UNIVERSITÀ DEGLI STUDI DI MILANO

GRADUATE SCHOOL IN MATHEMATICAL SCIENCES

DEPARTMENT OF MATHEMATICS ”FEDERIGO ENRIQUES”

PHD COURSE
IN MATHEMATICS AND STATISTICS FOR COMPUTATIONAL SCIENCES
XXV CYCLE

Doctoral Thesis

SPARSE RECOVERY BY NONCONVEX LIPSHITZIAN MAPPINGS

settore scientifico-disciplinare: INF/01

Author: ALESSANDRO ADAMO

Supervisor: Prof. Alberto BERTONI

Co-Supervisor: PhD. Giuliano GROSSI

PhD Programme Coordinator: Prof. Giovanni NALDI

Acknowledgements

This dissertation would not have been possible without the guidance and the help of several individuals who, in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

Would like to express my heartfelt gratitude to my advisor Prof. Alberto Bertoni for supporting me and guiding my research with mathematical tips. A special thanks to Ph.D. Giuliano Grossi whereby I have worked hard on common topics and whereby I had the pleasure to work. I want to thank Ph.D. Raffaella Lanzarotti for introducing me to some interesting image processing problems, and to her valuable support to my English.

I have to thank Prof.ssa Paola Campadelli for her availability to listen students' problems and her will to resolve it. I have to thank all the guys of the LAIV laboratory, for relaxation moments and for interesting discussions carried out. I thank Prof. Federico Pedersini to give me precious suggestions related to electronics and signal processing and for funny chats. I have to thank the Department of Computer Science of the University of Milan, for having accommodate me in a comfortable environment, giving me the opportunity to work with stimulating colleagues.

I would also like to thank Prof. Giovanni Naldi and Prof. Massimo Emiliano Tarallo for their availability and for their valuable advice analysis of the developed methods. I would to thank all the Department of Mathematics of the University of Milan to give me the chance to follow interesting courses that have stimulated me and which gave me new ideas for my research.

I would not have contemplated this road if not for my parents, Alonzo and Lorella that believing in me and supporting my choices, gave me the strength to overcome the difficulties.
I have to give a special thanks to my grandmother Lucia, that has always loved and raised me as a son.

I have to thank Salvatore, Fortuna and Vito, for welcoming me into their family and for treating me like a son.

Finally I have to thank my girlfriend Sabrina for having believed in my ability and for having brought with me a path of growth, letting me know to be the woman of my life and soon my wife.

Contents

List of Figures

List of Tables

Introduction

In recent years, the sparsity concept has attracted considerable attention in areas of applied mathematics and computer science, especially in signal and image processing fields [41, 29, 55]. The general framework of sparse representation is now a mature concept with solid basis in relevant mathematical fields, such as probability, geometry of Banach spaces, harmonic analysis, theory of computability, and information-based complexity. Together with theoretical and practical advancements, also several numeric methods and algorithmic techniques have been developed in order to capture the complexity and the wide scope that the theory suggests. All these discoveries explain why sparsity paradigm has progressively interested a broad spectrum of natural science and engineering applications.

Sparse recovery relays over the fact that many signals can be represented in a sparse way, using only few nonzero coefficients in a suitable basis or overcomplete dictionary. The problem can be described as follows. Given a fixed vector $s \in \mathbb{R}^n$ and a matrix $\Phi \in \mathbb{R}^{n \times m}$ with $m > n$, determine the sparsest solution α^*, i.e.

$$\alpha^* = \arg\min_{\alpha} ||\alpha||_0, \quad \text{s.t. } \Phi\alpha = s \tag{BP0}$$

where $||\cdot||_0$ is the ℓ_0 quasinorm, that represents the number of non-zero entries of the vector α.
Unfortunately, this problem, also called ℓ_0-*norm minimization*, is not only NP-hard [85], but but also hard to approximate within an exponential factor of the optimal solution [87]. Nevertheless, many heuristics for the problem has been obtained and proposed for many applications. Among them we recall a greedy pursuit technique that approximates a sparse solutions to an underdetermined linear system of equations. Successively, several greedy-based extended heuristics that directly attempt to solve the ℓ_0-norm minimization have been proposed, for instance, Matching Pursuit (MP) [77], Orthogonal Matching Pursuit (OMP) [91] and Stagewise Orthogonal Matching Pursuit (StOMP) [45].

A second key contribution [31] relaxes the problem by using the ℓ_1-norm for evaluating sparsity and solving the relaxed problem by linear programming. Typical algorithm in this class of algorithms is Basis Pursuit (BP) [29]

This thesis provides new regularization methods for the sparse representation problem with application to face recognition and ECG signal compression. The proposed methods are based on fixed-point iteration scheme which combines non-convex Lipschitzian-type mappings with canonical orthogonal projectors. The first are aimed at uniformly enhancing the sparseness level by shrinking effects, the latter to project back into the feasible space of solutions. In particular the algorithm LIMAPS (Lipshitzian Mappings for Sparse recovery) is proposed as heuristics for (BP0). This algorithm is based on a parametric class G_λ of nonlinear mappings

$$G_\lambda : \{\alpha \quad | \quad s = \Phi\alpha\} \to \{\alpha \quad | \quad s = \Phi\alpha\}.$$

First of all, the problem (BP0) is relaxed to the problem

$$\alpha^* = \arg\min_\alpha ||\alpha||_{<\lambda>}, \qquad \text{s.t. } \Phi\alpha = s \tag{REL}$$

where, for all $\lambda > 0$, $||\cdot||_{<\lambda>}$ is a suitable pseudonorm such that $||\alpha||_0 \approx ||\alpha||_{<\lambda>}$ for large λ.

The main result we obtain in this part states under reasonable conditions, the minima of (REL) are asymptotically stable fixed points of G_λ with respect to the iterative system

$$\alpha_{t+1} = G_\lambda(\alpha_t)$$

Then, the LIMAPS algorithm requires a suitable sequence $\{\lambda_t\}$ with $\lim_{t\to\infty} \lambda_t = \infty$ Roughly speaking, this implies that $||\cdot||_{<\lambda>} \approx ||\cdot||_0$ for large t. LIMAPS implements the system

$$\alpha_{t+1} = G_\lambda(\alpha_t)$$

for obtaining a sparse solution as $t \to \infty$.

In many applications, it is often required to solve the variant of (BP0) in which the sparsity level is a given as a constant:

$$\alpha^* = \arg\min_\alpha ||\Phi\alpha - s||_2^2, \qquad \text{s.t. } ||\alpha||_0 \leq 0 \tag{LS0}$$

In this thesis we propose a heuristic for (LS0) the algorithm k-LIMAPS . An empirical evidence of convergence of k-LIMAPS to good solutions is discussed.

In the second part of this thesis we study two applications in which sparseness has been successfully applied in recent areas of the signal and image processing: the face recognition problem and the ECG signal compression problem.

In the last decades, the face recognition (FR) problem has received increasing attention. Despite excellent results have been achieved, the existing methods suffer when applied in uncontrolled conditions. Such bottleneck represents a serious limit for their real applicability. In this work we propose two different algorithms

able to work on crucial topics in the context of uncontrolled conditions: variation in illumination conditions, continuous occlusions and large expression variations. Experimental results based on *k*-LIMAPS show that the algorithm has high recognition rate, showing good stability performances both in case of manually and automatically registered images.

In the last few years, the need of ECG signal recordings has been enormously augmented due to the increasing interest in health care. Portable ECG recording systems (e.g., Holters) record ECG signals continuously for long time periods ranging between several hours and few days. We present two novel and efficient signal compression algorithms aimed at finding the sparsest representation of the ECG signals based on *k*-LIMAPS . The idea behind these algorithms is to use to sparse represent the ECG signal to compress a dictionary created from signal itself. The saved coefficients are then discretized and rescaled in a convenient range and compressed by a lossless entropy-based algorithm. Experimentally we have shown the effectiveness of our methods which reaches high compression rate maintaining an acceptable percent root-mean square difference level.

0 0

Part I
Theoretical Foundations

Chapter 1
Sparse Recovery and Compressive Sensing

Abstract Shannon-Nyquist sampling theorem is one of the central principle in signal processing. To reconstruct without error a signal $s(t)$ with no frequencies higher than B Hertz by the sampled signal $s_c(t)$, it is sufficient a sampling frequency $A > 2B$.
In the last few years a further development called *compressive sensing* has emerged, showing that a signal can be reconstructed from far fewer measurements than what is usually considered necessary, provided that it admits a sparse representation.
In this chapter we provide a brief introduction of the basic theory underlying compressive sensing and discuss some methods to recovery a sparse vector in efficient way.

1.1 Introduction

Compressive sensing (CS) has emerged as a new framework for signal acquisition and sensor design [47, 28]. It provides an alternative to Shannon / Nyquist sampling when signal under acquisition is known to be sparse or compressible. Instead of taking periodic signal samples of length n, we measure inner products with $p \ll n$ measurement vectors and then recover the signal via sparsity seeking optimization algorithm. In matrix notation, the measurement vector y can be expressed as

$$y = \Psi s = \Psi \Phi \alpha$$

where the rows of $p \times n$ matrix Ψ contain the measurement vectors, Φ is an $n \times n$ compression matrix, α is the sparse compressed signal and s is the sampled signal. While the matrix $\Psi\Phi$ is rank deficient, and hence loses information in general, it can be shown to preserve the information in sparse and compressible signals if it satisfies the Restricted Isometry Property (RIP) [15]. The standard CS theory states that robust signal recovery is possible from $p = \mathcal{O}(p \log \frac{n}{p})$ measurements.

Many fundamental works are proposed by Càndes, Chen, Sauders, Tao and Romberg [31, 21, 26, 27, 23] in which are shown that a finite dimensional signal having a sparse or compressible representation can be recovered exactly from a small set of linear non adaptive measurements.

This chapter starts with preliminary notations on linear algebra and continue with an introduction to the compressive sensing problem and recall some of the most important results in literature that summarize under which conditions compressive sensing algorithms are able to recover the sparsest representation of a signal into a given basis.

1.2 Preliminary Notations on Linear Algebra

The set of all $n \times 1$ column vectors with complex number entries is denoted by $\mathbb{C}^n$, the i-th entry of a columns vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ is denoted by x_i.
The set of all $n \times m$ rectangular matrices with complex number entries is denoted by $\mathbb{C}^{n \times m}$. The elements in the i-th row and j-th column of a matrix A is denoted by $A_{i,j}$.

Let $A \in \mathbb{C}^{n \times m}$ a rectangular matrix, the left multiplication of a matrix A with a scalar λ gives another matrix λA of the same size as A. The entries of λA are given by $\lambda(A)_{i,j} = (\lambda A)_{i,j} = \lambda A_{i,j}$. Similarly, the right multiplication of a matrix A with a scalar λ is defined to be $(A\lambda)_{i,j} = A_{i,j}\lambda$. If A is an $n \times m$ matrix and B is an $m \times p$ matrix, the result AB of their multiplication is an $n \times p$ matrix defined only if the number of columns m in A is equal to the number of rows m in B. The entries of the product matrix smatrix AB are defined as $(AB)_{i,j} = \sum_{k=1}^{m} A_{i,k} B_{k,j}$. The matrix addition is defined for two matrices of the same dimensions. The sum of two $m \times n$ matrices A and B,s denoted by $A + B$, is again an $m \times n$ matrix computed by adding corresponding elements $(A + B)_{i,j} = A_{i,j} + B_{i,j}$.

The dot product, or scalar product, is an algebraic operation that takes two equal-length vectors and returns a single number. Let $a = (a_1, \ldots, a_n) \in \mathbb{C}^n$ and $b = (b_1, \ldots b_n) \in \mathbb{C}^n$, the dot product can be obtained by multiplying the transpose of the vector a with the vector b and extracting the unique coefficient of the resulting 1×1 matrix is defined as

$$a^T b = \sum_{i=1}^{n} a_i b_i$$

Let $A \in \mathbb{C}^{n \times m}$, the adjoint matrix is a matrix $A^* \in \mathbb{C}^{m \times n}$ obtained from A by taking the transpose and then taking the complex conjugate of each entry. Formally the adjoint matrix is defined as

$$A^* = (\overline{A})^T = \overline{A^T}$$

where A^T denotes the transpose and $\overline{A}$ denotes the matrix with complex conjugated entries. If the matrix is $A \in \mathbb{R}^{n\times m}$, the adjoint matrix is given by

$$A^* = A^T$$

A square matrix $A \in \mathbb{C}^{n\times n}$ is called invertible if there exists a matrix $B \in \mathbb{C}^{n\times n}$ such that

$$AB = BA = I$$

where I denotes the $n \times n$ identity matrix and the multiplication used is ordinary matrix multiplication. If the matrix B exist, is uniquely determined by A and is called the inverse of A, denoted as A^{-1}. An orthogonal matrix is a square matrix with complex entries whose columns and rows are orthogonal unit vectors. Equivalently, a matrix A is orthogonal if its adjoint matrix is equal to its inverse:

$$A^* = A^{-1}$$

which entails

$$A^*A = AA^* = I$$

A symmetric $n \times n$ complex matrix A is said to be positive definite if

$$z^*Az > 0,\ \forall z \in \mathbb{C}^n$$

Similarly a symmetric $n \times n$ complex matrix A is said to be positive semi-definite if

$$z^*Az \geq 0,\ \forall z \in \mathbb{C}^n$$

Let $A \in \mathbb{C}^{n\times m}$, the column rank of a matrix A, denoted as rank(A), is the maximum number of linearly independent column vectors of A. It is known that the maximum number of linearly independent rows of a matrix is always equal to the maximum number of linearly independent columns.

For a rectangular matrix $A \in \mathbb{C}^{n\times m}$, by definition rank$(A) \leq \min\{n,m\}$. When rank$(A) = \min\{n,m\}$ A is said full rank, otherwise the matrix is said deficient rank.

A diagonal matrix is a matrix $D = (d_{i,j}) \in \mathbb{C}^{n\times n}$ such that

$$d_{i,j} = 0 \text{ iff } i \neq j$$

Denote with $D = diag(d_1,\ldots,d_n) \in \mathbb{R}^{n\times n}$ the diagonal matrix with diagonal vector $(d_1,\ldots,d_n)$. It holds that

$$diag(a_1,\ldots,a_n) + diag(b_1,\ldots,b_n) = diag(a_1+b_1,\ldots,a_n+b_n) \tag{1.1}$$

$$diag(a_1,\ldots,a_n)diag(b_1,\ldots,b_n) = diag(a_1b_1,\ldots,a_nb_n) \tag{1.2}$$

The diagonal matrix $D = diag(d_1,\ldots,d_n)$ is invertible if and only if all the entries of $(d_1,\ldots,d_n)$ are all non-zero. In this case, we have

$$diag(d_1,\ldots,d_n)^{-1} = diag(\frac{1}{d_1},\ldots,\frac{1}{d_n})$$

The Hadamard product is a binary operation that takes two matrices of the same dimensions, and produces another matrix where each element i,j is the product of elements i,j of the original two matrices. Let $A,B \in \mathbb{C}^{n\times m}$ two rectangular matrices of the same dimensions, the Hadamard product $A \odot B$ is a matrix, of the same dimension as the operands, with elements given by

$$(A \odot B)_{i,j} = A_{i,j}B_{i,j}$$

The Hadamard product is commutative, associative and distributive over addition. That is,

$$A \odot B = B \odot A \tag{1.3}$$
$$A \odot (B \odot C) = (A \odot B) \odot C \tag{1.4}$$
$$A \odot (B + C) = A \odot B + A \odot C \tag{1.5}$$
$$\tag{1.6}$$

The identity matrix under Hadamard multiplication of two $n \times m$ matrices is $n \times m$ matrix where all elements are equal to 1.

Given $x \in \mathbb{R}^n$, the Euclidean norm of x is defined as

$$||x||_2 = (x^T x)^{\frac{1}{2}}$$

The Euclidean norm measures the distance from the point x to the origin. It is almost immediate from the definition that the following properties hold:

- $||x||_2 \geq 0$ for all $x \in \mathbb{R}^n$
- $||x||_2 = 0$ iff $x = 0$
- For any $\gamma \in \mathbb{R}$, $\gamma > 0$, $||\gamma x||_2 = |\gamma|\,||x||_2$

1.2.1 Singular Value Decomposition theorem

Let A be an $n \times m$ matrix of rank r, with $n < m$. Then there exist unitary orthogonal matrices U, V respectively of order n and m such that

$$A = UDV^*$$

where D has the form

$$D = \begin{pmatrix} D_1 & 0 \\ 0 & 0 \end{pmatrix}$$

and $D_1 = diag(\sigma_1,\ldots,\sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ is a non singular diagonal matrix. The diagonal elements of $D = diag(\sigma_1,\ldots,\sigma_r,0,\ldots,0)$ are called *singular values*

of A and the number of non-zeros σ_i's is equal to the rank of A. The ratio $\frac{\sigma_1}{\sigma_r}$, with $\sigma_r \neq 0$ can be regarded as a condition number of A.

1.2.2 Moore-Penrose pseudoinverse

Let $A = U \begin{pmatrix} D_1 & 0 \\ 0 & 0 \end{pmatrix} V^*$ be the singular value decomposition of A. Then the Moore-Penrose pseudoinverse of A is defined as

$$A^\dagger = V \begin{pmatrix} D_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^* \tag{1.7}$$

It is possible to prove that the Moore-Penrose pseudoinverse $A^\dagger$ is the unique matrix X satisfying:

$$AXA = A \tag{1.8}$$
$$XAX = X \tag{1.9}$$
$$(XA)^* = XA \tag{1.10}$$
$$(AX)^* = AX \tag{1.11}$$
$$\tag{1.12}$$

Let us consider the linear system

$$Ax = b$$

and let $x_0 = A^\dagger b$. Then for $x \neq x_0$ we have either

$$||Ax - b||_2 > ||Ax_0 - b||_2 \tag{1.13}$$

or

$$||Ax - b||_2 = ||Ax_0 - b||_2 \text{ and } ||x||_2 > ||x_0||_2 \tag{1.14}$$

In other words, $A^\dagger b$ is either the unique least square solution, or it is the least square solution of minimum norm.
An important case is when the rows of A are linearly independent; in this case $A^\dagger b$ is the unique least square solution and $A^\dagger$ can be obtained by

$$A^\dagger = (A^T A)^{-1} A^T \tag{1.15}$$

The product of the matrix A and the n-dimensional vector x can be written in terms of dot product of vectors as follow:

$$Ax = \begin{bmatrix} a_1^T x \\ a_2^T x \\ \vdots \\ a_n^T x \end{bmatrix}$$

where $a_1, a_2, \ldots, a_n$ denote the rows of the matrix A. Let $\mathcal{N}_A$ be the null space of A, i.e. $\mathcal{N}_A = \{x | Ax = 0\}$; $x \in \mathcal{N}_A$ is orthogonal to each row vectors of A. The row space of the matrix A is the span of the row vectors of A; it follows that a vector $x \in \mathcal{N}_A$ iff it is perpendicular to every vector in the row space of A. The dimension of the null space of A is called nullity. It holds that $\text{rank}(A) + \text{nullity}(A) = m$.

Let $x \in \mathbb{R}^n = V \oplus W$. Then x can be uniquely decomposed into

$$x = x_1 + x_2 \text{ with } x_1 \in V \text{ and } x_2 \in W$$

The transformation that maps x into x_1 is called *projector matrix* onto V along W and is denoted by P. The vector $x_1 = Px$ transformed by the projector P is called the projection of x onto V along W. The necessary and sufficient condition for a square matrix P of order n to be the projection matrix is given by

$$P^2 = P$$

Let Q denote the projector that transforms x in x_2. Then

$$Px + Qx = (P + Q)x$$

Because the equation above has to hold for any $x \in \mathbb{R}^n$, it must hold that

$$I = P + Q$$

Let P a square matrix be the projector matrix onto V along W. Then $Q = I - P$ satisfies $Q^2 = I - 2P + P^2 = Q$ indicating that Q is the projection matrix onto W along V. We have that

$$PQ = P(I - P) = P - P^2 = 0$$

implying that the $span(Q)$ constitutes the null space of P, similarly $QP = 0$ imply that the $span(P)$ constitutes the null space of Q.

By coming back to the general case, let $Q = A^\dagger A$

$$Q^2 = A^\dagger (AA^\dagger A) = A^\dagger A = Q$$

Q is the projection on the null space of A and $P = I - Q$ is the projection on the orthogonal complement of the null space of A, denoted by $\mathcal{N}_A^\perp$.

The null space also plays a role in the solution to a nonhomogeneous linear system of linear equations

$$Ax = b \tag{1.16}$$

If u and v are two possible solutions of the equations to the linear system (1.16), then

$$A(u-v) = Au - Av = b - b = 0$$

Thus the difference of any two solutions of the equation (1.16) lies in $\mathscr{N}_A$. It follows that any solution to the equation (1.16) can be expressed as the sum of a fixed solution v and an arbitrary element of $\mathscr{N}_A$. That is, the solution set of (1.16) is defined as

$$\{v+x | x \in \mathscr{N}_A\}$$

where v is any fixed vector satisfying $Av = b$. The solution set of (1.16), also called affine space and denoted by $\mathscr{A}_{A,b}$, is the translation of the null space of A by the vector v.

1.2.3 Norm, Pseudo-Norm and Quasi-Norm in $\mathbb{R}^n$

For every p with $p \in (0 < p < \infty)$, let us consider the functional $||.||_p : \mathbb{R}^n \to \mathbb{R}^+$ defined by:

$$||x||_p = (\sum |x_i|^p)^{\frac{1}{p}} \tag{1.17}$$

This functional is extended to $p = 0$ and $p = \infty$ as follows:

$$||x||_0 = \lim_{p \to 0} ||x||_p^p = |supp(x)| \tag{1.18}$$

$$||x||_\infty = \lim_{p \to \infty} ||x||_p = \max_i |x_i| \tag{1.19}$$

with $supp(x) = \{i | x_i \neq 0\}$ is the support of the vector x.

It is known that $||.||_p$, with $1 \leq p \leq \infty$, is a norm , i.e. it holds:

$$||x||_p = 0 \text{ iff } x = 0 \tag{1.20}$$

$$||\alpha x||_p = |\alpha| ||x||_p \tag{1.21}$$

$$||x+y||_p \leq ||x||_p + ||y||_p \tag{1.22}$$

In particular, $\mathbb{R}^n$ equipped by $||.||_p$ is a Banach space [9].
If $0 < p < 1$, $||.||_p$ is a quasinorm [108], i.e. it satisfies the norm axioms, except that the triangular inequality which is replaced by

$$||x+y||_p = \gamma(||x||_p + ||y||_p) \tag{1.23}$$

for some $\gamma > 1$. A vector space with an associated quasinorm is called a quasi-normed vector space .

If $p = 0$, the triangular inequality (1.22) holds but does not hold the scaling axiom (1.21). In this case $||.||_0$ is called pseudonorm .

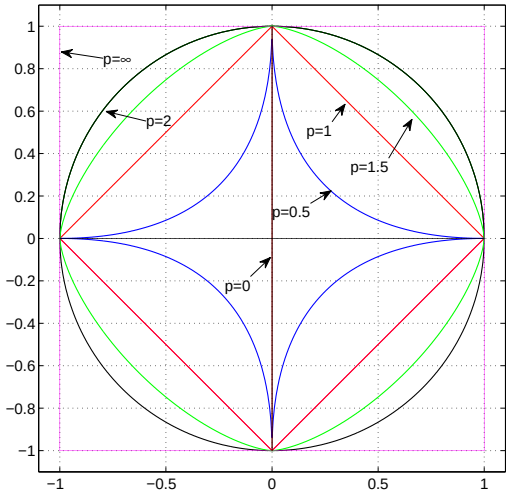

Fig. 1.1 Unit spheres in $\mathbb{R}^2$ for the l_p norms with $p = 1, 2, \infty$, for the l_p quasinorm with $p = 0.5$ and for the pseudonorm l_0.

In figure 1.1 are depicted on the plane the unit sphere and some quasinorms $\{x \mid ||x||_p = 1\}$ for different values of p.

1.2.4 Metric Space and Normed Vector Space

A vector space over a field F is a set V equipped with operation sum $+ : V \times V \to V$ and scalar multiplication $. : F \times V \to V$, such that the following conditions hold:

$$x+y=y+x \quad \text{(commutativity)} \tag{1.24}$$
$$(x+y)+z=x+(y+z) \quad \text{(associativity of vector addition)} \tag{1.25}$$
$$\alpha(\beta x)=(\alpha\beta)x \quad \text{(associativity of scalar multiplication)} \tag{1.26}$$
$$x+0=0+x=x \quad \text{(additive identity)} \tag{1.27}$$
$$1x=x \quad \text{(scalar multiplication identity)} \tag{1.28}$$
$$x+(-x)=0 \quad \text{(existence of inverse)} \tag{1.29}$$
$$\alpha(x+y)=\alpha x+\alpha y \quad \text{(distributivity of vector sums)} \tag{1.30}$$

We define a metric space an ordered pair (M,d) where M is a set and d is a metric function

$$d : M \times M \to \mathbb{R}^+$$

such that for any $x,y,z \in M$ the following properties holds:

$$d(x,0)=0 \quad \text{iff } x=y \text{ (identity)} \tag{1.31}$$
$$d(x,y)\geq 0 \quad \text{(non negativity)} \tag{1.32}$$
$$d(x,y)=d(y,x) \quad \text{(simmetry)} \tag{1.33}$$
$$d(x,z)\leq d(x,y)+d(y,z) \quad \text{(triangular inequality)} \tag{1.34}$$

The function d is also called distance function .

A normed vector space is a pair $(V,||.||)$ where V is a vector space and $||.||$ is a norm function on V. Observe that $||x-y||$ is a distance function.

We call x the limit of the sequence (x_n) if for each real number $\varepsilon > 0$, there exists a natural number N such that, for every $n > N$, $|x_n - x| < \varepsilon$ where $|.|$ denote the absolute value. The sequence (x_n) is said to converge to or tend to the limit x, and we denote it as $x_n \to x$. A sequence $\{x_n\}_n$ of real numbers is said to have the Cauchy property, if for every positive real number ε, there is a positive integer N such that for all natural numbers $m,n < N$

$$|x_m - x_n| < \varepsilon$$

A metric space M is called complete, if every Cauchy sequence of points in M has a limit that is also in M or, alternatively, if every Cauchy sequence in M converges in M.

1.2.5 Convex, Smooth and Lipschitzian Functions

A set V is convex if the line segment between any two points in V lies in V, i.e., if for any $x_1,x_2 \in C$ and and any θ with $0 \leq \theta \leq 1$, we have

$$\theta x_1 + (1-\theta)x_2 \in V$$

A real valued function $f : V \to \mathbb{R}$ defined on a convex set $V \subseteq \mathbb{R}^n$ in a vector space is said to be convex if for any points $x, y \in V$ and any $\alpha \in [0,1]$ it holds

$$f(\alpha x + (1-\alpha)) \leq \alpha f(x) + (1-\alpha) f(y) \tag{1.35}$$

A function $f : V \to \mathbb{R}$ is said to be strictly convex if for any $\alpha \in (0,1)$

$$f(\alpha x + (1-\alpha)) < \alpha f(x) + (1-\alpha) f(y) \tag{1.36}$$

If f is twice differentiable on its open domain and the Hessian $\nabla^2 f(x)$

$$\nabla^2 f(x)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \text{ with } i, j = 1, \ldots, n$$

exists for each $x \in \mathrm{dom} f$, then it is convex if and only if the Hessian matrix is positive semidefinite

$$\nabla^2 f(x) \succeq 0 \; \forall x \in \mathrm{dom} f$$

If the Hessian matrix is positive definite, i.e., $\nabla^2 f(x) \succ 0$, $\forall x \in \mathrm{dom} f$, f is strictly convex.

Let f be a function defined on a open set on the real line and let k a non negative integer. If the derivatives $f', f'', \ldots, f^{(k)}$ exist and are continuous the function f is said to be of class C^k . If the function f has derivatives of all orders is said to be of class C^∞ or smooth . The class C^0 is the class of all continuous functions, the class C^1 is the class of all differentiable functions whose derivative is continuous.

Let (M, d_M) and (N, d_N) two metric spaces and let $f : M \to N$ a function. The function f is said to be Lipschitz continuous if there exist a constant γ, called Lipschitz constant such that $\forall x, y \in M$

$$d_N(f(x), f(y)) \leq \gamma d_N(x, y) \tag{1.37}$$

The smallest constant γ is called the best Lipschitz constant. If $\gamma = 1$ the function is called a short map and if the Lipschitz constant is $0 \leq \gamma < 1$ the function is called a contraction . A function is called locally Lipschitz continuous if for every $x \in M$ if there exists a neighborhood U such that f restricted to U is Lipschitz continuous. A function f defined over M is said to be Hölder continuous or uniform Lipschitz condition of order α on M if if there exists a constant λ such that

$$d_N(f(x), f(y)) \leq \lambda d_N(x, y)^\alpha \quad \forall x, y \in M \tag{1.38}$$

1.3 Basis and Frames

A set of column vectors $\Phi = \{\phi_i\}_{i=1}^n$ is called *basis* for $\mathbb{R}^n$ if the vectors $\{\phi_1, \ldots, \phi_n\}$ are linearly independent. Each vector in $\mathbb{R}^n$ is then uniquely represented as a linear combination of basis vectors, i.e. for any vector $s \in \mathbb{R}^n$ there exist a unique set of coefficients $\alpha = \{\alpha_i\}_{i=1}^n$ such that

$$s = \Phi\alpha \tag{1.39}$$

The set of coefficients α can be univocally reconstructed as follows:

$$\alpha = \Phi^{-1}s \tag{1.40}$$

The reconstruction is particularly simple when the set of vectors $\{\phi_1, \ldots, \phi_n\}_{i=1}^n$ are orthonormal, i.e.

$$\phi_i^* \phi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{1.41}$$

In this case $\Phi^{-1} = \Phi^*$.

A generalization of the basis concept, that allow to represent a signal by a set of linearly dependent vectors is the *frame* [76].

Definition 1.1. A frame in $\mathbb{R}^n$ is a set of vectors $\{\phi_i\}_{i=1}^m \subset \mathbb{R}^n$, with $n < m$, corresponding to a matrix Φ such that there are $0 < A \leq B$ and

$$A||\alpha||_2^2 \leq ||\Phi^T\alpha||_2^2 \leq B||\alpha||_2^2 \; \forall \alpha \in \mathbb{R}^n \tag{1.42}$$

Since $A > 0$, condition (1.42), is equivalent to require that rows of Φ are linearly independent, i.e. $\text{rank}(\Phi) = n$.

If $A = B$ then the frame Φ is called *A-tight frame* , while if $A = B = 1$, then Φ is called *Parseval frame*.

By remembering the results presented in section (1.2.2), given a frame Φ in $\mathbb{R}^n$, the linear system

$$s = \Phi\alpha \tag{1.43}$$

admits an unique least square solution α_{LS}, that can be obtained by

$$\alpha_{LS} = (\Phi^T\Phi)^{-1}\Phi^T s = \Phi^\dagger s \tag{1.44}$$

Is simple to show that the solution in (1.44) is the smallest l_2 norm vector

$$||\alpha_{LS}||_2^2 \leq ||\alpha||_2^2 \tag{1.45}$$

for each coefficient vector α such that $x = \Phi\alpha$, and it is also called *least square solution*.

1.4 Sparse and Compressible Signals

We say that a vector $x \in \mathbb{R}^n$ is k-sparse when $|supp(x)| \leq k$, i.e $||x||_0 \leq k$. Let $\Sigma_k = \{\alpha : ||\alpha||_0 \leq k\}$ the set of all k-sparse vectors.
A signal admits a sparse representation in some frame Φ, if $s = \Phi\alpha$ with $||\alpha||_0 \leq k$.

In real world only few signals are true sparse, rather they can be considered *compressible* , in the sense that they can be well approximated by a sparse signal.
We can quantify the compressibility of a signal s by calculating the error between the original signal and the best approximation $\hat{s} \in \Sigma_k$

$$\sigma_k(s)_p = \min_{\hat{s}\in\Sigma_k} ||s - \hat{s}||_p \tag{1.46}$$

If $s \in \Sigma_k$ then $\sigma_k(s)_p = 0$ for any p. Another way to think about compressible signals is to consider the rate of decay of their coefficients. For many important class of signals there exist bases such that the coefficients obey a power law decay, in which case the signal are highly compressible. Specifically, if $s = \Phi\alpha$ and we sort the coefficients α_i such that $|\alpha_1| \geq |\alpha_2| \geq \cdots \geq |\alpha_m|$, then we say that the coefficients obey a power law decay if there exist constants C_1, $q > 0$ such that

$$|\alpha_i| \leq C_1 i^{-q}$$

The larger q is, the faster the magnitudes decay, and the more compressible a signal is. Because the magnitude of their coefficients decay so rapidly, compressible signals can be represented accurately by $k \ll m$ coefficients. Specifically, for such signal there exist constants C_2, $r > 0$ depending only on C_1 and q such that

$$\sigma_k(s)_2 \leq C_2 k^{-r}$$

In fact, one can show that $\sigma_k(s)_2$ will decay as k^{-r} if and only if the sorted coefficients α_i decay as $i^{r+\frac{1}{2}}$ [39].

1.5 Underdetermined Linear System and Sparsest Solution

Let consider a matrix $\Phi \in \mathbb{R}^{n\times m}$ with $n < m$ and a vector s, the system $\Phi\alpha = s$ has more unknowns than equations, and thus it has no solutions if s is not in the span of the columns of the matrix Φ, or infinitely many if s is in the span of the dictionary Φ.

We consider the sparse recovery problem, where the goal is to recover a high-dimensional sparse vector α from an observation s:

$$s = \Phi\alpha \tag{1.47}$$

A well-posed problem stems from a definition given by Jacques Hadamard. He believed that mathematical models of physical phenomena should have three properties:

- a solution must exists
- the solution is unique
- the solution's behavior hardly changes when there's a slight change in the initial condition

Problems that are not well-posed in the sense of Hadamard are termed ill-posed. Inverse problems are often ill-posed .

In ill-posed problems we desire to find a single solution of system $s = \Phi\alpha$, and in order to select one well defined solution additional criteria are needed.
A way to do this is the *regularization technique*, where a function $J(\alpha)$ that evaluates the desirability of a would-be solution α is introduced, with smaller values being preferred.
Defining the general optimization problem

$$\arg\min_{\alpha\in\mathbb{R}^m} J(\alpha) \quad \text{subject to } \Phi\alpha = s \tag{PJ}$$

where $\alpha \in \mathbb{R}^m$ is the vector we wish to reconstruct, $s \in \mathbb{R}^n$ are available measurements, Φ is a known $n \times m$ matrix is also called sensing matrix or dictionary.

It is now in the hands of $J(\alpha)$ to govern the kind of solution we may obtain. We are interested in the underdetermined case with fewer equations than unknowns, i.e. $n < m$, and ask whether it is possible to reconstruct α with a good accuracy.
By fixing $J(\alpha) = ||\alpha||_0$, we can constrain the solution of (1.47) to be sparsest as possible.

The problem can be formulated as

$$\arg\min_{\alpha\in\mathbb{R}^m} ||\alpha||_0 \text{ s.t. } \Phi\alpha = s \tag{P0}$$

where $||\alpha||_0 = |\text{supp}\{\alpha\}|$.

Problem (P0) requires searches over all subsets of columns of Φ, a procedure which clearly is combinatorial in nature and has high computational complexity. It is proved that (P0) is NP-hard [85].
In fact, under the so called Restricted Isometry Conditions[15] over the sensing matrix Φ, described with more details in the next session, the sparse recovery problem P0 [20, 24] can be relaxed to the convex l_1 problem

$$\arg\min_{\alpha\in\mathbb{R}^m} ||\alpha||_1 \text{ s.t. } \Phi\alpha = s \tag{P1}$$

where $||\alpha||_1 = \sum_{i=1}^{m} |\alpha_i|$ denotes the l_1 norm of vector α.

Problem (P1) can be reformulated as a linear programming (LP) [99] problem

$$\begin{aligned} & \min_{t\in\mathbb{R}^m} \sum_{i=1}^m t_i \\ \text{s.t.} \quad & -t_i \leq \alpha_i \leq t_i \\ & \Phi\alpha = s \qquad t_i \geq 0 \end{aligned} \tag{1.48}$$

This problem can be solved exactly with for instance interior point methods or with the classical simplex algorithm.

The linear programming formulation (1.48) results inefficient in most cases, for this reason many algorithms able to solve (P1) have been proposed in literature: for example greedies Basis Pursuit (BP) [29], Stagewise Orthogonal Matching Pursuit (StOMP) [45] and the Orthogonal Matching Pursuit (OMP) [91, 35] or other optimization methods like Least Angle Regression (LARS) [48] or the Smoothed ℓ_0 (SL0) [82, 81] that are able to find the approximated solution to the problem (P1) and (P0) respectively.

In the next session are recalled the conditions for the matrix Φ under which the sparsest solution of the problem (P0) can be recovered uniquely.

1.5.1 Null Space Property and Spark

In this section we introduce a necessary and sufficient condition for to ensure that the unique solution of (P1) is also the solution of (P0). At this regard, given $\eta \in \mathbb{R}^m$ and $\Lambda \subset \{1,2,\ldots,m\}$, we denote η_Λ the vector

$$(\eta)_i = \begin{cases} \eta_i & i \in \Lambda \\ 0 & i \notin \Lambda \end{cases}$$

Definition 1.2. A sensing matrix $\Phi \in \mathbb{R}^{n\times m}$ has the Null Space property (NSP) of order k, if there is $0 < \gamma < 1$ such that for $\eta \in \mathscr{N}_\Phi$ and $\Lambda \subset \{1,2,\ldots,m\}$, $|\Lambda| \leq k$, it holds

$$||\eta_\Lambda||_1 \leq \gamma ||\eta_{\Lambda^c}||_1 \tag{1.49}$$

Notice that to verify the Null Space Property of a sensing matrix is not an easy task, because we have to check each point in the null space with a support less than k.

A general necessary and sufficient condition [42] for solving problem (P0) is that the sensing matrix Φ has the Null Space Property [43, 62]. Moreover, in [98] it is shown that if a sensing matrix Φ has the Null Space Property it is guaranteed that the unique solution of (P1) is also the solution of (P0).
Moreover, it is proved that if Φ has the Null Space Property, the unique minimizer of the (P1) problem is recovered by basis pursuit (BP).

The column rank of a matrix Φ is the maximum number of linearly independent column vectors of Φ. Equivalently, the column rank of Φ is the dimension of the

column space of Φ.

Another criteria to assert to existence of a unique sparsest solution to a linear system is based on the concept of spark of a matrix the notion called spark[43] defined as:

Definition 1.3. Given a matrix Φ, $spark(\Phi)$ is the smallest number s such that there exists a set of s columns in Φ which are linearly-dependent.

$$spark(\Phi) = \min_{z \neq 0} ||z||_0 \text{ s.t. } \Phi z = 0$$

While computing the rank of a matrix is an easy goal, from a computational point of view, the problem of computing the spark is difficult. In fact, it has been proved to be an NP-hard problem [111]. The spark gives a simple criterion for uniqueness of sparse solutions. By definition, each vector z in the null space of the matrix $\Phi z = 0$ must satisfy $||z||_0 \geq spark(\Phi)$, since these vectors combine linearly columns from Φ to give the zero vector.

Theorem 1.1. *[43] Given a linear system $\Phi\alpha = s$, if α is a solution verifying $||\alpha||_0 < \frac{spark(\Phi)}{2}$, then α is the unique sparsest solution.*

Proof. Let β an alternative solution such that $\Phi\beta = s$, and $||\beta||_0 \leq ||\alpha||_0$. This implies that $\Phi(\alpha - \beta) = 0$. By definition of spark

$$||\alpha||_0 + ||\beta||_0 \geq ||\alpha - \beta||_0 \geq spark(\Phi) \tag{1.50}$$

Since $||\alpha||_0 < \frac{spark(\Phi)}{2}$, it follows that $||\beta||_0 \leq ||\alpha||_0 < \frac{spark(\Phi)}{2}$. By (1.50)

$$spark(\Phi) \leq ||\alpha||_0 + ||\beta||_0 < \frac{spark(\Phi)}{2} + \frac{spark(\Phi)}{2} = spark(\Phi)$$

that is impossible. □

1.5.2 Restricted Isometry Property

Compressive sensing allows to reconstruct sparse or compressible signals accurately from a very limited number of measurements, possibly contaminated with noise. Compressive sensing relies on properties of the sensing matrix such as the restricted isometry property.

The Null Space Property is necessary and sufficient condition to ensure that any k-sparse vector α is recovered as the unique minimizer of the problem (P1). When the signal s is contamined by noise it will be useful to consider strongly condition like the Restricted Isometry Property condition [22] on matrix Φ, introduced by Candes and Tao and defined as

Definition 1.4. A matrix Φ satisfies the Restricted Isometry Property (RIP) of order k if there exists a $\delta_k \in (0,1)$ such that

$$(1-\delta_k)||\alpha||_2^2 \leq ||\Phi\alpha||_2^2 \leq (1+\delta_k)||\alpha||_2^2 \tag{1.51}$$

holds for all $\alpha \in \Sigma_k$

If a matrix Φ satisfies the RIP of order $2k$, then we can interpret (1.51) as saying that Φ approximately preserves the distance between any pair of k-sparse vectors. If the matrix Φ satisfies the RIP of order k with constant δ_k, then for any $k' < k$ we automatically have that Φ satisfies the RIP of order k' with constant $\delta_{k'} \leq \delta_k$.

In Compressive Sensing [68] , random matrices are usually used as random projections of a high-dimensional but sparse or compressible signal vector onto a lower-dimensional space that with high probability contain enough information to enable signal reconstruction with small or zero error. Random matrices drawn according to any distribution that satisfies the Johnson-Lindenstrauss contraction inequality, in [12] was shown that with high probability the random sensing matrix Φ has the Restricted Isometry Property.

Proposition 1.1. *Let Φ, be a random matrix of size $n \times m$ drawn according to any distribution that satisfies the contraction inequality*

$$\mathbf{P}\left[\left|\,||\Phi\alpha||_2 - ||\alpha||_2\,\right| \leq \varepsilon||\alpha||_2\right] \leq 2e^{-nc_0(\varepsilon)}, \textit{with } 0 < \varepsilon < 1$$

where $c_0(\varepsilon) > 0$ is a function of ε. If $\Phi \sim N(0, \frac{1}{n}I)$, $c_0 = \frac{\varepsilon^2}{4} - \frac{\varepsilon^3}{6}$ is a monotonically increasing function.
For a given Gaussian matrix Φ, for any $\alpha \in \mathbb{R}^m$, Λ such that $|\Lambda| = k < n$ and any $0 < \delta < 1$, we have that

$$(1-\delta)||\alpha||_2^2 \leq ||\Phi\alpha||_2^2 \leq (1+\delta)||\alpha||_2^2$$

with a probability

$$\mathbf{P}\left[(1-\delta) \leq \frac{||\Phi\alpha||_2^2}{||\alpha||_2^2} \leq (1+\delta)\right] > 1 - 2(\frac{12}{\delta})^k e^{-nc_0(\delta/2)}$$

For large m (number of columns of Φ), estimating and testing the Restricted Isometry Constant is computational impractical. A computationally efficient, yet conservative bounds on Restricted Isometry Property can be obtained through the mutual coherence.
In the next session we introduce some bounds for of the mutual coherence of a dictionary Φ.

1.5.3 Coherence

Mutual coherence is a condition that implies the uniqueness and recoverability of the sparsest solution. While computing Restricted Isometry Property, Null Space Property and spark are NP-hard problems, the coherence og a matrix can be easily computed.

Definition 1.5. Let $\phi_1, \ldots, \phi_m$ the columns of the matrix Φ. The mutual coherence of Φ is then defined as

$$\mu(\Phi) = \max_{i<j} \frac{|\phi_i^T \phi_j|}{||\phi_i||_2 ||\phi_j||_2}$$

By Schwartz inequality, $0 \leq \mu(\Phi) \leq 1$. We say that a matrix Φ is incoherent if $\mu(\Phi) = 0$.

For $n \times n$ unitary matrices, columns are pairwise orthogonal, so the mutual coherence is obviously zero. For full rank $n \times m$ matrices Φ with $m > n$, $\mu(\Phi)$ is strictly positive, and it is possible to show [109] that

$$\mu(\Phi) \geq \sqrt{\frac{m-n}{n(m-1)}}$$

with equality being obtained only for a family of matrices named Grassmanian frames. Moreover, if Φ is a Grassmanian frame, the $spark(\Phi) = n+1$, the highest value possible.

Mutual coherence is easy to compute and give a lower bound to the spark. In order to outline this result, we briefly recall the Gershgorin's theorem for localizing eigenvalues of a matrix. Given a $n \times n$ matrix $A = \{a_{i,j}\}$, let be $R_k = \sum_{j \neq k} |a_{k,j}|$. The complex disk $z = \{z | |z - a_{k,k}| \leq R_k\}$ is said Gershgorin's disk with $(1 \leq k \leq n)$. It holds that for Gershgorin's theorem [57], every eigenvalues of A belongs to (at least one) Gershgorin's disk.

Theorem 1.2. *[43] For any matrix $\Phi \in \mathbb{R}^{n \times m}$ the spark of the matrix is bounded by a function of coherence as follows:*

$$spark(\Phi) \geq 1 + \frac{1}{\mu(\Phi)}$$

Proof. Since normalizing the columns does not change the coherence of a matrix, without loss of generality we consider each column of the matrix Φ normalized to the unit l_2-norm. Let $G = \Phi^T \Phi$ the Gram matrix of Φ.
Consider an arbitrary minor from G of size $p \times p$, built by choosing a subset of p columns from the matrix Φ and computing their sub Gram matrix M. We have $|\phi_i^T \phi_j| = 1$ if $k = j$ and $|\phi_i^T \phi_j| \leq \mu(\Phi)$ if $k = j$, as consequence $R_k \leq (p-1)\mu(\Phi)$. It follows that Gershgorin's disks are contained in $\{z | |1 - z| \leq (p-1)\mu(\Phi)\}$. If $(p-1)\mu(\Phi) < 1$, by Gershgorin's theorem, 0 can not be eigenvalues of M, hence every p-subset of columns of Φ is composed by linearly independent vectors. We

conclude that a subset of columns of Φ linearly dependent should contain $p \geq 1 + \frac{1}{\mu(\Phi)}$ elements, hence $spark(\Phi) \geq 1 + \frac{1}{\mu(\Phi)}$. $\square$

Previous result, together with theorem (1.1) gives the following condition implying the uniqueness of the sparsest solution of a linear system $\Phi\alpha = s$.

Theorem 1.3. *[43] If a linear system $\Phi\alpha = s$ has a solution α such that $||\alpha||_0 < \frac{1}{2}(1 + \frac{1}{\mu(\Phi)})$, than α is the sparsest solution.*

1.6 Algorithms for Sparse Recovery

The problem we analyze in this section is to approximate a signal s using a linear combination of k columns of the dictionary $\Phi \in \mathbb{R}^{n\times m}$. In particular we seek a solution of the minimization problem

$$\arg_{\Lambda\subset\{1,\ldots,m\}\ |\Lambda|=k} \min_{|\Lambda|=k} \min_{\alpha_\lambda} ||\sum_{\lambda\in\Lambda} \phi_\lambda \alpha_\lambda - s||_2^2 \tag{1.52}$$

fixed k with $(1 \leq k \leq m)$.

The real difficulties for solving problem (1.52) lies in the optimal selection of the index set Λ, since the "brute force" algorithm for the optimization requires to test all $\binom{m}{k} \geq \left(\frac{mk}{k}\right)$ subsets of k columns of Φ; this seems prohibitive for real instances.

The algorithms for sparse recovery can be divided in three classes:

- **Basis Pursuit methods** where is desired the sparsest solution in the ℓ_1 sense and there is an underdetermined system of linear equations $\Phi\alpha = s$ that must be satisfied exactly. The sparsest solution can be easily solved by classical linear programming algorithms.
- **Greedy methods** where an approximation of the optimal solution is found by a sequence of locally optimal choices.
- **Convex or Non-convex methods** that relax the combinatorial sparse problem to a related convex/non-convex programming and solve it with iterative methods.

1.6.1 Basis Pursuit

Basis Pursuit (BP) finds the best representation of a signal s by minimizing the ℓ_1 norm of the coefficients α of the representation. Ideally, we would like that some components of α to be zero or as close to zero as possible.
In [99] is shown how (P1), can be recasted into a linear programming problem (LP)of the form

$$\min_{\alpha\in\mathbb{R}^m} c^T\alpha \text{ s.t. } \Phi\alpha = s, \alpha \geq 0 \tag{1.53}$$

where $c^T x$ is the objective function, $\Phi\alpha = s$ is a collection of equality constraints and $\alpha \geq 0$ is a set of bounds.
The objective function of (P1) is not linear, however we can transfer the nonlinearities to the set of constraints by adding new variables $t_1, \ldots, t_n$. This gives:

$$\begin{aligned} &\min_{t\in\mathbb{R}^m} \sum_{i=1}^{m} t_i \\ &\text{s.t. } |\alpha_i| \leq t_i \quad i = 1,\ldots,m \\ &\quad \Phi\alpha = s \end{aligned}$$

Observing that $|\alpha_i| \leq t_i$ if and only if $-t_i \leq \alpha_i \leq t_i$ we can transform the original problem (P1) into the following linear programming problem subject to m inequalities constraints.

$$\begin{aligned} &\min \sum_{i=1}^{m} t_i \\ &\text{s.t. } \alpha_i \leq t_i \\ &\quad \alpha_i \geq -t_i \quad i = 1,\ldots,m \\ &\quad \Phi\alpha = s \end{aligned} \qquad (P_{\ell_1})$$

Note that $\alpha_i \leq t_i$ implies that $I\alpha - It \leq 0$ and $\alpha_i \geq t_i$ implies that $I\alpha I + t \geq 0$, thus we have the problem in LP form

$$\begin{aligned} &\min c^T t \\ &\text{s.t. } I\alpha - It \leq 0 \\ &\quad I\alpha + It \geq 0 \\ &\quad \Phi\alpha = s \end{aligned} \qquad \text{(LP)}$$

where $c = [1,1,\ldots,1]^T$.

In order to reduce the size of (LP) problem is that of using the *dual problem* . From duality theory, starting with a linear program in standard form (1.53), we can rewrite the problem as the following dual linear program:

$$\max b^T s \text{ s.t. } \Phi^T s + z = c, \alpha \geq 0 \qquad (1.54)$$

Using this equivalence we can rewrite the problem (1.54) in terms of dual variables y and v which correspond to the constraints from the primal problem without restrictions

$$\begin{aligned} &\min b^T y \\ &\text{s.t. } \Phi^T y - 2v = -e,\ 0 \leq v \leq e \end{aligned} \qquad \text{(DLP)}$$

Once the size of the original problem (P_{ℓ_1}) was reduced, the dual problem (DLP) can be solved efficiently by a linear solver.

1.6.2 Greedy Algorithms

In literature many greedy algorithms was proposed to perform sparse recovery. Many signal analysis methods look for a linear expansion of the unknown signal s in terms of functions ϕ_i.

$$s = \sum_{i=1}^{m} \alpha_i \phi_i \tag{1.55}$$

We may say that in such a way the unknown signal s is explained using atoms (functions ϕ_i) from the dictionary Φ, used for decomposition. MP algorithm finds a sub-optimal solution to the problem of an adaptive approximation of a signal in a redundant set (dictionary) of atoms. If the dictionary Φi is an orthonormal basis, then the coefficients an are given simply by the inner products of the dictionary's atoms ϕ_i with the signal $\alpha_i =< s, \phi_i >$. We would like to use a dictionary $\Phi = \{\phi_i\}_{i=1,\ldots,m}$ that would properly reveal the intrinsic properties of an unknown signal, or, almost equivalently, would give low entropy of the α_i and possibilities of good lossy compression.

We may relax the requirement of exact signal representation (1.55), and try to automatically choose the atoms ϕ_{λ_t}, optimal for the representation of a given signal s, from a redundant dictionary Φ. The expansion becomes an approximation, and uses only the functions ϕ_{λ_t} chosen from the redundant dictionary Φ. In practice, the dictionary contains orders of magnitude more candidate functions ϕ_{λ_t} than the number k of functions chosen for the representation:

$$s \approx \sum_{t=1}^{k} \alpha_t \phi_{\lambda_t} \tag{1.56}$$

A criterion of optimality of a given solution for a fixed dictionary Φ, signal s, and number of used functions k can be formulated as minimization of the reconstruction error of representation

$$\varepsilon = ||s - \sum_{t=1}^{k} \alpha_t \phi_{\lambda_t}||_2^2$$

Finding the minimum requires checking all the possible combinations (subsets) of k functions from the dictionary, which leads to a combinatorial explosion. Therefore, the problem is intractable even for moderate dictionary sizes. Matching pursuit algorithm, proposed in [77], finds a sub-optimal solution by means of an iterative procedure.

In the first step, the atom ϕ_{λ_1} which gives the largest product with the signal s is chosen from the dictionary Φ, composed of normalized atoms ($||\phi_{\lambda_t}||_2^2 = 1$). In each of the consecutive steps, the atom ϕ_i is matched to the signal r_t which is the residual left after subtracting results of previous iterations:

$$\begin{aligned} r_0 &= s \\ r_t &= < r_t, \phi_t > \phi_t + r_{t-1} \\ \phi_{\lambda_i} &= \arg \max_{\phi_{\lambda_i} \in D} | < r_t, \phi_{\lambda_i} > | \end{aligned}$$

For a complete dictionary the procedure converges to s with $k \to \infty$ [77].

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Require: - a dictionary $\Phi \in \mathbb{R}^{n \times m}$
- a signal $s \in \mathbb{R}^n$
- a stopping criteria

1: $r_0 = s, \alpha_0 = 0, \Lambda_0 = \emptyset, t = 0$
2: **while** [cond] **do**
3: $\lambda_t \in \arg\max_{j=1,\ldots m} | < r_{t-1}, \phi_j > |$ $< match >$
4: $\Lambda_{t+1} = \Lambda_t \cup \{\lambda_t\}$ $<$ *identify: if multiple maxima exist, choose only one*$>$
5: $\alpha_{t+1} = \arg\min_{\beta: supp(\beta) \subseteq \Lambda_{t+1}} ||\Phi\beta - s||_2^2$ $< update >$
6: $r_{t+1} = s - \Phi\alpha_{t+1}$
7: $t = t + 1$
8: **end while**

Ensure: $\hat{\alpha} = \alpha_t = arg\min_{\beta: supp(\beta) \subseteq \Lambda_{t+1}} ||\Phi\beta - s||_2^2$

Another greedy algorithm extensively used to find the sparsest solution of the problem (P0) is the so called Orthogonal Matching Pursuit (OMP) algorithm proposed in [35, 91] and analized by Tropp and Gilbert [113]. OMP adds a least square minimization to each step of the Matching Pursuit. The t-th approximant of s is

$$\begin{aligned} \hat{s}_t &= \arg\min_{\alpha} ||s - \sum_{t=1}^{k} \alpha_t \phi_{\lambda_t}||_2^2 \\ \text{s.t. } \hat{s} \in & \quad \{\phi_\lambda : \lambda \in \Lambda_t\} \end{aligned} \tag{1.57}$$

that can be performed by standard least squares techniques. A detailed pseudo-code of the OMP algorithm can be viewed in 1.

1.6.3 Relaxation Algorithms

An alternative way to solve the (P0) problem is to relax it to the highly discontinuous ℓ_0-norm, continuous or even smooth approximations. Examples of such relaxation is to replace the ℓ_0 norm with convex norm as the ℓ_1, with nonconvex norms like the ℓ_p for some $p \in (0,1)$ or with smooth functions like $\sum_{i=1}^{m}(1 - e^{-\lambda\alpha_i^2})$,

$\sum_{i=1}^{m} \log(1+\lambda\alpha_i^2)$ or $\sum_{i=1}^{m} \frac{\alpha_i^2}{\lambda+\alpha_i^2}$.

The problem of using ℓ_0 norm, needed for a combinatorial search of its minimization, are both due to the fact that the ℓ_0 norm of a vector is a discontinuous function of that vector.
The main idea of the Smoothed l_0 (SL0) algorithm, proposed and analyzed in [82, 81], is to approximate this discontinuous function by a suitable continuous one, and minimize it by means of minimization algorithms function i.e. with steepest descent gradient method.
The continuous functions which approximation $||\alpha||_0$, should have a parameter which determines the quality of the approximation.
Consider the single variable family of functions

$$f_\sigma(\alpha) = e^{\frac{-\alpha^2}{2\sigma^2}}$$

and note that

$$\lim_{\sigma\to 0} f_\sigma(\alpha) = \begin{cases} 1, & \text{if } \alpha = 0 \\ 0, & \text{if } \alpha \neq 0 \end{cases}$$

Defining $F_\sigma(\alpha) = \sum_{i=1}^{m} f_\sigma(\alpha_i)$ it is clear that $||\alpha||_0 \approx m - F_\sigma(\alpha)$ for small values of σ.

Algorithm 2 Smoothed ℓ_0 (SL0)

Require: - a dictionary $\Phi \in \mathbb{R}^{n\times m}$ and the Moore-Penrose pseudo inverse $\Phi^\dagger$
- a signal $s \in \mathbb{R}^n$
- a suitable decreasing sequence for $\sigma = \{\sigma_1, \ldots \sigma_T\}$
- a stopping criteria

1: $\alpha_0 = \Phi^\dagger s$
2: **while** [cond] **do**
3: $\sigma = \sigma_t$ $<$ *match* $>$
4: Maximize the function F_σ on the feasible set $\mathscr{S}$ using L iterations of the steepest ascent algorithm (followed by projection onto the feasible set):
5: **for** $j = 1, \ldots, L$ **do**
6: $\Delta\alpha = [\alpha_1 e^{\frac{-|\alpha_1|^2}{2\sigma^2}}, \ldots, \alpha_m e^{\frac{-|\alpha_m|^2}{2\sigma^2}}]$
7: $\alpha \leftarrow \alpha - \mu\Delta\alpha$ $<$ *where* μ *is a small positive constant* $>$
8: $\alpha \leftarrow \alpha - \Phi^\dagger(\Phi\alpha - s)$ $<$*orthogonal projection*$>$
9: **end for**
10: $t = t+1$
11: **end while**

Ensure: $\hat{\alpha} = \alpha_t$

We can find the minimum ℓ_0 norm solution by maximizing the $F_\sigma(\alpha)$ subject to $\Phi\alpha = s$ for a very small value of σ. The σ parameter determines how smooth the function F_σ is, the smaller value of σ, closer behavior of F_σ to ℓ_0 norm.

The idea is than to use a decreasing sequence for σ, for maximizing F_σ for each value of σ, the initial value of the maximization algorithm is the maximizer of F_σ for the previous larger value of σ. If we gradually decrease the value of σ, for each value of σ the maximization algorithm starts with an initial solution near the actual maximizer of F_σ and hence we hope to escape from getting trapped into local maxima and reach to the actual maximum for a small values of σ, which gives the ℓ_0 norm solution.

Chapter 2
Lipschitzian Mappings for Sparse Representation

Abstract In this chapter we present two new algorithms (LIMAPS and k-LIMAPS respectively) for the following Sparse Recovery Problem

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t. } s = \Phi\alpha \tag{2.1}$$

After a general introduction in sec. 2.1, in sec. 2.2 we describe the problem of Sparse Recovery faced in this work, of which the two proposed algorithms are heuristics.
In sec. 2.4 we introduce a parametric class $\{G_\lambda | \lambda \geq 0\}$ of nonlinear mappings and in sec. 2.5 we discuss some properties of iterative scheme based on these operators.
In sec. 2.5.1, for a given sequence of λ_t, the iterative scheme

$$\alpha_{t+1} = G_{\lambda_t}(\alpha_t) \tag{2.2}$$

is discussed, proving the convergence when $\sum \frac{1}{\lambda_t} < \infty$.
In sec. 2.5.2 the iterative scheme $\alpha_{t+1} = G_{\lambda_t}(\alpha_t)$ is studied in the simple bidimensional case. It is proved that, for sufficiently large λ, the sparsest solution is "near" to a fixed point of G_λ.
In sec. 2.5.3, we find a connection between the fixed point of G_λ and a relaxation of the problem (2.1). First of all, we introduce a family $\{||\cdot||_{\langle\lambda\rangle} \quad \lambda > 0\}$ of pseudonorms (see lemma 8). Then we study the relaxed problem

$$\min_{\alpha} ||\alpha||_{\langle\lambda\rangle} \quad \text{s.t. } s = \Phi\alpha \tag{2.3}$$

The main result (2.3) asserts that, under reasonable assumptions, the minima of (2.3) are asymptotically stable fixed points of G_λ.
In sec. 2.5.4 and sec. 2.6 the algorithms LIMAPS and k-LIMAPS are introduced, together a brief discussion on empirical evidence of convergence of k-LIMAPS in sec. 2.6.1.

2.1 General Considerations

Consider the underdetermined system of linear equations

$$s = \Phi\alpha^* + \eta \tag{2.4}$$

where η represents an additive noise with mean and variance

$$E[\eta] = 0 \tag{2.5}$$

$$E[\eta^2] = \sigma^2 I \tag{2.6}$$

and s the vector of observations. The matrix Φ is a $n \times m$ sensing matrix with $n < m$. The matrix Φ can be considered as the modeling operator that links α^*(the model) to the $s - \eta$ (the noise free data). We assume that the vector α has few entries significantly different from 0, i.e. α^* is approximately sparse. In this case, we can promote the sparsity via ℓ_1 regularization to overcome the singular nature of Φ. It is reasonable to recover α^*, under suitable conditions on the matrix Φ, minimizing the objective function:

$$\begin{aligned} E[\frac{1}{2}||s-\eta-\Phi\alpha||_2^2] + \lambda||\alpha||_1 & \\ &= \frac{1}{2}||s-\Phi\alpha||_2^2 + \frac{1}{2}E[(s-\Phi\alpha)\eta] + \frac{1}{2}E[\eta^2] + \lambda||\alpha||_1 \\ &= \frac{1}{2}||s-\Phi\alpha||_2^2 + \lambda||\alpha||_1 + \sigma^2 \end{aligned} \tag{2.7}$$

The minimizer is independent from σ^2, and can be obtained solving the optimization problem

$$\min_{\alpha} \frac{1}{2}||s-\Phi\alpha||_2^2 + \lambda||\alpha||_1 \tag{QP}$$

that is closely related to quadratic programming. The parameter λ represents the trade off between the ℓ_2-norm of the reconstruction error and the ℓ_1-norm of the solution vector.

When an estimate of the noise level σ^2 in the signal is available, a different formulation is often useful:

$$\min_{\{\alpha\,|\,||s-\Phi\alpha||_2^2 \le \sigma^2\}} ||\alpha||_1 \tag{BP}$$

This formulation, called Basis Pursuit (BP) denoise problem[29], minimizes the ℓ_1 norm giving a maximum misfit.

When the ℓ_1 norm estimator τ of a "good" solution is available, a third formulation is given by the Lasso [110]

$$\min_{\{\alpha | ||\alpha||_1 \leq \tau\}} \frac{1}{2}||s - \Phi\alpha||_2^2 \tag{LS}$$

2.2 Problem Formulation and Summary of Results

In this chapter we present two new algorithms for sparse recovery problem called LIMAPS and k-LIMAPS respectively. The LIMAPS algorithm tries to find the sparsest solution of the reformulation of the basis pursuit denoise problem in ℓ_0 norm

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad ||s - \Phi\alpha||_2^2 \leq \sigma \tag{BP0 Noisy}$$

In this work we consider only the model without noise ($\sigma = 0$). The problem can be rewritten as:

$$\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad s = \Phi\alpha \tag{BP0}$$

In sec. 2.5.3 we introduce a family $\{||\cdot||_{\langle\lambda\rangle} | \quad \lambda > 0\}$ of pseudonorm 2.5.3 that approximate $||\cdot||_0$ for large values of λ, and we consider the relaxation of BP0:

$$\min_{\alpha} ||\alpha||_{\langle\lambda\rangle} \quad \text{s.t. } s = \Phi\alpha \tag{REL}$$

In sec. 2.4 we consider a family of mappings $\{G_\lambda \quad | \quad \lambda > 0\}$, sec. 2.4 and we prove that, under reasonable conditions, a solution of (REL) is an asymptotically stable fixed point of G_λ 2.3.
This fact suggests, given a suitable sequence $\{\lambda_t\}$ with $\lim_{t\to\infty} = \infty$. to consider the iterative scheme called LIMAPS

$$\alpha_{t+1} = G_{\lambda_t}(\alpha_t)$$

as heuristics for solving (BP0). This iterative schema in the base of the algorithm LIMAPS presented in 2.5.4.

The second proposed algorithm, called k-LIMAPS , is an heuristics for the following problem:

$$\min_{\alpha} \frac{1}{2}||s - \Phi\alpha||_2^2 \quad \text{s.t.} \quad ||\alpha||_0 \leq k \tag{LS0}$$

2.3 A Source of Inspiration

In recent years, a new family of numerical algorithms has been gradually built. This family, addressing the optimization problem (QP), is the *Iterative Shrinkage* algorithms [17, 86, 13].
Roughly speaking, in these iterative methods, each iteration comprises a multipli-

cation by Φ and its adjoint, along with a scalar shrinkage step on the obtained result.

If Φ is unitary, the minimization of (QP) can be founded easily, with a closed form solution that leads to shrinkage. Using the identity $\Phi\Phi^T = I$, and exploiting the fact that ℓ_2 norm is unitary invariant, we have:

$$\begin{aligned} f(x) =& \frac{1}{2}||s-\Phi\alpha||_2^2 + \lambda\rho(\alpha) \\ =& \frac{1}{2}||\Phi(\Phi^T s-\alpha)||_2^2 + \lambda\rho(\alpha) \\ =& \frac{1}{2}||\Phi^T s-\alpha||_2^2 + \lambda\rho(\alpha) \end{aligned}$$

and denoting $\alpha^0 = \Phi^T s$, we get:

$$\begin{aligned} f(x) =& \frac{1}{2}||\alpha^0-\alpha||_2^2 + \lambda\rho(\alpha) \\ =& \sum_{k=1}^{m}\left[\frac{1}{2}(\alpha_k^0-\alpha_k)^2 + \lambda\rho(\alpha_k)\right] \\ =& \sum_{k=1}^{m} g(\alpha_k^0,\alpha_k) \end{aligned}$$

The minimization of the scalar function of the form $g(x,a) = \frac{1}{2}(x-a)^2 + \lambda\rho(x)$ with respect to x requires that we either zero the gradient in the case of smooth function $\rho(\cdot)$, or show that the sub-gradient of g contains the zero, for non differentiable $\rho(\cdot)$. The problem can be solved, by finding $\hat{x}_{\text{opt}} = S_{\rho,\lambda}(a)$ that is the global minimizer of the scalar objective function $g(x,a)$.

Let us now consider the convex function $\rho(x) = |x|^p$ with $1 \leq p < 2$. The scalar objective function $g(x,a)$ then can be rewritten as

$$g(x,a) = \frac{1}{2}(x-a)^2 + \lambda|x|^p$$

The condition of minimum is thean expressed by

$$g'(x,a) = x-a+p\lambda|x|^{p-1}\text{sgn}(x) = 0$$

from which we can derive

$$a = x+\lambda p|x|^{p-1}\text{sgn}(x) \tag{2.8}$$

Inverting the equation (2.8) we obtain

$$\hat{x}_{\text{opt}} = S_\lambda(a) \tag{2.9}$$

that maps the input a in the global minimizer $\hat{x}_{\text{opt}}$. Notice that:

(a) if a is near the origin, then $S_\lambda(a) \approx 0$. This implies that if $|S_\lambda(a)| \leq \varepsilon$ then $|\alpha| \approx p\lambda|x|^{p-1}$ and then $|S_\lambda(a)| \approx (p\frac{1}{\lambda}a)^2$.
(b) if a is large (tending to infinity), $S_\lambda(a) \approx a - p\lambda|a|^{p-1}\text{sgn}(a)$, implying $|a| > |S_\lambda(a)|$.

The function $S_\lambda(a)$ results to be a *shrinkage function*.

2.4 Sparsity Promotion Mappings

The aim of this section is to establish asymptotic fixed point approaches for a thresholding-based iterative process for model selection relying on a family of shrinkage function represented by the uniformly Lipschitzian nonlinear mappings $\mathscr{F} = \{f_\lambda \mid \lambda \in \mathbb{R}^+\}$, where the choice of λ is important in controlling the shrinking effects, i.e., to drive the search towards the sparsest solutions. It provides then a fusion with the classical orthogonal projectors built on the Moore-Penrose pseudo-inverse, in order to restore the feasibility at each iteration step.

2.4.1 Uniformly Lipschitzian Shrinking Mappings

The key feature of shrinking functions is to attenuate coefficients adaptively with respect to scale and time, taking into proper account the general strength constraints of the signal. This task is committed to classes of smooth sigmoidal based shrinkage maps that may be used to promote the sparsity. These classes of shrinkage functions introduce a little variability among coefficients with small amplitude, performing low attenuation of coefficients with high amplitude and stronger attenuation on coefficients with small amplitudes. These functions depend on a parameter λ that make possible the control of the attenuation degree imposed to coefficients. Moreover they have the following features:

smoothness: the shrinkage function induces small variability among data with close values.
shrinkage: a strong attenuation is imposed for small values, conversely weak attenuation is imposed for large values.
vanishing attenuation: attenuation decreases to zero when the amplitude of the coefficient tends to infinity.
Lipschitzianity: given a class of functions

$$\{f_\lambda \mid f_\lambda : \mathbb{R} \to \mathbb{R}, \lambda \in \Lambda\}$$

the class is called Lipschitzian continuous with Lipschitz constant $K \geq 0$ if for all $\lambda \in \Lambda$, it holds

$$\frac{|f_\lambda(x)-f_\lambda(y)|}{|x-y|} \leq K \tag{2.10}$$

for all x and y in $\mathbb{R}$, with $x \neq y$.

Name	$f(x)$	$\frac{\mathrm{d}f(x)}{\mathrm{d}x}$
Exponential	$x(1-e^{-\lambda\|x\|})$	$e^{-\lambda\|x\|}(\lambda\|x\|-1)+1$
Exponential 2	$x(\frac{2}{1+e^{-\lambda\|x\|}}-1)$	$\frac{2\lambda\|x\|e^{\lambda\|x\|}+e^{2\lambda\|x\|}-1}{(e^{\lambda\|x\|}+1)^2}$
Hyperbolic Tangent	$x\tanh(\lambda\|x\|)$	$\tanh(\lambda\|x\|)+\lambda\|x\|\sec h^2(\lambda\|x\|)$
Absolute Value	$x(\frac{\lambda\|x\|^2}{1+\lambda\|x\|^2})$	$\frac{\lambda x^2(\lambda x^2+3)}{(\lambda x^2+1)^2}$
Square Root	$x\frac{\lambda\|x\|}{\sqrt{1+\lambda\|x\|^2}}$	$\frac{\lambda\|x\|(\lambda x^2+2)}{(\lambda x^2+1)^{\frac{3}{2}}}$
Error Function	$xerf(\lambda\|x\|)$	$erf(\lambda\|x\|)+\lambda\|x\|\frac{d}{d\lambda\|x\|}erf(\lambda\|x\|)$
Gudermannian	$x\frac{2}{\pi}atan(sinh(\frac{\pi}{2}\lambda\|x\|))$	$\lambda\|x\|sech(\frac{\pi}{2}\lambda\|x\|)+\frac{2}{\pi}atan(sinh(\frac{\pi}{2}\lambda\|x\|))$

Table 2.1 Sparsity promotion mappings $f(x)$ and first derivatives $\frac{df(x)}{dx}$.

In table 2.1 we list some Lipschitzian smooth sigmoidal based shrinkage maps and their derivatives that have the properties required above.

Let us introduce some property for a uniformly Lipschitzian mapping based on a family of nonlinear maps

$$\mathscr{F} = \{f_\lambda \mid \lambda \in \mathbb{R}^+\}, \tag{2.11}$$

where the choice of λ is important in controlling the sparsity effects. In particular, we restrict all our analysis on the exponential family, for its well-known characteristics.

For $\lambda > 0$, let $f_\lambda : \mathbb{R} \to \mathbb{R}$ be the shrinkage map

$$f_\lambda(x) = x(1-e^{-\lambda|x|}). \tag{2.12}$$

Let be $\mathscr{F} = \{f_\lambda \mid \lambda > 0\}$. The function f_λ is an odd function with continuous derivative $f'_\lambda(x) = (\lambda|x|-1)e^{-\lambda|x|}+1$. Since $\sup_{x\in\mathbb{R}}|f'_\lambda(x)| = 1+e^{-2}$, as a direct consequence of the intermediate value theorem $|f_\lambda(x)-f_\lambda(y)| \leq (1+e^{-2})|x-y|$, for each $\lambda > 0$ and $x,y \in \mathbb{R}$. Thus, mapping (2.12) is uniformly Lipschitzian with respect to λ with Lipschitz constant $1+e^{-2}$. Moreover, given that $|f'_\lambda(x)| < 1$ on the interval $(-1/\lambda, 1/\lambda)$, the mapping (2.12) is contractive within that interval with fixed point

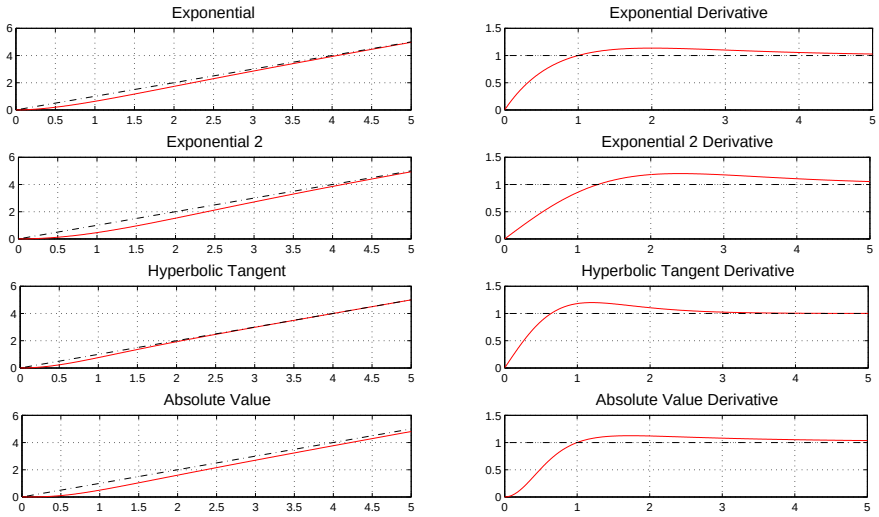

Fig. 2.1 Sparsity promotion mappings $f(x)$ and first derivatives $\frac{df(x)}{dx}$.

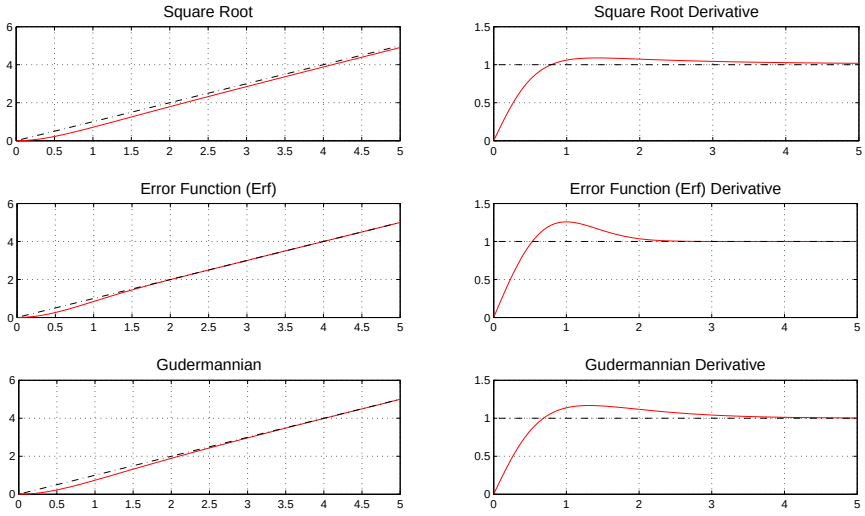

Fig. 2.2 Sparsity promotion mappings $f(x)$ and first derivatives $\frac{df(x)}{dx}$.

at the origin [10]. It obeys to the shrinkage rule since $|f_\lambda(x)| \leq |x|$ for all $x \in \mathbb{R}$ and it is nondecreasing, as shown by its plotting in Figure 2.3. Just for comparison, in figure is displayed also the soft threshold function $\mathscr{S}_\tau(x) = \max(|x| - \tau, 0)\,\mathrm{sgn}(x)$ which arises frequently in sparse signal processing and compressed sensing. The latter, differently from (2.12), is discontinuous and $\mathscr{S}_\tau(x) = 0$ iff $|x| \leq \tau$.

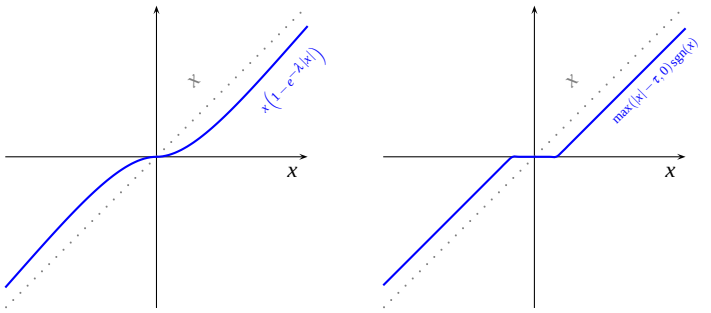

Fig. 2.3 The graphs of shrinking function (2.12) and the well known soft threshold function.

To deal with high dimensional data, we extend mapping (2.12) to many dimensions , obtaining the one-parameter family of nonlinear mappings $\mathscr{F}_m = \{F_\lambda : \mathbb{R}^m \to \mathbb{R}^m \mid \lambda \in \mathbb{R}^+\}$, where the k-th component $(F_\lambda(x))_k$ of F_λ

$$(F_\lambda(x))_k = f_\lambda(x),\ (1 \leq k \leq m) \tag{2.13}$$

Analogously to the scalar case, the function $[f_\lambda(x_1)/x_1, \ldots, f_\lambda(x_m)/x_m]$ represents a symmetric sigmoid function in m dimensions, where larger values of λ give sharper sigmoids, in the limit becoming a Heaviside multi-dimensional step function.

Now we come back to the problem

$$\Phi\alpha = s$$

where Φ is an $n \times m$ sensing matrix of full rank, and s is the vector of observations. The set of possible solutions $\mathscr{A}_{\Phi,s} = \{x \mid \Phi x = s\}$ is the affine space $\nu + \mathscr{N}_\Phi$, where $\mathscr{N}_\Phi = \{y \mid \Phi y = 0\}$ is the null space of Φ and ν is the solution with minimum ℓ_2 norm. We recall that $\nu = \Phi^\dagger s$, where $\Phi^\dagger = (\Phi^T\Phi)^{-1}\Phi^T$ is the Moore-Penrose pseudo inverse of Φ.

Let P be the projector onto $\mathscr{N}_\Phi$. For each x in $\mathbb{R}^m$ is projected in a point $y \in \mathscr{A}_{\Phi,s} = \nu + \mathscr{N}_\Phi$ as follow:

$$y = Px + \nu \tag{2.14}$$

These early assumptions suggest to define a new mapping by composing the shrinking (2.13) and the feasibility (2.14). As a consequence, we get the self-mapping family $G_\lambda : \mathscr{A}_{\Phi,s} \to \mathscr{A}_{\Phi,s}$, which has the form

$$G_\lambda(\alpha) = PF_\lambda(\alpha) + \nu, \tag{2.15}$$

Since $\alpha \in \mathscr{A}_{\Phi,s}$, it holds $\alpha = x + \nu$ for a suitable $x \in \mathscr{N}_\Phi$; as a consequence:

$$\begin{aligned} G_\lambda(\alpha) &= P(x+\nu) + \nu - P(\alpha \odot e^{-\lambda|\alpha|}) \\ &= x + \nu - P(\alpha \odot e^{-\lambda|\alpha|}) \\ &= \alpha - P(\alpha \odot e^{-\lambda|\alpha|}) \end{aligned} \tag{2.16}$$

2.5 Iterative scheme and Fixed Point Characterization

The aim of this section is to show that a fixed point iterative scheme involving mapping (2.15) becomes an effective procedure to find good approximation of the sparsest solution to the linear system (2.4), providing that a suitable sequence of λ_t would be supplied.

By denoting with $T_\lambda : \mathbb{R}^m \to \mathscr{N}_\Phi$ the mapping defined by $T_\lambda(\alpha) = P\alpha \odot e^{-\lambda|\alpha|}$, a fixed point α^* of mapping (2.15) satisfies:

$$\alpha^* = G_\lambda(\alpha^*) \quad \Longleftrightarrow \quad T_\lambda(\alpha^*) = 0. \tag{2.17}$$

Thus, each α^* in the set $\operatorname{Fix} G_\lambda \equiv \mathscr{K}_{T_\lambda} \subseteq \mathscr{A}_{\Phi,s}$ uniquely corresponds to the point $\alpha^* \odot e^{-\lambda|\alpha^*|} \in \mathscr{N}_\Phi$, being $\mathscr{K}_{T_\lambda}$ and $\mathscr{N}_\Phi$ the kernel of T_λ and the null space of projector P respectively.

For mapping (2.15) we study the convergence of sequences of successive approximations (Picard process) with a given initial value. In particular, we take into exam two cases: the first with fixed parameter λ and the latter providing a sequence $\{\lambda_t\}_{t\geq 0}$, tending to become arbitrarily large as $t \to \infty$. Both the analysis will be useful in the study of minimizers for sparsity promotion functionals.

2.5.1 Convergence for increasing sequences of λ

In this case, for a given incresing sequence of real numbers $\{\lambda_t\}_{t\geq 0}$, we define the iterates as

$$\begin{cases} \alpha_0 = \alpha \in \mathscr{A}_{\Phi,s} \\ \alpha_t = G_{\lambda_{t-1}}(\alpha_{t-1}), \quad t \geq 1 \end{cases} \tag{2.18}$$

and we study the convergence of the sequence $\{\alpha_t\}_{t\geq 0}$.

The first lemma provides an inductive form of the general term α_t.

Lemma 1 *Let $\{\alpha_t\}$ be the sequence generated by (2.18), then it holds:*

$$\alpha_t = P\alpha - P\left(\sum_{k=0}^{t-1} \alpha_k \odot e^{-\lambda_k|\alpha_k|}\right) + \nu\,, \quad t \geq 1. \tag{2.19}$$

Proof. To prove the lemma we proceed by induction on n. The case $t = 1$ is trivial. For $t > 1$, by definition we have

$$\alpha_t = G_{\lambda_{t-1}}(\alpha_{t-1}) = P\alpha_{t-1} - P\left(\alpha_{t-1} \odot e^{-\lambda_{t-1}|\alpha_{t-1}|}\right) + \nu\,.$$

Assume inductively that the equality (2.19) is true for every positive integer less than t. Thus, the induction hypothesis and the idempotency of P imply that

$$\begin{aligned} P\alpha_{t-1} &= P^2\alpha - P^2\left(\sum_{k=0}^{t-2} \alpha_k \odot e^{-\lambda_k|\alpha_k|}\right) + P\nu \\ &= P\alpha - P\left(\sum_{k=0}^{t-2} \alpha_k \odot e^{-\lambda_k|\alpha_k|}\right), \end{aligned}$$

because

$$P\nu = (I - \Phi^\dagger\Phi)\Phi^\dagger s = \Phi^\dagger s - \Phi^\dagger\Phi\Phi^\dagger s = 0.$$

Hence, the proposition holds for t, and this completes the proof. □

By limiting the sum of reciprocals of $\{\lambda_t\}$, the sequence $\{\alpha_t\}$ results to be convergent, as stated in the following lemma.

Lemma 2 *Let $\{\lambda_t\}$ be a real positive sequence. Then, if $\sum_{t=0}^{\infty} 1/\lambda_t < +\infty$, the sequence $\{\alpha_t\}$ converges as $t \to +\infty$.*

Proof. First of all observe that the univariate mapping

$$x \mapsto |x|e^{-\lambda|x|} \leq \frac{1}{e\lambda}$$

is bounded, which implies

$$\left\| \alpha_k \odot e^{-\lambda_k|\alpha_k|} \right\| \leq \frac{\sqrt{m}}{e}\frac{1}{\lambda_k}.$$

For any $n > n'$ and since $||P|| < 1$ we have:

$$\begin{aligned} ||\alpha_n - \alpha_{n'}|| &= ||P(\sum_{k=n'}^{n-1} \alpha_k \odot e^{-\lambda_k|\alpha_k|})|| \\ &\leq ||\sum_{k=n'}^{n-1} \alpha_k \odot e^{-\lambda_k|\alpha_k|}|| \\ &\leq \sum_{k=n'}^{n-1} ||\alpha_k \odot e^{-\lambda_k|\alpha_k|}|| \\ &\leq \frac{\sqrt{m}}{e}\sum_{k=n'}^{n-1}\frac{1}{\lambda_k} \end{aligned} \tag{2.20}$$

For all $\varepsilon > 0$ therefore, there always exists a index γ such that, for $n, n' > \gamma$ it holds that

$$|\sum_{k=n'}^{n} \frac{1}{\lambda_k}| \leq \frac{e}{\sqrt{m}}\varepsilon$$

It follows that if $\sum_{k=n'}^{n} \frac{1}{\lambda_k}$ converges also $\{\alpha_k\}$ is convergent.

Given the previous bound, we show now that the sequence (2.18) is a Cauchy sequence.

We conclude that the hypothesis done on the sequence $\{\lambda_t\}$ is sufficient to get the convergence in norm of $\sum_{k=0}^{+\infty} \alpha_k \odot e^{-\lambda_k|\alpha_k|}$. □

2.5.2 Fixed points of G_λ in the bidimensional case

In this section we discuss the fixed points of G_λ in the simple bidimensional case.

Consider the family of Lipshitzian functions with parameter $\lambda > 0$

$$G_\lambda : \mathscr{A}_{\Phi,s} \to \mathscr{A}_{\Phi,s}$$

given by

$$G_\lambda(\alpha) = \alpha - P(\alpha \odot e^{-\lambda|\alpha|})$$

where P is the projector on $\mathscr{N}_\Phi$. We can assume without loss of generality that $||v|| = 1$ because

$$G_\lambda\left(\frac{\alpha}{||v||}\right) = \frac{1}{||v||} G_{\frac{\lambda}{||v||}}(\alpha)$$

The linear system in the bidimensional case, is then:

$$(\phi_1, \phi_2)\begin{bmatrix}\alpha_1 \\ \alpha_2\end{bmatrix} = s$$

where $s \in \mathbb{R}$ is fixed. Without loss of generality, we can consider $||(\phi_1, \phi_2)|| = 1$, which can be written in terms of a unique parameter θ, as for instance $(\phi_1, \phi_2) = (-\sin\theta, \cos\theta)$ and $\mathscr{N}_\Phi = \{(z,w) \,|\, \phi_1 z + \phi_2 w = 0\} = \{x(\cos\theta, \sin\theta) \,|\, x \in \mathbb{R}\}$. The solutions $\begin{bmatrix}\alpha_1 \\ \alpha_2\end{bmatrix}$ are in the affine space $v + \mathscr{N}_\Phi$, with $v = \begin{bmatrix}-\sin\theta \\ \cos\theta\end{bmatrix} s$. Without loss of generality, we can consider the case $s = 1$, then $v = \begin{bmatrix}-\sin\theta \\ \cos\theta\end{bmatrix}$.

The problem is transformed into the study of the fixed points of $G_\lambda(\alpha)$, with fixed λ. Consider the bijection $\psi : \mathbb{R} \to \mathscr{A}_{\Phi,s}$ given by:

$$x \mapsto x(\cos\theta, \sin\theta) + (-\sin\theta, \cos\theta)$$

With respect to the bijection, the operator $G_\lambda : \mathscr{A}_{\Phi,s} \to \mathscr{A}_{\Phi,s}$ becomes the operator $S_\lambda : \mathbb{R} \to \mathbb{R}$. The picture in figure 2.4 shows the above bijection in which the point

$xu = x(\cos\theta, \sin\theta)$ is mapped on $\alpha = (x\cos\theta, \sin\theta) + (-\sin\theta, \cos\theta)$ and $G_\lambda(\alpha)$ is mapped on $S_\lambda(x)(\cos\theta, \sin\theta)$.

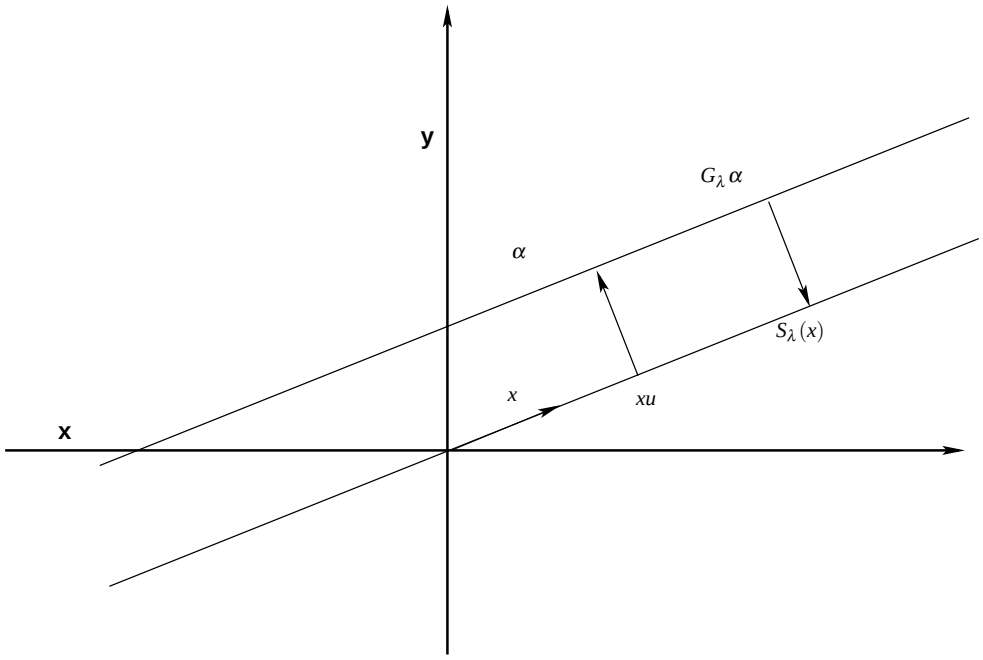

Fig. 2.4 Representation of the bijection in which the point $xu = x(\cos\theta, \sin\theta)$ is mapped on $\alpha = (\cos\theta, \sin\theta) + (-\sin\theta, \cos\theta)$ and the map $G_\lambda(\alpha)$ is mapped on $S_\lambda(x)$.

An explicit write for S_λ is given by:

$$x \mapsto \langle xu + v - (xu + v) \odot e^{-\lambda|xu+v|}), u\rangle$$

where $\langle\cdot,\cdot\rangle$ denotes the usual dot product in $\mathbb{R}^2$. Thus, in this setting, we characterize the fixed points of the transformation

$$\begin{aligned} S_\lambda(x) = x - \cos\theta(x\cos\theta - \sin\theta)e^{-\lambda|x\cos\theta-\sin\theta|} - \\ \sin\theta(x\sin\theta + \cos\theta)e^{-\lambda|x\sin\theta+\cos\theta|} \end{aligned} \tag{2.21}$$

varying the parameters λ and θ.

The equation for the determination of the fixed points is:

$$e^{-\lambda|x\cos\theta-\sin\theta|}e^{-\lambda|x\sin\theta+\cos\theta|} = \frac{1 + x\tan\theta}{1 - x\cot\theta} \tag{2.22}$$

For sake of simplicity, we suppose that $\sin\theta, \cos\theta > 0$. In order to identify the interval where fixed points lie, we obtain:

Lemma 3 *If* $x \geq \tan\theta$ *then* $0 < S_\lambda(x) < x$

Proof. Since $x \geq \frac{\sin\theta}{\cos\theta}$, then $x\cos\theta - \sin\theta \geq 0$, and $x\cos\theta + \sin\theta > 0$, then

$$S_\lambda(x) - x = -\cos\theta(x\cos\theta - \sin\theta)e^{-\lambda(x\cos\theta - \sin\theta)} - \sin\theta(x\sin\theta + \cos\theta)e^{-\lambda(x\sin\theta + \cos\theta)} < 0$$

which concludes the proof. □

Under the same hypothesis it holds, with similar proof:

Lemma 4 *If* $x \leq -\cot\theta$ *then* $0 > S_\lambda(x) > x$

By lemma 3 and 4, the presence of fixed points out of the interval $(-\cot\theta, \tan\theta)$ can be excluded.
In this case we can therefore conclude that the fixed points of the transformation $S_\lambda(x)$ must be in the interval $(-\cot\theta, \tan\theta)$. We can drop the absolute values, obtaining:

$$S_\lambda(x) = x - \cos\theta(x\cos\theta - \sin\theta)e^{-\lambda(x\cos\theta - \sin\theta)} - \sin\theta(x\sin\theta + \cos\theta)e^{-\lambda(x\sin\theta + \cos\theta)}$$

The equation for the fixed points is then:

$$e^{-\lambda(\sin\theta - \cos\theta)}e^{\lambda x(\cos\theta + \sin\theta)} = \frac{1 + x\tan\theta}{1 - x\cot\theta}$$

The function

$$y_1(x) = \frac{1 + x\tan\theta}{1 - x\cot\theta} \tag{2.23}$$

is a rectangular hyperbola that cuts the x-axis in $x = -\cot\theta$. The hyperbola has the vertical asymptote $x = \tan\theta$ and intersects the y-axis in the point $y = 1$. The function

$$y_2(x) = e^{-\lambda(\sin\theta - \cos\theta)}e^{+\lambda(\sin\theta + \cos\theta)x} \tag{2.24}$$

is a growing exponential function.
We can remark:

- If $\sin\theta > \cos\theta$, the function (2.24) intersects the ordinate axis in a point with coordinate $\hat{y} < 1$. This guarantees the existence of a fixed point $\hat{x} < 0$; if $\lambda \gg 1$ it holds that $\hat{x} \approx -\cot\theta$. We conclude observing that, for λ sufficiently large, $\hat{x}$ is a sparse solution.
- If $\sin\theta < \cos\theta$, the function (2.24) intersects the ordinate axis in a point of coordinate $\hat{y} > 1$. This guarantees the existence of a fixed point $\hat{x} > 0$; if $\lambda \gg 1$ it holds that $\hat{x} \approx \tan\theta$.

The point $\hat{x}$ is asymptotically stable if $|S'(\hat{x})| < 1$. If $\lambda \gg 1$ it holds that $\hat{x} \approx \tan\theta$ and

$$S'(\hat{x}) \approx 1 - \cos^2\theta - sin^2\theta(1 - \lambda\tan\theta)e^{-\lambda\tan\theta} \tag{2.25}$$

It follows that, if $\lambda \gg 1$, $|S'(\hat{x})| < 1$, hence the point $\hat{x}$ is asymptotically stable. Similarly, if $\sin\theta > \cos\theta$, the fixed point $\hat{x} = -\cot\theta$ is asymptotically stable for $\lambda \gg 1$.

We now analyze in details, for illustrative purposes, the case $\theta = \frac{\pi}{4}$. The function S_λ can be rewritten as

$$S_\lambda(x) = x - \frac{1}{2}(x-1)e^{-\frac{\lambda}{\sqrt{2}}|1-x|} - \frac{1}{2}e^{-\frac{\lambda}{\sqrt{2}}|1+x|} \tag{2.26}$$

Posing for simplicity $t = \frac{\lambda}{\sqrt{2}}$, we have

$$S_\lambda(x) = x - \frac{1}{2}(x-1)e^{-t|1-x|} - \frac{1}{2}e^{-t|1+x|}. \tag{2.27}$$

The fixed points are in the interval $(-1,1)$ and they are solutions of the equation

$$e^{2tx} = \frac{1+x}{1-x} \tag{2.28}$$

We observe that

- for each t, $x_1 = 0$ is fixed point
- if $\frac{d}{dx}e^{2tx}|_{x=0} > \frac{d}{dx}\frac{1+x}{1-x}|_{x=0}$ iff $t > 1$. In this case there are two other fixed points x_2 and x_3 such that $-1 < x_2 < 0 < x_3 < 1$.
- if $0 < t \leq 1$ it exists only the fixed point $x_1 = 0$.

To study the fixed point stability, we calculate:

$$S'(x) = 1 - \frac{1}{2}e^{-t(1-x)(1+t(x-1))} - \frac{1}{2}e^{-t(1+x)(1-t(x+1))} \tag{2.29}$$

Let $S'(x_1) = S'(0) = 1 - e^{-t}(1-t)$. If $t > 1$ the fixed point x_1 is instable and if $0 < t < 1$ the point x_1 is asymptotically stable. Since for t sufficiently high, $x_2 \approx -1$ and $x_3 \approx 1$, then:

$$S'(x_2) \approx S'(1) = \frac{1}{2}(1 - e^{-2t}(1-2t)) \tag{2.30}$$

If $t > 1$ is $-e^{-2t(1-2t)} < e^{-2}$, $S'(1) \leq \frac{1}{2}(1+e^{-2}) < 1$. The point x_3, that exists for $t > 1$, is asymptotically stable. Similarly for the fixed point x_2.

Now we return to the original transformation in the affine space $\mathscr{A}_{\Phi,s}$

$$G_\lambda(\alpha) = \alpha - P(\alpha \odot e^{-\lambda|\alpha|}) \tag{2.31}$$

In the bijection between $\mathscr{A}_{\Phi,s}$ and $\mathbb{R}$, given by $\alpha = xu + v$, with $u = (\cos\theta, \sin\theta)$ and $v = (-\sin\theta, \cos\theta)$, for sufficiently large λ, the point $\hat{\alpha} = xu + v$ is the asymptotically stable fixed point of the iterative system generated by G_λ. We observe that:

- if $\hat{x} \approx \tan\theta$, then

$$\hat{\alpha} \approx \tan\theta(\cos\theta, \sin\theta) + (-\sin\theta, \cos\theta) = (0, \frac{1}{\cos\theta})$$

- if $\hat{x} \approx \cot\theta$, then

$$\hat{\alpha} \approx -\cot\theta(\cos\theta, \sin\theta) + (-\sin\theta, \cos\theta) = (-\frac{1}{\sin\theta}, 0)$$

The fixed points $\hat{\alpha}$, are then very close to the minimizers of the problem (BP0), as $\lambda \to +\infty$ they satisfy:

$$\hat{\alpha} = \min_{\alpha \in \mathscr{A}_{\Phi,s}} ||\alpha||_0 \tag{2.32}$$

2.5.3 Sparsity Minimization

In this subsection, our objective is to study the sparsity minimization property of the fixed points of the iterative system. To do this, let us now consider the family of functions

$$g_\lambda(z) = 1 - e^{-\lambda|z|}(1 + \lambda|z|) \tag{2.33}$$

with parameter $\lambda > 0$. We can easily verify that the first derivative of the function (2.33) is given by

Lemma 5

$$g'_\lambda(z) = \lambda^2 z e^{-\lambda|z|}$$

A graphical representation of the function $g_\lambda(z)$ can be obtained observing that $g_\lambda(z)$ is even, is increasing for $z \geq 0$, it holds that $0 \leq g_\lambda(z) < 1$, if $|z| \ll \frac{1}{\lambda}$ is $g_\lambda(z) \approx \lambda^2 z^2$ while $|z| \gg \frac{1}{\lambda}$ is $g_\lambda(z) \approx 1$. In particular, $g'_\lambda(z)$ and $g''_\lambda(z)$ are continuous functions.

Respect to the parameter λ, if holds that:

Lemma 6

$$\lim_{\lambda \to \infty} g_\lambda(z) = \begin{cases} 0 & z = 0 \\ 1 & z \neq 0 \end{cases} \tag{2.34}$$

A further property of (2.33) is the following:

Lemma 7 *$\forall x, y \in \mathbb{R}$ it holds:*

$$g_\lambda(x+y) \leq g_\lambda(x) + g_\lambda(y) \tag{2.35}$$

Proof. Since $g_\lambda(z) = g_1(\lambda z)$, we can limit ourselves to the study in the case of $\lambda = 1$. Considering the function

$$G(x,y) = e^{-x-y}(1+x+y) - e^{-x}(1+x) - e^{-y}(1+y) + 1$$

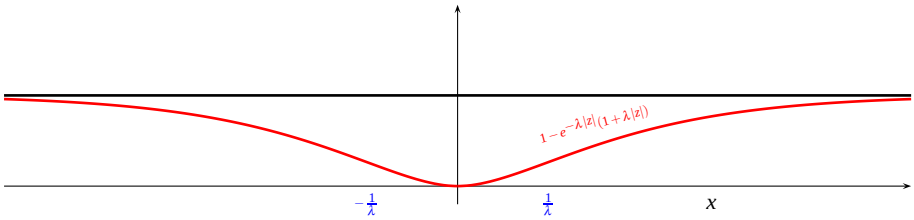

Fig. 2.5 Parametric function g_λ.

$\forall x,y \geq 0$. *We observe that the point* $(0,0)$ *is the minimum of* $G(x,y)$*, then:*

$$G(x,y) \geq G(0,0) = 0$$

i.e.

$$e^{-x-y}(1+x+y) - e^{-x}(1+x) - e^{-y}(1+y) + 1 \geq 0$$

This is equivalent to

$$1 - e^{-x-y}(1+x+y) \leq 1 - e^{-x}(1+x) + 1 - e^{-y}(1+y) \tag{2.36}$$

We conclude that

$$\begin{aligned} g_1(x+y) &= 1 - e^{-|x+y|}(1+|x+y|) \\ &= 1 - e^{-|x|-|y|}(1+|x|+|y|) \\ &= 1 - e^{-|x|}(1+|x|) + 1 - e^{-|y|}(1+|y|) \\ &= g_1(x) + g_1(y) \end{aligned}$$

□

Let us now study the functional with parameter $\lambda > 0$:

$$||\cdot||_{<\lambda>} : \mathbb{R}^m \to \mathbb{R}^+ \tag{2.37}$$

where $||(\alpha_1,\ldots,\alpha_m)||_{<\lambda>} = \sum_{k=1,\ldots,m} g_\lambda(\alpha_k)$.

The principal properties of $||\cdot||_{<\lambda>}$ are summarized in the following points:

Lemma 8 $\forall \lambda > 0$, $||\cdot||_{<\lambda>}$ *is a pseudonorm*

Proof. • *For all* $\alpha, \beta \in \mathbb{R}^m$ *it holds that*

$$\begin{aligned} ||\alpha||_{<\lambda>} = 0 &\iff \sum_{k=1,\ldots,m} g_\lambda(\alpha_k) = 0 \\ &\iff g_\lambda(\alpha_k)(1 \leq k \leq m) \\ &\iff \alpha = 0 \end{aligned} \tag{2.38}$$

- *For all $\alpha, \beta \in \mathbb{R}^m$ it holds that*

$$||\alpha+\beta||_{<\lambda>} \leq ||\alpha||_{<\lambda>} + ||\beta||_{<\lambda>}$$

Infact:

$$\begin{aligned} ||\alpha+\beta||_{<\lambda>} &= \sum_{k=1,\ldots,m} g_\lambda(\alpha_k+\beta_k) \\ &\leq \sum_{k=1,\ldots,m} g_\lambda(\alpha_k) + g_\lambda(\beta_k) \\ &= \sum_{k=1,\ldots,m} g_\lambda(\alpha_k) + \sum_{k=1,\ldots,m} g_\lambda(\beta_k) \\ &= ||\alpha||_{<\lambda>} + ||\beta||_{<\lambda>} \end{aligned} \tag{2.39}$$

Lemma 9 $\forall \alpha \in \mathbb{R}^m$, $\lim_{\lambda\to\infty} ||\alpha||_{<\lambda>} = ||\alpha||_0$

Proof.

$$\begin{aligned} \lim_{\lambda\to\infty} ||\alpha||_{<\lambda>} = & \\ &= \sum_{k=1,\ldots,m} \lim_{\lambda\to\infty} g_\lambda(\alpha_k) \\ &= |\{k \mid \alpha_k \neq 0\}| = ||\alpha||_0 \end{aligned} \tag{2.40}$$

Returning to the original problem:

$$\min ||\alpha||_0 \quad \text{s.t.} \quad \alpha \in v + \mathscr{N}_\Phi \tag{2.41}$$

it can be relaxed as

$$\min ||\alpha||_{<\lambda>} \quad \text{s.t.} \quad \alpha \in v + \mathscr{N}_\Phi \tag{2.42}$$

The justification of such relaxation is given by (9): $\lim_{\lambda\to\infty} ||\alpha||_{<\lambda>} = ||\alpha||_0$. Let $a_1, \ldots, a_r$ with $r \leq m-n$, an orthogonal basis of the null space $\mathscr{N}_\Phi$, where $a_k = (a_{k,1}, \ldots, a_{k,m})$ with $1 \leq k \leq r$. Recalling that the vector v is orthogonal to $a_1, \ldots, a_r$, the null space $\mathscr{N}_\Phi$ can be rewritten as:

$$\mathscr{N}_\Phi = \left\{ \alpha \in \mid \alpha = \sum_{k=1,\ldots,r} x_k a_k, (x_1, \ldots, x_r) \in \mathbb{R}^r \right\}$$

Given the orthogonal basis $a_1, \ldots, a_r$, the projector $P : \mathbb{R}^m \to \mathscr{N}_\Phi$ is defined by:

$$P(y) = \sum_{k=1,r} \langle y, a_k \rangle a_k. \tag{2.43}$$

Given the vector $\alpha = \sum_{k=1,\dots,r} x_k a_k + v$, the problem (2.42) can be recasted in the problem:

$$\min_{x\in\mathbb{R}^r} \sum_{j=1,\dots,m} g_\lambda(\alpha_j) \quad \text{with} \quad \alpha_j = \sum_{k=1,\dots,r} x_k a_{k,j} + v_j \tag{2.44}$$

Note that, fixed an orthogonal basis, the problem (2.44), turns out to be an unconstrained problem on the entire space $\mathbb{R}^r$. The aim of the following results is to show that the fixed point of the operator $L_\lambda(\alpha)$ are local minima of the problem (2.44). A necessary condition of minimum is to nullify the gradient:

$$\frac{\partial}{\partial x_s}\left(\sum_{j=1,\dots,n} g_\lambda(\alpha_j)\right) = 0, \quad \forall (s = 1,\dots,r).$$

It holds that

$$0 = \frac{\partial}{\partial x_s}\left[\sum_{j=1,\dots,n} g_\lambda(\alpha_j)\right] = \sum_{j=1,\dots,n} g'_\lambda(\alpha_j)\frac{\partial}{\partial x_s} = \sum_{j=1,\dots,n} g'_\lambda(\alpha_j) a_{s,j} = (g', a_s),$$

where $g' = (g'_\lambda(\alpha_1),\dots,g'_\lambda(\alpha_n))$. Since $P\alpha \odot e^{-\lambda|\alpha|} = 0$ is the fixed point condition of the operator $G_\lambda(\alpha)$, and since $g' = \lambda^2 \alpha \odot e^{-\lambda|\alpha|}$, we can conclude that the necessary condition of minimum becomes

$$P(g') = \sum_{s=1,\dots,r} \langle g', a_s\rangle a_s = 0. \tag{2.45}$$

Previous result can be stated in following theorem:

Theorem 2.1. *Let α^* a solution of local minimum of*

$$\min ||\alpha||_{<\lambda>} \text{ s.t. } \alpha \in v + \mathscr{N}_\Phi$$

then, it holds that

$$P(\alpha^* \odot e^{-\lambda|\alpha^*|}) = 0$$

It follow that the point α^* is a fixed point of $G_\lambda(\alpha) = \alpha - P(\alpha \odot e^{-\lambda|\alpha|})$ □

2.5.3.1 Fixed Point Stability

Aiming to show both the local minimum property and the stability property of the fixed point, we use the second partial derivative test. Necessary condition of relative minimum of the problem (2.42) is that

$$\frac{\partial}{\partial x_k}\left(\sum_{j=1,\dots,m} g_\lambda(\alpha_j)\right) = 0, \quad \forall k = 1,\dots,r \tag{2.46}$$

or equivalently

$$\left(\sum_{j=1,\dots,m} g'_\lambda(\alpha_j) a_{k,j} \right) = 0, \quad \forall k = 1,\dots,r \tag{2.47}$$

Let P the projector in $\mathscr{N}_{\Phi}$, i.e. $P\alpha = \sum_{k=1,\dots,m} \langle \alpha, a_k \rangle a_k$; the condition 2.47 can be rewritten as

$$Pg'_\lambda(\alpha) = 0.$$

A sufficient condition since α is a relative minimum, is that the Hessian matrix

$$H(\alpha) = \left(\frac{\partial^2}{\partial x_s \partial x_u} \left(\sum_{k=1,\dots,m} g_\lambda(\alpha_k) \right) \right)$$

must be positive defined. With a direct calculus

$$H(\alpha) = \sum_{k=1,\dots,m} g''_\lambda(\alpha_k) a_{s,k} a_{u,k} \tag{2.48}$$

Let us consider the transformation $G_\lambda : \mathscr{A}_{\Phi,s} \to \mathscr{A}_{\Phi,s}$ defined as

$$G_\lambda(\alpha) = \alpha - P(\alpha \odot e^{-\lambda|\alpha|}).$$

We are interested in the study of the iterative system

$$\alpha_{t+1} = G_\lambda(\alpha_t) \tag{2.49}$$

Suppose that α^* is a relative minimum of (2.42) such that the Hessian $H(\alpha^*)$ is positive definite, i.e. has all the eigenvalues greater than zero.
We now show that a fixed point α^* is a asymptotically stable fixed point of the iterative system (2.49).

In this regard, consider the bijection $\phi : \eta + \mathscr{N}_{\Phi} \to \mathbb{R}^m$ given by

$$x_k = \langle \alpha, a_k \rangle, \quad \alpha = \eta + \sum_k x_k a_k.$$

Let $L_\lambda : \mathbb{R}^m \to \mathbb{R}^m$, given by $L_\lambda = \phi G_\lambda \phi^{-1}$, it is convenient to study instead of (2.49) the isomorphic iterative system

$$x(t+1) = L_\lambda(x(t)) \tag{2.50}$$

In terms of components

$$x_k(t+1) = x_k(t) - \frac{1}{\lambda^2} \sum_{j=1,\dots,r} g'_\lambda(\alpha_j) a_{k,j}$$

with $\alpha_j = \eta_j + \sum_{l=1} x_l a_{l,j}$.

Let x^* the fixed point of L_λ, corresponding to α^*, i.e. $x^* = \langle \alpha^*, a_k \rangle$. The linearization $x^* + \Delta(t) \approx x(t)$ of the system (2.50) in the neighborhood of x^* is given by

$$\Delta_k(t+1) = \Delta_k(t) - \sum_{s=1,\dots,r} \left[\frac{\partial}{\partial x_s} \frac{1}{\lambda^2} \sum_{j=1,\dots,m} g'_\lambda(\alpha_j) a_{k,j} \right]_{\alpha=\alpha^*} \Delta_s(t)$$

Since

$$\frac{\partial}{\partial x_s}\Big(\sum_{j=1,\dots,m} g'_\lambda(\alpha_j)\Big) = \sum_{j=1,\dots,m} g''_\lambda(\alpha_j) a_{k,j} a_{s,j} = H_{k,s}(\alpha)$$

results that

$$\Delta(t+1) = \Delta(t) - \frac{1}{\lambda_2} \sum_{s=1,\dots,r} H_{k,s}(\alpha^*)\Delta_s(t)$$

or in vector notation

$$\Delta(t+1) = (I - \frac{1}{\lambda^2} H(\alpha^*))\Delta(t) \tag{2.51}$$

We can conclude that

Lemma 10 *If the eigenvalues ε of $I - \frac{1}{\lambda^2}H(\alpha^*)$, satisfy that $|\varepsilon| < 1$, than α^* is asymptotic stable fixed point of the system (2.49).*

Since the eigenvalues ε of $I - \frac{1}{\lambda^2}H(\alpha^*)$ are related to the eigenvalues of γ of $H(\alpha^*)$ by $\varepsilon = 1 - \frac{\eta}{\lambda^2}$, then we can study the eigenvalues of $H(\alpha^*)$.

Recall that $g'_\lambda(z) = \lambda^2 z e^{-\lambda|z|}$, results that

$$g''_\lambda(x) = \lambda^2(1 - \lambda|z|)e^{-\lambda|z|}. \tag{2.52}$$

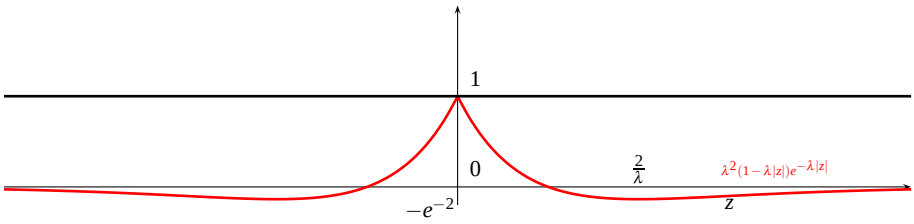

Fig. 2.6 Graph of $\frac{1}{\lambda^2}g''_\lambda(z)$.

The function $\frac{1}{\lambda^2}g''_\lambda(z)$, illustrated in Fig. 2.6, is an even function, with the maximum value in 0, end minimum value in $\pm\frac{2}{\lambda}$, in particular results that $-e^{-2} \leq \frac{1}{\lambda^2}g''_\lambda(z) \leq 1$.

Theorem 2.2. *For all $\alpha \in \nu + \mathscr{N}_\Phi$, let ε be the maximum eigenvalue ε of $\frac{1}{\lambda^2}H(\alpha)$; then $\varepsilon \leq 1$.*

Proof. Recall that $\frac{1}{\lambda^2}H_{s,k}(\alpha) = \sum_{j=1,\dots,m} \rho_j a_{s,j} a_{k,j}$, with $\rho_j = \frac{1}{\lambda^2}g''_\lambda(\alpha_j) \leq 1$. We know that:

$$\varepsilon = \max_{||(v_1,\dots,v_r)||=1} \sum_{i,s} v_i \frac{1}{\lambda^2} Hi,s(\alpha) v_s$$

It result that

$$\begin{aligned}\sum_{i,s} v_i \frac{1}{\lambda^2} H_{i,s}(\alpha) v_s &= \sum_{i,s} v_i \left(\sum_{j=1,\dots,m} \rho_j a_{s,j} a_{i,j} \right) v_s \\ &= \sum_j \rho_j \left(\sum_{i=1} v_i a_{i,j} \right)^2 \\ &\leq \sum_{j=1,\dots,m} \left(\sum_{i=1,\dots,r} v_i a_{i,j} \right)^2 \end{aligned} \tag{2.53}$$

Cansidering an orthonormal basis $a_1, \dots, a_r, a_{r+1}, \dots, a_m$ *of* $\mathbb{R}^m$*, the matrix* $A = (a_{i,j})$ *is orthogonal and then*

$$\begin{aligned}\sum_{j=1,\dots,m} \left(\sum_{i=1,\dots,r} v_i a_{i,j} \right)^2 &= ||(v_1, \dots, v_r, 0, \dots, 0)A||^2 \\ &= ||(v_1, \dots, v_r, 0, \dots, 0)||^2 \quad \textit{for orthogonality of } A \\ &= ||v||^2 = 1 \end{aligned} \tag{2.54}$$

We conclude with

$$\varepsilon = \max_{(v_1,\dots,v_r 0)||^2=1} \sum_{i,s} v_i \frac{1}{\lambda^2} H_{i,s}(\alpha) v_s \leq 1 \tag{2.55}$$

□

We are able to prove the main result of this section:

Theorem 2.3. *Let* α^* *a relative minimum of (2.42), with* $H(\alpha^*)$ *positive defined, then* α^* *is the asymptotically stable fixed point of the iterative system (2.49).*

Proof. Each eigenvalue ε *of* $I - \frac{1}{\lambda^2} H(\alpha^*)$ *is equal to* $\varepsilon = 1 - \gamma$*, where* γ *is an eigevalue of* $\frac{1}{\lambda^2} H(\alpha^*)$*. By theorem 2.2, we know that* $\gamma \leq 1$*; furthermore, since* $H(\alpha^*)$ *is positive defined, it holds that:*

$$0 < \gamma \leq 1$$

Since $\varepsilon = 1 - \gamma$*, we conclude that* $0 \leq \varepsilon < 1$*. Since, each eigenvalue* ε *of* $I - \frac{1}{\lambda^2} H(\alpha^*)$ *verifies that* $|\varepsilon| < 1$*,* α^* *is asymptotically stable fixed point.*

2.5.4 *The* LIMAPS *Algorithm*

Stating the role of the parameter λ in the family of Lipschitzian-type mappings $\mathscr{F}$, we call it *sparsity ratio* because it determines how strong the overall increment of the sparsity level should be within each step of the iterative process. In fact, when applied iteratively, for small λ this kind of mappings should promote sparsity by forcing the magnitude of all components α_i to become more and more close to zero (recall that the map is contractive within $(-1/\lambda, 1/\lambda)$). On the other hand, for high values of λ, the chance to reduce the magnitudes of the α_i diminishes, fixing its value over the time. Hence, for gaining sparsity, the scheduling of sparsity ratio λ should start from small values and then increase according to the iteration step n.

This behavior is exhibited by the algorithm LIMAPS (which stands for LIPSCHITZIAN MAPPINGS FOR SPARSITY) introduced in [2], whose pseudo-code is sketched in Algorithm 3.

Algorithm 3 LIMAPS

Require: - a dictionary $\Phi \in \mathbb{R}^{n\times m}$
- its pseudo-inverse $\Phi^{\dagger}$
- a signal $s \in \mathbb{R}^n$
- a sequence $\{\lambda_t\}_{t\geq 0}$

1: $t \leftarrow 0$
2: $\alpha \leftarrow v$
3: **while** [cond] **do**
4: $\quad \lambda \leftarrow \lambda_t$ <*sparsity ratio update*>
5: $\quad \beta \leftarrow f_\lambda(\alpha)$ <*increase sparsity*>
6: $\quad \alpha \leftarrow \beta - \Phi^{\dagger}(\Phi\beta - s)$ <*orthogonal projection*>
7: $\quad t \leftarrow t+1$ <*step update*>
8: **end while**

Ensure: a fixed-point $\alpha = P\alpha + v$

Remark 1 As said informally above, its ability to find desired solutions is given by wise choices which will be adopted for the sequence $\{\lambda_t\}_{t\geq 0}$, together with choosing a good dictionary. Among many candidates respecting the constraints imposed by (2), one of the most promising sequence, at least on empirical grounds, is the geometric progression whose t-th term has the form

$$\lambda_t = \gamma\lambda_{t-1} = \theta\gamma^t \qquad \text{for } t \geq 1,$$

where $\lambda_0 = \theta$ and $\gamma > 1$ are positive and fixed constants.

Remark 2 In order to have faster computations, the projection operation P must be split into the two matrix products of steps 5 and 6 in pseudocode LIMAPS .

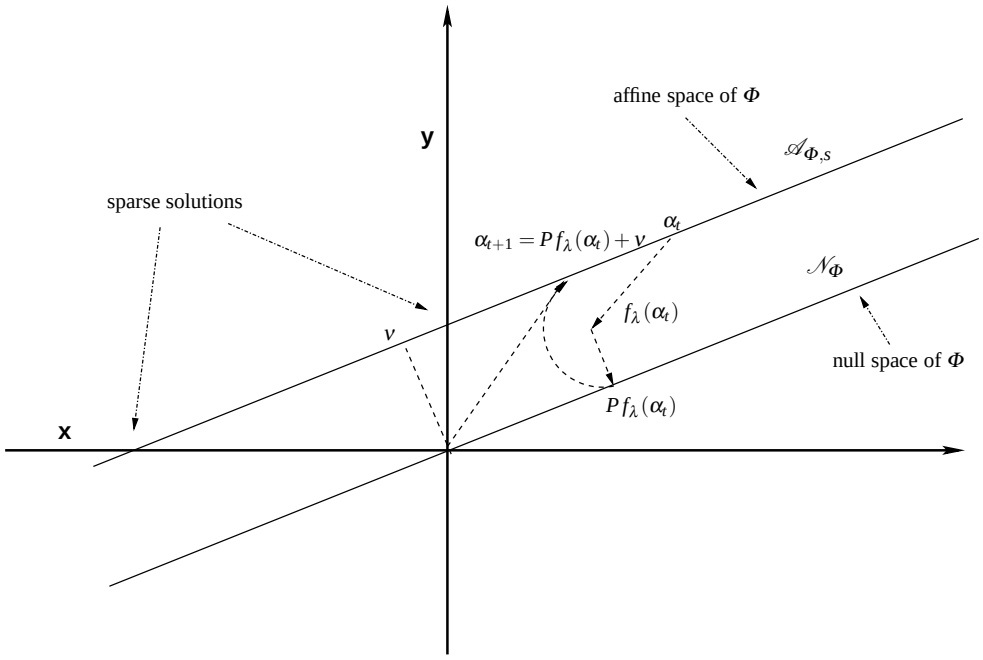

Fig. 2.7 2D example of the LIMAPS iteration scheme. Starting from a point α_t in the affine space $\mathscr{A}_{\Phi,s}$, the point is shrinked applying the f_λ function. After that, the shrinked point is projected into the null space of Φ. As last step, the point is shifted into the affine space by adding the least square solution ν

.

Remark 3 The stop condition of the **while** loop (line 3: of Algorithm 3) may capture different events leading to a correct termination of the iterative system to a solution having minimum error and, hopefully low sparsity.

In realistic computations we found solutions with very small error when the algorithm reaches values near machine precision. Possible choices may include to bound the difference between two successive iterates, that is, until $||\alpha_n - \alpha_{n-1}|| \geq \varepsilon$, or the discrepancy between the value $||P\left[\sum_{k=0}^{\infty} \alpha_k \odot e^{-\lambda_k |\alpha_k|}\right]||$ and zero.

Remark 4 LIMAPS algorithm has a very simple iteration scheme given by step 5 and 6 of the pseudocode. Each iteration consists of two vector additions and two matrix multiplications.

2.6 Sparsity Approximation with Best k-Basis Coefficients

The main drawback of LIMAPS is represented by the need of providing a right sequence for the parameter λ indexing the function family in (2.11) so as to achieve a convergent sequence $\{\alpha_t\}_{t\geq 0}$. Moreover, even if it has good performances on sparse recovery, in general there is no way to upper bound the sparsity level which LIMAPS carries out, unless to accomplish an unnatural thresholding on the final coefficient vector.

In this section we suggest a new fixed point iterative method inspired by the same nonlinear operator on which LIMAPS is based on. In particular, it is shown that the method is able to adaptively find a suitable sequence $\{\lambda_t\}$ for approximately solving the problem (LS0). This choice should also be made in the light of relevant constraints imposed on the objective function of such a problem, that is to choose k coefficients not null and discard the remaining $m-k$.

Fixed $1 \leq k \leq n$, a possible strategy for finding k-sparse solutions using LIMAPS consists on choosing $\lambda_t = \sigma_t^{-1}$ at time $t \geq 0$ satisfying

$$\sigma_t = \hat{\alpha}_{k+1} \tag{2.56}$$

being $\hat{\alpha}$ the absolute values of α rearranged in descending order and $\hat{\alpha}_{k+1}$ its k-th element. The goal of this choice is double:

1. to speed up the process aimed at dropping the smallest coefficients, i.e., those corresponding to elements $\hat{\alpha}_j \leq \sigma_t$, which have indices in the set $\Lambda(t) = \{j : |\alpha_j| \leq \sigma_t\}$;
2. to minimize the solution error induced by α "adjusting" the not null coefficients, i.e., those corresponding to elements $\hat{\alpha}_j > \sigma_t$ which have indices in the set $\Lambda_c(t) = \{j : |\alpha_j| > \sigma_t\}$.

Based upon this strategy, the method should ideally force the σ_t values in such a way to have

$$\lim_{t\to+\infty} g_{\sigma_t}(|\alpha_j|) = 1 - e^{-|\alpha_j|/\sigma_t} = \begin{cases} 1, & \text{if } j \in \Lambda(t) \\ 0, & \text{if } j \in \Lambda_c(t) \end{cases}.$$

Clearly, this requires that $\forall\, j \in \Lambda(t)$ the ratio between the absolute value of the coefficient $|\alpha_j|$ and the parameter σ_t tends to infinite, while $\forall\, j \in \Lambda_c(t)$, $|\alpha_j|$ have to be an infinitesimal of order greater than that of σ_t, leading to

$$\lim_{t\to+\infty} f_{\sigma_t}(\alpha_j) = \alpha_j g_{\sigma_t}(|\alpha_j|) = \begin{cases} \alpha_j, & \text{if } j \in \Lambda(t) \\ 0, & \text{if } j \in \Lambda_c(t) \end{cases}.$$

The overall algorithm, called k-LIMAPS (which stands for k-COEFFICIENTS LIPSCHITZIAN MAPPINGS FOR SPARSITY), is sketched in Algorithm 4.

It should be noted that the last step of the algorithm accomplishes a thresholding of the final point carried out by the while loop because in some cases it can

Algorithm 4 k-LIMAPS

Require: - dictionary $\Phi \in \mathbb{R}^{n\times m}$
- signal $x \in \mathbb{R}^n$
- least square solution $v = \Phi^{\dagger}x$
- sparsity level k

1: $\alpha \leftarrow v$
2: **while** [cond] **do**
3: $\sigma \leftarrow \text{sort}\,(|\alpha|)$
4: $\lambda \leftarrow 1/\sigma_{k+1}$
5: $\alpha \leftarrow \alpha - P\left(\alpha \odot e^{-\lambda|\hat{\alpha}|}\right)$
6: **end while**
7: $\hat{\alpha} \leftarrow P_{\mathscr{C}_k}(\alpha)$

Ensure: An approximate solution $\hat{\alpha} \in \mathscr{C}_k$.

have some noise among the null coefficients, that is those with indices in the set Λ^c. However, experimentally we found that such coefficients reach arbitrary close to zero values as the number of loops increases, making the threshold step not strictly necessary. In Fig. 2.8 we plot the α coefficients showing thus annealing-like behavior which hits the not required coefficients exhibited by k-LIMAPS already at the beginning of the first iterations.

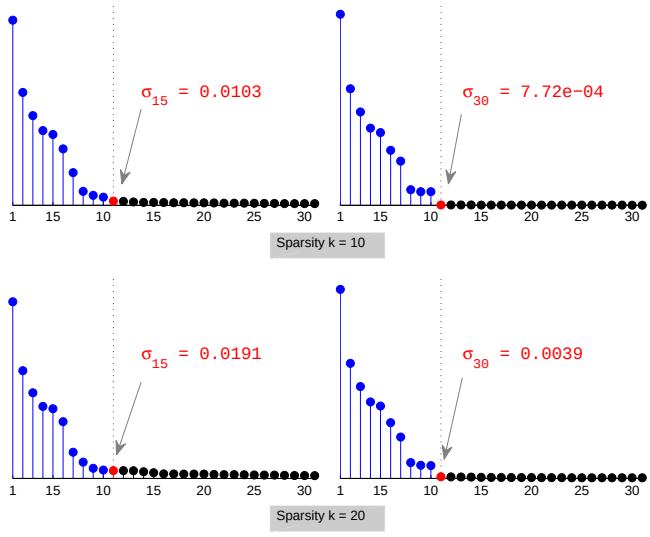

Fig. 2.8 Sorted absolute values of the α coefficients. The red stencils represent the absolute values of σ_t at varius times. They separate the null coefficients (black stencils) from the absolute values of those not null (blue stencils).

The k-LIMAPS algorithm relies on the nonlinear self-mapping

$$\alpha \mapsto \alpha - P\left(\alpha \odot e^{-\lambda|\alpha|}\right) \tag{2.57}$$

over the affine convex set (affine space) $\mathscr{A}_{\Phi,s} = \{\alpha \in \mathbb{R}^m : \Phi\alpha = x\}$, where $\lambda > 0$ is a parameter.

Starting with an initial guess $\alpha \in \mathbb{R}^m$ and applying the mapping (2.57), the sequences $\{\alpha_t\}_{t>0}$ obtained by the iterative step

$$\alpha_{t+1} = \alpha_t - P\left(\alpha_t \odot e^{-\lambda_t|\alpha_t|}\right), \tag{2.58}$$

where $\{\lambda_t\}_{t>0}$ given by (2.56). Points that at the same time minimize the problem (LS0) and are fixed points of (2.57) are those we are looking for. To this end, after a fixed number of iterations, k-LIMAPS uses the nonlinear orthogonal projection $P_{\mathscr{C}_k}$ onto the set $\mathscr{C}_k = \{\beta \in \mathbb{R}^m : \|\beta\|_0 \leq k\}$ expressed by

$$P_{\mathscr{C}_k}(\alpha) = \arg\min_{\beta \in \mathscr{C}_k} \|\alpha - \beta\|^2. \tag{2.59}$$

Note that, due to the nonconvexity of $\mathscr{C}_k$, the solution of problem (2.59) is not unique.

2.6.1 Empirical Convergence for k-LIMAPS

To provide empirical evidence on the convergence ratio, in Fig. 2.9 we plot the curves given by the norm

$$||P\alpha e^{-\lambda|\alpha|}|| \tag{2.60}$$

during the first simulation steps of system (2.58). They are chosen as examples for highlighting how it behaves and how is in general the slope of the curves which result to be decaying in all simulations. Here in particular, k-sparse random instances $s \in \mathbb{R}^n$ and random matrix dictionaries $\Phi \in \mathbb{R}^{n\times m}$ with fixed size $n = 100$ and various $m = 200, \ldots, 1000$. The different slopes are mainly due to the ratio m/n rather than the values imposed to the algorithm by means of the sparsity parameter k. In fact, the curves do not significantly change when we use values for $k > k^*$, where k^* is the optimum sparsity of the given signal s.

2.7 Simulation of LIMAPS Algorithm

To show the effectiveness of LIMAPS algorithm we directly compared it with some algorithms for sparsity recovery well-known in literature, as Matching Pursuit (MP) [77], Orthogonal Matching Pursuit (OMP) [91, 113], Stagewise Orthogonal Matching Pursuit (StOMP) [45], LASSO [48], LARS [48], Smoothed L0 (SL0) [82], Iterative Soft Thresholding [49], Accelerated iterative hard thresholding (AIHT) [17]

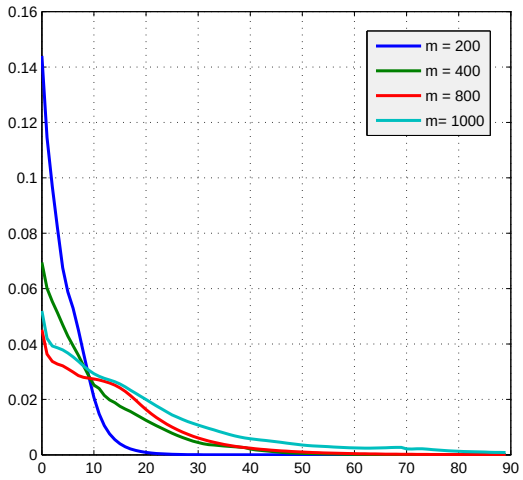

Fig. 2.9 Plotting of the norm in (2.60) with sparsity $k = 10$, size $n = 100$ and $m = 200, 400, 800, 1000$.

and Improved SL0 (ISL0)[66]. In order to make the ISL0 algorithm behavior more stable, in our implementation we used the explicit pseudo inverse calculation instead of the conjugate gradient method, so penalizing its time performances in case of big size instances.

In all tests, the frames Φ and the optimum coefficients α^* are randomly generated using the noiseless Gaussian-Bernoulli stochastic model, i.e., for all $i, j \in [1, \dots, m]$:

$$\Phi_{ij} \sim \mathcal{N}(0, n^{-1}) \quad \text{and} \quad \alpha_i^* \sim x_i \cdot \mathcal{N}(0, \sigma),$$

where $x_i \sim \text{Bern}(p)$. In this way each coefficient α_i^* has probability p to be active and probability $1 - p$ to be inactive. When the coefficient α_i^* is active, its value is randomly drawn with a Gaussian distribution having zero mean and standard deviation σ. Conversely, if the coefficient is not active the value is set to zero. As far as the parameters are concerned, we fix $\lambda_0 = 10^{-3}$ and $\gamma = 1.01$ because they have given good results in all considered instances, coming out essentially independent from the size $n \times m$ of the frames and size m of the coefficient vectors.

We evaluate the performances of the algorithms measuring relative error and computation time:

1. as errors we consider the Signal-to-Noise-Ratio (SNR) and the Sum of Squares Error (SSE) of found approximate solution α with respect the optimum α^*. Precisely:

$$\text{SNR} = 20 \log_{10} \frac{\|\alpha\|}{\|\alpha - \alpha^*\|}, \quad \text{SSE} = \|s - \Phi\hat{\alpha}\|^2;$$

2. as computation time we take the CPU time spent in the execution of the algorithm cores, without including the computation of instances generation or the pseudo-inverse matrix of the dictionary in our and SL0 algorithms.

The simulations were performed on AMD Athlon II X4 630 Processor 64 bit, 2.8 GHz processor with 4 GB of memory, using MATLAB with SparseLab (`http://sparselab.stanford.edu`) and Toolbox Sparse Optimization (`http://www.ceremade.dauphine.fr/~peyre/matlab/`) for algorithm implementation. The algorithm LIMAPS is available online at the URL `http://dalab.dsi.unimi.it/limaps`.

Among the many experiments done, in Figure 2.10, Figure 2.11 and Figure 2.12 we report the average SNR, times and the relative number (in %) of correctly recovered atoms values respectively, obtained from executions on instances of $n = 200$ equations and m ranging from 300 to 1400 variables, moving the percentage of sparsity k from 10% to 50% over n. For each n, m and k 100 instances of dictionary and coefficients were randomly generated.

As can be noted, our algorithm outperforms all the others with regard to the reconstruction quality, reaching arbitrary precision and keeping a CPU execution time comparable with the others. The most interesting results are obtained with the sparsity levels between the 30% and the 50% over n, where our algorithm keeps a good accuracy in terms of SNR.

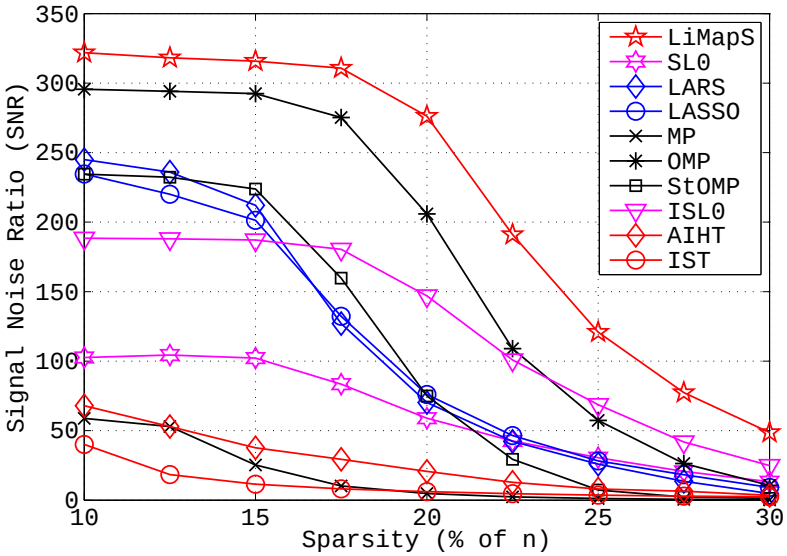

Fig. 2.10 Averages SNR of the algorithms vs. sparsity, expressed in percentage of the number of equations n.

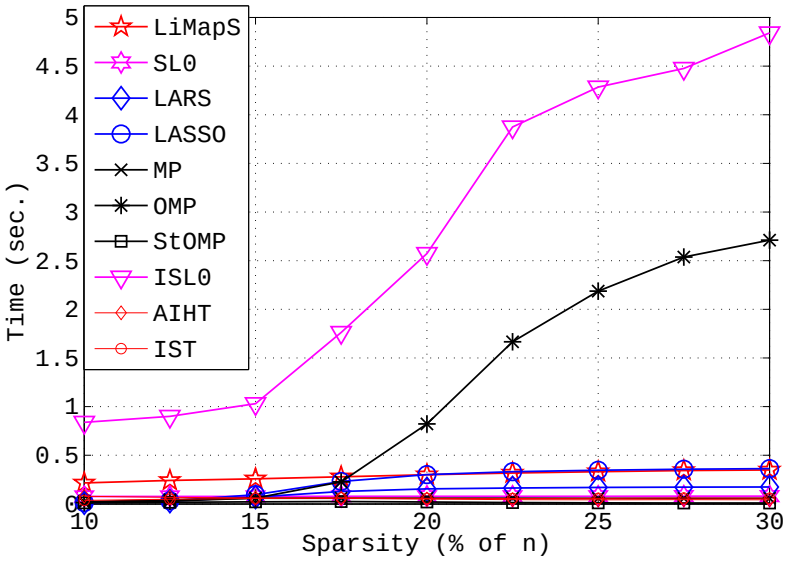

Fig. 2.11 Averages computational times of the algorithms vs. sparsity, expressed in percentage of the number of equations n.

A second kind of experiment was aimed at studing the algorithm behavior when the sparsity level k is low (e.g., 50% over n), that is when algorithms find more difficulties to converge toward the sparsest solution in general. To this end, we have generated random instances of dimensions $n = 400$ and $m = 800$ with a sparsity level $k = 200$, doing also in this case 100 trials. The results are outlined in Table 2.2, listed by error averages $\mu_{\text{SSE}}, \mu_{\text{SNR}}$ and mean times μ_{time} together with their relative standard deviations σ. Again LIMAPS gives the best results in terms of SNR and of SSE with lower standard deviations while the times remain comparable with other algorithms. Finally, it must be noted that the SSE of solutions found by LIMAPS vanishes at each iteration of while cycle (statement 1: in Algorithm 1) since they are remapped every time onto the feasible space of (BP0).

2.8 Simulation of k-LIMAPS Algorithm

In order to empirically study how the k-LIMAPS algorithm performs, we have carried out two kinds of experiments on synthetic and real data respectively. The first was conducted on random instances assumed to have the sparsity property, while the second was aimed to learn a dictionary for a class of electrocardiogram (ECG) signals taken from standard benchmark. Other applications that can benefit from

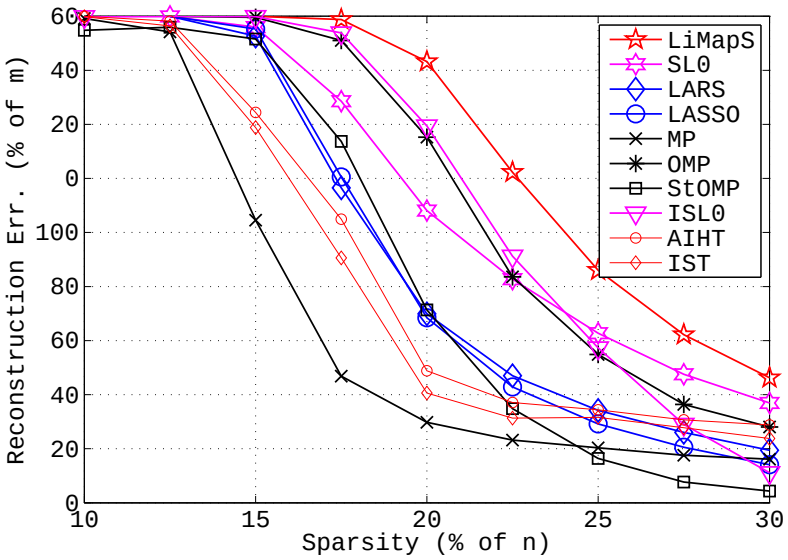

Fig. 2.12 Relative number (in %) of correctly recovered atoms not equal to zero. An atom is considered correctly reconstructed if the deviation from the true value of the estimated value is less than 5%.

Table 2.2 Averages and standard deviations of the results obtained by the algorithms from 1000 trials with instances of dimensions $m = 800$, $n = 400$ and $k = 200$.

	μ_{SSE}	σ_{SSE}	μ_{SNR}	σ_{SNR}	μ_{time}	σ_{time}
LiMapS	**1.5e-24**	1.3e-24	**249.8**	114.9	0.79	0.23
SL0	4.8e-24	7.6e-25	24.7	38.3	**0.15**	0.01
ISL0	4.3e-16	3.7e-15	82.7	89.5	9.10	12.70
IST	6.2e+04	2.2e+05	-0.1	5.0	0.33	0.07
AIHT	5.8e+03	1.6e+03	5.4	1.28	0.06	0.02
LASSO	1.3e+02	1.1e+03	8.3	2.0	1.79	0.20
LARS	2.3e-10	2.3e-09	6.3	2.2	0.79	0.07
MP	2.4e+04	4.3e+03	1.9	0.7	0.18	0.01
OMP	2.4e+00	2.8e-01	1.6	5.5	11.4	0.79
StOMP	3.8e+05	1.4e+05	2.4	0.7	0.02	0.01

the sparsity and overcompleteness concepts include compression, regularization in inverse problems and feature extraction.

By synthetic instances we mean a collection of instances of problem BP0 satisfying sparsity requirements and defined by an ensemble of matrices Φ of size $n \times m$

and an ensemble of k-sparse vectors $s \in \mathbb{R}^n$. All matrices have been sampled from the uniform spherical ensemble, while each vector s was a single realization of a random variable having k nonzeros sampled from a standard iid $\mathscr{N}(0,1)$ distribution.

The OMP and k-LIMAPS algorithms are compared measuring their performances on each realization according to the quantitative criterion given by the mean square error:

$$\text{MSE} = \frac{\| \Phi\alpha - s \|^2}{n}$$

A diagram of the integral of the error depicts the performances of the two algorithms for a wide variety of instances. The average value of such cumulative error measure is displayed as a function of $\rho = k/n$ and $\delta = n/m$. Fig. 2.13 displays a grid of $\delta - \rho$ values, with δ ranging through 50 equispaced points in the interval [.01, .5] and ρ ranging through 100 equispaced points in [.01, 1]; here the signal length is fixed to $n = 100$. Each point on the grid shows the cumulated mean square error between the original and reconstructed, averaged over 100 independent realizations at a given k, m.

It can be noticed that MSE of OMP increases particularly when δ tends to .5 and ρ tends to 1, while k-LIMAPS is less sensitive with respect to these saturation values.

To show the effectiveness of our algorithm on real data, we focus on the dictionary learning task for sparse representation applied to ECG signals. Instances are taken from the Physionet bank [59], specifically in the class of normal sinus rhythm, collecting many patient records with normal cardiac activity. We took a long ECG registration relative to a single patient and we split the signal into segments of length $n = 128$, each one corresponding to a second of the signal registration and sampled with frequency $f_s = n$, then we divide the blocks so obtained into two groups: training set and test set.

To perform the dictionary learning task we use KSVD[5] and MOD[50] techniques working in conjunction with both the pursuit algorithm OMP and our nonlinear method k-LIMAPS as sparsity recovery algorithms. In the training phase, the algorithms perform 50 iteration steps with a fixed sparsity level of 64 coefficients (50% of the signal length), over a dataset collecting 512 samples randomly picked from training set. At the end of the learning phase, the dictionaries carried out by the learning algorithms were tested on 5000 signals picked from the test set using the same sparse recovery algorithms (OMP or k-LIMAPS) previously applied in the training phase.

To evaluate the accuracy of the signal reconstruction, one of the most used performance measure in the ECG signal processing field is the root mean square difference or PRD, together with its normalized version PRDN (which does not depend on the signal mean), defined respectively as:

$$\text{PRD} = 100 * \frac{\|s - \hat{s}\|_2}{\|s\|_2} \quad \text{and} \quad \text{PRDN} = 100 * \frac{\|s - \hat{s}\|_2}{\|s - \overline{s}\|_2},$$

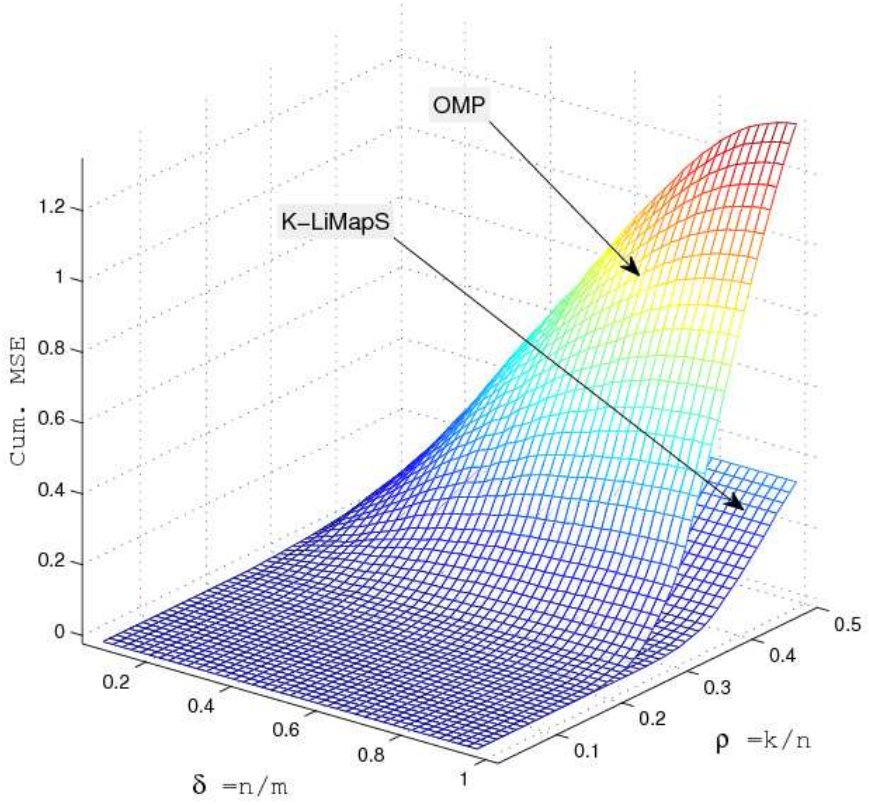

Fig. 2.13 Each point on the grid shows the cumulative MSE between the original s and reconstructed $\Phi\alpha$ signals, averaged over 100 independent realizations. The grid of $\delta - \rho$ values is done with δ ranging through 50 equispaced points in the interval [.01, 5] and ρ ranging through 100 equispaced points in [.01, 1].

where s and $\hat{s}$ are the original and the reconstructed signals respectively, while $\bar{s}$ is the original signal mean.

As it can be observed in Tables 2.3 and 2.4 our sparse recovery algorithm, applied to the dictionary learning, obtains the best results on average for both training algorithms MOD and KSVD, with standard deviations comparable to that of OMP.

The convergence error is a parameter in evaluating such a kind of algorithms. In figure 2.14 we report all MSEs ensured by the algorithms: also in this case k-LIMAPS outperforms OMP with both MOD and KSVD algorithms.

Qualitatively speacking, the signals recovered using dictionaries trained with OMP suffer from a significant error in the more "flat" regions, which are mainly localized nearby the most prominent features of a normal electrocardiogram, given by the three graphical deflections seen on a typical ECG signal and called QRS complex.

Table 2.3 PRD over 5000 test signals.

	PRD mean (%)	PRD std. dev.
KSVD-LiMapS	15.86	5.26
MOD-LiMapS	16.16	5.05
KSVD-OMP	17.92	5.13
MOD-OMP	17.41	4.93

Table 2.4 PRDN over 5000 test signals.

	PRDN mean (%)	PRDN std. dev.
KSVD-LiMapS	16.17	5.26
MOD-LiMapS	15.86	5.05
KSVD-OMP	17.92	5.13
MOD-OMP	17.42	4.92

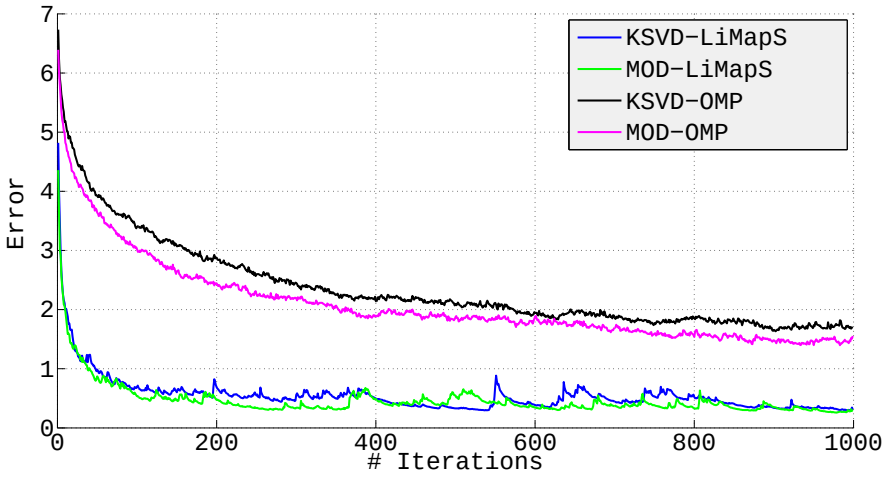

Fig. 2.14 Mean square error over the training set during each iteration of the learning process.

2.8.1 Empirical Phase Transition

Following [16], one of the main aspects of the CS systems is its ability to recover k-sparse signals when the $n \sim k$, as the problem size grows, i.e. $n \to \infty$. Each sparse recovery algorithm exhibits a phase transition property, such that, when no noise is present, it exists a k_n^* such that for any $\varepsilon > 0$, as k_n^*, $n \to \infty$, the algorithm successfully

recovers all k-sparse vectors, provided that $k < (1-\varepsilon)k_n^*$ and does not recover all k-sparse vectors if $k > (1-\varepsilon)k_n^*$.

We assume Φ is a $n \times m$ matrix with $n < m$, drawn from i.i.d. $\mathcal{N}(0, n^{-1})$, the normal distribution with mean 0 and variance n^{-1} and let $\alpha \in \mathbb{R}^m$ a real m dimensional vector with $k < n$ non zero entries.

For the problem (s, Φ) we seek the sparsest vector α such that $s = \Phi\alpha$. When the solution of (P1) is the same as the solution of the problem (BP0), α is called a point of ℓ_1/ℓ_0 equivalence.
Following the convention used by Donoho [44], we denote $\rho = \frac{k}{n}$ and $\delta = \frac{n}{m}$ a normalized measure of problem indeterminacy and a normalized measure of the sparsity respectively, and we define regions $(\delta, \rho) \in [0,1]^2$ that describe the difficulty of a problem instance, in which there is a high probability on the draw of Gaussian matrix Φ that for large problem sizes $(k, n, m) \to \infty$, all $\alpha \in \Sigma_k$ are points of ℓ_1/ℓ_0 equivalence.
A problem can be considered difficult to recover if the sparsity measure and the problem indeterminacy measure are high.

The region where ℓ_1/ℓ_0 equivalences occur for all $\alpha \in \Sigma_k$ is given by (δ, ρ) for

$$\rho \leq (1-\varepsilon)\rho_S(\delta) \tag{2.61}$$

for any $\varepsilon > 0$, where the function $\rho_S(\delta)$ defines a curve below which there is exponentially high probability on the draw of a matrix Φ with Gaussian i.i.d. entries that every k-sparse vector is a point of ℓ_1/ℓ_0 equivalence .

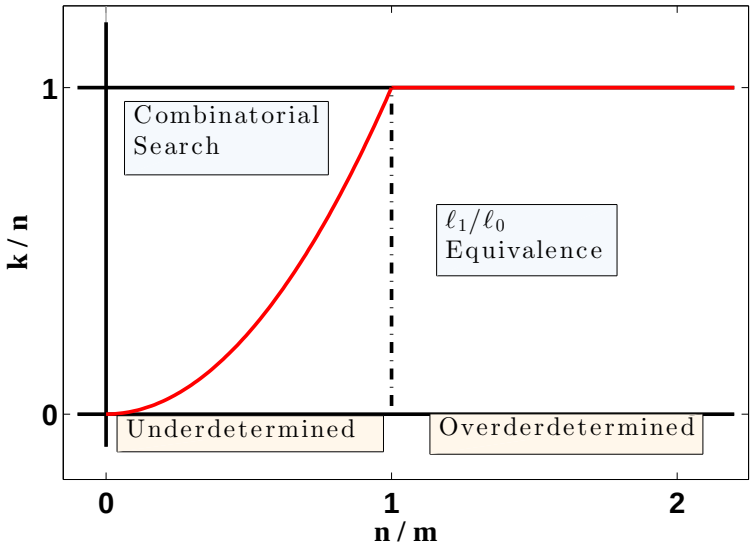

Fig. 2.15 Donoho-Tanner [44] Phase Transition.

Any problem instance with parameters (k,n,m), $\forall \varepsilon > 0$, if $\frac{k}{n} = \rho < (1-\varepsilon)\rho_S(\delta)$, then with high probability in the draw of a matrix Φ with entries drawn i.i.d. from $\mathcal{N}(0,n^{-1})$ every $\alpha \in \Sigma_k$ is a point of ℓ_1/ℓ_0 equivalence.

Rudelson and Vershynin in [100] provided a sufficient condition under which Gaussian matrices will recover all $\alpha \in \Sigma_k$.
The next theorem shows the main result in terms of lower bound on the phase transition $\rho_S^{RV}(\delta)$ for Gaussian matrices.

Theorem 2.4. *For any $\varepsilon > 0$ as $(k,n,m) \to \infty$, there is an exponentially high probability on the draw of Φ with Gaussian i.i.d. entries that every $\alpha \in \Sigma_k$ is a point of ℓ_1/ℓ_0 equivalence if $\rho < (1-\varepsilon)\rho_S^{RV}(\delta)$, where $\rho_S^{RV}(\delta)$ is the solution of*

$$\rho = \frac{1}{12 + 8\log(\frac{1}{\rho\delta})\beta^2(\rho\delta)}$$

with

$$\beta(\rho\delta) = exp\left(\frac{\log(1+2\log(\frac{e}{\rho\delta}))}{4\log(\frac{e}{\rho\delta})}\right)$$

The curve $(\delta, \rho_S^{RV}(\delta))$ is the theoretical curve that separates the successful recoverability area positioned below from unrecoverable instances described by the portion of phase space above. In Fig. 2.15, the Donoho-Tanner [44] phase transition is illustrated. The area under the red curve represents the ℓ_1/ℓ_0 equivalence area.

For a given algorithm, we estimate the phase transition measuring the capability of sparse recovery through extensive experiments. We fix the number of equations of the undetermined system to $n = 100$ and we move the number of variables m and the sparsity level k through a grid of 900 δ and 100 ρ, with δ varying from 0.01 to 1.0 and with ρ varying from 0.01 to 1.0. At each (δ,ρ) combination, we perform 100 problem instances.

Each problem instance is randomly generated using the Gaussian-Bernoulli stochastic model, with each frame entry $\Phi_{i,j} \sim \mathcal{N}(0,n^{-1})$. Each entry belonging to the optimal solution α^* is modeled as

$$\alpha^* \sim x_i.\mathcal{N}(0,\sigma)$$

with $x_i \sim Bern(p)$ distributed as a Bernoulli random variable with parameter p, probability of coefficient's activity. Finally the vector of known terms s of the linear system is calculated by $\Phi\alpha^* = s$.

A problem instance generated as described above is thus a triplet (Φ, s, α^*) consisting of a frame $\Phi \in \mathbb{R}^{n\times m}$ and a k sparse coefficients vector α^*.

To better highlight the reconstruction performances obtained by each algorithm, we chose to plot the phase transition plane in terms of Signal-to-Noise-Ratio (SNR), that compares the level of the desired k sparse vector α^* to the level of noise of the estimated vector α.

In our experiments we estimated the phase transition of four algorithms: OMP, LASSO (LARS), SL0 and our LIMAPS algorithm. For each one of them we performed $9 * 10^6$ sparse reconstructions using a dedicated workstation to reduce the computational time required for the simulations.

In Figures 2.16,2.17,2.18, 2.19, 2.20 we depict empirical phase transitions estimated through instances extracted from the Gaussian-Bernoulli stochastic model described above. The transitions obtained in terms of probability of reconstruction obey the following ordering:

$$\text{LIMAPS} > \text{SL0} > \text{LASSO} > \text{OMP}$$

Otherwise, if we consider the obtained reconstruction quality in terms of Signal-to-Noise-Ration, the SL0 algorithm obtains the worst results, especially in the area between $0.01 \leq \delta \leq 0.8$.

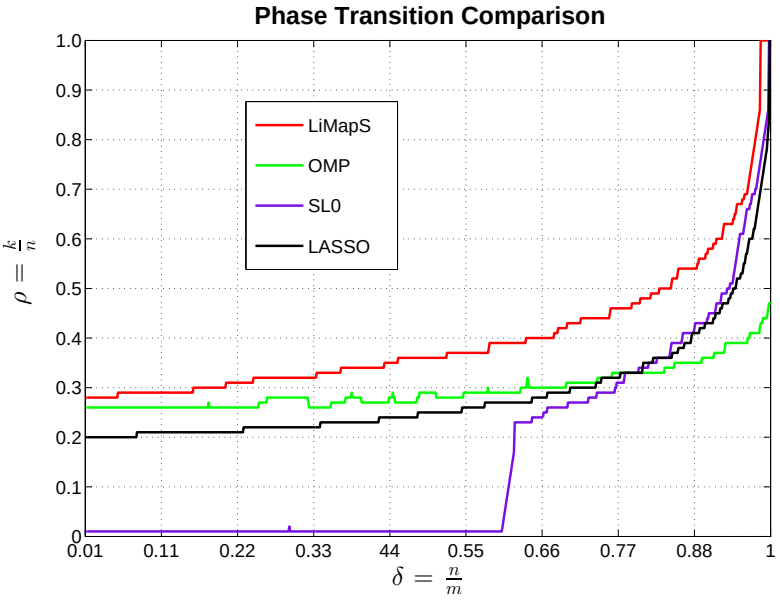

Fig. 2.16 1D Phase Transition of the OMP, LIMAPS , LASSO and SL0 algorithm. In this graph, a signal is considered reconstructed if the SNR of the estimated signal is greater or equal to 100dB.

To show the behavior of the LIMAPS algorithm with different nonlinear sparse promoting mappings, extensive experiments are conducted varying dictionaries. The chosen dictionaries are matrices commonly used or having features easily found in real cases. The dictionaries used in these experiments are described below:

- **Gaussian Dictionary**: A Gaussian random dictionary $\Phi \in \mathbb{R}^{n \times m}$ has each entry $\Phi_{i,j}$ drawn from i.i.d. $\mathcal{N}(0, n^{-1})$ normal distribution with mean 0 and variance n^{-1}.

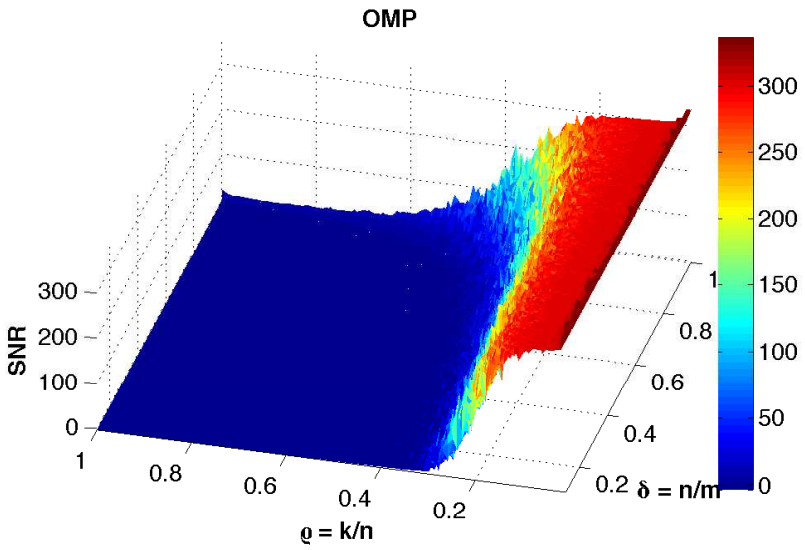

Fig. 2.17 Signal to Noise phase transition of the OMP algorithm.

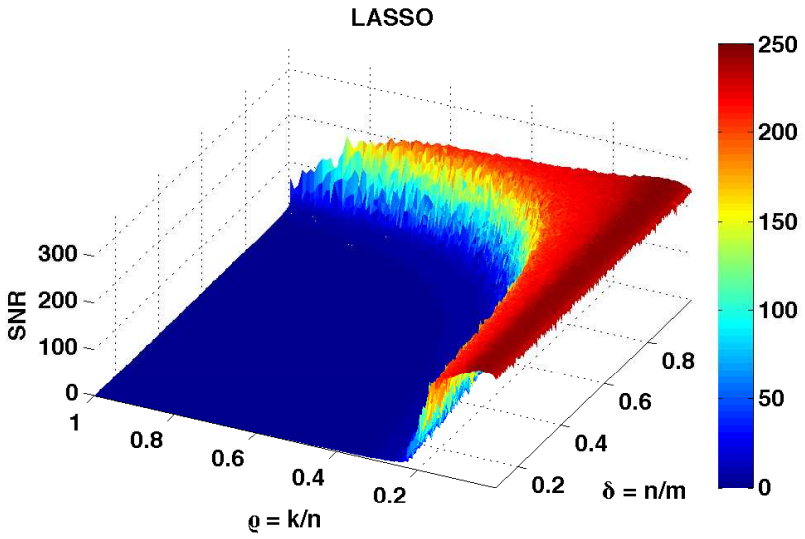

Fig. 2.18 Signal to Noise phase transition of the LASSO (LARS) algorithm.

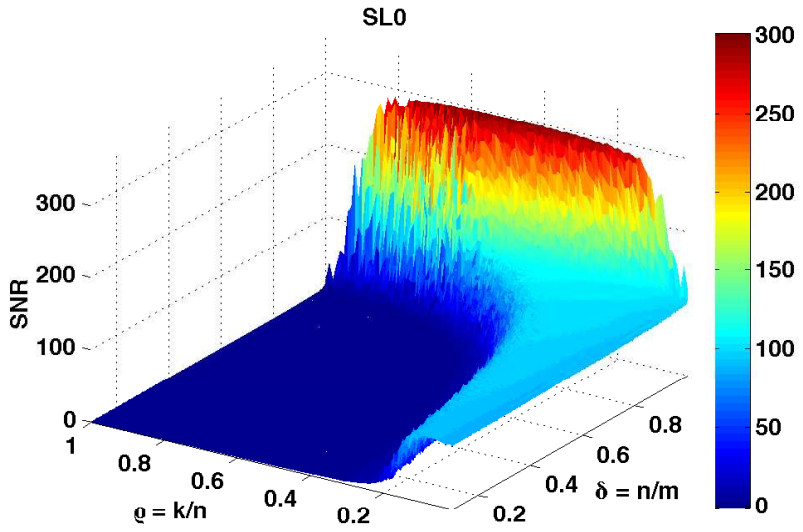

Fig. 2.19 Signal to Noise phase transition of the SL0 algorithm.

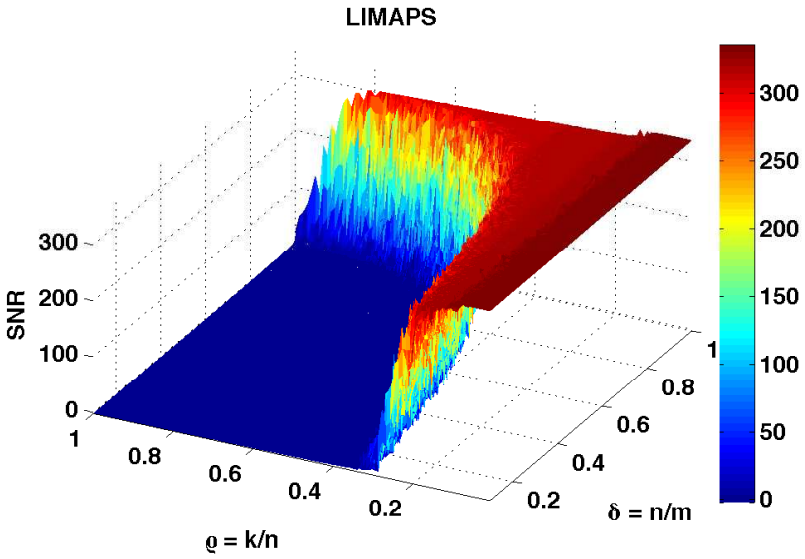

Fig. 2.20 Signal to Noise phase transition of the LIMAPS algorithm.

- **Partial Circulant Dictionary**: Given a vector $\phi = (\phi_1, \ldots, \phi_m)$, the partial circulant matrix $C \in n \times m$ is defined as

$$C = \begin{pmatrix} \phi_1 & \phi_m & \ldots & \phi_2 \\ \phi_2 & \phi_1 & \ldots & \phi_3 \\ \vdots & \vdots & & \vdots \\ \phi_n & \phi_{n-1} & \ldots & \phi_{m-n} \end{pmatrix}$$

  where each entry ϕ_i is drawn from i.i.d. $\mathcal{N}(0, n^{-1})$normal distribution with mean 0 and variance n^{-1}.
- **Partial Toeplitz Dictionary**: Given a vector $\phi = (\phi_1, \ldots, \phi_{m+n})$, the partial Toeplitz matrix $T \in n \times m$ is defined as

$$T = \begin{pmatrix} \phi_1 & \phi_2 & \ldots & \phi_m \\ \phi_2 & \phi_3 & \ldots & \phi_{m+1} \\ \vdots & \vdots & & \vdots \\ \phi_n & \phi_{n+1} & \ldots & \phi_{m+n} \end{pmatrix}$$

  whit each entry ϕ_i randomly extracted from the i.i.d. $\phi_i \sim \mathcal{N}(0, n^{-1})$ random variable.
- **Ternary Dictionary**: Each entry of the Ternary dictionary Φ has a probability p to have value equal to zero. If the entry $\Phi_{i,j}$ is not equal to zero, has probability 0.5 to take value 1 and probability 0.5 to take value -1.
- **Bernoulli Dictionary**: Each entry $\Phi_{i,j}$ of the Bernoulli dictionary Φ has probability $p = 0.5$ to take value 1 and probability $p = 0.5$ to take value -1.
- **Fourier Dictionary**: the entries of the Fourier dictionary $\Phi \in \mathbb{C}^{n \times m}$ are given by

$$\Phi_{j,k} = \frac{1}{\sqrt{m}} \exp^{\frac{2\pi i j k}{m}}$$

- **Sparse dictionary**: Sparse dictionay $\Phi \in \mathbb{R}^{n \times m}$ is a matrix populated primarily with zeros. The entries of the dictionary that differ from zero are randomly choosen from a Gaussian random variable. Formally, each entry $\phi_{i,j}$ has probability p to be active and probability $1 - p$ to be inactive. When $\phi_{i,j}$ is active, its value is randomly drawn with a Gaussian distribution having zero mean and standard deviation $\frac{1}{n}$. Conversely, if the coefficient is not active, the value is set to zero. In our experiments, p is set to 0.5.

For each dictionary and for each sparsity promoting mapping listed in table (2.4.1) we estimate the phase transition measuring the capability of sparse recovery in terms of Signal to Noise (SNR) of the LIMAPS algorithm. We fix the number of equations of the undetermined system to $n = 100$ and we move the number of variables m and the sparsity level k through a grid of 50 δ and 90 ρ, with δ varying from 0.01 to 1.0 and with ρ varying from 0.01 to 1.0. At each (δ, ρ) combination, we performed 100 problem instances. The results are shown in Appendix B of this

thesis. As we can see the LIMAPS algorithm reaches a high reconstruction area in almost all couples sparsity promoting mapping - dictionary.

2.9 Conclusions

We developed new heuristics to solve efficiently the sparse recovery of signals described by underdetermined linear systems. They consist in alternating two lipschitzian maps: one promotes the sparsity of each near-feasible solution (or point) falling outside the affine space associated with the linear transformation and the other remaps such a solution in the nearest point of the feasible space.

The so derived heuristics one based on a iteration schemes which converge to good solution coinciding, in many cases, with the sparsest one solution admitted.

With the experimental results conduced in section 2.7, we highlight the high solution quality and a good average time complexity in practice, comparable with the fastest well-known reconstruction algorithms; in particular, such technique is promising because it exhibits very good performances (high SNR) also in case of very high sparsity (near $n/2$), values for which many others fail.

In particular, we have considered both the case of random generated instances and the case of real data picked to ECG signal database with application to the dictionary learning. We directly compare all accomplished tests with the well-known greedy method called Orthogonal Matching Pursuit and we show that the proposed method outperforms the latter one obtaining less noisy solutions in both kinds of experiment.

We point out that the theoretical analysis of the algorithms is particularly difficult and it deserves further studies. The main open problems are related to the clarity into details the properties of the iterative systems generated by the mappings G_λ. In particular, we proved that the local minima of the $||\alpha||_{<\lambda>}$ over the set $\{\alpha \quad | \quad s = \Phi\alpha\}$ are asymptotically stable fixed points of G_λ. An open problem is to show that the sequence

$$\alpha_{t+1} = G_\lambda(\alpha_t)$$

converges for any initial point α_0.

Part II
Applications

Chapter 3
Face Recognition

Abstract In the first part of this chapter, we present a new holistic approach for face recognition [3] that even with few training samples is robust against both poorly defined and poorly aligned training and testing data. Working in the conventional feature space yielded by the Fisher's Linear Discriminant analysis, it uses the sparse representation algorithm, namely *k*-LIMAPS introduced in chapter 2, as general classification criterion. Thanks to its particular search strategy, it is very fast and able to discriminate among separated classes lying in the low-dimension Fisherspace. In the second part of this chapter, we introduce a local-based FRS namely *k*-LIMAPS_LFR, proposing two possible local features: either raw sub-images or Gabor features. Both these variants combine weak classifiers based on random local information, creating a new robust classifier able to recognize faces in presence of occlusions.

3.1 Introduction

In the last decades the face recognition problem has been widely studied involving biological researchers, psychologists, and computer scientists. This interest is motivated by the numerous applications it involves, such as human-computer interaction (HCI), content-based image retrieval (CBIR), security systems and access control systems [125]. Unfortunately there is still a big disparity between the performances achieved by existing automatic face recognition systems (FRSs) [125, 97] and human ability in solving this task. In particular, the existing methods behave very well under controlled conditions, but their performances drop down significantly when dealing with uncontrolled conditions [125, 112, 97]. The term *uncontrolled conditions* refers to several problems affecting the images, including variations in the environmental conditions (lighting, clutter background), variations in the acquired face (expressions, poses, occlusions), and even the quality of the acquisition (focus/blurred). All these problems have high probability to happen in real applications, thus they need to be faced to have a robust face recognition system (FRS).

Many solutions have been proposed to face each single problem: several illumination invariant FRSs have been presented (e.g. [107, 46, 60, 83]), and also systems dealing with variations of the expression and occlusions (e.g. [84, 7]). However, such systems are specialized on one problem, while in real applications, it is necessary a system able to deal with any possible imperfection, even added together. In this perspective a big effort has been done: the FRSs proposed in [122, 38, 104, 95] deal with uncontrolled images in general, achieving high performances. But again when trying to adopt them in real applications other problems arise. The first concerns the *number of images per subjects* required for training: many FRSs [122, 38] behaves well only if a sufficiently representative training set is available which, however, is not possible in many applications. On the contrary in literature we find works where the training phase requires only one image per subject (facing the so called *Small Sample Size problem*) [70, 96], but then the performances are too poor. Another question concerns the step of face cropping: most approaches [104, 95] present results on face images cropped using manual annotated landmarks, but this is not indicative of the performances we would have applying the methods on automatic detected faces. In fact it has been amply demonstrated that the system performances decrease drastically in presence of misalignment [18]. This problem has been tackled in [124, 118] showing extensive results. Other factors to take into account evaluating a FRS are its scalability, namely, "does the system perform well even with big galleries?", and the computational cost of the algorithm: real-time is often required in applicative contexts.

Existing FRSs can be classified in holistic (H) and local-based (L). The holistic approaches are suitable in case of low quality images considering they do not require to design and extract explicit features. The most popular are Eigenface [115], Fisherface [14] and Laplacianface [64]. More recently a new approach [123] based on the sparse representation theory [41, 25] has been proposed, proving its effectiveness. This method aims at recognizing a test image as a sparse representation of the training set, assuming that each subject can be represented as a linear combination of the corresponding images in the training set. The main disadvantage of this method, and of all the holistic approaches in general, is that it requires a very precise (quasi-perfect) alignment of all the images both in the training and in the test sets: even small errors affect heavily the performances [37]. Besides, they require to have numerous images per subject for training. All these characteristics are not conceivable for real world applications. The local-based methods extract local features either on the whole face [92] or in correspondence to peculiar fiducial points [120]. By construction, such methods are more robust to variations caused by either illumination or pose changes. Moreover they are more suitable to deal with face partial occlusions [90, 79] that may occur in real world applications. Their main disadvantages are the computational cost and the fact that they require a certain image resolution and quality, which cannot be guaranteed in real world applications.

In this chapter we propose both a holistic method and a local approach, highlighting their strengths and weaknesses.

The holistic approach follows [123], while being fast, robust and completely automatic. The crucial peculiarity consists in the adopted sparse approximation algo-

rithm, solving an underdeterminend linear system using the *k*-LiMAPS algorithm [2] presented in chapter 2. Such method is based on suitable Lipschitzian type mappings providing an easy and fast iterative scheme which leads to capture sparsity in the face subspace spanned by the training set. With this change the method achieves higher performances both in presence of unregistered and uncontrolled images.

The local-based approach we propose is also based on the *k*-LiMAPS algorithm, while it extracts local and multiscale information (either raw sub-images or Gabor features) in correspondence to the visible parts of faces. Such setting makes the approach suitable to deal with partial occlusions caused by either accessories (e.g. sunglasses, scarves or hats), or hands or hair on the faces, or even external sources that partially occlude the camera view. The main novelty of our algorithm is that it attempts to solve the face recognition problem with a set of weak classifiers combined by the majority vote rule to create a strong FRS that classifies among multiple linear regression models, being robust to partial occlusions and misalignments.

3.2 Holistic Face Recognition by *k*-LiMAPS Algorithm

In this section we propose a completely automatic and fast FRS based on the sparse representation (SR) method. Both the training and the test sets are preprocessed with the off-the-shelf face detector presented in [116] plus the eyes and mouth locator presented in [19]. The obtained face sub-images are projected in the Fisher space and then sparsity is accomplished applying the recently proposed algorithm *k*-LiMAPS [2]. Such method is based on suitable Lipschitzian type mappings providing an easy and fast iterative scheme which leads to capture sparsity in the face subspace spanned by the training set.

We tested out method on the Yale, Yale B Extended [56], ORL [88], BANCA [11] and FRGC version 2.0 database [93], and compared it with the SRC method. These experiments prove that, despite the system is completely automatic, it is robust with respect to misalignments and variations in expression or illumination.

3.2.1 Eigenfaces and Fisherfaces

Holistic Face Recognition algorithms deal with face images trying to extract global features describing the face in its wholeness. In this section we outline two fundamental techniques used to extract interesting features useful for solving the face recognition problem. These techniques are low sensitive to large variations in lighting intensity, direction and number of light sources and to different facial expressions.

The first is the principal component analysis (PCA) that extracts a set of features called Eigenfaces which maximize the total scatter over the whole training set. The second method exploits the information given by the labels of the training set to

extract features called Fisherfaces which are the most discriminative as possible among the different classes.

3.2.1.1 Eigenfaces

The Eigenface method [114, 69], is based on the principal component analysis (PCA) also called Karhunen–Loéve transformation for dimensionality reduction . It applies a linear projection from the image space to a lower dimensional feature space so that the chosen directions maximize the total scatter across all classes, i.e. across all images of all faces. Choosing the projection which maximizes total scatter, the principal component analysis retains unwanted variations such as for example facial expressions and illuminations.

Let $\{x_1,\ldots,x_N\}$ with $x_i \in \mathbb{R}^n$ be a set of N images taking values in an n-dimensional image space, and assume that each image x_i belongs to one of the C classes $\{1,\ldots,C\}$. Let us consider a linear transformation mapping the original n-dimensional image space into an l-dimensional feature space , with $l < n$. The new feature vectors $y_i \in \mathbb{R}^l$ are defined by:

$$y_i = W^T x_i \text{ with } i = 1,\ldots,N \tag{3.1}$$

where $W \in \mathbb{R}^{n\times l}$ is an orthonormal column matrix.

Let S_T the total scatter matrix defined as

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T$$

where $\mu \in \mathbb{R}^n$ is the mean image of all samples, then after applying the linear transformation W^T, the scatter of the transformed feature vector $\{y_1,\ldots,y_N\}$ is $W^T S_T W$. Principal component analysis choose the projection W_{opt} such that the determinant of the total scatter matrix of the projected samples is maximized

$$W_{opt} = \arg\max \left|W^T S_T W\right|$$

with $W_{opt} = [w_1,\ldots,w_l]$ is the set of n-dimensional eigenvectors of S_T corresponding to the l largest eigenvalues . Considering these eigenvectors have the same dimension of the original images, they are also called Eigenfaces.

If the principal component analysis is presented with images of faces under varying illumination , the projection matrix W_{opt} will contain principal components which retain, in the projected feature space, the variation due to lighting. For this reason, the points in the projected space will not be well clustered according to the subject identity.

The eigenvectors that catch the variance of lighting are the first eigenvectors, thus, discarding them permits a reduction of the variation due to lighting, and consequently a better clustering of the projected samples.

Fig. 3.1 Examples of Eigenfaces of some subjects of the YaleB database.

3.2.1.2 Fisherfaces

The principal component analysis projections are optimal for reconstruction from a low dimensional basis but they may not be optimal from a discrimination standpoint.
Since the learning set is labeled, it makes sense to use this information to build a more reliable method for reducing the dimensionality of the feature space .

Fisher Linear Discriminant analysis (FLD) [54, 14] is a class specific method, in the sense that it tries to reshape the scatter in order to make it more reliable for classification. This method selects the projection matrix W of (3.1) such that the ratio of the between class scatter and the within class scatter is maximized. We define the between class scatter matrix as:

$$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

and let the within class scatter matrix be

$$S_W = \sum_{i=1}^{C} \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

where μ_i is the mean image of the class X_i and N_i the number of samples in the class X_i. If S_W is non singular, the optimal projection W_{opt} is chosen as the matrix with orthonormal columns which maximizes the ratio between the determinants of the between and the within class scatter matrix of the projected samples respectively

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{3.2}$$

The matrix $W_{opt} = [w_1, \ldots, w_l]$ is the matrix that contains the generalized eigenvectors of S_B and S_W corresponding to the l generalized eigenvalues

$$S_B w_i = \lambda_i S_W w_i \text{ with } i = 1, \ldots, l$$

Fig. 3.2 Examples of Fisherfaces of some subjects of the YaleB database.

The maximum number of nonzero generalized eigenvalues is $C-1$, so an upper bound on l is $C-1$, where C is the number of classes.

In the face recognition problem, a possible difficulty that can be appear is that the within class scatter matrix $S_W \in \mathbb{R}^{n\times n}$ can be singular. This stems from the fact that the rank of the matrix S_W is at most $N-C$, and in general the number of images N in the learning set is much smaller then the number n of pixels in each image, making possible to choose the matrix W such that the within class scatter of the projected samples can be made exactly zero.

To avoid this problem in [14] it was proposed an alternative to the criterion (3.2) called Fisherfaces , that projects the image set to a lower dimensional space such that the within class scatter matrix S_W is nonsingular . This can be achieved by using the principal component analysis to reduce the dimensions of the feature space to $N-C$ and then applying the standard Fisher linear discriminant (3.2) to reduce the dimension to $C-1$. The projection matrix W_{opt} can be rewritten as

$$W_{opt}^T = W_{fld}^T W_{pca}^T \tag{3.3}$$

where

$$W_{pca} = \arg\max_W \left|W^T S_T W\right|$$
$$W_{fld} = \arg\max_W \frac{\left|W^T W_{pca}^T S_B W_{pca} W\right|}{\left|W^T W_{pca}^T S_W W_{pca} W\right|}$$

where W_{pca} is the projection matrix of the principal component analysis and W_{fld} is the projection matrix of the Fisher linear discriminant.

3.2.2 Classification Based on Linear Sparse Representation

The classification process aiming at identifying a subject within a fixed group of individuals can be successfully carried out using techniques based on sparse representation [123].

This process can be recasted into the problem of finding the sparsest representation of a test image, usually represented by a vector in a suitable feature space $x \in \mathscr{X} \subseteq \mathbb{R}^n$, into a frame (called *dictionary* in this context) $\Phi = [\phi_1, \ldots, \phi_m]$ assumed to be a wide collection ($m > n$) of vectors (or *atoms*) in $\mathscr{X}$, each one coding a subject belonging to a fixed training set. In this setting, a sparse representation for x means to linearly combine the fewest possible training images, i.e., the smallest number of atoms such that $x = \sum_i \alpha_i \phi_i$, or equivalently in matricial form

$$\Phi\alpha = x. \tag{3.4}$$

Ideally, under condition of underdetermination of the system (3.4) the sparsest solution can be found as a unique solution of the NP-hard optimization problem BP0

In order to make effective the general techniques based on sparsity promotion two main issues must be taken into account:

1. *projection* – it helps both in extracting holistic features to discriminate among subjects and in projecting high-dimensional face images to low-dimensional feature spaces, so making computations faster.
2. *approximation* – the recovery by the exact superposition of few atoms is sometimes unlikely for holistic feature spaces, therefore it is better for the sparse representation to focus on constructing the best approximation of an image with a linear combination of k, or fewer, atoms from the dictionary, as stated by the sparse approximation problem:

$$\min_{\alpha \in \mathbb{R}^m} \| \Phi\alpha - x \| \quad \text{subject to} \quad \| \alpha \|_0 \leq k, \tag{LS0}$$

where $\| \cdot \|$ represents the standard Euclidean norm.

Following [123], the previous defined sparsity framework can be used to determine, given c distinct classes or subjects, at which one a given test image belongs. To this end data are arranged such that the training samples from the i-th class are represented as column vectors of the matrix $A_i = [x_1, \ldots, x_{n_i}] \in \mathbb{R}^{n \times n_i}$. The training set collecting all subjects is then obtained by stacking all matrices A_i into matrix $A = [A_1, \ldots, A_c]$.

As a usual practice, a dimensional reduction is carried out by linearly projecting high-dimensional face images to low-dimensional feature spaces by means of a suitable matrix W. Thus the dictionary Φ for the sparse recovery is determined as $\Phi = WA$. Successively, in the classification stage, given a test image x, the projected sample $y = Wx$ is used to find a sparse vector α such that $\Phi\alpha \approx y$, i.e., to solve one out the many sparsity problems as, for instance, the problem referred by (LS0).

In the purpose of solving the membership i of the test image x, one looks for the linear span of the training samples associated to the subject i that better approximates the feature vector y. In other words, by denoting with $\hat{\alpha}_i$ the coefficient vector whose only nonzero entries are the ones in α associated to class i (zeroing all others entries), the identity of y is found minimizing the residual with the linear combination $\Phi\hat{\alpha}_i$, i.e., by applying the discrepancy rule:

$$\text{identity}(y) = \text{argmin}_{i\in[1,\ldots,c]} \|y - \Phi\hat{\alpha}_i\|. \tag{3.5}$$

3.2.3 *Classification by* k-LIMAPS

The holistic FRS we propose here, namely k-LIMAPS _HFR, follows the setting of the general framework introduced in the previous section, while adopting the k-LIMAPS algorithm to solve the sparsity promotion.

The entire process for face recognition using k-LIMAPS is summarized in the flow diagram of Figure 3.3 and consists in the following steps.

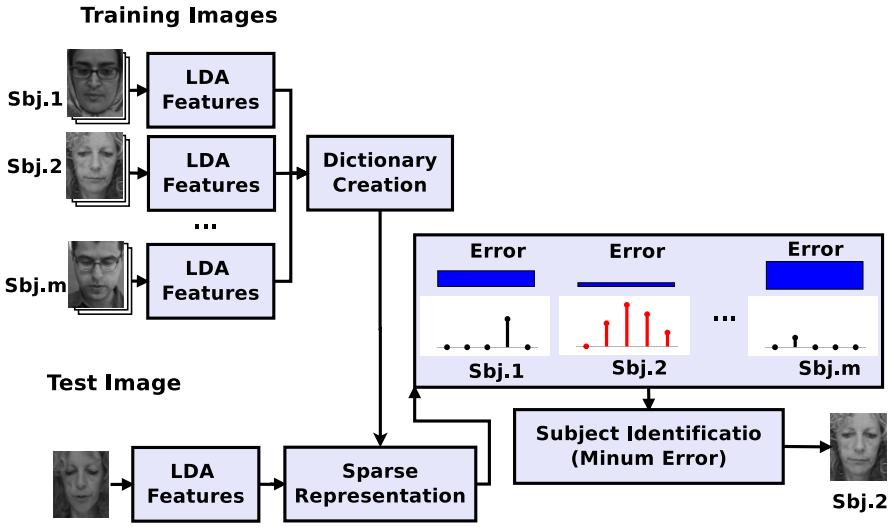

Fig. 3.3 The k-LIMAPS _FR system.

1. *Projection* – which embed training and test n-dimensional images in the LDA space ($(c-1)$-*dimensional vectors*) in order to extracts holistic facial features using the matrix W_{LDA} as projector.
2. *Sparsity* – which finds a sparse representation of a test image x by building the dictionary $\Phi = W_{\text{LDA}}A \in \mathbb{R}^{(c-1)\times n}$ (with $c-1 < n$) and applying k-LIMAPS to

the projected test image $y = W_{\mathrm{LDA}}\, x$ using dictionary Φ, an initial guess $\alpha_0 = Py$, and the sparsity parameter s.
Classify – which finds the identity of x by applying rule (3.5).

3.2.4 Databases

As claimed before, our goal is to define a FRS that works independently of the acquisition conditions. In order to verify this independence, we test our FRS on several databases that differ significantly one from the others with respect to several aspects: the illumination conditions, the photo sharpness, the face pose and expression, the context, and the subject cardinality. In particular in table (3.1) we report the results obtained on the following databases:

1. **Yale db**: contains 165 grayscale images of 15 individuals with homogeneous background. There are 11 images per subject, where the expression (neutral, happy, sad, sleepy, and wink) and the illumination conditions (central/right/left-light) are varied. The subjects are also acquired both with or without glasses.
2. **YaleB + Extended YaleB db** [56]: contains 21888 grayscale images of 38 individuals acquired in 9 poses and 64 different illuminations, all with homogeneous background. For our experiments we considered the images with frontal faces under any illumination conditions (2432 images).
3. **ORL db** [88]: contains 400 grayscale images of 40 distinct subjects in frontal position and with homogeneous background. The images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses).
4. **BANCA db** [11]: it is a large, realistic and challenging multi-modal database. For our experiments we refer to the sections Controlled and Adverse. *Controlled*: it consists of 2080 images of of 52 people placed in front of the camera and looking down as if reading. The images were acquired in 4 different sessions. The images have homogeneous background while the illumination conditions vary from daylight to underexposed. *Adverse*: like the Controlled section it consists of 2080 images. The main difference is that the background is non-uniform and the image quality and illumination are poorer.
5. **FRGC version 2.0 db** [93]: this dataset reports images of 466 people acquired in several sessions (from 1 to 22, varying from person to person), over two periods (Fall 2003 and Spring 2004). A session consists of six images: four *controlled* and two *uncontrolled*, both acquired with either neutral or smiling face expression. Controlled images are acquired in frontal pose, with homogeneous illumination, while the uncontrolled ones represent smaller faces, often blurred and acquired in several illumination conditions. For our experiments we considered only the subjects with at least three sessions per period. This brought us to 384 subjects, in the case of the uncontrolled section, and 394 subjects for the controlled one.

Table 3.1 Database description. From the left to the right the columns are the database name, number of subjects, number of images, background, illumination, expression, timing and the image quality.

Database	N. sbj	N. Imgs.	Back.	Illum.	Expr.	Timing	Qual.
Yale	15	165	hom.	varies	varies	n	good
ORL	40	400	hom.	varies	varies	y	good
BANCA Controlled	52	2080	hom.	good	reading	y	good
BANCA Adverse	52	2080	clutter	poor	reading	y	bad
Extended Yale B (frontal)	38	2432	hom.	varies	neutral	n	good
FRGC v.2 Controlled	394	5726	hom.	good	varies	y	good
FRGC v.2 Uncontrolled	384	5248	clutter	poor	varies	y	bad

3.2.5 Experimental results

In this section we present the experimental results obtained running our system on several public databases.
All the experiments have been carried out on images automatically localized with the face detector proposed in [117] followed by the eyes and mouth locator presented in [19]. *No human intervention* is required. The misalignment we deal with is exemplified in Fig. 3.4.

Fig. 3.4 Examples of automatic cropping on uncontrolled images (first line from the FRGC v.2 db; second line from the BANCA Adverse db).

The number of images in the training set has been deliberately kept low (k varying between 3 and 5) in order to emulate real world settings. The results we report have been obtained mediating over 100 trials; at each iteration, k images are randomly selected for training and the remaining are used to construct the test set. Comparisons have been carried out with the state-of-the-art SRC [123], with a feature space dimension equal to 100, which is a good compromise between the performances and the computational costs.
We first set up several experiments referring to a subset of the FRGC 2.0 dataset. The choice of this database is due to its high subject cardinality and to its rich-

Table 3.2 The face recognition rate (%) on the FRGC 2.0 controlled, varying the cardinality. In brackets we report the number of features which brought to such percentage.

# Subj	50	100	150	200	239
$k=3$	97.6 (100)	96.4 (180)	95.6 (200)	94.9 (340)	93.9 (360)
$k=4$	98.4 (100)	98.3 (200)	97.0 (250)	96.9 (390)	95.4 (490)
$k=5$	98.8 (160)	98.2 (230)	98.2 (280)	97.2 (340)	97.2 (390)

ness in the acquisition conditions (both controlled and uncontrolled), allowing to analyze our FRS under several critical aspects. In particular, we first explored the *system scalability* : considering only the controlled images of people with neutral expressions, we tested the system performances incrementing the subjects cardinality. As shown in Table 3.2, the decrease of performances is more important for small values of *k*.
Second, we investigated how the *expression variation* influences the performances. In the first two columns of Table 3.3 we report the results obtained by both our algorithm and the SRC, varying *k* and the pool of images: either neutral or neutral and smiling of the FRGC 2.0 database. As we can see, the expression variation causes a loss of less than one percentage point for both our method and the SRC, showing a desirable invariance to the expressions.
We explored the system behavior on *uncontrolled* images reporting the results in the last column of Table 3.3. This is the more realistic and challenging scenario, where the subjects are non-collaborative and the acquisition conditions non-optimal. In this case the performances are poorer, reflecting the challenge of the task. The low quality of these images affects the recognition percentage in two ways: first the face locator is less precise, resulting in more misaligned faces (see Fig. 3.4). Second, the feature extractor itself has to deal with less discriminative information deleted by blurring, and even misleading information caused by shadows or glasses. What we highlight however is the large gap between the performance we achieve and the SRC ones. Confirming that our method is more robust in presence of misalignment and unfavorable conditions.

Table 3.3 The face recognition rate (%) on 239 subjects of the FRGC 2.0 controlled, neutral versus neutral and smiling and FRGC 2.0 uncontrolled.

	Neutral		Neutral and Smiling		Uncontrolled	
	k-LiMapS	**SRC**	**k-LiMapS**	**SRC**	**k-LiMapS**	**SRC**
$k=3$	93.9 (360)	92.8	93.2 (380)	91.8	77.1 (390)	68.4
$k=4$	95.4 (490)	95.3	94.6 (500)	94.7	82.8 (360)	74.7
$k=5$	97.2 (390)	96.6	96.3 (460)	96.2	87.2 (380)	79.1

Secondly we investigate the behavior of our algorithm on different databases described in 3.2.4, like Yale, Extended Yale B, Banca Controlled and Adverse,

ORL and the FRGC v.2 Controlled and Uncontrolled. With this experiment we wanted to demonstrate the robustness of our method independently of the database in analysis.

Table 3.4 The face recognition rate (%) on some databases.

Database	**k = 3**		**k = 4**		**k = 5**	
	***k*-LiMaPS_HFR**	**SRC**	***k*-LiMaPS_HFR**	**SRC**	***k*-LiMaPS_HFR**	**SRC**
Yale	95.7 (26)	36.7	97.4 (46)	39.9	98.0 (44)	41.0
ORL	89.6 (52)	65.5	93.0 (66)	83.0	95.3 (54)	89.4
BANCA Co.	90.8 (148)	81.1	94.3 (154)	89.9	96.5 (190)	93.8
BANCA Ad.	86.3 (148)	77.4	90.7 (154)	87.5	93.9 (198)	91.9
Ext. Yale B	89.3 (109)	45.7	94.2 (114)	69.9	96.3 (146)	79.7
FRGC v.2 Co.	90.1 (565)	87.8	94.0 (625)	92.4	96.0 (685)	94.4
FRGC v.2 Un.	72.5 (530)	65.6	79.7 (530)	72.1	84.6 (634)	75.1

The experiments have been carried out keeping low the number of images per subjects in the training set and repeating 100 times each single setting. In table 3.4 we report the average results; regarding the standard deviations, we remark they are always very low (varying between 0.013 and 0.019), indicating a good stability of the system. Results are presented for each database, reporting for each k: the best results obtained varying the number of features n (in brackets we indicate the corresponding value of n) and the results obtained on the same data running the SRC algorithm.
In the following we highlight some aspects crucial for the applicability of the system in real applications: the robustness to possible misalignment produced by the localization step (sec. 3.2.5.1); the low criticality of the parameter setting (sec. 3.2.5.2); the robustness in a wide range of acquisition conditions (sec. 3.2.4); the low computational cost in testing phase (sec. 3.2.5.3).

3.2.5.1 Face localization

In all the experiments the images have been cropped automatically: we applied the face detector proposed in [116], followed by the eyes and mouth locator (EML) presented in [19]. The faces missed by the Viola-Jones face detector are not considered for the subsequent steps, while the misalignments caused by the EML is managed by the FRS.
In order to give a quantitative estimate of the misalignment error, we computed the relative error measure introduced by Jesorsky [67] defined as

$$d_{eye} = \frac{\max(||C_l - \tilde{C}_l||, ||C_r - \tilde{C}_r||)}{||C_l - C_r||}$$

where the values $\tilde{C}_{r/l}$ stand for the eye positions output by the localization module, while the values $C_{r/l}$ are the ground truth of the right and left eye centers respectively. This measure, which normalizes the localization error over the inter-ocular distance, is scale independent. We analogously define the d_{mouth} measure as the mouth detection error normalized over the inter-ocular distance. According to these measures, we can say that on all the datasets the method achieves both d_{eye} and $d_{mouth} \leq 0.25$ in about $99-100\%$ of the processed images, except for the uncontrolled sections of the BANCA and FRGC databases where it achieves the 95%. A more precise localization (d_{eye} and $d_{mouth} \leq 0.10$) is attained in the 85% of controlled images and 60% of the uncontrolled sections. Figure 3.4 shows qualitatively the level of misalignment we deal with.

3.2.5.2 Parameters setting

The algorithm requires to fix three parameters: k, that is the number of images per subject in the training set, n: the feature space dimensionality, and s: the number of atoms selected by k-LIMAPS .

a. Regarding k, it is obvious that the bigger it is the better the performances are. However, in order to emulate real world settings, we keep k low (varying it between 3 and 5). This choice frames our FRS as a solution to the small sample size (SSS) problem.
b. The feature space dimensionality n depends on both the database and the value of K. This is the reason why, whenever possible, it is advisable to tune n for each experiment. Results reported in table 3.4 are based on tuned values of n.
   We remark, however that good performances can be achieved setting n equal to any value between the number of subjects in the training set and the number, ω, of eigenvalues greater than a fixed value (e.g., 10^{-6}): the performance trend in this range changes slowly, while overcoming the superior limit ω they drop down drastically.
c. The k-LIMAPS algorithm requires to set the number s of atoms to select from the dictionary. This is not a critical parameter (varying it between 2 and 10 no significant differences are encountered). In the experiments we set $s = k$: so doing in the best case k-LIMAPS will select all and only the k atoms in the dictionary corresponding to the current target subject.

3.2.5.3 Computational costs

A last but not least aspect to take in consideration when designing and implementing a FRS is its computational cost: traditional approaches in the field of the compressive sensing require to solve linear systems, adopting the expensive simplex method. If on one hand this allows to achieve high performances,

on the other hand, it is computational very expensive preventing by itself the applicability in real applications.
The *k*-LIMAPS _HFR is very fast being based on an iterative search strategy, that in the case of face recognition arrives to convergence after few iterations (namely 5-10).
In particular, the MATLAB implementation of our algorithm runs each image in 0.01 seconds when referring to the most populous dictionary (FRGC controlled), resulting in a real-time solution for the FR problem.

3.3 Face Recognition with Occlusions by *k*-LIMAPS

Occlusions and variation of expressions are the most common difficulties in applications involving automatic FRSs. Sources of occlusions can be apparel like sunglasses, eyeglasses, scarves and hats as well as hands or hair covering part of a face or even other objects placed between the camera and the face. Other kind of occlusions can be considered extreme variation of illumination like dark shadows or excessive lighting.
Robustness to occlusions is essential for real world FRSs. If faces are partially occluded, holistic methods based on Eigenfaces of Fisherfaces [14] cannot be applied, since all the features extracted from the training set would be corrupted. In this work we propose a local-based FRS namely *k*-LIMAPS _LFR, combining a set of weak local classifiers obtaining a strong and robust classifier, suitable to solve the face recognition problem under occlusions. This algorithm exploits two possible local features: the first, described in 3.3.2, called Multi-scale Random Tessellations (MRT), consist of natural patches extracted from face images; the second, are Gabor features (Gf) that well describe contours and local variations in spatial domain.
The *k*-LIMAPS _LFR system consists of two phases: the dictionary construction and the test phase. Regarding the first one, we proceed selecting k unoccluded images per subject and extracting from each of them a high number Z, of local features (the same pool of features for all the images). Z dictionaries are then constructed, one for each feature z, linearizing the corresponding information extracted from each training images and placing them side by side in the matrix Φ_z.
The testing phase is articulated in the following steps. At first the test image is classified according to the possible occlusion (unoccluded / glasses / scarf) performed through the use of the EML locator. In the case the EML locator correctly localize eyes, the subject must be checked by the scarf detector described in [80] in order to determine the scarf possible presence. Otherwise, if the EML fails to locate eyes, the face are classified as occluded by sunglasses. On the basis of this information, we inhibit the occluded regions referring to off-line defined masks. In particular two binary masks, one corresponding to sunglasses and the other to scarves. The masks are mathematically described as $M(x,y)$:

$$M(x,y) = \begin{cases} 0, & \text{if the pixel (x,y) is not occluded} \\ 1, & \text{otherwise} \end{cases}$$

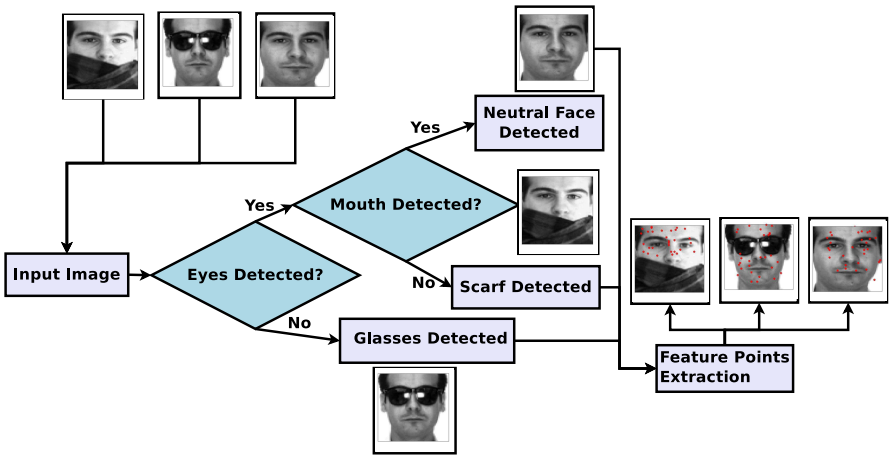

Fig. 3.5 Classification of occlusions and feature points extraction.

M is used to define a feature property. Let z be a feature in Z and let (x_z, y_z) be the center of the feature support, h and w be its height and width respectively. Then, given $0 < \sigma < 1$, we say that z is σ-unoccluded if

$$\frac{\sum_{i=\lfloor x_p - \frac{w}{2} \rfloor}^{\lceil x_p + \frac{w}{2} \rceil} \sum_{i=\lfloor y_p - \frac{h}{2} \rfloor}^{\lceil y_p + \frac{h}{2} \rceil} M(i,j)}{wh} \leq \sigma$$

Applying this inspection, a subset S of dictionaries is selected for classification so that

$$\Phi_z \in S \text{ if } z \text{ is } \sigma\text{-unoccluded.}$$

Each $\Phi_z \in S$ is a weak classifier behaving as the one presented in section 3.2.2. The final decision is determined applying the majority vote rule as described in the next subsection. Finally, at the end of this section we detail two possible features to be casted in the *k*-LIMAPS _LFR system.

3.3.1 Combining Classifiers

Let $D = \{D_1, \ldots, D_L\}$ be a set of classifiers, called also pool or ensemble, such that $D_i : \mathbb{R}^n \to \Omega$, where $\Omega = \{\omega_1, \ldots, \omega_c\}$, assigns to each $x \in \mathbb{R}^n$ a class label $\omega_j \in \Omega$. The majority vote method for combining classifiers decisions, is one

of many methods that assign to x the class label ω_j that is supported by the majority of the classifiers D_i.
In general given the votes of L classifiers $D_1,\ldots,D_L$, we define a majority vote classifier as follows

$$C(x) = \arg\max_i \sum_{j=1}^{L} w_j I(D_j(x) = i) \tag{3.6}$$

where $w_1,\ldots,w_L$ are weights such that $\sum_j w_j = 1$, and $I(.)$ is an indicator function. If weights are $w_j = \frac{1}{L}$, (3.6) can be rewritten as

$$C(x) = \text{mode}\{D_1(x),\ldots,D_L(x)\} \tag{3.7}$$

where mode$(.)$ is the value that appears most often in the set of classifiers.
Finding independent classifiers is one aim of classifier fusion methods for the following reason. Let L be odd, $\Omega = \{\omega_1,\omega_2\}$, and all classifiers have the same classification accuracy $p \in [0,1]$. The majority vote method with independent classifiers decisions gives an overall correct classification accuracy calculated by the binomial formula

$$p_{maj} = \sum_{i=0}^{\lfloor L/2 \rfloor} \binom{L}{i} p^{L-i}(1-p)^i$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. The majority vote method with independent classifiers is guaranteed to give a higher accuracy than individual classifiers when $p > 0.5$ [72, 71].

3.3.2 Local Features

The first pool of features we are interested in investigating consists of patches extracted varying randomly the position, the height and the width. These raw data capture local details, varying the scale and thus the retailed details.
In figure 3.6 we can see examples of random patches maintained in presence of different kind of occlusions. We observe that both the screaming and neutral expression images are unoccluded, thus their occlusion masks are set to 0 for all pixels.
The second pool of features we investigate is obtained applying a bank of Gabor filters in correspondence to a certain number of points randomly extracted from the non occluded face portion.
A Gabor filter is a linear band pass filter, widely used for edge detection, obtained by modulating an harmonic sinusoidal function with a Gaussian function. The filter has a real and an imaginary component representing orthogonal directions.

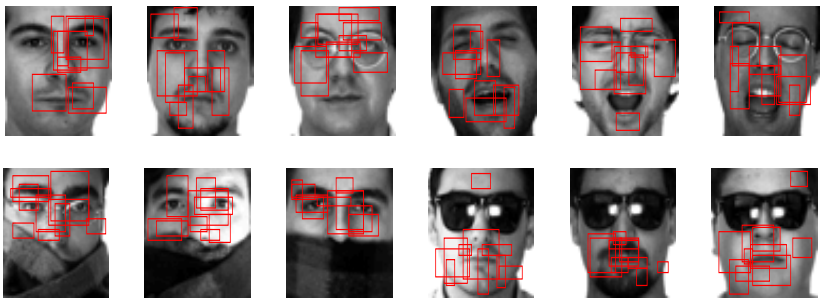

Fig. 3.6 Example of Random Pieces for different occlusions.

The impulse response of a Gabor filter is defined as

$$g_{real}(x,y;\lambda,\theta,\phi,\sigma,\gamma) = e^{-\frac{x'^2+y'^2\gamma^2}{2\sigma^2}}\cos(2\pi\frac{x'}{\lambda}+\phi)$$
$$g_{imag}(x,y;\lambda,\theta,\phi,\sigma,\gamma) = e^{-\frac{x'^2+y'^2\gamma^2}{2\sigma^2}}\sin(2\pi\frac{x'}{\lambda}+\phi)$$

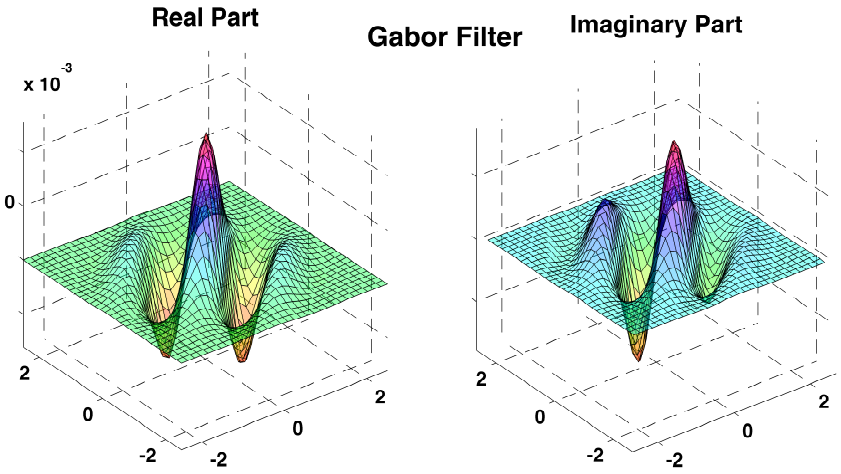

Fig. 3.7 Real (cosine-type) and imaginary (sine-type) part of the impulse response of a Gabor filter.

where $x' = x\cos(\theta)+y\sin(\theta)$ and $y' = -x\sin(\theta)+y\cos(\theta)$.
The parameter λ is the wavelength of the sinusoidal function, θ is the orientation of the Gabor function , σ represents the sigma of the Gaussian envelope, γ

specifies the ellipticity of the support of the Gabor function and ϕ is the phase offset.
The real and the imaginary components of the Gabor filter may be used individually or combined to form a complex number as defined below:

$$g(x,y;\lambda,\theta,\phi,\sigma,\gamma) = e^{-\frac{x'^2+y'^2\gamma^2}{2\sigma^2}} e^{i(2\pi\frac{x'}{\lambda}+\phi)}$$

The response of a Gabor filter to an image is obtained by a 2D convolution operation. Let $I(x,y)$ denotes the image and let $G(x,y;\lambda,\theta,\phi,\sigma,\gamma)$ be the response of a Gabor filter $g(x,y;\lambda,\theta,\phi,\sigma,\gamma)$ at point (x,y) on the image plane. The response $G(.)$ of the Gabor filter is obtained as

$$\begin{aligned} G(x,y;\lambda,\theta,\phi,\sigma,\gamma) &= I(u,v) * g(x,y;\lambda,\theta,\phi,\sigma,\gamma) \\ &= \int\int_{-\infty}^{+\infty} I(u,v) g(x-u,y-v;\lambda,\theta,\phi,\sigma,\gamma) du dv \end{aligned}$$

In figure 3.8 we show the amplitude of response of Gabor filters with different orientations and scales applied to a face image.
Wiskott at al. in [121] suggest a set of Gabor filters with 5 spatial frequencies and 8 distinct orientations to create an efficient filter bank.
Parameters are resumed in table 3.5

Table 3.5 Parameters of Gabor filters.

Parameter	Symbol	Values
Orientation	θ	$\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\}$
Wavelength	λ	$\{4, 4\sqrt{2}], 8, 8\sqrt{2}, 16\}$
Phase	ϕ	$\{0, \frac{\pi}{2}\}$
Gaussian Radius	σ	$\{4, 4\sqrt{2}], 8, 8\sqrt{2}, 16\}$
Aspect Radius	γ	1

Gabor filters are among the most popular tools for facial feature extraction. Their use in automatic face recognition system is motivated by two major factors: their computational properties and their biological relevance.
When exploited for feature extraction, a filter bank with several filters is usually created and used to extract multiorientational and multi-scale features from the given face image.
By convolving face images with the Gabor kernels of the filter bank previously defined, we would have a total of 40 Gabor filter response images. These images could be used as features, but their high dimensionality would make this approach impractical.
To reduce the feature space, our approach consists in generating random points spatially located in non occluded area of the image, and in calculating local features through the application of the Gabor filter bank.

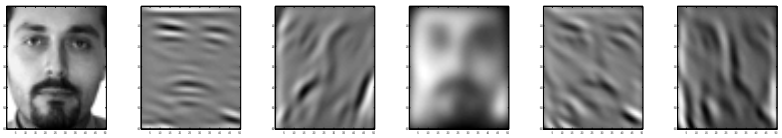

Fig. 3.8 Example of features extracted convolving a subject with Gabor filters with different orientations and scales.

3.3.3 Occlusion and Expression Database

Our goal is to test the *k*-LIMAPS _LFRS system in case of partial occlusions or expression variations. To this end we adopted the AR [78] database that contains images of faces acquired with different occlusions and with different expressions.
This face database was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B. It contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women). Images show frontal view faces with different facial expressions, illumination conditions, and occlusions (sun-glasses and scarves). The pictures were taken at the CVC under strictly controlled conditions. No restriction on clothing, make-up, hair style, etc. were imposed to participants. Each person participated in two sessions, separated by two weeks (14 days) time.
The same pictures were taken in both sessions.
All images are stored in raw file format of the form G-xx-yy.raw. The letter 'G' represents the gender of the subject and can assume the value "M" for males and "F" for females, 'xx' is a unique person identifier (from "00" to "70" for males and from "00" to "56" for females) and 'yy' specifies the features of each image; its meanings are described in the following table:

1 Neutral expression
2 Smile
3 Anger
4 Scream
5 left light on
6 right light on
7 all side lights on
8 wearing sun glasses
9 wearing sun glasses and left light on
10 wearing sun glasses and right light on
11 wearing scarf
12 wearing scarf and left light on
13 wearing scarf and right light on
14 to 26 second session (same conditions as 1 to 13)

3.3.4 Experimental results

In this section we present the experimental results obtained on the AR database running our local-FRS using both Multiscale Random Tessellation and Gabor features
.

Table 3.6 Face recognition rate (%) for AR registered with 100 subjects, 100 trials.

	Type	Train	Test	k-LIMAPS_LFR MRT 200 pieces	k-LIMAPS_LFR Gf 300 points	SRC	PFI
1	Glasses	$[1:4,14:17]$	$[8,21]$	98.15	94.50	97.50	97.50
2	Scarf	$[1:4,14:17]$	$[11,24]$	98.27	97.30	93.00	93.50
3	Glasses	$[1,5:7,14,18:20]$	$[8:10,21:23]$	95.01	95.00	96.00	97.70
4	Scarf	$[1,5:7,14,18:20]$	$[11:13,24:26]$	96.45	97.17	91.50	82.50
5	Glasses	$[1,2,5:7]$	$[8:10,21:23]$	87.80	91.30	79.30	84.50
6	Scarf	$[1,2,5:7]$	$[11:13,24:26]$	92.05	95.53	82.30	73.20
7	Neutral	$[1,2,5:7]$	$[14,15,18:20]$	98.94	99.76	96.20	98.80
8	Expression	$[1:4,7]$	$[15:17]$	95.77	96.47	91.00	97.00

In all the experiments the images are automatically located with the face detector proposed in [117] and automatically registered by the eyes and mouth locator (EML) presented in [19]. In case of sunglasses, the EML reports the eyes absence, thus we refer to the face localized by the face detector. In case of scarves, the eyes are generally correctly localized, allowing a partial registration.

The results we report have been obtained mediating over 100 experiments, varying randomly the patches for the Multiscale Random Tessellation (MRT) and the point locations for the Gabor features (Gf).

We setup two sessions of experiments. In the first we adopt 8 images per subject for the training set ($k = 8$), according to the experiments reported in literature [123]. In the second we reduce k to 5 in order to test the system behavior in more realistic conditions. In all cases the training images are non occluded. Comparisons have been carried out with the state-of-the-art SRC [123] and with the algorithm proposed in [103, 105] that use a large features sets extracted locally in each image.

In the first session, the training sets are constructed as proposed in [123]. In particular the authors propose two scenarios: in the first (adopted for the experiments 1 and 2 in the table (3.6)) the training sets represents each subject under different expressions, while the illumination condition is constant. In the second (adopted for the experiments 3 and 4 in the table (3.6)), the expression is always neutral while the illumination conditions vary over the training images. In the second session all illumination and neutral and smiling expressions are used for training. Only in the

case of test over the expressions, we enrich the training with all the expressions, reducing the illumination variability.

Experiments are run both on manually registered images and on automatic registered images. Results are respectively summarized in in table (3.6) and in table (3.7). We observe that in all the experiments we obtained a very low standard deviation, varying between 0.022 and 0.037 indicating good stability results.

Table 3.7 Face recognition rate (%) for AR unregistered with 72 subjects, 100 trials.

	Type	**Train**	**Test**	***k*-LIMAPS** MRT 200 pieces	***k*-LIMAPS** GC 300 points	**SRC**	**PFI**
1	Glasses	$[1:4,14:17]$	$[8,21]$	84.40	82.74	52.56	83.80
2	Scarf	$[1:4,14:17]$	$[11,24]$	95.51	96.50	88.46	70.90
3	Glasses	$[1,5:7,14,18:20]$	$[8:10,21:23]$	78.21	81.60	56.55	74.50
4	Scarf	$[1,5:7,14,18:20]$	$[11:13,24:26]$	92.86	95.95	84.33	71.40
5	Glasses	$[1,2,5:7]$	$[8:10,21:23]$	65.60	66.84	44.02	62.80
6	Scarf	$[1,2,5:7]$	$[11:13,24:26]$	87.18	91.42	75.50	64.20
7	Neutral	$[1,2,5:7]$	$[14,15,18:20]$	91.64	94.02	83.59	92.50
8	Expression	$[1:4,7]$	$[15:17]$	80.80	85.76	74.64	91.20

As we can see in table 3.6, our local-FRS using 300 Gabor features results the best classifier in almost all cases, except for the first experiment that use training images with controlled frontal illumination but with variations in expressions. In this case the best results are obtained adopting the MRT features. This behavior of the Gabor features could be explained by the fact that the Gabor filter bank is very sensitive to edges caused by expression variations, corrupting the training information.

In the second most realistic scenario, where the images are automatically registered, the worst performances are obtained in the case of glasses occlusion. This result confirms the difficulty already shown in table 3.6 of recognizing a person viewing only the lower half of the face. Here the performance drop down even more because of the higher misalignment: in case of occlusion only the face detector is adopted to locate the face.

As general consideration we can remark that the local-based FRS does not require specific tuning, except the number of features to generate randomly. Obviously, a small feature cardinality Z would compromise the system performances, but at a certain level we observed that further increasing of Z would be useless for the performance improvement, while augmenting significantly the computational costs. Experimentally we set $Z = 300$ as a trade-off between performances and computational costs. In any case, considering the weak classifiers are independent, it may

always be possible to evaluate in parallel the classification stage and merging results by majority vote rule.

3.4 Conclusion

In the first part of this chapter we have proposed a new approach for face recognition in the framework of sparse recovery. Experimentally, we have shown its capability in producing well separated classes in uncontrolled contexts, proving its applicability in real contexts.

We remark that the system has been applied on automatically localized faces, showing its ability in dealing with misalignment. Its applicability in real contexts is reinforced by both the good performances achieved having few examples per subject for training, and its low computational costs.

In the second part we have illustrated a new local-FRS that combines weak classifiers to obtain a robust one able to work in difficult conditions such as partial occlusions, with different kind of environmental luminosity conditions and with different facial expressions.

Experimental results show that the algorithm has high recognition rate, showing good stability performances both in case of manually registered images and in case of automatic registration.

The promising results encourage us to research methods to reduce the number of classifier necessary to obtain a stable FRSs, investigating new ensemble techniques and different kind of features to improve the classification capacity under occlusions and under variation of expressions.

Chapter 4
ECG Signal Compression

Abstract In this chapter we present a novel and efficient signal compression algorithm aimed at finding the sparsest representation of electrocardiogram (ECG) signals. In order to achieve high compression rate (CR), the method generates, for each signal, its own base using the first seconds of the signal itself. The algorithm requires the user to fix a desired percent root square difference (PRD). After a preprocessing phase, where some kind of noise is suppressed, the ECG signal is windowed and sparse represented by the k-LIMAPS algorithm with a number of coefficients adaptively estimated from the data. The found coefficients are then discretized and rescaled in a convenient range and compressed with a lossless entropy-based compression algorithm. To evaluate the proposed method, the technique is tested over a large number of both normal and abnormal ECG signals extracted from the MIT-BIH Arrhythmia database. The performances are measured in terms of percent root-mean square difference (PRD), normalized percent root mean square difference (PRDN), compression ratio (CR) and signal to noise ratio (SNR). Our algorithm shows best results if compared with other methods proposed in literature, reaching comparable compression ratios with lower root mean square difference error.

4.1 Introduction

In the last few years, the need of ECG signal recordings has been enormously augmented due to the increasing interest in health care. Portable ECG recording systems (e.g. holters) record ECG signals continuously for long time periods ranging between several hours and a few days. One of the most important problem of holter systems is the huge space required to store long time records; for example a one channel ECG signal sampled at a frequency of 512Hz with 11 bits of quantization resolution, recorded for a day (24 hours) require an amount of 58MB of storage size.

In recent years, many ECG compression algorithms have been developed to encode digital ECG signals. They can be classified into two major categories: lossless and lossy algorithms.

The lossless algorithms such as the Run-Length Encoding [74], Huffman [65] and Lempel-Ziv-Welch [119] do not show reconstruction error while the compression ratio (CR) is generally small, conversely the lossy algorithms like Foveation-SPIHT[32] and DCeq-MMP [51] pay a quantization error to obtain higher compression level.

The most classical compression algorithms apply some data transformations to extract the most informative characteristics exploiting the redundancy of data. The most informative coefficients are then encoded using different lossless compression algorithms.

Within this group many methods based on the Discrete Wavelet Transform (DWT) [4], like the Set Partitioning in Hierarchical Tree (SPIHT) algorithm [73] show good results in ECG signal coding.

In recent years, compressive sensing and sparse recovery theory has generated significant interest in the signal processing community because of its potential to enable signal reconstruction from significantly fewer data samples than suggested by conventional sampling theory. Compared to conventional ECG compression algorithms, sparse recovery has some important advantages:

(a) It transfers the computational burden from the encoder to the decoder, and thus offers simpler hardware implementations for the encoder.
(b) The location of the largest coefficients in the wavelet domain does not need to be encoded.

Based on the fact that electrocardiogram (ECG) signals can be approximated by a linear combination of a few coefficients taken from a Wavelet basis, in [94] compressed sensing-based approach for ECG signal compression. For solving ECG denoising and compression problem, in [58] was proposed an algorithm based on a sparse separable 2-dimensional transform for both complete and overcomplete dictionaries.

In this chapter we propose a new compression algorithm that seeks, for each block of a signal, the sparsest solution of an underdetermined linear system, in a frame generated on the basis of the first second of the signal itself.

Conversely to the algorithms in the literature, we do not compress the signal using random projection in a lower space, but we use the sparsest representation of the signal in a natural basis.

4.2 The Electrocardiogram

A great variety of electrical signals are produced by the human body due to the chemical activities both in the nerves and in the muscles. Among the others, the heart leads to a characteristic pattern of voltage variations.

The registration and analysis of these bioelectrical activities are very important processes in fields such as the clinical research.

The heart's electrical activity is measured by electrodes that are placed on the skin. The amplitudes, polarities and also times and duration of the different components of the ECG mainly depend on the location of the electrodes on the body. When electrodes are placed with medical purposes (4.1), the standard locations are on both the arms near the wrists, the left leg near the ankle, and several points of the chest called precordial positions. Moreover, a reference electrode is usually placed on the right leg near the ankle.

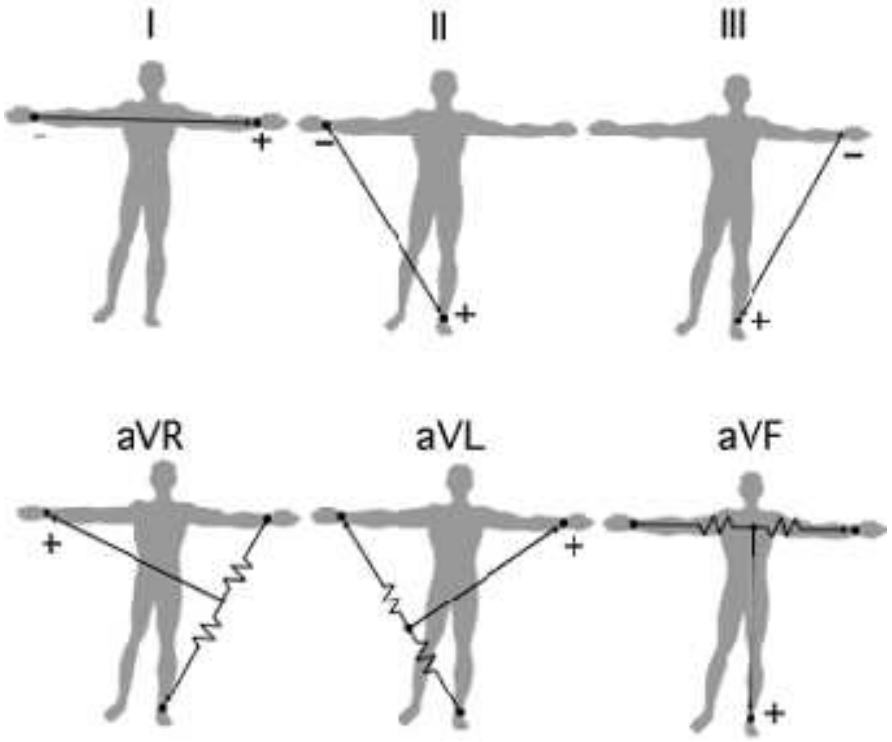

Fig. 4.1 The standard leads (top) and the augmented leads (bottom) reflect the limb electrodes (left arm, right arm, left leg) used to record the heart's electrical axis in the frontal plane.

ECG can be viewed as pseudo periodical signals, characterized by elementary beats in the specific waveform PQRST. The cardiac cycle begins with the P wave which corresponds to the period of atrial depolarization in the heart. It follows the QRS complex, which is the most recognizable feature of an ECG waveform and corresponds to the period of ventricular repolarization. The T wave succeeds the QRS complex ending the cardiac cycle. Occasionally a small U wave can be present at the end of the signal, although not containing significant diagnostic information.

Another characteristic of the ECG is the interval, that is the timing between pairs of ECG features (i.e. RR or ST intervals). Such feature is of great interest since it

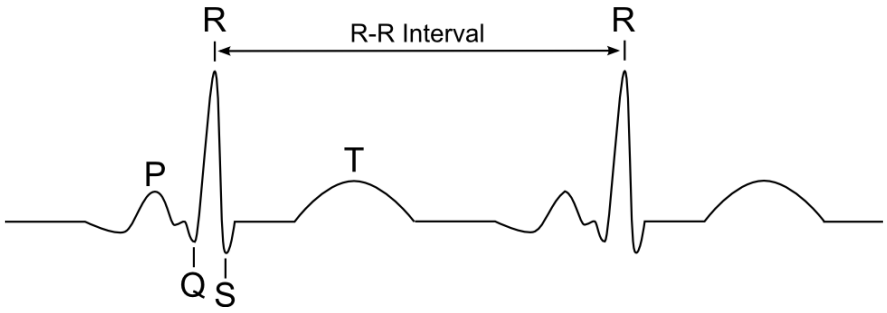

Fig. 4.2 Wave definitions of the cardiac cycle. In figure are noted the P wave, the QRS complex, the T wave and the R-R interval.

provides a measure of the state of the heart and can indicate the presence of certain cardiological conditions [34] or diseases.

4.3 ECG Preprocessing

The aim of the preprocessing phase is to improve the quality of the electrocardiogram signals. The kinds of noise that can disturb the ECG signal are both low frequency baseline wander (BW) and high frequency (50-60 Hz) noise. The first is caused by the respiratory activity while the latter by either electrical power line, poor electrode contacts, body movements or muscular activities.

In the next subsections we describe the preprocessing phase used to suppress noise in our experiments.

4.3.1 Baseline Wander

Some considerations should be taken into account in designing a linear time invariant highpass filter aimed at removing the baseline wander:

- The cut-off frequency should be chosen so that the clinical information in the ECG signal remains undistorted while the baseline wander is removed. Indeed, if we chose a too high cut-off frequency, the filtered signal would contain unwanted artifacts with oscillatory component highly correlated to the beat rate. To this end a reasonable cut-off can be fixed at 0.5 Hz: if the patient is retired, the frequency content of the baseline wander is usually below 0.5 Hz and the lowest frequency components of the ECG spectra are approximately 0.67 Hz.
- Linear phase is highly desirable in order to prevent phase distortions which would alter wave properties of the cardiac cycle such as the duration of the QRS, the ST-T segment level and the duration of the T wave.

The Finite Impulse Response (FIR) filters would satisfy the two requests, but they have the disadvantage to have high order. On the contrary, the Infinite Impulse Response (IIR) filters have a lower order, but they have a nonlinear phase response. To overcame this problem we adopt a forward-backward IIR filter, rectifying the nonlinear phase.

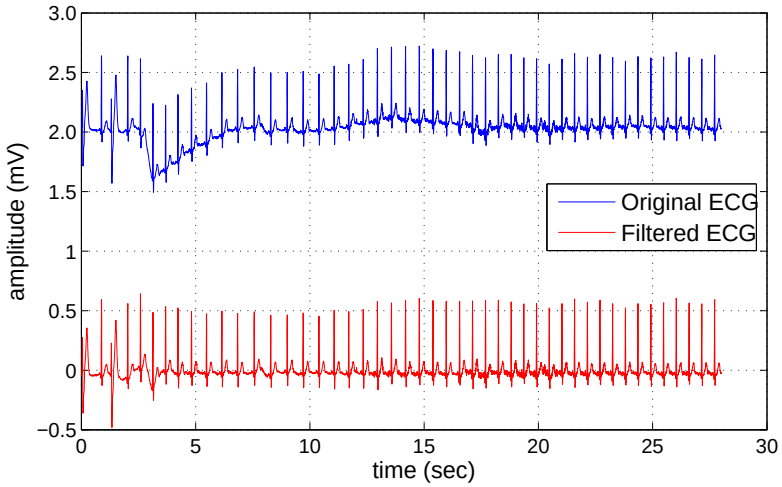

Fig. 4.3 Initial ECG sample (top) and after the IIR filter for baseline wander (bottom).

4.3.2 Powerline Interference

Another common source of noise in ECG signals is the electromagnet field caused by powerline. This kind of noise is characterized by 50 or 60 Hz (depending on the country power line frequency), possibly accompanied by several armonics.

Various precautions can be attempted to reduce the effects of this kind of interference. For example we could either surround electrical devices that can cause line noise or ground and shield the location of the devices. Unfortunately, these shrewdnesses are often not sufficient to remove the power line noise, thus requiring to perform signal processing. To this end linear time band stop notch filters are generally applied.

4.4 R - Peak Detection

A basic task necessary for our electrocardiogram (ECG) compression algorithm is the R wave peak detector. This task is necessary both for building the dictionary used in the sparse representation phase and for the compression stage that they require RR signal bloks. To detect the peaks in ECG signals one can encounter some difficulties, i.e. irregular peak form, irregular distance between peaks and presence of low-frequency component due to patient breathing etc.

In literature many algorithms for QRS and R peak detection were presented. The principal techniques are based on the filtering and thresholding of the signal [63, 89], wavelets [36, 1] and template matching [30]. In our algorithm we use a simple and inexpensive method to detect R peaks, that does not preserve information about the Q and S waves, that are not useful for our purpose.

Let assume to have a digital ECG signal $s = (s_1, \ldots, s_n) \in \mathbb{R}^n$. The first step in the ECG peak detection algorithm is to remove the low-frequency components. We apply a direct Fast Fourier Transform (FFT), remove the low frequencies by thresholding and reconstruct the ECG signal by applying the Inverse Fast Fourier Transform (IFFT).
The second step is to find local maxima using a windowed nonlinear filter. This process puts to zero all the values in the window except the local maximum.
The last step is to remove the small values that called false positives, i.e. high voltage T waves. To do this, we threshold the local maxima found in the previous step with appropriate value that can be easily set equal to a percentage of the absolute maximum of the signal s.

4.5 Compression Method

The algorithm for ECG signal compression and reconstruction is summarized in Fig. 4.4. It consists of a block-wise sparse coding preceded by standard preprocessing and dictionary creation, and followed by a final lossless compression of sparse coefficients, as detailed in the following.

4.5.1 Dictionary Creation and Signal Normalization

The ECG signal $s = (s_1, \ldots, s_n) \in \mathbb{R}^n$ can be modeled as a mixture of two cardiac activities:

$$s = s_{VA} + s_{AA} \tag{4.1}$$

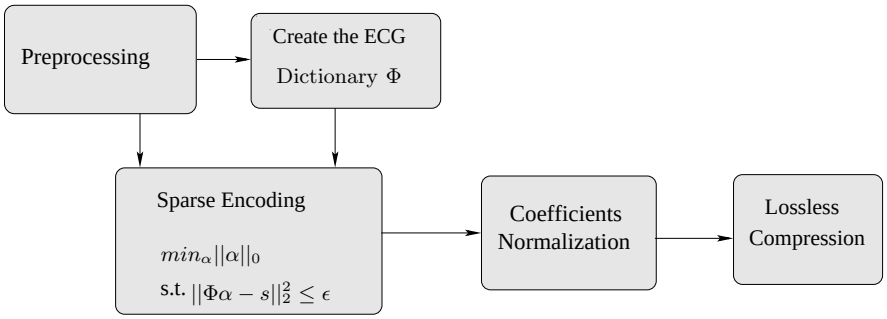

Fig. 4.4 Block diagram of the signal encoding phase.

where $s_{AA} \in \mathbb{R}^n$ denotes the atrial activity signal and $s_{VA} \in \mathbb{R}^n$ represent the ventricular activity signal.

The generation of good sparse models for the atrial s_{AA} and the ventricular s_{VA} signals requires the use of dictionaries capable to fit the signal structure.
The approach proposed in [40] is based on the decomposition of the ECG signal s in a redundant dictionary $\Phi \in \mathbb{R}^{n\times m}$ that is a composition of two sub dictionaries: $\Phi_{AA} \in \mathbb{R}^{n\times k_1}$ suited for the representation of the atrial activity and $\Phi_{VA} \in \mathbb{R}^{n\times k_2}$ able to represent the ventricular activity (with $m = k_1 + k_2$).
In this way the ECG signal can be represented as

$$s = \Phi_{VA}\alpha_{VA} + \Phi_{AA}\alpha_{AA} \tag{4.2}$$

The subdictionary Φ_{VA} in [40] was proposed to be generated by all possible translations of the Generalized Gaussian function

$$g_{VA}(t) = c\exp(-\frac{|t-a|^d}{b}) \tag{4.3}$$

where c is a normalization constant, a represents the position, b is the scale and d is the peakiness, conversely the subdictionary Φ_{AA} was proposed to be generated by all translations of the real Gabor function

$$g_{AA}(t) = c\exp\left(-\left(\frac{t-a}{b}\right)^2\right)\cos\left(\frac{2\pi k(t-a)}{n} - \Delta\theta\right) \tag{4.4}$$

where n is the signal length, a the peak position, c the normalization constant, b the scale and $\Delta\theta$ the phase.

In [75] was proposed to learn the subdictionaries from training samples by alternating dictionary learning phases for Φ_{AA} and Φ_{VA}.

In [53, 30] ECG signals, were represented using a $n \log_2 n$ waveforms analytically represented by wavelets functions (like the Haar, Coiflet4, Deaubuchies18, Battle3 and Symmlet8) as atoms of the dictionary Φ.

Alternative dictionaries are proposed in [61]. The first kind of dictionary, called Resampled Cardiac Pattern, uses as atoms cardiac pattern taken between the middles of successive RR intervals of the ECG signal. Each segment contains the P wave, the QRS complex and the T wave. The second kind of overcomplete dictionary is called Resampled R-centered Cardiac Patterns and is built taking beats patterns from an ECG signal each of which is elastically shrinked and stretched with respect to the peak of the R wave, until it is moved in the middle of the waveform.

Algorithm 5 Dictionary Creation

Require: - the ECG signal s
- the number m of atoms
- the length n of atoms
- the vector of peak positions RR

for $i = 1, \ldots, m$ **do**
$\mu = \frac{RR_{i+1} - RR_i}{2}$ $<$ *calculate the mean of the current RR interval* $>$
$s_l = s_{RR_i:(RR_i+\mu-1)}$ $<$ *set the left half part of the current RR interval into* s_l $>$
$s_r = s_{(RR_i+\mu):(RR_{i+1}-1)}$ $<$ *set the right half part of the current RR interval into* s_r $>$
$z = (0, \ldots, 0) \in \mathbb{R}^{n-(RR_{i+1}-RR_i)}$ $<$ *create the zero padding vector* $>$
$\phi_i = [s_l^T, z^T, s_r^T]^T$ $<$ *stretch the atom to the fixed size* $>$
end for
return $\Phi = \{\phi_1, \ldots, \phi_m\}$ $<$ *the dictionary* Φ $>$

Following the idea in [61] to use natural patterns as representation basis of the ECG signal, in our algorithm we propose a dictionary constructed from pieces of RR interval sampled from the ECG signal that we want to compress. Given an ECG signal, an overcomplete dictionary is built using the cardiac patterns of the signal extracted from an initial transitory of few seconds' duration. To do this, the signal is segmented by taking patterns between successive RR intervals. Each segment s contains two halves of the QRS complex with the T-wave on the left and the P-wave on the right. s is then normalized extending its length to the size of the sample rate by adding the right number of zeros in the middle.

The normalization meaning is obviously that of concentrating the distribution or RR-peaks at the extrema of each segment in order to facilitate the sparse representation stage. In fact, the more the signal is regular and closer to be a cyclostationary process, the higher the sparse representation will be. Thus, the light overhead given by the normalization phase is widely compensated by the high level of sparsity that could be potentially reached. This stage should then imply a first significant

numerosity reduction of coefficients involved in the sparse representation and consequently a first decisive step toward high compression ratio.

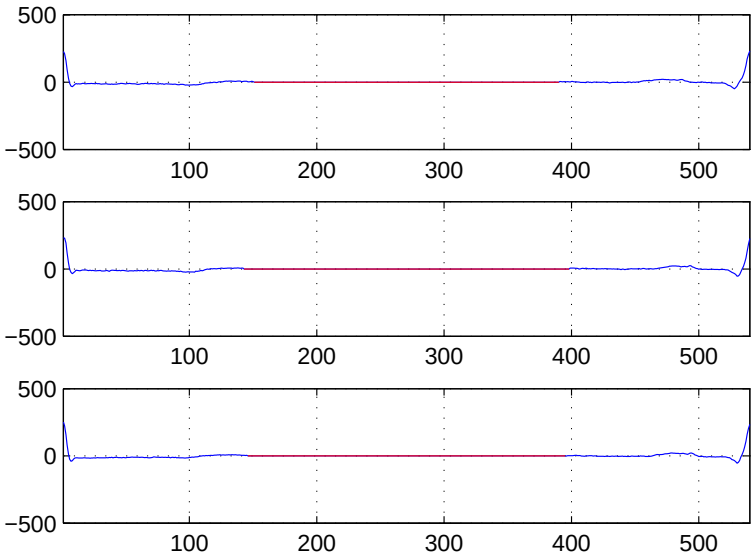

Fig. 4.5 Example of atoms from the original signal s. In red the zero padding for atom length normalization.

After normalization, taken a set of m heart-beat segments ϕ_i ($i = 1 \ldots, m$) of size n, with $n < m$, the dictionary Φ is obtained stacking all basis vectors (or *atoms*) so that $\Phi = [\phi_1, \ldots, \phi_m]$, represents a linear mapping into a suitable feature space $\mathscr{S} \subseteq \mathbb{R}^n$. This algorithm is summarized in Algorithm (5). Each atom of the dictionary Φ so builted, ensures to represent the clinical information such as the PR interval, the P wave configuration, the RR duration etc.

4.5.2 Sparse Representation

The main purpose of the sparsity promotion phase is to adaptively choose the sparsity level k in order to guarantee an upper bound on the discrepancy between the i-th ECG segment s_i and its sparse representation $\hat{s}_i$. To this end we introduce the main criterion involved in such approximation in order to respect the bounded error requirement, namely the Percent Root-Mean Square Difference (PRD) defined as:

$$\mathrm{PRD}(s_i, \hat{s}_i) = 100 \frac{\|s_i - \hat{s}_i\|}{\|s_i\|}.$$

In this setting, the early stage compression can be recasted into the problem of finding the sparsest representation of the *i*-th segment $s_i \in \mathscr{S}$ into the dictionary Φ. A sparse representation for s_i is a linear combination of as less as possible basis elements, i.e., the smallest number of atoms such that $s_i = \sum_{j=1}^{m} \alpha_j \phi_j$, or equivalently, in matricial notation

$$\Phi\alpha = s_i.$$

According to P0 and fixed a bound $\mathrm{PRD}_{\max}$ on the PRD, for each segment s_i the algorithm based on *k*-LIMAPS (called *k*-LIMAPS_ECG) aims at approximately solving the previous system with the regularization problem

$$\min_{\alpha \in \mathbb{R}^m} \|\alpha\|_0 \quad \text{subject to} \quad \mathrm{PRD}(s_i, \Phi\alpha) \leq \mathrm{PRD}_{\max}, \tag{4.5}$$

whose *k*-LIMAPS -based solver pseudo-code is listed in Algorithm 6.

Algorithm 6 *k*-LIMAPS_ECG - Sparse representation

Require: - dictionary pair $\langle \Phi, \Phi^\dagger \rangle$
- signal s_i
- guessed initial sparsity level k_0
- upper bound $\mathrm{PRD}_{\max}$

$\alpha_i \leftarrow k$-LIMAPS $(s_i, \Phi, \Phi^\dagger, k_0)$
if $\mathrm{PRD}(s_i, \Phi\alpha_i) > \mathrm{PRD}_{\max}$ **then**
  while $\mathrm{PRD}(s_i, \Phi\alpha_i) > \mathrm{PRD}_{\max}$ **do**
    $k_i \leftarrow k_i + 1$
    $\alpha_i \leftarrow k$-LIMAPS $(s_i, \Phi, \Phi^\dagger, k_i)$
  end while
else
  if $\mathrm{PRD}(s_i, \Phi\alpha_i) < \mathrm{PRD}_{\max}$ **then**
    while $\mathrm{PRD}(s_i, \Phi\alpha_i) < \mathrm{PRD}_{\max}$ **do**
      $k_i \leftarrow k_i - 1$
      $\alpha_i \leftarrow k$-LIMAPS $(s_i, \Phi, \Phi^\dagger, k_i)$
    end while
  end if
end if
Ensure: a k_i-sparse vector α_i s.t. $s_i \approx \hat{s}_i = \Phi\alpha_i$

The algorithm takes as input the dictionary $\Phi \in \mathbb{R}^{n \times m}$, the pseudoinverse matrix $\Phi^\dagger$, the set of RR intervals S, the initial sparsity level k_0 and the maximum percentage root mean square difference ($\mathrm{PRD}_{\max}$) (4.6) allowed.
It starts finding the k_0-sparsest representation for each $s_i \in S$ RR interval, then it evaluates the PRD and dichotomously searches the minimum k_j such that $\hat{s}_i = \Phi\alpha$, with α k_j-sparse vector and $\mathrm{PRD}(s_i, \hat{s}_i) < \mathrm{PRD}_{\max}$

4.5.3 Quantization and Compression

For long term signal of N heart-beats the data we have to compress are of three kinds:

1. the set of coefficients (amended by zeros elements) contained in the vectors $\alpha_1, \ldots, \alpha_N$ whose number amount to $K = \sum_{i=1}^{N} k_i$. Fixed q bits for 2^q quantization levels we simply rescale and quantize each coefficient c into the integer number
$$c_q = \text{round}\left(\frac{c - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}} (2^q - 1) \right) \tag{4.6}$$
where $\alpha_{\min}$ and $\alpha_{\max}$ are the minimum and maximum value coefficients.
2. Once processed coefficients, we have to compress the positions of coefficients different from zero, i.e., the characteristic vectors $\{\chi_{\alpha_1}, \ldots, \chi_{\alpha_N}\}$, one for each n-tuple α_i, identifying the positions of coefficients not null. The concatenation of the binary vectors χ_{α_i} forms a sparse bit-vector with K ones over nN bits, with $K \ll nN$. An effective way to compress them is to use of Huffman coding [65] for bit-vectors. It is partitioned into blocks of fixed size l and statistics are collected on the frequency of occurrence of the 2^l bit patterns. Based on these statistics the set of blocks is Huffman encoded and the bitmap itself is encoded as a sequence of such codewords For sparse vectors the l-bit block consisting of zeros only and blocks with only a single 1-bit have much higher probabilities than the other blocks so the average codeword length of the Huffman code will be smaller than l.
3. The last set of numbers $\rho_1, \ldots, \rho_N$ to be compressed are those involved in the zero-padding normalization task, accomplished for each heart-beat. An effective way to encode such a list is with the delta encoding, which stores the differences between pairs of successive values in the list. Such encoding is particularly useful in our case where most of the differences are small. The final sequence of differences can be thus compressed much more efficiently by most of the compression algorithms.

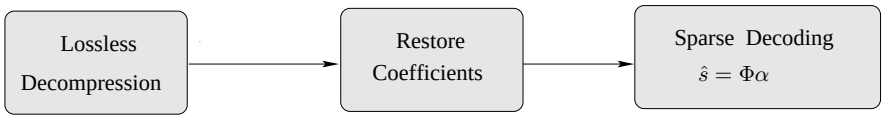

Fig. 4.6 Block diagram of the signal reconstruction phase.

4.6 Performance Evaluation

The compression performances for a cardiac signal x of fixed duration is computed, as usual, in terms of Compression Ratio (CR) [102]:

$$\mathrm{CR} = \frac{\#\text{ bits of } x}{\#\text{ bits of } \hat{x}}.$$

In our setting the signal to be compressed is a sequence of N cardiac segments $x = [s_1, \ldots, s_N]$ picked sequentially by a sliding window without overlapping. As already mentioned in the previous section, to accomplish this task we have to compress three kinds of data: not null coefficients in each vectors α_i, characteristic bit-vectors χ_i and padding numbers ρ_i. Putting all tighter, the CR thus becomes

$$\mathrm{CR} = \frac{N \times f_s \times 11}{B_\alpha + B_\chi + B_\rho},$$

where f_s the sampling frequency and B_α, B_χ, B_ρ the number of bits coding coefficients, characteristic vectors and padding numbers respectively.

However, in the case of lossy data compression, the *data compression ratio* does not provide sufficient detail on the performances because it does not reflect distinction between the original signal and the reconstructed signal.

Regarding the error, since PRD [6] measure heavily depends on the mean value $\bar{s}$ of the ECG signal s, it is sometime more suitable to use the normalized PRD measure given by

$$\mathrm{PRDN} = 100\frac{\|s - \hat{s}\|}{\|s - \bar{s}\|}.$$

where $\bar{s}$ is the mean value of the signal s.

Another measure typically used to evaluate the reconstruction error is the *root mean square error* (RMS) defined as

$$RMS = \sqrt{\frac{1}{n}||s - \hat{s}||_2^2} \tag{4.7}$$

where n is the length of the window over which the RMS is evaluated [126]. The RMS quantifies the amount of the original signal discarded by the compression method, as measured by the difference between the signal s and $\hat{s}$.

An important aspect of performance evaluation is the choice of the ECG database. As the performance of a method depends on the noise level, the evaluation should be based on data representative of the application of interest.
In the next session we describe the PhysioNet MIT-BIH Arrhythmia database [59], used to evaluate the performances of our algorithm.

4.7 PhysioNet MIT-BIH Arrhythmia Database

PhysioNet offers free web access to large collections of recorded physiologic signals (PhysioBank) and related open-source software (PhysioToolkit) [59].
Since 1975 the Massachusetts Institute of Technology (MIT) together with the Beth Israel Deaconess Medical Center have carried out research concerning medical examinations analysis and related points.
One of the first major product of that effort was the MIT-BIH Arrythmia database (downloadable from the PhysioNet web site [59]) which was completed in 1980.
This database contains standard test material for evaluation of arrythmia detectors.

The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three recordings were chosen random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample.

The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range and each signal was independently annotated from two or more cardiologists.

4.8 Experimental Results

In order to evaluate the compression performances of the *k*-LIMAPS based compression algorithm, we have performed extended computer simulations. Tests were conducted using the records 100, 101, 102, 109, 111, 112, 113, 115, 117, 119 and 121 extracted from the MIT-BIH Arrythmia database [63]. These signals are 30 minutes registrations sampled at 11 bits with a sampling rate of 360 Hz.

Each recording was splitted in two parts: the first one of about 10 minutes, used for the dictionary creation phase, the second one for the sparse representation and signal compression.

Our experiments were performed on an AMD Athlon II X4 630 Processor 64 bit, 2.8 GHz processor with 4 GB of memory, implementing our algorithm in MATLAB v.2011b.

In table (4.1) we can see results for the compression of the selected records. Each one is quantized with two different quantization rate: 6 and 7 bits. With the 6 bits

Table 4.1 Examples of original and compressed ECG waveforms, along with the reconstruction error with 6 bits of quantization. The signal duration for dictionary grabbing is about 10 min, while the block of compressed ECG is about 20 min.

Sig.Num.	CR	PRD	PRDN	SNR	Time Tot.	Time Dict.	Time Comp.
100	78.20	0.72	18.03	34.27	30.09	9.47	20.62
101	80.24	0.68	14.66	38.40	30.08	11.23	18.84
102	50.54	0.69	18.45	33.81	30.08	9.86	20.22
103	46.32	0.75	12.57	41.47	30.08	10.22	19.87
109	24.86	1.43	13.70	39.76	30.08	8.36	21.73
111	31.05	0.98	26.20	26.79	30.07	10.29	19.78
112	34.06	0.71	16.58	35.94	30.08	8.42	21.67
113	37.42	1.08	14.08	39.20	30.08	12.33	17.75
115	38.26	0.65	9.76	46.53	30.08	11.27	18.82
117	38.94	0.61	14.42	38.73	30.08	14.12	15.96
119	16.26	3.74	32.19	22.67	30.09	10.36	19.73
121	26.67	0.67	17.36	35.02	30.08	11.88	18.20

Table 4.2 Examples of original and compressed ECG waveforms, along with the reconstruction error with 7 bits of quantization. The signal duration for dictionary grabbing is about 10 min, while the block of compressed ECG is about 20 min.

Sig.Num.	CR	PRD	PRDN	SNR	Time Tot.	Time Dict.	Time Comp.
100	75.12	0.68	17.22	35.19	30.09	9.47	20.62
101	76.46	0.60	12.91	40.95	30.08	11.23	18.84
102	48.47	0.68	18.16	34.11	30.08	9.86	20.22
103	44.33	0.69	11.57	43.14	30.08	10.22	19.87
109	23.57	1.04	9.97	46.12	30.08	8.36	21.73
111	29.44	0.73	19.51	32.69	30.07	10.29	19.78
112	32.55	0.68	15.99	36.66	30.08	8.42	21.67
113	35.49	0.75	9.82	46.42	30.08	12.33	17.75
115	36.57	0.61	9.18	47.75	30.08	11.27	18.82
117	37.13	0.56	13.38	40.23	30.08	14.12	15.96
119	15.24	1.90	16.36	36.20	30.09	10.36	19.73
121	25.29	0.60	15.63	37.12	30.08	11.88	18.20

quantization we can see that the compression rate increases respect to the 7 bits quantization. The only signal that drastically decrease the quality passing from a quantization of 7 bits to 6 bits is the record 119. This fact is due to the irregularity of the RR interval.

In figure (4.7) we can see an example of compressed signal extracted from the record 100 of the MIT-BIH Arrythmia database. From the top we illustrate the original signal without baseline, the reconstructed signal and the error vector. We note that the errors are equally distributed over all the signal.

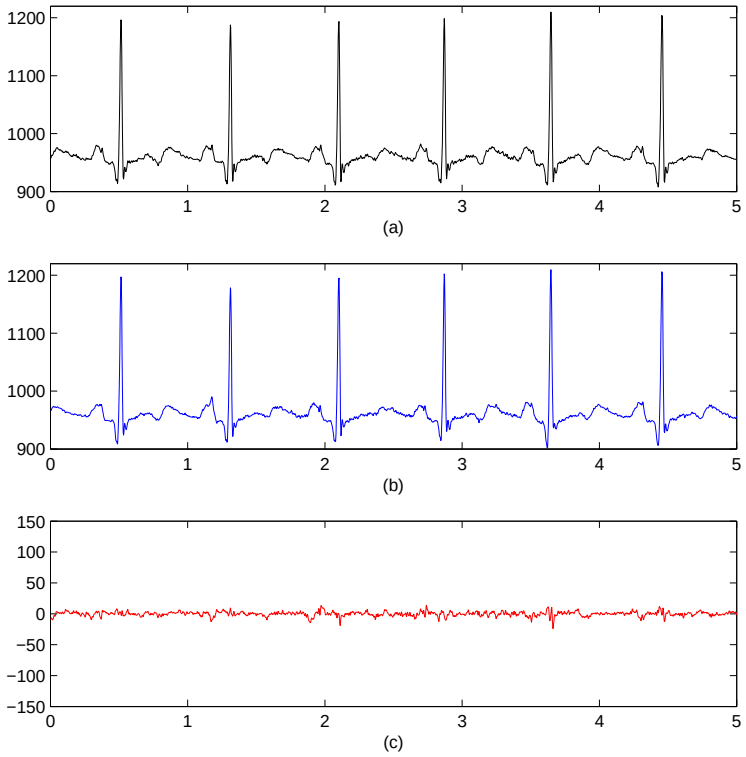

Fig. 4.7 Compression of the record 100 with $CR = 75.12$ and $PRD = 0.68$. (a) Original signal. (b) Reconstructed signal. (c) Error signal.

In table (4.3) we summarize comparative results with the algorithms DCeq-MMP [52] (Multididimensional Multiscale Parser) and with Foveation [33] + SPIHT algorithm (Set Partitioning in Hierarchical Trees [101, 106]) for records 100, 102, 115, 117 and 119.

In all the experiments conducted, the PRD of each signal was calculated with respect to the original non-processed signal mean. This turns out to be necessary because different preprocessing filtering can change significantly the average signal magnitude, making the results incomparable.
As we can see for almost all the records, the compression rate obtained by our algorithm is the highest, maintaining a PRD less or comparable to other algorithms. An exception is found for the record 119: in this case this the irregularities of the RR interval and the presence of large muscular noise, the atoms of the dictionary do not represent properly the signal when only few coefficients are adopted.

Table 4.3 Performance comparison of different ECG compression schemes

Method	Record	CR	PRD
DCeq - MMP [52]	100	24	3.30
Foveation + SPIHT [33]	100	20	0.52
k-LIMAPS (7 bit)	100	74.12	0.68
k-LIMAPS (6 bit)	100	78.20	0.72
DCeq - MMP [52]	102	10	2.03
k-LIMAPS (7 bit)	102	48.47	0.68
k-LIMAPS (6 bit)	102	50.54	0.69
DCeq - MMP [52]	115	30.6	2.92
k-LIMAPS (7 bit)	115	36.57	0.61
k-LIMAPS (6 bit)	115	38.26	0.65
DCeq - MMP [52]	117	13	0.91
Foveation + SPIHT [33]	117	32	0.51
k-LIMAPS (7 bit)	117	37.13	0.56
k-LIMAPS (6 bit)	117	38.19	0.61
DCeq - MMP [52]	119	20	1.83
DCeq - MMP [52]	119	10	1.07
DCeq - MMP [52]	119	8	0.91
Foveation + SPIHT [33]	119	20	0.49
k-LIMAPS (7 bit)	119	15.24	1.90
k-LIMAPS (6 bit)	119	16.26	3.74

4.9 Garanteed-Percentage Root Mean Square Difference Error

In this section we present a variant of the ECG compression algorithm explained in 4.5. The main difference concerns the dictionary construction: in this variant it is a concatenation of two dictionaries, one extracted directly from a portion of the signal to be compressed, the second is a random generated matrix used with the purpose of improving the ability of sparse recovery of *k*-LIMAPS into a natural base. Moreover no zero padding is added to the natural signals, thus no noise in the superposition of atoms are added and propagated to the compressed coefficients. The algorithm proposed in this section ensures that the recovered signal has a guaranteed PRD error.

4.9.1 Dictionary Creation

The dictionary is created by slicing a natural ECG signal with a fixed window size. Atoms are selected as non overlapped pieces of signal that contain (in average) only one beat. To do this, in the case the ECG signal used, belongs to subjects at rest, we can simply choose a window size of dimension approximately equal to the sampling rate. This assumption is justified by the fact that in normal conditions, the average heart beat rate is in the interval of 60-80 bpm (beat per minute). If an estimate of the average beats per minutes is available, another possible choice for the temporal resolution is given by the formula

$$tr = \lceil \frac{abpm}{60} * fs \rceil$$

where $abpm$ is the average beat per minutes and fs is the sampling rate frequency.

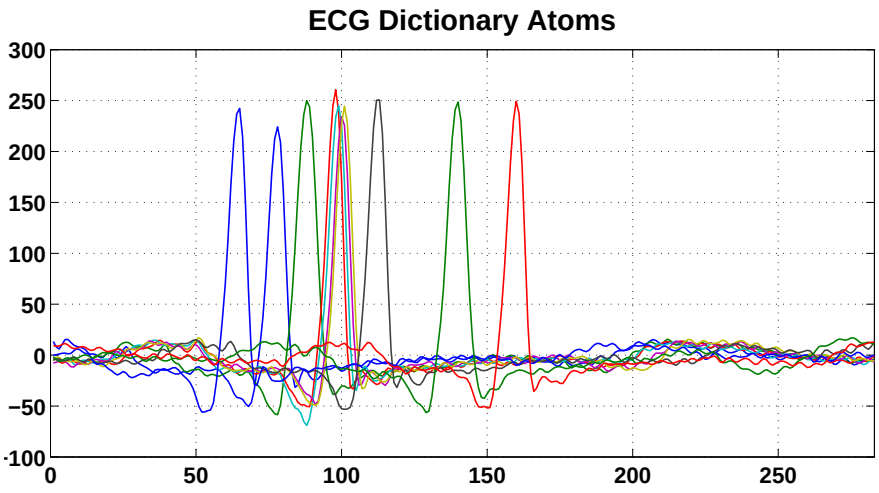

Fig. 4.8 Example of Dictioanry Atoms.

In figure 4.8 we can see examples of dictionary atoms extracted by slicing the ECG signal with a temporal resolution equal to the sampling rate.

A generalization of the notion is presented in [8], where two fundamental measures of coherence are introduced: worst-case coherence and the average coherence , among the columns of a matrix.

Let Φ be an $n \times m$ matrix with unitary norm columns $\phi_i \in \mathbb{R}^n$, [8] defines

$$\mu(\Phi) = \max_{i,j:i\neq j} | < \phi_i, \phi_j > | \tag{4.8}$$

$$\nu(\Phi) = \frac{1}{m-1} \max_i | \sum_{j:j\neq i} < \phi_i, \phi_j > | \tag{4.9}$$

respectively the worst-case coherence and the average coherence of Φ. The smaller the worst-case coherence, the less similar the columns. On the other hand, the average coherence is a measure of the spread of the columns of a dictionary matrix within the n-dimensional unit ball. TNote that, in addition to having zero worst-case coherence, orthonormal bases also have zero average coherence.

The uniqueness of the sparsest solution of an underdetermined linear system depends on the worst-case coherence of the dictionary Φ. Recalling the theorem 1.3: i f a linear system $\Phi\alpha = s$ has a solution α such that $||\alpha||_0 < \frac{1}{2}(1 + \frac{1}{\mu(\Phi)})$, than α is the sparsest solution. Futhermore, the worst case analysis tells us that we can recover superposition of k atoms as long as

$$k \lesssim \frac{1}{\mu(\Phi)} \approx \sqrt{m} \tag{4.10}$$

The resulting worst case bounds for recoverable sparsity levels turn out to be overly pessimistic and quite in contrast to the much better performance archived in practice.

The natural bases extracted from the ECG signal, does not guarantee a low worst case coherence, making difficult to recover the sparsest solution of the underdetermined system.

It was established in [8] that when the average coherence $\nu(\Phi)$ is sufficiently smaller than $\mu(\Phi)$, a number of guarantees can be provided for sparse signal processing.

To decrease the average coherence of our dictionary, we concatenate the dictionary containing atoms extracted from natural ECG signal with a dictionary random sampled from a Gaussian distribution.

Formally let Φ be the dictionary created by segmenting the natural ECG signal, and let Ψ be the dictionary sampled from a Gaussian distribution $\mathcal{N}(0, \frac{1}{n})$, we can recast the regularization problem 4.5 as

$$\min_{\alpha_i \in \mathbb{R}^m} \| \alpha_i \|_0 \quad \text{subject to} \quad \text{PRD}(s_i, \Sigma\alpha_i) \leq \text{PRD}_{\max}, \tag{4.11}$$

with Σ defined as

$$\Sigma = [\Phi|\Psi]$$

that is the concatenation of the two dictionaries.

As proved in [8], we note that the average coherence of Gaussian matrix $\Psi \in \mathbb{R}^{n\times p}$ with i.i.d. $\mathcal{N}(0, \frac{1}{n})$ entries, is

$$\nu(\Psi) \leq \frac{\sqrt{15 \log p}}{n}$$

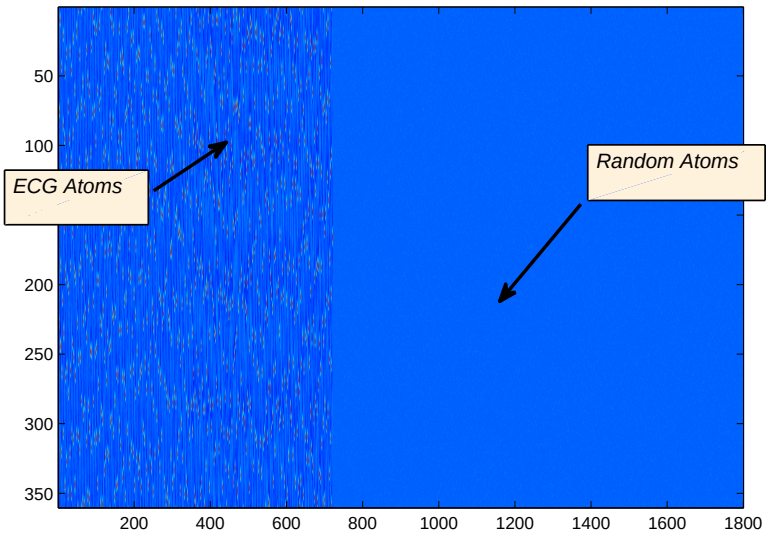

Fig. 4.9 ECG Dictionary with Gaussian atoms. The left part of the dictionary contains atoms sampled from ECG signal, right part are atoms random sampled from Gaussian distribution.

with probability exceeding $1-2p^{-2}$ as long as $p > n \geq 60 \log p$.

Conversely, the dictionary Φ extracted by natural ECG signal has a high average coherence, hence concatenating the two dictionaries we can establish the following relation

$$\nu(\Psi) \leq \nu(\Sigma) \leq \nu(\Phi)$$

4.9.2 Experimental Results

To analyze the compression performances of the method described in this session, we have performed a wide range of experiments using the records taken from the MIT-BIH Arrythmia database [63], as we have done to evaluate the k-LIMAPS _ECG in section 4.5.

Each record signal was divided in two partitions: one of about 12 minutes for the creation of the subdictionary, and the other to be used to measure the compression capability of the method.

In table (4.4) we resume the results of compression performances reached by the Garanteed-Percentage Root Mean Square Difference Error algorithm, obtained over a selection of ECG records taken from the MIT-BIH Arrythmia database.

Table 4.4 *k*-LIMAPS Compression Results for the Garanteed-Percentage Root Mean Square Diffenrence Error algorithm.

Record	CR	PRD	PRDN	SNR	Time dic.	Time sign.
100	41.87	0.49	12.42	41.72	7.57	22.51
101	37.85	0.52	11.14	43.90	11.21	18.87
102	23.04	0.49	14.23	39.00	8.17	21.92
103	30.35	0.57	9.41	47.27	9.01	21.07
109	18.02	0.59	6.09	55.96	6.12	23.97
111	26.61	0.55	15.24	37.63	8.67	21.42
112	20.32	0.50	13.50	40.05	6.07	24.02
113	22.18	0.66	9.88	46.29	12.20	17.88
115	23.56	0.59	8.82	48.57	10.27	19.81
117	32.11	0.55	15.89	36.79	16.57	13.51
119	13.04	0.71	7.11	52.87	11.41	18.67
121	33.94	0.48	14.41	38.75	11.28	18.81
201	23.14	0.49	15.50	37.29	10.33	19.76

After the sparse representation, each obtained coefficient is quantized in 7 bits. We can observe a decreasing of compression performances if compared with the method that uses the dictionary obtained by signal length normalization with zero padding. Conversely, the PRD obtained in the compression phase, in each signal respects the imposed limit of 0.5.

The records with lowest compression rate are 119 and 109. The irregularity of the RR intervals and the low quality recording, compromise the performances obtained by the compression algorithm in these two examples.

4.10 Conclusion

This chapter proposes a new algorithm to perform electrocardiographic signal compression. It rest on the recent but strengthen paradigm of sparsity representation on overdetermined dictionary. It is developed on the basis of a recent optimization algorithm which iterates suitable nonconvex mappings onto the solution affine space after a brief trajectory accomplished inside the space. The novelty is mainly represented by the idea of extracting base waveforms from the cardiac signal in order to provide succinct representation via linear combination for the long term part of

the signal. Therefore the compression task is recasted into the problem of finding a sparse representation of signal blocks and successive encoding of the information held by few coefficients saved along with their positions.

Experimentally we have shown the effectiveness of our method which reaches high compression rate maintaining an acceptable percent root-mean square difference level. The proposed method has been experimentally compared with the state of the art algorithms in our knowledge.

Subsequently has been proposed a variant of the *k*-LIMAPS _ECG compression algorithm, that has as main difference from the previous a different representation dictionary. The dictionary is a composition of two different subdictionaries: the first one extracted directly from a portion of the signal to be compressed, the second one is a random generated Gaussian matrix used with the purpose of improving the ability of sparse recovery of *k*-LIMAPS into a natural base.

With this compression algorithm, experiments have shown lower compression rate if compared with previous method but capable of guaranteeing a percent root-mean square difference level in reconstruction phase.

These techniques appears to be applicable in real contexts, especially in offline outpatient applications. Future work suggests the necessity to make it computationally tractable in embedded mobile devices, like Holter, to meet the medical needs of the last few years and enabling the application of our algorithm in environments like personal area networks (PAN).

Appendix A
MATLAB Code

A.1 LiMapS

```
function [ alpha ] = LiMapS(s, D, DINV, gamma, maxIter, ftype)
%
% [ alpha ] = LIMAPS(s, D, DINV, gamma, maxIter, ftype)
%
%   Returns the sparsest vector alpha which satisfies underdetermined
%   system of equations  D*alpha=s, using  Lipschitzian Mapping for Sparsity
%   Recovery algorithm.
%   The dictionary matrix  D should have more columns than rows. The
%   number of rows of dictionary D  must be equal to the length of the
%   column observation vector(or matrix) s.
%
%     The observations signal(s) s and the dictionary D necessarily, must
%   be provided by the user. The other parameters have defult values, or
%   may be provided by the user.
%
%   This algorithm work fine with matrix too, returning the sparsest matrix
%   alpha that satisfy the underdetermined system of equations D*alpha=s.
%   Then the dictionary D is of size [n x m], s must be of size [ s x T ] and
%   as consiquence the coefficients matirx alpha is of size [ m x T]
%
%   s:
%       The observation vector (matrix)
%
%   D:
%       The dictionary
%
%   DINV (optional):
%       The pseudo-inverse of the dictionary matrix D
%
%   gamma (optional):
%       The increase factor (default 1.01)
%
```

```
%   maxIter (optional):
%       Maximum number of iterations (default 1000)
%
%   ftype (optional):
%       String that represent the shrinking function
%               - 'exp'     (default) exponential
%               - 'exp2'    exponential function
%               - 'htan'    hyperbolic function
%               - 'abs'     absolute value function
%               - 'sqrt'    square root function
%               - 'err'     error function
%               - 'gud'     Gudermannian function
%
% Authors: Alessandro Adamo and Giuliano Grossi
%
% WEB PAGE:
% ------------------
%    http://dalab.dsi.unimi.it/limaps
%
% HISTORY:
%--------------
% Version 2.0: 9 Feb. 2013
%        official version.
%

if(nargin<2||nargin>6)
    error('Wrong parameters number');
end
if(nargin==2)
    DINV=pinv(D);
    gamma=1.01;
    maxIter=1000;
    ftype='exp';
end
if(nargin==3)
    gamma=1.01;
    maxIter=1000;
    ftype='exp';
end
if(nargin==4)
    maxIter=1000;
    ftype='exp';
end
if(nargin==5)
    ftype='exp';
end


%% --  INITIALIZATION  --
alpha = DINV*s;

lambda = 1/max(abs(alpha(:)));
epsilon=1e-5;                   % stopping criteria
alpha_min=1e-2;
```

```
% choose the shrinking function
f=shrinkage_function (ftype);

%% --  CORE  --
for i=1:maxIter

    alphaold=alpha;
    % apply sparsity constraction mapping: increase sparsity
    beta = f(alpha,lambda);

    beta(abs(beta)<alpha_min)=0;

    % apply the orthogonal projection
    alpha = beta-DINV*(D*beta-s);

    % update the lambda coefficient
    lambda = lambda*gamma;

    % check the stopping criteria
    if (norm(alpha(:)-alphaold(:))<epsilon)
        break;
    end

    if (mod(i,100))
        if(sum(abs(f(alpha,lambda)./alpha)>1/lambda)>size(D,1))
           break;
        end
    end

end

%% -- REFINEMENT --

% threshold the coefficients
alpha(abs(alpha)<alpha_min) = 0;

% final refinements of the solution
for i=1:size(alpha,2);
    alpha_ind = alpha(:,i)≠0;
    D1 = D(:,alpha_ind);
    alpha(alpha_ind,i) = alpha(alpha_ind,i) - pinv(D1)*(D1*alpha(alpha_ind,i)-s(:,i));
end

end

function [ f ] = shrinkage_function ( type )

    f = @exponential_func;
    switch(type)
        case 'exp'
            f = @exponential_func;
        case 'exp2'
            f = @exponential2_func;
```

```
        case 'htan'
            f = @hyperbolic_tangent_func;
        case 'abs'
            f = @absolute_value_func;
        case 'sqrt'
            f = @square_root_func;
        case 'err'
            f = @error_func;
        case 'gud'
            f = @gudermannian_func;
    end

end

function [ y ] = exponential_func(x,lambda)
    y=x.*(1-exp(-lambda*abs(x)));
end

function [ y ] = exponential2_func(x,lambda)
    y=x.*(2./(1+exp(-lambda*abs(x)))-1);
end

function [ y ] = hyperbolic_tangent_func(x,lambda)
    y=x.*tanh(lambda*abs(x));
end

function [ y ] = absolute_value_func(x,lambda)
    y=x.*(lambda*x.^2./(1+lambda*x.^2));
end

function [ y ] = square_root_func(x,lambda)
    y=x.*(lambda*abs(x))./sqrt(1+(lambda*x).^2);
end

function [ y ] = error_func(x,lambda)
    y=x.*erf(lambda*abs(x));
end

function [ y ] = gudermannian_func(x,lambda)
    y=x.*(2/pi*atan(sinh(pi/2*lambda*abs(x))));
end
```

A.2 *k*-LiMapS

```
function [ alpha ] = KLiMapS(s, D, k, DINV, maxIter, ftype)
%
% [ alpha ] = KLIMAPS(s, D, k, DINV, maxIter, ftype)
%
%   Returns the k-sparse vector alpha which satisfies underdetermined
%   system of equations  D*alpha=s, using  Lipschitzian Mapping for Sparsity
%   Recovery algorithm.
%   The dictionary matrix  D should have more columns than rows. The
```

```
%   number of rows of dictionary D  must be equal to the length of the
%   column observation vector(or matrix) s.
%
%     The observations signal(s) s and the dictionary D necessarily, must
%   be provided by the user. The other parameters have defult values, or
%   may be provided by the user.
%
%   This algorithm work fine with matrix too, returning the sparsest matrix
%   alpha that satisfy the underdetermined system of equations D*alpha=s.
%   Then the dictionary D is of size [n x m], s must be of size [ s x T ] and
%   as consiquence the coefficients matirx alpha is of size [ m x T]
%
%   s:
%       The observation vector (matrix)
%
%   D:
%       The dictionary
%
%   k:
%       Number of coefficients to select
%
%   DINV (optional):
%       The pseudo-inverse of the dictionary matrix D
%
%   gamma (optional):
%       The increase factor (default 1.01)
%
%   maxIter (optional):
%       Maximum number of iterations (default 1000)
%
%   ftype (optional):
%       String that represent the shrinking function
%               - 'exp'     (default) exponential
%               - 'exp2'    exponential function
%               - 'htan'    hyperbolic function
%               - 'abs'     absolute value function
%               - 'sqrt'    square root function
%               - 'err'     error function
%               - 'gud'     Gudermannian function
%
% Authors: Alessandro Adamo and Giuliano Grossi
% Version: 1.0
% Last modified: 1 Aug. 2011.
%
% WEB PAGE:
% ------------------
%    http://dalab.dsi.unimi.it/klimaps
%
% HISTORY:
%--------------
% Version 2.0: 9 Feb. 2013
%        official version.
%
    if(nargin<3||nargin>6)
```

```
        error('Wrong parameters number');
    end
    if(nargin==3)
        DINV=pinv(D);
        maxIter=1000;
        ftype='exp';
    end
    if(nargin==4)
        maxIter=1000;
        ftype='exp';
    end
    if(nargin==5)
        ftype='exp';
    end

    %% --  INITIALIZATION  --
    alpha = DINV*s;
    a=sort(abs(alpha));
    lambda = 1/a(end-k);

    epsilon=1e-4;                    % stopping criteria

    % choose the shrinking function
    f=shrinkage_function (ftype);

    %% --  CORE  --
    for i=1:maxIter

        alphaold=alpha;

        % apply sparsity constraction mapping: increase sparsity
        beta = f(alpha,lambda);

        % apply the orthogonal projection
        alpha = beta-DINV*(D*beta-s);
        % update the lambda coefficient
        a=sort(abs(alpha));
        lambda = 1/a(end-k);

        % check the stopping criteria
        if (norm(alpha-alphaold)/norm(alphaold)<epsilon || isnan(lambda))
            break;
        end

    end

    %% -- REFINEMENT --

    % final refinements of the solution
    alpha(abs(alpha)≤1/lambda)=0;

    idx = alpha≠0;
    D1 = D(:,idx);
    alpha(idx) = alpha(idx) - pinv(D1)*(D1*alpha(idx)-s);
```

```
end


function [ f ] = shrinkage_function ( type )

    f = @exponential_func;
    switch(type)
        case 'exp'
            f = @exponential_func;
        case 'exp2'
            f = @exponential2_func;
        case 'htan'
            f = @hyperbolic_tangent_func;
        case 'abs'
            f = @absolute_value_func;
        case 'sqrt'
            f = @square_root_func;
        case 'err'
            f = @error_func;
        case 'gud'
            f = @gudermannian_func;
    end

end

function [ y ] = exponential_func(x,lambda)
    y=x.*(1-exp(-lambda*abs(x)));
end

function [ y ] = exponential2_func(x,lambda)
    y=x.*(2./(1+exp(-lambda*abs(x)))-1);
end

function [ y ] = hyperbolic_tangent_func(x,lambda)
    y=x.*tanh(lambda*abs(x));
end

function [ y ] = absolute_value_func(x,lambda)
    y=x.*(lambda*x.^2./(1+lambda*x.^2));
end

function [ y ] = square_root_func(x,lambda)
    y=x.*(lambda*abs(x))./sqrt(1+(lambda*x).^2);
end

function [ y ] = error_func(x,lambda)
    y=x.*erf(lambda*abs(x));
end

function [ y ] = gudermannian_func(x,lambda)
    y=x.*(2/pi*atan(sinh(pi/2*lambda*abs(x))));
end
```

Appendix B
Phase Transition

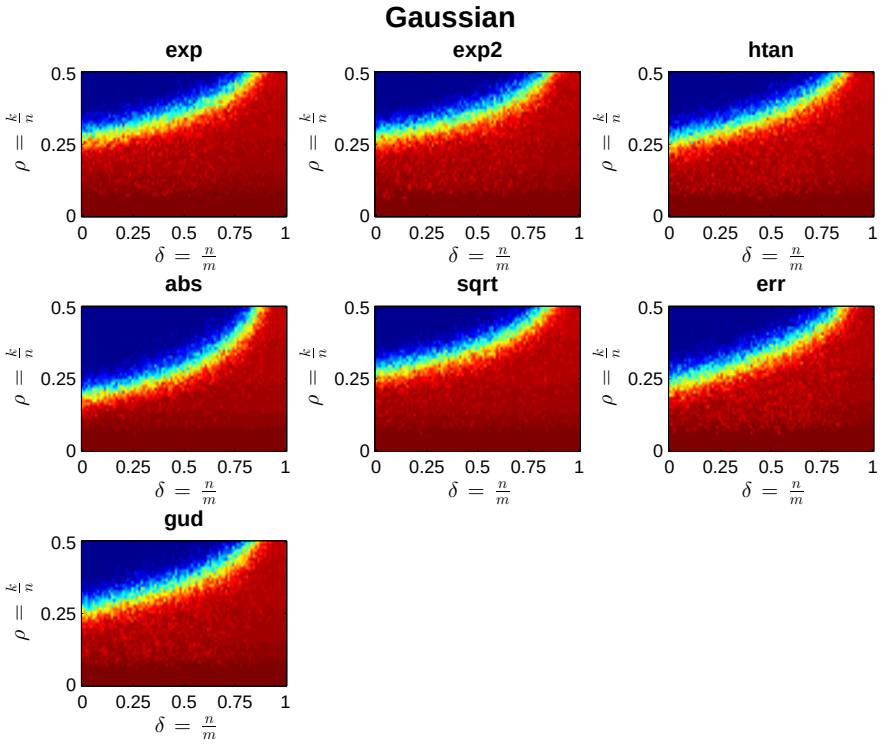

Fig. B.1 Phase transition for Guassian matrices.

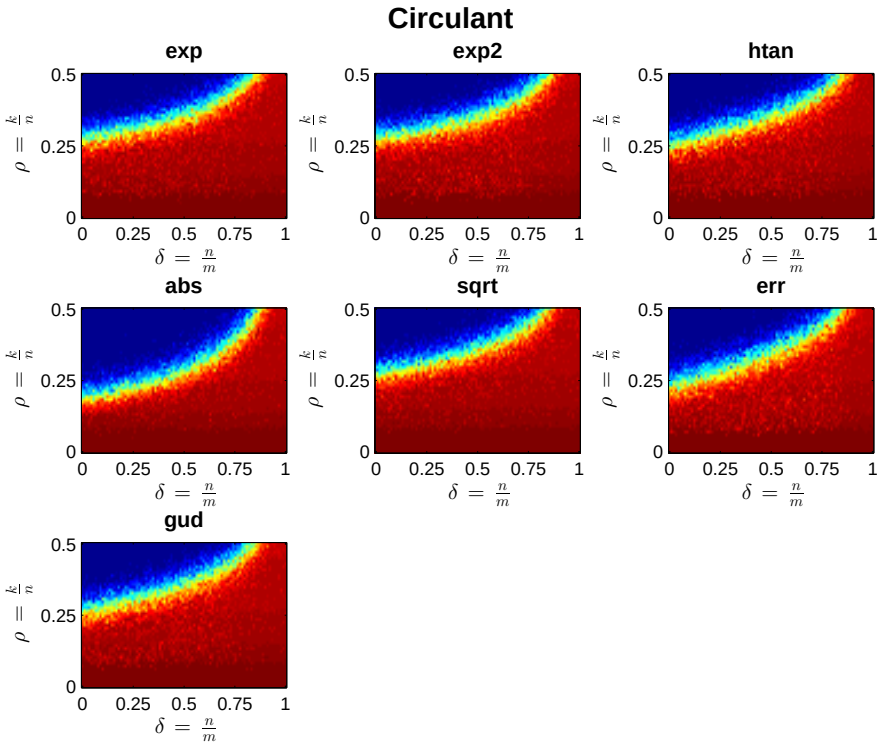

Fig. B.2 Phase transition for Circulant matrices.

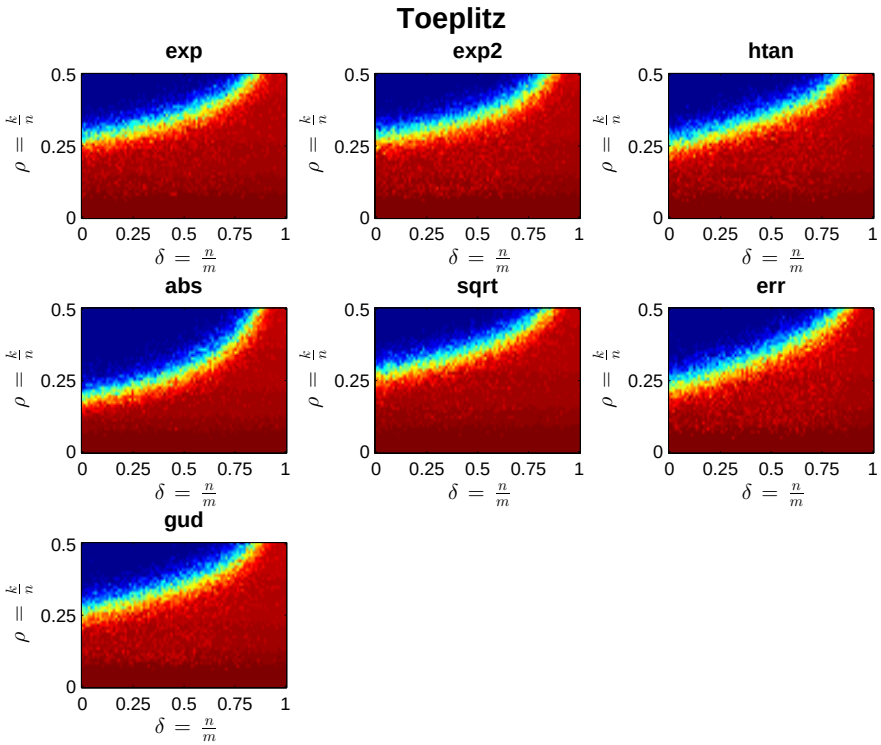

Fig. B.3 Phase transition for Toeplitz matrices.

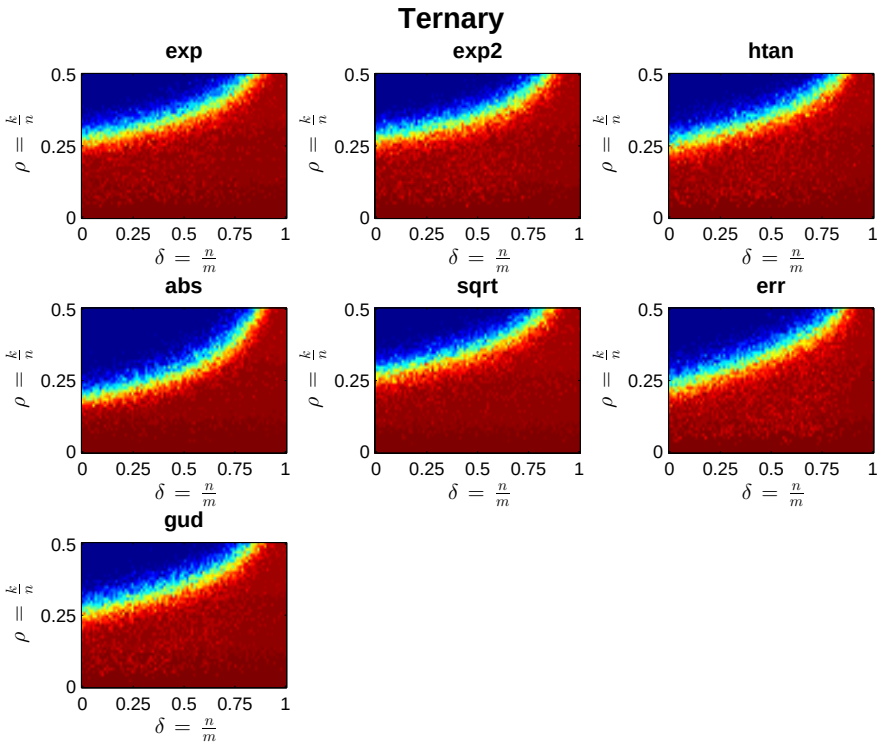

Fig. B.4 Phase transition for Ternary matrices.

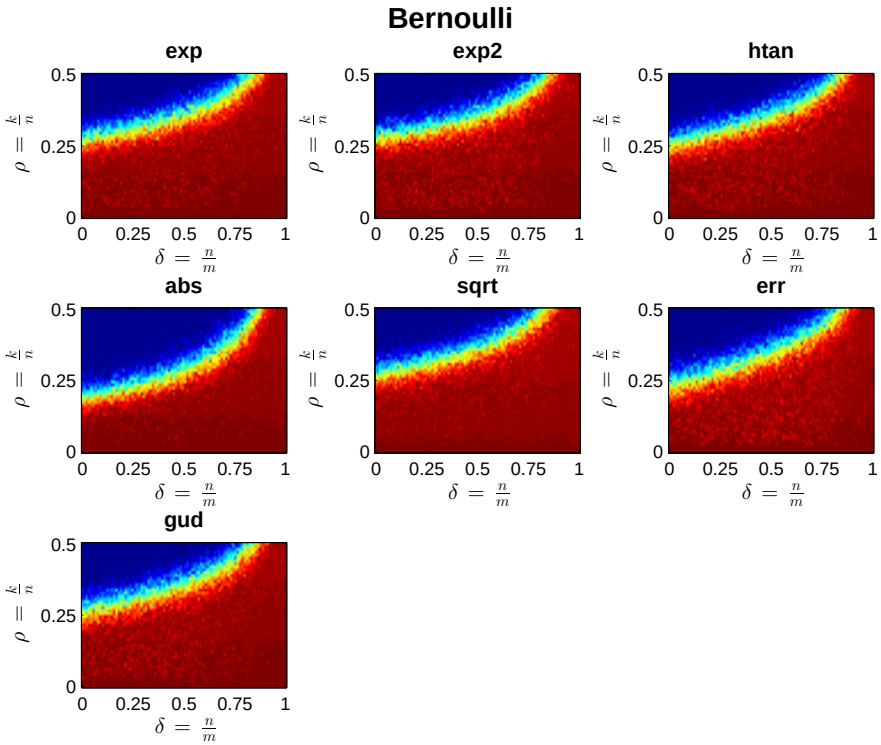

Fig. B.5 Phase transition for Bernoulli matrices.

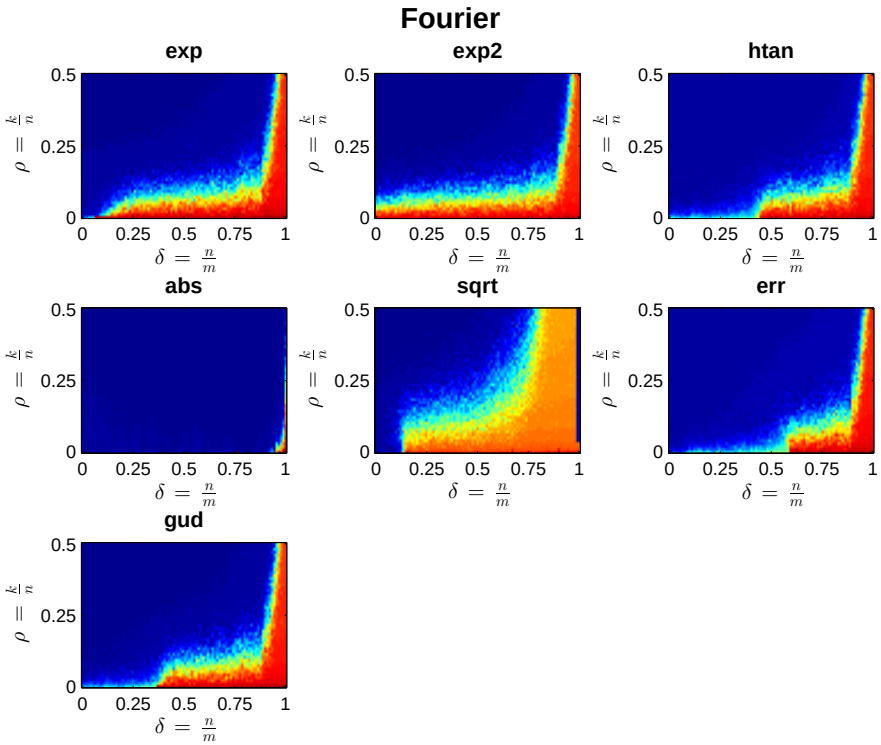

Fig. B.6 Phase transition for Fourier matrices.

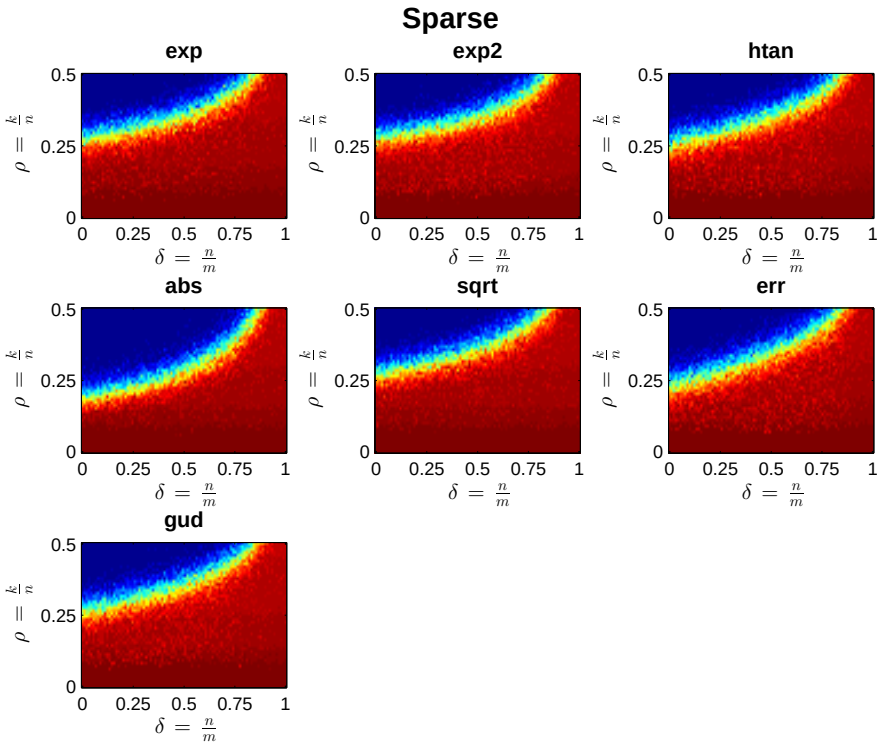

Fig. B.7 Phase transition for Sparse matrices.

References

1. *Wavelets for QRS detection*, volume 2, November 2002.
2. Alessandro Adamo and Giuliano Grossi. A fixed-point iterative schema for error minimization in k-sparse decomposition. pages 167–172, 2011.
3. Alessandro Adamo, Giuliano Grossi, and Raffaella Lanzarotti. Sparse representation based classification for face recognition by k-limaps algorithm. In *ICISP*, pages 245–252, 2012.
4. P. S. Addison. Wavelet transforms and the ECG: a review. *Physiol Meas*, 26(5), October 2005.
5. M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, November 2006.
6. Sabah M. Ahmed, Anwer Al-Shrouf, and Mohammed Abo-Zahhad. ECG data compression using optimal non-orthogonal wavelet transform. *Medical engineering & physics*, 22(1):39–46, 2000.
7. K. Ayarpadi, E. Kannan, R. R. Nair, T. Anitha, R. Srinivasan, and P.G. Scholar. Face Recognition under Expressions and Lighting Variations using Masking and Synthesizing. 2(1):758–763, 2012.
8. Waheed U. Bajwa, A. Robert Calderbank, and Sina Jafarpour. Why gabor frames? two fundamental measures of coherence and their role in model selection. *CoRR*, abs/1006.0719, 2010.
9. S. Banach. *Théorie des opérations linéaires*. Varsovie, 1932.
10. Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae III*, pages 133–181, 1922.
11. BANCA DB. Address: `http://www.ee.surrey.ac.uk/Research/VSSP/banca/`. web.
12. Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A Simple Proof of the Restricted Isometry Property for Random Matrices. *Constructive Approximation*, 28(3):253–263, 2008.
13. Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, March 2009.
14. P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
15. Jeffrey D. Blanchard, Coralia Cartis, and Jared Tanner. Compressed sensing: How sharp is the restricted isometry property? *SIAM Review*, 53(1):105–125, 2011.
16. Jeffrey D. Blanchard, Coralia Cartis, Jared Tanner, and Andrew Thompson. Phase transitions for greedy sparse approximation algorithms. *CoRR*, abs/1004.1821, 2010.
17. Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752–756, 2012.
18. P. Campadelli, R. Lanzarotti, and G. Lipori. Automatic facial feature extraction for face recognition. In Delac and Grgic [37], pages 31–58.
19. P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye and mouth localization. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(3), 2009.
20. E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589–592, may 2008.
21. E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, February 2006.
22. E. J. Candes and T. Tao. Decoding by Linear Programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.
23. E. J. Candes and T. Tao. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, December 2006.

24. E.J. Candès and Y. Plan. Near-ideal model selection by l1 minimization. *The Annals of Statistics*, 37(5A):2145–2177, 2008.
25. Emmanuel Candes, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math*, 59(8):1207–1223, 2005.
26. Emmanuel J. Candes and Justin Romberg. Quantitative Robust Uncertainty Principles and Optimally Sparse Decompositions. *Found. Comput. Math.*, 6(2):227–254, April 2006.
27. Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, August 2006.
28. Rick Chartrand. Exact reconstructions of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14:707–710, 2007.
29. Chen, Scott Shaobing, Donoho, David L., Saunders, and Michael A. Atomic Decomposition by Basis Pursuit. *SIAM Rev.*, 43(1):129–159, jan 2001.
30. S. Chen, H. Chen, and H. Chan. A real-time QRS detection method based on moving-averaging incorporating with wavelet denoising. *Computer Methods and Programs in Biomedicine*, 82(3):187–195, June 2006.
31. Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Rev.*, 43(1):129–159, January 2001.
32. Iulian B. Ciocoiu. ECG signal compression using 2D wavelet foveation. In *Proceedings of the 2009 International Conference on Hybrid Information Technology*, ICHIT '09, pages 576–580, New York, NY, USA, 2009. ACM.
33. Iulian B. Ciocoiu. Ecg signal compression using 2d wavelet foveation. In Geuk Lee, Daniel Howard, Jeong Jin Kang, Dominik Slezak, Tae Nam Ahn, and Chung-Huang Yang, editors, *ICHIT*, volume 321 of *ACM International Conference Proceeding Series*, pages 576–580. ACM, 2009.
34. Gari D. Clifford, Francisco Azuaje, and Patrick McSharry. *Advanced Methods And Tools for ECG Data Analysis*. Artech House, Inc., Norwood, MA, USA, 2006.
35. Geoffrey Davis, Stéphane Mallat, and Zhifeng Zhang. Adaptive Time-Frequency Decompositions with Matching Pursuits. *Optical Engineering*, 33, 1994.
36. Gael de Lannoy, Arnaud de Decker, and Michel Verleysen. A supervised wavelet transform algorithm for r spike detection in noisy ecgs. In *BIOSTEC (Selected Papers)*, pages 256–264, 2008.
37. Kresimir Delac and Mislav Grgic, editors. *Face Recognition*. I-Tech Education and Publishing, 2007.
38. W. Deng, J. Hu, J. Guo, W. Cai, and D. Feng. Emulating biological strategies for uncontrolled face recognition. *Pattern Recognition*, 43(6):2210–2223, 2010.
39. Ronald A. DeVore. Nonlinear approximation. *ACTA NUMERICA*, 7:51–150, 1998.
40. O. Divorra Escoda, L. Granai, M. Lemay, J. Molinero Hernandez, P. Vandergheynst, and J. Vesin. Ventricular and Atrial Activity Estimation Through Sparse ECG Signal Decompositions. In *Proc. of IEEE ICASSP'06*, 2006.
41. David L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.
42. David L. Donoho. High-dimensional centrally-symmetric polytopes with neighborliness proportional to dimension. Technical report, Comput. Geometry, (online) Dec, 2005.
43. David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, March 2003.
44. David L. Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *CoRR*, abs/0906.2530, 2009.
45. David L. Donoho, Yaakov Tsaig, Iddo Drori, and Jean-Luc Starck. Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, 58(2):1094–1121, 2006.

46. S. Du and R.K. Ward. Adaptive Region-Based Image Enhancement Method for Robust Face Recognition Under Variable Illumination Conditions. *IEEE Transactions on Circuits and Systems for video technology*, 20(9):1165–1175, 2010.
47. Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, Mar. 2008.
48. Bradley Efron, Trevor Hastie, Lain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.
49. M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Science+Business Media, LLC, 2010.
50. K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference - Volume 05*, ICASSP '99, pages 2443–2446, Washington, DC, USA, 1999. IEEE Computer Society.
51. Eddie B. L. Filho, Eduardo A. B. da Silva, Waldir S. S. Junior, and Murilo B. de Carvalho. ECG compression using multiscale recurrent patterns with period normalization. In *ISCAS*, 2006.
52. Eddie B. L. Filho, Nuno M. M. Rodrigues, Eduardo A. B. da Silva, Murilo B. de Carvalho, Sérgio M. M. de Faria, and Vítor Manuel Mendes da Silva. On ecg signal compression with 1-d multiscale recurrent patterns allied to preprocessing techniques. *IEEE Trans. Biomed. Engineering*, 56(3):896–900, 2009.
53. Monica Fira and Liviu Goras. Biomedical signal compression based on basis pursuit. ICHIT, 541-545 (2009)., 2009.
54. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.
55. M. Fornasier. *Theoretical Foundations and Numerical Methods for Sparse Recovery*. Radon Series on Computational and Applied Mathematics Series. De Gruyter, 2010.
56. A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.
57. S. Gershgorin. Ueber die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. SSSR Ser. Mat.*, 1:749–754, 1931.
58. A. Ghafari, H. Palangi, Massoud Babaie-Zadeh, and Christian Jutten. ECG denoising and compression by sparse 2D separable transform with overcomplete mixed dictionaries. In *Proceedings of the 19-th international IEEE workshop on Machine Learning for Signal Processing (MLSP 2009)*, page 6 pages, Grenoble, France, September 2009.
59. A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215 PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.
60. R. Gopalan and D. Jacobs. Comparing and combining lighting insensitive approaches for face recognition. *Computer Vision and Image Understanding*, 114(1):135–145, 2010.
61. L. Goras and M. Fira. *Preprocessing method for improving ECG signal classification and compression validation.*, pages 20–25. Hackensack, NJ: World Scientific, 2010.
62. R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325, December 2003.
63. P. S. Hamilton and W. J. Tompkins. Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. *IEEE Trans Biomed Eng*, 33(12):1157–1165, December 1986.
64. X. He, S.Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27:328–340, 2005.
65. D.A. Huffman. A Method for the Construction of Minimum Redundancy Codes. *Proceedings IRE 40*, 40(10):1098–1101, 1952.

66. Mashud Hyder and Kaushik Mahata. An improved smoothed ℓ_0 approximation algorithm for sparse representation. *IEEE Transactions on Signal Processing*, 58(4):2194–2205, 2010.
67. O. Jesorsky, K.J. Kirchberg, and R.W. Frischholz. Robust face detection using the Hausdorff distance. *Lecture Notes in Computer Science*, 2091:212–227, 2001.
68. William Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984.
69. M. Kirby and L. Sirovich. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103–108, jan 1990.
70. M. Koç and A. Barkana. A new solution to one sample problem in face recognition using FLDA. *Applied Mathematics and Computation*, 217(24):10368–10376, 2011.
71. L. Lam and S. Y. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *Trans. Sys. Man Cyber. Part A*, 27(5):553–568, September 1997.
72. Louisa Lam and Ching Y. Suen. Optimal combinations of pattern classifiers. *Pattern Recogn. Lett.*, 16(9):945–954, September 1995.
73. Z Lu, D Y Kim, and W A Pearlman. Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. *IEEE Transactions on Biomedical Engineering*, 47(7):849–856, 2000.
74. T.J. Lynch. *Data compression techniques and applications*. Lifetime Learning Publications, 1985.
75. Boris Mailhé, Rémi Gribonval, Frédéric Bimbot, Mathieu Lemay, Pierre Vandergheynst, and Jean-Marc Vesin. Dictionary learning for the sparse modelling of atrial fibrillation in ecg signals. In *ICASSP*, pages 465–468. IEEE, 2009.
76. Stephane Mallat. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, 2 edition, sep 1999.
77. Stéphane Mallat and Zhifeng Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.
78. Aleix Martínez and Robert Benavente. The ar face database. Technical Report 24, Computer Vision Center, Bellatera, Jun 1998.
79. Aleix M. Martinez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):748–763, 2002.
80. Rui Min, Angela D'angelo, and Jean-Luc Dugelay. Efficient scarf detection prior to face recognition. In *EUSIPCO 2010, 18th European Signal Processing Conference, August 23-27, 2010, Aalborg, Denmark*, Aalborg, DANEMARK, 08 2010.
81. G. Hosein Mohimani, Massoud Babaie-Zadeh, Irina Gorodnitsky, and Christian Jutten. Sparse Recovery using Smoothed ℓ^0 (SL0): Convergence Analysis. *CoRR*, abs/1001.5073, 2010.
82. G. Hosein Mohimani, Massoud Babaie-Zadeh, and Christian Jutten. Fast Sparse Representation Based on Smoothed ℓ_0 Norm. In *ICA*, pages 389–396, 2007.
83. A. Nabatchian, E. Abdel-Raheem, and M. Ahmadi. Illumination invariant feature extraction and mutual-information-based local matching for face recognition under illumination variation and occlusion. *Pattern Recognition*, 44(10-11):2576–2587, 2011.
84. P. Nagesh and B. Li. A compressive sensing approach for expression-invariant face recognition. *Proc. Int'l Conf. on Computer Vision and Pattern Recognition*, pages 1518–1525, 2009.
85. B. K. Natarajan. Sparse Approximate Solutions to Linear Systems. *SIAM J. Comput.*, 24(2):227–234, 1995.
86. D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Technical report, California Institute of Technology, Pasadena, 2008.
87. Tyler Neylon. *Sparse solutions for linear prediction problems*. PhD thesis, New York, NY, USA, 2006. AAI3221982.

88. ORL DB. AT & T laboratories cambridge. web.
89. J. Pan and W. J. Tompkins. A real-time QRS detection algorithm. *IEEE Trans Biomed Eng*, 32(3):230–236, March 1985.
90. Bo-Gun Park, Kyoung-Mu Lee, and Sang-Uk Lee. Face recognition using face-arg matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1982–1988, 2005.
91. Y. C. Pati, R. Rezaiifar, Y. C. Pati R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993.
92. C. Perez, L. Cament, and L. E. Castillo. Methodological improvement on local Gabor face recognition based on feature selection and enhanced Borda count. *Pattern Recognition*, 44(4):951–963, 2011.
93. P.J. Phillips, P.J. Flynn, T. Scruggs, and K.W. Bowyer. Overview of the face recognition grand challenge. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.
94. Luisa F. Polania, Rafael E. Carrillo, Manuel Blanco-Velasco, and Kenneth E. Barner. Compressed sensing based method for ecg compression. In *ICASSP*, pages 761–764. IEEE, 2011.
95. V.K. Pothos, C. Theoharatos, and G. Economou. A local spectral distribution approach to face recognition. *Computer Vision and Image Understanding*, 116(6):663–675, 2012.
96. L. Qiao, S. Chen, and X. Tan. Sparsity preserving discriminant analysis for single training image face recognition. *Pattern Recognition Letters*, 31(5):422–429, 2010.
97. J. Rabia and R.A. Hamid. A survey of face recognition techniques. *Journal of Information Processing Systems*, 5, 2009.
98. H. Rauhut. Compressive sensing and structured random matrices. In M. Fornasier, editor, *Theoretical Foundations and Numerical Methods for Sparse Recovery*, volume 9 of *Radon Series Comp. Appl. Math.*, pages 1–92. deGruyter, 2010.
99. M. Rudelson and R. Vershynin. Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements. In *Annual Conference on Information Sciences and Systems*, pages 207–212, 2006.
100. M. Rudelson and R. Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.
101. Amir Said, William A. Pearlman, and Senior Member. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, 1996.
102. David Salomon. *A Concise Introduction to Data Compression*. Undergraduate topics in computer science. pub-SV, 2008.
103. William Robson Schwartz, Huimin Guo, and Larry S. Davis. A robust and scalable approach to face identification. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 476–489, Berlin, Heidelberg, 2010. Springer-Verlag.
104. W.R. Schwartz, H. Guo, J. Choi, and L.S. Davis. Face identification using large feature sets. *IEEE transactions on image processing*, 21(4):2245–2255, 2012.
105. W.R. Schwartz, Huimin Guo, Jonghyun Choi, and Larry S. Davis. Face identification using large feature sets. *Image Processing, IEEE Transactions on*, 21:2245 – 2255, 2012/04// 2012.
106. J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.
107. A. Shashua and T. Riklin-Raviv. The quotient image: Class-based re- rendering and recognition with varying illuminations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:129–139, 2001.
108. Michael Starbird. R. H. Bing's human and mathematical vitality. In Charles E. Aull and Robert Lowen, editors, *Handbook of the history of general topology*, volume 2 of *History of Topology*, pages 453–466. Kluwer Academic, Dordrecht, 1998. (San Antonio, TX, 1993). MR 1795160. Zbl 0932.01042.
109. Thomas Strohmer and Robert W. Heath. Grassmannian frames with applications to coding and communication. *Applied and Computational Harmonic Analysis*, 14(3):257–275, may 2003.
110. Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

111. Andreas M. Tillmann and Marc E. Pfetsch. The Computational Complexity of RIP, NSP, and Related Concepts in Compressed Sensing. *arXiv*, 2012.
112. A. Tolba, A. El-Baz, and A. El-Harby. Face recognition: A literature review. *Int. J. Signal Process.*, 2:88–103, 2006.
113. Joel A. Tropp, Anna, and C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 53:4655–4666, 2007.
114. Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, jan 1991.
115. M. Turker and A. Pentland. Face recognition using Eigenfaces. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
116. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1:511–518, 2001.
117. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1:511–518, 2001.
118. A. Wagner and J. Wright. Toward a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(2):372–386, 2012.
119. T.A. Welch. A Technique for High-Performance Data Compression. *Computer*, 17:8–19, 1984.
120. Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.
121. Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 19:775–779, 1997.
122. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Yi Ma. Robust Face Recognition via Sparse Representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, feb 2009.
123. John Wright, Allen Y. Yang, Arvind Ganesh, Shankar S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.
124. S. Yan, H. Wang, J. Liu, X. Tang, and T.S. Huang. Misalignment-robust face recognition. *IEEE transactions on image processing*, 19(4):1087–96, 2010.
125. W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM, Computing Surveys*, 35(4):399–458, 2003.
126. Hehong Zou and Ahmed H. Tewfik. Parametrization of compactly supported orthonormal wavelets. *IEEE Transactions on Signal Processing*, 41(3):1428–1431, 1993.

Index