Université de Liège
Faculté des Sciences Appliquées
Dept. of Electrical Engineering and Computer Science

Geometric algorithms for component analysis with a view to gene expression data analysis

PhD thesis of **Michel Journée**

supervised by Prof. Rodolphe Sepulchre

Spring 2009

Abstract

The research reported in this thesis addresses the problem of *component analysis*, which aims at reducing large data to lower dimensions, to reveal the essential structure of the data. This problem is encountered in almost all areas of science – from physics and biology to finance, economics and psychometrics – where large data sets need to be analyzed.

Several paradigms for component analysis are considered, e.g., *principal component analysis*, *independent component analysis* and *sparse principal component analysis*, which are naturally formulated as an optimization problem subject to constraints that endow the problem with a well-characterized *matrix manifold* structure. Component analysis is so cast in the realm of optimization on matrix manifolds. Algorithms for component analysis are subsequently derived that take advantage of the geometrical structure of the problem.

When formalizing component analysis into an optimization framework, three main classes of problems are encountered, for which methods are proposed. We first consider the problem of optimizing a smooth function on the set of n-by-p real matrices with orthonormal columns. Then, a method is proposed to maximize a convex function on a compact manifold, which generalizes to this context the well-known *power method* that computes the dominant eigenvector of a matrix. Finally, we address the issue of solving problems defined in terms of large positive semidefinite matrices in a numerically efficient manner by using *low-rank* approximations of such matrices.

The efficiency of the proposed algorithms for component analysis is evaluated on the analysis of *gene expression data* related to breast cancer, which encode the expression levels of thousands of genes gained from experiments on hundreds of cancerous cells. Such data provide a snapshot of the biological processes that occur in tumor cells and offer huge opportunities for an improved understanding of cancer. Thanks to an original framework to evaluate the biological significance of a set of components, well-known but also novel knowledge is inferred about the biological processes that underlie breast cancer.

Hence, to summarize the thesis in one sentence: *We adopt a geometric point of view to propose optimization algorithms performing component analysis, which, applied on large gene expression data, enable to reveal novel biological knowledge.*

Acknowledgments

Completing this thesis has been a very enriching and often overwhelming experience, for which I am indebted to many people. I express my sincere appreciation to all of them.

My deepest gratitude naturally goes to my advisor, Rodolphe Sepulchre. While granting all the freedom I wanted, he also provided all the guidance, encouragement and support I needed. When frustrations arose, he somehow always managed to convert them into fresh enthusiasm, motivation and optimism.

I am also very grateful to my co-advisor Pierre-Antoine Absil (Université catholique de Louvain) for many motivating discussions as well as for insightful comments and advices. I highly appreciated his hospitality during my two weeks long visit in Cambridge.

I wish to acknowledge my closest collaborators Andrew Teschendorff (University of Cambridge, UK), Francis Bach (Ecole Normale Supérieure, Paris), Peter Richtárik (Université catholique de Louvain) and Yurii Nesterov (Université catholique de Louvain). They shared a lot of their expertise and research insight with me. I specifically warmly thank Francis Bach for his friendly welcome when I came in Paris to visit him.

I owe my warm thanks to all my colleagues and friends from the research unit, past and present, for creating a pleasant and inspiring atmosphere over the last four years.

I would also like to show my genuine appreciation to the members of the Jury for their interest and time devoted to this work.

I acknowledge financial support from the Belgian National Fund for Scientific Research (FNRS) as well as from the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

Un merci tout particulier à ma famille, pour son soutien inconditionnel et permanent.

Contents

Chapter 1

Introduction

Today's society lives in an era of data overload. With the advent of information technologies, data have been digitized and their amount started to grow at an ever-spiraling rate. This revolution arises in any sector. In cancer research, an overwhelming amount of data is collected from high-throughput experiments on genetic material. These data offer unprecedented opportunities for the discovery of novel knowledge on the biological processes behind cancer. Unfolding the biological information withheld by the raw data could pave the way for improved diagnosis, treatments and drugs.

In this thesis, we analyze data that result from *microarray* experiments, which quantify, in a single assay, the extend of *transcription* of a large portion of all genes in a cell. The transcription is the synthesis of RNA molecules from the DNA, i.e., it is a process that makes static genetic information *functional*. Microarray experiments provide thus a snapshot of the biological events that occur in the analyzed cell. This technology is enabling genetic diseases like cancer to be studied in unprecedented detail, both at transcriptomic and genomic levels. A significant challenge that needs to be overcome is to unravel the complex mechanism that gives rise to the measured *expression levels* of the genes. This would drastically improve our understanding of the close relation between the quantitative transcriptome of a cell and its *phenotype*, i.e., the external traits of the cell.

Throughout the thesis, we investigate strategies to infer knowledge about these biological mechanisms from large *gene expression data*. Our analysis focuses on data resulting from microarray experiments in the context of breast cancer. The data are structured in a matrix $A \in \mathbf{R}^{m\times n}$, which stores the expression levels of n genes gained from the analysis of m cancerous cells. The number of genes considered in a single study (n) is typically around ten thousand and several hundreds of experiments (m) are usually conducted.

The tools discussed in the thesis perform an approximate factorization of the data matrix into the product of two matrices $Y \in \mathbf{R}^{m\times p}$ and $Z \in \mathbf{R}^{n\times p}$,

$$A = YZ^T + E, \tag{1.1}$$

with the *rank* $p \ll \min(m, n)$. The matrix $E \in \mathbf{R}^{m\times n}$ is an error term that has to be made as small as possible. The columns of both A and Y can be interpreted as samples of random variables, in which case A and Y are seen as random row vectors. In this probabilistic view, the factorization model (1.1) derives p linear combinations of the n original variables, called the *components* or the *factors*. The rank p is usually chosen substantially smaller than n such that the information stored in n variables is concentrated in p components. Model (1.1) performs then a *component analysis* of the data A. The methods for component analysis in this thesis are all *non-parametric* in the sense that no assumption is made on the probability distributions of the components.

Component analysis is a highly appreciated tool for the analysis of data containing a large number of interrelated variables. The components are expected to capture the *essential* structure of the data and to highlight information that is otherwise hidden in the large database. In the case of gene expression data, the components are expected to characterize distinct biological functions. The expression level in a cell is in fact determined by a whole range of biological processes, some of which act to reduce this number, while others act to increase it. It is therefore natural to model gene expression levels as the net sum of a complex superposition of cooperating and counteracting biological processes.

Many methods can be found in the literature that perform an approximate matrix factorization in the sense of (1.1). Minimizing a particular matrix norm of the approximation error E cast the factorization model (1.1) as a matrix optimization problem,

$$\min_{Y,Z} \|A - YZ^T\|.$$

Extra a priori information on the data can be incorporated either as constraints (i.e., a restriction of the search space) or as penalties (i.e., additional terms in the objective function). Mathematical optimization per se gives great flexibility in the problem formulation.

The optimization problems considered throughout the thesis feature *geometric constraints*, which are dictated by biologically motivated assumptions on Y and Z. These constraints enforce the solutions to lie on a *matrix manifold*. A *differentiable manifold* is a mathematical space that is locally Euclidean but with a global structure that may be more complex. Intuitively, a manifold can be seen as a smoothly curved space. Importantly, the geometry of a manifold is entirely determined intrinsically, without the need of an "external Euclidean world". If the elements of that space have a natural representation in the form of a matrix, we have a *matrix* manifold. This property is essential to provide practical algorithms in matrix algebra formulation (i.e., that can be run on a computer).

The optimization methods discussed in the thesis deal with these geometric constraints in a natural manner by locally treating the manifold as a Euclidean space, which evolves at each iteration. Because of this local similarity to a Euclidean space, most classical unconstrained optimization methods can be adapted to manifolds while their convergence properties are preserved. Hence, instead of traditional methods for constrained optimization on a flat space, we favor approaches that perform an unconstrained optimization on particular curved spaces.

The idea of treating problems naturally defined on manifolds in a differential-geometric framework goes back to Luenberger [Lue72] but raised significant interest first in the control systems community, essentially with the work of Brockett (e.g., [Bro72, Bro93]). The issue was to describe differential equations whose solutions evolve on a manifold. Interest in differential geometry for numerically efficient algorithms was further sparked in a book by Helmke and Moore [HM94]. In recent years, several algorithms have been proposed that rest on a conversion from differential-geometric computations into matrix computations. Notable results have been obtained on fundamental problems of linear algebra, such as eigenvalue computation or invariant subspace computation [Abs03]. The general problem of optimizing any smooth function on a manifold is addressed in the monograph [AMS08].

The main objective of this thesis is to demonstrate that differential-geometric methods are natural and effective in the context of component analysis of large data sets. Specifically, this geometric reasoning leads to numerically efficient algorithms for component analysis of low (usually linear) complexity with the number n of variables in the data. This property is essential to analyze gene expression data, for which the number n of genes is much larger than the number m of samples. Furthermore, in view of the prompt development of high throughput technologies, the number of analyzed variables is expected to grow in the future more rapidly than the number of conducted experiments.

1.1 Contributions of the thesis

This thesis is motivated by the analysis of large gene expression data and is devoted to the development of algorithms for component analysis that inherently exploit the geometric structure of the problem.

The contributions of the thesis are threefold.

First, methods for component analysis are turned into an optimization problem on a matrix manifold. We first reformulate *principal component analysis*, probably the most popular method in this context, and extend these formulations to more refined approaches, such as *independent component analysis* and *sparse principal component analysis*.

Second, the resulting optimization problems are cast in three main classes, for which existing optimization methods are reviewed and novel ones are proposed. This leads to new algorithms for component analysis, which are numerically efficient, and thus suitable for large data.

Third, the efficiency of these algorithms for component analysis is illustrated on large gene expression data related to breast cancer. Interestingly, these algorithms enable to infer new knowledge of the biology underlying cancer.

Specific contributions of the thesis are listed below.

- We review formulations and provide novel algorithms for *independent component analysis* (ICA), which is an important method for component analysis. ICA imposes the p components described by the matrix Y in (1.1) to be as *statistically independent* as possible. ICA algorithms optimize a *contrast* function that estimates the degree of statistical independence of these components. This optimization has typically to be performed on the set of matrices with orthonormal columns, which is a non Euclidean matrix space endowed with a well-characterized manifold structure. Combining the contrasts discussed in this thesis with the proposed optimization methods provide known as well as novel algorithms for ICA. Related papers are [JTAS07, JAS07, JTA$^+$08].

- We address the problem of *sparse principal component analysis* (sparse PCA) that imposes the columns of the matrix Z in (1.1) to contain many zeros (i.e., to be *sparse*) by simultaneously maximizing the variance captured by the components in the matrix Y. Sparsity is enforced for the sake of interpretability: components that are linear combinations of a small number of the original variables are easier to interpret. It is furthermore expected that a given cellular process involves only a small fraction of genes. We propose several formulations of this problem in the form of the maximization of convex functions on compact manifolds. Our formulation deals with single-unit as well as block versions of sparse PCA. The former and the latter are aimed at extracting a single component of the data or more components at once, respectively. To the best of our knowledge, block formulations of the sparse PCA problem have not been previously proposed in the literature. An original gradient-based optimization approach is derived that generalizes to this context the well-known *power method* for computing the largest eigenvalue of a matrix. This *generalized power method* provides new practical algorithms for sparse PCA. This work has been submitted for publication in the *Journal of Machine Learning Research* [JNRS08].

- Sparse PCA is essentially the problem of finding an optimal pattern of zero and nonzero entries in the matrix Z and is thus a problem of combinatorial nature. The new generalized power method provides patterns of sparsity that are only *locally* optimal. Convex relaxations have been suggested in recent years in order to near the global solution. These relaxations need to solve optimization problems defined on a set of positive semidefinite matrices of potentially large dimension and are therefore intractable for practical problems. However, because these relaxations are tight (i.e., exact) for rank-one matrices, *low-rank* solutions are expected (and observed in practice). In this context, we propose an approach that rests on low-rank positive semidefinite matrices to reduce the computational complexity of solving the convex relaxations. It turns out that the resulting optimization problem lies on a manifold endowed with a *quotient* geometry. The corresponding material has been submitted for publication in the *SIAM Journal on Optimization* [JBAS08].

- Besides these contributions of algorithmic nature, we suggest an original strategy to gain biological information from a set of components extracted from breast cancer gene expression data. This strategy is intended to compare the proposed algorithms for component analysis in terms of biological significance. But first and foremost, it enables to infer novel and valued knowledge on the biological mechanisms behind breast cancer. This biological analysis of breast cancer data has been published in *PLoS Computational Biology* [TJA$^+$07].

1.2 Outline of the thesis

This thesis is organized as follows.

In Chapter 2, we introduce three methods for component analysis, *principal component analysis*, *independent component analysis*, and *sparse principal component analysis*, and cast them as a constrained optimization problem. Emphasis is placed on the geometric nature of the constraints involved by these problems.

In Chapter 3, we provide some extensive details on the challenge that motivates this thesis, i.e., the analysis of gene expression data, and propose a framework to evaluate the biological significance of components extracted from these data.

Each of the Chapters 4, 5, 6 is devoted to a specific class of optimization problems. Existing optimization methods are reviewed and new ones are proposed, which specialize to new algorithms for component analysis. These algorithms are systematically applied on the same breast cancer data and compared in terms of biological significance through the framework developed in Chapter 3.
Specifically, in Chapter 4, we address the problem of optimizing a smooth function on the *Stiefel manifold*, which is the set of matrices with orthonormal columns. The discussed optimization methods provide algorithms for principal component analysis (PCA) and independent component analysis (ICA).
In Chapter 5, we derive and analyze the *generalized power method* to maximize convex functions on compact sets. New algorithms for sparse principal component analysis (sparse PCA) are subsequently obtained.
In Chapter 6, we propose a method to perform computations with low-rank positive semidefinite matrices and which enables to solve convex relaxations of the sparse PCA problem in an efficient manner.

The objectives and the achievements of the thesis are summarized in the concluding Chapter 7, which also raises some perspectives and future research directions.

1.3 Abbreviations and notations

The following conventions and notations are used throughout the thesis.

$\mathbf{R}^n$	the space of all n-dimensional real column vectors.
$\mathbf{R}^{n\times p}$	the space of all n-by-p real matrices.
$\mathbf{R}_*^{n\times p}$	the noncompact Stiefel manifold, i.e., the set of full-rank matrices of $\mathbf{R}^{n\times p}$.
$\mathbf{S}^n$	the space of all n-by-n real symmetric matrices.
$\mathcal{S}^{n-1}$	the unit Euclidean sphere in $\mathbf{R}^n$, i.e., the set of unit-norm vectors of $\mathbf{R}^n$.
$[\mathcal{S}^{n-1}]^p$	the product of p unit Euclidean sphere of $\mathbf{R}^n$, i.e., the set of matrices of $\mathbf{R}^{n\times p}$ with unit-norm columns.
$\mathrm{St}(p,n)$	the Stiefel manifold, i.e., the set of matrices of $\mathbf{R}^{n\times p}$ with orthonormal columns.
$\mathcal{O}(n)$	the orthogonal group, i.e., the set of orthogonal matrices of $\mathbf{R}^{n\times n}$.
$\mathcal{SO}(n)$	the special orthogonal group, i.e., the set of orthogonal matrices of $\mathbf{R}^{n\times n}$ with positive determinant.
$\mathcal{SP}$	the spectahedron, i.e., the set of positive semidefinite matrices of $\mathbf{S}^n$ with unit trace.
$\mathcal{E}$	the elliptope, i.e., the set of positive semidefinite matrices of $\mathbf{S}^n$ with unit diagonal elements.
$\mathrm{Conv}(\mathcal{Q})$	convex hull of the set $\mathcal{Q}$, i.e., the smallest convex set that contains Q.
$\{0,1\}^{n\times p}$	the set of all binary matrices of dimension n-by-p.
e_i	ith canonical basis vector of $\mathbf{R}^n$.
1_n	constant vector of all ones of dimension n.
I_n	identity matrix of dimension n.
$\mathrm{sign}(t)$	sign of the scalar $t \in \mathbf{R}$.
t_+	the "positive part" function $t_+ = \max\{0,t\}$ for $t \in \mathbf{R}$.
$\langle \eta, \zeta \rangle$	metric, i.e., the inner product of η and ζ.
$\mathbb{E}[x]$	expectation of the random variable x.
$\mathrm{Var}[x]$	variance of the random variable x.

Given a vector $x \in \mathbf{R}^n$ and a matrix $X \in \mathbf{R}^{n\times p}$, we define the notations,

x_i (for a vector x)	ith coordinate of the vector x.
x_i (for a matrix X)	ith column of the matrix X.
x_{ij}	element at position (i,j) of the matrix X .
$\\|x\\|_1$	ℓ_1 norm of x, i.e., $\\|x\\|_1 = \sum_i \lvert x_i \rvert$.
$\\|x\\|_0$	cardinality or ℓ_0 "norm" of x, i.e., the number of nonzero coefficients of x.
$\mathrm{Tr}(X)$	trace of the square matrix $X \in \mathbf{R}^{n\times n}$, i.e., the sum of its diagonal elements, $\mathrm{Tr}(X) = \sum_{i=1}^n x_{ii}$.
$\mathrm{Tr}[X]_+$	the sum of the positive eigenvalues of the matrix $X \in \mathbf{S}^n$.

$\|X\|_F$	Frobenius norm of X, i.e., the square root of the sum of the absolute squares of its elements, $\|X\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{p}\|x_{ij}\|^2} = \sqrt{\mathrm{Tr}(XX^T)}$.
$\|X\|_0$	the number of nonzero entries in the matrix X.
$X \succeq 0$	semidefinite positivity, i.e., the eigenvalues of the (symmetric) matrix X are all nonnegative.
$\mathrm{diag}(X)$	vector that equals the diagonal of X.
$\mathrm{Diag}(X)$	diagonal matrix with the same diagonal elements as X.
$\mathrm{Off}(X)$	matrix with entries identical to those of X except on the diagonal, which contains only zero-valued elements, i.e., $\mathrm{Off}(X) = X - \mathrm{Diag}(X)$.
$\mathrm{qf}(X)$	Q factor of the QR decomposition of X as $X = QR$, where $Q \in \mathbf{R}^{n\times p}$ has orthonormal columns and $R \in \mathbf{R}^{p\times p}$ is an upper triangular matrix.
$\mathrm{uf}(X)$	U factor of the polar decomposition of X as $X = US$, where $U \in \mathbf{R}^{n\times p}$ has orthonormal columns and $S \in \mathbf{R}^{p\times p}$ is a symmetric positive semidefinite matrix.

Given a function $f : \mathbf{R}^{n\times p} \to \mathbf{R} : X \mapsto f(X)$, we use the following notations:

$\mathrm{D}_X f(X_0)[\eta]$	directional derivative of f at X_0 in a direction η with respect to the variable X, i.e., $\mathrm{D}_X f(X_0)[\eta] = \lim\limits_{t\to 0} \frac{f(X_0+t\eta)-f(X_0)}{t}$.
$\nabla_X f(X_0)$	Euclidean gradient of f at X_0 with respect to the variable X, i.e., $[\nabla_X f(X_0)]_{i,j} = \left.\frac{\partial f}{\partial X_{i,j}}\right\|_{X_0}$.
$\mathrm{grad}_X f(X_0)$	differential-geometric gradient of f at X_0 with respect to the variable X, i.e., generalization of the Euclidean gradient $\nabla_X f(X_0)$ to a manifold.

The subscript X is useful for functions of several variables. It is omitted if no confusion is possible (e.g. $\mathrm{grad} f(X_0)$). Depending on the context, the symbol ∇ can also denote a Riemannian connection. It is then used with Greek letters, e.g., $\nabla_\eta \zeta$.

The following acronyms are also used in the thesis.

CR	cancer-related (pathway)
EMT	epithelial-mesenchymal transition (pathway)
ER	estrogen receptor status
EVD	eigenvalue decomposition
ICA	independent component analysis
PCA	principal component analysis
PEI	pathway enrichment index
PSD	positive semidefinite
SVD	singular value decomposition.

Chapter 2

A geometric framework to component analysis

The purpose of *component analysis* is to make sense of high-dimensional data by reducing it to a few number of *components* expected to extract the essential characteristics of the data. This analysis tool allows us to glimpse into the hidden and simplified structure that underlies the data. Component analysis has applications in virtually all areas of science, both exact and human, where large data sets are encountered, e.g., physics, meteorology, image processing, genetics, finance, psychometrics.

In this chapter we address the problem of component analysis from a geometrical perspective. We first turn the problem into an optimization problem (Section 2.1). We then highlight the rich geometrical structure that underlies these formulations (Section 2.2). Specifically, component analysis is cast as an optimization on a *matrix manifold*. This class of problems has been intensively studied in recent years and a whole bunch of methods are available.

2.1 Component analysis as constrained optimization

Three different approaches for component analysis are investigated in the thesis, which are reviewed in this section: *principal component analysis*, *independent component analysis* and *sparse principal component analysis*. Emphasis is placed on the formulation of these methods as an optimization problem.

These methods analyze a data matrix $A \in \mathbf{R}^{m\times n}$ that encodes m samples of n variables. Without loss of generality, the data A is assumed to be *centered*, i.e., the mean of the columns is set to zero. For such data, the Gram matrix A^TA equals the sample covariance matrix between the n variables, up to multiplication by a positive scalar factor. For the sake of clarity, this factor is omitted in the sequel and we consider the covariance matrix to "equal" the Gram matrix. Finally, given a matrix X, the notation x_i refers to the ith column of X, and x_{ij} denotes the element of X at position (i, j).

2.1.1 Principal component analysis

Principal component analysis (PCA) is probably the most widespread method for data analysis. It was originally discussed by Pearson in 1901 as a method to compute the line (or the plane) of "closest fit" to a cloud of data points in a high-dimensional space [Pea01]. Interest in PCA was raised by Hotelling in the 1930s, who adopted a statistical perspective and considered the data matrix $A \in \mathbf{R}^{m\times n}$ to result from m samples of a random vector of dimension n. Hotelling proposed a method to identify "a fundamental set of independent variables" that underlie the observed ones, which he called the *principal components* [Hot33].

In Hotelling's conception, PCA aims at changing the basis in which the data is represented to obtain interesting statistical properties. The first component of the random vector describing the data should capture maximum variance and the succeeding components should account for as much as possible of the variation in the data, while being uncorrelated with the first components. In other words, the data viewed from the new basis should have a covariance matrix that is diagonal (i.e., the principal components are uncorrelated) and with decreasing diagonal elements (i.e., the first principal components capture maximum variance). In this context, changing the basis of a vector means to multiply that vector with an *orthogonal* matrix,[1] whose columns are the coordinates of the new basis vectors. Hence, the matrix $\bar{Y} \in \mathbf{R}^{m\times n}$ that contains m samples of the n principal components is obtained by multiplying the data matrix A with an orthogonal matrix $\bar{Z} \in \mathbf{R}^{n\times n}$ (i.e., $\bar{Z}^T\bar{Z} = I_n$),

$$\bar{Y} = A\bar{Z}, \quad \text{or} \quad A = \bar{Y}\bar{Z}^T, \tag{2.1}$$

because the inverse of an orthogonal matrix is simply the transpose of that matrix.

For the purpose of component analysis, one is rarely interested in *all* the n components, but rather in the first few ones, which account for maximum variation in the data. Let us therefore truncate the component matrix $\bar{Y} \in \mathbf{R}^{m\times n}$ and store its p first columns in a matrix $Y \in \mathbf{R}^{m\times p}$, where $p < n$ is the desired number of components. Similarly, the orthogonal matrix $\bar{Z} \in \mathbf{R}^{n\times n}$ is truncated into a matrix $Z \in \mathbf{R}^{n\times p}$. PCA consists thus to view the n-dimensional data "as well as possible" from an orthonormal basis of dimension p. Equation (2.1) then becomes

$$A = YZ^T + E, \tag{2.2}$$

where the error term $E \in \mathbf{R}^{m\times n}$ compensates the approximation made by reducing the data to p components. For the components to be uncorrelated, the Gram matrix Y^TY should be diagonal, i.e., the columns of Y should be mutually orthogonal.

The forthcoming theorems point out the close relationship between the matrix factorization model (2.2) and the *singular value decomposition* (SVD) of the matrix A.

Theorem 2.1.1 *Given a real matrix $A \in \mathbf{R}^{m\times n}$, there exists a factorization of the form*

$$A = U\Sigma V^T,$$

[1] An orthogonal matrix is a square matrix with *orthonormal* columns, i.e., columns of unit-norm that are mutually orthogonal.

where $U \in \mathbf{R}^{m\times m}$ *and* $V \in \mathbf{R}^{n\times n}$ *are orthogonal matrices, i.e.,* $U^T U = I_m$ *and* $V^T V = I_n$*, and* $\Sigma \in \mathbf{R}^{m\times n}$ *is a nonnegative diagonal matrix. Such a factorization is a* singular value decomposition *(SVD) of A.*

Proof. See, e.g., Golub and Van Loan [GVL89], Theorem 2.5.1. □

The *singular values* σ_i, i.e., the diagonal elements of Σ, are usually ordered in a decreasing manner,

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0,$$

where $r = \min(m, n)$. This makes the SVD of A unique, provided that all nonzero singular values are distinct. This decomposition is a powerful tool to characterize *low-rank* approximations of a matrix. Consider the SVD of the matrix A as an expansion in *rank-one* matrices,[2]

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

where the vectors u_i and v_i are the ith column of matrices U and V, respectively. The number of rank-one matrices in this expansion, i.e., the number of nonzero singular values, defines the *rank* of the matrix A, denoted rank(A). A *rank-p singular value decomposition* of A with $p \leq \text{rank}(A)$ is so naturally defined by the truncated sum

$$A_p \stackrel{\text{def}}{=} \sum_{i=1}^{p} \sigma_i u_i v_i^T$$

or, in matrix terms,

$$A_p = U_p \Sigma_p V_p^T,$$

where U_p (resp. V_p) is formed by the p first columns of U (resp. V) and Σ_p is the p-by-p upper-left diagonal submatrix of Σ.

The following theorem provides an interesting characterization of the first singular value in the form of an optimization problem.

Theorem 2.1.2 *Given a real matrix* $A \in \mathbf{R}^{m\times n}$*, the first singular value verifies*

$$\sigma_1 = \max_{x \in \mathbf{R}^n_*} \frac{\|Ax\|_2}{\|x\|_2},$$

where $\mathbf{R}^n_*$ *is the vector space* $\mathbf{R}^n$ *with the origin removed.*

Proof. See, e.g., Golub and Van Loan [GVL89], Theorem 8.3.1 with $k = 1$. □

This property can be rewritten as the maximization of the *Rayleigh quotient* of the matrix $A^T A$,

$$\sigma_1^2 = \max_{x \in \mathbf{R}^n_*} \frac{x^T A^T A x}{x^T x}. \tag{2.3}$$

Since the Rayleigh quotient is invariant by multiplication of the vector x by a scalar, the optimization can be restrained to the unit-norm vectors without loss of generality,

$$\sigma_1^2 = \max_{\substack{x \in \mathbf{R}^n \\ x^T x = 1}} x^T A^T A x. \tag{2.4}$$

[2]A rank-one matrix is the matrix product of a column vector with a row vector.

The following theorem indicates that the Rayleigh quotient also characterizes the *right singular vector* v_1.

Theorem 2.1.3 *The maximizer of the optimization problem (2.4) is the dominant right singular vector* v_1*. It is unique if the two largest singular values are distinct, i.e.,* $\sigma_1 > \sigma_2$*.*

Proof. Write the first- and second-order optimality conditions of (2.4). □

A similar characterization can be derived for the other singular values and singular vectors.

Corollary 2.1.4 *Given a real matrix* $A \in \mathbf{R}^{m\times n}$*, the ith singular value verifies*

$$\sigma_i = \max_{x\in\mathbf{R}^n_*} \frac{\|(A - A_{i-1})x\|_2}{\|x\|_2}.$$

Proof. The ith singular value of $A = \sum_{j=1}^r \sigma_j u_j v_j^T$ is the first singular value of the truncated sum $\sum_{j=i}^r \sigma_j u_j v_j^T = A - A_{i-1}$. □

These properties allow us to relate the SVD of a data matrix A to its principal components.

Theorem 2.1.5 *Let* $U_p\Sigma_p V_p^T$ *be the rank-p SVD of the data matrix* $A \in \mathbf{R}^{m\times n}$ *encoding m samples of n variables. A number p of principal components is obtained by posing*

$$Z = V_p \quad \textit{and thus,} \quad Y \stackrel{\textit{def}}{=} AZ = U_p\Sigma_p,$$

in the model (2.2).

Proof. First, the covariance matrix of the components is given by $Y^TY = \Sigma_p^2$, which is diagonal and has decreasing diagonal elements. Then, the components successively capture maximum variance in the data. In fact, the first column z_1 of Z is chosen such that the first component $y_1 = Az_1$ has maximum variance, i.e.,

$$\mathrm{Var}[y_1] = \sigma_1^2 = \underset{\substack{z\in\mathbf{R}^n\\ z^Tz=1}}{\mathrm{argmax}}\ z^TA^TAz = \underset{\substack{z\in\mathbf{R}^n\\ z^Tz=1}}{\mathrm{argmax}}\ \mathrm{Var}[Az],$$

by virtue of Theorem 2.1.3. Similarly, the other components $y_i = Az_i$ capture maximum variance from the residual data matrix

$$A - A_{i-1} = \sum_{j=i}^{r} \sigma_j u_j v_j^T.$$

The components described by the matrix Y satisfy thus the properties to be *principal* components. □

As an interesting "side product" of PCA, the Frobenius norm of the error $E \stackrel{\text{def}}{=} A - YZ^T$ in the PCA model (2.2) is minimal. In fact, the Frobenius norm of a matrix, defined as the square root of the sum of the squares of its elements, is characterized by the singular values as follows

$$\|A\|_F^2 \stackrel{\text{def}}{=} \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2 = \mathrm{Tr}(A^TA) = \mathrm{Tr}(\Sigma^2) = \sum_{i=1}^{\mathrm{rank}(A)} \sigma_i^2.$$

The rank-p SVD of a matrix A is therefore the best rank-p approximation of A in term of minimization of the Frobenius norm of the error.

Theorem 2.1.6 *Any solution of*

$$\min_{\substack{X \in \mathbf{R}(m,n) \\ \operatorname{rank}(X) \leq p}} \|A - X\|_F,$$

is provided by a rank-p singular value decomposition of A.

Proof. See, e.g., Golub and Van Loan [GVL89], Theorem 2.5.2. □

2.1.2 Independent component analysis

PCA aims at finding a new orthogonal basis, in which the data is represented at best. One can go one step further and search an *arbitrary* basis, i.e., not necessarily an orthogonal one, in which the structure of the data is expected to be even more apparent. Relaxing the orthogonality condition on the matrix Z in the PCA setting (2.2) releases a significant degree of freedom, which can be used to enforce further properties on the components. Consider the couple

$$(Y, Z) \stackrel{\text{def}}{=} (UQ\bar{\Sigma}, V\Sigma Q\bar{\Sigma}^{-1}),$$

where $U\Sigma V^T$ is the rank-p SVD of A (i.e., $U \in \mathbf{R}^{m\times p}$ and $V \in \mathbf{R}^{n\times p}$ have orthonormal columns, $\Sigma \in \mathbf{R}^{p\times p}$ is positive and diagonal), the matrix $Q \in \mathbf{R}^{p\times p}$ is orthogonal (i.e., $Q^TQ = I_p$) and $\bar{\Sigma} \in \mathbf{R}^{p\times p}$ is a positive diagonal matrix that normalizes the columns of Z to unit norm,

$$\bar{\Sigma} \stackrel{\text{def}}{=} \sqrt{\operatorname{Diag}(Q^T\Sigma^2 Q)}.$$

The matrix Z defines a non-orthogonal basis of p unit-norm vectors in $\mathbf{R}^n$. The particular case where Q is set to the identity matrix recovers PCA.

The so-obtained components present several interesting properties. First, the product YZ^T is invariant with respect to the rotation matrix Q and so, by virtue of Theorem 2.1.6, the Frobenius norm of the error $E \stackrel{\text{def}}{=} A - YZ^T$ is minimal. Furthermore, because the matrix Y has orthogonal columns, the corresponding components are uncorrelated. They, however, do not *individually and sequentially* capture maximum variance in the data. For instance, the variance explained by the first component can be smaller than σ_1^2. They nonetheless *mutually* explain the same variance as the principal components.[3]

The released degree of freedom Q can be used to provide a better representation of the data A by enforcing new properties on the components described by the matrix Y. Typically, the components are assumed to be *statistically independent.* Statistical independence is a much stronger property than uncorrelatedness. *Independent component analysis* (ICA) aims

[3]The variance explained by the p uncorrelated components described by the matrix Y is the sum of the variance individually explained by each of them, i.e.,

$$\operatorname{Var}[Y] \stackrel{\text{def}}{=} \sum_{i=1}^{p} \operatorname{Var}[y_i] = \operatorname{Tr}(Y^TY) = \operatorname{Tr}(\bar{\Sigma}^2) = \operatorname{Tr}(Q^T\Sigma^2 Q) = \operatorname{Tr}(\Sigma^2).$$

The variance mutually explained by the p components is hence unaffected by the rotation Q and equals the variance explained by the p principal components.

at finding an orthogonal matrix $Q \in \mathbf{R}^{p\times p}$ to maximize a *contrast function* that estimates the statistical independence of the components $Y = UQ\bar{\Sigma}$, i.e., ICA solves the problem

$$\max_{\substack{Q\in\mathbf{R}^{p\times p}\\ Q^TQ=I_p}} f(Q), \tag{2.5}$$

for a contrast function f. Many contrasts are proposed in the literature. Some of them are reviewed in Chapter 4.

2.1.3 Sparse principal component analysis

Definite physical meanings are often associated to the n axes of the space in which the m data points described by the matrix $A \in \mathbf{R}^{m\times n}$ are originally represented. For instance, when dealing with gene expression data, each axis stands for a specific gene. Usual methods for component analysis, such as PCA and ICA, represent the data in a new basis to facilitate its analysis. The new basis vectors are linear combinations of, usually, *all* the original ones and the simple physical interpretation of the axes is therefore lost. Hence, it seems natural to seek a trade-off between the conflicting goals of representing the data at best and having a readily interpretable basis. This trade-off is typically obtained with basis vectors that are linear combinations of only a *small* number of the original variables. Such basis vectors are naturally easier to interpret. In the mathematical model (2.2), this means that the matrix Z, whose columns define the coordinates of the new basis vectors, should contain many zeros, i.e., the matrix Z should be *sparse*. The associated *sparse components* involve thus as few of the n original variables as possible.

The objective of *sparse principal component analysis* (sparse PCA) is to find a reasonable trade-off between principal components (which explains as much variability in the data as possible) and sparse components (which are readily interpretable). Sparse PCA inevitably sacrifices some of the variance explained by the principal components for the sake of interpretability.

Mathematical formulations of sparse PCA are generally derived from PCA as an optimization problem (e.g., problem (2.4)) by adding a penalty term, which enforces sparsity in the matrix Z. For instance, the vector z_1 associated to the dominant sparse principal component is computed as the solution of the optimization problem

$$z_1 = \arg\max_{\substack{z\in\mathbf{R}^n\\ z^Tz=1}} z^TA^TAz - \gamma\|z\|_0, \quad \text{with } \gamma \geq 0, \tag{2.6}$$

that corresponds to (2.4) in the case $\gamma = 0$. The additional term penalizes the number of nonzero components in z (i.e., the ℓ_0-norm or cardinality of z). Another common way to enforce sparsity is to penalize the ℓ_1-norm of the unit-norm vector z, i.e.,

$$z_1 = \arg\max_{\substack{z\in\mathbf{R}^n\\ z^Tz=1}} z^TA^TAz - \gamma\|z\|_1, \quad \text{with } \gamma \geq 0, \tag{2.7}$$

where $\|z\|_1 = \sum_{i=1}^n |z_i|$. Unit-norm vectors with minimum ℓ_1-norm are in fact the canonical basis vectors e_i, i.e., unit-norm vectors of maximum sparsity.

Interestingly, these problems can be reformulated as the maximization of a *convex* function on the set of unit-norm vectors. A function f is convex if any straight line segment joining two points on the graph of f always lies above this graph, i.e.,

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2),$$

for any *feasible* points x_1 and x_2 (i.e., points that satisfy the constraints) and $0 \leq \theta \leq 1$. This valued property allows for the design of fast optimization methods, as discussed in Chapter 5.

2.1.4 Convex relaxations

Sparse PCA essentially consists in finding the optimal pattern of zero and nonzero elements in a vector z, which is a problem of combinatorial complexity. Computing *the* optimal solution of (2.6) is therefore intractable for large dimension n.

When dealing with hard combinatorial optimization problems, one usually tries to *relax* the constraints to obtain *convex* optimization problems.[4] Convexity is a very desirable property, because any locally optimal solution is automatically a global one.[5] Solutions of the convex relaxation are expected to provide "good" approximations of the solution of the original combinatorial problem.

Several convex relaxations of sparse PCA have been proposed in recent years. For illustration, the sparse PCA problem (2.6) is turned into a convex problem in two steps. First, the sphere is made convex by lifting the unit-norm vector z into a matrix $Z = zz^T$ that is rank-one. This rank-one constraint has however to be dropped for the sake of convexity. The relaxation consists then to admit any element (even with a rank larger than one) of the *spectahedron*

$$\mathcal{SP} = \{Z \in \mathbf{R}^{n \times n} \mid Z^T = Z,\ Z \succeq 0,\ \mathrm{Tr}(Z) = 1\},$$

i.e., the convex set of symmetric positive semidefinite matrices with unit trace. A relaxation of the optimization problem (2.6) is thus provided by

$$\max_{Z \in \mathcal{SP}} \mathrm{Tr}(A^T A Z) - \gamma \|Z\|_0, \tag{2.8}$$

which is tight (i.e., exact) for rank-one matrices: given any rank-one solution $Z = zz^T$ of (2.8), the unit-norm vector z yields a solution to (2.6). To make problem (2.8) convex, the cardinality penalty is replaced by a convex ℓ_1-penalty term,

$$\max_{Z \in \mathcal{SP}} \mathrm{Tr}(A^T A Z) - \gamma \sum_{i,j} |z_{ij}|. \tag{2.9}$$

Even if convexity significantly reduces the complexity of the original combinatorial problem, the convex optimization problem (2.9) requires to search a space of dimension $\mathcal{O}(n^2)$, which

[4] An optimization problem is *convex* if a convex objective function is minimized on a convex set of points. A set of points is convex if it contains all the straight line segments between any two of its points. Note that maximizing a *concave* function, i.e., the negative of a convex function, on a convex set also provides a convex problem. We refer to Boyd and Vanderberghe [BV04] for more details on convex problems.

[5] A local solution optimizes the objective function among the points that are near it, whereas the global solution is the optimal one among all possible points.

is practically intractable for large n. Because this solution is eventually rounded to a rank-one matrix to reconstruct a unit-norm vector solving (2.6), an approximate low-rank solution of (2.9) is often sufficient. Furthermore, because the relaxation (2.8) is tight for rank-one matrices, one can reasonably expect the solution of (2.9) to be low-rank. Hence, to reduce computational complexity, the positive semidefinite matrix is factored as $Z = WW^T$ with $W \in \mathbf{R}^{n\times l}$ and the rank l that is much smaller than n. The problem to solve becomes

$$\max_{\substack{W\in\mathbf{R}^{n\times l}\\ \mathrm{Tr}(W^TW)=1}} \quad \mathrm{Tr}(W^TA^TAW) - \gamma \sum_{i,j} |(WW^T)_{ij}|, \tag{2.10}$$

which searches a space of dimension nl. Although the convexity is lost, the number of local solutions of (2.10) decreases as l increases. The rank-one case (i.e., $l = 1$) is very close from the original combinatorial problem, whereas the full-rank case (i.e., $l = n$) recovers the convex relaxation (2.9). The parameter l enables to "interpolate" between these two limit cases.

A fundamental issue with problem (2.10) is that its solutions are not isolated. For any solution W and any orthogonal matrix $Q \in \mathbf{R}^{l\times l}$, i.e., $Q^TQ = I_l$, the matrix WQ is also a solution. In other words, problem (2.10) is invariant by right multiplication of the search variable by an orthogonal matrix. As explained in the sequel, this inherent symmetry of the problem has an important impact on the optimization method used for solving (2.10).

2.2 Optimization on matrix manifolds

In the previous section, approaches for component analysis are turned into optimization problems involving *constraints*, i.e., restrictions of the search space. In this section we take a closer look on these constraints and highlight the geometry that underlies them.

Specifically, these constraints endow the search space with the structure of a *matrix manifold*, which casts the problem of component analysis in the realm of *optimization on matrix manifolds*. Several efficient algorithms have been proposed in recent years to solve this class of problems. We refer to the monograph [AMS08] for the state-of-the-art in this area.

Although all the encountered constraints induce a manifold structure on the search space, the resulting geometry can be rather different. We therefore separate them in three classes: *spherical*, *orthonormality* and *invariance* constraints. A spherical constraint enforces a vector to be of unit-norm. It is faced for instance by PCA for maximizing the Rayleigh quotient of the covariance matrix of the data (i.e., problem (2.4)). The sparse PCA formulations (2.6) and (2.7) also involve spherical constraints. Orthonormality constraints enforce a matrix to have mutually orthonormal columns and arise, e.g., to perform ICA (problem (2.5)). Finally, invariance constraints are introduced whenever the optimization problem presents *symmetries*. The Rayleigh quotient (2.3), for instance, is invariant by multiplication of the vector x by a scalar. Another symmetry is encountered for solving convex relaxations of sparse PCA by means of low-rank arguments, e.g., problem (2.10) is invariant by right multiplication of W with an orthogonal matrix.

2.2.1 Spherical constraints

Spherical constraints express that an optimization has to be performed on the *unit Euclidean sphere*,

$$\mathcal{S}^{n-1} = \{x \in \mathbf{R}^n \mid x^T x = 1\}.$$

i.e., the set of unit-norm vectors in $\mathbf{R}^n$. Examples include the one-unit sparse PCA problems (2.6) and (2.7).

Constraint optimization is often tackled by, somehow, converting the optimization problem into a more familiar unconstrained setting, for which well-known algorithms can be used (e.g., *steepest-descent/ascent*, *Newton's*, *conjugate-gradient* or *trust-region* methods). This is exactly the way the common *penalty*, *log-barrier* and *augmented Lagrangian* methods proceed: they solve a sequence of unconstrained problems. These methods handle *any* constraint, without however taking advantage of their possibly interesting structure. Details on classical methods for unconstrained as well as constrained optimization can be found, e.g., in [NW06]. In the particular case of spherical constraints, the most efficient approach is probably to use well-known tools for unconstrained optimization while simultaneously exploiting the geometry of the sphere to enforce the constraint.

Algorithms for unconstrained optimization are iterative, i.e., they compute a sequence of points (the *iterates*) that converges towards the solution of the problem. To adapt these algorithms to optimization problems on the sphere, let us define the *tangent space* to the sphere at an iterate $x \in \mathcal{S}^{n-1}$ as the set of vectors that are orthogonal to x,

$$T_x\mathcal{S}^{n-1} = \{\eta \in \mathbf{R}^n \mid \eta^T x = 0\}, \tag{2.11}$$

which is a Euclidean space (i.e., a vector space) of dimension $n-1$. To obtain a new iterate $x_+ \in \mathcal{S}^{n-1}$ that is closer to the solution, let us move away from the point x in a "good" direction $\eta \in T_x\mathcal{S}^{n-1}$, i.e., x_+ is given by the unit-norm vector

$$x_+ \overset{\text{def}}{=} \frac{x+\eta}{\|x+\eta\|}.$$

At each iteration, an update direction η of the tangent space at the current iterate has to be found, i.e., the problem amounts to search an element η in a vector space. Classical iterations for unconstrained optimization can hence be used. For instance, to perform *steepest-ascent* optimization, the vector η is chosen as an element of the tangent space that points as well as possible in the direction of the gradient of the objective function.

To sum up, the constrained optimization problem is lifted at each iterate to a vector space, where methods for unconstrained optimization can be used. This trick is possible because the sphere can be *locally* assimilated to a Euclidean space, i.e., the sphere is a *differentiable manifold*. Intuitively, a differentiable manifold is a space that is locally Euclidean, but which can have a much richer global structure. We refer to do Carmo [Car92] for a formal definition of the concept. Interestingly, the geometry of a manifold is *intrinsic*, i.e., all geometrical properties can be defined without "leaving" the manifold. An "external Euclidean world" is thus not needed.

Although an intrinsic characterization of the tangent space to the sphere exists, the definition (2.11) refers to the *embedding space* $\mathbf{R}^n$. This embedding space enables to perform computations with abstract geometrical objects in numerical algebra terms. Manifolds for which this transfer from geometry to numerics is possible are termed *matrix manifolds.* There are essentially two main categories of matrix manifolds: the *embedded submanifolds* of a Euclidean space, such as the sphere which is *embedded* in $\mathbf{R}^n$ and the *quotient manifolds*, which are discussed in the sequel.

For completeness, the *dimension* of a manifold is defined as the dimension of the tangent space. Hence, the sphere $\mathcal{S}^{n-1}$ is a manifold of dimension $n-1$.

2.2.2 Orthonormality constraints and embedded geometries

Orthonormality constraints pose the optimization problem on the *Stiefel manifold*,

$$\mathrm{St}(p,n) = \{x \in \mathbf{R}^{n\times p} | x^T x = I_p\},$$

which is the set of n-by-p matrices with orthonormal columns. Examples include the ICA problem (2.5). The Stiefel manifold has dimension $np - \frac{1}{2}p(p+1)$ and is embedded in $\mathbf{R}^{n\times p}$. In the particular case $p = 1$, it specializes to the sphere.

The optimization strategy discussed in the case of the sphere, i.e., to lift the problem to the tangent space at each iterate, is valid for *any* embedded matrix manifold. It can thus be used in the present case.

For *orthogonal matrices*, i.e., for the square case $p = n$, a further structure is available. The Stiefel manifold is then equipped with a *Lie group* structure, and is therefore renamed the *orthogonal group*

$$\mathcal{O}(n) \stackrel{\mathrm{def}}{=} \mathrm{St}(n,n) = \{x \in \mathbf{R}^{n\times n} | x^T x = I_n\}.$$

The main property of Lie groups is that the product of two elements of the group remains in the group, i.e., the product of two orthogonal matrices is an orthogonal matrix. This provides new simple ways to move on the orthogonal group and thus to satisfy the orthonormality constraints at each iteration. As a furthermore important property of Lie groups, the *whole* manifold can be mapped to the tangent space at the identity (i.e, the identity matrix I_n in the case of the orthogonal group).[6] The initial problem can hence be rewritten as an unconstrained optimization on a vector space.

These considerations suggest a large diversity of optimization methods to handle orthonormality constraints.

2.2.3 Invariance constraints and quotient geometries

Invariance constraints are introduced to deal with objective functions that have *symmetries.* Optimizing functions with symmetries entails difficulties of theoretical and practical nature.

[6] More precisely, this property holds for compact and *connected* Lie groups. Since the orthogonal group is not connected, one usually restricts the search space to the *special orthogonal group*, which is the set of orthogonal matrices with positive determinant.

For instance, the Newton method for solving the unconstrained problem

$$\max_{x \in \mathbf{R}_*^n} \frac{x^T A^T A x}{x^T x}, \tag{2.3}$$

yields the iteration $x \to 2x$, which does not increase the objective function (see Proposition 2.1.2 in [AMS08]). This lack of convergence is due to the invariance of the Rayleigh quotient by multiplication of the vector x with a scalar. This symmetry in fact prevents the *Hessian*, i.e., the "curvature matrix", of the objective function from being positive definite at the solution, which is required for the well-posedness of the Newton method and most second-order methods[7] [NW06].

The concept of *quotient manifold* enables to circumvent this issue. A few definitions are first required. We consider that two vectors $x, y \in \mathbf{R}_*^n$ are *equivalent* if they point in the same directions, i.e., $y = tx$ for a certain $t \in \mathbf{R}_*$. The *equivalence class* of x, denoted $[x]$, is defined as the set of elements of $\mathbf{R}_*^n$ that are equivalent to x, i.e.,

$$[x] \stackrel{\text{def}}{=} \{y \in \mathbf{R}_*^n \mid y = tx,\ t \in \mathbf{R}_*\}.$$

Such an equivalence class is thus a straight line of $\mathbf{R}^n$ passing though the origin, i.e., a direction in $\mathbf{R}^n$. The set of all these equivalence classes is a *quotient* of the *total space* $\mathbf{R}_*^n$. This quotient is furthermore endowed with a manifold structure, i.e., it is a *quotient manifold*. Considering the Rayleigh quotient from this manifold removes the inherent invariance by scalar multiplication. The subspace of symmetry of the Rayleigh quotient is in fact reduced to a single point of the quotient. It seems thus natural to optimize that function over the quotient manifold instead of the total space $\mathbf{R}_*^n$.

In case of problem (2.10), the objective function is invariant by right multiplication of the search variable W by an orthogonal matrix. The equivalence class of $W \in \mathbf{R}_*^{n \times l}$ is thus defined by the set

$$[W] \stackrel{\text{def}}{=} \{WQ | Q \in \mathbf{R}^{l \times l}, Q^T Q = I_l\}.$$

The set of all these equivalence classes is the quotient manifold of the total space $\mathbf{R}_*^{n \times l}$ by the orthogonal group $\mathcal{O}(l)$, denoted $\mathbf{R}_*^{n \times l}/\mathcal{O}(l)$.[8] Each point of this quotient manifold is thus a set of matrices. The minimizers of problem (2.10) are isolated on this new search space.

Equivalence classes are abstract objects that cannot be "defined " on a computer. Therefore, for numerical computations, any point $[x]$ of a quotient manifold is parameterized by a particular element x of the Euclidean total space, i.e., the total space provides a matrix representation to the elements of the quotient. By solving problems defined on a quotient manifold, attention should however be paid to have successive iterates x_i in the total space that map to distinct points $[x_i]$ in the quotient manifold. Ways to satisfy this requisite are discussed in Chapter 6.

[7] Second-order optimization methods exploit both first- and second-order derivative information on the objective function and converge usually faster than simpler first-order optimization methods.

[8] $\mathbf{R}_*^{n \times l}$ is the noncompact Stiefel manifold of *full-rank* matrices in $\mathbf{R}^{n \times l}$. The nondegeneracy condition is required to deal with differentiable manifolds.

Let us finally mention that, to get around the invariance of the Rayleigh quotient (2.3), one can also constrain the vector x to unit-norm, as previously suggested with problem (2.4). Getting rid of the symmetry of problem (2.10) by adding a suitable constraint is however cumbersome. Quotient manifolds provide in this sense a much more natural approach.

2.3 Summary

Component analysis is the problem of reducing large data to lower dimension in order to highlight the essential information hidden in the raw data. In this chapter we review several approaches for component analysis, i.e., *principal component analysis* (PCA), *independent component analysis* (ICA) and *sparse principal component analysis* (sparse PCA), and cast them as a constrained optimization problem. The constraints involved by these problems are of three kinds: spherical, orthonormality and invariance constraints. Each of them endow the problem with an interesting manifold structure. Exploiting this geometry is expected to lead to efficient algorithms for component analysis.

Chapter 3

Motivating problem: analysis of gene expression data

The algorithms for component analysis proposed in the thesis are evaluated on the analysis of gene expression data related to breast cancer. In this chapter, important details on gene expression data are first provided (Section 3.1). We then address the challenges and opportunities posed by these data and sketch how component analysis allows to progress towards meeting these goals (Section 3.2). Specifically, we suggest leads to evaluate the biological significance of the components and show how novel knowledge on the biology of breast cancer could emerge (Section 3.3).

3.1 What are gene expression data?

The transcriptome is the set of all messenger RNA (mRNA) molecules in a given cell. Unlike the genome, which is roughly similar for all the cells of an organism, the transcriptome may vary from one cell to another according to the biological role of the cell as well as to the external stimuli. In this sense, it reflects the events that occur within the cell. The quantity of a given mRNA is determined by a complex interaction between cooperative and counteracting biological processes. Understanding this intricate mechanism is an important step in elucidating the relation between the transcriptome of a cell and its phenotype.

Microarrays provide a quantitative measure of the amount of the mRNA molecules in a cell, called the *expression level* of the genes. This technology is revolutionary for life science research because, instead of looking at a very small part of the genome, it provides an overall view of it in a single assay.

In the last decade, several microarray technologies have been developed. In order to grasp the main concepts used by these technologies, let us consider the standard case of *complementary DNA (cDNA) microarrays*. A cDNA microarray is a small chip on which strands of DNA are attached at fixed spots. Each spot contains a huge number of identical DNA sequences to mark a specific gene of the genome. There can be several thousands of such spots on a single chip. Gene expression can be measured by comparing two mRNA samples, e.g., a test sample and a control sample. Both samples are first reverse-transcribed into complementary DNA (cDNA) and labeled using fluorescent dyes (e.g., red for the test sample and green for the control sample). The labeled cDNA molecules are mixed and washed over the chip so that they can hybridize to their complementary sequences in the spots. The color and intensity of fluorescence of each spot reflects the amount of hybridization of both samples to the corresponding probe. If the test sample mRNA is in abundance, the spot will be red. Conversely, it will be green if the control sample mRNA is in abundance. If both are equal, the spot will be yellow. While if neither are present, it will appear black. Relative intensity of the dyes enables to quantify the up-regulation or down-regulation of the genes in the test sample with respect to the control sample. We refer to Riva et al. [RCTH05] and references therein for more details on microarrays.

A gene expression database stores the results related to a couple of experiments, which compare a set of test samples against a control sample on distinct microarray chips. The test samples are usually drawn from cells with a common feature, e.g., cancerous cells of several unhealthy tissues or patients. The control sample, on the other hand, is collected from normal (i.e., healthy) cells. The same control sample is usually poured on all the arrays. Sometimes, however, to better compare tumor against normal tissue, control and test samples are taken from the same individual. For the sake of completeness, let us mention that some microarray technologies (e.g., oligonucleotide microarrays) estimate the absolute levels of gene expression. Two separate chips are thus required when the differential expression of a sample against a control is of interest.

Gene expression data typically contain the expression levels of several thousand genes over hundred experiments and are stored in a matrix $A \in \mathbf{R}^{m\times n}$, where n is the number of analyzed genes and m is the number of experiments. The element (i, j) of the matrix A depicts thus to the expression level of gene j during the ith experiment.

In this thesis, we analyze gene expression data sets related to breast cancer and which are briefly detailed on Table 3.1. We focus on breast cancer for two reasons. First, for this type of cancer many large patient cohorts that have been profiled with microarrays are available. Second, breast cancer is a highly heterogeneous disease and hence it provides a more challenging (and hence suitable) arena in which to compare and evaluate different methodologies.

Study	Genes (n)	Samples (m)	Reference
Vijver	13319	295	[VHV+02]
Wang	14913	285	[WKZ+05]
Naderi	8278	135	[NTBM+07]
JRH-2	14223	101	[SWL+06]

Table 3.1: Breast cancer data sets.

3.2 Component analysis of gene expression data

The challenge to take up with gene expression data is to unravel the mechanisms that give rise to the measured mRNA levels. At the early beginning of bioinformatics, people though that each gene was responsible for a particular biological function. But Nature is not so simple: genes interact. The cell is a huge network between genes, proteins and further biomolecules. With the advent of high-throughput technologies, such as microarrays, researchers started to unveil little parts of this network. Although the biological mechanisms behind gene expression are extremely complex, it is hoped that some "simple" structures, which involve a few genes only, can explain the most of specific biological processes.

Our main objective in the analysis of gene expression data is to identify genes that are systematically coexpressed under similar experimental conditions. The inferred sets of genes are presumably responsible for some specific biological functions that underlie the observed data.

The methods for component analysis reviewed in Chapter 2 perform an approximate matrix factorization of the gene expression matrix A into the product of two matrices $Y \in \mathbf{R}^{m\times p}$ and $Z \in \mathbf{R}^{n\times p}$,

$$A = YZ^T + E, \tag{3.1}$$

with the rank p that is much smaller than n, and the error term $E \in \mathbf{R}^{m\times n}$. The matrix Y contains the "expression levels" of the p components for the m experiments. The columns of the matrix Z provide an interpretation to these components as linear combinations of the n original variables, i.e., the columns of Z depict the "activation pattern" over genes of the components.

Under "good" mathematical assumptions on the components, one expects each of them to reflect important biological functions encoded in the data. Thus, the main modeling hypothesis that underlies the factorization (3.1) is that the expression level of a gene is determined by a linear superposition of biological processes, some of which try to express it, while other contending processes try to suppress it. The genes that are the most differentially activated in the columns of Z characterize the biological functions caught by the associated component.

We expect the three methods for component analysis discussed the thesis (i.e., PCA, ICA and sparse PCA) to be useful in this context. First, PCA has been shown in several studies to be a relevant tool for modeling and analysis of gene expression data (see, e.g., [ABB00, HMM+00, ABB03]). Second, the assumption of statistically independent components seems very natural to express that the main biological functions behind gene expression data take place independently, from a biological point of view. The value of ICA has also been illustrated by several studies (see, e.g., [Lie02, MMSM02, LB03, KM03, SHK+03, DMB04, ZYW+05, FVLH06, HZ06, CXW+08, LUG+08, KVG+08, SLK+08, KML+09]). Sparse PCA, finally, is expected to highlight "simple" structures in the genome that involve a few genes only, but explain a significant amount of specific biological processes encoded in the gene expression data. It is in fact reasonable to assume that most of these biological processes correspond to activation or inhibition of small sets of genes. *Biological pathways*, for instance, which are well-known series of biochemical reactions in and around a cell that ensure a certain biological function, involve usually a few genes only.

Estimating the rank p of the factorization to provide the most informative components is a hard outstanding problem. While some approaches to estimating p exist [CG07], for example, the *Bayesian information criterion* [HLK01], we decide to infer the same number of components for each method. There are two reasons for this. First, because of the still relatively small sample sizes of microarray experiments, estimating the correct number of components is difficult. It has therefore been conventional to use a fixed number of components (e.g., [LB03, CRT$^+$04]). Second, using the same number of components for each algorithm facilitates their comparison. In the thesis, we arbitrarily chose to infer *ten* components for each data set and each method.

3.3 Biological significance of the components

An important issue that is posed by component analysis is how biological knowledge could emerge from the inferred components. In order to evaluate the biological significance of the components, the methods we use in the sequel aim at "correlating" the components with established biological knowledge.

A first performance criterion is to estimate how well the inferred components map to known biological pathways. As previously mentioned, pathways define groups of genes that interact when a certain biological function is required. They are thought to provide a "good" validation framework because breast cancer is caused by aberrations in the activation of specific pathways that upset the delicate balance between expression and repression in otherwise healthy tissue.

A second performance criterion is to investigate how well the components relate to *regulatory motifs* and *transcription factors*.[1] Genes tagged with a common regulatory motif are controlled by the same transcription factor and are thus likely to be coexpressed [XLK$^+$05]. As a consequence, they are expected to appear in the same component.

Finally, gene expression datasets are usually provided with clinical data about each sample (i.e., patient). This information can be used to check whether the inferred components are associated with breast cancer phenotypes.

It should be mentioned that it is customary to evaluate components against the *Gene Ontology* (GO) [Con00] rather than against the biological pathways or the regulatory motifs. We however consider that GO does not provide the best framework to evaluate the components since many genes with the same GO term annotation may not be part of the same biological pathway or may not be under the control of the same regulatory motif, and vice versa. Furthermore, evaluating methods for component analysis in the explicit context of biological pathways and regulatory motifs is a new idea, proposed by our biologist collaborator Andrew

[1] A *transcription factor* is a protein that binds to the DNA to control the transcription of the genetic information. A given transcription factor can bind only at a specific sequence of nucleotides, the *regulatory motif*. Regulatory motifs are either located in the *promoter region* of the gene or in *three prime untranslated region* (3' UTR).

Teschendorff.

In the following sections, we derive quantitative statistical estimators for the overlap between components and pathways/regulatory motifs (Section 3.3.1) as well as for the association between components and clinical data (Section 3.3.2). These estimators, when used together, enable to infer meaningful information on the biology of breast cancer (Section 3.3.3).

3.3.1 Pathway enrichment index

The *pathway enrichment index* (PEI) evaluates how well the components map to known pathways and regulatory motifs. The *enrichment* of a component in a pathway or a regulatory motif is the statistical significance of the overlap between genes derived from established biology (i.e., the pathway or the regulatory motif) and genes that underlie the component. Components scoring a high PEI are more clearly related to important known biological functions.

While research in cancer biology is still at the stage of trying to elucidate all the pathways that may be involved, several efforts are underway in building up pathway databases. To compile a comprehensive list of pathways known to be directly or indirectly involved in cancer biology, we use the Molecular Signatures Database (MSigDB) [STM+05], which include 522 distinct pathways curated from the literature and from other databases such as KEGG[2] and CGAP.[3] We augment this list with known oncogenic pathways provided by Bild et al. [BYC+05], and cancer signalling pathways from NETPATH,[4] yielding a total of 536 pathways. The latter pathways are frequently altered in cancer and hence expected to be captured by the inferred components. Each of the pathways is described by small sets of genes known to participate together when a certain biological function is required.

A list of 173 regulatory motifs is provided by Xie et al. [XLK+05]. For each such motif, the associated *regulatory gene module* is defined as the set of genes having this motif in their promoters or 3' UTR, as provided in MSigDB [STM+05]. Testing the inferred components for enrichment of regulatory modules provides putative links between components and the transcription factors that bind to these motifs.

In the component analysis model (2.2), the columns of Y contain the expression level of the component for the m experiments, while the columns of Z reflect the activation pattern over genes of the corresponding component. The most differentially activated genes in the columns of Z are considered to underlie the biological functions mapped by the components. These genes are conventionally identified by setting a threshold, typically two or three standard deviations from the mean, and selecting those genes whose absolute weights exceed this threshold. To focus on the pathways/regulatory modules that dominate a component, we use the more stringent threshold of *three* standard deviations on either side from the zero mean, which picks out the 0.2% of genes in the tails of the signed weight distributions.

[2] http://www.genome.jp/kegg/
[3] http://cgap.nci.nih.gov/
[4] http://www.netpath.org

The significance of enrichment of genes from a pathway/regulatory module in a component is evaluated by using the *hypergeometric test*. Let n denote the total number of genes in the database, n_1 the number of genes selected from the component, n_2 the number of genes in the pathway and t_{12} the number of genes that are present in both sets. Under the null-hypothesis, where the selected genes are chosen randomly, the number t_{12} follows a hypergeometric distribution [BS04]. Specifically, the probability distribution is

$$P(t) = \binom{n_1}{t} \prod_{k=0}^{t-1} \frac{n_2 - k}{n - k} \prod_{k=0}^{n_1-t-1} \frac{n - n_2 - k}{n - t - k} = \frac{\binom{n_2}{t}\binom{n-n_2}{n_1-t}}{\binom{n}{n_1}},$$

where the binomial coefficient $\binom{n}{k}$ denotes the number of combinations of k elements in a set of n elements. For each component-pathway pair, this yields a p-value $P(t > t_{12})$, which estimates how enriched the component is in terms of genes from that particular pathway. Correction for multiple testing is done using the Benjamini-Hochberg procedure to obtain an estimate for the false discovery rate (FDR) [BH95]. A component is then declared enriched for a certain pathway if the Benjamini-Hochberg corrected p-value is less than 0.05. Hence, we would expect approximately 5% of significant tests to be false positives. Finally, we count the number of pathways enriched in at least one component and defined the PEI as the corresponding fraction of enriched pathways. More details on the PEI can be found in Teschendorff at al. [TJA+07].

3.3.2 Association with clinical data

The four breast cancer data set of Table 3.1 are provided with three categorical phenotypes: *estrogen receptor status*, *histological grade*, and *clinical outcome*.

The estrogen receptor status (ER) is either positive or negative, the positive case meaning that a significant number of cancer cells have estrogen receptors. Such cells are more likely to grow and multiply in a high-estrogen environment. ER-positive cancers can thus be treated by hormonal therapy, the goal of which is to starve the breast cancer cells of estrogen. The histological grade is an indicator of prognosis in breast cancer. It is a score given on a three-tier scale that estimates how much the tumor cells resemble or differ from the normal cells of the same tissue type. A low grade cancerous cell looks almost like a normal tissue and grows thus slowly. On the other hand, high grade cells grow rapidly and are very aggressive. Clinical outcome is either dead or alive.

To evaluate statistical significance of an association between a component and a phenotype, we separate the weights in the corresponding column of the matrix Y across the different categories and test these groups of weights to assess whether they are drawn from equal probability distributions. The component and the phenotype are considered to be related if this null-hypothesis is rejected. The distribution of all weights of the component is otherwise independent of the phenotype. The Wilcoxon rank-sum test is used for the two binary phenotypes and the Kruskal-Wallis test is used for histological grade. Both tests are based upon the null-hypothesis that the groups come from distributions with equal medians.

3.3.3 Inference of novel biological knowledge

Biological knowledge could emerge from the combination of the two estimators of biological significance of the preceding sections. The PEI characterizes each component by the differential activation pattern of cancer-related pathways and regulatory modules. For those components associated with a phenotype, it is hence possible to link the corresponding pathways and regulatory modules with the phenotype. In other words, the components are used as an intermediary to link pathways/regulatory modules with phenotypes. Such relationships are valuable biological information. In the forthcoming chapters, we show that this approach for evaluating the biological significance of components leads to well-known biological relationships but also to novel ones.

3.4 Summary

The present chapter details the problem of analyzing gene expression data, which motivates the research presented in this thesis. Gene expression data are large databases that store the expression level of thousands of genes for a couple of cells. They open new perspectives in the understanding of genetic diseases, such as cancer.

Gene expression data provide a challenging framework to evaluate the algorithms for component analysis of the thesis. Specifically, we propose strategies to evaluate the biological relevance of components extracted from breast cancer gene expression data and show how new knowledge on the biology of cancer could emerge from this study.

The idea of using components as computational tools to link pathways or motifs to phenotypes in presented in [TJA$^+$07].

Chapter 4

Optimization on the Stiefel manifold and its application to ICA

The Stiefel manifold, denoted $\mathrm{St}(p,n)$ with $p \leq n$, is defined as the space of p-dimensional orthonormal bases in an n-dimensional Euclidean space, i.e., the set of matrices of $\mathbf{R}^{n\times p}$ with orthonormal columns,

$$\mathrm{St}(p,n) = \{x \in \mathbf{R}^{n\times p} \mid x^T x = I_p\}.$$

It is an embedded submanifold of $\mathbf{R}^{n\times p}$ of dimension $np - \frac{1}{2}p(p+1)$ [AMS08]. In the particular case $p = 1$, the Stiefel manifold corresponds to the unit Euclidean sphere,

$$\mathcal{S}^{n-1} = \{x \in \mathbf{R}^n \mid x^T x = 1\}.$$

The square case $p = n$ provides the orthogonal group,

$$\mathcal{O}(n) = \{x \in \mathbf{R}^{n\times n} \mid x^T x = I_n\}.$$

In the present chapter, we focus on optimization problems of the form

$$\min_{x\in\mathrm{St}(p,n)} f(x), \tag{P$_1$}$$

for a smooth objective function $f : \mathrm{St}(p,n) \to \mathbf{R}$. Aside differentiability, no assumption is imposed on the function f. This class of optimization problems encloses formulations of principal component analysis (PCA) and independent component analysis (ICA). The optimization methods discussed in this chapter compute a *local* solution of problem (P$_1$), i.e., a solution that is optimal with respect to the neighboring points rather than with respect to all the points of $\mathrm{St}(p,n)$.

This chapter is organized as follows. PCA and ICA are first cast into optimization problems on the Stiefel manifold (Sections 4.1 and 4.2). Optimization methods for solving problem (P$_1$) are then discussed (Section 4.3). For the particular case of the orthogonal group ($p = n$), further optimization strategies are conceivable due to additional geometrical properties (Section 4.4). All these optimization methods provide algorithms for ICA (Section 4.5), which are evaluated on the analysis of gene expression data (Section 4.7). Some numerical experiments are also proposed, which compare the convergence of the discussed optimization methods (Section 4.6).

4.1 Principal component analysis

If $A \in \mathbf{R}^{m\times n}$ is a matrix encoding m samples of n variables, with n being large, PCA aims at finding linear combinations of these variables, the *principal components*, which are uncorrelated and explain as much of the variance in the data as possible. Although it goes back to the beginning of the 20th century with the seminal article of Pearson [Pea01] and, somewhat latter, the contribution of Hotelling [Hot33], PCA is still an active topic of research, with many papers and several books devoted to it. The problem has been investigated independently by different fields of research, leading to a large variety of algorithms and *names*. From an algorithmic viewpoint, the proper orthogonal decomposition [Lum67], the Karhunen-Loève transform [Ger81], the singular value decomposition [GVL89] and principal component analysis [Jol04] are *essentially* the same.

The objective in this thesis is not to compete with state-of-the-art methods for PCA. Nevertheless, discussing PCA from an optimization perspective is useful to formulate extensions, such as ICA or sparse PCA.

When introducing PCA in Chapter 2, we started from the approximate factorization model

$$A = YZ^T + E, \tag{2.2}$$

where the matrix $Z \in \mathbf{R}^{n\times p}$ defines the new orthonormal basis in which to view the data and the matrix $Y \in \mathbf{R}^{m\times p}$ contains m samples of the p principal components. As stated in Theorem 2.1.5, the matrices Y and Z are obtained through the rank-p singular value decomposition $U_p\Sigma_p V_p^T$ of the data matrix A, i.e.,

$$Z = V_p \quad \text{and} \quad Y = U_p\Sigma_p.$$

Since the singular value decomposition minimizes the Frobenius norm of the approximation error E (Theorem 2.1.6), PCA can be cast as the optimization problem

$$\begin{array}{ll} \min\limits_{Y,Z} & \|A - YZ^T\|_F \\ \text{s.t.} & Y \in \mathbf{R}^{m\times p}, \\ & \mathrm{Off}(Y^TY) = 0, \\ & Z \in \mathrm{St}(p,n), \end{array} \tag{4.1}$$

which constrains the columns of Y to be orthogonal and those of Z to be orthonormal. In the hypothetic case where one extracts only *one* principal component, problem (4.1) consists in maximizing the Rayleigh quotient of the covariance matrix A^TA,[1] i.e.,

$$\min_{\substack{y\in\mathbf{R}^m \\ z\in\mathcal{S}^{n-1}}} \|A - yz^T\|_F^2 = \min_{\substack{y\in\mathbf{R}^m \\ z\in\mathcal{S}^{n-1}}} \mathrm{Tr}(A^TA) - 2y^TAz + y^Ty \tag{4.2}$$

$$= \mathrm{Tr}(A^TA) - \max_{z\in\mathcal{S}^{n-1}} z^TA^TAz, \tag{4.3}$$

[1]Because the columns of A are assumed to be centered, the Gram matrix A^TA equals the sample covariance matrix of the data up to a scalar multiplier, which is simply omitted for the sake of clarity.

where $y = Az$ is the optimizer of (4.2) at any $z \in \mathcal{S}^{n-1}$. Instead of optimizing with respect to y first and then to z, the alternative gives the "dual" formulation

$$\min_{\substack{y \in \mathbf{R}^m \\ z \in \mathcal{S}^{n-1}}} \|A - yz^T\|_F^2 = \mathrm{Tr}(A^T A) - \max_{\bar{y} \in \mathcal{S}^{m-1}} \bar{y}^T A A^T \bar{y}, \tag{4.4}$$

with $y = (\bar{y}^T A A^T \bar{y})^{\frac{1}{2}} \bar{y}$ and the solution $z = \frac{A^T y}{\|A^T y\|}$ of (4.2) at any $y \in \mathbf{R}^m$. Extracting one principal component amounts to computing the dominant eigenvector of either $A^T A$ or AA^T. In the case of gene expression data where the number m of samples in much smaller than the number n of variables, it is naturally recommended to solve (4.4) instead of (4.3). Further components are obtained by computing the first principal component of the residual data matrix $A - yz^T$. Such a sequential evaluation of the components is termed a *deflation* process.

Block algorithms for PCA, which extract several components at once, solve for instance the optimization problem

$$\max_{Z \in \mathrm{St}(p,n)} \mathrm{Tr}(Z^T A^T A Z N) \tag{4.5}$$

which has the same solution Z as (4.1) provided that the diagonal parameter matrix N has distinct diagonal elements [Bro91, AMS08]. The solution Y of (4.1) is then given by $Y = AZ$.

4.2 Independent component analysis

Independent component analysis (ICA) provides a linear representation of the data in terms of components that are statistically independent. In the approximate matrix factorization model

$$A = YZ^T + E, \tag{2.2}$$

the columns of Y are assumed to contain samples drawn from statistically independent random variables. Random variables are, per definition, statistically independent if their conditional probabilities are equal to the "unconditional" (i.e., marginal) probabilities. In other words, random variables are independent if the value of any one variable does not carry any information on the value of any other variable. ICA was originally dedicated to the *blind source separation* problem, which recovers independent sources from linear mixtures of them [Com94].

The approximation error E is usually minimized in the least square sense and the components are enforced to be uncorrelated, which is a necessary condition for statistical independence. As discussed in Chapter 2, ICA amounts then to finding the orthonormal transformation of the principal components to maximize statistical independence. Let $\bar{Y} \in \mathbf{R}^{m \times p}$ contain m samples of the p principal components (described by a random *row* vector $\bar{y}$ of dimension p). Since the independent nature of random variables is not altered by a scaling of these variables, the random variables in $\bar{y}$ are assumed, without lost of generality, to be *white*, i.e., they all have a unit variance. The orthogonal columns of $\bar{Y}$ are thus also of unit-norm, i.e., they are orthonormal.

In practice, statistical independence has to be appraised by means of finite sets of samples. ICA maximizes thus a *contrast function*

$$f : \mathcal{O}(p) \to \mathbf{R} : Q \mapsto f(Q)$$

that provides a quantitative estimate of independence between the p components $\bar{y}Q$. We refer to Comon [Com94] for the definition of this concept. The only requirement on the contrast function is that it approaches, with probability one, to a prescribed extremum (say zero) if and only if the random variables are statistically independent and as the number of samples m goes to infinity. This leaves many possibilities for the contrast function, leading to a variety of ICA algorithms, which may also differ in their numerical implementation.

In this *square* ICA setting, dimension reduction is exclusively achieved during the preprocessing by PCA. This task can however be shared by both PCA and ICA steps by computing the best rank-$\bar{p}$ factorization of the data A with $\bar{p} > p$ and identifying then a matrix $Q \in \mathrm{St}(p, \bar{p})$ to maximize statistical independence, i.e.,

$$\max_{Q \in \mathrm{St}(p,\bar{p})} f(Q), \tag{4.6}$$

where the contrast function $f : \mathrm{St}(p, \bar{p}) \to \mathbf{R}$ estimates the statistical independence of the p components $\bar{y}Q$ and the random vector $\bar{y}$ of the principal components is of dimension $\bar{p}$. The components obtained with this approach are still uncorrelated and they potentially reach a better statistical independence. The error $E \stackrel{\text{def}}{=} A - YZ^T$ is however not minimal. This *soft* dimension reduction approach is for instance considered by Theis at al. [TCA09] for the analysis of biomedical imaging data.

We review below some standard contrast functions. These functions rest on various concepts of probability and information theory, which are explained, e.g., in [Mac02, CT06]. Our intention here is simply to highlight the essence of these contrasts as well as the main differences among them, without going into the details.

Mutual information

Statistical independence is typically characterized by the *mutual information* $I(x)$, defined as the Kullback-Leibler divergence between the joint distribution and the product of the marginal distributions of the multivariate random variable $x = (x_1 \ldots, x_p)$,

$$I(x) \stackrel{\text{def}}{=} \int \mathrm{p}(x) \log \frac{\mathrm{p}(x)}{\mathrm{p}(x_1) \ldots \mathrm{p}(x_p)} \, dx_1 \ldots dx_p,$$

where $\mathrm{p}(x)$ denotes the probability density function of x. The mutual information is nonnegative and equals zero if and only if the random variables x_i are all mutually statistically independent.

Practical formulations of the mutual information rest on the expansions

$$I(x) = \sum_{i=1}^{p} S(x_i) - S(x) \quad \text{and} \quad I(x) = J(x) - \sum_{i=1}^{p} J(x_i), \tag{4.7}$$

with the *differential entropy* $S(x)$ and *negentropy* $J(x)$ that are defined by

$$S(x) \stackrel{\text{def}}{=} \int \mathrm{p}(x)\log(\mathrm{p}(x))dx, \quad \text{and} \quad J(x) \stackrel{\text{def}}{=} S(g) - S(x),$$

respectively, where g is a Gaussian variable with same mean and variance as x. In the second expansions in (4.7), i.e., the expansion of the mutual information in terms of negentropies, the random vector x is assumed to have a zero mean and to be white, i.e., its covariance matrix is the identity. As previously mentioned, this assumption can always be enforced in the context of ICA. Contrast functions are obtained from (4.7) by posing $x = \bar{y}Q$,

$$f_S(Q) = \sum_{i=1}^{p} S(\bar{y}Qe_i) - S(\bar{y}) \quad \text{and} \quad f_J(Q) = J(\bar{y}) - \sum_{i=1}^{p} J(\bar{y}Qe_i), \tag{4.8}$$

where e_i is the ith canonical basis vector. More details on the derivation of these expressions can be found in [LF03] for f_S and in [Com94] for f_J.

Both contrasts f_S and f_J require to estimate *unidimensional* entropies and negentropies. This is for instance achieved by means of *cumulants*.[2] Let x be a *standardized* one-dimensional random variable x, i.e., x has zero mean and unit variance. A truncated Edgeworth expansion of the probability distribution of x provides the following fourth-order approximation of the negentropy,

$$J(x) \approx \frac{1}{12}\kappa_3^2(x) + \frac{1}{48}\kappa_4^2(x) + \frac{7}{48}\kappa_3^4(x) - \frac{1}{8}\kappa_3^2(x)\kappa_4(x),$$

where κ_i denotes the ith cumulant [Com04]. *Order statistics* also provide efficient estimators of the entropy/negentropy. Given m samples of x, the order statistic is the set of samples $\{x^1, \ldots, x^m\}$ rearranged in non-decreasing order, i.e., $x^1 \leq \ldots \leq x^m$. The differential entropy of x is estimated by the formula,

$$S(x) \approx \frac{1}{m-k} \sum_{j=1}^{m-k} \log\left(\frac{m+1}{k}(x^{(j+k)} - x^{(j)})\right),$$

where k is usually set to $\sqrt{m}$. This expression is derived from a statistical estimator proposed by Vasicek [Vas76].

[2] Given a unidimensional random variable x, let $g_x(t)$ be the *cumulant-generating function* defined as

$$g_x(t) \stackrel{\text{def}}{=} \mathbb{E}[e^{tx}].$$

Cumulants are defined by the derivatives $\kappa_n(x) \stackrel{\text{def}}{=} \left.\frac{\mathrm{d}^n g_x(t)}{\mathrm{d}t^n}\right|_{t=0}$. Cumulants up to order 4 are given by

$$\begin{aligned}
\kappa_1(x) &= \mathbb{E}[x], \\
\kappa_2(x) &= \mathbb{E}[x^2] - \kappa_1^2(x), \\
\kappa_3(x) &= \mathbb{E}[x^3] - 3\kappa_2(x)\kappa_1(x) - 4\kappa_1^3(x), \\
\kappa_4(x) &= \mathbb{E}[x^4] - 4\kappa_3(x)\kappa_1(x) - 3\kappa_2^2(x) - 6\kappa_2(x)\kappa_1^2(x) - \kappa_1^4(x).
\end{aligned}$$

The cumulant $\kappa_1(x)$ is the mean and the cumulant $\kappa_2(x)$ is the variance.

Nongaussianity

The *central limit theorem* states that a sum of independent random variables of any distribution converges (in distribution) to a Gaussian variable as the number of terms tends to infinity. Each linear combination of random variables is thus expected to be more Gaussian than the original ones. Independent components should therefore be as non-Gaussian as possible. A whole range of contrast functions are derived from estimators of *nongaussianity* of random variables. The following estimator, for instance, measures the "distance" between the probability distributions of a random variable x and a Gaussian variable g with same mean and variance as x,

$$\eta(x) = (\mathbb{E}[G(x)] - \mathbb{E}[G(g)])^2, \tag{4.9}$$

where $\mathbb{E}[\cdot]$ is the expectation operator and G is a smooth even function [HKO01]. The choice $G(x) = \frac{1}{4}x^4$ recovers the cumulant $\kappa_4(x)$,

$$\eta(x) = \kappa_4(x)^2,$$

which, in case of a standardized random variable, is the *kurtosis*, a well-known estimator of nongaussianity. In this particular case, the intuitive relationship between nongaussianity and statistical independence is corroborated by the following result.

Theorem 4.2.1 *The kurtosis of the sum of two independent variables x_1 and x_2 presents a smaller absolute value than the largest absolute value of the kurtosis among these variables, i.e.,*

$$|\kappa_4(x_1 + x_2)| \leq \max(|\kappa_4(x_1)|, |\kappa_4(x_2)|).$$

Proof. See [Mat01]. □

Further possibilities for the function G in (4.9) are suggested by Hyvärinen et al. [HKO01].

Contrast functions are obtained by measuring the nongaussianity of the components $\bar{y}Q$,

$$f(Q) = \sum_{i=1}^{p} (\mathbb{E}[G(\bar{y}Qe_i)] - \mathbb{E}[G(g)])^2, \tag{4.10}$$

with g a Gaussian random variable of zero mean and unit variance. These contrasts are used by the FastICA algorithm [HKO01], which is probably the most popular algorithm for ICA.

Joint diagonalization of cumulant matrices

The Nth-order cumulant tensor $\mathcal{C}_x^{(N)}$ of a p-dimensional random vector $x = (x_1, \ldots, x_p)^T$ is defined in an element-wise manner as

$$(\mathcal{C}_x^{(N)})_{i_1 \ldots i_N} \stackrel{\text{def}}{=} \kappa(x_{i_1}, \ldots, x_{i_N}), \quad i_1, \ldots, i_N \in [1, p],$$

where $\kappa(x_{i_1}, \ldots, x_{i_N})$ is the joint cumulant of the N random variables $x_{i_1}, \ldots x_{i_N}$.[3] In case of a zero-mean random vector x, the second-order cumulant is the covariance matrix. If the random vector x has mutually independent components, the cumulant tensors of order $N \geq 2$ are all diagonal. ICA is thus performed by computing the rotation that diagonalizes as well as possible the cumulant tensors of the components. In practice, one cannot consider cumulants of any order. Because the second-order cumulant tensor of the principal components is already diagonal and the cumulant of order three is identically zero for symmetric probability distributions, one diagonalizes as well as possible the *fourth*-order cumulant tensor. Performing numerical computations with tensors can, however, be very tricky. An alternative consists in deriving a set of matrices from $\mathcal{C}_x^{(N)}$, e.g., the *cumulant matrices*, that are diagonal in case of statistical independence. Cumulant matrices are symmetric matrices defined element-wise by

$$(C_x(M))_{i_1 i_2} \stackrel{\text{def}}{=} \sum_{i_3, i_4} (\mathcal{C}_x^4)_{i_1 i_2 i_3 i_4} M_{i_3 i_4}.$$

They are diagonal for any $M \in \mathbf{R}^{p \times p}$ if the tensor $\mathcal{C}_x^4$ is diagonal. Furthermore, these matrices are efficiently evaluated without the computation of the whole tensor $\mathcal{C}_x^4$,

$$C_x(M) = \mathbb{E}[(x^T M x) x x^T] - \mathbb{E}[x x^T] \operatorname{Tr}(M \mathbb{E}[x x^T]) - \mathbb{E}[x x^T](M + M^T) E[x x^T],$$

where the random vector x is assumed to have a zero mean [CS93]. A set of cumulant matrices is constructed by picking some matrices M. This set of matrices contains the same information as the whole cumulant tensor if the selected M form an orthogonal basis for the Euclidean space of the symmetric matrices of $\mathbf{R}^{p \times p}$ [Car99].

As a further property, an orthogonal transform $x = \bar{y} Q$ with $Q \in \mathcal{O}(p)$ results in the similarity transform

$$C_x(M) = Q^T C_{\bar{y}}(M) Q,$$

whatever the matrix M. ICA is thus performed by maximizing the contrast function

$$f(Q) = \sum_i \| \operatorname{Diag}(Q^T C_{\bar{y}}(M_i) Q) \|_F^2, \tag{4.11}$$

proposed in [CS93], or by minimizing

$$f(Q) = \sum_i \| \operatorname{Off}(Q^T C_{\bar{y}}(M_i) Q) \|_F^2, \tag{4.12}$$

where $\operatorname{Off}(x) = x - \operatorname{Diag}(x)$ extracts the non-diagonal elements of the matrix x [Car99, PS00, WSC05, AG06]. These two problems either maximize the diagonal elements or minimize the off-diagonal elements of the cumulant matrices $C_x(M)$, which is consistent with the objective of diagonalizing these matrices as well as possible.

[3] Let x_1, x_2, x_3 and x_4 be zero mean random variables. The second and fourth order joint cumulants are given by

$$\kappa(x_1, x_2) = \mathbb{E}[x_1 x_2],$$
$$\kappa(x_1, x_2, x_3, x_4) = \mathbb{E}[x_1 x_2 x_3 x_4] - \mathbb{E}[x_1 x_2]\mathbb{E}[x_3 x_4] - \mathbb{E}[x_1 x_3]\mathbb{E}[x_2 x_4] - \mathbb{E}[x_1 x_4]\mathbb{E}[x_2 x_3].$$

Nonlinear correlation

Pearson's correlation coefficient is a typical measure of correlation between random variables. Given two random variables x_1 and x_2, it is defined as the covariance between these variables divided by the product of their standard deviations,

$$\mathrm{corr}(x_1, x_2) = \frac{\mathbb{E}[(x_1 - \mu_1)(x_2 - \mu_2)]}{\sqrt{\mathbb{E}[(x_1 - \mu_1)^2]\mathbb{E}[(x_2 - \mu_2)^2]}},$$

where μ_1 and μ_2 are the means of x_1 and x_2, respectively. Estimators of statistical independence are derived by extending this measure to higher-order statistics. The *$\mathcal{F}$-correlation* between two random variables x_1 and x_2, for instance, is defined by

$$\rho_{\mathcal{F}}(x_1, x_2) = \max_{f_1, f_2 \in \mathcal{F}} \mathrm{corr}(f_1(x_1) f_2(x_2)), \tag{4.13}$$

where $\mathcal{F}$ is a vector space of functions from $\mathbf{R}$ to $\mathbf{R}$. It can be proven that two random variables x_1 and x_2 are statistically independent if and only if the $\mathcal{F}$-correlation $\rho_{\mathcal{F}}$ is zero, up to some conditions on the function space $\mathcal{F}$ [BJ03]. In particular, $\mathcal{F}$ should have infinite dimension.

Although this measure of independence provides a contrast to compute *two* independent components, it can be extended in a heuristic manner to a larger number of components [BJ03]. The maximization over the infinite dimensional space of functions $\mathcal{F}$ is approximated by means of *kernel methods*,[4] which transform the problem into a generalized eigenvalue problem of dimension pm, where p and m are the number of components and the number of samples, respectively. Further kernel-based estimators of statistical independence have been recently proposed (see, e.g., [GHS$^+$05, AS08]).

4.3 Optimization methods on the Stiefel manifold

Consider the optimization problem

$$\min_{x \in \mathrm{St}(p,n)} f(x), \tag{P$_1$}$$

where the objective $f : \mathrm{St}(p, n) \to \mathbf{R}$ is a smooth function and the search variable x is constrained onto the Stiefel manifold.

Unconstrained optimization problems on $\mathbf{R}^n$ can be solved by *line-search* methods that repeatedly shift the iterate $x \in \mathbf{R}^n$ in a direction $\eta \in \mathbf{R}^n$ with a certain step size $t \geq 0$,

$$x_+ = x + t\eta, \tag{4.14}$$

such that the objective function decreases from x to the new iterate x_+. Also very common, the *trust-region* method approximates at each iteration the objective by a (usually quadratic) model function and optimizes this model on a restricted domain, the *trust region*.

Methods for unconstrained optimization are usually well-understood and efficient. One would of course like to use them, as much as possible, once the search variable has to satisfy

[4] See, e.g., [Sai88, SS01] for more details on kernel methods.

some constraints. A basic method to enforce a constraint is to construct a penalty function $\varphi : \mathbf{R}^n \to \mathbf{R}$, which is zero when the constraint holds and positive elsewhere and to minimize the penalized objective $f(x) + \gamma\varphi(x)$, where the positive weight γ is steadily increased. This approach is however often computationally inefficient and the constraint is, after all, only approximately satisfied, which is not sufficient in most problems.

Another method is to project each iterate onto the *constraint surface*, i.e., the sets of points of $\mathbf{R}^n$ that satisfy the constraint. For instance, in the case of the spherical constraint $x \in \mathcal{S}^{n-1}$, each iterate is normalized to unit-norm. This approach can be successful in special cases (see Chapter 5) but can also totally fail. In fact, the iterate is first moved, sometimes far away from the constraint surface, to minimize the objective, and has then to be projected back to satisfy the constraint. Big movements are operated in the embedding Euclidean space, whereas the resulting update on the constraint surface can be rather small.

More efficient methods for constrained optimization can however be obtained by taking advantage of the rich geometry that sometimes underlie the constraints. As briefly explained in Section 2.2, the manifold structure of problem (P_1) enables to adapt methods for unconstrained optimization in the context of orthonormality constraints. Intuitively, enforcing the search direction to be *tangent* to the constraint surface should reduce the tendency to move away from it. In recent years, several classical tools for unconstrained optimization have been tailored to tackle manifold constraints. In the next sections, we review the major concepts and achievements. Most of the following material is extracted from the monograph [AMS08].

Let us beforehand mention that methods for unconstrained optimization typically work on $\mathbf{R}^n$, but can be readily extended to matrix search spaces (i.e., $\mathbf{R}^{n\times p}$): either the search variable is vectorized (the columns are stacked on top of each other) or better, the methods are rewritten with matrix variables. For instance, iteration (4.14) remains valid for an iterate $x \in \mathbf{R}^{n\times p}$ and a search direction $\eta \in \mathbf{R}^{n\times p}$.

4.3.1 Line-search on a manifold

The notion of *tangent vector* to a manifold is essential in this context. This concept is somewhat intuitive for embedded manifolds. In the case of the sphere, for instance, a tangent vector at a point $x \in \mathcal{S}^{n-1}$ is an element $\eta \in \mathbf{R}^n$ that is orthogonal to x, i.e., $\eta^T x = 0$. The tangent vector η is so defined as an element of the embedding space $\mathbf{R}^n$. The purpose of differential geometry, however, is to define any property of a manifold in an *intrinsic* manner, i.e., without referring to an "external world". This is essential to treat manifolds without the need of embedding them in a larger space (e.g., for quotient manifolds, that are also encountered in this thesis).

Consider a manifold $\mathcal{M}$, a point $x \in \mathcal{M}$, a smooth function $f : \mathcal{M} \to \mathbf{R}$ and a smooth curve $\gamma : \mathbf{R} \to \mathcal{M} : t \mapsto \gamma(t)$ such that $\gamma(0) = x$. Let $\dot\gamma(0)$ be a mapping that takes the function f and returns the directional derivative

$$\mathrm{D}f(x)[\dot\gamma(0)] \stackrel{\text{def}}{=} \left.\frac{\mathrm{d}(f(\gamma(t)))}{\mathrm{d}t}\right|_{t=0}.$$

This mapping defines a tangent vector to the curve γ at x. Since γ is included in $\mathcal{M}$, the mapping $\dot{\gamma}(0)$ is also a tangent vector to the manifold $\mathcal{M}$. The set of all tangent vectors to $\mathcal{M}$ at a point $x \in \mathcal{M}$ is the *tangent space* to $\mathcal{M}$ at x, denoted $T_x\mathcal{M}$. Because its elements are derivative operators, it is a vector space.

A line-search on a manifold $\mathcal{M}$ consists of selecting a tangent vector $\eta \in T_x\mathcal{M}$ and moving from the current iterate $x \in \mathcal{M}$ with a certain step size along a curve $\gamma(t) \in \mathcal{M}$ that verifies $\gamma(0) = x$ and $\dot{\gamma}(0) = \eta$. This generalizes the classical line-search iteration (4.14) to

$$x_+ = R_x(t\eta), \tag{4.15}$$

with the step size $t \geq 0$ and the *retraction* $R_x(\eta)$. The retraction $R_x(\eta)$ is a mapping from the tangent space to the manifold such that the curve $\gamma : \mathbf{R} \to \mathcal{M} : t \mapsto \gamma(t) = R_x(t\eta)$ passes through x at $t = 0$ and its tangent vector at $t = 0$ is η.

Both cases of the sphere and the Stiefel manifold are discussed below.

Example 4.3.1 (Line-search on the sphere) *Let $t \mapsto \gamma(t)$ be a curve on $\mathcal{S}^{n-1}$, i.e., $\gamma(t) \in \mathbf{R}^n$ such that*

$$\gamma(t)^T\gamma(t) = 1. \tag{4.16}$$

at any t. Let f be a differentiable function defined in the neighborhood of $\gamma(0)$. The directional derivative of f along the curve γ at $t = 0$ is

$$\begin{aligned}
\left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0} &= \sum_{i=1}^{n} \left.\frac{\partial f}{\partial \gamma_i}\right|_{\gamma(0)} \left.\frac{\mathrm{d}\gamma_i(t)}{\mathrm{d}t}\right|_{t=0} \\
&= \underbrace{\left[\left.\frac{\mathrm{d}\gamma_1(t)}{\mathrm{d}t}\right|_{t=0}, \ldots, \left.\frac{\mathrm{d}\gamma_n(t)}{\mathrm{d}t}\right|_{t=0}\right]}_{\dot{\gamma}(0)^T} \underbrace{\left[\left.\frac{\partial f}{\partial \gamma_1}\right|_{\gamma(0)}, \ldots, \left.\frac{\partial f}{\partial \gamma_n}\right|_{\gamma(0)}\right]^T}_{\nabla f(\gamma(0))},
\end{aligned}$$

where $\gamma_i(t)$ denotes the ith component of the vector $\gamma(t)$. The mapping $f \to \dot{\gamma}(0)^T\nabla f(\gamma(0))$ is thus a tangent vector at $\gamma(0)$. In the basis $\{\left.\frac{\partial\cdot}{\partial\gamma_1}\right|_{\gamma(0)}, \ldots, \left.\frac{\partial\cdot}{\partial\gamma_n}\right|_{\gamma(0)}\}$, the coordinates of this vector are provided by $\dot{\gamma}(0)$.

The differentiation of (4.16) with respect to t yields

$$\dot{\gamma}(t)^T\gamma(t) + \gamma(t)^T\dot{\gamma}(t) = 0.$$

The tangent space at a point x is hence the set of vectors of $\mathbf{R}^n$, which are orthogonal to x, i.e.,

$$T_x\mathcal{S}^{n-1} = \{\eta \in \mathbf{R}^n \mid \eta^T x = 0\}.$$

This is consistent with intuition. Since any elements $x \in \mathcal{S}^{n-1}$ and $\eta \in T_x\mathcal{S}^{n-1}$ are both represented by vectors in $\mathbf{R}^n$, they can be added. A possible retraction is given by

$$R_x(\eta) = \frac{x + \eta}{\|x + \eta\|_2},$$

which projects the sum $x + \eta$ back onto the sphere.

Example 4.3.2 (Line-search on the Stiefel manifold) *Let $\gamma(t)$ be a curve on* $\mathrm{St}(p,n)$*, i.e., $\gamma(t) \in \mathbf{R}^{n\times p}$ and*

$$\gamma(t)^T\gamma(t) = I_p. \tag{4.17}$$

for any t. The tangent vector $\dot\gamma(t)$ to the curve γ is represented by a matrix of $\mathbf{R}^{n\times p}$, which can be decomposed as

$$\dot\gamma(t) = \gamma(t)\Omega(t) + \gamma_\perp(t)K(t),$$

where $\Omega(t) \in \mathbf{R}^{p\times p}$, $K(t) \in \mathbf{R}^{(n-p)\times p}$ and $\gamma_\perp(t) \in \mathbf{R}^{n\times(n-p)}$ spans the orthogonal complement of the subspace spanned by $\gamma(t)$. The differentiation of (4.17) yields

$$\dot\gamma(t)^T\gamma(t) + \gamma(t)^T\dot\gamma(t) = 0.$$

Hence, $\Omega(t)^T + \Omega(t) = 0$, i.e., $\Omega(t)$ is skew-symmetric. The tangent space corresponds thus to the set

$$T_x\mathrm{St}(p,n) = \{x\Omega + x_\perp K \mid \Omega = -\Omega^T, K \in \mathbf{R}^{(n-p)\times p}\}.$$

Possible retractions are provided by

$$R_x(\eta) = \mathrm{qf}(x+\eta) \qquad \textit{or} \qquad R_x(\eta) = \mathrm{uf}(x+\eta),$$

where $\mathrm{qf}(x)$ *denotes the Q factor of the QR decomposition of the matrix x (i.e., $x = QR$, where $Q \in \mathrm{St}(p,n)$ and $R \in \mathbf{R}^{p\times p}$ is an upper triangular matrix) and* $\mathrm{uf}(x)$ *denotes the U factor of the polar decomposition of x (i.e., $x = US$, where $U \in \mathrm{St}(p,n)$ and $S \in \mathbf{S}^p$ is a positive semidefinite matrix.).*

4.3.2 First-order differential-geometric methods

In the line-search iteration (4.14), the search direction η has to be a *descent* direction, i.e., a small shift in that direction decreases the objective. *Steepest-descent* methods perform a search in the direction opposite to the gradient of the objective at the current iterate. The notion of "gradient of a function on a manifold" needs to be defined. For that purpose, we endow the manifold $\mathcal{M}$ with a *Riemannian metric* to obtain a *Riemannian manifold.* A Riemannian metric $\langle\cdot,\cdot\rangle_x$ is an inner product on the tangent space $T_x\mathcal{M}$. This metric induces a norm

$$\|\eta\|_x \stackrel{\text{def}}{=} \sqrt{\langle\eta,\eta\rangle_x},$$

on $T_x\mathcal{M}$. The gradient is derived from the previously defined notion of directional derivative. Given a smooth function $f : \mathcal{M} \to \mathbf{R}$, the *gradient* of f at $x \in \mathcal{M}$ is the element $\mathrm{grad} f(x) \in T_x\mathcal{M}$ that satisfies

$$\langle \mathrm{grad} f(x), \eta\rangle_x = \mathrm{D}f(x)[\eta], \quad \forall \eta \in T_x\mathcal{M}. \tag{4.18}$$

The gradient points in the direction of maximum ascent of objective function,

$$\frac{\mathrm{grad} f(x)}{\|\mathrm{grad} f(x)\|_x} = \operatorname*{arg\,max}_{\substack{\eta\in T_x\mathcal{M}\\ \|\eta\|_x=1}} \mathrm{D}f(x)[\eta].$$

In case of the Euclidean space $\mathbf{R}^{n\times p}$, the definition (4.18) corresponds to the classical element-wise construction

$$[\mathrm{grad} f(x)]_{ij} = \left.\frac{\partial f}{\partial x_{ij}}\right|_x, \tag{4.19}$$

at the point $x \in \mathbf{R}^{n\times p}$ and for the metric $\langle \eta, \zeta\rangle_x = \mathrm{Tr}(\eta^T\zeta)$.

In the case of a manifold $\mathcal{M}$ embedded in $\mathbf{R}^{n\times p}$, it should first be noted that, at a point $x \in \mathcal{M}$, the Euclidean space $\mathbf{R}^{n\times p}$ is uniquely decomposed into the two subspaces,

$$\mathbf{R}^{n\times p} = T_x\mathcal{M} \oplus N_x\mathcal{M},$$

where the *normal space* $N_x\mathcal{M}$ is the set of elements in $\mathbf{R}^{n\times p}$ that are orthogonal to all the elements of $T_x\mathcal{M}$ according to the metric $\langle\cdot,\cdot\rangle_x$. Any element $\eta \in \mathbf{R}^{n\times p}$ is so decomposed into the sum

$$\eta = \mathrm{P}_x\eta + \mathrm{P}_x^{\perp}\eta,$$

where $\mathrm{P}_x\eta \in T_x\mathcal{M}$ and $\mathrm{P}_x^{\perp}\eta \in N_x\mathcal{M}$. Let $\nabla f(x)$ denote the *Euclidean gradient* of f at x, i.e., the gradient of f computed in the embedding space according to (4.19). The gradient of f on the manifold $\mathcal{M}$ corresponds then to

$$\mathrm{grad} f(x) = \mathrm{P}_x\nabla f(x), \tag{4.20}$$

which satisfies the definition (4.18) since

$$\langle \mathrm{grad} f(x), \eta\rangle_x = \langle \nabla f(x), \eta\rangle_x = \mathrm{D}f(x)[\eta],$$

for any $\eta \in T_x\mathcal{M}$.

The following two examples provide detailed insight into both projections P_x and $\mathrm{P}_x^{\perp}$ in the specific cases of the sphere and the Stiefel manifolds.

Example 4.3.3 (Projections for the sphere) *Consider the metric* $\langle \eta, \zeta\rangle_x \overset{def}{=} \eta^T\zeta$ *on the tangent space* $T_x\mathcal{S}^{n-1} = \{\eta \in \mathbf{R}^n \mid \eta^T x = 0\}$. *The normal space corresponds then to* $N_x\mathcal{S}^{n-1} = \{\alpha x \mid \alpha \in \mathbf{R}\}$. *Given an element* $\eta \in \mathbf{R}^n$, *projections onto the tangent space and the normal space are respectively given by*

$$\mathrm{P}_x\eta = (I_n - xx^T)\eta \quad \textit{and} \quad \mathrm{P}_x^{\perp}\eta = xx^T\eta.$$

Example 4.3.4 (Projections for the Stiefel manifold) *Consider the metric* $\langle \eta, \zeta\rangle_x \overset{def}{=} \mathrm{Tr}(\eta^T\zeta)$ *on the tangent space*

$$T_x\mathrm{St}(p,n) = \{x\Omega + x_{\perp}K \mid \Omega = -\Omega^T, K \in \mathbf{R}^{(n-p)\times p}\}.$$

By considering the identity $\mathrm{Tr}(S\Omega) = 0$ *for any symmetric matrix* S *and skew-symmetric matrix* Ω, *the normal space corresponds to*

$$N_x\mathrm{St}(p,n) = \{xS \mid S = S^T\},$$

and the projections are given by

$$\mathrm{P}_x\eta = (I_n - xx^T)\eta + x\,\mathrm{skew}(x^T\eta) \quad \textit{and} \quad \mathrm{P}_x^{\perp}\eta = x\,\mathrm{sym}(x^T\eta),$$

where $\mathrm{sym}(M) \overset{def}{=} \frac{1}{2}(M + M^T)$ *and* $\mathrm{skew}(M) \overset{def}{=} \frac{1}{2}(M - M^T)$ *extract, respectively, the symmetric and the skew-symmetric part of the square matrix* M.

Finally, to complete the description of the steepest-descent method, the step size t in the iteration (4.15) can be chosen by classical backtracking methods. It is customarily chosen to satisfy the Armijo condition

$$f(R_x(t\eta)) \leq f(x) + \sigma\langle \mathrm{grad} f(x), t\eta\rangle_x,$$

where $x \in \mathcal{M}$ is the current iterate, $\sigma \in (0,1)$ and $R_x(\eta)$ is a retraction. Accumulation points of the steepest-descent method are proved to be stationary points of the objective function (i.e., a point for which $\mathrm{grad} f(x) \in T_x\mathcal{M}$ is zero), provided that the step size t satisfies the Armijo condition at each iteration. Furthermore, since the iteration performs a descent mapping, the local minimizers are the only stable accumulation points of the algorithm.

4.3.3 Second-order differential-geometric methods

Many optimization methods exploit both first- and second-order derivative information on the objective function. Such methods usually converge faster than the simple steepest-gradient method, i.e., they often converge superlinearly whereas the steepest-gradient method converges only linearly.

On a vector space, second-order derivative information on a smooth function is provided by the *Hessian*. The Hessian at a point $x \in \mathbf{R}^n$ of a function $f : \mathbf{R}^n \to \mathbf{R} : x \mapsto f(x)$ is a matrix $H(x) \in \mathbf{R}^{n\times n}$ such that

$$[H(x)]_{ij} \stackrel{\text{def}}{=} \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_x .$$

Most methods do not require an explicit evaluation of the Hessian but only its application on a particular direction $\eta \in \mathbf{R}^n$, i.e., the product $H(x)\eta$. This corresponds to the derivative of the gradient in the direction η. The notion of "directional derivative of a vector field" is generalized to manifolds thanks to the concept of *Riemannian connection*. The discussion below provides some intuition about this concept as well as practical guidelines for its numerical evaluation. We refer to Absil et al. [AMS08] for a rigorous definition.

Let ζ be a vector field on a manifold $\mathcal{M}$, i.e., a map that assigns to each point $x \in \mathcal{M}$ a tangent vector $\zeta_x \in T_x\mathcal{M}$. The gradient of a function is a typical example of vector field. Let $\nabla_\eta \zeta_x \in T_x\mathcal{M}$ denote the directional derivative of the vector field ζ at $x \in \mathcal{M}$ in a direction $\eta \in T_x\mathcal{M}$. Consider the following examples.

Example 4.3.5 (Riemannian connection on the sphere) *Let $\bar\zeta$ be a vector field on $\mathbf{R}^n$. Let ζ be the associated vector field on $\mathcal{S}^{n-1}$ that assigns to any point $x \in \mathcal{S}^{n-1}$ the tangent vector*

$$\zeta_x \stackrel{\text{def}}{=} \mathrm{P}_x\bar\zeta_x = (I_n - xx^T)\bar\zeta_x.$$

The directional derivative of $\bar\zeta$ at x in a direction $\eta \in T_x\mathcal{S}^{n-1}$ is

$$\mathrm{D}\bar\zeta_x[\eta] = \lim_{t\to 0}\frac{\bar\zeta_{x+t\eta} - \bar\zeta_x}{t}.$$

The "classical" directional derivative of ζ (i.e., computed in the Euclidean sense in $\mathbf{R}^n$) at x in the direction η is given by

$$\lim_{t\to 0}\frac{\zeta_{x+t\eta}-\zeta_x}{t} = (I_n - xx^T)\mathrm{D}\bar{\zeta}_x[\eta] - (x\eta^T + \eta x^T)\bar{\zeta}_x,$$

which is, in general, not an element of the tangent plane $T_x\mathcal{S}^{n-1}$. The Riemannian connection is simply the projection of this derivative onto $T_x\mathcal{S}^{n-1}$,

$$\begin{aligned}\nabla_\eta\zeta_x &\stackrel{def}{=} \mathrm{P}_x\left(\lim_{t\to 0}\frac{\zeta_{x+t\eta}-\zeta_x}{t}\right)\\ &= (I_n - xx^T)\mathrm{D}\bar{\zeta}_x[\eta] - \eta x^T\bar{\zeta}_x.\end{aligned}$$

Example 4.3.6 (Riemannian connection on the Stiefel manifold) *Let $\bar{\zeta}$ be a vector field on $\mathbf{R}^{n\times p}$. Let ζ be the associated vector field on $\mathrm{St}(p,n)$ that assigns to any point $x \in \mathrm{St}(p,n)$ the tangent vector*

$$\begin{aligned}\zeta_x &\stackrel{def}{=} \mathrm{P}_x\bar{\zeta}_x\\ &= (I_n - xx^T)\bar{\zeta}_x + x\,\mathrm{skew}(x^T\bar{\zeta}_x).\end{aligned}$$

The directional derivative of $\bar{\zeta}$ at x in a direction $\eta \in T_x\mathrm{St}(p,n)$ is

$$\mathrm{D}\bar{\zeta}_x[\eta] = \lim_{t\to 0}\frac{\bar{\zeta}_{x+t\eta}-\bar{\zeta}_x}{t}.$$

Again, the Euclidean directional derivative of ζ (i.e., computed in the embedding space $\mathbf{R}^{n\times p}$) at x in the direction η,

$$\lim_{t\to 0}\frac{\zeta_{x+t\eta}-\zeta_x}{t} = \mathrm{P}_x\mathrm{D}\bar{\zeta}_x[\eta] - \eta\,\mathrm{sym}(x^T\bar{\zeta}_x) - x\,\mathrm{sym}(\eta^T\bar{\zeta}_x),$$

is, in general, not an element of the tangent space. The Riemannian connection is the projection of this derivative onto $T_x\mathrm{St}(p,n)$,

$$\begin{aligned}\nabla_\eta\zeta_x &\stackrel{def}{=} \mathrm{P}_x\left(\lim_{t\to 0}\frac{\zeta_{x+t\eta}-\zeta_x}{t}\right)\\ &= \mathrm{P}_x\mathrm{D}\bar{\zeta}_x[\eta] - \mathrm{P}_x\eta\,\mathrm{sym}(x^T\bar{\zeta}_x).\end{aligned}$$

Given a Riemannian connection, the Hessian of a function f at a point $x \in M$ is naturally defined by

$$\mathrm{Hess}f(x)[\eta] = \nabla_\eta\mathrm{grad}f(x),$$

for any tangent vector $\eta \in T_x\mathcal{M}$ and represents the derivative of the gradient in the direction η. This definition enables to generalize standard second-order optimization methods to manifolds. For instance, *Newton's method* consists in solving the equation

$$\mathrm{Hess}f(x)[\eta] = -\mathrm{grad}f(x)$$

with respect to η at each iterate x. The update is then performed according to $x_+ = R_x(\eta)$. The *trust-region* method minimizes at each iteration a quadratic model of the objective on a trust region of radius Δ,

$$\begin{aligned} &\max_{\eta \in T_x\mathcal{M}} f(x) + \langle \mathrm{grad} f(x), \eta \rangle_x + \frac{1}{2} \langle \mathrm{Hess} f(x)[\eta], \eta \rangle_x \\ &\text{subject to } \langle \eta, \eta \rangle_x \leq \Delta^2. \end{aligned} \tag{4.21}$$

Again, the next iterate is computed according to $x_+ = R_x(\eta)$. Since problem (4.21) is defined on a vector space, classical optimization strategies can be directly used. The *conjugate gradient* method has also been generalized to manifolds.

All these optimization methods are supported by a convergence theory whose results are similar to the ones related to classical unconstrained optimization. In particular, trust-region methods on manifolds converge globally to stationary points of the objective function if the inner iteration (i.e., the iteration used to solve (4.21) at a given $x \in \mathcal{M}$.) produces a model decrease that is better than a fixed fraction of the *Cauchy decrease* [ABG07]. Since the iteration is moreover a descent method, convergence to saddle points or local maximizers is not observed in practice. For appropriate choices of the inner iteration stopping criterion, trust-region methods converge locally superlinearly towards the non-degenerate local minimizers of the objective function.

We refer to Absil et al. [AMS08] for the complete description and convergence analysis of these differential-geometric optimization methods. The trust-region algorithm is also detailed in [ABG07].

4.4 Optimization methods on the orthogonal group

In the particular case $p = n$, the Stiefel manifold inherits the properties of a *Lie group*.

Definition 4.4.1 (Group) *A* group *is a set G endowed with a product called the* group operation *such that*

1. *Given $x, y \in G$, $x \cdot y$ is also in G;*

2. *Given $x, y, z \in G$, $(x \cdot y) \cdot z = x \cdot (y \cdot z)$;*

3. *There is an identity element $\mathbf{1}$, such that $x \cdot \mathbf{1} = \mathbf{1} \cdot x = x$ for any $x \in G$;*

4. *For each element $x \in G$, there is any inverse $x^{-1} \in G$ such that $x \cdot x^{-1} = x^{-1} \cdot x = \mathbf{1}$.*

Definition 4.4.2 (Lie group) *A* Lie group *is a differentiable manifold $\mathcal{M}$ and a differentiable group operation that satisfy the four group properties.*

The orthogonal group $\mathcal{O}(n) = \{x \in \mathbf{R}^{n \times n} \mid x^T x = I_n\}$ is a Lie group for the matrix multiplication. In fact, the product of two orthogonal matrices is an orthogonal matrix. The matrix multiplication is associative. The identity matrix is the identity element. The inverse of an orthogonal matrix, finally, is an orthogonal matrix. In view of these geometrical properties, further methods can be considered for solving the optimization problem

$$\min_{x \in \mathcal{O}(n)} f(x), \tag{4.22}$$

with a smooth function $f : \mathcal{O}(n) \to \mathbf{R}$. The first group property in Definition 4.4.1 suggests the iterations

$$x_+ = xy \quad \text{or} \quad x_+ = yx, \tag{4.23}$$

where the orthogonal update $y \in \mathcal{O}(n)$ is constructed such that the objective decreases from x to x_+. Thanks to the Lie group structure of $\mathcal{O}(n)$, the orthonormality constraint can be maintained at each iterate in a very natural manner.

Note that the iteration $x_+ = yx$ with $y \in \mathcal{O}(n)$ could also be used for optimization on the Stiefel manifold, i.e., $x \in \mathrm{St}(p, n)$ with $p < n$. Left matrix multiplication of an element of $\mathrm{St}(p, n)$ with an element of $\mathcal{O}(n)$ remains in $\mathrm{St}(p, n)$. Formally, the orthogonal group *acts transitively* on the Stiefel manifold.

4.4.1 Jacobi rotations

A common choice for the update y in (4.23) is provided by the *Jacobi rotation* [GVL89, Com94, Car99]. A Jacobi rotation $y^{k,l}(t) \in \mathcal{O}(n)$ is defined element-wise by

$$[y^{k,l}(t)]_{ij} \stackrel{\text{def}}{=} \begin{cases} \cos(t) & \text{if } i = k \text{ and } j = k, \\ \sin(t) & \text{if } i = k \text{ and } j = l, \\ -\sin(t) & \text{if } i = l \text{ and } j = k, \\ \cos(t) & \text{if } i = l \text{ and } j = l, \\ 1 & \text{if } i = j \text{ and } i \neq k, l, \\ 0 & \text{otherwise,} \end{cases} \tag{4.24}$$

and performs a planar rotation of angle t in the subspace of $\mathbf{R}^n$ spanned by the two canonical basis vectors e_k and e_l.

A Jacobi algorithm consists in selecting two directions $\{e_k, e_l\}$ at each iteration and to compute the rotation t that maximizes the objective function on the subspace spanned by these vectors. At each iteration, a "line-search" for the best angle t needs to be performed. The Jacobi algorithm was initially proposed to diagonalize a symmetric matrix and to compute its eigenvalue decomposition, but it can be extended to any optimization problem on the orthogonal group (see e.g., [HH97]). Overall, it is a very efficient algorithm once the "line-search" inner problems have closed-form solutions. This is definitely the case for the eigenvalue decomposition [GVL89]. Some ICA contrasts are also endowed with this interesting property, e.g., the ICA contrast (4.11) to approximately joint diagonalize a set of matrices [CS93]. An alternative, otherwise, is to perform an exhaustive search on the interval $t \in [0, 2\pi]$. This approach is used by the RADICAL algorithm [LF03] to maximize an order-statistics based estimator of the mutual information. Interestingly, exhaustive search can be restrained to the interval $t \in [0, \frac{\pi}{2}]$ for ICA contrasts, because the sign and the order of the components is insignificant.

A Jacobi algorithm usually sweeps until convergence through all possible pairs of basis vectors in a sequential and ordered fashion. Such an algorithm is known as the *sequential cyclic* Jacobi algorithm. The Jacobi algorithm arouses also much interest because it is appropriate for parallel computations (see, e.g., [GVL89, EP90] and references therein).

4.4.2 Geodesic flows

For any compact and connected Lie group $\mathcal{M}$, there exists an *exponential map*

$$\mathrm{Exp} : T_{\mathbf{1}}\mathcal{M} \to \mathcal{M},$$

from the tangent space at the identity $T_{\mathbf{1}}\mathcal{M}$ to the manifold $\mathcal{M}$ that is smooth and *surjective*, i.e., for any point in $x \in \mathcal{M}$, there exists an element $\eta \in T_{\mathbf{1}}\mathcal{M}$ such that $\mathrm{Exp}(\eta) = x$. This exponential map enables to lift a problem defined on a Lie group into a problem defined on a vector space, where traditional methods for unconstrained optimization could be used.

The orthogonal group is compact but *not* connected. It consists of two connected components: the matrices with positive determinant (dextrorsum orthonormal frames) and the matrices with negative determinant (senestrorsum orthonormal frames). A compact and connected Lie group is obtained by considering the matrices with positive determinant only. This defines the *special orthogonal group* $\mathcal{SO}(n)$. In the case of ICA, since the sign of the components is unimportant, the search space can be restricted to $\mathcal{SO}(n)$ instead of $\mathcal{O}(n)$ without loss of generality. As discussed previously in the Example 4.3.2, the tangent space at the identity to $x \in \mathcal{SO}(n)$ is given by the set of skew-symmetric matrices,

$$\mathfrak{so}(n) \stackrel{\text{def}}{=} T_{\mathbf{1}}\mathcal{SO}(n) = \{\Omega \mid \Omega^T = -\Omega\}.$$

The vector space $\mathfrak{so}(n)$ satisfies the properties to be a *Lie algebra* and is therefore called the Lie algebra of $\mathcal{SO}(n)$.[5] An exponential map for $\mathcal{SO}(n)$ is provided by the matrix exponential

$$\mathrm{Exp}(\eta) = e^{\eta} \stackrel{\text{def}}{=} I_n + \eta + \frac{\eta^2}{2!} + \ldots + \frac{\eta^k}{k!} + \ldots,$$

which, given a skew-symmetric matrix η, provides an orthogonal matrix. The optimization problem (4.22) is thus naturally lifted into

$$\min_{\eta \in \mathfrak{so}(n)} \bar{f}(\eta). \tag{4.25}$$

where $\bar{f} : \mathfrak{so}(n) \to \mathbf{R} : \bar{f}(\eta) = f(\mathrm{Exp}(\eta))$.

Classical optimization methods can be used for solving (4.25). For illustration, consider the simplest case of the steepest-descent method, i.e., the iteration is

$$\eta_+ = \eta - t\,\mathrm{grad}\bar{f}(\eta) \tag{4.26}$$

[5]A Lie algebra $\mathfrak{g}$ is a vector space with a *Lie bracket* $[\cdot,\cdot] : [\mathfrak{g},\mathfrak{g}] \to \mathfrak{g}$ that satisfies

$$\begin{aligned}[x,x] &= 0,\\ [x+y,z] &= [x,z]+[y,z],\\ [x,[y,z]]+[y,[z,x]]+[z,[x,y]] &= 0,\end{aligned}$$

for any elements $x, y, z \in \mathfrak{g}$. The vector space $\mathfrak{so}(n)$ is a Lie algebra for the Lie bracket $[x,y] = xy - yx$.

with the step size $t \geq 0$. To any iterate η corresponds an orthogonal matrix $x = \mathrm{Exp}(\eta)$. The gradient $\mathrm{grad}\bar{f}(\eta)$ is the vector of $\mathfrak{so}(n)$ that satisfies

$$\langle \mathrm{grad}\bar{f}(\eta), \zeta \rangle = \mathrm{D}\bar{f}(\eta)[\zeta],$$

for any direction $\zeta \in \mathfrak{so}(n)$. If one endows the vector space $\mathfrak{so}(n)$ with the metric $\langle \eta_1, \eta_2 \rangle \stackrel{\mathrm{def}}{=} \mathrm{Tr}(\eta_1^T \eta_2)$, the following holds

$$\begin{aligned} \mathrm{D}\bar{f}(\eta)[\zeta] &= \lim_{t \to 0} \frac{\bar{f}(\eta + t\zeta) - \bar{f}(\eta)}{t} \\ &= \lim_{t \to 0} \frac{f(\mathrm{Exp}(\eta + t\zeta)) - f(\mathrm{Exp}(\eta))}{t} \\ &= \mathrm{Tr}(\nabla f(\mathrm{Exp}(\eta))^T \, \mathrm{DExp}(\eta)[\zeta]), \end{aligned}$$

where $\nabla f(x)$ is the Euclidean gradient of f at x and

$$\mathrm{DExp}(\eta)[\zeta] = \zeta + \frac{1}{2!}(\eta\zeta + \zeta\eta) + \frac{1}{3!}(\eta\eta\zeta + \eta\zeta\eta + \zeta\eta\eta) + \dots.$$

This series is however difficult to evaluate, excepted at the point $\eta = 0$. The usual trick consists in *shifting* the computation of the gradient to the origin of $\mathfrak{so}(n)$. Let us consider the mapping $\mathfrak{so}(n) \to \mathcal{SO}(n) : \eta \mapsto x\mathrm{Exp}(\eta)$ that maps the origin of $\mathfrak{so}(n)$ onto the current iterate $x \in \mathcal{SO}(n)$. The steepest-descent iteration (4.26) is then rewritten in the form

$$\begin{aligned} \eta_+ &= -t\mathrm{grad}\bar{f}_x(0) \\ x_+ &= x\mathrm{Exp}(\eta_+), \end{aligned} \tag{4.27}$$

with the function $\bar{f}_x : \mathfrak{so}(n) \to \mathbf{R} : \bar{f}_x(\eta) = f(x\mathrm{Exp}(\eta))$ and where the gradient is evaluated at the origin. The following closed-form expression is now available,

$$\mathrm{grad}\bar{f}_x(0) = \mathrm{skew}(x^T \nabla f(x)). \tag{4.28}$$

The exponential map between the Lie algebra and the Lie group provides very elegant methods for optimization. Specifically, straight lines of $\mathfrak{so}(n)$ are mapped onto *geodesics* of $\mathcal{SO}(n)$, i.e., curves of minimum distance between two points. The iteration (4.27) performs therefore a *geodesic search* on $\mathcal{SO}(n)$. The numerical evaluation of the matrix exponential is, however, a usually very costly operation. Interestingly, if the skew-symmetric η has only one non-zero element at position (k, l), i.e.,

$$\eta(k, l) = 1 \quad \text{and} \quad \eta(k, l) = -1 \quad \text{with} \quad k < l, \tag{4.29}$$

the matrix exponential $e^{t\eta}$ is provided by the Jacobi rotation (4.24). The Jacobi algorithm, described in Section 4.4.1, performs hence a search along geodesics that are computationally cheap, but which do not point in directions of steepest-descent.

As previously mentioned, the Jacobi algorithm usually sweeps through the matrices η satisfying (4.29) in an ordered fashion. An alternative would be to choose at each iteration the matrix η that is the closest to the gradient (4.28). This would achieve a trade-off between finding a steepest-descent direction and minimizing the computational expense. Let us finally mention that such an approach is also conceivable for rotations on three-dimensional subspaces, for which a closed-form expression of the matrix exponential is also available.[6]

4.5 Algorithms for independent component analysis

Algorithms for ICA are obtained by combining a contrast function with an optimization method. Applying the optimization methods of Sections 4.3 and 4.4 on the contrasts of Section 4.2 recovers some well-known algorithms, that we now briefly review.

The following list of algorithms for ICA is not exhaustive. This research topic is very active and the related literature is vast. We only review a couple of algorithms that match within the geometric framework discussed in this chapter.

The Jacobi algorithm is widely used for optimizing ICA contrasts. In his seminal paper on ICA, Comon proposes a Jacobi algorithm to optimize the estimator of the mutual information based on cumulants [Com94]. The JADE algorithm [CS93, Car99] uses Jacobi rotations to diagonalize as well as possible a set of cumulant matrices. These two methods are provided with closed-form expressions for the best rotation angle t at each iteration. RADICAL [LF03] is another Jacobi algorithm that maximizes an order-statistics based estimator of the mutual information. But in contrast to the previous algorithms, the geodesic search is performed in an exhaustive manner. As a further example, SOBI [BAMCM97] is a Jacobi algorithm that approximately joint diagonalizes covariance matrices extracted from time-series data (i.e., data for which the order of the samples is meaningful).

A gradient optimization on the orthogonal group of the contrast used by the RADICAL algorithm is discussed in [JTAS07, JAS07]. The KernelICA algorithm [BJ03] maximizes a kernel approximation of the $\mathcal{F}$-correlation (4.13) by gradient-descent on the orthogonal group. The main contributions on Lie group methods for ICA are due to Nishimori [Nis99] and Plumbley [Plu03, Plu05].

FastICA [HH00], probably the most popular algorithm for ICA, maximizes measures of non-gaussianity on the sphere. A rather heuristic optimization method is used in the original formulation, which has been later improved by exploiting the manifold structure of the problem [WRZ$^+$06, SKH08].

For completeness, let us mention that the algorithms for ICA discussed in this thesis post-process the principal components by computing a suitable rotation, and are therefore commonly qualified as *orthogonal algorithms* for ICA. Some further approaches do, however, not compute principal components and manage to identify from scratch a non-orthogonal

[6] Consider a skew-symmetric matrix η that is zero, excepted at the 3-by-3 skew-symmetric submatrix ω formed by the intersection of the rows and columns i, j, and k of η. Assume that $\|\omega\|_2 = 1$, i.e., the largest singular value of ω is one. The matrix exponential $e^{t\eta}$ is an identity matrix, excepted at the submatrix formed by the intersection of the rows and columns i, j, and k that equals

$$I_3 + \omega \sin(t) + \omega^2(1 - \cos(t)).$$

This last equation is known as the Rodriguez formula [MSZ94]. The orthogonal matrix $e^{t\eta}$ corresponds to a rotation on the three-dimensional subspace spanned by the canonical basis vectors e_i, e_j and e_k.

p-dimensional basis Z of $\mathbf{R}^n$ in which the data is represented at best. Optimization is then performed on the noncompact Stiefel manifold $\mathbf{R}_*^{n\times p}$, i.e., the set of n-by-p full-rank matrices, or the *general linear group* GL_n in the square case $p = n$. The well-known Infomax algorithm belongs to this category of methods [BS95, LGS99]. A whole bunch of non-orthogonal algorithms for ICA are also based on the joint approximate diagonalization of a set of matrices (see, e.g., [Pha01, AG06, Afs06, WLZ07] and references therein).

4.6 Numerical experiments

In this section, some of the discussed optimization methods are compared on the minimization of a same contrast, e.g., the objective function

$$f : \mathcal{O}(p) \to \mathbf{R} : x \mapsto f(x) = \sum_i \|\mathrm{Off}(x^T C_i x)\|_F^2. \tag{4.30}$$

whose minimization performs the joint approximate diagonalization of the matrices C_i.

Considered are manifold-based optimization methods, Jacobi algorithms as well as approaches based on the exponential mapping. A couple of objects need to be specified to use these algorithms. First, the Euclidean gradient of the function f, used in the equations (4.20) and (4.28), is given by

$$\nabla f(x) = 4 \sum_i Cx\,\mathrm{Off}(x^T C_i x).$$

Similarly, the Euclidean directional derivative of this gradient in a direction η, required to evaluate the Riemannian connection (see Example 4.3.6), is given by

$$\mathrm{D}\nabla f(x)[\eta] = 4 \sum_i C\eta\,\mathrm{Off}(x^T C_i x) + C\eta\,\mathrm{Off}(\eta^T C_i x) + C\eta\,\mathrm{Off}(x^T C_i \eta).$$

The retraction used by the manifold-based methods is done by QR factorization (see Example 4.3.2). Concerning the Jacobi algorithm, we refer to Cardoso and Souloumiac [CS93] for a closed-from expression of the best angle t that optimizes the contrast at each iteration.

In our experiments, the matrices C_i are cumulant matrices drawn from a data matrix A that is artificially constructed as the product $A = SH$, where $S \in \mathbf{R}^{m\times p}$ contains m samples of p statistically independent random variables (the *sources*) and $H \in \mathbf{R}^{p\times n}$ is a mixing matrix. Each column of S contains the pixel values of a black-and-white image of dimension 256-by-512. Ten images are considered, which are expected to have independent pixels distributions.[7] The matrix $H \in \mathbf{R}^{10\times 10}$ is chosen randomly according to a Gaussian distribution.

Numerical results obtained with MATLAB are presented in Figure 4.1. The two methods "Gradient" and "Trust-region" are adaptations of the classical gradient-descent and trust-region methods on manifolds. For the trust-region approach, the parameter θ in equation (10) of [ABG07] is set to one to ensure a quadratic convergence. "Jacobi (cyclic)" denotes the sequential cyclic Jacobi algorithm which computes Jacobi rotations in an ordered fashion,

[7] These images are available at the URL http://www.cis.hut.fi/projects/ica/data/images/

whereas "Jacobi (gradient)" is the algorithm proposed at the end of Section 4.4.2, which computes at each iteration the Jacobi rotation that is the closest to the gradient. "Exp. mapping", finally, performs a gradient-descent along geodesics of the orthogonal group according to (4.27). In both algorithms "Gradient" and "Exp. mapping", the step size t is computed by backtracking search. Specifically, we use the *Armijo* step size (see Definition 4.2.2 in [AMS08] with parameters $\bar{\alpha} = 2$, $\beta = 0.5$ and $\sigma = 0.01$).

Figure 4.1 shows that the trust-region approach achieves quadratic convergence, whereas the other methods converge only linearly. Although the cyclic Jacobi algorithm is the slowest to converge in terms of number of iterations, it is the fastest in terms of computational time for low accuracies. Computing a Jacobi rotation is in fact very cheap. When comparing the two gradient methods "Gradient" and "Exp. mapping", it turns out that these methods have identical rates of convergence. The "Gradient" algorithm is however somewhat faster in time, probably because a QR factorization is numerically cheaper to evaluate than a matrix exponential. Finally, "Jacobi (gradient)" has a better convergence than "Jacobi (cyclic)". In a close neighborhood of the solution, the convergence of "Jacobi (gradient)" is even comparable to the one of the steepest-descent algorithms. Nevertheless, because evaluating a gradient at each iteration is somewhat costly, the common cyclic Jacobi algorithm is still the fastest in time.

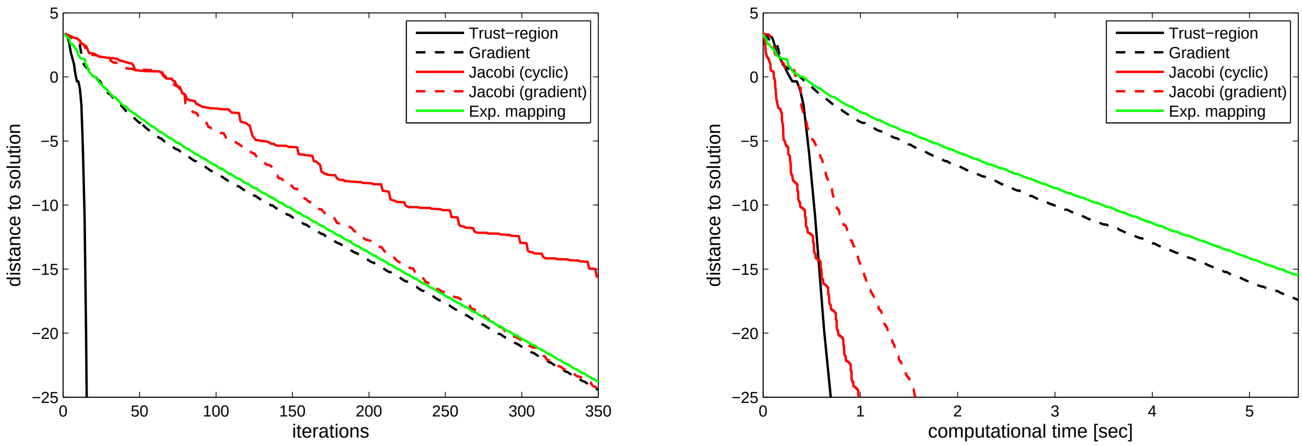

Figure 4.1: Convergence of several optimization methods for minimizing the function (4.30). All methods are initialized from the same initial point and converge towards the same local minimizer x^*. The vertical axis is the distance $\log\ (f(x)\ -\ f(x^*))$.

4.7 Analysis of gene expression data

In this section, we use PCA and ICA to analyze the four breast cancer cohorts described in Table 3.1. PCA has been intensively applied in the context of gene expression data (see, e.g., [ABB00, HMM$^+$00, ABB03]). Several studies have also shown the value of ICA, notably Liebermeister [Lie02], who was the first to apply ICA to gene expression data. Important results on some bacterial and human databases are also detailed in [MMSM02, LB03, SHK$^+$03].

PCA is performed by computing the singular value decomposition of the data and four different implementations of the ICA paradigm are considered: FastICA [HH00], JADE [Car99], KernelICA [BJ03] and RADICAL [LF03]. Ten components are inferred for each data set and method, i.e., $p = 10$. Importantly, in the context of gene expression analysis, statistical independence is typically imposed in the space of the genes rather than in the space of the experiments, i.e., the columns of Z in the factorization (2.2) are seen as samples of statistically independent random variables,. This provides activation patterns over genes that are as independent as possible. Going back to Section 2.1.2, ICA then amounts to computing a rotation matrix Q that maximizes the statistical independence of the columns of $Z = V\Sigma Q\bar{\Sigma}^{-1}$. Statistical quantities are so estimated from a much larger number of measurements.

This study has been published in PLoS Computational Biology [TJA$^+$07]. The most significant results are summarized below.

Pathway enrichment analysis

Because of the statistical independence assumption inherent in ICA, one expects the components to map more closely to known pathways than an alternative linear decomposition method, like PCA, that does not use the statistical independence criterion. In Figure 4.2 (plots (A) and (B)), the pathway enrichment index (PEI), that we have defined in Section 3.3.1, is shown for two lists of pathways, for each of the methods and the four breast cancer sets. Figure 4.2 shows that across the four cohorts the PEI is higher for ICA algorithms when compared with PCA. It is also noteworthy that when comparing the various ICA algorithms with each other we do not observe any appreciable difference in their respective PEI, although these algorithms significantly differ in the contrast and the optimization method they use.

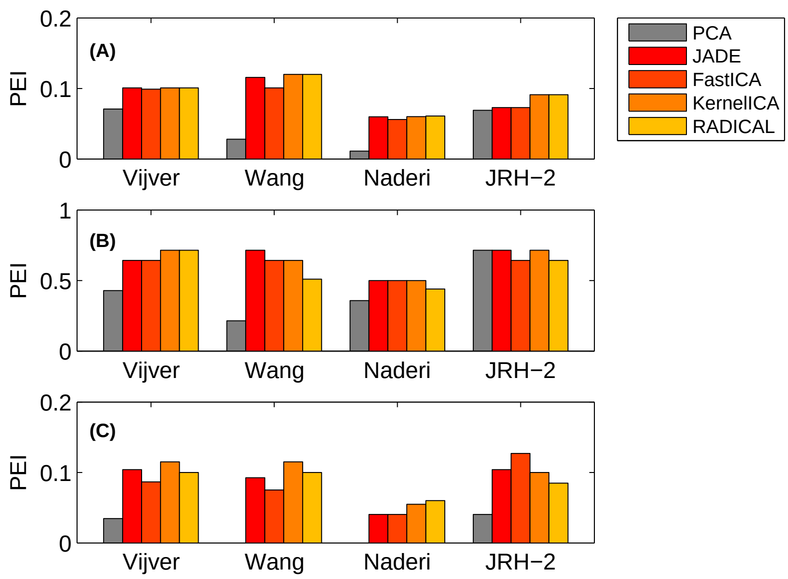

Figure 4.2: Pathway enrichment index (PEI) based on 536 biological pathways (A), 14 cancer-signalling and oncogenic pathways (B) and 173 motif-regulatory gene sets (C).

As a further validation that ICA outperforms PCA, let us investigate the relation of the derived components with regulatory modules. Figure 4.2 in plot (C) shows that PCA performs worst out of all algorithms. In two cohorts ("Wang" and "Naderi"), none of the PCA components is associated with any of the 173 distinct regulatory modules. In contrast, the components derived by ICA algorithms are consistently associated with regulatory modules. These results show that ICA provides a more biologically meaningful decomposition of breast cancer expression data than PCA.

Figure 4.3 lists the pathways that are most frequently and consistently differentially activated by the four ICA algorithms across all four breast cancer cohorts. Among these pathways are those related to estrogen signalling as well as to other important breast cancer signalling pathways such as the EGFR1 and TGF-β pathways. We also find cell-adhesion, immune-response, cell-cycle, and metabolic pathways to be commonly differentially activated across the cohorts. While breast cancer studies have found study-specific gene clusters associated with cell-cycle, estrogen-response, cell-adhesion, and immune-response functions, our results show that expression variation across breast tumors can be understood in terms of single pathways (i.e., a fixed common set of genes for all studies) that relate to these biological functions.

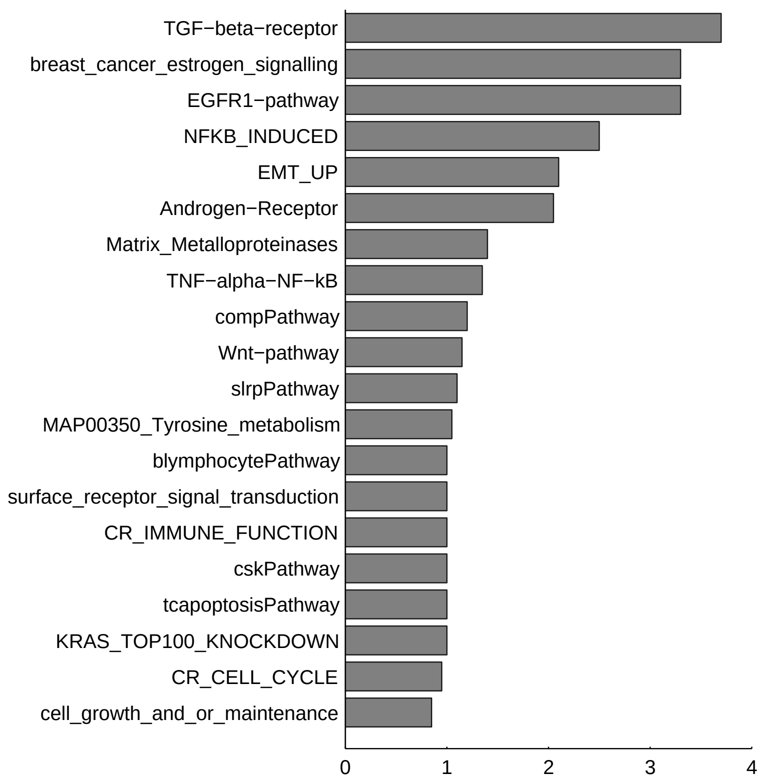

Figure 4.3: Twenty of the most frequently mapped pathways by ICA. The scores give the average number of ICA components in which the pathway is mapped.

Correlation with clinical data

Statistical testing between inferred components and clinical data reveals a complex pattern of significant associations with several components differentiating breast tumors according to estrogen receptor (ER) status and histological grade (Figure 4.4). It is notable that in all cohorts ICA components associating with clinical outcome are also found, while PCA generally does not. Another feature is the fact that more and stronger phenotype associations are uncovered by using ICA as compared with PCA.

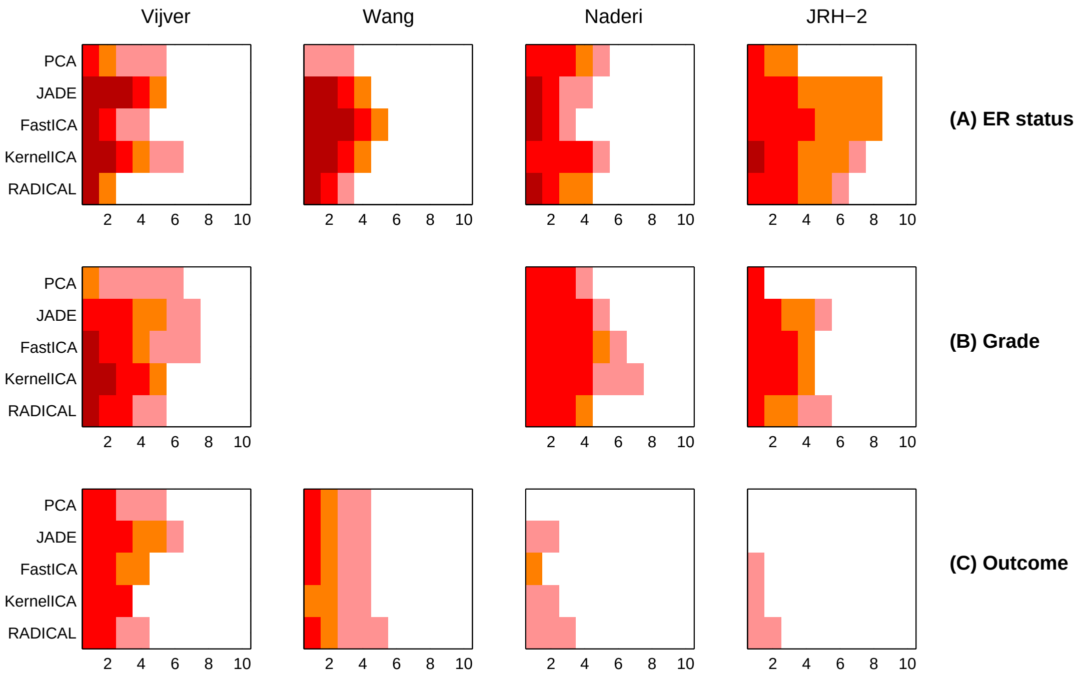

Figure 4.4: Heatmaps of association between components and breast cancer phenotypes. For each data set and each method, ten p-values are represented that assess the strength of association between each component and a phenotype. Color-code: p-value $< 10^{-10}$ (dark red), p-value < 0.001 (red), p-value < 0.01 (orange), p-value < 0.05 (pink) and p-value > 0.05 (white). For Wang's cohort, grade information is unavailable

Since we characterize each component in terms of the differential activation pattern of cancer-related pathways and regulatory modules, for those components associated with a phenotype we are able to link the corresponding pathways and regulatory modules with the phenotype. This leads to several well-known but also novel observations. Let us briefly review a few of them.

First, as expected, ICA components that are strongly associated with ER status are frequently mapped to the estrogen signalling pathway. Second, ICA components that map to the CR (cancer related) cell-cycle pathway [BCC+03] are frequently associated with either grade or outcome. The association between cell-cycle genes and grade or outcome is well-known [SNM+03, SWL+06, TNBM+06]. Third, we observe that pathways relating to immune response functions and the classical complement pathway are frequently correlated with ER status. For example, we find in each of the four major breast cancer cohorts an ICA component that maps to the CR immune response pathway [BCC+03], and which is consistently overactivated in ER- relative to ER+ tumors. Fourth, in all studies where grade information is available, an ICA component mapping to epithelial-mesenchymal transition (EMT) signalling pathway [JGT+03] is found to be associated with histological grade. Specifically, ICA reveals a component driving upregulation of genes involved in EMT in poorly differentiated tumors relative to low-grade tumors across the three studies where grade information is available. The latter associations linking immune response and EMT pathways with ER status and histological grade, respectively, are novel.

The parallel analysis for regulatory motifs and breast cancer phenotypes also provides direct links between the associated transcription factors and clinical variables. We refer to Teschendorff et al. [TJA+07] for the details.

Importantly, ICA facilitates the identification of many of the biological associations in comparison with PCA. Significant associations revealed by any one of the four ICA algorithms in all cohorts are, in fact, not consistently found by PCA. Some particular associations are even not identified by PCA in any cohort.

From statistical independence to sparsity

Statistical independence can be viewed as a "weak manner" to impose *sparsity* in the loading vectors, i.e., to enforce the columns of the matrix Z to contain many zeros. In fact, the ICA-inferred activation patterns over genes are as super-Gaussian as possible, which means that most of the entries are close to zero, excepted a few ones which might be large. It appears thus in this study that components involving a few genes only by still explaining a great part of the variability in the data explain more of the hidden biology than the regular principal components. This motivates the forthcoming investigations on *sparse principal component analysis*, which imposes sparsity in a "strong manner", i.e., by clearly setting the entries of the matrix Z to zero.

4.8 Summary

The present chapter is devoted to smooth optimization problems defined on the Stiefel manifold, i.e., the set of n-by-p matrices with orthonormal columns. The discussed applications concern principal component analysis (PCA) and especially independent component analysis (ICA). Typical objective functions for ICA are first reviewed, which relate to statistical estimators of independence between random variables. Various optimization methods are then discussed, that inherently exploits the rich geometry of the Stiefel manifold. Further optimization methods are also suggested for the limit case of the orthogonal group. Overall, these methods rest on classical tool for unconstrained optimization, but take advantage of the manifold structure to enforce orthonormality constraints. The numerical efficiency of these methods is compared on simple test problems. PCA and ICA are then applied on the four breast cancer cohorts. It turns out that ICA identifies important biological relationships, so far unseen by PCA.

The results of this chapter have been published in [TJA+07, JTAS07, JAS07, JTA+08].

Chapter 5

Generalized power method and its application to sparse PCA

In the present chapter, we focus on optimization problems of the form

$$\max_{x \in \mathcal{Q}} f(x), \tag{P$_2$}$$

where $\mathcal{Q}$ is a compact subset of a Euclidean space $\mathbf{E}$ and the function $f : \mathbf{E} \to \mathbf{R}$ is convex[1] but not necessarily smooth. The applications discussed in the sequel consider sets $\mathcal{Q}$ that are compact embedded manifolds, such as the sphere and the Stiefel manifold are. Due to the convexity of the objective function, we are able to propose a simple gradient-type scheme, the *generalized power method*, which appears to be well suited for problems of the class (P$_2$).

In the particular case when $\mathcal{Q}$ is the unit Euclidean ball in $\mathbf{R}^n$ and $f(x) = x^T C x$ for some symmetric positive definite matrix $C \in \mathbf{R}^{n \times n}$, this gradient scheme specializes to the *power method*, which aims at maximizing the Rayleigh quotient $R(x) = \frac{x^T C x}{x^T x}$ and thus at computing the largest eigenvalue and the corresponding eigenvector of C. By letting the matrix C be the Gram matrix $A^T A$, the (generalized) power method solves the problem

$$\max_{\substack{z \in \mathbf{R}^n \\ z^T z = 1}} z^T A^T A z, \tag{5.1}$$

which computes the first principal component of the column-centered data matrix $A \in \mathbf{R}^{m \times n}$ encoding m samples of n variables. Variations of PCA are obtained by adding a suitable penalty term to (5.1) that preserves the convexity of the objective function. In the specific case of penalties that enforce sparsity, the generalized power method performs *sparse principal component analysis* (sparse PCA).

This chapter is organized as follows. First, formulations for sparse PCA in the form of (P$_2$) are derived (Section 5.1). The generalized power method is then proposed and analyzed

[1] A function $f : \mathbf{E} \to \mathbf{R}$ is *convex* if for all $x_1, x_2 \in \mathbf{E}$ and θ with $0 \le \theta \le 1$, we have

$$f(\theta x_1 + (1-\theta) x_2) \le \theta f(x_1) + (1-\theta) f(x_2),$$

i.e., the chord between x_1 and x_2 is above or on the graph of the function f. We refer to Boyd and Vandenberghe [BV04] for further the properties of convex functions.

(Section 5.2). New algorithms for sparse PCA are subsequently obtained (Section 5.3). These algorithms are first evaluated on random test problems (Section 5.4) and then on the analysis of breast cancer gene expression data (Section 5.5).

5.1 Sparse principal component analysis

Principal and independent components are, in general, combinations of all the input variables. For instance, the vector $z \in \mathbf{R}^n$ that solves the PCA problem (5.1) is not expected to have many zero coefficients. In most applications, however, the original variables have concrete physical meaning and the extracted components appear especially interpretable if they are composed only from a small number of the original variables. In the case of gene expression data, one would like to find "simple" structures in the genome, expected to involve a few genes only, that explain a significant amount of the specific biological processes underlying the data. Sparse principal component analysis (sparse PCA) has the objective to explain *as much* variability in the data as possible, using components constructed from *as few* variables as possible. A reasonable trade-off between *statistical fidelity* and *interpretability* has thus to be found.

For about a decade, sparse PCA has been a topic of active research. Historically, the first suggested approaches were based on ad-hoc methods involving post-processing of the components obtained from classical PCA. For example, Jolliffe et al. [Jol95] consider using various rotation techniques to find sparse loading vectors in the subspace identified by PCA. Cadima et al. [CJ95] propose to simply set to zero the PCA loadings which are in absolute value smaller than some threshold constant.

In recent years, more involved approaches have been put forward, which consider the conflicting goals of explaining variability and achieving representation sparsity simultaneously. These methods usually cast the sparse PCA problem in the form of an optimization problem, aiming at maximizing explained variance penalized for the number of non-zero loadings. For instance, the SCoTLASS algorithm proposed by Jolliffe et al. [JTU03] aims at maximizing the Rayleigh quotient of the covariance matrix of the data under the ℓ_1-norm based Lasso penalty [Tib96]. Zou et al. [ZHT06] formulate sparse PCA as a regression-type optimization problem and impose the Lasso penalty on the regression coefficients. d'Aspremont et al. [AEJL07] in their DSPCA algorithm exploit convex optimization tools to solve a convex relaxation of the sparse PCA problem. Shen and Huang [SH08] adapt the singular value decomposition (SVD) to compute low-rank matrix approximations of the data matrix under various sparsity-inducing penalties. Greedy methods, which are typical for combinatorial problems, have been investigated by Moghaddam et al. [MWA06]. Finally, d'Aspremont et al. [ABE08] propose a greedy heuristic accompanied with a certificate of optimality.

Let us mention that sparse PCA has a wide range of applications. Besides the problem of component analysis discussed in this thesis, some examples are proposed by d'Aspremont et al. [ABE07]. *Compressed sensing*, for instance, is the problem of finding a vector $x \in \mathbf{R}^n$ from measurements $y = Ax + e$ where $A \in \mathbf{R}^{m\times n}$ is a known matrix and the unknown vector of error $e \in \mathbf{R}^m$ has a low cardinality [CT05]. This NP hard problem can be solved by linear programming provided that the *restricted isometry condition* is satisfied. One way to check this condition is to solve a sparse PCA problem [ABE07].

In the following sections, we consider several formulations of the sparse principal component analysis of a data matrix $A \in \mathbf{R}^{m \times n}$ as the maximization of a convex function on a compact set. These formulations aim at extracting either *one* dominant sparse principal component ("single-unit sparse PCA") or p components at once ("block sparse PCA"). While the basic formulations involve maximization of a *nonconvex* function on a space of dimension involving n, reformulations are derived that cast the problem into the form of maximization of a *convex* function on the unit Euclidean sphere in $\mathbf{R}^m$ (in the $p = 1$ case) or the Stiefel manifold in $\mathbf{R}^{m \times p}$ (in the $p > 1$ case). The advantage of the reformulation becomes apparent when trying to solve problems with many variables ($n \gg m$) since this manages to avoid searching a space of large dimension. By applying the general gradient scheme to the proposed sparse PCA reformulations of the form (P_2), algorithms are obtained with per-iteration computational cost $\mathcal{O}(nmp)$. These algorithms can also address the case of starting out with the sole knowledge of the covariance matrix between the n variables, i.e, without access to the data matrix. One simply needs to identify a factorization of this positive semidefinite matrix as the product $A^T A$, e.g., by eigenvalue decomposition or by Cholesky decomposition.

5.1.1 Sparse PCA as a maximization with spherical constraints

The "single-unit" formulations of sparse PCA come in two variants, depending on the type of penalty that is used to enforce sparsity: either ℓ_1 or ℓ_0 (cardinality).[2]

Single-unit sparse PCA via ℓ_1-penalty

Consider the optimization problem

$$\phi_{\ell_1}(\gamma) \stackrel{\text{def}}{=} \max_{z \in \mathcal{B}^n} \sqrt{z^T A^T A z} - \gamma \|z\|_1, \tag{5.2}$$

with sparsity-controlling parameter $\gamma \geq 0$, sample covariance matrix $A^T A$, and the unit Euclidean ball $\mathcal{B}^n = \{x \in \mathbf{R}^n \mid x^T x \leq 1\}$.

The solution $z^*(\gamma)$ of problem (5.2) in the case $\gamma = 0$ equals the dominant right singular vector of A and provides thus the first principal component of the data matrix A. The optimal value of the problem is given by

$$\phi_{\ell_1}(0) = (\lambda_1(A^T A))^{\frac{1}{2}} = \sigma_1(A),$$

where λ_1 and σ_1 denote the largest eigenvalue and the largest singular value, respectively. There is no reason to expect the vector $z^*(0)$ to be sparse. On the other hand, for large enough γ, one necessarily has $z^*(\gamma) = 0$, obtaining maximum sparsity. Indeed, since

$$\max_{z \neq 0} \frac{\|Az\|_2}{\|z\|_1} = \max_{z \neq 0} \frac{\|\sum_i z_i a_i\|_2}{\|z\|_1} \leq \max_{z \neq 0} \frac{\sum_i |z_i| \|a_i\|_2}{\sum_i |z_i|} = \max_i \|a_i\|_2 = \|a_{i^*}\|_2,$$

[2]Our single-unit cardinality-penalized formulation is identical to that of d'Aspremont et al. [ABE08].

one has $\|Az\|_2 - \gamma\|z\|_1 < 0$ for all nonzero vectors z whenever γ is chosen to be strictly bigger than $\|a_{i^*}\|_2$. From now on we assume that

$$\gamma < \|a_{i^*}\|_2. \tag{5.3}$$

A trade-off can be found between the value $\|Az^*(\gamma)\|_2$ and the sparsity of the solution $z^*(\gamma)$. The penalty parameter γ is introduced to "continuously" interpolate between the two extreme cases described above, with values in the interval $[0, \|a_{i^*}\|_2)$. It depends on the particular application whether sparsity is valued more than the explained variance, or vice versa, and to what extent. Due to these considerations, we consider the solution of (5.2) to provide a *sparse principal component* of A.

Reformulation. The objective function in (5.2) is not convex, nor concave. The feasible set is furthermore of a high dimension for large n. These shortcomings are overcome by considering the following reformulation,

$$\begin{aligned}
\phi_{\ell_1}(\gamma) &= \max_{z\in\mathcal{B}^n} \|Az\|_2 - \gamma\|z\|_1 \\
&= \max_{z\in\mathcal{B}^n} \max_{x\in\mathcal{B}^m} x^T Az - \gamma\|z\|_1 && (5.4) \\
&= \max_{x\in\mathcal{B}^m} \max_{z\in\mathcal{B}^n} \sum_{i=1}^n z_i(a_i^T x) - \gamma|z_i| \\
&= \max_{x\in\mathcal{B}^m} \max_{\bar{z}\in\mathcal{B}^n} \sum_{i=1}^n |\bar{z}_i|(|a_i^T x| - \gamma), && (5.5)
\end{aligned}$$

where $z_i = \operatorname{sign}(a_i^T x)\bar{z}_i$. In view of (5.3), there is some $x \in \mathcal{B}^n$ for which $a_i^T x > \gamma$. Fixing such x, solving the inner maximization problem for $\bar{z}$ and then translating back to z, we obtain the closed-form solution

$$z_i^* = z_i^*(\gamma) = \frac{\operatorname{sign}(a_i^T x)[|a_i^T x| - \gamma]_+}{\sqrt{\sum_{k=1}^n [|a_k^T x| - \gamma]_+^2}}, \quad i = 1, \ldots, n. \tag{5.6}$$

Problem (5.5) can therefore be written in the form

$$\phi_{\ell_1}^2(\gamma) = \max_{x\in\mathcal{S}^{m-1}} \sum_{i=1}^n [|a_i^T x| - \gamma]_+^2. \tag{5.7}$$

The objective function in (5.7) is differentiable and convex, and hence all local and global maxima must lie on the boundary, i.e., on the unit Euclidean sphere $\mathcal{S}^{m-1}$. Also, in the case when $m \ll n$, formulation (5.7) requires to search a space of a much lower dimension than the initial problem (5.2).

Sparsity. In view of (5.6), an optimal solution x^* of (5.7) defines a sparsity pattern of the vector z^*. In fact, the coefficients of z^* indexed by

$$\mathcal{I} = \{i \mid |a_i^T x^*| > \gamma\} \tag{5.8}$$

are active while all others must be zero. Geometrically, active indices correspond to the defining hyperplanes of the polytope

$$\mathcal{D} = \{x \in \mathbf{R}^m \mid |a_i^T x| \leq 1\}$$

that are (strictly) crossed by the line joining the origin and the point $\frac{x^*}{\gamma}$. It is even possible to say something about the sparsity of the solution without the knowledge of x^*: for any $i = 1, \ldots, n$ such that $\gamma \geq \|a_i\|_2$, it holds that

$$z_i^*(\gamma) = 0. \tag{5.9}$$

Single-unit sparse PCA via cardinality penalty

Instead of the ℓ_1-penalization, the authors of [ABE08] consider the formulation

$$\phi_{\ell_0}(\gamma) \stackrel{\text{def}}{=} \max_{z \in \mathcal{B}^n} z^T A^T A z - \gamma \|z\|_0, \tag{5.10}$$

which directly penalizes the number of nonzero components (cardinality) of the vector z.

Reformulation. The reasoning of the previous section suggests the reformulation

$$\phi_{\ell_0}(\gamma) = \max_{x \in \mathcal{B}^m} \max_{z \in \mathcal{B}^n} (x^T A z)^2 - \gamma \|z\|_0, \tag{5.11}$$

where the maximization with respect to $z \in \mathcal{B}^n$ for a fixed $x \in \mathcal{B}^m$ has the closed-form solution

$$z_i^* = z_i^*(\gamma) = \frac{[\operatorname{sign}((a_i^T x)^2 - \gamma)]_+ a_i^T x}{\sqrt{\sum_{k=1}^n [\operatorname{sign}((a_k^T x)^2 - \gamma)]_+ (a_k^T x)^2}}, \quad i = 1, \ldots, n. \tag{5.12}$$

In analogy with the ℓ_1 case, this derivation assumes that

$$\gamma < \|a_{i^*}\|_2^2,$$

so that there is $x \in \mathcal{B}^n$ such that $(a_i^T x)^2 - \gamma > 0$. Otherwise $z^* = 0$ is optimal. Formula (5.12) is easily obtained by analyzing (5.11) separately for fixed cardinality values of z. Hence, problem (5.10) is cast in the following form,

$$\phi_{\ell_0}(\gamma) = \max_{x \in \mathcal{S}^{m-1}} \sum_{i=1}^n [(a_i^T x)^2 - \gamma]_+. \tag{5.13}$$

Again, the objective function is convex, albeit non-smooth, and the new search space is of particular interest if $m \ll n$. A different derivation of (5.13) for the $n = m$ case can be found in [ABE08].

Sparsity. Given a solution x^* of (5.13), the set of active indices of z^* is given by

$$\mathcal{I} = \{i \mid (a_i^T x^*)^2 > \gamma\}.$$

Geometrically, active indices correspond to the defining hyperplanes of the polytope

$$\mathcal{D} = \{x \in \mathbf{R}^m \mid |a_i^T x| \leq 1\}$$

that are (strictly) crossed by the line joining the origin and the point $\frac{x^*}{\sqrt{\gamma}}$. As in the ℓ_1 case, we have

$$z_i^*(\gamma) = 0, \tag{5.14}$$

for any $i = 1, \dots, n$ such that $\gamma \geq \|a_i\|_2^2$.

5.1.2 Sparse PCA as a maximization with orthonormality constraints

In many applications, several components need to be identified. The traditional approach consists in incorporating an existing single-unit algorithm in a deflation scheme and to compute the desired number of components sequentially (see, e.g., d'Aspremont et al. [AEJL07]). In the case of Rayleigh quotient maximization it is well-known that computing several components at once instead of computing them one-by-one by deflation with the classical power method might present better convergence whenever the largest eigenvalues of the underlying matrix are close to each other (see, e.g., Parlett [Par80]). Therefore, block approaches for sparse PCA are expected to be more efficient on ill-posed problems. Block formulations of sparse PCA come also in two variants, with either an ℓ_1 or an ℓ_0 (cardinality) penalty to enforce sparsity.

Block sparse PCA via ℓ_1-penalty

Consider the following block generalization of (5.4),

$$\phi_{\ell_1,m}(\gamma) \stackrel{\text{def}}{=} \max_{\substack{X \in \mathrm{St}(p,m) \\ Z \in [\mathcal{S}^{n-1}]^p}} \mathrm{Tr}(X^T A Z N) - \gamma \sum_{j=1}^{p} \sum_{i=1}^{n} |z_{ij}|, \tag{5.15}$$

where $\gamma \geq 0$ is a sparsity-controlling parameter and $N = \mathrm{Diag}(\mu_1, \dots, \mu_p)$, with positive entries on the diagonal and

$$[\mathcal{S}^{n-1}]^p = \{X \in \mathbf{R}^{n \times p} \mid \mathrm{Diag}(X^T X) = I_p\},$$

is the space of n-by-p matrices with unit-norm columns (i.e., the product of p spheres $\mathcal{S}^{n-1}$). The dimension p corresponds to the number of extracted components and is assumed to be smaller or equal to the rank of the data matrix, i.e., $p \leq \mathrm{Rank}(A)$. It will be shown below that under some conditions on the parameters μ_i, the case $\gamma = 0$ recovers PCA. In that particular instance, any solution Z^* of (5.15) has orthonormal columns, although this is not explicitly enforced. For positive γ, the columns of Z^* are not expected to be orthogonal anymore. Most existing algorithms for computing a set of sparse principal components, e.g., [ZHT06, AEJL07, SH08], also do not impose orthogonal loading directions. Simultaneously enforcing sparsity and orthogonality seems to be a hard (and perhaps questionable) task.

Reformulation. Since problem (5.15) is completely decoupled in the columns of Z, i.e.,

$$\phi_{\ell_1,m}(\gamma) = \max_{X \in \mathrm{St}(p,m)} \sum_{j=1}^{p} \max_{z_j \in \mathcal{S}^{n-1}} \mu_j x_j^T A z_j - \gamma \|z_j\|_1,$$

the closed-form solution (5.6) of (5.4) is easily adapted to the block formulation (5.15),

$$z_{ij}^* = z_{ij}^*(\gamma) = \frac{\operatorname{sign}(a_i^T x_j)[\mu_j |a_i^T x_j| - \gamma]_+}{\sqrt{\sum_{k=1}^n [\mu_j |a_k^T x_j| - \gamma]_+^2}}. \tag{5.16}$$

This leads to the reformulation

$$\phi_{\ell_1,m}^2(\gamma) = \max_{X \in \mathrm{St}(p,m)} \sum_{j=1}^p \sum_{i=1}^n [\mu_j |a_i^T x_j| - \gamma]_+^2, \tag{5.17}$$

which maximizes a convex function $f : \mathbf{R}^{m \times p} \to \mathbf{R}$ on the Stiefel manifold $\mathrm{St}(p, m)$.

Sparsity. A solution X^* of (5.17) again defines the sparsity pattern of the matrix Z^*: the entry z_{ij}^* is active if

$$\mu_j |a_i^T x_j^*| > \gamma,$$

and equal to zero otherwise. For $\gamma > \max_{i,j} \mu_j \|a_i\|_2$, the trivial solution $Z^* = 0$ is optimal.

Block PCA. For $\gamma = 0$, problem (5.17) is equivalently written in the form

$$\phi_{\ell_1,m}^2(0) = \max_{X \in \mathrm{St}(p,m)} \operatorname{Tr}(X^T A A^T X N^2), \tag{5.18}$$

which has been well studied (see, e.g., Brockett [Bro91] and Absil et al. [AMS08]). The solutions of (5.18) span the dominant p-dimensional invariant subspace of the matrix AA^T. Furthermore, if the parameters μ_i are all distinct, the columns of X^* are the p dominant eigenvectors of AA^T, i.e., the p dominant left-eigenvectors of the data matrix A. The columns of the solution Z^* of (5.15) are thus the p dominant right singular vectors of A, i.e., the PCA loading vectors. Such a matrix N with distinct diagonal elements enforces the objective function in (5.18) to have isolated maximizers. In fact, if $N = I_p$, any point X^*Q with X^* a solution of (5.18) and $Q \in \mathcal{O}(p)$ is also a solution of (5.18). In the case of sparse PCA, i.e., $\gamma > 0$, the penalty term enforces isolated maximizers. The technical parameter N is therefore set to the identity matrix in what follows.

Block sparse PCA via cardinality penalty

The single-unit cardinality-penalized case can also be naturally extended to the block case,

$$\phi_{\ell_0,m}(\gamma) \stackrel{\mathrm{def}}{=} \max_{\substack{X \in \mathrm{St}(p,m) \\ Z \in [\mathcal{S}^{n-1}]^p}} \operatorname{Tr}(\operatorname{Diag}(X^T A Z N)^2) - \gamma \|Z\|_0, \tag{5.19}$$

where $\gamma \geq 0$ is the sparsity-inducing parameter and $N = \operatorname{Diag}(\mu_1, \ldots, \mu_p)$ with positive entries on the diagonal. In the case $\gamma = 0$, problem (5.21) is equivalent to (5.18) and therefore corresponds to PCA, provided that all μ_i are distinct.

Reformulation. Again, this block formulation is completely decoupled in the columns of Z,

$$\phi_{\ell_0,m}(\gamma) = \max_{X \in \mathrm{St}(p,m)} \sum_{j=1}^p \max_{z_j \in \mathcal{S}^{n-1}} (\mu_j x_j^T A z_j)^2 - \gamma \|z_j\|_0,$$

so that the solution (5.12) of the single unit case provides the optimal columns z_i,

$$z_{ij}^* = z_{ij}^*(\gamma) = \frac{[\mathrm{sign}((\mu_j a_i^T x_j)^2 - \gamma)]_+ \mu_j a_i^T x_j}{\sqrt{\sum_{k=1}^n [\mathrm{sign}((\mu_j a_k^T x_j)^2 - \gamma)]_+ \mu_j^2 (a_k^T x_j)^2}}. \tag{5.20}$$

The reformulation of problem (5.19) is thus

$$\phi_{\ell_0,m}(\gamma) = \max_{X \in \mathrm{St}(p,m)} \sum_{j=1}^p \sum_{i=1}^n [(\mu_j a_i^T x_j)^2 - \gamma]_+, \tag{5.21}$$

which maximizes a convex function $f : \mathbf{R}^{m\times p} \to \mathbf{R}$ on the Stiefel manifold $\mathrm{St}(p,m)$.

Sparsity. For a solution X^* of (5.21), the active entries z_{ij}^* of Z^* are given by the condition

$$(\mu_j a_i^T x_j^*)^2 > \gamma.$$

Hence for $\gamma > \max\limits_{i,j} \mu_j \|a_i\|_2^2$, the optimal solution of (5.19) is $Z^* = 0$.

5.2 Maximization of convex functions on compact sets

In this section, we propose and analyze a simple gradient-type method for maximizing a convex function $f : \mathbf{E} \to \mathbf{R}$ on a compact set $\mathcal{Q}$,

$$f^* = \max_{x \in \mathcal{Q}} f(x), \tag{P$_2$}$$

where $\mathbf{E}$ an arbitrary vector space. Unless explicitly stated otherwise, the function f is *not* assumed to be differentiable.

Let $\mathbf{E}^*$ be the conjugate space of $\mathbf{E}$, i.e., the space of all linear functionals on $\mathbf{E}$. By $\langle s, x\rangle$ we denote the action of $s \in \mathbf{E}^*$ on $x \in \mathbf{E}$. For a self-adjoint positive definite linear operator $G : \mathbf{E} \to \mathbf{E}^*$ we define a pair of norms on $\mathbf{E}$ and $\mathbf{E}^*$ as follows

$$\begin{array}{lcll} \|x\| & \stackrel{\text{def}}{=} & \langle Gx, x\rangle^{\frac{1}{2}}, & x \in \mathbf{E}, \\ \|s\|_* & \stackrel{\text{def}}{=} & \langle s, G^{-1}s\rangle^{\frac{1}{2}}, & s \in \mathbf{E}^*. \end{array} \tag{5.22}$$

Although the forthcoming theory is developed in this general setting, the sparse PCA formulations of Section 5.1 require either the choice $\mathbf{E} = \mathbf{E}^* = \mathbf{R}^m$ or $\mathbf{E} = \mathbf{E}^* = \mathbf{R}^{m\times p}$. In both cases, G is the corresponding identity operator for which we obtain

$$\begin{array}{lcllll} \langle s, x\rangle & = & s^T x, & \|x\| = \langle x, x\rangle^{\frac{1}{2}} = \|x\|_2, & x, s \in \mathbf{R}^m, & \text{and} \\ \langle s, x\rangle & = & \mathrm{Tr}(s^T x), & \|x\| = \langle x, x\rangle^{\frac{1}{2}} = \|x\|_F, & x, s \in \mathbf{R}^{m\times p}. & \end{array}$$

In this chapter, the notation $\nabla f(x)$ refers to any subgradient of function f at x. By $\partial f(x)$ we denote its subdifferential.[3] At any point $x \in \mathcal{Q}$ we introduce some measure for the first-order

[3]The *subgradient* generalizes the notion of gradient to convex but non-smooth functions. The subgradient of a convex function f at a point $x \in \mathbf{E}$ is an element $\nabla f(x) \in \mathbf{E}^*$ such that

$$f(y) - f(x) \geq \langle \nabla f(x), y - x\rangle$$

for any $y \in \mathbf{E}$. The *subdifferential* $\partial f(x)$ is the set of all subgradients of f at x.

optimality conditions,

$$\Delta(x) \stackrel{\text{def}}{=} \max_{y \in \mathcal{Q}} \langle \nabla f(x), y - x \rangle.$$

Clearly, $\Delta(x) \geq 0$ and it vanishes only at the points where the gradient $\nabla f(x)$ belongs to the normal cone[4] to the convex hull[5] of the set $\mathcal{Q}$, denoted $\mathrm{Conv}(\mathcal{Q})$, at x. The optimality conditions are hence satisfied once $\Delta(x) = 0$.[6]

5.2.1 A gradient algorithm

Consider the following simple algorithmic scheme, which maximizes at each iteration the best linear approximation of the objective function f. By virtue of its convexity, a *lower bound* to the objective is repeatedly maximized.

Algorithm 1: Gradient scheme

input : Initial iterate $x_0 \in \mathbf{E}$.
output: x_k, approximate solution of (P_2)
begin
  $k \longleftarrow 0$
  repeat
    $x_{k+1} \in \operatorname{Arg\,max}\{f(x_k) + \langle \nabla f(x_k), y - x_k \rangle \mid y \in \mathcal{Q}\}$
    $k \longleftarrow k + 1$
  until *a stopping criterion is satisfied*
end

Depending on the nature of the set $\mathcal{Q}$, the inner problem, i.e., the maximization of a linear function on $\mathcal{Q}$, can sometimes be very simple. For instance, in the special cases of the sphere and the ball of radius $r > 0$, i.e.,

$$\mathcal{Q} = \{x \in \mathbf{E} \mid \|x\| = r\} \quad \text{and} \quad \mathcal{Q} = \{x \in \mathbf{E} \mid \|x\| \leq r\},$$

the main step of Algorithm 1 can be written in an explicit form,

$$x_{k+1} = r \frac{G^{-1} \nabla f(x_k)}{\|\nabla f(x_k)\|_*}. \tag{5.23}$$

5.2.2 Convergence analysis

The following theorems indicate that the proposed gradient method has best theoretical convergence properties when either f or $\mathcal{Q}$ are strongly convex. Such a situation can always be enforced by adding a strongly convex regularizing term to the objective function, constant on the feasible set. We do not, however, prove any results concerning the quality of the obtained solution. Even the goal of obtaining a local maximizer is in general unattainable, and we must be content with convergence to a stationary point. Our first convergence result is straightforward.

[4]The *normal cone* to a compact and convex set $\mathcal{Q}$ at a point $x \in \mathcal{Q}$ is the set $\{s \in \mathbf{E}^* \mid \langle s, y - x \rangle \geq 0, \text{ for all } y \in \mathcal{Q}\}$.

[5]The *convex hull* of a set Q is the smallest convex set that contains Q.

[6]The normal cone to the set $\mathrm{Conv}(\mathcal{Q})$ at $x \in \mathcal{Q}$ is *smaller* than the normal cone to the set $\mathcal{Q}$. Therefore, the optimality condition $\Delta(x) = 0$ is *stronger* than the standard one.

Theorem 5.2.1 *Let sequence $\{x_k\}_{k=0}^{\infty}$ be generated by Algorithm 1 as applied to a convex function f. Then the sequence $\{f(x_k)\}_{k=0}^{\infty}$ is monotonically increasing and $\lim_{k\to\infty} \Delta(x_k) = 0$. Moreover,*

$$\Delta_k \stackrel{def}{=} \min_{0\leq i\leq k} \Delta(x_i) \leq \frac{f^* - f(x_0)}{k+1}. \tag{5.24}$$

Proof. From convexity of f, it holds that

$$f(x_{k+1}) \geq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle = f(x_k) + \Delta(x_k),$$

and therefore, $f(x_{k+1}) \geq f(x_k)$ for all k. By summing up these inequalities for $k = 0, 1, \ldots, N-1$, we obtain

$$f^* - f(x_0) \geq f(x_k) - f(x_0) \geq \sum_{i=0}^{k} \Delta(x_i),$$

and the result follows. □

For a sharper analysis, we need some technical assumptions on f and $\mathcal{Q}$.

Assumption 5.2.2 *The norms of the subgradients of f are bounded from below on $\mathcal{Q}$ by a positive constant, i.e.,*

$$\delta_f \stackrel{def}{=} \min_{\substack{x\in\mathcal{Q} \\ \nabla f(x)\in\partial f(x)}} \|\nabla f(x)\|_* > 0. \tag{5.25}$$

This assumption is not too binding because of the following result.

Proposition 5.2.3 *Assume that there exists a point $\bar{x} \notin \mathcal{Q}$ such that $f(\bar{x}) < f(x)$ for all $x \in \mathcal{Q}$. Then*

$$\delta_f \geq \frac{\min_{x\in\mathcal{Q}} f(x) - f(\bar{x})}{\max_{x\in\mathcal{Q}} \|x - \bar{x}\|} > 0.$$

Proof. Because f is convex, for any $x \in \mathcal{Q}$ it holds that

$$0 < f(x) - f(\bar{x}) \leq \langle \nabla f(x), x - \bar{x} \rangle \leq \|\nabla f(x)\|_* \, \|x - \bar{x}\|.$$

□

For our next convergence result we need to assume either strong convexity of f or strong convexity of the set $\operatorname{Conv}(\mathcal{Q})$.

Assumption 5.2.4 *Function f is* strongly convex, *i.e., there exists a constant $\sigma_f > 0$ such that for any $x, y \in \mathbf{E}$*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\sigma_f}{2}\|y - x\|^2. \tag{5.26}$$

Convex functions satisfy this inequality for *convexity parameter* $\sigma_f = 0$.

Assumption 5.2.5 *The set* $\operatorname{Conv}(\mathcal{Q})$ *is* strongly convex, *i.e., there exists a constant $\sigma_{\mathcal{Q}} > 0$ such that for any $x, y \in \operatorname{Conv}(\mathcal{Q})$ and $\alpha \in [0,1]$ the following inclusion holds,*

$$\{\alpha x + (1-\alpha) y + \frac{\sigma_{\mathcal{Q}}}{2}\alpha(1-\alpha)\|x - y\|^2 z \mid z \in \mathbf{E}, \|z\| = 1\} \subset \operatorname{Conv}(\mathcal{Q}). \tag{5.27}$$

Convex sets satisfy this inclusion for *convexity parameter* $\sigma_{\mathcal{Q}} = 0$.

As indicated in the forthcoming Theorem 5.2.8, a better analysis of Algorithm 1 is possible if $\mathrm{Conv}(\mathcal{Q})$, the convex hull of the feasible set of problem (P_2), is strongly convex. Note that in the case of the two formulations (5.7) and (5.13) of the sparse PCA problem, the feasible set $\mathcal{Q}$ is the unit Euclidean sphere. Since the convex hull of the unit sphere is the unit ball, which is a strongly convex set, the feasible set of our sparse PCA formulations satisfies Assumption 5.2.5.

Example 5.2.6 (Strong convexity of the ball) *In the special case of the sphere*

$$\mathcal{Q} = \{x \in \mathbf{E} \mid \|x\| = r\},$$

for some $r > 0$, there is a simple proof that Assumption 5.2.5 holds with $\sigma_{\mathcal{Q}} = \frac{1}{r}$. Indeed, for any $x, y \in \mathbf{E}$ and $\alpha \in [0,1]$, we have

$$\begin{aligned} \|\alpha x + (1-\alpha)y\|^2 &= \alpha^2\|x\|^2 + (1-\alpha)^2\|y\|^2 + 2\alpha(1-\alpha)\langle Gx, y\rangle \\ &= \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2. \end{aligned}$$

Thus, for $x, y \in \mathcal{Q}$ we obtain,

$$\|\alpha x + (1-\alpha)y\| = \left[r^2 - \alpha(1-\alpha)\|x-y\|^2\right]^{\frac{1}{2}} \leq r - \frac{1}{2r}\alpha(1-\alpha)\|x-y\|^2.$$

Hence, we can take $\sigma_{\mathcal{Q}} = \frac{1}{r}$.

The relevance of Assumption 5.2.5 is justified by the following technical observation.

Proposition 5.2.7 *Let Assumption 5.2.5 be satisfied. Then for any $x \in Q$, the following holds,*

$$\Delta(x) \geq \frac{\sigma_{\mathcal{Q}}}{2}\|\nabla f(x)\|_* \, \|y(x) - x\|^2, \tag{5.28}$$

where $y(x) \overset{def}{\in} \arg\max_{y \in \mathcal{Q}} \langle \nabla f(x), y - x\rangle$.

Proof. At an arbitrary $x \in \mathcal{Q}$, it holds that

$$\langle \nabla f(x), y(x) - y\rangle \geq 0, \quad y \in \mathrm{Conv}(\mathcal{Q}).$$

We use this inequality for

$$y = y_\alpha \overset{\mathrm{def}}{=} x + \alpha(y(x) - x) + \frac{\sigma_{\mathcal{Q}}}{2}\alpha(1-\alpha)\|y(x) - x\|^2 \, \frac{G^{-1}\nabla f(x)}{\|\nabla f(x)\|_*}, \ \alpha \in [0,1].$$

In view of Assumption 5.2.5, $y_\alpha \in \mathrm{Conv}(\mathcal{Q})$. Therefore,

$$0 \geq \langle \nabla f(x), y_\alpha - y(x)\rangle = (1-\alpha)\langle \nabla f(x), x - y(x)\rangle + \frac{\sigma_{\mathcal{Q}}}{2}\alpha(1-\alpha)\|y(x) - x\|^2 \, \|\nabla f(x)\|_*.$$

Since α is an arbitrary value from $[0,1]$, the result follows. □

We are now ready to refine our analysis of Algorithm 1.

Theorem 5.2.8 (Convergence) *Let f be convex and let Assumption 5.2.2 and at least one of Assumptions 5.2.4 and 5.2.5 be satisfied. If $\{x_k\}$ is the sequence of points generated by Algorithm 1, then*

$$\sum_{k=0}^{N} \|x_{k+1} - x_k\|^2 \leq \frac{2(f^* - f(x_0))}{\sigma_{\mathcal{Q}}\delta_f + \sigma_f}. \tag{5.29}$$

Proof. Indeed, in view of our assumptions and Proposition 5.2.7, it holds that

$$f(x_{k+1}) - f(x_k) \geq \Delta(x_k) + \frac{\sigma_f}{2}\|x_{k+1} - x_k\|^2 \geq \frac{1}{2}(\sigma_{\mathcal{Q}}\delta_f + \sigma_f)\|x_{k+1} - x_k\|^2.$$

□

We cannot in general guarantee that the algorithm converges to a unique local maximizer. In particular, if started from a local minimizer, the method does not move away from this point. However, the above statement guarantees that all of its limit points satisfy the first-order optimality condition.

5.2.3 Maximization with spherical constraints

Consider $\mathbf{E} = \mathbf{E}^* = \mathbf{R}^m$ with $G = I_m$ and $\langle x, y\rangle = x^T y$, and let $\mathcal{Q}$ be a sphere of radius r,

$$\mathcal{Q} = r \cdot \mathcal{S}^{m-1} = \{x \in \mathbf{R}^m \mid \|x\|_2 = r\}.$$

Problem (P_2) takes on the form

$$f^* = \max_{x \in r\cdot\mathcal{S}^{m-1}} f(x). \tag{5.30}$$

Since $\mathrm{Conv}(\mathcal{Q})$ is strongly convex ($\sigma_{\mathcal{Q}} = \frac{1}{r}$), Theorem 5.2.8 is meaningful for any convex function f ($\sigma_f \geq 0$). It has already be mentioned (see (5.23)) that the main step of Algorithm 1 can be written down explicitly,

$$x_{k+1} = r\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|_2}.$$

Note that the single-unit sparse PCA formulations (5.7) and (5.13) conform to the setting (5.30). The following examples illustrate the connection to classical algorithms.

Example 5.2.9 (Power method) *In the special case of a quadratic objective function*

$$f(x) = \tfrac{1}{2}x^T C x$$

for some positive definite matrix $C \in \mathbf{S}^m$ on the unit sphere ($r = 1$), it holds that

$$f^* = \tfrac{1}{2}\lambda_1(C),$$

and Algorithm 1 is equivalent to the power iteration method *for computing the largest eigenvalue of C (see, e.g., Golub and Van Loan [GVL89]). Hence for $\mathcal{Q} = \mathcal{S}^{m-1}$, we can think of our scheme as a generalization of the power method. Indeed, our algorithm performs the following iteration,*

$$x_{k+1} = \frac{Cx_k}{\|Cx_k\|_2}, \quad k \geq 0.$$

Both δ_f and σ_f are equal to the smallest eigenvalue of C, and hence the right-hand side of (5.29) is equal to

$$\frac{\lambda_1(C) - x_0^T C x_0}{2\lambda_{min}(C)}. \tag{5.31}$$

Example 5.2.10 (Shifted power method) *If C is not positive semidefinite in the previous example, the objective function is not convex and our results are not applicable. However, this complication is circumvented by instead running the algorithm with the shifted quadratic function*

$$\bar{f}(x) = \frac{1}{2}x^T(C + \omega I_m)x,$$

where $\omega > 0$ is chosen for $\bar{C} = \omega I_m + C \in \mathbf{S}^m$ to be positive definite. On the feasible set, this change only adds a constant term to the objective function. The method, however, produces different sequence of iterates. The constants δ_f and σ_f are also affected and, correspondingly, the estimate (5.31).

5.2.4 Maximization with orthonormality constraints

Consider $\mathbf{E} = \mathbf{E}^* = \mathbf{R}^{m\times p}$, the space of m-by-p real matrices, with $p \leq m$. Note that the case $p = 1$ recovers the setting of the previous section. This space is assumed to be equipped with the trace inner product, $\langle X, Y\rangle = \mathrm{Tr}(X^T Y)$. The induced norm, denoted by $\|X\|_F \stackrel{\text{def}}{=} \langle X, X\rangle^{\frac{1}{2}}$, is the Frobenius norm (we let G be the identity operator). We can now consider various feasible sets, the simplest being a sphere or a ball, i.e.,

$$\mathcal{Q} = \{X \in \mathbf{R}^{m\times p} \mid \|X\|_F = r\} \;\text{ or }\; \mathcal{Q} = \{X \in \mathbf{R}^{m\times p} \mid \|X\|_F \leq r\}.$$

Due to the nature of applications in this chapter, let us concentrate on the situation when $\mathcal{Q}$ is a special subset of the sphere of radius $r = \sqrt{p}$, the Stiefel manifold $\mathrm{St}(p, m)$,

$$\mathcal{Q} = \mathrm{St}(p, m) = \{X \in \mathbf{R}^{m\times p} \mid X^T X = I_p\}.$$

Problem (P_2) then takes on the following form

$$f^* = \max_{X\in \mathrm{St}(p,m)} f(X). \tag{5.32}$$

The set $\mathrm{Conv}(\mathcal{Q})$ is not strongly convex ($\sigma_{\mathcal{Q}} = 0$), and hence Theorem 5.2.8 is meaningful only if f is strongly convex ($\sigma_f > 0$). At every iteration, the algorithm needs to maximize a linear function over the Stiefel manifold. The following standard result shows how this can be done.

Proposition 5.2.11 *Let $C \in \mathbf{R}^{m\times p}$, with $p \leq m$, and denote by $\sigma_i(C)$, $i = 1, \dots, p$, the singular values of C. Then*

$$\max_{X\in \mathrm{St}(p,m)} \langle C, X\rangle = \mathrm{Tr}[(C^T C)^{\frac{1}{2}}] = \sum_{i=1}^{p} \sigma_i(C), \tag{5.33}$$

and a maximizer X^* is given by the U factor in the polar decomposition of C,

$$C = US, \quad U \in \mathrm{St}(p, m),\; S \in \mathbf{S}^p,\; S \succeq 0.$$

If C is of full rank, then we can take $X^* = C(C^T C)^{-\frac{1}{2}}$.

Proof. Existence of the polar factorization in the nonsquare case is covered by Theorem 7.3.2 in [HJ85]. Let $C = V\Sigma W^T$ be the singular value decomposition of A, i.e., V is m-by-m orthogonal, W is p-by-p orthogonal, and Σ is m-by-p diagonal with values $\sigma_i(A)$ on the diagonal. Then

$$\begin{aligned}\max_{X\in \mathrm{St}(p,m)} \langle C, X\rangle &= \max_{X\in \mathrm{St}(p,m)} \langle V\Sigma W^T, X\rangle \\ &= \max_{X\in \mathrm{St}(p,m)} \mathrm{Tr}(\Sigma(W^T X^T V)) \\ &= \max_{Z\in \mathrm{St}(p,m)} \mathrm{Tr}(\Sigma Z^T) = \max_{Z\in \mathrm{St}(p,m)} \sum_{i=1}^{p} \sigma_i(C) z_{ii} \leq \sum_{i}^{p} \sigma_i(C).\end{aligned}$$

The third equality follows since the function $X \mapsto V^T X W$ maps $\mathrm{St}(p, m)$ onto itself. It remains to note that

$$\langle C, U\rangle = \mathrm{Tr}(S) = \sum_i \lambda_i(S) = \sum_i \sigma_i(S) = \mathrm{Tr}[(S^T S)^{\frac{1}{2}}] = \mathrm{Tr}[(C^T C)^{\frac{1}{2}}] = \sum_i \sigma_i(C),$$

Finally, in the full rank case we have $\langle C, X^*\rangle = \mathrm{Tr}[C^T C(C^T C)^{-\frac{1}{2}}] = \mathrm{Tr}[(C^T C)^{\frac{1}{2}}]$. □

Let the symbol $\mathrm{uf}(C)$ denote the U factor of the polar decomposition of matrix $C \in \mathbf{R}^{m\times p}$, or equivalently, $\mathrm{uf}(C) = C(C^T C)^{-\frac{1}{2}}$ if C is of full rank. In view of the above result, the main step of Algorithm 1 can be written in the form

$$x_{k+1} = \mathrm{uf}(\nabla f(x_k)). \tag{5.34}$$

The block sparse PCA formulations (5.17) and (5.21) conform to the setting (5.32). Here is one more example.

Example 5.2.12 (Rectangular Procrustes problem) *Let $C, X \in \mathbf{R}^{m\times p}$ and $D \in \mathbf{R}^{p\times p}$ and consider the following problem,*

$$\min\{\|C - DX\|_F^2 \mid X^T X = I_p\}. \tag{5.35}$$

Since $\|C - DX\|_F^2 = \|C\|_F^2 + \langle DX, DX\rangle - 2\langle CD, X\rangle$, by a similar shifting technique as in the previous example we can cast problem (5.35) in the following form

$$\max\{\omega\|X\|_F^2 - \langle DX, DX\rangle + 2\langle CD, X\rangle \mid X^T X = I_p\}.$$

For $\omega > 0$ large enough, the new objective function is strongly convex. In this case our algorithm becomes similar to the gradient method proposed by [FND08]. The standard Procrustes problem in the literature is a special case of (5.35) with $p = m$.

5.3 Algorithms for sparse principal component analysis

The solutions of the sparse PCA formulations of Section 5.1 provide locally optimal patterns of zero and nonzero entries for the vector $z \in \mathcal{S}^{n-1}$ (in the single-unit case) or the matrix $Z \in [\mathcal{S}^{n-1}]^p$ (in the block case). The sparsity-inducing penalty term used in these formulations biases however the values assigned to the nonzero entries, which should be readjusted by considering the sole objective of maximum variance. An algorithm for sparse PCA combines thus a method that identifies a "good" pattern of sparsity with a method that fills the active entries. In the sequel, we discuss the general block sparse PCA problem. The single-unit case is recovered in the particular case $p = 1$.

Methods for pattern-finding

The application of the general method (Algorithm 1) to the four sparse PCA formulations (5.7), (5.13), (5.17) and (5.21) leads to Algorithms 2, 3, 4 and 5 below, that provide a locally optimal pattern of sparsity for a matrix $Z \in [\mathcal{S}^{n-1}]^p$. This pattern is defined as a binary matrix $P \in \{0,1\}^{n\times p}$ such that $p_{ij} = 1$ if the loading z_{ij} is active and $p_{ij} = 0$ otherwise. So, P is an indicator of the coefficients of Z that are zeroed by our method. The computational complexity of the single-unit algorithms (Algorithms 2 and 3) is $\mathcal{O}(nm)$ operations per iteration. The block algorithms (Algorithms 4 and 5) have complexity $\mathcal{O}(nmp)$ per iteration.

Algorithm 2: Single-unit sparse PCA method based on the ℓ_1-penalty (5.7)

input : Data matrix $A \in \mathbf{R}^{m\times n}$
Sparsity-controlling parameter $\gamma \geq 0$
Initial iterate $x \in \mathcal{S}^{m-1}$

output: A locally optimal sparsity pattern $P \in \{0,1\}^n$

begin

repeat

$x \longleftarrow \sum_{i=1}^n [|a_i^T x| - \gamma]_+ \operatorname{sign}(a_i^T x) a_i$

$x \longleftarrow \frac{x}{\|x\|_2}$

until *a stopping criterion is satisfied*

Construct a binary vector $P \in \{0,1\}^n$ such that $\begin{cases} p_i = 1 & \text{if } |a_i^T x| > \gamma \\ p_i = 0 & \text{otherwise.} \end{cases}$

end

Post-processing

Once a "good" sparsity pattern P is identified, the active entries of Z still have to be filled. To this end, we consider the optimization problem

$$(X^*, Z^*) \stackrel{\text{def}}{=} \underset{\substack{X\in \mathrm{St}(p,m) \\ Z\in[\mathcal{S}^{n-1}]^p \\ Z_{\bar{P}}=0}}{\arg\max} \operatorname{Tr}(X^T AZN), \tag{5.36}$$

where $\bar{P} \in \{0,1\}^{n\times p}$ is the complement of P, $Z_{\bar{P}}$ denotes the entries of Z that are constrained to zero and $N = \operatorname{Diag}(\mu_1, \ldots, \mu_p)$ with strictly positive μ_i. Problem (5.36) assigns the active part of the matrix Z to maximize the variance explained by the resulting components. Without loss of generality, each column of P is assumed to contain active elements.

Algorithm 3: Single-unit sparse PCA algorithm based on the ℓ_0-penalty (5.13)

input : Data matrix $A \in \mathbf{R}^{m\times n}$
Sparsity-controlling parameter $\gamma \geq 0$
Initial iterate $x \in \mathcal{S}^{m-1}$
output: A locally optimal sparsity pattern $P \in \{0,1\}^n$
begin
repeat
$x \longleftarrow \sum_{i=1}^{n}[\mathrm{sign}((a_i^T x)^2 - \gamma)]_+ \, a_i^T x \, a_i$
$x \longleftarrow \frac{x}{\|x\|_2}$
until *a stopping criterion is satisfied*
Construct a binary vector $P \in \{0,1\}^n$ such that $\begin{cases} p_i = 1 & \text{if } (a_i^T x)^2 > \gamma \\ p_i = 0 & \text{otherwise.} \end{cases}$
end

In the single-unit case $p = 1$, an explicit solution of (5.36) is available,

$$\begin{aligned} X^* &= u, \\ Z_P^* &= v \;\text{ and }\; Z_{barP}^* = 0, \end{aligned} \tag{5.37}$$

where σuv^T with $\sigma > 0$, $u \in \mathcal{S}^{m-1}$ and $v \in \mathcal{S}^{\|P\|_0 - 1}$ is a rank-one singular value decomposition of the matrix A_P, that corresponds to the submatrix of A containing the columns related to the active entries.

Although an exact solution of (5.36) is hard to compute in the block case $p > 1$, a local maximizer can be efficiently computed by optimizing alternatively with respect to one variable while keeping the other one fixed. The following two lemmas provide an explicit solution to each of these subproblems.

Lemma 5.3.1 *For a fixed $Z \in [\mathcal{S}^{n-1}]^p$, a solution X^* of*

$$\max_{X \in \mathrm{St}(p,m)} \mathrm{Tr}(X^T AZN)$$

is provided by the U factor of the polar decomposition of the product AZN.

Proof. See Proposition 5.2.11. □

Lemma 5.3.2 *The solution*

$$Z^* \stackrel{def}{=} \arg\max_{\substack{Z \in [\mathcal{S}^{n-1}]^p \\ Z_{\bar{P}} = 0}} \mathrm{Tr}(X^T AZN), \tag{5.38}$$

is at any point $X \in \mathrm{St}(p,m)$ defined by the two conditions $Z_P^* = (A^T XN)_P D$ and $Z_{\bar{P}}^* = 0$, where D is a positive diagonal matrix that normalizes the columns of Z^* to unit norm, i.e.,

$$D = \mathrm{Diag}((A^T XN)_P^T (A^T XN)_P)^{-\frac{1}{2}}.$$

Algorithm 4: Block Sparse PCA algorithm based on the ℓ_1-penalty (5.17)

input : Data matrix $A \in \mathbf{R}^{m\times n}$
Sparsity-controlling parameter $\gamma \geq 0$
Initial iterate $X \in \mathrm{St}(p, m)$
output: A locally optimal sparsity pattern $P \in \{0,1\}^{n\times p}$
begin
  repeat
    for $j = 1, \ldots, m$ **do**
      $x_j \longleftarrow \sum_{i=1}^{n} [|a_i^T x_j| - \gamma]_+ \operatorname{sign}(a_i^T x) a_i$
    $X \longleftarrow \mathrm{uf}(X)$
  until *a stopping criterion is satisfied*
  Construct a binary matrix $P \in \{0,1\}^{n\times p}$ such that $\begin{cases} p_{ij} = 1 & \text{if } |a_i^T x_j| > \gamma \\ p_{ij} = 0 & \text{otherwise.} \end{cases}$
end

Proof. The Lagrangian of the optimization problem (5.38) is

$$\mathcal{L}(Z, \Lambda_1, \Lambda_2) = \mathrm{Tr}(X^T AZN) - \mathrm{Tr}(\Lambda_1(Z^T Z - I_m)) - \mathrm{Tr}(\Lambda_2^T Z),$$

where the Lagrangian multipliers $\Lambda_1 \in \mathbf{R}^{p\times p}$ and $\Lambda_2 \in \mathbf{R}^{n\times p}$ have the following properties: Λ_1 is an invertible diagonal matrix and $(\Lambda_2)_P = 0$. The first-order optimality conditions of (5.38) are thus

$$\begin{aligned} A^T XN - 2Z\Lambda_1 - \Lambda_2 &= 0 \\ \mathrm{Diag}(Z^T Z) &= I_p \\ Z_{\bar{P}} &= 0. \end{aligned}$$

Hence, any stationary point Z^* of (5.38) satisfies $Z^*_P = (A^T XN)_P D$ and $Z^*_{\bar{P}} = 0$, where D is a diagonal matrix that normalizes the columns of Z^* to unit norm. The second-order optimality condition imposes the diagonal matrix D to be positive. Such a D is unique and given by $D = \mathrm{Diag}((A^T XN)_P^T (A^T XN)_P)^{-\frac{1}{2}}$. □

The alternating optimization scheme is summarized in Algorithm 6, which computes a local solution of (5.36). It should be noted that Algorithm 6 is a post-processing heuristic that, strictly speaking, is required only for the ℓ_1 block formulation (Algorithm 4). In fact, since the cardinality penalty only depends on the sparsity pattern P and not on the actual values assigned to Z_P, a solution (X^*, Z^*) of Algorithms 3 or 5 is also a local maximizer of (5.36) for the resulting pattern P. This explicit solution provides a good alternative to Algorithm 6. In the single unit case with ℓ_1 penalty (Algorithm 2), the solution (5.37) is available.

Sparse PCA algorithms

To sum up, we propose four sparse PCA algorithms, each combining a method to identify a "good" sparsity pattern with a method to fill the active entries of the p loading vectors. They

Algorithm 5: Block Sparse PCA algorithm based on the ℓ_0-penalty (5.21)

input : Data matrix $A \in \mathbf{R}^{m \times n}$
Sparsity-controlling parameter $\gamma \geq 0$
Initial iterate $X \in \mathrm{St}(p, m)$

output: A locally optimal sparsity pattern $P \in \{0,1\}^{n \times p}$

begin
  repeat
    for $j = 1, \ldots, m$ **do**
      $x_j \longleftarrow \sum_{i=1}^{n} [\mathrm{sign}((a_i^T x_j)^2 - \gamma)]_+ \, a_i^T x_j \, a_i$
    $X \longleftarrow \mathrm{uf}(X)$
  until *a stopping criterion is satisfied*
  Construct a binary matrix $P \in \{0,1\}^{n \times p}$ such that $\begin{cases} p_{ij} = 1 & \text{if } (a_i^T x_j)^2 > \gamma \\ p_{ij} = 0 & \text{otherwise.} \end{cases}$

end

are summarized in Table 5.1.[7]

Deflation scheme.

For the sake of completeness, we recall a classical deflation process for computing p sparse principal components with a single-unit algorithm (d'Aspremont et al. [AEJL07]). Let $z \in \mathbf{R}^n$ be a unit-norm sparse loading vector of the data A. Subsequent directions are sequentially obtained by computing a dominant sparse component of the residual matrix $A - yz^T$, where $y = Az$ is the vector that solves

$$\min_{y \in \mathbf{R}^m} \|A - yz^T\|_F.$$

Further deflation techniques for sparse PCA have been proposed in [Mac08].

5.4 Numerical experiments

In this section, we evaluate the proposed power algorithms against existing sparse PCA methods. Three competing methods are considered in this study: a greedy scheme aimed at computing a local maximizer of (5.10) (approximate greedy search algorithm in d'Aspremont et al. [ABE08]), the SPCA algorithm (Zou et al. [ZHT06]) and the sPCA-rSVD algorithm (Shen and Huang [SH08]). We do not include the DSPCA algorithm (d'Aspremont et al. [AEJL07]) in our numerical study. This method solves a convex relaxation of the sparse PCA problem and has a large computational complexity of $\mathcal{O}(n^4\sqrt{\log(n)})$ compared to the other methods. Table 5.2 lists the considered algorithms.

[7]Our algorithms are named GPower where the "G" stands for *generalized* or *gradient*.

Algorithm 6: Alternating optimization scheme for solving (5.36)

input : Data matrix $A \in \mathbf{R}^{m \times n}$
Sparsity pattern $P \in \{0,1\}^{n \times p}$
Matrix $N = \mathrm{Diag}(\mu_1, \ldots, \mu_p)$
Initial iterate $X \in \mathrm{St}(p, m)$
output: A local minimizer (X, Z) of (5.36)
begin
  repeat
    $Z \longleftarrow A^T X N$
    $Z_{\bar{P}} \longleftarrow 0$
    $Z \longleftarrow Z \ \mathrm{Diag}(Z^T Z)^{-\frac{1}{2}}$
    $X \longleftarrow \mathrm{uf}(AZN)$
  until *a stopping criterion is satisfied*
end

	Computation of P	Computation of Z_P
GPower$_{\ell_1}$	Algorithm 2	Equation (5.37)
GPower$_{\ell_0}$	Algorithm 3	Equation (5.12)
GPower$_{\ell_1,p}$	Algorithm 4	Algorithm 6
GPower$_{\ell_0,p}$	Algorithm 5	Equation (5.20)

Table 5.1: New algorithms for sparse PCA.

GPower$_{\ell_1}$	Single-unit sparse PCA via ℓ_1-penalty
GPower$_{\ell_0}$	Single-unit sparse PCA via ℓ_0-penalty
GPower$_{\ell_1,p}$	Block sparse PCA via ℓ_1-penalty
GPower$_{\ell_0,p}$	Block sparse PCA via ℓ_0-penalty
Greedy	Greedy method
SPCA	SPCA algorithm
rSVD$_{\ell_1}$	sPCA-rSVD algorithm with an ℓ_1-penalty (“soft thresholding”)
rSVD$_{\ell_0}$	sPCA-rSVD algorithm with an ℓ_0-penalty (“hard thresholding”)

Table 5.2: Sparse PCA algorithms we compare in this section.

5.4.1 Implementation

All numerical experiments are performed in MATLAB. Our implementations of the GPower algorithms are initialized at a point for which the associated sparsity pattern has *at least one* active element. In case of the single-unit algorithms, such an initial iterate $x \in \mathcal{S}^{m-1}$ is chosen parallel to the column of A with the largest norm, i.e.,

$$x = \frac{a_{i^*}}{\|a_{i^*}\|_2}, \qquad \text{where} \qquad i^* = \arg\max_i \|a_i\|_2. \tag{5.39}$$

For the block GPower algorithms, a suitable initial iterate $X \in \mathrm{St}(p, m)$ is constructed in a block-wise manner as $X = [x|X_\perp]$, where x is the unit-norm vector (5.39) and $X_\perp \in \mathrm{St}(p, m-1)$ is orthogonal to x, i.e., $x^T X_\perp = 0$. We stop the GPower algorithms once the relative change of the objective function is small,

$$\frac{f(x_{k+1}) - f(x_k)}{f(x_k)} \leq \epsilon = 10^{-4}.$$

MATLAB implementations of the SPCA algorithm and the greedy algorithm have been rendered available by Zou et al. [ZHT06] and d'Aspremont et al. [ABE08]. We have, however, implemented the sPCA-rSVD algorithm on our own (Algorithm 1 in [SH08]), and use it with the same stopping criterion as for the GPower algorithms. This algorithm initializes with the best rank-one approximation of the data matrix. This is done with the `svds` function in MATLAB.

Given a data matrix $A \in \mathbf{R}^{m \times n}$, the considered sparse PCA algorithms provide p unit-norm sparse loading vectors stored in the matrix $Z \in [\mathcal{S}^{n-1}]^p$. The samples of the associated components are provided by the p columns of the product AZ. The variance explained by these p components is an important comparison criterion of the algorithms. In the simple case $p = 1$, the variance explained by the component $y = Az$ is

$$\mathrm{Var}[y] = z^T A^T A z.$$

When z corresponds to the first principal loading vector, the variance is $\mathrm{Var}[y] = \sigma_1^2$, with σ_1 the largest singular value of A. In the case $p > 1$, the derived components are likely to be correlated. Hence, summing up the variance explained individually by each of the components overestimates the variance explained simultaneously by all the components. This motivates the notion of *adjusted variance* proposed by Zou et al. [ZHT06]. The adjusted variance of the p components $Y = AZ$ is defined as

$$\mathrm{AdjVar}[Y] = \mathrm{Tr}(R^2),$$

where $Y = QR$ is the QR decomposition of the component matrix Y, i.e., $Q \in \mathrm{St}(p, m)$ and R is an p-by-p upper triangular matrix.

5.4.2 Results on random test problems

The sparse PCA algorithms are compared on random data matrices $A \in \mathbf{R}^{m \times n}$ generated according to a Gaussian distribution, with zero mean and unit variance.

Trade-off curves

Let us first compare the single-unit algorithms, which provide a unit-norm sparse loading vector $z \in \mathbf{R}^n$. We first plot the variance explained by the extracted component against the cardinality of the resulting loading vector z. For each algorithm, the sparsity-inducing parameter is incrementally increased to obtain loading vectors z with a cardinality that decreases from n to 1. The results displayed in Figure 5.1 are averages of computations on 100 random matrices with dimensions $m = 100$ and $n = 300$. The considered sparse PCA methods aggregate in two groups: GPower$_{\ell_1}$, GPower$_{\ell_0}$, Greedy and rSVD$_{\ell_0}$ outperform the SPCA and the rSVD$_{\ell_1}$ approaches. It seems that these latter methods perform worse because of the ℓ_1 penalty term used in them. If one, however, post-processes the active part of z according to (5.37), as we do in GPower$_{\ell_1}$, all sparse PCA methods reach the same performance.

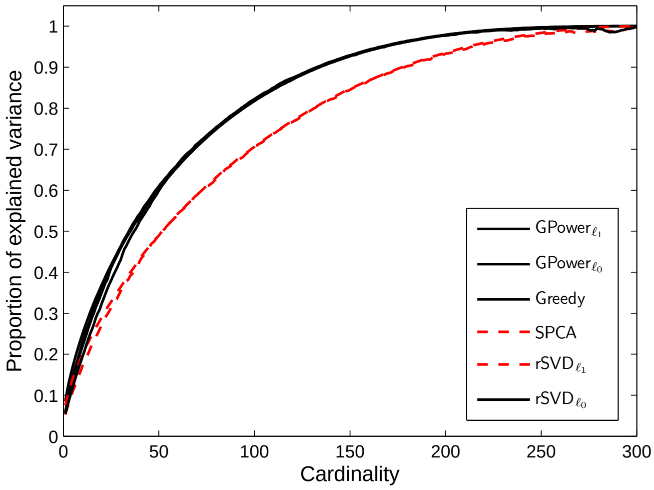

Figure 5.1: Trade-off curves between explained variance and cardinality. The vertical axis is the ratio $\frac{\mathrm{Var}[y_{\mathrm{sPCA}}]}{\mathrm{Var}[y_{\mathrm{PCA}}]}$, where y_{sPCA} is a components obtained by sparse PCA and y_{PCA} is the first principal component. The considered algorithms aggregate in two groups: GPower$_{\ell_1}$, GPower$_{\ell_0}$, Greedy and rSVD$_{\ell_0}$ (top curve), and SPCA and rSVD$_{\ell_1}$ (bottom curve). For a fixed cardinality value, the methods of the first group explain more variance. Post-processing algorithms SPCA and rSVD$_{\ell_1}$ with equation (5.37), results, however, in the same performance as the other algorithms.

Controlling sparsity with γ

Among the considered methods, the greedy approach is the only one to directly control the cardinality of the solution, i.e., the desired cardinality is an input of the algorithm. The other methods require a parameter controlling the trade-off between variance and cardinality. Increasing this parameter leads to solutions with smaller cardinality, but the resulting number of nonzero elements can not be precisely predicted. In Figure 5.2, we plot the average relationship between the parameter γ and the resulting cardinality of the loading vector z for the two algorithms GPower$_{\ell_1}$ and GPower$_{\ell_0}$. In view of (5.9) (resp. (5.14)), the entries i of the loading vector z obtained by the GPower$_{\ell_1}$ algorithm (resp. the GPower$_{\ell_0}$ algorithm) satisfying

$$\|a_i\|_2 \leq \gamma \qquad (\text{resp. } \|a_i\|_2^2 \leq \gamma) \tag{5.40}$$

have to be zero. Taking into account the distribution of the norms of the columns of A, this provides for every γ a theoretical upper bound on the expected cardinality of the resulting vector z.

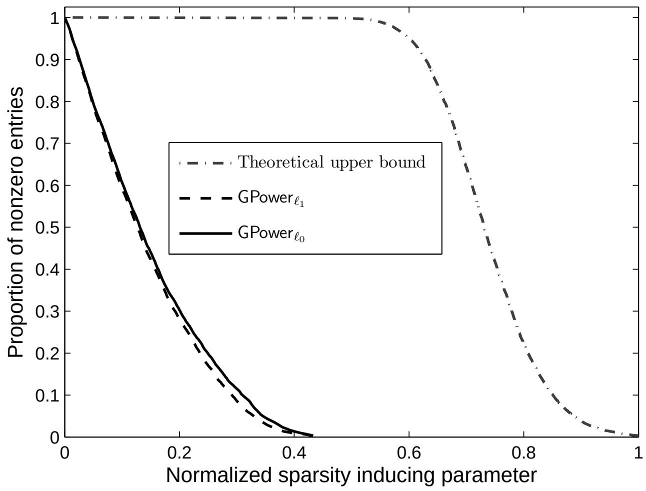

Figure 5.2: Dependence of cardinality on the value of the sparsity-inducing parameter γ. In case of the GPower$_{\ell_1}$ algorithm, the horizontal axis shows $\frac{\gamma}{\|a_{i^*}\|_2}$, whereas for the GPower$_{\ell_0}$ algorithm, we use $\frac{\sqrt{\gamma}}{\|a_{i^*}\|_2}$. The theoretical upper bound is therefore identical for both methods. The plots are averages based on 100 test problems of size $m = 100$ and $n = 300$.

"Greedy" versus the rest

The considered sparse PCA methods feature different empirical computational complexities. In Figure 5.3, we display the average time required by the sparse PCA algorithms to extract one sparse component from Gaussian matrices of dimensions $m = 100$ and $n = 300$. One immediately notices that the greedy method slows down significantly as cardinality increases, whereas the speed of the other considered algorithms does not depend on cardinality. Since on average "Greedy" is much slower than the other methods, even for low cardinalities, we discard it from all following numerical experiments.

Computational time

In Tables 5.3 and 5.4 we compare the speed of the remaining algorithms. Table 5.3 deals with problems with a fixed aspect ratio $\frac{n}{m} = 10$, whereas in Table 5.4, m is fixed at 500, and exponentially increasing values of n are considered. For the GPower$_{\ell_1}$ method, the sparsity-inducing parameter γ was set to 10% of the upper bound $\gamma_{\max} = \|a_{i^*}\|_2$. For the GPower$_{\ell_0}$ method, γ was set to 1% of $\gamma_{\max} = \|a_{i^*}\|_2^2$ in order to aim for solutions of comparable cardinalities (see (5.40)). These two parameters have also been used for the rSVD$_{\ell_1}$ and the rSVD$_{\ell_0}$ methods, respectively. Concerning SPCA, the sparsity parameter has been chosen by trial and error to get, on average, solutions with similar cardinalities as obtained by the other methods. The values displayed in Tables 5.3 and 5.4 correspond to the average running times of the algorithms on 100 test instances for each problem size. In both tables, the new methods GPower$_{\ell_1}$ and GPower$_{\ell_0}$ are the fastest. The difference in speed between GPower$_{\ell_1}$ and GPower$_{\ell_0}$ results from different approaches to fill the active part of z: GPower$_{\ell_1}$ requires

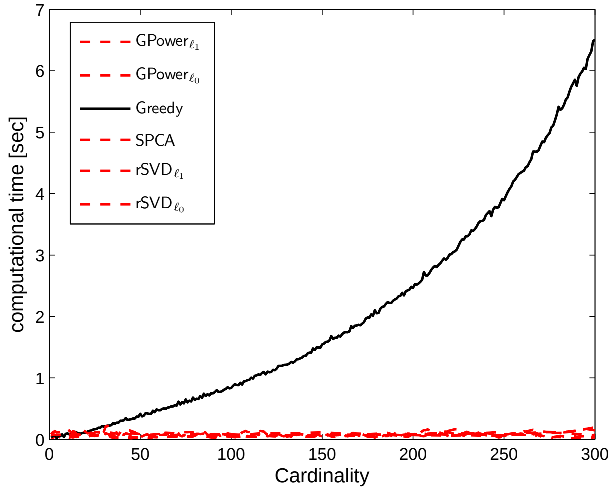

Figure 5.3: The computational complexity of "Greedy" grows significantly if it is set out to output a loading vector of increasing cardinality. The speed of the other methods is unaffected by the cardinality target.

to compute a rank-one approximation of a submatrix of A (see equation (5.37)), whereas the explicit solution (5.12) is available to GPower$_{\ell_0}$. The linear complexity of the algorithms in the problem size n is clearly visible in Table 5.4.

$m \times n$	100×1000	250×2500	500×5000	750×7500	1000×10000
GPower$_{\ell_1}$	0.10	0.86	2.45	4.28	5.86
GPower$_{\ell_0}$	0.03	0.42	1.21	2.07	2.85
SPCA	0.24	2.92	14.5	40.7	82.2
rSVD$_{\ell_1}$	0.21	1.45	6.70	17.9	39.7
rSVD$_{\ell_0}$	0.20	1.33	6.06	15.7	35.2

Table 5.3: Average computational time for the extraction of one component (in seconds).

$m \times n$	500×1000	500×2000	500×4000	500×8000	500×16000
GPower$_{\ell_1}$	0.42	0.92	2.00	4.00	8.54
GPower$_{\ell_0}$	0.18	0.42	0.96	2.14	4.55
SPCA	5.20	7.20	12.0	22.6	44.7
rSVD$_{\ell_1}$	1.20	2.53	5.33	11.3	26.7
rSVD$_{\ell_0}$	1.09	2.26	4.85	10.5	24.6

Table 5.4: Average computational time for the extraction of one component (in seconds).

Different convergence mechanisms

Figure 5.4 illustrates how the trade-off between explained variance and sparsity evolves in the time of computation for the two methods GPower$_{\ell_1}$ and rSVD$_{\ell_1}$. In case of the GPower$_{\ell_1}$

algorithm, the initialization point (5.39) provides a good approximation of the final cardinality. This method then works on maximizing the variance while keeping the sparsity at a low level throughout. The rSVD$_{\ell_1}$ algorithm, in contrast, works in two steps. First, it maximizes the variance, without enforcing sparsity. This corresponds to computing the first principal component and requires thus a first run of the algorithm with random initialization and a sparsity-inducing parameter set at zero. In the second run, this parameter is set to a positive value and the method works to rapidly decrease cardinality at the expense of only a modest decrease in explained variance. So, the new algorithm GPower$_{\ell_1}$ performs faster primarily because it combines the two phases into one, simultaneously optimizing the trade-off between variance and sparsity.

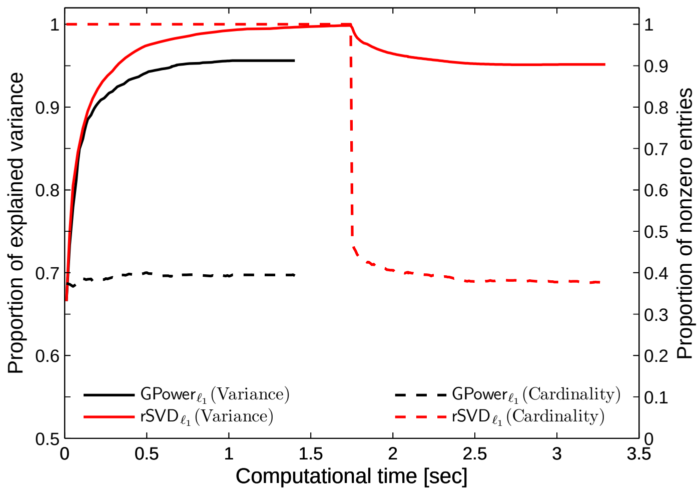

Figure 5.4: Evolution of the variance (solid lines and left axis) and cardinality (dashed lines and right axis) in time of computation for the methods GPower$_{\ell_1}$ and rSVD$_{\ell_1}$ on a test problem with $m = 250$ and $n = 2500$. The vertical axis is the ratio $\frac{\mathrm{Var}[y_{\mathrm{sPCA}}]}{\mathrm{Var}[y_{\mathrm{PCA}}]}$, where the component y_{sPCA} is obtained by sparse PCA and y_{PCA} is the first principal component. The rSVD$_{\ell_1}$ algorithm first solves unconstrained PCA, whereas GPower$_{\ell_1}$ immediately optimizes the trade-off between variance and sparsity.

Extracting a couple of components

Similar numerical experiments, which include the methods GPower$_{\ell_1,p}$ and GPower$_{\ell_0,p}$, have been conducted for the extraction of more than one component. A deflation scheme is used by the non-block methods to sequentially compute p components. These experiments lead to similar conclusions as in the single-unit case, i.e., the methods GPower$_{\ell_1}$, GPower$_{\ell_0}$, GPower$_{\ell_1,p}$, GPower$_{\ell_0,p}$ and rSVD$_{\ell_0}$ outperform the SPCA and rSVD$_{\ell_1}$ approaches in terms of variance explained at a fixed cardinality. Again, these last two methods can be improved by post-processing the resulting loading vectors with Algorithm 6, as it is done for GPower$_{\ell_1,p}$. The average running times for problems of various sizes are listed in Table 5.5. The new power-like methods are significantly faster on all instances.

$m \times n$	50×500	100×1000	250×2500	500×5000	750×7500
GPower$_{\ell_1}$	0.22	0.56	4.62	12.6	20.4
GPower$_{\ell_0}$	0.06	0.17	2.15	6.16	10.3
GPower$_{\ell_1,p}$	0.09	0.28	3.50	12.4	23.0
GPower$_{\ell_0,p}$	0.05	0.14	2.39	7.7	12.4
SPCA	0.61	1.47	13.4	48.3	113.3
rSVD$_{\ell_1}$	0.30	1.15	7.92	37.4	97.4
rSVD$_{\ell_0}$	0.28	1.10	7.54	34.7	85.7

Table 5.5: Average computational time for the extraction of $p = 5$ components (in seconds).

5.5 Analysis of gene expression data

In this section, the discussed algorithms for sparse PCA are used to analyze the four breast cancer cohorts described in Table 3.1. For consistency of comparison with PCA and ICA, we follow the methodology proposed in Section 3.3 and used in Section 4.7 to evaluate ICA against PCA. Ten components are thus inferred by the algorithms from each data set. Let us however first validate the observations made in the previous Section 5.4 on random test problems.

Trade-off curves

Figure 5.5 plots the proportion of adjusted variance versus the cardinality for the "Vijver" data set. The other data sets have similar plots. As for the random test problems, this performance criterion does not discriminate among the different algorithms. All methods have in fact the same performance, provided that the SPCA and rSVD$_{\ell_1}$ approaches are used with post-processing by Algorithm 6.

Computational time

The average computational time required by the sparse PCA algorithms on each data set is displayed in Table 5.6. The indicated times are averages on all the computations performed to obtain cardinality ranging from n down to 1.

Biological significance

Although most papers on sparse PCA validate their results on gene expression data (e.g., [ZHT06, AEJL07, SH08]), they essentially provide trade-off curves between variance and cardinality, without deeply analyzing the obtained components from a biological perspective. In this thesis, we also evaluate the sparse PCA methodology in terms of biological significance.

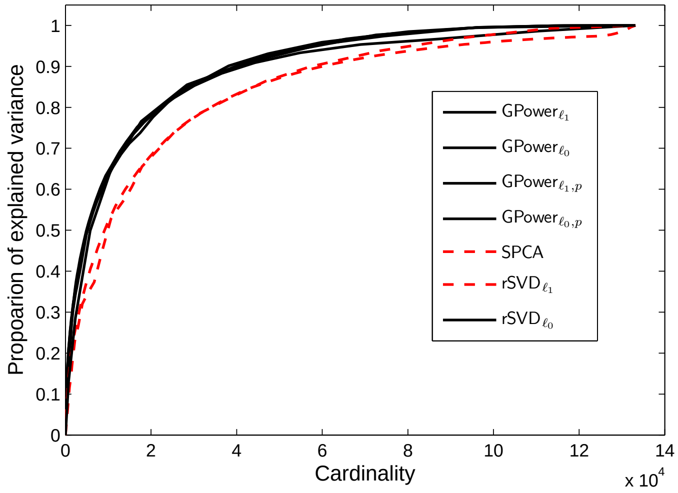

Figure 5.5: Trade-off curves between explained variance and cardinality (case of the "Vijver" data). The vertical axis is the ratio $\frac{\mathrm{AdjVar}[Y_{\mathrm{sPCA}}]}{\mathrm{AdjVar}[Y_{\mathrm{PCA}}]}$, where the components Y_{sPCA} are obtained by sparse PCA and Y_{PCA} are the dominant principal components.

	Vijver	Wang	Naderi	JRH-2
GPower$_{\ell_1}$	7.72	6.96	2.15	2.69
GPower$_{\ell_0}$	3.80	4.07	1.33	1.73
GPower$_{\ell_1,p}$	5.40	4.37	1.77	1.14
GPower$_{\ell_0,p}$	5.61	7.21	2.25	1.47
SPCA	77.7	82.1	26.7	11.2
rSVD$_{\ell_1}$	46.4	49.3	13.8	15.7
rSVD$_{\ell_0}$	46.8	48.4	13.7	16.5

Table 5.6: Average computational time (in seconds).

First, Table 5.7 displays the pathway enrichment index (PEI) based on the set of 536 pathways related to cancer, while Table 5.8 is based on the set of 173 regulatory modules. The values in both tables correspond to the largest PEI obtained among all possible cardinalities. The results for PCA and ICA are given for comparison.[8] This analysis clearly indicates that the sparse PCA methods perform better than PCA and ICA in this context. Furthermore, the new GPower algorithms, and especially the block formulations, provide largest PEI for both types of biological information.

Among the pathways that are the most frequently found in the components are important estrogen signalling and breast cancer signalling pathways such as the EGFR1 and TGF-β pathways, as well as the immune-response, cell-cycle pathways (Figure 5.6). When compared to ICA (Figure 4.3), the average number of components in which these pathways are found is somewhat larger with sparse PCA.

[8]ICA is represented by the JADE algorithm. The analysis of Section 4.7 showed, in fact, that all the considered ICA algorithms reach almost the same performance in terms of PEI (Figure 4.2).

	Vijver	Wang	Naderi	JRH-2
PCA	0.0728	0.0466	0.0149	0.0690
ICA (JADE)	0.1007	0.1157	0.0597	0.0728
GPower$_{\ell_1}$	**0.1493**	0.1026	0.0728	0.1250
GPower$_{\ell_0}$	0.1250	0.1250	0.0672	0.1026
GPower$_{\ell_1,p}$	0.1418	0.1250	**0.1026**	**0.1381**
GPower$_{\ell_0,p}$	0.1362	**0.1287**	0.1007	0.1250
SPCA	0.1362	0.1007	0.0840	0.1007
rSVD$_{\ell_1}$	0.1213	0.1175	0.0914	0.0914
rSVD$_{\ell_0}$	0.1175	0.0970	0.0634	0.1063

Table 5.7: PEI based on a set of 536 cancer-related pathways.

	Vijver	Wang	Naderi	JRH-2
PCA	0.0347	0	0.0289	0.0405
ICA (JADE)	0.1040	0.0925	0.0405	0.1040
GPower$_{\ell_1}$	0.1850	0.0867	0.0983	0.1792
GPower$_{\ell_0}$	0.1676	0.0809	0.0925	**0.1908**
GPower$_{\ell_1,p}$	**0.1908**	**0.1156**	**0.1329**	0.1850
GPower$_{\ell_0,p}$	0.1850	0.1098	**0.1329**	0.1734
SPCA	0.1734	0.0925	0.0809	0.1214
rSVD$_{\ell_1}$	0.1387	0.0809	0.1214	0.1503
rSVD$_{\ell_0}$	0.1445	0.0867	0.0867	0.1850

Table 5.8: PEI based on a set of 173 motif-regulatory gene sets.

Finally, Figure 5.7 displays the association of the components with clinical data. Considered are the components obtained with the sparsity parameter γ that led to the PEI reported in Table 5.7, which is the largest among all possible cardinalities. Overall, when compared with Figure 4.4, it turns out that sparse PCA provides slightly less components to be correlated with these phenotypes than ICA. The associations found by sparse PCA are however very strong, statistically speaking. Specific relationships between pathways and phenotypes previously identified by ICA (in Section 4.7) are again revealed by sparse PCA. For instance, components strongly correlated with the ER status often map to CR immune response pathway in all breast cancer cohorts. The association found by ICA between the EMT pathway and the grade is also identified by sparse PCA in the three studies where grade information is available.

To summarize, the components inferred by sparse PCA map to a large number of pathways. They are furthermore strongly associated with phenotypes. Hence, sparse PCA seems very promising for analyzing gene expression data. In a deeper study, we expect sparse PCA to identify novel associations between pathways and phenotypes, so far unseen by PCA and ICA.

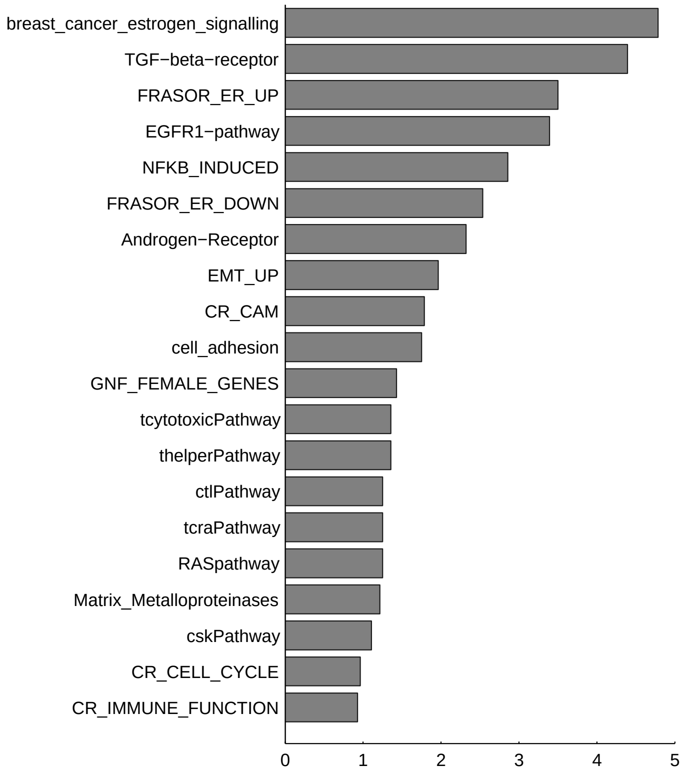

Figure 5.6: Twenty of the most frequently mapped pathways by sparse PCA. The scores give the average number of components in which the pathway is mapped.

5.6 Summary

This chapter is devoted to the maximization of convex (and not necessarily smooth) functions on compact sets. The considered problems deal with single-unit and block formulations of sparse PCA, aimed at extracting a single sparse dominant principal component of a data matrix, or more components at once, respectively. While the initial formulations involve nonconvex functions, and are therefore computationally intractable, they are rewritten into the form of an optimization problem involving maximization of a convex function on a compact set, being either a unit Euclidean sphere or the Stiefel manifold. This structure allows for the design and iteration complexity analysis of a simple gradient scheme which applied to our sparse PCA setting results in four new algorithms for computing sparse principal components of a matrix $A \in \mathbf{R}^{m \times n}$. The proposed algorithms compute a locally optimal solution of the sparse PCA problem, which inherently is of combinatorial nature. They appear to be faster if either the objective function or the feasible set are strongly convex, which holds in the single-unit case and can be enforced in the block case. Furthermore, the dimension of the feasible sets does not depend on n but on m and on the number p of components to be extracted. This is a highly desirable property if $m \ll n$. Applied on gene expression data, these algorithms provide components that deliver a rich biological interpretation.

The results of this chapter have been submitted for publication in the *Journal of Machine*

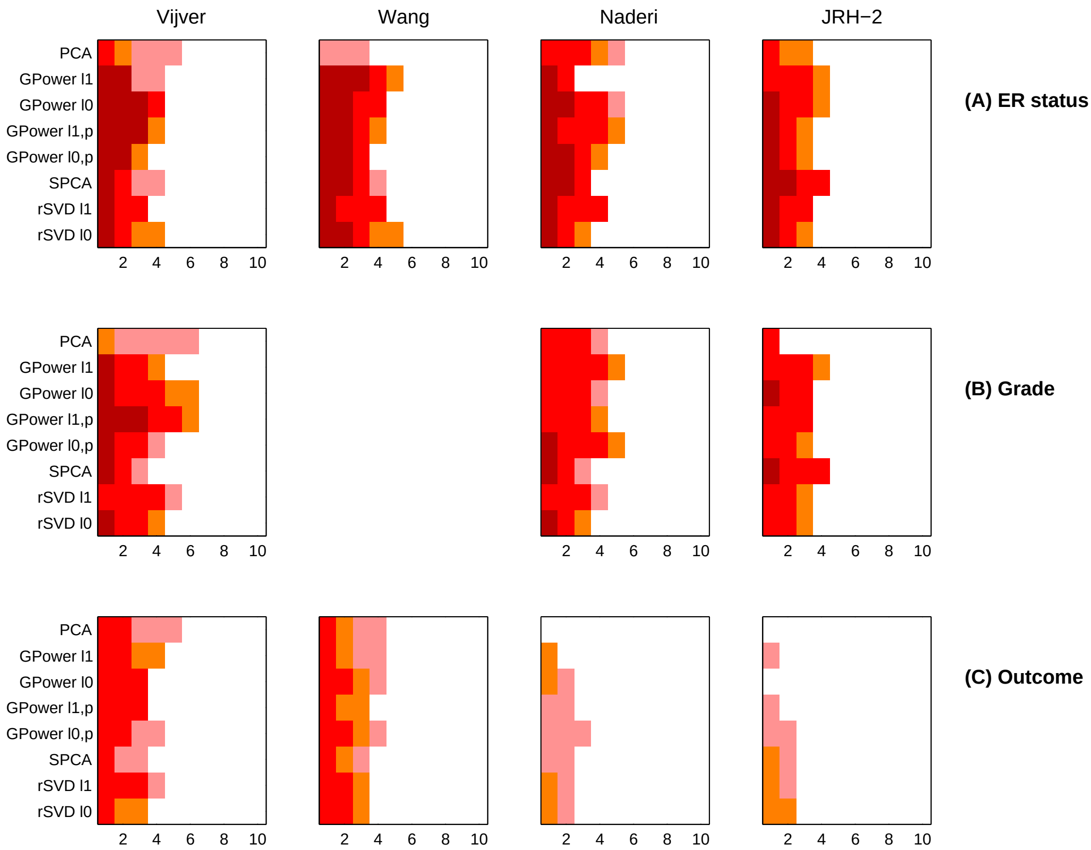

Figure 5.7: Heatmaps of association between components and breast cancer phenotypes. For each data set and each method, ten p-values are represented that assess the strength of association between each component and a phenotype. Color-code: p-value $< 10^{-10}$ (dark red), p-value < 0.001 (red), p-value < 0.01 (orange), p-value < 0.05 (pink) and p-value > 0.05 (white). For Wang's cohort, grade information is unavailable

Learning Research [JNRS08].

Chapter 6

Optimization over low-rank positive semidefinite matrices and its application to sparse PCA

In the present chapter, we focus on the optimization problem

$$\begin{array}{ll} \min_{X \in \mathbf{S}^n} & f(X) \\ \text{s.t.} & \operatorname{Tr}(C_i X) = b_i,\ C_i \in \mathbf{S}^n, b_i \in \mathbf{R},\ i = 1, \ldots, k, \\ & X \succeq 0, \end{array} \tag{P$_3$}$$

where f is a smooth function and $\mathbf{S}^n$ is the set of the symmetric matrices of $\mathbf{R}^{n \times n}$. Problem (P$_3$) is convex provided that the objective function f is convex. This assumption is however not required by the forthcoming optimization method, which computes then a local solution.

Under certain circumstances, problem (P$_3$) presents a low-rank solution, i.e., a solution X^* such that

$$\operatorname{rank}(X^*) = r \ll n.$$

This situation is often observed for convex relaxations of combinatorial optimization problems – such as the convex relaxations of the sparse PCA problem derived in the sequel – , which expand the dimension of the search space to optimize over the set of symmetric positive semidefinite matrices of the size of the original problem. To a combinatorial problem of dimension n corresponds then a convex relaxation of dimension $\mathcal{O}(n^2)$. Interestingly, these relaxations are usually *tight*, i.e., exact, for rank-one matrices and one can reasonably expect the existence of low-rank solutions.

Even when convexity significantly reduces the complexity of the problem, searching the relaxed solution in a space of dimension $\mathcal{O}(n^2)$ is still an infeasible task for large-scale problems. Convex relaxations are therefore mainly introduced as a tool to obtain lower and upper bounds on the problem of interest. Solving the relaxed problem would however provide a close approximation to the solution of the original problem.

The optimization method discussed in this chapter imposes a *low-rank* constraint on the solution in order to make a direct computation of a relaxed solution tractable even for large problems. More precisely, the positive semidefinite matrix X is parameterized as the product $X = WW^T$ where the number of independent columns of $W \in \mathbf{R}_*^{n\times l}$ fixes the rank of X. Problem (P_3) is then solved in terms of the new variable W and a space of dimension $O(nl)$ has to be searched. In the case of problems with low-rank solutions, the dimension l at which the problem should be solved remains much smaller than n.

The new optimization problem, defined in terms of the variable W, is invariant by right multiplication of W with an orthogonal matrix. This symmetry suggests to introduce a quotient manifold structure in the optimization problem.

This chapter is organized as follows. Convex relaxations for sparse PCA are first derived (Section 6.1). We then propose a method for solving (P_3) based on low-rank matrices (Section 6.2). The efficiency of the method is evaluated on the resolution of convex relaxations of sparse PCA as well as of the maximal cut of a graph, which is also a problem of combinatorial nature (Section 6.3). The resulting algorithms for sparse PCA are finally applied on breast cancer gene expression data (Section 6.4).

6.1 Convex relaxations of sparse PCA

Consider the sparse PCA formulation

$$\max_{z\in\mathcal{S}^{n-1}} z^T A^T A z - \gamma\|z\|_0, \tag{6.1}$$

that we have encountered in Section 5.1.1 and where the sparsity-inducing parameter is nonnegative, $\gamma \geq 0$. Finding the optimal pattern of nonzero elements in a loading vector $z \in \mathbf{R}^n$ is a problem of combinatorial complexity. Recently, two convex relaxations have been derived that require to minimize nonlinear convex functions on the *spectahedron*, the convex set of symmetric positive semidefinite matrices with unit trace, i.e.,

$$\mathcal{SP} = \{X \in \mathbf{S}^n | X \succeq 0, \mathrm{Tr}(X) = 1\}.$$

6.1.1 First convex relaxation

The authors of [AEJL07] relax problem (6.1) in two steps. First, a convex feasible set is obtained by lifting the unit-norm vector variable z into a matrix variable Z that belongs to the spectahedron,

$$\begin{array}{ll} \max\limits_{Z\in\mathbf{S}^n} & \mathrm{Tr}(A^TAZ) - \gamma\|Z\|_0 \\ \text{s.t.} & \mathrm{Tr}(Z) = 1, \\ & Z \succeq 0. \end{array} \tag{6.2}$$

The relaxation (6.2) is tight for rank-one matrices. In such cases, the vector variable z in (6.1) is related to the matrix variable Z according to $Z = zz^T$. Then, for problem (6.2) to be convex, the cardinality penalty is replaced by a convex l_1 penalty,

$$\begin{array}{ll}\max\limits_{Z\in\mathbf{S}^n} & \mathrm{Tr}(A^TAZ)-\gamma\sum_{i,j}|Z_{ij}| \\ \text{s.t.} & \mathrm{Tr}(Z)=1, \\ & Z\succeq 0.\end{array} \tag{6.3}$$

Finally, a smooth approximation to (6.3) is obtained by replacing the absolute value function by a close differentiable approximation. For instance, the function $h_\kappa(x)=\sqrt{x^2+\kappa^2}$ with $x,\kappa\in\mathbf{R}$ is smooth and approaches the absolute value of x as κ decreases. A too small value for the smoothing parameter κ might however lead to ill-conditioned Hessians and thus to numerical problems.

The problem

$$\begin{array}{ll}\max\limits_{Z\in\mathbf{S}^n} & \mathrm{Tr}(A^TAZ)-\gamma\sum_{i,j}h_\kappa(Z_{ij}) \\ \text{s.t.} & \mathrm{Tr}(Z)=1, \\ & Z\succeq 0,\end{array} \tag{6.4}$$

which maximizes a concave function on a convex set, is convex and fits within the framework (P_3).

6.1.2 Second convex relaxation

As illustrated in Section 5.1.1, problem (6.1) is equivalently rewritten in the form

$$\max_{x\in\mathcal{S}^{m-1}}\sum_{i=1}^n[(a_i^Tx)^2-\gamma]_+. \tag{6.5}$$

Again, the vector x is lifted into a matrix X of the spectahedron, which leads to the problem

$$\begin{array}{ll}\max\limits_{X\in\mathbf{S}^m} & \sum_{i=1}^n[a_i^TXa_i-\gamma]_+ \\ \text{s. t.} & \mathrm{Tr}(X)=1, \\ & X\succeq 0,\end{array} \tag{6.6}$$

that is equivalent to (6.5) in case of rank-one matrices $X=xx^T$. Problem (6.6) maximizes a convex function and is thus nonconvex. Nevertheless, as shown by d'Aspremont et al. [ABE07], whenever restricted to the subset of rank-one matrices, the convex objective in (6.6) equals the concave function

$$f(X)=\sum_{i=1}^n\mathrm{Tr}[X^{\frac{1}{2}}(a_i^Ta_i-\gamma I_m)X^{\frac{1}{2}}]_+, \tag{6.7}$$

where the function $\mathrm{Tr}[X]_+$ denotes the sum of the positive eigenvalues of X. A convex relaxation of (6.1) is thus provided by

$$\begin{array}{ll}\max\limits_{X\in\mathbf{S}^m} & \sum_{i=1}^n\mathrm{Tr}[X^{\frac{1}{2}}(a_i^Ta_i-\gamma I_m)X^{\frac{1}{2}}]_+ \\ \text{s. t.} & \mathrm{Tr}(X)=1, \\ & X\succeq 0,\end{array} \tag{6.8}$$

that is tight in case of rank-one solutions. We are not aware of any smoothing method that would preserve the convexity of the only piecewise smooth objective function in (6.8). Although this might be abusive, we will apply the forthcoming smooth optimization method in this non-smooth context. Interestingly, the objective in (6.8) is a *spectral* function, i.e., a function of a symmetric matrix X that depends only on the eigenvalues of X.

6.2 Optimization over low-rank positive semidefinite matrices

We propose in this section an approach for solving the problem

$$\begin{array}{ll} \min\limits_{X \in \mathbf{S}^n} & f(X) \\ \text{s.t.} & \operatorname{Tr}(C_i X) = b_i,\ C_i \in \mathbf{S}^n, b_i \in \mathbf{R},\ i = 1, \dots, k, \\ & X \succeq 0, \end{array} \tag{P$_3$}$$

that is able to deal with a large dimension n once the following assumptions hold.

Assumption 6.2.1 *Problem* (P$_3$) *presents a low-rank solution* X^**, i.e.,*

$$\operatorname{rank}(X^*) = r \ll n.$$

Assumption 6.2.2 *The symmetric matrices* C_i *satisfy*

$$C_i C_j = 0,$$

for any $i, j \in \{1, \dots, k\}$ *such that* $i \neq j$.

Assumption 6.2.2 is fulfilled, e.g., by the spectahedron

$$\mathcal{SP} = \{X \in \mathbf{S}^n | X \succeq 0, \operatorname{Tr}(X) = 1\},$$

and the elliptope[1]

$$\mathcal{E} = \{X \in \mathbf{S}^n | X \succeq 0, \operatorname{diag}(X) = 1_n\}, \tag{6.9}$$

where 1_n is the vector of all ones. Although the function f is often convex in the considered applications – in which case (P$_3$) is a convex problem –, this assumption is not required by the proposed optimization method, which identifies then a local solution of (P$_3$).

Assumption 6.2.1 suggests to factor the optimization variable X as

$$X = WW^T, \tag{6.10}$$

with $W \in \mathbf{R}^{n \times l}$ and $l \ll n$, and to consider the nonconvex problem

$$\begin{array}{ll} \min\limits_{W \in \mathbf{R}^{n \times l}} & f(WW^T) \\ \text{s.t.} & \operatorname{Tr}(W^T C_i W) = b_i,\ C_i \in \mathbf{S}^n, b_i \in \mathbf{R},\ i = 1, \dots, k, \end{array} \tag{6.11}$$

[1] The elliptope is also known as the set of correlation matrices.

which searches a space of dimension $\mathcal{O}(nl)$. The parameter l should ideally equal the rank r, which is usually unknown. The proposed algorithm for solving (P_3) combines thus a method that finds a local minimizer W of (6.11) with an approach that increments l until a sufficient condition is satisfied for W to provide the solution WW^T of (P_3).

A further potential difficulty of problem (6.11) is that the solutions are not isolated. For any solution W and any orthogonal matrix $Q \in \mathcal{O}(l)$, the matrix WQ also provides a solution. In other words, problem (6.11) is invariant by right multiplication of the search variable with an orthogonal matrix. This issue is not harmful for simple gradient schemes but it greatly affects the convergence of second-order methods (see, e.g., [AMS08] and [AILH09]). In order to take into account the inherent symmetry of the solution, the algorithm developed in this chapter does not optimize over the Euclidean space $\mathbf{R}^{n\times l}$. Instead, one considers a search space, whose points are the equivalence classes $\{WQ | Q \in \mathbf{R}^{l\times l}, Q^TQ = I_l\}$. The minimizers of (6.11) can be isolated in that *quotient* space.

The idea of reformulating a convex problem into a nonconvex one by factorization of the matrix unknown is not new and was investigated by Burer and Monteiro [BM03] for solving semidefinite programs (SDP). While the setup considered in [BM03] is general but restricted to gradient methods, we further exploit the particular structure of the equality constraints (Assumption 6.2.2) and propose second-order methods that lead to a descent algorithm with guaranteed superlinear convergence. The authors of [GP07] also exploit the factorization (6.10) to efficiently solve optimization problems that are defined on the elliptope (6.9). Whereas the algorithms in [GP07] evolve on the *Cholesky manifold* – a submanifold of $\mathbf{R}^{n\times l}$ whose intersection with almost all equivalence classes is a singleton– the methods proposed here work conceptually on the entire quotient space and numerically in $\mathbf{R}^{n\times l}$, using the machinery of Riemannian submersions.

In the following sections, we derive conditions for an optimizer of (6.11) to represent a solution of the original problem (P_3) (Section 6.2.1). A meta-algorithm for solving (P_3) based on the factorization (6.10) is built upon these theoretical results (Section 6.2.2). We then describe the geometry of the underlying quotient manifold and propose an algorithm for solving (6.11) based on second-order derivative information (Section 6.2.3).

6.2.1 Optimality conditions

Optimality conditions of both problems (P_3) and (6.11) are now derived and analyzed. They provide theoretical insight about the rank l at which (6.11) should be solved as well as conditions for an optimizer of (6.11) to represent a solution of the original problem (P_3).

First-order optimality conditions

Lemma 6.2.3 *A symmetric matrix* $X \in \mathbf{S}^n$ *solves* (P_3) *if and only if there exist a vector* $\sigma \in \mathbf{R}^k$ *and a symmetric matrix* $S \in \mathbf{S}^n$ *such that the following holds,*

$$\begin{aligned} &\mathrm{Tr}(C_i X) = b_i, \\ &X \succeq 0, \\ &S \succeq 0, \\ &SX = 0, \\ &S = \nabla f(X) - \textstyle\sum_{i=1}^k \sigma_i C_i. \end{aligned} \tag{6.12}$$

Proof. These are the first-order optimality conditions of (P_3). □

The first-order optimality conditions (6.12) are necessary and sufficient in case of convex optimization problems [BV04]. In the case of a nonconvex objective function f, we consider any point that satisfies these optimality conditions as a solution of (P_3). Only the (local) minimizers are actually stable for the optimization method proposed in the sequel, which is a descent algorithm for f.

Lemma 6.2.4 *If* W *is a local optimum of (6.11), then there exists a vector* $\lambda \in \mathbf{R}^k$ *such that*

$$\begin{aligned} &\mathrm{Tr}(W^T C_i W) = b_i, \\ &(\nabla f(WW^T) - \textstyle\sum_{i=1}^k \lambda_i C_i)W = 0. \end{aligned} \tag{6.13}$$

If the $\{C_i W\}_{i=1,\dots,k}$ *are linearly independent, the vector* λ *is unique.*

Proof. These are the first-order optimality conditions of (6.11). □

Given a local minimizer W of (6.11), one readily notices that all but one condition of Lemma 6.2.3 hold for the symmetric positive semidefinite matrix WW^T. Comparison of Lemma 6.2.3 and Lemma 6.2.4 provides thus the following relationship between the problems (6.11) and (P_3).

Theorem 6.2.5 *A local minimizer* W *of problem (6.11) provides the solution* WW^T *of problem* (P_3) *if the matrix*

$$S_W \stackrel{def}{=} \nabla f(WW^T) - \sum_{i=1}^k \lambda_i C_i \tag{6.14}$$

is positive semidefinite for the Lagrangian multipliers λ_i *that satisfy (6.13).*

Proof. Check the conditions of Lemma 6.2.3 for the tuple $\{X, S, \sigma\} = \{WW^T, S_W, \lambda\}$. □

Under Assumption 6.2.2, the Lagrangian multipliers in (6.13) have the closed-form expression

$$\lambda_i = \frac{\mathrm{Tr}(W^T C_i \nabla f(WW^T) W)}{\mathrm{Tr}(W^T C_i^2 W)}. \tag{6.15}$$

Hence, a closed-form expression is available for the dual matrix S_W in (6.14) at an optimizer W of (6.11).

Second-order optimality conditions

Let $\mathcal{L}(W, \lambda)$ denote the Lagrangian of the nonconvex problem (6.11),

$$\mathcal{L}(W, \lambda) \stackrel{\text{def}}{=} f(WW^T) - \sum_{i=1}^{k} \lambda_i (\operatorname{Tr}(W^T C_i W) - b_i).$$

The optimality conditions (6.13) can be rewritten in the form

$$\nabla_\lambda \mathcal{L}(W, \lambda) = 0 \quad \text{and} \quad \nabla_W \mathcal{L}(W, \lambda) = 0.$$

In the following, we consider the Lagrangian multipliers λ_i to be given by (6.15).

Lemma 6.2.6 *For a local minimizer $W \in \mathbf{R}^{n \times l}$ of (6.11), it holds that*

$$\operatorname{Tr}(\eta^T \mathrm{D}_W \nabla_W \mathcal{L}(W, \lambda)[\eta]) \geq 0$$

for any matrix $\eta \in \mathbf{R}^{n \times l}$ that satisfies

$$\operatorname{Tr}(\eta^T C_i W) = 0, \; i = 1, \ldots, k. \tag{6.16}$$

Proof. These are the second-order optimality conditions of (6.11). □

Lemma 6.2.7 *For any matrix $\eta \in \mathbf{R}^{n \times l}$ such that $W\eta^T = 0$, the following equality holds*

$$\frac{1}{2} \operatorname{Tr}(\eta^T \mathrm{D}_W \nabla_W \mathcal{L}(W, \lambda)[\eta]) = \operatorname{Tr}(\eta^T S_W \eta).$$

Proof. By noting that $\nabla_W \mathcal{L}(W, \lambda) = 2 S_W W$, one has

$$\frac{1}{2} \operatorname{Tr}(\eta^T \mathrm{D}_W \nabla_W \mathcal{L}(W, \lambda)[\eta]) =$$

$$\operatorname{Tr}(\eta^T S_W \eta) + \operatorname{Tr}(\eta^T \mathrm{D}_W(\nabla f(WW^T))[\eta] W) - \sum_{i=1}^{k} \mathrm{D}_W \lambda_i[\eta] \operatorname{Tr}(\eta^T C_i W),$$

where the two last terms cancel out by virtue of the condition $W\eta^T = 0$. □

Theorem 6.2.8 *A local minimizer W of problem (6.11) provides the solution $X = WW^T$ of problem* (P_3) *if it is rank deficient.*

Proof. For the minimizer $W \in \mathbf{R}^{n \times l}$ to span an r-dimensional subspace in $\mathbf{R}^n$ (with $l > r$), the following factorization has to hold,

$$W = \bar{W} M^T,$$

with the full-rank matrices $\bar{W} \in \mathbf{R}_*^{n \times r}$ and $M \in \mathbf{R}_*^{l \times r}$. Let $M_\perp \in \mathbf{R}^{l \times (l-r)}$ be an orthogonal basis for the orthogonal complement of the column space of M, i.e., $M^T M_\perp = 0$ and $M_\perp^T M_\perp = I_{l-r}$. For any matrix $\bar{\eta} \in \mathbf{R}^{n \times (l-r)}$, the matrix $\eta = \bar{\eta} M_\perp^T$ satisfies

$$W\eta^T = 0,$$

and the conditions (6.16) hold. By virtue of Lemmas 6.2.6 and 6.2.7,

$$\mathrm{Tr}(\eta^T S_W \eta) \geq 0,$$

for all the matrices $\eta = \bar{\eta} M_\perp^T$, i.e., the matrix S_W is positive semidefinite and $X = WW^T$ is a solution of problem (P$_3$). □

Corollary 6.2.9 *In the case $l = n$, any local minimizer $W \in \mathbf{R}^{n\times n}$ of problem (6.11) provides the solution $X = WW^T$ of problem* (P$_3$).

Proof. If W is rank deficient, the matrix $X = WW^T$ is optimal for (P$_3$) by virtue of Theorem 6.2.8. Otherwise, the matrix S_W is zero because of the second condition in (6.13) and X is also optimal for (P$_3$). □

6.2.2 A meta-algorithm for solving the initial problem

The algorithm we propose for solving (P$_3$) consists in solving a sequence of nonconvex problems (6.11) of increasing dimension until the resulting local minimizer W represents a solution of the initial problem (P$_3$). Both Theorems 6.2.5 and 6.2.8 provide conditions to check this fact. When problem (6.11) is solved in a dimension l smaller than the unknown rank r, none of these conditions can be fulfilled. The dimension l is thus incremented after each resolution of (6.11). In order to ensure a monotone decrease of the objective function through the iterations, the optimization algorithm that solves (6.11) is initialized with a matrix corresponding to W with an additional zero column appended, i.e.,

$$W_0 \stackrel{\text{def}}{=} [W|0^{n\times 1}],$$

where $0^{n\times 1}$ denotes an n-by-1 vector full of zeros. Since this initialization occurs when the local minimizer $W \in \mathbf{R}^{n\times l}$ of (6.11) does not represent the solution of (P$_3$), W_0 is a saddle point of the nonconvex problem for the dimension $l+1$. This can be a critical issue for many optimization algorithms. Fortunately, in the present case, a descent direction from W_0 can be explicitly evaluated. The matrix

$$\eta \stackrel{\text{def}}{=} [0^{n\times l} \,|\, v],$$

where $0^{n\times l}$ is a zero matrix of the size of W and v is the eigenvector of S_W related to the smallest algebraic eigenvalue verifies $W_0\eta^T = 0$ and hence, by virtue of Lemma 6.2.7,

$$\frac{1}{2}\,\mathrm{Tr}(\eta^T \mathrm{D}_W \nabla_W \mathcal{L}(W_0, \lambda)[\eta]) = v^T S_W v \leq 0,$$

for the Lagrangian multipliers λ given in (6.15). All these elements lead to the meta-algorithm displayed in Algorithm 7. The parameter ε sets a threshold on the eigenvalues of S_W to decide about the nonnegativity of this matrix. ε is chosen to be 10^{-12} in our implementation.

Algorithm 7: Meta-algorithm for solving problem (P_3) [2]

input : Initial rank l_0, initial iterate $W^{(0)} \in \mathbf{R}^{n \times l_0}$ and parameter ε.
output: Solution X of problem (P_3).

```
begin
    l ⟵ l_0
    W_l ⟵ W^(0)
    stop ⟵ 0
    while stop ≠ 1 do
        Initialize an optimization scheme with W_l to find a local minimum W_l^* of (6.11)
        by exploiting a descent direction η_l if available.
        if l = l_0 and rank(W_l^*) < l then
            stop = 1
        else
            Find the smallest eigenvalue λ_min and the related eigenvector V_min of the
            matrix S_W (6.14).
            if λ_min ≥ −ε then
                stop = 1
            else
                l ⟵ l + 1
                W_l ⟵ [W_l^* | 0]
                A descent direction from the saddle point W_l is given by η_l = [0 | V_min].
    X ⟵ W_l^* W_l^{*T}
end
```

It should be mentioned that, to check the optimality for the initial problem (P_3) of a local minimizer W_l^*, the rank condition of Theorem 6.2.8 is computationally cheaper to evaluate than the nonnegativity condition of Theorem 6.2.5. Nevertheless, the rank condition does not provide a descent direction to escape saddle points. It furthermore requires to solve problem (6.11) at a dimension that is strictly greater than r, the rank of the solution of (P_3). Hence, this condition is only used at the initial rank l_0 and holds in general if l_0 is chosen larger than the unknown r. Numerically, the rank of $W_{l_0}^*$ is computed as the number of singular values that are greater than a threshold fixed at 10^{-6}. The algorithm proposed by Burer and Monteiro [BM03] exploits exclusively the rank condition of Theorem 6.2.8. Each optimization of (6.11) is therefore initialized in a random manner and the algorithm in [BM03] is not a descent algorithm.

By virtue of Corollary 6.2.9, Algorithm 7 stops at the latest once $l = n$. The numerical experiments reported in the forthcoming Section 6.3 indicate that in practice, however, the algorithm stops at a rank l that is much lower than the dimension n. If $l_0 < r$, then the algorithm stops once l equals the rank r of the solution of (P_3). These applications also illustrate that the magnitude of the smallest eigenvalue $\lambda_{\min}$ of the matrix S_W can be used

[2] A MATLAB implementation of Algorithm 7 with the manifold-based optimization method of Section 6.2.3 can be downloaded from http://www.montefiore.ulg.ac.be/~journee.

to monitor convergence. The value $|\lambda_{\min}|$ indicates in fact whether the current iterate is close to satisfying the optimality conditions (6.12). This feature is of great interest once an approximate solution to (P_3) is sufficient. The threshold ε set on $\lambda_{\min}$ controls then the accuracy of the result.

A trust-region scheme based on second-order derivative information is proposed next for computing a local minimum of (6.11). This method is provided with a convergence theory that ensures the iterates converge towards a local minimizer.

Hence, the proposed algorithm presents the following notable features. First, it converges toward the solution of problem (P_3) by ensuring a monotone decrease of the objective function. Then, the magnitude of the smallest eigenvalue of S_W provides a means to monitor the convergence. Finally, the inner problem (6.11) is solved by second-order methods featuring superlinear local convergence.

6.2.3 Inner iteration as an optimization on a quotient manifold

We now derive an optimization scheme that locally solves the nonconvex and nonlinear problem

$$\begin{array}{ll} \min\limits_{W \in \mathbf{R}^{n\times l}} & \bar{f}(W) \\ \text{s.t.} & \operatorname{Tr}(W^T C_i W) = b_i,\ C_i \in \mathbf{S}^n, b_i \in \mathbf{R},\ i = 1, \dots, k, \end{array} \tag{6.17}$$

where $\bar{f}(W) = f(WW^T)$ for some $f : \mathbf{S}^n \to \mathbf{R}$.

As previously mentioned, problem (6.17) is invariant by right-multiplication of the variable W by orthogonal matrices. The critical points of (6.17) are thus non isolated. To get rid of this symmetry, let $\mathcal{M}$ define the set of all the equivalence classes of the form

$$[W] \stackrel{\text{def}}{=} \{WQ \mid Q \in \mathbf{R}^{l\times l},\ Q^T Q = I_l\}, \tag{6.18}$$

where $W \in \mathbf{R}_*^{n\times l}$ satisfies the quadratic equality constraints in (6.17), i.e., W belongs to the manifold

$$\bar{\mathcal{M}} \stackrel{\text{def}}{=} \{W \in \mathbf{R}_*^{n\times l} \mid \operatorname{Tr}(W^T C_i W) = b_i,\ i = 1, \dots, k\},$$

which is embedded in $\mathbf{R}_*^{n\times l}$.[3] The set $\mathcal{M}$ is the *quotient* of the manifold $\bar{\mathcal{M}}$ by the orthogonal group $\mathcal{O}(l)$,

$$\mathcal{M} = \bar{\mathcal{M}}/\mathcal{O}(l).$$

It can be furthermore proven that the quotient $\mathcal{M}$ is a differentiable manifold, i.e., it is a quotient manifold.

Let us turn problem (6.17) onto the quotient manifold $\mathcal{M}$, i.e.,

$$\min_{[W]\in\mathcal{M}} \phi([W]), \tag{6.19}$$

with the function $\phi : \mathcal{M} \to \mathbf{R} : [W] \mapsto \phi([W]) = \bar{f}(W)$. The minimizers of (6.19) are isolated on the search space $\mathcal{M}$. As discussed in Section 4.3, several methods for unconstrained

[3] $\mathbf{R}_*^{n\times l}$ is the noncompact Stiefel manifold of *full-rank* matrices in $\mathbf{R}^{n\times l}$. The nondegeneracy condition is required to deal with differentiable manifolds.

optimization have been generalized to search spaces that are manifolds (see, e.g., [AMS08]). Let us now discuss the practical implementation of these algorithms in the context of quotient manifolds.

A quotient manifold is an abstract mathematical space that cannot be directly "represented" on a computer. However, any point of the quotient $\mathcal{M}$, i.e, an equivalence class $[W]$ is completely characterized by any one of its element, i.e., a matrix W of the *total space* $\bar{\mathcal{M}}$, which is embedded in $\mathbf{R}_*^{n\times l}$. Hence, the quotient $\mathcal{M}$ is neatly parameterized by n-by-l matrices, which is suitable for numerical computations.

Let us now characterize another important concept: the *tangent space* to a quotient manifold. In Section 4.3, a tangent vector to a manifold has been defined by considering a smooth curve $\gamma : \mathbf{R} \to \mathcal{M} : t \mapsto \gamma(t)$ on the manifold. The tangent vector $\dot{\gamma}(0)$ is then the mapping that, given a function f, returns the derivative of f along that curve at $\gamma(0)$. These concepts are immediately transposed to the total space $\bar{\mathcal{M}}$ by considering that to any smooth curve

$$\gamma : \mathbf{R} \to \mathcal{M} : t \mapsto \gamma(t) = [W(t)]$$

on the quotient manifold corresponds a smooth curve

$$\bar{\gamma} : \mathbf{R} \to \bar{\mathcal{M}} : t \mapsto \bar{\gamma}(t) = W(t).$$

on the total space. Since the derivative of the function f along γ at $\gamma(0)$ is identical to the derivative along $\bar{\gamma}$ at $\bar{\gamma}(0)$ of the function $\bar{f}$ defined by $\bar{f}(W) = f([W])$, one can relate the tangent vectors of the quotient manifold $\mathcal{M}$ to the tangent vectors of $\bar{\mathcal{M}}$. The latter have a well-defined matrix representation,

$$T_W\bar{\mathcal{M}} = \{\eta \in \mathbf{R}^{n\times l} \mid \operatorname{Tr}(W^T C_i \eta) = 0,\ i = 1, \dots, k\}.$$

To any smooth curve on the quotient $\mathcal{M}$ correspond however infinitely many smooth curves on the total space $\mathcal{M}$, e.g., some curves are almost parallel to the equivalence classes, and some other are rather orthogonal to them. A small shift of a point $W \in \bar{\mathcal{M}}$ along its equivalence class does not modify the point $[W]$ on the quotient $\mathcal{M}$, and is thus useless for our goal of solving the optimization problem (6.19). For the sake of numerical efficiency, it appears natural to move the iterates along curves that are orthogonal to the equivalence classes. In other words, to a smooth curve on $\mathcal{M}$ passing through $[W]$, one would like to associate a curve on $\bar{\mathcal{M}}$ passing through W and that is orthogonal to the equivalence class $[W]$. This induces a decomposition of the tangent space $T_W\bar{\mathcal{M}}$ in two orthogonal subspaces, the *vertical space* $\mathcal{V}_W\mathcal{M}$ and the *horizontal space* $\mathcal{H}_W\mathcal{M}$. The vertical space $\mathcal{V}_W\mathcal{M}$ is the tangent space to the equivalence classes,

$$\mathcal{V}_W\mathcal{M} = \{W\Omega \mid \Omega \in \mathbf{R}^{l\times l},\ \Omega^T = -\Omega\}.$$

The horizontal space $\mathcal{H}_W\mathcal{M}$, on the other hand, is the orthogonal complement of $\mathcal{V}_W\mathcal{M}$ in $T_W\bar{\mathcal{M}}$. In case of the Euclidean metric $\langle \eta_1, \eta_2 \rangle_W \stackrel{\text{def}}{=} \operatorname{Tr}(\eta_1^T \eta_2)$ for any $\eta_1, \eta_2 \in T_W\bar{\mathcal{M}}$, the horizontal space corresponds to

$$\mathcal{H}_W\mathcal{M} = \{\eta \in T_W\bar{\mathcal{M}} \mid \eta^T W = W^T \eta\}. \tag{6.20}$$

Expression (6.20) results from the equality $\mathrm{Tr}(S\Omega) = 0$ that holds for any symmetric matrix S and skew-symmetric matrix Ω of compatible dimension. A unique matrix representation of the tangent space to the quotient manifold $\mathcal{M}$ is so provided by the elements of the horizontal space $\mathcal{H}_W\mathcal{M}$.

By extending the discussion of Section 4.3.2 to quotient manifolds, the gradient of a function ϕ is obtained by projecting the Euclidean gradient of the function $\bar{f}$ onto the horizontal space, i.e.,

$$\mathrm{grad}\phi([W]) = P_W(\nabla \bar{f}(W)),$$

where $P_W : \mathbf{R}^{n\times l} \to \mathcal{H}_W\mathcal{M}$ is a projection.

In order to specify precisely the projection, let $N_W\bar{\mathcal{M}}$ be the normal space to $\bar{\mathcal{M}}$ at W, i.e., the orthogonal complement of $T_W\bar{\mathcal{M}}$ in $\mathbf{R}^{n\times l}$ with respect to the chosen Euclidean metric,

$$N_W\bar{\mathcal{M}} = \{\sum_{i=1}^{k} \alpha_i C_i W \mid \alpha \in \mathbf{R}^k\}.$$

The Euclidean space $\mathbf{R}^{n\times l}$ is so uniquely divided into three mutually orthogonal subspaces,

$$\mathbf{R}^{n\times l} = \mathcal{H}_W\mathcal{M} \oplus \mathcal{V}_W\mathcal{M} \oplus N_W\bar{\mathcal{M}}.$$

We are now ready to derive a closed-form expression for the projection P_W.

Theorem 6.2.10 *Let W be a point on $\bar{\mathcal{M}}$. For a matrix $\eta \in \mathbf{R}^{n\times l}$, the projection*

$$P_W : \mathbf{R}^{n\times l} \to \mathcal{H}_W\mathcal{M}$$

is given by

$$P_W(\eta) = \eta - W\Omega - \sum_{i=1}^{k} \alpha_i C_i W,$$

where Ω is the skew-symmetric matrix that solves the Sylvester equation

$$\Omega W^T W + W^T W \Omega = W^T \eta - \eta^T W,$$

and with the coefficients

$$\alpha_i = \frac{\mathrm{Tr}(\eta^T C_i W)}{\mathrm{Tr}(W^T C_i^2 W)}.$$

Proof. Any vector $\eta \in \mathbf{R}^{n\times l}$ presents a unique decomposition

$$\eta = \eta_{\mathcal{V}_W\mathcal{M}} + \eta_{\mathcal{H}_W\mathcal{M}} + \eta_{N_W\bar{\mathcal{M}}},$$

where each element $\eta_{\mathcal{X}}$ belongs to the Euclidean space $\mathcal{X}$. The orthogonal projection $\mathcal{P}_W$ extracts the component that lies in the horizontal space,

$$P_W(\eta) = \eta - W\Omega - \sum_{i=1}^{k} \alpha_i C_i W,$$

where Ω is a skew-symmetric matrix. The parameters Ω and α are determined from the linear equations

$$W^T P_W(\eta) = P_W(\eta)^T W,$$
$$\mathrm{Tr}(W^T C_i P_W(\eta)) = 0, \quad i = 1 \dots k,$$

which are satisfied by any element of the horizontal space. □

The projection P_W provides simple formulas to compute derivatives of the function ϕ (defined on the quotient manifold) from derivatives of the function $\bar{f}$ (defined in the Euclidean space). As previously mentioned, the gradient of the function ϕ defined on the manifold corresponds to

$$\mathrm{grad}\phi([W]) = P_W(\nabla \bar{f}(W)).$$

Similarly, the Hessian of ϕ in a direction $\eta \in \mathcal{H}_W\mathcal{M}$ is given by

$$\mathrm{Hess}\phi([W])[\eta] \stackrel{\mathrm{def}}{=} \nabla_\eta \mathrm{grad}\phi([W]) = P_W(\mathrm{D}(\mathrm{grad}\phi([W]))[\eta]),$$

where ∇ is a Riemannian connection and the directional derivative $\mathrm{D}(\cdot)[\cdot]$ is performed in the Euclidean sense in $\mathbf{R}^{n\times l}$.[4]

Finally, a last ingredient required for optimizing on manifolds is a *retraction*

$$\mathcal{R}_W : \mathcal{H}_W\mathcal{M} \to \bar{\mathcal{M}},$$

that moves the current iterate $W \in \bar{\mathcal{M}}$ in a direction η (an element of the horizontal space at W) to obtain a matrix representing a new point on the manifold $\mathcal{M}$. Such a mapping is for example obtained by projecting the matrix $\bar{W} = W + \eta$ along the Euclidean space $N_W\bar{\mathcal{M}}$,

$$\mathcal{R}_W(\eta) = \bar{W} + \sum_{i=1}^{k} \alpha_i C_i \bar{W}, \tag{6.21}$$

[4] Let $\bar{\zeta}$ be a vector field on $\mathbf{R}_*^{n\times l}$, e.g., the Euclidean gradient $\nabla \bar{f}(W)$. Let ζ be the associated vector field on the quotient manifold $\mathcal{M}$, which assigns to any point $[W] \in \mathcal{M}$ the horizontal vector $\zeta = P_W\bar{\zeta}$. The Riemannian connection of ζ in a direction $\eta \in \mathcal{H}_W\mathcal{M}$ corresponds to the projection of the Euclidean directional derivative of ζ in the direction η,

$$\nabla_\eta \zeta([W]) \stackrel{\mathrm{def}}{=} P_W \mathrm{D}(\zeta)[\eta] = P_W \mathrm{D}(P_W\bar{\zeta})[\eta].$$

The directional derivative $\mathrm{D}(P_W\bar{\zeta})[\eta]$ is computed as follows,

$$\mathrm{D}(P_W(\bar{\zeta}))[\eta] = \mathrm{D}\bar{\zeta}[\eta] - \eta\Omega - W\mathrm{D}\Omega[\eta] - \sum_{i=1}^{k} \alpha_i C_i \eta - \sum_{i=1}^{k} \mathrm{D}\alpha_i[\eta] C_i W,$$

where $\mathrm{D}\Omega[\eta]$ is the solution of the Sylvester equation

$$\mathrm{D}\Omega[\eta]W^TW + W^TW\mathrm{D}\Omega[\eta] = \eta^T\bar{\zeta} - \bar{\zeta}^T\eta + W^T\mathrm{D}\bar{\zeta}[\eta] - \mathrm{D}\bar{\zeta}[\eta]^TW - \Omega(\eta^TW + W^T\eta) - (\eta^TW + W^T\eta)\Omega,$$

and

$$\mathrm{D}\alpha_i[\eta] = \frac{1}{\mathrm{Tr}(W^TC_i^2W)}(\mathrm{D}\bar{\zeta}[\eta]C_iW + \bar{\zeta}^TC_i\eta) - \frac{\mathrm{Tr}(\eta^TC_iW)}{\mathrm{Tr}(W^TC_i^2W)^2}(\eta^TC_i^2W + W^TC_i^2\eta).$$

until the quadratic equality constraints in (6.17) are satisfied. Under Assumption 6.2.2, the coefficients α_i are easily computed as the solution of the quadratic polynomial

$$\alpha_i^2 \operatorname{Tr}(\bar{W}^T C_i^3 \bar{W}) + 2\alpha_i \operatorname{Tr}(\bar{W}^T C_i^2 \bar{W}) + \operatorname{Tr}(\bar{W}^T C_i \bar{W}) = b_i, \quad i = 1, \ldots, k.$$

In case of the elliptope $\mathcal{E}$, equation (6.21) becomes

$$\mathcal{R}_W(\eta) = \operatorname{Diag}((W+\eta)(W+\eta)^T)^{-\frac{1}{2}}(W+\eta),$$

For the spectahedron $\mathcal{SP}$, the retraction (6.21) is given by

$$\mathcal{R}_W(\eta) = \frac{W+\eta}{\sqrt{\operatorname{Tr}((W+\eta)^T(W+\eta))}}.$$

In our implementation of Algorithm 7, we use the trust-region method described in [ABG07, AMS08] for solving the inner problem (6.17). As previously mentioned, this optimization method is provided with a convergence theory whose results are similar to the ones related to classical unconstrained optimization. We set the parameter θ in equation (10) of [ABG07] to one to ensure a quadratic convergence.

The complexity of this manifold-based optimization algorithm for solving problem (6.17) is dominated by the computational cost required to evaluate the objective $\bar{f}(W)$, the gradient $\nabla \bar{f}(W)$ and the directional derivative $\mathrm{D}(\nabla \bar{f}(W))[\eta]$. Hence, the costly operations are performed in the Euclidean space $\mathbf{R}^{n \times l}$, whereas all manifold-related operations, such as evaluating a metric, a projection and a retraction, are of linear complexity with the dimension n.

6.3 Numerical experiments

In this section, we evaluate the new optimization method on several tests problems. First, a common benchmark setup is provided by the SDP relaxation of the maximal cut of a graph. Then, the two convex relaxations of sparse PCA are considered. We finally address the problem of finding a "good" rank-one approximation to a positive semidefinite matrix of larger rank. This is essential to reconstruct a loading vector $z \in \mathbf{R}^n$ from a matrix solution of a convex relaxation of sparse PCA.

6.3.1 The max-cut SDP relaxation

The maximal cut of an undirected and weighted graph corresponds to the partition of the vertices in two sets such that the sum of the weights associated to the edges crossing between these two sets is the largest. Computing the maximal cut of a graph is *NP-complete*, i.e., "hard". Several convex relaxations to that problem have been proposed. The most studied one, which is the basis of a 0.878-approximation algorithm [GW95], is the following *semidefinite program* (SDP),

$$\begin{array}{ll} \min\limits_{X \in \mathbf{S}^n} & \operatorname{Tr}(AX) \\ \text{s.t.} & \operatorname{diag}(X) = 1_n, \\ & X \succeq 0, \end{array} \tag{6.22}$$

where n is the number of vertices in the graph, $A = -\frac{1}{4}L$ with L the Laplacian matrix[5] of the graph and 1_n is the vector of all ones. This relaxation is tight in case of a rank-one solution.

As previously mentioned, the elliptope,

$$\mathcal{E} = \{X \in \mathbf{S}^n | X \succeq 0, \mathrm{diag}(X) = 1_n\},$$

satisfies Assumption 6.2.2. Hence, problem (6.22) is a good candidate for the proposed framework. Using the rank-l factorization $X = WW^T$ turns the problem on the quotient manifold $\mathcal{M}_{\mathcal{E}} = \bar{\mathcal{M}}_{\mathcal{E}}/\mathcal{O}(l)$, where

$$\bar{\mathcal{M}}_{\mathcal{E}} = \{W \in \mathbf{R}_*^{n\times l} \mid \mathrm{diag}(WW^T) = 1_n\}.$$

The Euclidean gradient and Hessian of the objective function $\bar{f}(W) = \mathrm{Tr}(W^T AW)$ are respectively given by

$$\nabla \bar{f}(W) = 2AW \quad \text{and} \quad \mathrm{D}\nabla \bar{f}(W)[\eta] = 2A\eta, \tag{6.23}$$

for any direction $\eta \in \mathbf{R}^{n\times l}$.

The per-iteration complexity of Algorithm 7 in the present context is of order $O(n^2 l)$. This complexity is dominated by both the manifold-based optimization (i.e., computation of the gradient and the Hessian (6.23)) and the eigenvalue decomposition of the dual variable S_W, that are $O(n^2 l)$. The computational cost related to the manifold-based optimization is however reduced if the matrix A is sparse, which is often the case for Laplacian matrices.

In Table 6.1, we present computational results obtained with Algorithm 7 for computing the maximal cut of a set of graphs. The parameter n denotes the number of vertices of these graphs and corresponds thus to the size of the variable X in (6.22). More details on these graphs can be found in [BM03] and references therein. The proposed low-rank optimization method is compared to the SDPLR algorithm proposed by Burer and Monteiro [BM03], which also rests on the low rank factorization $X = WW^T$ to solve semidefinite programs (SDP). The rank of the optimizer W^* indicates that low-rank methods are highly relevant in this context. They in fact search the solution in a space of significantly reduced dimension. Concerning computational time, it is important to realize that Algorithm 7 is implemented in MATLAB, whereas a C implementation of the SDPLR algorithm is provided by the authors of [BM03]. Although this renders a rigorous comparison of the computational load difficult, Table 6.1 suggests that both methods perform similarly.

In Figure 6.1, we illustrate the monotone convergence of the Algorithm 7 in the particular case of the graph "toruspm3-15-50". The number of iterations is displayed on the bottom abscissa, whereas the top abscissa stands for the rank l. As indicated in Figure 6.2, the smallest eigenvalue $\lambda_{\min}$ of the dual matrix S_W monotonically increases to zero and provides so some insight on the accuracy of the current iterate.

6.3.2 The sparse PCA problem

The new optimization method is used to solve the two convex relaxations (6.4) and (6.8).

[5]Let the *adjacency matrix* $W \in \mathbf{S}^n$ be a symmetric matrix such that the entry w_{ij} is the weight on the edge between the vertices i and j, or zero if there is no edge between these two vertices. The *Laplacian matrix* is defined by $L = \mathrm{Diag}(W1_n) - W$.

			Objective values		**CPU time** (sec)	
Graph	n	Rank(W^*)	Algo. 7	SDPLR	Algo. 7	SDPLR
toruspm3-8-50	512	8	-527.81	-527.81	17	3
toruspm3-15-50	3375	15	-3474.79	-3474.76	1051	181
torusg3-8	3375	7	-3187.61	-3188.09	375	228
G1	800	13	-12083.2	-12083.1	57	35
G11	800	5	-629.16	-629.15	53	15
G14	800	13	-3191.57	-3191.53	82	13
G22	2000	18	-14136.0	-14135.9	358	101
G32	2000	5	-1567.58	-1567.57	158	69
G35	2000	14	-8014.57	-8014.33	525	68
G36	2000	13	-8005.60	-8005.80	459	115
G58	5000	8	-20111.3	-20135.4	1881	1119

Table 6.1: Computational results of Algorithm 7 (implemented in MATLAB) and the SDPLR algorithm (implemented in C) on various graphs.

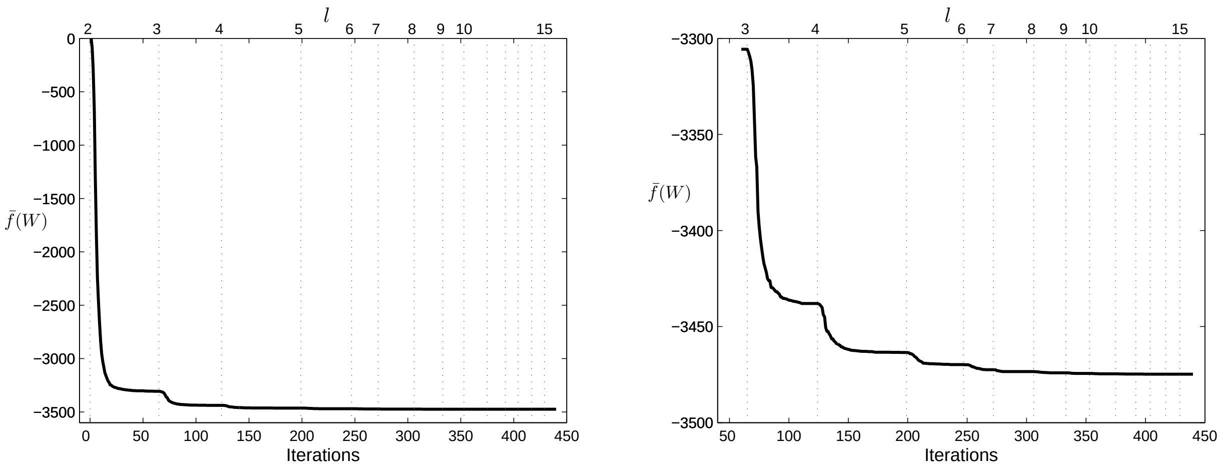

Figure 6.1: Monotone decrease of the objective function in (6.22), i.e., $\bar{f}(W) = \mathrm{Tr}(W^T AW)$, through the iterations (bottom abscissa) and with the rank l (top abscissa) in the case of the graph “toruspm3-15-50”.

First convex relaxation

Let us factor the variable Z in problem (6.4) into the product WW^T and apply the proposed optimization method on the quotient manifold $\mathcal{M}_{\mathcal{SP}} = \bar{\mathcal{M}}_{\mathcal{SP}}/\mathcal{O}(l)$ where

$$\bar{\mathcal{M}}_{\mathcal{SP}} = \{W \in \mathbf{R}_*^{n\times l} \mid \mathrm{Tr}(W^T W) = 1\}.$$

The function to maximize is

$$\bar{f}(W) = \mathrm{Tr}(W^T A^T AW) - \gamma \sum_{i,j} h_\kappa((WW^T)_{ij}) \tag{6.24}$$

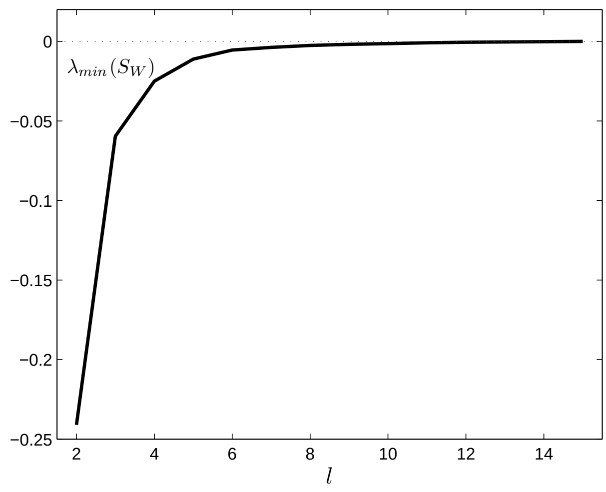

Figure 6.2: Evolution of the smallest eigenvalue of S_W with the rank l (case of the graph "toruspm3-15-50"). The matrix S_W tends to be positive semidefinite as l increases, in which case the product WW^T provides a solution to the convex relaxation (6.22).

with $h_\kappa(x) = \sqrt{x^2 + \kappa^2}$ for $x, \kappa \in \mathbf{R}$. Details on the derivation of the first- and second-order derivatives of $\bar{f}$ can be found in the Appendix. The computational complexity of Algorithm 7 in this context is $O(n^2 l)$. It should be mentioned that the DSPCA algorithm derived in [AEJL07] and that is tuned to solve (6.3) features a per-iteration complexity of order $O(n^3)$.

In Figure 6.3, we illustrate the monotone convergence of Algorithm 7 on a random Gaussian matrix A of size 50-by-50. For comparison, the optimal value of the non-smooth problem (6.3) is computed with the DSPCA algorithm [AEJL07]. The sparsity weight factor γ is chosen to 5 and the smoothing parameter κ equals 10^{-4}.

First, although the smooth objective in (6.4) provides an underestimate to the non-smooth objective in (6.3), the maximizers of both problems (6.3) and (6.4) are still very close. Then, we should mention that all numerical experiments performed with the DSPCA algorithm resulted in a rank-one matrix. A similar observation holds for the smooth problem (6.4), since the objective function remains almost constant for ranks larger than one. Hence, to speed up the computations one could compute a *rank-one* approximate solution of (6.4), i.e., to stop, quite heuristically, Algorithm 7 after the iteration $l = 1$. On the right hand plot of Figure 6.3, the smallest eigenvalue $\lambda_{\min}$ of the matrix S_W appears as a way to monitor convergence.

In Figure 6.4, we provide some insight on the computational time required by a MATLAB implementation of Algorithm 7 for solving the sparse PCA problem (6.4). Square Gaussian matrices A are considered, i.e., $m = n$. On the left hand plot, Algorithm 7 is compared with the DSPCA algorithm and the above mentioned heuristic (i.e., computing a rank-one approximate solution of the problem). The right hand plot highlights the quadratic complexity of Algorithm 7 with the size n of the problem.

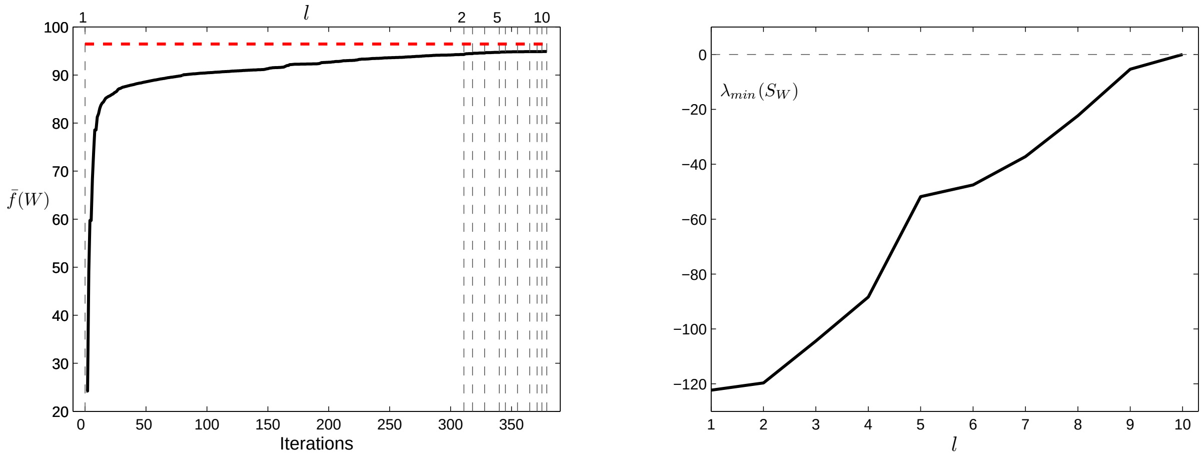

Figure 6.3: Left: monotone increase of the smooth objective function in problem (6.4), i.e., $\bar{f}(W) = \mathrm{Tr}(W^T A^T A W) - \gamma \sum_{i,j} h_\kappa((WW^T)_{ij})$, through the iterations (bottom abscissa) and with the rank l (top abscissa). The dashed horizontal line represents the maximum of the non-smooth objective function in (6.3), computed with the DSPCA algorithm. Right: evolution of the smallest eigenvalue of S_W with the rank l.

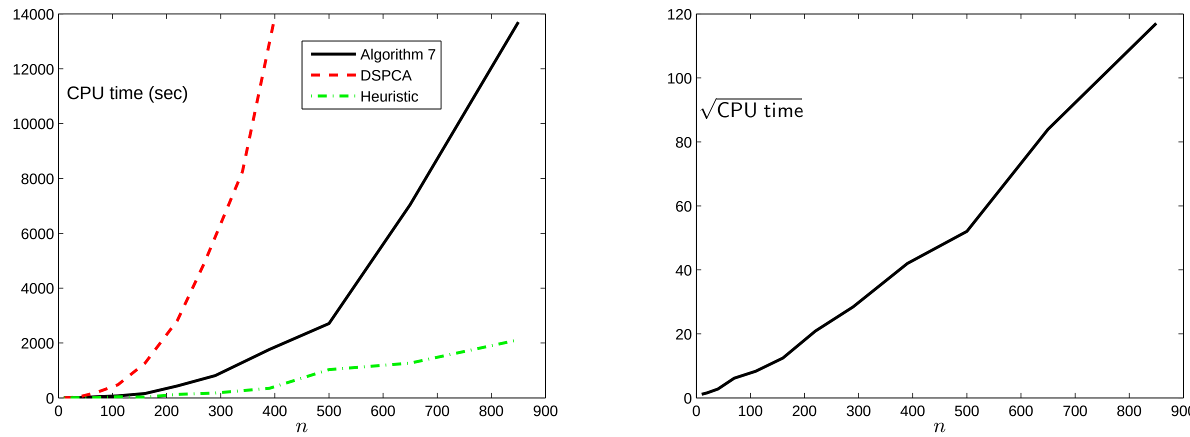

Figure 6.4: Left: Computational time for solving (6.4) versus the problem size in the case $m = n$. Right: Quadratic complexity with n of Algorithm 7.

Second convex relaxation

To solve problem (6.8), we consider the factorization $X = WW^T$ and perform an optimization on the quotient manifold $\mathcal{M}_{\mathcal{SP}}$. As shown by d'Aspremont et al. [ABE07], the spectral function (6.7) equals the function

$$\bar{f}(W) = \sum_{i=1}^{n} \mathrm{Tr}[W^T(a_i^T a_i - \gamma I_m)W]_+, \tag{6.25}$$

for $X = WW^T$. The gradient and Hessian of $\bar{f}$ are evaluated on the basis of explicit formulae derived in the papers [Lew96, LS01] for computing the first- and second-order derivatives of a spectral function. Details on these derivations can be found in the Appendix. As previously mentioned, Algorithm 7 is used to maximize the function (6.25), although it is only piecewise smooth. All the performed numerical simulations converged however successfully to the solution of (6.8). The computational complexity of Algorithm 7 for solving (6.8) is of order $O(nm^2l)$, i.e., linear in the dimension n. The convex relaxation (6.8) of the sparse PCA problem (6.1) is thus adapted for data with more variables than samples, such as gene expression data.

In Figure 6.5, we illustrate the convergence of Algorithm 7 for solving the sparse PCA problem (6.8) for a random Gaussian matrix A of size $m = 100$ and $n = 500$. The sparsity parameter γ is chosen at 5 percent of the upper bound $\bar{\gamma} = \max_i \|a_i\|_2^2$, as discussed in Chapter 5, equation (5.14). The smallest eigenvalue $\lambda_{\min}$ of the matrix S_W presents a monotone decrease once it gets sufficiently close to zero.

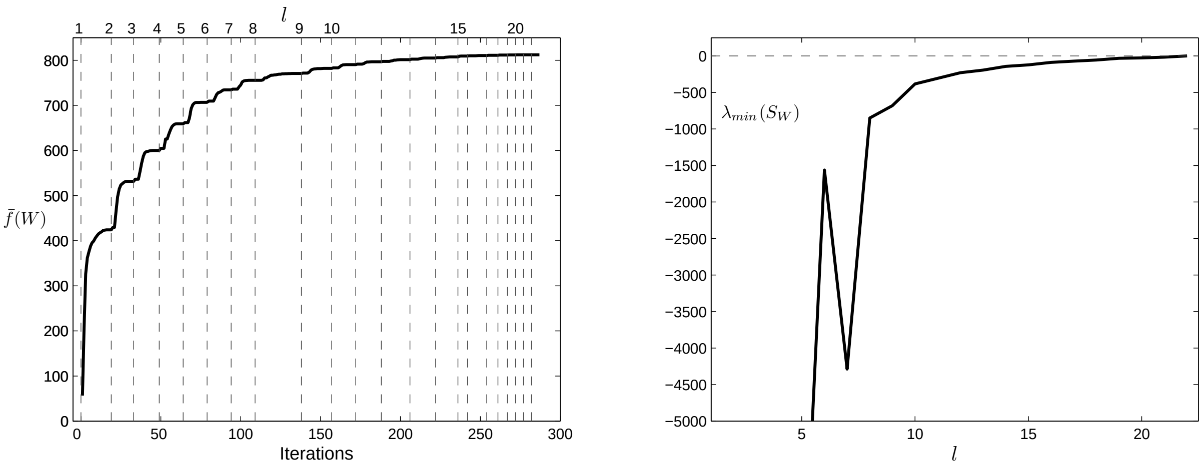

Figure 6.5: Left: monotone increase of the objective function through the iterations (bottom abscissa) and with the rank l (top abscissa). Right: evolution of the smallest eigenvalue of S_W with the rank l.

In Figure 6.6, we plot the CPU time required by a MATLAB implementation of Algorithm 7 for the sparse PCA problem (6.8) versus the dimension n of the matrix A. The dimensions m and l are fixed at 100 and 50, respectively. The data matrix A is generated according to a Gaussian distribution. Figure 6.6 depicts the linear complexity of the method with the dimension n.

6.3.3 Rounding to a rank-one matrix

Both convex relaxations (6.4) and (6.8) result from the reformulation of a problem on unit-norm vectors into a problem on the matrices of the spectahedron. These reformulations are exact if the matrix solution is rank-one. This rank-condition had however to be drop to end up with a convex problem. As a consequence, the solutions of both relaxations (6.4) and (6.8) have in general a rank larger than one. This matrix solution needs to be rounded to a rank-one matrix of the spectahedron, from which a unit-norm vector can be reconstructed

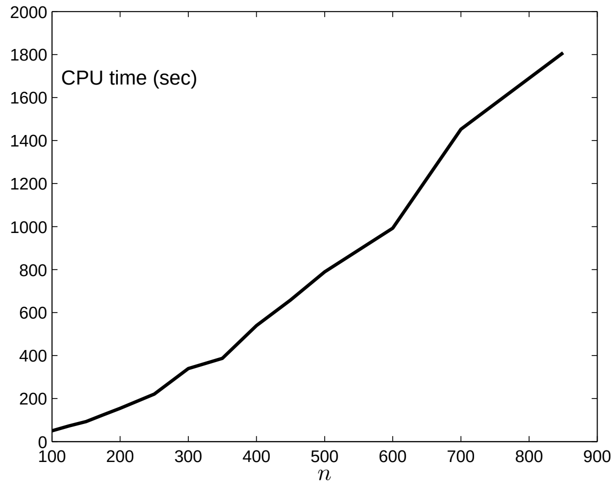

Figure 6.6: Computational time for solving problem (6.8) at the rank $l = 50$ versus the problem size n. The dimension m is fixed at 100.

that is expected to provide a good approximation to the original problem.

As previously mentioned, all numerical experiments performed with the DSPCA algorithm [AEJL07], which solves the non-smooth convex problem (6.3), led to a rank-one solution. The solution of the smooth convex relaxation (6.4) is thus expected to tend to a rank-one matrix for a smoothing parameter κ that gets sufficiently close to zero. This fact is illustrated in Figure 6.7. It should be mentioned that a matrix X of the spectahedron has nonnegative eigenvalues whose sum is one. Hence, X is rank one if and only if its largest eigenvalue equals one. In order to deal with potential numerical problems in the case of a very small smoothing parameter κ, we solve a sequence of problems of the class (6.4) with a parameter κ that is monotonically decreased, and initialize each new problem with the solution of the previous one.

Concerning the convex relaxation (6.8), solutions with a rank larger than one are usually obtained. The solution matrix X has thus to be projected onto the subset of rank-one matrices of the spectahedron to recover a vector variable x that approximately solve the original problem (6.5). A convenient heuristic is to consider the dominant eigenvector of the matrix X. A better solution is probably obtained with the following homotopy method. Consider the optimization problem

$$\begin{array}{ll} \max\limits_{X \in \mathbf{S}^m} & \mu f_{cvx}(X) + (1-\mu) f_{ccv}(X) \\ \text{s. t.} & \mathrm{Tr}(X) = 1, \\ & X \succeq 0, \end{array} \tag{6.26}$$

with the concave function,

$$f_{ccv}(X) = \sum_{i=1}^{n} \mathrm{Tr}[X^{\frac{1}{2}}(a_i^T a_i - \gamma I_m) X^{\frac{1}{2}}]_+,$$

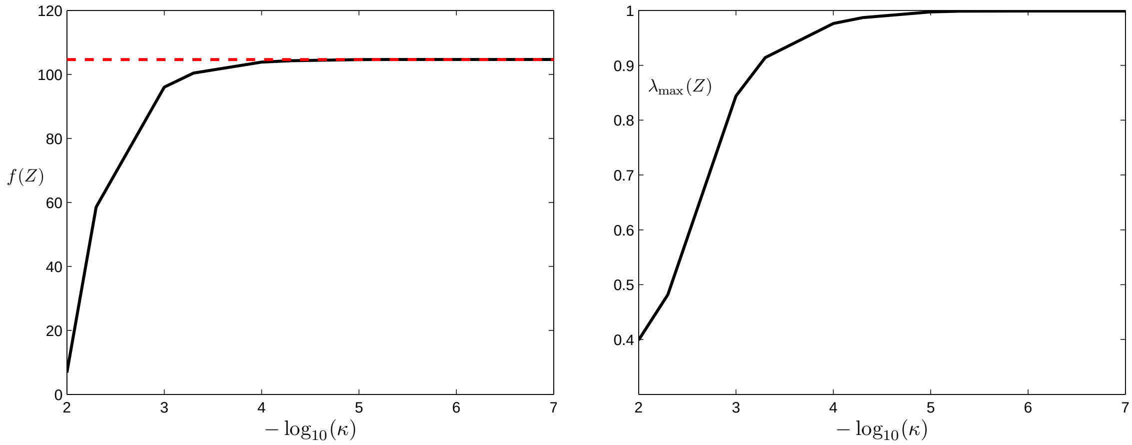

Figure 6.7: Left: evolution of the maximum objective in (6.4) with the smoothing parameter κ. The dashed horizontal line represents the maximum of the non-smooth objective function in (6.3). Right: evolution of the largest eigenvalue of the solution of (6.4). A value of one reflects a rank-one matrix.

the convex function,

$$f_{cvx}(X) = \sum_{i=1}^{n} [a_i^T X a_i - \gamma]_+,$$

and the parameter $0 \leq \mu \leq 1$. As previously mentioned, the functions f_{ccv} and f_{cvx} are identical for rank-one matrices and equal the objective of the original problem (6.5). For $\mu = 0$, problem (6.26) corresponds to the convex relaxation (6.8) and has solutions with a rank typically larger than one. On the other hand, if $\mu = 1$, the solutions of (6.26) are *extreme points* of the spectahedron, i.e., rank-one matrices. The parameter μ is so introduced to continuously interpolate these two extreme situations. By solving a sequence of problems (6.26) with an increasing parameter μ, the solution of (6.8) is projected onto the rank-one matrices of the spectahedron. Problem (6.26) is however no longer convex once $\mu > 0$. The proposed optimization method converges then towards a local maximizer of (6.26). Details on the derivation of the first- and second-order derivatives of $\bar{f}_{ccv}(W) \stackrel{\text{def}}{=} f_{ccv}(WW^T)$ and $\bar{f}_{ccv}(W) \stackrel{\text{def}}{=} f_{cvx}(WW^T)$ can be found in the Appendix.

Computational results obtained on a random Gaussian matrix $A \in \mathbf{R}^{150\times 50}$ are presented in Figure 6.8. The homotopy method is compared with the usual approach, which projects the symmetric positive semidefinite matrix X onto the rank-one matrix xx^T where x is the dominant eigenvector of X normalized to unit-norm. Let f_{EVD} denote the objective function evaluated at this rank-one matrix,

$$f_{EVD}(X) \stackrel{\text{def}}{=} f_{ccv}(xx^T) = f_{cvx}(xx^T).$$

As previously, the maximum eigenvalue is used in Figure 6.8 to monitor the rank of a matrix X of the spectahedron. The continuous plots display the evolution of the functions f_{ccv} and f_{EVD} during the resolution of the convex problem (6.8), i.e., $\mu = 0$ in (6.26). The point A represents the solution obtained with Algorithm 7 by solving (6.8) at the rank $l = 1$, whereas the points B and B' stand for the exact solution of (6.8), which is of rank larger than one. The dashed plots illustrate the effect of the parameter μ, that is linearly increased by steps of 0.05 between the points B and C. For a sufficiently large parameter μ, problem (6.26) presents a rank-one solution (point C). One clearly notices that the objective function of the original problem (6.5), which equals f_{EVD}, is larger at C than at B'. Hence, the rounding method based on (6.26) provides a better rank-one solution than the usual approach based on the eigenvalue decomposition of X.

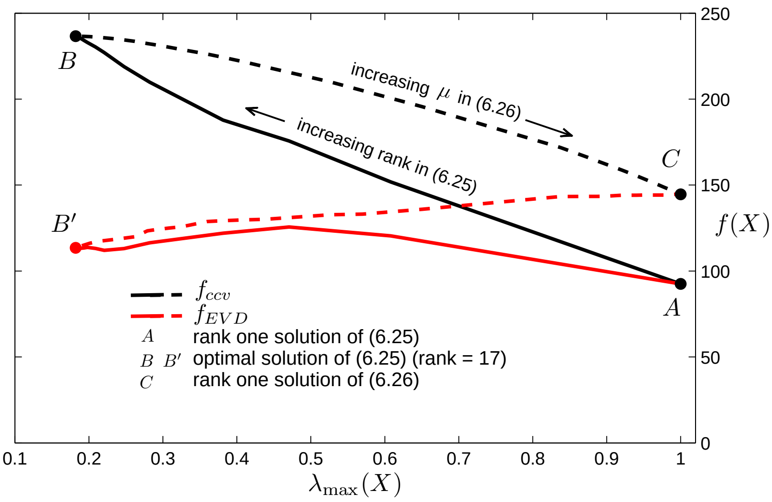

Figure 6.8: Evolution of the functions f_{ccv} (black plots) and f_{EVD} (red plots) in two situations:
(1) resolution of the convex problem (6.8), i.e., $\mu = 0$ in (6.26) (continuous plots).
(2) rounding of the solution of (6.8) to a rank-one matrix by gradual increase of μ in (6.26) (dashed plots).

6.4 Analysis of gene expression data

We discuss in this section the value of solving convex relaxations of sparse PCA for the analysis of gene expression data.

First, the sparse PCA algorithm that results from the first relaxation (6.4) has a numerical complexity that is *quadratic* with the dimension n of the data. The computational time required for analyzing random data with up to 850 variables is illustrated in Figure 6.4. This algorithm seems thus impractical in the context of gene expression data, where the number of variables is around ten thousand.

The numerical complexity of the algorithm related to the second relaxation (6.8) of sparse PCA, however, is *linear* with the number of variables in the data. Nevertheless, in view of Figure 6.6 where data matrices up to the dimension 850-by-100 are considered, analyzing gene expression data by solving this relaxation is probably inconvenient.

Computational results are reported in Table 6.2, which compares two algorithms for computing an approximate solution of the sparse PCA problem (6.5) in the context of the breast cancer gene expression data of Table 3.1. First, we consider an algorithm that computes an approximate solution of rank $l = 10$ of the convex relaxation (6.8). This solution is rounded afterwards to a rank-one matrix by retaining its dominant eigenvector. We do not use the above discussed homotopy method, which is too expensive in this large scale context. Furthermore, for the sake of computational efficiency, the rank is automatically set at $l = 10$, without using the incremental strategy of Algorithm 7. On the other hand, we use the algorithm GPower$_{\ell_0}$ proposed in Chapter 5, which rests on the generalized power method to compute a local solution of (6.5). Both algorithms are initialized identically, i.e., if the algorithm GPower$_{\ell_0}$ is initialized with a vector x, Algorithm 7 is initialized with the rank-one matrix xx^T. Given a breast cancer cohort, the sparsity-inducing parameter γ is set to a constant value for both algorithms: the value that led to the largest PEI in Table 5.7 for the GPower$_{\ell_0}$ method. Ten components are systematically computed by using the deflation scheme described in Section 5.3.

As a first observation, the algorithm based on the convex relaxation (6.5) requires a significant amount of computational effort (Table 6.2A), although it has been reduced to its simplest form, i.e., the rank is fixed to ten instead of being gradually increased, and the rounding of the solution to a rank-one matrix is done by eigenvalue decomposition. This excess in computational effort is furthermore not rewarded by a significant improvement in terms of objective value (Table 6.2B), and the PEI is not necessarily increased (Table 6.2C). Better results could possibly be obtained for rank larger than ten or by using the homotopy method for the rounding, but at the expense of computational time. If one furthermore reminds that in practice the sparsity-inducing parameter γ is tuned by trial-and-error, solving the convex relaxation (6.5) is virtually intractable for a practical component analysis of large data.

6.5 Summary

This chapter is devoted to optimization problems defined in terms of a positive semidefinite matrix X of potentially large dimension, but whose solutions are expected to be of low rank. The proposed optimization method rests on the factorization $X = WW^T$, where the number of columns of W fixes the rank of X. This factorization suggests a reformulation of the original problem as an optimization on a particular quotient manifold. A second-order optimization method is derived and conditions are provided for the rank of the factorization to ensure equivalence with the original problem. The resulting algorithm solves a sequence of nonconvex optimization problems of much lower dimension than the original one and presents a monotone convergence towards the sought solution.

The proposed algorithm seems particularly well adapted to solve convex relaxations of combinatorial problems, which usually have low-rank solutions. A low-rank approximate solution is furthermore often sufficient for such problems since the obtained solution is usually rounded to a rank-one matrix to provide an approximate solution of the initial problem. The number of columns in the matrix W thus provides a tuning parameter to explore this trade-off between the complexity of the problem (i.e., combinatorial problem versus convex problem) and computational efficiency. For sparse PCA, specifically, the low-rank approach reduces

(A) Computational time (in seconds)

	Vijver	Wang	Naderi	JRH-2
Convex relaxation (Algorithm 7)	24040	21690	8600	8080
Combinatorial problem (GPower_{ℓ_0})	4.2	4.8	1.9	3.1

(B) Objective value of (6.5) reached by the first component (to maximize)

	Vijver	Wang	Naderi	JRH-2
Convex relaxation (Algorithm 7)	58376.9	116939.9	19342.2	9757
Combinatorial problem (GPower_{ℓ_0})	58376.7	116939.1	19339.3	9757
Gap between the two methods	0.23	0.82	2.92	10^{-4}

(C) PEI based on a set of 536 cancer-related pathways.

	Vijver	Wang	Naderi	JRH-2
Convex relaxation (Algorithm 7)	0.1194	0.1287	0.0560	0.1175
Combinatorial problem (GPower_{ℓ_0})	0.1250	0.1250	0.0672	0.1026

Table 6.2: Comparison of two algorithms for computing ten components by the resolution of the sparse PCA problem (6.5) in the context of breast cancer gene expression data.

the per-iteration numerical complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ to solve the convex relaxation (6.4) and from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ in the case of problem (6.8). These sparse PCA algorithms are, however, still numerically expensive for a practical analysis of gene expression data.

Although a locally optimal solution of sparse PCA might probably be sufficient in the context of component analysis, finding the best solution can very important for further applications, such as compressed sensing [ABE08]. Finally, besides sparse PCA, the proposed optimization algorithm is also well-suited for a rather large diversity of problems, e.g., the maximal cut of a graph, the best low-rank approximation of a correlation matrix [BX05, GP07], as well as convex problems in the context of *clustering* and *embedding* [KSJ07].

The results of this chapter have been submitted for publication in the *SIAM Journal on Optimization* [JBAS08].

Chapter 7

Conclusion and perspectives

In this thesis, a differential-geometric point of view is adopted to study the problem of *component analysis*, aimed at reducing large data to lower dimensions and revealing so the essential underlying structure. This problem is encountered in almost all areas of science – from physics, chemistry and biology to finance, economics and psychometrics – where large data sets need to be analyzed.

Our investigations on this topic are motivated by the analysis of breast cancer data, which store the expression levels of ten thousand genes gained from experiments on hundreds of cancerous cells. Such data provide a snapshot of the biological processes that occur in tumor cells and offer novel opportunities for an improved understanding of the biology of breast cancer and progress in diagnosis, treatments as well as drugs. The main challenge in analyzing these data is to unravel the complex biological mechanisms that give rise to the measured expression levels.

New algorithms for component analysis

The main contribution of the thesis is the proposal of several new algorithms for component analysis. These algorithms concern *principal component analysis* (PCA), which explains the raw data in terms of uncorrelated components, as well as two extensions of PCA: *independent component analysis* (ICA) and *sparse principal component analysis* (sparse PCA). The former infers components that are as statistically independent as possible, and the latter manages to preserve the interpretability of the components.

The algorithms of the thesis result from a formulation of component analysis as a constrained optimization problem. It turns out that the constraints involved in these settings endow the problem with a simple but rich *manifold* structure. Component analysis is so cast in the realm of optimization on matrix manifolds. The resulting algorithms rest on efficient and well-understood strategies from unconstrained optimization, while simultaneously taking advantage of the geometric structure of the problem to enforce the constraints. Some well-known algorithms for component analysis are recovered in this way, but also new ones have been obtained. The efficiency of these algorithms has been systematically illustrated on random test problems and on the analysis of breast cancer data. Importantly, their numerical complexity is low (most often linear) with the number of variables in the data, which is a valued property to analyze large gene expression data sets.

Efficient optimization methods for important classes of problems

When formalizing component analysis into an optimization framework, three main classes of problems have been encountered, for which methods have been proposed.

First, in Chapter 4, we considered the optimization of a smooth function on the set of n-by-p real matrices with orthonormal columns. This set is endowed with a manifold structure, i.e., it is the *Stiefel manifold*, which specializes in the extreme cases $p = 1$ and $p = n$ to the *unit Euclidean sphere* and to the *orthogonal group*, respectively. We explained how classical methods for unconstrained optimization (e.g., steepest-descent, Newton's or trust-region methods) are typically adapted to handle these types of constraints. Specifically, the underlying manifold structure enables to view the problem around each iterate of the optimization process from the Euclidean tangent space, where it appears like an unconstrained optimization problem. The iterates are then successively computed along curves of the manifold. The orthonormality constraint is so maintained at each iteration in the most natural manner. In the particular case of the orthogonal group, the manifold is equipped with a *Lie group* structure, which enables to consider further optimization methods.

Then, in Chapter 5, we proposed the *generalized power method*, a simple gradient-type method to maximize convex and not necessarily smooth functions on compact sets. When applied to maximize the Rayleigh quotient of a square matrix on the sphere, this algorithm specializes to the well-known *power method*, which computes the dominant eigenvector and eigenvalue of that matrix. Due to the convexity of the objective, the generalized power method converges rapidly to a local maximizer of the problem, and constraints such as spherical constraints or orthonormality constraints (which relate to compact manifolds) are tackled with ease.

Finally, in Chapter 6, we addressed the issue of solving problems defined in terms of large positive semidefinite matrices in a numerically efficient manner by using low-rank arguments. Given a symmetric positive semidefinite matrix variable X of $\mathbf{R}^{n\times n}$, the proposed method rests on the factorization $X = WW^T$ with $W \in \mathbf{R}^{n\times l}$ that enforces a rank at most equal to l to the matrix X. The dimension l enables to find a trade-off between computational efficiency (i.e., $l \ll n$) and fidelity in the original problem (i.e., $l = n$). This setup appears especially appropriate whenever the original problem has a low-rank solution, as it is often expected for convex relaxations of combinatorial problems.

Novel knowledge on breast cancer biology

Applied on breast cancer gene expression data, the proposed algorithms for component analysis enabled to infer novel biological knowledge. Specifically, we proposed an original framework to evaluate the biological significance of obtained components. Components are simultaneously tested for statistical association with pathways and regulatory modules on the one hand, and with clinical data (i.e., phenotypes) on the other hand. In this way, components are used as an intermediate object to link pathways/regulatory modules with phenotypes. Some of the relationships that were unravelled are well-known, but novel ones have also been identified. For instance, the *cancer-related immune response* pathway is consistently correlated with estrogen receptor status. Similarly, the *epithelial-mesenchymal transition signalling* pathway is found to be associated with histological grade. Some of these associations could be inferred from principal components. Some other ones, so far unseen by PCA, have been identified by ICA and confirmed latter by sparse PCA. The analysis performed with sparse PCA in Chapter 5 was only preliminary, but already very promising. There is a good chance that a deeper study of our sparse PCA results would reveal novel associations.

Perspectives for future research

First, the next step towards an improved component analysis of gene expression data is probably to incorporate a priori information on the data in a more *refined* way. The methods covered in this thesis enforce properties such as orthogonality, sparsity or statistical independence on the components. Although strongly biologically motivated, these assumptions provide a very *crude* model of the complex structure of gene expression data. A priori biological knowledge is probably better modeled in the form of a *graph*. Certain genes have in fact well-known affinities, while others almost never interact. This information can be readily transposed into a graph, the vertices of which denote the genes. Edges would then link genes that are likely to be coexpressed, e.g., genes tagged by a common regulatory motif. Consequently, it seems natural to develop methods for component analysis that take a priori information in the form of a graph into account. A first attempt could be to "bias" the computation of the principal components with this graph information, for instance by solving the problem

$$\max_{\substack{z\in\mathbf{R}^n \\ z^Tz=1}} z^TA^TAz + \varphi(z),$$

which maximizes the Rayleigh quotient of the covariance matrix A^TA subject to a penalty $\varphi(z)$ that enforces the graph structure on the vector z, i.e., genes that are connected in the graph should be coexpressed in z. Interestingly, if the penalty $\varphi(z)$ is convex, the new generalized power method can be used.

Another limitation of the present work is that we exclusively applied the *generalized power method* to solve problems defined either on the sphere or on the Stiefel manifold, which are compact *embedded submanifolds* of Euclidean spaces. One can naturally wonder whether an adaptation to compact *quotient manifolds* of Euclidean spaces is possible and meaningful. One could for instance want to optimize a function, which is convex in the total space $\mathbf{R}^{n\times p}$, on the set of p-dimensional subspaces of $\mathbf{R}^n$, i.e., the *Grassmann manifold*, which is a quotient manifold.

Finally, several parameterizations are conceivable for the set of fixed-rank symmetric positive semidefinite matrices. Besides the product $X = WW^T$ with $W \in \mathbf{R}^{n\times l}$, which leads to the quotient manifold $\mathbf{R}_*^{n\times l}/\mathcal{O}(l)$ discussed in this thesis, the symmetric positive semidefinite matrix X of rank l is, for instance, also described by the product $X = URU^T$, where

$U \in \mathrm{St}(l,n)$ is an n-by-l matrix with orthonormal columns and $R \in \mathbf{S}^l$ is a symmetric *positive* definite matrix (i.e., a full-rank matrix). This factorization also encounters symmetries since for any l-by-l orthogonal matrix Q, the product UQ is an element of $\mathrm{St}(l,n)$ and $Q^T RQ$ is symmetric positive definite. This suggests the quotient manifold $(\mathrm{St}(l,n) \times P_l)/\mathcal{O}(l)$, where P_l is the set of l-by-l symmetric positive definite matrices, i.e., the *positive symmetric cone* of dimension l [BS08]. It could be interesting to investigate which parameterizations provide the most efficient algorithms for solving the optimization problems mentioned in Chapter 6.

Concluding words

In the last decade, many algorithms based on geometric arguments have appeared that solve a large diversity of problems. This thesis is highly inspired by this current of research and a geometric point of view is adopted to tackle problems arising in the context of component analysis of large data. Exploiting the geometric structure of a problem is not only neat, elegant and natural. It also provides an inspiring framework to solve the problem. But first and foremost, the most efficient algorithms often lie at the interface of geometry and linear algebra.

Appendix

In this Appendix, we provide some details on the computation of the first- and second-order derivatives of three objectives functions related to the convex relaxations of the sparse PCA problem discussed in Chapter 6.

1. Derivatives of the function (6.24)

Consider the function

$$\bar{f}(W) = \mathrm{Tr}(W^T A^T A W) - \gamma \sum_{i,j} h_\kappa((WW^T)_{ij})$$

with $h_\kappa(x) = \sqrt{x^2 + \kappa^2}$ for $x, \kappa \in \mathbf{R}$.

The Euclidean gradient and Hessian of $\bar{f}$ at W are respectively

$$\nabla \bar{f}(W) = 2A^T A W - 2\gamma M W,$$

and

$$\mathrm{D}\nabla \bar{f}(W)[\eta] = 2A^T A\eta - 2\gamma(M\eta + M'W),$$

where the matrices M and M' are constructed in an element-wise manner as follows,

$$m_{ij} = \left.\frac{\mathrm{d}h_\kappa(x)}{\mathrm{d}x}\right|_{(WW^T)_{ij}}$$

and

$$m'_{ij} = (\eta W^T + W\eta^T)_{ij} \left.\frac{\mathrm{d}^2 h_\kappa(x)}{\mathrm{d}x^2}\right|_{(WW)^T_{ij}}.$$

The first- and second-order derivatives of the smooth function h_κ are

$$\frac{\mathrm{d}h_\kappa(x)}{\mathrm{d}x} = \frac{x}{\sqrt{x^2+\kappa^2}} \quad \text{and} \quad \frac{\mathrm{d}^2 h_\kappa(x)}{\mathrm{d}x^2} = \frac{1}{\sqrt{x^2+\kappa^2}} - \frac{x^2}{(x^2+\kappa^2)^{\frac{3}{2}}}.$$

2. Derivatives of the function (6.25)

Consider the function

$$\bar{f}(W) = \sum_{i=1}^{n} \mathrm{Tr}[W^T(a_i^T a_i - \gamma I_m)W]_+,$$

which is piecewise differentiable. The variable W is an m-by-l matrix.

The Euclidean gradient and Hessian of $\bar{f}$ are evaluated on the basis of explicit formulae derived in the papers [Lew96, LS01] for computing the first- and second-order derivatives of a spectral function. For the sake of clarity, we denote $B_i = a_i^T a_i - \gamma I_m$. Let

$$W^T C_i W = VDV^T$$

be an eigenvalue decomposition of the symmetric matrix $W^T B_i W$, i.e., the l-by-l matrices V and D are orthogonal and diagonal, respectively. The gradient of $\bar{f}$ at a point of differentiability W is given by

$$\nabla \bar{f}(W) = 2 \sum_{i=1}^{n} B_i W V D' V^T,$$

where D' is a diagonal matrix such that

$$d'_{ii} \stackrel{\text{def}}{=} \max(0, \operatorname{sign}(d_{ii})).$$

The Hessian of $\bar{f}$ at W is given by

$$\mathrm{D}\nabla \bar{f}(W)[\eta] = 2 \sum_{i=1}^{n} B_i(\eta V D' V^T + W V D'' V^T),$$

where the symmetric matrix D'' is constructed in an element-wise manner as follows

$$d''_{ij} \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } i = j \\ \frac{d'_{jj} - d'_{ii}}{d_{jj} - d_{ii}} H_{ij} & \text{otherwise} \end{cases},$$

with $H = V^T(\eta^T B_i W + W^T B_i \eta)V$.

3. Derivatives of the function f_{cvx}

Consider the function

$$\bar{f}_{cvx}(W) \stackrel{\text{def}}{=} f_{cvx}(WW^T) = \sum_{i=1}^{n} [a_i^T W W^T a_i - \gamma]_+,$$

which is piecewise differentiable. The variable W is an m-by-l matrix.

At a point of differentiability W, the Euclidean gradient and Hessian of $\bar{f}_{cvx}$ are respectively given by

$$\nabla \bar{f}_{cvx}(W) = 2 \sum_{i=1}^{n} \max(0, \operatorname{sign}(a_i^T W W^T a_i - \gamma)) a_i a_i^T W,$$

and

$$\mathrm{D}\nabla \bar{f}_{cvx}(W)[\eta] = 2 \sum_{i=1}^{n} \max(0, \operatorname{sign}(a_i^T W W^T a_i - \gamma)) a_i a_i^T \eta.$$

References

[ABB00] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, 97(18):10101–10106, 2000.

[ABB03] O. Alter, P. O. Brown, and D. Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *PNAS*, 100(6):3351–3356, 2003.

[ABE07] A. d' Aspremont, F. R. Bach, and L. El Ghaoui. Full regularization path for sparse principal component analysis. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 177–184, 2007.

[ABE08] A. d' Aspremont, F. R. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

[ABG07] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, 7(3):303–330, 2007.

[Abs03] P.-A. Absil. *Invariant Subspace Computation: A Geometric Approach.* PhD thesis, Faculté des Sciences Appliquées, Université de Liège, 2003.

[AEJL07] A. d' Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49:434–448, 2007.

[Afs06] B. Afsari. Simple LU and QR based non-orthogonal matrix joint diagonalization. In *ICA*, pages 1–7, 2006.

[AG06] P.-A. Absil and K. A. Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, 2006.

[AILH09] P.-A. Absil, M. Ishteva, L. De Lathauwer, and S. Van Huffel. A geometric Newton method for Oja's vector field. *Neural Comput.*, 21(5):1415–1433, 2009.

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, Princeton, NJ, 2008.

[AS08] C. Alzate and J.A.K. Suykens. A regularized kernel CCA contrast function for ICA. *Neural Networks*, 21(2-3):170–181, 2008.

[BAMCM97] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Trans. on Signal Processing*, 45:434–444, 1997.

[BCC+03] H. Brentani, O. L. Caballero, A. A. Camargo, A. M. da Silva, and W. A. da Silva et al. The generation and utilization of a cancer-oriented representation of the human transcriptome by using expressed sequence tags. *PNAS*, 100(23):13418–13423, 2003.

[BH95] Y. Benjamini and Y. Hochberg. *Journal of the Royal Statistical Society. Series B*, 57(1):289–300, 1995.

[BJ03] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2003.

[BM03] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2):329–357, 2003.

[Bro72] R. W. Brockett. System theory on group manifolds and coset spaces. *SIAM J. Control*, 10(2):265–284, 1972.

[Bro91] R. W. Brockett. Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. *Linear Algebra Appl.*, 146:79–91, 1991.

[Bro93] R. W. Brockett. Differential geometry and the design of gradient algorithms. In *Proc. of Symposia in Pure Mathematics*, volume 54, pages 69–92. Amer. Math. Soc., 1993.

[BS95] A. J. Bell and T. J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[BS04] T. Beißbarth and T. P. Speed. GOstat: find statistically overrepresented gene ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465, 2004.

[BS08] S. Bonnabel and R. Sepulchre. Geometric distance and mean for positive semidefinite matrices of fixed rank. *ArXiv*, 2008.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BX05] S. Boyd and L. Xiao. Least-squares covariance matrix adjustment. *SIAM J. Matrix Anal. Appl.*, 27(2):532–546, 2005.

[BYC+05] A. H. Bild, G. Yao, J. T. Chang, Q. Wang, A. Potti, D. Chasse, M.-B. Joshi, D. Harpole, J. M. Lancaster, A. Berchuck, J. A. Olson, Jeffrey R. Marks, H. K. Dressman, M. West, and J. R. Nevins. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, 2005.

[Car92] M. P. do Carmo. *Riemannian Geometry.* Birkhäuser Boston, 1992.

[Car99] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, 1999.

[CG07] R. Cangelosi and A. Goriely. Component retention in principal component analysis with application to cDNA microarray data. *Biology Direct*, 2:2, 2007.

[CJ95] J. Cadima and I. T. Jolliffe. Loadings and correlations in the interpretation of principal components. *Journal of Applied Statistics*, 22:203–214, 1995.

[Com94] P. Comon. Independent Component Analysis, a new concept ? *Signal Processing, Elsevier*, 36(3):287–314, 1994. Special issue on Higher-Order Statistics.

[Com04] P. Comon. Canonical tensor decompositions. Research Report RR-2004-17, I3S, 2004.

[Con00] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[CRT+04] A.-S. Carpentier, A. Riva, P. Tisseur, G. Didier, and A. Hénaut. The operons, a criterion to compare the reliability of transcriptome analysis tools: ICA is more reliable than ANOVA, PLS and PCA. *Computational Biology and Chemistry*, 28(1):3–10, 2004.

[CS93] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE ProceedingsF*, 140(46):362–370, 1993.

[CT05] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Trans. on Information Theory*, 51(12):4203–4215, 2005.

[CT06] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 2006.

[CXW+08] L. Chen, J. Xuan, C. Wang, I. Shih, Y. Wang, Z. Zhang, E. Hoffman, and R. Clarke. Knowledge-guided multi-scale independent component analysis for biomarker identification. *BMC Bioinformatics*, 9(1):416, 2008.

[DMB04] A. Dragomir, S. Mavroudi, and A. Bezerianos. SOM-based class discovery exploring the ICA-reduced features of microarray expression profiles: Research paper. *Comp. Funct. Genomics*, 5(8):596–616, 2004.

[EP90] P. J. Eberlein and H. Park. Efficient implementation of jacobi algorithms and jacobi sets on distributed memory architectures. *J. on Parallel and Distributed Computing*, 8(4):358–366, 1990.

[FND08] C. Fraikin, Yu. Nesterov, and P. Van Dooren. A gradient-type algorithm optimizing the coupling between matrices. *Linear Algebra and its Applications*, 429(5-6):1229–1242, 2008.

[FVLH06] A. Frigyesi, S. Veerla, D. Lindgren, and M. Hoglund. Independent component analysis reveals new and biologically significant structures in micro array data. *BMC Bioinformatics*, 7:290, 2006.

[Ger81] J. Gerbrands. On the relationship between SVD, KLT and PCA. *Pattern Recognition*, 14:375–381, 1981.

[GHS+05] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *J. Machine Learning Research*, 6:2075–2129, 2005.

[GP07] I. Grubisic and R. Pietersz. Efficient rank reduction of correlation matrices. *Linear Algebra Appl.*, 422:629–653, 2007.

[GVL89] G. H. Golub and C. F. Van Loan. *Matrix Computations (second edition).* The Johns Hopkins University Press, 1989.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.

[HH97] U. Helmke and K. Hüper. The Jacobi method: A tool for computation and control. In C.F. Martin C.I. Byrnes, B.N. Datta and D.S. Gilliam, editors, *Systems and Control in the Twenty–First Century*, pages 205–228. Birkhäuser, Boston, 1997.

[HH00] A. Hyvärinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Compututation*, 12(7):1705–1720, 2000.

[HJ85] R. A. Horn and C. A. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, UK, 1985.

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Sons, 2001.

[HLK01] L. Hansen, J. Larsen, and T. Kolenda. Blind detection of independent dynamic components, 2001.

[HM94] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. Springer, 1994.

[HMM+00] N. S. Holter, M. Mitra, A. Maritan, M. Cieplak, J. R. Banavar, and N. V. Fedoroff. Fundamental patterns underlying gene expression profiles: simplicity from complexity. *PNAS*, 97(15):8409–8414, 2000.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[HZ06] D. S. Huang and C. H. Zheng. Independent component analysis-based penalized discriminant method for tumor classification using gene expression data. *Bioinformatics*, 22(15):1855–1862, 2006.

[JAS07] M. Journée, P.-A. Absil, and R. Sepulchre. Gradient-optimization on the orthogonal group for independent component analysis. In *7th International Conference on Independent Component Analysis and Blind Signal Separation (ICA 2007)*, 2007.

[JBAS08] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization for semidefinite convex problems. *Submitted to SIAM Journal on Optimization (preprint available on ArXiv)*, 2008.

[JGT+03] M. Jechlinger, S. Grunert, I.H. Tamir, E. Janda, and S. Lüdemann et al. Expression profiling of epithelial plasticity in tumor progression. *Oncogene*, 22(46):7155–7169, 2003.

[JNRS08] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Submitted to Journal of Machine Learning Research (preprint available on ArXiv)*, 2008.

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

[Jol04] I. T. Jolliffe. *Principal Component Analysis, Second Edition*. Springer Series in Statistics. Springer Science, 2004.

[JTA+08] Michel Journée, Andrew Teschendorff, Pierre-Antoine Absil, Simon Tavaré, and Rodolphe Sepulchre. Geometric optimization methods for the analysis of gene expression data. In A. N. Gorban, B. Kégl, D. C. Wunsch, and A. Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*, chapter 12, pages 271–292. Springer Berlin Heidelberg, 2008.

[JTAS07] M. Journée, A. E. Teschendorff, P.-A. Absil, and R. Sepulchre. Geometric optimization methods for independent component analysis applied on gene expression data. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, pages 1413–1416, 2007.

[JTU03] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[KM03] D. P. Kreil and D. J. C. MacKay. Reproducibility assessment of independent component analysis of expression ratios from DNA microarrays. *Comparative and Functional Genomics*, 4(3):300–317, 2003.

[KML+09] W. Kong, X. Mou, Q. Liu, Z. Chen, C. Vanderburg, J. Rogers, and X. Huang. Independent component analysis of alzheimer's DNA microarray gene expression data. *Mol Neurodegener*, 4:1–5, 2009.

[KSJ07] B. Kulis, A. Surendran, and Platt. J. Fast low-rank semidefinite programming for embedding and clustering. In *in Eleventh International Conference on Artifical Intelligence and Statistics, AISTATS 2007*, 2007.

[KVG+08] W. Kong, C. Vanderburg, H. Gunshin, J. Rogers, and X. Huang. A review of independent component analysis application to microarray gene expression data. *Biotechniques*, 45(5):501–520, 2008.

[LB03] S.-I. Lee and S. Batzoglou. Application of independent component analysis to microarrays. *Genome Biology*, 4:R76, 2003.

[Lew96] A. S. Lewis. Derivatives of spectral functions. *Math. Oper. Res.*, 21(3):576–588, 1996.

[LF03] E. G. Learned-Miller and J. W. Fisher III. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

[LGS99] T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent component analysis using an extended Infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Comput.*, 11(2):417–441, 1999.

[Lie02] W. Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18:51–60, 2002.

[LS01] A. S. Lewis and H. S. Sendov. Twice differentiable spectral functions. *SIAM J. Matrix Anal. Appl.*, 23(2):368–386, 2001.

[Lue72] D. G. Luenberger. The gradient projection method along geodesics. *Management Science*, 18:620–631, 1972.

[LUG+08] D. Lutter, P. Ugocsai, M. Grandl, E. Orso, F. Theis, E. Lang, and G. Schmitz. Analyzing M-CSF dependent monocyte/macrophage differentiation: expression modes and meta-modes derived from an independent component analysis. *BMC Bioinformatics*, 9:100, 2008.

[Lum67] J. L. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178, 1967.

[Mac02] D. J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[Mac08] L. Mackey. Deflation methods for sparse PCA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1017–1024, 2008.

[Mat01] H. Mathis. *Nonlinear Functions for Blind Separation and Equalization*. PhD thesis, Swiss Federal Institute of Technology, Zürich, Switzerland, 2001.

[MMSM02] A.-M. Martoglio, J. W. Miskin, S. K. Smith, and D. J. C. MacKay. A decomposition model to track gene expression signatures: preview on observer-independent classification of ovarian cancer. *Bioinformatics*, 18(12):1617–1624, 2002.

[MSZ94] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1994.

[MWA06] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 915–922. MIT Press, Cambridge, MA, 2006.

[Nis99] Y. Nishimori. Learning algorithm for independent component analysis by geodesic flows on orthogonal group. In *Proc. International Joint Conference on Neural Networks IJCNN '99*, volume 2, pages 933–938, 1999.

[NTBM+07] A. Naderi, A. E. Teschendorff, N. L. Barbosa-Morais, S. E. Pinder, and A. R. Green et al. A gene expression signature to predict survival in breast cancer across independent data sets. *Oncogene*, 26:1507–1516, 2007.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization (Second edition)*. Springer, 2006.

[Par80] B. N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1980. Prentice-Hall Series in Computational Mathematics.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[Pha01] D.-T. Pham. Joint approximate diagonalization of positive definite matrices. *SIAM J. on Matrix Anal. and Appl.*, 22:1136–1152, 2001.

[Plu03] M. Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Trans. on Neural Networks*, 14:534–543, 2003.

[Plu05] M. Plumbley. Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. *Neurocomputing*, 67:161–197, 2005.

[PS00] L. Parra and C. Spence. Convolutive blind separation of non-stationary sources. *IEEE Trans. on Speech and Audio Proc.*, pages 320–327, 2000.

[RCTH05] A. Riva, A.-S. Carpentier, B. Torrésani, and A. Hénaut. Comments on selected fundamental aspects of microarray analysis. *Computational Biology and Chemistry*, 29(5):319–336, 2005.

[Sai88] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.

[SH08] H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

[SHK+03] S. A. Saidi, C. M. Holland, D. P. Kreil, D. J. C. MacKay, D. S. Charnock-Jones, C. G. Print, and S. K. Smith. Independent component analysis of microarray data in the study of endometrial cancer. *Oncogene*, 23(39):6677–6683, 2003.

[SKH08] H. Shen, M. Kleinsteuber, and K. Huper. Local convergence analysis of fastica and related algorithms. *IEEE Trans. on Neural Networks*, 19(6):1022–1032, 2008.

[SLK+08] R Schachtner, D Lutter, P Knollmüller, A M M. Tomé, F J J. Theis, G Schmitz, M Stetter, P Gómez G. Vilda, and E W W. Lang. Knowledge - based gene expression classification via matrix factorization. *Bioinformatics*, 24(15):1688–97, 2008.

[SNM+03] C. Sotiriou, S. Y. Neo, L. M. McShane, E. L. Korn, and P. M. Long et al. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *PNAS*, 100(18):10393–10398, 2003.

[SS01] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[STM+05] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. From the cover: Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, 102(43):15545–15550, 2005.

[SWL+06] C. Sotiriou, P. Wirapati, S. Loi, A. Harris, and S. Fox et al. Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis. *J Natl Cancer Inst*, 98(4):262–272, 2006.

[TCA09] F. J. Theis, T. P. Cason, and P.-A. Absil. Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. In *ICA*, volume 5441 of *Lecture Notes in Computer Science*, pages 354–361. Springer, 2009.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(2):267–288, 1996.

[TJA+07] A. Teschendorff, M. Journée, P.-A. Absil, R. Sepulchre, and C. Caldas. Elucidating the altered transcriptional programs in breast cancer using independent component analysis. *PLoS Computational Biology*, 3(8):1539–1554, 2007.

[TNBM+06] A. E. Teschendorff, A. Naderi, N. L. Barbosa-Morais, S. E. Pinder, and I. O. Ellis et al. A consensus prognostic gene expression classifier for er positive breast cancer. *Genome Biol*, 7(10):R101, 2006.

[Vas76] O. Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society: Series B*, 38:54–59, 1976.

[VHV+02] M. J. van de Vijver, Y. D. He, L. J. van't Veer, H. Dai, and A. A. Hart et al. A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, 347(25):1999–2009, 2002.

[WKZ+05] Y. Wang, J. G. Klijn, Y. Zhang, A. M. Sieuwerts, and M. P. Look et al. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, 365(9460):671–679, 2005.

[WLZ07] F. Wang, Z. Liu, and J. Zhang. Nonorthogonal joint diagonalization algorithm based on trigonometric parameterization. *IEEE Trans. on Signal Processing*, 55(11):5299–5308, 2007.

[WRZ+06] G. Wang, N. Rao, Z.-L. Zhang, Q. Mo, and P. Wang. An extended online FastICA algorithm. In *Proc. of the International Symposium on Neural Networks*, pages 1109–1114, 2006.

[WSC05] W. Wang, S. Sanei, and J. A. Chambers. Penalty function-based joint diagonalization approach for convolutive blind separation of nonstationary sources. *IEEE Trans. on Signal Processing*, 53(5):1654–1669, 2005.

[XLK+05] X. Xie, J. Lu, E. J. Kulbokas, T. R. Golub, and V. Mootha et al. Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals. *Nature*, 434:338 – 345, 2005.

[ZHT06] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[ZYW+05] X. W. Zhang, Y. L. Yap, D. Wei, F. Chen, and A. Danchin. Molecular diagnosis of human cancer type by gene expression profiles and independent component analysis. *Eur J Hum Genet*, 13(12):1303–1311, 2005.