Hugo BREHIER

GRADUATION INTERNSHIP REPORT

HOST ORGANIZATION: LEME, Université Paris-Nanterre

INTERNSHIP SUBJECT: Sparse and Robust Principal Components Analysis

INTERSHIP LOCATION: Campus de Ville d'Avray

CITY: Ville d'Avray COUNTRY: FRANCE

Promotion: 2019/2020

Internship Supervisor: Arnaud Breloy

ENSAI Internship Advisor: Valentin Patilea

Contents

Foreword

Internship environment

The internship initially took place at the LEME laboratory, in the campus of Ville d'Avray (University Paris Nanterre). Starting from the second week, I had to work remotely due to the lockdown imposed by the Covid-19 situation. This inconvenience did not affect the program of the internship, as my mission is primarily about theoretical derivations, and validations that can be done with my personal computer. I had regular contact with my supervisors so that the internship could be conducted properly.

Notations

a	scalar
$\mathbf{a}$	vector
$\mathbf{A}$	matrix
$\langle \cdot, \cdot \rangle$	inner product (defaults are the dot product and Frobenius inner product)
$\|\cdot\|$	norm (default is the ℓ_2 norm)
$\mathrm{vec}\,(\mathbf{A})$	vectorization of a matrix obtained by vertical stacking of its columns
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of two matrices
$\mathbb{1}_{\{\cdot\}}$	Indicator function
$\mathrm{sgn}\,(\cdot)$	sign of a scalar
$(\cdot)_+$	maximum between a scalar and zero
$\mathrm{diag}\,(\cdot)$	diagonal matrix built from a set of elements
$\mathbf{A}^T$	transpose of a matrix
$\mathbf{A}^{-1}$	Inverse of a matrix
$\mathrm{rk}\,(\mathbf{A})$	rank of a matrix
$\mathbf{A}^\perp$	orthogonal complement of a subspace
$\mathrm{span}\,(\mathbf{A})$	span of the columns of a matrix
$\mathbf{I}_k$	Identity matrix of size $k \times k$

Definitions

Definition 0.0.1. *Real Stiefel Manifold*
For $k < p$, $\mathrm{St}(p,k)$ denotes the set of matrices whose columns form an orthonormal basis, i.e.:

$$\mathrm{St}(p,k) = \{\mathbf{U} \in \mathbb{R}^{p\times k} : \mathbf{U}^T\mathbf{U} = \mathbf{I}_k\}$$

Definition 0.0.2. *Eigenvalue Decomposition*
Any symmetric matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$ can be decomposed as :

$$\mathbf{A} \overset{\text{EVD}}{=} \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T,$$

where $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_p] \in \mathrm{St}(p,p)$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ contain respectively the eigenvectors and eigenvalues of $\mathbf{A}$, i.e., $\mathbf{A}\mathbf{q}_i = \lambda_i\mathbf{q}_i,\ \forall i \in [\![1,p]\!]$

Definition 0.0.3. *Singular Value Decomposition (SVD)*
Any matrix $\mathbf{A} \in \mathbb{R}^{p\times k}$ can be decomposed as:

$$\mathbf{A} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathrm{St}(p,p), \mathbf{V} \in \mathrm{St}(k,k), \mathbf{\Sigma} = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0) \in \mathbb{R}^{p\times k}$ with $\sigma_1 > \ldots > \sigma_r > 0$ the singular values of $\mathbf{A}$ and $r = \mathrm{rk}\,(\mathbf{A})$.

Definition 0.0.4. *Thin Singular Value Decomposition (TSVD)*
For $k < p$, any matrix $\mathbf{A} \in \mathbb{R}^{p\times k}$ can be decomposed as :

$$\mathbf{A} \overset{\text{TSVD}}{=} \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}^T,$$

where $\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathrm{St(p,k)}, \mathbf{\Sigma}_\mathrm{k} = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_\mathrm{k}) \in \mathbb{R}^{\mathrm{k}\times\mathrm{k}}$, and $\mathbf{V} \in \mathrm{St}(k,k)$. This decomposition corresponds to the (compact) SVD of $\mathbf{A}$ keeping only the first k columns of $\mathbf{U}$ and $\mathbf{\Sigma}$.

Definition 0.0.5. *Matrix norms*
Let $\mathbf{A} \in \mathbb{R}^{p\times k}$ with $[\mathbf{A}]_{i,j} = a_{ij}$, $p \geq 1$ and $q \geq 1$, we define the following:

Frobenius norm:	$\|\mathbf{A}\|_F = \sqrt{\mathrm{Tr}\,(\mathbf{A}^T\mathbf{A})} = \left(\sum_{i,j} \lvert a_{ij}\rvert^2\right)^{\frac{1}{2}}$
ℓ_p-norm:	$\|\mathbf{A}\|_p = \left(\sum_{i,j} \lvert a_{ij}\rvert^p\right)^{\frac{1}{p}}$
$\ell_{p,q}$-norm (mixed norm)	$\|\mathbf{A}\|_{p,q} = \left(\sum_i \left(\sum_j \lvert a_{ij}\rvert^p\right)^{p/q}\right)^{1/q}$
ℓ_0-pseudonorm	$\|\mathbf{A}\|_0 = \sum_{i,j} \mathbb{1}_{\{a_{ij}\neq 0\}}$

1 | Principal components analysis

1.1 Introduction

In many data sets, the information usually lies in a linear subspace of much lower dimension than the ambient observation space. This principle is illustrated in Figure 1.1, where a 2D subspace contains most of the data information whereas the ambient space is in 3D.

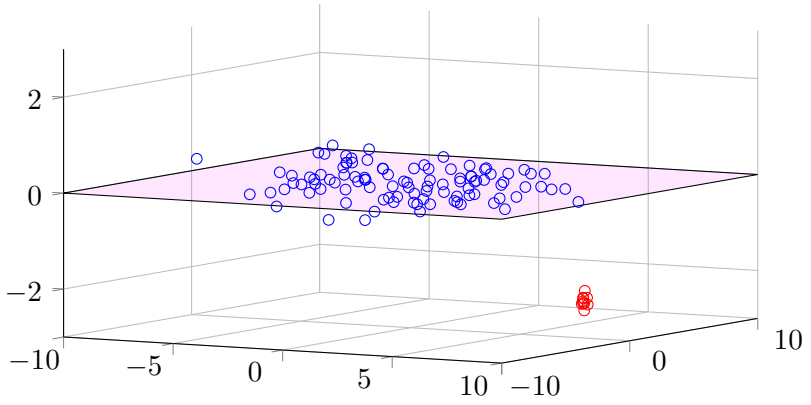

Figure 1.1: Illustration of data contained in a low-dimensional subspace (outliers represented in red).

Thus, learning algorithms are often interested in recovering this underlying structure from the data. Let the demeaned (centered) data set be denoted $\{\mathbf{x}_i\}_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^p$, $\forall i \in [\![1, n]\!]$, and aggregated in a matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$. The general aim is to find an orthonormal basis of a *linear subspace* $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathrm{St}(p, k)$ (cf. Definition 0.0.1) with $k \ll p$ such that the projection

$$\tilde{\mathbf{x}}_i = \mathbf{U}^T \mathbf{x}_i \in \mathbb{R}^k \tag{1.1}$$

offers a meaningful representation for the data, i.e., that captures the maximum amount of information. These basis vectors $[\mathbf{u}_1, \dots, \mathbf{u}_k]$ are generally referred to as the *loading vectors*, while the k entries of $\tilde{\mathbf{x}}_i$ are referred to as the *principal components*. This transformation can serve both for exploratory analysis (easier interpretation) and dimension reduction (pre-procesing to counter the curse of dimensionality).

The principal components analysis (PCA) [Jolliffe, 2002] is the most celebrated solution to this problem: it consists in selecting $\mathbf{U}$ as the k leftmost singular vectors (cf. Definition 0.0.3) of the data matrix $\mathbf{X}$. In the following, we will present three formulations of the subspace recovery problem leading to this solution.

1.2 Statistical formulation

This formulation is inspired from [Hotelling, 1933], which expresses the subspace recovery with the two following requirements:

1. find the k-dimensional subspace in which the projected data has maximal variance,
2. find a basis $\mathbf{U}$ such that $\tilde{\mathbf{x}}_i = \mathbf{U}^T\mathbf{x}_i$ produces uncorrelated entries with decreasing variance,

while assuming that $\{\mathbf{x}_i\}_{i=1}^n$ is a n-sample of a random variable $\mathbf{x}$ of mean $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ (centered) and covariance matrix $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{\Sigma}$.

The theoretical captured variance is $\mathbb{E}\left[\mathrm{Tr}\left(\mathbf{U}^T\mathbf{x}\mathbf{x}^T\mathbf{U}\right)\right] = \mathrm{Tr}\left(\mathbf{U}^T\mathbf{\Sigma}\mathbf{U}\right)$, so the first requirement yields the problem:

$$\max_{\mathbf{U}\in\mathrm{St}(p,k)} \mathrm{Tr}\left(\mathbf{U}^T\mathbf{\Sigma}\mathbf{U}\right). \tag{1.2}$$

A solution to this problem is given by any orthonormal basis that spans subspace associated to the k strongest eigenvectors of $\mathbf{\Sigma}$. This ambiguity is alleviated by the second requirement, which dictates to select $\mathbf{U}_k$ as (see Appendix A):

$$\mathbf{\Sigma} \overset{\text{EVD}}{=} [\mathbf{U}_k|\mathbf{U}_{rst}]\,\mathbf{\Lambda}\,[\mathbf{U}_k|\mathbf{U}_{rst}]^T \tag{1.3}$$

Nevertheless, the covariance matrix $\mathbf{\Sigma}$ is actually unknown. In practice, this parameter is replaced by an estimate, such as the sample covariance matrix:

$$\hat{\mathbf{\Sigma}}_{\text{SCM}} = \frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T, \tag{1.4}$$

which corresponds to the maximum likelihood estimator (MLE) of the covariance matrix in a Gaussian setting, i.e., assuming $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Notice that the corresponding estimated principal components can also be computed directly through the SVD of $\mathbf{X}$ (see Appendix A), which explains the standard association 'PCA = SVD of the data matrix'.

1.3 Geometric formulation

This formulation is inspired by [Pearson, 1901], and formulates the subspace recovery as the problem of finding the subspace to which data points are the 'closest to'.

Definition 1.3.1. *Distance to a subspace*
Let the distance of a sample $\mathbf{x}_i$ to a subspace spanned by $\mathbf{U} \in \mathrm{St}(p,k)$ be :

$$\mathrm{dist}\left(\mathbf{x}_i, \mathbf{U}\right) = \inf_{\mathbf{x}\in\mathbf{U}} \mathrm{dist}\left(\mathbf{x}, \mathbf{x}_i\right) = \|\mathbf{x}_i - \mathbf{\Pi}\mathbf{x}_i\| = \left\|\mathbf{x}_i - \mathbf{U}\mathbf{U}^T\mathbf{x}_i\right\| \tag{1.5}$$

Where $\mathbf{\Pi}$ is the orthogonal projector onto $\mathrm{span}\left(\mathbf{U}\right)$.

The subspace that minimizes the average squared distance to a data set $\{\mathbf{x}_i\}_{i=1}^n$ can thus be defined as the solution of the problem

$$\min_{\mathbf{U}\in\mathrm{St}(p,k)} \frac{1}{n}\sum_{i=1}^n \mathrm{dist}^2\left(\mathbf{x}_i, \mathbf{U}\right) \tag{1.6}$$

Notice that (see Appendix A)

$$\frac{1}{n}\sum_{i=1}^n \mathrm{dist}^2\left(\mathbf{x}_i, \mathbf{U}\right) = -\,\mathrm{Tr}\left(\mathbf{U}^T\hat{\mathbf{\Sigma}}_{\text{SCM}}\mathbf{U}\right) + \text{cst}. \tag{1.7}$$

Hence, the considered problem is equivalent to the previous formulation :

$$\max_{\mathbf{U}\in\mathrm{St}(p,k)} \mathrm{Tr}\left(\mathbf{U}^T\hat{\mathbf{\Sigma}}_{\text{SCM}}\mathbf{U}\right). \tag{1.8}$$

The resulting solution is, again, obtained with the k strongest eigenvectors of $\hat{\mathbf{\Sigma}}_{\text{SCM}}$ (or equivalently the k leftmost singular vectors of $\mathbf{X}$). Notice that this solution is valid up to a rotation $\tilde{\mathbf{U}} = \mathbf{U}\mathbf{R}$ for $\mathbf{R} \in \text{St}(k,k)$, since $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T = \mathbf{U}\mathbf{R}\mathbf{R}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T$. This ambiguity can be alleviated using the arguments from the previous section: the standard PCA sequentially maximizes the variance w.r.t. the principal component indexes (unique solution), while any rotation of this solution captures the same variance (although it becomes distributed among the principal components more evenly).

1.4 Low-rank formulation

This approach links subspace recovery to the 'best' (in the least-squared sense) low-rank approximation of the data matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$, which is formulated as :

$$\min_{\mathbf{U}\in\text{St}(p,k),\mathbf{Z}} \|\mathbf{X} - \mathbf{U}\mathbf{Z}\|_F^2. \tag{1.9}$$

Where $\mathbf{U} \in \mathbb{R}^{p\times k}$ and $\mathbf{Z} \in \mathbb{R}^{k\times n}$. A solution to this problem is given by the Eckart-Young Theorem, found in [Eckart and Young, 1936], as $\mathbf{U}^\star = \mathbf{U}_k$ and $\mathbf{Z}^\star = \mathbf{\Sigma}_k\mathbf{V}_k^T$ from the rank-k truncated SVD of $\mathbf{X}$:

$$\begin{aligned} &\mathbf{X} \overset{\text{SVD}}{=} \mathbf{U}_x\mathbf{\Sigma}_x\mathbf{V}_x^T \\ &\mathbf{U}_k = [\mathbf{u}_1^x, \dots, \mathbf{u}_k^x],\ \mathbf{\Sigma}_k = \text{diag}\,(\sigma_1^x, \dots, \sigma_k^x),\ \mathbf{V}_k = [\mathbf{v}_1^x, \dots, \mathbf{v}_k^x]. \end{aligned} \tag{1.10}$$

Thus, the subspace offering the best least-square approximation of the data is spanned by $\mathbf{U}_k$, allowing us to retrieve the standard PCA from this argument. Interestingly, replacing $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$ (optimal $\mathbf{Z}$ for fixed $\mathbf{U}$) into (1.9) leads to the problem (1.6), which highlights a direct link with the geometric approach. Finally, this formulation can also be put on perspective with orthogonal dictionary learning [Bao et al., 2013], where an additional penalty is applied on $\mathbf{Z}$ in order to promote some additional structure in the regression problem.

1.5 A visual example

We consider the synthetic toy example from [Abdi and Williams, 2010], where several wines and their characteristics are reported in the following table:

	Hedonic	Meat	Dessert	Price	Sugar	Alcohol	Acidity
Wine 1	14	7	8	7	7	13	7
Wine 2	10	7	6	4	3	14	7
Wine 3	8	5	5	10	5	12	5
Wine 4	2	4	7	16	7	11	3
Wine 5	6	2	4	13	3	10	3

The PCA applied to these samples shows that the first two loading vectors account for 94% of the total variance of this data. Figure 1.2 displays the wines in this new 2D space, as well as the correlation between principal components and the original variables. We can interpret that the first principal component stands for characteristics found in cheaper wines while the second principal component represents a wine sweetness.

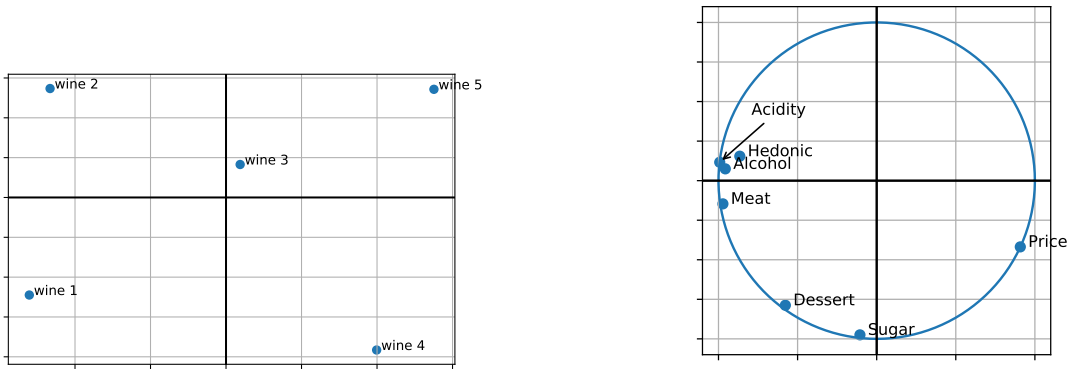

Figure 1.2: Representation of the wines in their two first principal components (left) and correlation between principal components and original variables (right).

1.6 Challenges related to PCA

Unfortunately, the SVD of the data matrix does not solve every aspect of the dimension reduction. In this manuscript, we will notably consider the two following issues:

- **Robustness**: It is well known that the sample covariance matrix is very sensitive to outliers. Thus, performing PCA from this estimate can lead to irrelevant solutions when the sample set is partially corrupted. Several approaches exist to overcome this issue, such as the use of robust estimators of the covariance matrix, or robust geometric costs [Ding et al., 2006, Lerman and Maunu, 2017]. These methods are generally referred to as *robust subspace recovery*.

- **Sparsity**: A limitation of PCA is that the loading vectors are dense, i.e. they create principal components that are linear combinations of all the original variables. Since these variables have a fundamental meaning, such as assets in finance or genes in biology, loading vectors with only a few non-zero entries could facilitate the statistical interpretation (variable selection). The problem of finding such vectors is referred to as *sparse PCA* and motivated numerous works over the past decades.

The aim of this work will therefore be to develop methods that enjoy best of both worlds, i.e., new robust sparse PCA algorithms.

2 | Robust subspace recovery

The traditional PCA tends to yield an inaccurate estimation of the 'true' (i.e. relevant) underlying subspace if outliers are present in the dataset. The general aim of robust subspace recovery [Lerman and Maunu, 2018] is to derive alternate subspace learning methods that alleviate this issue. In this scope, many approaches have been proposed, such as the use of robust covariance matrix estimators [Huber, 2004, Chapter 8], the use of a robust distance rather than the squared Euclidean one [Lerman and Maunu, 2017], or the recovery of a low-rank plus sparse decomposition of the data matrix [Candès et al., 2011].

In the following, we will detail two geometric formulations for robust subspace recovery, that will be involved in the derivation of the contribution in Chapter 4 .

2.1 Spherical PCA

Spherical PCA [Maronna et al., 2019, Section 6.11.1] considers minimizing the squared distance between U and normalized samples, i.e. using (1.6) with:

$$\text{dist}_{\text{sph}}\left(\mathbf{x}_i, \mathbf{U}\right) = \text{dist}\left(\mathbf{x}_i/|\mathbf{x}_i|, \mathbf{U}\right) \tag{2.1}$$

Following similar derivations as in the geometric approach, this leads to the EVD of the so-called normalized sample covariance matrix :

$$\hat{\mathbf{\Sigma}}_{\text{NSCM}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbf{x}_i \mathbf{x}_i^T}{\mathbf{x}_i^T \mathbf{x}_i} \tag{2.2}$$

or equivalently the SVD of the matrix concatenating normalized samples $\{\mathbf{x}_i/|\mathbf{x}_i|\}_{i=1}^n$. The resulting estimator is very robust but not accurate at low signal to noise ratio or when samples are impulsive.

2.2 Huber-type geometric subspace recovery

In [Lerman and Maunu, 2017] is proposed a robust cost function F with parameters q and δ inspired by [Huber, 1964]. The main idea is to set a quadratic loss for small errors, and a linear one for greater error values.

$$\begin{aligned} F_{q,\delta}(\mathbf{U}, \mathbf{X}) &= \sum_{\substack{i \\ \text{dist}^{2-q}(\mathbf{x}_i, \mathbf{U}) \geq q\delta}} \left(\text{dist}^q(\mathbf{x}_i, \mathbf{U})\right) + \sum_{\substack{i \\ \text{dist}^{2-q}(\mathbf{x}_i, \mathbf{U}) < q\delta}} \left(\frac{1}{2\delta}\text{dist}^2(\mathbf{x}_i, \mathbf{U}) + (q\delta)^{q/(2-q)} - \frac{(q\delta)^{2/(2-q)}}{2\delta}\right) \\ &= \sum_{i=1}^{n} \rho(\text{dist}(\mathbf{x}_i, \mathbf{U})), \end{aligned} \tag{2.3}$$

where

$$\rho(d) = \begin{cases} \frac{1}{2\delta}d^2 + (q\delta)^{q/(2-q)} - \frac{(q\delta)^{2/(2-q)}}{2\delta} & \text{if } d^{2-q} < q\delta \\ d^q & \text{if } d^{2-q} \geq q\delta \end{cases} \tag{2.4}$$

This objective function F generalizes some subspace recovery methods :

- for $(q = 2, \delta = 0)$ we retrieve the standard PCA through its geometric formulation.
- for $(q = 1, \delta = 0)$ we retrieve the geometric median subspace, also found in (vanilla) R1-PCA [Ding et al., 2006].

The function ρ represents a a trade-off between these two options, as it gets closer to the ℓ_1 norm when $q = 1$ as δ tends to zero (cf. Figure 2.1).

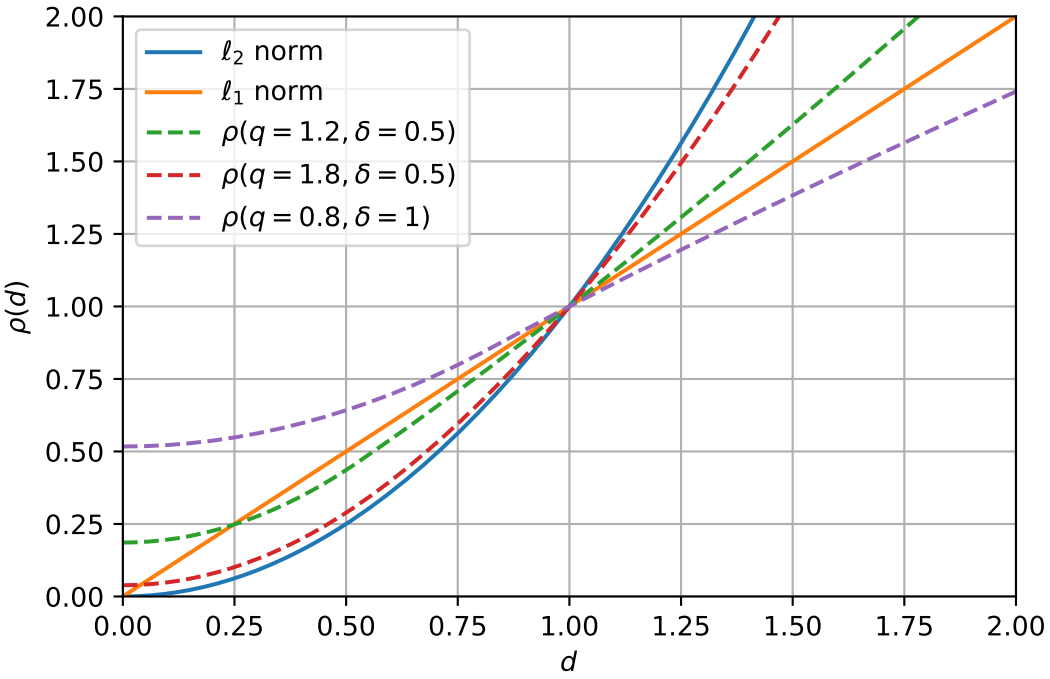

Figure 2.1: Different robust functions

The minimizer of F over $\mathbf{U}$ can be computed using an iterative algorithm (cf. Chapter 4 for more details). The resulting estimator is referred to as Fast Median Subspace (FMS) .

2.3 Simulations

In this section, we illustrate on synthetic data the interest of a robust cost function for subspace recovery in the presence of outliers.

2.3.1 Setting

In the following, synthetic data will be drawn according to a Gaussian probabilistic model.

Definition 2.3.1. *Probabilistic PCA Model (PPCA) [Tipping and Bishop, 1999]*
In this model, the samples are drawn as the sum of a structured signal plus noise :

$$\mathbf{x} = \mathbf{W}\mathbf{s} + \mathbf{n} \tag{2.5}$$

with $\mathbf{W} \in \mathbb{R}^{p \times k}$ *a full rank matrix,* $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^k$ *and* $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \in \mathbb{R}^p$*. We then have the representation* $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ *with* $\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$*.*

A link with the so-called factor model in econometrics [Jolliffe, 2002] is also to be pointed out since it generally considers (2.5) with $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi})$, where $\mathbf{\Psi}$ is still diagonal but with different variance for each entry.

Note that The maximum likelihood estimator (MLE) of $\mathbf{\Sigma}$ for this model corresponds to the regularization of the sample covariance matrix, where the last $p-k$ eigenvalues are averaged. Hence, from the point of view of subspace recovery (i.e., estimating span($\mathbf{W}$), in which $\mathbf{s}$ lies in), this model still leads to the standard PCA since eigenvectors of the sample covariance matrix are kept as estimators.

Next, the outliers will be generated according to a Haystack-type model, which consists in a specific mixture of PPCA models, defined as follows.

Definition 2.3.2. *Haystack Model [Lerman et al., 2012]*
In this model, samples $\{\mathbf{x}_i\}_{i=1}^n$ are drawn as inliers and outliers as follows:

$$\begin{aligned}\{\mathbf{x}_i\}_{i=1}^n &= \{\{\mathbf{x}_i^{in}\}_{i=1}^{n_{in}}, \{\mathbf{x}_i^{out}\}_{i=n_{in}+1}^{n}\} \\ \mathbf{x}^{in} &\sim \mathcal{N}(\mathbf{0}, \sigma_s^2 \mathbf{U}_0 \mathbf{U}_0{}^T + \mathbf{I}_p) \\ \mathbf{x}^{out} &\sim \mathcal{N}(\mathbf{0}, \sigma_o^2 \mathbf{U}_0^{\perp} (\mathbf{U}_0^{\perp})^T + \mathbf{I}_p)\end{aligned}$$

where $\mathbf{U}_0 \in \mathrm{St}(p,k)$ is the underlying signal subspace basis, σ_s^2 and σ_o^2 are respectively the signal and outlier to noise ratio (SNR), and $(n-n_{in})/n$ is the fraction of outliers in the dataset.

2.3.2 Results

We consider a Haystack model with $\mathbf{U}_0 \in \mathrm{St}(p,k)$ generated randomly by filling a $p \times k$ matrix with realizations of a standard univariate normal distribution and orthonormalizing it (by SVD to compute the polar factor, although it is also feasible by QR decomposition). We compare PCA, Spherical PCA and FMS (set with $q=1$ and $\delta=1$) in terms of average recovered fraction of energy (AFE), defined as :

$$\mathrm{AFE}(\hat{\mathbf{U}}) = \mathbb{E}\left[\mathrm{Tr}(\hat{\mathbf{U}}^T \mathbf{U}_0 \mathbf{U}_0{}^T \hat{\mathbf{U}})\right]/k \tag{2.6}$$

for a given estimator $\hat{\mathbf{U}}$. This expectation is evaluated through 100 Monte-Carlo runs.

First, we consider uncorrupted data following the PPCA model (Definition 2.3.1), the AFE is measured as the sample size n increases gradually from 15 to 150, with parameters $p=50$, $k=5$. The results are displayed in Figure 2.2.

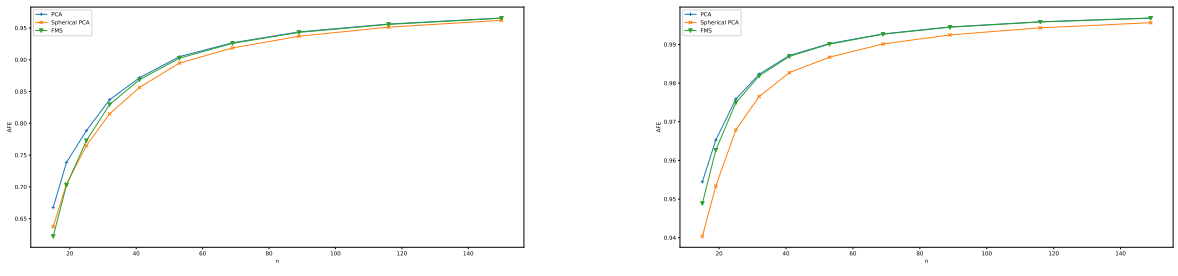

Figure 2.2: AFE of different algorithms as the sample size increases, for $\sigma_s^2 = 10$ (left) and $\sigma_s^2 = 100$ (right)

Second, with data from a Haystack-like Model (Definition 2.3.2) and fixing $n=100$, the AFE is measured as the number of outliers $n-n_{in}$ increases from 0 to $n/2$ and the SNR of outliers σ_o^2 increases from 0.1 to 20. 100 Monte-Carlo samples are drawn at each point. The results are in Figure 2.3.

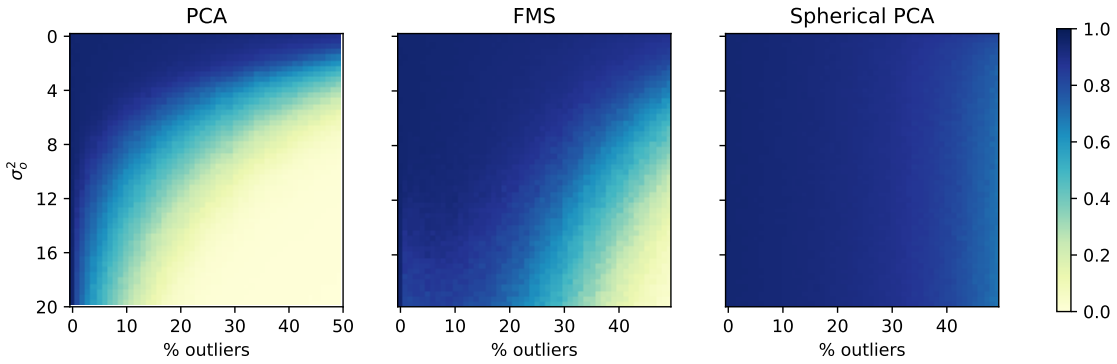

Figure 2.3: AFE of different algorithms depending on outlier percentage and power

From both of these experiments, we observe that the robust objective function of FMS yields an interesting trade-off between the performance of PCA when there is no outliers, and the robustness of Spherical PCA to numerous and powerful outliers.

3 | Sparse PCA

As mentioned previously sparse loading vectors are desirable in order to ease the statistical interpretation. Obtaining such a basis requires to modify the traditional PCA, which generally produces dense loading vectors. A straightforward approach consists in thresholding the entries of the principal components. However this simple process lowers the explained variance and impacts their orthonormality. Alternate methods considered applying rotations to the principal components (i.e., maintaining the explained variance fixed) in order to sparsify them [Kaiser, 1958].

More recent works considered exploring the sparsity versus explained variance trade-off directly through the formulation of optimization problems. A standard approach consists in adding a sparsity promoting penalty [Hastie et al., 2015, Rish and Grabarnik, 2014] in a standard PCA formulation (either statistical or geometric). Indeed, the popularity of the LASSO (see Appendix B) in regression analysis inspired researchers to apply a similar ℓ_1 regularization on PCA loading vectors $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k]$. A general approach would be to consider the problem:

$$\begin{array}{ll} \min\limits_{\mathbf{U}} & f(\mathbf{X}, \mathbf{U}) + \lambda \|\mathbf{U}\|_1 \\ \text{s.t.} & \mathbf{U} \in \mathrm{St}(p, k) \end{array} \tag{3.1}$$

With f a generic cost function. However, these approaches are non-trivial from the point of view of optimization, as satisfying the orthonormality constraints with ℓ_1 penalties is a challenging issue. Various relaxations, which are detailed in the following, were thus proposed to alleviate this issue.

3.1 Sequential methods

Sequential approaches consist in obtaining loading vectors one by one while using constraints of orthonormality and sparsity. This method thus aims to split a complex problem on $\mathrm{St}(p, k)$ in to a sequence of k simpler ones. Such approach can be described by the following algorithm:

Algorithm 1 Generic sequential SPCA

1: **Entry:** $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$
2: **for** $i = 1, \dots, k$ **do**
3: $\quad \mathbf{u}_i \leftarrow \arg\min_{\mathbf{u}} f(\mathbf{X}, \mathbf{u}) + \lambda \|\mathbf{u}\|_1$ s.t. $\|\mathbf{u}\| = 1, \langle \mathbf{u}, \mathbf{u}_j \rangle = 0 \quad \forall j < i$
4: **end for**
5: **Output:** $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{R}^{p \times k}$

An example of seminal formulation of this process is the SCoTLASS algorithm [Jolliffe et al., 2003] which aims to find loading vectors that sequentially maximise variance under sparsity constraints :

$$\begin{array}{ll} \max\limits_{\mathbf{u}_i} & \mathbf{u}_i^T \mathbf{X}\mathbf{X}^T \mathbf{u}_i \\ \text{s.t.} & \mathbf{u}_i^T \mathbf{u}_i = 1 \\ & \mathbf{u}_i^T \mathbf{u}_j = 0, \ \forall\, j < i \\ & \|\mathbf{u}_i\|_1 \leq \lambda \end{array} \tag{3.2}$$

The resolution of this series of problems is however complex and time consuming. To alleviate this issue [d'Aspremont et al., 2007] proposed a convex semidefinite programming relaxation of the problem (3.2).

3.2 Deflation methods

Deflation methods consist in sequential ones that relax the orthogonality constraint between loading vectors. An approximate orthonormality is however achieved by deflating the data matrix (generally, performing an orthogonal projection) at each step in order remove the influence of the previously obtained loading vectors. Such approach can be described by the following algorithm:

Algorithm 2 Generic deflation SPCA

1: **Entry:** $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$
2: **for** $i = 1, \dots, k$ **do**
3: $\quad \mathbf{u}_i \leftarrow \text{argmin}_{\mathbf{u}}\ f(\mathbf{X}, \mathbf{u}) + \lambda \|\mathbf{u}\|_1$ s.t. $\|\mathbf{u}\| = 1$
4: $\quad \mathbf{X} \leftarrow \mathbf{X} - \phi(\mathbf{u}_i)$ where ϕ is a deflation function [Mackey, 2009].
5: **end for**
6: **Output:** $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{R}^{p \times k}$

An example of deflation method, called rSVD [Shen and Huang, 2008], comes from the low-rank approximation formulation. The problem is formulated as a rank-one approximation of the data :

$$\begin{array}{ll} \min_{\tilde{\mathbf{u}}, \tilde{\mathbf{v}}} & \left\|\mathbf{X} - \tilde{\mathbf{u}}\tilde{\mathbf{v}}^T\right\|_F^2 + \lambda \|\tilde{\mathbf{u}}\|_1 \\ \text{s.t.} & \|\tilde{\mathbf{v}}\| = 1 \end{array} \tag{3.3}$$

Then, we set $\mathbf{u} = \tilde{\mathbf{u}}/\|\tilde{\mathbf{u}}\|$ to have a unit-norm loading vector. Finally, the following loading vectors can be sequentially obtained with the same procedure applied on the data matrix deflated by the solution $\tilde{\mathbf{u}}\tilde{\mathbf{v}}^T$. Note that for $\lambda = 0$, this procedure coincide with the rank k SVD of the data matrix and yields orthonormal loading vectors. Otherwise, this method does not enforce orthogonality, but it is hoped that the deflation allows for a close approximation. This method was also associated to G-Power [Journée et al., 2010] in [Hu et al., 2016], yielding a block method that recovers several loading vectors at once (rSVD-GPB).

3.3 Relaxing orthogonality

Several works considered a relaxation of (3.1) by dropping the orthogonality constraint on the loading vectors, i.e. they consider the problem

$$\begin{array}{ll} \min\limits_{\mathbf{U}} & f(\mathbf{X}, \mathbf{U}) + \lambda \|\mathbf{U}\|_1 \\ \text{s.t.} & ||\mathbf{u}_i|| = 1,\ \forall\ i \in [\![1, k]\!] \end{array} \tag{3.4}$$

In this case, the orthogonality is expected to occur naturally from the nature of the regression problem expressed by f, as an underlying subspace is intuitively better recovered if loading vectors do not align.

As main examples, we can cite SPCA [Zou et al., 2006], which applies this relaxation to a low-rank matrix recovery formulation, and SPCArt [Hu et al., 2016], which applies the same principle on a cost function based on an attach to a pre-estimated subspace basis i.e. we consider the problem :

$$\begin{array}{ll} \min\limits_{\mathbf{Z}, \mathbf{R}} & \|\mathbf{U} - \mathbf{Z}\mathbf{R}\|_F^2 + \lambda \sum_{i=1}^k \|\mathbf{z}_i\|_1 \\ \text{s.t.} & \forall i \quad \|\mathbf{z}_i\| = 1 \\ & \mathbf{R} \in \text{St}(k, k) \end{array} \tag{3.5}$$

i.e., SPCArt searches for a rotation $\mathbf{R}$ and a sparse basis $\mathbf{Z}$ that approximate the PCA loadings $\mathbf{U}$.

3.4 ℓ_0-norm proxies

In [Benidis et al., 2016] was proposed the use of smooth sparsity promoting penalties (proxies of the ℓ_0-norm) on the loading vectors. This results in a relaxation of the strict sparsity (i.e., only produces entries that are close to zero). However, a main interest of these proxies the possibility to derive majorization-minimization algorithms with practical iterations that do not relax the orthonormality.

3.5 Augmented Lagrangian methods

Including the augmented Lagrangian (cf. Appendix C) in the cost function has become a standard technique in order to relax a strict constraint in an optimization problem. In the scope of sparse PCA, two types of approaches have been considered:

- Including directly the augmented Lagrangian corresponding to the orthonormality constraint on the loading vectors, which was for example used in the ALSPCA algorithm [Lu and Zhang, 2009].
- Resorting to variable splitting method, i.e., putting the constraint only on a slack variable while constraining its equality with the main one. In this context, the process is also referred to as manifold ADMM [Lai and Osher, 2014, Kovnatsky et al., 2016] and has, for example, been used to derive the SOFAR algorithm [Uematsu et al., 2019].

4 | Contribution to robust sparse PCA

In this chapter, we propose a new algorithm for robust sparse PCA. This algorithm is driven by the formulation of an optimization problem that includes a robust geometric cost and a sparsity promoting penalty. The proposed resolution method through variable splitting draws its inspiration from the SOFAR algorithm in [Uematsu et al., 2019], which allows for a 'semi-relaxation' of the orthogonality constraint on the loading vectors. The corresponding estimate will be referred to as median sparse PCA (MSPCA). The notable features of the proposed method are the following:

- the use of a Huber-type geometric subspace recovery cost in order to ensure robustness to outliers in the dataset,
- the possibility to promote row-sparsity (variable selection) with an $\ell_{2,1}$-norm penalty without relaxing orthogonality of the loading vectors,
- the possibility to distribute the algorithm through the alternating direction method of multipliers (ADMM).

4.1 MSPCA: Formulation

We consider the problem

$$\begin{array}{ll} \min\limits_{\mathbf{U}} & \frac{1}{n} F_{q,\delta}(\mathbf{U}, \mathbf{X}) + \lambda \rho_s(\mathbf{U}) \\ \text{s.t.} & \mathbf{U} \in \mathrm{St}(p, k) \end{array} \tag{4.1}$$

where

- The data fitting term $F_{q,\delta}(\mathbf{U}, \mathbf{X})$ corresponds to the robust subspace recovery cost from [Lerman and Maunu, 2017] (cf. section 2.2 and (2.3)-(2.4)).
- The function $\rho_s(\cdot)$ is a sparsity promoting penalty that can be either the ℓ_1 or the $\ell_{2,1}$-norm.

However, it is well know that orthonormality constraints and non-smooth functions are hard to tackle simultaneously. Hence, we consider a reformulation of the problem through a variable splitting method, as discussed in [Lai and Osher, 2014, Kovnatsky et al., 2016]. The idea to apply a block-coordinate descent on separated variables while constraining them to be equal by using the augmented Lagrangian method (cf. Appendices C and D). Formally, we consider the problem

$$\begin{array}{ll} \min\limits_{\mathbf{U},\mathbf{V},\mathbf{\Gamma}} & L(\mathbf{U}, \mathbf{V}, \mathbf{\Gamma}) \\ \text{s.t.} & \mathbf{U} \in \mathrm{St}(p, k) \end{array} \tag{4.2}$$

with

$$L(\mathbf{U}, \mathbf{V}, \mathbf{\Gamma}) = \frac{1}{n} F_{q,\delta}(\mathbf{U}, \mathbf{X}) + \lambda \rho_s(\mathbf{V}) + \gamma \|\mathbf{U} - \mathbf{V}\|_F^2 + \langle \mathbf{\Gamma}, \mathbf{U} - \mathbf{V} \rangle \tag{4.3}$$

and where $\mathbf{\Gamma}$ is a matrix of Lagrange multipliers. Notice that the first terms in (4.3) correspond to the objective function of (4.1) splitted between two variables $\mathbf{U}$ and $\mathbf{V}$, while the additional terms correspond to the augmented Lagrangian method applied on the constraint $\mathbf{U} = \mathbf{V}$. Also note that the sparsity promoting penalty is now applied on an unconstrained variable, which will allow for practical derivations using proximal operators.

4.2 MSPCA: Algorithm

The proposed algorithm corresponds to the augmented Lagrangian and block coordinate descent (ALM-BCD) applied on (4.3). The resulting algorithm is summarized in the box Algorithm 4 and the derivation of each step is detailed in the following sections.

4.2.1 U-update

This step consists in fixing $\mathbf{V}$ and $\mathbf{\Gamma}$ and seeking for the update

$$\mathbf{U} = \underset{\mathbf{U}\in\mathrm{St}(p,k)}{\operatorname{argmin}} \quad L(\mathbf{U}, \mathbf{V}, \mathbf{\Gamma}), \tag{4.4}$$

i.e., we consider the problem

$$\begin{array}{ll} \min\limits_{\mathbf{U}} & \frac{1}{n} F_{q,\delta}(\mathbf{U}, \mathbf{X}) + \gamma \|\mathbf{U} - \mathbf{V}\|_F^2 + \langle \mathbf{\Gamma}, \mathbf{U} - \mathbf{V} \rangle \\ \text{s.t.} & \mathbf{U} \in \mathrm{St}(p, k) \end{array} \tag{4.5}$$

The solution to this problem admits no closed-form solution. We therefore tailor a majorization-minimization algorithm (cf. Appendix D) in order to evaluate a local minimum of the objective.

The majorization step is applied thanks to the following proposition:

Proposition 4.2.1. *The objective function in* (4.5) *is majorized at point* $\mathbf{U}_t$ *by the surrogate function*

$$g(\mathbf{U}|\mathbf{U}_t) = -\operatorname{Tr}\left(\mathbf{U}^T\left(\frac{q}{n}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}_t + 2\gamma\mathbf{V} - \mathbf{\Gamma}\right)\right) \tag{4.6}$$

with $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \ \ldots, \ \tilde{\mathbf{x}}_n]$ *and*

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i / \max\left(\operatorname{dist}^{(2-q)/2}(\mathbf{x}_i, \mathbf{U}_t), \sqrt{q\delta}\right). \tag{4.7}$$

Equality is achieved at $\mathbf{U}_t$.

Proof. First, we leverage the result of [Lerman and Maunu, 2017], stating that $F_{q,\delta}$ can be majorized at point $\mathbf{U}_t$ by the quadratic surrogate function

$$\begin{array}{rcl} H_{q,\delta}(\mathbf{U}, \mathbf{X}|\mathbf{U}_t) & = & \sum_{\operatorname{dist}^{2-q}(\mathbf{x}_i,\mathbf{U}_t)\geq q\delta} \left(\frac{q}{2}\frac{\operatorname{dist}^2(\mathbf{x}_i,\mathbf{U})}{\operatorname{dist}^{2-q}(\mathbf{x}_i,\mathbf{U}_t)} + \left(1-\frac{q}{2}\right)\operatorname{dist}^q(\mathbf{x}_i,\mathbf{U}_t)\right) \\ & & + \sum_{\operatorname{dist}^{2-p}(\mathbf{x}_i,\mathbf{U}_t)<q\delta} \left(\frac{1}{2\delta}\operatorname{dist}^2(\mathbf{x}_i,\mathbf{U}) + (q\delta)^{q/(2-q)} - \frac{(q\delta)^{2/(2-q)}}{2\delta}\right) \\ & = & \frac{q}{2}\sum_{i=1}^n \operatorname{dist}^2(\tilde{\mathbf{x}}_i, \mathbf{U}) + \text{cst}. \end{array} \tag{4.8}$$

with $\tilde{\mathbf{x}}_i$ defined in (4.7). Now, remark that

$$\sum_{i=1}^n \operatorname{dist}^2(\tilde{\mathbf{x}}_i, \mathbf{U}) = \sum_{i=1}^n \left\|\tilde{\mathbf{x}}_i - \mathbf{U}\mathbf{U}^T\tilde{\mathbf{x}}_i\right\|^2 = \left\|\tilde{\mathbf{X}} - \mathbf{U}\mathbf{U}^T\tilde{\mathbf{X}}\right\|_F^2 = -\operatorname{Tr}\left(\mathbf{U}^T\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}\right) + \text{const}. \tag{4.9}$$

Denote $f_B(\mathbf{U}) = \operatorname{Tr}\left(\mathbf{U}^T\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}\right)$. The function $-f_B(\mathbf{U})$ is quadratic concave, so it can be majorized at point $\mathbf{U}_t$ by its first order Taylor expansion [Sun et al., 2016]:

$$\begin{aligned} -f_B(\mathbf{U}) &\leq -f_B(\mathbf{U}_t) - \operatorname{Tr}\left(\left(\frac{\partial f_B(\mathbf{U}_t)}{\partial \mathbf{U}_t}\right)^T(\mathbf{U} - \mathbf{U}_t)\right) \\ &\leq -f_B(\mathbf{U}_t) - \operatorname{Tr}\left((2\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}_t)^T(\mathbf{U} - \mathbf{U}_t)\right) \\ &\leq -2\operatorname{Tr}\left(\mathbf{U}^T\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}_t\right) + \text{cst}. \end{aligned} \tag{4.10}$$

Adding the last two (linear) terms of (4.5) in their trace form, this linear majorizer yields the surrogate g as in (4.6). □

From the majorizer in Proposition 4.2.1, the minimization of the surrogate g is equivalent to:

$$\mathbf{U}_{t+1} = \underset{\mathbf{U}\in \mathrm{St}(p,k)}{\operatorname{argmax}} \operatorname{Tr}\left(\mathbf{U}^T\left(\frac{q}{n}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}_t + 2\gamma\mathbf{V} - \mathbf{\Gamma}\right)\right) \tag{4.11}$$

which corresponds to a orthogonal Procrustes problem.

Definition 4.2.1. *Orthogonal Procrustes problem [Golub and Van Loan, 2012, p. 327–328]*
Let $\mathbf{A} \in \mathbb{R}^{p\times k}, \mathbf{B} \in \mathbb{R}^{p\times q}$, an orthogonal Procrustes problem is expressed as

$$\begin{aligned}\mathbf{R} &= \underset{\mathbf{Q}\in \mathrm{St}(q,k)}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{B}\mathbf{Q}\|_F \\ &= \underset{\mathbf{Q}\in \mathrm{St}(q,k)}{\operatorname{argmax}} \operatorname{Tr}(\mathbf{Q}^T\mathbf{B}^T\mathbf{A}).\end{aligned} \tag{4.12}$$

A solution to this problem is the orthogonal factor of the polar decomposition of $\mathbf{B}^T\mathbf{A}$, expressed as

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \text{ with } \mathbf{B}^T\mathbf{A} \overset{\text{TSVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{4.13}$$

We also note that [Koschat and Swayne, 1991] consider the case where a diagonal matrix weights the residuals (of the difference) and derive an algorithm.

In our case $q = p$ and $\mathbf{B} = \mathbf{I}_p$, the problem is reduced to finding a nearest orthogonal matrix [Higham, 1989]. Thus, our solution can be computed as:

$$\mathbf{U}_{t+1} = \mathbf{U}_C\mathbf{V}_C^T \tag{4.14}$$

with $\mathbf{C} \overset{\text{TSVD}}{=} \mathbf{U}_C\mathbf{\Sigma}_C\mathbf{V}_C^T$ and $\mathbf{C} = \frac{q}{n}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}_t + 2\gamma\mathbf{V} - \mathbf{\Gamma}$. The resulting algorithm to compute the $\mathbf{U}$-update is summarized in the box Algorithm 3.

Algorithm 3 $\mathbf{U}$-Update $(\mathbf{U}, \mathbf{V}, \mathbf{\Gamma}, \gamma, q, \delta)$

Entry: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$
repeat
  Form $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n] \in \mathbb{R}^{p\times n}$ with $\tilde{\mathbf{x}}_i = \mathbf{x}_i / \max\left(\mathrm{dist}^{(2-q)/2}(\mathbf{x}_i, \mathbf{U}), \sqrt{q\delta}\right)\ \forall i = 1, \ldots, n$
  Compute $\mathbf{C} = \frac{q}{n}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U} + 2\gamma\mathbf{V} - \mathbf{\Gamma}$
  Compute the thin SVD: $\mathbf{C} = \mathbf{U}_C\mathbf{\Sigma}_C\mathbf{V}_C^T$
  Update $\mathbf{U}$ by : $\mathbf{U} = \mathbf{U}_C\mathbf{V}_C^T$
until convergence
Output: $\mathbf{U} \in \mathbb{R}^{p\times k}$

4.2.2 V-update

This step consists in fixing $\mathbf{U}$ and $\mathbf{\Gamma}$ and seeking for the update

$$\mathbf{V} = \underset{\mathbf{V}}{\operatorname{argmin}} \quad \gamma\left\|\tilde{\mathbf{U}} - \mathbf{V}\right\|_F^2 + \lambda\rho_s(\mathbf{V}), \tag{4.15}$$

with $\tilde{\mathbf{U}} = \mathbf{U} + \frac{1}{2\gamma}\mathbf{\Gamma}$, i.e. we have incorporated the inner product denoting the Lagrangian constraint inside the norm. This gives the problem the form of the proximal operator of the function ρ_s.

Definition 4.2.2. *Proximal operator [Parikh et al., 2014]*
Let $\mathbf{A}, \mathbf{B}, \in \mathbb{R}^{p\times k}$. The proximal operator (first introduced by [Moreau, 1965]) of a convex function ρ_s is defined as :

$$\mathrm{Prox}_{(\lambda/2\gamma)\rho_s}(\mathbf{A}) = \underset{\mathbf{B}}{\operatorname{argmin}} \quad \gamma\|\mathbf{A} - \mathbf{B}\|_F^2 + \lambda\rho_s(\mathbf{B}) \tag{4.16}$$

which is unique due to the strict convexity of the objective function.

In our case, two updates are thus possible depending on the chosen penalty:

- If ρ_s is the ℓ_1-norm, the solution is given by

$$\mathbf{V} = S_{\lambda/2\gamma}(\tilde{\mathbf{U}}) \tag{4.17}$$

with the entry-wise Soft-Thresholding operator, defined (entry by entry) by:

$$[S_{\lambda/2\gamma}(\tilde{\mathbf{U}})]_{ij} = \text{sgn}([\tilde{\mathbf{U}}]_{ij}) \left(|[\tilde{\mathbf{U}}]_{ij}| - \frac{\lambda}{2\gamma} \right)_+ \tag{4.18}$$

- If ρ_s is the $\ell_{2,1}$-norm, the solution is given by

$$\mathbf{V} = T_{\lambda/2\gamma}(\tilde{\mathbf{U}}) \tag{4.19}$$

with the row-wise Thresholding operator, defined (row by row) by:

$$[T_{\lambda/2\gamma}(\tilde{\mathbf{U}})]_{i:} = \left(1 - \frac{\lambda/2\gamma}{\left\| [\tilde{\mathbf{U}}]_{i:} \right\|} \right)_+ [\tilde{\mathbf{U}}]_{i:} \tag{4.20}$$

4.2.3 Γ-update

This step consists in the standard dual update of ALM for the matrix $\mathbf{\Gamma}$ (gradient ascent) :

$$\mathbf{\Gamma} + 2\gamma(\mathbf{U} - \mathbf{V})$$

4.2.4 Final algorithm

The full algorithm is summarized in the box Algorithm 4. An additional optional step consists in multiplying γ by a constant factor $\nu > 1$ at each iteration, which allows for a better convergence towards a feasable solution [Uematsu et al., 2019].

Algorithm 4 ALM-BCD for MSPCA $(k, \nu, \lambda, \gamma, q, \delta)$

Entry: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$
Initialize $\mathbf{U} \in \mathbb{R}^{p \times k}$ by the first k loadings of the PCA of $\mathbf{X}$
Initialize $\mathbf{V} \in \mathbb{R}^{p \times k}$ randomly
Initialize $\mathbf{\Gamma} \in \mathbb{R}^{p \times k}$ by a matrix of zeros
repeat
    Update $\mathbf{U}$ by following Algorithm 3
    Update $\mathbf{V}$ by following (4.18) or (4.20)
    Update $\mathbf{\Gamma}$ by : $\mathbf{\Gamma} + 2\gamma(\mathbf{U} - \mathbf{V})$
    Update (optional) γ by : $\nu\gamma$
until convergence
Output: $\mathbf{U} \in \mathbb{R}^{p \times k}$

4.3 dMSPCA: Distributed version via consensus ADMM

We make use of ADMM and its extension to global consensus problems (reviewed in Appendix C) to develop an algorithm that can be distributed across a network. We hereby call this algorithm dMSPCA. The main interest from this formulation is that data can be separated into m subsets $\{\mathbf{X}_i\}$ (of size n_i) that are not located at the same node. Formally, we consider the problem

$$\begin{array}{ll} \min\limits_{\{\mathbf{U}_i\},\mathbf{V}} & \sum_{i=1}^m \frac{1}{n_i} F_{q,\delta}(\mathbf{U}_i, \mathbf{X}_i) + \lambda \rho_s(\mathbf{V}) \\ \text{s.t} & \mathbf{U}_i \in \mathrm{St}(p,k) \\ & \mathbf{U}_i = \mathbf{V}, \quad \forall i = 1, \dots, m \end{array} \tag{4.21}$$

We consider that only the i-th processing unit has access to the subset $\mathbf{X}_i$. On the other hand, $\mathbf{V}$ is called the *global consensus variable* and constrains every local variable, driving a consensus.

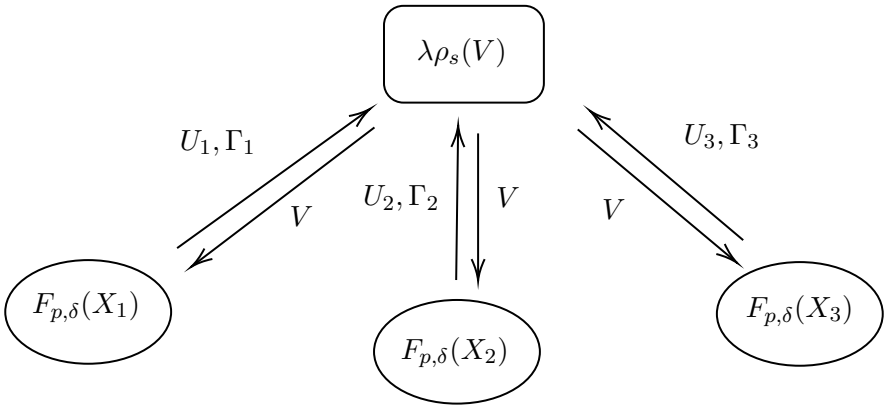

Figure 4.1: Illustration of process separation

The augmented Lagrangian associated to the problem (4.21) is:

$$\begin{aligned} L_p(\{\mathbf{U}_i\}, \mathbf{V}, \{\mathbf{\Gamma}_i\}) &= \lambda \rho_s(\mathbf{V}) + \sum_{i=1}^m \left(\frac{1}{n_i} F_{q,\delta}(\mathbf{U}_i, \mathbf{X}_i) + \gamma \|\mathbf{U}_i - \mathbf{V}\|_F^2 + \langle \mathbf{\Gamma}_i, \mathbf{U}_i - \mathbf{V} \rangle \right) \\ &= \lambda \rho_s(\mathbf{V}) + \sum_{i=1}^m l_p(\mathbf{U}_i, \mathbf{V}, \mathbf{\Gamma}_i) \end{aligned} \tag{4.22}$$

Where $\mathbf{\Gamma}_i$ is the local variable of Lagrange multipliers associated to $\mathbf{U}_i$. Following the ADMM, the $(t+1)$-th iteration of the distributed algorithm is as follow:

$$\begin{aligned} \mathbf{U}_i^{t+1} &= \operatorname*{argmin}_{\mathbf{U}_i \in \mathrm{St}(p,k)} \quad l_p(\mathbf{U}_i, \mathbf{V}^t, \mathbf{\Gamma}_i^t) \quad \forall i = 1, \dots, m \\ \mathbf{V}^{t+1} &= \operatorname*{argmin}_{\mathbf{V}} \quad L_p(\{\mathbf{U}_i^{t+1}\}, \mathbf{V}, \{\mathbf{\Gamma}_i^t\}) \\ \mathbf{\Gamma}_i^{t+1} &= \mathbf{\Gamma}_i^t + 2\gamma(\mathbf{U}_i^{t+1} - \mathbf{V}^{t+1}) \quad \forall i = 1, \dots, m \end{aligned} \tag{4.23}$$

The $\mathbf{U}_i$-updates and $\mathbf{\Gamma}_i$-updates are identical to the ones already covered in the previous section. Crucially, they can be processed separately and in parallel. The $\mathbf{V}$-update is slightly modified as follows:

$$\mathbf{V}^{t+1} = \operatorname*{argmin}_{\mathbf{V}} \quad \lambda \rho_s(\mathbf{V}) + \gamma m \left\| \mathbf{V} - \widetilde{\mathbf{U}}^{t+1} \right\|_F^2 \tag{4.24}$$

with $\widetilde{\mathbf{U}}^{t+1} = \frac{1}{m} \sum_{i=1}^m \left(\mathbf{U}_i^{t+1} + \frac{1}{2\gamma} \mathbf{\Gamma}_i^t \right)$. Again, the update corresponds to a proximal operator and depends on the chosen penalty:

$$\begin{array}{ll} \ell_1\text{-norm:} & \mathbf{V}^{t+1} = S_{\lambda/2\gamma m}(\widetilde{\mathbf{U}}^{t+1}) \quad \text{defined in (4.18)} \\ \ell_{2,1}\text{-norm:} & \mathbf{V}^{t+1} = T_{\lambda/2\gamma m}(\widetilde{\mathbf{U}}^{t+1}) \quad \text{defined in (4.20)} \end{array} \tag{4.25}$$

The full algorithm is summarized in the box Algorithm 5.

Algorithm 5 Consensus ADMM for dMSPCA $(k, \nu, \lambda, \gamma, q, \delta, m)$

Entry: $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_m] \in \mathbb{R}^{p \times n}$
Initialize $\mathbf{U}_i \in \mathbb{R}^{p \times k}$ by the first k loadings of the PCA of $\mathbf{X}_i \quad \forall i = 1, \ldots, m$
Initialize $\mathbf{V} \in \mathbb{R}^{p \times k}$ randomly
Initialize $\mathbf{\Gamma}_i \in \mathbb{R}^{p \times k}$ by a matrix of zeros $\quad \forall i = 1, \ldots, m$
repeat
    Update $\mathbf{U}_i$ by following Algorithm 3 using $\mathbf{X}_i, \mathbf{\Gamma}_i \quad \forall i = 1, \ldots, m$
    Update $\mathbf{V}$ by following (4.25)
    Update $\mathbf{\Gamma}_i$ by : $\mathbf{\Gamma}_i + 2\gamma(\mathbf{U}_i - \mathbf{V}) \quad \forall i = 1, \ldots, m$
    Update γ by : $\nu\gamma$
until convergence
Output: $\mathbf{U} \in \mathbb{R}^{p \times k}$

5 | Experiments

In this chapter, the MSPCA algorithm (with $F_{q,\delta}$ as $q = 1$ and $\delta = 1$) is compared to two state of the art algorithm from [Hu et al., 2016, algorithms 1 and 3], namely SPCArt and rSVD-GPB. Experiments are conducted both on Monte-Carlo simulations and real data[1].

5.1 Performance criteria

For a given matrix of loading vectors $\mathbf{U} \in \mathbb{R}^{p\times k}$, we study the following criteria.

Recovered energy

For Monte-Carlo simulations, the energy captured by the subspace basis estimates will be measured using the AFE (2.6). For a real data matrix denoted $\mathbf{X}_\mathrm{r}$, the 'true' underlying signal subspace is actually unknown. Therefore, we will alternatively consider the cumulative percentage of explained variance (CPEV):

$$\mathrm{CPEV}(\mathbf{U}) = \mathrm{Tr}(\mathbf{U}^T\mathbf{X}_\mathrm{r}\mathbf{X}_\mathrm{r}^T\mathbf{U})/\mathrm{Tr}\left(\mathbf{X}_\mathrm{r}\mathbf{X}_\mathrm{r}^T\right), \tag{5.1}$$

which will be studied either globally and for each loading vector.

Sparsity of the subspace basis

The degree of sparsity (SP) of the subspace basis will be measured by the percentage of zero-valued entries

$$\mathrm{SP}(\mathbf{U}) = 1 - \frac{\|\mathbf{U}\|_0}{p \times k}. \tag{5.2}$$

which will be studied either globally and for each loading vector. To account for the variable selection capacity of a method, we will also consider the degree of row-sparsity (RowSP), measured by the percentage of rows (within the basis) which are constituted of zeros only:

$$\mathrm{RowSP}(\mathbf{U}) = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}_{\left\{\|[\mathbf{U}]_{i:}\|_0=0\right\}} \tag{5.3}$$

Note that, in practice, we will consider equal to zero not only true zeros, but also values lower than the threshold $\epsilon = 1e^{-10}$.

Orthogonality of the subspace basis

The non-orthogonality (NOR) of a subspace basis will be measured using:

$$\mathrm{NOR}(\mathbf{U}) = \left\|\mathbf{U}^T\mathbf{U} - \mathbf{I}_k\right\|_F \tag{5.4}$$

[1] All algorithms have been coded in Python, including the usage of Numpy for Linear Algebra, Numba for some optimization, JobLib for multicore computation.

5.2 Simulation study on synthetic data

The data generating process follows the PPCA model from Definition 2.3.1. The sparse signal subspace basis $\mathbf{U}_0$ is generated as

$$\mathbf{U}_0 = \begin{bmatrix} \mathbf{U}_d \\ \mathbf{0} \end{bmatrix},$$

Where $\mathbf{U}_d \in \mathbb{R}^{d\times k}, d \leq p$, is an orthogonal basis and $\mathbf{0} \in \mathbb{R}^{(p-d)\times k}$ is a matrix of zeros.

5.2.1 General performance

Let : $n = 200, p = 200, d = 0.2 \times p, k = 20$ and $\sigma_s^2 = 16$. We vary the sparsity penalty parameter λ and compute the AFE, SP and NOR over 200 Monte-Carlo simulations.

Figure 5.1 displays the results when all algorithms use ℓ_1 norm sparsity penalization. Figure 5.2 shows the results when MSPCA algorithm use an $\ell_{2,1}$-norm sparsity penalization (which can not be used with the other methods). From this experiment, we can observe that MSPCA achieves state of the art performance (except for high degrees of sparsity, i.e. where the performance drop occurs), while not relaxing the orthogonality property of the loading vectors.

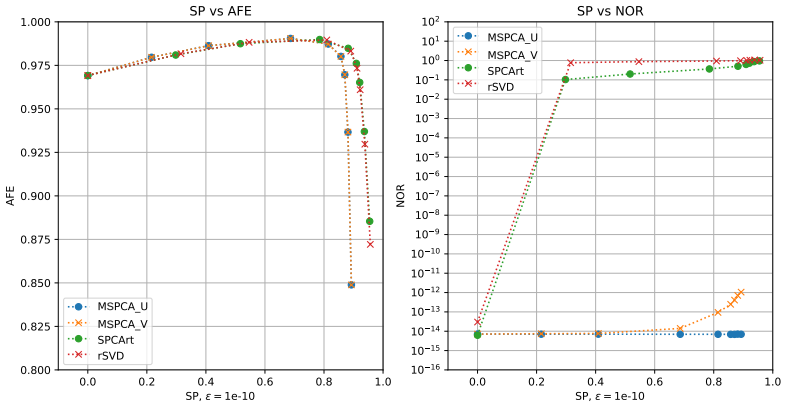

Figure 5.1: (SP,AFE) and (SP,NOR) depending on λ, with ℓ_1 norm

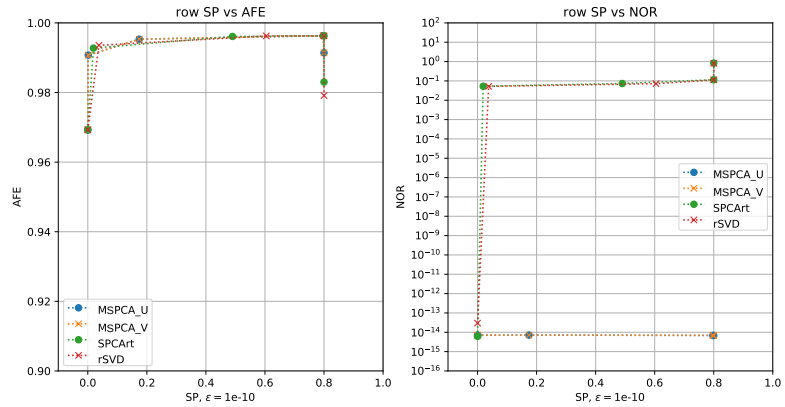

Figure 5.2: (Row SP,AFE) and (Row SP,NOR) depending on λ, with $\ell_{2,1}$ norm

5.2.2 Robustness

In this experiment, we are interested in testing robustness to outliers: the data is generated using the Haystack-type model described in Definition 2.3.2. We compare again MSPCA to SPCArt and rSVD-GPB as well a Spherical PCA. We also test MSPCA without sparsity constraint, which corresponds to the FMS. We vary the outliers SNR σ_o^2 and the number of outliers $n - n_{in}$ for fixed $p = 100, n = 100, k = 15, d = 0.2p$ and $\sigma_s^2 = 10$. We then compute the AFE on 100 Monte Carlo draws at each point.

In Figure 5.3 we initialize all algorithms with standard PCA. In Figure 5.4 we initialize rSVD-GPB and MSPCA with the Spherical PCA. From this experiment, we can observe that MSPCA achieves an interesting performance-robustness trade-off, while allowing for the inclusion of a sparsity promoting penalty.

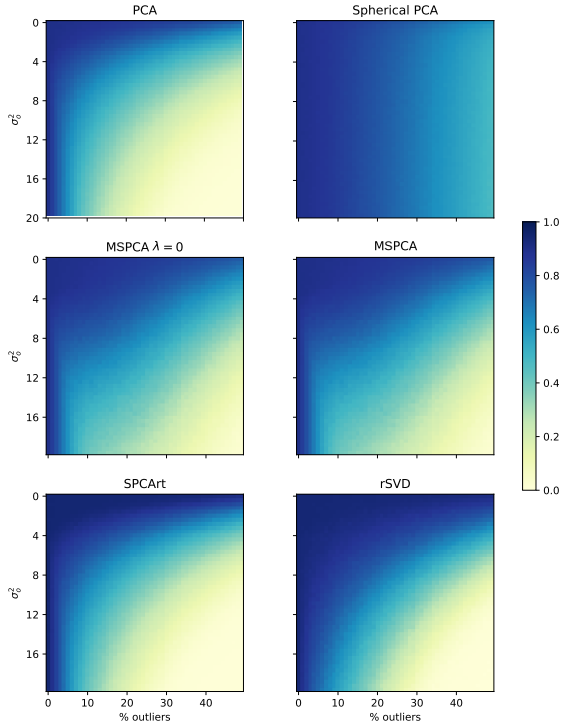

Figure 5.3: AFE of different algorithms depending on outlier percentage and power

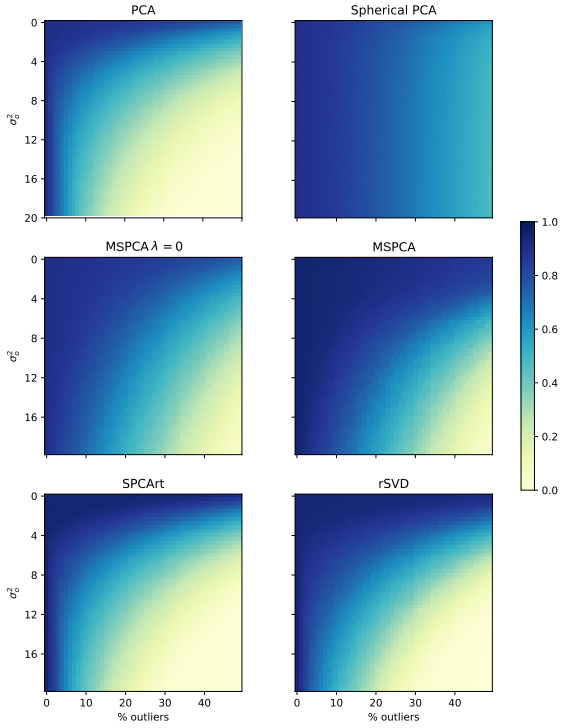

Figure 5.4: AFE of different algorithms depending on outlier percentage and power, with Spherical PCA initialization.

5.3 Study on real data: gene dataset

We consider the microarray dataset from [Golub et al., 1999]. This dataset consist of 47 patients with acute lymphoblastic leukemia (ALL) and 25 patients with acute myeloid leukemia (AML). Each of the 72 patients had bone marrow samples obtained at the time of diagnosis. Furthermore, the observations have been assayed with Affymetrix Hgu6800 chips, resulting in 7129 gene expressions (Affymetrix probes). This data set has become a benchmark for sparse PCA, as it is one of the most widely studied and cited microarray data set.

After centering the data, we select $k = 6$ and compare the CPEV (and NOR) versus sparsity trade-off that can be achieved by MSPCA, SPCArt and rSVD-GPB (all using ℓ_1 norm sparsity penalization). The results are displayed in Figure 5.5, where we observe that MSPCA reaches the same performance as state-of-the-art algorithms on real data while maintaining orthogonality of the loading vectors.

For further analysis, we compare the CPEV and SP for each loading vector, as the sparsity pattern may give a contrasted picture of the general performance (i.e. more sparsity in less important loading vectors). We select an equivalent CPEV-SP ratio (third to last point on Figure 5.5), corresponding to $\lambda = \frac{1}{\sqrt{p}}$ for SPCArt and rSVD-GPB and $\lambda = 100$ for MSPCA. The results are displayed in Figures 5.7 and 5.8, while Figure 5.6 also compares the correlation of principal components.We observe that MSPCA's first loading vector is slightly denser, however it preserves better the uncorrelatedness of the principal components.

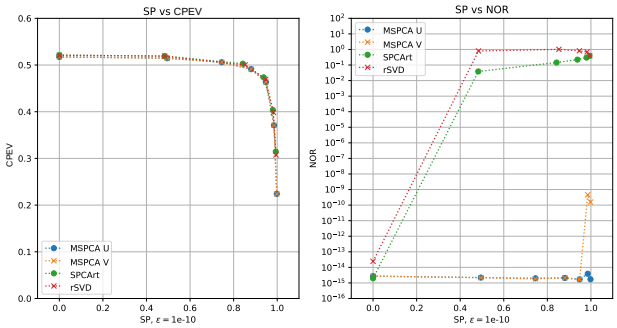

Figure 5.5: (SP, CPEV) and (SP,NOR) depending on λ with ℓ_1 norm

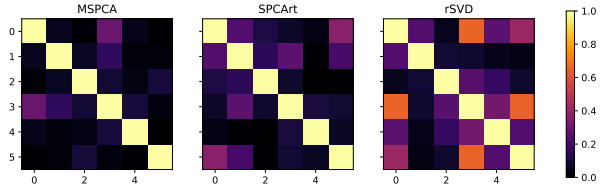

Figure 5.6: Correlations of Principal Components with ℓ_1 norm

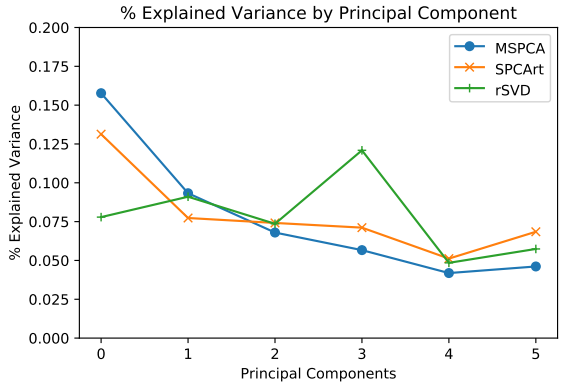

Figure 5.7: Explained variance by principal component with ℓ_1 norm

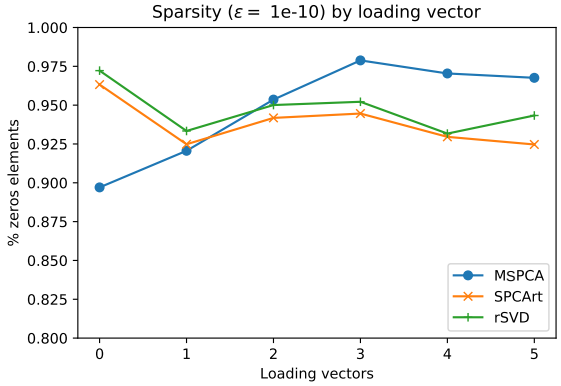

Figure 5.8: Sparsity of the loading vectors with ℓ_1 norm

We now use $\ell_{2,1}$-norm penalization for MSPCA and compare sparsity row-by-row in Figure 5.9. We observe that MPSCA can achieve better performance in terms of row-sparsity, which is due to the fact that the state of the art does allow for the used of $\ell_{2,1}$-norm penalty.

To conduct further analysis, we select $\lambda = 250$ for MSPCA , the third to last point on Figure 5.9. The results for individual vectors are in Figures 5.10, 5.11 and 5.12. This time, MPSCA's first loading vector is sparse. Altogether, MSPCA's loading vectors are capturing variance in an ordered manner (contrarily to rSVD-GPB) and remain orthogonal to each other (contrarily to SPCArt) while achieving similar performance of explained variance.

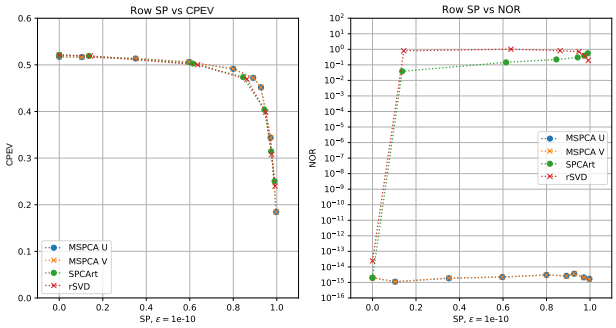

Figure 5.9: (Row SP, CPEV) and (Row SP,NOR) depending on λ with $\ell_{2,1}$ norm

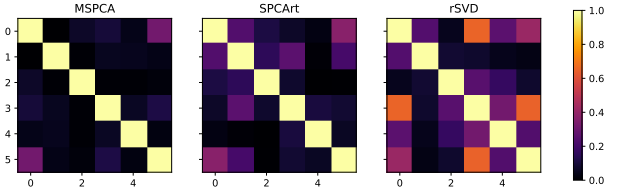

Figure 5.10: Correlations of Principal Components with $\ell_{2,1}$-norm

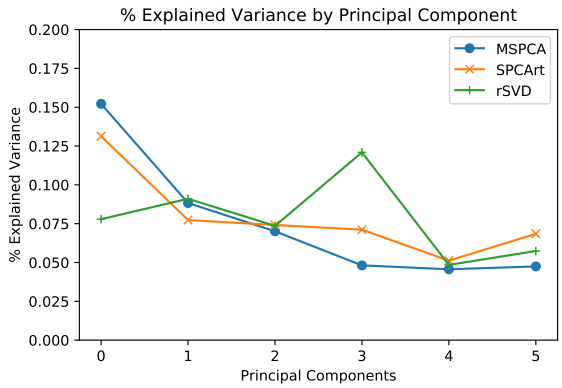

Figure 5.11: Explained variance by principal component with $\ell_{2,1}$-norm

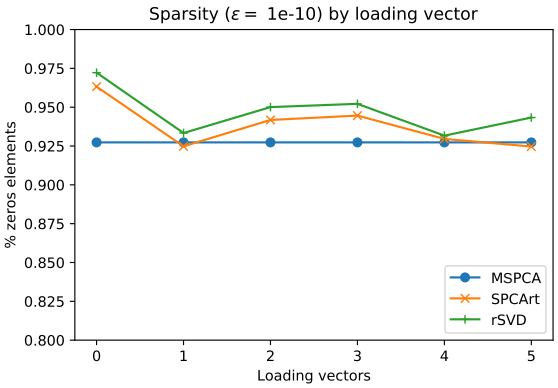

Figure 5.12: Sparsity of the loading vectors with $\ell_{2,1}$-norm

5.4 Study of the distributed version

In this part, we study the distributed version of MSPCA. In this experiment, the differents nodes (where local variables are updated) are represented by different cores on a CPU. The sparsity tuning is done so that the sparsity is roughly in the same zone as other methods.

Results with ℓ_1-norm sparsity penalization are in Figures 5.13, 5.14 and 5.15, while those for $\ell_{2,1}$-norm sparsity penalization are in Figures 5.16, 5.17 and 5.18. The distributed version loses some performance compared to the non-distributed case, which was to be expected due to the increased complexity of handling distributed data. The ordering of loading vectors explained variance is lost, however the loading vector with most variance explained is more sparse. The orthogonality of the loading vectors is also well-preserved.

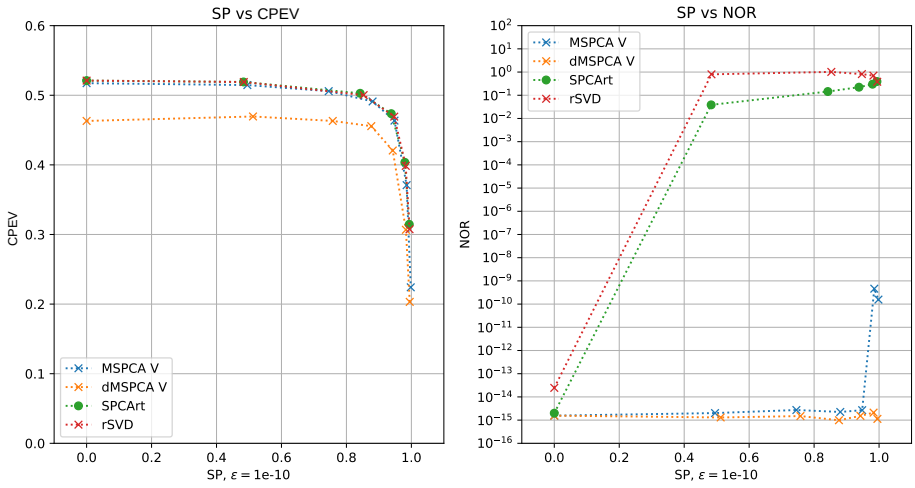

Figure 5.13: (SP, CPEV) and (SP,NOR) depending on λ with ℓ_1-norm

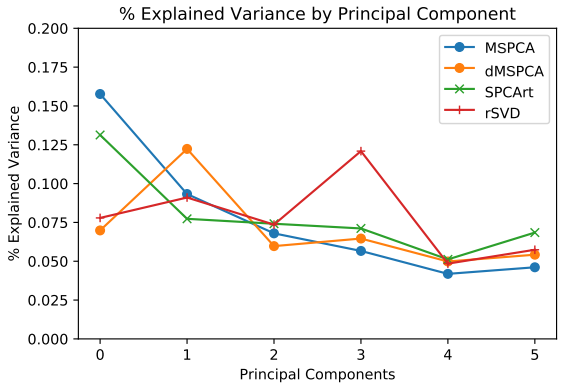

Figure 5.14: Explained variance by principal component with ℓ_1-norm

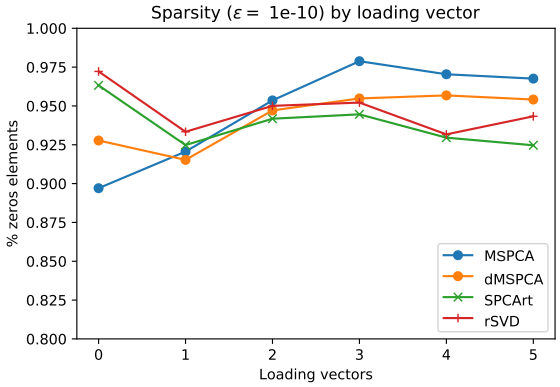

Figure 5.15: Sparsity of the loading vectors with ℓ_1-norm

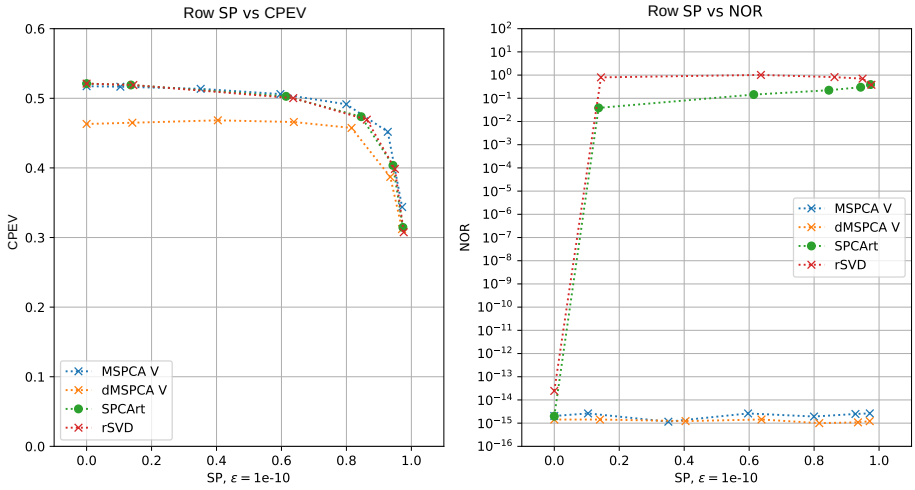

Figure 5.16: (Row SP, CPEV) and (Row SP,NOR) depending on λ with $\ell_{2,1}$-norm

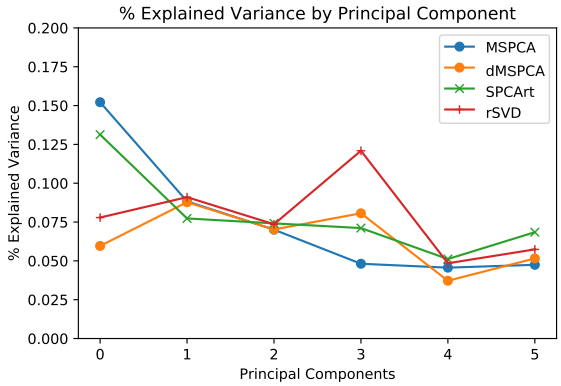

Figure 5.17: Explained variance by principal component with $\ell_{2,1}$-norm

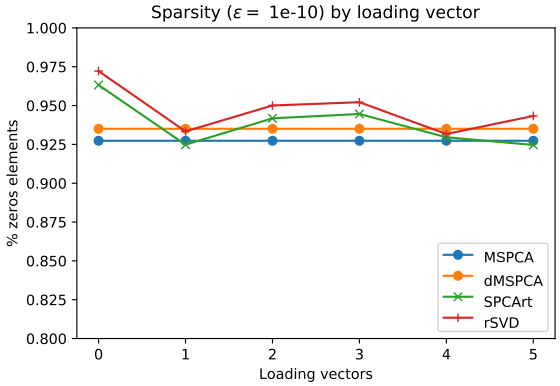

Figure 5.18: Sparsity of the loading vectors with $\ell_{2,1}$-norm

6 | Manifold optimization for MSPCA

The algorithm proposed in chapter 4 resorts to variable splitting in order to deal with sparsity promoting penalties under the orthonormality constraint. In this chapter, we consider the alternate use of a recently proposed proximal gradient method within the Riemannian optimization framework [Chen et al., 2020], which does not require any splitting relaxation.

6.1 Background

6.1.1 Semi-smooth functions

Our optimization problem takes the following form:

$$\min_{\mathbf{U}\in \mathrm{St}(p,k)} \quad \frac{1}{n} F_{q,\delta}(\mathbf{U}, \mathbf{X}) + \lambda \rho_s(\mathbf{U}) \tag{6.1}$$

Where F is smooth but non convex whereas ρ_s is convex nonsmooth, and the variable is constrained on the Stiefel manifold $\mathrm{St}(p, k)$. The concept of smoothness is related to that of Lipschitz continuity.

Definition 6.1.1. *Lipschitz continuity*
A function f is Lipschitz continuous on a set $\mathcal{X}$ with constant $L > 0$ if :

$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\| \quad \forall x_1, x_2 \in \mathcal{X} \tag{6.2}$$

All functions of class C^1 i.e. with a continuous derivative, are Lipschitz continuous (but the converse is not true).

The level of smoothness of a function is measured by the integer n in C^n, the class of regularity to which the function belongs to. Being of class C^∞ i.e. indefinitely derivable is the most smooth a function can be. Now, we are ready to introduce semi-smoothness.

Definition 6.1.2. *Semi-smoothness*
A function f is said to be semi-smooth at a point x if:

- *it is Lipschitz continuous in the neighbourhood of x*
- *it admits directional derivatives in x in all directions*
- *when $h \to 0$, we have : $\sup_{J\in\partial f(x+h)} \|f(x+h) - f(x) - Jh\| = o(\|h\|)$*

The third condition is linked to the derivation of a Newton Method for semi-smooth functions (which will be used later). Here, ∂f denotes Clarke's generalized differential [Clarke, 1990, Chapter 2].

Definition 6.1.3. *Clarke's generalized Jacobian*
Consider $F : \mathbb{R}^n \to \mathbb{R}^m$ with usual Jacobian $JF(\mathbf{x}) \in \mathbb{R}^{m\times n}$ (the matrix of partial derivatives). The generalized Jacobian of F denoted $\partial F(\mathbf{x})$ is the convex hull of matrices $\mathbf{Z} \in \mathbb{R}^{m\times n}$ obtained as the limit of a sequence of the form $JF(\mathbf{x}_i)$, where $\mathbf{x}_i \to \mathbf{x}$ and with differentiable $F(\mathbf{x}_i)$. Symbolically:

$$\partial F(\mathbf{x}) = co\{\mathbf{Z} = \lim JF(\mathbf{x}_i) : \mathbf{x}_i \to \mathbf{x}, \mathbf{x}_i \in \Omega_F\} \tag{6.3}$$

Where Ω_F is the set of points at which F is differentiable and co denotes the convex hull.

Notably, all convex functions (thus norms) are semi-smooth everywhere.

6.1.2 Proximal gradient method

Consider the problem

$$\min_x \quad f(x) + h(x), \tag{6.4}$$

with f a possibly non-convex, twice-differentiable function and h a convex function but possibly not differentiable everywhere, with a well-defined proximal operator. The proximal gradient descent [Parikh et al., 2014] (in a classical Euclidian setting) is an algorithm that performs the following iterations:

$$\begin{aligned} x_{t+1} &= \operatorname*{argmin}_y \quad h(y) + f(x_t) + \langle \nabla f(x_t), y - x_t \rangle + \frac{1}{2l}\|y - x_t\|^2 \\ &\triangleq \operatorname{Prox}_{lh}(x_t - l\nabla f(x_t)), \end{aligned} \tag{6.5}$$

meaning that x_{t+1} minimizes $h(y)$ plus a local quadratic majorizer of $f(y)$ around x_t, with $l \in \left(0, \frac{1}{L}\right]$ (L being the Lipschitz constant of the gradient ∇f). This constant has the property of upper-bouding the Hessian $\nabla^2 f$ in the sense that $\nabla^2 f$ is positive semidefinite since f is convex and $L\mathbf{I} \succeq \nabla^2 f$. This remark allows to recast the Proximal Gradient Method in the majorization-minimization framework (cf. Appendix D).

Also note that it is possible to rewrite the minimization problem in terms of a descent direction $v_k = y - x_t$ as

$$v_t = \operatorname*{argmin}_v \quad h(x_t + v) + \langle \nabla f(x_t), v \rangle + \frac{1}{2l}\|v\|^2, \tag{6.6}$$

which formalization will be helpful in the upcoming discussions.

6.1.3 First-order Riemannian optimization methods

A manifold $\mathcal{M}$ a smooth topological space that is—in short—locally homeomorphic to the Euclidean space around each point. Notably, differentiable manifolds admit a tangent space $T_x\mathcal{M}$ at each point $x \in \mathcal{M}$, which consists in the set of all tangent vectors to $\mathcal{M}$ at x. The set of tangent spaces of $\mathcal{M}$ is called the tangent bundle, denoted $T\mathcal{M}$. Endowing each tangent space $T_x\mathcal{M}$ with an inner product $\langle \cdot, \cdot \rangle_x$ that is smooth w.r.t. x (referred to as *Riemannian metric* on $\mathcal{M}$) yields a *Riemannian Manifold*. The study of Riemannian manifolds and their geometry (metrics, connections, geodesics and distances, etc.) is a field on its own. In the following, we simply recall the tools that are necessary to express a *Riemannian gradient descent* algorithm (i.e., the generalization of the standard gradient descent to a Riemannian manifold). For more details, the reader is referred to the reference [Absil et al., 2008].

Let $F : \mathcal{M} \to \mathcal{N}$ be a function and γ be a curve in $\mathcal{M}$ with $\gamma(0) = x$ and velocity $\gamma'(0) = e$. The mapping $DF(x) : T_x\mathcal{M} \to T_{F(x)}\mathcal{N}, e \to DF(x)[e] := (F \circ \gamma)'(0)$ is called the differential of F at x, which generalizes the notion of directional derivative in Euclidian space. Now, let $f : \mathcal{M} \to \mathbb{R}$ and consider the problem :

$$\min_{x \in \mathcal{M}} \quad f(x). \tag{6.7}$$

The Riemannian gradient descent applied to f then consists of the iterations:

$$x_{t+1} = R_{x_t}(-\alpha_t \operatorname{grad} f(x_t)) \quad t = 0, 1, 2..., \tag{6.8}$$

where

- R_{x_t} is a *retraction*, i.e., a mapping $R : T\mathcal{M} \to \mathcal{M}$ such that $R_x(0) = x$ and $DR_x(0)[v] = v$ (which can be seen, from the Euclidean point of view, as a generalization of the projection on the set $\mathcal{M}$)

- $\text{grad}\, f(x_t) \in T_x\mathcal{M}$ is the *Riemannian gradient* of f at x_t, which is entirely defined by the relation

$$Df(x)[v] = \langle v, \text{grad}\, f(x)\rangle_x. \tag{6.9}$$

  Notice that the notion of gradient thus depends on the chosen metric.

- $\alpha_t > 0$ is the step size at iteration t.

Thus, the Riemannian gradient algorithm (represented in Figure 6.1) follows at each step the direction of the negative gradient while staying on the manifold thanks to the retraction operator.

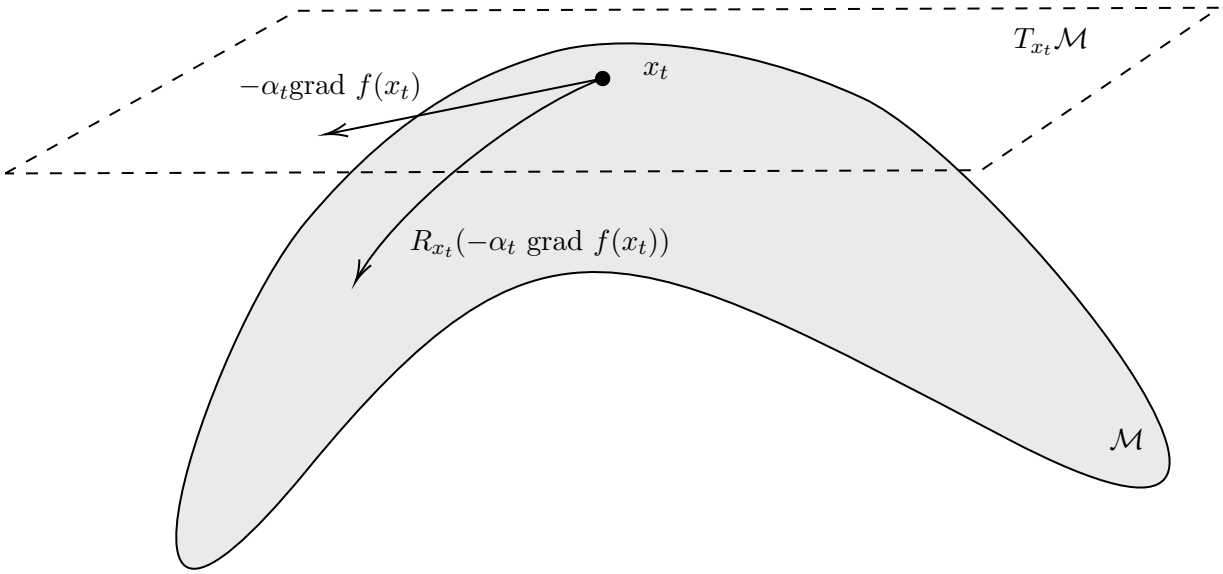

Figure 6.1: Riemannian gradient descent

In our context, the Stiefel Manifold $\text{St}(p,k)$ (cf. definition 0.0.1) is indeed a differentiable manifold with a tangent space at each point $\mathbf{U}$ defined by

$$T_{\mathbf{U}}\text{St}(p,k) = \{\mathbf{Z} \in \mathbb{R}^{p\times k} : \mathbf{U}^T\mathbf{Z} + \mathbf{Z}^T\mathbf{U} = \mathbf{0}\} \tag{6.10}$$

This manifold can be endowed with the metric $\langle \mathbf{A}, \mathbf{B}\rangle_{\mathbf{U}} = \text{Tr}\,(\mathbf{A}^T\mathbf{B}) \quad \forall \mathbf{A}, \mathbf{B} \in T_{\mathbf{U}}\text{St}(p,k)$ (which is simply the Frobenius inner product restricted to tangent vectors). In this case, we have the Riemannian gradient [Absil et al., 2008]

$$\text{grad}\, f(\mathbf{U}) = \text{Proj}_{T_{\mathbf{U}}\text{St}(p,k)}(\nabla f(\mathbf{U})) \tag{6.11}$$

where ∇f is the Euclidian gradient, and with the orthogonal projection

$$\text{Proj}_{T_{\mathbf{U}}\text{St}(p,k)}(\mathbf{X}) = (\mathbf{I}_p - \mathbf{U}\mathbf{U}^T)\mathbf{X} + \frac{1}{2}\mathbf{U}(\mathbf{U}^T\mathbf{X} - \mathbf{X}^T\mathbf{U}). \tag{6.12}$$

Moreover, retractions on $\text{St}(p,k)$ can be computed either with the Q factor of a QR decomposition, or the orthogonal factor of a polar decomposition (which can be computed through SVD). Hence, Riemannian gradient descend methods can be applied on $\text{St}(p,k)$ from these tools.

6.2 Proximal gradient method for the Stiefel manifold

Since the objective function in (6.1) is non-smooth but semi-smooth, we cannot directly use a Riemannian gradient descent algorithm. A recent solution has been proposed in [Chen et al., 2020], where proximal gradient descent were extended to manifold optimization, and where an example is specifically tailored for $\mathrm{St}(p,k)$. In this section, we first recall the key elements from [Chen et al., 2020] before applying the corresponding algorithm to the considered problem.

6.2.1 Extending the Proximal Gradient Method

The main idea is to recast the formulation from (6.6) in order to obtain a direction descent that lies in the proper space. Hence, the t^{th} direction descent $\mathbf{V}_t$ is defined as the solution of the of the problem

$$\begin{array}{ll} \min\limits_{\mathbf{V}} & \lambda\rho_s(\mathbf{U}_t+\mathbf{V}) + \left\langle \nabla \frac{1}{n} F_{q,\delta}(\mathbf{U}_t,\mathbf{X}), \mathbf{V}\right\rangle + \frac{1}{2l}\|\mathbf{V}\|_F^2 \\ \text{s.t.} & \mathbf{V} \in T_{\mathbf{U}_t}\mathrm{St}(p,k) \end{array} \tag{6.13}$$

Notice that a constraint is set to ensure that $\mathbf{V}_t$ belongs to the tangent space of the current point. Also note that, in theory, the Euclidian gradient should be replaced with its Riemannian counterpart. However, it is not necessary to compute this quantity in practice since we have the relation $\langle \mathbf{V}, \mathrm{grad}\, \frac{1}{n} F_{q,\delta}(\mathbf{U}_t,\mathbf{X})\rangle = \langle \mathbf{V}, \nabla \frac{1}{n} F_{q,\delta}(\mathbf{U}_t,\mathbf{X})\rangle$. The resolution of (6.13) can be performed using a semi-smooth Newton (SSN) method, which is detailed in subsection 6.2.2. Once $\mathbf{V}_t$ is obtained, a step size α_t can be evaluated by backtracking linesearch, and the iterate is obtained by applying a standard retraction on the Stiefel manifold, i.e., $\mathbf{U}_{t+1} = R_{\mathbf{U}_t}(\alpha_t \mathbf{V}_t)$.

For the problem (6.1), we have the Euclidean gradient:

$$\begin{aligned} &\nabla \frac{1}{n} F_{q,\delta}(\mathbf{U}_t,\mathbf{X}) = -2\frac{1}{n}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{U}_t \\ &\text{with} \quad \tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1,\dots,\tilde{\mathbf{x}}_n] \text{ and } \tilde{\mathbf{x}}_i = \mathbf{x}_i / \max\left(dist^{(2-q)/2}(\mathbf{x}_i,\mathbf{U}), \sqrt{q\delta}\right) \end{aligned} \tag{6.14}$$

which is sufficient to apply the presented Riemannian proximal gradient method. Algorithm 6 summarizes the corresponding method, hereby referred to as ManMSPCA.

Algorithm 6 ManMSPCA

1: **Entry:** $\mathbf{X} = [\mathbf{x}_1,\dots,\mathbf{x}_n] \in \mathbb{R}^{p\times n}$
2: **Initialize** $\mathbf{U} \in \mathbb{R}^{p\times k}$ by the first k loadings of the PCA of $\mathbf{X}$
3: **for** $t = 0, 1, 2, \dots$ **do**
4: **Solve** (6.13) for $\mathbf{V}_t$
5: **Compute** α_t (by backtracking linesearch)
6: **Set** $\mathbf{U}_{t+1} = R_{\mathbf{U}_t}(\alpha_t \mathbf{V}_t)$
7: **end for**
8: **Output:** $\mathbf{U} \in \mathbb{R}^{p\times k}$

6.2.2 Semi-smooth Newton method for a descent direction (6.13)

Let us define the operator

$$\begin{aligned} \mathcal{A}_t : \ & \mathbb{R}^{p\times k} \to \mathbb{R}^{k\times k} \\ & \mathbf{V} \mapsto \mathbf{V}^T\mathbf{U}_t + (\mathbf{U}_t)^T\mathbf{V} \end{aligned} \tag{6.15}$$

Then, the Lagrangian function associated to (6.13) is

$$\mathcal{L}(\mathbf{V},\mathbf{\Lambda}) = \lambda\rho_s(\mathbf{U}_k+\mathbf{V}) + \left\langle \nabla\frac{1}{n}F_{q,\delta}(\mathbf{U}_t,\mathbf{X}), \mathbf{V}\right\rangle + \frac{1}{2l}\|\mathbf{V}\|_F^2 - \langle \mathcal{A}_t(\mathbf{V}), \mathbf{\Lambda}\rangle \tag{6.16}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{k\times k}$ denotes a matrix of Lagrange multipliers. The so-called Karush-Kuhn-Tucker (KKT) conditions for (6.13) yield the system

$$0 \in \partial_{\mathbf{V}} \mathcal{L}(\mathbf{V}, \mathbf{\Lambda}) \text{ and } \mathcal{A}_t(\mathbf{V}) = \mathbf{0}. \tag{6.17}$$

Let $B(\mathbf{\Lambda}) = \mathbf{U}_t - l\left(\nabla \frac{1}{n} F_{q,\delta}(\mathbf{U}_t, \mathbf{X}) - \mathcal{A}_t^*(\mathbf{\Lambda})\right)$ where $\mathcal{A}_t^*$ is the adjoint operator of $\mathcal{A}_t$ (defined as $\mathcal{A}_t^* : \mathbb{R}^{k\times k} \to \mathbb{R}^{p\times k}, \mathbf{\Lambda} \mapsto \mathbf{U}_t\mathbf{\Lambda} + \mathbf{U}_t\mathbf{\Lambda}^T$) and denote $\tau_s(\mathbf{V}) = \rho_s(\mathbf{U}_t + \mathbf{V})$. Then by definition of the proximal operator as well as its postcomposition and precomposition properties [Parikh et al., 2014], we have :

$$\begin{aligned} \text{Prox}_{l\lambda\rho_s}(B(\mathbf{\Lambda})) - \mathbf{U}_t &= \text{Prox}_{l\lambda\tau_s}(B(\mathbf{\Lambda}) - \mathbf{U}_t) \\ &= \underset{\mathbf{V}}{\text{argmin}} \quad \lambda\rho_s(\mathbf{U}_t + \mathbf{V}) + \frac{1}{2l}\|B(\mathbf{\Lambda}) - \mathbf{U}_t - \mathbf{V}\|_F^2 \end{aligned} \tag{6.18}$$

By developing $B(\mathbf{\Lambda})$, expanding the norm as a trace and omitting the terms not depending on $\mathbf{V}$, we get:

$$\mathbf{V}(\mathbf{\Lambda}) = \text{Prox}_{l\lambda\rho_s}(B(\mathbf{\Lambda})) - \mathbf{U}_t = \underset{\mathbf{V}}{\text{argmin}} \quad \mathcal{L}(\mathbf{V}, \mathbf{\Lambda}). \tag{6.19}$$

Reporting this solution in the second KKT condition (dual feasability), we get

$$E(\mathbf{\Lambda}) = \mathcal{A}_t(\mathbf{V}(\mathbf{\Lambda})) = (\mathbf{V}(\mathbf{\Lambda}))^T\mathbf{U}_t + (\mathbf{U}_t)^T\mathbf{V}(\mathbf{\Lambda}) = \mathbf{0}. \tag{6.20}$$

The problem consists thus in solving for $\mathbf{\Lambda}$ by finding the roots of this function. This solution will be obtained using the Semi-Smooth Newton (SSN) method [Izmailov and Solodov, 2014]. An iteration of the method consists in performing

$$\mathbf{\Lambda}_{t+1} = \mathbf{\Lambda}_t - J_t^{-1}E(\mathbf{\Lambda}_t) \quad k = 0, 1, 2, \ldots, \tag{6.21}$$

where J_t denotes an element of Clarke's generalized Jacobian. To use the SSN method, we thus need to compute the generalized Jacobian of E. Then:

$$\begin{aligned} \text{vec}\,(E(\mathbf{\Lambda})) &= \text{vec}\left(\mathbf{V}(\mathbf{\Lambda})^T\mathbf{U}_t\right) + \text{vec}\left((\mathbf{U}_t)^T\mathbf{V}(\mathbf{\Lambda})\right) \\ &= \mathbf{K}_{kk}(\mathbf{I}_k \otimes (\mathbf{U}_t)^T)\,\text{vec}\,(\mathbf{V}(\mathbf{\Lambda})) + (\mathbf{I}_k \otimes (\mathbf{U}_t)^T)\,\text{vec}\,(\mathbf{V}(\mathbf{\Lambda})) \\ &= (\mathbf{K}_{kk} + \mathbf{I}_{kk})(\mathbf{I}_k \otimes (\mathbf{U}_t)^T)\,\text{vec}\,(\mathbf{V}(\mathbf{\Lambda})) \end{aligned} \tag{6.22}$$

Where $\mathbf{K}_{kk} \in \mathbb{R}^{kk\times kk}$ is a commutation matrix [Harville, 2012]. We also have the expression

$$\begin{aligned} &\text{vec}\,(\mathbf{V}(\mathbf{\Lambda})) = \text{vec}\,(\text{Prox}_{l\lambda\ \rho_s}(B(\mathbf{\Lambda})) - \mathbf{U}_t) \\ &= \text{Prox}_{l\lambda\ \rho_s}\left(\text{vec}\left(\mathbf{U}_t - l\nabla\frac{1}{n}F_{q,\delta}(\mathbf{U}_t, \mathbf{X})\right) + l\,\text{vec}\,(\mathbf{U_t\Lambda}) + l\,\text{vec}\left(\mathbf{U}_t\mathbf{\Lambda}^T\right)\right) - \text{vec}\,(\mathbf{U}_t) \\ &= \text{Prox}_{l\lambda\ \rho_s}\left(\text{vec}\left(\mathbf{U}_t - l\nabla\frac{1}{n}F_{q,\delta}(\mathbf{U}_t, \mathbf{X})\right) + 2l(\mathbf{I}_k \otimes \mathbf{U}_t)\,\text{vec}\,(\mathbf{\Lambda})\right) - \text{vec}\,(\mathbf{U}_t)\,. \end{aligned} \tag{6.23}$$

Let us denote

$$\begin{aligned} f(\mathbf{x}) &= \text{Prox}_{l\lambda\ \rho_s}\left(\text{vec}\left(\mathbf{U}_t - l\nabla\frac{1}{n}F_{q,\delta}(\mathbf{U}_t, \mathbf{X})\right) + \mathbf{x}\right) \\ \mathbf{A} &= (\mathbf{K}_{kk} + \mathbf{I}_{kk})(\mathbf{I}_k \otimes (\mathbf{U}_t)^T) \\ \mathbf{B} &= 2l(\mathbf{I}_k \otimes \mathbf{U}_t) \end{aligned} \tag{6.24}$$

We can then use the Jacobian Chain Rule [Clarke, 1990, Theorem 2.6.6] to compute the generalized Jacobian of E:

$$\frac{\partial\,\text{vec}\,(E(\mathbf{\Lambda}))}{\partial\,\text{vec}\,(\mathbf{\Lambda})} = \frac{\partial\mathbf{A}f(\mathbf{B}\,\text{vec}\,(\mathbf{\Lambda}))}{\partial\,\text{vec}\,(\mathbf{\Lambda})} = \mathbf{A}\frac{\partial f(\mathbf{x}_b)}{\partial\mathbf{x}_b}\mathbf{B} \tag{6.25}$$

Remark that :

$$\begin{aligned}\mathbf{x}_b &= \mathbf{B}\,\mathrm{vec}\,(\mathbf{\Lambda}) = 2l(\mathbf{I}_k \otimes \mathbf{U}_t)\,\mathrm{vec}\,(\mathbf{\Lambda}) \\ &= l\,\mathrm{vec}\,(\mathbf{U_t\Lambda}) + l\,\mathrm{vec}\left(\mathbf{U}_t\mathbf{\Lambda}^T\right) \\ &= l\,\mathrm{vec}\,(\mathcal{A}_t^*(\mathbf{\Lambda}))\end{aligned} \tag{6.26}$$

And recall:

$$\mathrm{vec}\,(B(\mathbf{\Lambda})) = \mathrm{vec}\left(\mathbf{U}_t - l\nabla\frac{1}{n}F_{q,\delta}(\mathbf{U}_t, \mathbf{X}) + l\mathcal{A}_t^*(\mathbf{\Lambda})\right) \tag{6.27}$$

Then:

$$f(\mathbf{x}_b) = \mathrm{Prox}_{l\lambda\ \rho_s}(\mathrm{vec}\,(B(\mathbf{\Lambda}))) \tag{6.28}$$

Putting everything back together, we get the following as a representation of $\partial\,\mathrm{vec}\,(E(\mathbf{\Lambda}))$:

$$\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda})) = 2l(\mathbf{K}_{kk} + \mathbf{I}_{kk})(\mathbf{I}_k \otimes (\mathbf{U}_t)^T)\mathcal{J}\left[\mathrm{vec}\,(B(\mathbf{\Lambda}))\right](\mathbf{I}_k \otimes \mathbf{U}_t) \tag{6.29}$$

Where $\mathcal{J}[\mathbf{y}]$ is the Generalized Jacobian of $\mathrm{Prox}_{l\lambda\ \rho_s}$ at $\mathbf{y}$.

Note that $\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda}))$ is singular. Indeed:

$$\det\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda})) = \det\left(\mathbf{K}_{kk} + \mathbf{I}_{kk}\right) \times \det\left(2l(\mathbf{I}_k \otimes (\mathbf{U}_t)^T)\mathcal{J}\left[\mathrm{vec}\,(B(\mathbf{\Lambda}))\right](\mathbf{I}_k \otimes \mathbf{U}_t)\right) \tag{6.30}$$

We can show that -1 is an eigenvalue of $\mathbf{K}_{kk}$ associated to a vector $\mathbf{v}$. Indeed, Let $\mathbf{v} = \mathrm{vec}\,(\mathbf{A})$ where $\mathbf{A}^T = -\mathbf{A}$ (skew-symmetric). Then, by properties of the vec operator and commutation matrices :

$$\mathbf{K}_{kk}\mathbf{v} = \mathbf{K}_{kk}\,\mathrm{vec}\,(\mathbf{A}) = -\mathbf{K}_{kk}^T\mathbf{K}_{kk}\,\mathrm{vec}\,(\mathbf{A}) = -\mathbf{v} \tag{6.31}$$

Then we have:

$$(\mathbf{K}_{kk} + \mathbf{I}_{kk})\mathbf{v} = \mathbf{K}_{kk}\mathbf{v} + \mathbf{I}_{kk}\mathbf{v} = -\mathbf{v} + \mathbf{v} = \mathbf{0} \tag{6.32}$$

Thus 0 is an eigenvalue of $(\mathbf{K}_{kk} + \mathbf{I}_{kk})$ associated to $\mathbf{v}$, so $\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda}))$ is non-invertible. We therefore resort to a regularized version of SSN called ASSN [Xiao et al., 2017] to allievate this issue. First, we solve for a modified Newton Method descent direction $\mathbf{d}_t$:

$$\left(\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda}_t)) + \eta\mathbf{I}_{kk}\right)\mathbf{d}_t = -\,\mathrm{vec}\,(E(\mathbf{\Lambda}_t)) \tag{6.33}$$

Where η is the regularization parameter dampening the singularity of $\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda}_t))$. Indeed, for positive semidefinite matrices $\mathbf{A}$ and $\mathbf{B}$: $\det(\mathbf{A}+\mathbf{B}) \geq \det(\mathbf{A}) + \det(\mathbf{B})$, implying:

$$\det\left(\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda}_t)) + \eta\mathbf{I}_{kk}\right) \geq \eta^{kk} \tag{6.34}$$

$\eta\mathbf{I}_{kk}$ is obviously positive semidefinite. It is also known that the proximal operator is Lipschitz continuous and monotone , then [Xiao et al., 2017, Lemma 2.4] tells us that its Generalized Jacobian contains positive semidefinite elements. Thus, $\mathcal{J}\left[\mathrm{vec}\,(B(\mathbf{\Lambda}))\right]$ has positive semidefinite elements. We can deduce that $\mathcal{G}(\mathrm{vec}\,(\mathbf{\Lambda}_t))$ is also positive semidefinite. Afterwards, a strategy is devised to decide whether to retain or not the direction $\mathbf{d}_t$; if $\|\mathrm{vec}\,(E(\mathbf{\Lambda}_t + \mathbf{d}_t))\|$ is sufficiently less than $\|\mathrm{vec}\,(E(\mathbf{\Lambda}_t))\|$, we do accept $\mathbf{d}_t$, and we set:

$$\mathrm{vec}\,(\mathbf{\Lambda}_{t+1}) = \mathrm{vec}\,(\mathbf{\Lambda}_t) + \mathbf{d}_t \tag{6.35}$$

Otherwise a *safeguard* step is taken.

6.3 Experiments

We study the method on the same set of real data as previously. Figure 6.2 shows the evolution of the performance-sparsity tradeoff. We then compare the variance explained and sparsity by loading vector as well as the correlation of principal components. For this purpose, we fix the sparsity tuning parameter to an optimal following the result in Figure 6.2. Indeed, we take a value that put us in the inflexion point of the curve. The results are in Figures 6.3, 6.4 and 6.5.

We observe that ManMSPCA is perfoming under the other methods. The orthogonality is however well-preserved and the principal components found by ManMSPCA are the least correlated of all methods. The explained variances of ManMSPCA are similar to MSPCA in pattern but lower, while the sparsity pattern is homogeneous.

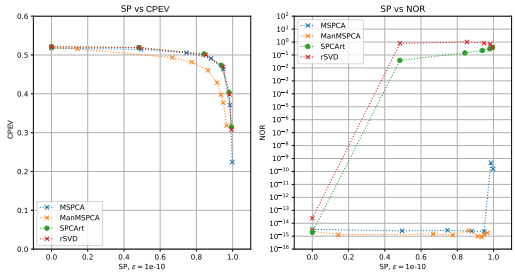

Figure 6.2: (SP, CPEV) and (SP,NOR) depending on λ with ℓ_1 norm

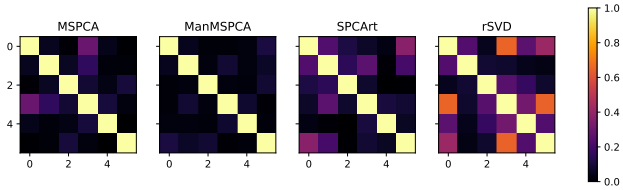

Figure 6.3: Correlations of Principal Components with ℓ_1 norm

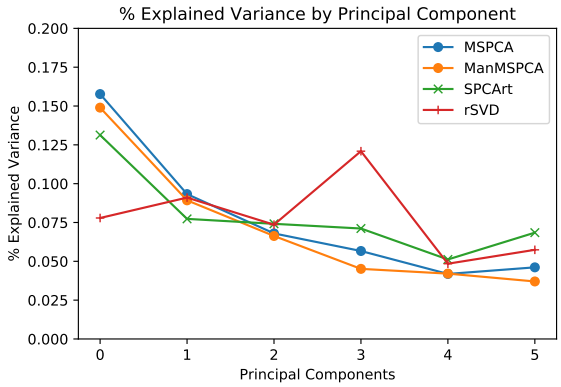

Figure 6.4: Explained variance by principal component with ℓ_1 norm

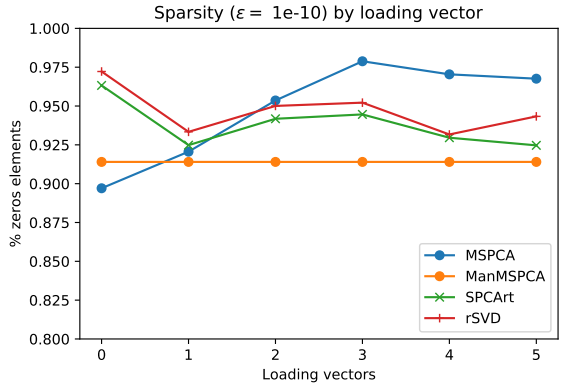

Figure 6.5: Sparsity of the loading vectors with ℓ_1 norm

7 | Conclusion

In this manuscript, we proposed a novel algorithm for robust sparse PCA, namely MSPCA. Compared to existing state-of-the-art methods, its main features are the following: it does not relax the orthogonality of the loading vectors, it achieves an interesting performance-robustness trade-off thanks to the use of a robust cost function, it can be distributed (formulation suited to large datasets across a network). In terms of explained-variance/sparsity tradeoff, MSPCA achieves state-of-the-art performance on the tested real dataset. Moreover, we also showed that an improved sparsity pattern suited to variable selection (row-sparsity in the loading vectors) can be achieved by using a $\ell_{2,1}$-norm sparsity promoting penalty.

Interestingly, the proposed approach (formulation and algorithmic resolution) can be extended to various PCA formulation by modifying the data-fitting term. In this scope, we can consider the use of robust covariance matrix estimators [Huber, 2004, Mahot, 2012, Breloy et al., 2016] in order to improve the performance-robustness trade off in non-Gaussian settings and the use of a probabilistic [Tipping and Bishop, 1999] model in order to handle missing data through EM-type algorithms.

We also explored the use of Riemannian optimization to avoid the splitting of the matrix variable in the MSPCA algorithm (i.e., strictly non-relaxed orthonormality constraint). However, the performance of the corresponding algorithm (ManMSPCA) appeared to be degraded compared to the one of the initial version. It remains to be seen if this issue can be alleviated with a more careful tuning of the algorithm parameters. Another interesting lead would be to reduce the computational load of the ASSN method involved in this algorithm, which is currently quite high for large data.

A limitation of PCA is that it is a linear method. Adding non-linearities would allow to find better representations of complex data. To this end, we can use Neural Networks or Kernel methods.

Neural networks that set the output equal to the input with the goal of minimizing the reconstruction error are called Autoencoders. They typically have at least one layer that has a smaller number of neurons acting as a bottleneck for dimensionality reduction. Simply, an Autoencoder with a single fully-connected hidden layer, a linear activation function and a squared ℓ_2-norm cost function, learns a weight matrix that spans the same subspace as PCA loading vectors [Plaut, 2018]. Autoencoders can go further by using deeper networks and non-linear activation functions [Hinton and Salakhutdinov, 2006] or manifold-valued data [Chakraborty et al., 2020].

Similarly, kernel PCA [Schölkopf et al., 1999] uses the theory of Kernel methods, first used in Support Vector Machines by Vapnik. It brings nonlinearity to PCA by embedding data into a high dimensional feature space, since all operations can be expressed only in terms of inner products of the data.

However, these methods lack interpretability, which is what prompted us to search for sparse loading vectors. Autoencoders cannot be constrained to give orthogonal loading vectors. For deep Autoencoders, the learned weights are difficult to interpret. In similar fashion, the feature space for kernel methods is very high-dimensional and no basis of it is formed.

Moving forward, we could implement sparsity in a one-hidden-layer autoencoder with non-linear activation [Makhzani and Frey, 2014].

Appendices

A | Calculations around PCA

A.1 Derivation of PCA vector by vector

Denote $\mathbf{\Sigma} = \mathbb{E}[\mathbf{x}\mathbf{x}^T]$ the covariance matrix of $\mathbf{x}$.
Remark that $\text{Var}\,(\mathbf{u}_1^T\mathbf{x}) = \mathbf{u}_1^T\mathbf{\Sigma}\mathbf{u}_1$. The problem is then :

$$\max_{\mathbf{u}_1} \mathbf{u}_1^T\mathbf{\Sigma}\mathbf{u}_1 \text{ s.t. } \|\mathbf{u}_1\| = 1$$

In an unconstrained way:

$$\max_{\mathbf{u}_1} \mathbf{u}_1^T\mathbf{\Sigma}\mathbf{u}_1 - \lambda(\mathbf{u}_1^T\mathbf{u}_1 - 1)$$

By differenciating w.r.t $\mathbf{u}_1$, we get:

$$\mathbf{\Sigma}\mathbf{u}_1 = \lambda\mathbf{u}_1 \tag{A.1}$$

Which is the definition of an eigenvector. Recall what we want to maximize:

$$\mathbf{u}_1^T\mathbf{\Sigma}\mathbf{u}_1 = \lambda\mathbf{u}_1^T\mathbf{u}_1 = \lambda$$

Thus we choose the eigenvector associated to the highest eigenvalue of $\mathbf{\Sigma}$, λ_1.

For the second PC, we need it to be uncorrelated to the first one:

$$\mathbf{u}_1^T\mathbf{\Sigma}\mathbf{u}_2 = \mathbf{u}_2^T\mathbf{\Sigma}\mathbf{u}_1 = \lambda_1\mathbf{u}_2^T\mathbf{u}_1 = 0$$

Thus, the problem can be expressed as :

$$\max_{\mathbf{u}_2} \mathbf{u}_2^T\mathbf{\Sigma}\mathbf{u}_2 - \lambda(\mathbf{u}_2^T\mathbf{u}_2 - 1) - \phi(\mathbf{u}_2^T\mathbf{u}_1)$$

Differenciating again, we get:

$$\begin{aligned} & \mathbf{\Sigma}\mathbf{u}_2 - \lambda\mathbf{u}_2 - \phi\mathbf{u}_1 = 0 \\ \iff & \mathbf{u}_1^T\mathbf{\Sigma}\mathbf{u}_2 - \lambda\mathbf{u}_1^T\mathbf{u}_2 - \phi\mathbf{u}_1^T\mathbf{u}_1 = 0 \\ \iff & \phi = 0 \end{aligned}$$

Thus :

$$\mathbf{\Sigma}\mathbf{u}_2 = \lambda\mathbf{u}_2$$

We then choose the eigenvector associated with the second highest eigenvalue of $\mathbf{\Sigma}$, λ_2.

Going on, while maintaining orthogonality of loading vectors, leads to finding the following eigenvectors of $\mathbf{\Sigma}$ for $\mathbf{u}_2, \ldots, \mathbf{u}_k$. Going to $k = p$, we retrieve as solution (Definition 0.0.2):

$$\mathbf{\Sigma} \overset{\text{EVD}}{=} \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{A.2}$$

Where eigenvalues and their corresponding eigenvector have been sorted in decreasing order.

A.2 Link between statistical and geometric approachs

We can rewrite the distances :

$$\begin{aligned}\text{dist}^2(\mathbf{x}_i, \mathbf{U}) &= \left\|\mathbf{\Pi}^{\perp}\mathbf{x}_i\right\|^2 \\ &= (\mathbf{\Pi}^{\perp}\mathbf{x}_i)^T\mathbf{\Pi}^{\perp}\mathbf{x}_i \\ &= \mathbf{x}_i^T(\mathbf{\Pi}^{\perp})^T\mathbf{\Pi}^{\perp}\mathbf{x}_i \\ &= \mathbf{x}_i^T\mathbf{\Pi}^{\perp}\mathbf{x}_i \quad \text{(by property of a projection matrix)}\end{aligned}$$

Where $\mathbf{\Pi}$ and $\mathbf{\Pi}^{\perp}$ are the orthogonal projector onto,respectively, span $(\mathbf{U})$ and span $(\mathbf{U}^{\perp})$.

We retrieve the problem for the geometrical approach :

$$\begin{aligned}&\min_{\mathbf{U}\in\text{St}(p,k)} \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T(\mathbf{I}-\mathbf{U}\mathbf{U}^T)\mathbf{x}_i \\ \iff &\min_{\mathbf{U}\in\text{St}(p,k)} \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i \\ \iff &\max_{\mathbf{U}\in\text{St}(p,k)} \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i \\ \iff &\max_{\mathbf{U}\in\text{St}(p,k)} \frac{1}{n}\sum_{i=1}^{n}\text{Tr}\left(\mathbf{x}_i^T\mathbf{U}\mathbf{U}^T\mathbf{x}_i\right) \quad \text{(since the quantity is a scalar)} \\ \iff &\max_{\mathbf{U}\in\text{St}(p,k)} \text{Tr}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{U}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{U}\right) \quad \text{(by linearity and cyclicity of the trace)} \\ \iff &\max_{\mathbf{U}\in\text{St}(p,k)} \text{Tr}\left(\mathbf{U}^T\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i\mathbf{x}_i^T)\mathbf{U}\right) \quad \text{(by left and right distributivity of matrix multiplication)}\end{aligned}$$

We recognize the problem of the statistical formulation under a Gaussian hypothesis. Thus, we obtain the same result for $\mathbf{U}$.

A.3 Link between EVD and SVD in PCA

Remark A.3.1. *The eigenvalue decomposition of a real symmetric matrix can be expressed with orthogonal eigenvectors and real eigenvalues. Then:*

$$\begin{aligned}&\mathbf{A}\mathbf{A}^T \overset{\text{EVD}}{=} \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \\ \iff &\mathbf{A}\mathbf{A}^T \overset{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T \\ &= \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T\end{aligned}$$

i.e we can retrieve eigenvalues and eigenvectors of $\mathbf{A}\mathbf{A}^T$ from the SVD of $\mathbf{A}$. An equivalent computation of eigenvectors and eigenvalues from the SVD is:

$$\mathbf{X} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{E}\mathbf{V}^T \tag{A.3}$$

Wih $\mathbf{U}$ the eigenvectors and $\mathbf{\Lambda} = \frac{1}{n}\mathbf{E}^2$ the eigenvalues.

B | LASSO

B.1 Sparsity through the ℓ_1 norm

In regression analysis , linear regression with ℓ_1 regularization is called LASSO regression [Tibshirani, 1996], while ℓ_2 regularization is called Ridge regression.

Figure B.1 shows how an undetermined linear system can be regularized as to push the solution towards a unique point that is sparse in the variables.

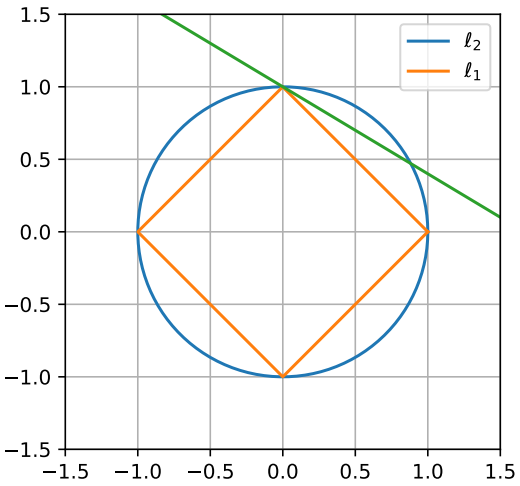

Figure B.1: Regularized solution of an underdetermined linear system

We can observe that the ℓ_1 regularized solution is unique and on the point $(0, 1)$, thus creating some sparsity. The ℓ_2 regularized solution does not share this property.

Formally, we define LASSO regression as :

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1 \tag{B.1}$$

Where $\mathbf{y} \in \mathbb{R}^m, \mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^n$.

This problem is in fact separable for $\mathbf{A} \in \mathrm{St}(n, n)$ (i.e $m = n$) since the ℓ_2 norm is unitarily invariant :

$$\|\mathbf{A}\mathbf{x}\|^2 = (\mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{x}) = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2$$

Which is true for $\mathbf{A}^T$ too since the matrix is square. Then:

$$\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 = \left\|\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x})\right\|^2 = \|\tilde{\mathbf{x}} - \mathbf{x}\|^2 = \sum_{i=1}^{n}(\tilde{x}_i - x_i)^2$$

Where $\tilde{\mathbf{x}} = \mathbf{A}^T\mathbf{y}$ is the OLS solution for an orthogonal $\mathbf{A}$.

Then, this LASSO problem becomes separable in n independant univariate problems, one per x_i :

$$\min_{\mathbf{x}} \frac{1}{2}\sum_{i=1}^{n}(\tilde{x}_i - x_i)^2 + \lambda\sum_{i=1}^{n}|x_i| = \min_{x_i}\sum_{i=1}^{n}\frac{1}{2}(\tilde{x}_i - x_i)^2 + \lambda|x_i| \tag{B.2}$$

The solution is the so-called Soft-Thresholding of $\tilde{\mathbf{x}}$:

$$S_\lambda(\tilde{\mathbf{x}}) = [S_\lambda(\tilde{x}_i)]_i = \operatorname{sgn}(\tilde{x}_i)\left(|\tilde{x}_i| - \lambda\right)_+ \tag{B.3}$$

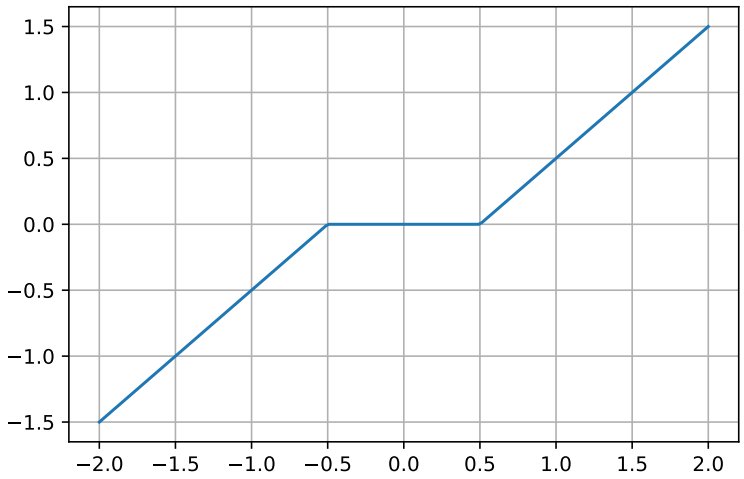

Figure B.2: Soft-Thresholding operator with $\lambda = 0.5$

For the general case $\mathbf{A} \in \mathbb{R}^{m\times n}$, several algorithms exist to compute a solution :

- Least Angle Regression : notable because it computes the entire regularization path (i.e. coefficients as λ changes) at an efficient cost
- Coordinate Descent : cycling through all x_i one by one, keeping other fixed, and solving for this univariate case.
- Proximal Gradient Descent : computing a gradient step and applying the proximal operator on it.

B.2 Extensions and structured sparsity through mixed norms

Several additions have been made to enhance the LASSO such as the Elastic-Net [Zou and Hastie, 2005] which combine ℓ_1 and ℓ_2 constraints. This gives the advantage of selecting all highly correlated variables with a similar coefficient i.e. a grouping effect.

Mixed norms (Definition 0.0.5) allow for structured sparsity. [Yuan and Lin, 2006] introduced the Group LASSO for selecting (predefined) groups of variables. In fact, thresholding is done row-wise instead of entry-wise.[Kowalski, 2009]

C | Augmented Lagrangian methods

Several sources are available on the matter. [Boyd and Vandenberghe, 2004] gives an overview of convex optimization. [Bertsekas, 1996] and [Bertsekas, 1999] are also references for optimization. Moreover, [Boyd et al., 2011] introduces key points before their contribution.

C.1 From Dual Ascent...

Let's consider the convex, equality constrained, optimization problem :

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{C.1}$$

With $\mathbf{x} \in \mathbb{R}^n, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$ and where $f : \mathbb{R}^n \to \mathbb{R}$ is convex.

The unconstrained, equivalent Lagrange formulation of (C.1) is:

$$\min_{\mathbf{x},\mathbf{y}} \quad L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$$

The Lagrange dual function is then defined as :

$$g(\mathbf{y}) = \inf_{\mathbf{x}} L(\mathbf{x}, \mathbf{y})$$

We can find an alternative formulation using the convex conjugate denoted:

$$f^*(\mathbf{y}) = \sup_{\mathbf{x}}(\mathbf{y}^T\mathbf{x} - f(\mathbf{x}))$$

Indeed :

$$\begin{aligned} g(\mathbf{y}) &= \inf_{\mathbf{x}} \left(f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \right) \\ &= -\mathbf{b}^T\mathbf{y} + \inf_{\mathbf{x}} \left(f(\mathbf{x}) + (\mathbf{A}^T\mathbf{y})\mathbf{x} \right) \\ &= -\mathbf{b}^T\mathbf{y} - f^* \left(-\mathbf{A}^T\mathbf{y} \right) \end{aligned}$$

$g(\mathbf{y})$ gives us a lower bound on problem (C.1) . A natural problem is to find the best (closest) lower bound, i.e. :

$$\begin{aligned} &\text{maximize } g(\mathbf{y}) \\ &\text{subject to } \mathbf{y} \succcurlyeq 0 \end{aligned} \tag{C.2}$$

We can refer to problem (C.1) as the *primal problem*, while (C.2) is called the *Lagrangian dual problem*. If both problems yield the same value, we call this property *strong duality*, otherwise (the lower bound is not equal) we have *weak duality*.

The method of Dual Ascent solves the dual problem via gradient ascent.

Suppose there is strong duality. We can recover a primal optimal point $\mathbf{x}^*$ from a dual optimal point $\mathbf{y}^*$:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}^*)$$

Moreover, first denote $\mathbf{x}^+ = \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{y})$. Then we have by property of the convex conjugate (assuming differentiability):

$$\nabla g(\mathbf{y}) = \mathbf{A}\nabla f^* \left(-\mathbf{A}^T\mathbf{y}\right) - \mathbf{b} = \mathbf{A}\mathbf{x}^+ - \mathbf{b}$$

Now, we can define the method of Dual Ascent. Denote k an iteration of the algorithm. Then we follow the updates:

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}^k) \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + \alpha^k(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \end{aligned} \tag{C.3}$$

Where $\alpha^k > 0$ is a step size. The first step is the x-minimization step, and the second step the dual update (via gradient ascent). Choosing α^k appropriately , $\mathbf{x}^k$ and $\mathbf{y}^k$ converge to optimal points. The assumptions made are however restrictive and often do not hold. We thus need a more robust method.

C.2 ... To the Augmented Lagrangian Method

Augmented Lagrangian bring robustness to dual ascent : for example, strict convexity of f is not required. In this form, the method is also known as the method of multipliers, originating from [Hestenes, 1969]. First, a penalty term is added to give a higher cost to infeasible points [Bertsekas, 1975]. In particular, a *quadratic penalty* term is added [Bertsekas, 1999, section 4.2]:

$$\mathbf{x}(\rho) = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

The solution to the original problem is:

$$\mathbf{x}^* = \lim_{\rho\to\infty} \mathbf{x}(\rho)$$

This can be viewed as a constrained problem:

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \\ &\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{C.4}$$

Equivalently :

$$L_p(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T\mathbf{A}\mathbf{x} - \mathbf{b} + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \tag{C.5}$$

Clearly, it is equivalent to problem (C.1) since the added term is zero for a feasible $\mathbf{x}$. Formally, this is called the *Augmented Lagrangian* form of (C.1). Its dual function is :

$$g_p(\mathbf{y}) = \inf_{\mathbf{x}} L_p(\mathbf{x}, \mathbf{y})$$

Dual ascent is then applied to those new terms. This is the aforementioned *method of multipliers.*

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} L_p(\mathbf{x}, \mathbf{y}^k) \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \end{aligned} \tag{C.6}$$

Here the step size ρ is found to allow dual feasibility of $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})$. Indeed, optimality conditions state primal and dual feasibility :

$$\mathbf{A}\mathbf{x}^* - \mathbf{b} = 0, \quad \nabla f(\mathbf{x}^*) + \mathbf{A}^T\mathbf{y}^* = 0$$

Thus, since $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} L_p(\mathbf{x}, \mathbf{y}^k)$:

$$\begin{aligned} \nabla_{\mathbf{x}} L_p(\mathbf{x}^{k+1}, \mathbf{y}^k) &= 0 \\ \nabla_{\mathbf{x}} f(\mathbf{x}^{k+1}) + \mathbf{A}^T \left(\mathbf{y}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \right) &= 0 \end{aligned}$$

Then, we can identify $\mathbf{y}^{k+1}$ as $\mathbf{y}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b})$ from the condition of dual feasability.

C.3 Parallelization with the Alternating Direction Method of Multipliers

The ADMM [Boyd et al., 2011] go further by studying decomposability of L_p in several variables. In this case, this is simply a use of Block Coordinate Descent (see below). However, they also study the opportunity of parallelizing the computation by means of a consensus formulation (see [Boyd et al., 2011, section 7]). Consider the following:

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{m} f_i(\mathbf{x}_i) + g(\mathbf{z}) \\ &\text{subject to } \mathbf{x}_i = \mathbf{z} \quad \forall i = 1, \dots, m \end{aligned} \tag{C.7}$$

Where ,$\forall i = 1, \dots, m$, we have $\mathbf{x}_i \in \mathbb{R}^n$, local variables each associated with the i-th block of data and $\mathbf{z}$ the global consensus variable. $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex objective functions each associated with the i-th local variable and g is a global regularizer.
It is called a global consensus problem since the constraint is that all local variables should agree through the global variable.

The Augmented Lagrangian is

$$\begin{aligned} L_p(\{\mathbf{x}_i\}, \mathbf{z}, \{\mathbf{y}_i\}) &= g(\mathbf{z}) + \sum_{i=1}^{m} f_i(\mathbf{x}_i) + \mathbf{y}_i^T(\mathbf{x}_i - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{x}_i - \mathbf{z}\|^2 \\ &= g(\mathbf{z}) + \sum_{i=1}^{m} l_i(\mathbf{x}_i, \mathbf{z}, \mathbf{y}_i) \end{aligned} \tag{C.8}$$

Then, the updates of the local variables can be done in parallel. Only the global variable is centralized.

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \arg\min_{\mathbf{x}_i} l_i(\mathbf{x}_i, \mathbf{z}^k, \mathbf{y}_i^k) \quad \forall i = 1, \dots, m \\ \mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} L_p(\{\mathbf{x}_i^{k+1}\}, \mathbf{z}, \{\mathbf{y}_i^k\}) \\ \mathbf{y}_i^{k+1} &= \mathbf{y}_i^k + \rho(\mathbf{x}_i^{k+1} - \mathbf{z}^{k+1}) \quad \forall i = 1, \dots, m \end{aligned} \tag{C.9}$$

D | Elements of optimization

D.1 Block Coordinate Descent

The study of this algorithmic method can trace its roots to [Ortega and Rheinboldt, 1970, section 8.2]. We briefly review this method of optimization. For more information,refer to [Bertsekas, 1999, section 2.7].

Let's consider a function to minimize over a partitioned vector :

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to } \mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_m \end{aligned} \tag{D.1}$$

Where $\forall i = 1, \dots, m$ $\mathbf{x}_i$ is a vector and $\mathcal{X}_i$ is a closed convex set. Assume that for every $\mathbf{x} \in \mathcal{X}$ and $\forall i = 1, \dots, m$

$$\begin{aligned} &\min_{\xi} f(\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \xi, \mathbf{x}_{i+1}, \dots, \mathbf{x}_m) \\ &\text{s.t. } \xi \in \mathcal{X}_i \end{aligned}$$

has a feasible solution.

The following algorithm is known as *Block Coordinate Descent*. It consists of generating the next iterate $\mathbf{x}^{k+1} = [\mathbf{x}_1^{k+1}, \dots, \mathbf{x}_m^{k+1}]$ given the current one $\mathbf{x}^k = [\mathbf{x}_1^k, \dots, \mathbf{x}_m^k]$ according to the iteration:

$$\mathbf{x}_i^{k+1} = \arg\min_{\xi \in \mathcal{X}_i} f(\mathbf{x}_1^{k+1}, \dots, \mathbf{x}_{i-1}^{k+1}, \xi, \mathbf{x}_{i+1}^k, \dots, \mathbf{x}_m^k) \quad \forall i = 1, \dots, m \tag{D.2}$$

i.e. optimization is done on $\mathbf{x}_i^k$ while fixing other 'block coordinates'. Moreover, they are here updated in *cyclic* order.

Theorem D.1.1. *Convergence of Block Coordinate Descent*

Suppose that f is continuously differentiable over $\mathcal{X}$ and that for every $\mathbf{x} \in \mathcal{X}$ and $\forall i = 1, \dots, m$, the minimum below :

$$\min_{\xi \in \mathcal{X}_i} f(\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \xi, \mathbf{x}_{i+1}, \dots, \mathbf{x}_m)$$

is unique. Then let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by block coordinate descent (D.2). Then, a limit point of $\{\mathbf{x}_k\}$ is a stationary point.

A natural idea is to use this separation over the optimization variables of a problem. Note that coordinate descent is used extensively for its simple structure : in LASSO, etc.

D.2 Majorization-Minimization

The general MM framework is briefly reviewed below (courtesy of my internship supervisor A. Breloy). For more information, refer to [Ortega and Rheinboldt, 1970, section 8.3] or more recently

to [Sun et al., 2016].
Consider the following optimization problem:

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}}\ f(\mathbf{x}),$$

where $f : \mathcal{X} \to \mathbb{R}$ is a continuous function and $\mathcal{X}$ is a closed set. Given an initial point $\mathbf{x}^0 \in \mathcal{X}$, the MM procedure minimizes f over $\mathcal{X}$ by updating $\mathbf{x}$ iteratively as

$$\mathbf{x}^{t+1} \in \underset{\mathbf{x}\in\mathcal{X}}{\text{argmin}}\ g(\mathbf{x}|\mathbf{x}^t), \tag{D.3}$$

where $g(\cdot|\mathbf{x}^t) : \mathcal{X} \to \mathbb{R}$ is a *surrogate function* of f satisfying the following property:

$$\mathbf{x}^t \in \underset{\mathbf{x}\in\mathcal{X}}{\text{argmin}}\ g(\mathbf{x}|\mathbf{x}^t) - f(\mathbf{x}).$$

In other words, $g(\cdot|\mathbf{x}^t)$ upperbounds f globally over set $\mathcal{X}$ up to a constant:

$$g(\mathbf{x}|\mathbf{x}^t) - f(\mathbf{x}) \geq c^t \triangleq \{g(\mathbf{x}^t|\mathbf{x}^t) - f(\mathbf{x}^t)\}, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{D.4}$$

The sequence $\{f(\mathbf{x}^t)\}_{t\in\mathbb{N}}$ generated by (D.3) is non-increasing since

$$f(\mathbf{x}^{t+1}) \overset{(D.4)}{\leq} g(\mathbf{x}^{t+1}|\mathbf{x}^t) - c^t \overset{(D.3)}{\leq} g(\mathbf{x}^t|\mathbf{x}^t) - c^t = f(\mathbf{x}^t).$$

The MM procedure suggests thus the possibility of minimizing f by iteratively seeking for a sequence of surrogate functions $\{g(\cdot|\mathbf{U}^t)\}_{t\in\mathbb{N}}$ that are easy to minimize over the feasible set. This procedure in recapped in Figure D.1.

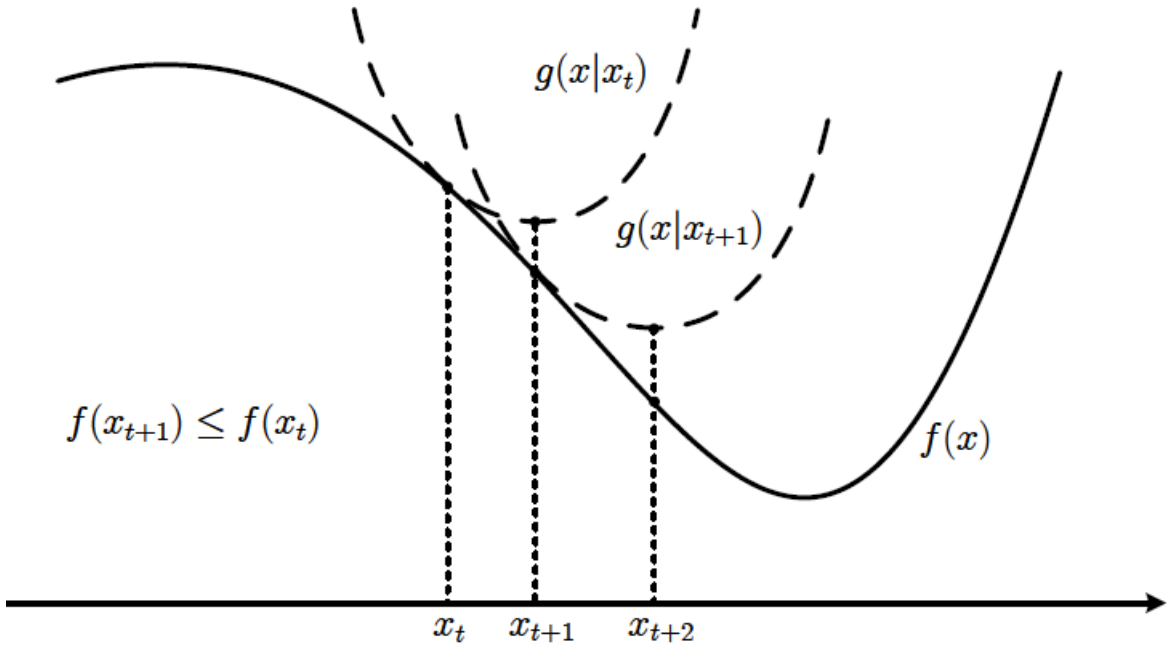

Figure D.1: MM principle: "Iteratively minimizing a smooth local tight upperbound of the objective".

Bibliography

[Abdi and Williams, 2010] Abdi, H. and Williams, L. J., "Principal component analysis," *WIREs Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. (Cited on page 3.)

[Absil et al., 2008] Absil, P.-A., Mahony, R., and Sepulchre, R., *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008. (Cited on pages 26 and 27.)

[Bao et al., 2013] Bao, C., Cai, J.-F., and Ji, H., "Fast Sparsity-Based Orthogonal Dictionary Learning for Image Restoration," In *The IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on page 3.)

[Benidis et al., 2016] Benidis, K., Sun, Y., Babu, P., and Palomar, D. P., "Orthogonal sparse pca and covariance estimation via procrustes reformulation," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6211–6226, 2016. (Cited on page 11.)

[Bertsekas, 1999] Bertsekas, D., *Nonlinear Programming*, Athena Scientific, 1999. (Cited on pages 38, 39, and 41.)

[Bertsekas, 1975] Bertsekas, D. P., "On penalty and multiplier methods for constrained minimization," In Mangasarian, O., Meyer, R., and Robinson, S., editors, *Nonlinear Programming 2*, pages 165 – 191. Academic Press, 1975. (Cited on page 39.)

[Bertsekas, 1996] Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods (Optimization and Neural Computation Series)*, Athena Scientific, 1 edition, 1996. (Cited on page 38.)

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011. (Cited on pages 38 and 40.)

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004. (Cited on page 38.)

[Breloy et al., 2016] Breloy, A., Ginolhac, G., Pascal, F., and Forster, P., "Robust Covariance Matrix Estimation in Heterogeneous Low Rank Context," *IEEE Transactions on Signal Processing*, vol. 64, pp. 1–1, 2016. (Cited on page 32.)

[Candès et al., 2011] Candès, E. J., Li, X., Ma, Y., and Wright, J., "Robust Principal Component Analysis?," *J. ACM*, vol. 58, no. 3, 2011. (Cited on page 5.)

[Chakraborty et al., 2020] Chakraborty, R., Bouza, J., Manton, J., and Vemuri, B. C., "ManifoldNet: A Deep Neural Network for Manifold-valued Data with Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. (Cited on page 32.)

[Chen et al., 2020] Chen, S., Ma, S., Man-Cho So, A., and Zhang, T., "Proximal Gradient Method for Nonsmooth Optimization over the Stiefel Manifold," *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 210–239, 2020. (Cited on pages 25 and 28.)

[Clarke, 1990] Clarke, F. H., *Optimization and Nonsmooth Analysis*, Society for Industrial and Applied Mathematics, 1990. (Cited on pages 25 and 29.)

[d'Aspremont et al., 2007] d'Aspremont, A., Ghaoui, L. E., Jordan, M. I., and Lanckriet, G. R. G., "A Direct Formulation for Sparse PCA Using Semidefinite Programming," *SIAM Review*, vol. 49, no. 3, pp. 434–448, 2007. (Cited on page 10.)

[Ding et al., 2006] Ding, C., Zhou, D., He, X., and Zha, H., "R1-PCA: Rotational Invariant L1-Norm Principal Component Analysis for Robust Subspace Factorization," In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 281–288, New York, NY, USA. Association for Computing Machinery, 2006. (Cited on pages 4 and 6.)

[Eckart and Young, 1936] Eckart, C. and Young, G. M., "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, 1936. (Cited on page 3.)

[Golub and Van Loan, 2012] Golub, G. H. and Van Loan, C. F., *Matrix computations*, volume 3, JHU press, 2012. (Cited on page 14.)

[Golub et al., 1999] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.," *Science*, vol. 286, no. 5439, pp. 531–537, 1999. (Cited on page 21.)

[Harville, 2012] Harville, D., "Matrix Algebra From a Statistician's Perspective," *Technometrics*, vol. 40, 2012. (Cited on page 29.)

[Hastie et al., 2015] Hastie, T., Tibshirani, R., and Wainwright, M., *Statistical Learning with Sparsity: The Lasso and Generalizations*, Chapman & Hall/CRC, 2015. (Cited on page 9.)

[Hestenes, 1969] Hestenes, M. R., "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1969. (Cited on page 39.)

[Higham, 1989] Higham, N. J., "MATRIX NEARNESS PROBLEMS AND APPLICATIONS," 1989. (Cited on page 14.)

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R., "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. (Cited on page 32.)

[Hotelling, 1933] Hotelling, H., "Analysis of a complex of statistical variables into principal components," *Journal of educational psychology*, vol. 24, no. 6, pp. 417, 1933. (Cited on page 1.)

[Hu et al., 2016] Hu, Z., Pan, G., Wang, Y., and Wu, Z., "Sparse Principal Component Analysis via Rotation and Truncation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 875–890, 2016. (Cited on pages 10 and 18.)

[Huber, 2004] Huber, P., *Robust Statistics*, Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series. Wiley, 2004. (Cited on pages 5 and 32.)

[Huber, 1964] Huber, P. J., "Robust Estimation of a Location Parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964. (Cited on page 5.)

[Izmailov and Solodov, 2014] Izmailov, A. and Solodov, M., *Newton-Type Methods for Optimization and Variational Problems*, 2014. (Cited on page 29.)

[Jolliffe, 2002] Jolliffe, I., *Principal Component Analysis*, Springer Series in Statistics. Springer, 2002. (Cited on pages 1 and 7.)

[Jolliffe et al., 2003] Jolliffe, I. T., Trendafilov, N. T., and Uddin, M., "A Modified Principal Component Technique Based on the LASSO," *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pp. 531–547, 2003. (Cited on page 9.)

[Journée et al., 2010] Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R., "Generalized Power Method for Sparse Principal Component Analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, 2010. (Cited on page 10.)

[Kaiser, 1958] Kaiser, H., "The varimax criterion for analytic rotation in factor analysis," *Psychometrika*, vol. 23, no. 3, pp. 187–200, 1958. (Cited on page 9.)

[Koschat and Swayne, 1991] Koschat, M. and Swayne, D., "A weighted Procrustes criterion," *Psychometrika*, vol. 56, pp. 229–239, 1991. (Cited on page 14.)

[Kovnatsky et al., 2016] Kovnatsky, A., Glashoff, K., and Bronstein, M. M., "MADMM: a generic algorithm for non-smooth optimization on manifolds," In *European Conference on Computer Vision*, pages 680–696. Springer, 2016. (Cited on pages 11 and 12.)

[Kowalski, 2009] Kowalski, M., "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303 – 324, 2009. (Cited on page 37.)

[Lai and Osher, 2014] Lai, R. and Osher, S., "A Splitting Method for Orthogonality Constrained Problems," *J. Sci. Comput.*, vol. 58, no. 2, pp. 431–449, 2014. (Cited on pages 11 and 12.)

[Lerman and Maunu, 2017] Lerman, G. and Maunu, T., "Fast, robust and non-convex subspace recovery," *Information and Inference: A Journal of the IMA*, vol. 7, no. 2, pp. 277–336, 2017. (Cited on pages 4, 5, 12, and 13.)

[Lerman and Maunu, 2018] Lerman, G. and Maunu, T., "An Overview of Robust Subspace Recovery," *CoRR*, vol. abs/1803.01013, 2018. (Cited on page 5.)

[Lerman et al., 2012] Lerman, G., McCoy, M. B., Tropp, J. A., and Zhang, T., "Robust computation of linear models, or How to find a needle in a haystack," *CoRR*, vol. abs/1202.4044, 2012. (Cited on page 7.)

[Lu and Zhang, 2009] Lu, Z. and Zhang, Y., "An Augmented Lagrangian Approach for Sparse Principal Component Analysis," *Mathematical Programming*, vol. 135, 2009. (Cited on page 11.)

[Mackey, 2009] Mackey, L. W., "Deflation Methods for Sparse PCA," In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 1017–1024. Curran Associates, Inc., 2009. (Cited on page 10.)

[Mahot, 2012] Mahot, M., *Robust covariance matrix estimation in signal processing*, Theses, École normale supérieure de Cachan - ENS Cachan, 2012. (Cited on page 32.)

[Makhzani and Frey, 2014] Makhzani, A. and Frey, B. J., "k-Sparse Autoencoders," *CoRR*, vol. abs/1312.5663, 2014. (Cited on page 32.)

[Maronna et al., 2019] Maronna, R., Martin, R., Yohai, V., and Salibián-Barrera, M., *Robust Statistics: Theory and Methods (with R)*, Wiley Series in Probability and Statistics. Wiley, 2019. (Cited on page 5.)

[Moreau, 1965] Moreau, J. J., "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299, 1965. (Cited on page 14.)

[Ortega and Rheinboldt, 1970] Ortega, J. M. and Rheinboldt, W. C., "Iterative Solution of Nonlinear Equations in Several Variables," page ii. Academic Press, 1970. (Cited on page 41.)

[Parikh et al., 2014] Parikh, N., Boyd, S., et al., "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. (Cited on pages 14, 26, and 29.)

[Pearson, 1901] Pearson, K., "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, vol. 6, no. 2, pp. 559–572, 1901. (Cited on page 2.)

[Plaut, 2018] Plaut, E., "From Principal Subspaces to Principal Components with Linear Autoencoders," 2018. (Cited on page 32.)

[Rish and Grabarnik, 2014] Rish, I. and Grabarnik, G., *Sparse Modeling: Theory, Algorithms, and Applications*, 2014. (Cited on page 9.)

[Schölkopf et al., 1999] Schölkopf, B., Smola, A. J., and Müller, K.-R., *Kernel Principal Component Analysis*, page 327–352, MIT Press, Cambridge, MA, USA, 1999. (Cited on page 32.)

[Shen and Huang, 2008] Shen, H. and Huang, J. Z., "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of Multivariate Analysis*, vol. 99, no. 6, pp. 1015 – 1034, 2008. (Cited on page 10.)

[Sun et al., 2016] Sun, Y., Babu, P., and Palomar, D., "Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning," *IEEE Trans. on Signal Process.*, vol. PP, no. 99, pp. 1–1, 2016. (Cited on pages 13 and 42.)

[Tibshirani, 1996] Tibshirani, R., "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. (Cited on page 36.)

[Tipping and Bishop, 1999] Tipping, M. E. and Bishop, C., "Probabilistic Principal Component Analysis," *Journal of the Royal Statistical Society, Series B*, vol. 21, no. 3, pp. 611–622, Available from http://www.ncrg.aston.ac.uk/Papers/index.html, 1999. (Cited on pages 6 and 32.)

[Uematsu et al., 2019] Uematsu, Y., Fan, Y., Chen, K., Lv, J., and Lin, W., "SOFAR: Large-Scale Association Network Learning," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 4924–4939, 2019. (Cited on pages 11, 12, and 15.)

[Xiao et al., 2017] Xiao, X., Li, Y., Wen, Z., and Zhang, L., "A Regularized Semi-Smooth Newton Method with Projection Steps for Composite Convex Programs," *Journal of Scientific Computing*, 2017. (Cited on page 30.)

[Yuan and Lin, 2006] Yuan, M. and Lin, Y., "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society, Series B*, vol. 68, pp. 49–67, 2006. (Cited on page 37.)

[Zou and Hastie, 2005] Zou, H. and Hastie, T., "Regularization and variable selection via the Elastic Net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005. (Cited on page 37.)

[Zou et al., 2006] Zou, H., Hastie, T., and Tibshirani, R., "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006. (Cited on page 10.)