Università degli Studi di Bari "Aldo Moro"
Dipartimento di Informatica
Dottorato di Ricerca in Informatica
CICLO XXVII

Non-negative factorization methods for extracting semantically relevant features in Intelligent Data Analysis

Gabriella Casalino

S.S.D.: INF/01

Supervisor: Dott. Corrado Mencar
Co-supervisor: Prof. Nicoletta Del Buono
Coordinator: Prof. Donato Malerba

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Science

ESAME FINALE 2015

Abstract

Intelligent Data Analysis (IDA) is a methodology for extracting useful knowledge from data, with special emphasis on human involvement in the analysis process. Within IDA, dimensionality reduction methods play an important role, as they enable to represent data in low-dimensional spaces. With this representation, it is indeed possible to discover hidden structures in data by disregarding irrelevant information.

Non-negative Matrix Factorization (NMF) is a low-rank approximation method that is widely used for dimensionality reduction and clustering. Its characteristic non-negativity constraint leads to representing data as linear additive combinations of latent factors, which, in turn, can be interpreted as building-blocks of the final data. NMF can play a prominent role among dimensionality reduction methods within IDA, yet classical approaches to NMF may fail to provide data representations that are semantically relevant, hence easily interpretable, for the data analyst.

In this thesis, new variants of NMF have been proposed, with the aim of extracting semantically relevant features, so as to improve the usefulness of NMF within IDA. The common theme of these variants is the ability of injecting prior information in the factorization process. The first proposal concerns an initialization method for NMF based on Subtractive fuzzy Clustering (SC). In fact, NMF needs some initial matrices before starting the factorization process. Several alternatives exist, but most of them require the a-priori specification of the rank, i.e. the dimensionality of the new subspace the data will lie in the new representation. The use of SC enables to automatically determine the rank, by exploiting some additional information concerning

the similarity of data being provided by the analyst. This approach has been applied to document clustering, where an improvement of the interpretability of the results coming from NMF has been empirically observed. Moreover, when NMF is used for clustering documents in latent topics, the resulting prototypes are more representative of these topics than when other initialization techniques are used.

The second proposal is mainly focused on the optimization process of NMF. The point of departure is the observation that classical NMF returns a new representation of data that can be hardly described in terms of parts, a part being a selection of features of the original space where a linear correlation holds for a subset of data. To enforce a part-based representation, the standard NMF optimization process has been modified so as to take into account a binary mask that regulates the factorization process so as to describe data as composition of parts conforming to the mask. The resulting Masked NMF (MNMF) puts at the analyst's disposal a tool to query the dataset so as to extract a subset of the available data which can be represented in terms of user-specified parts. This approach, called Query-based NMF, is accompanied by some metrics that evaluate the quality of the query in terms of representativeness of MNMF results as well as their conformity to the query. The whole approach has been tested on some synthetic data and a benchmark dataset so as to show the potential benefits within IDA.

The third and last proposal is a modification of Non-negative Matrix Underapproximation (NMU), which in turn is a variant of NMF where the factorization process is carried out via an iterative approximation of the original data matrix in rank-one matrices. Here, NMU has been modified in order to accommodate some constraints that enhance the interpretability of the final results. In essence, these constraints enforce sparsity and spatial (local) information, thus resulting particularly suited for hyper-spectral images, where the proposed approach has been successfully applied to classify the pixels of real-world images according to the materials of the scanned objects.

These three proposals show that a proper injection of expert knowledge in the factorization process enables the discovery of hidden structures in data that could be easily interpreted by the data analyst. In all cases, an interaction is established between the analyst and the computational machinery, thus achieving an intelligent support for data analysis.

Part of this research work has been accomplished within a 5-months studentship at the University of Mons, under the supervision of prof. Nicolas Gillis.

Some results of this research work have been published in the following papers:

- Casalino G., Del Buono N., Mencar C., Subtractive clustering for seeding non-negative matrix factorizations, Information Sciences, Volume 257, 1 February 2014, Pages 369-387, ISSN 0020-0255, http://dx.doi.org/10.1016/j.ins.2013.05.038;

- Casalino G., Del Buono N., Mencar C., (2014) Part-Based Data Analysis with Masked Non-negative Matrix Factorization, 440–454. In Computational Science and Its Applications – ICCSA 2014.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The amount of available data has grown dramatically over the past fifty years. Every year more than 200 Exabytes of data are generated. Huge quantities of digital data are produced daily from different sources: numerical data from satellites or sensors, textual data — both structured and unstructured — from web sites, emails, forums, newsgroups, public and private digital archives, images, videos, are just some examples. Data overload is a fact of life for all of us in the information era.

Although this profusion of information potentially allows to satisfy all information needs, it also presents some limits: the larger is the amount of data the fewer are the possibilities to capture, discover and understand useful knowledge to guide action or decision making [Bierig et al.; Liu and Motoda, 2007].

Clearly human capabilities prove to be unsuitable to process big amounts of data, therefore automatic mechanisms, which are able to assist humans in extracting useful information and knowledge from rapidly growing volumes of digital data are indispensable and an extensive effort of research in this direction has been made in the last years.

Intelligent Data Analysis (IDA) aims to the intelligent application of human expertise and computational models for advanced data analysis. Automatic tools which strive for involving the analyst in the process of data analysis and extracting useful patterns from big data can be enumerated among IDA methods. In this scenario, techniques coming from different areas (such as statistics, artificial intelligence, data mining, machine learning, optimization, dynamic programming)

which favour the interaction with users and produce understandable knowledge could be favoreably exploited in IDA.

Non-negative Matrix Factorizations are powerful techniques recently proposed to uncover latent low-dimensional structures intrinsic in high-dimensional data and provide a non-negative part-based representation of data [Berry et al., 2007; Cichocki et al., 2009; Gillis, 2014; Lee and Seung, 1999, 2001; Zhang, 2011]. Non-negativity enhances meaningful interpretations of mined information and distinguishes NMF from other traditional dimensionality reduction algorithms, such as Principal Component Analysis (PCA) [Jolliffe, 1986] or Singular Value Decomposition (SVD) [Golub and Van Loan, 2001].

However, the understandability of the results obtained by applying classical NMF is not guaranteed a priori as they often do not correspond with the intuitive notions of parts in the original data. Several variants of constraints and various regularization terms have been proposed to improve NMF capabilities so as to make the extracted parts easier to understand by the data analyst.

This thesis aims to propose three NMF variants by enhancing understandability so as to improve the applicability of NMF in IDA.

First, an initialization technique for NMF based on the Subtractive fuzzy clustering algorithm (SC) is proposed. It has been applied to document clustering application showing its ability to enhance the quality of the clustering results in term of interpretability of the cluster centers.

Second, an approach for injecting user knowledge in the factorization process, by masking the basis matrix (one of the products of NMF) is presented. Masking enables the decomposition of data into user-defined parts, which are consequently easy to understand by the analyst. The results of Masked NMF helps the analyst to identify which subset of the available data are best represented by the specified parts, thus extracting potentially useful knowledge from large quantities of data.

Finally a constrained modification of Non-Negative Matrix Underapproximation (NMU) algorithm for hyperspectral images is described[1]. Non-negative matrix factorizations are widely used in hyperspectral imaging due to their capability of separating constituent materials of the objects represented in the images, and

[1]This research result has been achived in collaboration with professor Nicolas Gillis, after a five months studentship at the University of Mons (Belgium).

to correctly classify the pixels according to these materials. Moreover it has been shown that sparsity and local information about pixels in images, when incorporated in the factorization process, lead to better results in term of separation of materials and classification of pixels [Gillis and Plemmons [2013]; Gillis et al. [2012]]. A new NMU algorithm incorporating prior information *PNMU* (both sparsity and local information) has been developed. Tests on real datasets have shown that the proposed method outperforms the standard NMU algorithm and its constrained versions.

This thesis is organized as follows: in paragraph 2.1 an overview of IDA and its objective is given, while paragraph 2.2 focuses on NMF techniques. Than the three proposed approaches are reported in chapters 4, 5, and 6 respectively. For each method, experimental results demostrating the effectiveness are shown. Future perspectives are sketched in the conclusing chapter 7.

Chapter 2

Preliminaries and Problem Statement

2.1 Intelligent Data Analysis

Data are collections of values or measurements. They can be numbers, words, observations or even descriptions of things. In this chapter we will simply refer to data as a collection of numerical values recording the magnitude of different attributes and/or features that describe the problem under study.

Hand writes *"data analysis is what we do when we turn data into information"* [Hand, 1997]. Intelligent data analysis is the intelligent way to do it. Moreover, he gives a definition of information: *"It is what we extract from data when we attempt to answer some questions. Before extracting information which can shed light on a question, one must be clear about what that question is"*. This is a crucial point in IDA: the analysis is driven by the questions that the analyst wants to answer to, otherwise it would be *unintelligent*.

IDA is an iterative process that enables the combination of human expertise and computational models to automatically extract useful patterns, event correlations and in general, understandable knowledge which would otherwise remain hidden in the data under consideration [Berthold and Hand, 1999]. A data analyst could be interested in describing data by finding patterns and anomalies, or just by summarizing them; in this case the term *exploratory data analysis* is

used. Contrariwise, when the analyst is interested in verifying some hypoteses about the structure in data, e.g. differences among groups of data, evolution of the attribute values, etc., the term *confirmatory data analysis* is used. IDA is a multidisciplinary discipline that comes from the intersection of several research fields, the most important ones are Statistics and Machine Learning.

IDA and Knowledge Discovery from Data (KDD) are tightly correlated, yet with some noteworthy differences. Both are aimed at identifying valid, novel, potentially useful, and ultimately understandable patterns in data [Fayyad et al., 1996]; however IDA emphasizes the importance of the prior knowledge possessed by human experts that intelligently guide the analysis process in an interactive and iterative way [Berthold et al., 2010]. Data Mining is one step of the KDD process and refers to the set of tools that allow to *automatically* extract knowledge from large amounts of data [Fayyad et al., 1996]. However, a full automatization of the data analysis process is impossible [Berthold et al., 2010], for this reason IDA is focused on the human contribution to the analysis process.

Holmes and Peek [2007] categorize IDA methods in three main classes: data exploration, classification and prediction, dimensionality reduction . Data exploration plays a fundamental role in data analysis. Analysts look at data for discovering relations among features, trends, anomalies or outliers, relations among features and classes, etc. Most of these techniques rely on visual tools to represent information. IDA-based approaches for data exploration integrate automatic techniques with a-priori user knowledge in the exploration process, thus enabling user interaction. Classification and prediction methods are used in several domains dealing with real data. Machine learning literature provides many different techniques for classification (both supervised, semi-supervised or unsupervised) and prediction. Most of them are based on some automatic learning tools to acquire knowledge that can be used for classifying (or predicting) unobserved data. However, only few of them are capable of yielding knowledge that is intelligible to users (e.g. knowledge expressed in form of rules), a mandatory requirement for their use within IDA. Learning interpretable knowledge from data is a topic of current research in Machine Learning and Computational Intelligence. In this context are located dimensionality reduction techniques, that represent data in a reduced space through feature selection and extraction. This facilitates to man-

age, understand and visualize data. Because of their tight relationship with NMF, a brief overview of such techniques is outlined in the following subsection.

2.1.1 Dimensionality Reduction techniques

Often, in high-dimensional data not all the measured variables are "important" for understanding the underlying phenomena of interest. Hence, mechanisms that transform data and reduce the number of original variables are frequently used.

Let $X \in \mathbb{R}^{n \times m}$ be the observation data matrix, where each columns vector is composed by n observations for each of the m dimensional variable in $x = (\mathbf{x_1}, \ldots, \mathbf{x_m})^\top$. In this formalization, the dimension of data is meant the number of variables that are measured on each observation, while the term dimensionality of X indicates the number m of original features. A dimensionality reduction method is a transformation of a given data matrix X into a meaningful representation $S \in \mathbb{R}^{n \times k}$ of reduced dimensionality $k \leq m$ [Van der Maaten et al., 2008]. The low dimensional vectors $s = (\mathbf{s_1}, \ldots, \mathbf{s_k})^\top$, with $k \leq m$ capture information in the original data, according to some particular criteria. The components of s are called "hidden components" or "latent factors", while — depending on the particular research context one is working with— the m multivariate vectors are alternatively named "variables", "attributes" or "features". Dimensionality reduction methods mitigate *the curse of dimensionality* [Bellman, 1961], which refers to difficulties related to data analysis when data dimensionality increases; these methods are able to overcome problems coming from data sparseness and noise, and can be adopted as a visualization tool to show multivariate data in a human intelligible form.

Dimensionality reduction techniques can be categorized in two classes: (i) *feature selection* and (ii) *feature extraction*. A feature selection method is a process that selects a subset of k original (and supposed relevant) features for spanning a reduced space that may better describe the phenomena of interest. Feature selection mechanisms reduce the computational costs, but a good trade-off between accuracy of the results and efficiency is needed.

On the oher hand, feature extraction methods try to capture hidden properties of data and discover the minimum number of uncorrelated or lowly correlated

factors that can be used to better describe the phenomena of interest. It is accomplished by the creation of new features obtained as functions of the original data. Reduction of the computational complexity of data both in time (for elaboration) and in space (for storage) and the discovery of latent structure hidden in data, (meaningful structures and/or unexpected relationships among variables) are some of the advantages resulting from feature extraction methods.

The simplest dimensionality reduction methods are linear and derive each of the $k \leq m$ components of the new variables in S as a linear combination of the original variables:

$$S = XA, \tag{2.1}$$

or equivalently

$$X = SB, \tag{2.2}$$

being $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{k \times m}$ appropriate linear transformation weight matrices. Equation (2.2) makes clear the motivation why the new variables in S are called hidden or latent factors. Principal Component Analyisis (PCA) [Hotelling, 1933; Jolliffe, 1986; Pearson, 1901], Factor Analysis (FA) [Spearman, 1904], Indipendent Component Analysis (ICA) [Hyvärinen, 1999], Linear Discriminant Analysis (LDA) [Fisher, 1936], CUR decomposition [Mahoney and Drineas, 2009] are all well known linear dimensionality reduction techniques used for analyzing multivariate data. Among linear dimensionality reduction methods, the most widely used in the context of IDA is PCA.

2.1.1.1 Principal Component Analysis

Principal component analysis is the best, in the least-square error sense, linear dimensionality reduction technique [Jackson, 2003; Jolliffe, 1986]. It is based on the covariance matrix of the variables and seeks to reduce the dimensionality of data matrix X by finding few orthogonal linear combinations (the principal components – PCs) of the original variables with the largest variance. The first PC is the linear combination of the original data with the largest variance; the second PC is the linear combination with the second largest variance and orthogonal to

the first PC, and so on. The principal components are given by:

$$Y = XU, \tag{2.3}$$

where $U \in \mathbb{R}^{m \times m}$ is an orthogonal weight matrix computed as the orthogonal factor of the spectral decomposition of the covariance matrix $X^\top X$ of the stardardized data matrix X[1]. Therefore, the columns of the matrix U are the eigenvectors of the covariance matrix. These eigenvectors (principal axes) map a data vector from the original space of m variables to a new space of k variables which are uncorrelated over the dataset. Hence keeping only the first $k < m$ principal components a dimensional reduction on k-dimensional subspace of the original data is derived.

Moreover, it is proven that the transformed data matrix, obtained by only considering the first $k < m$ principal components, is the best least-squares k-approximation of the original data X (this result is known as the *Eckart–Young–Mirsky theorem* [Golub et al., 1987]).

Figure 2.1 shows the behaviour of PCA of a data matrix collecting points that belong to a bivariate Gaussian distribution centered in the coordinates $(1, 3)$. Standard deviation of data is 3 in the direction $(0.878, 0.478)$ and 1 in the orthogonal direction. The first principal component ($PC1$) captures information in the direction of the maximum variability in data, instead the second principal component ($PC2$) is orthogonal to the first one and captures information in the second most variable direction. The principal axes are therefore the bases of the rotated space and are centered in the center of the points. This is a simple example where the dimensionality of the original data space and that of the transformed one are the same. As an example of dimensionality reduction, one can represent the same points using the first principal axis only: in this case a one dimensional space is obtained where data points are projected onto.

In many applications, the most of data variance can be captured by the first two (or three) PCs: this makes the PCA a widely used visualization tool in IDA. However, even though the PCs are uncorrelated variables constructed as linear

[1]Since the values of the variance of data depends on the scale of the variables, usually the original data contained in X are subject to a standardization process so that each variable has mean zero and standard deviation one.

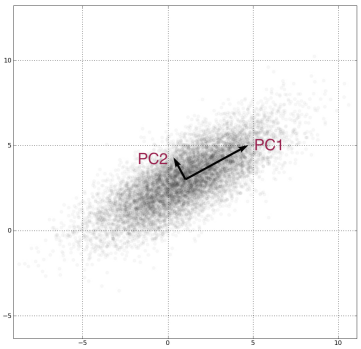


Figure 2.1: Graphical illustration of PCA. From Wikipedia, the free encyclopedia.

combinations (with mixed signs) of the original variables, and have some desirable properties (they are orthogonal and ordered in a descreasing manner w.r.t. the variance of original data), they do not necessarily correspond to meaningful physical quantities. Hence, a clear interpretation of the results provided by PCA is sometimes difficult to be derived.

To clarify this point consider the computer vision problem of human face recognition, where PCA has been largely adopted to obtain a set of basis images — *the eigenfaces* — that can be linearly combined to reconstruct images in the original dataset of face [Turk and Pentland, 1991]. As it can be observed by Figure 2.3 (left panel), eigenfaces are not physically intuitive and far to correspond to what humans use to explain why a face is a face. In particular, because of the presence of negative signs in the components of principal axes, PCA reconstructs the original data adding up some basis images and subtracting others: this may not make sense in some applications. A simply question can be posed:"What does it mean to subtract a face basis?"

These considerations can be extended to documents, genes, preferences, questionaries and to all non-negative data. In the following section 2.2, a review of Non-negative Matrix Factorization is given. It is able to represent original data by only additive, not subtractive, combinations of some basis vectors. This characteristic of parts-based representation is appealing because it reflects the intuitive notion of combining parts to form a whole providing more distinct and clearer

dimensionality reduction results and a easier understandability of the obtained results.

2.2 Non-Negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a computational technique for linear dimensionality reduction of a given data matrix X, which is able to explain data in terms of additive combination of non-negative factors that represent realistic *building blocks* for the original data (provided that data are non-negative too) [Cichocki et al., 2009; Gillis, 2014; Lee and Seung, 1999, 2001; Zhang, 2011].

The non-negativity constraint is useful for learning part-based representations and has a twofold motivation. First in many applications one knows that the quantities involved cannot be negative (for example by the rules of physics). Second intuitively parts are generally combined additively (and not subtracted) to form a whole and physiological principles assume that humans learn objects as part-based [Lee and Seung, 1999]. Hence, non-negativity potentially enhances meaningful interpretations of information mined from a given data matrix, allowing to a better understanding of the results obtained by the analysis process: this makes NMF a suitable computational models for IDA.

2.2.1 NMF mathematical formulation

Formally, given a nonnegative data matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \in \mathbb{R}_+^{n \times m}$, where $\mathbf{x}_i \in \mathbb{R}_+^n$ are $n-$dimensional column vectors representing samples[1], NMF aims to approximate X into the product of two lower rank non-negative matrices — a *basis matrix* $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k] \in \mathbb{R}_+^{n \times k}$ and an *encoding matrix* $H = (h_{ij}) \in \mathbb{R}_+^{k \times m}$ — such that

$$X \approx WH, \tag{2.4}$$

or, equivalently,

$$\mathbf{x}_j \approx \sum_{i=1}^{k} \mathbf{w}_i h_{ij}. \tag{2.5}$$

[1]Henceforth a matrix is denoted with an uppercase letter, e.g. X, its elements with the corresponding lowercase letter, e.g. x_{ij}, a column vector in lowercase boldface, e.g. $\mathbf{x}_i$

where W and H both have non-negative elements (namely, $W \geq 0$ and $H \geq 0$) and the product matrix (WH) is of rank k with $(n+m)k \leq nm$.

2.2.2 NMF optimization problem

To compute a non-negative matrix factorization (2.4) of a given data matrix X, some quality measures have to be taken into account to evaluate how well the product (WH) approximates the data matrix X. Particularly, some divergence function

$$D : \mathbb{R}_+^{n \times r} \times \mathbb{R}_+^{r \times m} \to \mathbb{R}_+,$$

can be adopted if it possesses the following properties: (i) it is continuously differentiable (at least once) in both variables; (ii) it is individually convex in W and H; and (iii) it equals 0, if and only if $X = WH$ [Dhillon and Sra, 2005].

It should be observed that the divergence D is a function of the factor matrices W and H, but it is also parametrized by the input data matrix X. This dependence can be expressed by writing $D(X; W, H)$ [Tandon and Sra, 2010]. Using the previous formalization, the NMF problem may be re-written as a non-linear constrained optimization problem over the divergence D, that is:

$$\min_{W \geq 0, H \geq 0} D(X; W, H). \tag{2.6}$$

The most frequently adopted instance of (2.6) leads to the minimization of

$$\min_{W \geq 0, H \geq 0} D(X; W, H) = \|X - WH\|_F^2, \tag{2.7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Many other divergence measures have also been used, the interested reader can refer to [Dhillon and Sra, 2005].

The most popular approach to numerically solve the NMF optimization problem is the multiplicative update algorithm proposed by [Lee and Seung, 2001]. Particularly, it can be shown that the square Frobenius distance measure (2.7) is non-increasing under the Lee and Seung's iterative update rules described in Algorithm 1.

Lee and Seung update rules can be interpreted as a diagonally rescaled gradi-

Algorithm 1 Lee and Seung multiplicative update rules (NMF)

Initialize non-negative matrices $W^{(0)}$ and $H^{(0)}$
while Stopping criteria are not satisfied **do**
$W \leftarrow W \odot (VH^\top) \oslash (WHH^\top)$
$H \leftarrow H \odot (W^\top V) \oslash (W^\top WH)$
end while
{$\odot$ and $\oslash$ denotes the Hadamard product, that is the element-wise matrix multiplication and the element-wise division, respectively.}

ent descent method (i.e., a gradient descent method using a rather large learning rate).

Additional constraints on the factors W and/or H, such as sparsity or orthogonality, may be (approximately) enforced by means of the introduction of penalty functions or constraint modification of the minimization problem (2.6). Some examples are the non-negative sparse encoding factorization (NMFSC) proposed in [Hoyer, 2002], which has the peculiarity of controlling the statistical sparsity of the factor matrices W and H, in order to discover part-based representations that are more separate than those given by standard NMF. Furthermore, orthogonal non-negative matrix factorizations (ONMF) attempt to obtain the basis and/or the encoding matrix with columns as orthogonal as possible, in order to minimize the number of basis components required to represent the data and the redundancy between different basis [Choi, 2008b]. Constrained variants of NMF will be detailed in paragraph 3.2.

2.2.3 NMF drawbacks

Uniqueness

NMF are not free from drawbacks. First, it should be pointed out that the basis and the encoding factors in (2.4) are not unique. For instance, when an arbitrary invertible matrix $A \in \mathbb{R}^{k \times k}$ can be found, such that the two matrices $W' = WA$ and $H' = A^{-1}H$ are non-negative, then another factorization $X \approx W'H'$ exists. Such a transformation is always possible if A is an invertible non-negative matrix. However, in this situation, the result of the transformation is simply a scaling and permutation of the original matrices [Berman and Plemmons, 1979]. The

NMF problem is a difficult non-convex optimization problem. Indeed, it has been proved in [Vavasis, 2007] that finding the global optimum for the minimization problem in (2.7) is a NP-hard problem, and so a numerical method for solving the NMF problem could only guarantee at most locally optimal solutions.

Factorization rank

On the other hand, to obtain NMF for a given data matrix X, the number k of rank factor must be provided. This number k is problem dependent and identifies the number of factors to be used in order to explain data and it plays a fundamental role in the factorization process. In fact, different values of k lead to different factorization results. This is an open issue when dealing with NMF as well as with methods devoted to reduce the dimensionality of multivariate data (for instance, SVD needs to fix the number of singular components or in the k-means clustering the user has to choose the number of desired clusters [Maisog, 2009]). This problem can be solved as done in PCA or ICA, where the number of components to be considered the true signals and those to be considered merely noice is decided a-posteriori using classic methods such as Cattell's scree test [Cattell, 1966] or Kaiser's rule [Kaiser, 1960].

Initialization

All algorithms for computing NMF are iterative and require initialization of the basis and the encoding matrices [Berry et al., 2007; Wild et al., 2004; Wild, 2003]. As previously observed, the computed non-negative factorization is not unique: one can obtain different answers depending on the initialization of matrices W and H. Hence different initializations may find different local minima in the search space. Moreover, the efficiency of many NMF algorithms is affected by the selection of the starting matrices: poor initialization often results in slow convergence or lower error reduction. Furthermore, the problem of selecting an appropriate initialization becomes more complex when additional structures or constraints are imposed on the factorized matrices, or when the data possess special meaning. Different initialization mechanisms have been proposed in literature: some of them lead to rapid error reduction and faster convergence of the adopted NMF

algorithm, others lead to a good overall error accuracy at convergence [Boutsidis and Gallopoulos, 2008; Buciu et al., 2006]. However, there does not exist a definitive suggestion about the best initialization strategy to be adopted for different NMF algorithms [Del Buono and Lucarelli, 2010]. A taxonomy of the initialization techniques for NMF is given in paragraph 3.1.

2.2.4 Interpretation of the basis and encoding matrices

The results of NMF applied to a data matrix X have an immediate geometrical interpretation. According to (2.5), the columns of the matrix W are basis vectors spanning a subspace in $k \leq n$ dimensions, called NMF-subspace, while each column of the encoding matrix H represents the new coordinates of the corresponding data sample in the NMF-subspace. From a numerical point of view, each data sample is approximated by a linear combination of vectors in W, where the linear coefficients are grouped in a column of H corresponding to the data sample. Therefore, the elements h_{ij} codify the amount of the factors (i.e. the columns of W) used to reconstruct each sample of X in the NMF-subspace.

The coefficients h_{ij} in each column of H define the importance of each basis vector in approximating the data sample: if a coefficient is very small, then the corresponding basis vector is useless in approximating the sample. Under some hypoteses[1], the basis vectors can be interpreted as prototypes of data clusters. In this case, the coefficients h_{ij} can be easily interpreted as membership degrees of each sample to each cluster.

Examples of successful applications of NMF are: basic student skills describing student questionnaire results in educational data mining [Desmarais, 2011]; topics represented as bag of words in text mining [Shahnaz et al., 2006]; anatomic parts of images describing human faces in face identification problems [Guillamet and Vitrià, 2002; Sun et al., 2009]; part-based representation of digital characters for object recognition [Guillamet and Vitrià, 2003; Liu and Zheng, 2004]; community categories extracted to describe users networks [Wang et al., 2011]; diversified portfolio describing trends in stock markets in financial data mining [Drakakis et al., 2008; Ribeiro et al., 2009]; topics used to clusterize social tags

[1]The use of NMF in clustering applications will be detailed in paragraph 3.2.2.

data [Chen et al., 2014]; users-items relations in recommender systems [Gu et al., 2010]; chemical constituents in air pollution relevations [Hopke, 1985; Kim et al., 2003]; musical instrument frequencies for music classification [Benetos et al., 2005, 2006a,b]; endmembers of consituent materials of hyperspectral images [Burger and Geladi, 2007; Gillis and Plemmons, 2013; Gillis et al., 2013; Jia and Qian, 2007; Ma et al., 2014; Pauca et al., 2006]; genes in microarray data [Brunet et al., 2004; Carmona-Saez et al., 2006; Devarajan, 2008; Ding et al., 2006; Mejía-Roa et al., 2008; **?**]. Other successful applications of NMF, where interpretability is a key requirement, belong to molecular pattern discovery [Gao and Church, 2005; Kim and Park, 2007] and object detection [Casalino et al., 2012].

A key aspect of NMF, that is advantageous for its application in IDA, stands in the possibility of approximating data samples as linear combination of factors, where the factors are subsets of the same features used to represent data samples. Therefore, unlike other low-rank approximation techniques, NMF allows to represent data as composition of parts, being each part expressed with the same features used in data. This makes the results of NMF easily interpretable for the analyst, who can intelligently guide the factorization process, in order to achieve results that are interesting and useful for understanding the problem at hand.

2.2.5 Comparison of NMF and PCA

As stated before, PCA can be used as tool in IDA because of its dimensionality reduction and visualization capability. However, it presents some drawbacks (such as the presence of mixed sign values) and several research papers demonstrated that it is outperformed by NMF in many applications such as face recognition [Cichocki et al., 2009; Guillamet and Vitrià, 2003]. In the following, some of the differences among these two techniques are briefly highlighted [Zinovyev et al., 2013].

Uniqueness. PCA is able to find the global minimum of the optimization problem, while NMF usually is trapped into local minima: this implies that the set of principal components is unique, while NMF has multiple solutions (in terms of basis and encoding matrices).

To overcome NMF non-uniqueness problem, bootstrapping techniques can be

used, several executions of the factorization are performed and the most frequent solutions selected.

Ranking. Principal components are naturally ranked accordingly to the quantity of variance they explain. On the contrary, factors in NMF have no ordering and are all equally important. This causes a problem to appropriate choose the value of the rank parameter k. When PCA is applied, no specification of the value k is provided: all the eigenpairs are computed and then the most important components are selected according to the proportion of variance one wants to preserve. Instead, when NMF is applied, the parameter k has to be specified (by user) as input parameter for the factorization. The choice of the rank value is problem dependent: usually, different factorizations are performed with different rank values and then the results are evaluated accordingly to the target of the analysis.

Orthogonality. Principal components are orthogonal directions which capture the variance in data. On the other hands, factors obtained by NMF are positive vectors that better approximate data, but they are not necessarily orthogonal. They are the bases of the hypercone containing all data and are able to preserve local data structure in this subspace. Figure 2.2 shows the principal components and the factors returned by PCA and NMF (left and right panels, respectively) when applied to non-negative 2-dimensional data matrix.

The orthogonality constraint is a desiderable property, however this implies the presence of some negative values in the elements of principal components that, as previously highlighted, does not make sense in some contexts. The non-negativity constraint is always violated by PCA, even when it is applied to non-negative data. Hence, the interpretability of data is lost when moving from original data space to the reduced low dimensionality subspace. From Figure 2.2 (left panel) it can be observed that, starting from samples in the positive orthant, after transforming them by PCA, samples belonging to the line assume negative values. On the contrary (right panel), NMF preserves the non-negativity of data that leads to a part-based representation.

The interpretability of the factors is one of the strength point in NMF. The parts-based representation obtained by NMF is more intuitive and human-understandable than the holistic results of PCA. A clear example is illustrated

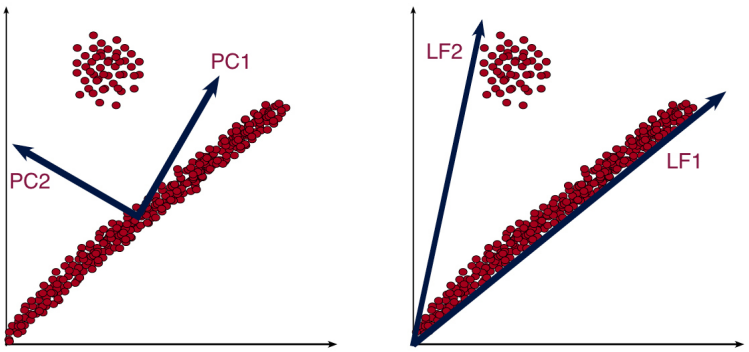


Figure 2.2: Comparison between principal factors (left panel) and NMF latent factors (right panel).

in Figure 2.3 in the context of facial image recognition problem [Lihong et al., 2008]. PCA provides for the eigenfaces that are prototypical faces containing all kinds of facial traits (left panel), while NMF basis vectors represent particular facial traits: different kinds of eyes, noses, mouths (right panel).

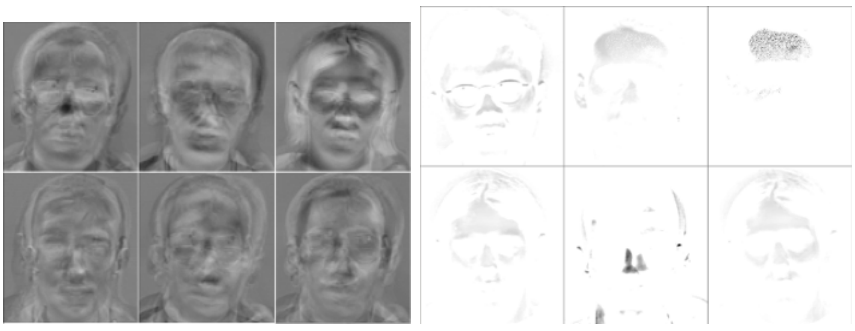


Figure 2.3: Comparison between bases extracted with PCA (left panel) and NMF (right panel).

Chapter 3

Related Work

In this chapter the literature of NMF techniques for injecting prior knowledge in the factorization process, will be shown. First, a taxonomy of the initilization mechanisms for NMF will be shown in section 3.1. Than the constrained versions of NMF will be detailed in section 3.2 to show their influence to intelligently analyze data

3.1 Initialization mechanisms for NMF

All algorithms for NMF are iterative. They require the computation of initial matrices $W^{(0)}$ and/or $H^{(0)}$ by some numerical mechanism and then alternately update W and H until the divergence measure does not present appreciable changes (i.e., $D(X; W, H)$ is bounded by a fixed tolerance). This mechanism unavoidably yields to converge to locally optimal solutions. Hence, in practice, the local minima from several different starting points should be compared, using the results of the best local minimum found. However, this is prohibitive on large, realistically-sized problems.

Not only different NMF algorithms produce different NMF factors, but also the same NMF algorithm, run with slightly different initial matrices, could produce different NMF factors. Therefore, the choice of the initial pair $(W^{(0)}, H^{(0)})$ plays a crucial role for the convergence speed of the iterative algorithm and for the improvement of the algorithm performance. Moreover, when NMF is ap-

plied in a particular context, such as text document clustering, it can be of some importance that initial factors also possess meaningful interpretations.

Different initialization mechanisms have been proposed to improve the performance of the NMF algorithms: some of them lead to rapid error reduction and faster convergence of the standard NMF algorithm, others lead to better overall error accuracy at convergence. A wide range of approaches have been presented in literature to alleviate the initialization problem: random initialization, prototype-based clustering methods (such as k-means and Fuzzy C-means) [Wild et al., 2004; Wild, 2003; Xue et al., 2008], feature extraction techniques (such as, SVD, PCA and ICA) [Boutsidis and Gallopoulos, 2008; Buciu et al., 2006; Casalino et al., 2011], meta-heuristic search algorithms (such as genetic algorithms) [Janecek and Tan, 2011; Rezaei et al., 2011].

As evidenced by some authors [Albright et al., 2006; Boutsidis and Gallopoulos, 2008], the goodness of an initialization strategy for NMF can be assessed either when (i) rapid error reduction and fast convergence occur or (ii) a better overall error at convergence is obtained. Some experimental evidences have been obtained when the first criteria are considered, however, the quality of the pair (W, H) in terms of overall error is quite difficult to understand especially when other features such as interpretability or added structure reveal to be important. Hence, initialization of NMF remains an open problem and to date there does not exist a definitive suggestion about the best initialization strategy to be adopted for any NMF algorithm or a standard procedure to justify the choice of a particular NMF initialization scheme.

Initialization schemes can be classified in:

- **simple mechanisms**, based on some kind of randomization;

- **complex schemes**, based on clustering algorithms or on some alternative low rank factorization;

- **evolutionary schemes**, based on techniques mimicking optimization strategies observed in nature.

Figure 3.1 sketches a purposely conceived categorization of the initialization algorithms appeared in the survey of literature on NMF.

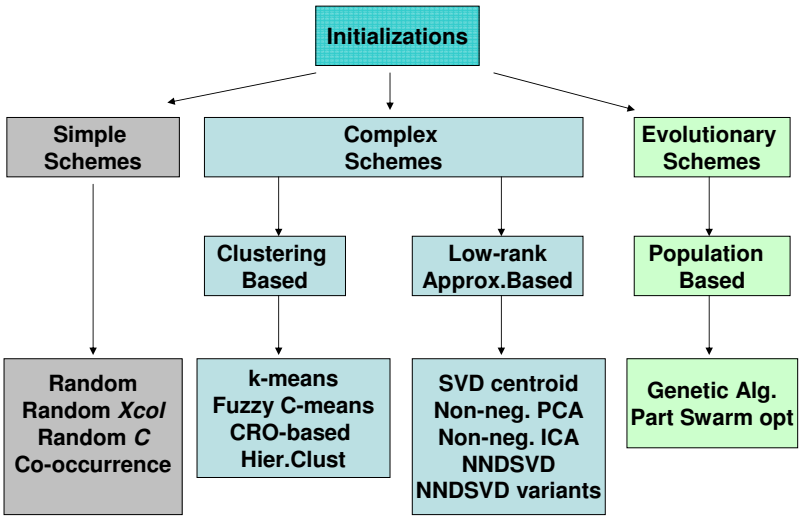


Figure 3.1: Classification of the initialization schemes which appeared in the literature panorama.

The class of simple initialization mechanisms includes: (i) the random initialization, (ii) several variants of random choices of columns in X used to build $W^{(0)}$ together with random or zeros initialization of $H^{(0)}$. Random initialization produces dense matrices $W^{(0)}$ and $H^{(0)}$ of dimension $n \times r$ and $r \times m$, respectively, with elements randomly generated in a specified subinterval of positive numbers (generally, [0,1]). This simple initialization mechanism has the advantages of requiring low computational cost and negligible processing time, nevertheless it does not generally provide a good first estimate for NMF algorithms [Smilde et al., 2004], especially when alternating least squares algorithms are adopted [Albright et al., 2006].

Random variants include:

(i) random Xcol initialization. This scheme computes each column of the basis matrix $W^{(0)}$ by averaging p columns of X randomly chosen and generates

$H^{(0)}$ (when required) via the random initialization. Random $Xcol$ initialization builts basis vectors from the given data matrix; hence, as observed in [Albright et al., 2006], when X is sparse, this initialization scheme forms a sparse initial basis matrix $W^{(0)}$, which represents a more reasonable choice than using random initialization. However, performance of NMF algorithms initialized by random $Xcol$ scheme are comparable by those obtained using random initialization [Langville, 2005].

(ii) random-C initialization. This scheme computes each column of the basis matrix $W^{(0)}$ by averaging a random subset of vectors chosen from the longest (in the 2-norm) columns of the data matrix X and generates $H^{(0)}$ (when required) via the random initialization. In the case the data matrix X is very sparse, on the average this method chooses the densest columns for initialization. The main idea underlying this variant of random $Xcol$ initialization is that the choice of the densest columns might be more likely to be the centroid centers [Albright et al., 2006].

(iii) co-occurrence initialization. This scheme firstly forms the co-occurrence matrix $XX^\top$ and then randomly chooses the k columns of the initial factor $W^{(0)}$ among the densest columns of the co-occurrence matrix and generates $H^{(0)}$ (when required) via the random initialization. The co-occurrence scheme has the advantage of producing a basis matrix which includes some hidden information on the initial data (i.e., term-term similarities when a document clustering scenario is considered), however, it requires higher computational cost than simple random initialization.

Complex initialization strategies exploit clustering algorithms or some alternative low rank factorization scheme to construct the initial pair $(W^{(0)}, H^{(0)})$ in order to provide structured initialization pairs. Schemes among this class can be classified into:

(i) clustering based methods. These mechanisms provide initializations based on a data-density sample of initial data locations. Schemes in this class construct initial low rank factors $W^{(0)}$ and $H^{(0)}$ adopting some clustering method.

Clustering based initializatiosn appeared in literature are: spherical k-means (k-means) initialization (alzso known as centroid initialization) [Xue et al., 2008] and Fuzzy C-Means (FCM) initialization [Rezaei et al., 2011; Zheng et al., 2007]. Specifically, the former is used to partition the columns of the data matrix X into k clusters and select the centroid representative vector for each cluster (these are known as cluster centroid vectors or concept vectors). Then these vectors are used to initialize the columns of $W^{(0)}$. The elements on the encoding matrix $H^{(0)}$, instead, are initialized with the distances from each data point to every centroid. Moreover, $H^{(0)}$ could be the binary partition matrix, or can be computed either random or as $\arg\min_{H\geq 0} \|X - W^{(0)}H\|_F$. The FCM scheme, instead, initializes the matrix $W^{(0)}$ with the cluster centers and $H^{(0)}$ with the fuzzy partition matrix (more details can be found in [Zheng et al., 2007]). It should be pointed out that both spherical k-means and FCM initializations need to be themselves initialized.

(ii) low-rank approximation based methods. These mechanisms provide initialization based on the eigenstructure of the initial data. Among low-rank approximation based methods, we can include:

- **SVD-centroid initialization**, which initializes $W^{(0)}$ with the centroid decomposition of the SVD factor V[1] of the data matrix X, when this factor is available [Langville, 2005];
- **PCA and ICA initialization**, which initialize the pair $(W^{(0)}, H^{(0)})$ with the non-negative principal and independent components, respectively, extracted from the initial data matrix X (the non-negativity of the components obtained from PCA and ICA is met by truncation of the negative values) [Zhao et al., 2008];

[1]The singular value decomposition (SVD) of a data matrix $X \in \mathbb{R}^{n\times m}$ is $X = U\Sigma V^{\mathrm{T}}$, where $U \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, $\Sigma \in \mathbb{R}^{n\times m}$ is a diagonal matrix, and $V \in \mathbb{R}^{m\times m}$ is the transpose of an orthogonal matrix. The columns of U are orthonormal eigenvectors of XX^{T}, the columns of V are orthonormal eigenvectors of $X^{\mathrm{T}}X$. They are called: the left-singular vectors and right-singular vectors of X respectively. The diagonal entries Σ_{ii} are known as the singular values of X, and they are the square roots of eigenvalues from U or V in descending order.

- **Lanczos bidiagonalization process** to get a low-rank approximation of a non-negative matrix. This mechanism has been recently proposed as initialization strategy for all existing ONMF algorithms and contains a little random because of free choice of the vector used to start the bidiagonalization process see [Wang et al., 2012];

- **Non-negative Double Singular Value Decomposition (NNDSVD)**. This is the only initialization mechanism containing no randomization and is based on two SVD processes, the one approximating the data matrix, the other approximating positive section of the resulting partial SVD factors via peculiar properties of unit rank matrices. Specifically, first the matrix X is decomposed into its rank k SVD, $X = \sum_{i=1}^{k} \sigma_i C_i$, where $C_i = u_i v_i^\top$. Then each C_i is decomposed into positive and negative components $C_i = C_i^+ - C_i^-$, being C_i^+ the nearest (in sense of Frobenius norm) positive approximation of rank 2 to C_i. Then the SVD of C_i^+ is computed to obtain its dominant singular triplet. After these two SVD processes, the first column and row vectors in $W^{(0)}$ and $H^{(0)}$ are initialized using dominant singular triplet of X, while subsequent column and row vectors in $W^{(0)}$ and $H^{(0)}$ are equal respectively to the singular triplets of C_i^+.

  NNDSVD initialization proved to increase the performance of NMF algorithm on several datasets [Anbumalar et al., 2013; Boutsidis and Gallopoulos, 2008; Heinrich et al., 2008; Xiaojun, 2011] and its great advantage lies in the uniqueness of the computed initial factors (we address the reader to [Boutsidis and Gallopoulos, 2008] for detailed description of NNDSVD initialization and properties).

Generally speaking, complex initialization strategies require higher computational costs, but they could produce fast error reductions, high convergence rates in NMF algorithms and reduce to the minimum or definitely do not require the use of any randomization step.

The last class of initialization schemes is the class of evolutionary initializations which generally operate on a population of solutions in the search space with techniques typical of evolutionary processes. Evolutionary initialization schemas

Initializ.	Pros	Cons	Costs
Random	easy; cheap to compute;	dense $(W^{(0)}, H^{(0)})$ with no intuitive meaning; k assigned a priori	$O((n+m)k)$
k-means	reduces NMF iterations; intuitive meaning of $W^{(0)}$	dense $W^{(0)}$; expensive; k assigned a priori	$O(nmk)$
FC-means	intuitive meaning of $(W^{(0)}, H^{(0)})$	dense $(W^{(0)}, H^{(0)})$ expensive; k assigned a priori	$O(nmk)$
NNDSVD	no randomization; structured $(W^{(0)}, H^{(0)})$	expensive r assigned a priori;	$O(nmk)$

Table 3.1: Pros, cons and the computational complexity of the initialization methods for NMF which have been adopted in the experimental session.

include genetic algorithms, particle swarm optimizations, differential evolution, etc. [Janecek and Tan, 2011; Snasel et al., 2008; Stadlthanner et al., 2006]. As complex initialization schemas, evolutionary initializations present high computational costs, but demonstrate the advantage of parallel implementation. To conclude this brief reviews of initialization mechanisms appeared in the literature panorama, in Table 3.1, the main advantages and the principal drawbacks of the most adopted initialization schemas have been summarize, together with the indication of the computational costs required to perform one step of these initializations [Albright et al., 2006; Wild et al., 2004]. It should be pointed out that both complex and evolutionary schemas need some iterations to meet a reasonable stopping criteria or the reach an appreciable value of fitness functions. Particularly, the column of costs in Table 3.1 reports the computational complexity required by the most adopted complex initialization schemes: hidden in this notation there is a factor that depends on the number of iterations required by the specific scheme. This number, which is unknown a priori, depends on the iterative nature of any complex initialization scheme which has to satisfy some stopping criteria before providing the initial pair $(W^{(0)}, H^{(0)})$.

3.2 Constrained NMF

The key feature of NMF is to decompose the original data as combinations of parts. However, without any constraint the resulting parts could not be as intuitive as to help analyst in a clear understanding of data. In order to be easy to understand, parts should be composed by a small number of features; however, this structural requirement must be imposed in the factorization process. This can be achieved through different possible variants of NMF which have been proposed in literature.

More specifically, the objective function

$$f(W, H) = \|X - WH\|_F^2, \tag{3.1}$$

that is minimized by the NMF factorization process[1] can be modified in several ways in order to introduce additional properties on the resulting matrices. For example, a penalty term could be added to $f(W, H)$ in order to enforce sparseness [Hoyer, 2004] as well as to enhance smoothness [Essid and Févotte, 2013] or to improve clustering ability of NMF [Ding et al., 2005; Li and Ding, 2006, 2013]. Hence, a more general objective function can be formulated

$$f\left(W, H\right) = \|X - WH\|_F^2 + \alpha J_1\left(W\right) + \beta J_2\left(H\right), \tag{3.2}$$

where the penalty terms $J_1\left(W\right)$ and $J_2\left(H\right)$ add contraints to the original problem, whilst the regularization parameters α and β balance the trade-off between the approximation error and additional constraints.

Penalization terms are used in order to constrain the factorization process to yield more interpretable results, so as to be more suitable for IDA. In the following, constrained variants of NMF have been reviewed.

[1] In this chapter we mainly consider NMF based on the error function described in Equation 3.1, but other divergence measures could be used (e.g. generalized Kullback-Leibler divergence, α-divergence). Anyway, technical details apart, the general ideas described in the section still hold.

3.2.1 Sparse NMF

Sparseness is a quality that *"refers to a representational scheme where only a few units (out of a large population) are effectively used to represent typical data vectors"* [Hoyer, 2004]. Sparse representation of hidden factors makes them easier to be interpreted because the resulting parts are structurally simple.

In fact, NMF naturally promotes a sparse representation of data. The matrices W and H describe the relationships among the original features and the latent factors, and among the latent factors and the samples, respectively. Thus, there will be many zero-entries in these matrices where such relationships are not present in data. When the basis matrix W is sparse, basis vectors representing data subspace are sparse, thus only few features are used to describe the latent factors. This enables a part-based representation where each part is very simple and, therefore, easy to understand by the analyst. Similarly, when the enconding matrix H is sparse, then each sample is described by few (or just one) latent factors. This means that it possible to easily explain data samples as a composition of few parts.

Sparseness is desirable because it enhances interpretability; however it could negatively affect the accuracy of the approximation. Thus, sparseness should be regulated, but this is not possible in standard NMF unless some additional constraints are added. In [Hoyer, 2002, 2004], the classical NMF optimization algorithm has been modified to include the sparseness constraint. The basic idea is to introduce a measure of sparseness of a k-dimensional vector $\mathbf{x}$ as follows[1]:

$$\text{sparseness}(x) = \frac{\sqrt{k} - (\|\mathbf{x}\|_1)/(\|\mathbf{x}\|_2)}{\sqrt{k} - 1}. \tag{3.3}$$

This measure is then used to design a projected gradient descent algorithm that controls both sparseness and non-negativity. In essence, this algorithm essentially takes a step in the direction of the negative gradient of the cost function (3.1), and subsequently projects the solution onto the constraint space, that is the cone of

[1]The function in Equation 3.3 yields values in the interval $[0, 1]$, where 0 indicates the minimum degree of sparseness obtained when all the elements x_i have the same absolute value, while 1 indicates the maximum degree of sparseness, which is reached when only one component of the vector x is different from zero.

non-negative matrices with a prescribed degree of sparseness ensured by imposing the degree of sparseness to s_W and s_H for the matrices W and H, respectively.

Depending on the specific application of NMF, a desired degree of sparseness for W and H can be imposed. For example, when data samples represent images, high sparseness in both the encoding and the bases matrices is convenient. This allows to generate small *pieces* (factors) of the whole images, and few of them are used to describe each image. Differently, in a medical application where each data sample represent the symptoms of a patient and latent factors are diseases, we should expect to have a sparse encoding matrix H (because we expect patients have one or few more diseases) but W could be dense (since each disease could cause a large number of symptoms).

The prominent role of the data analyst to intelligently guide the factorization process is clear from these simple examples. Based on the questions the analyst wants to ask, and depending on the problem she needs to solve, the NMF process is modified by tuning the sparseness degree of its factors. Many variants of sparse NMF have been proposed subsequently to Hoyer's paper [Hoyer, 2002]. Some examples are Sparse Nonnegative Matrix Factorization, SNMF [Gao and Church, 2005; Kim and Park, 2007; Liu et al., 2003; Pauca et al., 2006], non-smooth Nonnegative Matrix Factorization [Pascual-Montano et al., 2006], LocalNMF [Feng et al., 2002; Li et al., 2001], Non-Negative Matrix Underapproximation (NMU) [Gillis and Glineur, 2010].

3.2.2 Orthogonal NMF and clustering capabilities

Dimensionality reduction can be exploited for endowing NMF with clustering capabilities. The theoretical relationship between NMF (with additional orthogonal constraints on its factors), k-means and spectral-based clustering was demonstrated [Ding et al., 2005], while the mathematical equivalence between orthogonal NMF and a weighted variant of spherical k-means was proved together with some indications about the cases in which orthogonal NMF should be preferred over k-means and spherical k-means [Pompili et al., 2012] .

Clustering is one of the most useful tools in IDA, since it produces a summarized view of data that helps the analyst to understand data by means of compact

and informative representations of large collections of samples [Berthold et al., 2010]. Many different clustering methods exist in literature, like hiearchical clustering, prototype-based clustering and density-based clustering (just to cite the most important ones). Hierarchical clustering yields a collection of nests groups of data, while in prototype-based clustering groups are represented in a compressed form through a prototype, i.e. an element belonging to the same domain of data. Finally, in density-based clustering groups are formed in regions of data space where data are more crowded. The choice of the most appropriate method is up to the data analyst.

In the case that prototype-based clustering is a convenient method for the problem at hand, NMF could be a valid tool. NMF has been widely used in clustering applications [Shahnaz et al., 2006; Xu et al., 2003] where the factors W and H have been interpreted in terms of cluster centroids and cluster membership, respectively.

From a geometric point of view, columns of W are the axes of the sub dimensional space where samples are spanned. They represent latent feature estracted from data. Vector samples are clustered according to their closeness to these basis vectors.

NMF without constraints finds a convex hull containing data points. However Ding et al. [2005] pointed out that adding orthogonality constraint to NMF algorithms is necessary to improve their clustering capabilities. In fact, the bases obtained from orthogonal NMF tend to point to the center of the clusters. The minimization problem in Equation 3.1 has been modified imposing orthogonality constraint to the rows of the encoding matrix H as follows:

$$\min_{W\geq 0, H\geq 0} \|X - WH\|_F^2, \quad s.t. \quad HH^\top = I. \tag{3.4}$$

Orthogonality constraint on the matrix H forces samples belonging to the same cluster to be closer to same bases. In the same manner, a feature clustering can be achieved by imposing the orthogonality constraint on the columns of the basis matrix W (i.e. $W^\top W = I$).

As a natural consequence, [Ding et al., 2005] proposed a new minimization problem. Simultaneously clustering of both features and objects (i.e. co-

clustering) has been archieved imposing orthonormality constraints on both columns of W and rows of H.

$$\min_{W\geq 0, H\geq 0} \|X - WH\|_F^2, \quad s.t.\, W^{\mathrm{T}}W = I,\, HH^{\mathrm{T}} = I. \tag{3.5}$$

In this representation the matrix W is the clustering indicator matrix, and the rows of the matrix H are the cluster centers for the features clustering problem; whilst the matrix H is the clustering indicator matrix, and the columns of the matrix W are the cluster centers for the objects clustering problem. However, this double orthogonality constraint is very restrictive and it leads to a rather poor matrix low-rank approximation. Different multiplicative updates for NMF preserving orthogonality were recently proposed [Choi, 2008a; Del Buono, 2009; J. and S., 2010]. To overcome the limits of the two factor orthogonal NMF, tri-factors NMF–TNMF has been proposed. Particularly, TNMF adds an extra factor to absorb the different scales of X,W,H and to allow different number of clusters for features and objects, that is

$$X \approx USV, \tag{3.6}$$

being $X \in \mathbb{R}_+^{n\times m}$, $U \in \mathbb{R}_+^{n\times k}$, $S \in \mathbb{R}_+^{k\times l}$, $V \in \mathbb{R}_+^{l\times m}$, where the number of rows in S correspond to the number of feature-clusters k, whilst the number of columns to the number of objects-clusters l.

The interested reader can find a deep investigation about NMF algorithms with ortogonality constraint and their application in clustering on [Lazar and Doncescu, 2009; Li and Ding, 2006, 2013; Mirzal, 2011].

3.2.3 Semi-Supervised NMF

NMF is an unsupervised machine learning algorithm, in fact it allows to *automatically* extract human-significative feature from data and to reduce the dimensionality of data. As it has been shown in the previous paragraph, classical NMF algorithms, and constrained ones, are widely used in clustering applications. They group data in a unsupervised way, but without taking in account any prior information of data. However, when class labels are available, this knowledge

could be injected in the factorization process, to improve the quality of clustering. Labelling dataset could be difficult, expensive, or time consuming, and often incomplete labels are available. Semi-supervised learning methods use a large amount of unlabeled data, together with labelled data, to train the process [Pise and Kulkarni, 2008].

Different algorithms have been also proposed in the context of NMF to inject a-priori knowledge. This can be done extending the objective function in Equation 3.1 to include extra terms containing the available a-priori knowledge (that could be class labels associated to the samples or pairwise constraints provided by the user, which indicate data to be clustered together –*must link*– and data that have not to be clustered together –*cannot link*). Research on NMF is going in the direction of considering it an interactive tool, instead of a black box. Semi-supervised NMFs allows to modify the factorization process taking in account the knowlege of the analyst. Some examples are [Cai et al., 2009; Chen et al., 2007, 2008, 2010; Cho and Saul, 2011; He et al., 2014a,b; Heiler and Schnörr, 2006; Jing et al., 2012; Lee et al., 2010; Liu and Wu, 2010; Liu et al., 2012; Lyubimov and Kotov, 2013; Wang et al., 2009, 2004; Yang and Hu, 2007].

Chapter 4

Subtractive clustering for seeding non-negative matrix factorizations

All algorithms for computing non-negative matrix factorization are iterative, therefore particular emphasis must be placed on a proper initialization of NMF because of its local convergence. The problem of selecting appropriate starting matrices becomes more complex when data possess special meaning as in document clustering. The first proposal presented in this thesis concerns the adoption of the subtractive clustering algorithm as a scheme to generate initial matrices for non-negative matrix factorization algorithms. It has been applied to a document clustering application and it has been shown to enhance the quality of the clustering results in term of interpretability of the cluster centers.

4.1 Proposed method

In this section, an initialization schema for NMF algorithms, inspired by the subtractive clustering (SC) algorithm [Chiu, 1994], is presented. In particular, after recalling the main steps performed by the subtractive clustering, the use of the cluster results to generate the initial $(W^{(0)}, H^{(0)})$ pair for any NMF iterative algorithm is illustrated.

It should be pointed out that, all clustering methods adopted to initialize NMF algorithms [Wild et al., 2004; Wild, 2003; Xue et al., 2008] need to fix the number of clusters corresponding to the rank factor k, defining the dimensionality of the NMF-subspace. SC, instead, is able to automatically discover the most appropriate value of k, when an estimation of distance among data is provided.

The proposed method has been applied to document clustering. Document representation as term-document matrix and clustering approach using NMF are detailed in the following paragraphs.

4.1.1 Data representation

In Information Retrieval (IR), the well known *bag-of-words* approach is commonly used to represent document corpora as numerical matrices. In this context, a collection of m documents is represented as a term-document matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \in \mathbb{R}_+^{n \times m}$ where each document $\mathbf{x}_i \in \mathbb{R}_+^n$ is an $n-$dimensional column vector of terms. Each entry x_{ij} indicates the contribution (*weight*) of the i-th term to the specification of the semantics of the j-th document. Several term weighting schemes have been proposed in literature [Polettini, 2004], however three general components could be identified:

$$a_{ij} = g_i t_{ij} d_j. \tag{4.1}$$

Where g_i is the global weight of the i-th term, t_{ij} is the local weight of the i-th term in the j-th document and d_j is the normalization factor for the j-th document. In this thesis *term frequency–inverse document frequency - (tf-idf)* weighting scheme has been used. It is a numerical statistic that is intended to reflect how important a word is to a document in a collection. In this scheme the local weight of the i-th term in the j-document is evaluated by term-frequency measure tf:

$$t_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} = tf_{ij}, \tag{4.2}$$

with $i = 1, \ldots, n$ and $j = 1, \ldots, m$, where n_{ij} is the frequency of the i-th term in the j-document, $\sum_k n_{kj}$ is the frequency of all the terms in document j; the

global weight of the i-th term is evaluated by inverse term frequency measure idf:

$$g_i = \log \frac{m}{|\{d : t_i \in d\}|} = idf_i, \tag{4.3}$$

where m is the total number of documents, the denominator counts the number of documents where i-th term appears, and d_i is the $L2$ normalization.

4.1.2 NMF for document clustering

Xu et al. [2003] proposed the use of NMF techniques for document clustering. Given a term-document matrix X, a nonnegative matrix factorization produces basis vectors $\mathbf{w_i}$ that isolate a subset of words denoting a particular concept or topic, while each column vector in H contains the encoding values of the linear combination of basis vectors to be used to approximate X. Therefore, each document can be identified as a combination of specific basis vectors, and therefore, it can be grouped as belonging to specific topics with different weights. Basis vectors in W represent cluster centers. Differently from other dimensionality reduction techniques they are directly interpretable from human expert, since each column $\mathbf{w_i}$ is a *bag of word*. From a geometric point of view, each concept represents an axis in the new subspace where data are spanned. Documents are vectors in this subspace, represented as linear combination of basis vectors weighted by coefficients in H. Figure 4.1 shows a simple example with three documents represented in a subspace described by two bases $\mathbf{w_1}$ and $\mathbf{w_2}$. "Insurance" and "car" are the semantics associated to the bases. Documents are grouped to the most close concept in term of euclidean distance. In the example document one belongs to the cluster identified by the concept car, and documents two and three to the custer insurance. Each document $\mathbf{x_i}$ is assigned to the cluster center that corresponds to the basis with the highest value in the column $\mathbf{h_i}$. Algorithm 2 specifies the steps used to cluster a document collection with NMF techniques. To make the solution unique, the authors require that the Euclidean lengths of $\mathbf{w_i}$ are one. For this purpose columns of W are normalized, and columns of H are accordingly adjusted in order to preserve the factorization results (lines 5.14 and 5.15).

Different NMF techniques and different initialization methods lead to differ-

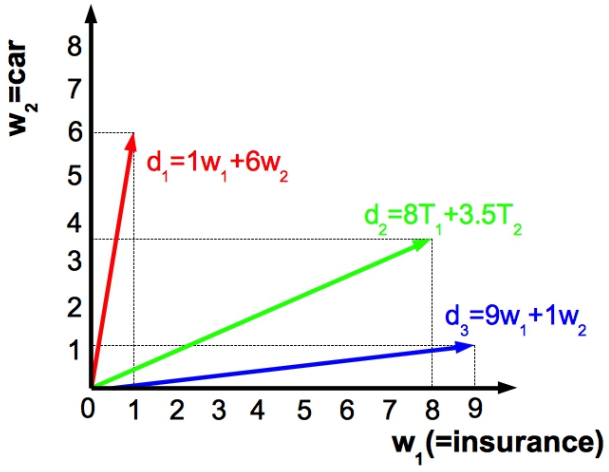


Figure 4.1: Example of document space created by NMF.

ent solutions. Particularly, as it has been discussed in Sec. 2.2.3 the number of concepts k is problem dependent. The choice of this parameter influences the effectiveness of an intelligent data analysis of NMF results. The proposed initialization method based on Subtractive clustering algorithm could be able to suggest a suitable number of concepts needed to describe the document collection.

4.1.3 Subtractive Clustering Initialization

Consider the data matrix $X = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}]$, where without loss of generality, each column vector $\mathbf{x_j} \in \mathbb{R}^n$ is assumed to be normalized to have unitary L_2 norm.

SC assumes that each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the actual cluster center, based on the potential of surrounding data points defined as follows:

$$\mathbf{p_j} = \sum_{k=1}^{m} \exp\left(-\frac{4}{r_a^2}\|\mathbf{x_j} - \mathbf{x_k}\|_2^2\right), \tag{4.7}$$

being r_a a positive constant representing a normalized radius defining a neighborhood. According to (4.7), high potential values correspond to a data point with many neighborhood data points. Hence, the potential of each data point is

Algorithm 2 Document Clustering based on NMF

Require: Document collection D

Require: Number of concepts k

1: Construct the term-document matrix X of document collection D

2: Perform NMF to carry out W and H

3: Normalize matrices W and H with:

$$w_{ij} \leftarrow \frac{w_{ij}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2}} \tag{4.4}$$

$$h_{ij} \leftarrow h_{ij} \sqrt{\sum_{i=1}^{n} w_{ij}^2} \tag{4.5}$$

for each column $j = 1 \ldots m$ of W and H.

4: Use columns of W as cluster centers

5: Use terms in each basis $\mathbf{w_i}$ to get the prototype of each cluster

6: Use coefficients $\mathbf{h_{ij}}$ to assign each document to a cluster.
Assign document d_j to cluster k if:

$$k = \arg\max_i h_{ij} \tag{4.6}$$

computed and then the point with the highest potential is selected as the first cluster center. Then, in order to avoid that points near the first cluster center could be selected as another center of the same cluster, an amount of potential proportional to the distance of each data point from the first cluster center is subtracted from it. After the potential reduction, the data point with the highest remaining potential is selected as the second cluster and the potential of each data point is further reducted, according to their distance to the second cluster center. Formally, after the k-th cluster center $\tilde{\mathbf{x}}_\mathbf{k}$ has been obtained with potential $\tilde{\mathbf{p}}_\mathbf{k}$, the potential of each data point is reduced by:

$$\mathbf{p_j} \leftarrow \mathbf{p_j} - \tilde{\mathbf{p}}_\mathbf{k} \exp\left(-\frac{4}{r_b^2} \|\mathbf{x_j} - \tilde{\mathbf{x}}_\mathbf{k}\|_2^2\right), \quad j = 1, \ldots, m, \tag{4.8}$$

where r_b is a positive constant. The process of finding new cluster centers and reducing the potential of all data iterates until the remaining potential of all data points is bounded by some fraction of the potential $\tilde{\mathbf{p_1}}$ of the first cluster center.

The stopping criterion usually adopted is $\tilde{\mathbf{p}}_\mathbf{r} < 0.15\tilde{\mathbf{p}}_\mathbf{1}$.

After the stopping criterion is satisfied, the SC applied to X provides: the number k of clusters, the cluster centroids and their potential values $\tilde{\mathbf{p}}_\mathbf{r}$, $r = 1, \ldots, k$.

The initial matrices $W^{(0)}$ and $H^{(0)}$ are constructed as follows. The basis matrix collects the cluster centroid vectors $\tilde{\mathbf{x}}_\mathbf{k}$ ordered by decreasing values of their potential $\tilde{\mathbf{p}}_\mathbf{k}$, i.e., $W^{(0)} = [\tilde{\mathbf{x}}_\mathbf{1}, \tilde{\mathbf{x}}_\mathbf{2}, \ldots, \tilde{\mathbf{x}}_\mathbf{r}]$. The encoding matrix $H^{(0)}$ provides the degree to which each data is assigned to each cluster. Particularly, the elements $h_{rj}^{(0)}$, $r = 1, \ldots, k$ and $j = 1, \ldots, m$, provide the fuzzy membership value for the j-th data in the r-th cluster and are computed by

$$h_{rj}^{(0)} = \frac{\exp\left(-\frac{1}{2}\frac{\|x_j - \mathbf{w}_\mathbf{r}^{(0)}\|_2^2}{\sigma^2}\right)}{\sum_{i=1}^{k} \exp\left(-\frac{1}{2}\frac{\|x_j - \mathbf{w}_\mathbf{i}^{(0)}\|_2^2}{\sigma^2}\right)}, \tag{4.9}$$

being k the total number of clusters and $\sigma^2 = \frac{{r_a}^2}{8}$.

Membership degrees are normalized because SC does not ensure the approximation $X \approx WH$ with the original membership degrees. By using (4.9) each column $\mathbf{x_j}$ is defined by the weighted mean of the prototypes in $W^{(0)}$.

4.1.4 Significance of the parameters in SC

SC is based on two hyper-parameters, r_a and r_b. The value of r_a confines the influence of data samples in determining the potential of a sample within a radius of length r_a. Indeed, it is easy to show that a sample located at distance r_a from another sample contributes to the potential of the latter by less than 0.02. The value of r_a could be therefore interpreted as the minimum distance that is acceptable for two samples to belong to different clusters.

SC uses the parameter r_b to reduce the potential of candidate cluster prototypes that are too close to an actual prototype. The potential of a candidate cluster prototype located at distance r_b from the current cluster prototype is indeed reduced by less than 0.02. Thus, the interpretation of r_b is the minimum distance that is acceptable for two cluster prototypes. Since cluster prototypes belong to different clusters by definition, then r_b should not be smaller than r_a.

Usually, the setting $r_b = 1.5r_a$ is suggested in literature.

4.1.5 Computational complexity

The computational complexity of SC is defined by the summation of two main terms. The first term involves the computation of a distance matrix between sample couples, which requires $O(nm^2)$ arithmetic operations. Once the distance matrix is computed, SC makes a number of iterations equal to the number of clusters to be discovered; in each cycle the computation of (8) is carried out, which requires $O(m)$ operations.

Most of the time is therefore required to compute the distance matrix, while the second step is very fast. However, the distance matrix can be computed just once for each dataset, and re-used in each run of SC. This favours re-running SC several times for finding the best configuration of hyper-parameters r_a and r_b.

4.2 Experimental Results

This paragraph is devoted to the illustration of the session of experiments that have been performed in order to demonstrate the effectiveness of the proposed initialization method. Five document datasets, which are described in subsection 4.2.1, have been used. Particularly, these experiments allow us both to assess the main characteristic of the proposed SC initialization scheme when compared with other complex initialization methods (i.e., Fuzzy C-means - FCM, k-means, Non-negative Singular Value Decomposition - NNSVD) and to provide some insights on the the performance of different NMF algorithms (i.e. Lee and Seung Multiplicative Update algorithm, Alternate Least Squared - ALS, Orthogonal NMF - ONMF, Sparse NMF - NMFSC), when these are initialized by complex initialization schemes. Besides the initialization proposal and the most commonly used complex initialization schemes, also the simple random initialization mechanism has been included, which could be considered the benchmark initialization scheme for any NMF algorithm. A study on the effects of hyper-parameters regulating the behaviour of the proposed SC initialization scheme has been preliminary conducted and the results discussed in subsection 4.2.3. Evaluations of

Dataset	Documents	Terms	Classes	Sparsity
CSTR	639	4016	4	98.56%
WebKB4	2785	7287	4	98.94%
Reuters8	5485	14551	8	97.37%
Reuters10	201	4184	10	99.74%
Newsgroups10	500	13709	10	99.15%

Table 4.1: Summary of the dataset statistics.

both initializations and NMF algorithms performances are provided and the obtained results are discussed in subsection 4.2.4 together with some results about the behaviour of NMF algorithms when stopping criteria are varied.

All the numerical results have been obtained by implementing the algorithms in Matlab 7.7 codes (the used codes for initialization and NMF are all publicly available) and running them on a machine equipped with an Intel Core Quad CPU Q6600 2.40 GHz.

4.2.1 Datasets

In the experiments, five datasets, which are detailed below, have been employed to evaluate the performance of the proposed initialization on different conditions. The dataset differ in number of terms and documents, contents and number of classes in which documents are categorized. Table 4.1 summarizes the dataset statistics reporting the number of documents, the number of terms composing the dictionary of the collection, the number of classes in which the documents are a priori grouped and the percentage of sparsity (i.e., percentage of zero elements) of the term-by-document matrix.

Each dataset has been pre-processed, by removing the stop words using an English common words dictionary, applying Porter stemming algorithm and leaving out local or global frequent terms. The term-by-document matrix has been composed using the standard TF-IDF weights, with cosine normalization. The open source software Term-Matrix Generator(TMG)[1] has been applied for these tasks.

[1] http://scgroup20.ceid.upatras.gr:8000/tmg/

CSTR. This dataset contains *URCS Technical Reports*: the abstracts of technical reports (TRs) published in the Department of Computer Science at University of Rochester between 1991 and 2002[1]. The dataset contains 639 documents expressed by a dictionary composed of 4016 terms. These documents are grouped into four categories corresponding to four research areas: Natural Language Processing (NLP), Robotics/Vision, Systems and Theory;

WebKB4. This dataset contains the web-pages collected by the World Wide Knowledge Base (WebKb) project of the CMU text learning group. These pages were collected from the Computer Science departments of various universities in 1997 and manually classified into seven different classes: student, faculty, staff, department, course, project, and other. The WebKB4 dataset includes the documents from the four most populous categories: project, course, faculty and student. A subset[2] containing only $2,785$ documents expressed by a dictionary composed of $7,287$ terms, have been used.

Reuters. The Reuters-21578 dataset[3] contains documents collected from the Reuters newswire in 1987. The full dataset contains $21,578$ documents manually classified into 52 categories. Due to the fact that the class distribution for these documents is very skewed and the categories are highly correlated, we considered two sub-collections: Reuters8 and Reuters10, which differ in terms of document-term ratio and number of classes.

- **Reuters8.** This is a high dimensional dataset[4] composed by a dictionary of $14,551$ terms and $5,485$ documents grouped into only eight categories: acq, crude, earn, grain, interest, money, ship, trade.
- **Reuters10.** This is a simpler dataset[5] created by selecting 20 files each from the 10 largest classes in the Reuters-21578 collection. There are 10 directories labelled by the topic name (acq, corn, crude, earn,

[1] http://www.cs.rochester.edu/trs/
[2] http://web.ist.utl.pt/acardoso/datasets/
[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[4] Defined starting from the training set at: http://web.ist.utl.pt/acardoso/datasets
[5] http://archive.ics.uci.edu/ml/datasets/Reuters+Transcribed+Subset

grain, interest, money, ship, trade, wheat), each of them containing 20 files of transcriptions.

Newsgroups. The entire Newsgroups10 [1] dataset contains about 20,000 articles, subdivided into 20 categories. The first 50 documents, for each of the top 10 categories have been considered, obtaining a small subset of only 500 documents, named Newsgroups10. This dataset is used to verify the results on a dataset with a small number of documents for each category and a relatively high number of categories.

4.2.2 Evaluation metrics

The primary concern has been the evaluation of overall initialization-factorization process. The initialization strategies have been compared both in terms of effectiveness of the starting pair $(W^{(0)}, H^{(0)})$ and of run-time required to compute these initial factors (this time was evaluated in seconds). The initial error – that is the value assumed by the divergence measure D into the pair $(W^{(0)}, H^{(0)})$ – has been used to measure the initial reconstruction error. It should be pointed out that the run-time value for the random initialization has been omitted (being negligible) and that this initialization scheme produces full initial matrices with poor accuracy.

NMF algorithms with different initializations have been compared in terms of values of the divergence function D into the final pair $(W^{(fin)}, H^{(fin)})$, where the matrices $W^{(fin)}$ and $H^{(fin)}$ are obtained when the stopping criteria are satisfied. The behaviour of the divergence measure D during the iteration process has been also provided. For a fair comparison among all the algorithms, the same stopping criteria have been adopted: common maximum number of iterations ($maxiter = 500$) and fixed tolerance ($toll = 10^{-6}$) for the difference between two subsequent values of the divergence measure D.

[1] http://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

4.2.3 Effects of SC parameters

The first step performed during the experimental session was the empirically evaluation of the influence of the hyper-parameters (i.e., r_a and r_b) in the SC initialization. It should be pointed out that these parameters influence the clusters number, so that they determine the proper rank factor k in the context of NMF. In fact, a strong point of the SC initialization scheme lies in its capability of suggesting the most suitable rank value (i.e., the number of final clusters obtained at the end of the SC process) to be used for a given dataset, when average distance between data is estimated. Table 4.2 reports the computational times (in seconds) needed to compute the matrix of Euclidean distances among documents for each adopted dataset.

CSTR	Newsgroup10	WebKB4	Reuters8	Reuters10
2.84	3.65	82.25	158.42	0.48

Table 4.2: Computational times (in seconds) required to construct the matrices of Euclidean distances between different columns of each data matrix

Being, r_a and r_b the hyper-sphere cluster and penalty radius in the data space, respectively, they can be estimated on the basis of the distances among the documents in the term-by-document matrix. Since documents in each dataset possess unit Euclidean norm, they lie on the surface of the unitary hyper-sphere, so that the distance value among any two documents varies in the interval $[0, \sqrt{2}]$ [1] (i.e., the minimum value 0 corresponds to two identical documents, the maximum value $\sqrt{2}$ corresponds to term totally different documents).

Due to the high multidimensionality of the datasets adopted in the experiments, high distance values among documents were observed, hence only a rightmost-side subinterval of $[0, \sqrt{2}]$ was considered for determining r_a values (the used interval corresponds to the 5th-95th percentile range of the observed

[1] Given two documents $x_i, x_j \in \mathbb{R}^n$, the distance between x_i and x_j is evaluated by $d = \|x_i - x_j\|_2$. When $x_i = x_j$, the vectors are overlapped and their distance $d = 0$. On the contrary, when the document vectors are orthogonal, and lie in the unit sphere (i.e. in the 2-dimensional case $x_i = \begin{matrix}1\\0\end{matrix}$ and $x_j = \begin{matrix}0\\1\end{matrix}$) their distance, measured by the Euclidean norm, is $d = \sqrt{\sum_{r=1}^{n} (x_i - x_j)^2} = \sqrt{2}$.

document distance values). Table 4.3 indicates the range interval for parameters r_a for each of the adopted dataset.

CSTR	Newsgroup10	WebKB4	Reuters8	Reuters10
[1.3476, 1.4142]	[1.3851, 1.4142]	[1.3612, 1.4142]	[1.3148, 1.4142]	[1.3221, 1.4137]

Table 4.3: Interval values for the parameter r_a obtained from the 5th-95th percentile range of document distance values for each dataset used in the experimental session.

The SC algorithm was run for different values of r_a in the chosen range and the number of clusters suggested by the SC was observed. Since this algorithm is also subjected to the r_b penalty parameter, for each fixed value of r_a, the variation of clusters number when $r_b = \alpha * r_a$, with $\alpha \in [1, 2]$ has been evaluated. Particularly, a large variation in the final numbers of clusters was observed. In particular, the increase of α produces a reduction of the number of clusters (since more documents are clustered in each cluster). Hence, this parameter identifies the granularity of the results.

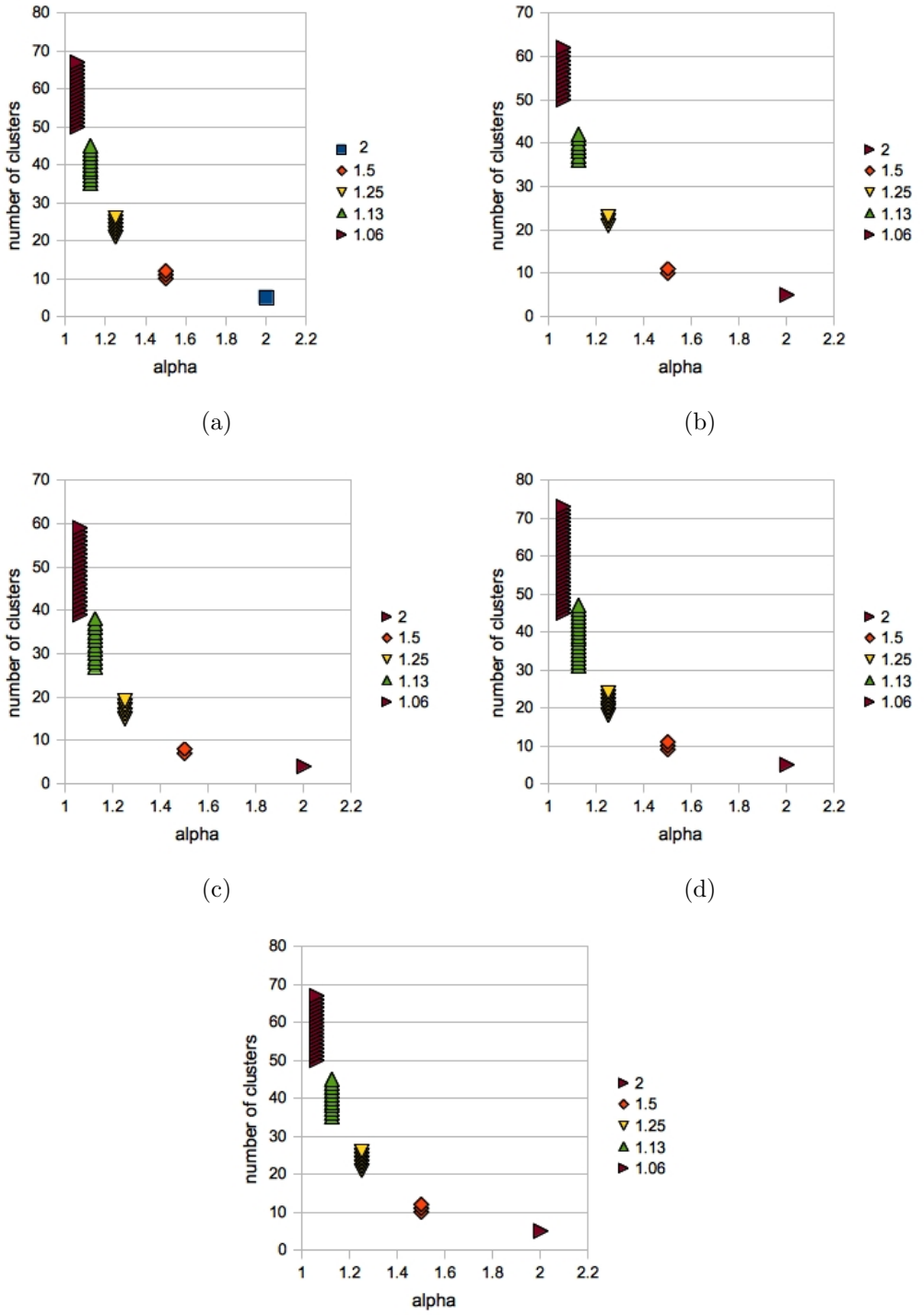
Figure 4.2 shows the effects on the number of different clusters obtained when SC scheme is applied to the selected datasets, when the parameter r_b is varied. The results evidence that the SC algorithm is quite sensitive to the parameter r_b.

Accordingly to this preliminary study on the effects of the hyper-parameters, the value $\alpha = 1.5$[1] has been selected to be used in the following part of the experimental session. This choice reflects a more stable behaviour of the SC scheme and suggests a number of clusters more suitable from an interpretability point of view.

4.2.4 Results

The results of the numerical experiments conducted to compare the SC scheme with other initialization methods could be analyzed from different points of view: the particular NMF algorithm to be initialized, the used dataset, the number of suggested bases. First, the effectiveness of the initialization methods has been evaluated; then the influence of different initializations on NMF algorithm performance has been discussed.

[1] $r_b = 1.5 r_a$



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

Figure 4.2: Suggested number of clusters when the coefficient α is varied. Pictures refer to the adopted datasets: (a) CSTR , (b) Newsgroup10, (c) Reuters8, (d) Reuters10, (e) WebKB4.

Tables 4.4-4.8 summarize the effectiveness of both initialization methods and NMF algorithms (initialized with different schemes) on approximating the adopted datasets. The first two columns report the initial errors and the run-time values of every initialization scheme, the remaining columns report the evaluation of the NMF algorithm, that is the final error and the number of iterates performed till the stopping criteria are satisfied. Each row in the table points out the initialization schemes adopted. For each dataset, the initialization-factorization process has been performed for different values of the rank factor k. The choice of the particular value corresponds to the number of clusters suggested by SC algorithm with the parameters configuration set as described in subsection 4.2.3. This value has been used as a priori information for the other initialization schemes in order to produce data approximations belonging to a low-rank space of the same dimensions.

It has been observed that the SC initialization scheme requires the lowest computational time to compute the pair $(W^{(0)}, H^{(0)})$, but it demonstrates a slightly increase of the initial error with respect to the others complex initialization mechanisms. This behaviour depends on the construction of the basis vector in W, which are taken to be the most representative (in term of potential) documents of the collection. The final errors for NMF algorithms initialized by SC are nearly always comparable with the performance obtained by other initializations. However, it has to be reminded that the NMF problem is a non-convex optimization task, so the initialization-factorization process could only guarantee locally optimal solutions. On the other hand, the number of iterations performed by NMF algorithms when initialized by SC scheme, results to be lower (in half of the trials) than the other complex initialization schemes, except for the k-means initialization.

As concerning the results obtained by k-means initialization, it should be reminded that NMF are equivalent to a relaxed form of k-means clustering and the latter represents a low rank approximation of the original data matrix [Ding et al., 2005]. Basically, k-means seeks a point-wise way to solve the formulation of clustering as a matrix factorization. Hence, the initial error for the k-means initialization results lower than those given by other initialization schemes, and is closer to the final errors (with respect to all the NMF algorithms). In fact,

(a) Rank factor $k = 10$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0163	320.66	293.44	70	282.43	112	293.53	316	286.12	341
FCM	7.7878	310.10	282.53	244	282.63	84	283.24	499	286.67	500
NNSVD	0.1277	300.43	283.45	130	282.42	34	284.14	366	286.27	500
Rand	-	$6.1031e^7$	282.19	500	282.44	48	283.01	499	286.26	324
k-means	40.2523	288.83	286.60	9	282.79	48	288.53	327	288.99	42

(b) Rank factor $k = 11$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0118	317.40	291.59	269	280.38	97	291.75	166	284.48	321
FCM	10.0858	310.10	280.28	500	280.74	147	281.38	499	284.48	500
NNSVD	0.1387	300.65	281.57	166	280.37	114	282.29	240	284.67	500
Rand	-	$6.9296e^7$	281.02	500	281.05	163	281.32	499	284.88	475
k-means	41.83	287.92	285.58	13	280.74	80	287.87	307	288.21	51

(c) Rank factor $k = 12$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0086	315.41	290.54	147	278.82	123	290.74	410	283.31	404
FCM	9.4267	310.10	279.12	500	278.98	140	279.53	499	283.64	500
NNSVD	0.1566	300.93	279.75	185	278.48	117	280.50	237	283.10	500
Rand	-	$8.5154e^7$	278.78	500	278.67	93	279.62	499	283.21	500
k-means	81.0650	286.09	283.83	13	278.62	174	286.20	357	286.72	57

Table 4.4: Performance of the NMF algorithms initialized with different strategies applied to CSTR dataset for different value of the rank factor k.

during the initialization phase, k-means algorithm finds a quite optimal clusters, which are only refined by NMF algorithms (with high convergence rate). Despite of its good convergence rate, k-means scheme requires high computational cost in the initialization which grows quickly as the dimensions of the term-by-document matrix increase.

Figure 4.3 provides some illustrative examples of the performance of all the initialization methods used in this paper combined with the NMFSC algorithm when applied to some of the adopted datasets. As it can be observed, SC scheme provides initial values that enable the NMFSC algorithm to reduce the value of the initial error after very few iterations and at a low overall cost, at levels slightly better than those obtained after running the algorithm with one of the other initializations. Similar behaviour of the SC-NMFSC initialization-factorization scheme can be observed for all trials we have conducted and for any adopted

(a) Rank factor $k = 10$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0082	257.59	234.57	59	229.78	47	234.61	19	232.39	258
FCM	16.9350	246.45	229.89	500	229.76	42	230.44	499	232.43	500
NNSVD	0.3620	238.50	230.17	76	229.76	21	230.33	150	232.23	187
Rand	-	$1.5705e^8$	229.78	500	229.83	56	229.95	470	232.28	500
k-means	138.7318	234.36	232.67	40	229.81	71	233.13	75	234.50	135

(b) Rank factor $k = 11$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0077	254.74	233.39	60	228.55	55	233.44	222	231.42	395
FCM	18.7174	246.45	228.48	500	228.48	44	229.05	499	231.23	500
NNSVD	0.3750	237.87	228.90	89	228.43	32	229.09	151	231.19	198
Rand	-	$1.9538e^8$	228.47	428	228.54	34	228.69	499	231.35	500
k-means	185.8935	233.19	231.60	90	228.91	66	232.10	202	233.69	156

Table 4.5: Performance of the NMF algorithms initialized with different strategies applied to Newsgroups10 dataset for different rank factor k.

(a) Rank factor $k = 7$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.4858	2904.19	2467.31	500	2427.13	284	2468.36	499	2496.88	500
FCM	399.5160	2666.17	2426.72	500	2429.59	226	2432.04	403	2498.70	500
NNSVD	0.5269	2531.84	2432.45	310	2426.58	254	2434.38	499	2510.31	500
Rand	-	$9.5854e^8$	2425.92	500	2426.58	263	2431.59	499	2503.27	500
k-means	1030.903	2497.71	2475.80	14	2427.13	273	2484.15	441	2523.77	126

(b) Rank factor $k = 8$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.4429	2831.15	2455.04	500	2413.06	456	2456.54	499	2487.87	500
FCM	393.3133	2666.17	2413.46	500	2411.08	171	2420.49	499	2488.48	500
NNSVD	0.5802	2532.94	2417.45	347	2411.08	154	2419.62	379	2501.39	500
Rand	-	$1.2174e^9$	2410.78	500	2411.08	166	2413.53	499	2486.59	500
k-means	660.6835	2465.12	2444.43	18	2413.06	240	2478.61	499	2499.64	147

Table 4.6: Performance of the NMF algorithms initialized with different strategies applied to Reuters8 dataset for different rank factor k.

(a) Rank factor $k = 9$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0020	96.38	86.53	92	83.04	52	86.71	373	84.33	294
FCM	3.3035	95.36	83.13	311	83.11	103	83.63	499	84.47	500
NNSVD	0.1201	91.95	83.69	152	83.10	52	84.17	499	84.53	349
Rand	-	$1.6569e^7$	82.94	427	83.02	195	83.57	317	84.46	500
k-means	10.1670	85.61	84.94	15	83.18	166	86.33	333	85.82	53

(b) Rank factor $k = 10$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0016	94.48	85.53	94	82.04	142	85.73	322	83.52	304
FCM	3.6644	95.36	81.97	500	82.25	186	82.66	494	83.63	500
NNSVD	0.1371	92.08	82.74	205	82.05	66	83.26	499	83.66	279
Rand	-	$1.9581e^7$	81.95	433	82.12	189	82.57	499	83.66	486
k-means	9.9055	84.88	83.96	79	82.12	212	85.12	216	84.88	139

(c) Rank factor $k = 11$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.0027	92.72	84.62	94	81.10	140	84.82	324	82.78	500
FCM	4.0298	95.36	80.98	500	80.99	95	81.69	499	82.77	500
NNSVD	0.1471	92.52	81.83	211	81.04	500	82.39	310	82.83	500
Rand	-	$2.3196e^7$	81.16	500	81.11	102	81.63	499	82.90	419
k-means	7.6596	84.03	83.01	17	81.18	153	84.12	271	84.10	51

Table 4.7: Performance of the NMF algorithms initialized with different strategies applied to Reuters10 for different rank factor k.

(a) Rank factor $k = 10$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.1988	1437.30	1310.25	176	1288.84	43	1310.47	381	1301.51	500
FCM	52.1340	1358.41	1289.35	500	1289.22	78	1292.43	499	1302.25	500
NNSVD	0.6578	1329.05	1291.99	335	1289.27	95	1292.28	499	1302.54	500
Rand	-	$4.7344e^8$	1289.27	500	1289.14	183	1290.72	327	1303.32	500
k-means	1687.948	1312.55	1306.05	26	1290.86	323	1315.80	499	1311.59	100

(b) Rank factor $k = 11$

	Eff. of init.		NMFLS		ALS		ONMF		NMFSC	
Initial.	Time	Err.	Err.	iter	Err.	iter	Err.	iter	Err.	iter
SC	0.1393	1426.05	1304.92	174	1283.76	137	1305.15	356	1297.55	500
FCM	57.7798	1358.41	1285.17	500	1283.94	268	1285.37	499	1298.48	500
NNSVD	0.6718	1330.34	1286.52	310	1284.25	140	1288.61	499	1298.11	500
Rand	-	$5.6985e^8$	1285.05	500	1283.91	102	1290.28	413	1297.25	500
k-means	2281.688	1307.42	1300.67	42	1284.83	206	1308.70	499	1307.46	79

Table 4.8: Performance of the NMF algorithms initialized with different strategies applied WebKB4 dataset for different rank factor k.

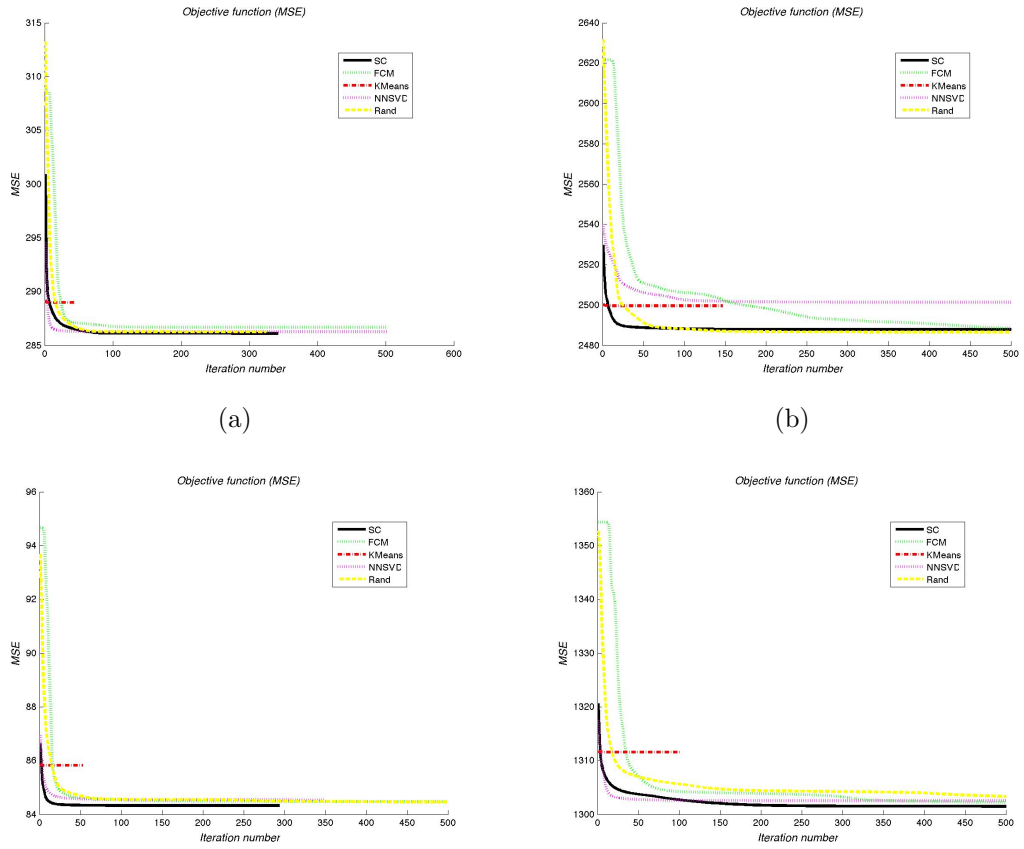


Figure 4.3: Objective function behaviour during the iteration process for the NMFSC algorithm applied on the different datasets: (a) CSRT with $k = 10$, (b) Reuters8 with $k = 8$, (c) Reuters10 with $k = 9$, (d) WebKB4 with $k = 10$.

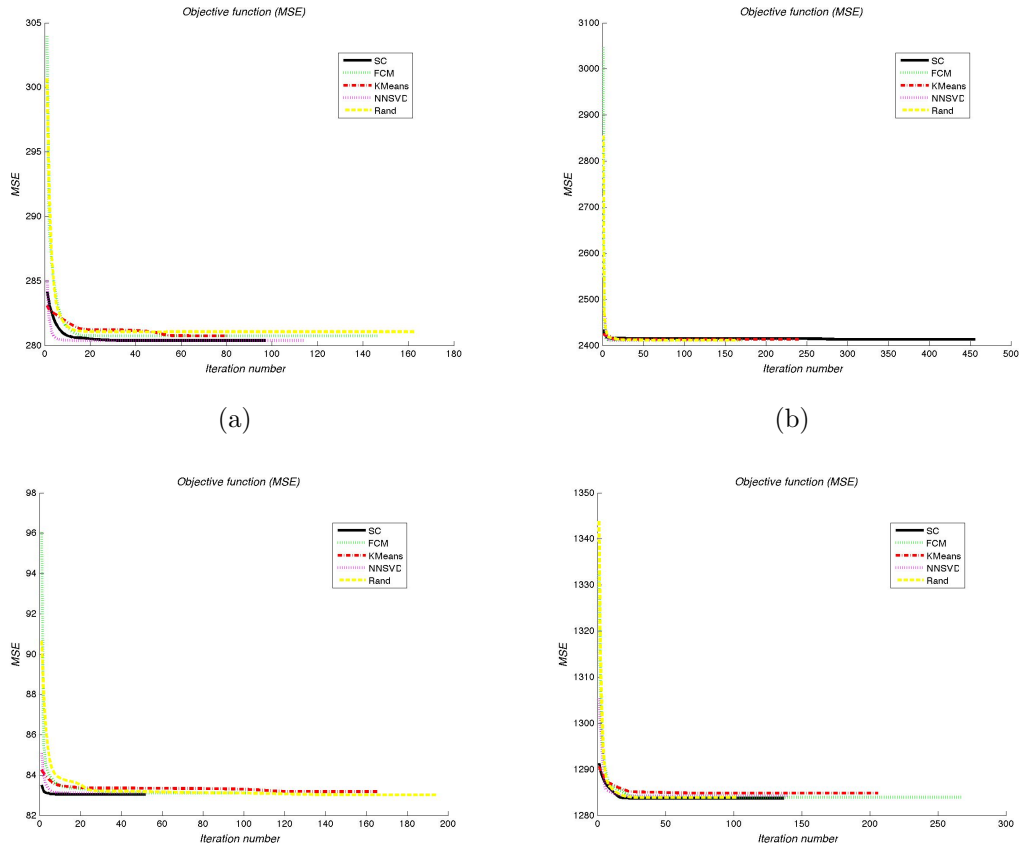


Figure 4.4: Objective function behaviour during the iteration process for the ALS algorithm applied on the different datasets: (a) CSRT with $k = 11$, (b) Reuters8 with $k = 8$, (c) Reuters10 with $k = 9$, (d) WebKB4 with $k = 11$.

dataset.

Figure 4.3 provides some illustrative examples of the performance of all the initialization methods used in this paper combined with the ALS algorithm when applied to some of the adopted datasets. Also in this case the initial values provided by the SC scheme allow to a fast reduction of the objective function after few iterations leading to a final error which is comparable with the values reached by NNSVD (which demonstrated to be the better initialization scheme for ALS algorithm). Tables 4.9 and 4.10 report the final error and the number of iterates of NMFSC and ALS algorithms, respectively, when initialized with

different schemes and the tolerance *toll* is varied.

From the illustrative plots depicting the behaviour of the objective function during the learning process, some considerations about the influence of the stopping criteria on the NMF algorithms can also be drawn . As previously observed, both NMFSC and ALS algorithm appear to take some advantages in terms of fast convergence when initialized by SC scheme. This means that the curve depicting the values of the objective function presents a knee behaviour before converging to its asymptotic behaviour. Varying the tolerance value adopted in the stopping criteria does not influence the presence of this elbow, but only the number of least significant digits in which two subsequent values of the divergence measure D differs. Consequently, the number of iterations performed by the NMF algorithm to satisfy the fixed tolerance decreases (respectively, increases) when the fixed tolerance is reduced.

Moreover either for NMFLS or for ONMF initialized by SC did not show this kind of behaviour in the course of this investigation. For NMFLS and ONMF, only the NNSVD scheme outperforms the other complex initializations.

4.2.5 Effects of the rank on the cluster granularity

As demonstrated in [Xu et al., 2003], when applied on document data corpora, NMF can be interpreted in terms of document clustering: particularly each column vector of the basis matrix W is a semantic feature that could be represented by its ten highest frequency words. Moreover, each column of the encoding matrix H allows to represent documents of X in the subspace defined by W. In this scenario, each document in the original dataset X is assigned to the semantic feature with the highest value in its columns.

Although clustering is an unsupervised method, labeled data are commonly used to assign documents to some classes which are a-priori known. Particularly, when NMF is used for tackling a document clustering problem, the factor rank k is chosen equal to the number of original classes in which the data have been initially grouped.

Reasonably, this class structure might be quite different from the unknown cluster structure underlying the data, so that this a-priori choice of the rank

(a) Dataset Reuters10 with rank value $k = 9$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	84.78	7	84.42	16	84.35	35	84.33	91	84.33	168	84.33	294
FCM	94.66	2	94.66	2	84.57	71	84.49	227	84.48	345	84.48	500
NNSVD	85.07	8	84.67	21	84.58	51	84.54	126	84.54	216	84.54	349
Rand	85.44	19	84.83	36	84.57	79	84.55	131	84.46	500	84.46	500
k-means	85.86	2	85.83	4	85.83	7	85.82	14	85.82	27	85.82	53

(b) Dataset CSTR, with rank value $k = 10$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	287.63	15	286.37	54	286.14	89	286.13	127	286.12	209	286.12	341
FCM	308.66	2	308.64	3	308.64	3	286.68	197	286.68	332	286.68	500
NNSVD	286.69	10	286.40	20	286.30	47	286.28	86	286.27	253	286.27	500
Rand	287.92	22	286.38	62	286.29	87	286.27	150	286.27	210	286.27	324
k-means	289.03	2	289.01	3	289.00	7	289.00	12	289.00	22	289.00	42

(c) Dataset Reuters8 with rank value $k = 8$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	2489.78	25	2488.48	81	2487.96	188	2487.88	393	2487.87	500	2487.87	500
FCM	2621.88	4	2491.61	352	2488.48	500	2488.48	500	2488.48	500	2488.48	500
NNSVD	2506.39	49	2502.09	126	2501.48	268	2501.40	457	2501.39	500	2501.39	500
Rand	2489.55	62	2487.04	147	2486.71	247	2486.61	426	2486.60	500	2486.60	500
k-means	2499.85	4	2499.70	9	2499.65	20	2499.64	41	2499.64	78	2499.64	147

(d) Datset WebKB4 with rank factor $k = 10$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	1305.08	21	1302.17	139	1301.58	265	1301.53	334	1301.51	500	1301.51	500
FCM	1354.33	2	1354.33	2	1354.33	3	1354.33	4	1354.33	5	1302.26	500
NNSVD	1303.54	17	1302.88	39	1302.62	104	1302.56	250	1302.55	430	1302.55	500
Rand	1308.16	32	1304.74	150	1304.23	303	1303.32	500	1303.32	500	1303.32	500
k-means	1311.67	2	1311.64	3	1311.60	10	1311.60	23	1311.60	50	1311.60	100

Table 4.9: Behaviour of NMFSC algorithm when varying the tolerance value adopted in the stopping criteria.

(a) Dataset Reuters10 with rank factor $k = 9$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	83.21	3	83.07	6	83.05	13	83.04	21	83.04	33	83.04	52
FCM	83.71	8	83.36	16	83.12	43	83.11	59	83.11	77	83.11	103
NNSVD	83.26	6	83.13	10	83.11	15	83.11	29	83.11	45	83.11	52
Rand	84.00	9	83.22	28	83.21	33	83.02	153	83.02	172	83.02	195
k-means	83.74	6	83.42	16	83.36	28	83.19	135	83.19	147	83.19	166

(b) Dataset CSTR with rank factor $k = 11$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	280.84	10	280.43	27	280.38	37	280.38	43	280.38	45	280.38	97
FCM	280.96	14	280.77	18	280.75	23	280.75	29	280.75	38	280.75	147
NNSVD	280.55	6	280.41	10	280.38	17	280.38	30	280.38	35	280.38	114
Rand	281.33	12	281.10	18	281.06	27	281.06	40	281.06	48	281.06	163
k-means	281.68	12	281.22	20	281.20	25	280.75	65	280.75	74	280.75	80

(c) Dataset Reuters8 with rank factor $k = 8$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	2415.55	11	2415.31	18	2415.15	66	2415.14	69	2415.14	70	2413.07	456
FCM	2411.72	9	2411.01	22	2410.99	26	2410.99	26	2410.99	26	2411.08	171
NNSVD	2411.46	11	2411.09	19	2411.07	23	2411.07	24	2411.07	24	2411.08	154
Rand	2412.08	27	2412.02	29	2411.05	66	2411.05	67	2411.05	67	2411.08	166
k-means	2413.25	17	2413.13	21	2413.07	41	2413.06	58	2413.06	62	2413.07	240

(d) Dataset WebKB4 with rank factor $k = 11$

	10		10^{-2}		10^{-3}		10^{-4}		10^{-5}		10^{-6}	
Initial	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter	Err	iter
SC	1283.89	18	1283.78	21	1283.76	26	1283.76	32	1283.76	36	1283.76	137
FCM	1284.42	24	1283.97	36	1283.93	45	1283.93	47	1283.93	47	1283.94	268
NNSVD	1284.51	15	1284.29	20	1284.26	27	1284.26	33	1284.27	80	1284.26	140
Rand	1284.37	19	1283.91	29	1283.89	36	1283.89	41	1283.89	42	1283.91	102
k-means	1285.40	22	1285.09	30	1284.85	74	1284.84	111	1284.84	161	1284.84	206

Table 4.10: Behaviour of ALS algorithm when varying the tolerance value adopted in the stopping criteria.

factor could lead to poor results with incorrect extracted features.

On the other hand, SC initialization scheme can help the proper analysis of the structure of the data due to its capability of suggesting a most suitable value for the parameter k.

The following tables report the semantic feature extracted by the NMFSC algorithm initialized with SC scheme when CSTR dataset is considered. The reported basis vectors have been obtained with rank value $k = 4$ and $k = 11$. The first value represents the number of classes into which the originally documents have been grouped while the second value represents the embedded information automatically extracted by the SC initialization algorithm.

The four a priori classes in the CSTR dataset were: Natural Language Processing (NLP), Robotics/Vision, Systems and Theory.

The feature extracted by the NMF with $k = 4$ are not able to describe exhaustively these topics as shown in Table 4.11.

W_1	W_2	W_3	W_4
memori	set	task	object
share	select	manipul	recognit
program	class	robot	imag
parallel	polynomi	control	train
cach	hierarchi	plan	view
perform	prove	visual	learn
coher	complet	method	system
applic	string	freedom	imageri
data	bound	space	represent
multiprocessor	collaps	real	supervis

Table 4.11: Example of semantic feature extracted with NMFSC algorithm and SC initializiation when CSTR dataset is considered. Rank value set to $k = 4$.

Table 4.12, on the other hand, reports the features extracted at the end of the NMF process performed with rank $k = 11$. Particularly, the last row of the table reports the semantic concepts which have been associated to the extracted

features.

As it can be observed, these semantic concepts slightly differ from the a-priori classes, but they appear to be more specific and reflect the presence of some geometric properties underlying the data that are captured by the clusters.

These results assess the effectiveness of the proposed initialization method. A point of originality in using SC initialization scheme lies in its capability of suggesting the most suitable rank value k for a given dataset. In fact, as discussed in Sec. 4.1.4, the parameters r_a and r_b allow to set the radius of the hyper-sphere clusters according with the locality property of the data documents in the Euclidean space. This means that documents which were close in the original space remain close in the subspace obtained at the end of the NMF learning process. As it has been shown in the example, the granularity of the problem suggested by the SC method enables the extraction of more interpretable semantic features than using the number of known classes as factor rank. The increased interpretability reflects the concepts enclosed in the most representative documents selected by the SC scheme to be the basis vector in $W^{(0)}$.

W_1	W_2	W_3	W_4	W_5
lock	heterogen	coher	train	database
syncrhron	loop	cach	recognit	itemset
barrier	load	memori	object	mine
scalab	balanc	share	learn	algorithm
multiprogram	processor	hardware	view	rule
mutual	netwotk	protocol	image	cluster
reader	schedul	softwar	system	associ
writer	compil	perform	supervis	frequent
schedul	parallel	multiprocessor	unlabel	transact
exclus	commun	dsm	geometr	discoveri
parallel programming	**heterogeneous networks**	**operating systems**	**object recognition**	**datamining**

W_6	W_7	W_8	W_9	W_{10}	W_{11}
visual	select	hard	parallel	object	probabilist
fixat	set	nontrivi	program	imag	log
predict	polynomi	rice	transform	featur	space
respons	hierarchi	theorem	data	kei	automata
neural	reduct	prove	control	method	ture
model	function	bound	local	recognit	error
filter	string	count	compil	cluster	finit
task	complet	properti	model	system	class
cortic	collaps	nondetermin	optim	invari	machin
scene	equival	circuit	array	base	nondeterminist
neural networks for visual.	**complexity theory**	**computability theory**	**algorithms**	**image recognition**	**finite state automata**

Table 4.12: Example of semantic features extracted using NMFSC algorithm and SC initializiation when CSTR dataset is considered. Rank factor $k = 11$.

Chapter 5

Part-based data analysis with Masked Non-negative Matrix Factorization

The second proposal concerns a novel masked nonnegative matrix factorization algorithm which is used either to explain data as a composition of interpretable parts (which are actually hidden in them) as well as to introduce knowledge in the factorization process. Masking enables the decomposition of data into user-defined parts, which are consequently easy to understand by the analyst. The results of Masked NMF enables the analyst to understand which subsets of the available data are best represented by the specified parts, thus extracting potentially useful knowledge from large quantities of data.

5.1 Proposed method

The non-negativity characterization of NMF makes it a useful tool for Intelligent Data Analysis (IDA). In fact, the non-negativity makes NMF capable of representing data as an additive combination of common factors. Moreover, if such factors have some physical meaning (i.e., they can be interpreted in the domain of the considered problem), NMF allows to explain data as a composition of parts, being each part a factor. The problem of interpreting parts as small selections

of features is faced. More precisely, the column vectors of W are constrained so that only a small subset of elements is non-zero. This representation of parts could be very useful for IDA, since it is able to highlight some local linear relationships existing among features that hold for a subset of data. To this purpose, a new optimization problem for NMF has been introduced, which constrains the columns of the base matrix W to possess a small number of non-zero elements. Then, a query-based approach is adopted, where the structure of the base matrix is defined by a user-provided mask matrix. In this way, the analyst can specify the parts she is interested to discover in data: the proposed technique, in fact, extracts the subset of data that are actually represented by these parts.

5.1.1 Masked NMF

The non-negativity constraints imposed by NMF often are not enough to produce factors that represent useful knowledge. Usually these columns are very dense; moreover, different configurations of W and H lead to the same approximation of X, thus it could be difficult to associate a physical meaning to the factors.

To overcome the limits of classical NMF and to inject a-priori knowledge in the factorization process, the concept of part has been introduced. From the vector representation of data it is possible to observe that each sample is represented by a vector $x \in \mathbb{R}_+^n$ of n features $\{f_1, ..., f_n\}$. A part p is defined as a sparse vector in $\mathbb{R}^n$ where at least two components are non-zero. A feature belongs to a part iff its value is non-zero. In this way the factorization process is constrained to describe data as a linear correlation of different parts, whose features are linearly correlated among them. The structure of the part (i.e. the features set to zero, thus excluded by the part), as well as the number of parts, constitutes the a-priori knowledge and is user-defined.

To obtain basis factors that are able to extract parts, the columns $\mathbf{w}_k$ in W are constrained to contain only few non-zero elements. Factors possessing this type of structure enable the elicitation of local linear relationships in subsets of data.

A binary matrix $P \in \{0, 1\}^{n \times k}$, with the same dimensions of the basis matrix W is used as mask for the NMF problem. Particularly, the mask matrix P is

$$P = \begin{array}{c} \begin{array}{ccc} \textit{Part 1} & \textit{Part 2} & \textit{Part 3} \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{array}{l} \text{Feature 1} \\ \text{Feature 2} \\ \text{Feature 3} \\ \text{Feature 4} \\ \text{Feature 5} \\ \text{Feature 6} \end{array} \end{array}$$

Figure 5.1: Example of query matrix P.

used to identify the parts that the analyst would like to extract from data. This is accomplished by defining P as a set of k column vectors, where each element in a column is 1 if the corresponding feature has to be selected, 0 if it has not be considered. Figure 5.1 shows an example of query matrix P. Features one and four have been selected in part one. It means that the analyst is looking for linear relationships between these two features in data. Similarly features two, three and five have been selected in part two, and features one, five and six in part three.

To incorporate the additional constraint described above, the NMF minimization problem (2.7) has been extended to automatically impose the structure of the mask P to the basis matrix W:

$$\min_{W \geq 0, H \geq 0} \frac{1}{2} \|X - (P \odot W)H\|_F^2 + \frac{1}{2}\lambda \left\| P \odot \tilde{W} \right\|_F^2, \tag{5.1}$$

where $\tilde{w}_{ij} = \exp(-w_{ij})$ and $P \in \{0,1\}^{n \times k}$ and $\lambda \geq 0$ is a regularization parameter.

The objective function in (5.1) is composed by two terms: the first one represents a weighted modification of the classical NMF problem where the mask matrix P is used to fix the structure the basis matrix W has to possess. The second term is a penalty term used to enhance the elements w_{ij} corresponding to elements $p_{ij} = 1$. For this purpose the exponential function has been chosen: when the value of an entry w_{ij} of W is small it is increased by the penalty term, when it is high the penalty tends to zero. The choice of the exponential func-

tion allow us to prevent that zero values correspond to features that we want to include in the parts. The regularization parameter $\lambda \geq 0$ is used to balance the influence of the two terms.

5.1.2 Updating Rules

The objective function (5.1) automatically imposes the structure of the mask P in the factor matrix W, minimizing the non-relevant elements in W and maximizing (when they are actually present) the relevant elements in it. It should be observed, however, that the objective function (5.1) is not convex in both variables W and H. So, it is thus unrealistic to find the global minima for it. However, an iterative updating algorithm to obtain the local optima of (5.1) can be derived.

First the minimization of the objective function (5.2) is discussed:

$$O = \frac{1}{2} \|X - (P \odot W)H\|_F^2 + \frac{1}{2}\lambda \left\| P \odot \tilde{W} \right\|_F^2, \tag{5.2}$$

with $\tilde{w}_{ij} = exp(-w_{ij})$ and $P \in \{0,1\}^{n \times k}$

It can can be rewritten as:

$$\begin{aligned} O = \frac{1}{2} trace \left((X - (P \odot W)\, H)^\top (X - (P \odot W)\, H) \right) + \\ + \frac{1}{2}\lambda trace \left(\left(P \odot \tilde{W} \right)^\top \left(P \odot \tilde{W} \right) \right). \end{aligned} \tag{5.3}$$

Particularly, denoted by $\Psi = [\psi_{ij}]$ and $\Phi = [\phi_{ij}]$ the Lagrangian multiplier for the constraints $w_{ij} \geq 0$ and $h_{ij} \geq 0$, the Lagrangian function associated to the minimization problem (5.1) is given by:

$$\begin{aligned} \mathcal{L} = \frac{1}{2} trace \left((X - (P \odot W)\, H)^\top (X - (P \odot W)\, H) \right) \\ + \lambda trace \left(\left(P \odot \tilde{W} \right)^\top \left(P \odot \tilde{W} \right) \right) + trace\left(\Psi W\right) + trace\left(\Phi H\right), \end{aligned} \tag{5.4}$$

The partial derivatives of $\mathcal{L}$ with respect to W and H are:

$$\frac{\partial \mathcal{L}}{\partial W} = (P \odot W) HH^\top - P \odot \left(XH^\top\right) + \frac{1}{2}\lambda (-2) P \odot \tilde{W} + \Psi; \tag{5.5}$$

$$\frac{\partial \mathcal{L}}{\partial H} = (P \odot W)^\top (P \odot W) H - (P \odot W)^\top X + \Phi. \tag{5.6}$$

Imposing the Karush-Kuhn-Tucker conditions for the optimality:

$$\frac{\partial \mathcal{L}}{\partial W} = 0; \qquad \frac{\partial \mathcal{L}}{\partial H} = 0; \tag{5.7}$$

$$W \odot \Psi = 0; \qquad H \odot \Phi = 0; \tag{5.8}$$

$$W \geq 0; \qquad H \geq 0, \tag{5.9}$$

the following equations, for w_{ij} and h_{ij}, are given:

$$\begin{aligned} &- \left((P \odot W) HH^\top\right)_{ij} w_{ij} + \\ &+ \left(P \odot \left(XH^\top\right)\right)_{ij} w_{ij} + \lambda \left(P \odot \exp\left(-W\right)\right)_{ij} w_{ij} = 0, \end{aligned} \tag{5.10}$$

$$\left(- (P \odot W)^\top (P \odot W) H\right)_{ij} h_{ij} + \left((P \odot W)^\top X\right)_{ij} h_{ij} = 0. \tag{5.11}$$

These equations lead to the following updating rules:

$$w_{ij} \leftarrow w_{ij} \frac{\left[P \odot \left(XH^\top\right)\right]_{ij} + \lambda \left(P \odot \tilde{W}\right)_{ij}}{\left[(P \odot W) \left(HH^\top\right)\right]_{ij} + \epsilon}; \tag{5.12}$$

$$h_{ij} \leftarrow h_{ij} \frac{\left[(P \odot W)^\top X\right]_{ij}}{\left[(P \odot W)^\top (P \odot W) H\right]_{ij} + \epsilon}, \tag{5.13}$$

where the constant $\epsilon = 10^{-12}$ has been introducted to prevent division by zero. We will refer to equations 5.12 and 5.13 as Masked NMF (MNMF).

5.1.3 Normalization

The data matrix X has to be normalized so as to lay in the unit sphere (i.e. $\|x_i\|_2 = 1$ for $i = 1, \ldots, m$). This requirement has been adopted because NMF

works in a vectorial space, where data are vectors and not points. Normalization eliminates information related to the lenght of the vectors, but preserving directions. After a MNMF run, columns of W are normalized in L_2 together with the matrix H in order to preserve the factorization results. This is accomplished by:

$$w_{ij} \leftarrow \frac{w_{ij}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2}}, \tag{5.14}$$

$$h_{vt} \leftarrow h_{vt} \sqrt{\sum_{i=1}^{n} w_{iv}^2}, \tag{5.15}$$

for each column $j = 1 \ldots k$ of W, and each column $t = 1 \ldots m$ of H.

5.1.4 Representativeness

A *representativeness* measure has been defined to estimate the effectiveness of the mask matrix P in representing structure of data. It quantifies the matching between structures holding in data and the structure imposed to the basis vectors in W by the mask matrix P.

For each sample $\mathbf{x_i}$ the representativeness measure Rep_s has been evaluated as in (5.16). It measures the effectiveness of the parts in W in reconstucting each sample and it is composed by two factors. The first is the inverse of the reconstruction error of each sample (in term of mean squared error, MSE). In fact the lower is the reconstruction error, the better is the reconstuction of the samples using parts in W and coefficients in H. The second factor of the function indicates the contribution of the parts in the reconstruction. This factor is used to enhance the *weights* of samples reconstucted by parts the analyst is looking for. Formally:

$$Rep_s\left(\mathbf{x_i}, W, \mathbf{h_i}\right) = \frac{1}{\|\mathbf{x_i} - W\mathbf{h_i}\|_F^2} \sum_{j=1}^{k} h_{ji}, \tag{5.16}$$

where $\mathbf{x_i}$ e $\mathbf{h_i}$ are respectively the i-th columns of X and H, h_{ji} is the element of H on the j-th row and i-th column.

An example of the measure Rep_s for all sample in a dataset is shown in

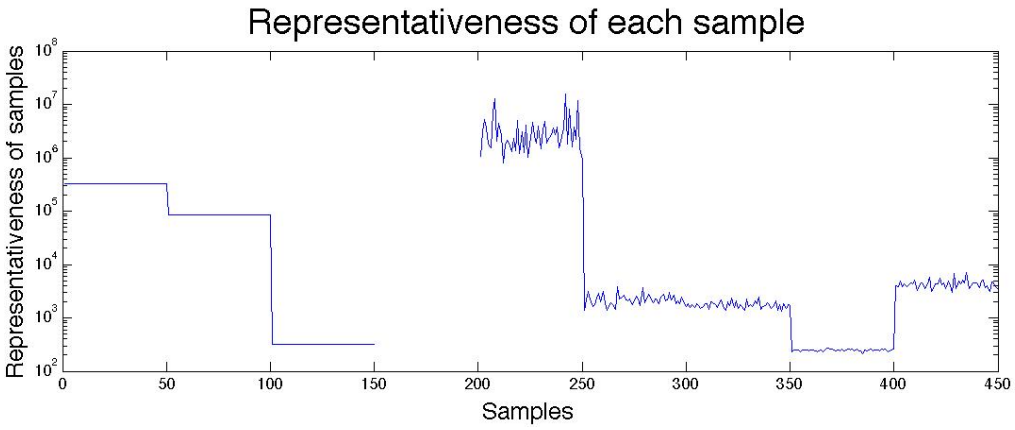


Figure 5.2: Example of the representativeness measure Rep_s of each sample in a dataset.

Figure 5.2. In the example the samples in $[151, 200]$ have representativeness values equal to zero, because the parts in the mask matrix P are not able to describe them[1].

Using a thereshold t_R on the values of representativeness, it is possible to select the set G_t of *well represented* samples:

$$G_t = \{\mathbf{x_i} \in X / Rep_s(\mathbf{x_i}, W, \mathbf{h_i}) \geq t_R\}. \tag{5.17}$$

A global measure of representativeness for the factorization results is than given by:

$$Rep\,(X, W, H) = \frac{|G_t|}{m}, \tag{5.18}$$

where $|G|$ is the cardinality of the set G_t and m is the number of samples.

$Rep\,(X, W, H)$ has values in the interval $[0, 1]$. The lowest value 0 means that parts in W are not representative for the dataset X at the precision level t_R. On the contrary 1 means that parts in W are completely adeguate to describe data in X. Values in the interval indicate the percentage of samples well represented. The mask matrix P is not directly used in the representativeness measure described in (5.18), but it is implicitly involved, as it determines the structure of the parts

[1] A deep investigation on the MNMF results is given in the section 5.2.

in W.

The choice of thereshold t_R is related to the precision of the reconstruction the analyst is looking for. Samples with a value of Rep_s higher than thereshold t_R belong to G_t (5.17):

$$\frac{1}{\|\mathbf{x_i} - W\mathbf{h_i}\|_F^2} \sum_{j=1}^{k} h_{ji} \geq t. \tag{5.19}$$

The reconstruction error of samples is inversely proportional to thereshold t_R:

$$\|\mathbf{x_i} - W\mathbf{h_i}\|_F^2 \leq \frac{\sum_{j=1}^{k} h_{ji}}{t_R}. \tag{5.20}$$

Since columns in H are normalized in L_2, each entry h_{ij} has values in $[0, 1]$. The sum of the elements in each columns could have maximum value equal to the number k of parts:

$$\|\mathbf{x_i} - W\mathbf{h_i}\|_F^2 \leq \frac{k}{t_R}. \tag{5.21}$$

5.1.5 Conformity

The representativeness measure is not sufficient to evaluate the effectiveness of a MNMF result. In fact, it does verity that the structure of the vector bases $\mathbf{w_i}$ matches with the structure of the parts $\mathbf{p_i}$. Indeed when the analyst is looking for parts that does not *capture* the structure in data, values in $\mathbf{w_i}$ could be close to zero. In this case MNMF uses parts that are different from those the analyst was looking for. It is therefore necessary to quantify the similarity between the structure in P and the structure obtained in the basis matrix W after MNMF.

The conformity measure of each column of W and P is defined as the cosine of the two corresponding column vectors:

$$Conf_p\left(\mathbf{w_i}, \mathbf{p_i}\right) = \frac{\mathbf{w_i}^{\mathrm{T}}\mathbf{p_i}}{\|\mathbf{w_i}\|_2 \|\mathbf{p_i}\|_2}. \tag{5.22}$$

It evaluates the angle between the basis vector $\mathbf{w_i}$ and the corresponding mask vector $\mathbf{p_i}$. When $\mathbf{w_i} = \mathbf{p_i}$ the vectors are overlapped and this angle is 0 (the cosine

is 1). On the other hand the greater is this angle, the bigger is the difference between the vectors $\mathbf{w_i}$ and $\mathbf{p_i}$. Since the zero-elements in the basis vectors $\mathbf{w_i}$ are imposed by $\mathbf{p_i}$, these differences derive from elements in $\mathbf{w_i}$ that have low values for some selected features. From a semantically point of view, this means that these features do not contribute to the part. Thus conformity has values in $[0, 1]$. When structure in $\mathbf{w_i}$ does not match with structure in $\mathbf{p_i}$, conformity value is zero, whilst when the two structures overlap it is one.

It is possible to use a threshold t_C suggesting when the two structures coincide.

A global measure of conformity is given by:

$$Conf\,(W, P) = \min_{1 \leq i \leq k} \left(Conf_p\,(\mathbf{w_i}, \mathbf{p_i})\right). \tag{5.23}$$

5.1.6 Query based MNMF

The query based MNMF algorithm has been designed in order to allow the analyst to specify what parts she is interested to discover. The proposed approach extracts the subset of data that are actually represented by the parts, discarding the data in the matrix X that do not find a neat representation by the parts.

As it has been shown in subsection 2.2.4, when a NMF of a given data matrix X is computed, each sample is approximated in a low-rank subspace (of k dimensionality) by Equation 2.5. Particularly, the elements of each columns of the encoding matrix H codify the information needed to identify the factors (columns of W) used to reconstruct each sample of X in the low-rank subspace. Therefore, the elements in a column of H identify the importance of each basis vector in approximating the data sample: if a coefficient is very small, then the corresponding basis vector is useless in approximating the sample; as a consequence, the data sample does not contain the part represented by this basis vector. Information stored in the matrix H can be used therefore for Intelligent Data Analysis.

Algorithm

Algorithm 3 formally describes the proposed approach to analyse data through MNMF. Particulalrly, the steps of the proposed approach are justified and de-

scribed in the following.

Given a data matrix X, the query based MNMF algorithm first executes MNMF with a mask matrix P where the data analyst has previously specified the relationships she is interested to discover. MNMF is an iterative updating algorithm based on the classical multiplicative algorithm [Lee and Seung, 2001]. It alternatively updates the matrices W and H according to the rules in (5.12) and (5.13) (lines 3 and 4 of algorithm 3) while stopping criterion is not satisfied (line 2).

After MNMF optimization, the resulting matrices (W, H) are normalized in L_2 (lines 6 and 7).

Then the *conformity* of the basis matrix W to the mask matrix P, $Conf_p\,(W, H)$ (5.23) is verified (line 9). If the structure of W does not correspond to the parts specified by the analyst, the process is stopped. Low values of conformity means that parts, the analyst is looking for are not present in data. Therefore the algorithm returns the set of column indices of W that differ from the corresponding columns of P. This feedback allow the analyst to modify her query, and re-run the algorithm with a new mask matrix P.

Otherwise, when the basis matrix catches the structure in P, each sample is evaluated in term of a *representativeness* measure Rep_s (5.18) (line 10). Low values of representativity for a subset of samples means that MNMF was not able to find the parts in W in that subset, so the analysis could be restricted to the subset of data G_t (line 11) where the parts have been recognized.

The samples in the matrix X that have not been reconstructed using the parts which the analyst is looking for, are then removed from the matrix X. The remaining columns after this removal procedure form a new data matrix that is denoted by X' (line 13). This approach allows the selection of the samples in data that are actually represented by the specified parts.

At the end of the selection process, MNMF could be re-run for the subset of the selected data samples. The objective of this last step is to re-compute the values in the base and encoding matrices without taking into account data samples that are not composed by the selected parts. This provides a more precise estimation of the parts and their contribution in the data samples.

Algorithm 3 QMNMF

Require: $X \in \mathbb{R}_+^{n \times m}$ {Dataset}
Require: $P \in \{0,1\}^{n \times k}$ {Query mask}
Require: λ {Regularization parameter}
Require: $W_0 \in \mathbb{R}_+^{n \times k}$ and $H_0 \in \mathbb{R}_+^{k \times m}$ {Initial matrices W and H}
Require: $t_R > 0$ and $t_C \in [0,1]$ {Thresholds for Representativeness and Conformity measures}

1: Normalize X
2: **while** stopping criterion not satisfied **do**
3: update matrix W according to (5.12)
4: update matrix H according to (5.13)
5: **end while**
6: Normalize matrix W according to (5.14)
7: Adjust matrix H according to (5.15)
8: Evaluate $Conf\,(W,P)$ according to (5.23)
9: **if** $Conf\,(W,P) > t_c$ **then**
10: Evaluate $Rep_s\,(\mathbf{x_i}, W, \mathbf{h_i})$, $\forall i = 1,\ldots,n$, according to (5.2)
11: Evaluate G_t for thereshold t, according to (5.17)
12: Compute the column index set $J = j \,:\, x_j \in G_t$
13: Select data samples $X' = X[1:n, J]$ {all rows and columns in J}
14: **return** $W \in \mathbb{R}_+^{n \times k}$ {selected parts}
15: **return** $H \in \mathbb{R}_+^{k \times m}$ {coefficents}
16: **return** $X' \in \mathbb{R}_+^{n \times m'}$ {data subset}
17: **else**
18: Compute the column index set
$R = r \,:\, Conf_p\,(\mathbf{w_r}, \mathbf{p_r}) < t_C$ with $r = 1,\ldots,k$
19: **return** R {parts not representing structures in data}
20: **end if**

Stop-criteria

Note that since MNMF is an iterative algorithm that converges to zero, the adopted stopping criteria (line 2) is based on the difference between two following values of the objective function: the computation of updates stops when the difference is lower than a prescribed small value ε.

Formally, set $E = Obj(i) - Obj(i-1)$, where $Obj(i)$ indicates the value of the objective function at the $i-th$ iterate, then

$$stop = \begin{cases} true & \text{if } E \leq \varepsilon \\ false & \text{otherwise.} \end{cases}$$

Initialization

Being the MNMF algorithm based on the gradient descent method, it is sensitive to the starting point. As stardart choice, the matrices W and H have been initialized using two random matrices W_0 and H_0 (line 3), however different initializations can lead to better results [Boutsidis and Gallopoulos, 2008; Casalino et al., 2011, 2014], further experiments will be aimed to examine this aspect.

5.2 Experimental Results

This section illustrates the experiments that have been performed in order to show the effectiveness of the proposed method. First, a preliminary data analysis has been conducted. For this purpose, a synthetic dataset has been constructed and three mask matrices P has been adopted queries. The behaviour of Query-based MNMF algorithm has been illustrated, when an optimal mask, a correct mask and a wrong mask are used. Further experiments have been conducted on the well known Iris dataset to better highlight the semantics associated to the selected parts.

5.2.1 Synthetic dataset

A dataset $X \in \mathbb{R}^{6\times 450}$ of six features has been synthetically generated in a specific way to evaluate the ability of the proposed approach in finding parts in data.

To generate the data samples, two random variables from a Gaussian distribution with mean 50 and variance 5 have been used: $s_1 \sim \mathcal{N}(50, 5)$ and $s_2 = \alpha s_1$ (negative values have been cropped to 0). Four combinations of two out of six features have been generated, namely $(1, 2)$, $(2, 3)$, $(3, 5)$, $(4, 6)$. For each combination of features (i_1, i_2) a correponding random basis $\mathbf{c}$ has been defined, so that $c_{i_1} = s_1$, $c_{i_2} = s_2$ and $c_i = 0$ for $i \notin \{i_1, i_2\}$. A multiplicative factor $\alpha = 3$ has been used for random bases corresponding to combinations $(1, 2)$ and $(3, 5)$,

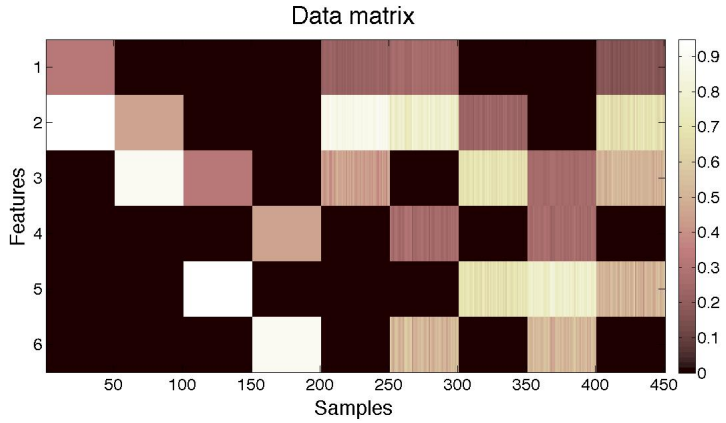


Figure 5.3: Graphical illustration of the synthetic dataset X.

whilst a value $\alpha = 2$ for $(2, 3)$ and $(4, 6)$. Finally, the dataset, has been generated, in blocks of 50 samples, each block being defined as a combination of the random bases $\mathbf{c}$. Let denote $\mathbf{c}_h$, with $h = 1, \ldots, 4$, the random bases corresponding respectively to the combinations $(1, 2)$, $(2, 3)$, $(3, 5)$, $(4, 6)$, the synthetic dataset is constructed as follows: $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, $\mathbf{c}_4$, $\mathbf{c}_1 + \mathbf{c}_2$, $\mathbf{c}_1 + \mathbf{c}_3$, $\mathbf{c}_2 + \mathbf{c}_3$, $\mathbf{c}_3 + \mathbf{c}_4$, $\mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3$.

Figure 5.3 shows a graphical representation of the data matrix. It should be observed that the boxes represent fifty sequential data generated with the same linear combination.

Optimal Mask

In the optimal case, Query-based MNMF is used to query data matrix with the mask P_{opt} that imposes on the factors matrix W the same structure occurring in the dataset (Figure 5.4). Figure 5.5 shows basis matrix W and encoding matrix H returned after applying MNMF to the data matrix. As it can be observed, the factor W possesses the same structure of P_{opt}. This means that parts that are looked for, are actually in data. This result is confirmed by the conformity values of the columns of W and P, $Conf_P\left(\mathbf{w_i}, \mathbf{p_{opt_i}}\right)$,that are 0.8959, 0.9498, 0.8969, 0.9491, for $i = 1 \ldots 4$. Empirical tests suggest that values of conformity lower than a threshold $t_C = 0.80$ denote bases with structures that do not allow a good reconstruction of the data. In this case values are close to the maximum conformity value 1.

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 5.4: Optimal mask matrix P containing all the parts in data.

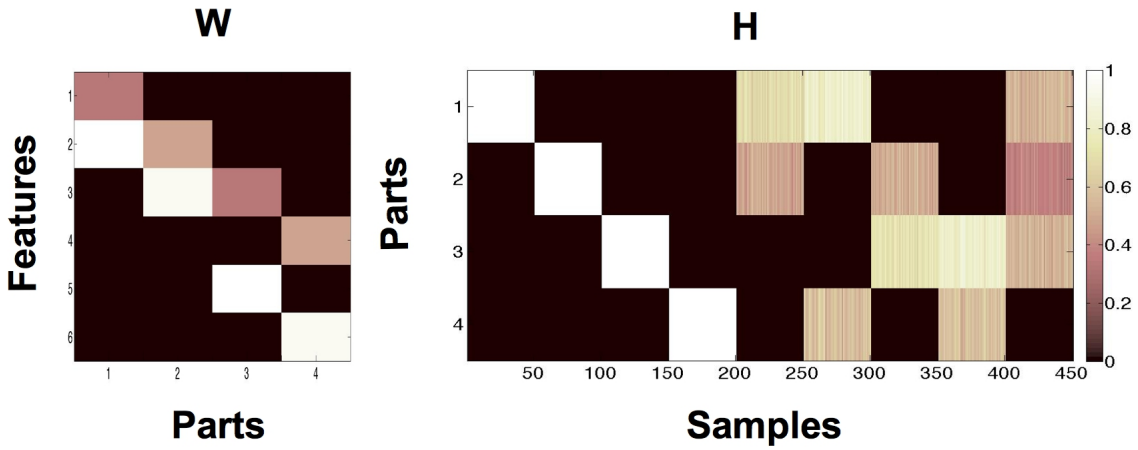


Figure 5.5: Basis matrix W and encoding matrix H obtained applying MNMF to synthetic data with optimal query mask P.

From the encoding matrix H, it could be pointed out that MNMF is able to recognize samples in dataset that were constructed using parts in P. In fact each block of samples has been reconstructed using the parts that correspond to the random bases that have been used to generate original samples.

Representativity of the mask mask P_{opt} is the maximum value 1 for $t_R = 10^7$. This means that all samples in dataset have been reconstructed with parts in P with high precision.

However this is the best scenario. In the following paragraphs the behavior of QMNMF is shown when the analyst is looking for parts that partially cover the structure in data, or, in the worst scenario, are not able to describe data.

$$P_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Figure 5.6: Mask matrix P_1 containing parts in data.

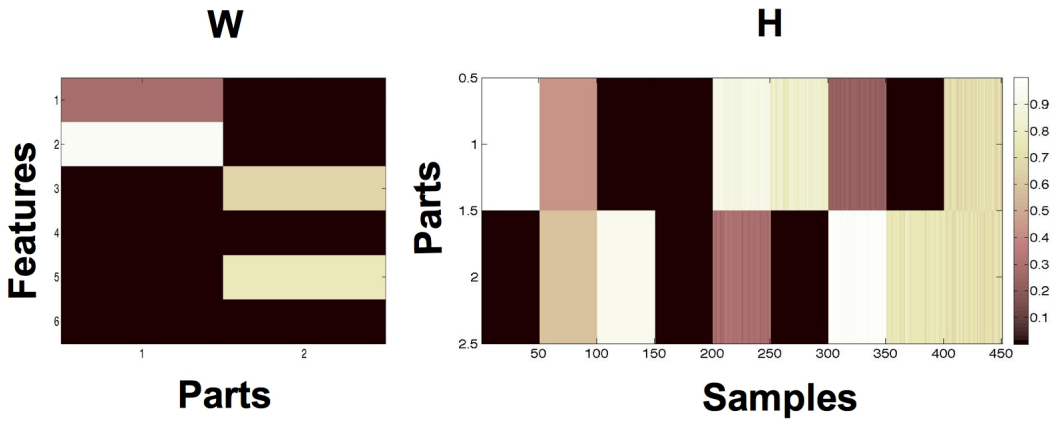


Figure 5.7: Basis matrix W and encoding matrix H obtained applying MNMF to synthetic data with query mask P_1.

Ideal case

In the ideal case, Query-based MNMF is used to query data matrix with the mask P_1 that imposes on the factors matrix W a partial structure occurring in the dataset (Figure 5.6). The mask P_1 allows to verify if the proposed QMNMF is able to recognize as relevant the examples in dataset that have been constructed using the parts specified by the query mask. The query submitted to the algorithm does not cover all the examples in the dataset, so it is expected that the procedure selects a subset of it containing only the relevant data. The following detailed analysis shows the behavior of MNMF in reconstructing samples when they contain the parts in P_1, linear relationship of these parts, and when there are not parts in the mask adequate to describe them.

Figure 5.7 shows the basis matrix W and the encoding matrix H returned after applying MNMF the matrix. As it can be observed, the factor W possesses the same of structure of P_1. Moreover the values of the conformity measure of

the two bases $\mathbf{w_1}$ and $\mathbf{w_2}$ are respectively 0.87 and 0.9977. These values are close to the maximum value one. This suggests that parts in P do actually catch the structure in data. From the encoding matrix H, it could be pointed out that MNMF is able to recognize samples in dataset that were constructed using parts in P_1. Samples from 1 to 50 composed by the the first two features in X (whose linear correlation is captured by $\mathbf{w}_1$) have been correctly reconstructed in H using only the first part. Similarly, samples from 101 to 150 have been reconstructed using the part $\mathbf{w}_2$ representing the relationship between the features three and five.

Query Based MNMF algorithm, after applying MNMF, selects samples that have values of representativeness Rep_s higher than a thereshold t_R. Figure 5.8 shows values of Rep_s and the samples that have been selected using a threshold $t_R = 10^3$: blocks of samples $[1, 50]$, $[101, 150]$, $[201, 250]$, $[301, 350]$, $[401, 450]$. Of course the choice of the threshold t_R is problem dependent. Moreover, different thresholds t_R lead to different solutions. The choice of a suitable threshold is duty of the analyst. The value of the reconstuction error she expects could suggest the value of t_R.

MNMF can recognize parts that are composed linearly to describe data. This is the case of the samples from 401 to 450 that have been generated adding data composed by the first two features and data composed by the third and fifth. These samples have been reconstructed in the matrix H using both the bases $\mathbf{w}_1$ and $\mathbf{w}_2$ capturing the linear correlation between respectively first and second features, and third and fifth features. Samples from 201 to 250 have been generated adding data composed by the first two features and data composed by the second and third. Since these features are partially captured from parts in P_1, the algorithm returns them. The same behaviour is observed in samples from 301 to 350.

When the algorithm does not find parts that are able to correctly reconstruct the samples in data it returns values of representativeness Rep_s lower than threshold t_R. This beavior suggests to the analyst that the parts she is looking for in data are not enough to describe them.

An extreme case are the samples from 151 to 200 that have been completely constructed using a part that is not in P_1, the algorithm returns no parts for this

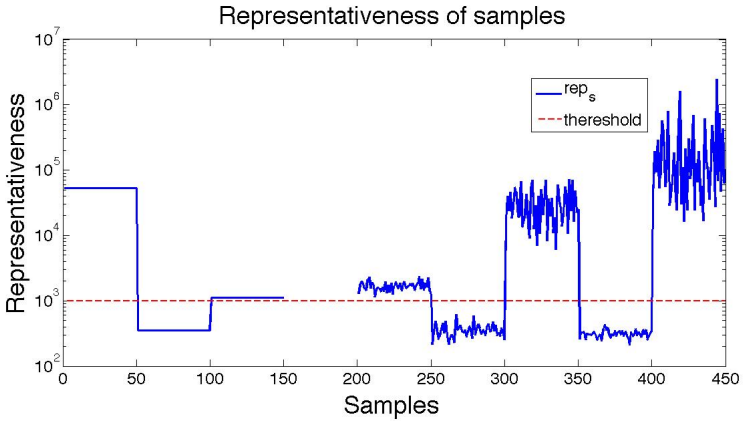


Figure 5.8: Values of Rep_s for samples in synthetic dataset.

$$P_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Figure 5.9: Mask matrix P_2 not containing parts in data.

samples. Hence, the proposed Query Based MNMF algorithm suggests which parts correctly reconstruct data.

The value of total representativity quantifies the matching between the structure in P and the structure in data. In this case it is 0.56, that means 50% of samples uses part in P_1.

The non-ideal case

In the non-ideal case, Query-based MNMF is used to query data matrix with the mask P_2 that does not represent the structure hidden in data (i.e., both columns in P_2 represent parts that are not present in data (Figure 5.9). This mask allows to verify the behaviour of the proposed Query-basd MNMF when the analyst is looking for parts that do not correctly decribe data.

The basis matrix W returned by MNMF with matrix mask P_2 has columns composed by one of the two values very close to the maximum value 1, and the second one close to 0 (Figure 5.10). This behaviour suggests that the algorithm has not been able to impose the structure to the basis matrix, which, instead,

$$W_2 = \begin{pmatrix} 0.13 & 0 \\ 0 & 0.996 \\ 0 & 0 \\ 0 & 0.08 \\ 0 & 0 \\ 0.99 & 0 \end{pmatrix}$$

Figure 5.10: Basis matrix W obtained applying MNMF to synthetic data with mask matrix P_2.

uses canonical bases. The conformity measure confirms this observation. In fact, conformity values $Conf_P(\mathbf{w_i}, \mathbf{p_i})$ of basis and mask matrix columns are 0.7939 and 0.7664.

When the mask matrix does not contain parts in data, QMNMF algorithm stops. The analyst should then modify her query according to the received feedback.

5.2.2 Iris Dataset

In this section, the behaviour of the proposed approach has been illustrated when the well known Iris dataset is adopted [Bache and Lichman, 2013]. The dataset is composed by 150 samples grouped in three different classes: Iris-Setosa, Iris-Veriscolor, Iris-Virginica (Figure 5.12). Each sample is a four dimensional vector describing: sepal length, sepal width, petal length, petal width. Figure 5.11[1] shows data represented in subspaces generated by couples of features.

Two experiments have been conducted in order to highlight the use of a specific mask to select features and extract samples which are described by these parts. Particularly, the aim is to discover if there exists any linear correlation between the features in the data samples. The use of a real dataset better highlights the semantic associated to the parts.

Relationship between lengths and widths of iris flowers

The goal of the first experiment is to verify the existence of relationships between sepal and petal lengthts and sepal and petal widths of Iris Dataset. Features involved in relationships that are looked for have been selected and a mask matrix

[1] "Anderson's Iris data set" by Indon - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons

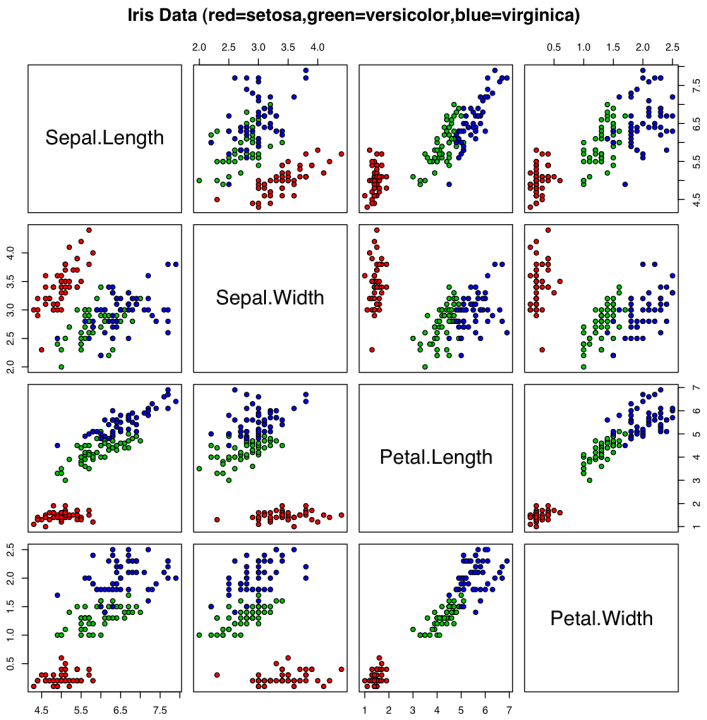


Figure 5.11: The scatterplot of Iris flower data set, collected by Edgar Anderson and popularized in the Machine learning community by Ronald Fisher.

Figure 5.12: Iris Setosa, Iris Versicolor, Iris Virginica.

P_3 (Figure 5.13) has been generated according to this choice. Particularly, the first part specifies a relationships between sepal and petal lengthts (represented by the first and third features of dataset), whilst the second part specifies relationships between sepal and petal widths (represented by the second and fourth features of dataset).

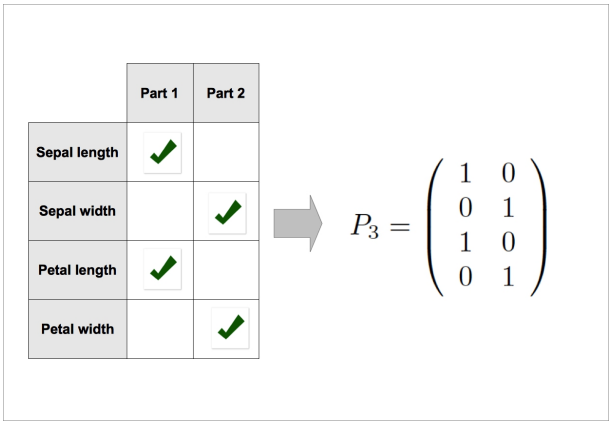


Figure 5.13: Query mask P_3 used to verify relationships between sepal and petal lengthts and sepal and petal widths of Iris Dataset.

MNMF has been executed on Iris dataset with mask P_3 and parameter $\lambda = 0.5$. The basis matrix W_3 illustrated in Figure 5.14 preserves the structure imposed by the query masks, moreover the parts are represented by significative values. This result suggests to the analyst that parts she is looking for are actually present in data, i.e., there is correlation between the selected features in data. The conformity measure confirms this result. In fact the conformity of the first columns of W_3 and P_3 is $Conf\left((\mathbf{w_3})_1, (\mathbf{p_3})_1\right) = 0.97$ that is close to the maximum. On the other hand the conformity of the second columns of W_3 and P_3 is $Conf\left((\mathbf{w_3})_2, (\mathbf{p_3})_2\right) = 0.88$, which is smaller than the previous one, but still significative.

After establishing the presence of relationships in data, the aim of the algo-

$$W_3 = \begin{pmatrix} 0.85 & 0 \\ 0 & 0.96 \\ 0.53 & 0 \\ 0 & 0.29 \end{pmatrix}$$

Figure 5.14: Basis matrix W_3 obtained with MNMF, masks P_3 and $\lambda = 0.5$.

rithm is to find the subset of samples in which these relationships hold. Query-based MNMF algorithm uses the representativeness measure with a defined threshold $t_R = 10^4$ for this purpose. Figure 5.15 shows representativeness measure $Rep_s(\mathbf{x_i}, W, \mathbf{h_i})$ of samples in Iris dataset computed with MNMF, query mask P_3 and $\lambda = 0.5$.

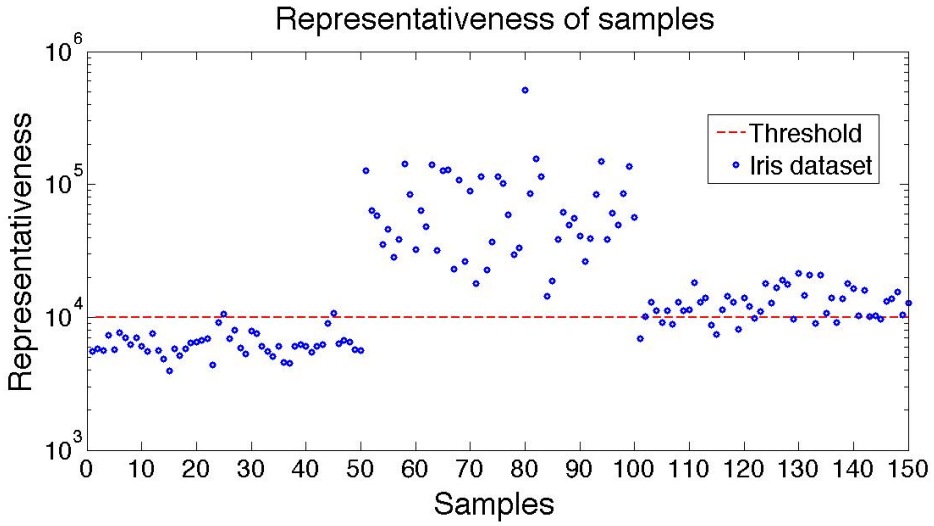


Figure 5.15: Representativeness $Rep_s(\mathbf{x_i}, W, \mathbf{h_i})$ of samples in Iris dataset computed with MNMF, query mask P_3 and $\lambda = 0.5$.

It could be pointed out that the three classes are well separated (the first fifty samples belong to class Setosa, from 51 to 100 to Versicolor and the last fifty to Virginica). This means that the two parts contribute with different values to reconstruct data. Using a threshold $t_R = 10^4$, the algorithm selects samples belonging to classes versicolor and to virginica. Using a more selective threshold, only samples belonging to the class versicolor are selected. The choice of the threshold value influences the results returned from the algorithm. The threshold value needs to be provided by the analyst who is able to exploit his specific knowledge of the problem under consideration to intelligently analyze data. After selection MNMF is re-run on the modified dataset composed by samples with representativity values higher than this threshold t_R. The reconstruction error, evaluated in term of Mean Squared Error (Equation 5.2.2) obtained removing samples that have not been well reconstructed is 0.0023, which is smaller than

the error obtained with the entire dataset (0.0176): the approximation of the subset of data is better than that of the entire dataset. This result is explained by the fact that parts in P_3 are more suitable to describe data in the subset instead of those in the whole dataset. Raising the threshold t_R and reducing dataset to samples belonging to class versicolor, the obtained reconstructing error is $6.51 * 10^{-4}$, much smaller than the previous ones.

$$MSE = \frac{1}{2}\frac{\|X - WH\|_F^2}{m}. \tag{5.24}$$

These considerations help the analyst to conclude that a relationship between lengths and widths of Iris flowers holds for the samples belonging to the class Versicolor and some samples of Virginica.

Relationship between sepals and petals of iris flowers

The goal of the second experiment is to verify the existence of relationships between sepal and petal dimensions of iris flowers. MNMF has been executed on Iris dataset with mask P_4 illustrated in Figure 5.16 with parameter $\lambda = 0.5$.

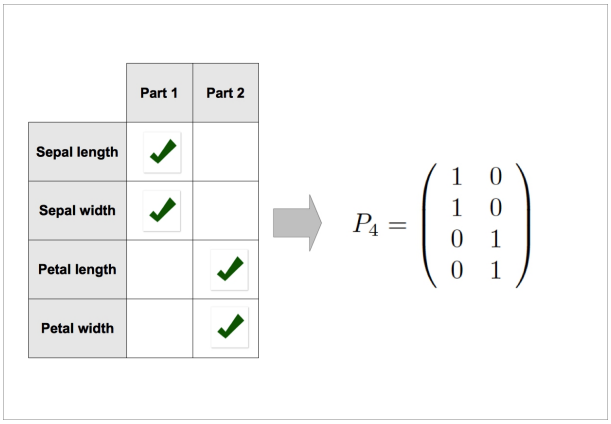


Figure 5.16: Query mask P_4 used to verify relationships between sepal and petal measures of Iris Dataset.

In this example, as well as in the first one, the basis matrix W_4 illustrated in Figure 5.17 preserves the structure imposed by the query mask, and the parts

$$W_4 = \begin{pmatrix} 0.88 & 0 \\ 0.48 & 0 \\ 0 & 0.95 \\ 0 & 0.31 \end{pmatrix}$$

Figure 5.17: Basis matrix W_4 obtained with MNMF, masks P_4 and $\lambda = 0.5$.

contain significative values. Thus a correlation between the selected features exists for a subset of samples. Conformity measures confirms this result. In fact they are close to the maximum value one: $Conf\left((\mathbf{w_4})_1, (\mathbf{p_4})_1\right) = 0.96$ and $Conf\left((\mathbf{w_4})_2, (\mathbf{p_4})_2\right) = 0.89$.

The subset of samples in which this relationships hold is suggested by QM-NMF algorithm according to representativeness values of samples. Figure 5.19 shows the representativeness measure $Rep_s\left(\mathbf{x_i}, W, \mathbf{h_i}\right)$ obtained with MNMF, mask matrix P_4 and $\lambda = 0.5$. Using a threshold $t_R = 5 \cdot 10^5$ most of samples belonging to class Setosa have been discarded. On the contrary, most of samples belonging to classes Virginica and Versicolor are preserved. This value is smaller than the recostraction error on the entire dataset (which is 0.0035).

Further confirmation of this result could be obtained observing the encoding matrix H_4 obtained with MNMF mask P_4 (Figure 5.18). Observing the graph one can figures out that samples from 1 to 50 (belonging to the classe Setosa) can be represented using only the first bases $\mathbf{w_1}$. In fact, the elements in $\mathbf{w_1}$ assume almost the maximum value, that is 1, while the elements in $\mathbf{w_2}$ assume values close to zero. This explain low values of representativeness. Moreover this means that in this subset of data there is a linear relationship between the sepal features of the Iris, but not between the petal features. On the contrary samples from 51 to 150 (belonging to Versicolor and Virginica), have been reconstructed using both bases $\mathbf{w_1}$ and $\mathbf{w_2}$.

After identifying the subset of samples, original dataset is modified and MNMF is executed on it. The reconstruction error obtained removing samples that have not been well reconstructed is 0.0012 smaller than that obtained with the entire dataset 0.0035.

Even in this case, results confirm that there are linear correlations between sepal and petal features holding for samples belonging to Versicolor and Virginica.

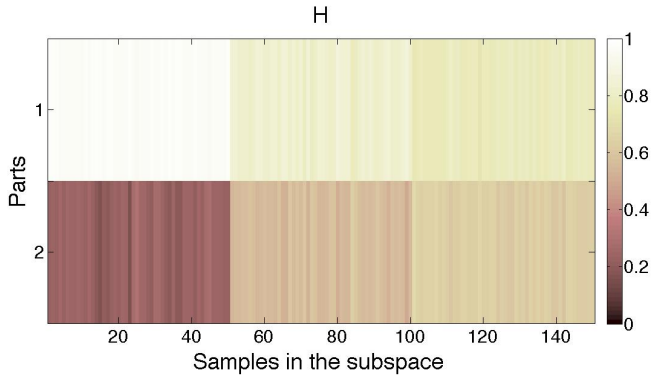


Figure 5.18: Encoding matrix H_4 obtained with MNMF mask P_4, $\lambda = 0.5$.

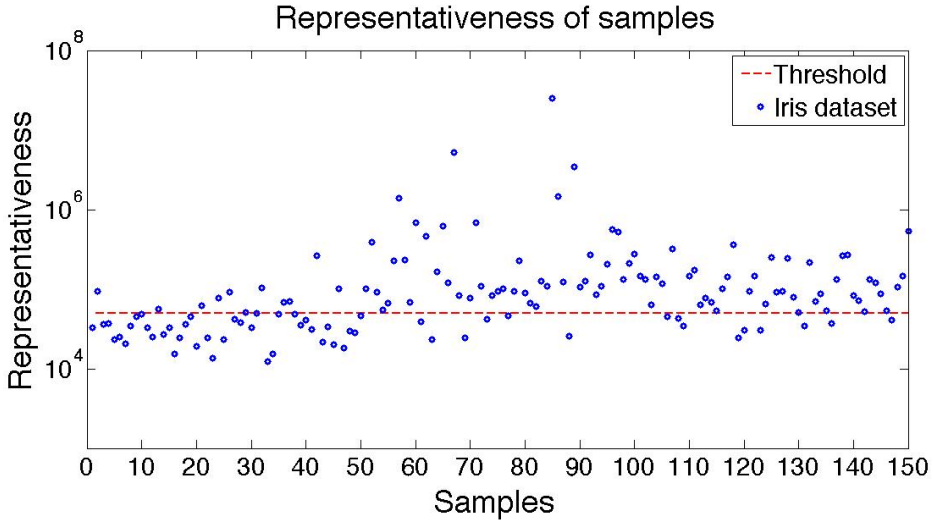


Figure 5.19: Representativeness $Rep_s(\mathbf{x_i}, W, \mathbf{h_i})$ of samples in Iris dataset computed with MNMF, query mask P_4 and $\lambda = 0.5$.

Observations

It is worth to highlight that the threshold t_R is correlated to the precision of the reconstruction of each sample. In fact representativeness of each sample is defined by Equation 5.16 in terms of inverse of reconstuction error of each sample and sum of coefficients in encoding matrix H. Thus from examples one and two it could be pointed out that results obtained in the second experiments have higher precision than those obtined in the first. From a semantic point of view, this means that relationships between sepal and petal measures are more significative than those between lengths and widths.

Data visualization

As it has been widely discussed in subsection 2.2.4, NMF is a dimensionality reduction technique, where column vectors of W are basis defining the new subspace, and coefficients in encoding matrix H specify coordinates of samples in the new subspace.

When MNMF is adopted, each basis vector $\mathbf{w_i}$ has a semantic meaning according to features that have been selected in it. Moreover when a factorization rank k equals to two or three is chosen, it is possible to visualize data in the subspace. Further analysis could be conducted in this space.

Figure 5.20 illustrates data samples represented in the subspace defined by parts $(\mathbf{w_3})_1$ and $(\mathbf{w_3})_2$ corresponding to the semantic concepts lengths and widths respectively. The graph shows that samples are grouped in two well separated groups. The first group is composed by samples belonging to class Setosa whilst the second group is composed from samples belonging to classes Versicolor and Virginica, which are mixed. Moreover, samples in the first group have lenght values varying in $[0.75, 0.87]$, and widths values varying in $[0.4, 0.6]$. Instead samples in the second group have length values varying in $[0.87, 0.95]$, and widths values varying in $[0.3, 0.5]$ (this values are normalized in L_2). Furthermore the position of points in the space suggests that the first dimension is more important in defining samples than the second one, in fact samples are more close to the first axes.

Similar considerations can be derived observing Figure 5.21, that illustrates

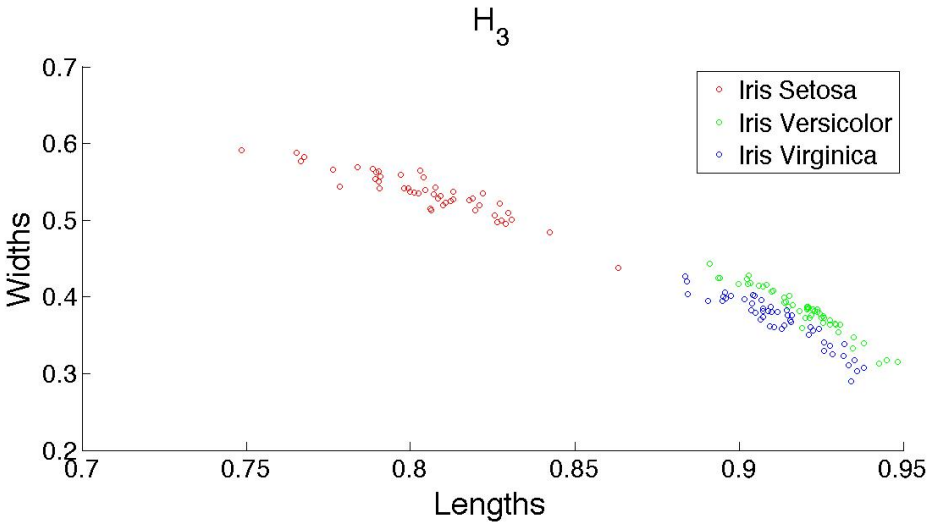


Figure 5.20: Data samples represented in the subspace defined by parts $(\mathbf{w_3})_1$ and $(\mathbf{w_3})_2$ corresponding to the semantic concepts lengths and widths respectively.

data samples represented in the subspace defined by parts $(\mathbf{w_4})_1$ and $(\mathbf{w_4})_2$ corresponding to the semantic concepts sepal and petal dimensions respectively. The class Setosa is well separed from the others. The sepal distances have more influence on data than the petal ones, indeed data are more close to the first axis. Samples belonging to the class Setosa have the biggest sepals ($[0.95, 1]$) and the smallest petals ($[0.2, 0.4]$). The classes Versicolor and Virginica are not well separed, however the samples belonging to the two classes present differences in the dimensions of the sepals and the petals. The class Versicolor has samples with sepals length in $[0.8, 0.85]$, and petals in $[0.4, 0.6]$, whilst the Virginica's sepals vary in $[0.75, 0.8]$, and petals in $[0.6, 0.7]$.

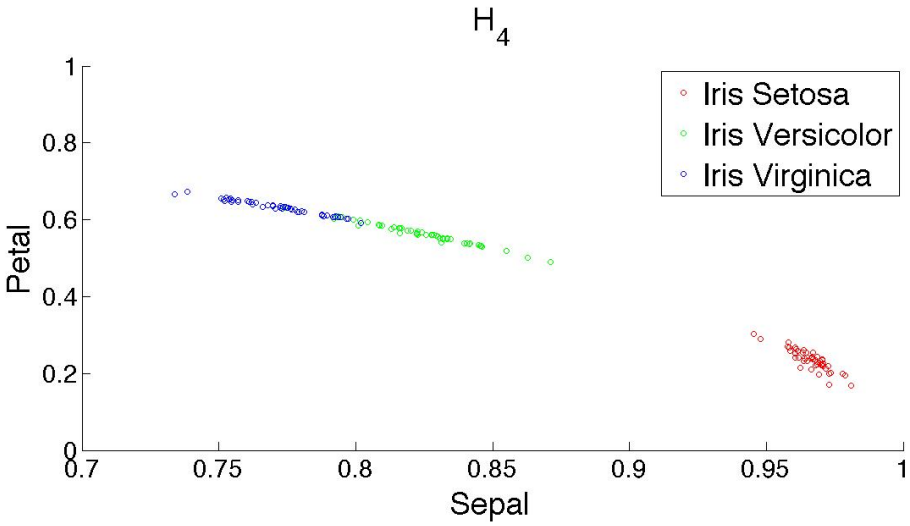


Figure 5.21: Data samples represented in the subspace defined by parts $(\mathbf{w_4})_1$ and $(\mathbf{w_4})_2$ corresponding to the semantic concepts sepal and petal dimensions respectively.

Chapter 6

Priors for Nonnegative Matrix Underapproximation of Hyperspectral Images

The third proposal concerns a modification of a constrained non-negative matrix factorization algorithm that has been used to analyze hyperspectral images (HSI). Basic information on hyperspectral images, together with the application of factorization processes to analyze them will be detailed in sections 6.1.1 and 6.1.2, respectively. Than the proposed method will be explained together with its mathematical formulation. Finally, section 6.2 will show the effectiveness of the proposed method in analyzing four real dataset.

6.1 Proposed method

Hyperspectral images represent the same scene at different wavelengths. They are widely used in data analysis processes detecting constituent materials represented in the images. For this purpose NMF has been shown to be a useful analysis tool. NMU is a recent non-negative matrix factorization algorithm that has been effectively used to analyze hyperspectral images. In this thesis a modification of the NMU algorithm has been proposed in order to incorporate prior information in the factorization process. In particular spatial information of the pixels in the

images and a sparsity constraint on the abundance matrix[1] have been added. This constrained version of NMU allows to extract materials that are represented in the HSI in a more efficient way than the classical NMU, and its previous constrained modifications.

6.1.1 Hyperspectral Images

A hyperspectral image (HSI) is a three dimensional data cube providing the electromagnetic reflectance of a scene at varying wavelengths, measured by hyperspectral remote sensors [Gillis et al., 2012]. Reflectance is the percentage of the light hitting a material that is then reflected by that material (as opposed to being absorbed or transmitted) [Shippert, 2003]. Reflectance varies with wavelength for most materials because energy at certain wavelengths is scattered or absorbed to different degrees [Smith, 2006]. Some materials will reflect the light at certain wavelengths, while other materials will absorb it at the same wavelengths.

This property of hyperspectral images is used to uniquely identify constitutive materials in a scene and classify pixels according to materials they contain. Figure 6.2 shows an hyperspectral datacube of Urban dataset, acquired with the imaging spectrometer Hymap©, and the spectral signatures (plots of reflectance curves: reflectance versus wavelength) for vegetation and soil.

Hyperspectral datacubes can be represented by two dimensional pixel-wavelength matrices $M = [\mathbf{m_1}, \mathbf{m_2}, \ldots, \mathbf{m_m}] \in \mathbb{R}_+^{n \times m}$. Columns $\mathbf{m_i} \in \mathbb{R}_+^n$ are original images that have been converted into n-dimensional column vectors (stacking the columns of the image matrix into a single vector). Rows $\mathbf{m_j} \in \mathbb{R}_+^m$ are the spectral signatures of the pixels (see Figure 6.2). Each entry m_{ij} represents the reflectance of the i-th pixel at the j-th wavelength.

6.1.2 Non-Negative Matrix Factorization for Hyperspectral Unmixing

The spectral signature of each pixel results from the additive combination of the non-negative spectral signatures of its constitutive materials [Gillis et al.,

[1] In hyperspectral imaging abundances are the relative contributions of each material to the to each pixel.

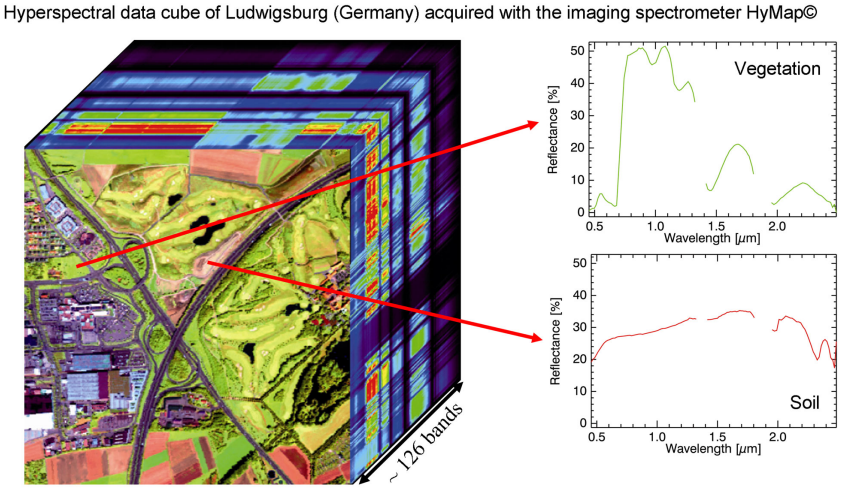


Figure 6.1: Example of Hyperspectral Image.

2012]. Due to its part-based representation and non-negativity constraint, NMF is a suitable tool to analyze hyperspectal images. It is used to identify spectral signatures of constitutive materials, and classify pixels according to materials they contain (end-members).

More precisely, given an hyperspectral datacube represented by a two dimensional matrix $M \in \mathbb{R}_+^{n\times m}$, NMF approximates it with the product of two factor matrices $U \in \mathbb{R}_+^{n\times k}$ and $V \in \mathbb{R}_+^{k\times m}$ such that the spectral signature of each pixel (rows of matrix M) is approximated by the additive linear combination of the spectral signatures of the constitutive materials (rows of matrix V), weighted by coefficients u_{ij} representing the the abundance of the j-th endmember at the i-th pixel. For each pixel i, we have:

$$\mathbf{m_i} \approx \sum_{j=1}^{k} u_{ij}\mathbf{v_j}, \tag{6.1}$$

where $\mathbf{m_i}$ are the rows of the data matrix M and $\mathbf{v_j}$ are the rows of the matrix V[1].

Figure 6.2 shows NMF approximation of the Hyperspectral datacube of Ur-

[1]Note that when NMF is used to approximate the rows of the data matrix, row vectors $\mathbf{v_j}$ are the bases of the new subspace, and U is the coefficient matrix.

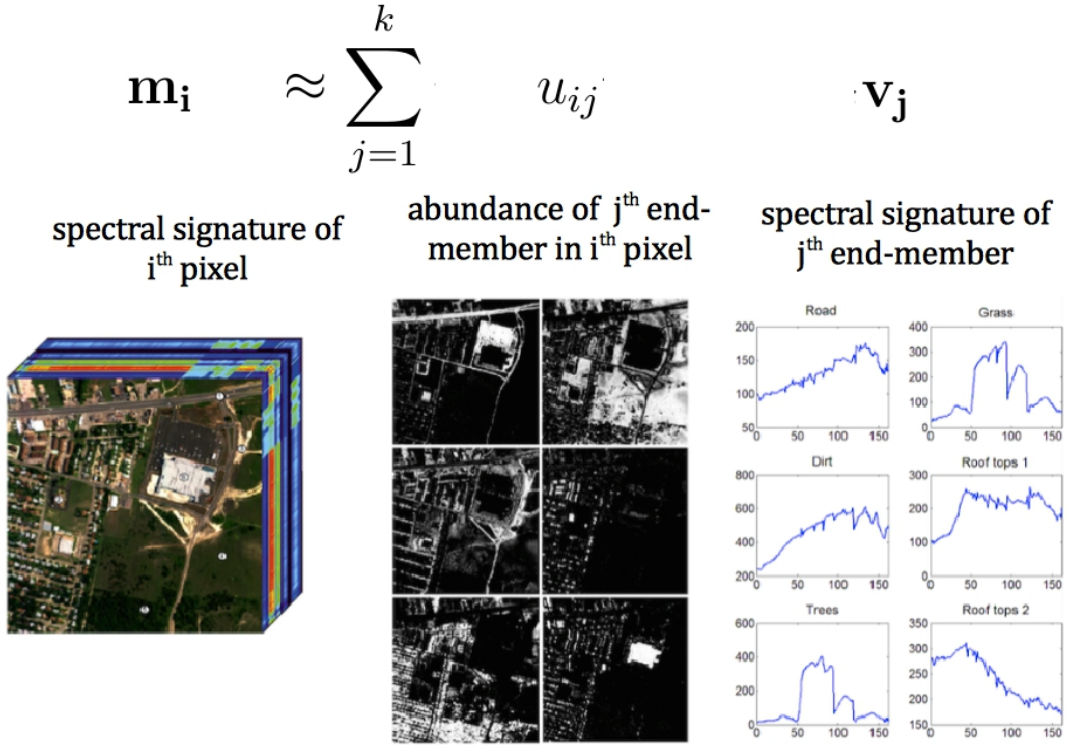


Figure 6.2: Example of non-negative matrix factorization of a hyperspectral image.

ban dataset[1]. The abundances of endmembers (columns of U) represent images in which only pixels belonging to the actual material are visible. The spectral signatures of constitutive materials are represented by row in V. In the example of Figure 6.2, mainly six constitutive materials are present: road, grass, dirt, two kind of roof tops and trees.

6.1.3 Non Negative Matrix Underapproximation

Experimental observations have shown that NMF is not able to correctly separate the end-members, due to the non-uniqueness of its solutions.

More recently, Gillis and Glineur [2010] have proposed a new algorithm to

[1] Available at http://www.agc.army.mil/hypercube/

solve NMF problems sequentially: the Nonnegative Matrix Underapproximation (NMU). They have shown that NMU outperforms NMF for hyperspectral unmixing for the following reasons:

1. The solution is unique (under some assumptions) [Gillis and Plemmons, 2011];
2. The factorization rank does not need to be chosen a priori;
3. Solutions are sparser than NMF leading to better decomposition into parts [Gillis and Glineur, 2010].

NMU is based on a recursive approach to solve NMF by imposing the upper bound constraint $uv^{\mathrm{T}} \leq M$ to the factor matrices, that ensures their non-negativity.

Formally, given a data matrix $M \in \mathbb{R}^{m \times n}$ and a rank $1 \leq k \leq \min(m, n)$, NMU solves the following rank-one optimization problem:

$$\begin{aligned} \text{minimize :} \quad & \left\| M - uv^{\mathrm{T}} \right\|_F^2 \\ \text{subject to :} \quad & uv^{\mathrm{T}} \leq M, \\ & u \geq 0, v \geq 0. \end{aligned} \tag{6.2}$$

where $u \in \mathbb{R}_+^m$ and $v \in \mathbb{R}_+^n$. A non-negative residual matrix $R = M - uv^{\mathrm{T}} \geq 0$ is than obtained, and the same procedure can be recursively applied on the residual matrices R_i obtained at each iteration. After k steps, NMU provides a rank-k NMF of the data matrix M.

Further modifications of original NMU algorithm were made by adding prior information into the model. Particularly the sparsity constraint on the abundance matrix [Gillis and Plemmons, 2013] and spatial information about pixels [Gillis et al., 2012] have been proposed and their ability in improving the NMU performances has been shown.

In this thesis a modification of NMU including both sparsity constraint and spatial information is proposed, and experimental results show the effectiveness of the method in detecting constitutive materials (end-members) so as to correctly classify pixels according to materials.

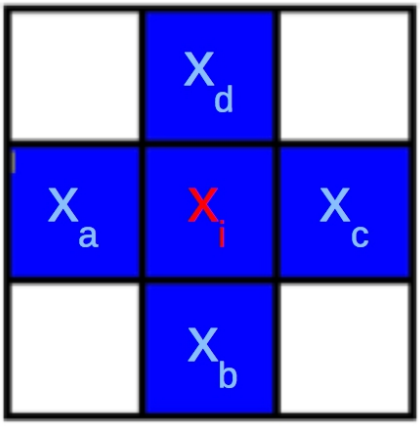


Figure 6.3: Example of neighbor pixels of a given pixel x_i.

6.1.4 Spatial Information

The key idea of the proposed variant is that in each hyperspectral image, neighbor pixels are likely to contain the same materials. In [Gillis et al., 2012] it has been demonstrated that the addition of spatial information to NMU improves the decomposition of the hyperspectral images. Figure 6.3 shows neighboring pixels of a given pixel x_i. In the algorithm the diagonal pixels have not been included in the neighborhood. However it could be easily modified to include them.

To do so the following regularization term is added to the objective function (6.2). The aim is to minimize distances (in terms of abundances) between each pixel and its neighbors:

$$\sum_{i=1}^{m} \sum_{j \in \mathcal{N}(i)} |u_i - u_j| = 2 \|Nu\|_1 , \tag{6.3}$$

where $\mathcal{N}(i)$ is the set of neighboring pixels of pixel i, and $N \in \mathbb{R}^{K \times m}$ is a neighbor matrix that indicates for each pixel x_i its neighbors such that each pair (i, j) of neighboring pixels is represented by a row in which:

$$N(k, i) = 1 \quad \text{and} \quad N(k, j) = -1 \tag{6.4}$$

$$X = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{pmatrix} \qquad N = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 6.4: Example of neighbor matrix N of a simple matrix X.

with $1 \leq i < j \leq m$ and K is the number of neighboring pairs ($K \leq 4m$, because each pixel has at most 4 neighbors: the pixels in the middle of the image have four neighbors, on the border three, on the corner only two). Equation 6.1.4 shows a simple example of neighbor matrix N of a small pixel matrix X. The l_1-norm is more suitable for image analysis due to its ability to preserve the edges, on the contrary the l_2-norm would smooth them out.

6.1.5 Sparsity Information

The key idea of the proposed modification is that each material is present in a relatively small number of pixel, and each pixel contains a small number of constitutive materials. Thus it is possible to constraint column vectors of U to possess only few non-zero elements. Gillis and Plemmons [2013] demonstrate that sparse NMU leads to better decompositions than standard NMU. A regularization term, based on the l_1-norm heuristic approach, is added to the objective function (6.2), in order to minimize the non-zero entries of u: $\|u\|_1$ where $\|u\|_2 = 1$.

6.1.6 Optimization problem

Adding the sparsity constraint and the spatial information to NMU (6.2), leads to the following minimization problem:

$$\min_{u\in\mathbb{R}_+^m, v\in\mathbb{R}_+^n} \left\|M - uv^{\mathrm{T}}\right\|_F^2 + \varphi \left\|u\right\|_1 + \mu \left\|Nu\right\|_1, \qquad (6.5)$$
$$\text{such that} \qquad \left\|u\right\|_2 = 1,\ uv^{\mathrm{T}} \leq M.$$

The objective function is composed by three terms: the first one is the classical mean of squared residuals, the second term imposes the sparsity to the abundance matrix U, and the third term adds spatial information. The regularization parameters μ and φ are used to balance the influence of the three terms. We will refer to (6.5) as *Priors in NMU* (PNMU).

In [Gillis and Glineur, 2010] approximate solutions for the NMU problem 6.2 are obtained by solving the Lagrangian dual

$$\max_{\Lambda\geq 0} \min_{x\geq 0, y\geq 0} L\left(x, y, \Lambda\right) = \left\|M - xy^{\mathrm{T}}\right\|_F^2 + 2\sum_{i,j} \left(xy^{\mathrm{T}} - M\right)_{ij} \Lambda_{ij}, \qquad (6.6)$$

where $\Lambda \in \mathbb{R}^{m\times n}$ is the matrix containing the Lagrangian multipliers of the underapproximation constraints[1]. The authors prove that for a fixed Λ, the problem $\min_{x\geq 0, y\geq 0} L\left(x, y, \Lambda\right)$, called Lagrangian Relaxation of 6.2, is equivalent to

$$\max_{x\geq 0, y\geq 0} x^{\mathrm{T}}\left(M - \Lambda\right) y, \text{such that } \left\|x\right\|_2 = \left\|y\right\|_2 = 1. \qquad (6.7)$$

In order to inject prior information in the factorization process, the regularization terms for the sparsity and spatial information have been added to 6.7.

For a fixed y approximate solutions of (6.5) can be obtained by solving the

[1] In the following variables x and y are used to indicate the column vectors u_r and v_r for $r = 1 \ldots k$

subproblem:

$$\max_{\|x\|_2=1, x\geq 0, \|y\|_2=1, y\geq 0} \left(\underbrace{x^{\mathrm{T}} (M - \Lambda) y - \varphi \|x\|_1 - \mu \|Nx\|_1}_{= f(x)} \right). \tag{6.8}$$

6.1.7 Algorithm

Algorithm 4 formally describes the alternating scheme to solve the NMU problem with sparsity and spatial constraints (6.5).

Alternate scheme

A simple exact block-coordinate descent scheme is used to find good solutions to problem (6.8). This is achieved by applying the following alternating scheme that optimizes one block of variables while keeping the other fixed:

1. $y \leftarrow \frac{\max\left((M-\Lambda)^{\mathrm{T}} x, 0\right)}{\left\|\max\left((M-\Lambda)^{\mathrm{T}} x, 0\right)\right\|_2}$ (lines 21 and 22);
2. $x \leftarrow \mathcal{P}\left(Lx + \nabla f(x)\right)$ (from line 12 to line 12),
   where L is the Lipschitz constant of $\nabla f(x)$ which is equals to the largest eigenvalue of B and with
   $\mathcal{P}(s) = \begin{cases} \frac{\max(0,s)}{\|\max(0,s)\|_2} & \text{if } \|\max(0,s)\|_2 \geq 1 \\ \max(0,s) & \text{otherwise;} \end{cases}$
3. $\Lambda \leftarrow \max\left(0, \Lambda + \alpha_k \left(xy^{\mathrm{T}} - M\right)\right)$ (from line 24 to line 30).

The variables y and Λ are updated as in the original NMU algorithm, whilst the update of x has been modified in order to take into account the penalty terms.

The update of x involves the derivative of L_1-norm that is non-differentiable. Gillis et al. [2012] suggest to use iteratively re-weighted least squares (*IRWLS*) to approximate L_1-norm. After k iteration it is possible to replace $\|Nx\|_1$ with:

$$\|Nx\|_1 \approx x^{\mathrm{T}} \left(\underbrace{N^{\mathrm{T}} W^{(k)\mathrm{T}} W^{(k)} N}_{= B} \right) x, \tag{6.9}$$

where $W^{(k)} = \operatorname{diag}\left(W^{(k)}\right)$ and $w_i^{(k)} = \left(\left|Nx^{(k)}\right|_i + \varepsilon\right)^{-\frac{1}{2}}$ (lines 12 and 32).

For this reason gradient descent equation (line 17) becomes:

$$\nabla f(x) = (M - \Lambda) y - \varphi - \mu (Bx). \tag{6.10}$$

Moreover since B is very large (but sparse), it is computational costly to exactly compute its largest eigenvalue, so Gillis et al. [2012] propose to use several steps of the power method (line 13) to understimate the Lipschitz constant L. In this way the algorithm takes larger steps.

Initialization

Variables (x, y, Λ) have been initialized with an approximate solution of NMU using the algorithm from [Gillis and Plemmons, 2013] (line 4).

Heuristic for the choice of the penalty parameter μ

The heuristic for the choice of the penalty parameter μ proposed in [Gillis et al., 2012], has been used (line 15):

$$\mu = \mu_k \frac{\|Ay\|_\infty}{\|Bx\|_\infty}, \tag{6.11}$$

for some $\mu_k \in [0, 1]$.

Heuristic for the choice of the penalty parameter φ

The heuristic for the choice of the penalty parameter φ proposed in [Gillis and Plemmons, 2013], has been used (line 7):

$$\varphi = \varphi_k \|(M - \Lambda) y\|_\infty, \tag{6.12}$$

for some $\varphi_k \in \mathbb{R}_+$.

Algorithm 4 NMU incorporating spatial and sparseness information

Require: $M \in \mathbb{R}_+^{m\times n}$, $k \in \mathbb{N}_+$, $\varphi_k \in \mathbb{R}_+$, $0 \leq \mu_k \leq 1$, $\epsilon \in \mathbb{R}_+$,maxiter, iter.

Ensure: $(U, V) \in \mathbb{R}_+^{m\times k} \times \mathbb{R}_+^{k\times n}$ s.t. $UV \leq M$ with U containing sparseness and locality information.

Generate the matrix N according to (6.4);
for $k = 1 :$ r **do**
  $z =$ rand(n,1); % Estimate of the eigenvector of B associated with the largest eigenvalue
  $[x, y, \Lambda] =$ rank-one underapproximation(M); % Initialization of (x, y) with an approximate solution to NMU (6.2)
  $u_k \leftarrow x$; $v_k \leftarrow y$; $x \leftarrow \frac{x}{\|x\|_2}$; $y \leftarrow \frac{y}{\|y\|_2}$;
  $w_i = (|Nx|_i + \epsilon)^{-0.5}$; $W = diag(w)$; % Initialization of IRWLS weights
  $\varphi = \varphi_k \|(M - \Lambda)\, y\|_\infty$ % Setting of the sparsity parameter φ
  $x = \max(0, (x - \varphi_k))$; $x = \frac{x}{\|x\|_2}$;
  for $p = 1 :$ maxiter **do**
    $A = M - \Lambda$;
    % Update of x
    $B = (WN)^{\mathrm{T}}(WN)$;
    for $l = 1 : iter$ $z = Bz$; $z = \frac{z}{\|z\|_2}$; %Power method
    for $l = 1 :$ iter **do**
      $\mu = \mu_k \frac{\|Ay\|_\infty}{\|Bx\|_\infty}$; % Setting of the spatial parameter μ
      $L = \max\left(\varepsilon, \mu\left(z^{\mathrm{T}}Bz\right)\right)$ % Approximated Lipschitz constant
      $\nabla f(x) = Ay - \mu Bx - \varphi_k$;
      $x \leftarrow \mathcal{P}(Lx + \nabla f(x))$;
    end for
    % Update of y
    $y \leftarrow \max\left(0, A^{\mathrm{T}}x\right)$;
    if $\|y\|_2 \neq 0$ then $y \leftarrow \frac{y}{\|y\|_2}$;
    %Update of Λ and save (x, y)
    if $x \neq 0$ and $y \neq 0$ **then**
      $\sigma = x^{\mathrm{T}}Ay$; $u_k \leftarrow x$; $v_k \leftarrow \sigma y$;
      $\Lambda \leftarrow \max\left(0, \Lambda - \frac{1}{j+1}\left(M - u_k v_k^{\mathrm{T}}\right)\right)$;
    else
      $\Lambda \leftarrow \frac{\Lambda}{2}$;
      $x \leftarrow \frac{u_k}{\|u_k\|_2}$; $y \leftarrow \frac{v_k}{\|v_k\|_2}$;
    end if
    % Update of the weights
    $w_i = (|Nx|_i + \epsilon)^{-0.5}$; $W = diag(w)$;
  end for
  $M = \max\left(0, M - u_k v_k^{\mathrm{T}}\right)$;
end for

6.2 Experimental Results

In this section, experiments which have been conducted are shown. Four datasets, differing for the total number of pixels and materials have been used, in order to test the effectiveness of the proposed method in correctly detecting materials in images and reduce noise in the basis elements.

All the numerical results have been obtained by implementing the algorithms in Matlab 7.8 codes and running them on a machine equipped with an Intel(R) Xeron(R) CPU E5420 Dual Core 250 GHz, RAM 8.00 GB

It has to be pointed out that parameters φ for the sparsity information and μ for local information strongly influence the results and the convergence of the algorithm. For this reason, different run of the algorithm 4, varying the parameters φ and μ, have been performed.

Differently from classical NMU that is completely unsupervised, NMU with prior information requires human supervision for tuning this parameters. In the following sections the influence of the parameters on the results will be discussed.

6.2.1 Hubble

Hubble database consists of 100 hyperspectral images of the Hubble telescope[1] composed by 128×128 pixels. Figure 6.5 shows the eight materials the images are composed of: Honeycomb Side, Copper Stripping, Green Glue, Aluminum, Solar Cell, Honeycomb Top, Black Rubber Edge and Bolts.

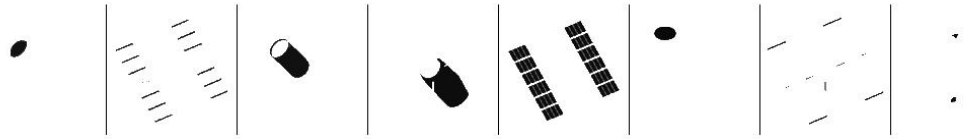


Figure 6.5: Materials of the Hubble teleschope. From left to right: Honeycomb Side,Copper Stripping, Green Glue, Aluminum, Solar Cell, Honeycomb Top, Black Rubber Edge and Bolts.

[1] Available at http://www.agc.army.mil/hypercube/

The following set of parameters have been used for the experiment: $maxiter = 500$, $\varphi_k \in [0, 1]$, $\mu_k \in [0, 1]$, with a step of 0.1, $iter = 10$ and a rank $k = 8$.

Figure 6.6 shows the effect of the parameters φ and μ on the results of PNMU. High values of the locality parameter μ lead to a loss of the sharpness in the abundance images (see Figure 6.6 (b)), whilst high values of the sparsity parameter φ lead to a loss of information in the abuncance images that does not correctly classify pixels belonging to different materials (see Figure 6.6¡ (a)).

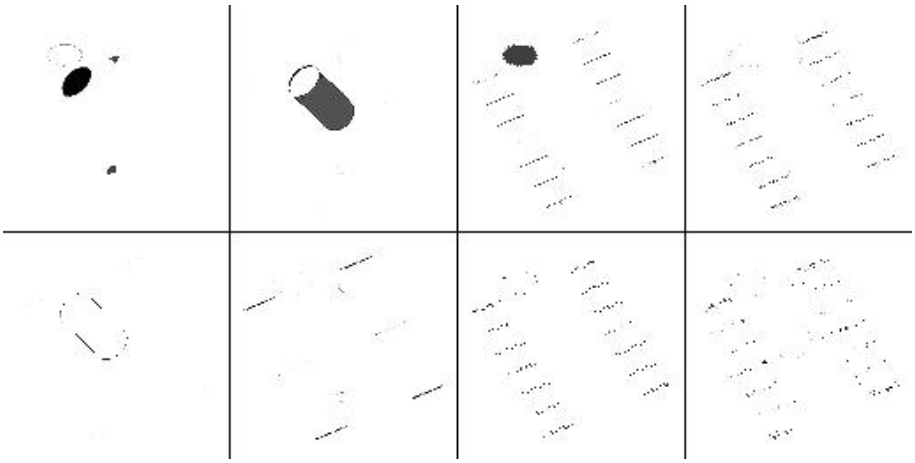
Moreover it has to be pointed out that the sparsity and locality parameters influence each other. In fact when the sparsity is high there are few pixels to be taken into account, and the locality information applyied to them enhances its influence. Figure 6.7 shows an example of bases obtained with a high value of locality information $\mu_k = 0.6$ and high sparisity $\varphi_k = 0.6$. As it could be observed the borders of the images are lost, the images are confused and the algorithm is not able to recognize the pure materials.

Figure 6.8 shows a comparison of the bases obtained with standard NMU (a), NMU with sparsity constraint (b), NMU with local information (c), PNMU (d). It could be observed that the standard NMU does not converge to a good solution, since bases does not represent pure materials, but a mix of them. The sparsity constraint improves this result leading to more separate bases. However it could be noted that the segments in the images are dashed. When local information is added to NMU, the bases have well defined borders, but materials are more mixed than in the sparse case. Finally, when both the sparsity and local prior informations are added to NMU, bases are more sparse but the borders are more clear. Moreover PNMU is able to separate more materials, giving a better decomposition.

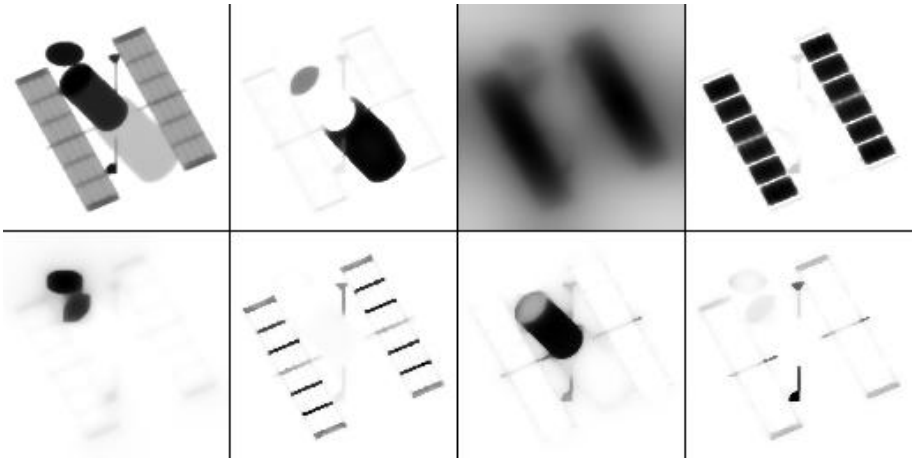
6.2.2 Cuprite

The second dataset that has been used is *Cuprite dataset*[1]. It is more complex than Hubble dataset in terms of number of pixels, and details of the depicted images. It consists of 188 images with 250×191 pixels, and in literature, about 20 different materials (minerals) have been identified Gillis et al. [2012]. Cuprite

[1]Available at http://speclab.cr.usgs.gov/PAPERS.imspec.evol/aviris.evolution.html.It



(a) Basis elements of PNMU for Hubble telescope when only the local information is added to NMU ($\mu_k = 0$ and $\varphi_k = 0.1$).



(b) Basis elements of PNMU for Hubble telescope when only the sparsity costraint is added to NMU ($\mu_k = 0.1$ and $\varphi_k = 0$).

Figure 6.6: Comparison of bases obtained with PNMU varying the locality and sparsity constraints.

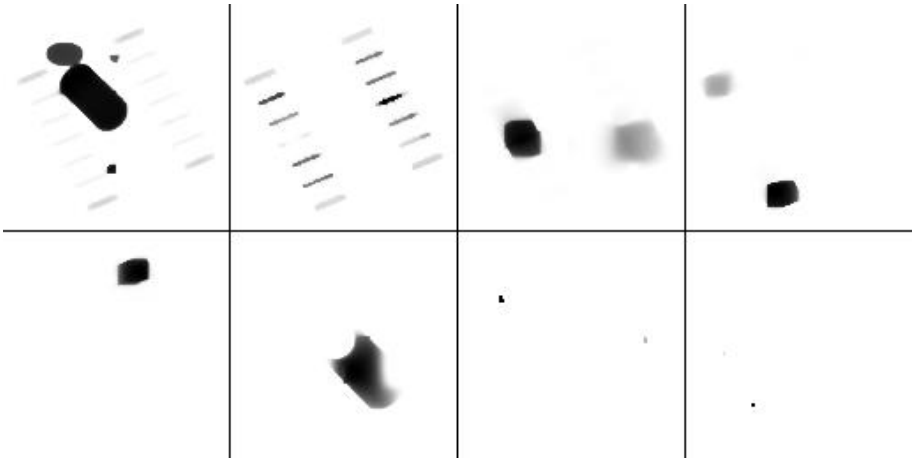


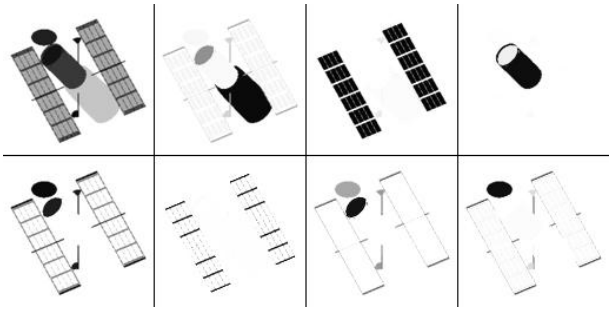
Figure 6.7: Basis elements of PNMU for Hubble telescope with $\mu_k = 0.6$ and $\varphi_k = 0.6$.

has been used to show the effectiveness of the method, in detect materials and classify image pixels according to the materials they belong to, when it is used in real contexts.

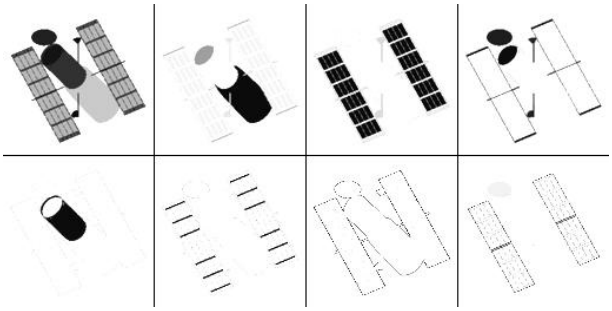
The experiments have been runned with the following set of parameters: $maxiter = 500$, $iter = 10$, $\varphi_k \in [0, 1]$, $\mu_k \in [0, 0.5]$, with a step of 0.05 and a rank $k = 21$.

The locality parameter μ is strongly dependent by the complexity of the scene that is depicted in the images. In fact it enhances the membership values of the neighborhood of a given pixel, to the material to which the pixel belongs. The more the images are detailed, the stronger is the influence of the local information on the factorization results. For this reason the experiments have been runned with a maximum value of $\mu_k = 0.5$, higher values give not significative results, and with a small step. Figure 6.9 shows basis elements obtained with PNMU with parameters μ_k that regulates the local information equals to the maximum value 0.5. It could be observed that the results are poor, and the basis images are blurry and sparse (even if sparsity has not been imposed to the algorithm).

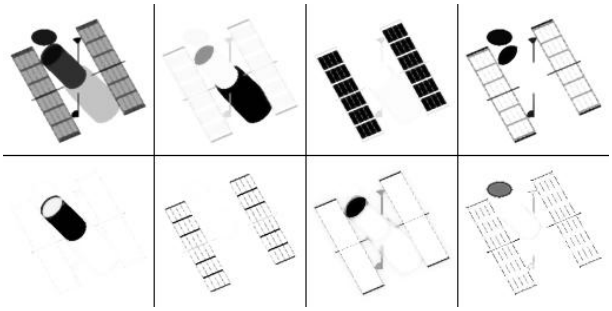
Similarly, high values of sparsity lead to poor results. Figure 6.10 shows basis elements obtained with PNMU with parameters φ_k, that regulates the sparsity of the bases, equals to the maximum value 1. The method does not converge, some bases have few non-zero pixels belonging to the actual material, and the



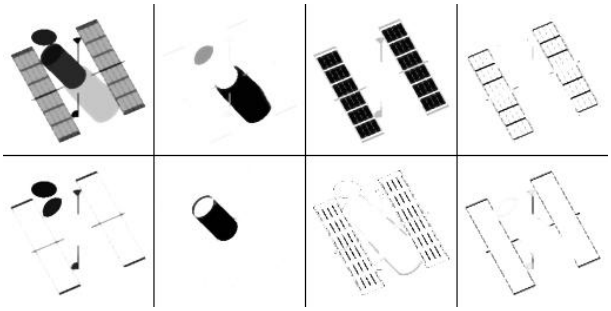
(a) Basis elements of NMU for Hubble telescope (PNMU with parameters $\mu_k = 0$ and $\varphi_k = 0$).



(b) Basis elements of NMU with sparsity constraint for Hubble telescope (PNMU with parameters $\mu_k = 0$ and $\varphi_k = 0.2$).



(c) Basis elements of NMU with local information for Hubble telescope (PNMU with parameters $\mu_k = 0.3$ and $\varphi_k = 0$).



(d) Basis elements of PNMU for Hubble telescope with parameters $\mu_k = 0.3$ and $\varphi_k = 0.2$.

Figure 6.8: Comparison of bases obtained with NMU variants for Hubble dataset.

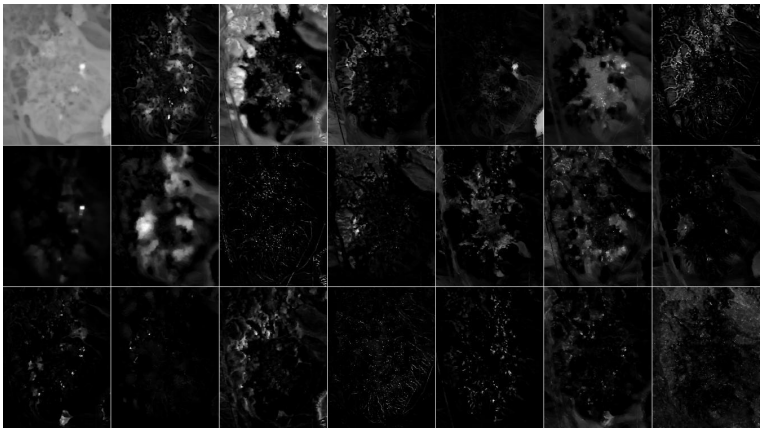


Figure 6.9: Basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.5$ and $\varphi_k = 0$

remaining bases do not capture any material.

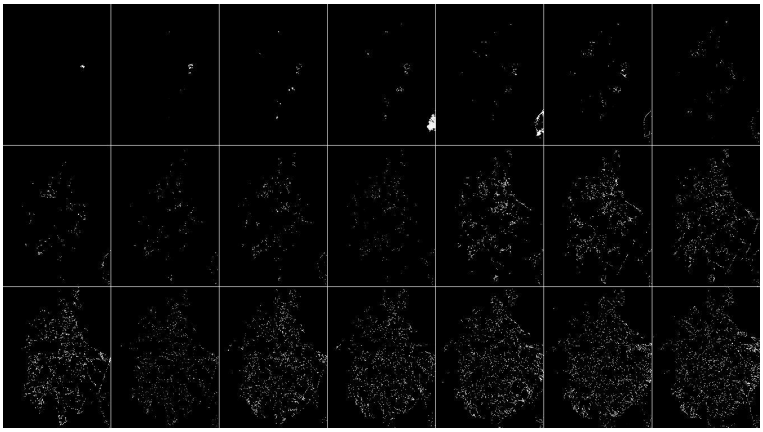
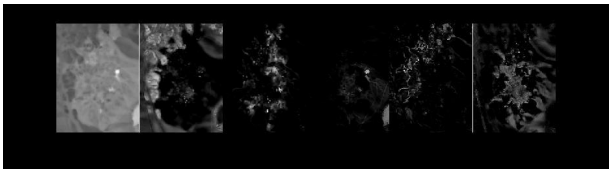


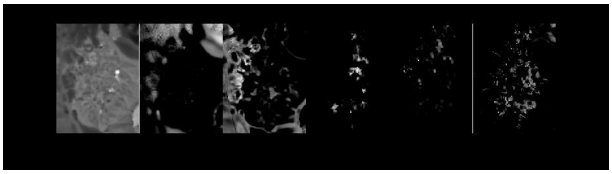
Figure 6.10: Basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0$ and $\varphi_k = 1$.

As it has been previously discussed with Hubble dataset, sparsity and locality parameters influence each other. Indeed the higher the sparsity is, the less the pixels in the image are, so the influence of the locality parameter, on the final results, exponentially grows. Figure 6.11 shows the first six bases obtained fixing the locality ($\varphi_k = 0.5$) and varying the sparsity ($\mu_k = 0.1, 0.2, 0.3, 0.4, 0.6$). Small variations of the sparsity cause big losses of the clarity in the bases. For this reason a small step is used to vary the parameters.

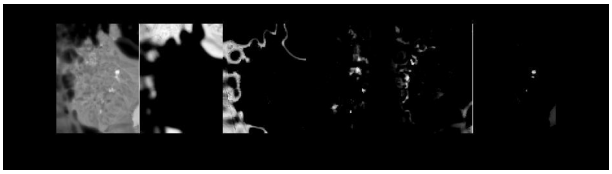


(a) First six basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.5$ and $\varphi_k = 0.1$.

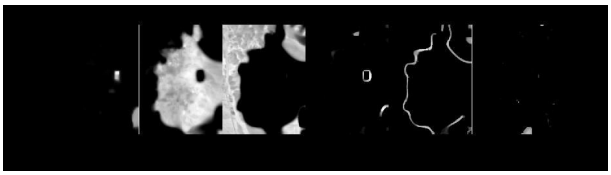
(b) First six basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.5$ and $\varphi_k = 0.2$.



(c) First six basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.5$ and $\varphi_k = 0.3$.



(d) First six basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.5$ and $\varphi_k = 0.4$.



(e) First six basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.5$ and $\varphi_k = 0.6$.

Figure 6.11: Study of the influence of the locality term on the bases when the sparsity grows.

The proposed method has been compared with the standard NMU algorithm Gillis and Glineur [2010] and its variants with sparsity information Gillis and Plemmons [2013] and local information Gillis et al. [2012] on Cuprite dataset. Figures 6.12 and 6.13 show the basis elements obtained with the four variants of the NMU algorithm. Figure 6.12(a) reports the bases obtained with standard NMU, it could be observed that the images are not clear, and the bases contain noise pixels. The sparsity constraint on the basis vectors allow to remove the noise, but nevertheless the borders of the part of the images that belong to the specific materials are not well defined (figure 6.12(b)). Local information about pixels in the images helps to obtain better results in term of edge definition, but the basis images are still noisy (figure 6.13(a)). Combining sparsity constraint and local information in the factorization process improves the results. Figure 6.13(b) shows the bases obtained with PNMU for Cuprite dataset with parameters $\mu_k = 0.1$ and $\varphi_k = 0.2$. The images are clear, there is not noise, and the edges are well defined.

6.2.3 Urban

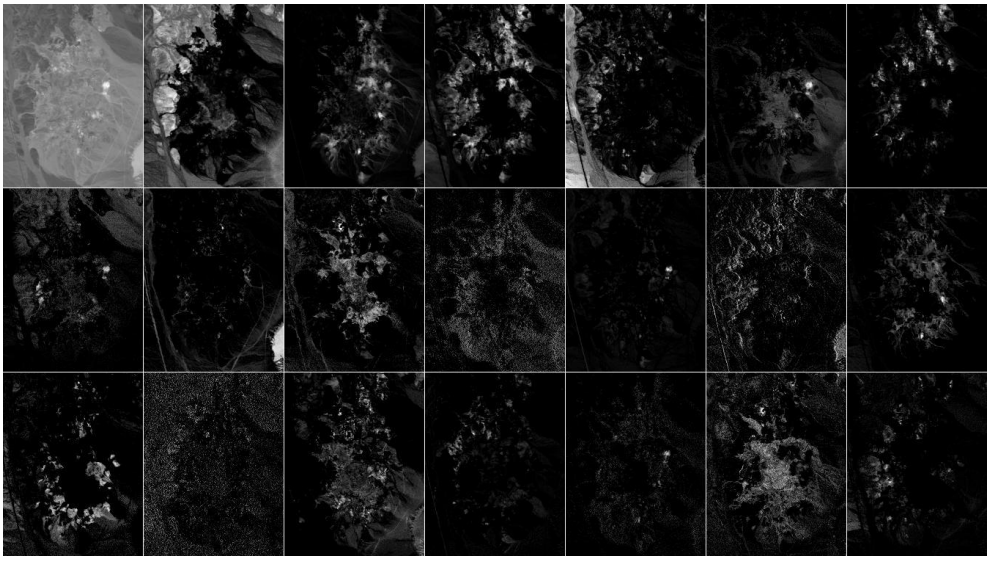
The HYDICE Urban dataset[1] consists of 210 images at different spectral bands, each composed by 307×307 pixels. Data are mainly composed of 6 materials: road, dirt, trees, roofs, grass and metal.

The following set of parameters have been used for the experiment: $maxiter = 500$, $\varphi_k \in [0, 1]$, $\mu_k \in [0, 1]$, with a step of 0.1, $iter = 10$ and a rank $k = 6$.

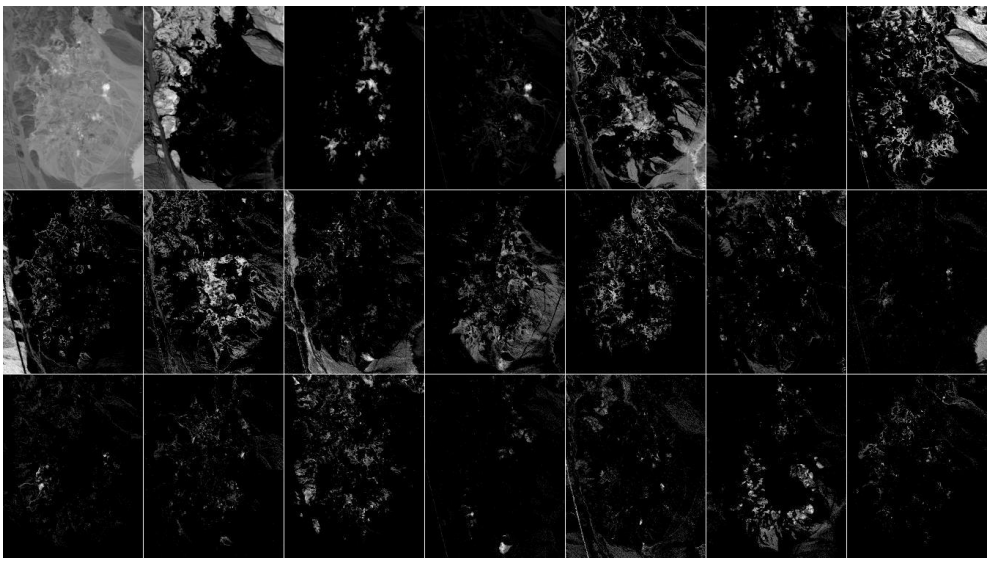
Without loss of precision, the dataset has been pre-processed reducing its dimensionality to facilitate the computation. For each image the number of pixels has been halved considering only one pixel every two.

Figure 6.14 shows the best solution obtained with $\mu_k = 0.1$ and $\varphi_k = 0.3$. Results suggest that even when the value of locality parameter μ is low ($\mu_k = 0.1$) it gives too much information to the process leading to shade images. Moreover a justification to this result is given by the complexity of the depicted scene. When the selected areas are small (as in the case of the fifth and sixth bases) the influence of the local information is stronger than in the other bases. How-

[1] Available at http://www.agc.army.mil/hypercube/

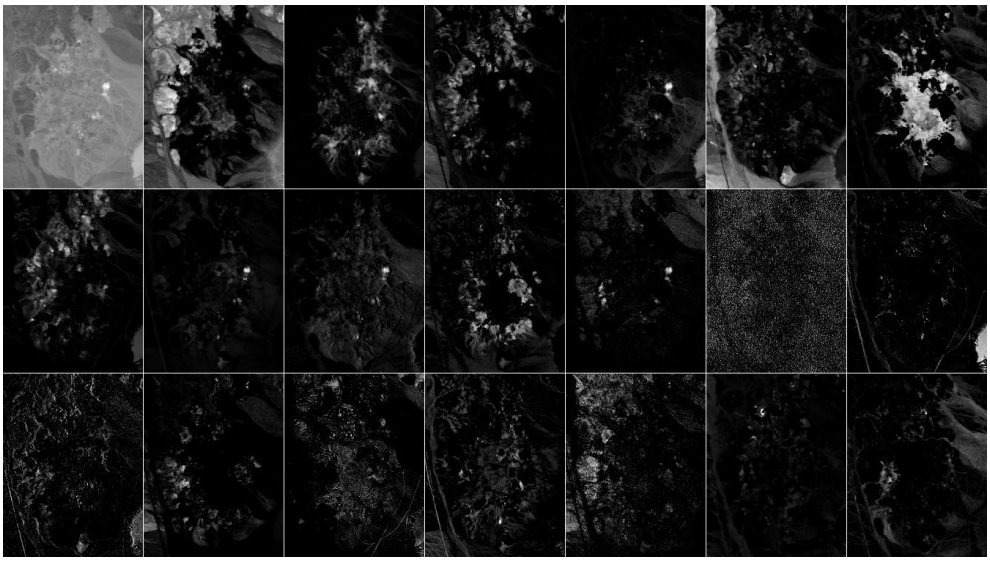


(a) Basis elements of NMU for Cuprite dataset (PNMU with parameters $\mu_k = 0$ and $\varphi_k = 0$).

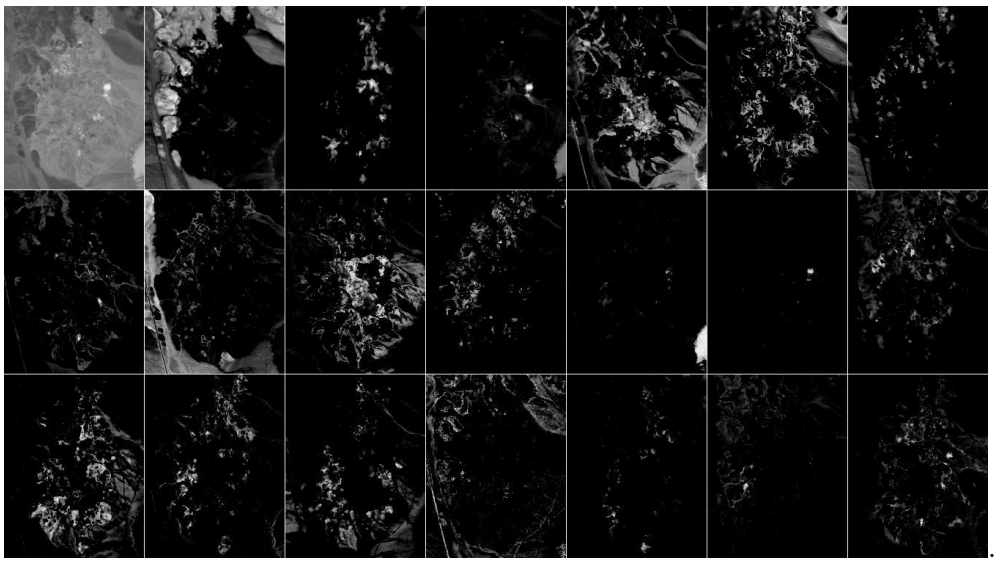


(b) Basis elements of NMU with sparsity constraint for Cuprite dataset (PNMU with parameters $\mu_k = 0$ and $\varphi_k = 0.2$).

Figure 6.12: Comparison of bases obtained with NMU and NMU with sparsity.



(a) Basis elements of NMU with local information for Cuprite dataset (PNMU with parameters $\mu_k = 0.1$ and $\varphi_k = 0$).



(b) Basis elements of PNMU for Cuprite dataset with parameters $\mu_k = 0.1$ and $\varphi_k = 0.2$.

Figure 6.13: Comparison of bases obtained with NMU with local information and with PNMU

ever it is worth to note that materials in the bases are well separated. Further improvements will be addressed to separately tune parameters μ and φ for each basis.

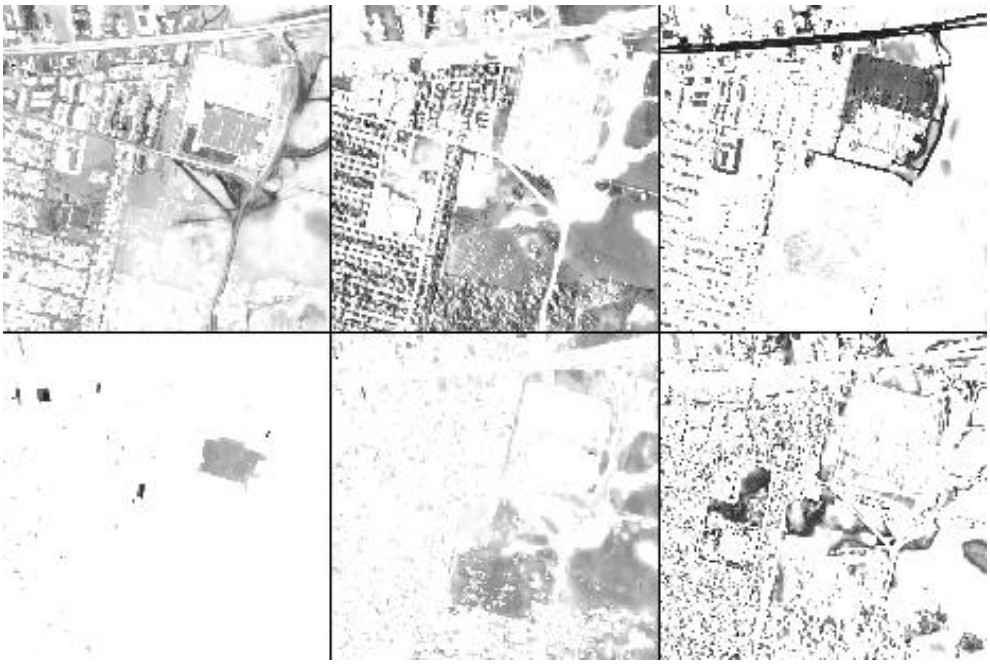


Figure 6.14: Basis elements of PNMU for Urban dataset with parameters $\mu_k = 0.1$ and $\varphi_k = 0.3$.

6.2.4 San Diego airport

The San Diego airport dataset consists of 158 images composed by 400×400 pixels. There are four basic types of materials: road surfaces, roofs, trees and grass, but there are mainly three different types of road surfaces [Gillis and Plemmons, 2013].

The same pre-processing phase, as for Urban dataset, has been conducted to accelerate the computation.

The following set of parameters have been used for the experiment: $maxiter = 300$, $\varphi_k \in [0.2, 0.3]$ with a step of 0.01, $\mu_k \in [0, 1]$ with a step of 0.1, $iter = 10$, and a rank $k = 8$.

As for the Urban dataset, the complexity of the scene compromises the accuracy of the results. Particularly PNMU is able to recognize roofs in the first basis, roads of type two in the forth basis, grass in the third, and roads of type one in the second mixed with other materials. It is clear that basis two needs more sparsity, whilst in the case of basis one and seven local information has too much influence, and the images result blurry.

Bases three, six and eight have captured some outliers. This behaviour could be prevented by using a lower bound on the density of u.

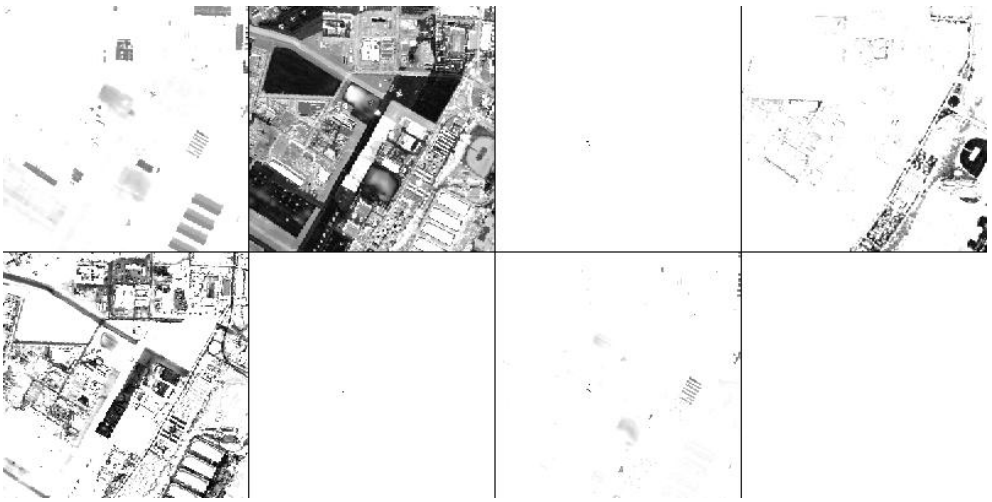


Figure 6.15: Basis elements of PNMU for San Diego airport dataset with parameters $\mu_k = 0.01$ and $\varphi_k = 0.27$.

Chapter 7

Conclusions and Future Work

In this thesis the use of matrix factorization as intelligent data analysis tool, has been studied. Particularly three different approaches have been proposed.

NMF has been widely used in document clustering applications, due to its capability to suggest interpretable concepts to group data. However, NMF algorithms are iterative, and very sensitive to the initial basis and encoding matrices. The use of the subtractive clustering (SC) schema has been proposed because, differently from other schemes, it does not require the specification of the number of clusters (and the corresponding rank used in factorization) but other hyper-parameters that may be more significant in problems where the number of clusters cannot be known a-priori. The experimental results, based on benchmark document collections, provide some interesting properties emerging from the use of SC for initializing the factorization algorithms. Particularly, it has been shown that SC is able to suggest a suitable rank k for detecting interpretable concepts from documents.

A novel NMF algorithm, namely Masked NMF has been proposed in order to overcome the limitations of classical NMF and to introduce knowledge in the factorization process, making the proposed MNMF algorithm a useful tool for IDA. The query-based approach has been adopted to allow the analyst to specify what parts she is interested to discover. As shown in the numerical examples, the proposed approach is able to extract the subset of data that are actually represented by the parts, discarding the data in the matrix X that do not find a neat representation by the parts and returning the subset of samples that contains

the selected parts. Future work can be addressed to assess the performance of the query based MNMF approach on different real datasets as well as to further investigate its capability of selecting local features hidden in data. Particularly, an automatic tool to support the analyst in selecting parts by mask matrices will be developed. The problem of find the best mask according to some criteria is a combinatorial problem with an exponential complexity. A genetic algorithm to optimize the mask matrix P, searching in the space of all the possible combinations of P, will be investigated.

A variant of standard NMU applyied to hyperspectral images has been proposed in order to use prior information into the factorization process. Sparsity constraint on the abundance matrix U has been shown to lead to better results than standard NMU. Local information about pixels were shown to reduce noise, because pixel belonging to the neighborhood are more willing to belong to the same material. In this thesis a variant of NMU, named PNMU, that adds both sparsity and local information to the process has been proposed. Experiments on the Hubble telescope dataset have shown the effectiveness of the method in detecting materials represented in images, and the influence of the priors on the results. Additional experiments on Cuprite dataset have shown the effectiveness of the method when real data are considered. Further experiments on Urban dataset and San Diego airport dataset have shown some limits of the proposed method. Particularly the necessity of allow the separate tuning of the sparsity and locality parameters for the different bases, has emerged. For this reason further work will be addressed in this direction.

References

R. Albright, J. Cox, D. Duling, A. Langville, and C. Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. Technical report, NCSU Technical Report Math 81706, 2006. 19, 20, 21, 24

S. Anbumalar, R. Anandanatarajan, and P. Rameshbabu. Sparse non-negative matrix factorization and its application in overlapped chromatograms separation. *International Journal of Computer Applications*, 21:1–10, 2013. 23

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`. 73

R. E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, 1961. 6

E. Benetos, M. Kotti, C. Kotropoulos, J. J. Burred, G. Eisenberg, M. Haller, and T. Sikora. Comparison of subspace analysis-based and statistical model-based algorithms for musical instrument classification. In *2nd Workshop on Immersive Communication and Broadcast Systems (ICOB '05)*, Berlin, Germany, October 2005. 15

E. Benetos, M. Kotti, and C. Kotropoulos. Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, volume V, pages 221–224, 2006a. 15

E. Benetos, M. Kotti, and C. Kotropoulos. Applying supervised classifiers based on non-negative matrix factorization to musical instrument classification. In *ICME*, pages 2105–2108. IEEE, 2006b. 15

A. Berman and R. Plemmons. *Nonnegative matrices in the mathematical sciences*. Academic Press, 1979. 13

M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173, 2007. 2, 13

M. Berthold and D. J. Hand, editors. *Intelligent Data Analysis: An Introduction*. Springer-Verlag New York, Inc., 1st edition, 1999. ISBN 3540658084. 4

M. R. Berthold, C. Borgelt, F. Hppner, and F. Klawonn. *Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data*. Springer Publishing Company, Incorporated, 1st edition, 2010. 5, 28

R. Bierig, F. Piroi, M. Lupu, and A. Hanbury. Conquering data: The state of play in intelligent data analytics. 1

C. Boutsidis and E. Gallopoulos. Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41:1350–1362, April 2008. ISSN 0031-3203. 14, 19, 23, 67

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*, 101(12):4164–4169, March 2004. ISSN 1091-6490. doi: 10.1073/pnas.0308531101. 15

I. Buciu, N. Nikolaidis, and I. Pitas. On the initialization of the dnmf algorithm. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 4671–4674. IEEE, 2006. 14, 19

J. E. Burger and P. L. M. Geladi. Hyperspectral image data conditioning and regression analysis. In *Techniques and Applications of Hyperspectral Image Analysis*, pages 127–153. John Wiley & Sons, Ltd, 2007. 15

D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *Proc. 2009 Int. Joint Conference on Artificial Intelligence (IJCAI'09)*, 2009. 30

P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. D. Pascual-Montano. Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, 7:78, 2006. 15

G. Casalino, N. Del Buono, and C. Mencar. Subtractive initialization of nonnegative matrix factorizations for document clustering. In Anna Fanelli, Witold Pedrycz, and Alfredo Petrosino, editors, *Fuzzy Logic and Applications*, volume 6857 of *Lecture Notes in Computer Science*, pages 188–195. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-23712-6. 19, 67

G. Casalino, N. Del Buono, and M. Minervini. Nonnegative matrix factorizations performing object detection and localization. *Appl. Comp. Intell. Soft Comput.*, 2012:15:15–15:15, January 2012. ISSN 1687-9724. doi: 10.1155/2012/781987. 15

G. Casalino, N. Del Buono, and C. Mencar. Subtractive clustering for seeding non-negative matrix factorizations. *Information Sciences*, 257(0):369 – 387, 2014. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2013.05.038. 67

R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966. 13

J. Chen, S. Feng, and J. Liu. Topic sense induction from social tags based on non-negative matrix factorization. *Information Sciences*, 280(0):16 – 25, 2014. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2014.04.048. 15

Y. Chen, M. Rege, M. Dong, and J. Hua. Incorporating user provided constraints into document clustering. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 103–112, Oct 2007. doi: 10.1109/ICDM.2007.67. 30

Y. Chen, M. Rege, M. Dong, and J. Hua. Non-negative matrix factorization for semi-supervised data clustering. *Knowledge and Information Systems*, 17(3):355–379, 2008. ISSN 0219-1377. doi: 10.1007/s10115-008-0134-6. 30

Y. Chen, L. Wang, and M. Dong. Non-negative matrix factorization for semisupervised heterogeneous data coclustering. *Knowledge and Data Engineering,*

IEEE Transactions on, 22(10):1459–1474, Oct 2010. ISSN 1041-4347. doi: 10.1109/TKDE.2009.169. 30

S. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent Fuzzy Systems*, 2:267–278, 1994. 31

Y. Cho and L. K. Saul. Nonnegative matrix factorization for semi-supervised dimensionality reduction. *CoRR*, abs/1112.3714, 2011. 30

S. Choi. Algorithms for orthogonal nonnegative matrix factorization. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1828–1832. IEEE, june 2008a. 29

S. Choi. Algorithms for orthogonal nonnegative matrix factorization. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1828–1832. IEEE, june 2008b. 12

A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley Publishing, 2009. ISBN 0470746661, 9780470746660. 2, 10, 15

N. Del Buono. A penalty function for computing orthogonal non-negative matrix factorizations. In *ISDA*, pages 1001–1005. IEEE Computer Society, 2009. ISBN 978-0-7695-3872-3. 29

N. Del Buono and M. Lucarelli. Comparative studies on initializations for non-negative matrix factorization algorithms. Technical report, University of Bari Aldo Moro, 2010. 14

M. Desmarais. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. best paper award. In Mykola Pechenizkiy, Toon Calders, Cristina Conati, Sebastián Ventura, Cristóbal Romero, and John C. Stamper, editors, *EDM*, pages 41–50, 2011. ISBN 978-90-386-2537-9. 14

K. Devarajan. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Computational Biology*, 4(7), 2008. 15

I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *Proceeding of Neural Information Processing Systems*, pages 283–290. Curran Associates Inc, 2005. 11

C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and k-means - spectral clustering. In *Proceedings of the SIAM Data Mining Conference*, pages 606–610. SIAM, 2005. 25, 27, 28, 44

C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006. 15

K. Drakakis, S. Rickard, R. de Fréin, and A. Cichocki. Analysis of financial data using non-negative matrix factorization. *International Mathematical Forum*, 3 (38):1853 – 1870, 2008. 15

S. Essid and C. Févotte. Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring. *IEEE Transactions on Multimedia*, 15(2):415–425, Feb. 2013. doi: http://dx.doi.org/10.1109/TMM.2012.2228474. 25

U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, 1996. ISBN 0-262-56097-6. 5

T. Feng, S.Z. Li, H. Y. Shum, and H. J. Zhang. Local non-negative matrix factorization as a visual representation. In *Proceedings of the 2Nd International Conference on Development and Learning*, ICDL '02, page 178. IEEE Computer Society, 2002. ISBN 0-7695-1459-6. 27

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936. 7

Y. Gao and G. Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975, 2005. 15, 27

N. Gillis. The why and how of nonnegative matrix factorization. In M. Signoretto J.A.K. Suykens and A. Argyriou, editors, *Regularization, Optimization, Kernels, and Support Vector Machines*, Machine Learning and Pattern Recognition Series. Chapman and Hall/CRC, 2014. 2, 10

N. Gillis and F. Glineur. Using underapproximations for sparse nonnegative matrix factorization. *Pattern Recogn.*, 43(4):1676–1687, apr 2010. ISSN 0031-3203. doi: 10.1016/j.patcog.2009.11.013. 27, 86, 87, 90, 101

N. Gillis and R. J. Plemmons. Dimensionality reduction, classification, and spectral mixture analysis using non-negative underapproximation. *Optical Engineering*, 50(2):027001–027001–16, 2011. doi: 10.1117/1.3533025. 87

N. Gillis and R. J. Plemmons. Sparse nonnegative matrix underapproximation and its application to hyperspectral image analysis. *Linear Algebra and its Applications*, 438(10):3991 – 4007, 2013. ISSN 0024-3795. doi: http://dx.doi.org/10.1016/j.laa.2012.04.033. 3, 15, 87, 89, 92, 101, 104

N. Gillis, R. J. Plemmons, and Q. Zhang. Priors in sparse recursive decompositions of hyperspectral images, 2012. 3, 84, 87, 88, 91, 92, 95, 101

N. Gillis, D. Kuang, and H. Park. Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization. *CoRR*, abs/1310.7441, 2013. 15

G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Edition)*. The Johns Hopkins University Press, 2001. 2

G.H. Golub, A. Hoffman, and G.W. Stewart. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its Applications*, 88–89(0):317 – 327, 1987. ISSN 0024-3795. doi: http://dx.doi.org/10.1016/0024-3795(87)90114-5. 8

Q. Gu, J. Zhou, and C. H. Q. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210. SIAM, 2010. 15

D. Guillamet and J. Vitrià. Non-negative Matrix Factorization for Face Recognition. In *CCIA '02: Proceedings of the 5th Catalonian Conference on AI*, pages 336–344. Springer-Verlag, 2002. 14

D. Guillamet and J. Vitrià. Evaluation of distance metrics for recognition based on non-negative matrix factorization. *Pattern Recogn. Lett.*, 24(9-10):1599–1605, June 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(02)00399-9. 14, 15

D. J. Hand. Intelligent data analysis: Issues and opportunities. In Xiaohui Liu, Paul R. Cohen, and Michael R. Berthold, editors, *IDA*, volume 1280 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 1997. 4

Y. He, H. Lu, L. Huang, and X. Shi. Non-negative matrix factorization with pairwise constraints and graph laplacian. *Neural Processing Letters*, pages 1–19, 2014a. ISSN 1370-4621. doi: 10.1007/s11063-014-9350-0. 30

Y. He, H. Lu, and S. Xie. Semi-supervised non-negative matrix factorization for image clustering with graph laplacian. *Multimedia Tools and Applications*, 72 (2):1441–1463, 2014b. ISSN 1380-7501. doi: 10.1007/s11042-013-1465-1. 30

M. Heiler and C. Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *J. Mach. Learn. Res.*, 7:1385–1407, December 2006. ISSN 1532-4435. 30

K. E. Heinrich, M. W. Berry, and R. Homayouni. Gene tree labeling using nonnegative matrix factorization on biomedical literature. *Computational Intelligence and Neuroscience*, 2008:12, 2008. 23

J. H. Holmes and N. Peek. Intelligent data analysis in biomedicine. *Journal of Biomedical Informatics*, 40(6):605–608, 2007. 5

P. K. Hopke. *Receptor modeling in environmental chemistry.* Wiley and Sons, 1985. 15

H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.*, 24, 1933. 7

P. O. Hoyer. Non-negative sparse coding. In *Neural networks for signal processing XII(Proc. IEEE workshop on neural networks for signal processing)*, pages 557–565, 2002. 12, 26, 27

P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, December 2004. ISSN 1532-4435. 25, 26

A. Hyvärinen. Survey on independent component analysis. *Neural Comp. Surveys*, 2:94–128, 1999. 7

Yoo J. and Choi S. Orthogonal nonnegative matrix tri-factorization for co-clustering: multiplicative updates on Stiefel manifolds. *Information Processing and Management*, 46:559–570, 2010. 29

J. E. Jackson. *A User's Guide to Principal Components*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2003. ISBN 0471471348. 7

A. Janecek and Y. Tan. Using population based algorithms for initializing nonnegative matrix factorization. In Ying Tan, Yuhui Shi, Yi Chai, and Guoyin Wang, editors, *Advances in Swarm Intelligence*, volume 6729 of *Lecture Notes in Computer Science*, pages 307–316. Springer Berlin / Heidelberg, 2011. 19, 24

S. Jia and Y. Qian. A complexity constrained nonnegative matrix factorization for hyperspectral unmixing. In Mike E. Davies, Christopher J. James, Samer A. Abdallah, and Mark D. Plumbley, editors, *ICA*, volume 4666 of *Lecture Notes in Computer Science*, pages 268–276. Springer, 2007. ISBN 978-3-540-74493-1. 15

L. Jing, J. Yu, T. Zeng, and Y. Zhu. Semi-supervised clustering via constrained symmetric non-negative matrix factorization. In FabioMassimo Zanzotto, Shusaku Tsumoto, Niels Taatgen, and Yiyu Yao, editors, *Brain Informatics*, volume 7670 of *Lecture Notes in Computer Science*, pages 309–319.

Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35138-9. doi: 10.1007/978-3-642-35139-6_29. 30

I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986. 2, 7

H. F. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20(1):141–151, April 1960. 13

E. Kim, P. K. Hopke, and Edgerton E. S. Source identification of atlanta aerosol by positive matrix facorization. *Journal of the Air and Waste Management Association*, pages 733–739, 2003. 15

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. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm134. 15, 27

A. N. Langville. Experiments with the nonnegative matrix factorization and the reuters10 dataset. Technical report, Slides from SAS Meeting, 2005. 21, 22

C. Lazar and A. Doncescu. Non negative matrix factorization clustering capabilities; application on multivariate image segmentation. In Leonard Barolli, Fatos Xhafa, and Hui-Huang Hsu, editors, *CISIS*, pages 924–929. IEEE Computer Society, 2009. ISBN 978-0-7695-3575-3. 29

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. ISSN 0028-0836. doi: 10.1038/44565. 2, 10

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001. 2, 10, 11, 65

H. Lee, J. Yoo, and S. Choi. Semi-supervised nonnegative matrix factorization. *Signal Processing Letters, IEEE*, 17(1):4–7, Jan 2010. ISSN 1070-9908. doi: 10.1109/LSP.2009.2027163. 30

S. Z. Li, X. Hou, H. Zhang, and Qi. Cheng. Learning spatially localized, parts-based representation. In *Computer Vision and Pattern Recognition*, volume 1, pages 207–212, 2001. doi: 10.1109/CVPR.2001.990477. 27

T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 362–371, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2701-9. 25, 29

T. Li and C. H. Q. Ding. Nonnegative matrix factorizations for clustering: A survey. In *Data Clustering: Algorithms and Applications*, pages 149–176. CRC Press, 2013. 25, 29

Z. Lihong, G. Zhuang, and X. Xu. Facial expression recognition based on pca and nmf. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 6826–6829, June 2008. doi: 10.1109/WCICA.2008.4593968. 17

H. Liu and H. Motoda. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007. ISBN 1584888784. 1

H. Liu and Z. Wu. Non-negative matrix factorization with constraints. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010. 30

H. Liu, Z. Wu, D. Cai, and T. S. Huang. Constrained nonnegative matrix factorization for image representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34 (7):1299–1311, 2012. 30

W. Liu and N. Zheng. Non-negative matrix factorization based methods for object recognition. *Pattern Recogn. Lett.*, 25(8):893–897, June 2004. ISSN 0167-8655. doi: 10.1016/j.patrec.2004.02.002. 14

W. Liu, N. Zheng, and X. Lu. Non-negative matrix factorization for visual coding. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, volume 3, pages 293–296, apr 2003. 27

N. Lyubimov and M. Kotov. Non-negative matrix factorization with linear constraints for single-channel speech enhancement. In Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier, editors, *INTERSPEECH*, pages 446–450. ISCA, 2013. 30

W. K. Ma, J. M. B. Dias, T. H. Chan, N. Gillis, P. D. Gader, A. J. Plaza, A. M. Ambikapathi, and C. Y. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Process. Mag.*, 31(1): 67–81, 2014. doi: 10.1109/MSP.2013.2279731. 15

M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009. doi: 10.1073/pnas.0803205106. 7

J. Maisog. *Non-negative Matrix Factorization: Assessing Methods for Evaluating the number of Components, and the Effect of Normalization Thereon.* PhD thesis, Georgetown University, May 2009. 13

E. Mejía-Roa, P. Carmona-Saez, R. Nogales, C. Vicente, M. Vázquez, X. Y. Yang, C. García, F. Tirado, and A. D. Pascual-Montano. bionmf: a web-based tool for nonnegative matrix factorization in biology. *Nucleic Acids Research*, 36: 523–528, 2008. 15

A. Mirzal. Clustering and latent semantic indexing aspects of the nonnegative matrix factorization. *arXiv preprint arXiv:1112.4020*, pages 1–28, 2011. 29

A. Pascual-Montano, J.M. Carazo, K. Kochi, D. Lehmann, and R.D. Pascual-Marqui. Nonsmooth nonnegative matrix factorization (nsnmf). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):403–415, March 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.60. 27

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. ISSN 0024-3795. doi: http://dx.doi.org/10.1016/j.laa.2005.06.025. 15, 27

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901. 7

N. N. Pise and P. Kulkarni. A survey of semi-supervised learning methods. In *Computational Intelligence and Security*, pages 30–34. IEEE Computer Society, 2008. ISBN 978-0-7695-3508-1. 30

N. Polettini. *The Vector Space Model in Information Retrieval- Term Weighting Problem*, 2004. 32

F. Pompili, N. Gillis, P. A. Absil, and F. Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *CoRR*, abs/1201.0901, 2012. 27

M. Rezaei, R. Boostaniand, and M. Rezaei. An efficient initialization method for nonnegative matrix factorization. *Journal of Applied Sciences*, 11:354–359, 2011. 19, 22

B. Ribeiro, C. Silva, A. Vieira, and J. C. das Neves. Extracting discriminative features using non-negative matrix factorization in financial distress data. In Mikko Kolehmainen, Pekka J. Toivanen, and Bartlomiej Beliczynski, editors, *ICANNGA*, volume 5495 of *Lecture Notes in Computer Science*, pages 537–547. Springer, 2009. ISBN 978-3-642-04920-0. 15

F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Inf. Process. Manage.*, 42(2):373–386, March 2006. ISSN 0306-4573. doi: 10.1016/j.ipm.2004.11.005. 14, 28

P. Shippert. Introduction to hyperspectral image analysis. *Online Journal of Space Communication.*, 2003. 84

A. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis Applications in the Chemical Sciences*. Wiley-VCH, 2004. 20

R. Smith. Introduction to hyperspectral imaging. *Microimages*, 2006. 84

V. Snasel, J. Platos, and P. Kromer. Developing genetic algorithms for boolean matrix factorization. In *Proceedings of DATESO*. VSB, 2008. 24

C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904. 7

K. Stadlthanner, F.J. Theis, E.W. Lang, A.M. Tome, C.G. Puntonet, P. Gomez Vilda, T. Langmann, and G. Schmitz. Sparse nonnegative matrix factorization applied to microarray data sets. In *Independent Component Analysis and Blind Signal Separation*, volume 3889 of *Lecture Notes in Computer Science*, pages 254–261. Springer Berlin / Heidelberg, 2006. 24

X. Sun, Q. Zhang, and Z. Wang. Face recognition based on nmf and svm. *Electronic Commerce and Security, International Symposium*, 1:616–619, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/ISECS.2009.98. 14

R. Tandon and S. Sra. Sparse nonnegative matrix approximation: new formulations and algorithms. Technical report, MPI Technical Report, 2010. 11

M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, January 1991. ISSN 0898-929X. doi: 10.1162/jocn.1991.3.1.71. 9

L.J.P. Van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review, 2008. 6

S. A. Vavasis. On the complexity of non-negative matrix factorization. *SIAM Journal of Optimization*, 20:1364–1377, 2007. 13

C. Wang, S. Yan, L. Zhang, and H. Zhang. Non-negative semi-supervised learning. In David A. Van Dyk and Max Welling, editors, *AISTATS*, volume 5 of *JMLR Proceedings*, pages 575–582. JMLR.org, 2009. 30

F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *Data Min. Knowl. Discov.*, 22(3):493–521, May 2011. ISSN 1384-5810. doi: 10.1007/s10618-010-0181-y. 14

X. Wang, X. Xie, and L. Lu. An effective initialization for orthogonal nonnegative matrix factorization. *Journal of Computational Mathematics*, 30:34–46, 2012. 23

Y. Wang, Y. Jia, C. Hu, and M. Turk. Fisher non-negative matrix factorization for learning local features. In *Asian Conference on Computer Vision*, January 2004. 30

S. Wild, J. Curry, and A. Dougherty. Improving non-negative matrix factorizations through structured initialization. *Pattern Recognition*, 37:2217–2232, 2004. 13, 19, 24, 32

S.M. Wild. *Seeding non-negative matrix factorizations with the spherical K-Means clustering*. PhD thesis, University of Colorado, April 2003. 13, 19, 32

Z. Xiaojun. M. Berry and j. Kogan (eds.): Text mining: applications and theory. *Inf. Retr.*, 14(2):208–211, April 2011. ISSN 1386-4564. doi: 10.1007/s10791-010-9153-5. 23

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3. 28, 33, 50

Y. Xue, C. S. Tong, Y. Chen, and W.S. Chen. Clustering-based initialization for non-negative matrix factorization. *Applied Mathematics and Computation*, pages 525–536, 2008. 19, 22, 32

Y. Yang and Bao-Gang Hu. Pairwise constraints-guided non-negative matrix factorization for document clustering. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pages 250–256, Nov 2007. doi: 10.1109/WI.2007.66. 30

Z. Zhang. Nonnegative matrix factorization: Models, algorithms and applications. In DawnE. Holmes and LakhmiC. Jain, editors, *DATA MINING: Foundations and Intelligent Paradigms*, volume 2, pages 99–134. Springer Berlin Heidelberg, 2011. 2, 10

L. Zhao, G. Zhang, and X. Xu. Facial expression recognition based on pca and nmf. In *Proceedings of the 7th World Congress on Intelligent Control and Automation*, pages 6826–6829. Springer, 2008. 22

Z. Zheng, J. Yangb, and Y. Zhuc. Initialization enhancer for non-negative matrix factorization. *Engineering Applications of Artificial Intelligence*, 20:101–110, 2007. 22

A. Zinovyev, U. Kairov, T. Karpenyuk, and E. Ramanculov. Blind source separation methods for deconvolution of complex signals in cancer biology. *CoRR*, abs/1301.2634, 2013. 15