UCLouvain

École polytechnique de Louvain

Orthogonal Non-negative Matrix Factorization

From ONMF to NN-PCA

Author: **Bruno Losseau**
Supervisors: **Pierre-Antoine Absil, Nicolas Gillis**
Readers: **François Glineur,**
Academic year 2018-2019
Mathematical Engineering

ABSTRACT

Low Rank Matrix Approximation (LRMA) is a well studied problem in the context of Matrix Factorisation (MF). Adding Constraints (CLRMA) makes it possible to extract information from large datasets. We investigate how the combination of the non-negative and orthogonality constraints, the Orthogonal Non-negative Matrix Factorisation ONMF, affects the modelling of the problem, and the link it has with other constraint MF. We show in what way it relates to the NN-PCA model. Based on Pompili's work we focus on the hard-enforcement of the orthogonality to develop algorithms, taking advantage of the problem's geometry to solve it on the Stiefel manifold using a quadratic penalty method. We present the RQNN-PCA algorithm.

Acknowledgments

Though this Master Thesis is my own work, I could not have done it without the support of many.

I would first like to thank my supervisors, Professor Pierre-Antoine Absil and Professor Nicolas Gillis, constantly supportive.
Prof. Absil thank you for your explanations and availability, you always let your door open and welcomed me, preciously answering my questions no matter where they led.
Prof. Gillis, we persevered in making Hangouts work as I really needed your guidance. You would each time suggest papers, datasets, ideas that would bring me back right on track.

I also would like to thank Professor Francois Glineur for accepting to be part of the jury as reader. I hope you will enjoy the discoveries and the direction I took.

During my research I've had many interrogations.
I thank all who took the time to answer them, in particular Prof. Paatero and Prof. Wang.
I thank Becca for accepting to read and comment my approach to k-means and PCA.

This Thesis is the accomplishment of my studies. My first Master year I met Taku and Prof. Papavasiliou, your passion and curiosity in the project we lead were contaminating. I wholeheartedly thank you for making my studies so gripping.

Last but not least, I want to thank my family. You were my very best support. Thank you for all your energy and encouragements.

Contents

List of Figures

List of Algorithms

Abreviations

BSS	Blind Source Separation
CLRMA	Constraint LRMA
EVD	Eigen Value Decomposition
ICA	Independent Component Analysis
LDR	Linear Dimensionality Reduction
LRMA	Low Rank Matrix Approximation
MF	Matrix Factorisation
ML	Machine Learning
LRMA	Low Rank Matrix Approximation
NMF	Non-negative MF
NN-PCA	Nonnegative PCA
ONMF	Orthogonal NMF
OPNMF	Orthogonal Projected NMF
PCA	Principal Component Analysis
PMF	Positive Matrix Factorisation
P-ONPMF-R	Projected Orthogonal Nonnegative Penalised MF on Riemanian manifold algorithm
PNMF	Projected NMF
RALM	Riemanian manifold Augmented Lagrangian Method
REPMS	Riemanian manifold Exact Penalty Method with Smoothing
RQNN-PCA	Riemanian manifold Quadratic Penalty NN-PCA algorithm
SVD	Singular Value Decomposition
VQ	Vector Quantization

(NMF)

$$\underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad ||M-WH||_F^2 \quad \text{s.t} \quad W\geq 0, H\geq 0$$

(ONMF)

$$\underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad ||M-WH||_F^2 \quad \text{s.t} \quad W^TW=I_{k\times k},\ W\geq 0, H\geq 0$$

(OPNMF)

$$\underset{W\in\mathbb{R}^{m\times k}}{\text{minimize}} \quad ||M-WW^TM||_F^2 \quad \text{s.t} \quad W\geq 0,\ W^TW=I$$

(PNMF)

$$\underset{W\in\mathbb{R}^{m\times k}}{\text{minimize}} \quad ||M-WW^TM||_F^2 \quad \text{s.t} \quad W\geq 0$$

(NN-PCA)

$$\underset{W\in\mathbb{R}^{m\times k}}{\max} \quad ||M^TW||_F^2 \quad \text{s.t} \quad W\geq 0,\ W^TW=I$$

(maxPCA)

$$\underset{W\in\mathbb{R}^{m\times k}}{\text{maximize}} \quad ||M^TW||_F^2 \quad \text{s.t} \quad W^TW=I$$

(minPCA)

$$\underset{W\in\mathbb{R}^{m\times k}}{\text{minimize}} \quad ||M-WW^TM||_F^2 \quad \text{s.t} \quad W^TW=I$$

(k-means)

$$\underset{H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad ||M-MH^TH||_F^2 \quad \text{s.t} \quad H=(BB^T)^{-\frac{1}{2}}B,\ B_{ij}=\{0,1\},\ \sum_{i=1}^{k}B_{ij}=1$$

Figure 1: Summary of the models discussed chapter 1

Introduction

Not a long time ago, not in a galaxy far far away...

In 2014, the article "*Two Algorithms for Orthogonal Nonnegative Matrix Factorization with Application to Clustering*" [PGAG14] is published, co-written by my supervisors Prof. Gillis and Prof. Absil, by my reader Prof. Glineur, and by Pompili - a PHD student who visited UCLouvain in 2012. This article introduces a novel approach to build ONMF algorithms.
In 2018 the article has been quoted more than 69 times with most of the recent papers mentioning it. During my research I received from Prof. Gillis the PHD Thesis manuscript of Pompili, "*Nonnegative Matrix Factorization with Orthogonality Constraints: A Geometric Approach*", in which a third algorithm is hidden, the P-ONPMF-R.[Pom12][1] Pompili presented his Thesis back home at the University of Perugia. It is not available on the internet and hardly anyone heard of it. I received it raw with no abstract or any introduction. Yet there has been a lot of research on ONMF algorithms following his work. *Not a long time ago...* Over the last five years a considerable amount of new publications on the subject have appeared. How did the research evolve following his article? Was his hidden algorithm discovered somewhere else? Could it be introduced now to the scientific community?

When choosing the subject on ONMF for my Master Thesis, Professor Absil encouraged me to take a Machine Learning course (LELEC2870), where I was introduced to PCA, ICA, k-means and many others, but there was almost no reference to NMF or ONMF. Working on my Thesis I knew that there are strong bonds between these notions. Simultaneously, on my own initiative I chose a course of medical imaging (LGBIO2050) in which we did tumour identification, segmentation and edge detection. But again, surprisingly, there was still no reference to either of the two models. Yet, as will be demonstrated in this Thesis, ONMF is hiding behind even the simplest hard-clustering task and can be used for such applications. *Not in a galaxy far far away...*

This Master Thesis showcases Pompili's unreleased algorithm P-ONPMF-R (that I changed to OPNMF-R or RQNN-PCA) which follows the OPNMF model. The first chapter presents how NMF with orthogonality constraints surged and

[1]The manuscript is not readily available online and we did not get any authorisation, thus to access it please contact us at bruno losseau @ student uclouvain be

investigates its background. It links the poorly known OPNMF and various well-known machine learning tools. Above all it relates for the first time ONMF to NN-PCA with great detail. It sets the background for the second chapter where we introduce Pompili's hidden algorithm into nowadays's context. After investigating different possibilities, we successfully made it accessible using Manopt, a Matlab toolbox. Finally, in the third chapter, we showcase RQNN-PCA and compare it to the ONPMF using synthetic datasets.

Chapter 1

ONMF in the context of Machine Learning: Dimensionality Reduction and Clustering

In this chapter we position (O-P)NMF in the context of Machine Learning. We apprehend the utility and uses of (O-P)NMF, comparing them to the different models afloat. Example A is used all the way.

We center mainly on introducing and making links between models, using sometimes algorithms to present results in the example A. Second chapter will be more focussing on the algorithms.

The framework employs a historical approach. After introducing section 1 matrix factorisation (MF), we acquaint section 2 Principal Component Analysis (PCA) and Vector Quantization (VQ) as historical techniques of Linear Dimensionality Reduction (LDR) and Clustering.
In the 3rd section we look at how NMF surged in the nineties as it suggests a new paradigm compared to PCA and VQ. We then see in section 4 the reasons why NMF can sometimes be insufficient and how ONMF appeared as a way to enhance the researched effects of NMF. A few concrete applications in different fields are explained.
In the 5th section we bring in PNMF and OPNMF. We proof that ONMF with hard orthogonality enforced is OPNMF (with hard orthogonality enforced), i.e. nonnegative PCA.

1.1 Matrix Factorisation (MF)

Applying standard notations, we assume that data is represented as a matrix $M \in \mathbb{R}^{m \times n}$ where the columns represent the data points in a linear space of dimension m.
In a facetious way we can write down that M contains n column-data points $[m_1, m_2, ..., m_n]$.

Given $k < \min(m, n)$, MF consists in finding two matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ such that

$$M \approx \tilde{M} = W_{m \times k} H_{k \times n} \tag{MF}$$

The difference between the MF and M is considered as an error/noise and is put in $N \in \mathbb{R}^{m \times n}$:

$$M = WH + N$$

We have that

- each column of W is a basis vector of the reduced rank representation,
- each column of H is an encoding and it is a one-to-one correspondence of a data point in M.

The MF of M given k is not unique: fixing any $W \neq 0$ we could find a H and vis versa.

MF is a LDR technique: we project M to W using a linear combination of weights. MF can be understood in the form of a probabilistic hidden variables model [LS99]: the visible data points $M_{:1}$, $M_{:2}, \ldots, M_{:n}$ in the bottom layer of nodes are generated from the hidden variables $W_{:1}$, $W_{:2}, \ldots$, $W_{:k}$ in the top layer of nodes. Each visible variable is represented by a probability distribution with mean $M_{:i} = \sum_{a=1}^{k} W_{:a} H_{ai}$. A common assumption is to admit that the noise N is Gaussian.

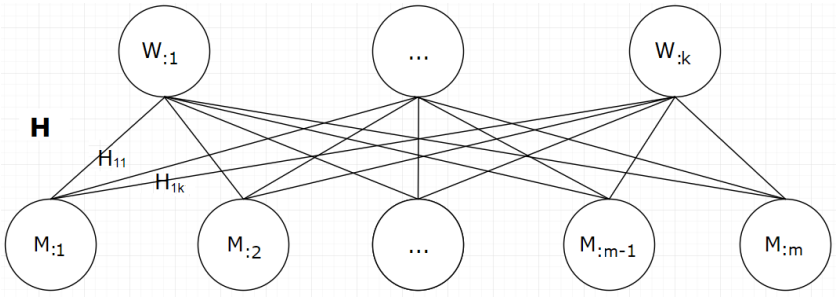

Figure 1.1: Viewing MF as a probabilistic hidden variables model, NMF can be seen as a two-layer directed graphical model with one layer of observed random variables and one layer of hidden random variables

Example A0: MF intuition

MF can drastically reduce the amount of data to stock. Suppose M represents a bank of $n = 1\,000$ grayscale images, each column containing a picture of size $m = 64 \times 64$ pixels. Then M carries $4\,096\,000$ elements. Factorising it with $k = 50$, W and H sum up to a total of $254\,800$ elements.
MF may lead to loss of information. Each data point is taken as a weighted sum of the reference data. When recombining M from W and H, it leads to an approximation of M. The smaller the k the more we might lose information on M.

1.1.1 Definitions

Hermitian matrix A matrix $Q \in \mathbb{R}^{m\times m}$ is hermitian if $Q = Q^T$.

Given $B \in \mathbb{R}^{m\times n}$, if a matrix A is such that $A = BB^T$ then A is hermitian positive semi-definite (p.s.d).

Orthogonality Two non-zero real vectors are said to be orthogonal if their inner product is zero. A square matrix $Q \in \mathbb{R}^{m\times m}$ is said to be orthogonal if $Q^TQ = QQ^T = I$ with I being the identity matrix. A rectangular matrix $Q \in \mathbb{R}^{m\times n}$ is orthogonal if $Q^TQ = I_{n\times n}$ or if $QQ^T = I_{m\times m}$.

Eigenvalue decomposition (EVD) The EVD decomposition of a square matrix $A \in \mathbb{R}^{m\times m}$ is a factorisation written as

$$A = V\Lambda V^T$$

where V is orthogonal and contains the eigenvectors, Λ is diagonal and contains the eigenvalues associated to each eigenvector in descending order. A hermitian p.s.d matrix has positive eigenvalues. The rank of a matrix is the number of non-zero singular values.

Trace It is the sum of all elements on the main diagonal of a matrix. For $A = [a_{ij}] \in \mathbb{R}^{m\times n}$,

$$Tr(A) = \sum_{i=1}^{\min(m,n)} a_{ii}$$

The trace is cyclic: $Tr(XYZ) = Tr(ZXY)$.
If A is square with EVD $A = V\Lambda V$, then as V is orthogonal $Tr(A) = Tr(V\Lambda V^T) = Tr(V^TV\Lambda) = Tr(\Lambda)$.
A matrix and its transpose has the same trace.

Euclidian norms: 2-norm and Frobenius norm The 2-norm of a matrix A is the maximum eigenvalue of the matrix

$$||A||_2 = \lambda_{\max(A)}$$

The Frobenius-norm of A is the extension of the squared Euclidian norm of vectors

$$||A||_F = Tr(A^TA)^{\frac{1}{2}} = \sqrt{\sum_i \sum_j |A_{ij}|^2}$$

If A is square then $||A||_F = Tr(A^TA)^{\frac{1}{2}} = Tr(V\Lambda^TV^TV\Lambda V^T)^{\frac{1}{2}} = Tr(V^TV\Lambda\Lambda)^{\frac{1}{2}} = Tr(\Lambda^2)^{\frac{1}{2}} = (\sum \lambda^2)^{\frac{1}{2}}$. Therefore $||A||_2 \leq ||A||_F$.

The matrix 2-norm and the Frobenius norm are invariant with respect to orthogonal transformation. In the following when not indicated the norm is the Euclidean norm.

Isometry Let X and Y be metric spaces with metrics dX and dY. A map $f : X \to Y$ is called an isometry or distance preserving if for any $a, b \in X$ one has

$$d_Y(f(a), f(b)) = d_X(a, b)$$

Using the Frobenius norm, given A and a transformation matrix R, if $||AR||_F = ||A||_F$ then R is an isometry.

Orthogonal matrix properties

An orthogonal matrix R is necessarily invertible (with inverse $R^{-1} = R^T$) in the reals. The determinant of any square orthogonal matrix is either $+1$ or -1. As a linear transformation, an orthogonal matrix preserves the dot product of vectors, and therefore acts as an isometry of Euclidean space (a distance-preserving transformation between metric spaces), such as a rotation or a reflection.

1.2 Machine Learning (ML) tools

In 1959 A. Samuel coined the expression 'Machine learning' as "programming of a digital computer to behave in a way which, if done by human beings or animals, would be described as involving the process of learning" [Sam59].

This can be narrowed into two types of learning:

- Supervised. The computer is given a training set in which each input element is labelled (associated with the desired output). It can thus learn to recognise how to label the data at hand to classify new data in the future.
- Unsupervised. The computer has only access to the inputs with no example of any desired outputs. The learning algorithm must find by itself a structure in the input to separate the data into clusters. Doing so it can find hidden patterns in the data, make sense "of its own".

Classification is a supervised task whilst clustering is unsupervised. MF is an unsupervised technique. We focus only on the latter.

Historically there are two main topics in unsupervised ML:

- Clustering. Regroups similar points and depicts them by a single one. It maps the n data points to a reduced space of dimension k. Hard clustering is when each point can only be put in one cluster, whereas soft/fuzzy clustering places each column of M in one or more clusters allowing them to overlap
- Dimensionality reduction and feature selection. One and the other reduce the size of the data points from m to k, but they do so differently. A dimensionality reduction creates new combinations of attributes, whereas feature selection includes and excludes attributes in the data without modifying them.

Clustering and dimensionality reduction can both be performed on m or n.

Typically clustering is done by k-means and dimensionality reduction by PCA. We are going to see how they are related to MF, but we first need to present the Singular Value Decomposition (SVD).

1.2.1 The Singular Value Decomposition (SVD)

Introduced in 1873 by E. Beltrami, in 1874 by C. Jordan, the SVD matrix decomposition is one of the most well known and studied MF.

SVD

For $A \in \mathbb{R}^{m\times n}$, there exists orthogonal matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ such that

$$A = U\Sigma V^T, \qquad p = \text{rank}(m,n) \tag{SVD}$$

with

$$\Sigma = \left(\begin{array}{cccc|c} \sigma_1 & 0 & \cdots & 0 & \\ 0 & \sigma_2 & \cdots & 0 & 0_{p\times(n-p)} \\ \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & \sigma_p & \\ \hline \multicolumn{4}{c|}{0_{(m-p)\times p}} & 0_{(m-p)\times(n-p)} \end{array}\right), \qquad \sigma_1 > \sigma_2 > \cdots > \sigma_p > 0$$

.

The σ_i are the singular values of A. The proof using EVD decomposition can be found in [Doo17, section 2.2].
Using (SVD), an approximated (MF) could be written $\tilde{M} = WH$ with $W = U_{m\times k}\Sigma_{k\times k}$ and $H = (V^T)_{k\times n}$. It is a truncated SVD: if $k \geq p$ then $M = \tilde{M}$, else we cannot reconstruct perfectly the matrix M and $M \approx \tilde{M}$. $\tilde{M}$ is called a low-rank matrix approximation (LRMA) when $k \leq p$.
To appreciate the quality of a MF, the Frobenius norm $||M - \tilde{M}||_F^2$ is most often used as it represents the square of the sum of the $(p-k)$ omitted eigenvalues. It has also the advantage of implicitly assuming that the noise N is Gaussian in the probabilistic hidden variables model presented figure 1.1.

SVD and EVD

Knowing the SVD $A = U\Sigma V^T$, then the EVD of the positive-semi-definite matrix AA^T is

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$$

with $\Sigma\Sigma^T$ square and diagonal. Hence U contains the eigenvectors of AA^T and the eigenvalues are σ_i^2. In the same way V contains the eigenvectors of A^TA and the eigenvalues are also σ_i^2. The square roots of the eigenvalues σ_i are named the singular values of A ; thus the eigenvalues are not necessarily positive but the singular values are always positive.

Example B1: SVD in 2D

We consider the 2-D unit circle. Its coordinates can be represented by the unitary matrix I_2. Applying a sheering matrix M we deform the unit circle to an ellipsoid. This can be replicated via 3 transformations: a) an initial rotation V^T that is orthogonal, b) a scaling along the coordinate axis Σ that is a diagonal matrix, c) a final rotation U once again orthogonal.

Example B2: orthonormal matrix transformation and Frobenius norm

For $A \in \mathbb{R}^{m \times n}$, we compute its reduced rank SVD: $A = U_{m \times k} \Sigma_{k \times k} V_{k \times n}$ for any $k \leq m, n$. As U and V are orthogonal, $U^T A$ and AV^T are isometries, thus

$$||A||_F = ||U^T A||_F = ||AV^T||_F$$

.

1.2.2 Principal Component Analysis (PCA): the Swiss Army Knife of data analysis

PCA is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. PCA has been widely used for many years in Machine Learning and it is so convenient to employ that in [VL18] it is described as the 'Swiss Army Knife of data analysis'. Introduced by Karl Pearson in 1901 [Pea01], the central idea of PCA is to make sense out of datasets based on a statistical viewpoint: it reduces the number of dimensions of many interrelated variables, while keeping a maximum variation present in the set [Jol01].

Mathematical approach to PCA

PCA is defined as an orthogonal linear transformation that expresses data in a new coordinate system such that the greatest variance comes to lie on the first coordinate (the Principal Component), the second greatest variance on the second coordinate (the second PC), and so on. In the new coordinate system the variables are decorrelated and only the ones containing the most information (the PC) are kept [VL18]. The full principal components decomposition $H_{m \times n}$ can be written, after the centering of M (subtracting to each sample its mean):

$$H = W^T M \tag{1.1}$$

where W is a $m \times m$ matrix of weights whose columns are the eigenvectors of the EVD $MM^T = W^T \Lambda W$.
Using the link between SVD and EVD, for the SVD decomposition seen previous subsection $M = U \Sigma V^T$ we also have $MM^T = U \Sigma \Sigma^T U^T$ where $\Sigma \Sigma^T = \Lambda$ and $U = W^T$. Thus we can use the SVD decomposition on the centered M to find the full PCA.
To only keep the k first principal components, we can select the k first columns of W and compute H of size $k \times m$. By removing columns of W it looses its orthogonality one of the two ways, only the property $W^T W = I_{k \times k}$ is kept.

PCA always considers that matrix M is centered, each observation has a zero mean.

PCA and Factor Analysis

Factor Analysis (FA) emerged at about the same time as PCA. It is found many times in the litterature at the beginings of NMF. Jolliffe explains the difference

between the two in [Jol01]. PCA involves extracting linear composites of observed variables. On the other hand FA is based on a formal model predicting observed variables from theoretical latent factors.

PCA and reconstruction, unicity

PCA keeps the most information possible of M, describing all the observations it contains in a reduced features space, mapping the matrix from $M_{m\times n}$ to a reduced matrix of size $H_{k\times n}$. If we want to reconstruct an approximation $\hat{M}$ of M from W and H, as W is always invertible (being an orthogonal matrix) and as its inverse is equal to its transpose, we have:

$$\hat{M} = \left[W^T\right]^{-1} H = \left[W^T\right]^T H = WH = WW^T M \tag{1.2}$$

This can be seen as an auto-encoder, where the first transformation is not an isometry but the second is. The proof is straightforward:

$$||W^T M||_F^2 = Tr(W^T M M^T W) = Tr(W W^T M M^T)$$

$$\begin{aligned} ||WH||_F^2 = Tr(WHH^T W^T) &= Tr(W^T W H H^T) \\ &= Tr(W^T W H H^T) = ||HH^T||_F = ||H||_F^2 \end{aligned}$$

A visualisation is made figure 1.2.

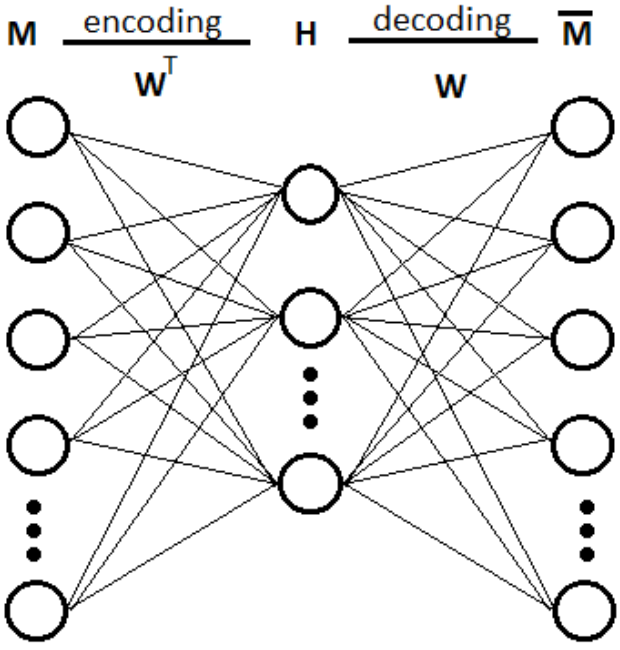

Figure 1.2: Viewing PCA as a linear autoencoder.

In the encoding we preserve as much variance as possible to store in a reduced space the information contained in M; with the decoding we can only represent M based on the information that is stored in H.

Understanding PCA using figure 1.1 can be confusing. This representation means setting the weights H as a combination of the visible data points and the hidden variables.

The LRMA obtained with PCA is unique as there exists only one $\Sigma_{k\times k}$.

Approach to the general model

There are a large number of PCA models. Mandel (1972) considers the retention of m PCs, based on the SVD, as already implicitly using a model [Jol01]. As it minimizes the reconstruction error by definition, the PCA solves the following LRMA problem, with M centered:

$$\begin{aligned} \underset{\tilde{M}}{\text{minimize}} \quad & ||M-\tilde{M}||_F^2 \\ \text{s.t} \quad & \text{rank}(\tilde{M}) = k \end{aligned}$$

It is equivalent to the unconstrained problem (as explained in [Gil17], "*using the standard least-square error leads to (PCA) that can be solved by using the singular value decomposition (SVD)*"):

$$\underset{W\in\mathbb{R}^{m\times k}, H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad ||M-WH||_F^2 \tag{1.3}$$

We emphasis that the unconstrained solution obtained with PCA results in columns of W that are orthonormal and columns of H that are orthogonal. We did see (1.2) that the solution of the problem is the same as when setting $H = W^TM$:

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k}}{\text{minimize}} \quad & ||M-WW^TM||_F^2 \\ \text{s.t} \quad & W^TW = I \end{aligned} \tag{minPCA}$$

The solution W is the eigenvectors of the data covariance matrix.

Using the definition of the Frobenius norm, the trace properties and $W^TW = I$, we have:

$$\begin{aligned} ||M-WW^TM||^2 &= \text{tr}[(M-WW^TM)(M^T-M^TWW^T)] \\ &= \text{tr}[MM^T] - 2\text{tr}[W^TMM^TW] + \text{tr}[W^TWW^TMM^TW] \\ &= \text{tr}[MM^T] - \text{tr}[W^TMM^TW] \end{aligned} \tag{1.4}$$

Thus (minPCA) can be written as a maximization problem, that is the standard model for PCA:

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k}}{\text{maximize}} \quad & ||M^TW||_F^2 \\ \text{s.t} \quad & W^TW = I \end{aligned} \tag{maxPCA}$$

Based on the orthogonal matrix properties (see section 1.1), this form enhances the following interpretation: PCA consists in obtaining a linear transformation close to an isometry, that maps M to a reduced dimensional space such that the maximum of information in M is maintained.

In the equations (minPCA) and (maxPCA), PCA aims at maximising the encoding of M to a space of reduced dimension, which is the same as minimising the error after decoding.

We can find the unique global minimum in polynomial time.

1.2.3 Clustering seen with Vector Quantization (VQ)

VQ was introduced by C. Shannon in the 1950s in the field of signal processing to solve the problem of analog to digital conversion. Quantization is the process of mapping input values from a large set (often continuous) to output values in a (countable) smaller set [VL18]. It is a clustering task: we want to store the n vectors represented by the columns of M by $k < n$ vectors described by the columns of W. An illustration is shown figure 1.3.

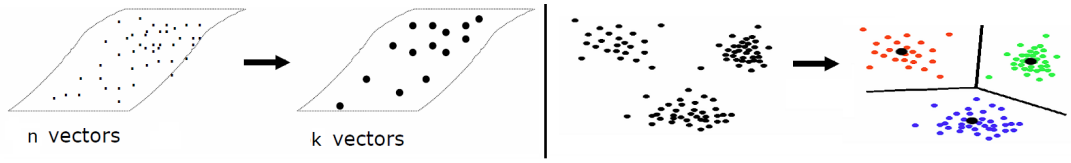

Figure 1.3: (Left) Projecting an input space into a discrete smaller output space. (right) The raw data in which each point represents a column of M is regrouped into 3 clusters, each being represented by a centroid and a Voronoi zone (in black). The centroids are the columns of W. [VL18]

Many methods exist to apply VQ, the most used unsupervised algorithm is the k-means that is a hard VQ. The goal is to minimise:

$$\sum_{i=1}^{k}\sum_{j=1}^{n} B_{ij}||M_{:j} - W_{:i}||_2^2 \tag{1.5}$$

with B a matrix of binary indicator variables such that $B_{ij} = 1$ if $M_{:j}$ is in the cluster of centroid $W_{:i}$. Every column j of B sums to 1 and each row i sums to C_i, the number of elements in each cluster :

$$\sum_{i=1}^{k} B_{ij} = 1 \quad \forall j \in 1 \dots n \qquad\qquad \sum_{j=1}^{n} B_{ij} = C_i \quad \forall i \in 1 \dots k$$

As we proceed to hard VQ, the rows $B_{:}j$ are orthogonal and BB^T is diagonal with on its diagonal the number of elements in each cluster.

The most common way to solve is as follows. At initialisation k vectors are arbitrarily selected as centroids. Each vector is then assigned a cluster represented by the nearest (in the sense of Euclidean distance) centroid. After that, each centroid is updated as the centre of mass (the mean) of all its cluster members. This is repeated as long as there are changes in the update of the centroids. It can be seen as a special case of the Expectation-Maximisation (EM) scheme using a centroid model with no variance and hard assignments, when the data is generated by a Gaussian distribution :

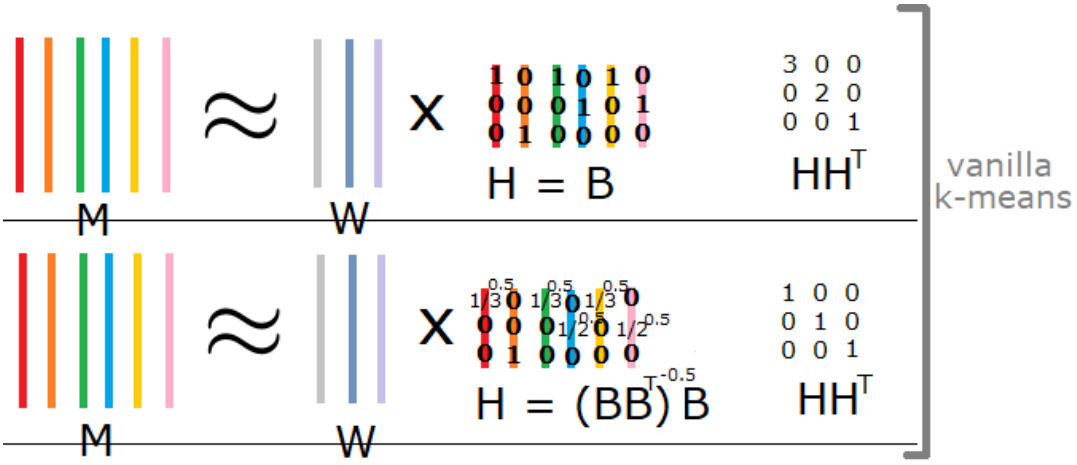

Figure 1.4: For $m = 6$ data points of dimension n, we cluster them using $k = 3$ centroids.

- The E-step, distribution estimation, update of B. Each column of M is assigned to one centroid.
- The M-step, parameter estimation, update of W. Each centroid is computed as the centre of mass of the points in the cluster.

VQ model for hard clustering seen as MF

It was first uncovered in 2005 [DHS05] that there exists an equivalence between the k-means algorithm and a constrained MF problem. In 2015 C. Bauckhage established rigorously the link between k-means clustering (1.5) and MF [Bau15]. The k-means solution satisfies:

$$\begin{aligned} &||M - MB^T(BB^T)^{-1}B||_F^2 \\ \text{with} \quad &B_{ij} = \{0, 1\} \\ &\sum_{i=1}^{k} B_{ij} = 1 \end{aligned} \tag{1.6}$$

This is a Constrained Low Rank Matrix Approximation (CLRMA).

We present an example figure 1.4, with the following explanation.

1. At first row, setting $H = B$ and $W = MB^T(BB^T)^{-1}$, we have HH^T diagonal with on its diagonal the number of elements in each cluster (it is not the case for H^TH).
2. At second row, setting $H = (BB^T)^{-\frac{1}{2}}B$ the scaled indicator matrix, the

problem can be written as:

$$\begin{aligned} \underset{H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad & ||M - MH^TH||_F^2 \\ \text{s.t} \quad & H = (BB^T)^{-\frac{1}{2}}B \\ & B_{ij} = \{0,1\} \\ & \sum_{i=1}^{k} B_{ij} = 1 \end{aligned} \tag{k-means}$$

- $W = MH^T$ reflects that each centroids $W_{:i}$ coincide with the mean (centre of mass) of all the $M_{:j}$ in the corresponding cluster.
- $HH^T = I$, H is an orthogonal matrix that is built from B.

We see that fixing H as the scaled indicator matrix it is orthogonal, but not all orthogonal matrix are scaled indicator matrix.

K-means clustering is a non convex np-hard problem. As finding a global minimum is not possible, we are reduced to search for a local minimum in polynomial time. To solve the EM strategy is most commonly used: we work out the problem with respect to H then W iteratively, each sub-problem being convex. The clustering might depend on the initialisation of the centroids.

1.2.4 Link between k-means and PCA

Comparing the objective functions in models (minPCA) and (k-means), we see that we can pass from one to the other simply by using the transpose:

$\left[MH^TH\right]^T = H^THM^T$, by posing $W = H^T$ we obtain $\left[MH^TH\right]^T = WW^TM^T$.

Figure 1.2 can then be associated to k-means. This seems reasonable in the sense that originaly PCA performs a dimensionality reduction diminishing the size m and k-means performs clustering reducing the size n of matrix M. Asking PCA to perform 'clustering' or k-means to perform 'dimensionality reduction' is in that way possible. However note that PCA always centers the data and not k-means.

From this reasoning, in a MF aspect, the two models (minPCA) and (k-means) differ only by the constraints added to the objective function. Both have an orthogonality constraint, k-means has constraints on top. In MF litterature, because with clustering we want to emphasis the constraints on the cluster indicators we prefer to use the H-factorisation setting $W = MH^T$, and because with PCA we want to point out the basis matrix we always employ the W-factorisation setting $H = W^TM$. In the remaining of the chapter we will separate the two matrix factorisations as they imply different interpretation, at the end of the chapter we will merge them.

1.2.5 Example A.1 - Image processing

Applying dimensionality reduction and clustering to face recognition

Comparing the VQ model to the PCA model, the two use the same Frobenius norm and thus their objective value can be compared. The minimum for PCA will in

general be smaller than the minimum for VQ because of the constraint restrictions. However as we are going to see with this example, the meaning of the obtained output W with PCA will be less obvious then with k-means.

As inspiration, the references [TP91, Fac94] have been used. We employ the CBCL, (MIT Center For Biological and Computation Learning) bank of $n = 2429$ grey-level 19-by-19 pixels images containing different faces. Given a new image of the face of someone known in the bank, the objective is to match it to the right label.

Each image A_j contains $D \times D$ grey-scale pixels that can be reorganised as a D^2-dimensional vector such that the lines of pixels are placed one after another in a 1-D vector

$$A_j^T = (a_1, a_2, a_3, \ldots, a_{D^2})$$

As $D = 19$, each A_j has 381 dimensions. We can merge all the images in a $D^2 \times n$ matrix M such that:

$$M = (A_1, A_2, \ldots, A_n)$$

We solve the problems for $k = 16$, using PCA and k-means as a clustering, diminishing the size n of M.

"Clustering" with PCA

A visual illustration is given figure 1.5. Using the interpretation of figure 1.1 for M^T instead of M:

- W contains the basis images, a set of "eigenfaces" that characterizes the variations between all the images. The columns of H represent the linear combination of the eigenfaces to reconstruct the original images.
- We can approximate any new image A_l by a linear combination of the eigenfaces $A_l \approx \tilde{A}_l = WH_l$. Comparing the weights associated to the new image H_l with those in the bank $H_{:i}$, $i \in 1 \ldots m$, we can match the new image to the right label by finding H_l such that it satisfies $\min_{H_{:i}} ||H_{:l} - H_{:i}||$.

The eigenfaces (output W) are 'ugly': the input contains only values between 0 and 255, the output contains values that are positive and negative that make no intuitive sense. The eigenfaces are combined using positive and negative weights to re-form the original faces, these weights have no meaning for a human. Nonetheless they are efficient as the face recognition works and all faces in the database can be reconstructed at best.

Clustering with k-means

Using k centroids, we map the n images to k images.

- The k images in the low dimensional space are this time prototypical faces instead of eigenfaces : the faces with similar characteristics are regrouped. As the matrix B is constrained to be unary column-wise (hard-clustering), then each image of the database is assigned a single prototype.

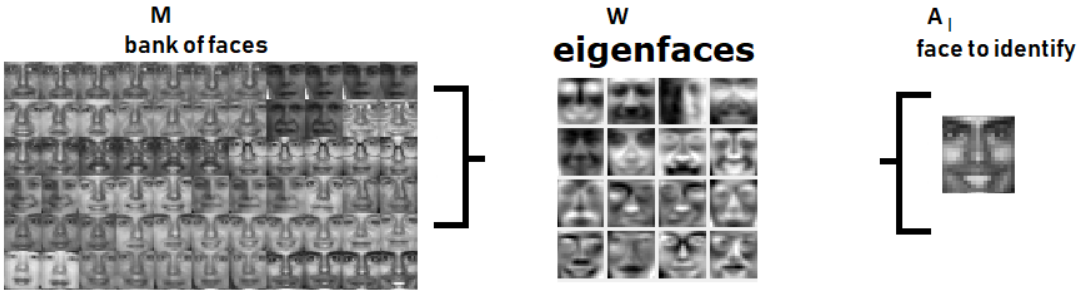

Figure 1.5: Illustration of example A.1 using the CBCL face database

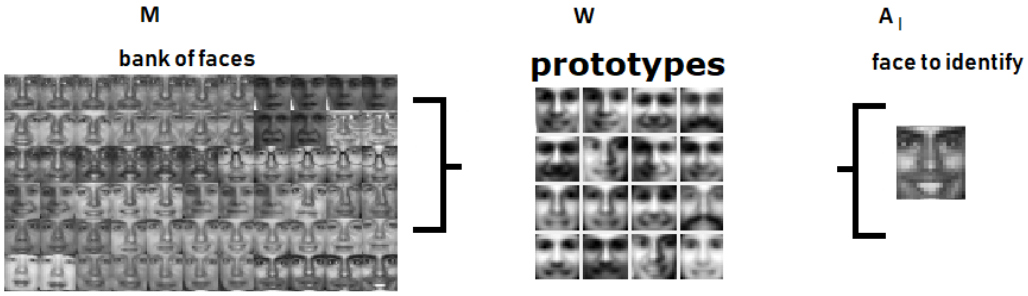

Figure 1.6: Illustration of example A.2 using the CBCL face database

- When introducing a new image A_l containing a known face, we could then associate it to its prototype and find which cluster of images it belongs to, but we would not be able to match it to the corresponding image in M.

With hard k-means we obtain figure 1.6 prototypical faces $W_{:i}$ that are images with positive values, the associated weights H to reconstruct the original faces are binary. In contrary to PCA, this makes the interpretation of W meaningfull for human beings. But the face recognition is not efficient, neither is the reconstruction.

Conclusion

We have discovered how both PCA and k-means can be expressed as MF. Each one possesses different properties that makes the applications and interpretations of the factorisation different. PCA can be seen as the 'best' factorisation as it finds the global minimum of the Frobenius norm in polynomial time, its minimum will be smaller than the minimum for k-means because of the constraints restrictions. k-means Allows to organise the data in clusters thus making possible a physical interpretation of W and H, the nature of the columns of W will be the same as the columns of M.

1.3 The rise of NMF

Often in data analysis the information contained in M is positive, such as in example A. With the PCA model the MF can be computed in polynomial time as an unconstrained problem but it does not lead to a representation such that W and H are positive, making it hard to interpret and give a meaning to the hidden variables. What happens to the matrix factorization if we enforce its non-negativity ? Starting from here we only consider the case where M is non-negative[1].

Non Negative Matrix Factorization

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad & ||M-WH||_F^2 \\ \text{s.t} \quad & W\geq 0, H\geq 0 \end{aligned} \tag{NMF}$$

The goal of this master thesis is to study NMF with orthogonality constraints. Orthogonality can be imposed on H, on W or both.

H-Orthogonal Non-Negative Matrix Factorisation

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad & ||M-WH||_F^2 \\ \text{s.t} \quad & HH^T = I_{k\times k} \\ & W\geq 0, H\geq 0 \end{aligned} \tag{H-ONMF}$$

W-Orthogonal Non-Negative Matrix Factorisation

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad & ||M-WH||_F^2 \\ \text{s.t} \quad & W^TW = I_{k\times k} \\ & W\geq 0, H\geq 0 \end{aligned} \tag{ONMF}$$

WH-Orthogonal Non-Negative Matrix Factorisation

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad & ||M-WH||_F^2 \\ \text{s.t} \quad & W^TW = I_{k\times k},\ HH^T = I_{k\times k} \\ & W\geq 0, H\geq 0 \end{aligned} \tag{WH-ONMF}$$

PCA dates from 1901, VQ from 1951. We had to wait (1999 and) 2001 to get NMF popularised. In the following we see how NMF and then ONMF appeared.

[1]The reader might be interested in the difference between non-negative and positive. The first implies that it can be zero, the second implies that it is strictly positive. Thus the difference is 'only' zero!

1.3.1 The human brain inspiration : learning by parts

In 1999 Lee and Seung publish the article "Learning the parts of objects by non-negative matrix factorization" [LS99]. In 2001 they present in details two algorithms to solve NMF [LS01]. The basis of their work is that there is psychological and cognitive evidence that in the brain "*perception of the whole is based on perception of its parts*".
Following example A1, both k-means and PCA use holistic representations of faces : the columns of W contain whole faces. To mimic the way the brain works the authors suggest to constrain W such that its columns contain parts of faces. NMF only allows nonnegative entries in the factorisation of every image i $M_{:i} \approx W_{:i}H_{i:}$, forcing additive combination, which is compatible with the intuition of combining parts to form a whole. Based on figure 1.1, if we constrain the weights in the MF of image i $H_{i:}$ and all W to be positive, to reconstruct the image i we can only take a weighted positive sum of the k images in W, forcing W to be a part-based representation.

Example A.2

In [LS99], the authors compare k-means and PCA to NMF based on the same dataset as in example A1. The results using their NMF algorithm are presented figure 1.7. We see that W is a part-based image representation and not a holistic one. The values stored in W are positive but they are not constrained to be in [0,255]. We saw with k-means that trying to identify a new image we only could place it in one cluster whereas here it can be associated to many clusters. With NMF face recognition, which works in a similar way to PCA, we can recompose the images in M based on W using the encoding matrix H. It can therefore be seen as both non-negative dimensionality reduction and soft clustering.

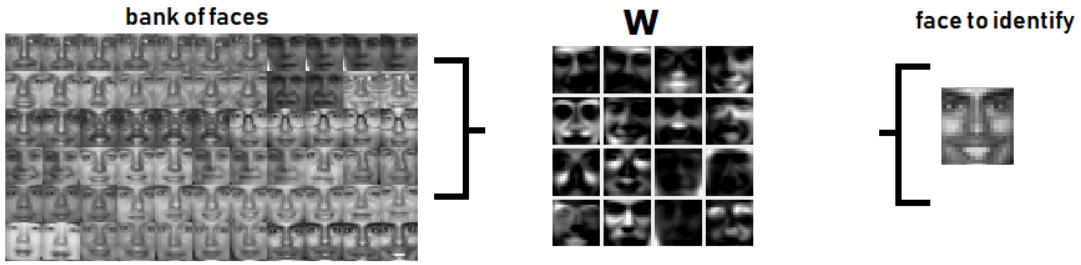

Figure 1.7: Illustration of example A with NMF as presented by Lee and Seung using the CBCL face database

Lee and Seung [LS99] compare the 3 methods (k-means, PCA, NMF) and show that NMF basis and encoding contain a large fraction of vanishing coefficients making both W and H sparse in contrast to the unary encoding of k-means and the fully distributed PCA encoding.

The authors model the problem using the Frobenius norm :

$$\begin{aligned} \underset{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}}{\text{minimize}} \quad & \sum_{i}^{m}\sum_{j}^{n}[M_{ij}-(WH)_{ij}]^2 \\ \text{s.t} \quad & W\geq 0, H\geq 0 \text{ element wise} \end{aligned} \tag{1.7}$$

NMF is non-convex but fixing H and solving for W (and vis-versa) turns the problem convex. Thus they suggest to implement a two-block coordinate update rule. W and H are randomly initialised as there are many possible non-negative factorisations. The authors state that the iterations converge to a local minimum. Launching many times the algorithm we might fall on different local minima.

We can reason that it has the same logic as the k-mean algorithm. They present the details in [LS01], suggesting to apply multiplicative updates (we will see how this has affected orthogonality constraints up to now in the domain of ONMF).

The two publications of Lee and Seung have made NMF very popular. Article [LS99] has shown the scientific community two applications (parts of face learning and semantic features of text) and asserts that NMF has a very broad range of potential applications. Giving a simple efficient way to solve NMF in [LS01], they brought much enthusiasm: [LS99] has more than 9 000 quotes and [LS01] has around 7 000 quotes. After this contribution they continued in their research fields, building robots or Artificial Intelligence that mimics the humans. In July 2018 they both joined Samsung AI R&D. They did not publish any other significant paper on NMF, but they have left a legacy.

1.3.2 NMF first appearance

The first studies on NMF were performed to find a reconstruction error N equal to zero, obtaining an equivalent like SVD decomposition that would be nonnegative, known as Non-negative Rank Factorization. For instance in 1973 a proof was written by A. Berman, L.B Thomas and A. Ben-Israel [BTBI73] stating that for any positive matrix M of rank r there exists a trivial non-negative factorisation where H is of rank r and W has r columns. There is still research on this topic [VGG18, generalisation of the problem]. The goal of (NMF) is however different : we search for a factorisation given a k that is smaller than r.

It is only between 1991 and 1994 that a similar problem to the NMF was formally introduced by P. Paatero and U. Tapper in the context of chemometrics [PT94]. From [Hé08], chemometrics aims to look at the data as a whole to find a relationship that explains a given situation. For example, it is now used for identifying the source contribution of fine particles of air pollution.

Funnily, they named their method Positive Matrix Factorisation and did not change it ([BEP15] in 2015) making chemometrics the only field in which a certain NMF is called PMF [2]. As for Lee and Seung [LS99], they start from the PCA

[2] Quoting an exchange we had with P. Paatero "original algorithm for PMF used logarithmic positivity constraints so that the factor elements were strictly positive, even if practically =0", thus the name.

explaining that it "*has been very successful in such fields as social sciences and psychometry. However, in physical sciences they have found only limited application.*" They insist that non-negativity is an intrinsic property in many environmental data and that PCA or FA that was accessible to them gave a matrix W that didn't possess this characteristic. At that time some 'elliptical' methods were suggested in the literature to add rotation matrix (orthogonal matrix) to the SVD decomposition and make the factorisation positive (see example B2). Paatero and Tapper were not convinced by this approach. They introduced the new procedure NMF (PMF)[3].

The difference between PCA and PMF is that PMF is a FA method, one of the hypothesis of the model is non-negativity. As for PCA and in contrary to the general NMF it finds a unique solution as it tries to obtain the (hopefully unique) model of a dataset that describes a physical/chemical situation, whereas NMF only wants to compress the data in non-negative elements, thus obtaining parts [4].

They achieved convincing results. In 2016 Paatero was awarded the Clarkson University honorary degree, part of the reason was for "transforming data into clear rationales for environmental policy and action". PMF has been applied recently to lake sediments, wastewater and soils. Article [PT94] has been quoted 3 000 times, way less then the two articles published by Lee and Seung [LS99]. We think the reason for this is that Paatero restricted the discovery to his own field of research, whereas Lee and Seung made it accessible to all. They used the PMF articles as inspiration for their NMF.

1.4 When NMF fails to learn by parts : introducing the ONMF

(NMF) received the attention of the scientific community for its strength of being able to learn by parts. In the encoding matrix a zero-value represents the absence of an event/component and a positive-value the presence of that event/component. It can however still result in holistic part-based representations.

Counter-example A.3, NMF can generate holistic representations.

Following example A2, in 2001 Li et. al. [LHZ01] ran the test on another face image database (the ORL database, 400 images of size 92×112, the CBCL database contained images of size 19×19). Using $k = 16$ we obtain the basis vectors shown figure 1.8.

The hidden faces are not intuitive and are not given as a part-based representation. The searchers explained that it was because the images in the ORL face database are not well hand aligned.

[3] Quoting the same exchange, P. Paatero said that "Almost similar non-negative models had been used by chemometricians before me, e.g. under the name of 'alternating regression' not as good, PMF was the first time it was solid".

[4] In regards to the pre-processing of the data in our NN-PCA model to come, we could get inspiration from the PMF field as it has been fully investigated how centering or normalising or other transformations affect the problem.

Figure 1.8: Illustration of example A using the ORL face database, the faces contained in W. NMF is used but does not give convincing results

The authors acknowledged that "*the components are combined to form a whole in the non-subtractive way*". But "*the additive parts learned by NMF are not necessarily localized*". The searchers suggest to enforce sparsity applying a "local" LNMF, pushing W and H towards orthogonality and putting the constraints in the objective function using the Lagrangian representation.

1.4.1 Enforcing sparsity and local features in NMF using orthogonality constraints

Sparseness given by NMF is somewhat a side-effect rather than a goal, one cannot in any way control the degree to which the representation is sparse. Orthogonality constraints are a way to obtain that property. The idea to use orthogonal matrix can first be found in 2001 in [LHZ01], they make the observation that PCA enforces W and H orthogonal, when NMF asserts the non-negativity. Adding orthogonality to the columns of W makes the basis vectors such that they are sparse and redundancy is minimised, the basis vectors disjoint. Adding orthogonality to the columns of H we obtain an approach similar to k-means clustering.

Example A.4

We apply models (H-ONMF), W-(ONMF), (WH-ONMF) to M, the 25 faces stored in W are presented figure 1.9 using the Matlab toolbox NMFTOOL[5]. We see that the part-based representation claimed by Lee and Seung can be obtained by restricting W to be orthogonal. We can make the following observations :

- (H-ONMF). As $HH^T = I_{k\times k}$, each column of H can contain only one positive value, the lines are disjoint. The values on each column are not necessarily all equal. H is a hard-clustering matrix associating 1 image to 1 prototype. We show the matrix W, we obtain holistic images as with k-means.
- W-(ONMF). As $W^TW = I_{k\times k}$, each line of W can contain only one positive value, the columns are disjoint. The values on each line are not necessarily all equal. Each prototype image of W is forced to store only a few pixels, these

[5]available at https://sites.google.com/site/nmftool/

Figure 1.9: Illustration of example A using the ORL face database, faces contained in W. Different ONMF are applied : top is model (H-ONMF), bottom is model W-(ONMF).

cannot be found in the other prototype images. A part-based representation is obtained.

- (WH-ONMF). As $HH^T = I_{k\times k}$ and $W^TW = I_{k\times k}$, the two previously observed effects should be enhanced but the constraints are too tight and the algorithms fail to obtain a result thus we do not show it. The most common way to enhance it is to use a tri-factorisation $M \approx WUH$ where W and H are orthogonal and U is free. We could make a parallel with SVD decomposition.

For the concrete application of example A the W-ONMF works when the H-ONMF doesn't. This seems legitimate because we want the matrix W to be separated into parts thus imposing orthogonality to it makes it a part-base representation. The W-ONMF is performing a clustering separating the parts of the faces. The H-ONMF instead insists on the clustering indicator and thus stays in a k-mean perspective of holistic faces.

1.4.2 Applications of NMF

NMF and the Blind Source Separation (BSS) problem

The BSS problem consists in extracting a small set of sources that explain most of the data : finding the "building blocks". It is considered as a dimensionality reduction problem. To solve the BSS problem most often a statistical Indepent Component Analysis (ICA) is applied, it can be linear or not. In PCA we were asking each column to be uncorelated, in ICA we ask each column of W to be independent. ICA can be seen as the minimisation of mutual information between

the transformed variables (the source) [Hyv97]. When the problem is linear and positive an alternative is to use the non statistical approach that is NMF. In 2014 A. Mirzal published a paper (it was republished in 2017 with almost no difference) comparing the two [Mir14], applying on four datasets 10 different NMF (among which the ONMF) and 3 different ICA. He emphasizes that ICA possesses uniqueness when NMF doesn't and thus we need to choose the representative reconstruction. But NMF gives in terms of reconstruction error similar results to ICA and when the number of sources increase it outperforms ICA. Also the results obtained are visually very satisfying. In 2017 a ML course [Par17] compares the results for a movie containing a series of 126 frames in which a human gives a signal. The results are shown figure 1.10, ICA identifies that the hands are moving, NMF identifies the 2 movements.

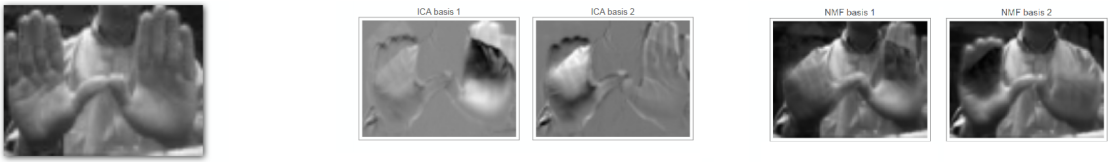

Figure 1.10: A movie (left) is analysed through ICA (center) and NMF (right). NMF gives the best representation to interpret the different sources. Images from reference [Par17].

As an example, in 2018 researchers found out that NMF outperforms ICA for identifying the pigment mixture of ancient chinese paintings [LLHY18] using hyperspectral images, making NMF possible in a near future to be used for painting conservation and restoration.

The reference in the domain of NMF for BSS is [CZPA09], quoted more than 1 600 times since 2009.

NMF and hyperspectral unmixing

My supervisor Nicolas Gillis is a specialist of hyperspectral unmixing using NMF. Based on the separable NMF assumption, with sparse NMF or with rank 2 NMF and hierarchical decomposition he manages to obtain amazing results. I could not include this in my thesis. Hyperspectral unmixing is highly related to orthogonality, as we will present an application in the third chapter.

NMF and text clustering

With M a term-by-term document matrix, the entries are the frequency of occurrence of each word inside each document. The basis elements obtained gather co-occuring words in the semantic space. NMF is an alternative to Latent Semantic Analysis (LSI) that is based on SVD decomposition, NMF is more interpretable than LSI.

1.4.3 Euclidian ONMF and k-means

ONMF : trade-off between orthogonality and non-negativity

We have seen that the orthogonality constraint can be added to the NMF problem to make the solution sparser and obtain a part-based representation in W. As the main goal of the architects behind the algorithms of ONMF is to obtain an interpretable NMF, and as the orthogonality constraint collides with the non-negativity constraint, most often the ONMF algorithms enforce strictly non-negativity and tend loosely impose the orthogonality.

However strictly enforcing orthogonality has a hard clustering effect. We study the link between ONMF with strict orthogonality enforced and k-means.

k-means as a particular case of ONMF

The (k-means) model constraints impose H to be non-negative (derived from a binary matrix). This can only lead to a matrix W positive too when M is positive and thus it results by design in NMF. Also, we saw that writing $H = (BB^T)^{-\frac{1}{2}}B$ it also respects $HH^T = I_{k\times k}$. Therefore it can be said that the k-means model is the H-ONMF with additional constraints. NMF with a weighted, binary, orthogonality constraint on H is equivalent to Euclidean k-means [PGAG14].

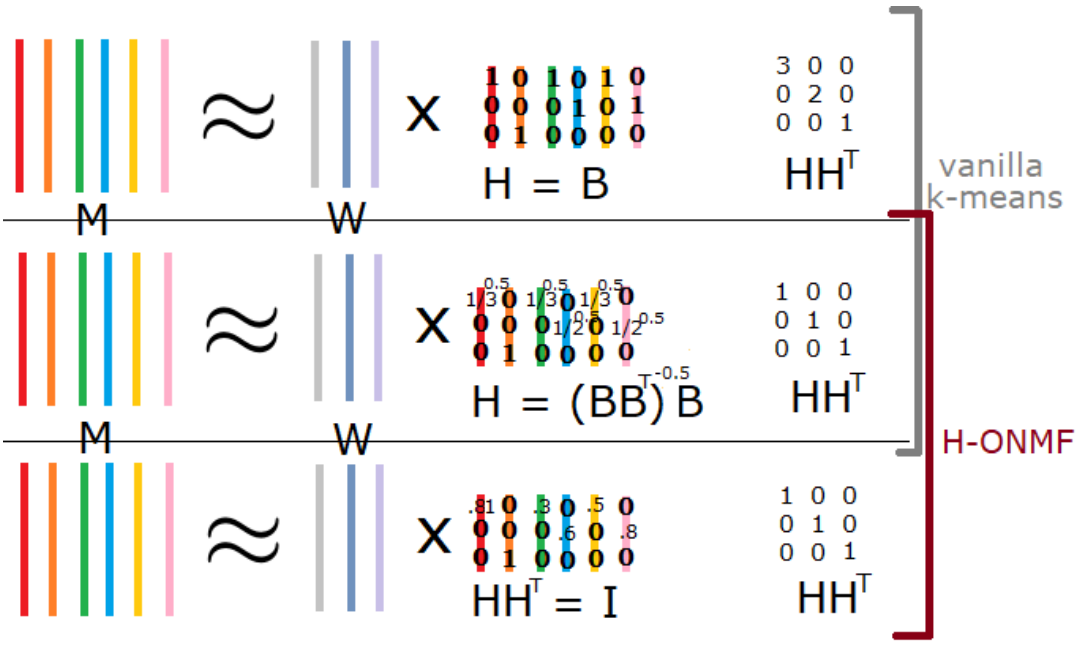

Figure 1.11: For $m = 6$ data points of dimension n, we cluster them using $k = 3$ centroids. We see that the difference between k-means and H-ONMF is that with H-ONMF the rows of H are not required to have their non-zero entries equal.

Figure 1.4 is completed by ONMF figure 1.11. We observe that with k-means the red, green and yellow (1st, 3rd and 5th) columns of M are associated to the

grey (1st) column of W given the coefficients in W. With ONMF it is the same except that the weights in each rown of W are not any more necessarily equal.

NMF, H-ONMF and spherical K-means

Pompili proved in [PGAG14] that imposing orthogonality instead of the binary assignment "*steers the problem from a pure k-means setting towards a mixed Euclidean and spherical variant of it, towards a more directional variant which still retains a dependence between data points*".

Vanilla K-means uses the Euclidean norm. With the same notations as in (1.5), if instead we apply the cosine similarity we obtain the spherical k-means algorithm that minimises, with $B_{k\times n}$ a matrix of binary indicator variables containing a single value per column:

$$\min_{B,W} \sum_{i=1}^{k}\sum_{j=1}^{n} B_{ij}(1-\cos(M_{:j},W_{:i})) \quad = \quad \min_{B,W} \quad \sum_{i=1}^{k}\sum_{j=1}^{n} B_{ij}\left(1-\frac{M_{:j}^T W_{:i}}{||M_{:j}||\,||W_{:i}||}\right) \tag{1.8}$$

Normalising the columns of W $||W_{:i}||=1$, (1.8) can also be written

$$\max_{B,||W_{:a}||=1} \sum_{i=1}^{k}\sum_{j=1}^{n} B_{ij}\left(\frac{M_{:j}^T}{||M_{:j}||}W_{:i}\right) \quad = \quad \max_{B,||W_{:a}||=1} \sum_{i=1}^{k}\sum_{j=1}^{n} B_{ij}\cos(M_{:j},W_{:i}) \tag{1.9}$$

In [PGAG14] they proove that H-ONMF is equivalent to:

$$\begin{aligned}\max_{B,||W_{:a}||=1,W>0} \sum_{i=1}^{k}\sum_{j=1}^{m} B_{ij}\left(M_{:j}^T W_{:i}\right)^2 &= \max_{B,||W_{:a}||=1,W>0} \sum_{i=1}^{k}\sum_{j=1}^{m} B_{ij}||M_{:j}||^2\left(\frac{M_{:j}^T}{||M_{:j}||}W_{:i}\right)^2 \\ &= \max_{B,||W_{:a}||=1,W>0} \sum_{i=1}^{k}\sum_{j=1}^{m} B_{ij}||M_{:j}||^2\cos^2(M_{:j},W_{:i})\end{aligned} \tag{1.10}$$

To obtain ONMF under this maximisation formula, they use formula 2.6 of their paper an argument that is very close to the one we will develop (1.13).

The two can then be compared. They both maximise the cosine of the angles between each centroid and the data points linked to it, making them invariant to scaling (contrarily to vanilla k-means). The differences are :

- (1.10) has $\sum_{j=1}^{k}||M_{:j}||^2$ when (1.9) doesn't, making ONMF sensitive to the norm of the data points when spherical k-means is only sensible to their direction.
- (1.10) maximises the sum-of-squares of the cosines when (1.9) maximises their sum.

The paper [PGAG14] explains that ONMF should be preferred to k-means and spherical k-means when scaling doesn't affect cluster assignment and data points with larger norm are more reliable.

They conclude by showing that solving ONMF is equivalent to finding a k-partitioning such that the sum of squares of the first singular values of each sub-matrix composed of all the elements in each cluster is maximised.

From the link of ONMF with spherical k-means they introduced in the paper [PGAG14] an EM-like algorithm that solves with strict orthogonality the Frobenius norm ONMF. To my knowledge it is the first NMF algorithm to strictly enforce non-negativity and orthogonality. In 2016 Yadohisa submitted his Doctoral Thesis [YH16] in which he develops "*two- and three-factor ONMF with KL and β-divergence in a fashion similar to that of the ONMF of Pompili et al. (2014)*" ('divergence' is acquainted chapter 2 of this thesis).

1.5 PNMF, making the bridge between the different concepts

1.5.1 PCA inspired PNMF

In 2005 Z. Yuan and E. Oja introduce the PNMF [OY05]. Quoting the article: *"In PCA or the related SVD, the image is projected on the eigenvectors of the image covariance matrix, each of which provides one linear feature. The representation of an image in this basis is distributed in the sense that typically all the features are used at least to some extent in the reconstruction. Another possibility is a sparse representation, in which any given image window is spanned by just a small subset of the available features. This approach is related to the technique of Independent Component Analysis which can be seen as a nongaussian extension of PCA and FA."*

They make the link with NMF as it has been shown that, similarly to ICA (introduced section 1.4.2 with BSS), non-negativity tends to result in sparse representations. However in 2005 it was well accepted that NMF doesn't necessarily result in a localized representation. From this observation they suggested a model that uses the matrix representation of PCA (1.1) without the constraint of orthogonality on W and the centering of M, but adding a non-negativity constraint on W. Taking a look at figure 1.2, they suggest to encode M in a reduced space accepting to loose information to preserve non-negativity. The decoding is then not as good as with PCA but the data contained in H is a part based representation. Also they do not center M as they want a non-negative representation in H. This results in a NMF where there is only one unknown matrix instead of two, making the two-block coordinates approach unnecessary. The number of free parameter decreases from $k \times (m + n)$ to $k \times m$.

To solve it they use the same technique as Lee and Seung in NMF, that is multiplicative updates. They make the remark that OPNMF (PNMF with the orthogonality constraint enforced) is equivalent to non-negative PCA (NNPCA). A 2018 article [VGSea18] interprets that PCA captures the components with the most variance explained across the dataset, while OPNMF captures the spatially more localized components that consistently co-vary across the dataset. The W matrix obtained using the PNMF model is already quite orthogonal but not completely.

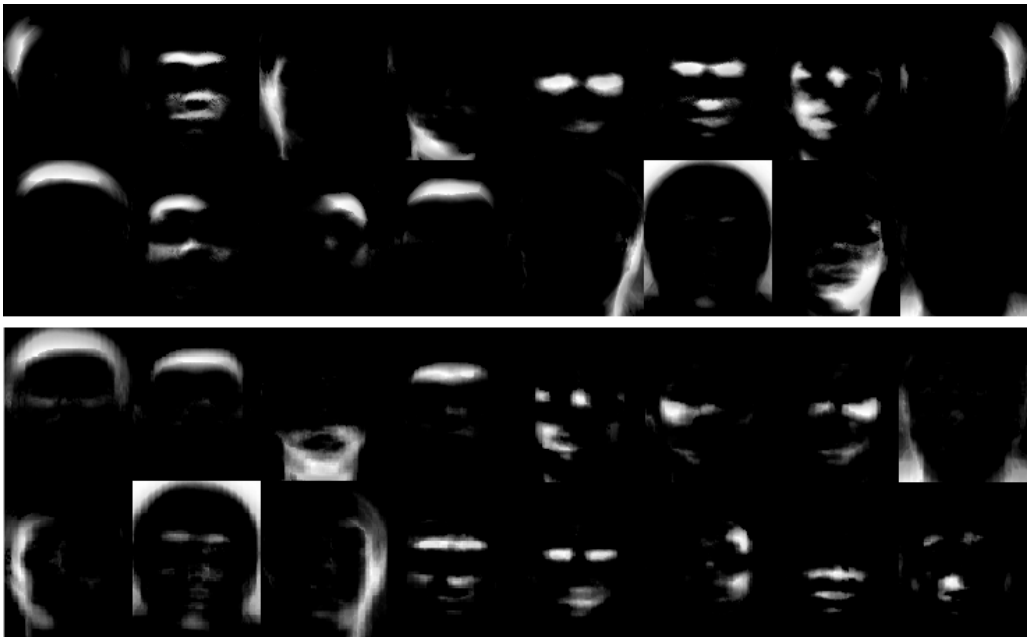

Figure 1.12: Illustration of example A using the ORL face database. The faces contained in W-PNMF (above) and W-OPNMF (below) gives results that are similar to ONMF

PNMF model

The PNMF model is:

$$\begin{aligned} \underset{W \in \mathbb{R}^{m \times k}}{\text{minimize}} \quad & ||M - WW^T M||_F^2 \\ \text{s.t} \quad & W \geq 0 \end{aligned} \tag{PNMF}$$

The OPNMF model is the PCA with an additional non-negativity constraint:

$$\begin{aligned} \underset{W \in \mathbb{R}^{m \times k}}{\text{minimize}} \quad & ||M - WW^T M||_F^2 \\ \text{s.t} \quad & W \geq 0 \\ & W^T W = I \end{aligned} \tag{OPNMF}$$

Example A.5

The effect and the interpretation of the sparsity are different if we solve W-PNMF or H-PNMF. Figure 1.12, we observe that the W-PNMF and W-OPNMF results in something alike the bottom image of figure 1.9, what he had with W-ONMF.

If instead we proceed to H-PNMF, we obtain again something similar to the H-ONMF (k-means alike), visualisation is almost the same as top of figure (1.9) (and thus not shown).

1.5.2 OPNMF, k-means and ONMF

PNMF and clustering

The transpose of the objective of PNMF function is: $||M^T - M^T W W^T||_F^2$, and by posing $H = W^T$ it becomes similar to the k-means objective function (k-means) :

$$\begin{aligned} \underset{H \in \mathbb{R}^{k \times m}}{\text{minimize}} \quad & ||M^T - M^T H^T H||_F^2 \\ \text{s.t} \quad & H \geq 0 \end{aligned} \tag{1.11}$$

We insist that the models (PNMF) and (1.11) are equivalent. The first form is related to and the second is related to the (k-means), except that H is of size $k \times m$ instead of $k \times n$.

From this reasoning we can derive a H-PNMF version:

$$\begin{aligned} \underset{H \in \mathbb{R}^{k \times n}}{\text{minimize}} \quad & ||M - M H^T H||_F^2 \\ \text{s.t} \quad & H \geq 0 \end{aligned} \qquad \begin{aligned} \underset{W \in \mathbb{R}^{n \times k}}{\text{minimize}} \quad & ||M^T - W W^T M^T||_F^2 \\ \text{s.t} \quad & W \geq 0 \end{aligned} \tag{1.12}$$

Instead of using model (1.12) when we want to reduce the dimension m, we could insert M^T in the (PNMF) model, then one would have $H = W$.

Very simply put we saw that (H-ONMF) is (k-means) with more freedom (figure (1.11)). H-PNMF is k-means with even more freedom: peaking at (k-means) the constraint on B is loosened to only impose non-negativity. We still have that $W = MH^T$, each centroid $W_{:i}$ coincides with the mean of all the $M_{:j}$ in the corresponding cluster, but now one $M_{:j}$ can be associated at the same time weightily to different clusters, this is a soft clustering version of k-means.

1.6 ONMF is OPNMF

Theorem 1

Given M non-negative, OPNMF is equivalent to ONMF.

Proof
We are showing the two sides of the statement.

- From the (OPNMF) model we can set $H = W^T M$. Then, as M is non-negative, H is of the same sign as W. Therefore the objective function and the conditions in (ONMF) are enforced by (OPNMF).
- In regards to (ONMF), for any given factor $W \geq 0$ with orthonormal columns, the second W-ONMF factor H is readily determined: $H = W^T M$. We can

obtain the latter by deriving, for a given W orthonormal:

$$\begin{aligned} \underset{H\in\mathbb{R}^{k\times n}>0}{\text{minimize}} \quad & ||M-WH||_F^2 \\ & = \text{tr}[(M-WH)(M^T-H^TW^T)] \\ & = \text{tr}[MM^T]-\text{tr}[MH^TW^T]-\text{tr}[WHM^T]+\text{tr}[HH^T] \\ H^* = W^TM & \end{aligned} \tag{1.13}$$

□

In conclusion, if we assume before-hand that H must always be set to its optimal value in W-ONMF, then it is OPNMF. The article [APD15] explains that for W-ONMF "if an oracle reveals that W is non-negative, then $H = M^TW \geq 0$". We prefer to ask for W to be of the same sign as H and impose M non-negative.

Pompili applied this result with no emphasis in his proof that ONMF is related to a spherical k-means variant in [PGAG14], equation 2.6. He also used this result page 23 in his Thesis.
It is only in his Thesis at page 55 that he does accentuate the result.

Non-Negative PCA

In the case M is non-negative, if ONMF is OPNMF, and if OPNMF is NN-PCA without the centering prepocessing, our problem maybe exists in the PCA literature. The first PCA with non-negativity constraints we could find dates from 1997, "*Determination of PCB Sources by a Principal Component Method with Nonnegative Constraints*" in [RC97]. It is cited by 100 articles. We did not dig into that.

The expression 'Non-negative PCA' (NN-PCA) commonly recognised in the litterature nowadays was coined in 2014 [MR16] by Montanari. He develops a model for $k = 1$ only. He enforces OPNMF seen under its maximisation problem form (model (maxPCA) with non-negativity constraint). Almost everywhere in the literature if we consider the minimisation problem (minPCA) with non-negative constraints then we talk about "OPNMF", if we consider the maximisation problem (maxPCA) with non-negative constraints we talk about "NN-PCA".

In 2018 Asteris developed an algorithm that solves NN-PCA for any k [APD15], as will be discussed it chapter 2.

Merging W- and H- representations and choosing a model for ONMF

Using the heavy W-ONMF and H-ONMF notations up to now allowed us to understand the strings behind the different models in the literature, making the links piece by piece, focusing sometimes on the cluster assignment and sometimes on the basis matrix. It should be clear at this point to the reader how ONMF is both performing dimensionality reduction and clustering. In the following, we would like to stop using W- and H- representations. We employ solely W-ONMF to represent all the problems. To obtain H-ONMF we then simply need to plug in the transpose of M:

$$\begin{aligned} \underset{W\in\mathbb{R}^{n\times k},H\in\mathbb{R}^{k\times m}}{\text{minimize}} \quad & ||M^T - WH||_F^2 \\ \text{s.t} \quad & W, H \geq 0 \\ & W^TW = I \end{aligned} \tag{1.14}$$

We already discussed section 1.2.4 how k-means is PCA with more constraints. OPNMF is also PCA with more constraints and it is K-means with less constraints. Something "magical" (a result proven **Theorem 1** is that when M is non-negative, ONMF and OPNMF models are equivalent. No mater what, given W orthogonal and M non-negative, $H = W^TM$. We can thus write the W-ONMF model as a NN-PCA model (without the centering of M but verifying its non-negativity):

$$\begin{aligned} \max_{W\in\mathbb{R}^{m\times k}} \quad & ||M^TW||_F^2 \\ \text{s.t} \quad & W \geq 0 \\ & W^TW = I \end{aligned} \tag{NN-PCA}$$

1.7 Conclusion

Since the discovery of (SVD), research went in the direction of obtaining a factorisation such that the nature of the data in W is of the same spirit as M, containing all the information. At the beginning one would use ellipsoid methods applying (orthogonal) transformation matrices to the SVD decomposition to re-frame the factorised matrix in the positive quadrant. FA was developed, in that context PMF was brought in and from it (NMF) arose. The goal of NMF is to reduce the dimension of M representing it by W that should contain 'parts', making it therefore also clustering. Step by step it was found out that adding soft orthogonality constraints on top of NMF turns the matrix W more interpretable, enhancing the clustering - the separation.

Orthogonality has intrinsically a decomposition property. Orthogonality with non-negativity has a hard clustering property.

NMF gives interpretable results that are often sparse, leading to a decomposition by parts property. However sparseness cannot be controlled. We saw how enforcing hard orthogonality in NMF leads to:

- a rigorous hard-clustering interpretation of NMF. For any matrix A, being column-orthogonal ($A^TA = I$) and non-negative implies that each row of A has only one non-negative element, being row-orthogonal ($AA^T = I$) and non-negative implies that each column of A has only one non-negative element. A is a bona fide cluster indicator.

- a rigorous NN-PCA interpretation of ONMF. Starting from the SVD decomposition (this means M not centered) using the PCA model, hard enforcing orthogonality and soft-enforcing non-negativity forces the coordinates to split among the principal vectors. This makes the principal vectors disjoint where each coordinate is non-zero in at most one vector, a part-based representation.

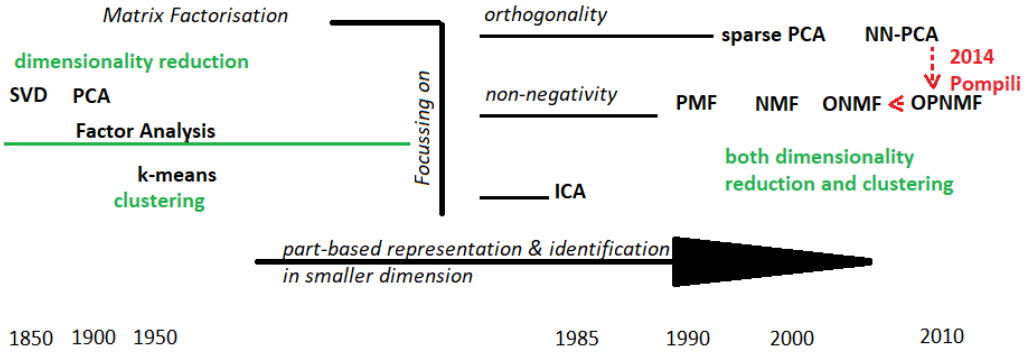

Figure 1.13: Timeline of chapter 1. In red, the ONMF algorithms as suggested by Pompili (EM-ONMF was introduced section 1.4.3 and ONPMF (wich is not OPNMF) will presented in the next chapter) makes the link with NN-PCA. The OPNMF comes out from the idea of NN-PCA, upon which Pompili derived his hidden P-ONPMF-R algorithm (also presented next chapter).

The literature often prefers to link NMF (and ONMF) to k-means. Studying the ONMF and investigating its premisses pushed us to embody the key element that follows the PCA model with an additional nonnegative constraint, thus choosing to use the NN-PCA model to study ONMF in the next chapters.

This chapter is recapitulate with a non exhaustive timeline as presented 1.13.

Chapter 2

Heading for strict orthogonality in ONMF algorithms

We have studied in the first chapter the ties between different models. It allowed us to understand fully how NMF with hard orthogonality is NN-PCA.

We are now to study how algorithms for ONMF are built and see that by construction they do not seek to enforce hard orthogonality.
Then, section 2 we introduce how the Stiefel manifold can help us to enforce hard orthogonality, we discuss and make a literature review of the paper [PGAG14], in which Pompili presented the first hard-orthogonal algorithm, the ONPMF.
In section 3 we introduce penalty methods on Manifolds and we follow the trail of the OPNMF unveiling the P-ONPMF-R algorithm, applying a quadratic penalty method.
At last, section 4 we introduce how Manopt can be used to develop ONMF algorithms on manifolds, we discover its limitations and its potential; also we adapt Pompili's work to it, suggesting to rebrand the P-ONPMF-R algorithm to RQNN-PCA.

2.1 Global context of NMF problems

We start by posing a broader context to the NMF problem seen in the first chapter. The numerical optimization of NMF (NMF) is a challenge because enforcing constraints both in W and H is non-convex. It is subject to inherent ambiguities since it doesn't possess a unique solution, $WH = WDD^{-1}H$ for any invertible non-negative matrix D. A NP hardness result has been shown by Vavasis in 2009 [Vav09].

Divergence measure

We have seen that the Frobenius norm (squared Euclidean distance) is a metric often used to check the quality of the MF. It is correct to employ when we can assume the noise N to be Gaussian (see section 1.2.2). There are many other possible ways to measure the discrepancy between the two and find the "entropy" of the result when N is not Gaussian.

In general the quality of the factorisation is assessed by the divergence:

$$D(M||WH) = \sum_{i=1}^{m} d(M_i, WH_i) \geq 0$$

It is a distance but it does not necessarily satisfy the triangular inequality and sometimes is not symmetric, from there the name "divergence" and not "metric".

There exist many divergences such as the the Kullback-Leibler divergence, Bregman divergence and Rényi divergence.
In this master thesis we focus on the squared Euclidean distance.

Cost function in NMF

Typically, when constraints are added on W or H to the CLRMA model, the algorithms used to solve it include all constraints other than non-negativity in the cost function using regularisation terms. All the added constraints push the solution to be sparse, but they are not necessarily strictly enforced. It can be written as follows:

$$\begin{aligned} D_F(M||WH) \ &= ||M - WH||_F^2 + \alpha_W J_W(W) + \alpha_H J_H(H) \\ &\text{s.t. } W \geq 0,\ H \geq 0 \end{aligned} \tag{2.1}$$

with α learning rates parameters between 0 and 1 that can be modified during the iterations (they need to be tuned), and J the added constraints.

In regards to the orthogonality constraint, we will show example C1 that such an approach leads to matrices that are approximately orthogonal, that is soft clustering. As the main objective of the constraint is to help W to contain local features, often we don't need a strict application of the constraint. But for some applications, such as hyperspectral imaging or BSS, it would be beneficial to apply the orthogonality strictly.

2.1.1 PNMF, OPNMF and ONMF algorithms: not achieving orthogonality?

Because the orthogonality and non-negativity constraints collide, we cannot in general obtain a strictly non-negative W for which the columns are strictly orthogonal. The solution is dependent on the initialisation and is not unique. Similar to (k-means) and all CLRMA, the problem is non-convex and NP-hard [Gil17]. Most often when the orthogonality constraint appears in the ONMF literature it is relaxed, and for many applications, with conclusive results. Since 2014, thanks to Pompili, research was made on the problem to hard-enforce the orthogonality, only tending to non-negativity. In this case, we have seen this chapter how the orthogonality steers the problem from (ONMF) to (NN-PCA).

When to choose which model

In his 2005 article Oja et. al. [OY05] emphasis the PNMF. When should we use OPNMF instead of PNMF? And OPNMF instead of ONMF? One could think that

it depends on the application: if we can allow soft clustering we should employ PNMF and if we expect to obtain hard clustering we should choose OPNMF. Nevertheless, because of the design of the algorithms in PNMF, OPNMF, ONMF, and because of the way the Multiplicative Updates[1] are derived, the final result is always non negative and tends to be orthogonal although not perfectly. For any of these models, when using MU as suggested by Oja et. al. [OY05], we obtain quite the same jumble. We illustrate this with an example.

Example C1: the Iris Dataset

This state of the art dataset contains a matrix M with $n = 4$ measurements of $m = 150$ flowers. We know that they belong to $k = 3$ species and want to identify each flower of the dataset. It was reported in [YO10] how PNMF obtains better results than K-means, NMF and ONMF (purity of 0.97). This result is such because the iris dataset doesn't contain enough information to be able to hard-cluster the flowers using an unsupervised technique, making soft-clustering a better approach. With the PNMF algorithm written by Yuan et. al. [YYO09], the OPNMF algorithm written by Aristeidis [ARD15], we compare the matrix W^TW that should be equal to the unity matrix. We see that they are not perfectly orthogonal and that OPNMF multiplicative updates don't necessarily lead to a more orthogonal result then PNMF. The same apply for the standard ONMF: as it enforces using the MU the strict non-negativity, the orthogonality is not obtained at the limit.

An OPNMF result

$$W^TW = \begin{bmatrix} 0.89 & 0.04 & 0.12 \\ 0.04 & 0.89 & 0.05 \\ 0.12 & 0.05 & 0.76 \end{bmatrix}$$

A PNMF result

$$W^TW = \begin{bmatrix} 0.82 & 0.03 & 0.13 \\ 0.03 & 0.96 & 0.00 \\ 0.13 & 0.00 & 0.88 \end{bmatrix}$$

An ONMF result

$$W^TW = \begin{bmatrix} 0.83 & 0.24 & 0.04 \\ 0.24 & 0.90 & 0.11 \\ 0.04 & 0.11 & 1.00 \end{bmatrix}$$

2.1.2 Sparse PCA with non-negativity using multiplicative updates: looping around

We present an algorithm that has "PCA" in its named, but follows the idea found in the general literature of ONMF (2.1), hard enforcing non-negativity and loosening the orthogonality constraint.

We saw how NMF arose from the context of PCA, making sense of the data using non-negative constraints instead of an orthogonal one. An other path that began in 1995 is sparse PCA, the idea being introduced by Jollife [Jol95], "Rotation of principal components: Choice of normalization constraints". What if we added a constraint to the PCA model (maxPCA) to limit the number of entries that are different to zero in each column of W? It would force the PCA to have a clustering effect. It is known as sparse PCA. With $\hat{\Sigma}_M \in \mathbb{R}^{m\times m}$ the covariance matrix M and

[1] If you don't know what are MU and how they are derived, this article introduces the topic

k the number of elements allowed in each "cluster", it is usualy written as:

$$\begin{aligned} \underset{v\in\mathbb{R}^m}{\text{maximize}} \quad & v^T\hat{\Sigma}_M v \\ \text{s.t} \quad & ||v||_2 = 1 \\ & ||v||_0 \leq k \end{aligned} \tag{2.2}$$

Many have worked on this model, from SCoTLASS to SPCA to DSPCA [ZS07]. Some also used this approach when NMF failed to learn by part ([Hoy04]). In 2007, Ron Zass introduced Non-negative Sparse PCA [ZS07]. Using the PCA model (maxPCA), he would add a non-negativity constraint and put the orthogonality constraint in the objective function. Then he adds a sparsenness constraint on top to encourage the problem to obtain a part based representation. He solves the NSPCA:

$$\underset{W\in\mathbb{R}^{m\times k}}{\text{maximize}} \quad \frac{1}{2}||M^T W||_F^2 - \frac{1}{4}\alpha||I - W^T W||_F^2 - \beta||W||_{L0} \quad s.t. W \geq 0$$

To get low overlaps in the CBCL face database (that we used example A) they set $\alpha = 5 \times 10^8$ and $\beta = 0 \cdots 2 \times 10^6$. As for OPNMF by Oja et. al.[OY05], the authors have loosened the orthogonality. We return to something very similar to O(P)NMF with strong non-negativity, looping around! Nonetheless, they use a non-negative coordinate gradient rule that preserves orthogonality. So if the initial W is orthogonal it will stay, in opposite to when using the multiplicative updates.

2.1.3 Achieving orthogonality and the importance of initialisation

Commonly in the general NMF, W and H are randomly initialised. This is important because the NMF doesn't possess a unique solution as in k-means. The problem is solved repetitively to assert we converge towards a solution. If we start repeatedly with the same initialisation we might loose possible solutions. For more information one can consult [Gil17].

Sometimes a unique initialisation is needed. For instance in the Non-negative Sparse PCA presented here-above one starts with the U of the (SVD) decomposition. In particular, when we want our solution to be near-orthogonal, it helps to initiate with a solution that is orthogonal. (PNMF) Uses this trick too. Consistently starting from the unique SVD initialisation may contain the risk (or advantage) to obtain always the same solution and skip possibilities.

2.2 Strictly enforcing orthogonality: Pompili's novel approach

In collaboration with Nicolas Gillis, Pierre-Antoine Absil and François Glineur, Filippo Pompili developed in 2014 the ONPMF algorithm different then what existed in the ONMF literature [PGAG14]. The "P" inside the name is for non-negative "Penalised" MF, as what he did is penalise the non-negativity in stead of the orthogonality:

$$\begin{aligned} D_F(M||WH) &= ||M - WH||_F^2 + \alpha_W J_W(W) + \alpha_H J_H(H) \\ &\text{s.t. } W^T W = I, \end{aligned} \tag{2.3}$$

When minimizing on W, the orthogonality constraint in this case arises a special structure that can be exploited.

2.2.1 The Stiefel Manifold

A manifold is a topological space that locally resembles the Euclidean space near each point. A submanifold of a manifold M is a subset S, which itself has the structure of a manifold, and for which the inclusion map $S \to M$ satisfies certain properties.

As it exploits the squared Frobenius norm, we can solve the formulation (2.3) on a structure that satisfies the constraint $W^T W = I_{k\times k}$; that structured is named the Stiefel Manifold $St(k, n)$. By definition the orthogonal Stiefel Manifold $St(k, n)$ is the set:

$$St(k, n) := \{X \in \mathbb{R}^{n\times k} : X^T X = I_k\}$$

of all $n \times k$ orthonormal matrices.

The Stiefel Manifold is an embedded submanifold of $\mathbb{R}^{n\times k}$, i.e. contained within. Its dimension is $dim(St(k, n)) = nk - k(k + 1)/2$.

The Stiefel Manifold is a Riemanian Manifold, it is a smooth and thus differentiable manifold. Each tangent space is equipped with an inner product $\langle ., . \rangle$ in a manner which varies smoothly from point to point.
Herewith, we can derive a gradient in the manifold for an unconstrained objective function in the set of orthonormal matrices, the Stiefel Manifold, at the condition that the objective function is smooth.

2.2.2 ONMF on Stiefel Manifold: Introducing ONPMF (ON-Penalised-MF)

The first to use the Stiefel manifold to solve ONMF were Yoo and Choi [YC08]. They exploited it in the context of formulation (2.1): when deriving the MU it was a way to empirically drive the iterates towards the Stiefel manifold.

Pompili was the first to strictly enforce orthogonality by solving ONMF on the Stiefel manifold. Based upon the H-ONMF, he derived the augmented Lagrangian[2], $\Lambda \in \mathbb{R}_+^{k\times n}$:

$$L_\rho(W, H, \Lambda) = \frac{1}{2}||M - WH||_F^2 + \langle \Lambda, -H \rangle + \frac{\rho}{2}||\min(V, 0)||_F^2$$

He would at each iteration solve one by one:

- W using the active set method, with:

$$\text{argmin}_{X\in R_+^{m\times k}} ||M - XH||_F^2$$

[2]He employed an augmented Lagrangian to tackle the zero-lock problem.

Figure 2.1: Illustration of example A using the ORL face database, the faces contained in W using Pompili's ONPMF.

- H in the Stiefel manifold, solving a projected gradient scheme:

$$\text{Proj}_{St}(H - \beta \nabla_H L_\rho(W, H, \Lambda))$$

  To find the step length β a backtracking line search is used. ρ is predetermined and is amplified iteration after iteration, making the solution increasingly non-negative. His algorithm is named ONPMF to account for the ρ.

- Λ, such that it is updated by $\max(0, \Lambda - \alpha V)$ where α is a predetermined sequence of step lengths decreasing to 0.

Pompili obtained convincing results. To get a hint at how the hard enforcement of the orthogonality changes the problem, we present figure 2.1. We compute matrix W using his ONPMF algorithm with the same dataset as in example A3 , the ORL dataset. It is performing hard clustering, therefore, each image strictly cannot overlap changing the solution compared to all the ONMF, NMF, PNMF and OPNMF algorithms previously seen.

His article has been quoted 57 times according to Google Scholar. The yearly quotation is presented in table 2.2.2.

Year	2014	2015	2016	2017	2018	Total
Number of references	3	7	22	16	19	57

Table 2.1: Yearly quotations of paper [PGAG14]

We decide to emphasis three papers:

- Article [KKT14] does a survey of the different ONMF algorithms in the literature. Commenting Pompili's algorithm, "*non-negativity was not strictly guaranteed in the algorithm. More worsely, it has three parameters to be set appropriately.*" When studying the results obtained with the different algorithms they skip the OPNMF: "*We omit Pompili's PGD based ONMF. Because the PGD scheme needs linear search with a fixed step-size and, thus, the speed depends on the other type of iterations*". They reported that "*their*

ONMF needs a higher computational cost than traditional ONMF algorithms." This article introduces Hierarchical ONMF that is now well used in the latest papers.

- Article [ZTS+16]. They apply the same approach as Pompili, but to semi-ONMF: they drop the non-negativity constraint on the matrix that is orthogonal. Instead of the Riemannian gradient descent Pompili used they propose a nonlinear Riemannian Conjugate Gradient descent. Instead of a fixed step size for the descent in the Stiefel manifold they use an adaptive Barzilai-Borwein (BB) step size. They use the OPNMF algorithm as their competitor to benchmark their NRCG-ONMF and obtain 'better results'.

- Article [LZQ+18]. Released November 2018, it does cite neither [ZTS+16] nor [PGAG14]. The authors state that they are the first to strictly enforce orthogonality. They also use semi-ONMF, they force W to be on the Stiefel Manifold "by implementing the Cayley transformation" and they use H-ONMF to update H efficiently. As it was released out of the blue and at the end of my Thesis, I did not investigate it any further.

We can see that today, all those who work on ONMF cite Pompili's ONPMF (or/and EM-like algorithm discussed in section 1), but no-one wrote an algorithm that strictly enforces orthogonality penalising non-negativity of both W and H. No-one took advantage that ONMF with hard orthogonality is naturally drawn to the OPNMF i.e. NN-PCA. Except for Asteris [APD15].

Asteris' NN-PCA

Asteris [APD15] suggests the "low rank NN-PCA" algorithm to find an approximate solution to the NN-PCA (NN-PCA). He uses the approximated W to compute the H of the MF (MF), judiciously applying the representation $M \approx WW^TM$. For this he states that "if an oracle reveals that W is non-negative, then $H = M^TW \geq 0$". This link between ONMF and OPNMF has been described earlier in chapter 1. He named his algorithm OPNMFS.

His low rank NN-PCA algorithm is the following:

1. Factorise $M_{m\times n}$ to $\bar{M}_{m\times n}$a using truncated SVD, a matrix of same dimension but of rank r small compared to m and n

2. Write explicitely the SVD decomposition of rank r $\bar{M} = \bar{U}\bar{\Sigma}V^T$, drop the V

3. for a predetermined amount of time, randomly select a $C_{r\times k}$ where each column is unit-norm, compute $A_{m\times k} = \bar{U}\bar{\Sigma}C$ and solve for W orthogonal and non-negative the Cauchy-Schwartz derived problem

$$\hat{W} = \arg\max \sum_{j=1}^{k} \langle w_j, a_j \rangle^2$$

At the end of the allocated time keep the best $\hat{W}$ and compute $H = \hat{W}^TM$.

Refer to the article [APD15] for detailed explanations. What is important for the further development of this thesis is that this procedure uses an enumeration of all the possible C. Consequently if we want to find the best solution we must try $2^{r\times k}$ different possibilities, which is not very convenient. I ran many tests using hyperspectral images and the results are awesome provided we have enough time to wait. Even if we decide to stop the random enumeration early we already obtain quite good results, but with no theoretical guarantee!

Another interesting point is the algorithms he uses to benchmark his ONMFS. These are Oja's OPNMF [OY05], Pompili's ONPMF [PGAG14], the vanilla and spherical k-means, the NSPCA (non-negative sparse PCA) [ZS07] and the NNSPAN. Except for the last one we already introduced all of them in the foregoing, which is a good news.

2.3 Pompili's unpublished legacy

Considering CLRMA with orthogonality and non-negativity constraints, Pompili developed in his Doctoral Thesis a novel and yet never published OPNMF algorithm. His approach and his findings are currently not accessible to the scientific community[3].

With his ONPMF algorithm discussed section 2.2.2, Pompili would obtain excellent clustering results. As demonstrated in **Theorem 1** , enforcing the orthogonality of H results in obtaining $W = MH^T$. Therefore using the OPNMF model should lead to the same result as ONMF.

Pompili used the H-OPNMF version instead of W-OPNMF to insist on the clustering aspect:

$$\begin{aligned} \underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \quad & \frac{1}{2}||M - MH^TH||_F^2 \\ \text{s.t} \quad & HH^T = I_{k\times k} \\ & H \geq 0 \end{aligned} \tag{2.4}$$

2.3.1 Setting the context

Solving it on the Stiefel manifold we are restricting the set of solutions to be orthogonal:

$$\begin{aligned} \underset{H\in\mathbf{St^{k,n}}}{\text{minimize}} \quad & \frac{1}{2}||M - MH^TH||_F^2 \\ \text{s.t} \quad & H \geq 0 \end{aligned} \tag{2.5}$$

This can be written in the general formulation:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{s.t } & x \in \mathcal{M} \\ & g_i(x) \leq 0 \text{ for } i \in \mathcal{I} \\ & h_j(x) = 0 \text{ for } i \in \mathcal{E} \end{aligned} \tag{MAN}$$

[3]Professor Wang, co-inventor of the OPNMF, confirmed this fact to me.

where M is a Riemannian manifold, $\mathcal{I}$ contains all the inequality constraints and $\mathcal{E}$ contains all the equality constraints. Solving an unconstrained problem in manifolds is something well studied. But can we solve problems in Manifolds with additional constraints?

2.3.2 Working with manifolds in Matlab: introducing Manopt and handling constraints

Manopt is the Matlab toolbox reference to solve unconstrained problems on manifolds [BMAS14]. As my supervisor Pierre-Antoine Absil is in contact with its main author Nicolas Boumal, he informed me about a discussion that toook place in August 2017 concerning the topic. Briefly, it highlighted that Manopt does not offer ready-to-use tools to solve optimization problems on a manifold with additional constraints. However, Boumal suggested two ways to deal with it:

- splitting methods such as the MADMM [KGB16][4],
- penalty methods, an important class of methods for constrained optimization, seeks the solution by replacing the original constrained problem by a sequence of unconstrained sub-problems. Pompili's ONPMF algorithm [PGAG14] uses an Augmented Lagrangian Penalty Method.

Because Pompili also used a penalty method for his unpublished P-ONPMF-R algorithm [Pom12], the remainder of this thesis focusses only on the latter.

An effort is currently made to integrate constraints using penalty methods in Manopt. A paper [LB18] was released in July 2018[5]. It comes with a toolbox accessible on Github. We will investigate it after introducing Pompili's work.

2.3.3 Pompili's approach to penalty methods

Two thirds of Pompili's Thesis is dedicated to the generic optimisation on the Riemanian manifold and his P-ONPMF-R algorithm. He uses a penalty method. Our goal is not to copy-paste the work of Pompili, but to present the most needed tools for the reader to be convinced of his algorithm (or baffle him).

As there is no abstract or introduction in his Thesis [Pom12], I introduce its structure.
Pompili presents chapter 1 constrained optimization on Euclidean spaces, more specifically the quadratic penalty method, based on Nocedal's renown [NW06] (quoted already more than 25 000 times). In chapter 3, applying to P-A Absil reference book on manifolds [AMS09], he writes a user manual about unconstrained optimisation on Riemanian manifolds. Finally in chapter 4, he submits his P-ONPMF-R algorithm with a proof of convergence. For that he uses the theorem presented in [YZS14]:

[4] I did not investigate the MADMM trail.

[5] I received it through my supervisor Professor Pierre-Antoine Absil, to my knowledge this publication is not yet available on the internet. Please contact me if you want an access.

Theorem 2
For the case where $\mathbb{M}$ is an embedded submanifold of $\mathbb{R}^m$, formed by a set of equality constraints, the optimality conditions can be derived directly from the traditional results on $\mathbb{R}^m$.

As the Stiefel manifold $\mathbf{St}(k, n)$ is an embedded submanifold of $\mathbb{R}^{n\times k}$ that enforces the equality constraint $HH^T = I$, it respects the **theorem 2.** From here Pompili adapted to the Manifold framework the results he presented in in his first chapter from [NW06].

Pompili considers the general framework (MAN) deriving the theory for both equality and inequality constraints whereas there are only inequality constraints in the ONMF model on Stiefel manifolds (2.5). He is always very cautious and meticulous, he works in the most general context. The absence of any abstract, introduction or link in his Thesis[Pom12], makes understanding his P-ONPMF-R algorithm arduous.

2.3.4 Constrained optimization in Euclidean spaces with penalty methods

We present the optimisation results on Euclidean spaces that are at the basis of Pompili's P-ONPMF-R algorithm.

They come from [NW06] with a few tweaks to adapt it to the standard manifold formulation (MAN).

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ & g_i(x) \geq 0, \quad i \in \mathcal{I} \\ s.t \quad & h_j(x) = 0, \quad j \in \mathcal{E} \end{aligned} \tag{EUC}$$

where the objective function f and the constraints $\{g_i\}, \{h_j\}$ are smooth real-valued functions on a subset of $\mathbb{R}^n$, $\mathcal{I}$ is the set containing the inequality constraints and $\mathcal{E}$ the set containing the equality constraints. When considering both the equality and inequality constraints they are writen c_i, $i \in \mathcal{I} \cup \mathcal{E}$.

There are two main ingredients that are employed for the proof of convergence by Pompili, the Linear Independance Constraint Qualification (LICQ), and the Karush-Kuhn-Tucker (KKT) condition. The first is always true, as we only have one constraint ($H \geq 0$) it will be independant of the non-existing others. The second can be simplified to the case where there are only inequalities.

Definition (Active set.)
The active set $\mathcal{A}(x)$ at any feasible x is

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} | g_i(x) = 0\}$$

Definition (Linearised feasible directions.)
Given a feasible point x and the active constraint set $\mathcal{A}(x)$, the set of linearized

feasible directions $\mathcal{F}(x)$ *is*

$$\mathcal{F}(x) = \{d : d^T \nabla h_j(x) = 0, \ \ \forall j \in \mathcal{E}, \text{and } d^T \nabla c_i(x) \geq 0 \forall i \in \mathcal{A} \cap \mathcal{I}\}$$

If there is no such d at a point x^*, it is reasonable to think that x^* constitutes a local minimizer candidate.

To ensure that the constraints are managable, we usually ask they respect some assumptions called constraint qualification. The most used is the following:

Definition (LICQ.)
Given the point x^* *and the active set* $\mathcal{A}(x^*)$*, we say that the linear independence constraint qualification (LICQ) holds if the set of active constraints gradients* $\{\nabla c_i(x^*), i \in \mathcal{A}(x^*)\}$ *is linearly independent.*

Definition (Lagrangian.)
The Lagrangian for (EUC) *is defined as*

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} c_i(x)$$

From this condition we can state the first-order necessary conditions:

Karush-Kuhn-Tucker KKT first order conditions
Suppose that x^* is a local solution of (EUC) and that the LICQ holds at x^*. Then there is a Lagrange multiplier vector λ^*, with components λ_i^*, $i \in \mathcal{E} \cup \mathcal{I}$, such that the following conditions are satisfied at (x^*, λ^*)

$$\begin{array}{lcll} \nabla_x \mathcal{L}(x^*, \lambda^*) & = & 0, & \\ h_j(x^*) & = & 0, & \forall j \in \mathcal{E}, \\ g_i(x^*) & \geq & 0, & \forall i \in \mathcal{I}, \\ \lambda_i^* & \geq & 0, & \forall i \in \mathcal{I}, \\ \lambda_i^* c_i(x^*) & = & 0, & \forall i \in \mathcal{E} \cup \mathcal{I} \end{array} \tag{2.6}$$

The Quadratic Penalty Method
Let $[x]^- = \max(0, -x)$. The quadratic penalty function is:

$$Q(x, \rho) = f(x) + \frac{\rho}{2} \sum_{j \in \mathcal{E}} h_j(x)^2 + \frac{\rho}{2} \sum_{i \in \mathcal{I}} ([g_i(x)]^-)^2$$

When f and the equality constraints h_j are with second order derivatives that exist and are continous, the inequality constraints g_i make Q less smooth: the function $\max(0, -x)^2$ has a second order derivative that is discontinuous in 0 (it can be 0 or 2) so Q is no longer twice differentiable in 0. Its first order derivative is continuous provided f is smooth.

In [NW06], the Quadratic Penalty Framework 1 is presented section 17.1. *Such approaches are sometimes known as "exterior penalty methods", because the penalty term for each constraint is nonzero only when x is infeasible with respect to that constraint. Often, the minimizers of the penalty functions are infeasible with respect to the original problem, and approach feasibility only in the limit as the penalty parameter grows increasingly large.*

They write a proof of convergence to a local minimum theorem 17.2 using LICQ and the 1st order KKT conditions.

Algorithm 1 Quadratic Penalty Framework

Input: x_0^s a starting point, $\rho_0 > 0$, $\tau_0 > 0$
Output: x_{final}

for $i = 1, 2, \ldots$ **do**
  Find an aprroximate minimizer x_i of $Q(.; \rho_i)$ starting at x_i^s,
    and terminating when $||\nabla Q(x, \rho_i)|| \leq \tau_i$;
  if final convergence test satisfied **then**
    Stop with approximate solution $x_{final} = x_i$;
  end if
  Choose new penalty parameter $\rho_{i+1} \geq \rho_i$;
  Choose new starting point x_{i+1}^s
end for

Exact penalty functions and Augmented Lagrangian Method
Still in [NW06] chapter 17, they also introduce the framework for exact penalty functions and augmented Lagrangian methods considering (EUC).

2.3.5 Pompili's way of handling the constraint: a quadratic penalty framework

At the time Pompili wrote his algorithm, there was no framework but the result of [YZS14] making it possible to adapt for example algorithm 1 to the Riemanian manifold structure.

Pompili adopts a quadratic exact penalty approach on manifolds to deal with the non-negativity constraint, following the notations and the ideas presented in chapters 12, 15 and 17 of the book Nonlinear Equations [NW06]. With our problem there are $k \times n$ inequalities: each element of H must be non-negative. Using the Frobenius norm all these non-negative constraints can be represented synthetically by only one inequality constraint. This is the major key to the success of the method when implementing an algorithm, as we will see later on with Manopt.

The quadratic penalty function is expressed as:

$$\begin{aligned}
\underset{H\in\mathbf{St^{k,n}}}{\text{minimize}} \quad Q_\rho(H) = \quad & \underset{H\in\mathbf{St^{k,n}}}{\text{minimize}} \quad f(H) + \frac{1}{2}\rho \sum_{i=1}^{k\times n}(g_i[(H)]^-)^2 \\
= \quad & \underset{H\in\mathbf{St^{k,n}}}{\text{minimize}} \quad \frac{1}{2}||M - MH^TH||_F^2 + \frac{1}{2}\rho\sum_{i=1}^{k}\sum_{j=1}^{n}([H_{ij}]^-)^2 \\
= \quad & \underset{H\in\mathbf{St^{k,n}}}{\text{minimize}} \quad \frac{1}{2}||M - MH^TH||_F^2 + \frac{1}{2}\rho||\,[H]^-\,||_F^2
\end{aligned} \tag{QP}$$

Pompili's algorithm alternates between increasing ρ and approximately minimizing $Q_\rho(H)$ using a gradient method directly on the Stiefel manifold. Pompili prooves the convergence of the algorithm in his thesis[Pom12]. It has an outer loop (augmenting ρ at every k) and an inner loop (approximately soving the problem with precision τ_k). We present a pseudo-code for P-ONPMF-R .

Algorithm 2 P-ONPMF-R, adapted from the Quadratic Penalty Framework 1

Input: M a data matrix, k the number of clusters,
$\{\tau_k\} > 0$ a decreasing sequence, $\eta > 0$, $\rho_0 > 0$, $C > 1$
Output: matrix H

Initialise the rows of $H^{(0)}$ with the first k right singular vectors of M and perform flip
for $i = 1, 2, \ldots$ **do**
  $H^{(i)} \leftarrow InexactlyMinQ\left(H^{(i-1)}, \rho_i, \tau_i\right)$
  $\rho_i \leftarrow C\rho_{i-1}$
  if $||\left[H^{(i)}\right]^-|| < \eta$ **then**
    Stop
  end if
end for

2.4 NN-PCA-R

In the following, first we explain the P-ONPMF-R algorithm. From there we suggest to rename it and reformulate it, moving to the NN-PCA framework. We discuss how the algorithm can be written in Manopt referring to the work in [LB18]. We explain why the Manopt solvers [LB18], not yet part of the toolbox, are not fit for the algorithm we introduce, RQNN-PCA, and how we can adapt them.

2.4.1 Pompili's way to make the algorithm 2 work

Based on the results presented in his Thesis[Pom12], Pompili obtains with P-ONPMF-R faster results than the ONPMF and the PNMF algorithm. Compared to Asteris' low rank NN-PCA algorithm we are not enumerating all possible

solutions but moving towards the solution using a steepest gradient descent. This seems very positive, but there are several breaks Pompili had to deal with to obtain such good results.

- **The initialisation.** In NMF as in k-means it is common to start with a random non-negative matrix $H^{(0)}$. As one needs to start with an orthogonal matrix, Pompili chooses to use the eigenvectors of the SVD decomposition as it is the representation that keeps as much information as possible. If one starts with any other initialisation (even orthogonal) the algorithm will perform very poorly, i.e. because of its design as will be discussed in the following points.

- **The flip.** After obtaining the initiate candidate $H^{(0)}$, Pompili performs on it a transformation. He compares the norm of the negative elements *neg* versus the norm of the positive elements *pos* in each column. If $neg > pos$ he flips the columns (all negative entries are turned to positive and vis versa). This helps the problem to start with a head up.

- **The penalty parameter and update ρ and C.** Updating ρ by multiplying it with a constant is a very aggressive strategy. $\rho_0 = 100$ and $C = 1.005 \dots 1.1$ depending on the dataset (use a small C with small datasets and a large C with large datasets). If C is too big the steepest gradient method might fail (or have lots of trouble) to find a new iterate in only a few steps even if τ is large, or we might have lost the "valley" in which the local minimum lies.

- **The inner loop.** *InexactlyMinQ* performs a gradient steepest descent with Armijo line search in the Stiefel manifold. It stops once the error is smaller than τ_k. τ_k is a decreasing sequence that is crucial to the speed of convergence of every iteration. At any k of the outer loop if it is too small then the inner loop might take (literally) ages to converge. If it is too big then the inner loop might not have enough time to make progress. Pompili is setting it to be very big, $\tau_0 = 100||M||_F$ and $\tau_k = \frac{100||M||_F}{\beta^k}$ with β a constant between 1 and 1.5 (1 by default works at best). With such a tactic there are only a few iterations per inner loop.

- **The stopping criterion.** Only examining the non-negativity of H, η is set to 0.001 for small datasets and 0.01 for larger datasets. This is because obtaining a fully non-negative matrix is not always achievable, what what one rather obtains is a marginally negative H.
  In every column there is one strictly positive value and all the others are close to zero but negative (we could decide to put negative values smaller then a threshold to zero once the algorithm has stopped, we then lose the orthogonality property of the matrix). We can interpret the non-negative elements as cluster indicators, which is enough for us.

So much tuning turns Pompili's P-ONPMF-R into a work of art. Everything in his algorithm is adjusted to the constraints up to the point where it can be misunderstood as the parameters focus mainly on the non-negativity: the stopping

criterion does not even investigate the norm of the objective function or its derivative and the progress that is made in diminishing it! It saves a lot of computational time, as instead of evaluating the Frobenius norm of $(M - MH^TH)$, it only evaluates the sum of all the non-negative elements in H. The iterations are stopped as soon as the matrix is sufficiently non-negative ; one is not focussed on the convergence result.

A way to see how the algorithm performs is that one starts at the lowest point of a valley (we begin with the SVD that is the best orthogonal factorisation in regards to the Frobenius norm), then little by little one changes the landscape increasing the penalty ρ on non-negativity. Every time ρ augments the landscape evolves smoothly (as the function and the constraints are 1st order smooth), one checks locally whether the minimum value is displaced and if so, we follow the steepest descent for a few steps.

2.4.2 Standardising P-ONPMF-R to RQ-NNPCA

Renaming P-ONPMF-R to OPNMF-R

Pompili titled his algorithm Projected Orthogonal Non-negative Penalised MF on Riemanian Manifold (P-OPNMF-R). This is intimidating and unappealing. As there are many different NMF models and algorithms with letters at the front such as 'Online', 'Parallel', etc., I suggest a few manipulations that would lead to a more standardised name so as to boost the algorithm's purpose and identity.

- Invert the first two letters. As there already exists an active OPNMF field when there is no PONMF or N-OPMF field in the literature, I advocate for the use of the OPNMF name.
- Keep the "-R" term. As it is solved on the Riemanian Manifold with orthogonality strictly enforced, contrarily to the other algorithms using MU to solve OPNMF, it is essential that the distinction is clearly made.
- Delete the "P" of penalised. It is intrinsic to the fact that we solve it by strictly enforcing orthogonality and putting the non-negativity constraint in the objective function, using the framework of solving constraint problems in manifolds (MAN). Although it gives precious information on how the algorithm works, it brings confusion because there is another "P" that stands for "projected". As there are many penalty methods it should at least mention that it is a quadratic penalty method! I suggest that we consider that the "-R" includes the scope of the "penalised".

Consequently, the name OPNMF-R would be more explicit to the scientific community clarifying what the algorithm does.

Renaming OPNMF-R to RQNN-PCA

To unravel Pompili's algorithm even better I suggest the standard name "RQNN-PCA".

NN-PCA-R for the model, i.e. Nonnegative PCA on Riemanian manifolds, and RQNN-PCA for the algorithm (Riemanian Quadratic penalty Nonnegative PCA).

Initially, this choice sounded frustrating as my thesis subject is "NMF with hard orthogonality constraints" and not "PCA with soft non-negative constraints". What Pompili does is to start with the SVD[6] solution, and iteration after iteration, he imposes it to become more non-negative, ensuring that the new iterate keeps the PCA quintessence (orthogonality) searching solutions only in the Riemanian manifold. With the equivalence of the minimization and maximization of PCA models (minPCA) and (maxPCA), instead of solving the problem as a minimization standard NMF problem with additional constraints, we can see it as a maximisation standard PCA problem with additional constraints. The main difference with PCA is that at input for the NN-PCA-R algorithms one must keep M nonnegative and shouldn't center it.

Using the (maxPCA) model is less heavy in computations (to evaluate the objective function there is only one product to make (W^TM), instead of two products and a subtraction ($M - WW^TM$)). Also, referring to the historical background in chapter 1, we have seen that the concept of extracting a non-negative PCA from the PCA was an issue thoroughly investigated before PMF and NMF existed, it is even part of the reason how NMF surged (if the reader is not convinced, I direct him towards section 1.3.2).

Adieu to Pompili's H-ONMF forever?

Pompili focussing only on the clustering aspects of ONMF does not make what's behind it fully apparent.

We saw how k-means embodies PCA (they both are dimensionality reduction enforcing orthogonality) with more constraints. As Pompili was the first focusing on orthogonality rather then non-negativity, and as by doing so the ONMF becomes OPNMF, he would always fall back on a problem related to the more general PCA with non-negativity constraints, something that already exists in the literature. He followed PCA to the point of using SVD as unique initialisation, loosing the multiple solutions that are possible with k-means.

We chose to drop the $H-$ point of view for the $W-$ end of the first chapter. We move to (NN-PCA-R) writing it with W and under its maximisation form.

Introducing NN-PCA-R model and algorithm

The (NN-PCA) can be written under the general formulation (MAN), if M is non-negative it differs with the PONMF model on Riemanian manifolds (2.5) only by the cost function f:

$$f = -\frac{1}{2}||M^TW||_F^2$$

[6]The reader might think that the name NN-SVD is better suited as we start with the SVD solution and as we do not centre the matrix M. The pre-processing of the data is different but we use the same model as PCA, thus my opinion. Also, the literature is keener on NN-PCA rather then NN-SVD (NN-SVD exists in the literature, but it imposes non-negativity of its three factorized matrix U, Σ, V) which doesn't interest us here.

.

The NN-PCA-R model we will use until the end of this thesis is, for M nonnegative:

$$\begin{aligned} \underset{W \in \mathbf{St^{k,m}}}{\text{minimize}} \quad & -\frac{1}{2}||M^T W||_F^2 \\ \text{s.t} \quad & W \geq 0 \end{aligned} \tag{NN-PCA-R}$$

We adapt Pompili's algorithm based on his work, as it never needs to compute f, except for the notation W algorithm P-ONPMF-R is exactly the same as algorithm RQNN-PCA .

Algorithm 3 RQNN-PCA, adapted from the Quadratic Penalty Framework 1

Input: M or M^T a data matrix, k the number of clusters,
$\{\tau_k\} > 0$ a decreasing sequence, $\eta > 0$, $\rho_0 > 0$, $C > 1$
Output: matrix W

Initialise the rows of $W^{(0)}$ with the first k left singular vectors of the input using SVD
for $i = 1, 2, \ldots$ **do**
  $W^{(i)} \leftarrow InexactlyMinQ\left(W^{(i-1)}, \rho_i, \tau_i\right)$
  $\rho_i \leftarrow C\rho_{i-1}$
  if $|| \left[W^{(i)}\right]^- || < \eta$ **then**
    Stop
  end if
end for

2.4.3 NN-PCA-R using Manopt toolbox

Very recently there has been much progress in the field of adapting penalty methods to smooth manifolds.

In 2018 Bergmann published *"Intrinsic formulation of KKT conditions and constraint qualifications on smooth manifolds"* [BH18], reaching beyond the work Pompili used [YZS14][7] and introducing a systematic discussion for (MAN).

In section 2.3.2 we introduced Manopt for solving unconstrained problems on manifolds and the 2018 paper [LB18] bringing penalty methods in Manopt. The article comes with a patch allowing us to use some penalty methods to plug in the (NN-PCA-R) model.

Based on (MAN), they consider in [LB18] the case where f,$\{g_i\}$, $\{h_j\}$ are twice continuously differentiable functions from $\mathcal{M}$ to $\mathbb{R}^n$. Problem (NN-PCA-R) satisfies the requirements.

Their tactic is the same as Pompili, extending some results by making "minor modifications to the Euclidean proofs". They go beyond Pompili's quadratic penalty method, not even mentioning it.

[7] In [YZS14], the KKT and also the second-order optimality conditions are derived for the problem (MAN) in the setting of a smooth Riemannian manifold, and under the assumption of LICQ.

For the inner iteration of all their solvers they use the RLBFGS (a quasi-newton method) published by Huang, Absil and Gallivan in 2016.

They develop a Riemannian augmented Lagrangian:

$$\mathcal{L}_\rho(x,\lambda,\gamma) = f(x) + \frac{\rho}{2}\left(\sum_{j\in\mathcal{E}}\left(h_j(x) + \frac{\gamma_j}{\rho}\right)^2 + \sum_{i\in\mathcal{I}}\max\left\{0, \frac{\lambda_i}{\rho} + g_i(x)\right\}^2\right) \quad \text{(RALM)}$$

They also develop a Riemannian exact penalty method following:

$$\underset{x\in\mathcal{M}}{\text{minimize}}\ Q(x) = \underset{x\in\mathcal{M}}{\text{minimize}}\ f(x) + \rho\left(\sum_{j\in\mathcal{E}}(|h_j(x)|) + \sum_{i\in\mathcal{I}}\max\{0, g_i(x)\}\right) \quad \text{(REPMS)}$$

The S stands for smoothing. Newton's method and the BFGS methods are not guaranteed to converge unless the function has a quadratic Taylor expansion near the optimum. As the exact penalty function to minimise in the REPMS is non smooth because of the constraints, they suggest the use of the pseudo-Huber loss function or the log-sum-exp function (as seen in LINMA2471 with Prof. Francois Glineur).

The surprise welcome party organised by k-means and NN-PCA

The paper [LB18] comes with examples solved on manifolds, among which k-means and NN-PCA for k=1.

- **k-means.** Based on the 2017 article [CMVW17]. Written with the notations of this thesis, given a set of data points $\{m_i\}_{i=1}^m$ stored as the column of $M_{n\times m}$. Let $D_{n\times n}$: $D_{ij} = ||m_i - m_j||^2$ the squared Euclidean distance matrix. In the Matlab example code of the patch we can see that D is computed as $D = M^T \times M$ (as we proceed to clustering with our notation). The model is:

$$\begin{aligned} \underset{W\in\mathbf{St^{k,n}}}{\text{minimize}} \quad & -\text{Trace}(W^T M^T M W) \\ \text{s.t} \quad & W \geq 0 \\ & WW^T\mathbf{e} = \mathbf{e} \end{aligned} \tag{2.7}$$

  We have that $\text{Trace}(W^T M^T MW) = ||MW||_F^2$.

$$\begin{aligned} \underset{W\in\mathbf{St^{k,n}}}{\text{minimize}} \quad & -||MW||_F^2 \\ \text{s.t} \quad & W \geq 0 \\ & WW^T\mathbf{e} = \mathbf{e} \end{aligned} \tag{2.8}$$

  For M non-negative, model (2.8) is the same as model (NN-PCA-R) except it has one more equality constraint. It implies that if $W_{ij} \neq 0$ and $W_{il} \neq 0$ then $W_{ij} = W_{lj}$. Using figure 1.4 where now W is the H, it visually confirms that ONMF is k-means with less constraints.

  Thus a first way to adapt our NN-PCA-R model in Manopt is to use the work for k-means and remove the equality constraint.

- **NN-PCA for k=1.** We already mentioned Montanari's NN-PCA [MR16]. We would like to recover a signal $v_0 \in \mathbb{R}^d$ such that $||v_0||_2 = 1$. We know that this signal follows the spiked model, that can be written as $Z = \sqrt{SNR}v_0v_0^T + N$ with N a Gaussian noise and SNR the signal to noise ratio. The classical PCA problem is:

$$\begin{aligned} \underset{v\in\mathbb{R}^d}{\text{minimize}} \quad & -v^T Z v \\ \text{s.t} \quad & ||v||_2 = 1 \end{aligned} \tag{2.9}$$

  Because of the model, as d augments the principal eigenvector of Z is asymptotically less indicative of v_0. To obtain a sparse representation Montanari suggests to impose the nonnegativity of v as prior knowledge, and thus solve NNPCA in 1D. Manopt implementation calculates the problem in the $d-1$ spherical manifold $\mathbf{S}^{d-1}$, that is the Stiefel manifold $\mathbf{St}^{1,d}$ (see [AMS09]).

  In this very particular case of FA, we do not possess M but only the covariance matrix MM^T with a gaussian noise. Nonetheless from this formulation we can say that the ONMF can be perceived as a generalisation of the NN-PCA.

Remarkably, we can plug ONMF into Manopt! Unfortunately, there is a catch. To have an image of the difference between Pompili's algorithm and Liu's algorithms, the first is a speedboat when the others are cruise boats. Liu is using many different parameters that must be tuned to each problem, using "classical" criteria: as long as there is an improvement in the iterates (evaluating the cost function every time) we keep running the algorithm, the ρ penalty criterion increases as the ϵ inner loop convergence criterion decreases proportionally. See [LB18]for the details.

More importantly, they store each constraint for every element of M separately, as their REPMS cannot use the squared Frobenius norm to only feed one as the QP. What about the RALM? We did not give a try. We can find traces of codes in Pompili's files that show he was working on it. We did not reach the further step about investigating the RALM.

The cruise boats cannot do much against the speedboat. For example with the Iris database used in example C1 , on our computer the solution for $k = 3$ plugging RQNN-PCA using our implementation takes less then 1 second to obtain a final result. On the other hand when we implement the NN-PCA-R model in Liu and Boumal's patch for Manopt it takes in the best case 40 seconds when tuning the parameters.

Example C2: the Iris Dataset

We cluster 150 flowers from a dataset containing 4 variables to separate them in 3 species. Compared to example C1 , as we are using penalised methods on Stiefel Manifold all the results orthogonal. But they are not necessarily non-negative any more. Both with the k-means and NN-PCA implemented in Manopt using the augmented Lagrangian RALM algorithm, starting with a random initialisation (in the Stiefel manifold) for k-means and with the SVD decomposition for NN-PCA we verify that we obtain a strictly orthogonal matrix in both cases, with only few slightly negative elements. Repeating the experiment, the final output is different with k-means when it is always the same with NN-PCA.

If we choose a random orthogonal initialisation for NN-PCA then we obtain solutions that are not as good as with the SVD+flip initialisation.

How the Matlab patch [LB18] is built

In 3 steps,

1. BOSS_.M Main file to solve a problem such as k-means, NN-PCA for 1D signal processing, or the ONMF I implemented. The parameters are set, the solver is chosen. It calls the client constraint.

2. CLIENTCONSTRAINT_.M There is 1 clientconstraint per problem. In it lies the description of the problem, feeding the functions f, ∇f, g_i, $\nabla g_i \forall i \in \mathcal{I}$, $\nabla h_j \forall j \in \mathcal{I}$. It calls the solver.

3. SOLVERS 4 solvers are implemented, all use RLBFGS for the inner iteration. First an augmented Lagrangian based on (RALM). Then 3 Riemannian exact penalty method (REPMS). As the objective function has a second order derivative not continuous in 0, one smooths it using the Huber-loss function, one using the log-exp trick, and one uses a modified RLBFGS method.

Pompili's algorithm defeats the current Manopt toolbox patch for constrained optimisation

- Pompili is implementing the basic quadratic penalty method whereas Liu developed the more advanced augmented Lagrangian and the exact penalty methods. The RQNN-PCA is advantaged. That is because of the way the constraints are fed. The four solvers of Liu need at input all the constraints separately, each element of W is non-negative (they can be written in $m \times k$ separate elements of size 1) and their gradients (they must be written in $m \times k$ separate elements of size $m \times k$). The spacial complexity is $O\left((mk)^2 + mk\right)$. If we want to use the ONMF to apply the example A3 ORL database, then we need 150000 elements of size 1 and 150000 elements of size 150000 to store the constraint, a total of more than 22 million . The solver of Pompili instead handles conveniently the Frobenius norm to input in the solver only one non-negative constraint of size 1 and its gradient of size $m \times n$. The spacial complexity is $O\left(mk\right)$. With the ORL database thus it needs to store "only" 150 001 elements to represent the constraints.

- Pompili uses a gradient method and not a RLBFGS in his quadratic penalty algorithm RQNN-PCA . Because of the design of his algorithm, we are not that much interested in converging in only a few outer loop iterations, we want to increase ρ with the number of iterations and the solution to move little by little towards nonnegativity. Computing a few approximate gradient steps per inner loop is sufficient. We ran many tests with RLBFGS, it was between twice slower or as fast compared to when using the gradient descent. Therefore because of the design of the algorithm it is not a default to use the gradient descent.

- The convergence criterion of Pompili is only based on the non-negativity of the iterates and not on many different parameters such as the norm of the gradient and the distance between two iterates. Recalling ONMF's figure 1.4, what we want in matrix W is to have one value non-negative in each column and the rest set to 0. If we have one value non-negative and most of the others slightly negative we can prospect that if we wait long enough for ρ to grow big all the negative elements will turn to 0 or will converge to very small negative values, but we don't need to wait as we already obtained the information about which element is non-negative and thus dominant.

2.4.4 Convergence of NN-PCA-R

Pompili wrote a proof of convergence for his P-ONPMF-R algorithm. We did study it thoroughly and there was no major flaws, except it is long. His result goes the following.

With algorithm 2 or 3, we develop a sequence of iterates x_k. Let x^* be an initial point of that sequence. Two behaviours of the sequence are possible:

- or it is a point where the non-negativity constraint is not enforced, x^* is infeasible but it is a stationary point
- or it is a point where the non-negativity constraint is enforced, x^* is feasible, then it is a KKT point with a unique Lagrange multiplier.

Pompili derived the conditions simply from the KKT and LICQ adapted to the Riemanian manifold.

In most cases the algorithm converges to a sequence in which there is one non-negative element per column, and all the others are close to zero but negative.

2.5 Conclusion

From the global context of NMF problems, we could understand how [PGAG14] was a breakthrough hard-enforcing orthogonality. The article has been well quoted but no-one really followed the track. No-one except for Asteris, suggesting to use the link ONMF has with NN-PCA to find a solution by enumeration in a reduced space from the SVD, naming it "subspace exploration". Pompili too had projects, from the OPNMF he would employ the Stiefel manifold to develop the P-ONPMF-R algorithm.

Solving constraint optimisation problems in manifolds has been recently developed. Owing to the examples available with the toolbox accompanying article [LB18], we could implement algorithms for the NN-PCA-R model, thus having an augmented Lagrangian solver at hand. After research we decided to present Pompili's P-ONPMF-R algorithm using Manopt under the PCA representation. We had to modify the solvers suggested in Manopt so that our problem could be solved as efficiently as Pompili with his custom made libraries, therefore forming the dilemma to employ a gradient method instead of the RLBFGS.

Because Pompili's code came raw with no explanations and some parameters badly tuned, because of the density of his Thesis [Pom12] with no transitions and introduction, because of the recent articles on the topic (many new interesting investigations appeared since 2017), and finally, because of his unique approach to the problem to make it efficient and tractable, it was a challenge to adapt and make his work accessible.

A Github repositery with the NN-PCA-R algorithm written in the canevas of Matlab patch for penalty methods in Manopt is accessible. We also wrote the NN-PCA-R in the format suggested by the authors, feeding separately all the nonnegativity constraints (one per element of the matrix) in the toolbox to solve small sized problems using REPMS and RALM.

Chapter 3

Case study: RQNN-PCA in Manopt

We implemented the RQNN-PCA quadratic penalty algorithm in Manopt.
As we have seen chapter 1 that (ONMF) is (OPNMF) i.e (NN-PCA) the first objective of this section is to verify numerically how our RQNN-PCA Manopt implementation proceeds. We will do so in regards to the Augmented Lagrangian ONPMFR algorithm already introduced by Pompili [PGAG14]. As our work is based on Pompili's never published P-ONPMF-R , we also would like to assert that the results obtained in Manopt with RQNN-PCA are at least as good as the ones we can obtain using his implementation.

We submit 2 codes with this thesis,

- POMPILI-ONMF that Pompili sent me with all his experiments. It is very generic as it is built to analyse many different algorithms and datasets, it is not very straightforward to use. We added our RQNN-PCA and used his structure to compare our implementation and results. Pompili asked us not to make the code open as it contains property materials, thus it is not available on Github.

- RQNN-PCA, our implementation following [LB18] using Manopt as a showcase. All the tests on real hyperspectral images were made with it. It is readily available on Github.

We introduce two criteria, the error and the negativity:

$$ER = \frac{||M||_F^2 - ||W^T M||_F^2}{||M||_F^2} \qquad NEG = \frac{||\min(W, 0)||_F}{||W||_F}$$

All the experiments were run on an Intel® Core™ i5-5300U CPU @2.30GHz with 8GB of RAM.

Standard settings for RQNN-PCA and P-ONPMF-R

- The outer loop criterion of convergence is only based on the non-negativity of the iterates:

$$NEG < \eta = 10^{-4}$$

- The inner loop criterion of convergence must follow "a decreasing sequence τ_k". In practice we keep this criterion constant, however we must tune it to each dataset. Setting $\beta = 1$ and GRADCONVTRESHOLD $= 100$:

$$\tau = \frac{\text{GRADCONVTRESHOLD}||M||_F}{\beta} = 100||M||_F$$

- We fix the penalty $\rho_0 = 100$ and the penalty rate $C = 1.1$

Standard settings for ONPMF

The parameters for ONPMF are kept as presented in article [PGAG14]:

$$\alpha_0 = 100, \quad \rho_0 = 100, \quad C = 1.01, \quad \eta = 10^{-3}$$

α is the parameter for the augmented Lagrangian update, ρ, C and η have the same definition as for the NN-PCA algorithm but they are set in a complete different way.

3.1 Benchmarking RQNN-PCA in Manopt using a synthetic dataset

Introducing ONPMF [PGAG14] Pompili showcased experiments on a synthetic dataset [1].

Synthetically building $\hat{M}_{m\times n}$, we test an algorithm to obtain an accurate representation $\hat{W}_{m\times k}$ of $W_{m\times k}$ given a Gaussian noise $N_{m\times n}$:

$$\hat{M}_{mn} = \hat{W}_{m\times k}\hat{H}_{k\times n} = W_{m\times k}H_{k\times n} + \epsilon N_{m\times n}$$

The experiment consists in checking how the results are affected by the noise as we augment ϵ, having built:

- $W_{m\times k}$ such that each column $W_{:i}$ $i = 1 \dots k$ represents the centroid of a non-negative cluster π_i.
  These are generated uniformly at random in the unit circle $[0, 1]^k$.

- $H_{k\times n}$ such that it meets the properties of ONMF (the orthogonality and nonnegativity of H imply 1 element non-zero per column), without enforcing the k-means constraint (all non-zero elements contained in a same row are equal). The reader can get the intuition by referring to figure 1.4.
  Each column of $M_{:j}$ is thus a multiple of its corresponding cluster centroid:

$$M_{:j} = \alpha W_{:k}$$

  where $\alpha > 0$ is picked uniformly at random in the interval [0.1, 1].

[1] To recompute the result we present using POMPILI-ONMF Matlab library, the heart of the code lies in RANDORTHMAT.M, it is called by LOADSYNTHDATASETBYNUMBER.M. We set the parameters in RUNEXPERIMENTS.M as *numItems* $= 450$, *dims* $= 10$, *numClusters* $= 6$, *noiseLevels* $=$ *logspace*$(-2, 0, 50)$ *illConds* $= [0]$ *and unbalances* $= [10]$.

Let $\{\pi'_i\}_{i=1}^k$ be the extracted clusters by an algorithm and $\{\pi_i\}_{i=1}^k$ the true clusters, the accuracy is defined as:

$$\text{Accuracy} = \max_{P\in[1,\dots,k]} \frac{1}{m \times n} \sum_{i=1}^{k} \left(|\pi_i \cap \pi'_{P(i)}|\right) \in [0,1]$$

with $[1,\dots,k]$ the set of permutations of $\{1,\dots,k\}$ (NMF find the clusters but not necessarily in the same order.)

Results For $k = 6$ clusters, $n = 450$ data points of $m = 10$ dimensions, we seek figure 3.1 how RQNN-PCA performs compared to ONPMF and P-ONPMF-R first without repetition.

- P-ONPMF-R lags behind as if it is k-means. We did not expect such a result. It was initialised with the same parameters as RQNN-PCA, the only difference is that one works with Pompili's environment and the other uses Manopt. We worked for months in Pompili's code with sometimes frustrations. These led us thriving in moving to Manopt. As we chose to make accessible the algorithm using Manopt, we leave the P-ONPMF-R and do not make any further investigation, noting that we obtained for example good results on the Hubble dataset.
- The behaviour of RQNN-PCA is similar to P-ONPMF-R:
  - When ϵ is very small, both obtain results close to a 100% accuracy but ONPMF seems to be better.
  - On the left ONPMF outperforms NNPCRA but not on the right. That is because the non-negativity criterion is by default more conservative in NNPCAR.
  - On average ONPMF takes 1.7s to converge, NN-PCA 5.5s with $\eta = 10^{-4}$ and 0.8s with $\eta = 10^{-4}$.
  - ONPMF needs more than 1000 outer loop iterations when RQNN-PCA needs more than 100. This is normal, as we have set $C = 1.01$ for ONPMF versus $C = 1.1$ for NN-PCA the number of iterations to have the penalty term big enough to influence objective function is different.

To be fair to ONPMF we run 10 repetitions with the parametrisation of ONPMF applied to both. The results are presented figure 3.2:

- We obtain the same behaviour as in figure 3.1. ONPMF is always with the same parametrization as it is our reference. What changes for RQNN-PCA compared to the left plot figure 3.1 is that we have decreased the penalty rate from $C = 1.1$ to $C = 1.01$. Thus it needs more than 1000 iterations and it takes on average 3.13 seconds to converge.

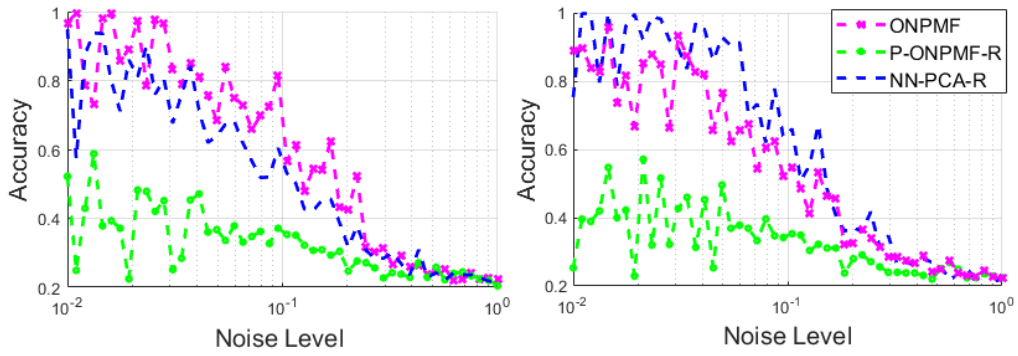

Figure 3.1: Accuracy of the 3 benchmarked algorithms given the noise level ϵ without repetition. On the left we have set the outer loop criterion η of P-ONPMF-R and RQNN-PCA from 10^{-4} to 10^{-3}. On the right is in the standard parametrisation.

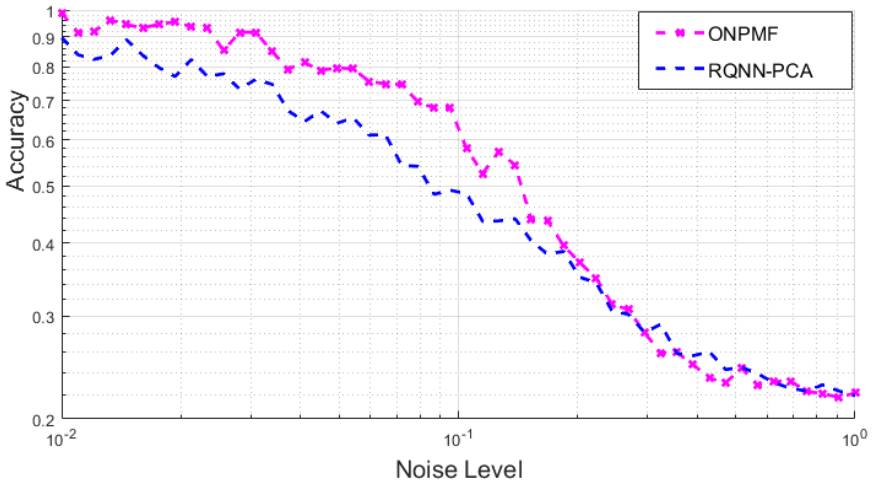

Figure 3.2: Accuracy of the 2 algorithms given the noise level ϵ with 10 repetitions. RQNN-PCA has been parametrised to the standard ONPMF settings, with $C = 1.1$ and $\eta = 10^{-3}$.

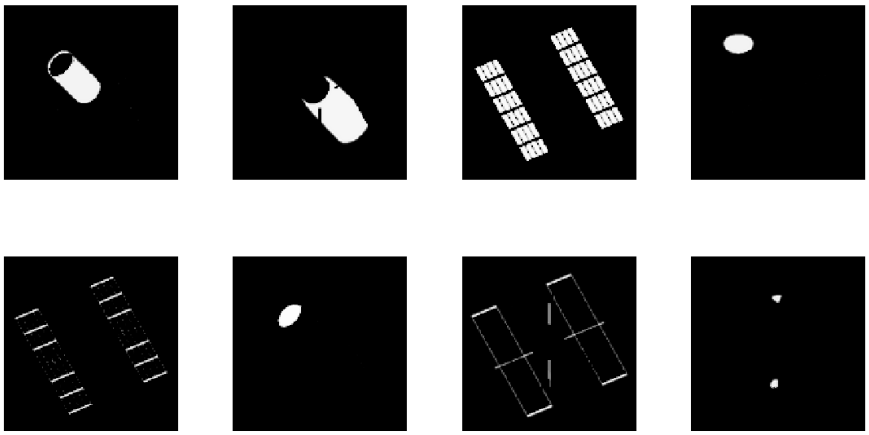

Figure 3.3: RQNN-PCA performs the hyperspectral clustering on the Hubble dataset.

This experiment allows us to validate RQNN-PCA with its standard settings. The quadratic RQNN-PCA can compete against the augmented Lagrangian ONPMF at the condition that it does not make too many outer-iterations, setting C large enough, and asking it to have an error at the end 10 times smaller than ONPMF. Building an augmented Lagrangian algorithm for the NN-PCA model would be a nice next step.

3.2 The synthetic Hubble hyperspectral dataset

We explain how the RQNN-PCA algorithm works and reacts using the simulated Hubble dataset [LNP+10]. It contains 100 evenly sampled spectral images of size 128×128 each. The spectral signatures range from 0.4 to $2.5\mu m$. There are 8 materials to separate. We can make the pure pixel assumption.

We start by running our RQNN-PCA algorithm 10 times with the standard setting. We obtain a convergence towards the same result that is presented figure 3.3, on average in 73s.

The objective is to tune the parameters adapting them to the dataset, to obtain the same results in less time.

1. Figure 3.4, we set$\rho_0 = 1$, $C = 1.05$ and we limit the number of inner iterations to 1, 5, 50 and 20 000. We see after 1000 outer iterations that up to the 200th iteration nothing happens, the ρ is too small for the weight of the nonnegative elements to have any influence, then suddenly the error explodes to finally converge at a higher level. This plot expresses very well our feeling about the name NN-PCA for the algorithm: we start at the minimum solution possible and then we push it to become non-negative. Impressively, even with the restriction of 1 inner iter per outer loop we obtain a result. Checking at

the duration of the algorithm after 1000 outer loop iterations, with 1 inner iteration maximum we did not yet obtain a final solution but it ran one minute less then the others.
$\rightarrow$ Asserting the number of inner iterations to be low is a key element to obtain quickly convergence in terms of seconds.

2. Figure 3.5. Setting $\rho = (1.05)^{200} = 1.74e^4$ to start where the SVD solution begins to change, fixing the maximum number of inner iterations to 20 000 After 400 outer iterations, we observe that augmenting the penalty rate from 1.05 to 2 decreases the number of iterations needed to obtain convergence. We assert that the algorithm converges to a point that does not satisfy the non-negativity constraint, therefore if we ask the non-negativity to be bellow 10^{-5} we will never reach it.

3. Figure 3.6. Tuning the maximum number of inner iterations with parameters $\rho_0 = 2$, $C = 1.74e^4$, we once again limit the number of inner iterations.

From all these tests we find that the time of convergence can vary a lot depending on the parameters. Its behaviour is always the same: suddenly the error ER augments as non-negativity NE becomes penalised. It then goes towards where we constrain it, tolerating to augment the reconstruction error by diminishing the non-negativity.

3.3 Conclusion

We successfully introduced ONMF to Manopt, solving it on the Stiefel manifold using a quadratic penalty method.

We did not use the readily available solvers that can be patched to Manopt from [LB18] as they are only fit to solve small problems. We had the ambition to be able to feed hyperspectral images and we succeeded. As we use a gradient method there is still room for improvement.

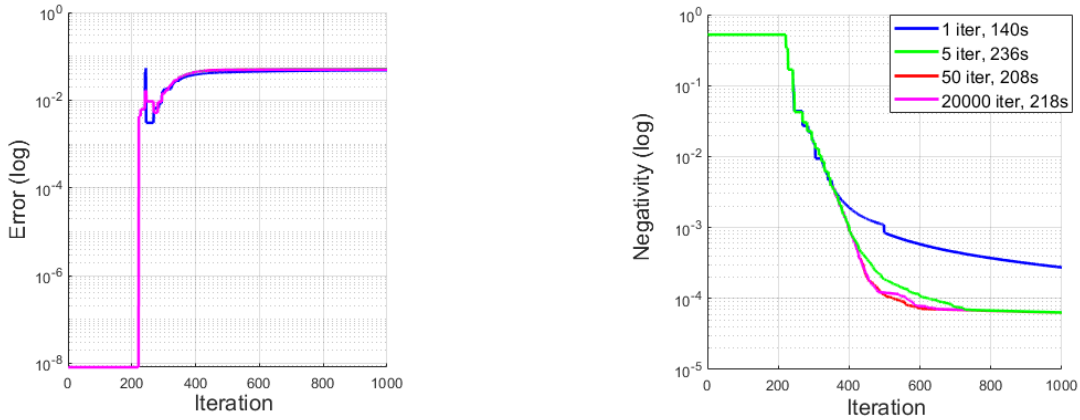

Figure 3.4: Tuning the maximum number of inner iterations with parameters $\rho_0 = 1$, $C = 1.05$.

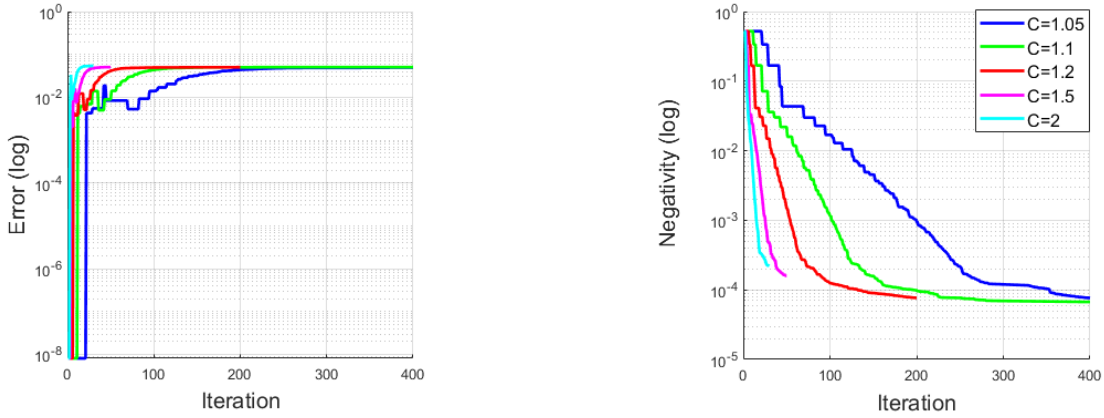

Figure 3.5: Tuning C with $\rho_0 = 1.74e^4$, letting free the maximum number of inner iterations.

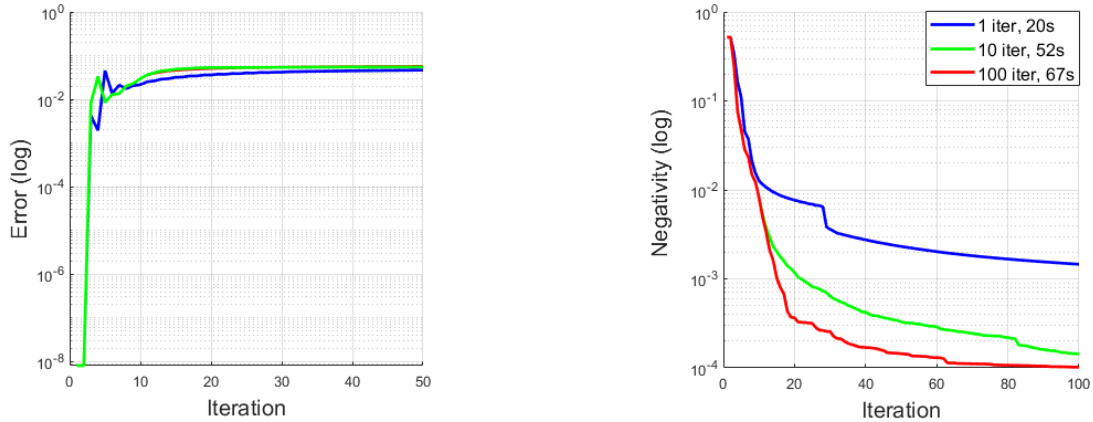

Figure 3.6: Tuning the maximum number of inner iterations with parameters $\rho_0 = 2$,$C = 1.74e^4$. Left we zoom on the 50 first iterations to better see the comportment.

Conclusion

The Master Thesis' purpose was to investigate the orthogonality constraint in the ONMF model. When hard enforcing it, we obtained **Theorem 1** that if M is non-negative and W orthogonal, then $H = W^T M$. From this result we explained how we fall back on the objective functions of both k-means seen as MF and PCA. This is the first fundamental result.

The three problems share the same orthogonality constraint, ONMF is k-means with one less constraint (the sum on the rows of W must be equal to one) and PCA with one more constraint (non-negativity).
The literature puts forth the connection between ONMF and k-means, almost never highlighting the bond with the PCA. At first we were concerned about it, wondering whether we were allowed to make such interpretation and how this could be written rigorously. NMF appeared as a way to learn by part, to make sense of the factorised data, whereas PCA is a statistical procedure for dimensionality reduction with minimal losses. By investigating the early literature we could understand how the ONMF model makes the bridge between the two.
In his Thesis, Pompili insists on the link between ONMF and k-means too. However the P-ONPMF-R algorithm he developed would make sense with a PCA interpretation, starting from the minimal loss representation that is SVD and enforcing the non-negativity progressively. Wanting to loose the least information possible, we can maximise a simpler objective function instead of the classical minimum problem considered by ONMF. Solving ONMF with hard orthogonality pushes the problem towards NN-PCA, choosing the latter representation is the second main contribution.

Considering ONMF with hard orthogonality is something still new. No-one but Asteris took advantage that we can obtain the ONMF by only solving for W. No-one but Pompili did it, using the geometry of the problem by solving it on the Stiefel manifold; he did not publish his algorithm. In the latter we need to account for the nonnegative constraints, but constraint manifold optimisation is not yet readily accessible. We saw that a way to deal with the nonnegativity is to use penalty methods, pushing it in the objective function.
Presently, research is focussed in adapting for the penalty methods the results of convergence from the Euclidean framework to different manifold frameworks.
Pompili used the quadratic penalty method with a gradient descent. New generic implementations of the Augmented Lagrangian and the exact penalty method are available in a patch for the Manopt Matlab toolbox, they use the RLBFGS. With

the quadratic penalty method, handling the constraints takes less memory space, allowing us to solve bigger sized matrices. After investigating the possibilities in Manopt we chose to adapt Pompili's algorithm under the name of RQNN-PCA. This is our third contribution.

We could investigate and show that our RQNN-PCA algorithm implemented in Manopt has the right behaviour, benchmarking it against ONPMF.

Bibliography

[AMS09] P-A Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[APD15] M. Asteris, D. Papailiopoulos, and A. Dimakis. Orthogonal nmf through subspace exploration. In C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 343–351. Curran Associates, Inc., 2015.

[ARD15] S. Aristeidis, S. Resnick, and C Davatzikos. Finding imaging patterns of structural covariance via non-negative matrix factorization. *NeuroImage 108*, pages 1–16, 2015.

[Bau15] C. Bauckhage. k-means clustering is matrix factorization. /, 2015. doi:arXiv:1512.07548.

[BEP15] G. Brown, S. Eberly, and P. Paatero. Methods for estimating uncertainty in pmf solutions: Examples with ambient air and water quality data and guidance on reporting pmf results. *Science of the total environment*, 2015.

[BH18] R. Bergmann and R Herzog. Intrinsic formulation of kkt conditions and constraint qualifications on smooth manifolds. /, 2018. doi:arXiv:1804.06214.

[BMAS14] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014. URL: http://www.manopt.org.

[BTBI73] A. Berman, L. B. Tomas, and A. Ben-Israel. Problems and solutions, edited by murray s. klamkin. *SIAM review*, 15(3), 1973.

[CMVW17] T. Carson, D. Mixon, S. Villar, and R. Ward. *Manifold optimization for k-means clustering*. International Conference on Sampling Theory and Applications, 2017.

[CZPA09] A. Cichocki, R. Zdunek, A. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Wiley, 2009.

[DHS05] C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. *Proc. SIAM Data Mining Conf*, 2005.

[Doo17] P. Dooren, van. Linma2380 - théorie des matrices - course at uclouvain, 2017.

[Fac94] The database of faces, 1994. doi:https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[Gil17] N. Gillis. Introduction to nonnegative matrix factorization. *SIAG/OPT Views and News 25*, 1:7–16, 2017.

[Hoy04] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[Hyv97] A. Hyvarinen. Independent component analysis by minimization of mutual information. *Helsinki University of Technology*, 1997.

[Hé08] K. Héberger. Chemoinformatics—multivariate mathematical–statistical methods for data evaluation. *Medical Applications of Mass Spectrometry*, 2008.

[Jol95] Ian T Jolliffe. Rotation of principal components: choice of normalization constraints. *Journal of Applied Statistics*, 22(1):29–35, 1995.

[Jol01] I.T. Jolliffe. *Principal Component Analysis, 2nd Edition*. Springer, 2001.

[KGB16] A. Kovnatsky, K. Glashoff, and M. Bronstein. Madmm: a generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision*, pages 680–696. Springer, 2016.

[KKT14] K. Kimura, M. Kudo, and Y. Tanaka. A fast hierarchical alternating least squares algorithm for orthogonal nonnegative matrix factorization. *JMLR: Workshop and Conference Proceedings*, 39:139–141, 2014.

[LB18] C. Liu and N. Boumal. Simple algorithms for optimization on riemannian manifolds with extra constraints. *Not yet accessible anywhere, thus put in appendix*, 2018. URL: https://github.com/losangle/Optimization-on-manifolds-with-extra-constraints.

[LHZ01] S. Li, X. Hou, and H. Zhang. Learning spatially localized, parts-based representation. *Computer Vision and Pattern Recognition - Proceedings of the 2001 IEEE Computer Society Conference*, 2001.

[LLHY18] Y. Liu, S. Lyu, M. Hou, and Q. Yin. The comparison between nmf and ica in pigment mixture identification of ancient chinese paintings. *Remote Sensing and Spatial Information Sciences*, 42:1169–1172, 2018.

[LNP+10] F. Li, M. Ng, R. Plemmons, S. Prasad, and Q. Zhang. Hyperspectral image segmentation, deblurring, and spectral analysis for material identification. In *Visual Information Processing XIX*, volume 7701, page 770103. International Society for Optics and Photonics, 2010.

[LS99] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 1999.

[LS01] D. Lee and H. Sung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2001.

[LZQ+18] J. Li, R. Zhu, A. Qu, H. Ye, and Z. Sun. Semi-orthogonal non-negative matrix factorization. *International Conference on Neural Information Processing 1*, 2018.

[Mir14] A. Mirzal. Nmf versus ica for blind source separation. *Advances in Data Analysis and Classification*, 2014.

[MR16] A. Montanari and E. Richard. Non-negative principal component analysis: Message passing algorithms and sharp asymptotics. *IEEE Transactions on Information Theory*, pages 1458–1484, 2016.

[NW06] Jorge Nocedal and Stephen J Wright. *Nonlinear Equations*. Springer, 2006.

[OY05] E. Oya and Z. Yang. Projective nonnegative matrix factorization for image compression and feature extraction. *Proceedings of the 14th Scandinavian Conference on Image Analysis SCIA*, pages 333–342, 2005.

[Par17] S. Paris. *CS598PS Machine Learning for Signal Processing*. University of Illinois at Urbana - Champaign, 2017.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, Series 6, 1901*, 2(11):559–572, 1901.

[PGAG14] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141:15–25, 2014.

[Pom12] F. Pompili. Nonnegative matrix factorization with orthogonality constraints: A geometric approach. *UNIVERSITÀ DEGLI STUDI DI PERUGIA Dottorato di Ricerca in Ingegneria dell'Informazione XXIV Ciclo*, 2012.

[PT94] P. Pattero and U. Tapper. Positive matric factorisation : a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5, 1994.

[RC97] P. Rachdawong and E. Christensen. Determination of pcb sources by a principal component method with nonnegative constraints. *Environmental science & technology*, 31(9):2686–2691, 1997.

[Sam59] A. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3:210–229, 1959.

[TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cogn. Neurosci.*, 3:71–86, 1991.

[Vav09] S. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM J. on Optimization*, 20:1364–1377, 2009.

[VGG18] A. Vandaele, F. Glineur, and N. Gillis. Algorithms for positive semidefinite factorization. *Computational Optimization and Applications*, pages 1–27, 2018.

[VGSea18] D. Varikuti, S. Genon, A. Sotiras, and et al. Evaluation of non-negative matrix factorization of grey matter in age prediction. *NeuroImage*, 2018.

[VL18] M. Verleysen and J. Lee. Elec2870 - machine learning: regression and dimensionality reduction, course at uclouvain, 2018.

[YC08] J. Yoo and S. Choi. Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifold. *Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning*, pages 140–147, 2008.

[YH16] Yadohisa and Hiroshi. Extensions of nonnegative matrix factorization for exploratory data analysis. 2016.

[YO10] Z. Yang and E. Oja. Linear and nonlinear projective nonnegative matrix factorization. 2010.

[YYO09] Z. Yuan, Z. Yang, and E. Oja. Projective nonnegative matrix factorization: Sparseness, orthogonality, and clustering. *Neural Process. Letters, submitted for publication*, 2009.

[YZS14] W. Yang, L. Zhang, and R. Song. Optimality conditions for the nonlinear programming problems on riemannian manifolds [accessed by pompili in 2011]. *Pacific Journal of Optimization*, 10(2):415–434, 2014.

[ZS07] R. Zass and A. Shashua. Nonnegative sparse pca. In *Advances in neural information processing systems*, pages 1561–1568, 2007.

[ZTS+16] W. Zhang, M. Tan, Q. Sheng, L. Yao, and Q. Shi. Efficient orthogonal non-negative matrix factorization over stiefel manifold. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1743–1752, 2016.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**