boldsymbol{A} \tag{3.1}$$

subject to $\boldsymbol{E} \geq 0$ and $\boldsymbol{A} \geq 0$; See Figure 3.1 for notations. The former nonnegativity constraint is relaxed in the so-called semi-NMF. The optimization problem is written in terms of the nonnegative least squares optimization, with

$$\arg \min_{\boldsymbol{A},\boldsymbol{E}\geq 0} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm and $\frac{1}{2}$ is included for convenience.

Under the nonnegativity constraints, the estimation of the entries of both matrices $\boldsymbol{E}$ and $\boldsymbol{A}$ is not convex. Luckily, the estimation of each matrix, separately, is a convex optimization problem. Most NMF algorithms take advantage of this property, with an iterative technique that alternates the optimization over each matrix while keeping the other one fixed. This is the so-called two block-coordinate strategy. The most

commonly used algorithms are the gradient descent rule and the multiplicative update rule (expressions are given in Section 3.3.3.1). For a recent survey of standard algorithms, see [Comon and Jutten, 2010, Chapter 13], as well as [Gillis, 2014] and references therein.

It is easy to notice that the matrix model (3.1) can be considered vector-wise, by dealing separately with each column of the matrix $\boldsymbol{X}$. Let

$$\boldsymbol{X} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_T],$$

$$\boldsymbol{E} = [\boldsymbol{e}_1 \quad \boldsymbol{e}_2 \quad \cdots \quad \boldsymbol{e}_N],$$

and a_{nt} be the (n,t)-th entry in $\boldsymbol{A}$. Then the NMF consists in estimating the nonnegative vectors $\boldsymbol{e}_n$ and scalars a_{nt}, for all $n = 1, \ldots, N$ and $t = 1, \ldots, T$, such that

$$\boldsymbol{x}_t \approx \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n. \tag{3.2}$$

Following this model, the resulting optimization problem is

$$\min_{a_{nt}, \boldsymbol{e}_n \geq 0} \mathcal{J},$$

where

$$\mathcal{J} = \frac{1}{2} \sum_{t=1}^{T} \left\| \boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n \right\|^2. \tag{3.3}$$

It is this vector-wise model that is investigated in deriving kernel-based NMF.

Without loss of generality, we illustrate the NMF with the problem of unmixing in hyperspectral imagery. In this case and following the notation in (3.2), each spectrum $\boldsymbol{x}_t$ of the image is decomposed into a set of spectra $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N$ (*i.e.,* endmembers), while $a_{1t}, a_{2t}, \ldots, a_{Nt}$ denote their respective abundances. Such physical problem allows us to incorporate additional constraints and impose structural regularity of the solution, as detailed in Section 3.4.

3.2.2 On kernelizing the NMF: the curse of the preimage problem

Some attempts have been made to derive nonlinear, kernel-based, NMF. These methods originate in mapping the columns of $\boldsymbol{X}$ with a nonlinear function $\Phi(\cdot)$, namely transforming $\boldsymbol{x}_t$ into $\Phi(\boldsymbol{x}_t)$ for $t = 1, \ldots, T$. Let $\mathcal{H}$ be the resulting feature space, with the

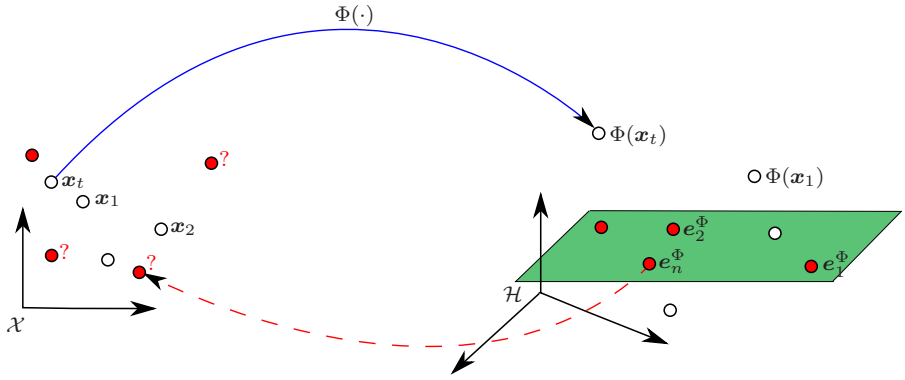


Figure 3.2: When applying NMF directly in the feature space, each column $\boldsymbol{x}_t$ of $\boldsymbol{X}$ is mapped to the feature space $\mathcal{H}$, where the basis elements $\boldsymbol{e}_n^\Phi$ are defined. Without any access to these elements, one needs to estimate their preimages (shown with ?).

associated norm $\|\Phi(\boldsymbol{x}_t)\|_\mathcal{H}$ and the corresponding inner product $\langle \Phi(\boldsymbol{x}_t), \Phi(\boldsymbol{x}_{t'})\rangle_\mathcal{H}$. The latter is evaluated with a kernel function $\kappa(\boldsymbol{x}_t, \boldsymbol{x}_{t'})$.

The NMF model is defined in the feature space with

$$\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n^\Phi, \tag{3.4}$$

written in matrix form as

$$\boldsymbol{X}^\Phi \approx \begin{bmatrix} \boldsymbol{e}_1^\Phi & \boldsymbol{e}_2^\Phi & \cdots & \boldsymbol{e}_N^\Phi \end{bmatrix} \boldsymbol{A},$$

where

$$\boldsymbol{X}^\Phi = \begin{bmatrix} \Phi(\boldsymbol{x}_1) & \Phi(\boldsymbol{x}_2) & \cdots & \Phi(\boldsymbol{x}_T) \end{bmatrix}.$$

Here, the elements $\boldsymbol{e}_n^\Phi$ lie in the feature space $\mathcal{H}$, since $\Phi(\boldsymbol{x}_t)$ belongs to the span of all these elements $\boldsymbol{e}_n^\Phi$. Essentially, all kernel-based NMF proposed so far have been considering this model [Zhang et al., 2006; Buciu et al., 2008; Ding et al., 2010; Li and Ngom, 2012; An et al., 2011]. Unfortunately, the model (3.4) suffers from an important weakness, inherited from kernel machines: one has no access to the elements in the feature space, but only to their inner products using the kernel function. The fact that the elements $\boldsymbol{e}_n^\Phi$ lie in the feature space $\mathcal{H}$ leads to several drawbacks in NMF, as shown next.

The first drawback of the model (3.4) is revealed when one computes the inner product of the images of any pair $(\boldsymbol{x}_{t'}, \boldsymbol{x}_t)$, in the feature space, namely

$$\langle \Phi(\boldsymbol{x}_{t'}), \Phi(\boldsymbol{x}_t)\rangle_\mathcal{H} \approx \sum_{n=1}^{N} a_{nt}\, \langle \Phi(\boldsymbol{x}_{t'}), \boldsymbol{e}_n^\Phi\rangle_\mathcal{H}.$$

Here, the left-hand-side is equivalent to $\kappa(\boldsymbol{x}_{t'}, \boldsymbol{x}_t)$. Unfortunately, the inner product $\langle \Phi(\boldsymbol{x}_{t'}), \boldsymbol{e}_n^\Phi \rangle_{\mathcal{H}}$ cannot be evaluated using the kernel function. To circumvent this difficulty, one should restrict the form of all the elements $\boldsymbol{e}_n^\Phi$, as investigated in [Lee et al., 2009; An et al., 2011] by writing them in terms of a linear combination of the images $\Phi(\boldsymbol{x}_t)$. By rearranging the coefficients of the linear combination in a matrix $\boldsymbol{W}$, the problem takes the form $\boldsymbol{X}^\Phi \approx \boldsymbol{X}^\Phi \boldsymbol{W} \boldsymbol{A}$. While this simplifies the optimization problem, it is however quiet different from the conventional NMF model (3.1).

Another downside of the model (3.4) is that one cannot impose the nonnegativity of the elements in the feature space, and in particular $\boldsymbol{e}_n^\Phi$. Therefore, the constraint $\boldsymbol{e}_n^\Phi \geq 0$ should be dropped. Only the coefficients a_{nt} can be set to nonnegative values. In this case, one cannot tackle the NMF problem. For this reason, only the constraint $\boldsymbol{A} \geq 0$ could be imposed, which yields the relaxed semi-NMF problem [Li and Ngom, 2012].

The most important drawback is that one has no access to the elements $\boldsymbol{e}_n^\Phi$, and therefore cannot extract the endmembers. Having a given matrix $\boldsymbol{X}$, only the abundance matrix $\boldsymbol{A}$ is determined. To estimate the columns of the other matrix $\boldsymbol{E}$, namely the endmembers, one needs to solve the so-called preimage problem. This ill-posed problem consists of estimating an input vector whose image, defined by the nonlinear map $\Phi(\cdot)$, is as close as possible to a given element in the feature space. In other words, one determines each column $\boldsymbol{e}_n$ of $\boldsymbol{E}$ by solving $\Phi(\boldsymbol{e}_n) \approx \boldsymbol{e}_n^\Phi$, for all $n = 1, \ldots, N$, which is a nonconvex, nonlinear, ill-posed optimization problem. This issue is obvious in all previous works on kernel-based NMF; see for instance [Pan et al., 2011]. Including the nonnegativity constraint to the preimage problem is a challenging problem [Kallas et al., 2011, 2013b].

Few attempts were conducted to circumvent some of these difficulties. The polynomial kernel $(\boldsymbol{x}_i^\top \boldsymbol{x}_j + \mathrm{c})^d$ with $c = 0$, is considered in [Buciu et al., 2008], restricting the derivation to this kernel as argued by the authors. [Pan et al., 2011] approximates the kernel by another one associated to a *nonnegative map*, which requires to solve another optimization problem prior to processing the one associated to the NMF. Moreover, the preimage problem needs to be solved subsequently.

For all these reasons, applying the nonnegative matrix factorization in the feature space has been often limited to preprocessing data before solving a classification task. Still, one has no access to the bases in the resulting relevant representation. Next, we propose a framework where both matrices can be exhibited, without suffering from the curse

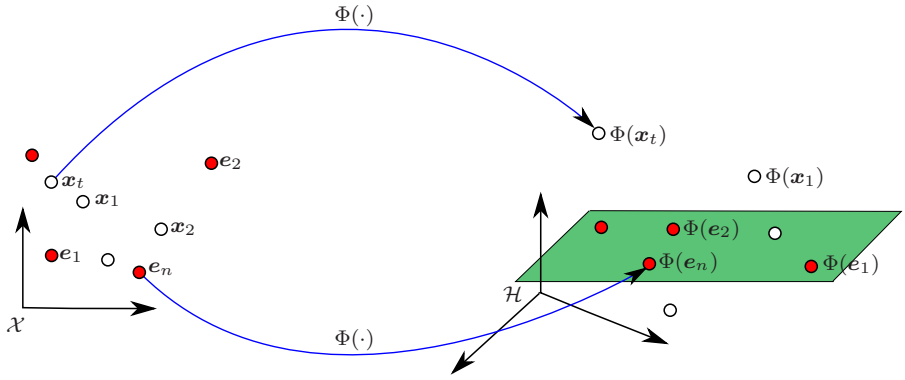


Figure 3.3: Illustration of the proposed KNMF. In contrast to the one in Figure 3.2, the proposed approach estimates the basis elements $\boldsymbol{e}_n$ directly in the input space $\mathcal{X}$. This strategy allows to overcome the curse of the preimage problem.

of the preimage problem. The core of the difference between these two approaches is illustrated in Figure 3.2 and Figure 3.3.

3.3 A Novel Framework for KNMF

In this section, we propose a novel framework to derive a kernel-based NMF, referred as **KNMF**, where the underlying model is defined by a basis in the input space, and therefore without the pain of solving the preimage problem. To this end, we consider the following matrix factorization model:

$$\boldsymbol{X}^{\Phi} \approx \boldsymbol{E}^{\Phi}\boldsymbol{A},$$

where

$$\boldsymbol{E}^{\Phi} = [\Phi(\boldsymbol{e}_1) \;\; \Phi(\boldsymbol{e}_2) \;\; \cdots \;\; \Phi(\boldsymbol{e}_N)].$$

The nonnegativity constraint is imposed to $\boldsymbol{A} \geq 0$ and $\boldsymbol{e}_n \geq 0$ for all $n = 1, \ldots, N$. Therefore, we have the following model:

$$\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n). \tag{3.5}$$

This means that we are estimating the basis elements $\boldsymbol{e}_n$ directly in the input space, as opposed to the model given in (3.4) where the elements $\boldsymbol{e}_n^{\Phi}$ lie in the feature space.

To estimate all $\boldsymbol{e}_n$ and a_{nt}, we consider a simple alternating technique to minimize the corresponding cost function

$$\mathcal{J} = \frac{1}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2, \tag{3.6}$$

thus yielding the optimization problem

$$\min_{a_{nt},\boldsymbol{e}_n}\sum_{t=1}^{T}\Big(-\sum_{n=1}^{N} a_{nt}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_{nt}a_{mt}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)\Big),$$

where $\kappa(\boldsymbol{x}_t,\boldsymbol{x}_t)$ is removed from the expression since it is independent of a_{nt} and $\boldsymbol{e}_n$. By taking its derivative with respect to a_{nt}, we obtain the following expression:

$$\nabla_{a_{nt}}\mathcal{J} = -\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\,\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m). \tag{3.7}$$

By taking the gradient of $\mathcal{J}$ with respect to $\boldsymbol{e}_n$, we obtain:

$$\nabla_{\boldsymbol{e}_n}\mathcal{J} = \sum_{t=1}^{T} a_{nt}\Big(-\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\,\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)\Big). \tag{3.8}$$

Here, $\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\cdot)$, which denotes the gradient of the kernel with respect to its argument $\boldsymbol{e}_n$, can be easily derived for most valid kernels, as given in Table 2.1. Expressions for the linear, polynomial and Gaussian kernels are given in Section 3.3.3. But before, we provide some insights on the physical interpretation of the proposed model, then we derive two algorithms to solve the above KNMF.

3.3.1 Remarks on the physical interpretation

We study the interpretation of the proposed KNMF by connecting it to several state-of-the-art models. The nonlinear model $\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)$ given in (3.5) is closely related to the intimate mixture with Hapke model. This model for microscopic mixture is described in Section 1.4.2, and summarized as follows. The Hapke model uses the widely known bidirectional reflectance distribution function for microscopic mixtures, which describes the relationship of observed reflectance to the albedo of materials within the scene under scrutiny [Hapke, 1981]. The single-scattering albedo (SSA) for a wavelength λ is

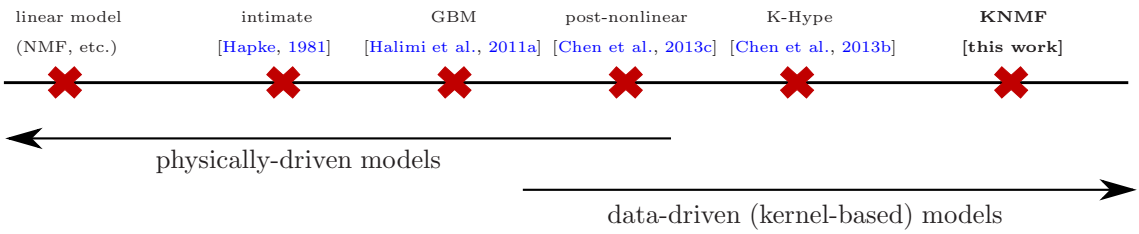


Figure 3.4: Schematic illustration of the physical interpretation confronted to data-driven nonlinearity in unmixing models.

defined as $w_\lambda = \sum_{n=1}^{N} f_n w_{n\lambda}$, where $\omega_{n\lambda}$ represents the material albedos and f_n represents the corresponding fractional proportions. It is easy to see that this microscopic mixing model is a linear model in the albedo domain, while it is nonlinear in the reflectance domain. Indeed, the model in the latter takes the form $\boldsymbol{x}_t \approx R\big(\sum_{n=1}^{N} f_{nt}\boldsymbol{w}_n\big)$, where R is the nonlinear Hapke's reflectance function and $\boldsymbol{w}_n$ is the vector of SSA at all wavelengths. By mapping the reflectance data to the albedo domain, the unknown microscopic proportions are estimated using the model $R^{-1}(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} f_{nt} R^{-1}(\boldsymbol{w}_n)$, where we have used $\boldsymbol{w}_n = R^{-1}(\boldsymbol{e}_n)$ as recommended in [Close et al., 2014]. The proposed nonlinear model has the same structure, where the difference lies in a nonlinearity R^{-1} characterized by a nonlinear kernel.

Machine learning with kernel-based models allow to alleviate missing physical interpretation of the underlying nonlinearity, as have been largely investigated in the literature (See 1.4.3 and references therein). Figure 3.4 attempts to categorize unmixing models/techniques in terms of both their "level" of physical interpretation and their data-driven modeling that describes the nonlinear relations. Consider for instance the post-nonlinear model of the form $\psi(\boldsymbol{E}\boldsymbol{a}_t)$; while it has a physical interpretation as stipulated in [Chen et al., 2013c], the nonlinear function $\psi(\cdot)$ is estimated from data with kernel-based methods, thus without any physical interpretation. It is worth noting that the linear and quadratic models can be viewed as special cases of kernel-based models.

3.3.2 Algorithms

In the following, we derive iterative techniques to minimize (3.6) using the two-block coordinate descent strategy, namely by alternating over the matrices $\boldsymbol{E}$ and $\boldsymbol{A}$, while keeping the other matrix fixed.

3.3.2.1 Additive update rule

In the first iterative algorithm, an additive update rule is presented to solve the optimization problem. It is based on a gradient descent scheme, alternating over both a_{nt} and $\boldsymbol{e}_n$, and is followed by a rectification function to impose their nonnegativity. A normalization step to impose the sum-to-one constraint on $\boldsymbol{a}_t$ can also be used.

By using a gradient descent scheme, we update a_{nt} according to $a_{nt} = a_{nt} - \eta_{nt} \nabla_{a_{nt}} \mathcal{J}$, where the stepsize η_{nt} can take different values for each pair (n, t). Replacing $\nabla_{a_{nt}} \mathcal{J}$ with its expression in (3.7), we get the following update rule:

$$a_{nt} = a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) - \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) \Big). \tag{3.9}$$

A similar procedure is applied to estimate the elements $\boldsymbol{e}_n$. The obtained update rule is given by

$$\boldsymbol{e}_n = \boldsymbol{e}_n - \eta_n \nabla_{\boldsymbol{e}_n} \mathcal{J}, \tag{3.10}$$

where the stepsize η_n can depend on n, and the expression of $\nabla_{\boldsymbol{e}_n} \mathcal{J}$ is given in (3.8). To impose the nonnegativity of the results, the negative values obtained by these update rules are set to zero. This is done by using the rectification function $x = \max(x, 0)$ over all a_{nt} and the entries in all the vectors $\boldsymbol{e}_n$. The abundance vectors can be normalized to have unit ℓ_1-norm, by substituting $\boldsymbol{a}_t$ with $\boldsymbol{a}_t / \|\boldsymbol{a}_t\|_1$ at each iteration.

Each of the above update rules followed by the rectification function can be expressed in a single formulation with the projected gradient descent scheme (PGD), proposed for the conventional NMF in [Lin, 2007b]. In this formulation, the update rule of the endmember $\boldsymbol{e}_n$ is

$$\boldsymbol{e}_n = \Big(\boldsymbol{e}_n - \eta_n \nabla_{\boldsymbol{e}_n} \mathcal{J} \Big)_+, \tag{3.11}$$

where $(\,\cdot\,)_+$ denotes the operator that projects its argument to the nonnegative set $\big\{ \boldsymbol{e}_n = [e_{n1} \;\; e_{n2} \;\; \cdots \;\; e_{nL}]^\top, e_{nl} \geq 0, l = 1, 2, \ldots, L \big\}$. Likewise, the update rule for the abundances is

$$a_{nt} = \left(a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) - \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) \Big) \right)_+, \tag{3.12}$$

where $\big(a_{nt}\big)_+ = \max(a_{nt}, 0)$ in this case.

3.3.2.2 Multiplicative update rule

The additive update rule is a simple procedure, however, the convergence is generally slow, and is directly related to the used stepsize value. In order to overcome these issues, we propose a multiplicative update rule, in the same spirit as in the conventional NMF [Lee and Seung, 2001].

To derive a multiplicative update rule for a_{nt}, the stepsize η_{nt} in (3.9) is chosen such that the first and the third terms in its right-hand-side cancel, that is

$$\eta_{nt} = \frac{a_{nt}}{\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)}.$$

Therefore, by substituting this expression in (3.9), we get the following update rule:

$$a_{nt} = a_{nt} \times \frac{\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)}{\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)}. \tag{3.13}$$

A normalization $\boldsymbol{a}_t = \boldsymbol{a}_t / \|\boldsymbol{a}_t\|_1$ can be considered to force the sum-to-one constraint. Compared with the additive rule, the above multiplicative rule has several interesting properties, such as the absence of any tunable stepsize parameter and the nonexistence of any rectification function. The latter property is due to the multiplicative nature which ensures that elements cannot become negative when one initializes with a nonnegative value.

A similar procedure is applied to estimate the basis elements $\boldsymbol{e}_n$, for $n = 1, \ldots, N$. The trick is that the expression of the gradient (3.8) can always be decomposed as $\nabla_{\boldsymbol{e}_n} \mathcal{J} = P - Q$, where P and Q are nonnegative. This is called the split gradient method [Lantéri et al., 2011]. It is obvious that this decomposition is not unique. Still, one can provide a multiplicative update rule for $\boldsymbol{e}_n$, with an expression depending on the used kernel function, as shown next for each of the most used kernels.

3.3.3 Kernels

All kernels studied in the literature of kernel machines can be investigated in the proposed framework. In the following, we derive expressions of the additive and multiplicative update rules for the most known kernel functions.

3.3.3.1 Back to the conventional linear NMF

A key property of the proposed KNMF framework is that the conventional NMF is a special case, when the linear kernel is used with $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{e}_n$, for any vector $\boldsymbol{z}$ from the input space. The gradient of the kernel is $\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \boldsymbol{z}$ in this case. By substituting this result in the above expressions, we get the additive update rules

$$\begin{cases} a_{nt} = \Big(a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m^\top \boldsymbol{e}_n - \boldsymbol{x}_t^\top \boldsymbol{e}_n \Big) \Big)_+ ; \\ \boldsymbol{e}_n = \Big(\boldsymbol{e}_n - \eta_n \sum_{t=1}^{T} a_{nt} \Big(-\boldsymbol{x}_t + \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m \Big) \Big)_+ , \end{cases}$$

as well as the multiplicative update rules

$$\begin{cases} a_{nt} = a_{nt} \times \dfrac{\boldsymbol{x}_t^\top \boldsymbol{e}_n}{\sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m^\top \boldsymbol{e}_n} ; \\ \boldsymbol{e}_n = \boldsymbol{e}_n \odot \dfrac{\sum_{t=1}^{T} a_{nt}\, \boldsymbol{x}_t}{\sum_{t=1}^{T} a_{nt} \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m} . \end{cases} \tag{3.14}$$

In the latter expression for updating $\boldsymbol{e}_n$, the element-wise operations are used, with the division and multiplication, the latter being the Hadamard product given by $\odot$. These expressions yield the well-known classical NMF. It is worth noting that in the case of the linear kernel, namely when the map $\Phi(\cdot)$ is the identity operator, the optimization problem (3.6) is equivalent to the minimization of the (half) Frobenius norm between the matrices $\boldsymbol{X}$ and $\boldsymbol{EA}$.

3.3.3.2 The polynomial kernel

The polynomial kernel is defined as $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = (\boldsymbol{z}^\top \boldsymbol{e}_n + c)^d$. Here, c is a nonnegative constant balancing the impact of high-order to low-order terms in the kernel. The kernel's gradient is given by:

$$\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{z}) = d\, (\boldsymbol{z}^\top \boldsymbol{e}_n + c)^{(d-1)} \boldsymbol{z}.$$

We consider the most common quadratic polynomial kernel with $d = 2$. Replacing $\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n, \boldsymbol{z})$ with this result, we obtain the additive update rules

$$\begin{cases} a_{nt} = \left(a_{nt} - \eta_{nt}\Big(\sum_{m=1}^{N} a_{mt}(\boldsymbol{e}_m^\top \boldsymbol{e}_n + c)^2 - (\boldsymbol{x}_t^\top \boldsymbol{e}_n + c)^2\Big)\right)_+; \\ \boldsymbol{e}_n = \left(\boldsymbol{e}_n - \eta_n \sum_{t=1}^{T} a_{nt}\Big(-2(\boldsymbol{x}_t^\top \boldsymbol{e}_n + c)\,\boldsymbol{x}_t + 2\sum_{m=1}^{N} a_{mt}(\boldsymbol{e}_m^\top \boldsymbol{e}_n + c)\,\boldsymbol{e}_m\Big)\right)_+, \end{cases}$$

and the multiplicative update rules

$$\begin{cases} a_{nt} = a_{nt} \times \dfrac{(\boldsymbol{x}_t^\top \boldsymbol{e}_n + c)^2}{\sum_{m=1}^{N} a_{mt}\,(\boldsymbol{e}_m^\top \boldsymbol{e}_n + c)^2}; \\ \boldsymbol{e}_n = \boldsymbol{e}_n \odot \dfrac{\sum_{t=1}^{T} a_{nt}(\boldsymbol{x}_t^\top \boldsymbol{e}_n + c)\boldsymbol{x}_t}{\sum_{t=1}^{T} a_{nt} \sum_{m=1}^{N} a_{mt}(\boldsymbol{e}_m^\top \boldsymbol{e}_n + c)\boldsymbol{e}_m}. \end{cases} \tag{3.15}$$

The division in the latter expression is also component-wise.

3.3.3.3 The Gaussian kernel

The Gaussian kernel is defined by $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \exp\left(\frac{-1}{2\sigma^2}\|\boldsymbol{e}_n - \boldsymbol{z}\|^2\right)$. In this case, its gradient is

$$\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = -\frac{1}{\sigma^2}\kappa(\boldsymbol{e}_n, \boldsymbol{z})(\boldsymbol{e}_n - \boldsymbol{z}).$$

The update rules of a_{nt} can be easily derived, in both additive and multiplicative cases. For the estimation of $\boldsymbol{e}_n$, the additive rule is

$$\boldsymbol{e}_n = \left(\boldsymbol{e}_n - \eta_n\Big(\frac{1}{\sigma^2}\sum_{t=1}^{T} a_{nt}\,\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)(\boldsymbol{e}_n - \boldsymbol{x}_t) - \frac{1}{\sigma^2}\sum_{t=1}^{T}\sum_{m=1}^{N} a_{nt}a_{mt}\,\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)(\boldsymbol{e}_n - \boldsymbol{e}_m)\Big)\right)_+.$$

As for the multiplicative algorithm, we split the corresponding gradient into the subtraction of two nonnegative terms. This is possible since all the matrices are nonnegative,

as well as the kernel values. We get the update rule:

$$\boldsymbol{e}_n = \boldsymbol{e}_n \odot \frac{\sum_{t=1}^{T} a_{nt}\Big(\boldsymbol{x}_t\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\boldsymbol{e}_n\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)\Big)}{\sum_{t=1}^{T} a_{nt}\Big(\boldsymbol{e}_n\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\boldsymbol{e}_m\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)\Big)}, \tag{3.16}$$

where the division is component-wise.

3.4 Extensions of KNMF

The above work provides a framework to derive extensions of the KNMF by including additional constraints and structural information. Several extensions are described in the following with constraints imposed on the endmembers and the abundances, typically motivated by the unmixing problem in hyperspectral imagery.

3.4.1 Constraints on the endmembers

Different constraints can be imposed on the endmembers, essentially to improve the smoothness of the estimates. It turns out that the derivatives, with respect to the abundances, of the unconstrained cost function $\mathcal{J}$ in (3.6) and the upcoming constrained cost functions are identical. Thus, the resulting update rules for the estimation of the abundances remain unchanged, as described with (3.12) for the additive scheme and (3.13) for the multiplicative scheme.

3.4.1.1 Smoothness with the ℓ_2-norm regularization

In the estimation of the basis elements $\boldsymbol{e}_n$, regular solutions with less variations are of interest, *e.g.*, less spiky [Piper et al., 2004]. This property is exploited by a smoothness constraint, as described next.

In the input space, this constraint can be formulated with the minimization of the norm of each endmember, namely $\frac{1}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|^2$. By combining this penalty term with the

cost function (3.6), we get

$$\mathcal{J}_{\text{2-norm}} = \frac{1}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2 + \frac{\lambda}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|^2.$$

The parameter λ controls the balance between the reconstruction accuracy (first term in the above expression) and the smoothness of all the endmembers $\boldsymbol{e}_n$ (second term).

To estimate the endmember $\boldsymbol{e}_n$, we consider the gradient of $\mathcal{J}_{\text{2-norm}}$ with respect to it, which yields the following additive update rule:

$$\boldsymbol{e}_n = \left(\boldsymbol{e}_n - \eta_n\Big(\sum_{t=1}^{T} a_{nt}\Big(\sum_{m=1}^{N} a_{mt}\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) - \nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)\Big) + \lambda\boldsymbol{e}_n\Big)\right)_+.$$

Using the split gradient method presented in Section 3.3.2.2, we get the corresponding multiplicative update rule. It turns out that one gets the same expressions as in the unconstrained case, with (3.14), (3.15) or (3.16), where the term $\lambda\,\boldsymbol{e}_n$ is added to the denominator.

Within the proposed KNMF framework, we can also consider a similar constraint in the feature space. The cost function becomes

$$\mathcal{J}^{\mathcal{H}}_{\text{2-norm}} = \frac{1}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2 + \frac{\lambda_{\mathcal{H}}}{2}\sum_{n=1}^{N}\|\Phi(\boldsymbol{e}_n)\|_{\mathcal{H}}^2. \tag{3.17}$$

The gradient with respect to $\boldsymbol{e}_n$ yields the additive update rule

$$\boldsymbol{e}_n = \left(\boldsymbol{e}_n - \eta_n\Big(\sum_{t=1}^{T} a_{nt}\Big(\sum_{m=1}^{N} a_{mt}\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) - \nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)\Big) + \lambda_{\mathcal{H}}\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_n)\Big)\right)_+.$$

Depending on the used kernel, the expression of the multiplicative update rule is similar to the one given in the unconstrained case, given in (3.14), (3.15) or (3.16), by adding the term $\lambda_{\mathcal{H}}\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_n)$ to the denominator. It is easy to see that, when dealing with the linear kernel where $\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_n) = \boldsymbol{e}_n$, the corresponding update rules are equivalent to the ones given with the constraint in the input space.

3.4.1.2 Smoothness with fluctuation regularization

Virtanen [2003] imposed smoothness by reducing the fluctuations between successive values. The resulting cost function of the KNMF with such constraint is

$$\mathcal{J}_{\text{fluct}} = \frac{1}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2 + \frac{\gamma}{2}\sum_{n=1}^{N}\sum_{l=2}^{L-1}|e_{ln} - e_{(l-1)n}|,$$

where γ is a tradeoff parameter and e_{ln} is the l-th entry of the vector $\boldsymbol{e}_n$. The derivative of the penalizing term with respect to e_{ln} equals to:

$$\begin{cases} +\gamma & \text{when } e_{ln} < e_{(l-1)n} \text{ and } e_{ln} < e_{(l+1)n}; \\ -\gamma & \text{when } e_{ln} > e_{(l-1)n} \text{ and } e_{ln} > e_{(l+1)n}; \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

Adopting the descent gradient scheme (3.11) and incorporating the above expression into $\nabla_{\boldsymbol{e}_n}\mathcal{J}$ given in (3.8), we can get the modified additive and multiplicative update rules for the endmembers estimation.

3.4.1.3 Smoothness with weighted-average regularization

Another smoothness regularization raised by [Chen and Cichocki, 2005] aims to reduce the difference between e_{ln} and a weighted average $\overline{e}_{ln} = \alpha\overline{e}_{(l-1)n} + (1-\alpha)e_{ln}$, where $\alpha \in]0\ 1[$ is a tunable parameter that determines the local smoothness range. For each endmember $\boldsymbol{e}_n$, this weighted average can be written in a matrix form as $\overline{\boldsymbol{e}}_n = \mathbf{T}\,\boldsymbol{e}_n$, where

$$\mathbf{T} = \begin{pmatrix} (1-\alpha) & 0 & \cdots & 0 \\ \alpha\,(1-\alpha) & (1-\alpha) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ \alpha^{L-1}(1-\alpha) & \cdots & \alpha\,(1-\alpha) & (1-\alpha) \end{pmatrix}.$$

For each $\boldsymbol{e}_n$, the cost function is defined as $\frac{1}{L}\|\boldsymbol{e}_n - \overline{\boldsymbol{e}}_n\|^2 = \frac{1}{L}\|(\mathbf{I}-\mathbf{T})\boldsymbol{e}_n\|^2$, where $\mathbf{I}$ is the identity matrix of appropriate size. By considering all N endmembers and introducing a regularization parameter ρ that controls the smoothing process, we get the cost function:

$$\mathcal{J}_{\text{av}} = \frac{1}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2 + \frac{\rho}{2L}\sum_{n=1}^{N}\|(\mathbf{I}-\mathbf{T})\boldsymbol{e}_n\|^2.$$

The gradient of the penalty term with respect to $\boldsymbol{e}_n$ takes the form $\rho\,\mathbf{Q}\boldsymbol{e}_n$, where $\mathbf{Q} = \frac{1}{L}(\mathbf{I}-\mathbf{T})^\top(\mathbf{I}-\mathbf{T})$. The additive update rule of the endmembers is easy to derive using the descent gradient method. The multiplicative update rule depends on the used kernel, with expressions similar to (3.14), (3.15) and (3.16), by adding the term $\rho\,\mathbf{Q}\boldsymbol{e}_n$ to the denominator.

3.4.1.4 Case of the Gaussian kernel

To sum up, consider for instance the Gaussian kernel. In this case, the multiplicative update rule of any of the aforementioned smoothness constraints takes the form

$$\boldsymbol{e}_n = \boldsymbol{e}_n \odot \frac{\displaystyle\sum_{t=1}^{T} a_{nt}\,\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)}{\displaystyle\sum_{t=1}^{T} a_{nt}\sum_{m=1}^{N} a_{mt}\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) + f(\boldsymbol{e}_n)},$$

where the division is component-wise. In this expression, the function $f(\boldsymbol{e}_n)$ can be $\lambda\boldsymbol{e}_n$, when dealing with the ℓ_2-norm regularization in the input space, or $\rho\,\mathbf{Q}\boldsymbol{e}_n$ when the weighted-average regularization is used. When the smoothness is operated in the feature space as given in (3.17), this function cancels since we have in this case $\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_n) = 0$. This property is common to all RBF kernels, since $\|\Phi(\cdot)\|^2$ is constant.

3.4.2 Constraints on the abundances

To satisfy a physical interpretation, two types of constraints are often imposed on the abundances, the sparseness and the spatial regularity. These constraints have no influence on the update rules for the endmembers estimation as given in Section 3.3. As a consequence, we shall study in detail the estimation of the abundances.

3.4.2.1 Sparseness regularization

Sparseness has been proved to be very attractive in many disciplines, namely by penalizing the ℓ_1-norm of the weight coefficients [Hoyer, 2004]. Typically for hyperspectral unmixing, each spectrum $\boldsymbol{x}_t$ can be represented by using a few endmembers, namely only a few abundances a_{nt} are nonzero. Since the latter are nonnegative, the ℓ_1-norm of

their corresponding vector is $\sum_{n=1}^{N} a_{nt}$. This leads to the following sparsity-promoting cost function

$$\mathcal{J}_{\text{sparse}} = \frac{1}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2 + \mu\sum_{t=1}^{T}\sum_{n=1}^{N} a_{nt},$$

where the parameter μ controls the tradeoff between the reconstruction accuracy and the sparseness level. By considering the derivative of $\mathcal{J}_{\text{sparse}}$ with respect to a_{nt}, the additive update rule is obtained as follows:

$$a_{nt} = \left(a_{nt} - \eta_{nt}\Big(\sum_{m=1}^{N} a_{mt}\,\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) - \kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \mu\Big)\right)_+.$$

To get the multiplicative update rule, we set the stepsize to

$$\eta_{nt} = \frac{a_{nt}}{\displaystyle\sum_{m=1}^{N} a_{mt}\,\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) + \mu},$$

which leads to

$$a_{nt} = a_{nt} \times \frac{\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)}{\displaystyle\sum_{m=1}^{N} a_{mt}\,\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) + \mu}.$$

3.4.2.2 Spatial regularization

Spatial regularization that favors spatial coherence is essential in many image processing techniques, as often considered in the literature with the total-variation (TV) penalty. This penalty was recently studied in [Iordache et al., 2012] for the linear unmixing problem in hyperspectral imagery. Motivated by this work, we derive in the following a TV-like penalty for incorporating spatial regularity within the proposed framework. It is worth noting that the derivations of the spatial regularization can be viewed as the application on the abundances of the method given in Section 3.4.1.3, by extending the one-direction smoothness (of e_{ln}) into the two-dimensional spatial regularization (of a_{nk}).

When transforming (*i.e.*, folding) a hyperspectral image of size $T = a \times b$ pixels into a matrix $\boldsymbol{X}$, the t-th column of $\boldsymbol{X}$ is filled with the (i,j)-th spectrum from the original image, with $i = \lceil \frac{t}{b} \rceil$ and $j = t - (i-1)b$, where $\lceil \cdot \rceil$ denotes the smallest integer greater

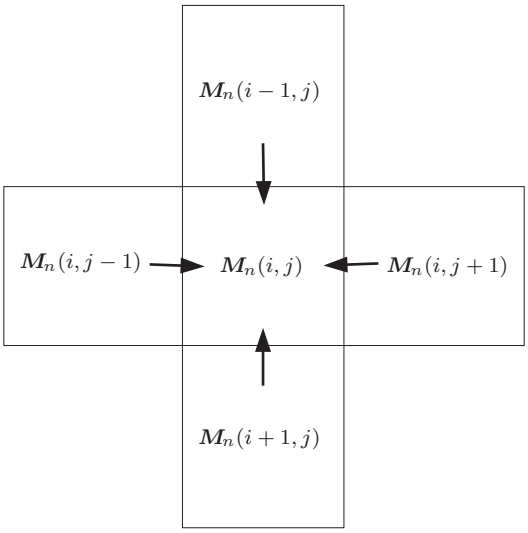


Figure 3.5: Schematic illustration of the spatial regularization.

than or equal to its argument. In the following, we denote by $\boldsymbol{M}_n$ the matrix of the n-th abundance defined by the entries $\boldsymbol{M}_n(i,j) = a_{nk}$, with $k = (i-1)b+j$ for $i = 1, 2, \ldots, a$ and $j = 1, 2, \ldots, b$. For any inner element $\boldsymbol{M}_n(i,j)$ belonging to the n-th abundance map, we shall use for spatial regularization the four geographical neighboring directions; cf. Figure 3.5.

The four spatial weighted averages of $\boldsymbol{M}_n(i,j)$ from its left, right, up and down sides are denoted as $\overline{\boldsymbol{M}_n}(i,j)_{\rightarrow}$, $\overline{\boldsymbol{M}_n}(i,j)_{\leftarrow}$, $\overline{\boldsymbol{M}_n}(i,j)_{\downarrow}$ and $\overline{\boldsymbol{M}_n}(i,j)_{\uparrow}$. They are expressed as follows:

$$\begin{cases} \overline{\boldsymbol{M}_n}(i,j)_{\rightarrow} = \alpha\overline{\boldsymbol{M}_n}(i,j-1)_{\rightarrow} + (1-\alpha)\boldsymbol{M}_n(i,j) \\ \overline{\boldsymbol{M}_n}(i,j)_{\leftarrow} = \alpha\overline{\boldsymbol{M}_n}(i,j+1)_{\leftarrow} + (1-\alpha)\boldsymbol{M}_n(i,j) \\ \overline{\boldsymbol{M}_n}(i,j)_{\downarrow} = \alpha\overline{\boldsymbol{M}_n}(i-1,j)_{\downarrow} + (1-\alpha)\boldsymbol{M}_n(i,j) \\ \overline{\boldsymbol{M}_n}(i,j)_{\uparrow} = \alpha\overline{\boldsymbol{M}_n}(i+1,j)_{\uparrow} + (1-\alpha)\boldsymbol{M}_n(i,j). \end{cases}$$

Rewriting in matrix form, we get

$$\begin{cases} \overline{\boldsymbol{M}}_n^{\top}(i,:)_{\rightarrow} = \boldsymbol{T}_{\rightarrow}\boldsymbol{M}_n^{\top}(i,:) \\ \overline{\boldsymbol{M}}_n^{\top}(i,:)_{\leftarrow} = \boldsymbol{T}_{\leftarrow}\boldsymbol{M}_n^{\top}(i,:) \\ \overline{\boldsymbol{M}}_n(:,j)_{\downarrow} = \boldsymbol{T}_{\downarrow}\boldsymbol{M}_n(:,j) \\ \overline{\boldsymbol{M}}_n(:,j)_{\uparrow} = \boldsymbol{T}_{\uparrow}\boldsymbol{M}_n(:,j), \end{cases}$$

where $\boldsymbol{T}_{\leftarrow} = \mathbf{T}_{\rightarrow}^{\top}$, $\boldsymbol{T}_{\uparrow} = \mathbf{T}_{\downarrow}^{\top}$, with

$$\boldsymbol{T}_{\rightarrow} = \begin{pmatrix} (1-\alpha) & 0 & \cdots & 0 \\ \alpha(1-\alpha) & (1-\alpha) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \alpha^{b-1}(1-\alpha) & \cdots & \alpha(1-\alpha) & (1-\alpha) \end{pmatrix},$$

and

$$\boldsymbol{T}_{\downarrow} = \begin{pmatrix} (1-\alpha) & 0 & \cdots & 0 \\ \alpha(1-\alpha) & (1-\alpha) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \alpha^{a-1}(1-\alpha) & \cdots & \alpha(1-\alpha) & (1-\alpha) \end{pmatrix}.$$

For each abundance $\boldsymbol{a}_n$, the associated cost function is:

$$\begin{aligned} R_n = \tfrac{1}{2} \sum_{i=1}^{a} \sum_{j=1}^{b} & \frac{\omega_l}{b} \|(\mathbf{I} - \boldsymbol{T}_{\rightarrow}) \boldsymbol{M}_n^{\top}(i,:)\|^2 + \frac{\omega_r}{b} \|(\mathbf{I} - \boldsymbol{T}_{\leftarrow}) \boldsymbol{M}_n^{\top}(i,:)\|^2 \\ & + \frac{\omega_u}{a} \|(\mathbf{I} - \boldsymbol{T}_{\downarrow}) \boldsymbol{M}_n(:,j)\|^2 + \frac{\omega_d}{a} \|(\mathbf{I} - \boldsymbol{T}_{\uparrow}) \boldsymbol{M}_n(:,j)\|^2, \end{aligned}$$

where $\omega_l, \omega_r, \omega_u$ and ω_d control the spatial effect ratios of left, right, up and down direction. In particular, $\omega_l = \omega_r = \omega_u = \omega_d$ denotes an average allocation of spatial effects. Considering the regularization term $\sum_{n=1}^{N} R_n$ for all N abundance maps, the cost function of the spatially-regularized KNMF is:

$$\mathcal{J}_{\text{spatial}} = \frac{1}{2} \sum_{t=1}^{T} \Big\| \Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n) \Big\|^2 + \sum_{n=1}^{N} R_n. \tag{3.19}$$

The update rule of the abundances for this cost function is obtained by locating a_{nt} in $\boldsymbol{M}_n$ using $a_{nt} = \boldsymbol{M}_n(i,j)$, with $i = \lceil \frac{t}{b} \rceil$ and $j = t - (i-1)b$. We get

$$\nabla_{a_{nt}} \Big(\sum_{n=1}^{N} R_n \Big) = \nabla_{\boldsymbol{M}_n(i,j)} R_n = \boldsymbol{G}(i,j),$$

where

$$\boldsymbol{G} = \omega_l \boldsymbol{M}_n \mathbf{Q}_{\rightarrow} + \omega_r \boldsymbol{M}_n \mathbf{Q}_{\leftarrow} + \omega_u \boldsymbol{M}_n^{\top} \mathbf{Q}_{\downarrow} + \omega_d \boldsymbol{M}_n^{\top} \mathbf{Q}_{\uparrow},$$

with

$$\begin{cases} \mathbf{Q}_{\rightarrow} = \frac{1}{b}(\mathbf{I} - \boldsymbol{T}_{\rightarrow})^{\top}(\mathbf{I} - \boldsymbol{T}_{\rightarrow}) \\ \mathbf{Q}_{\leftarrow} = \frac{1}{b}(\mathbf{I} - \boldsymbol{T}_{\leftarrow})^{\top}(\mathbf{I} - \boldsymbol{T}_{\leftarrow}) \\ \mathbf{Q}_{\downarrow} = \frac{1}{a}(\mathbf{I} - \boldsymbol{T}_{\downarrow})^{\top}(\mathbf{I} - \boldsymbol{T}_{\downarrow}) \\ \mathbf{Q}_{\uparrow} = \frac{1}{a}(\mathbf{I} - \boldsymbol{T}_{\uparrow})^{\top}(\mathbf{I} - \boldsymbol{T}_{\uparrow}). \end{cases}$$

By computing the gradient of (3.19) with respect to a_{nt}, namely $\nabla_{a_{nt}} \mathcal{J}_{\text{spatial}}$, we get the additive update rule for a_{nt}:

$$a_{nt} = \left(a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) - \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \boldsymbol{G}(i,j) \Big) \right)_+ ,$$

as well as the multiplicative update rule,

$$a_{nt} = a_{nt} \times \frac{\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)}{\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) + \boldsymbol{G}(i,j)},$$

where we have used the stepsize

$$\eta_{nt} = \frac{a_{nt}}{\sum_{m=1}^{N} a_{mt} \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) + \boldsymbol{G}(i,j)}.$$

3.5 Experiments

In this section, the relevance of the proposed KNMF and its extensions is studied on real hyperspectral images. The studied images are well-known hyperspectral images acquired by the AVIRIS. The raw images consist of 244 spectral bands, with the wavelength ranging from $0.4\mu m$ to $2.5\mu m$. The first image is a sub-image of 50×50 pixels taken from the well-known Cuprite image, where $L = 189$ spectral bands (out of 244) are of interest. The geographic composition of this area is known to be dominated by muscovite, alunite and cuprite, as investigated in [Clark et al., 1993]. The second image is a sub-image of 50×50 pixels from the Moffett Field image. This scene is known to consist of three materials: vegetation, soil and water. Before analysis, the noisy and water absorption bands were removed, yielding $L = 186$ spectral bands as recommended in [Dobigeon et al., 2008].

We introduce two criteria to evaluate the unmixing performance. The reconstruction error in the input space (RE) measures the mean distance between any spectrum and its reconstruction using the estimated endmembers and abundances, with

$$\mathrm{RE} = \sqrt{\frac{1}{TL}\sum_{t=1}^{T}\left\|\boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\boldsymbol{e}_t\right\|^2}.$$

Similarly, the reconstruction error in the feature space is

$$\mathrm{RE}^{\Phi} = \sqrt{\frac{1}{TL}\sum_{t=1}^{T}\left\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\Phi(\boldsymbol{e}_t)\right\|_{\mathcal{H}}^2}.$$

where

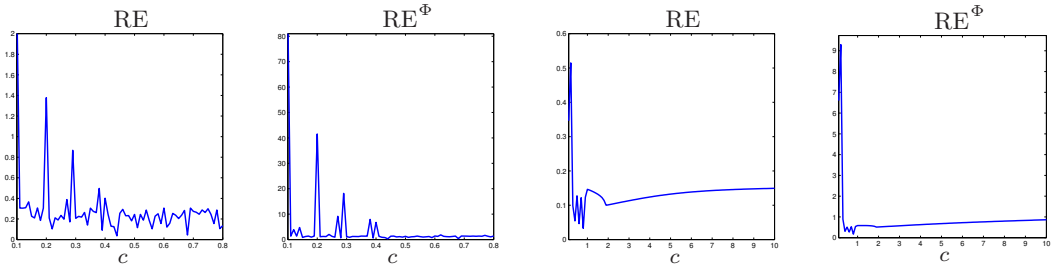
$$\left\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\Phi(\boldsymbol{e}_t)\right\|_{\mathcal{H}}^2 = \sum_{n=1}^{N}\sum_{m=1}^{N} a_{nt}a_{mt}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m) - 2\sum_{n=1}^{N} a_{nt}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \kappa(\boldsymbol{x}_t,\boldsymbol{x}_t).$$

3.5.1 State-of-the-art methods

As described in Section 1.2, many state-of-the-art hyperspectral unmixing algorithms either extract the endmembers (such as with VCA and N-Findr) or estimate the abundances (such as with FCLS, and nonlinear K-Hype and GBM-sNMF). In this case, solving the unmixing problem requires the joint use of two algorithms, one for endmember extraction and one for abundance estimation. The proposed KNMF estimates simultaneously the endmembers and the abundances, in the same spirit as some recently developed algorithms (such as MiniDisCo and ConvexNMF). We succinctly present all the comparing algorithms in the following.

Three supervised unmixing techniques are considered, where the endmembers are identified in prior using the endmember extraction technique VCA [Nascimento and Bioucas-Dias, 2005] presented in Section 1.2.1. Concerning the estimation of the abundances, we consider the techniques proposed for the linear mixing model, with FCLS [Heinz and Chang, 2001] as described in Section 1.3, and two nonlinear methods K-Hype [Chen et al., 2013b] and GBM-sNMF [Yokoya et al., 2014] presented in Section 1.4.

We also consider two non-kernel techniques that jointly extract the endmembers and estimate the abundances. The minimum dispersion constrained NMF (MiniDisCo) [Huck



(a) Cuprite image (b) Moffett image

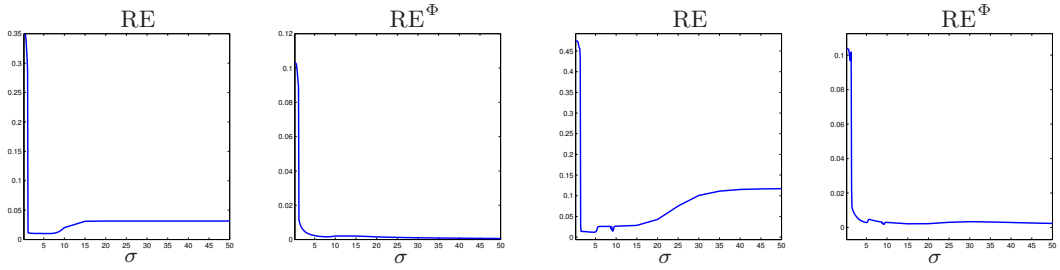
Figure 3.6: Influence on the reconstruction errors of the parameter c of the polynomial kernel for the KNMF with the multiplicative update rules.

et al., 2010] integrates the dispersion regularity into the NMF, by minimizing the variance of each endmember and imposing the sum of abundance fractions for every pixel to converge to 1. The resulting problem is solved with an alternate projected gradient scheme. In terms of convex optimization, the convex NMF (ConvexNMF) proposed in [Ding et al., 2010] restricts the basis matrix (endmember matrix in our problem) to a nonnegative linear combination of the samples, thus facilitating the interpretation.

Furthermore, we compare to other kernel-based NMF approaches. Proposed in [Li and Ngom, 2012], kernel convex-NMF (KconvexNMF) and kernel semi-NMF based on nonnegative least squares (KsNMF) are the kernelized versions corresponding respectively to the ConvexNMF in [Ding et al., 2010] and the alternating nonnegativity constrained least squares with the active set method in [Kim and Park, 2008]. Due to the curse of the preimage in the methods studied in [Zhang et al., 2006; Li and Ngom, 2012], neither the endmembers can be represented explicitly nor the reconstruction error can be evaluated. As opposed to these methods, the Mercer-based NMF introduced in [Pan et al., 2011] (MercerNMF) provides comparable results. It constructs a Mercer kernel that has a kernel map close to the one from the Gaussian kernel, under the nonnegative constraint on the embedded data. Conventional NMF is finally performed on these mapped data. It is noteworthy that learning the nonnegative embedding is computationally expensive.

3.5.2 Settings of the parameters

To provide comparable results, we estimate the optimal values of the parameters by conducting experiments on the KNMF with the multiplicative scheme (denoted by Poly$\odot$ and Gauss$\odot$, for the polynomial and Gaussian kernels, respectively), since these update rules do not depend on the stepsize parameter as in the case of the additive scheme



(a) Cuprite image (b) Moffett image

Figure 3.7: Influence on the reconstruction errors of the Gaussian bandwidth parameter σ for the KNMF with the multiplicative update rules.

(denoted by Poly$\oplus$ and Gauss$\oplus$, respectively). In order to explore the influence brought by the different regularizations to the unmixing performance, we use the same parameter values for the constrained extensions as in the KNMF. Note that the number of iterations is set to 200 for all experiments.

In the case of the polynomial kernel, we use the quadratic kernel with $d = 2$ since it is related to the generalized bilinear model as suggested in [Chen et al., 2013b]. The influence of the additive constant c in the kernel function is illustrated in Figure 3.6, yielding $c = 0.44$ for the Cuprite and $c = 0.72$ for the Moffett scene. A similar process determines the bandwidth parameter σ of the Gaussian kernel, employing the same set of candidate values $\{0.2, 0.3, \ldots, 9.9, 10, 15, 20, \ldots, 50\}$ for both images. The reconstruction errors are shown in Figure 3.7. Therefore, we fix $\sigma = 2.5$ and $\sigma = 3.3$ for the Cuprite and the Moffett images, respectively.

Concerning the stepsize parameter in the additive scheme, it is not only image-wise, but also involves a tradeoff between the estimation accuracy and the convergence rate.

3.5.3 Performance of the KNMF

Experiments are conducted on the linear (Lin$\oplus$/Lin$\odot$), polynomial (Poly$\oplus$/Poly$\odot$) and Gaussian (Gauss$\oplus$/Gauss$\odot$) kernels. The endmembers and the corresponding abundance maps estimated using these algorithms are shown in Figure 3.8 for the Cuprite image and in Figure 3.9 for the Moffett image. The efficiency of the KNMF is compared to the aforementioned well-known unmixing techniques, as presented in Table 4.5 in terms of the reconstruction errors in the input and feature spaces.

Table 3.1: Unmixing performance of the proposed KNMF

		Cuprite		Moffett	
		RE $\times 10^{-2}$	$\text{RE}^{\Phi} \times 10^{-2}$	RE $\times 10^{-2}$	$\text{RE}^{\Phi} \times 10^{-2}$
	FCLS	3.20	-	15.61	-
	K-Hype	2.12	-	5.27	-
	GBM-sNMF	0.98	-	2.09	-
	MiniDisCo	1.65	-	2.92	-
	ConvexNMF	1.61	-	2.58	
	KconvexNMF	-	25.64	-	35.95
	KsNMF	-	1.38	-	2.30
	MercerNMF	-	2.74	-	2.77
KNMF [this thesis]	Lin$\oplus$	0.96	0.96	2.90	2.90
	Lin$\odot$	**0.93**	0.93	**0.73**	0.73
	Poly$\oplus$	5.61	31.80	7.53	33.52
	Poly$\odot$	3.60	30.59	2.68	14.85
	Gauss$\oplus$	2.16	0.94	2.12	0.98
	Gauss$\odot$	1.05	**0.50**	1.24	**0.45**

Despite the fact that the linear kernel leaded to small reconstruction error in the input space, it does not outperform the Gaussian kernel in the feature space. As reflected in Figure 3.8, the inherent nonlinear correlation of the Cuprite image is revealed using the Gaussian kernel, which recognizes the three regions in the abundance maps; whereas the linear kernel is only capable to distinguish two regions. Considering the reconstruction error in the feature space, the unconstrained KNMF with the Gaussian kernel surpasses not only its linear and polynomial counterparts, but also all other methods including the kernel-based ones.

We also conduct an analysis on the different extensions. The results corresponding to the proposed regularizations are detailed in Figure 3.10 and Figure 3.11 for the smoothness of the endmembers, while constraints on the abundance maps are shown in Figure 3.12 for the sparseness regularization and Figure 3.13 for the spatial regularization. The influence of the regularization parameter γ in the smoothness with fluctuation regularization is shown in Figure 3.10 on an endmember estimated from the Cuprite image. Similarly, the influence brought by the weighted-average regularization is apparent as observed in Figure 3.11, where greater values of the regularization parameter ρ may over-smooth endmembers. The relevance of the sparsity is illustrated in Figure 3.12, where an increased value of the sparseness regularization parameter (up to $\mu = 2$) allows to better distinguish the regions in the abundance maps. Regarding the influence

of the spatial regularization, it is observed in Figure 3.13 on the Cuprite image that the larger the parameter ω is, the more homogeneous the abundance maps are.

3.6 Conclusion

In this chapter, we presented a new kernel-based framework for nonlinear NMF. The proposed approach provided a matrix decomposition where all the entries can be estimated, thanks to a model that circumvents the curse of the preimage problem. As a consequence to the hyperspectral unmixing task, it allows to estimate simultaneously the endmembers and abundances, as opposed to other kernel-based NMF where the endmembers cannot be extracted. Additive and multiplicative update rules were proposed with expressions depending on the used kernel functions, and several extensions were derived in order to incorporate constraints such as sparseness, smoothness and spatial regularity. The efficiency of these techniques was illustrated on well-known real hyperspectral images, with a comparative analysis using state-of-the-art techniques.

The proposed framework for KNMF opens the way to new developments and challenging issues. First, the algorithms developed in this chapter operate in a batch mode, namely, the entire data is processed at once. Such mode is inappropriate when dealing with large-scale and streaming data. In the succeeding chapter, we propose an online KNMF for an online setting, by providing algorithms using the stochastic gradient descent scheme. Secondly, one needs to choose the kernel function prior of applying the proposed KNMF. This corresponds in confronting for instance the linear kernel and the Gaussian one, and a fortiori the linear NMF in the input space and the KNMF in the corresponding feature space. This issue is investigated in Chapter 5 where we propose to combine the linear and kernel-based models and solve the corresponding optimization problem.

Although the proposed techniques to solve the KNMF are straightforward and simple to implement, the stationary point is not guaranteed in general (besides for the trivial linear kernel). This is due to the nonconvexity and nonlinearity of the optimization problem, which requires more elaborated optimization techniques to ensure that at least stationary points are attained. Some insights on this issue are given in Chapter 5 and a deeper analysis will be examined in future works.

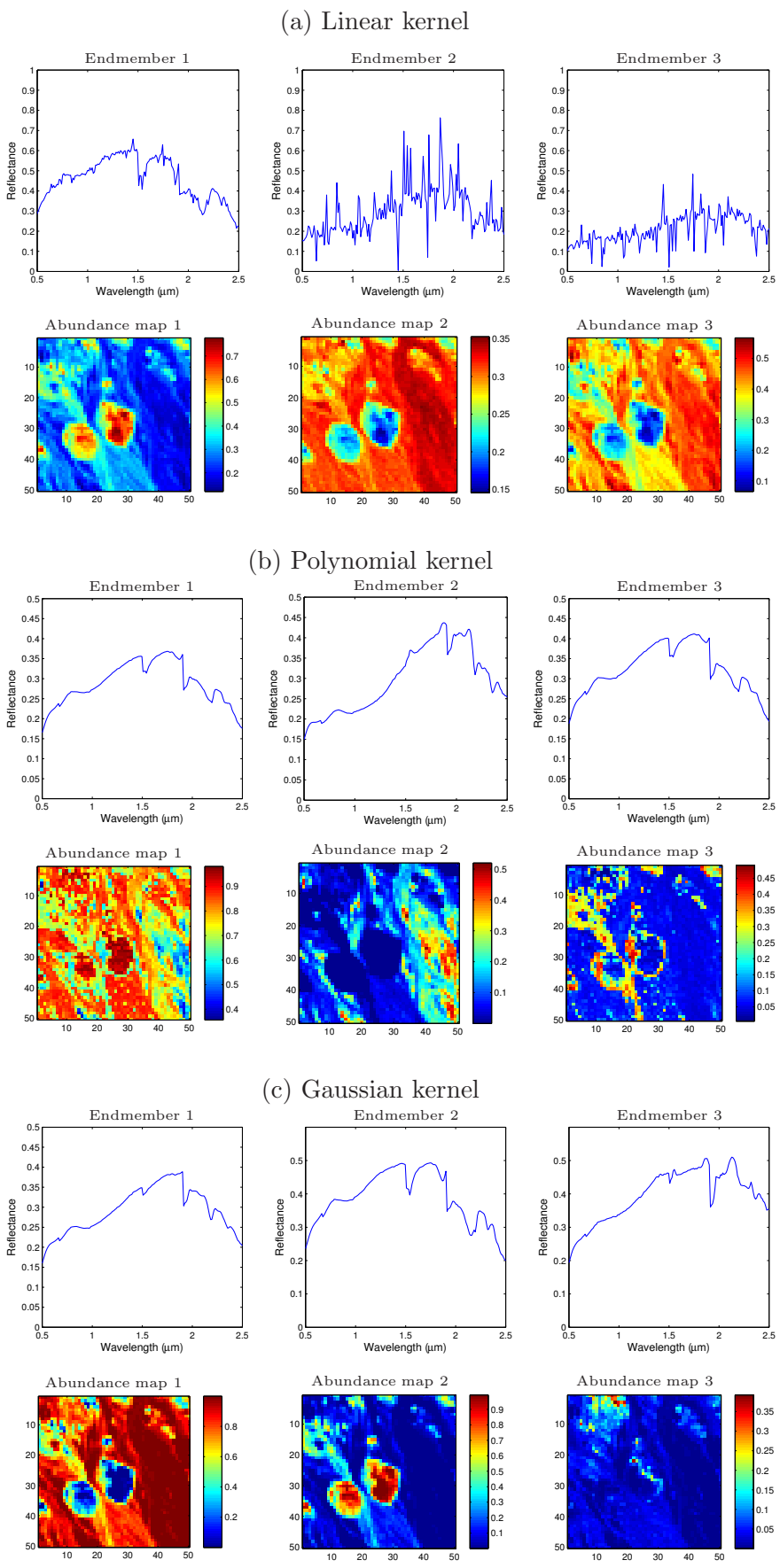


Figure 3.8: Cuprite image: Endmembers and corresponding abundance maps, estimated by the unconstrained KNMF with different kernels.

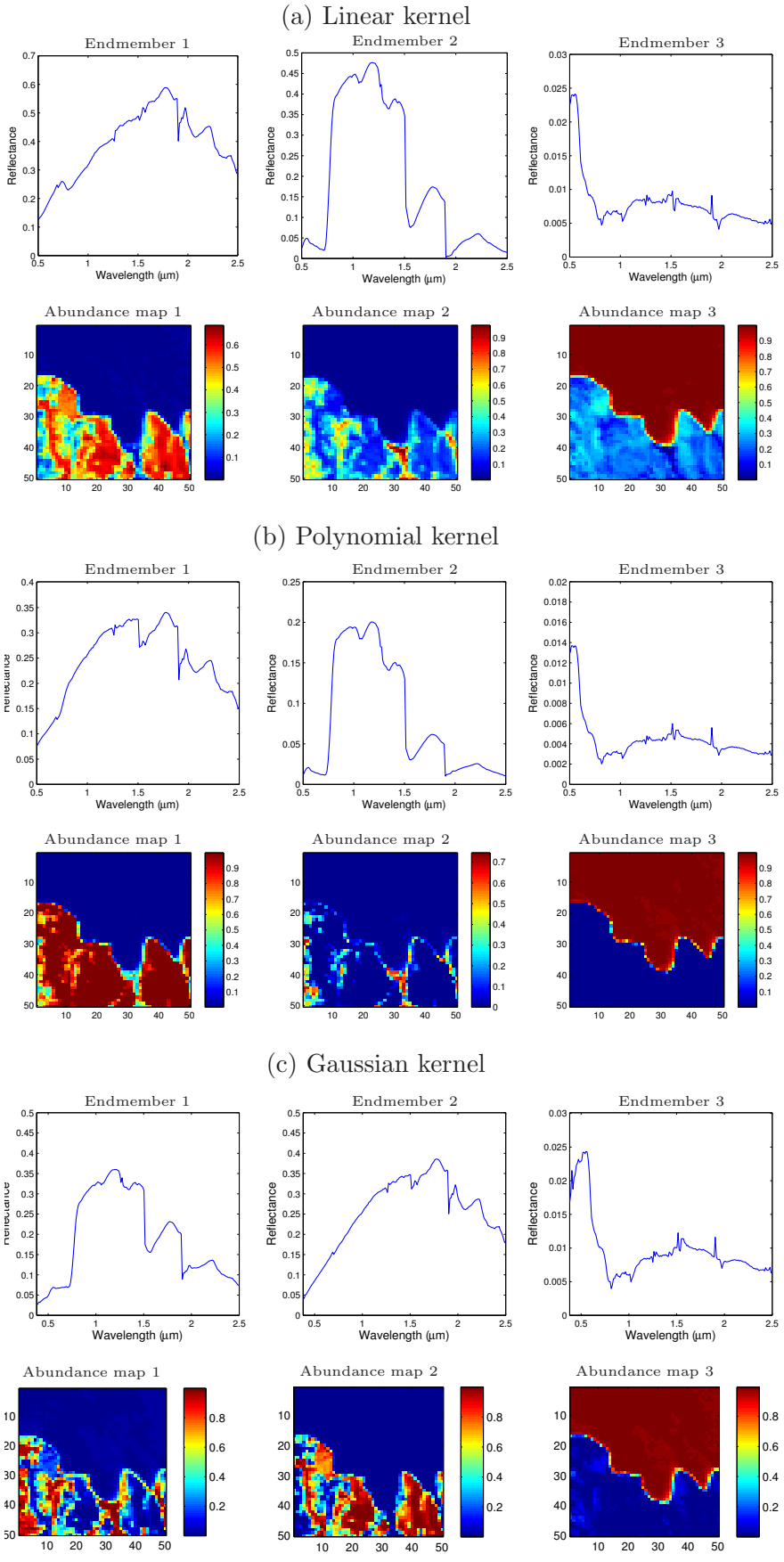


Figure 3.9: Moffett image: Endmembers and corresponding abundance maps, estimated by the unconstrained KNMF with different kernels.

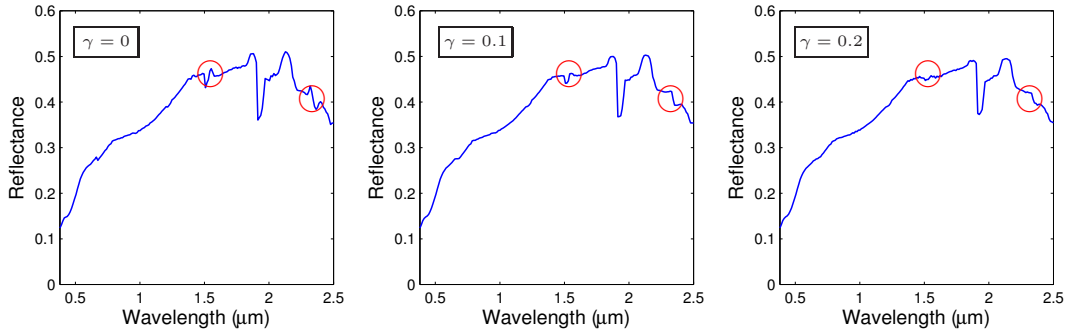


Figure 3.10: Influence of the smoothness with fluctuation regularization, illustrated on an endmember estimated from the Cuprite image, with different values of the regularization parameter γ.

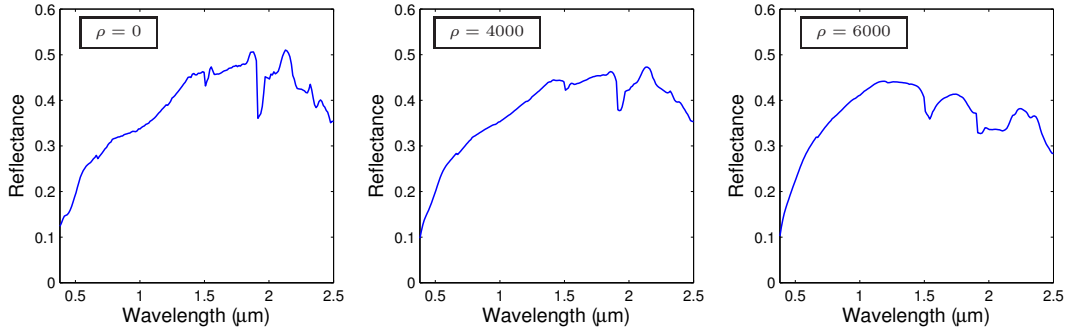


Figure 3.11: Influence of the weighted-average regularization, illustrated on an endmember estimated from the Cuprite image, with different values of the regularization parameter ρ.

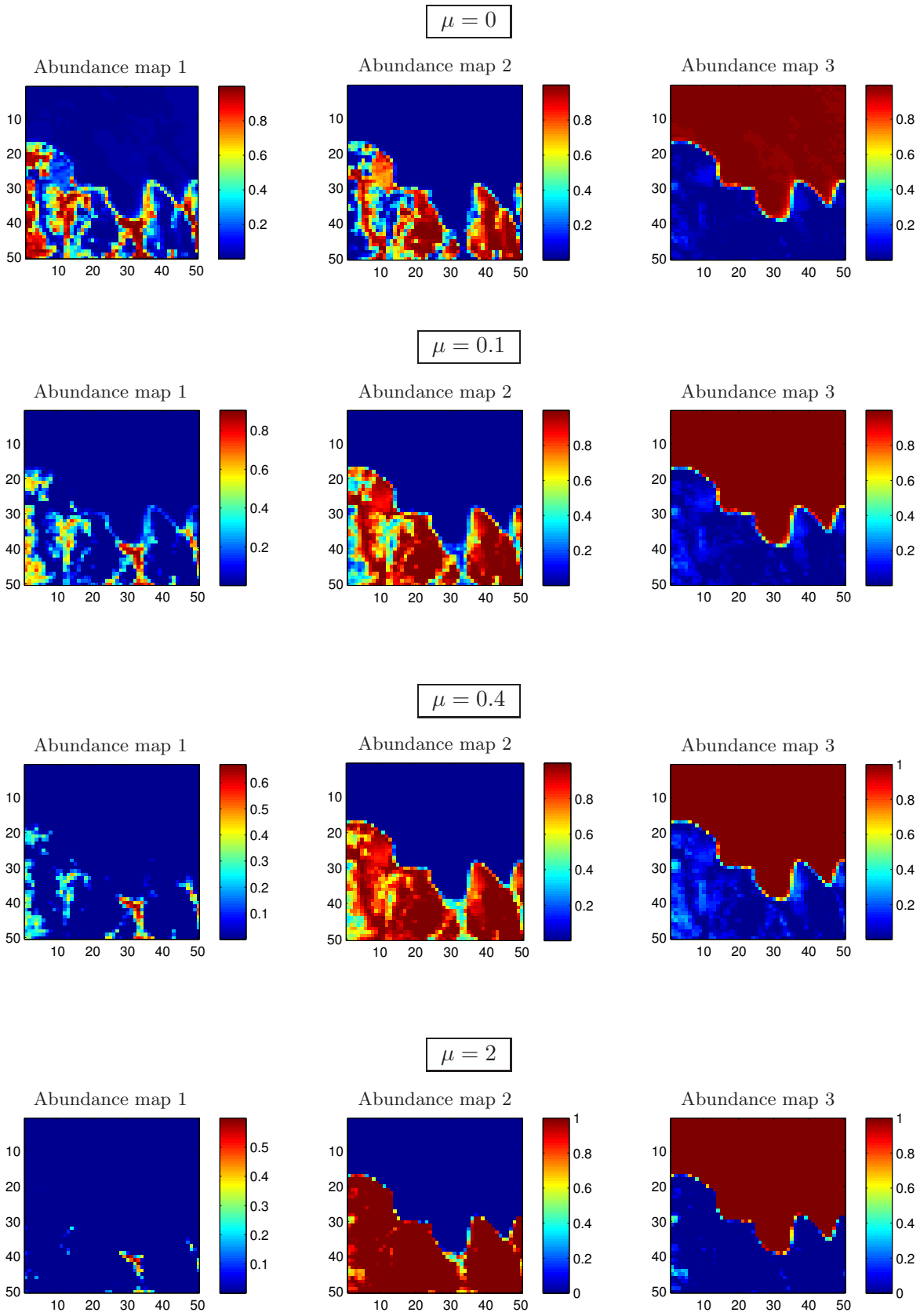


Figure 3.12: Influence of the sparseness regularization of the abundance maps for the Moffett image.

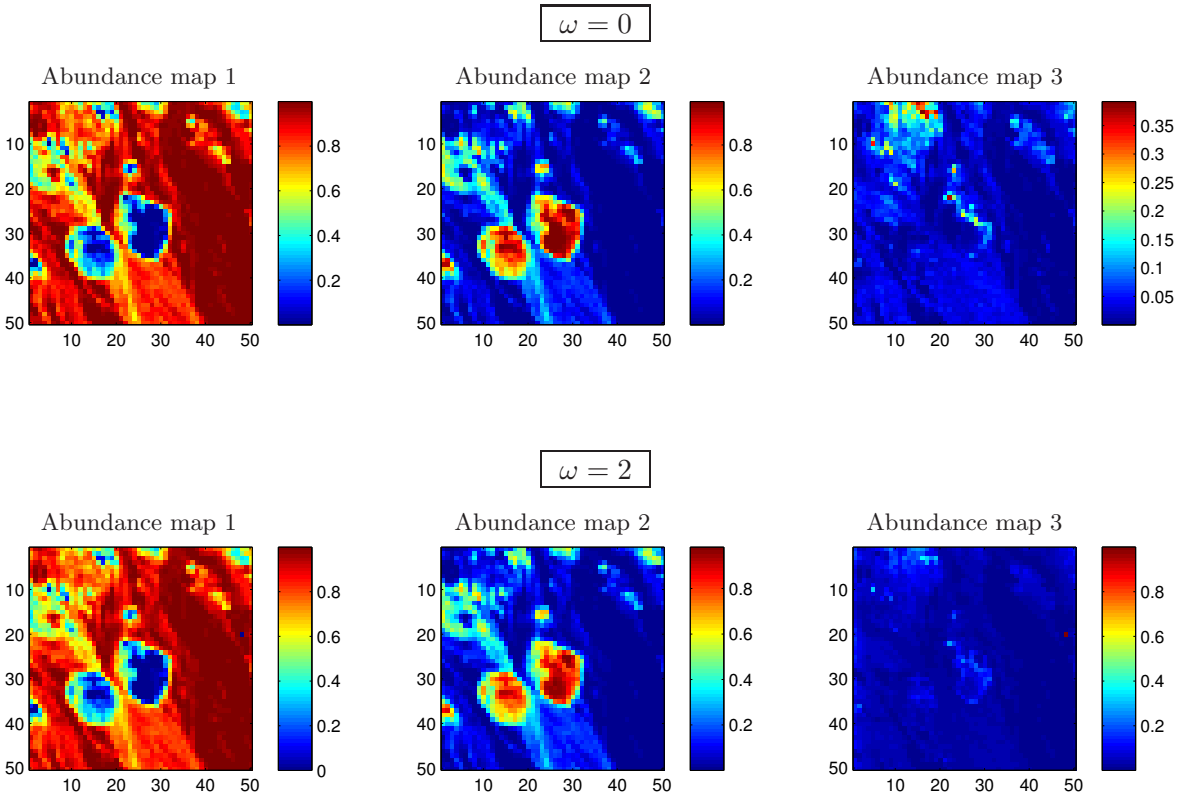


Figure 3.13: Influence of the spatial regularization of the abundance maps for the Cuprite image, with $\alpha = 0.5$.

Chapter 4

Online Kernel NMF

Contents

Addressing large-scale and streaming data is a challenging issue in data analysis. For this purpose, several online methods for NMF have been introduced recently, mainly restricted to the linear model. In this chapter, we propose a framework for online nonlinear NMF within the kernel-based framework described in the previous chapter. By exploring recent advances in the stochastic gradient descent and the mini-batch strategies, the proposed methods have a fixed complexity independent of the increasing number of samples. We derive several general updating rules, in both additive and multiplicative strategies, and present the case of the Gaussian kernel in detail. The performance of the proposed method is validated on unmixing synthetic and real hyperspectral images, by comparing to state-of-the-art online NMF techniques.

4.1 Introduction

To tackle the large-scale and streaming dynamic data, a couple of online NMF techniques have been proposed for the conventional linear NMF and its variants, as investigated in [Bucak and Gunsel, 2009; Wang et al., 2011; Guan et al., 2012; Cao et al., 2007]. For instance, online NMF with a constrained volume was presented in [Zhou et al., 2011b], while the projective online NMF (PONMF), analogue to its batch counterpart with an orthogonal and sparse representation, was proposed in [Yang et al., 2012; Wang and Lu, 2013]. Lefevre et al. [2011] studied an online version of the NMF with the Itakura-Saito divergence. To the best of our knowledge, published studies on online NMF have been mainly restricted to a linear model, whereas no online method exists for nonlinear, kernel-based, NMF.

In an online setting, the computational complexity of the algorithm remains a main concern to address. That is, the natural idea of performing sequentially batch NMF (or its variants) becomes inefficient and unfeasible, due to a time complexity proportional to the data number. To prohibit processing the whole data, early work presented in [Cao et al., 2007] factorizes the matrix composed by the previous basis matrix and novel samples. The incremental online NMF (IONMF) in [Bucak and Gunsel, 2009] makes the assumption that the encoding for the past samples is fixed, thereby alleviating the computational overhead. Since that published work, this assumption has been widely applied in online NMF algorithms, *e.g.*, [Wang et al., 2011; Guan et al., 2012; Zhou et al., 2011b], to name a few. Moreover, as the online NMF has a separable cost function with

respect to samples, Mairal et al. [2010] applied the stochastic gradient descent (SGD) strategy, which is a crucial complexity-reduction approach for online learning [Bottou, 2012]. This technique consists in substituting the real (but difficult to compute) gradient with the stochastic one, since the latter involves merely a single or a small subset of samples. In [Guan et al., 2012], the recent robust stochastic approximation technique is exploited for the linear model, where the SGD was improved with a smartly chosen stepsize and an average step on the results.

It is worth noting that the online NMF is closely related to the online dictionary learning and sparse coding, where the basic idea is to adaptably learn a dictionary from data, and to represent each sample as a sparse combination of the dictionary elements. The linear model considers an Euclidean least square loss function, with an ℓ_1-norm sparsity regularizer on the encoding vectors [Mairal et al., 2010]. Extensions include investigating an alternative Huber loss function as presented in [Wang et al., 2013b], and a nonlinear variation defined on a Riemannian manifold in [Ho et al., 2013].

In this chapter, we propose an online nonlinear NMF. In the same spirit of SGD, the batch KNMF described in Chapter 3 is extended to the online mode by keeping a tractable computational complexity. We provide the additive and multiplicative update rules of the general form for the basis matrix, and describe in more details the case of the Gaussian kernel. We also study the computational and memory complexity. Several extensions within this framework are discussed including sparse coding and smoothness of the basis vectors. The effectiveness of the proposed method is demonstrated on unmixing synthetic and real hyperspectral images.

4.2 Online KNMF (OKNMF)

Before introducing the online model, we succinctly review the KNMF in batch mode as proposed in Chapter 3. Consider the matrix factorization model $\boldsymbol{X}^{\Phi} \approx \boldsymbol{E}^{\Phi}\boldsymbol{A}$, or equivalently

$$\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n), \tag{4.1}$$

for all $t = 1, \ldots, T$. In a batch mode, namely when all the samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ are available at once, the cost function is

$$\mathcal{J}(\boldsymbol{E}, \boldsymbol{A}) = \frac{1}{2} \sum_{t=1}^{T} \left\| \Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n) \right\|_{\mathcal{H}}^2, \tag{4.2}$$

where the nonnegativity constraint is imposed on all the entries of the matrices $\boldsymbol{E}$ and $\boldsymbol{A}$. A two-block coordinate descent strategy was proposed in Chapter 3 to solve this constrained optimization problem in a batch mode.

In the following, we describe an online learning framework for KNMF. In the online setting, the samples arrive successively. An intuitive idea is to iteratively conduct the KNMF as in the batch mode. Unfortunately, as the samples number continuously grows, this approach suffers from an intractable computational complexity. By investigating the stochastic gradient, we propose an online KNMF (**OKNMF**) with a controlled computational complexity.

4.2.1 Problem formulation

Consider an online setting, where the samples are processed successively. Let $\boldsymbol{x}_k$ be the sample available at instant k. From (4.2), the cost function of the first k samples is

$$\mathcal{J}_k(\boldsymbol{E}_k, \boldsymbol{A}_k) = \frac{1}{2} \sum_{t=1}^{k} \left\| \Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n) \right\|_{\mathcal{H}}^2,$$

where $\boldsymbol{E}_k$ and $\boldsymbol{A}_k$ denote respectively the basis and encoding matrices obtained for the first k samples. We adopt the following assumption, initially proposed in [Bucak and Gunsel, 2009] and employed in online NMF methods such as [Wang et al., 2011; Guan et al., 2012; Mairal et al., 2010]: from instant k to instant $k+1$, the encoding vectors for the first k samples remain unchanged, *i.e.*, the matrix $\boldsymbol{A}_k$ is appended at each instant while keeping its entries unchanged: $\boldsymbol{A}_{k+1} = [\boldsymbol{A}_k \quad \boldsymbol{a}_{k+1}]$.

As a new sample $\boldsymbol{x}_{k+1}$ is available, one needs to estimate the new basis matrix $\boldsymbol{E}_{k+1}$, by updating $\boldsymbol{E}_k$, and the novel sample's encoding vector $\boldsymbol{a}_{k+1}$, to be appended to $\boldsymbol{A}_k$. Therefore, one estimates $\boldsymbol{E}_{k+1}$ and $\boldsymbol{a}_{k+1}$ by minimizing the following cost function,

subject to the nonnegativity constraints:

$$\begin{aligned}\mathcal{J}_{k+1}(\boldsymbol{E},\boldsymbol{A}) =& \frac{1}{2}\sum_{t=1}^{k+1}\Big\|\Phi(\boldsymbol{x}_t)-\sum_{n=1}^{N}a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2 \\ =& \frac{1}{2}\sum_{t=1}^{k}\Big\|\Phi(\boldsymbol{x}_t)-\sum_{n=1}^{N}a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2+\frac{1}{2}\Big\|\Phi(\boldsymbol{x}_{k+1})-\sum_{n=1}^{N}a_{n(k+1)}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2.\end{aligned} \tag{4.3}$$

It is easy to see that this cost function is expressed as a sum of sub-loss functions over data samples. By expanding this expression and removing the constant term $\frac{1}{2}\sum_{t=1}^{k+1}\kappa(\boldsymbol{x}_t,\boldsymbol{x}_t)$, the optimization problem becomes

$$\min_{\boldsymbol{a}_{k+1},\boldsymbol{E}}\sum_{t=1}^{k+1}\mathcal{L}(\boldsymbol{x}_t,\boldsymbol{a}_t,\boldsymbol{E}), \tag{4.4}$$

subject to the nonnegativity constraints, where the sub-loss function $\mathcal{L}(\boldsymbol{x}_t,\boldsymbol{a}_t,\boldsymbol{E})$ is

$$\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_{nt}a_{mt}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)-\sum_{n=1}^{N}a_{nt}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t).$$

In the following, we adopt an alternating technique to minimize the cost function in (4.4) over the unknown basis matrix $\boldsymbol{E}$ and encoding vector $\boldsymbol{a}_{k+1}$. While we consider the general form in the following, the case of the Gaussian kernel is described in more details in Section 4.2.4.

4.2.2 Basis matrix update

The gradient of (4.4) with respect to the vector $\boldsymbol{e}_n$ is:

$$\nabla_{\boldsymbol{e}_n}\mathcal{J}_{k+1}=\sum_{t=1}^{k+1}\nabla_{\boldsymbol{e}_n}\mathcal{L}(\boldsymbol{x}_t,\boldsymbol{a}_t,\boldsymbol{E}), \tag{4.5}$$

where

$$\nabla_{\boldsymbol{e}_n}\mathcal{L}(\boldsymbol{x}_t,\boldsymbol{a}_t,\boldsymbol{E})=a_{nt}\Big(\sum_{m=1}^{N}a_{mt}\,\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)-\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)\Big). \tag{4.6}$$

In this expression, $\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\cdot)$ denotes the gradient of the kernel with respect to its (first) argument $\boldsymbol{e}_n$. Expressions of the gradients of the most commonly-used kernels

are presented in Table 2.1. In the following, we derive additive and multiplicative update rules.

4.2.2.1 Additive update rules — SGD and ASGD

First, consider the projected gradient descent (PGD) update rule for KNMF, as presented in Section 3.3.2.1. The basis vectors $\boldsymbol{e}_n$ are updated according to

$$\boldsymbol{e}_n = \Big(\boldsymbol{e}_n - \eta_n \sum_{t=1}^{k+1} \nabla_{\boldsymbol{e}_n} \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})\Big)_+,$$

for $n = 1, \ldots, N$, where η_n is the stepsize parameter. Unfortunately, this rule cannot be considered in an online setting, since it deals with all the $k+1$ received samples and has a computational cost proportional to the number of samples.

The stochastic gradient descent (SGD) update alleviates this computational burden, by approximating the above gradient based on a single, randomly-chosen, sample $\boldsymbol{x}_t$ at each iteration, and is

$$\boldsymbol{e}_n = \Big(\boldsymbol{e}_n - \eta_n \nabla_{\boldsymbol{e}_n} \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})\Big)_+, \tag{4.7}$$

for $n = 1, ..., N$. Despite a drastically simplified procedure, the SGD asymptotically converges much slower than its batch mode counterpart [Bottou, 2012]. A compromise between these two modes is the mini-batch mode, which aggregates the gradients corresponding to a randomly picked set of samples. Let $\mathcal{I}$ be the subset of randomly picked samples employed for updating at each iteration. The update rule in the mini-batch mode takes the following form

$$\boldsymbol{e}_n = \Big(\boldsymbol{e}_n - \eta_n \sum_{\boldsymbol{x}_t \in \mathcal{I}} \nabla_{\boldsymbol{e}_n} \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})\Big)_+, \tag{4.8}$$

for $n = 1, ..., N$, where the mini-batch size, pre-fixed, is denoted in the following by $p = \mathrm{card}(\mathcal{I})$.

To accelerate the convergence of the SGD, we further consider an averaged stochastic gradient descent (ASGD) strategy, as initially proposed in [Polyak and Juditsky, 1992]. By averaging the results obtained by the SGD (with mini-batch) over iterations, the

ASGD is expressed as

$$\overline{\boldsymbol{e}_n^j} = \frac{1}{j - j_0} \sum_{j=j_0+1}^{j} \boldsymbol{e}_n^j,$$

or in the recursive form

$$\overline{\boldsymbol{e}_n^{j+1}} = (1 - \xi_j)\, \overline{\boldsymbol{e}_n^j} + \xi_j\, \boldsymbol{e}_n^{j+1}, \tag{4.9}$$

where j denotes the current iteration number, j_0 represents when to begin the averaging process (we set $j_0 = 1$), and $\xi_j = 1/\max(1, j - j_0)$ stands for the averaging rate [Xu, 2011; Bottou, 2012]. The theoretical results in [Polyak and Juditsky, 1992] show that the ASGD converges as good as the second-order SGD [Wang et al., 2011]. Whereas the latter needs the costly computation of the Hessian, the averaging in ASGD with (4.9) is implementation-friendly.

The stepsize parameters η_n should be appropriately set. One could be interested in revisiting the advanced optimization tools developed in the linear case, where the convexity of the loss function enables robust stochastic approximation [Guan et al., 2012] and second-order PGD [Wang et al., 2011], *i.e.*, the use of the approximate inverse of the Hessian as stepsize [Mairal et al., 2010]. However, the kernel-based loss function $\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})$ may be nonconvex in terms of $\boldsymbol{e}_n$, such as when the Gaussian kernel is used. Following recent theoretical results [Bottou, 2012; Xu, 2011], we adopt the stepsize $\eta_j = \eta_0(1 + \eta_0\lambda j)^{-1}$, equally for all the basis vectors. Hence, it starts at a predetermined value η_0 and diminishes asymptotically as $(\lambda j)^{-1}$, λ being a tunable parameter. According to [Bottou, 2012], this form of stepsize proves to be effective in SGD algorithms. Moreover, it leads to the best convergence speed when the loss function is convex, namely, its Hessian is positive-definite, with λ being the minimum eigenvalue of the Hessian and η_0 being a constant [Bottou, 2012; Xu, 2011]. Regardless of the possible nonconvex loss function under investigation, experiments show that such stepsize provides excellent results for the proposed OKNMF.

4.2.2.2 Multiplicative update (MU) rules

We present below the multiplicative update rules for the OKNMF, by revisiting the batch KNMF studied in Section 3.3.2.2. As opposed to the additive gradient descent update rules, the resulting methods lead to NMF with neither the projection/rectification to impose nonnegativity, nor the pain of choosing the stepsize parameter. To this end, we

split the gradient in (4.6) as the subtraction of two positive terms, denoted $\mathcal{G}^+(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})$ and $\mathcal{G}^-(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})$ such that

$$\nabla_{\boldsymbol{e}_n} \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) = \mathcal{G}^+(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) - \mathcal{G}^-(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}). \tag{4.10}$$

By setting the stepsize parameter as

$$\eta_n = \frac{\boldsymbol{e}_n}{\sum\limits_{\boldsymbol{x}_t \in \mathcal{I}} \mathcal{G}^+(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})},$$

this yields the following multiplicative update rule of the general form

$$\boldsymbol{e}_n = \boldsymbol{e}_n \odot \frac{\sum\limits_{\boldsymbol{x}_t \in \mathcal{I}} \mathcal{G}^-(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})}{\sum\limits_{\boldsymbol{x}_t \in \mathcal{I}} \mathcal{G}^+(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})}, \tag{4.11}$$

where the multiplication $\odot$ and the division are component-wise. Analogous to the aforementioned additive cases, three multiplicative update rules can be proposed, depending on the pre-defined value p of the number of samples investigated at each iteration:

- If $p = k+1$, all the samples are proceeded and (4.11) reduces to the multiplicative update rule for batch KNMF;
- If $p = 1$, then (4.11) uses a single randomly chosen sample, as with the stochastic gradient descent (4.7);
- If $1 < p < k+1$, then (4.11) operates with a mini-batch update of size equal to p, as its additive counterpart (4.8).

4.2.3 Encoding vector update

To estimate the encoding vector $\boldsymbol{a}_{k+1}$ for the newly available sample $\boldsymbol{x}_{k+1}$, the basis matrix $\boldsymbol{E}$ is fixed, as well as the previously estimated encoding vectors $\boldsymbol{a}_t$, for $t = 1, \ldots, k$. The optimization problem becomes

$$\min_{\boldsymbol{a}_{k+1} \geq \boldsymbol{0}} \frac{1}{2} \left\| \Phi(\boldsymbol{x}_{k+1}) - \sum_{n=1}^{N} a_{n(k+1)} \, \Phi(\boldsymbol{e}_n) \right\|_{\mathcal{H}}^2. \tag{4.12}$$

The kernel-based model is linear-in-the-parameters, as shown in (4.1) with respect to a_{nt}. As a consequence, we can investigate well-known algorithms from the classical least nonnegative least square (NNLS), such as the active set method [Lawson and Hanson, 1987] as implemented in [Wang et al., 2011; Guan et al., 2012], and the multiplicative update routine of NMF [Bucak and Gunsel, 2009; Zhou et al., 2011b].

Back to the cost function $\mathcal{J}_{k+1}(\boldsymbol{E}, \boldsymbol{A})$ in (4.3), its partial derivative with respect to $a_{n(k+1)}$ is

$$\nabla_{a_{n(k+1)}} \mathcal{J}_{k+1}(\boldsymbol{E}, \boldsymbol{A}) = \sum_{m=1}^{N} a_{m(k+1)}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) - \kappa(\boldsymbol{e}_n, \boldsymbol{x}_{k+1}).$$

Applying the gradient descent scheme yields the update rule

$$a_{n(k+1)} = a_{n(k+1)} - \eta'_n \nabla_{a_{n(k+1)}} \mathcal{J}_{k+1}, \tag{4.13}$$

for $n = 1, ..., N$, where η'_n denotes the stepsize parameter. Additionally, a rectification is necessary at each iteration in order to guarantee the nonnegativity of the entries in $\boldsymbol{a}_{k+1}$. By replacing the stepsize parameter η'_n in (4.13) with

$$\eta'_n = \frac{1}{\sum_{m=1}^{N} a_{m(k+1)}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)},$$

the multiplicative update rule for $a_{n(k+1)}$ can be expressed as

$$a_{n(k+1)} = a_{n(k+1)} \times \frac{\kappa(\boldsymbol{e}_n, \boldsymbol{x}_{k+1})}{\sum_{m=1}^{N} a_{m(k+1)}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)}, \tag{4.14}$$

for $n = 1, ..., N$. If the sum-to-one constraint needs to be satisfied, the resulting encoding vector can be divided at each iteration by its ℓ_1-norm, namely $\sum_{m=1}^{N} a_{m(k+1)}$.

4.2.4 Case of the Gaussian kernel

The update rules for a given kernel (belonging to but not restricted to the ones given in Table 2.1) can be derived, by appropriately replacing the expressions of $\kappa(\boldsymbol{e}_n, \boldsymbol{z})$ and $\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{z})$ in (4.5)-(4.6) for the SGD/ASGD algorithms, and splitting the gradient as in (4.10) for the MU algorithm. It is noteworthy that the trivial case of the linear kernel corresponds to the linear NMF in the batch mode, and to the IONMF [Bucak and Gunsel, 2009] in the online mode.

In the following, we describe in detail the derivation of the update rules for the Gaussian kernel, with $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \exp(\frac{-1}{2\sigma^2}\|\boldsymbol{e}_n - \boldsymbol{z}\|^2)$ and $\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = -\frac{1}{\sigma^2}\kappa(\boldsymbol{e}_n, \boldsymbol{z})(\boldsymbol{e}_n - \boldsymbol{z})$, for any $\boldsymbol{z} \in \mathcal{X}$. For the encoding vector update, the updating rule remains unchanged. For the basis matrix, the mini-batch SGD update (4.8) becomes

$$\boldsymbol{e}_n = \Big(\boldsymbol{e}_n - \frac{\eta}{\sigma^2}\sum_{\boldsymbol{x}_t\in\mathcal{I}} a_{nt}\big(\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)(\boldsymbol{e}_n - \boldsymbol{x}_t) - \sum_{m=1}^{N} a_{mt}\,\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)(\boldsymbol{e}_n - \boldsymbol{e}_m)\big)\Big)_+,$$

for $n = 1, ..., N$. The corresponding ASGD update is given by sequently addressing the above output of SGD with (4.9).

Splitting the gradient of the loss function $\nabla_{\boldsymbol{e}_n}\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E})$, as given in (4.10), yields the following two nonnegative terms:

$$\begin{cases} \mathcal{G}^+(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) = \dfrac{a_{nt}}{\sigma^2}\Big(\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)\boldsymbol{e}_n + \displaystyle\sum_{m=1}^{N} a_{mt}\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)\boldsymbol{e}_m\Big); \\ \mathcal{G}^-(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) = \dfrac{a_{nt}}{\sigma^2}\Big(\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)\boldsymbol{x}_t + \displaystyle\sum_{m=1}^{N} a_{mt}\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)\boldsymbol{e}_n\Big). \end{cases}$$

Setting the stepsize parameter as

$$\eta_n = \frac{\sigma^2\boldsymbol{e}_n}{\displaystyle\sum_{\boldsymbol{x}_t\in\mathcal{I}} a_{nt}\Big(\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)\boldsymbol{e}_n + \sum_{m=1}^{N} a_{mt}\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)\boldsymbol{e}_m\Big)},$$

leads to the following multiplicative update rule for $\boldsymbol{e}_n$:

$$\boldsymbol{e}_n = \boldsymbol{e}_n \odot \frac{\displaystyle\sum_{\boldsymbol{x}_t\in\mathcal{I}} a_{nt}\Big(\boldsymbol{x}_t\,\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\,\boldsymbol{e}_n\,\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)\Big)}{\displaystyle\sum_{\boldsymbol{x}_t\in\mathcal{I}} a_{nt}\Big(\boldsymbol{e}_n\,\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\,\boldsymbol{e}_m\,\kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)\Big)},$$

where the multiplication $\odot$ and the division are component-wise.

4.2.5 Complexity

We analyse the computational complexity in terms of time and memory usage for the proposed OKNMF framework, for the SGD, ASGD, and MU algorithms. Denote by

p the number of samples used at each update, namely $p = \text{card}(\mathcal{I})$, and assume $N \ll \min(L, k)$.

The time complexity for updating each basis vector $\boldsymbol{e}_n$ is $\mathcal{O}(pNL)$ per iteration, which holds for any commonly-used kernel listed in Table 2.1 due to a roughly equal time complexity of L for computing $\kappa(\cdot, \cdot)$. Thus, the total time complexity for the basis matrix update is $\mathcal{O}(ps_1N^2L)$, where N is the number of basis vectors and s_1 is the iteration number. In the online setting, the sequential batch update has an increasing value $\mathcal{O}(ks_1N^2L)$, k being the current number of available samples, and therefore becomes unrealistic in view of the streaming data. On the other hand, the case with $p = 1$ (namely SGD) shows poor convergence property in experiments, despite the lowest complexity $\mathcal{O}(s_1N^2L)$. The compromised mini-batch mode is most attractive not only for its fixed — tractable — complexity $\mathcal{O}(ps_1N^2L)$, but also for its satisfying performance in practice, as demonstrated with the experiments conducted in Section 4.4.2. The time complexity for the encoding vector update is $\mathcal{O}(s_2NL)$, s_2 being the iteration number. The total time complexity for the encoding matrix update remains unchanged, since the matrix is not modified but only appended.

The complexity in terms of memory usage is $\mathcal{O}(Lk + Nk)$ at instant k, by keeping in memory all the proceeded data and their encoding vectors. Since this quantity increases along with the sample number k, it becomes impractical when the data size is large. To alleviate this storage burden, a natural way is to retain in memory only the latest q samples with their encoding vectors. Termed "buffering strategy" in [Guan et al., 2012], this scheme can reduce the space complexity to a fixed value, with $\mathcal{O}(Lq + Nq)$.

4.3 Extensions of OKNMF

This section presents several extensions of the proposed OKNMF. We restrict the presentation to the most known regularizations, while other extensions can be conveniently incorporated into the OKNMF framework, in the same spirit as in Chapter 3 for the batch mode.

4.3.1 Sparse coding (sOKNMF)

As demonstrated in Section 3.4.2.1, sparseness is of particular interest in the hyperspectral unmixing problem, since it allows to represent each spectrum $\boldsymbol{x}_t$ with only few endmembers, namely only certain values of a_{nt} are non-zero, for $n = 1, \ldots, N$. Under the nonnegativity constraint, the ℓ_1-norm of the encoding vector $\boldsymbol{a}_t$ equals to $\sum_{n=1}^{N} a_{nt}$. It is clear that adding such regularization in the initial cost function brings no effect to the basis matrix updating, since it is independent of $\boldsymbol{e}_n$, $n = 1, ..., N$. By revisiting the optimization problem (4.12), a sparsity-promoting version is

$$\min_{\boldsymbol{a}_{k+1} \geq \boldsymbol{0}} \frac{1}{2} \left\| \Phi(\boldsymbol{x}_{k+1}) - \sum_{n=1}^{N} a_{n(k+1)} \, \Phi(\boldsymbol{e}_n) \right\|_{\mathcal{H}}^2 + \beta \sum_{n=1}^{N} a_{n(k+1)},$$

where the positive parameter β controls the tradeoff between the factorization accuracy and the level of sparsity. By taking the derivative with respect to $a_{n(k+1)}$ and appropriately choosing the stepsize, the multiplicative update rule of encoding vector $\boldsymbol{a}_{k+1}$ becomes

$$a_{n(k+1)} = a_{n(k+1)} \times \frac{\kappa(\boldsymbol{e}_n, \boldsymbol{x}_{k+1})}{\sum_{m=1}^{N} a_{m(k+1)} \, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) + \beta},$$

for $n = 1, ..., N$. It is easy to see the influence of sparse coding on the regularization of the update.

4.3.2 Smoothness of the basis vectors

As demonstrated in the previous chapter, the smoothness regularization is of great interest when dealing with hyperspectral unmixing, since the extracted bases need to be less "spiky" in order to get relevant endmembers. Different regularizations were proposed in Section 3.4.1 to promote the smoothness of the basis vectors in the batch KNMF.

Following the developments given in Section 3.4.1.1 and [Piper et al., 2004], a natural regularization is the commonly-used ℓ_2-norm penalty, which leads to

$$\mathcal{J}_{k+1}^{\text{2-norm}}(\boldsymbol{E}, \boldsymbol{A}) = \sum_{t=1}^{k+1} \mathcal{L}^{\text{2-norm}}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}),$$

where

$$\mathcal{L}^{\text{2-norm}}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) = \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) + \frac{\delta}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|^2.$$

The gradient of this cost function with respect to $\boldsymbol{e}_n$ is

$$\nabla_{\boldsymbol{e}_n}\mathcal{J}_{k+1}^{\text{2-norm}} = \sum_{t=1}^{k+1}\nabla_{\boldsymbol{e}_n}\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) + \frac{\delta}{2}\boldsymbol{e}_n.$$

Following a similar procedure described in Section 4.2 for the unregularized OKNMF, additive and multiplicative updating rules for the basis vectors can be easily derived.

Another way to promote smooth solutions for KNMF is described in Section 3.4.1.2 following the work in [Virtanen, 2003], by penalizing variations between successive values of the estimated vector, namely by minimizing $\sum_{l=2}^{L-1}|e_{ln} - e_{(l-1)n}|$ for all $n = 1, \ldots, N$. By adding such regularization term to the proposed OKNMF, the cost function becomes

$$\begin{aligned}\mathcal{J}_{k+1}^{\text{fluct}}(\boldsymbol{E}, \boldsymbol{A}) &= \sum_{t=1}^{k+1}\mathcal{L}^{\text{fluct}}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) \\ &= \sum_{t=1}^{k+1}\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{E}) + \frac{\gamma}{2}\sum_{n=1}^{N}\sum_{l=2}^{L-1}|e_{ln} - e_{(l-1)n}|.\end{aligned}$$

The derivative of the regularization term with respect to e_{ln} is given in expression (3.18).

Other smoothness promoting regularizations can be included as well to provide constrained OKNMF, such as the weighted-average penalty described in [Chen and Cichocki, 2005]; see Chapter 3 for more details. Due to the added regularization terms that are independent of the encoding vector $\boldsymbol{a}_{k+1}$, the update of the latter remains unchanged.

4.4 Experiments

In this section, the performance of the proposed algorithms is demonstrated on unmixing synthetic and real hyperspectral images. Four metrics are considered to evaluate the unmixing performance of the online NMF algorithms. The (root mean square) reconstruction error in the input space (RE) and in the feature space (RE^{Φ}) are defined in Section 3.5. When the ground-truth information is available, namely the real endmembers and abundances, the unmixing quality can be further evaluated by the averaged

spectral angle distance between endmembers (SAD) defined as

$$\text{SAD} = \frac{1}{N}\sum_{n=1}^{N} \arccos \frac{\boldsymbol{e}_n^\top \widehat{\boldsymbol{e}}_n}{\|\boldsymbol{e}_n\|\|\widehat{\boldsymbol{e}}_n\|}, \tag{4.15}$$

and the root mean square error of abundances (RMSE) defined as

$$\text{RMSE} = \sqrt{\frac{1}{NT}\sum_{t=1}^{T} \|\boldsymbol{a}_t - \widehat{\boldsymbol{a}}_t\|^2}. \tag{4.16}$$

4.4.1 State-of-the-art online algorithms

We first revisit several existing online algorithms, of the conventional linear NMF as well as its variations. Proposed in [Cao et al., 2007], a first approach for online NMF (**ONMF**) investigates the fact that, when updating the basis matrix, the old data matrix does not need to be available, but only the old basis matrix since they can be used to represent the old data. By using the full-rank decomposition theorem from linear algebra, this strategy significantly reduces the computational cost compared to re-running the conventional NMF. It turns out that it yields an inferior reconstruction accuracy, as demonstrated in many works, such as in [Guan et al., 2012] with face images and in our experiments with hyperspectral images. The trick of incremental subspace learning, initially proposed for principal component analysis, is revisited with incremental online NMF (**IONMF**) in [Bucak and Gunsel, 2009] to tackle the increasing complexity of the conventional NMF in online setting. The underlying assumption is that the new samples do not affect the past encoding vectors. Fixing the encoding vectors for the processed samples, this technique updates the two factorizing matrices with multiplicative updating rules, by incrementally aggregating the effects from the newly-available data. Wang et al. [2011] solves an optimization problem similar to IONMF by using additive updating rules. Therein, the update with first-order PGD is closely related to the work in [Mairal et al., 2010], where the online matrix factorization with sparse constraints on the encoding matrix is discussed, and NMF is viewed as a special case with supplementary nonnegativity constraints. Sharing the same basis matrix update scheme, the latter differs from the first-order PGD merely on the sparseness of the encoding vectors.

By replacing the stepsize in the first-order PGD by the approximation of the inverse Hessian, the second-order PGD (**HONMF**) is advocated in [Wang et al., 2011], and expected to outperform its first-order counterpart. Proposed in [Guan et al., 2012], the online NMF with robust stochastic approximation (**RSA**) benefits from the recent progress in choosing the stepsize and averaging over the results, with a convergence rate of $\mathcal{O}(1/\sqrt{K})$ guaranteed for the basis matrix update, where K is the iteration number.

While most works focus on the conventional linear model, a few online methods were developed for other variants of the NMF. In [Févotte et al., 2008], an online scheme is considered where the Itakura-Saito divergence is used as the measurement of dissimilarity between the input matrix and its approximation. The projective NMF, which decomposes $\boldsymbol{X}$ into the form $\boldsymbol{X} = \boldsymbol{E}\boldsymbol{E}^\top\boldsymbol{X}$, is extended to the online version in [Yang et al., 2012; Wang and Lu, 2013]. Similar with its batch counterpart, the online projective NMF (**PONMF**) yields orthogonal and sparse basis vectors. The volume-constrained online NMF in [Zhou et al., 2011b] adds the regularization term $\log|\det(\boldsymbol{E})|$ to the cost function, in order to enhance the uniqueness of the factorization. However, this technique is limited to the cases with a square basis matrix, *i.e.*, $L = N$.

4.4.2 Experiments with synthetic data

This section first presents the relevance of the SGD/ASGD/MU algorithms within the proposed OKNMF framework on two synthetic hyperspectral images. The performance is validated using a comparative analysis with state-of-the-art methods. Second, the performance of sparsity-promoting sOKNMF over the unregularized OKNMF is studied on synthetic data with sparse encoding matrices.

4.4.2.1 Performance of OKNMF

The performance of the proposed method is first evaluated by unmixing two synthetic hyperspectral images, each having 50 000 pixels. The $N = 3$ endmembers used for data generation are selected from the United States Geological Survey (USGS) digital spectral library used in [Bioucas-Dias and Nascimento, 2008], with each spectrum of $L = 224$ spectral bands. These spectra are shown in Figure 4.1. The first image is generated

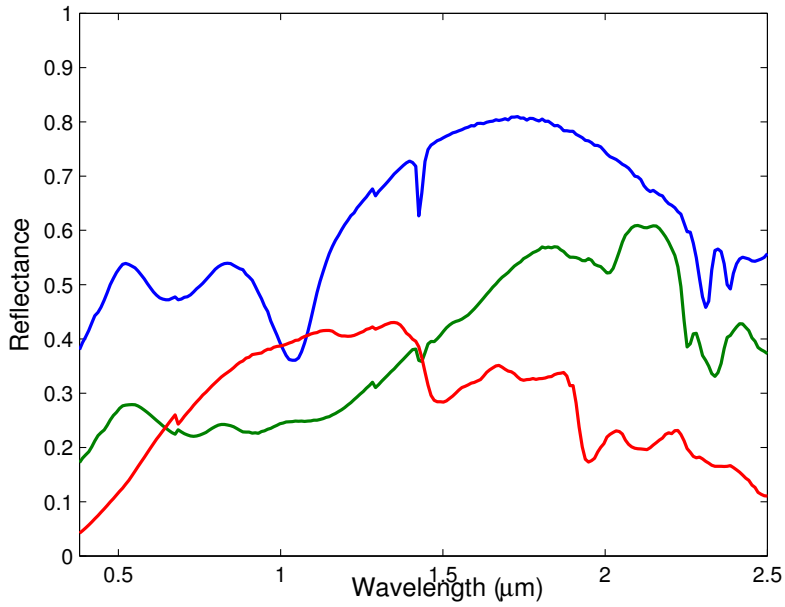


Figure 4.1: The USGS spectra used for synthetic images generation.

using the generalized bilinear model (GBM) defined in (1.6) with

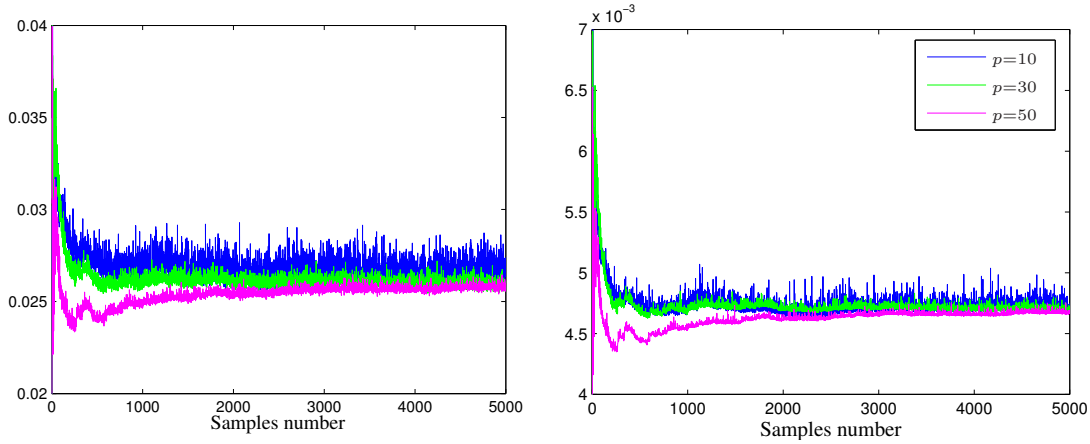
$$\boldsymbol{x}_t = \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \gamma_{ij,t}\, a_{it}\, a_{jt}\, (\boldsymbol{e}_i \odot \boldsymbol{e}_j) + \boldsymbol{n}_t,$$

where $\gamma_{nm} \in [0, 1]$ is generated from the uniform distribution. The second image is defined by using a polynomial post-nonlinear mixing model (PPNMM) given in (1.7) with

$$\boldsymbol{x}_t = \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n + b_t \left(\sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n \right) \odot \left(\sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n \right) + \boldsymbol{n}_t,$$

where the parameter b_t is uniformly generated within the range $[-0.3, 0.3]$ according to [Altmann et al., 2012]. For each studied image, the abundance values a_{nt} are uniformly generated within $[0, 1]$ and then normalized to meet the sum-to-one constraint. The images are corrupted by an additive Gaussian noise $\boldsymbol{n}_t \in \mathbb{R}^{L \times 1}$, with a signal-to-noise ratio SNR $= 30$ dB.

Experiments are conducted using the proposed algorithms, with the following settings. First, the bandwidth of the Gaussian kernel is set to $\sigma = 5.5$ for GBM image and $\sigma = 6.5$ for PPNMM image, as determined with the batch KNMF detailed in Chapter 3. Second, the mini-batch size is chosen within the form $p = \min\{\lceil \frac{k}{10} \rceil, m\}$. The value of this parameter allows to balance between the computational cost and the smoothness of the



(a) RE (b) RE^{Φ}

Figure 4.2: Influence of the value of the mini-batch size p on the reconstruction errors in the input and feature spaces, using the MU algorithm, with $p = 10$, 30, or 50. For illustration purpose, only the first 5000 samples from the synthetic PPNMM image are shown.

Table 4.1: Parameter Settings for the Synthetic Images

		GBM				PPNMM			
		σ	p	η_0	λ	σ	p	η_0	λ
OKNMF	SGD	5.5	30	0.25	2^{-8}	6.5	30	0.25	2^{-8}
	ASGD			2	2^{-9}			1	2^{-9}
	MU			-	-			-	-

convergence. To show its influence, we consider the MU algorithm (the other algorithms depend on the stepsize parameter, which makes difficult the characterization of the influence of p). As illustrated in Figure 4.2, a moderate value is close to 30. On one hand, small values of p such as 10 cause fluctuating convergence. On the other hand, an oversized $p = 50$ is computational expensive without anysignificant improvement due to the redundancy within the data. Last, concerning the algorithms SGD/ASGD with additive updates, the optimal values of η_0 and λ are determined with a 10-fold cross validation, on 1000 randomly selected pixels, using the candidate values $\eta_0 \in \{2^{-3}, 2^{-2}, \ldots, 2^{1}\}$ and $\lambda \in \{2^{-15}, 2^{-14}, \ldots, 2^{0}, 2^{1}\}$. Table 4.1 summarizes the parameter settings. The maximum number of iterations is set to 100.

The performance is validated using a comparative analysis with the aforementioned techniques: ONMF, IONMF, HONMF, RSA and PONMF. For the sake of fair comparison, we identically initialize the basis matrix $\boldsymbol{E}$ in all the compared algorithms, using the endmembers estimated by the NMF algorithm on a small subset of samples.

Table 4.2: Unmixing Performance for the Synthetic Images ($\times 10^{-2}$)

		GBM				PPNMM			
		SAD	RMSE	RE	RE^{Φ}	SAD	RMSE	RE	RE^{Φ}
	ONMF	100.39±2.75	48.87±0.02	66.56±28.51	6.68±0.00	104.66±18.3	48.82±0.04	79.52±65.8	6.68±0.00
	IONMF	③ 10.06±0.94	16.41±1.83	1.81±0.01	6.68±0.00	19.52±7.37	③ 15.06±2.29	③ 1.71±0.14	6.68±0.00
	HONMF	20.62±4.07	③ 14.47±4.97	② **1.57**±0.21	5.58±1.76	10.40±1.67	19.16±4.30	① **1.40**±0.01	4.68±0.85
	RSA	32.95±8.98	34.93±5.38	① **1.56**±0.19	635.38±0.04	33.96±8.63	35.56±5.11	② **1.54**±0.20	631.92±0.04
	PONMF	71.53±0.03	① **12.58**±0.01	③ 1.81±0.00	71.34±0.01	71.54±0.04	① **12.56**±0.01	1.79±0.00	70.46±0.01
OKNMF	SGD	12.48±2.42	17.17±2.77	2.51±0.19	③ 0.51±0.03	① **8.44**±3.46	19.70±2.71	2.65±0.13	② **0.45**±0.01
OKNMF	ASGD	① **9.19**±0.57	② **14.43**±1.30	2.25±0.01	① **0.47**±0.00	② **8.93**±1.70	② **15.01**±2.95	2.40±0.11	① **0.42**±0.01
OKNMF	MU	② **10.00**±1.48	17.08±4.89	2.38±0.13	② **0.49**±0.02	③ 9.42±3.18	18.72±3.85	2.60±0.08	③ 0.45±0.02

Considering the size of data, only 5 Monte-Carlo simulations are carried out for each algorithm. The resulting averages and deviations of the four aforementioned metrics are given in Table 4.2, where the smallest two values (often very close values) for each metric are highlighted.

The proposed OKNMF provides jointly the best averaged spectral angle distance between endmembers (SAD) and good root mean square error in terms of abundances (RMSE). The only competitive algorithm seems to be PONMF when dealing with the RMSE of abundances; however, the estimated endmembers are the worst with PONMF, up to eightfold compared to the proposed ASGD algorithm. The relevance of the jointly estimated endmembers and abundances can be measured with the reconstruction errors, RE and RE^{Φ}. The most accurate reconstruction is achieved by all the proposed algorithms (SGD/ASGD/MU) within the OKNMF framework, with the reconstruction error in the feature space. It is noticeable that these errors with RE^{Φ} are at least threefold lower than the ones obtained by all state-of-the-art algorithms using the linear model with RE. This means that the OKNMF (4.1) with the Gaussian kernel provides the most suitable factorization for the studied images, outperforming all other methods.

To compare the computational time of the online algorithms, experiments are conducted on the first 10000 pixels of the synthetic PPNMM image using a HP Intel® Core™ i7-3687U CPU at 2.10GHz computer. The MATLAB® (R2010) average implementation times per pixel in milliseconds are shown in Table 4.3. These results show that the SGD is twice faster than the state-of-the-art PONMF.

4.4.2.2 Performance of sOKNMF versus OKNMF

This section compares the extension with sparse coding (sOKNMF), presented in Section 4.3.1, with the unregularized OKNMF, on unmixing a series of synthetic data with

Table 4.3: Computational Time (ms/pixel)

	ONMF	6.7
	IONMF	23.7
	HONMF	30.1
	RSA	9.7
	PONMF	114.3
OKNMF	SGD	55.2
	ASGD	84.8
	MU	258.2

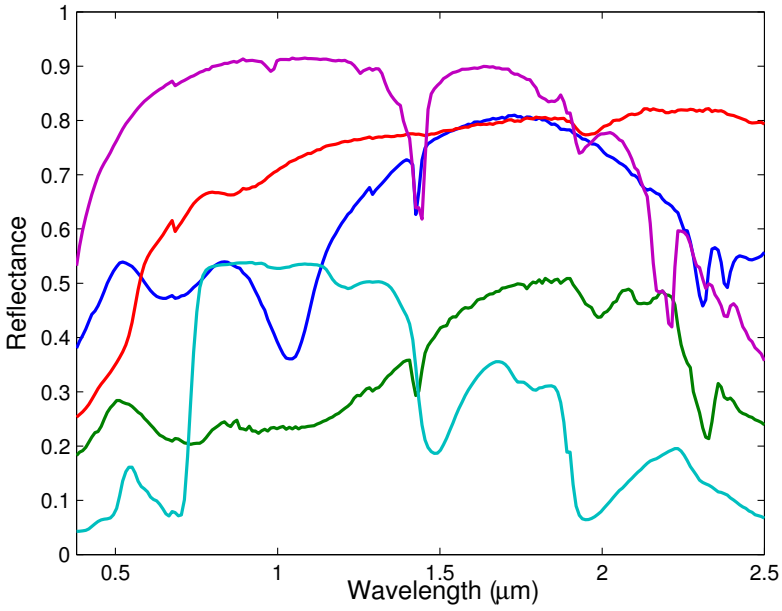
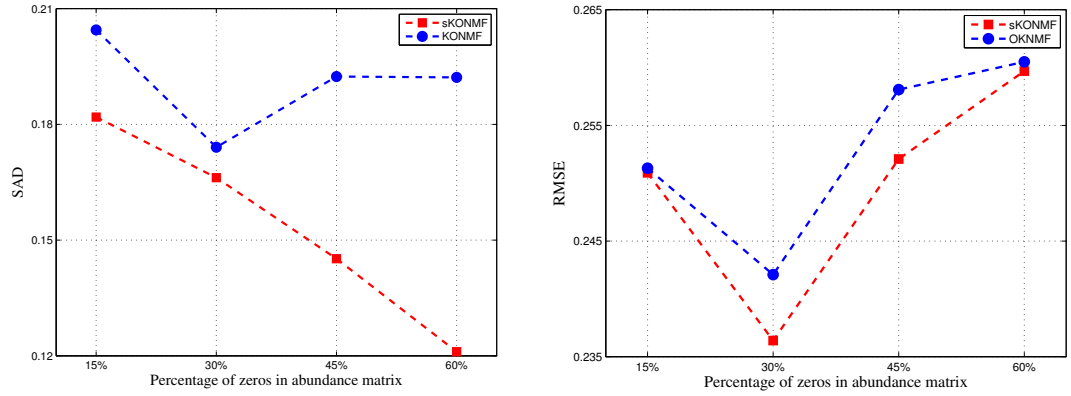


Figure 4.3: The USGS spectra used for synthetic images generation.

sparse abundance (encoding) matrix.

Four synthetic images, each of the size 50×50 pixels, are generated following the same settings of the aforementioned PPNMM model using five endmembers from the USGS digital spectral library, as shown in Figure 4.3. The four images have a different sparsity level on the abundances, set respectively to $s = 15\%$, 30%, 45% and 60%, and defined as follows: To impose the sparsity on the encoding matrix, a proportion of its entries are nullified, by ensuring at least one non-zero entry existing for each column. Concerning sOKNMF, it is noticeable that the parameter β should be tuned according to the sparsity of the unknown abundance (encoding) matrix. According to [Hoyer, 2004; Qian et al.,



(a) (b)

Figure 4.4: The averaged spectral angle distance (SAD, left figure) and the averaged root mean square error (RMSE, right figure) versus the percentage of zeros in the abundances, using OKNMF and sOKNMF. For the data with 15%, 30%, 45% and 60% zeros in abundance matrix, the results are archived respectively with $\beta = \hat{s} \times 10^{-1} = 0.024$, $\beta = \hat{s} \times 10^{-2} = 0.0037$, $\beta = \hat{s} \times 10^{-1} = 0.057$ and $\beta = \hat{s} \times 10^{-1} = 0.088$.

2011], a rough estimator from the input spectra is

$$\hat{s} = \frac{1}{\sqrt{L}} \sum_{l=1}^{L} \frac{\sqrt{T} - \|\boldsymbol{x}_{l*}\|_1 / \|\boldsymbol{x}_{l*}\|_2}{\sqrt{T} - 1},$$

where $\boldsymbol{x}_{l*}$ is the l-th row of $\boldsymbol{X}$, thus representing the l-th band over all the pixels. The set of candidate values for β is empirically set as $\beta \in \{0, \hat{s} \times 10^{-2}, \hat{s} \times 10^{-1}, \hat{s}\}$, with $\hat{s} = 0$ corresponding to the unregularized OKNMF. In all the experiments, the ASGD algorithm is applied with $\eta_0 = 2$ and $\lambda = 2 \times 10^{-8}$, the bandwidth parameter of the Gaussian kernel is set to $\sigma = 6.0$. The best results, averaged over ten Monte-Carlo simulations, in terms of SAM and RMSE are shown in Figure 4.4.

For all the synthetic images under study, the performance of OKNMF is improved by the sparsity-promoting sKONMF, in terms of both SAD and RMSE. These results are expected, since sOKMNF imposes a sparseness on the encoding vector, thus yielding more consistent solutions with the underlying sparse abundance matrices.

4.4.3 Experiments on real hyperspectral images

In order to study the performance of the proposed OKNMF by comparing it with the aforementioned state-of-the-art online NMF algorithms, we consider two well-known real hyperspectral images.

Figure 4.5: The RGB image of the Urban scene.

To provide a comparative analysis with results given in the previous chapter, the first image is the same as the one used in Section 3.5. The image is the Moffett image with 50×50 pixels.

To study the proposed OKNMF in a large-scale and streaming settings, the second image is the relatively big Urban image, as illustrated in Figure 4.5. It is available from the Hyperspectral Digital Imagery Collection Experiment (HYDICE), and contains 307×307 pixels. The original data is composed of $L = 210$ channels, with the wavelength ranging from $0.4\mu m$ to $2.5\mu m$. After removing the noisy bands, $L = 162$ clean bands are of interest. Widely-studied in the hyperspectral unmixing domain [Qian et al., 2011; Liu et al., 2011; Zhu et al., 2014], ground-truth information showed that this scene is composed at most of $N = 6$ endmembers, with asphalt road/parking, grass, tree, roof♯1, roof♯2/shadow, and concrete road as shown in Figure 4.6.

For both hyperspectral images, the parameter settings are obtained by performing a procedure similar to the one conducted for the synthetic images. Table 4.4 presents the used values. In practice, we simply retain for the Urban image the same values of the stepsize parameters (η_0, λ) as in the small Moffett image. Empirically, these values of

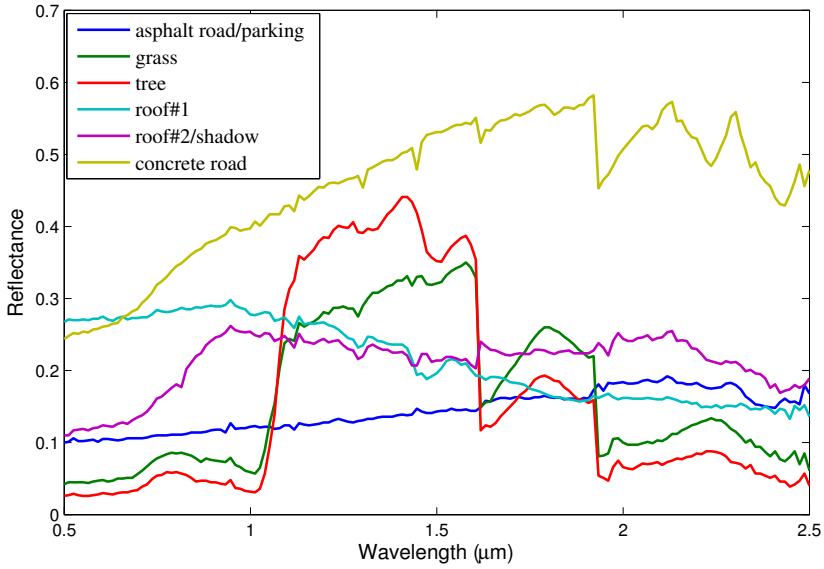


Figure 4.6: The six ground-truth endmembers in the Urban image.

Table 4.4: Parameter Settings for the Real Images

		Moffett				Urban			
		σ	p	η_0	λ	σ	p	η_0	λ
OKNMF	SGD	3.3	30	1	2^{-8}	3.0	30	1	2^{-8}
	ASGD			1	2^{-12}			1	2^{-12}
	MU			-	-			-	-

parameters perform well. Elaborated search techniques, such as cross-validation, achieve better accuracy, yet at a higher computational cost. Considering the data scale, ten Monte-Carlo simulations are carried out for each comparing algorithm for the Moffett image, and three Monte-Carlo simulations for the Urban image.

4.4.3.1 Performance comparison

The results in terms of reconstruction errors, RE and RE^{Φ}, are given in Table 4.5 for both hyperspectral images. Since ground-truth information on the endmembers is available for the Urban image, the accuracy of the estimated endmembers is further measured with the averaged spectral angle distance (SAD), as given in Table 4.6. For both images under study, the nonlinear model using the proposed online algorithms leads to the smallest reconstruction error in the feature space. Despite a relatively high reconstruction error in the input space, the nonlinear model results in the lowest SAD, namely it extracts

Table 4.5: Unmixing Performance for the Moffett and Urban Image ($\times 10^{-2}$)

		Moffett		Urban	
		RE	RE^{Φ}	RE	RE^{Φ}
	ONMF	11.94±3.49	7.36±0.01	231.48±23.23	7.88±0.01
	IONMF	① **0.82**±0.13	7.33±0.00	1.27±0.32	7.86±0.00
	HONMF	③ 1.03±0.69	9.60±3.69	② **0.87**±0.22	17.02±4.56
	RSA	1.21±0.01	84.66±0.12	① **0.69**±0.04	228.64±0.04
	PONMF	② **0.90**±0.02	23.82±0.93	③ 1.12±0.08	39.02±1.25
OKNMF	SGD	1.53±0.26	① **0.55**±0.08	3.16±0.15	① **0.95**±0.02
OKNMF	ASGD	1.54±0.21	② **0.57**±0.05	2.53±0.02	③ 1.02±0.01
OKNMF	MU	2.08±0.96	③ 0.93±0.27	2.45±0.07	② **1.01**±0.04

Table 4.6: Averaged spectral angle distance (SAD) for the Urban Image ($\times 10^{-2}$)

		asphalt	grass	tree	roof♯1	roof♯2	cr.road	**average**
	ONMF	-	-	-	-	-	-	≥ 133.75
	IONMF	53.45	46.84	41.38	24.47	③ 64.15	56.77	47.84
	HONMF	56.25	58.99	40.64	53.87	② **48.39**	47.93	51.09
	RSA	58.83	② **27.68**	③ 29.90	41.89	86.15	29.27	45.61
	PONMF	94.86	97.55	101.84	97.85	105.83	109.44	101.12
OKNMF	SGD	③ 35.63	42.06	51.27	③ 17.84	68.59	① **2.97**	③ 36.39
OKNMF	ASGD	① **29.02**	① **25.37**	① **6.93**	① **5.48**	71.32	② **3.41**	② **23.58**
OKNMF	MU	② **31.54**	③ 36.47	② **9.59**	② **15.92**	① **31.56**	③ 3.68	① **21.45**

the closest endmembers to the ground-truth by revealing the underlying nonlinearity in the image.

Figure 4.7 visualizes the estimated endmembers and their corresponding abundance maps for the Moffett image, and Figure 4.8 illustrates the estimate abundance maps for the Urban image. As observed, the proposed OKNMF is able to recognize regions that are the most consistent with the ground-truth, whereas the state-of-the-art techniques can only distinguish partly the regions while resulting in spiky/noisy endmembers and incoherent abundance maps.

4.5 Conclusion

This chapter presented a novel online kernel-based NMF method, termed OKNMF, to handle large-scale and streaming data. By exploiting stochastic gradient descent and mini-batch strategies in stochastic optimization, we developed additive and multiplicative update rules using the general kernel form, and detailed the case of the Gaussian kernel. The proposed methods maintain a fixed and tractable time complexity and

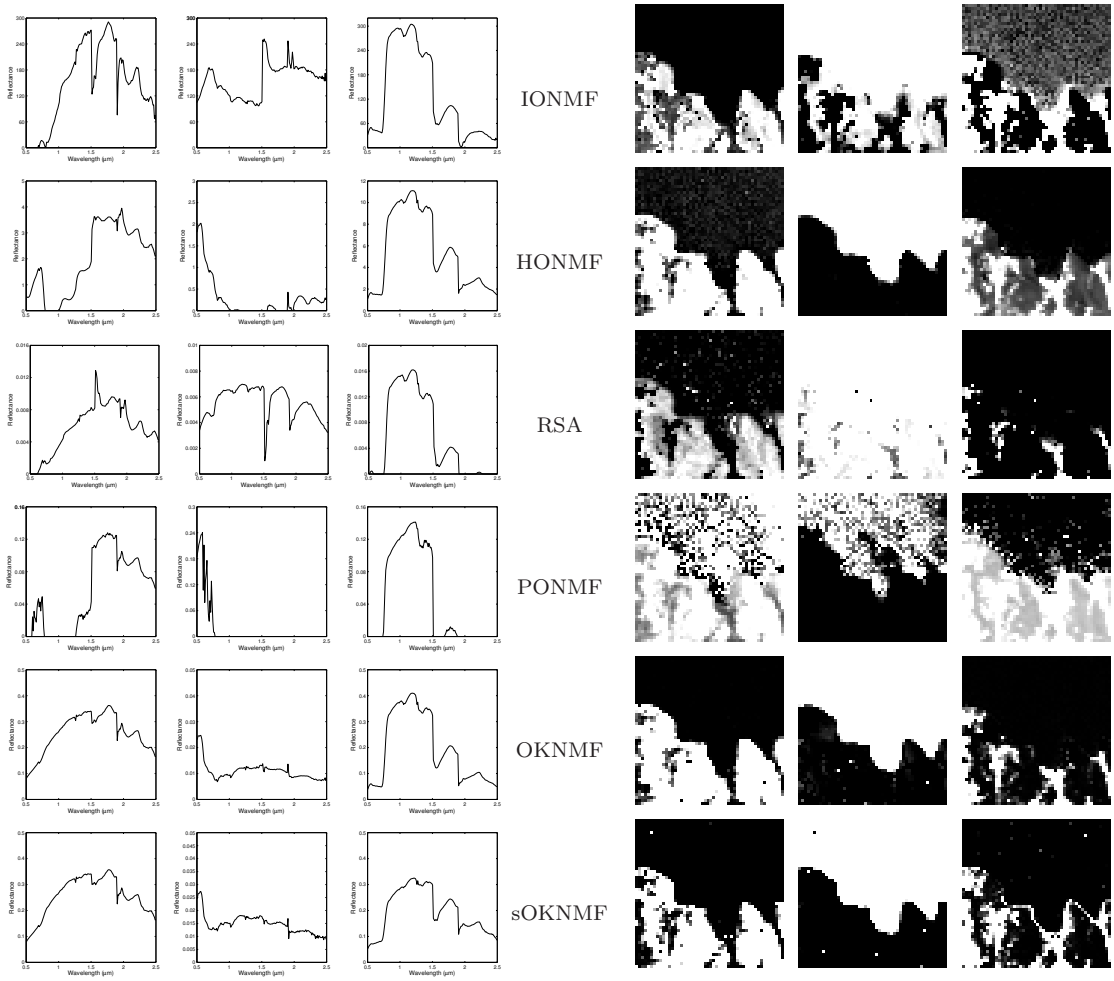


Figure 4.7: Left to right: estimated endmembers (soil, water, vegetation) and their corresponding abundance maps on the Moffett image. Top to bottom: IONMF, HONMF, RSA, PONMF, OKNMF with ASGD, sKONMF with $\beta = s \times 10^{-1} = 0.41$.

memory usage. Experimental results for unmixing synthetic and real hyperspectral images demonstrated the effectiveness of the proposed OKNMF. Not only it outperformed the state-of-the-art methods, but also the estimated endmembers are the closest to the ground-truth endmembers.

Future works include parallelization using GPU or distributing using computer clusters, as well as a further analysis on the convergence, which remains an open problem when dealing with nonconvex optimization problems. The study of more sophisticated optimization methods is of great interest, such as with limited-memory BFGS and conjugate gradient with line search.

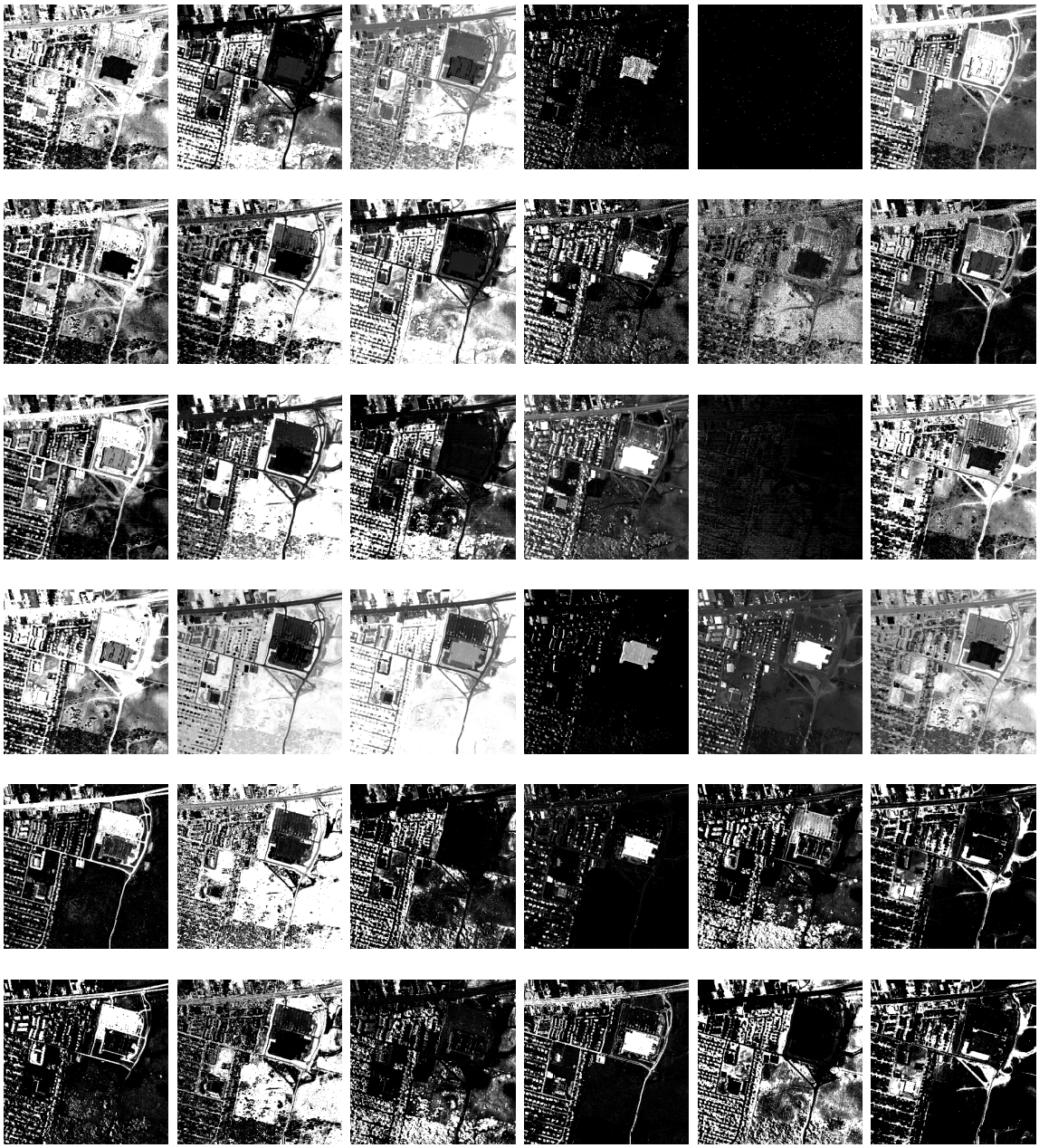


Figure 4.8: Estimated abundance maps on the Urban image. Left to right: asphalt road/parking, grass, tree, roof♯1, roof♯2/shadow, and concrete road. Top to bottom: IONMF, HONMF, RSA, PONMF, OKNMF with ASGD, sOKNMF with $\beta = s{\times}10^{-1} = 0.19$.

Chapter 5

Bi-objective KNMF: Linear versus Kernel-based Models

Contents

In this chapter, we revisit the KNMF as a multi-objective optimization problem, in particular a bi-objective one, where the objective functions defined in both input and feature spaces are taken into account. By taking the advantage of the sum-weighted method from the literature of multi-objective optimization, the proposed bi-objective KNMF determines a set of nondominated, Pareto optimal, solutions. Moreover, the corresponding Pareto front is approximated and studied. Experimental results on unmixing synthetic and real hyperspectral images confirm the efficiency of the bi-objective KNMF compared with the state-of-the-art methods.

5.1 Introduction

In either its linear conventional formulation or its nonlinear kernel-based formulation, as well as all of their variants, the NMF has been tackling a single-objective optimization problem. In essence, the underlying assumption is that it is known in prior that the linear model dominates the nonlinear one, or vice versa, for the data under study. To obtain such prior information about the given data is not practical in real-world applications. Moreover, it is possible that the combination of the linear and nonlinear models reveals the latent variables that are closer to the ground-truth than each single model considered alone. Independently from the NMF framework, such combination of the linear model with a nonlinear fluctuation was recently studied in [Chen et al., 2013b] with a nonlinearity depending only on the spectral content, and in [Chen et al., 2013c] with a nonlinearity defined by a post-nonlinear model. Within the same context, a multiple-kernel learning approach was studied in [Chen et al., 2012] and a Bayesian approach was investigated in [Altmann et al., 2014] with the so-called residual component analysis. While all these methods show the relevance of combining linear and nonlinear models, they share a major drawback: they only consist in estimating the abundances, while the endmembers need to be extracted in a pre-processing stage using any conventional linear technique (N-Findr, VCA, ... See Section 1.2.1). As opposed to such separation in the optimization problems, the NMF provides an elegant framework for estimating jointly the endmembers and the abundances. To the best of our knowledge, there has not been any study that combines the linear and nonlinear models within the NMF framework.

In this chapter, we study the bi-objective optimization problem that performs the NMF in both input and feature spaces, by combining the linear and kernel-based models. The

first objective function to optimize stems from the conventional linear NMF, while the second objective function, defined in the feature space, is derived from the kernel-based KNMF model proposed in Chapter 3. In case of two conflicting objective functions, there exists a set of nondominated, noninferior or Pareto optimal solutions. In order to acquire the Pareto optimal solutions, we investigate the sum-weighed method from the literature of multi-objective optimization, due to its ease for being integrated to the proposed framework. Moreover, we study the approximation of the corresponding Pareto front. Based on projected gradient descent scheme, the update rules are derived for the resulting sub-optimization problem when the feature space is induced by the Gaussian kernel. The complexity and the convergence of the algorithm are discussed, as well as the stopping criteria. Extensive experiments on synthetic and real hyperspectral images are conducted to study the relevance of solving such multi-objective optimization problem.

5.2 On Combining the Linear Model with a Nonlinear One

5.2.1 Augmenting the linear model with a nonlinearity

Several nonlinear models have been proposed within the hyperspectral unmixing scope, as outlined in Section 1.4 and reviewed in [Heylen et al., 2014; Dobigeon et al., 2014]. With few exceptions (such as the Mac-Mic mixing model that confronts the linear model to the nonlinear one), most of these nonlinear variations mainly consist in a combination of the linear model with an additive nonlinear term, as often advocated by a physical interpretation. These augmented linear models take the form

$$\boldsymbol{x}_t \approx \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n + \psi(\boldsymbol{E}, \boldsymbol{a}_t),$$

where ψ is an $\mathcal{X}$-valued nonlinear function, as detailed in the following. It is worth noting that the same abundances and endmembers intervene in both the linear and the nonlinear terms.

Bilinear models introduce bilinear mixtures of endmembers, such as the generalized bilinear model (GBM) [Halimi et al., 2011a] and the post-nonlinear mixing model [Altmann et al., 2012], as well as the GBM-based semi-NMF approach [Yokoya et al., 2014].

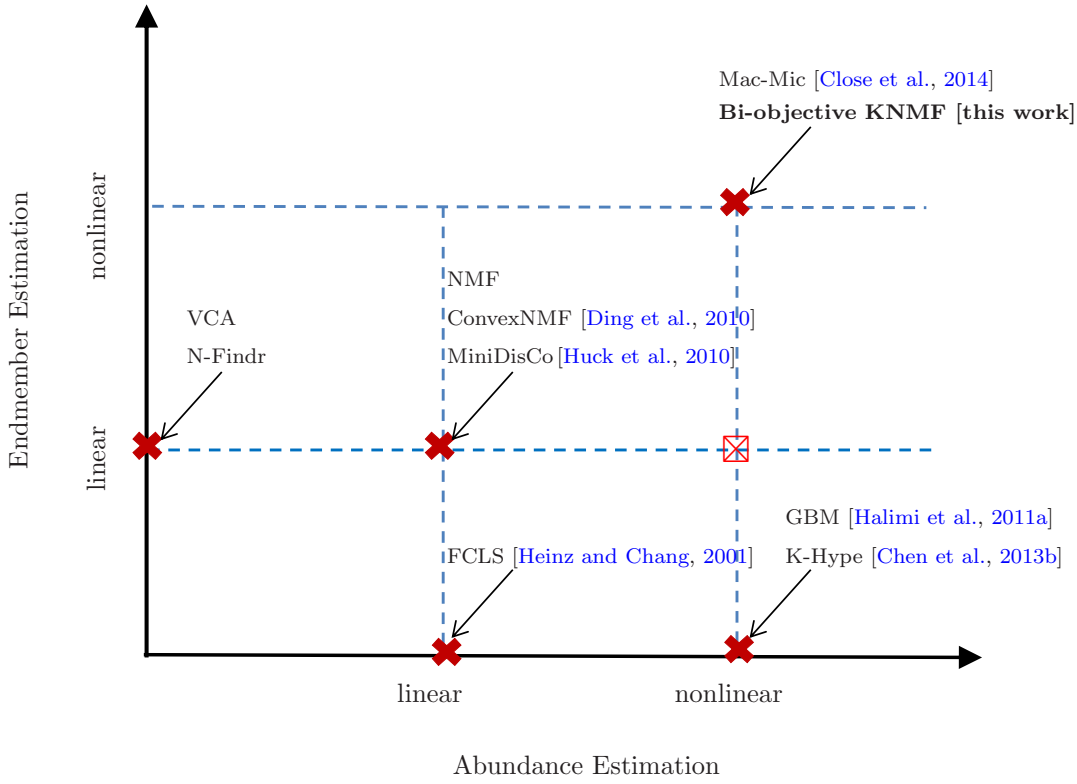


Figure 5.1: Schema illustrating linear versus nonlinear models, and single versus joint estimation. Marker ⊠ shows the combinations between VCA/N-Findr and GBM/K-Hype, for instance.

Several kernel-based models have been proposed to define the nonlinearity term ψ in some feature space. In [Chen et al., 2013b], the nonlinearity depends exclusively on the endmembers, namely $\psi(\boldsymbol{E})$. In [Chen et al., 2012], the above additive fluctuation is relaxed by considering a convex combination with multiple kernel learning. More recently, the abundances are incorporated in the nonlinear model, with a post-nonlinear model $\psi(\boldsymbol{E}\boldsymbol{a}_t)$ in [Chen et al., 2013c] and a Bayesian approach is used in [Altmann et al., 2014]. Another model is proposed in [Nguyen et al., 2013] in the context of supervised learning. All these methods consider that the endmembers $\boldsymbol{e}_n$ were already estimated using some linear technique such as N-Finder and VCA [Nascimento and Bioucas-Dias, 2005]; only the abundances are estimated with nonlinear models. Figure 5.1 presents a schematic illustration of these differences with respect to our work that is described in Section 5.3. See Section 3.3.1 for connections to the Mac-Mic [Hapke, 1981; Close et al., 2014].

5.2.2 On combining the linear NMF with a kernel-based one

The NMF allows to estimate simultaneously the endmembers and the abundances. It has been applied either in its linear model, *i.e.,* in the input space, or in a kernel-based formulation, *i.e.,* in the feature space. In the former as studied for instance in [Lee and Seung, 2001; Ding et al., 2010; Huck et al., 2010], each sample $\boldsymbol{x}_t$ is approximated with a linear combination of basis elements $\boldsymbol{e}_n$, by minimizing the distance in the input space between each $\boldsymbol{x}_t$ and $\widehat{\boldsymbol{x}}_t = \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n$, namely minimizing

$$\mathcal{J}_{\mathcal{X}}(\boldsymbol{E}, \boldsymbol{A}) = \frac{1}{2} \sum_{t=1}^{T} \left\| \boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n \right\|^2, \tag{5.1}$$

where the residual error is measured in the input space $\mathcal{X}$. In the kernel-based formulations conducted in [Zhang et al., 2006; Lee et al., 2009; Li and Ngom, 2012; Pan et al., 2011] (prior to our work described in Chapter 3), the basis elements $\boldsymbol{e}_n^{\Phi}$ belong to some kernel-induced feature space where the optimization occurs, by minimizing the distance between $\Phi(\boldsymbol{x}_t)$ and $\sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n^{\Phi}$.

To the best of our knowledge, there has not been any attempt to examine simultaneously linear and nonlinear NMF. This is mainly due to the fact that the endmembers are not the same in both representations. The linear endmembers are $\boldsymbol{e}_n \in \mathcal{X}$ while the nonlinear ones are $\boldsymbol{e}_n^{\Phi} \in \mathcal{H}$. As these endmembers belong to different spaces, a way to connect them is to map the latter to the input space, by estimating $\boldsymbol{e}'_n \in \mathcal{X}$ whose image $\Phi(\boldsymbol{e}'_n)$ is as close as possible to $\boldsymbol{e}_n^{\Phi}$. We fall once again in the curse of the pre-image problem as described in Section 2.4. Moreover, the issue here is a more difficult one, since the simultaneous optimization of the linear and nonlinear NMF yields two different sets of endmembers, $\boldsymbol{e}_n$ and $\boldsymbol{e}'_n$, without any connection between them which leads to difficult interpretation. For all these reasons, MercerNMF and KconvexNMF are not shown in Figure 5.1; while the underlying models are nonlinear, the endmembers cannot be estimated.

5.3 Bi-objective KNMF

In the previous chapters, we have defined a novel nonlinear model in the feature space using $\Phi(\boldsymbol{x}_t) \approx \widehat{\Psi}_t$, with

$$\widehat{\Psi}_t = \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n). \tag{5.2}$$

As a consequence, the endmembers $\boldsymbol{e}_n$ are estimated directly in $\mathcal{X}$. Under the nonnegativity of all $\boldsymbol{e}_n$ and a_{nt}, the optimization problem consists in minimizing the sum of the residual errors in the feature space $\mathcal{H}$, namely

$$\mathcal{J}_{\mathcal{H}}(\boldsymbol{E}, \boldsymbol{A}) = \frac{1}{2} \sum_{t=1}^{T} \left\| \Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n) \right\|_{\mathcal{H}}^2. \tag{5.3}$$

In this chapter, we combine the estimation of this model with the linear one. To this end, we minimize simultaneously $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$, namely the distance in the input space $\mathcal{X}$ between each $\boldsymbol{x}_t$ and $\widehat{\boldsymbol{x}}_t = \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n$, and the distance in the feature space $\mathcal{H}$ between $\Phi(\boldsymbol{x}_t)$ and $\widehat{\Psi}_t = \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n)$. The resulting problem is the bi-objective KNMF. In Section 5.4, we shall take advantage of the sum-weighted method to tackle this problem as a sequence of single-objective optimization problems, each corresponding to a fusion of the linear and nonlinear optimization problems, at different levels characterized by a parameter α, namely

$$\min_{\boldsymbol{E}, \boldsymbol{A}} \alpha\, \mathcal{J}_{\mathcal{X}}(\boldsymbol{E}, \boldsymbol{A}) + (1 - \alpha)\, \mathcal{J}_{\mathcal{H}}(\boldsymbol{E}, \boldsymbol{A}), \tag{5.4}$$

under the nonnegativity constraints.

5.3.1 Remarks on the physical interpretation

As opposed to augmenting the linear model with a nonlinearity (see Section 5.2.1), the proposed model is related to the Mac-Mic presented in [Close et al., 2014] (see Figure 5.1). Indeed, the latter confronts two models for each pixel, the linear model (called macroscopic) and the intimate mixing model (called microscopic) defined in Section 1.4.2. The proposed bi-objective KNMF can be also viewed as confronting two models, a "regularized" linear model and a "regularized" nonlinear one. One way to understand this property is through two complementary viewpoints of the bi-objective

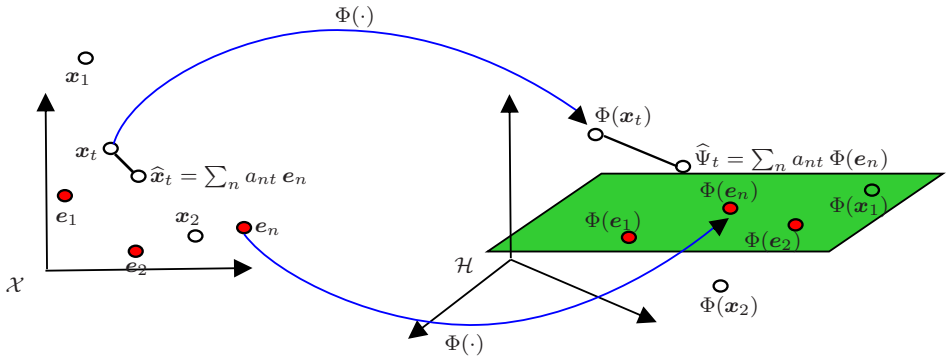


Figure 5.2: In the linear NMF, each sample $\boldsymbol{x}_t$ is approximated by $\widehat{\boldsymbol{x}}_t$ in the input space $\mathcal{X}$, while in the KNMF, the mapped sample $\Phi(\boldsymbol{x}_t)$ is approximated by $\widehat{\Psi}_t$ in the feature space $\mathcal{H}$. The proposed bi-objective KNMF solves simultaneously the two optimization problems.

optimization problem (5.4). In the first one, the investigated model is $\boldsymbol{x}_t \approx \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n$ (results from minimizing $\mathcal{J}_{\mathcal{X}}$), while the minimization of $\mathcal{J}_{\mathcal{H}}$ operates as a regularization, namely

$$\min_{\boldsymbol{E},\boldsymbol{A}} \sum_{t=1}^{T} \left\| \boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n \right\|^2 + \alpha'\, \mathcal{R}eg_{\mathcal{H}}(\boldsymbol{E}, \boldsymbol{A}),$$

where $\alpha' = 1/\alpha - 1$ controls the tradeoff between the fitness and the regularity of the solution. In the second viewpoint, one can say likewise that the underlying model is the nonlinear model $\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n)$, while the minimization of $\mathcal{J}_{\mathcal{X}}$ operates as a regularization by emphasizing that the nonlinear model should not be very "distinct" from the linear one.

5.3.2 Problem formulation

We propose to minimize simultaneously the objective functions $\mathcal{J}_{\mathcal{X}}(\boldsymbol{E}, \boldsymbol{A})$ and $\mathcal{J}_{\mathcal{H}}(\boldsymbol{E}, \boldsymbol{A})$, namely in both input and feature spaces as shown in Figure 5.2. Such problem is in a sense an ill-defined one. Indeed, it is not possible in general to find a solution that is optimal for both objective functions. As opposed to single-objective optimization problems where the main focus would be on the decision solution space, namely the space of all entries $(\boldsymbol{E}, \boldsymbol{A})$ (of dimension $LN + NT$), the bi-objective optimization problem brings the focus on the *objective space*, namely the space of the *objective vectors* $[\mathcal{J}_{\mathcal{X}}(\boldsymbol{E}, \boldsymbol{A}) \quad \mathcal{J}_{\mathcal{H}}(\boldsymbol{E}, \boldsymbol{A})]$. To study and solve this optimization problem, we revisit in our context the following definitions from the literature of multi-objective optimization:

- **Pareto dominance**: The solution $(\boldsymbol{E}_1, \boldsymbol{A}_1)$ is said to dominate $(\boldsymbol{E}_2, \boldsymbol{A}_2)$ if and only if $\mathcal{J}_{\mathcal{X}}(\boldsymbol{E}_1, \boldsymbol{A}_1) \leq \mathcal{J}_{\mathcal{X}}(\boldsymbol{E}_2, \boldsymbol{A}_2)$ and $\mathcal{J}_{\mathcal{H}}(\boldsymbol{E}_1, \boldsymbol{A}_1) \leq \mathcal{J}_{\mathcal{H}}(\boldsymbol{E}_2, \boldsymbol{A}_2)$, where at least one inequality is strict.

- **Pareto optimal**: A solution is a global (respectively local) Pareto optimal if and only if it is not dominated by any other solution (respectively in its neighborhood). That is, the objective vector $[\mathcal{J}_{\mathcal{X}}(\boldsymbol{E}^*, \boldsymbol{A}^*) \quad \mathcal{J}_{\mathcal{H}}(\boldsymbol{E}^*, \boldsymbol{A}^*)]$ corresponding to a Pareto optimal $(\boldsymbol{E}^*, \boldsymbol{A}^*)$ cannot be improved in any space (input or feature space) without any degradation in the other space.

- **Pareto front**: The set of the objective vectors corresponding to the Pareto optimal solutions forms the Pareto front in the objective space.

Various multi-objective optimization techniques have been successfully proposed *e.g.,* evolutionary algorithms, sum-weighted method, ε-constraint method, normal boundary intersection method, to name a few. See [Lampinen, 2000; Miettinen, 2008] for a survey. Among the existing methods, the sum-weighted or scalarization method has been always the most popular one, since it is straightforward and easy to implement [Das and Dennis, 1997; Ryu et al., 2010]. It converts a multi-objective problem into a single-objective problem by combining the multiple objectives. Under some conditions, the resulting objective vector belongs to the convex part of multi-objective problem's Pareto front. Thus, by changing appropriately the weights among the objectives, the Pareto front of the original problem is approximated. The main drawback of this method is that the nonconvex part of the Pareto front is often unattainable [Das and Dennis, 1997]. Nevertheless, it is the most practical one, in view of the complexity of the NMF problem, which is nonconvex, ill-posed and NP-hard [Vavasis, 2009].

5.4 Optimization with the Sum-weighted Method

Following the formulation introduced in the previous section, we study the minimization of the two objective functions $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$, under the nonnegativity constraints. The decision solution, of size $LN + NT$, corresponds to the entries in the unknown matrices $\boldsymbol{E}$ and $\boldsymbol{A}$. We transform this bi-objective optimization problem into an aggregated objective function (*i.e.*, sum-weighted objective function, also called scalarization value)

which is a convex combination of the two original objective functions, namely

$$\begin{aligned} \min_{\boldsymbol{E},\boldsymbol{A}} \quad & \alpha \mathcal{J}_{\mathcal{X}}(\boldsymbol{E},\boldsymbol{A}) + (1-\alpha)\mathcal{J}_{\mathcal{H}}(\boldsymbol{E},\boldsymbol{A}) \\ \text{subject to} \quad & \boldsymbol{E} \geq 0 \text{ and } \boldsymbol{A} \geq 0 \end{aligned} \tag{5.5}$$

where the weight $\alpha \in [0,1]$ controls the relative importance between objectives $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$. For a fixed value of α, this problem is called the sub-optimization problem. Its solution is a Pareto optimal for the original bi-objective problem, as proven in [Das and Dennis, 1997] for the general case. By solving the sub-optimization problem with a spread of values of α, we obtain an approximation of the Pareto front. It is obvious that the single-objective conventional NMF in (5.1) is given by $\alpha = 1$, while $\alpha = 0$ leads to the kernel-based formulation in (5.3).

Similar to the NMF, which is ill-posed, nonconvex and NP-hard [Vavasis, 2009], the optimization problem (5.5) is difficult to solve. It has no closed-form solution, a drawback inherited from most nonnegative constrained optimization problems. Moreover, the objective function is nonlinear, making the optimization problem more difficult. As in NMF algorithms, the global optimal solution cannot be guaranteed, thus the term Pareto optimal referred in the following is in the local sense.

Substituting the expressions given in (5.1) and (5.3) for $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$, the aggregated objective function becomes

$$\frac{\alpha}{2}\sum_{t=1}^{T}\Big\|\boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\,\boldsymbol{e}_n\Big\|^2 + \frac{1-\alpha}{2}\sum_{t=1}^{T}\Big\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\Big\|_{\mathcal{H}}^2.$$

After removing the constant terms that are independent of a_{nt} and $\boldsymbol{e}_n$, this objective function becomes

$$\begin{aligned} \mathcal{J}(\boldsymbol{E},\boldsymbol{A}) = \alpha \sum_{t=1}^{T}\Big(-\sum_{n=1}^{N} a_{nt}\boldsymbol{e}_n^\top \boldsymbol{x}_t + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_{nt}a_{mt}\boldsymbol{e}_n^\top \boldsymbol{e}_m\Big) \\ + (1-\alpha)\sum_{t=1}^{T}\Big(-\sum_{n=1}^{N} a_{nt}\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t) + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_{nt}a_{mt}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)\Big). \end{aligned} \tag{5.6}$$

In the following, we derive iterative techniques to minimize it with a two block-coordinate descent strategy, by alternating over the matrices $\boldsymbol{E}$ and $\boldsymbol{A}$, while keeping the other matrix fixed. The algorithm is outlined in Algorithm 1.

Algorithm 1 The proposed bi-objective KNMF, for a fixed $\alpha_m \in \{\alpha_1, \alpha_2, ..., \alpha_M\}$

Input: $k = 0$, warm start by $\boldsymbol{E}_m^0 = \boldsymbol{E}_{m-1}$ and $\boldsymbol{A}_m^0 = \boldsymbol{A}_{m-1}$
1: **repeat**
2: update $\boldsymbol{E}^{k+1}$ with (5.7), with stepsize obtained from Algorithm 2
3: update $\boldsymbol{A}^{k+1}$ with (5.12)
4: $k = k + 1$
5: **until** stopping criterion
Output: $\boldsymbol{E}_m$ and $\boldsymbol{A}_m$

5.4.1 Optimization over endmembers

Consider the minimization over $\boldsymbol{E}$ of the function $\mathcal{J}(\boldsymbol{E}, \boldsymbol{A})$, denoted $\mathcal{J}(\boldsymbol{E})$ in the following. The constrained optimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{E}} \quad & \mathcal{J}(\boldsymbol{E}) \\ \text{subject to} \quad & \boldsymbol{e}_n \geq 0 \text{ for } n = 1, \ldots, N, \end{aligned}$$

We apply the projected gradient descent strategy (PGD) presented in the previous chapters. At iteration k, the update rule takes the form

$$\boldsymbol{E}^{k+1} = \Big(\boldsymbol{E}^k - \eta_k \nabla_{\boldsymbol{E}} \mathcal{J}(\boldsymbol{E}^k)\Big)_+, \tag{5.7}$$

where η_k is the stepsize and $(\cdot)_+$ is the projection operator that maps its argument to the feasible nonnegative region.

In these expression, the gradient of (5.6) with respect to $\boldsymbol{e}_n$ is

$$\begin{aligned} \nabla_{\boldsymbol{e}_n} \mathcal{J}(\boldsymbol{E}) = \alpha \sum_{t=1}^{T} a_{nt} \Big(- \boldsymbol{x}_t + \sum_{m=1}^{N} a_{mt} \boldsymbol{e}_m \Big) \\ + (1-\alpha) \sum_{t=1}^{T} a_{nt} \Big(- \nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt} \nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) \Big), \end{aligned} \tag{5.8}$$

where $\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \cdot)$ denotes the gradient of the kernel with respect to its first argument $\boldsymbol{e}_n$, as given in Table 2.1. Without loss of generality, we restrict the presentation to the

Algorithm 2 The estimation of the optimal stepsize

Input: $0 < \rho < 1$

$\eta_k \leftarrow \eta_{k-1}$, $p = 1$
if η_k satisfies (5.9), **then**
  while η_k/ρ^p satisfies (5.9) **do**
    $\eta_k \leftarrow \eta_k/\rho^p$, $p \leftarrow p + 1$
  end while
else
  while η_k does not satisfy (5.9) **do**
    $\eta_k \leftarrow \eta_k\, \rho^p$, $p \leftarrow p + 1$
  end while
end if

Gaussian kernel for the objective function $\mathcal{J}_{\mathcal{H}}$. In this case, expression (5.8) becomes

$$\nabla_{\boldsymbol{e}_n}\mathcal{J}(\boldsymbol{E}) = \alpha \sum_{t=1}^{T} a_{nt}\Big(-\boldsymbol{x}_t + \sum_{m=1}^{N} a_{mt}\boldsymbol{e}_m\Big) + \frac{1-\alpha}{\sigma^2}\sum_{t=1}^{T} a_{nt}\Big(\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)(\boldsymbol{e}_n-\boldsymbol{x}_t) - \sum_{m=1}^{N} a_{mt}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)(\boldsymbol{e}_n-\boldsymbol{e}_m)\Big).$$

To estimate the stepsize η_k at each iteration k, we investigate the backtracking-Armijo line search, since it has proved effective for NMF [Lin, 2007b; Huck et al., 2010]. To this end, the stepsize is refined, either by increasing or decreasing its value ρ-fold, depending if the condition

$$\mathcal{J}(\boldsymbol{E}^k) - \mathcal{J}(\boldsymbol{E}^{k+1}) \leq \gamma\, \eta_k\, \mathrm{vec}(\nabla_{\boldsymbol{E}}\mathcal{J})^\top \mathrm{vec}(\boldsymbol{E}^{\mathrm{k}} - \boldsymbol{E}^{\mathrm{k+1}}) \tag{5.9}$$

is satisfied, where $\nabla_{\boldsymbol{E}}\mathcal{J} = [\nabla_{\boldsymbol{e}_1}\mathcal{J} \quad \nabla_{\boldsymbol{e}_2}\mathcal{J} \quad \cdots \quad \nabla_{\boldsymbol{e}_N}\mathcal{J}]$, $\mathrm{vec}(\cdot)$ reshapes this matrix into a column vector, and γ characterizes the decrease level and is often set to 1%. The algorithm that accelerates the stepsize search is outline in Algorithm 2.

5.4.2 Optimization over abundances

In order to minimize the function $\mathcal{J}(\boldsymbol{E}, \boldsymbol{A})$ over $\boldsymbol{A}$, denoted $\mathcal{J}(\boldsymbol{A})$ in the following, the PGD update rule for $\boldsymbol{A}$ can be derived in the same way as for $\boldsymbol{E}$. However, the stepsize estimation in PGD rule is very time consuming. To alleviate this problem, we develop the multiplicative update (MU) for $\boldsymbol{A}$. Both linear and nonlinear models are linear-in-the-parameters in terms of a_{nt}, which is not the case when dealing with the endmember matrix $\boldsymbol{E}$. Therefore, owing to the convexity of the subproblem $\mathcal{J}(\boldsymbol{A})$, the MU for

$\boldsymbol{A}$ yields a monotone decrease in the objective function. Denote by Λ_k the matrix of stepsize values at iteration k, where $(\Lambda_k)_{nt} = \lambda_{k,nt}$. The PGD update rule in terms of $\boldsymbol{A}$ is

$$\boldsymbol{A}^{k+1} = \Big(\boldsymbol{A}^k - \Lambda_k \nabla_{\boldsymbol{A}} \mathcal{J}(\boldsymbol{A}^k)\Big)_+. \tag{5.10}$$

Here, the stepsize balances the rate of convergence with the accuracy of optimization, and can be set differently depending on n and t.

In these expressions, the derivative of (5.6) with respect to a_{nt} is

$$\nabla_{a_{nt}} \mathcal{J}(\boldsymbol{A}) = \alpha\Big(- \boldsymbol{e}_n^\top \boldsymbol{x}_t + \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_n^\top \boldsymbol{e}_m\Big) + (1-\alpha)\Big(- \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)\Big). \tag{5.11}$$

In order to get the multiplicative update rule, we choose the stepsize parameter in (5.10) as

$$\lambda_{k,nt} = \frac{a_{nt}^k}{\alpha \sum_{m=1}^{N} a_{mt}^k \boldsymbol{e}_n^\top \boldsymbol{e}_m + (1-\alpha) \sum_{m=1}^{N} a_{mt}^k\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)},$$

which yields

$$a_{nt}^{k+1} = a_{nt}^k \, \frac{\alpha\, \boldsymbol{e}_n^\top \boldsymbol{x}_t + (1-\alpha)\, \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)}{\alpha \sum_{m=1}^{N} a_{mt}^k \boldsymbol{e}_n^\top \boldsymbol{e}_m + (1-\alpha) \sum_{m=1}^{N} a_{mt}^k\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)}. \tag{5.12}$$

It is noteworthy that the multiplicative update rule for $\boldsymbol{e}_n$ can be elaborated in the same way, by using the split gradient method. However, since the sub-optimization on $\boldsymbol{e}_n$ is possibly nonconvex[1], the monotone property is not guaranteed with an arbitrary kernel. That is, for a given weight α, although the aggregated objective function $\mathcal{J}$ globally decreases, the overshoot of stepsize in updating $\boldsymbol{E}$ may occur during iterations. This discussion is outlined in Table 5.1.

[1] In conventional NMF, the subproblem of estimating each matrix separately is convex. Thanks to this property, the monotone decreasing property of the multiplicative-style update rules was proved by constructing an auxiliary function as an upper bound [Lee and Seung, 2001; Lin, 2007a]. In our work, the proposed framework involves a nonconvex optimization problem on $\boldsymbol{e}_n$, since the Hessian matrix is no longer guaranteed to be positive semidefinite.

Table 5.1: The convexity and the optimization methods for each subproblem

	Convexity	PGD	MU
$\min_{\boldsymbol{E}} \mathcal{J}(\boldsymbol{E})$		✓	
$\min_{\boldsymbol{A}} \mathcal{J}(\boldsymbol{A})$	✓	✓	✓

5.4.3 Complexity, convergence and stopping criteria

The complexity of the PGD method for $\boldsymbol{E}$ in Algorithm 2 is $\mathcal{O}(pTLN^2)$, where p is the average number of checking condition (5.9). The complexity of the MU for $\boldsymbol{A}$ is $\mathcal{O}(TLN^2)$. Thus, the total complexity of Algorithm 1 is $\mathcal{O}(k(p+1)TLN^2)$ after k iterations. This complexity holds using any commonly-used kernel listed in Table 2.1, with roughly the same complexity $\mathcal{O}(L)$ for each kernel.

Similar to the PGD and MU rules initially presented for the linear NMF, the proposed algorithm is a stationary point method. See also the discussions on the convergence of the conventional NMF in [Gonzales and Zhang, 2005; Lin, 2007a]. We use a twofold stopping criterion, *i.e.*, either a stationary point is attained, or the preset maximum number of iterations is reached. To be more specific, the algorithm stops when either the condition $\|\mathcal{J}(\boldsymbol{E}^{k+1}, \boldsymbol{A}^{k+1}) - \mathcal{J}(\boldsymbol{E}^{k}, \boldsymbol{A}^{k}))\| < \varepsilon$ is satisfied, or $k = k_{\max}$. In the experiments, the threshold of the error difference between successive iteration is set to $\varepsilon = 10^{-4}$.

5.4.4 A posteriori analysis of the approximated Pareto front

It is worth noting that we apply the sum-weighted method as *a posteriori* method, where different Pareto optimal solutions are generated. The Decision Maker makes the final compromise among optimal solutions, from the conventional linear NMF to the Gaussian KNMF. Alternatively, in *a priori* methods, the Decision Maker specifies the weight α in advance to generate a solution. See [Miettinen, 2008] for more details.

All the points on the approximated Pareto front are optimal in some sense. To choose the α suitable to the data under scrutiny, we employ level diagrams approach studied in [Blasco et al., 2008]. This *a posteriori* method classifies the points on the Pareto front

according to their proximities to the ideal point, defined in our case by

$$\mathcal{J}^{**} = [\min \mathcal{J}_{\mathcal{X}} \quad \min \mathcal{J}_{\mathcal{H}}],$$

where $\min \mathcal{J}_{\mathcal{X}}$ and $\min \mathcal{J}_{\mathcal{H}}$ denote respectively the minimum values of the two objective functions obtained on the Pareto front. For this purpose, each point $\mathcal{J} = [\mathcal{J}_{\mathcal{X}} \quad \mathcal{J}_{\mathcal{H}}]$ is first normalized to $\overline{\mathcal{J}} = [\overline{\mathcal{J}_{\mathcal{X}}} \quad \overline{\mathcal{J}_{\mathcal{H}}}]$ using the maximum and minimum achieved values, that is

$$\overline{\mathcal{J}_{\mathcal{X}}} = \frac{\mathcal{J}_{\mathcal{X}} - \min \mathcal{J}_{\mathcal{X}}}{\max \mathcal{J}_{\mathcal{X}} - \min \mathcal{J}_{\mathcal{X}}},$$

and

$$\overline{\mathcal{J}_{\mathcal{H}}} = \frac{\mathcal{J}_{\mathcal{H}} - \min \mathcal{J}_{\mathcal{H}}}{\max \mathcal{J}_{\mathcal{H}} - \min \mathcal{J}_{\mathcal{H}}}.$$

The distance to the ideal point is then evaluated with an ℓ_p-norm. Of particular interest are the following norms:

- ℓ_1-norm: $\|\overline{\mathcal{J}}\|_1 = \overline{\mathcal{J}_{\mathcal{X}}} + \overline{\mathcal{J}_{\mathcal{H}}}$;
- ℓ_2-norm: $\|\overline{\mathcal{J}}\|_2 = \sqrt{\overline{\mathcal{J}_{\mathcal{X}}}^2 + \overline{\mathcal{J}_{\mathcal{H}}}^2}$;
- ℓ_∞-norm: $\|\overline{\mathcal{J}}\|_\infty = \max(\overline{\mathcal{J}_{\mathcal{X}}}, \overline{\mathcal{J}_{\mathcal{H}}})$;
- $\ell_{-\infty}$-norm: $\|\overline{\mathcal{J}}\|_{-\infty} = \min(\overline{\mathcal{J}_{\mathcal{X}}}, \overline{\mathcal{J}_{\mathcal{H}}})$.

It is clear that the points with small norms locate nearly to the ideal point; therefore the Decision Maker can choose a solution among them.

5.5 Experiments

In this section, the performance of the proposed bi-objective KNMF is demonstrated on the unmixing of synthetic and real hyperspectral images, by comparing with state-of-the-art unmixing techniques. We consider three supervised methods, namely FCLS [Heinz and Chang, 2001], K-Hype [Chen et al., 2013b] and GBM-sNMF [Yokoya et al., 2014], where NMF is applied in prior for endmember extraction. We further consider several unsupervised methods, including four NMF-based ones: MiniDisCo [Huck et al., 2010], ConvexNMF [Ding et al., 2010], KconvexNMF [Li and Ngom, 2012] and MercerNMF [Pan et al., 2011], as well as a nonlinear unmixing technique Mac-Mic [Close et al.,

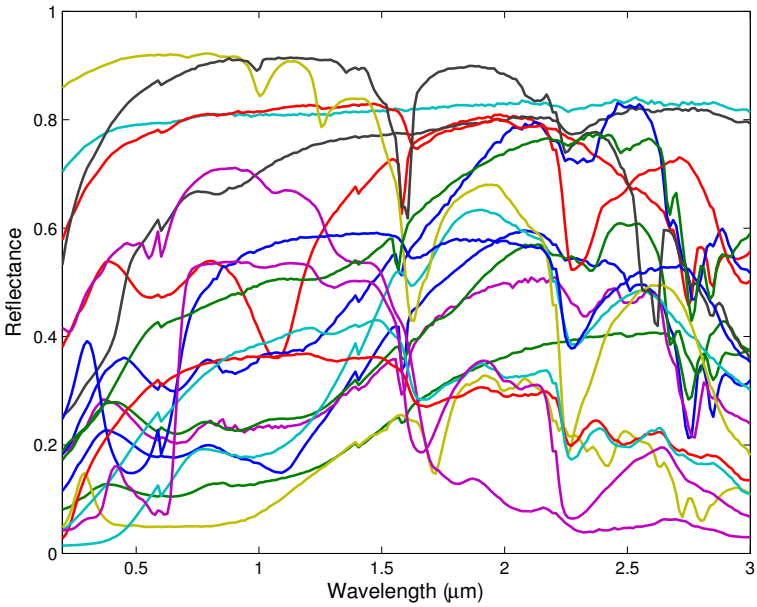


Figure 5.3: The USGS spectra used for synthetic data generation.

2014]. The unmixing performance is evaluated by two criteria, the averaged spectral angle distance between endmembers (SAD) and the root mean square error on the abundances (RMSE), as defined with (4.15) and (4.16) in Section 4.4.

5.5.1 Simulation with synthetic data

The performance of the proposed method is firstly studied on a series of synthetic images, each of size 20×20 pixels. The generalized bilinear model (GBM) is considered as defined in (1.6). The data are generated as given in [Bioucas-Dias and Nascimento, 2008] as follows. First, $N = 3$ or $N = 6$ endmembers are randomly selected from the candidate spectra set. This set is composed of 19 spectra drawn from the United States Geological Survey (USGS) digital spectral library, as given in Figure 5.3. Second, the abundance vectors are uniformly generated using a Dirichlet distribution on the simplex defined by the nonnegativity and the sum-to-one constraints. Last, the data are corrupted with a Gaussian noise at two different levels, with the signal-to-noise ratio of 30 dB and 15 dB.

Experiments are conducted employing the weight set $\alpha \in \{0, 0.1, \ldots, 0.9, 1\}$, which implies the model varying gradually from the nonlinear Gaussian NMF ($\alpha = 0$) to the conventional linear NMF ($\alpha = 1$). For each value of α from the weight set, Algorithm 1 is applied. The maximum iteration number is set as $k_{\max} = 2000$ in all the comparing

Table 5.2: Unmixing performance on synthetic data ($\times 10^{-2}$)

		$N = 3$				$N = 6$			
		SNR = 30dB		SNR = 15dB		SNR = 30dB		SNR = 15dB	
		SAD	RMSE	SAD	RMSE	SAD	RMSE	SAD	RMSE
	FCLS	-	32.48	-	31.99	-	30.01	-	28.17
	GBM-sNMF	-	28.91	-	27.48	-	27.79	-	26.49
	K-Hype	-	8.40	-	10.63	-	12.31	-	11.11
	MiniDisCo	8.20	10.49	11.60	12.24	14.53	② **7.66**	17.93	② **7.99**
	ConvexNMF	14.19	21.43	13.91	21.96	19.06	12.15	20.00	12.86
	KconvexNMF	-	14.40	-	16.36	-	12.45	-	12.40
	MercerNMF	-	16.02	-	15.94	-	① **7.60**	-	① **7.54**
	Mac-Mic	9.93	12.72	13.34	12.34	14.48	13.01	19.05	9.04
Bi-objective KNMF [this work]	$\alpha = 1$	8.29	25.26	11.14	24.18	15.88	37.01	24.08	36.44
Pareto Optimal	$\alpha = 0.9$	① **4.80**	① **4.67**	① **6.22**	③ 6.83	12.58	③ 8.83	21.97	③ 8.44
	$\alpha = 0.8$	② **5.34**	② **4.86**	② **6.40**	① **6.37**	② **11.78**	8.93	18.83	8.85
	$\alpha = 0.7$	③ 6.19	③ 6.13	③ 6.95	② **6.76**	① **11.77**	8.85	17.36	9.10
	$\alpha = 0.6$	7.10	7.81	7.49	7.62	③ 11.95	9.28	16.56	9.14
	$\alpha = 0.5$	7.85	9.06	7.95	8.40	12.27	9.93	16.11	9.80
	$\alpha = 0.4$	8.48	9.80	8.45	8.90	12.70	10.80	③ 15.46	10.45
	$\alpha = 0.3$	9.16	10.59	8.90	9.36	13.10	11.72	② **15.19**	10.89
	$\alpha = 0.2$	9.92	11.74	9.51	9.82	13.67	12.27	① **15.16**	11.14
	$\alpha = 0.1$	10.95	13.00	10.34	10.60	14.42	12.93	15.57	11.08
	$\alpha = 0$	12.32	17.55	12.54	15.54	15.22	13.83	16.42	12.32

methods. The bandwidth parameter in the Gaussian kernel is roughly set to $\sigma = 3.0$ for all the experiments. By performing ten Monte-Carlo simulations, the average values in terms of SAD and RMSE are compared with the state-of-the-art unmixing methods, as given in Table 5.2.

We observe the following. For all the considered numbers of endmembers and noise levels, the proposed bi-objective KNMF with the Pareto optimal outperforms not only state-of-the-art methods but also the linear ($\alpha = 1$) and Gaussian ($\alpha = 0$) NMF in terms of endmember estimation. Given a relatively small number of endmembers with $N = 3$, the proposed method also yields the smallest root mean square error on the abundances regardless of the noise level. For $N = 6$, it provides comparable results to MercerNMF and MiniDisCo, being slightly worse in terms of RMSE and slightly better in terms of SAD.

5.5.2 Experiments with the Urban image

As depicted in Figure 5.4, the real hyperspectral image studied is from the Urban image, acquired by the HYDICE sensor. A description of the Urban image is available in

Figure 5.4: The scene from the Urban image

Section 4.4.3. In this chapter, we take the top left part with 150×150 pixels from the original 307×307 pixels' image. According to the ground truth provided in [Jia and Qian, 2007; Fong and Hu, 2011], the studied area is mainly composed of four endmembers shown in Figure 5.5: asphalt, grass, tree and roof. In experiments, the weight set is chosen as $\alpha \in \{0, 0.04, \ldots, 0.96, 1\}$, and the maximum iteration number is set to $k_{\max} = 300$. Starting from $\alpha_1 = 0$, the matrix $\boldsymbol{E}_1$ is initialized by conducting NMF on 1000 randomly chosen samples, while the elements in $\boldsymbol{A}_1$ are generated using a $[0, 1]$ uniform distribution. The bandwidth in the Gaussian kernel is selected as $\sigma = 4.2$, after a preliminary analysis using the single-objective Gaussian NMF with the candidate set $\{0.2, 0.3, \ldots, 9.9, 10, 15, 20, \ldots, 50\}$.

The unmixing performance is shown in Table 5.3, with several ℓ_p-norms as described in Section 5.4.4. Methods that do not extract endmembers are not included, such as FCLS, sNMF, K-Hype, MercerNMF and KconvexNMF, since they poorly perform as shown in Table 5.2. Compared with the state-of-the-art methods, three endmembers out of four, *i.e.*, Asphalt, Tree and Roof, are better estimated by Pareto optima. The estimated abundance maps corresponding to the four endmembers are shown in Figure 5.8.

We compare in Table 5.4 the computational time of the proposed method with the aforementioned unmixing algorithms that jointly estimate the endmembers and abundances. Nonlinear methods, and in particular kernel-based ones, are time-consuming in general. Regarding the proposed bi-objective KNMF, its computational complexity is lower than the one of MercerNMF, for a fixed value of α. When considering a spread of values of α, the sub-optimization problems can be addressed in parallel.

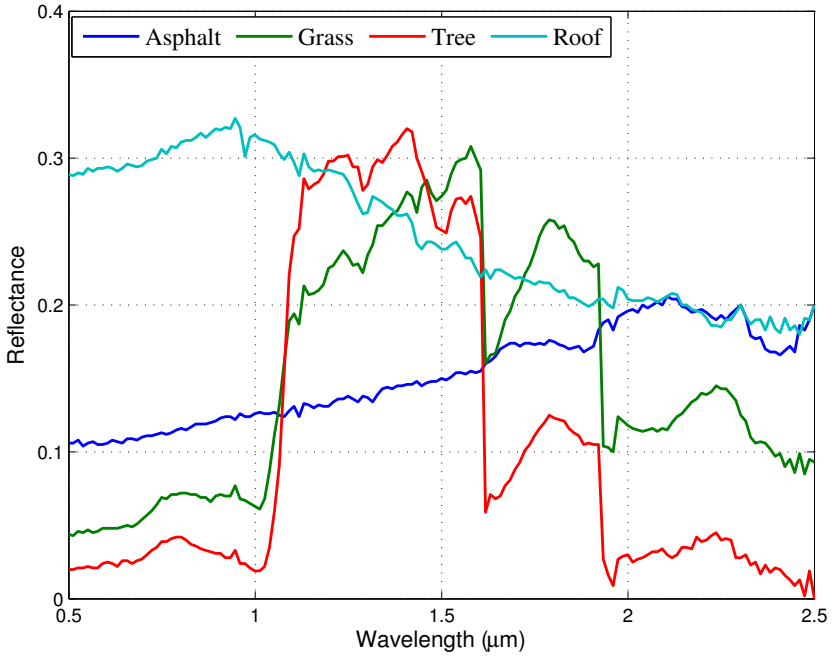


Figure 5.5: The four ground truth endmembers in the Urban image.

Table 5.3: Performance on the Urban image ($\times 10^{-2}$)

		Spectral Angle Distance				
		SAD	Asphalt	Grass	Tree	Roof
	MiniDisCo	③ 30.23	25.91	① **25.62**	13.86	55.51
	ConvexNMF	34.83	48.15	47.87	14.29	35.01
	Mac-Mic	33.53	③ 10.78	② **43.65**	53.00	26.68
this thesis	$\alpha = 1$ ($\ell_{-\infty}$-norm)	40.84	87.29	60.03	① **7.92**	③ 8.14
Pareto opt.	$\alpha = 0.48$ (ℓ_2-norm)	31.28	66.74	46.11	② **8.30**	① **3.95**
Pareto opt.	$\alpha = 0.40$ (ℓ_∞-norm)	30.45	64.18	③ 44.95	③ 8.37	② **4.30**
Pareto opt.	$\alpha = 0.04$ ($\ell_{-\infty}$-norm)	② **29.79**	① **8.34**	70.54	9.77	30.46
	$\alpha = 0$	① **28.55**	② **8.93**	62.31	10.10	32.84

Table 5.4: Estimated computational time (in seconds)

	MiniDisCo	220
	ConvexNMF	996
nonlinear	KconvexNMF	2622
nonlinear	MercerNMF	20332
nonlinear	Mac-Mic	4244
nonlinear	Bi-Objective NMF, average per α	5420

5.5.3 Approximating the Pareto front

Inherited from nonlinear multi-objective optimization problems, the determination of the whole Pareto front is intractable and the target becomes to approximate the Pareto front by a set of discrete points, as stated in [Lampinen, 2000]. To this end, we operate as follows: For each value of α, we obtain a solution (endmember and abundance matrices) from the proposed algorithm; by evaluating the objective functions $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$ at this solution, we get a single point in the objective space, as shown in Figure 5.6. Figure 5.7 shows the evolution of these objectives functions and the aggregated objective function $\mathcal{J}$, evaluated at the solution obtained for each α.

We observe the following:

1. Regarding the sum-weighted approach, the minimizer of the sub-optimization problem is proven to be a Pareto optimal for the original multi-objective problem, *i.e.*, the corresponding objective vector belongs to the Pareto front in the objective space [Das and Dennis, 1997]. For the Urban image, we obtain 25 (out of 26) nondominated solutions. The solution for $\alpha = 0$ is dominated by the solutions on the approximated Pareto front, with respect to both objectives. Such phenomenon is not surprising. Indeed, *there exist multiple Pareto optimal solutions in a problem only if the objectives are conflicting to each other*, as demonstrated in [Deb and Kalyanmoy, 2001][2]. As shown in Figure 5.6 and Figure 5.7, the obtained solutions are Pareto optimal within the objectives-conflicting interval $\alpha \in [0.04, 1]$.

2. A uniform distribution of the values of α from $[0, 1]$ does not lead to a uniform spread of the solutions on the approximated Pareto front. Moreover, the nonconvex part of the Pareto front cannot be attained using any weight. These are two major drawbacks of the sum-weighted method, as stated in [Das and Dennis, 1997] and illustrated in Figure 5.6.

Nevertheless, the obtained approximation of Pareto front is of high value. On one hand, it provides a set of nondominated solutions for the Decision Maker. On the other hand, an insight of the tradeoff between objectives $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$ reveals the underlying linearity/nonlinearity of the data under study.

[2]For example, the Pareto optimal solutions for the well-known Schaffer's function, defined by $\mathcal{J}(x) = [x^2 \quad (x-2)^2]$, are found only within the interval $[0, 2]$, where a tradeoff between the two objectives exists. See [Zitzler and Thiele, 1999].

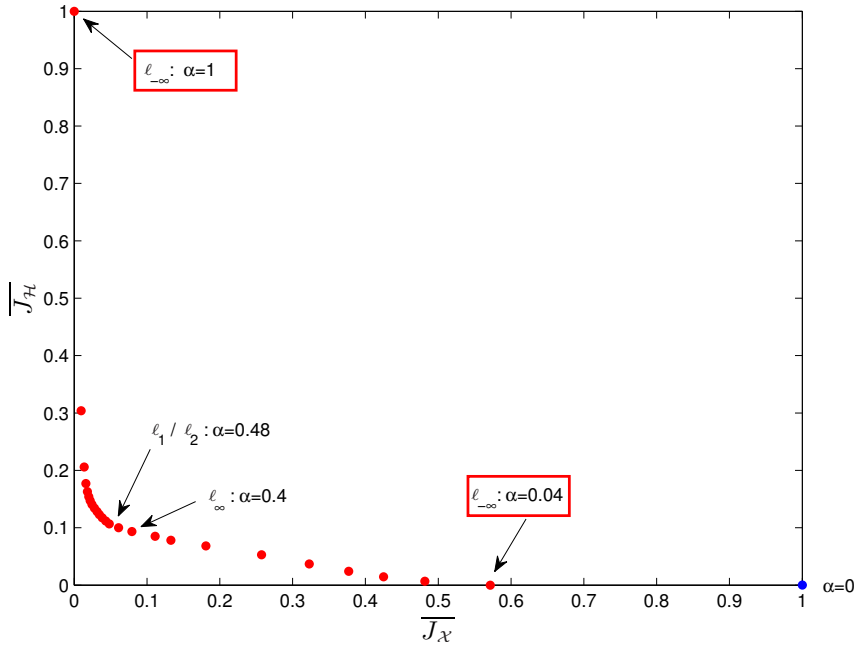


Figure 5.6: Illustration of the approximated Pareto front in the objective space for the Urban image. The (normalized) objective vectors of the 25 nondominated solutions, marked in red, approximate a part of the Pareto front; the single dominated solution is marked in blue.

5.6 Conclusion

This chapter presented a bi-objective nonnegative matrix factorization by exploiting the kernel machines, where the decomposition was performed simultaneously in the input and the feature spaces. The performance of the method was demonstrated for unmixing synthetic and real hyperspectral images. The approximation of the Pareto front was analyzed.

Future works include a more efficient way to determine the good value of the weight parameter α. In addition, we will incorporate physical-based unmixing models, namely the bilinear ones and the macroscopic-microscopic models, by defining appropriately the kernel in the proposed framework. Considering simultaneously several kernels, and consequently several feature spaces, is also under investigation.

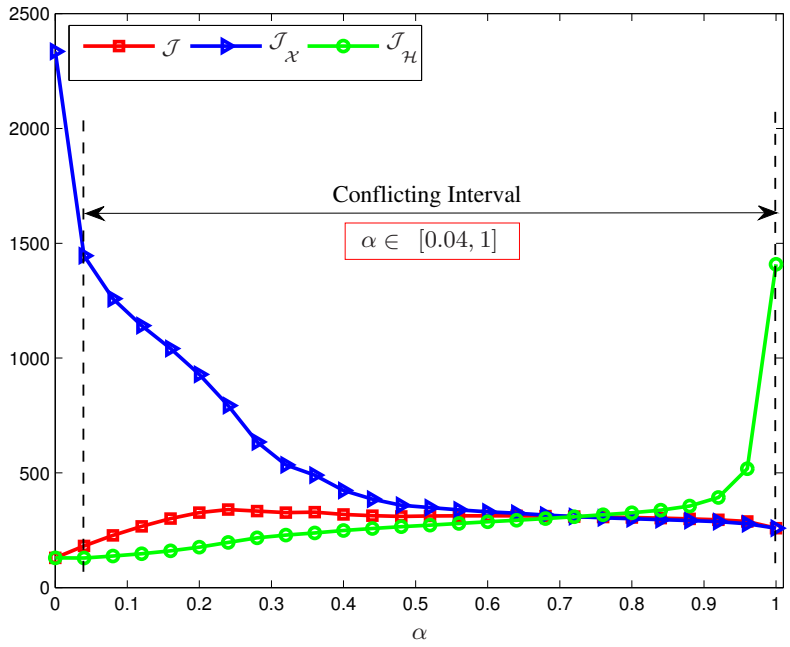


Figure 5.7: Visualization of the tradeoff between the objectives $\mathcal{J}_{\mathcal{X}}$ and $\mathcal{J}_{\mathcal{H}}$, and the change of the aggregated objective function $\mathcal{J}$, along with the increment of α for the Urban image.

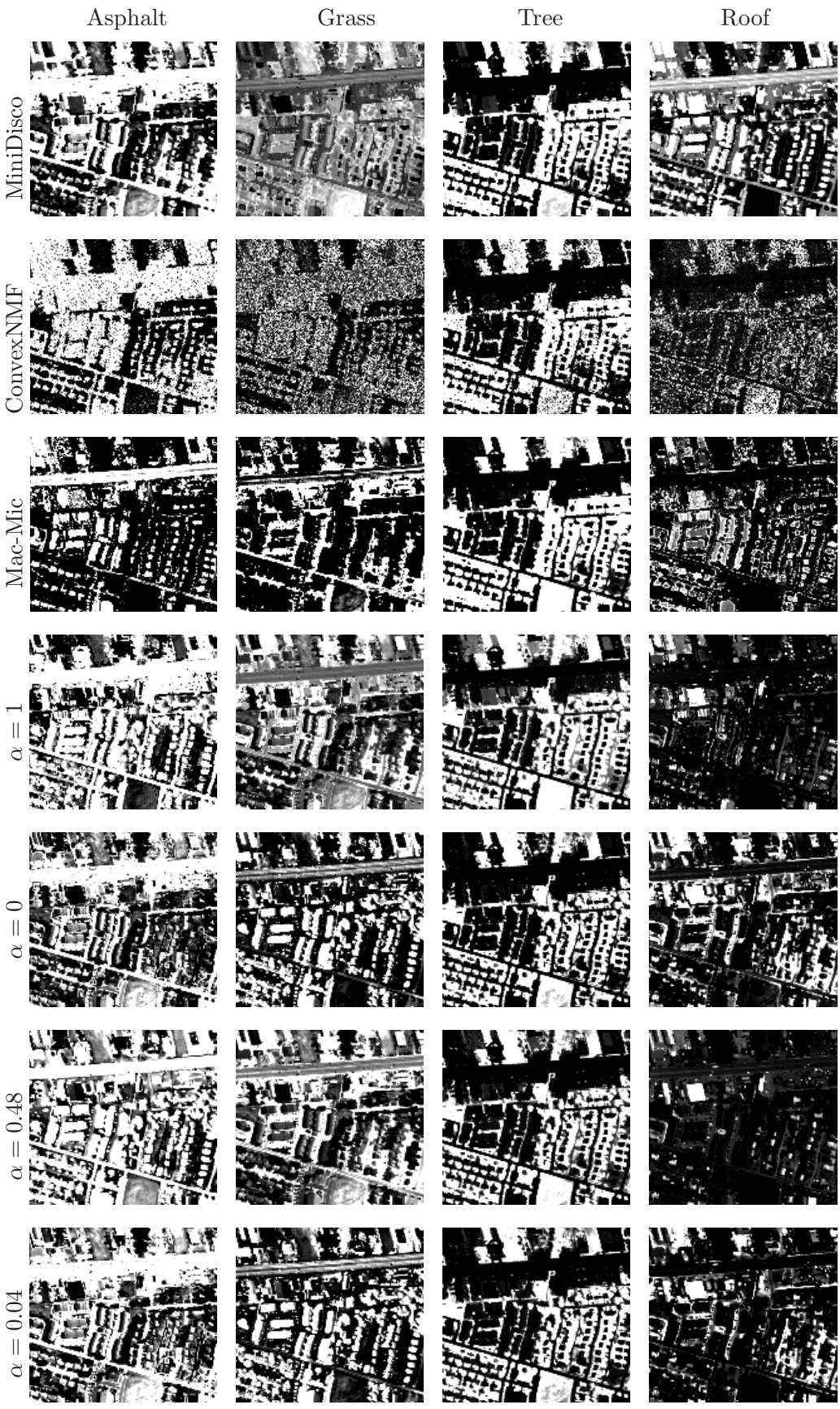


Figure 5.8: Estimated abundance maps for the Urban image. Left to right: Abundance maps for Asphalt, Grass, Tree and Roof. Top to bottom: MiniDisco, ConvexNMF, Mac-Mic, and the proposed bi-objective KNMF with $\alpha = 1$ (conventional linear NMF), $\alpha = 0$ (nonlinear Gaussian NMF), and Pareto optimal solutions $\alpha = 0.48$ (ℓ_1/ℓ_2-norm) and $\alpha = 0.04$ (ℓ_∞-norm).

Chapter 6

Correntropy Maximization via ADMM for Robust Unmixing

Contents

In hyperspectral images, some spectral bands suffer from low signal-to-noise ratio due to noisy acquisition and atmospheric effects, thus requiring robust techniques for the unmixing problem. This chapter presents a robust supervised spectral unmixing approach for hyperspectral images. The robustness is achieved by writing the unmixing problem as the maximization of the correntropy criterion subject to the most commonly used constraints. Two unmixing problems are derived: the first problem considers the fully-constrained unmixing, with both the nonnegativity and sum-to-one constraints, while the second one deals with the nonnegativity and the sparsity-promoting of the abundances. The corresponding optimization problems are solved efficiently using an alternating direction method of multipliers (ADMM) approach. Experiments on synthetic and real hyperspectral images validate the performance of the proposed algorithms for different scenarios, demonstrating that the correntropy-based unmixing is robust to outlier bands.

6.1 Introduction

By far, almost all the unmixing algorithms hugely suffer from noisy data and outliers within bands. Indeed, in real hyperspectral images for remote sensing, a considerable proportion (about 20%) of the spectral bands are noisy with low signal-to-noise ratio (SNR), due to the atmospheric effect such as water absorption [Zelinski and Goyal, 2006]. These bands need to be removed prior to applying any existing unmixing method; otherwise, the unmixing quality drastically decreases. Such sensitivity to outliers is due to the investigated ℓ_2-norm as a cost function in the fully-constrained least-squares method (FCLS) [Bioucas-Dias et al., 2012] and sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) [Bioucas-Dias and Figueiredo, 2010] algorithms, as well as all unmixing algorithms that explore least-squares solutions. It is worth noting that nonlinear unmixing algorithms also suffer from this drawback, including the kernel-based fully-constrained least-squares (KFCLS) [Broadwater et al., 2007], the nonlinear fluctuation methods [Chen et al., 2013b] and the post-nonlinear methods [Chen et al., 2013c]. See Chapter 1 for a review.

Information theoretic learning provides an elegant alternative to the conventional minimization of the ℓ_2-norm in least-squares problems, by considering the maximization of the so-called correntropy [Liu et al., 2007; Principe, 2010]. Due to its stability and robustness to noise and outliers, the correntropy maximization is based on theoretical

foundations and has been successfully applied to a wide class of applications, including cancer clustering [Wang et al., 2013a], face recognition [He et al., 2011], and recently hyperspectral unmixing [Wang et al., 2015], to name a few. In these works, the resulting problem is optimized by the half-quadratic technique [Nikolova and Ng, 2005], either in a supervised manner [He et al., 2011] or as an unsupervised nonnegative matrix factorization [Wang et al., 2013a, 2015].

In this chapter, we consider the hyperspectral unmixing problem by defining an appropriate correntropy-based criterion, thus taking advantage of its robustness to large outliers, as opposed to the conventional ℓ_2-norm criteria. By including constraints commonly used for physical interpretation, two unmixing problems are derived in detail: the first problem considers the fully-constrained unmixing, with both the nonnegativity (ANC) and sum-to-one (ASC) constraints, while the second one deals with the nonnegativity and the sparsity-promoting of the abundances. We propose to solve these constrained optimization problems with alternating direction method of multipliers (ADMM) algorithms. Indeed, the ADMM approach splits a hard problem into a sequence of small and handful ones [Boyd et al., 2011]. Its relevance to solve nonconvex problems was studied in [Boyd et al., 2011, Section 9]. We show that ADMM provides a relevant framework for incorporating different constraints arising in the unmixing problem. We present the so-called CUSAL (for *Correntropy-based Unmixing by variable Splitting and Augmented Lagrangian*), and study in particular two algorithms: CUSAL-FC to solve the fully-constrained (ANC and ASC) correntropy-based unmixing problem, and the CUSAL-SP to solve the sparsity-promoting correntropy-based unmixing problem.

6.2 Classical Unmixing Problems

Before proceeding, we outline the hyperspectral unmixing problems detailed in Chapter 1, by emphasizing on the models and problems that shall be revisited in this chapter.

The linear mixture model (LMM) assumes that each spectrum can be expressed as a linear combination of a set of pure material spectra, namely

$$\begin{aligned}\boldsymbol{x}_t &= \sum_{n=1}^{N} a_{nt}\,\boldsymbol{e}_n + \boldsymbol{n}_t \\ &= \boldsymbol{E}\boldsymbol{a}_t + \boldsymbol{n}_t,\end{aligned}$$

where $\boldsymbol{E} = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_N] \in \mathbb{R}^{L\times N}$ is the matrix of the N endmembers with $\boldsymbol{e}_n = [e_{1n} \ \cdots \ e_{Lr}]^\top$, $\boldsymbol{a}_t = [a_{1t} \ \cdots \ a_{Nt}]^\top$ is the abundance vector associated with the t-th pixel, and $\boldsymbol{n}_t \in \mathbb{R}^L$ is the additive noise. In matrix form for all pixels, we have

$$\boldsymbol{X} = \boldsymbol{E}\boldsymbol{A} + \boldsymbol{N},$$

where $\boldsymbol{A} = [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_T] \in \mathbb{R}^{N\times T}$ and $\boldsymbol{N}$ is the noise matrix.

In the following, the endmembers are assumed known, either from ground-truth information or by using any endmember extraction technique. The unmixing problem consists in estimating the abundances for each pixel. The easiest way to solve this problem is to consider the unconstrained least-squares optimization problem

$$\min_{\boldsymbol{a}_t} \|\boldsymbol{x}_t - \boldsymbol{E}\boldsymbol{a}_t\|_2^2, \tag{6.1}$$

for each $t = 1,\ldots,T$, where $\|\cdot\|_2$ denotes the conventional ℓ_2-norm. The solution to this conventional least-squares problem is given by the pseudo-inverse of the (tall) endmember matrix, with $\boldsymbol{a}_t = (\boldsymbol{E}^\top\boldsymbol{E})^{-1}\boldsymbol{E}^\top\boldsymbol{x}_t$. The least-squares optimization problems (6.1), for all $t = 1,\ldots,T$, are often written in a single optimization problem using the following matrix formulation $\min_{\boldsymbol{A}} \|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm. Its solution is

$$\boldsymbol{A}_{\text{LS}} = (\boldsymbol{E}^\top\boldsymbol{E})^{-1}\boldsymbol{E}^\top\boldsymbol{X}. \tag{6.2}$$

Finally, this optimization problem can be also tackled by considering all the image pixels at each spectral band, which yields the following least-squares optimization problem

$$\min_{\boldsymbol{A}} \sum_{l=1}^{L} \|\boldsymbol{x}_{l*} - \boldsymbol{e}_{l*}\boldsymbol{A}\|_2^2,$$

where $\boldsymbol{x}_{l*}$ its l-th row of $\boldsymbol{X}$ representing the l-th band over all pixels, and $\boldsymbol{e}_{l*}$ is the

row vector of the l-th spectral band over all the endmembers. While all these problem formulations have a closed-form solution, they suffer from two major drawbacks. The first one is that several constraints need to be imposed in order to have a physical meaning of the results. The second drawback is its sensitivity to noise and outliers, due to the use of the ℓ_2-norm as the fitness measure. These two drawbacks are detailed in the following.

To be physically interpretable, the abundances should be nonnegative (ANC) and satisfy the sum-to-one constraint (ASC). Considering these constraints, the fully-constrained least-squares problem is formulated as, for each $t = 1, \ldots, T$,

$$\min_{\boldsymbol{a}_t} \|\boldsymbol{x}_t - \boldsymbol{E}\boldsymbol{a}_t\|_2^2, \text{ subject to } \boldsymbol{a}_t \geq 0 \text{ and } \mathbf{1}^\top \boldsymbol{a}_t = 1,$$

where $\mathbf{1} \in \mathbb{R}^{N \times 1}$ denotes the column vector of ones and ≥ 0 is the nonnegativity applied element-wise, or in matrix form:

$$\min_{\boldsymbol{A}} \|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}\|_F^2, \text{ subject to } \boldsymbol{A} \geq 0 \text{ and } \mathbf{1}^\top \boldsymbol{a}_t = 1, \text{ for } t = 1, \ldots, T.$$

Since there is no closed-form solution when dealing with the nonnegativity constraint, several iterative techniques have been proposed, such as the *active set* scheme with the Lawson and Hanson's algorithm [Lawson and Hanson, 1987], the multiplicative iterative strategies [Lantéri et al., 2001], and the fully-constrained least-squares (FCLS) technique [Heinz and Chang, 2001]. More recently, the alternating direction method of multipliers (ADMM) was applied with success for hyperspectral unmixing problem, with the SUnSAL algorithm [Bioucas-Dias and Figueiredo, 2010].

Recent work in hyperspectral unmixing have advocated the sparsity in the abundance vectors [Bioucas-Dias and Figueiredo, 2010; Iordache et al., 2011, 2012]. In this case, each spectrum is fitted by a sparse linear mixture of endmembers, namely only the abundances with respect to a small number of endmembers are nonzero. To this end, the sparsity-promoting regularization with the ℓ_1-norm is included in the cost function, yielding the following constrained sparse regression problem [Bioucas-Dias and Figueiredo, 2010], for

each $t = 1, \ldots, T$,

$$\min_{\boldsymbol{a}_t} \|\boldsymbol{x}_t - \boldsymbol{E}\boldsymbol{a}_t\|_2^2 + \lambda\|\boldsymbol{a}_t\|_1, \text{ subject to } \boldsymbol{a}_t \geq 0,$$

where the parameter λ balances the fitness of the least-squares solution and the sparsity level. It is worth noting that the ASC is relaxed when the ℓ_1-norm is included. This problem is often considered by using the following matrix formulation:

$$\min_{\boldsymbol{A}} \|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}\|_F^2 + \lambda \sum_{t=1}^{T} \|\boldsymbol{a}_t\|_1, \text{ subject to } \boldsymbol{A} \geq 0.$$

6.2.1 Sensitivity to outliers

All the aforementioned algorithms rely on solving a (constrained) least-squares optimization problem, thus inheriting the drawbacks of using the ℓ_2-norm as the fitness measure. A major drawback is its sensitivity to outliers, where outliers are some spectral bands that largely deviate from the rest of the bands. Indeed, considering all the image pixels, the least-squares optimization problems take the form

$$\min_{\boldsymbol{A}} \sum_{l=1}^{L} \|\boldsymbol{x}_{l*} - \boldsymbol{e}_{l*}\boldsymbol{A}\|_2^2. \tag{6.3}$$

subject to any of the aforementioned constraints. From this formulation, it is easy to see how the ℓ_2-norm gives more weight to large residuals, namely to outliers in which the estimated values $\widehat{\boldsymbol{x}}_{l*} = \boldsymbol{e}_{l*}\boldsymbol{A}$ are far from the corresponding observations $\boldsymbol{x}_{l*}$. Moreover, it is common for hyperspectral images to present up to 20% of unusable spectral bands due to low signal-to-noise ratio essentially from atmospheric effects, such as water absorption. In the following section, we overcome this difficulty by considering the correntropy maximization principle from the information theoretic learning, which yields an optimization problem that is robust to outliers.

6.3 Correntropy-based Unmixing Problems

In this section, we present the correntropy and write the unmixing problems as correntropy maximization ones. Algorithms for solving these problems are derived in Section 6.4.

6.3.1 Correntropy

Within the framework of information theoretic learning, the correntropy was introduced as a generalized correlation function between two stochastic processes, with geometric and probabilistic interpretations [Principe, 2010]. With close connections with the M-estimation, it has been very useful in non-Gaussian signal processing, especially for impulsive noise environments [Liu et al., 2007]. The correntropy can be viewed as a nonlinear local similarity measure between two arbitrary random variables. For two random variables, $\boldsymbol{v}$ and its estimation $\widehat{\boldsymbol{v}}$ using some model/algorithm, it is defined by

$$\mathbb{E}[\kappa(\boldsymbol{v}, \widehat{\boldsymbol{v}})], \tag{6.4}$$

where $\mathbb{E}[\cdot]$ is the expectation operator, and $\kappa(\cdot, \cdot)$ is a shift-invariant positive definite kernel (see Chapter 2). The joint distribution of the variables $\boldsymbol{v}$ and $\widehat{\boldsymbol{v}}$ being unavailable in practice, the sample estimator of the correntropy is adopted instead by employing a finite set of data.

In the following, we consider the same notation as in the unmixing problem, where the variables are the observed spectrum and its estimation using some model/algorithm, namely $(\boldsymbol{x}_{l*}, \widehat{\boldsymbol{x}}_{l*})$. Considering a finite set of these pairs, namely $\{(\boldsymbol{x}_{l*}, \widehat{\boldsymbol{x}}_{l*})\}_{l=1}^{L}$, the correntropy is estimated by

$$\sum_{l=1}^{L} \kappa(\boldsymbol{x}_{l*}, \widehat{\boldsymbol{x}}_{l*}), \tag{6.5}$$

up to a normalization factor. The Gaussian kernel is the most commonly-used kernel for the correntropy. This leads to the following expression for the correntropy

$$\mathcal{C} = \sum_{l=1}^{L} \exp\left(\tfrac{-1}{2\sigma^2} \|\boldsymbol{x}_{l*} - \widehat{\boldsymbol{x}}_{l*}\|_2^2\right), \tag{6.6}$$

where σ denotes the bandwidth of the Gaussian kernel.

The maximization of the correntropy, given by

$$\max_{\boldsymbol{A}} \mathcal{C}$$

where $\boldsymbol{A}$ is the model's parameters, and is termed the maximum correntropy criterion. This criterion has been largely investigated for many applications in statistical signal processing and machine learning, including adaptive systems, unsupervised learning, and pattern recognition [Principe, 2010]. Equivalently, we consider in the following the minimization of the objective function $-\mathcal{C}$, which is often termed the negative of the correntropy.

It is noteworthy that well-known second-order statistics, such as the mean square error (MSE) depends heavily on the Gaussian assumption [Liu et al., 2007]. However, in presence of non-Gaussian noise and in particular large outliers (*i.e.*, observations greatly deviated from the data bulk), the effectiveness of the MSE-based algorithms will significantly deteriorate [Wu et al., 2015]. By contrast, the maximization of the correntropy criterion is appropriate for non-Gaussian signal processing, since it is robust in particular against large outliers, as shown next.

6.3.2 The underlying robustness of the correntropy criterion

In this section, we study the sensitivity to outliers of the correntropy maximization principle, by showing the robustness of the underlying mechanism. To this end, we examine the behavior of the correntropy in terms of the residual error defined by

$$\epsilon_l = \|\boldsymbol{x}_{l*} - \widehat{\boldsymbol{x}}_{l*}\|_2.$$

Thus, the correntropy defined in (6.6) becomes

$$\mathcal{C} = \sum_{l=1}^{L} \exp\left(\tfrac{-1}{2\sigma^2}\,\epsilon_l^2\right).$$

Compared with second-order statistics, *e.g.* MSE with $\frac{1}{L}\sum_{l=1}^{L}\epsilon_l^2$, the correntropy is more robust with respect to the outliers. This is illustrated in the profiles given in Figure 6.1 for $L = 1$, by comparing ϵ_l^2 and $1-\mathcal{C}$ in terms of the residual error ϵ_l. As the residual error increases, the second-order function keeps increasing dramatically. On the

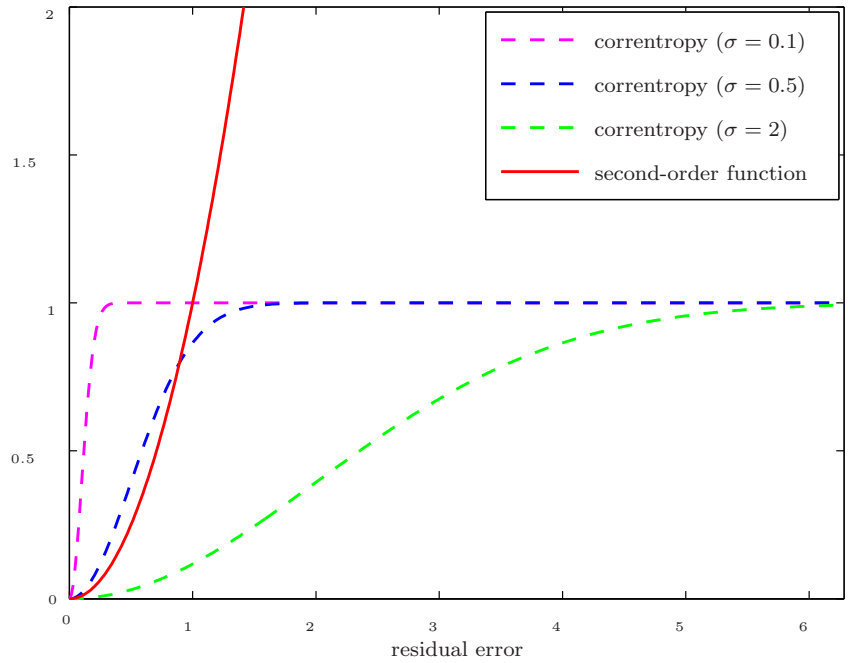


Figure 6.1: Illustration of the second-order objective function (ϵ_l^2, in solid red line) and the negative correntropy objective function ($1 - \mathcal{C}$, in dashed lines for different values of the kernel bandwidth), in terms of the residual error (ϵ_l).

contrary, the correntropy is only sensitive within a region of small residual errors, this region being controlled by the kernel bandwidth. For large magnitudes of the residual error, the correntropy falls to zero. Consequently, the correntropy criterion is robust to large outliers. In the following, we take advantage of this property in order to provide robust unmixing.

6.3.3 Correntropy-based unmixing problems

The correntropy-based unmixing problem consists in estimating the unknown abundance matrix $\boldsymbol{A}$, by minimizing the objective function $-\mathcal{C}$ (the negative of the correntropy), defined by

$$-\mathcal{C}(\boldsymbol{A}) = -\sum_{l=1}^{L} \exp\left(\tfrac{-1}{2\sigma^2}\|\boldsymbol{x}_{l*} - \boldsymbol{e}_{l*}\boldsymbol{A}\|_2^2\right), \tag{6.7}$$

where the Gaussian kernel was considered. Equivalently, using element-wise notation, we have

$$-\mathcal{C}(\boldsymbol{A}) = -\sum_{l=1}^{L} \exp\left(\tfrac{-1}{2\sigma^2}\sum_{t=1}^{T}\left(x_{lt} - \sum_{n=1}^{N} a_{nt}\, e_{ln}\right)^2\right).$$

Considering both the ANC and ASC constraints, the fully-constrained correntropy unmixing problem is defined by the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}} -\mathcal{C}(\boldsymbol{A}),\ \text{subject to } \boldsymbol{A} \geq 0 \\ \text{and } \mathbf{1}^\top \boldsymbol{a}_t = 1,\ \text{for } t = 1, \ldots, T. \end{aligned} \tag{6.8}$$

For the sake of promoting sparse representations, the objective function (6.7) can be augmented by the ℓ_1-norm penalty on the abundance matrix $\boldsymbol{A}$, leading to the following optimization problem:

$$\min_{\boldsymbol{A}} -\mathcal{C}(\boldsymbol{A}) + \lambda \sum_{t=1}^{T} \|\boldsymbol{a}_t\|_1,\ \text{subject to } \boldsymbol{A} \geq 0. \tag{6.9}$$

6.4 ADMM for Solving the Correntropy-based Unmixing Problems

We first briefly review the alternating direction method of multipliers (ADMM), following the expressions in [Boyd et al., 2011, Chap. 3]. Consider an optimization problem of the form

$$\min_{\boldsymbol{a}}\ f(\boldsymbol{a}) + g(\boldsymbol{a}),$$

where the functions f and g are closed, proper and convex. The ADMM solves the equivalent constrained problem

$$\min_{\boldsymbol{a},\boldsymbol{b}} f(\boldsymbol{a}) + g(\boldsymbol{b}) \text{ subject to } \boldsymbol{P}\boldsymbol{a} + \boldsymbol{Q}\boldsymbol{b} = \boldsymbol{c}, \tag{6.10}$$

such as having the particular constraint $\boldsymbol{a} = \boldsymbol{b}$ for instance. While this formulation may seem trivial, the optimization problem can now be tackled using the augmented Lagrangian method where the objective function is separable in $\boldsymbol{a}$ and $\boldsymbol{b}$. By alternating on each variable separately, the ADMM repeats a direct update of the dual variable. Its relevance to solve nonconvex problems is studied in [Boyd et al., 2011, Section 9]. In its scaled form, the ADMM algorithm is summarized in Algorithm 3.

Algorithm 3 The ADMM algorithm [Boyd et al., 2011]

Input: functions f and g, matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$, vector $\boldsymbol{c}$, parameter ρ

1: Initialize $k = 0$, $\boldsymbol{a}_0$, $\boldsymbol{b}_0$ and $\boldsymbol{d}_0$
2: **repeat**
3: $\quad \boldsymbol{a}_{k+1} = \arg\min_{\boldsymbol{a}} f(\boldsymbol{a}) + \frac{\rho}{2}\|\boldsymbol{P}\boldsymbol{a} + \boldsymbol{Q}\boldsymbol{b}_k - \boldsymbol{c} + \boldsymbol{d}_k\|_2^2$;
4: $\quad \boldsymbol{b}_{k+1} = \arg\min_{\boldsymbol{b}} g(\boldsymbol{b}) + \frac{\rho}{2}\|\boldsymbol{P}\boldsymbol{a}_{k+1} + \boldsymbol{Q}\boldsymbol{b} - \boldsymbol{c} + \boldsymbol{d}_k\|_2^2$;
5: $\quad \boldsymbol{d}_{k+1} = \boldsymbol{d}_k + \boldsymbol{P}\boldsymbol{a}_{k+1} + \boldsymbol{Q}\boldsymbol{b}_{k+1} - \boldsymbol{c}$;
6: $\quad k = k + 1$;
7: **until** stopping criterion

6.4.1 Correntropy-based unmixing with full constraints

In the following, we apply the ADMM algorithm to solve the correntropy-based unmixing problem in the fully-constrained case, presented in (6.8). Rewrite the variables to be optimized in a vector $\boldsymbol{a} \in \mathbb{R}^{RT\times 1}$, which is stacked by the columns of the matrix $\boldsymbol{A}$, namely $\boldsymbol{a} = [\boldsymbol{a}_1^\top \ \cdots \ \boldsymbol{a}_T^\top]^\top$. Rewrite also the following vectors in $\mathbb{R}^{RT\times 1}$: $\boldsymbol{b} = [\boldsymbol{b}_1^\top \ \cdots \ \boldsymbol{b}_T^\top]^\top$ and $\boldsymbol{d} = [\boldsymbol{d}_1^\top \ \cdots \ \boldsymbol{d}_T^\top]^\top$, where $\boldsymbol{b}_t = [z_{1t} \ \cdots \ z_{Nt}]^\top$ and $\boldsymbol{d}_t = [u_{1t} \ \cdots \ u_{Nt}]^\top$, for $t = 1, \ldots, T$. By following the formulation of the ADMM in Algorithm 3, we set

$$
\begin{aligned}
f(\boldsymbol{a}) &= -\mathcal{C}(\boldsymbol{a}) + \sum_{t=1}^{T} \iota_{\{1\}}(\boldsymbol{1}^\top \boldsymbol{a}_t) \\
g(\boldsymbol{b}) &= \iota_{\mathbb{R}_+^{RT}}(\boldsymbol{b}) \\
\boldsymbol{P} &= -\boldsymbol{I}, \boldsymbol{Q} = \boldsymbol{I} \ \text{ and } \ \boldsymbol{c} = \boldsymbol{0},
\end{aligned}
\tag{6.11}
$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{0} \in \mathbb{R}^{RT\times 1}$ is the zero vector and $\iota_{\mathcal{S}}(u)$ is the indicator function of the set $\mathcal{S}$ defined by

$$
\iota_{\mathcal{S}}(u) = \begin{cases} 0 & \text{if } u \in \mathcal{S}; \\ \infty & \text{otherwise.} \end{cases}
$$

Before that, we eliminate the T equality constraints, *i.e.*, the sum-to-one constraints, by replacing a_{Nt} with

$$
a_{Nt} = 1 - \sum_{n=1}^{N-1} a_{nt},
$$

for $t = 1, \ldots, T$. Let $\overline{\boldsymbol{a}} \in \mathbb{R}^{(N-1)T \times 1}$ be the reduced vector of $(N-1)$ unknowns to be estimated, stacked as follows

$$\overline{\boldsymbol{a}_t} = \begin{bmatrix} a_{1t} & \cdots & a_{(N-1)t} \end{bmatrix}^\top,$$

for $t = 1, \ldots, T$. By this means, the objective function in (6.11) is transformed from (6.7) into the reduced-form

$$f_1(\overline{\boldsymbol{a}}) = -\sum_{l=1}^{L} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=1}^{T} \epsilon_l(\overline{\boldsymbol{a}_t})^2\right), \tag{6.12}$$

where $\epsilon_l(\overline{\boldsymbol{a}_t}) = x_{lt} - e_{lN} - \sum_{p=1}^{N-1}(e_{lp} - e_{lN})a_{pt}$, for $l = 1, \ldots, L$. The gradient of (6.12) with respect to $\overline{\boldsymbol{a}}$ is stacked as

$$\frac{\partial f_1}{\partial \overline{\boldsymbol{a}}} = \left[\frac{\partial f_1}{\partial \overline{\boldsymbol{a}}_1}^\top \cdots \frac{\partial f_1}{\partial \overline{\boldsymbol{a}}_T}^\top\right]^\top \in \mathbb{R}^{(N-1)T \times 1},$$

where $\frac{\partial f_1}{\partial \overline{\boldsymbol{a}}_t} = \left[\frac{\partial f_1}{\partial \overline{a}_{1t}} \cdots \frac{\partial f_1}{\partial \overline{a}_{(N-1)t}}\right]^\top$, with the entries given by

$$\frac{\partial f_1(\overline{\boldsymbol{a}})}{\partial \overline{a_{nt}}} = \tfrac{1}{\sigma^2} \sum_{l=1}^{L} (e_{lN} - e_{ln}) \exp\left(\tfrac{-1}{2\sigma^2} \sum_{s=1}^{T} \epsilon_l(\overline{\boldsymbol{a}}_s)^2\right) \epsilon_l(\overline{\boldsymbol{a}_t}),$$

for all $n = 1, \ldots, (N-1)$ and $t = 1, \ldots, T$. Similarly, the function $\frac{\rho}{2}\|\boldsymbol{a} - \boldsymbol{b}_k - \boldsymbol{d}_k\|_2^2$ is expressed with respect to $\overline{\boldsymbol{a}}$ as

$$\phi(\overline{\boldsymbol{a}}) = \frac{\rho}{2} \sum_{t=1}^{T} \left(1 - \sum_{p=1}^{N-1} a_{pt} - z_{Nt,k} - u_{Nt,k}\right)^2 + \sum_{p=1}^{N-1} \left(a_{pt} - z_{pt,k} - u_{pt,k}\right)^2$$

with the entries in its gradient $\frac{\partial \phi}{\partial \overline{\boldsymbol{a}}}$ given by

$$\frac{\partial \phi(\overline{\boldsymbol{a}})}{\partial \overline{a_{nt}}} = \rho\Big(a_{nt} + \sum_{p=1}^{N-1} a_{pt} - 1 + z_{Nt,k} - z_{nt,k} + u_{Nt,k} - u_{nt,k}\Big), \tag{6.13}$$

for all $n = 1, \ldots, N-1$ and $t = 1, \ldots, T$.

The main steps of the algorithm CUSAL-FC (for correntropy-based unmixing with full constraints) are given in Algorithm 4. The subproblem of the $\boldsymbol{a}$-update (in line 3 of Algorithm 3) addresses a nonconvex problem without any closed-form solution. To

Algorithm 4 Correntropy-based unmixing with full constraints (**CUSAL-FC**)

1: Initialize $k = 0$, $\rho > 0$, $\eta > 0$, $\sigma > 0$; $\boldsymbol{a}_0$, $\boldsymbol{b}_0$ and $\boldsymbol{d}_0$;
2: **repeat**
3: **repeat**
4: $\overline{\boldsymbol{a}}_{k+1} = \overline{\boldsymbol{a}}_{k+1} - \eta\Big(\frac{\partial f_1}{\partial \overline{\boldsymbol{a}}_{k+1}} + \frac{\partial \phi}{\partial \overline{\boldsymbol{a}}_{k+1}}\Big)$;
5: **until** convergence
6: reform $\boldsymbol{a}_{k+1}$ using $\overline{\boldsymbol{a}}_{k+1}$;
7: $\boldsymbol{b}_{k+1} = \max(\boldsymbol{0}, \boldsymbol{a}_{k+1} - \boldsymbol{d}_k)$;
8: $\boldsymbol{d}_{k+1} = \boldsymbol{d}_k - (\boldsymbol{a}_{k+1} - \boldsymbol{b}_{k+1})$;
9: $k = k + 1$;
10: **until** stopping criterion

overcome this difficulty, we apply an ADMM variant that solves the subproblem iteratively using the gradient descent method, instead of solving it exactly and explicitly. This step is given in lines 3-5 of Algorithm 4. The solution of the $\boldsymbol{b}$-update in line 4 of Algorithm 3 becomes the projection of $\boldsymbol{a}_{k+1} - \boldsymbol{d}_k$ onto the first orthant, as shown in line 7 of Algorithm 4.

6.4.2 Sparsity-promoting unmixing algorithm

In order to apply the ADMM algorithm for the correntropy-based sparsity-promoting unmixing problem, we express the constrained optimization problem (6.9) as follows

$$\begin{aligned} f(\boldsymbol{a}) &= -\mathcal{C}(\boldsymbol{a}) \\ g(\boldsymbol{a}) &= \iota_{\mathbb{R}_+^{RT}}(\boldsymbol{a}) + \lambda\|\boldsymbol{a}\|_1 \\ \boldsymbol{P} &= -\boldsymbol{I}, \boldsymbol{Q} = \boldsymbol{I} \quad \text{and} \quad \boldsymbol{c} = \boldsymbol{0}. \end{aligned} \tag{6.14}$$

By analogy with the previous case, the $\boldsymbol{a}$-update in line 3 of Algorithm 3 is solved iteratively with the gradient descent method and is given in Algorithm 5 lines 3-5. The gradient of (6.14) with respect to $\boldsymbol{a}$ is stacked by $\frac{\partial f}{\partial \boldsymbol{a}_t}$, where

$$\frac{\partial f}{\partial \boldsymbol{a}_t} = -\frac{1}{\sigma^2}\sum_{l=1}^{L} \epsilon_l(\boldsymbol{a}_t) \exp\left(\frac{-1}{2\sigma^2}\sum_{s=1}^{T}(\epsilon_l(\boldsymbol{a}_s))^2\right) \boldsymbol{e}_l^\top,$$

for $t = 1, \ldots, T$, where $\epsilon_l(\boldsymbol{a}_t) = x_{lt} - \sum_{n=1}^{N} a_{nt}\, e_{ln}$.

Algorithm 5 Correntropy-based unmixing with sparsity-promoting (**CUSAL-SP**)

1: Initialize $k = 0$, $\rho > 0$, $\sigma > 0$, $\eta > 0$, $\lambda > 0$; $\boldsymbol{a}_0$, $\boldsymbol{b}_0$ and $\boldsymbol{d}_0$;
2: **repeat**
3: **repeat**
4: $\boldsymbol{a}_{k+1} = \boldsymbol{a}_{k+1} - \eta\Big(\dfrac{\partial f}{\partial \boldsymbol{a}_{k+1}} + \rho(\boldsymbol{a}_{k+1} - \boldsymbol{b}_k - \boldsymbol{d}_k)\Big)$;
5: **until** convergence
6: $\boldsymbol{b}_{k+1} = \max(\boldsymbol{0}, S_{\lambda/\rho}(\boldsymbol{a}_{k+1} - \boldsymbol{d}_k))$;
7: $\boldsymbol{d}_{k+1} = \boldsymbol{d}_k - (\boldsymbol{a}_{k+1} - \boldsymbol{b}_{k+1})$;
8: $k = k + 1$;
9: **until** stopping criterion

The $\boldsymbol{b}$-update in line 4 Algorithm 3 involves solving

$$\boldsymbol{b}_{k+1} = \arg\min_{\boldsymbol{b}} \iota_{\mathbb{R}_+^{RT}}(\boldsymbol{b}) + (\lambda/\rho)\|\boldsymbol{b}\|_1 + \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{a}_{k+1} - \boldsymbol{d}_k\|_2^2. \tag{6.15}$$

In [Boyd et al., 2011], the ADMM has been applied to solve various ℓ_1-norm problems, including the well-known LASSO [Tibshirani, 1996]. The only difference between (6.15) and the $\boldsymbol{b}$-update in LASSO is that in the latter, no nonnegativity term $\iota_{\mathbb{R}_+^{RT}}(\boldsymbol{b})$ is enforced. In this case, the $\boldsymbol{b}$-update in LASSO is the element-wise soft thresholding operation

$$\boldsymbol{b}_{k+1} = S_{\lambda/\rho}(\boldsymbol{a}_{k+1} - \boldsymbol{d}_k),$$

where the soft thresholding operator [Boyd et al., 2011] is defined by

$$S_b(\zeta) = \begin{cases} \zeta - b & \text{if } \zeta > b; \\ 0 & \text{if } \|\zeta\| < b; \\ \zeta + b & \text{if } \zeta < -b. \end{cases}$$

Following [Bioucas-Dias and Figueiredo, 2010], it is straightforward to project the result onto the nonnegative orthant in order to include the nonnegativity constraint, thus yielding

$$\boldsymbol{b}_{k+1} = \max(\boldsymbol{0}, S_{\lambda/\rho}(\boldsymbol{a}_{k+1} - \boldsymbol{d}_k)),$$

where the maximum function is element-wise. All these results lead to the correntropy-based unmixing algorithm with sparsity-promoting, as summarized in Algorithm 5.

6.4.3 On the initialisation and the bandwidth determination

We apply a three-fold stopping criterion for Algorithms 4 and 5, as recommended in general by [Boyd et al., 2011] and in [Bioucas-Dias and Figueiredo, 2010] for hyperspectral unmixing (with the SUnSAL algorithm) :

(i) the primal and dual residuals are small enough, namely $\|\boldsymbol{a}_{k+1} - \boldsymbol{b}_{k+1}\|_2 \leq \epsilon_1$ and $\rho\|\boldsymbol{b}_{k+1} - \boldsymbol{b}_k\|_2 \leq \epsilon_2$,

(ii) the primal residual starts to increase, *i.e.*, $\|\boldsymbol{a}_{k+1} - \boldsymbol{b}_{k+1}\|_2 > \|\boldsymbol{a}_k - \boldsymbol{b}_k\|_2$, or

(iii) the maximum iteration number is attained.

The threshold parameters are set to $\epsilon_1 = \epsilon_2 = \sqrt{RT} \times 10^{-5}$, as recommended in [Bioucas-Dias and Figueiredo, 2010].

The bandwidth σ in the Gaussian kernel should be well-tuned. A small value for this parameter punishes harder the outlier bands, thus increasing the robustness of the algorithm to outliers [He et al., 2011]. Moreover, the ADMM is applied to address a nonconvex objective function, thus no convergence is guaranteed theoretically, according to [Boyd et al., 2011]. Considering these issues, we propose to fix the bandwidth empirically as summarized in Algorithm 6 and described next. Following [He et al., 2011; Wang et al., 2015], we first initialize the bandwidth parameter as a function of the reconstruction error, given by

$$\sigma_0^2 = \frac{N}{8L}\|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}_{\mathrm{LS}}\|_F^2, \tag{6.16}$$

where $\boldsymbol{A}_{\mathrm{LS}}$ is the least-squares solution (6.2). In the case of a result too apart from that of least-squares solution, the parameter is augmented by $\sigma = 1.2\sigma$, until the condition

$$\frac{\|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}\|_F}{\|\boldsymbol{X} - \boldsymbol{E}\boldsymbol{A}_{\mathrm{LS}}\|_F} < 2$$

is satisfied. The algorithm divergence occurs if the stopping criterion (ii) is satisfied, namely the primal residual increases over iterations. In this case, either the parameter is too large due to an overestimated initialization, or it is too small. Accordingly, we either decrease it by $\sigma = \sigma_0/p$, or increase it by $\sigma = 1.2\sigma$, until the convergence of the ADMM algorihtm.

Algorithm 6 Tuning the bandwidth parameter σ

1: Initialize $\sigma = \sigma_0$ using (6.16); $p = 1$;
2: Do CUSAL with Algorithm 4 or Algorithm 5;
3: **if** stopping criterion (i) or (iii) is satisfied **then**
4: **if** condition $\frac{\|\boldsymbol{X}-\boldsymbol{E}\boldsymbol{A}\|_2}{\|\boldsymbol{X}-\boldsymbol{E}\boldsymbol{A}_{\mathrm{LS}}\|_2} < 2$ is satisfied, **then**
5: $\sigma^* = \sigma$ (optimal value)
6: **else**
7: increase $\sigma = 1.2\sigma$, and go to line 2
8: **end if**
9: **else**
10: **if** $\sigma > 1000\sigma_0$ (due to the overestimated σ_0) **then**
11: $p = p + 1$;
12: decrease $\sigma = \sigma_0/p$, and go to line 2
13: **else**
14: increase $\sigma = 1.2\sigma$, and go to line 2
15: **end if**
16: **end if**

6.5 Experiments

In this section, the performance of the proposed CUSAL algorithms, in both the fully-constrained (CUSAL-FC) and sparsity-promoting (CUSAL-SP) versions, are evaluated on synthetic and real hyperspectral images. A comparative study is performed considering six state-of-the-art methods proposed for linear and nonlinear unmixing models, described in Chapter 1: FCLS [Heinz and Chang, 2001], SUnSAL-FCLS [Bioucas-Dias and Figueiredo, 2010], BayGBM [Halimi et al., 2011a,b], BayPPNMM [Altmann et al., 2012], KFCLS [Broadwater et al., 2007] and the robust NMF (rNMF) [Févotte and Dobigeon, 2015]. Three metrics are considered to evaluate the unmixing performance, namely the root mean square error of abundances (RMSE) as defined in (4.16) (see Section 4.4), the averaged spectral angle distance between the input spectra and the reconstructed ones (SAD) as defined in (4.15) (see Section 4.4), as well as the signal-to-reconstruction error (SRE) to be defined in (6.17) for sparsity-promoting algorithms.

6.5.1 Experiments with synthetic data

In this section, the performance of the proposed fully-constrained (CUSAL-FC) and sparsity-promoting (CUSAL-SP) algorithms is evaluated on synthetic data.

We first compare the fully-constrained CUSAL-FC, presented in Section 6.4.1, with the state-of-the-art methods. A series of experiments are performed, mainly considering the

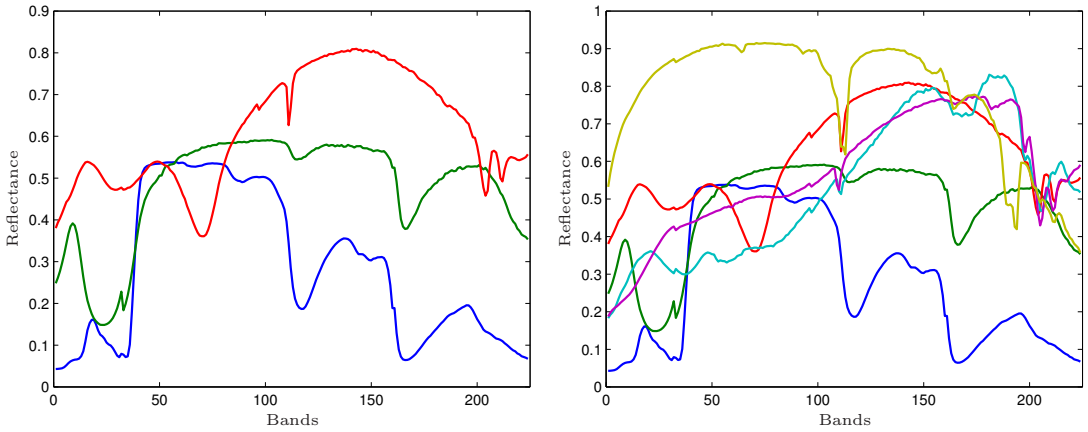
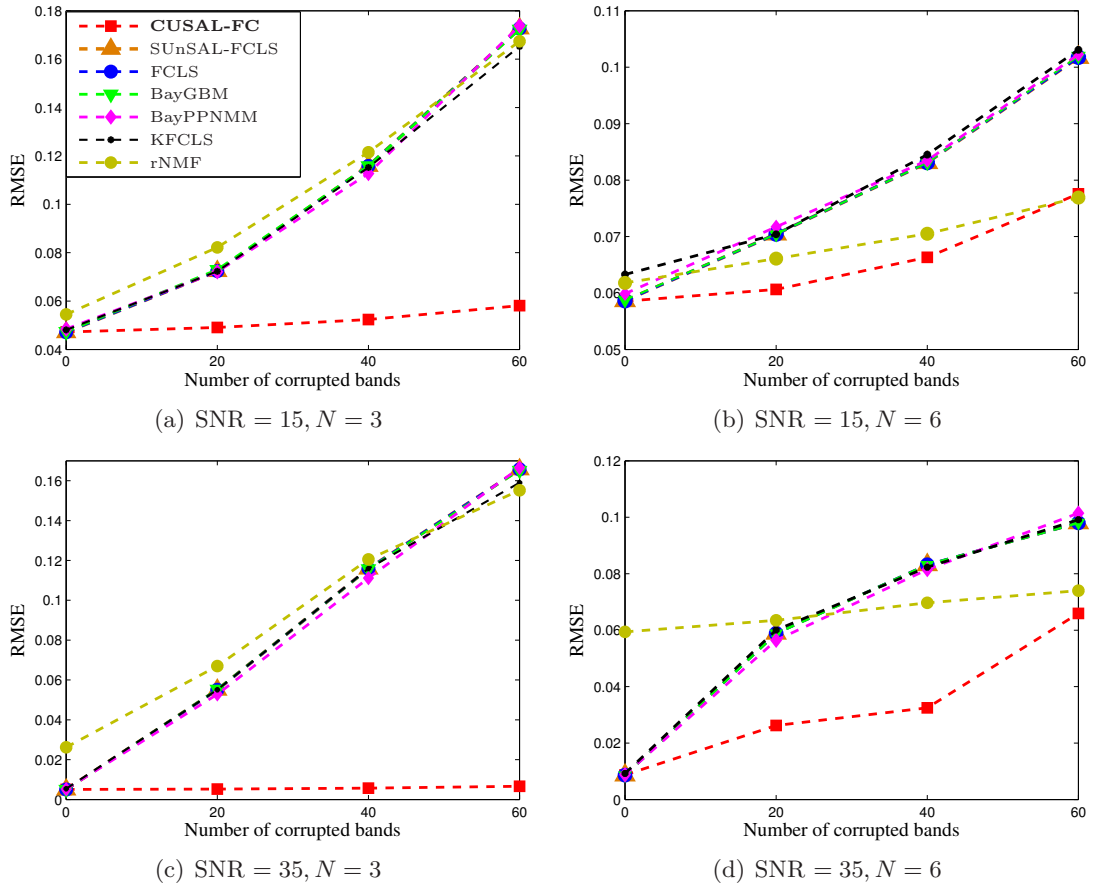


Figure 6.2: The $N = 3$ (left) and 6 (right) endmembers chosen for simulation from the USGS.

influence of four aspects: (i) mixture model, (ii) noise level, (iii) number of corrupted bands and (iv) number of endmembers.

Each image, of 50×50 pixels, is generated using either the linear mixing model or the polynomial post-nonlinear mixing model (PPNMM) (1.7), where the additive Gaussian noise has a SNR $\in \{15, 35\}$dB. The $N \in \{3, 6\}$ endmembers, as shown in Figure 6.2, are drawn from the USGS digital spectral library [Bioucas-Dias and Nascimento, 2008]. These endmembers are defined over $L = 244$ continuous bands with the wavelength ranging from $0.2\mu m$ to $3.0\mu m$. The abundance vectors $\boldsymbol{a}_t$ are uniformly generated using a Dirichlet distribution as in [Bioucas-Dias and Nascimento, 2008; Halimi et al., 2015a]. For PPNMM, the values of b_t are generated uniformly in the set $(-3, 3)$ according to [Altmann et al., 2012]. To imitate the noisy bands in the real hyperspectral images, several bands in the generated data are corrupted by replacing the corresponding rows of $\boldsymbol{X}$ with random values within $[0, 1]$. The number of corrupted bands varies in the set $\{0, 20, 40, 60\}$.

The unmixing performance is evaluated using the abundance root mean square error (RMSE), as defined in Section 4.4. Figure 6.3 and 6.4 illustrates the average of RMSE over 10 Monte-Carlo realizations, respectively on the LMM and PPNMM data. It is easy to see that, in presence of outlier bands, the proposed CUSAL-FC algorithm outperforms all the methods in terms of RMSE, for all mixture models, all noise levels and all numbers of endmembers. It is also shown that the performance of the proposed algorithm improves when increasing the SNR.

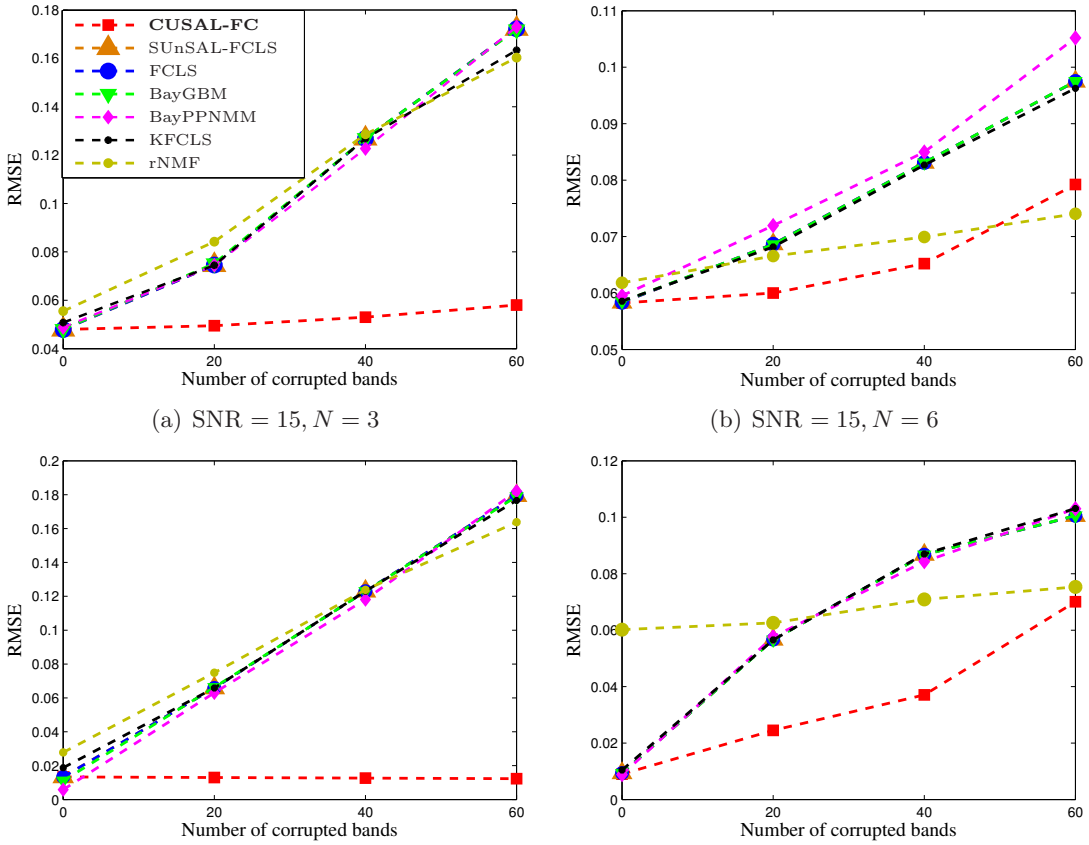


(a) SNR $= 15, N = 3$ (b) SNR $= 15, N = 6$

(c) SNR $= 35, N = 3$ (d) SNR $= 35, N = 6$

Figure 6.3: LMM data: The root mean square error (RMSE) with respect to the number of corrupted bands, averaged over ten Monte-Carlo realizations, for different number of endmembers and SNR.

The performance of the proposed the sparsity-promoting CUSAL-SP, presented in 6.4.2, is compared with the sparsity-promoting SUnSAL-sparse, as well as the FCLS, on a series of images with sparse abundance matrices. The influences of (i) the number of corrupted bands and (ii) the sparsity level of the abundances, are studied. Each image, of 15×15 pixels, is generated by the linear mixture model. The endmember matrix is composed of $N = 62$ signatures from the USGS, where the angle between any two different endmembers is larger than 10° [Iordache et al., 2011]. The K nonzero entries in each abundance vector $\boldsymbol{a}_t$ are generated by a Dirichlet distribution. The value of K (*i.e.*, the indicator of sparsity level) ranges from 4 to 20, while the number of corrupted bands varies in $\{0, 20, 40, 60\}$. We set the Gaussian noise by SNR $= 30$dB, a level that is commonly present in real hyperspectral images according to [Iordache et al., 2011]. For both sparsity-promoting algorithms, the regularization parameter λ is adjusted using the set $\{10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.



(a) SNR $= 15, N = 3$ (b) SNR $= 15, N = 6$

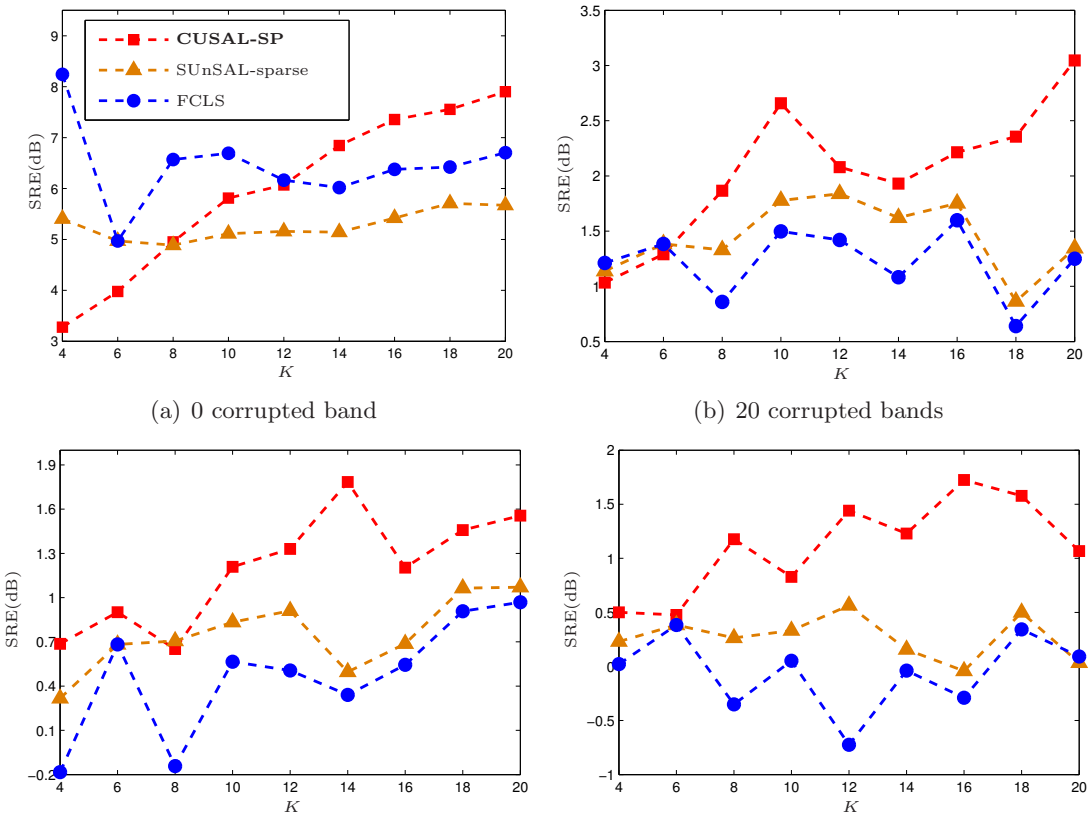
(c) SNR $= 35, N = 3$ (d) SNR $= 35, N = 6$

Figure 6.4: PPNMM data: The root mean square error (RMSE) with respect to the number of corrupted bands, averaged over ten Monte-Carlo realizations, for different number of endmembers and SNR.

The unmixing performance with the sparsity-promoting algorithms is evaluated using the signal-to-reconstruction error, measured in decibels, according to [Bioucas-Dias and Figueiredo, 2010; Iordache et al., 2011]. It is defined by

$$\text{SRE} = 10 \log_{10} \left(\frac{\sum_{t=1}^{T} \|\boldsymbol{a}_t\|_2^2}{\sum_{t=1}^{T} \|\boldsymbol{a}_t - \widehat{\boldsymbol{a}}_t\|_2^2} \right). \tag{6.17}$$

The results, averaged over ten Monte-Carlo realizations, are illustrated in Figure 6.5. Considering that the abundance matrix under estimation is sparse at different levels, we conclude the following: Concerning the case without outlier bands, the CUSAL-SP outperforms the SUnSAL-sparse for all values of $K > 8$ and FCLS for all values of $K > 12$. When the number of outlier bands increases, the proposed CUSAL-SP algorithm generally provides the best unmixing quality with the highest SRE value, especially for $K > 6$.



(a) 0 corrupted band

(b) 20 corrupted bands

(c) 40 corrupted bands

(d) 60 corrupted bands

Figure 6.5: LMM data: The averaged signal-to-reconstruction error (SRE) with respect to the sparsity level K, averaged over ten Monte-Carlo realizations. Comparison for various number of corrupted bands at SNR $= 30$.

6.5.2 Experiments with real data

This section presents the performance of the proposed algorithms on a real hyperspectral image. We consider a 250×190 sub-image taken from the Cuprite mining image, acquired by the AVIRIS sensor when flying over Las Vegas, Nevada, USA. The image has been widely investigated in the literature [Chen et al., 2013b; Iordache et al., 2011]. The raw data contains $L = 224$ spectral bands, covering a wavelength range $0.4\mu m - 2.5\mu m$. From these spectral bands, there are 37 relatively noisy ones with low SNR, namely the bands $1 - 3$, $105 - 115$, $150 - 170$, and $223 - 224$. The geographic composition of this area is estimated to include up to 14 minerals [Nascimento and Bioucas-Dias, 2005]. Neglecting the relatively similar signatures, we consider 12 endmembers as often investigated in the literature [Lu et al., 2013; Chen et al., 2013b]. The VCA technique is first applied to extract these endmembers on the clean image with $L = 187$ bands. Starting from $L = 187$ bands, the noisy bands, randomly chosen from the bands $1 - 3$,

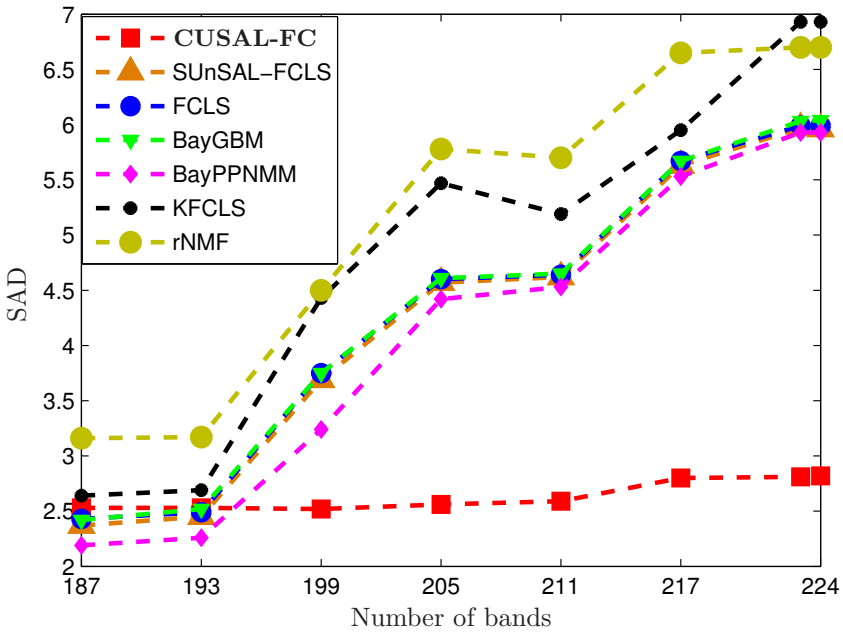


Figure 6.6: Cuprite image: The averaged spectral angle distance (SAD) using different number of bands, by starting with 187 clean spectral bands and gradually including the noisy bands.

$105-115$, $150-170$, and $223-224$, are gradually included to form a series of input data. Therefore, the experiments are conducted with $L = 187, 193, 199, 205, 211, 217, 223$ and 224 bands.

Since ground-truth abundances are unknown, the performance is measured with the averaged spectral angle distance (SAD) between the input spectra $\boldsymbol{x}_t$ and the reconstructed ones $\widehat{\boldsymbol{x}}_t$, as defined in (4.15) (see Section 4.4). The results are illustrated in Figure 6.6. The estimated abundance maps using 187, 205 and 224 bands are given in Figure 6.7, Figure 6.8, and Figure 6.9, respectively. In absence of noisy bands (*i.e.*, $L = 187$ bands), all the considered methods lead to satisfactory abundance maps, with BayPPNMM providing the smallest SAD and rNMF the worst. As the number of noisy bands increases, the unmixing performance of the state-of-the-art methods deteriorates drastically, while the proposed CUSAL yields stable SAD. The obtained results confirm the good behavior of the proposed CUSAL algorithms and their robustness in presence of corrupted spectral bands.

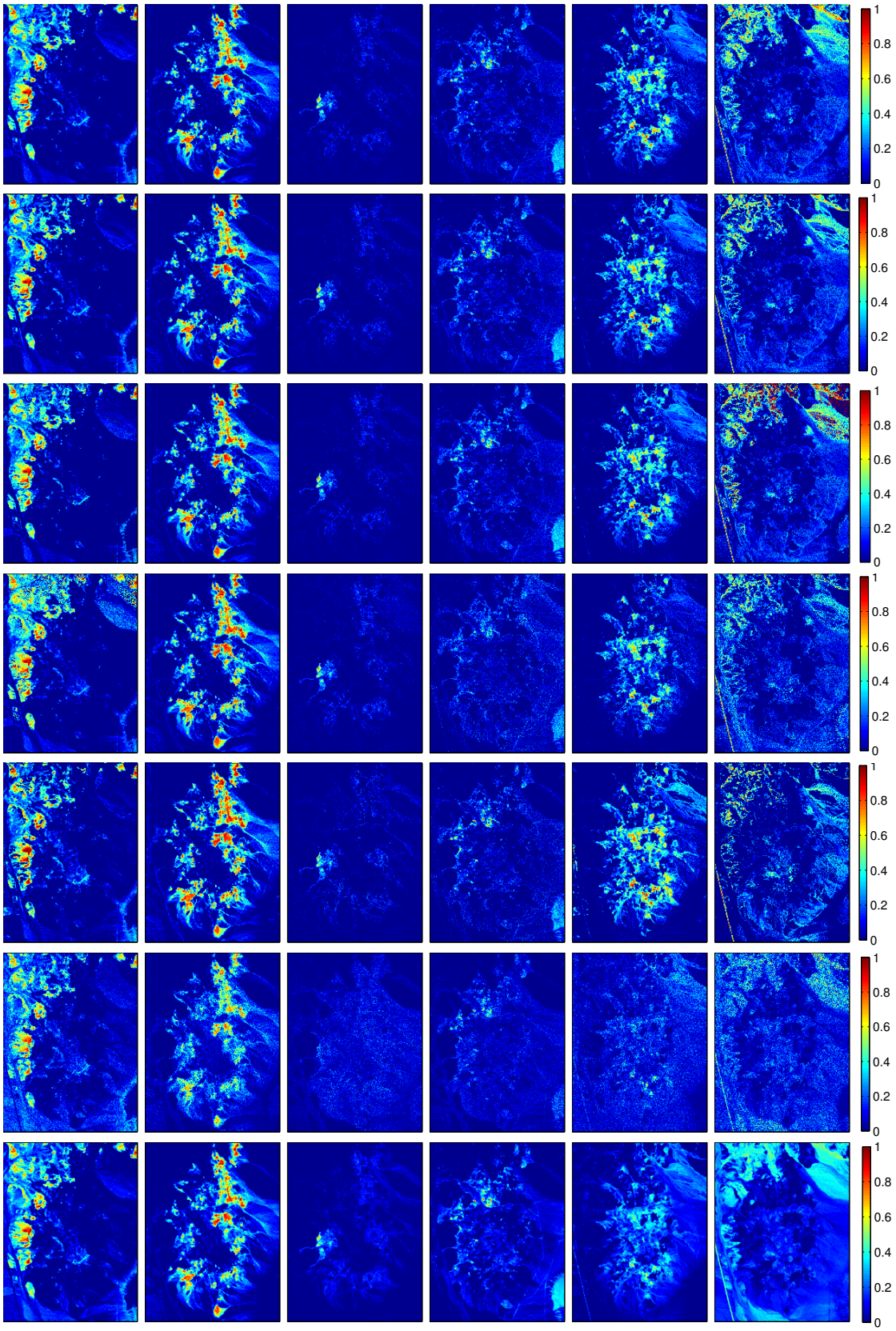


Figure 6.7: Cuprite image: Estimated abundance maps using 187 clean bands. **Left to right:** sphene, alunite, buddingtonite, kaolinite, chalcedony, highway. **Top to bottom:** SUnSAL-FCLS, FCLS, BayGBM, BayPPNMM, KFCLS, rNMF, CUSAL-FC.

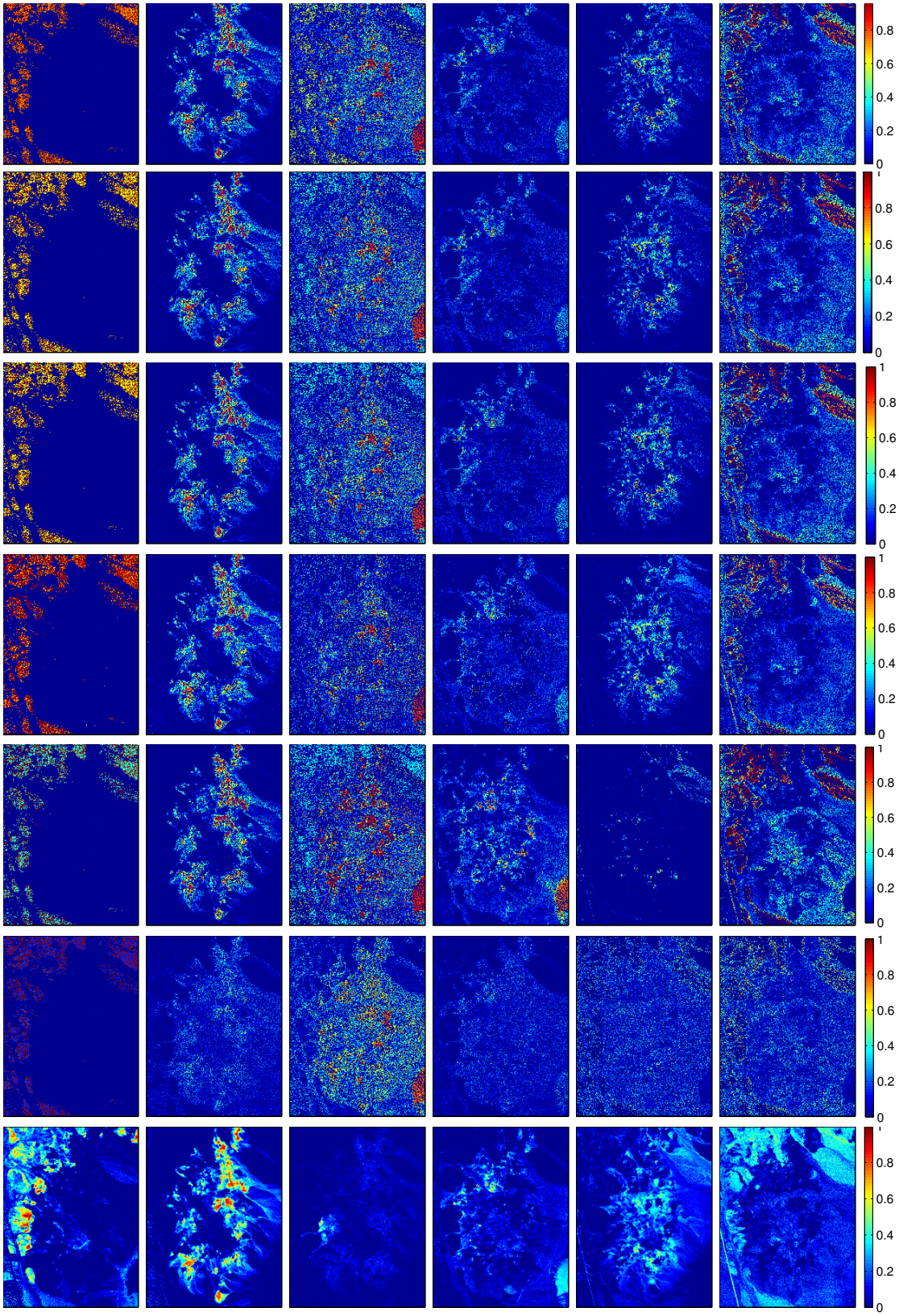


Figure 6.8: Cuprite image: Estimated abundance maps using 205 bands, with 187 clean bands. Same legend as Figure 6.7.

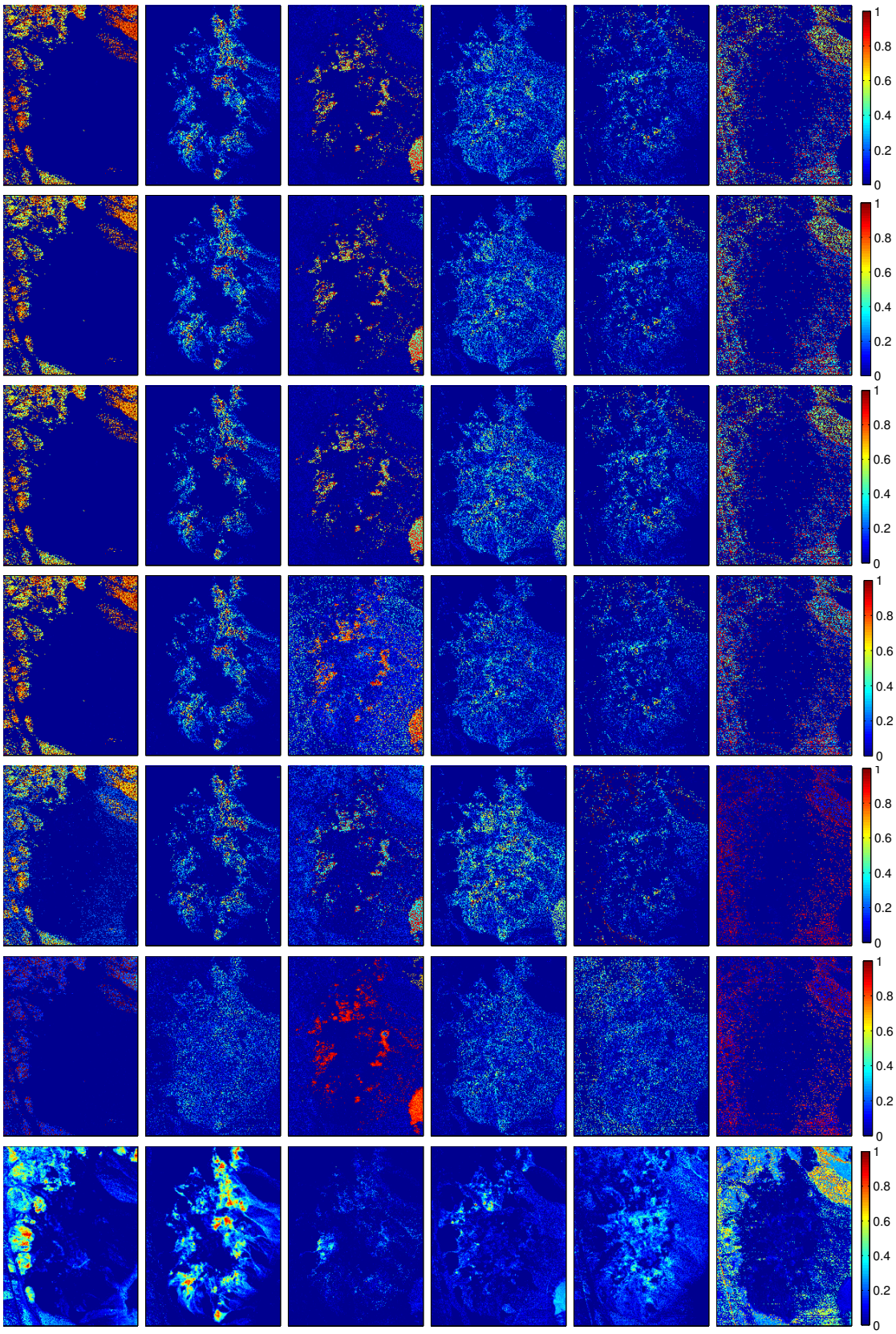


Figure 6.9: Cuprite image: Estimated abundance maps using all the 224 bands, with 187 clean bands. Same legend as Figure 6.7.

6.6 Conclusion

This chapter presented supervised robust unmixing algorithms based on the correntropy maximization principle, with robustness in terms of outliers and corrupted spectral bands. Two correntropy-based unmixing problems were addressed, the first with the nonnegativity and sum-to-one constraints, and the second with the nonnegativity constraint and a sparsity-promoting term. The alternating direction method of multipliers (ADMM) was investigated in order to solve the correntropy-based unmixing problems. The effectiveness and robustness of the proposed unmixing method were validated on synthetic and real hyperspectral images. Future works include the generalization of the correntropy criterion to account for the multiple reflection phenomenon [Halimi et al., 2011a; Fan et al., 2009], as well as incorporating nonlinear models [Halimi et al., 2015b]. Of great interest is robust unsupervised unmixing, with the joint estimation of endmembers and abundances.

Chapter 7

Conclusions and Perspectives

Contents

In this manuscript, we have investigated several methods to infer hidden patterns from data, by considering the hyperspectral unmixing problem. First, a kernel-based nonnegative matrix factorization (KNMF) model was proposed, which bypasses the preimage problem inherit from the kernel machines. To handle large-scale and streaming data, we also extended the the proposed method to an online mode, by maintaining a fixed and tractable computational complexity and memory usage. Next, we studied the bi-objective optimization problem that performs the NMF in both input and feature spaces, by combining the linear and kernel-based models. Finally, independent from the proposed KNMF, we derived a supervised unmixing method based on the correntropy maximization principle, which is shown to be robust to corrupted spectral bands. This chapter summarizes the thesis by outlining the aforementioned methods and by discussing future research directions.

7.1 Conclusion

The work of thesis consists in proposing several methods within the framework of kernel methods in machine learning, in order to address the hyperspectral unmixing problem. The main contributions can be summarized as follows.

In Chapter 3, we presented a new kernel-based NMF (KNMF) model that determines the decomposition bases directly in the input space, without suffering from the pre-image problem. The resulting optimization problem was solved using two-blocks coordinate descent alternating technique, where the additive and multiplicative update rules were proposed. Furthermore, motivated by the unmixing task in hyperspectral imagery, several extensions were derived with constraints imposed on the endmembers and the abundances, such as sparseness, smoothness, and spatial regularization with a total-variation-like penalty.

In Chapter 4, we extended the KNMF to an online mode in order to tackle large-scale and streaming dynamic data. We derived both the additive and multiplicative update rules of the general form, by investigating the stochastic gradient descent and the mini-batch strategies. The case of the Gaussian kernel was studied in details. For commonly-used kernels, the proposed algorithms keep a fixed and tractable complexity independent

of the increasing number of samples. Several extensions were considered within the proposed online framework, namely sparse coding and smoothness of the basis vectors.

In Chapter 5, we proposed a bi-objective KNMF by exploiting the kernel machines, where the decomposition was performed simultaneously in the input and the feature spaces. The first objective function to optimize stems from the conventional linear NMF, while the second one is from the KNMF. When these objective functions are conflicting, there exists a set of nondominated, noninferior or Pareto optimal solutions. By taking advantage of the sum-weighted method, we broke the original bi-objective problem into a sequence of single-objective optimization problems, each corresponding to a fusion of the linear and nonlinear models at a different level. The update rules were derived. Last, the corresponding Pareto front was approximated and analyzed.

In Chapter 6, we proposed a supervised spectral unmixing approach robust to outlier spectral bands, by investigating the maximization of the correntropy criterion. We derived two unmixing problems, the first one is the fully-constrained unmixing, and the second one is sparsity-promoting unmixing. Taking the advantages of alternating direction method of multipliers (ADMM), the corresponding optimization problems were solved efficiently.

7.2 Future Works

This thesis provided several important solutions for the hyperspectral unmixing problems. As part of future research, we would like to investigate the following aspects concerning improvements of our proposed methods.

- In Chapter 3, the optimization problem related to the proposed KNMF model was solved using the multiplicative update rules, following the spirit in linear NMF. However, due to the nonlinearity and nonconvexity in terms of the subproblem with respect to the endmembers/bases, the convergence to a stationary point is not guaranteed. Notice that KNMF can be viewed an extreme case of the bi-objective NMF. Therefore, the projected gradient descent (PGD) algorithm described in Algorithm 2 in Chapter 4 can be applied to yield stationary points. The disadvantage

is that the stepsize searching procedure in PGD is very time-consuming. Applying more efficient techniques, such as ADMM, to solve this nonlinear, nonconvex optimization problem deserves attention in the future.

- The bi-objective KNMF proposed in Chapter 5 detects holistically the nonlinearity in an image, namely each Pareto solution corresponds to a fusion of the conventional linear NMF and the kernel-based KNMF at a certain level. However, for real scenarios, it is more reasonable to assume that the nonlinearity varies from one pixel to another. To this end, a pixel-wise mixture model that determines the nonlinearity at each pixel should be considered as an improvement of the bi-objective KNMF model, from the perspective of hyperspectral unmixing. To this end, we rewrite the cost contributed by each pixel $\boldsymbol{x}_t$ separately, with

$$\min_{\boldsymbol{E},\boldsymbol{A},\boldsymbol{\mu}} \quad \sum_{t=1}^{T} \frac{\|\boldsymbol{x}_t - \sum_{n=1}^{N} a_{nt}\boldsymbol{e}_n\|^2}{\mu_t} + \frac{\|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\,\Phi(\boldsymbol{e}_n)\|^2_{\mathcal{H}}}{1-\mu_t} \tag{7.1}$$
$$\text{subject to} \quad \boldsymbol{E}, \boldsymbol{A} \geq 0 \text{ and } \mu_t \in (0,1) \text{ for } t = 1, \ldots, T.$$

Here, the t-th element μ_t of $\boldsymbol{\mu}$ corresponds to the nonlinearity level at pixel $\boldsymbol{x}_t$. Notice that a pixel tends to be linearly mixed when its nonlinearity level is close to 0, while a value near 1 means that the pixel is highly nonlinearly mixed. To solve the above problem, a three-block coordinate descent optimization alternating over $\boldsymbol{E}$, $\boldsymbol{A}$ and $\boldsymbol{\mu}$ could be considered. The balance in (7.1) between the linear and nonlinear functional norms (denoted respectively a and b) allows to have a convex cost function over $\mu_t \in (0,1)$ for fixed $\boldsymbol{E}$ and $\boldsymbol{A}$, with a closed-form solution for the optimum at $\mu^* = (1+\sqrt{b/a})^{-1}$. See [Chen et al., 2013a] and references therein.

- In Chapter 6, we established the correntropy maximization criterion under the assumption of the linear mixture model. Future works include the generalization of the correntropy criterion to account for the multiple reflection phenomenon [Halimi et al., 2011a; Fan et al., 2009], as well as incorporating recent advances in nonlinear models including variabilities [Halimi et al., 2015b]. As a result, existing nonlinear unmixing models could become robust to outlier bands. Moreover, the problem of extracting the endmembers using a robust unmixing method will receive attention in the future, namely by considering a robust KNMF.

Bibliography

Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret. Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery. *IEEE Transactions on Image Processing*, 21(6):3017–3025, 2012. 14, 96, 109, 144, 145

Y. Altmann, N. Dobigeon, S. McLaughlin, and J.-Y. Tourneret. Nonlinear spectral unmixing of hyperspectral images using gaussian processes. *IEEE Transactions on Signal Processing*, 61(10):2442–2453, May 2013. 20

Y. Altmann, N. Dobigeon, S. McLaughlin, and J.-Y. Tourneret. Residual component analysis of hyperspectral images - application to joint nonlinear unmixing and nonlinearity detection. *IEEE Transactions on Image Processing*, pages 2148–2158, 2014. 19, 108, 110

S. An, J.-M. Yun, and S. Choi. Multiple kernel nonnegative matrix factorization. In *Proc. of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1976–1979. IEEE, 2011. 53, 54

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950. 28, 29, 30, 31

M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. Huntington. Ice: a statistical approach to identifying endmembers in hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 42(10):2085–2095, Oct 2004. 6

J. M. Bioucas-Dias and M. A. Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *Proc. of IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, 2010. 7, 11, 12, 130, 133, 142, 143, 144, 147

J. M. Bioucas-Dias and J. M. P. Nascimento. Hyperspectral subspace identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435–2445, Aug 2008. 95, 121, 145

J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, April 2012. 6, 8, 11, 12, 130

X. Blasco, J. M. Herrero, J. Sanchis, and M. Martinez. A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908 – 3924, 2008. 119

L. Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pages 421–436. Springer, 2012. 83, 86, 87

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 7, 131, 138, 139, 142, 143

J. Broadwater and A. Banerjee. A comparison of kernel functions for intimate mixture models. In *Proc. of 2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, Aug 2009. 17

J. Broadwater, R. Chellappa, A. Banerjee, and P. Burlina. Kernel fully constrained least squares abundance estimates. In *Proc. of 2007 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4041–4044. IEEE, 2007. 16, 17, 130, 144

J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(12):4164–4169, 2004. 48

S. S. Bucak and B. Gunsel. Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, 42(5):788–797, 2009. 82, 84, 89, 94

G. Buchsbaum and O. Bloch. Color categories revealed by non-negative matrix factorization of munsell color spectra. *Vision Research*, 42(5):559–63, 2002. 48

I. Buciu and I. Pitas. Application of non-negative and local non negative matrix factorization to facial expression recognition. In *Proc. of 17th International Conference on Pattern Recognition*, volume 1, pages 288–291, Cambridge, UK, 2004. 48

I. Buciu, N. Nikolaidis, and I. Pitas. Nonnegative matrix factorization in polynomial feature space. *IEEE Transactions on Neural Networks*, 19(6):1090–1100, Jun. 2008. 53, 54

C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998. 26, 28, 38

B. Cao, D. Shen, J. Sun, X. Wang, Q. Yang, and Z. Chen. Detect and track latent factors with online nonnegative matrix factorization. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 2689–2694, 2007. 3, 82, 94

C. Chang, S. Chiang, A. Smith, and I. Ginsberg. Linear spectral random mixture analysis for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 40(2):375–392, Feb 2002. 7

J. Chen, C. Richard, and P. Honeine. Nonlinear unmixing of hyperspectral images based on multi-kernel learning. In *Proc. of IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Jun. 2012. 16, 19, 108, 110

J. Chen, C. Richard, A. Ferrari, and P. Honeine. Nonlinear unmixing of hyperspectral data with partially linear least-squares support vector regression. In *Proc. of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2174–2178, May 2013a. 158

J. Chen, C. Richard, and P. Honeine. Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model. *IEEE Transactions on Signal Processing*, 61(2):480–492, Jan. 2013b. 7, 16, 18, 57, 70, 72, 108, 110, 120, 130, 148

J. Chen, C. Richard, and P. Honeine. Estimating abundance fractions of materials in hyperspectral images by fitting a post-nonlinear mixing model. In *Proc. of IEEE Workshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing*, Jun. 2013c. 7, 16, 19, 57, 108, 110, 130

J. Chen, C. Richard, and P. Honeine. Nonlinear estimation of material abundances of hyperspectral images with ℓ_1-norm spatial regularization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2654–2665, May 2014. 19

Z. Chen and A. Cichocki. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. In *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep*, 2005. 64, 93

A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley Publishing, 2009. 2, 3, 48

R. Clark, G. Swayze, and A. Gallagher. Mapping minerals with imaging spectroscopy. *US Geological Survey, Office of Mineral Resources Bulletin*, 2039:141–150, 1993. 69

R. Close, P. Gader, J. Wilson, and A. Zare. Using physics-based macroscopic and microscopic mixture models for hyperspectral pixel unmixing. In *Proc. of SPIE Defense, Security, and Sensing*, pages 83901L–83901L. International Society for Optics and Photonics, 2012a. 16

R. Close, P. Gader, A. Zare, J. Wilson, and D. Dranishnikov. Endmember extraction using the physics-based multi-mixture pixel model. In *Proc. of SPIE Optical Engineering+ Applications*, pages 85150L–85150L. International Society for Optics and Photonics, 2012b. 16

R. Close, P. Gader, and J. Wilson. Hyperspectral unmixing using macroscopic and microscopic mixture models. *Journal of Applied Remote Sensing*, 8(1):083642–083642, 2014. 16, 57, 110, 112, 120

P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. Academic Press, Mar. 2010. ISBN 978-0-12-374726-6. 3, 4, 5, 7, 48, 52

I. Das and J. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997. 114, 115, 125

K. Deb and D. Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 047187339X. 125

K. Devarajan. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Comput Biol*, 4(7), Jul. 2008. 48

C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. of SIAM Data Mining Conference*, pages 606–610, 2005. 3, 48

C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135, New York, NY, USA, 2006. ACM. 49

C. Ding, T. Li, and M. I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, Nov. 2010. 4, 45, 50, 53, 71, 110, 111, 120

N. Dobigeon, J.-Y. Tourneret, and C.-I. Chang. Semi-supervised linear spectral unmixing using a hierarchical bayesian model for hyperspectral imagery. *IEEE Transactions on Signal Processing*, 56(7):2684–2695, Jul. 2008. 69

N. Dobigeon, J.-Y. Tourneret, C. Richard, J. Bermudez, S. McLaughlin, and A. Hero. Nonlinear unmixing of hyperspectral images: Models and algorithms. *IEEE Signal Processing Magazine*, 31(1):82–94, Jan 2014. 6, 12, 18, 109

D. Dranishnikov, P. Gader, A. Zare, and T. Glenn. Unmixing using a combined microscopic and macroscopic mixture model with distinct endmembers. In *Proc. of the 21st European Signal Processing Conference (EUSIPCO)*, pages 1–5, Sept 2013. 16

M. Essoloh, C. Richard, H. Snoussi, and P. Honeine. Distributed localization in wireless sensor networks as a pre-image problem in a reproducing kernel hilbert space. In *Proc. of 16th European Conference on Signal Processing*, pages 1–5, Lausanne, Switzerland, August 2008. 40, 45

W. Fan, B. Hu, J. Miller, and M. 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. 13, 153, 158

C. Févotte and N. Dobigeon. Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 24(12):4810–4819, Dec 2015. 7, 8, 144

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative Matrix Factorization with the Itakura-Saito Divergence: With Application to Music Analysis. *Neural Computation*, 21(3):793–830, Sep. 2008. 3, 48, 95

M. Fong and Z. Hu. Hyperactive: Hyperspectral image analysis toolkit. *UCLA Dept of Math, http://www. math. ucla. edu/wittman/hyper/hypermanual. pdf, accessed Apr.*, 3, 2011. 123

Z. Ghahramani. Unsupervised learning. In *Advanced lectures on machine learning*, pages 72–112. Springer, 2004. 26

N. Gillis. *Nonnegative Matrix Factorization: Complexity, Algorithms and Applications*. PhD thesis, Université catholique de Louvain, Feb. 2011. 49

N. Gillis. The why and how of nonnegative matrix factorization. *ArXiv e-prints*, Jan. 2014. 48, 52

E. Gonzales and Y. Zhang. Accelerating the Lee-Seung algorithm for non-negative matrix factorization. Technical report, Department of Computational and Applied Mathematics, Rice University, 2005. 119

N. Guan, D. Tao, Z. Luo, and B. Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1087–1099, July 2012. 82, 83, 84, 87, 89, 91, 94, 95

D. Guillamet, M. Bressan, and J. Vitria. A weighted non-negative matrix factorization for local representations. In *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–942–I–947 vol.1, 2001. 48

A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret. Nonlinear unmixing of hyperspectral images using a generalized bilinear model. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4153–4162, Nov. 2011a. 14, 57, 109, 110, 144, 153, 158

A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret. Unmixing hyperspectral images using the generalized bilinear model. In *Proc. of IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1886–1889, 2011b. 7, 14, 144

A. Halimi, N. Dobigeon, and J.-Y. Tourneret. Unsupervised unmixing of hyperspectral images accounting for endmember variability. *IEEE Transactions on Image Processing*, 24(12):4904–4917, december 2015a. 145

A. Halimi, P. Honeine, and J. M. Bioucas-Dias. Hyperspectral unmixing in presence of endmember variability, nonlinearity or mismodelling effects. *http://arxiv.org/abs/1511.05698*, pages 1–32, Nov. 2015b. 153, 158

A. Halimi, P. Honeine, M. Kharouf, C. Richard, and J.-Y. Tourneret. Estimating the intrinsic dimension of hyperspectral images using a noise-whitened eigengap approach. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–11, (in press) 2016. 5

B. Hapke. Bidirectional reflectance spectroscopy: 1. theory. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 86(B4):3039–3054, 1981. 15, 56, 57, 110

J. C. Harsanyi and C. I. Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4):779–785, Jul 1994. 6

R. He, W. Zheng, and B. Hu. Maximum correntropy criterion for robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1561–1576, 2011. 131, 143

M. Hein and O. Bousquet. Kernels, associated structures and generalizations. *Max-Planck-Institut fuer biologische Kybernetik, Technical Report*, 2004. 34

D. Heinz and C. 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, Mar. 2001. 6, 11, 70, 110, 120, 133, 144

R. Heylen, M. Parente, and P. Gader. A review of nonlinear hyperspectral unmixing methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):1844–1868, June 2014. 4, 10, 109

J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. of The 30th International Conference on Machine Learning*, pages 1480–1488, 2013. 83

P. Honeine and C. Richard. Solving the pre-image problem in kernel machines: A direct method. In *Proc. of 2009 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sept 2009. 44

P. Honeine and C. Richard. Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, 28(2):77–88, 2011. 19, 40, 41

P. Honeine and C. Richard. Geometric unmixing of large hyperspectral images: a barycentric coordinate approach. *IEEE Transactions on Geoscience and Remote Sensing*, 50(6):2185–2195, Jun. 2012. 7

P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. 49, 65, 99

K. Huang, N. Sidiropoulos, and A. Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, Jan. 2014. 49

A. Huck, M. Guillaume, and J. Blanc-Talon. Minimum dispersion constrained non-negative matrix factorization to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(6):2590–2602, Jun. 2010. 7, 8, 11, 70, 110, 111, 117, 120

M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2014–2039, June 2011. 10, 12, 133, 146, 147, 148

M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4484–4502, Nov. 2012. 12, 49, 66, 133

M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 52 (1):341–354, 2014. 12

S. Jia and Y. Qian. Spectral and spatial complexity-based hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):3867–3879, Dec. 2007. 123

S. Jia and Y. Qian. Constrained nonnegative matrix factorization for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1):161–173, Jan. 2009. 3, 7, 11, 49

M. Kallas, P. Honeine, C. Richard, C. Francis, and H. Amoud. Non-negative pre-image in machine learning for pattern recognition. In *Proc. of 2011 19th European Signal Processing Conference (EUSIPCO)*, pages 931–935, Aug 2011. 54

M. Kallas, P. Honeine, C. Francis, and H. Amoud. Kernel autoregressive models using yule-walker equations. *Signal Processing*, 93(11):3053–3061, November 2013a. 40, 45

M. Kallas, P. Honeine, C. Richard, C. Francis, and H. Amoud. Non-negativity constraints on the pre-image for pattern recognition with kernel machines. *Pattern Recognition*, 46(11):3066 – 3080, 2013b. 50, 54

N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, Jan 2002. 12

H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23 (12):1495–1502, June 2007. 3, 49

H. Kim and H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, Jul. 2008. 49, 71

P. M. Kim and B. Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome research*, 13(7):1706–1718, 2003. 48

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82 – 95, 1971. 36

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009. 3

J. Kwok and I. W. H. Tsang. The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks*, 15(6):1517–1525, Nov 2004. 42, 43

J. T. Kwok and I. W. Tsang. The pre-image problem in kernel methods. In *Proc. of 2003 International Conference on Machine Learning (ICML)*, pages 408–415, 2003. 43, 44

J. Lampinen. Multiobjective nonlinear pareto-optimization. *Pre-investigation Report, Lappeenranta University of Technology*, 2000. 114, 125

H. Lantéri, M. Roche, O. Cuevas, and C. Aime. A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints. *Signal Processing*, 81:945–974, May 2001. 133

H. Lantéri, C. Theys, C. Richard, and D. Mary. Regularized split gradient method for nonnegative matrix factorization. In *Proc. of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1133–1136, 2011. 59

C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems (Classics in Applied Mathematics)*. Society for Industrial Mathematics, 1987. ISBN 0898713560. 89, 133

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct. 1999. 3, 7, 8, 48, 49, 51

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of Advances in Neural Information Processing Systems*, pages 556–562, 2001. 3, 49, 59, 111, 118

H. Lee, A. Cichocki, and S. Choi. Kernel nonnegative matrix factorization for spectral EEG feature extraction. *Neurocomputing*, 72(13-15):3182–3190, 2009. 54, 111

A. Lefevre, F. Bach, and C. Févotte. Online algorithms for nonnegative matrix factorization with the itakura-saito divergence. In *Proc. of 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 313–316. IEEE, 2011. 82

D. Leggett. Numerical analysis of multicomponent spectra. *Analytical Chemistry*, 49: 276—281, 1977. 49

H. Li, T. Adal, W. Wang, D. Emge, and A. Cichocki. Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 48(1-2): 83–97, 2007. 49

S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Proc. of the 2001 IEEE Computer Society Conference onComputer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–207–I–212 vol.1, 2001. 48

T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Proc. of the 6th International Conference on Data Mining (ICDM)*, pages 362–371, 2006. ISBN 0-7695-2701-9. 48

Y. Li and A. Ngom. A new kernel non-negative matrix factorization and its application in microarray data analysis. In *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 371–378, San Diego, CA, USA, May. 2012. 4, 45, 50, 53, 54, 71, 111, 120

C. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596, Nov 2007a. 118, 119

C. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007b. 58, 117

W. Liu and N. Zheng. Non-negative matrix factorization based methods for object recognition. *Pattern Recognition Letters*, 25(8):893–897, 2004. 48

W. Liu, P. Pokharel, and J. C. Príncipe. Correntropy: properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11): 5286–5298, 2007. 130, 135, 136

X. Liu, W. Xia, B. Wang, and L. Zhang. An approach based on constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(2):757–772, Feb 2011. 101

R. Lorenzo and G. Durrett. Reproducing kernel hilbert spaces. Technical report, Massachusetts Institute of Technology, 2010. 28

X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li. Manifold regularized sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5): 2815–2826, May 2013. 7, 11, 148

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010. 3, 83, 84, 87, 94

Y. M. Masalmah and M. Veléz-Reyes. A full algorithm to compute the constrained positive matrix factorization and its application in unsupervised unmixing of hyperspectral imagery. In *Proc. of SPIE Defense and Security Symposium*, pages 69661C–69661C. International Society for Optics and Photonics, 2008. 50

L. Miao and H. 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, March 2007. 7

K. Miettinen. Introduction to multiobjective optimization: Noninteractive approaches. In *Multiobjective Optimization*, pages 1–26. Springer, 2008. 114, 119

S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Proc. of 1998 conference on advances in neural information processing systems II*, pages 536–542, Cambridge, MA, USA, 1999. MIT Press. ISBN 0-262-11245-0. 38, 42, 50

T. M. Mitchell. The Discipline of Machine Learning. Technical report, School of Computer Science, Carnegie Mellon University, Jul 2006. 26

S. Moussaoui, H. Hauksdottir, F. Schmidt, C. Jutten, J. Chanussot, D. Brie, S. Douté, and J. A. Benediktsson. On the decomposition of mars hyperspectral data by ica and bayesian positive source separation. *Neurocomputing*, 71(10):2194–2208, 2008. 7

J. Nascimento and J. M. Bioucas-Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, Apr. 2005. 6, 11, 70, 110, 148

J. M. Nascimento and J. M. Bioucas-Dias. Nonlinear mixture model for hyperspectral unmixing. In *Proc. of SPIE Europe Remote Sensing*, pages 74770I–74770I. International Society for Optics and Photonics, 2009. 13

N. Nguyen, J. Chen, C. Richard, P. Honeine, and C. Theys. Supervised nonlinear unmixing of hyperspectral images using a pre-image method. In *New Concepts in Imaging: Optical and Statistical Models, In Eds. D. Mary, C. Theys, and C. Aime*, volume 59 of *EAS Publications Series*, pages 417–437. EDP Sciences, 2013. 16, 19, 40, 45, 110

M. Nikolova and M. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific computing*, 27(3):937–966, 2005. 131

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. 49

B. Pan, J. Lai, and W. Chen. Nonlinear nonnegative matrix factorization based on Mercer kernel construction. *Pattern Recognition*, 44(10-11):2800 – 2810, 2011. 54, 71, 111, 120

V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and its Applications*, 416(1):29 – 47, 2006. 3, 49

J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin. Object characterization from spectral data using nonnegative factorization and information theory. In *Proc. of AMOS Technical Conference*, 2004. 62, 92

J. Plaza, E. Hendrix, I. García, G. Martín, and A. Plaza. On endmember identification in hyperspectral images without pure pixels: A comparison of algorithms. *Journal of Mathematical Imaging and Vision*, 42(2-3):163–175, 2012. 6

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. 86, 87

J. C. Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives.* Springer Science & Business Media, 2010. 130, 135, 136

Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly. Hyperspectral unmixing via $l_{1/2}$ sparsity-constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4282–4297, Nov. 2011. 3, 49, 99, 101

J. Ryu, S. Kim, and H. Wan. Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization. In *Proc. of 2009 WinterSimulation Conference (WSC)*, pages 623–633, Dec. 2010. 114

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, Jul. 1998. 38

B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In *Computational learning theory*, pages 416–426. Springer, 2001. 36

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521813972. 32, 34

P. Smaragdis. Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs. In *Proc. of the 5th International Conference, on Independent Component Analysis and Blind Signal Separation, ICA 2004*, pages 494–499. Granada, Spain, 22-24 Sep. 2004. 48

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 142

A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics. 28

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, USA, 1995. 28, 38

S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, Oct. 2009. 49, 114, 115

T. Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *Proc. of International Computer Music Conference (ICMC)*, volume 3, pages 231–234, 2003. 63, 93

D. Wang and H. Lu. On-line learning parts-based representation via incremental orthogonal projective non-negative matrix factorization. *Signal Processing*, 93(6):1608–1623, 2013. 82, 95

F. Wang, C. Tan, P. Li, and A. C. König. Efficient document clustering via online nonnegative matrix factorizations. In *Proc. of 11th SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, April 2011. 3, 82, 84, 87, 89, 94, 95

J. Wang and C. Chang. Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2601–2616, Sept 2006. 7

J. J. Wang, X. Wang, and X. Gao. Non-negative matrix factorization by maximizing correntropy for cancer clustering. *BMC bioinformatics*, 14(1):107, 2013a. 3, 131

N. Wang, J. Wang, and D. Y. Yeung. Online robust non-negative dictionary learning for visual tracking. In *Proc. of 2013 IEEE International Conference on Computer Vision (ICCV)*, pages 657–664. IEEE, 2013b. 83

Y. Wang, C. Pan, S. Xiang, and F. Zhu. Robust hyperspectral unmixing with correntropy-based metric. *IEEE Transactions on Image Processing*, 24(11):4027–4040, Nov 2015. 131, 143

S. Wild, J. Curry, and A. Dougherty. Improving non-negative matrix factorizations through structured initialization. *Pattern Recognition*, 37(11):2217–2232, Nov. 2004. 48

C. K. Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002. 43

M. E. Winter. N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *Proc. of SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 266–275. International Society for Optics and Photonics, 1999. 6, 11

Z. Wu, S. Peng, B. Chen, and H. Zhao. Robust hammerstein adaptive filtering under maximum correntropy criterion. *Entropy*, 17(10):7149, 2015. 136

W. Xu. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv preprint arXiv:1107.2490*, 2011. 87

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, New York, NY, USA, 2003. ACM. 3, 48

Z. Yang and E. Oja. Quadratic nonnegative matrix factorization. *Pattern Recognition*, 45(4):1500 – 1510, 2012. 4, 50

Z. Yang, H. Zhang, and E. Oja. Online projective nonnegative matrix factorization for large datasets. In *Proc. of 19th International Conference on Neural Information Processing*, pages 285–290. Springer, 2012. 82, 95

N. Yokoya, J. Chanussot, and A. Iwasaki. Generalized bilinear model based nonlinear unmixing using semi-nonnegative matrix factorization. In *Proc. of 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1365–1368, July 2012. 14

N. Yokoya, J. Chanussot, and A. Iwasaki. Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1430–1437, Feb. 2014. 14, 70, 109, 120

S. Young, P. Fogel, and D. M. Hawkins. Clustering scotch whiskies using non-negative matrix factorization. *Q&SPES News*, 14:11–13, 2006. 48

A. C. Zelinski and V. K. Goyal. Denoising hyperspectral imagery and recovering junk bands using wavelets and sparse approximation. In *Proc. of 2006 IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, pages 387–390, July 2006. 130

D. Zhang, Z. Zhou, and S. Chen. Non-negative matrix factorization on kernels. In *Lecture Notes in Computer Science*, volume 4099, pages 404–412. Springer, 2006. 4, 45, 50, 53, 71, 111

G. Zhou, S. Xie, Z. Yang, J. Yang, and Z. He. Minimum-volume-constrained nonnegative matrix factorization: Enhanced ability of learning parts. *IEEE Transactions on Neural Networks*, 22(10):1626–1637, Oct. 2011a. 49

G. Zhou, Z. Yang, S. Xie, and J. Yang. Online blind source separation using incremental nonnegative matrix factorization with volume constraint. *IEEE Transactions on Neural Networks*, 22(4):550–560, April 2011b. 82, 89, 95

F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan. Spectral unmixing via data-guided sparsity. *IEEE Transactions on Image Processing*, 23(12):5412–5427, 2014. 101

E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, Nov. 1999. 125

Fei ZHU

Doctorat : Optimisation et Sûreté des Systèmes

Année 2016

Factorisation en matrices non négatives à noyaux : application à l'imagerie hyperspectrale

Cette thèse vise à proposer de nouveaux modèles pour la séparation de sources dans le cadre non linéaire des méthodes à noyaux en apprentissage statistique, et à développer des algorithmes associés. Le domaine d'application privilégié est le démélange en imagerie hyperspectrale. Tout d'abord, nous décrivons un modèle original de la factorisation en matrices non négatives (NMF), en se basant sur les méthodes à noyaux. Le modèle proposé surmonte la malédiction de préimage, un problème inverse hérité des méthodes à noyaux. Dans le même cadre proposé, plusieurs extensions sont développées pour intégrer les principales contraintes soulevées par les images hyperspectrales. Pour traiter des masses de données, des algorithmes de traitement en ligne sont développés afin d'assurer une complexité calculatoire fixée. Également, nous proposons une approche de factorisation bi-objective qui permet de combiner les modèles de démélange linéaire et non linéaire, où les décompositions de NMF conventionnelle et à noyaux sont réalisées simultanément. La dernière partie se concentre sur le démélange robuste aux bandes spectrales aberrantes. En décrivant le démélange selon le principe de la maximisation de la correntropie, deux problèmes de démélange robuste sont traités sous différentes contraintes soulevées par le problème de démélange hyperspectral. Des algorithmes de type directions alternées sont utilisés pour résoudre les problèmes d'optimisation associés.

Mots clés : imagerie hyperspectrale - apprentissage automatique - modèles non linéaires (statistique) - factorisation - matrices nonnégatives.

Kernel Nonnegative Matrix Factorization: Application to Hyperspectral Imagery

This thesis aims to propose new nonlinear unmixing models within the framework of kernel methods and to develop associated algorithms, in order to address the hyperspectral unmixing problem.
First, we investigate a novel kernel-based nonnegative matrix factorization (NMF) model, that circumvents the pre-image problem inherited from the kernel machines. Within the proposed framework, several extensions are developed to incorporate common constraints raised in hypersepctral images analysis. In order to tackle large-scale and streaming data, we next extend the kernel-based NMF to an online fashion, by keeping a fixed and tractable complexity. Moreover, we propose a bi-objective NMF model as an attempt to combine the linear and nonlinear unmixing models. The decompositions of both the conventional NMF and the kernel-based NMF are performed simultaneously. The last part of this thesis studies a supervised unmixing model, based on the correntropy maximization principle. This model is shown robust to outlier bands. Two correntropy-based unmixing problems are addressed, considering different constraints in hyperspectral unmixing problem. The alternating direction method of multipliers (ADMM) is investigated to solve the related optimization problems.

Keywords: hyperspectral imagery - machine learning - nonlinear models - factorization (mathematics)- non-negative matrices.

Thèse réalisée en partenariat entre :

Ecole Doctorale "Sciences et Technologies"