"Sapienza", University of Rome

Mathematical Programming Methods for Minimizing the Zero-norm over Polyhedral Sets

Francesco Rinaldi

Dissertation for the Degree of
Philosophy Doctor
in
Operations Research

Tutor: Prof. Marco Sciandrone

Contents

Acknowledgements

F.R.

Introduction

The problem of finding sparse solutions to linear systems, i.e., solutions having many zero components, has recently received a great deal of attention in the research community. Such a phenomenon can be easily explained with the fact that many challenging real problems consist in searching a sparse solution to a linear system. In machine learning, for instance, the extraction of relevant features from massive datasets (see, e.g., [40]) is often modelled as a search for a sparse vector satisfying some linear inequality constraints. Some relevant problems in signal/image processing such as sparse approximation of signals, image denoising, image deblurring, can be also viewed as finding sparse solutions to underdetermined systems of linear equations (see, e.g., [26]).
In this work the general problem of finding a vector belonging to a polyhedral set P and having the minimum number of nonzero components has been considered. Formally, the problem is

$$\begin{aligned} &\min_{x \in R^n} \|x\|_0 \\ &x \in P \end{aligned} \tag{1}$$

where $\|x\|_0$ is the zero-norm of x defined as $\|x\|_0 = card\{x_i : x_i \neq 0\}$, $P \subset R^n$ is a non empty polyhedral set. This combinatorial optimization problem is NP-Hard as shown by Amaldi and Kann in [3].
In order to make the problem tractable, the simplest approach can be that of replacing the zero-norm, which is a nonconvex discontinuous function, by the ℓ_1 norm thus obtaining a linear programming problem which can be efficiently solved even when the dimension is very large. Under suitable assumptions on the polyhedral set P (defined by an underdetermined linear system of equations) it is possible to prove that a solution of (3.1) can be

obtained by solving the ℓ_1-norm problem (see, e.g., [38]). However, these assumptions may be not satisfied in many cases, and some experiments concerning machine learning problems and reported in [10] show that a concave optimization-based approach performs better than that based on the employment of the ℓ_1 norm.
The nonlinear approach experimented in [10] was originally proposed in [57], and is based on the idea of replacing the zero-norm by a continuously differentiable concave function. The replacement by the smooth concave problem is well-motivated (see [57]) both from a theoretical and a computational point of view:

- under suitable assumptions on the parameters of the concave function it is possible to show that the approximating problem is equivalent to the zero-norm problem;

- the Frank-Wolfe algorithm [34] with unitary stepsize is guaranteed to converge to a vertex stationary point in a finite number of iterations (this convergence result was proved for a general class of concave programming problems); thus the algorithm requires the solution of a finite sequence of linear programs for computing a sparse solution, and this may be quite advantageous from a computational point of view.

A similar concave optimization-based approach has been proposed in [90], where the idea is that of using the logarithm function instead of the step function. This formulation is practically motivated by the fact that, due to the form of the logarithm function, it is better to increase one variable while setting to zero another one rather than doing some compromise between both, and this should facilitate the computation of a sparse solution. The Frank-Wolfe algorithm with unitary stepsize has been applied to find a solution to the concave approximation, and good computational results have been obtained.
The main contributions of this work can be summarized as follows.

- New results on the equivalence between a specific optimization problem and a parameterized family of problems have been stated. By means of this analysis it is possible to derive, within a general framework, results about two previously known families of approximations schemes for the zero-norm problem.

- Two new families of approximation problems have been introduced. Thanks to the general results, the equivalence of these new approximations to the zero norm-problem has been showed.

- Some new theoretical results, which have an important impact on the computational efficiency of the Frank-Wolfe method when applied to concave optimization over polyhedra, have been described. In particular it is possible to prove that once the algorithm sets a variable to zero, it will not change this variable any more. This result suggests the definition of a version of the method that eliminates the variables set to zero, thus allowing for a dimensionality reduction which greatly increments the speed of the procedure. The global convergence of this modified version of the Frank-Wolfe method has been proved.

- Numerical experiments have been performed on test problems, and for finding sparse approximations of noisy signals. The obtained results show both the usefulness of the new concave formulations and the efficiency in terms of computational time of the new minimization algorithm.

- Concerning feature selection problems, which are of great importance in machine learning, a new algorithm has been developed. It combines the concave optimization-based approach (to eliminate irrelevant features) with linear Support Vector Machines (to guarantee predictive capability). An extensive computational experience has been performed on several datasets in order to show the efficiency of the proposed feature selection technique.

The thesis is organized as follows. The first chapter is mainly focused on analyzing known optimization methods for finding sparse approximations of signals. In the second chapter, several feature selection methods of the literature are presented and discussed. The third chapter is concerned with the general problem of minimizing the zero-norm over a polyhedral set, and presents the concave optimization-based contributions of the thesis. In the forth chapter, new methods for feature selection and sparse approximation

of signals are described. A feature selection technique combining concave optimization and linear Support Vector Machines is defined, and computational results about sparse approximations of noisy signals by concave programming are reported. The final chapter describes global optimization approaches for the minimization of concave separable functions over polyhedral sets.

The present thesis has been supported by the *Istituto di Analisi dei Sistemi ed Informatica* (IASI) "Antonio Ruberti", of the Italian National Research Council (CNR).

Notations and definitions

This short section is aimed to briefly introduce some useful notations and definitions that will be used throughout the entire text. A superscript k will generally be used to indicate the iteration number of an algorithm, while the subscript i will generally denote either the i-th component of a vector or the i-th row of a matrix. All vectors will be column vectors unless transposed to a row vector by a superscript T. The identity matrix in a real space of arbitrary dimension will be denoted by I, while a column vector of arbitrary dimension of all ones will be denoted by e. A generic diagonal matrix $D \in \mathbb{R}^{n \times n}$ will be denoted by $D = \mathrm{diag}(x)$ where $x \in R^n$ is the vector of the diagonal entries. For a vector $x \in R^n$, x_+ will denote the vector in R^n with components $(x_+)_i = \max\{x_i, 0\},\ i = 1, \ldots, n$. Similarly, x_* will denote the vector in R^n with components $(x_*)_i = (x_i)_*\ i = 1, \ldots, n$, where $(\cdot)_*$ is the step function defined as one for positive x_i and zero otherwise. The base of the natural logarithm will be denoted by ε and for $y \in R^n$, ε^y will denote a vector in R^n with components $\varepsilon^{y_i},\ i = 1, \ldots, n$. For two vectors x and y, $x \perp y$ will denote $x^T y = 0$. The norm $\|\cdot\|_p$ will denote the p-norm with $1 \leq p \leq \infty$. For a vector $x \in R^n$, $\|x\|_0$ will denote the zero norm of x, which is defined as follows: $\|x\|_0 = card\{x_i : x_i \neq 0\}$. Despite its name, the zero norm is not a true norm.

Chapter 1

Methods for sparse approximation of signals

This chapter is mainly focused on methods for finding sparse approximations of a signal. Sparse approximation problems arise in various fields, such as electrical engineering, applied mathematics and statistics. The goal in sparse approximation is that of approximating a given input signal by means of a linear combination of elementary signals. These elementary signals do usually belong to a large, linearly dependent collection. A preference for linear combinations involving only a few elementary signals is obtained by penalizing nonzero coefficients. A well-known penalty function is the number of elementary signals used in the approximation. Obviously the choice we make about the specified collection, the linear model and the sparsity criterion must be justified by the domain of the problem we deal with.

The first section of this chapter presents the problem of seeking the sparsest representation of a target signal. Some important theoretical results about the uniqueness of the sparsest representation are given, and some technique for finding sparse representations are described.

In the second section a noise-aware version of the problem is taken into account. Due to the presence of the noise, the signal cannot be recovered exactly. Hence the goal is seeking the sparsest representation that achieves a prescribed approximation error. This tough task can be tackled by using various methods. We give a wide overview of these methods throughout the section.

1.1 The Sparsest Exact Representation of a Signal

This section describes the problem of recovering the sparsest exact representation of a signal. Even if in the vast majority of situations it is not possible to construct a signal without error, finding an exact representation is very interesting from a theoretical point of view. In fact, all results regarding general sparse approximation problems somehow derive from results obtained analyzing this problem.
At the beginning of the section we formally describe the problem and give a fundamental result about its complexity. Then we give some conditions that can guarantee the uniqueness of the sparsest representation. Finally we describe convex relaxation methods for recovering a sparse representation of the input signal.

1.1.1 Formal Problem Statement

Consider a real-valued, finite-length, one-dimensional, discrete-time input signal b, which we view as an $m \times 1$ column vector in R^m with elements b_i $i = 1, \ldots, m$, and a *dictionary*

$$D = \{a_j \in R^m \ : \ j = 1, \ldots, n\}$$

of elementary discrete-time signals, usually called atoms, having the property that $\|a_j\|_2 = 1$ for $j = 1, \ldots, n$. We want to represent our signal as a linear combination of the atoms in this dictionary:

$$b = \sum_{j=1}^{n} x_j a_j \ .$$

In many applications the dictionary we deal with is *overcomplete*, which means $m < n$. In this case, the atoms form a linear dependent set and there exists an infinite number of approximations for a given input signal. We are basically interested in representations having as few nonzero coefficients x_j as possible. Then a function $P(x)$ measuring the *sparsity* of a solution x is

needed. The optimization problem we want to solve is

$$\begin{aligned} &\min_{x \in R^n} P(x) \\ &Ax = b \end{aligned} \tag{1.1}$$

with A an $R^{m \times n}$ matrix having as columns a_j the elementary signals of the dictionary D. A good measure of sparsity is the number of nonzero elements of the vector x. Hence, we can use the ℓ_0 quasi-norm and consider the new problem obtained from (1.1) by setting $P(x) = \|x\|_0$; thus we have:

$$\begin{aligned} &\min_{x \in R^n} \|x\|_0 \\ &Ax = b \ . \end{aligned} \tag{1.2}$$

This is a classical NP-Hard problem and was referred to as *minimum weight solution to linear equations* in [36]. Moreover, Amaldi and Kann [3] established that under a likely assumption, the problem is hard to approximate:

Theorem 1.1.1. *Let $DTIME(n^{polylog\ n})$ be the class of problems whose instances of size s can be solved in deterministic time $O(s^{polylog\ s})$, with polylog s any polynomial in log s.*
Assuming $NP \nsubseteq DTIME(n^{polylog\ n})$, problem (1.2) is not approximable within a factor of $2^{log^{1-\epsilon} n}$ for any $\epsilon > 0$, where n is the number of variables.

Equivalently, we can say that there is no polynomial time algorithm that computes a solution having a support at most $2^{log^{1-\epsilon} n}$ times larger than the support of an optimal solution, for any $\epsilon > 0$. $NP \nsubseteq DTIME(n^{polylog\ n})$ is stronger than $P \neq NP$, but it is a reasonable assumption anyway; it means that not all problems in NP can be solved in polynomial time.

1.1.2 Uniqueness of the Sparsest Representation

There exist conditions under which it is possible to guarantee the uniqueness of the optimal solution for problem (1.2). A fundamental concept for the study of uniqueness is the *spark*, a term originally introduced in [25]. It is defined as follows:

Definition 1.1.1. *The spark of a dictionary A is the smallest number of columns that form a linearly dependent set.*

By using the *spark*, we can give the first simple criterion for ensuring that the sparsest representation of a given input signal is unique:

Theorem 1.1.2. [25] *Let us consider an input signal $b \in R^m$ and a dictionary $A \in R^{m \times n}$. If there exists a solution $\tilde{x}$ of problem (1.2) such that*

$$\|\tilde{x}\|_0 < spark(A)/2 \ , \tag{1.3}$$

then $\tilde{x}$ is the unique sparsest representation of b.

Proof. Consider a different representation $\hat{x}$ satisfying the system

$$A\hat{x} = b \ .$$

We have that $A(\tilde{x} - \hat{x}) = 0$ and the vector $\tilde{x} - \hat{x}$ is in the null-space of A. By using definition of spark:

$$\|\tilde{x}\|_0 + \|\hat{x}\|_0 \geq \|\tilde{x} - \hat{x}\|_0 \geq spark(A) \ .$$

From condition (1.3) we have that an alternative representation must satisfy the following inequality:

$$\|\hat{x}\|_0 > spark(A)/2 \ .$$

Hence, we conclude that solution $\tilde{x}$ is necessarily the unique global optima of problem (1.2). □

Here, we find that global optimality can be checked simply by comparing the solution sparsity and the spark of matrix A. Clearly, calculating $spark(A)$ is a very tough task as a combinatorial search over all possible subsets of columns from A is required. Thus we need a simpler criterion to ensure uniqueness. The concept of *mutual coherence*, introduced in [56, 24, 25], can be used to obtain a new condition:

Definition 1.1.2. *The mutual coherence of a dictionary A, denoted by $\mu(A)$, is defined as the maximal absolute scalar product between two different atoms of A.*

$$\mu(A) = \max_{1 \leq j,k \leq n, \ j \neq k} |a_j^T a_k| \ . \tag{1.4}$$

The mutual coherence of a dictionary measures the similarity between the dictionarys atoms. For an orthogonal matrix A, $\mu(A) = 0$. For an over-complete matrix ($m < n$) we necessarily have $\mu(A) > 0$. If $\mu(A) = 1$, it implies the existence of two parallel atoms, and this causes confusion in the construction of sparse atom compositions.

Lemma 1.1.1. [25] *Given a dictionary $A \in R^{m\times n}$, the following relationship holds:*

$$spark(A) \geq 1 + \mu(A)^{-1} \ .$$

By using *mutual coherence* we attain the following theorem:

Theorem 1.1.3. [25] *Let us consider an input signal $b \in R^m$ and a dictionary $A \in R^{m\times n}$. If there exists a solution $\tilde{x}$ of problem (1.2) such that*

$$\|\tilde{x}\|_0 < (1 + \mu(A)^{-1})/2 \ , \tag{1.5}$$

then $\tilde{x}$ is the unique sparsest representation of b.

We notice that Theorem 1.1.3 is less powerful than Theorem 1.1.2 as it uses the *mutual coherence*, which represents a lower bound of *spark*.

1.1.3 ℓ_1 norm-based Relaxation Method

A good approach to solve the combinatorial problem we deal with is that of replacing it with a relaxed version that can be solved more efficiently. The ℓ_1 norm represents, in some sense, the best convex approximant of the ℓ_0 norm. Then using the ℓ_1 norm in place of the ℓ_0 norm is a natural strategy to obtain a convex problem we can easily handle. This is the well-known *Basis Pursuit Method* proprosed by Chen, Donoho and Saunders in [17]. The new convexified problem is:

$$\begin{aligned} &\min_{x\in R^n} \ \|x\|_1 \\ &Ax = b \ . \end{aligned} \tag{1.6}$$

It can be expressed as a linear programming problem and solved by means of modern interior-point methods or simplex methods.
Under some conditions it is possible to show the equivalence between the original problem (1.2) and the convexified problem (1.6) [25]:

Theorem 1.1.4. *Let us consider an input signal $b \in R^m$ and a dictionary $A \in R^{m \times n}$. If there exists a solution $\tilde{x}$ of problem (1.2) such that*

$$\|\tilde{x}\|_0 < (1 + \mu(A)^{-1})/2 \ , \tag{1.7}$$

then $\tilde{x}$ is both the unique solution of (1.2) and the unique solution of (1.6).

Proof. We define the following set of alternative approximations having larger support and an ℓ_1 norm value as good as the optimal solution:

$$S = \{ \ \hat{x} : \ \hat{x} \neq \tilde{x}, \ \|\hat{x}\|_1 \leq \|\tilde{x}\|_1, \ \|\hat{x}\|_0 > \|\tilde{x}\|_0 \ \text{ and } A\hat{x} = b \ \} \ .$$

By Theorem 1.1.3, we know that $\tilde{x}$ is the unique sparsest solution, then alternative solutions have more nonzero components. Thus, we can omit the expression $\|\hat{x}\|_0 > \|\tilde{x}\|_0$ from definition of C and, by using the new variable $d = \hat{x} - \tilde{x}$, we rewrite S as follows:

$$S_C = \{ \ d : \ d \neq 0, \ \|d + \tilde{x}\|_1 - \|\tilde{x}\|_1 \leq 0, \ \text{and } Ad = 0 \ \} \ .$$

We assume, without loss of generality, that the first k_0 components are nonzero. Then we can rewrite the expression $\|d+\tilde{x}\|_1 - \|\tilde{x}\|_1 \leq 0$ as follows:

$$\|d + \tilde{x}\|_1 - \|\tilde{x}\|_1 = \sum_{j=1}^{k_0} (|d_j + x_j| - |x_j|) + \sum_{j=k_0+1}^{n} |d_j| \leq 0 \ .$$

Using the inequality $|a + b| - |b| \geq -|a|$, we can relax our condition:

$$-\sum_{j=1}^{k_0} |d_j| + \sum_{j=k_0+1}^{n} |d_j| \leq \sum_{j=1}^{k_0} (|d_j + x_j| - |x_j|) + \sum_{j=k_0+1}^{n} |d_j| \leq 0 \ .$$

By adding and subtracting the term $\sum_{j=1}^{k_0} |d_j|$, we have:

$$S_C \subseteq \{ \ d : \ d \neq 0, \ \|d\|_1 - 2\sum_{j=1}^{k_0} |d_j| \leq 0, \ \text{and } Ad = 0 \ \} = S_C^1 \ .$$

Let us now consider the system $Ad = 0$ and rewrite it as follows:

$$-d = A^T Ad - d \ .$$

Taking the entry-wise absolute value, we can consider the following relaxation:

$$|d| = |(A^T A - I)d| \leq |A^T A - I| \cdot |d| \leq \mu(A)(\mathbf{1} - I) \cdot |d|$$

with $\mathbf{1}$ a rank-1 matrix having all components equal to 1. We can write the following relaxed set:

$$S_C^1 \subseteq \{\ d:\ d \neq 0,\ \|d\|_1 - 2\sum_{j=1}^{k_0} |d_j| \leq 0,\ \text{and}\ |d| \leq \frac{\mu(A)}{1+\mu(A)}\mathbf{1} \cdot |d|\ \} = S_C^2\ .$$

We can now restrict our quest for normalized vectors $\|d\|_1 = 1$ and consider the following set:

$$S_C^3 = \{\ d:\ d \neq 0,\ 1 - 2\sum_{j=1}^{k_0} |d_j| \leq 0,\ \text{and}\ |d| \leq \frac{\mu(A)}{1+\mu(A)} e\ \}$$

with $e^T = (1, \ldots, 1)$. A vector d belongs to S_C^3 if the following relations are satisfied:

$$1 - 2k_0\frac{\mu(A)}{1+\mu(A)} \leq 1 - 2\sum_{j=1}^{k_0} |d_j| \leq 0\ .$$

This means that if $k_0 < (1 + \mu(A)^{-1})/2$ the set will be necessarily empty and the unique solution of (1.2) is also the unique solution of (1.6). □

To summarize, if we solve problem (1.6) and find out that it has a sufficiently sparse solution, we know we have obtained the solution to problem (1.2) as well. Equivalently, if the input signal has a sparse enough representation, we can find it by solving problem (1.6).

1.1.4 A Basic Uncertainty Principle

The general results showed in the previous sections were first described for the special case when matrix A is the concatenation of two different orthogonal matrices (i.e. the identity matrix I and the Fourier matrix F). We have that

$$A = [I\ F]$$

and the fact that the system $Ax = b$ is underdetermineded simply means that there are different ways of representing the input signal b as a combination of columns from the identity matrix and of columns from the Fourier matrix. The first result obtained for this kind of matrices is similar to the one given in the previous section and was interpreted as a basic uncertainty principle. In fact, if we think of A as a time-frequency system, by the uniqueness of sparse representation, we have that a signal cannot be sparsely represented both in time and frequency.
A basic uncertainty principle concerning pairs of representations of a given vector b by means of two orthonormal bases Φ and Ψ , see [23, 24, 27], can be stated as follows:

Theorem 1.1.5. *Given a vector $b \in R^m$ and two orthonormal bases Φ and Ψ, b may be represented as*

$$b = \Phi x = \Psi y \ .$$

For this pair of representations we have the inequality:

$$\|x\|_0 + \|y\|_0 \geq \frac{2}{\mu(A)} \ . \tag{1.8}$$

When mutual coherence of the bases Φ and Ψ is small, representations x and y cannot be both very sparse. As a consequence of the theorem stated above, if there exists a sparse representation in terms of a dictionary $A = [\Phi \ \Psi]$, it must be necessarily unique. So the uncertainty principle provide us a bound on sparsity which ensures the uniqueness of such representation:

Theorem 1.1.6. *Given a vector $b \in R^m$ and a dictionary $A = [\Phi \ \Psi]$, if b is to be represented by using A, for any two feasible representations $x_1, \ x_2 \in R^{2m}$, we have*

$$\|x_1\|_0 + \|x_2\|_0 \geq \frac{2}{\mu(A)} \ . \tag{1.9}$$

Then for any given representation $\bar{x}$ the uniqueness is ensured by

$$\|\bar{x}\|_0 < \frac{1}{\mu(A)} \ .$$

We notice that, due to the special structure of A, a stronger uniqueness result than the one in Theorem 1.1.3 has been obtained.
Donoho and Huo proved in [24] that for a dictionary $A = [\Phi\ \Psi]$ under the stronger sparsity condition

$$\|x\|_0 < (1 + \mu(A)^{-1})/2$$

solving problem (1.6) is equivalent to solve problem (1.2). This is the same result obtained in Theorem 1.1.4 when the matrix A is a general dictionary.
In [27] Elad and Bruckstein improved the bound necessary for equivalence between problem (1.6) and (1.2):

Theorem 1.1.7. *Given a vector $b \in R^m$ and a dictionary $A = [\Phi\ \Psi]$, if there exists a sparse representation $b = A\tilde{x}$ such that*

$$\|\tilde{x}\|_0 < \frac{\sqrt{2} - 0.5}{\mu(A)}\ , \tag{1.10}$$

Then solution of problem (1.2) coincides with solution of problem (1.6).

Let us consider the case $\mu(A) = 1/\sqrt{m}$. As m goes to infinity the ratio between the two bounds becomes:

$$\frac{(\sqrt{2} - 0.5) \cdot \mu(A)^{-1}}{0.5\ (1 + \mu(A)^{-1})} = \frac{(\sqrt{2} - 0.5) \cdot \sqrt{m}}{0.5\ (1 + \sqrt{m})}\ \rightarrow\ 2\sqrt{2} - 1 = 1.8284\ .$$

Then the result obtained in Theorem 1.1.7 is better than the general result in Theorem 1.1.4 by a factor of almost 2.

1.2 Error-Constrained Approximation

In most practical situations it is not sensible to assume that the target signal can be exactly reconstructed by using the collection of elementary signals. Then a noise-aware variant of the problem described in the previous section must be considered. The goal is finding the sparsest representation that achieves a prescribed error.
At the beginning of the section we introduce the problem and show its NP-hardness. Then we give some bounds ensuring stability of the sparse

representation (stability replaces uniqueness in the noisy case).
In the third part we present the convex relaxation methods and analyze various computational approaches to attain a solution. Finally we describe other methods for seeking a sparse solution in the presence of noise, mainly focusing on the concave programming approach.

1.2.1 Sparse Representation in the Presence of Noise

We can now consider a generalized version of problem (1.2). Instead of finding the sparsest exact representation of a signal, we want to find the sparsest representation having a prescribed approximation error. This type of challenge is very common in numerical analysis, where a typical problem is that of approximating a complicated function by means of a short linear combination of more elementary functions, with an error committed which must be lower than a fixed bound. The exact constraint Ax=b is relaxed with an approximate equality measured using the quadratic penalty function

$$P(x) = \|Ax - b\|_2. \tag{1.11}$$

To state the problem formally, we consider an input signal $b \in R^m$ and fix an error tolerance δ. The problem to be solved is the following:

$$\begin{aligned} &\min_{x \in R^n} \|x\|_0 \\ &\|Ax - b\|_2 \leq \delta \ . \end{aligned} \tag{1.12}$$

The ℓ_2 norm used for evaluating the error $Ax - b$ can be replaced by other options, such as:

- the ℓ_1 norm;
- the ℓ_∞ norm;
- a weighted version of the ℓ_2 norm.

It is easy to see that problem (1.12) must have solutions at least as sparse as those obtained by solving (1.2).
Problem (1.12) can be viewed as a noise-aware version of problem (1.2). Consider a sparse vector x_0 and assume that

$$b = Ax_0 + z \ ,$$

where z is a noise term which is either stochastic or deterministic, such that $\|z\|_2 \leq \epsilon$. Finding x_0 is the aim of (1.12), which is the same as (1.2) when dealing with noiseless data $Ax_0 = b$.
Since problem (1.12) contains (1.2) as a special case, it is Np-Hard as well. Natarajan [65] presents an NP-Hardness proof for $\delta > 0$. The proof is by reduction from the Np-Hard problem of *exact cover by 3 sets*. Problem (1.12) is also hard to approximate as shown by Amaldi in [4].

1.2.2 Stability of Sparse Representations

Results obtained in the noisy case are similar to those obtained in the noiseless one, although uniqueness and equivalence no longer apply and are replaced by the notion of *stability*. An algorithm is said to be *globally stable* if it recovers the ideal noiseless reconstruction with an error at worst proportional to noise level even under the addition of arbitrary noise. Some rigorous bounds ensuring stability can be derived when dictionary A has a property of *mutual incoherence* and when it gives a sparse representation for the ideal noiseless signal. In practice, even if the problem of recovering the underlying representation is ill-posed in general, when the representation is sparse and the dictionary is incoherent, the ill-posedness can disappear.
Consider noisy observations $b = Ax_0 + z$ with $\|b - Ax_0\|_2 \leq \epsilon$. Obtain a sparse approximation x_0^δ by solving (1.12) with $\delta \geq \epsilon$. The following stability estimate, see [26], can be derived:

Theorem 1.2.1. *Consider the problem (1.12). Suppose that a sparse vector $x_0 \in R^n$ such that $\|b - Ax_0\|_2 \leq \epsilon$ satisfies*

$$\|x_0\|_0 < (1 + \mu(A)^{-1})/2 \ . \tag{1.13}$$

Then

a) x_0 is the unique sparsest representation;

b) the deviation of x_0^δ from x_0 is bounded by

$$\|x_0^\delta - x_0\|_2^2 \leq \frac{(\epsilon + \delta)^2}{1 - \mu(A)(2\|x_0\|_0 - 1)} \quad \forall\ \delta \geq \epsilon > 0 \ . \tag{1.14}$$

Equivalently, if there exists a sparse representation and the noise level is known, by solving problem (1.12) we get an approximation to the ideal

sparse decomposition of the noiseless signal in which the error is at worst proportional to the input noise level.

1.2.3 Convex Relaxation by the ℓ_1-norm

As finding a solution to problem (1.12) is a tough task, a pursuit algorithm similar to the one we considered in the noiseless case can be used in order to solve it. The noise-aware variant of the ℓ_1-norm problem is

$$\begin{aligned} &\min_{x \in R^n} \|x\|_1 \\ &\|Ax - b\|_2 \leq \delta \ . \end{aligned} \tag{1.15}$$

A solution to this convex quadratic problem can be easily calculated by using various standard approaches such as interior point methods [17] or active set methods. It is also close to the well-known LASSO technique employed in statistic regression [84]. All this methods replace problem (1.15) with the corresponding convex unconstrained version in the standard Lagrange form:

$$\min_{x \in R^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1 \ . \tag{1.16}$$

For an appropriate multiplier λ problem (1.15) and (1.16) have the same solution. This result can be proved using [73] Theorem 27.4.

1.2.4 ℓ_1-norm Algorithm: Stability and Support Properties

An input signal $b = Ax_0 + z$ with $\|b - Ax_0\|_2 \leq \epsilon$ is given. We solve (1.15) with $\delta \geq \epsilon$ and obtain a solution x_1^δ. The stability result obtained for the ℓ_1-norm is weaker than the one obtained for problem (1.12). We have that the optimal solution x_0 must be sparser and the tolerated error level larger:

Theorem 1.2.2. [26] *Consider the problem (1.12). Suppose that a sparse vector $x_0 \in R^n$ such that $\|b - Ax_0\|_2 \leq \epsilon$ satisfies*

$$\|x_0\|_0 < (1 + \mu(A)^{-1})/4 \ . \tag{1.17}$$

Then the deviation of x_1^δ from x_0 is bounded by

$$\|x_1^\delta - x_0\|_2^2 \leq \frac{(\epsilon + \delta)^2}{1 - \mu(A)(4\|x_0\|_0 - 1)} \quad \forall\ \delta \geq \epsilon > 0 \ . \tag{1.18}$$

We can equivalently say that the solution of the ℓ_1-norm problem has an error at worst proportional to the noise. The optimal solution x_0 for the ℓ_1-norm must be twice sparser than the one for the ℓ_0-norm.
Under some appropriate conditions, the ℓ_1-norm has the ability of recovering the correct sparsity pattern. Then the solution of (1.15) is not only as sparse as the ideal representation, but it only contains atoms belonging to the ideal sparse representation:

Theorem 1.2.3. [26] *Consider a sparse vector $x_0 \leq N$ such that*

$$\|b - Ax_0\|_2 \leq \epsilon \text{ and } \beta = \mu(A) \cdot N < 1/2 \ .$$

Define

$$\gamma = \frac{\sqrt{(1-\beta)} + (1-\beta)/\sqrt{N}}{1-2\beta} \ . \tag{1.19}$$

Solve the problem (1.12) with $\delta > \gamma \cdot \sqrt{N} \cdot \epsilon$. Then $supp(x_1^\delta) \subset supp(x_0)$.

1.2.5 Interior point Approach

Basis Pursuit in highly overcomplete dictionary leads to large-scale optimization problems. Interior point methods can be used to solve this kind of problems efficiently. An approach combining a primal-dual logarithmic barrier method, with a conjugate gradient solver was proposed by Chen, Donoho and Saunders in [17]. Problem (1.16) is rewritten as follows:

$$\begin{aligned} &\min c^T y + \tfrac{1}{2}\|p\|_2^2 \ , \\ &Hy + p = b \ , \quad y \geq 0 \ . \end{aligned} \tag{1.20}$$

with $H = [A \ -A]$, $c^T = \lambda(e^T \ e^T)$ and $y^T = (u^T \ v^T)$. In place of (1.20), the following perturbed linear program is solved by using a primal-dual log-barrier algorithm:

$$\begin{aligned} &\min c^T y + \tfrac{1}{2}\|p\|_2^2 + \tfrac{1}{2}\|\gamma y\|_2^2 \ , \\ &Hy + p = b \ , \quad y \geq 0 \ . \end{aligned} \tag{1.21}$$

The algorithm starts from a feasible (or nearly feasible) solution located near the "center" of the feasible region and improves the current solution

until the desired accuracy is achieved. The number of iterations required is small (usually a few dozen) and at each iteration a system of equations must be solved. Solving a system of equations is not an easy task in general: it takes order $O(n^3)$ time. In order to overcome this problem, dictionaries having *fast implicit algorithms* are considered. For this kind of algorithms special properties of the matrix accelerate the computation (i.e. Ax can be calculated without storing matrix A). When dealing with dictionary having fast implicit algorithms, a natural way to solve equations is by conjugate-gradient methods.

1.2.6 LASSO Algorithm

LASSO (Least Absolute Shrinkage and Selection Operator) is a widely used method for shrinkage and variable selection in linear models [84]. It achieves good prediction accuracy and, at the same time, gives a sparse solution. The main idea of LASSO is that of using the ℓ_1-norm constraint in the regularization step. In other words, the estimator is obtained by minimizing the empirical risk with the ℓ_1-norm of the regression coefficients bounded by a given positive number. Consider observations (x_i, y_i) with $i = 1, \ldots, m$. A standard multiple linear regression is given by:

$$y_i = \alpha + \beta_1 x_{i1} + \cdots + \beta_n x_{in} + \epsilon_i$$

with ϵ_i zero-mean random quantities. The LASSO estimate $(\hat{\alpha}, \hat{\beta})$ is defined by

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\ \beta} \sum_{i=1}^{m} (y_i - \alpha - \sum_{j=1}^{n} \beta_j x_{ij})^2 \quad s.t. \quad \|\beta\|_1 \leq \tau \ . \tag{1.22}$$

The parameter τ controls the amount of shrinkage that is applied to the model. Once parameter τ is fixed, problem (1.22) can be expressed as a constrained least squares problem with 2^n inequality constraints, corresponding to the 2^n possible signs of parameters β_j. Consider the set vectors δ_i with $i = 1, \ldots, 2^n$ having components $\delta_{ij} \in \{-1, 1\} \quad j = 1, \ldots, n$. Then constraint $\|\beta\|_1 \leq \tau$ can be equivalently rewritten as

$$\delta_i^T x \leq \tau \quad i = 1, \ldots, 2^n \ .$$

Hence problem (1.22) can be solved by introducing constraints $\delta_i^T x \leq \tau$ one at time. Here is the outline of the algorithm:

LASSO Algorithm

Initialization. Set $E = \{i_0\}$, where $\delta_{i_0} = sign(\hat{\beta}^0)$, $(\hat{\alpha}^0, \hat{\beta}^0)$ being the overall least square estimate, and $k = 1$.

1. Find the solution

$$(\hat{\alpha}^k, \hat{\beta}^k) = \arg\min_{\alpha,\ \beta} \sum_{i=1}^{m} (y_i - \alpha - \sum_{j=1}^{n} \beta_j x_{ij})^2$$
$$s.t. \quad \delta_i^T \beta \leq \tau \quad i \in E \ . \tag{1.23}$$

2. If $(\|\hat{\beta}^k\|_1 \leq \tau)$ then STOP.

3. Add i_k to the set E where $\delta_{i_k} = sign(\hat{\beta}^k)$.

4. Set $k = k + 1$ and go to step **1**.

As one element is added at each step, and the total number of elements is 2^n, the procedure converges in a finite number of steps. A different approach is that of writing each β_j as follows

$$\beta_j = \beta_j^+ - \beta_j^- \ ,$$

with β_j^+ and β_j^- non-negative variables.
Then the problem to be solved becomes

$$\begin{aligned} &\min_{\alpha,\ \beta} \sum_{i=1}^{m} (y_i - \alpha - \sum_{j=1}^{n} (\beta_j^+ - \beta_j^-)\ x_{ij})^2 \\ &s.t. \\ &\sum_{j=1}^{n} \beta_j^+ + \beta_j^- \leq \tau \\ &\beta_j^+ \geq 0 \quad \beta_j^- \geq 0 \ . \end{aligned} \tag{1.24}$$

The number of variables of this problem is twice as large as the number of variables of the original problem, but the number of constraints remains the same. The new problem (1.24) gives the same solution as the original one

and can be solved by standard quadratic techniques, with the convergence assured in $2n + 1$ steps.

1.2.7 Iterative Reweighted Least Squares

An Iteratively Reweighted Least Squares approach can be used to tackle problem (1.12) as shown in [47, 71]. Setting $X = diag(|x|)$, we have

$$\|x\|_1 = x^T X^{-1} x.$$

Thus, we can define an adaptively-weighted version of the squared ℓ_2-norm. Given the current solution x^k set $X^k = diag(|x^k|)$, we solve the following quadratic problem:

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda x^T X^{-1} x$$

Once we obtain a new solution x^{k+1}, the matrix X^{k+1} is built and a new iteration begins.

1.2.8 Iterative Shrinkage/Thresholding Methods

This class of methods has recently received considerable attention (See [32, 19, 21, 85]). The problem we want to solve, which generalizes problem (1.16), is as follows:

$$\min_x \phi(x) = f(x) + \tau g(x) \tag{1.25}$$

where $f : R^n \to R$ is smooth and convex and $g : R^n \to R$, usually called *regularization term*, is finite for all $x \in R^n$, but not necessarily smooth nor convex. It is usually assumed that g is *separable*:

$$g(x) = \sum_{i=1}^{n} g(x_i).$$

The approach generates a sequence of iterates x^k by solving separable subproblems of the following form:

$$x^{k+1} \in \arg\min_z (z - x^k)^T \nabla f(x^k) + \frac{\alpha_k}{2}\|z - x^k\|_2^2 + \tau g(z). \tag{1.26}$$

Different variants of the approach are distinguished by the choice of α_k or by the way the subproblems are solved.

1.2.9 A Concave Approximation of the ℓ_0-norm

A different way to find a sparse representation in the presence of noise is described by Bradley, Mangasarian and Rosen in [11]. This method is mainly based on the minimization of a concave function on a polyhedral set. The problem can be formally described as follows

$$\min_{x \in R^n} (1-\lambda)\|Ax-b\|_1 + \lambda\|x\|_0 \quad \lambda \in [0,1) \ . \tag{1.27}$$

As the number of nonzero components of the vector x and the error $\|Ax-b\|_1$ have to be minimized, the problem (1.27) can be considered as a multi-objective optimization problem. It can be easily noticed that when $\lambda = 0$ the problem is the classical ℓ_1-norm approximation problem. The case $\lambda = 1$ is of no interest as it leads to the trivial solution $x = 0$. When λ ranges in the interval $[0,1)$ the number of nonzero components can vary from n to 0, while the error $\|Ax-b\|_1$ is monotonically nondecreasing. Problem (1.27) can be equivalently rewritten as

$$\min_{(x,y) \in S} (1-\lambda)e^T y + \lambda\|x\|_0 \quad \lambda \in [0,1) \ , \tag{1.28}$$

with the set S defined as

$$S = \{(x,y) \mid x \in R^n, \ y \in R^m, \ -y \leq Ax-b \leq y\} \ . \tag{1.29}$$

In order to illustrate the idea underlying the concave approach, we observe that the ℓ_0-norm can be written as follows

$$\|x\|_0 = \sum_{j=1}^{n} s(|x_i|)$$

where $s : R \rightarrow R^+$ is the *step function* such that $s(t) = 1$ for $t > 0$ and $s(t) = 0$ for $t \leq 0$. Then the discontinuous step function is replaced by a continuously differentiable concave function $v(t) = 1 - \varepsilon^{-\alpha t}$, with $\alpha > 0$. Thus we have the smooth function

$$c(x) = \sum_{j=1}^{n} (1 - \varepsilon^{-\alpha|x_j|}) = e^T(e - \varepsilon^{-\alpha|x|}) \ , \tag{1.30}$$

satisfying the following relation

$$\|x\|_0 \geq c(x) \quad \forall \ x \in R^n \tag{1.31}$$

and such that

$$\lim_{\alpha\to\infty} c(x) = \|x\|_0 \ . \tag{1.32}$$

Hence a smooth approximation of (1.28) is obtained:

$$\min_{(x,y,z)\in T} (1-\lambda)e^T y + \lambda \sum_{j=1}^{n} (1-\varepsilon^{-\alpha z_j}) \quad \lambda \in [0,1) \ , \tag{1.33}$$

with the set T defined as

$$T = \{(x,y,z) \mid x,z \in R^n,\ y \in R^m,\ -y \le Ax-b \le y,\ -z \le x \le z\} \ . \tag{1.34}$$

1.2.10 Minimization of a Concave Function over a Closed Convex Set

We report here some important results about the minimization of a concave function over a closed convex set (See [73] for further details):

Proposition 1.2.1. *Let f be a concave function, and let C be a closed convex set contained in dom f. Suppose there are no half-lines in C on which f is unbounded below. Then:*

$$inf\ \{f(x) \mid x\ \in\ C\}\ =\ inf\ \{f(x) \mid x\ \in\ E\}, \tag{1.35}$$

where E is the subset of C consisting of the extreme points of $C \cap L^{\perp}$, being L the lineality space of C and $L^{\perp}$ the orthogonal complement of L. The infimum relative to C is attained only when the infimum relative to E is attained.

The following results are an immediate consequence of Proposition 1.2.1:

Corollary 1.2.1. *Let f be a concave function, and let C be a closed convex set contained in dom f. Suppose that C contains no lines. Then, if the infimum of f relative to C is attained at all, it is attained at some extreme points of C.*

Corollary 1.2.2. *Let f be a concave function, and let C be a nonempty polyhedral convex set contained in dom f. Suppose there are no half-lines in C on which f is unbounded below. Then the infimum of f relative to C is attained.*

Combining corollary (1.2.1) and (1.2.2) we can show that the problem (1.36) has a solution and this solution is a vertex:

Corollary 1.2.3. *Let f be a concave function, and let C be a nonempty polyhedral convex set contained in dom f. Suppose that C contains no lines, and that f is bounded below on C. Then the infimum of f relative to C is attained at one of the (finitely many) extreme points of C.*

1.2.11 Existence of an Exact Vertex Solution

In order to show some theoretical results, a general minimization problem is now considered:

$$\min_{s\in S} f(s) + \mu\|s\|_0, \tag{1.36}$$

where f is a concave function on R^p bounded below on S, μ is a nonnegative real number and S is a polyhedral set in R^k containing no lines that go to infinity in both directions. The smooth concave approximation obtained using (1.30) is as follows

$$\min_{s\in S} f(s) + \mu e^T(e - \varepsilon^{-\alpha|s|}) \ . \tag{1.37}$$

By taking into account (1.31), we notice that the minimum of the smooth problem (1.37) provides an underestimate to the minimum of (1.36). In order to prove the next theorem, a result about extreme points of a closed convex set is given:

Lemma 1.2.1. *Let $T \in R^{p_1} \times R^{p_2}$ be a convex closed set. Let $(\hat{x}\ \hat{y})$ be an extreme point of T. Then the vector $\hat{x}$ is an extreme point of the convex set*

$$T(\hat{y}) = \{x \mid (x\ \hat{y}) \in T\} \ . \tag{1.38}$$

Proof. A vector $(\hat{x}\ \hat{y}) \in T$ is an extreme point of T if and only if there is no way to express it as a convex combination of two different points in T:

$$(\hat{x}\ \hat{y}) = \Big((1-\lambda)x_1 + \lambda x_2 \quad (1-\lambda)y_1 + \lambda y_2\Big) \ , \tag{1.39}$$

with $(x_1\ y_1), (x_2\ y_2) \in T$ and $0 < \lambda < 1$, implies

$$(x_1\ y_1) = (x_2\ y_2) = (\hat{x}\ \hat{y}) \ .$$

By choosing $\hat{y} = y_1 = y_2$ in (1.39), it is easy to see that $\hat{x}$ is an extreme point of $T(\hat{y})$. □

The following theorem, proved by Bradley, Mangasarian and Rosen in their paper, shows that there exists a finite value of the smoothing parameter α such that the vertex solution of problem (1.36) is also a solution of problem (1.37):

Theorem 1.2.4. *Let f be a concave function bounded below on the polyhedral set S that contains no lines going to infinity in both directions. There exists a value $\alpha(\mu) > 0$ such that for all $\alpha \geq \alpha_0(\mu) > 0$ the smooth problem (1.37) has a vertex solution that is also a solution of the original problem (1.36).*

Proof. It is easy to notice that the problem (1.37) is equivalent to the following concave minimization problem

$$\min_{(s\ z)\in T} f(s) + \mu e^T(e - \varepsilon^{-\alpha z}) \tag{1.40}$$

with

$$T = \{(s\ z) |\ s \in S,\ -z \leq s \leq z\}\ .$$

The objective function of this problem is concave and bounded below on T. It follows by Corollary 1.2.3 that problem (1.40) has a vertex solution $(s(\alpha)\ z(\alpha))$ for each $\alpha > 0$. Since T has a finite number of vertices, there exists a vertex $(\hat{s}\ \hat{z})$ that is solution of (1.40) for some sequence of positive numbers $\{\alpha_i\}_{i=0}^{\infty}$. From (1.31), the following inequality holds:

$$\min_{s\in S} f(s) + \mu e^T(e - \varepsilon^{-\alpha|s|}) \leq \inf_{s\in S} f(s) + \mu\|s\|_0\ .$$

Hence for $\alpha_i \geq \alpha_0 = \alpha_0(\mu)$:

$$\begin{aligned} f(\hat{s}) + \mu e^T(e - \varepsilon^{-\alpha_i \hat{z}}) &= f(s(\alpha_i)) + \mu e^T(e - \varepsilon^{-\alpha_i z(\alpha_i)}) \\ &= \min_{(s\ z)\in T} f(s) + \mu e^T(e - \varepsilon^{-\alpha_i z}) \\ &= \min_{s\in S} f(s) + \mu e^T(e - \varepsilon^{-\alpha_i |s|}) \\ &\leq \inf_{s\in S} f(s) + \mu\|s\|_0\ . \end{aligned} \tag{1.41}$$

When $i \to \infty$, we have that

$$f(\hat{s}) + \mu\|\hat{s}\|_0 = \lim_{i\to\infty} \; f(\hat{s}) + \mu e^T(e - \varepsilon^{-\alpha_i \hat{z}}) \leq \inf_{s\in S} f(s) + \mu\|s\|_0 \; .$$

Since $(\hat{s} \; \hat{z})$ is a vertex of T, by Lemma 1.40 it follows that $\hat{s}$ is a vertex of S and a vertex solution of (1.36). □

A computational approach for solving problem (1.28) can be immediately described. In fact, it is easy to see that (1.28) is a special case of (1.37). Then by Theorem 1.2.4 it follows that by solving the smooth approximation (1.33) for a sufficiently large value of α, we obtain a solution of the original non-smooth problem.

1.2.12 The SLA Algorithm

A good method for minimizing a concave function on a polyhedral set is the Successive Linear Approximation (SLA) method , which is a finitely terminating stepless Frank-Wolfe algorithm. In [57] Mangasarian established the finite termination of the SLA for a differentiable concave function, and in [58] for a non-differentiable concave function using its subgradient. Here is given a general version of the SLA algorithm for minimizing a concave function f on a polyhedral set $X \subset R^n$, with f bounded below on X:

General SLA Algorithm

Initialization. $x^0 \in R^n$ randomly chosen.

1. Having x^k determine

$$x^{k+1} \in \arg \min_{x\in X} \nabla f(x^k)^T(x - x^k) \tag{1.42}$$

2. If $x^k \in X$ and

$$\nabla f(x^k)^T(x^{k+1} - x^k) = 0$$

then STOP.

3. Go to step **1**.

The next theorem shows that this algorithm generates a strictly decreasing finite sequence $\{f(x^k)\}_{k=0}^{\bar{k}}$ which terminates at a stationary point that may also be a global minimum solution.

Theorem 1.2.5. *Let f be a concave function bounded below on a polyhedral set X. The SLA converges at a vertex stationary point in a finite number of iterations.*

Proof. At each iteration k, the following linear programming problem

$$\min_{x \in X} \nabla f(x^k)^T (x - x^k) \tag{1.43}$$

must be solved at Step 2 of the algorithm. As f is bounded below on X and is a concave continuously differentiable function, for all $x \in X$ we can write

$$-\infty < \inf_{x \in X} f(x) - f(x^k) \leq f(x) - f(x^k) \leq \nabla f(x^k)^T (x - x^k) \ . \tag{1.44}$$

Hence we have that problem (1.43) admits a vertex solution x^{k+1} for any $x^k \in R^n$. If $x^{k+1} = x^k$ then x^k is a stationary point (provided $x^k \in X$) since

$$0 = \nabla f(x^k)^T (x^{k+1} - x^k) \leq \nabla f(x^k)^T (x - x^k) \quad \forall x \in X.$$

If $x^{k+1} \neq x^k$ then, using the assumptions on f, we have

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T (x^{k+1} - x^k) < f(x^k), \tag{1.45}$$

where x^{k+1} is a vertex of X. Since the number of vertices is finite and the sequence generated is strictly decreasing, it follows that the algorithm terminates in a finite number of iterations with a vertex stationary point. □

It is possible now to state the SLA algorithm for problem (1.33):

SLA Algorithm for finding a sparse representation in the presence of noise

Initialization. $x^0 \in R^n$ randomly chosen. Set $y^0 = |Ax^0 + b|$, $z^0 = |x^0|$.

1. Having $(x^k\ y^k\ z^k)$ determine

$$(x^{k+1}\ y^{k+1}\ z^{k+1}) \in \arg \min_{(x\ y\ z)\in T} (1-\lambda)e^T y + \lambda\alpha(\varepsilon^{-\alpha z^k})^T z \quad (1.46)$$

2. If $(x^k\ y^k\ z^k) \in T$ and

$$(1-\lambda)e^T y^k + \lambda\alpha(\varepsilon^{-\alpha z^k})^T z^k = (1-\lambda)e^T y^{k+1} + \lambda\alpha(\varepsilon^{-\alpha z^k})^T z^{k+1}$$

then STOP.

3. Go to step **1**.

By Theorem 1.2.5 we have the following finite termination result for this version of the SLA algorithm:

Corollary 1.2.4. *The SLA algorithm for problem (1.33) generates a finite sequence* $(x^k\ y^k\ z^k)$ *with strictly decreasing objective function values and terminates at a step* $\bar{k}$ *satisfying the minimum principle necessary optimality condition:*

$$(1-\lambda)e^T(y - y^{\bar{k}}) + \lambda\alpha(\varepsilon^{-\alpha z^{\bar{k}}})^T(z - z^{\bar{k}}) \geq 0 \quad \forall\ (x\ y\ z) \in T\ .$$

Chapter 2

Methods for Feature Selection

With the growth in size of databases, the problem of focusing on the most relevant information in a potentially overwhelming quantity of data has become increasingly important. For instance, people working in different fields such as engineering, astronomy, biology, remote sensing, economics, deal with larger and larger observations and simulations and need a way to effectively utilize these data. One of the most challenging problems in high-dimensional data analysis is *feature selection*. This problem basically consists in eliminating as many features as possible in a given problem while still carrying out a certain task with good accuracy. In classification the goal is that of discriminate between two given sets in a high-dimensional feature space by using as few of the considered features as possible.
Although feature selection is primarily performed to choose informative features, it can have other motivations: facilitating data visualization and data understanding, reducing the storage requirements, reducing the execution time, improving prediction performance. Some methods put more emphasis on one aspect than another.
In this chapter the various aspects of feature selection for classification are analyzed. The first section gives a brief description of classification, which is a very important task in machine learning, and feature selection. The second section is basically an overview of the existing methods for feature selection. The last section is focused on mathematical programming approaches to feature selection.

2.1 Feature Selection in Classification

In this section we present some key notions that make easier the understanding of this chapter. We start by giving a brief description of the classification problem in supervised learning. Then we put our attention on a specific task: the feature selection for classification. We formally define the problem and explain why is so important in data mining.

2.1.1 Supervised Learning and Classification

Consider a functional dependency g that maps points from an input space X to an output space Y. In *supervised learning*, the goal is extracting an estimate $\hat{g}$ of g from a given finite set of training data pairs (*training set*) containing the input x^i and the desired output y^i:

$$T = \{(x^i,\ y^i) \mid x^i \in X,\ y^i \in Y \ \text{ and } i = 1, \ldots, m\} \ .$$

When dealing with *classification* problems, the input space is divided into k subsets $X_1, \ldots, X_k \in X$ such that

$$X_i \cap X_j = \emptyset \quad i, j = 1, \ldots, k, \quad i \neq j$$

and the task becomes that of assigning a given input vector x to the subset it belongs to. From now on we only consider the classification task in the basic form of *binary classification*: we have two sets $X_1, X_2 \in X$, such that $X_1 \cap X_2 = \emptyset$, and we want to determine whether an input vector $x \in X$ belongs to X_1 or X_2. the training set for binary classification is formally defined as follows

$$T = \{(x^i,\ y^i) \mid x^i \in X,\ y^i \in \{\pm 1\} \ \text{ and } i = 1, \ldots, m\}$$

with the two classes X_1 and X_2 labelled by $+1$ and -1, respectively. The functional dependency $g : X \to \{\pm 1\}$, which determines the class of a given vector x, assumes the following form:

$$g(x) = \begin{cases} +1 & \text{if } x \in X_1 \\ \\ -1 & \text{if } x \in X_2 \ . \end{cases} \tag{2.1}$$

It is possible to use various classes of *learning machines*, which have different functional forms, for constructing an approximation $\hat{g}$ of g. Here are three widely-studied classes of learning machines:

1) Perceptron;

2) MultiLayer Perceptron Networks (MLPN),

3) Radial Basis Function Networks (RBFN),

4) Support Vector Machines (SVM).

A multilayer network typically consists of an *input layer*, which is basically a set of source nodes, one or more *hidden layers*, composed by various computational nodes, and an *output layer* of computational nodes. We can construct (*train*) the desired network $\hat{g}$ in a supervised manner by using a popular algorithm known as the *backpropagation algorithm* [74, 42].
A completely different approach is that of designing a neural network as a curve-fitting problem in a high-dimensional space by means of radial basis functions. A radial basis function network [69, 13] has three layers. The first layer is the input layer, a set of source nodes used to connect the network to its environment. The second layer,which is the only hidden layer, maps the input vector x into a hidden space of high dimensionality. The last layer (output layer) gives the response of the network to a given input vector x.
Support vector machines, introduced by Vapnik [86], represent another efficient tool for classification. This class of learning machines implements in an approximate way the method of *structural risk minimization*. Training a support vector machine requires the solution of a large dense quadratic programming problem. Different kinds of algorithms have been developed in the last twenty years in order to tackle this challenging task [66, 45, 60, 76, 68, 48, 20, 30]. The hope is that the estimate $\hat{g}$ of g obtained by using one of the learning machine models described above will *generalize*. A learning machine is said to generalize well when it is able to compute correctly the input-output mapping for *test data* not included in the training set. The generalization ability of a learning machine is strictly connected with its complexity. In fact, a complex estimate $\hat{g}$ usually approximates g poorly on points not in the training set. Such a phenomenon is referred to as *overfitting* or *overtraining*. A model which is too simple, however, is also not preferred as it gives too poor a fit to the training data. In order

to find the optimal complexity for our learning machine, we can utilize the *Occam's Razor* [9], named after William of Occam (1285-1349). This model selection criteria favors the simplest model possible that still grants good performance on the training data.
In order to evaluate the generalization ability of a learning machine, we can use the procedure of *cross-validation* [80]. We divide the training set T into k distinct segments $T_1, \ldots, T_k$. We then construct the function $\hat{g}$ by a learning algorithm using data from $k-1$ of the segments and test its performance using the remaining segment. This process is repeated for each of the k possible choices for the segment omitted from the training process, and the k results averaged to produce a single result. When k is equal to the number of training data we obtain the *leave-one-out method*.

2.1.2 Feature Selection Problem

In real-world classification problems requiring supervised learning, nothing is known about the mapping function g. The only information available is in the *features*, or components, of the vectors x^i contained in the training set. Features can be divided into three groups [46]:

1. *irrelevant features*: these features do not affect the target concept in any way (i.e. no useful information carried);

2. *redundant features*: a feature belonging to this group adds nothing new to the target concept (i.e. information already carried by other features);

3. *relevant features*: a relevant feature is neither irrelevant nor redundant.

As relevant features are unknown *a priori*, many candidate features are usually included in order to better describe the domain. Unfortunately, many of these features are *irrelevant* or *redundant* and their presence does not provide more discrimination ability. Furthermore, data sets with a large number of features and a limited number of training examples lead to the "curse of dimensionality": the data are very sparse and provide a poor representation of the mapping [8]. Then the only way to construct a good estimator $\hat{g}$ is to choose a small subset of predictive features, discarding irrelevant/redundant features. *Feature selection* for classification can be formally defined as a process to select a minimally sized subset of features, while still having good

performance and accurately estimating g over the training set.
Although feature selection is primarily performed to select informative features, it can have other motivations:

1. *data reduction*, to limit storage requirements and make algorithms faster;
2. *feature set reduction*, to save resources in the next round of data collection or during utilization;
3. *prediction accuracy improvement*, to obtain better classification performance;
4. *data understanding*, to better understand the process that generated the data.

In an n-dimensional space, methods for features selection ideally try to find the best subset of features among all the 2^n candidate subsets according to some criteria. It is evident that the search becomes too costly as n gets large. Therefore, a good procedure to prevent an exhaustive search of the subsets is needed. Various approaches to feature selection will be described in the next sections.

2.2 Machine Learning Approaches to Feature Selection

As the main aim of machine learning is addressing larger, more complex tasks, feature selection has become one of the most important issues in this field. Machine learning methods for feature selection can be divided into three classes: (*i*) Filters; (*ii*) Wrappers; (*iii*) Embedded methods.
Filters select subsets of variables as a pre-processing step, no matter what the predictor is. Wrapper algorithms utilize the learning machine of interest as a black box to score subsets of variable according to their predictive power. Embedded methods perform variable selection as a part of the training process and are usually specific to given learning machines.
The present section gives an overview of the existing machine learning approaches for feature selection.

2.2.1 Filter Methods

Filters are a special class of feature selection algorithms that are independent from the predictor used for classification, see Figure 2.2.1. In fact, performance of filters are evaluated only by means of some metrics calculated directly from data, without taking into account the learning machine that will be trained with reduced data. These algorithms are generally less expensive than wrapper ones.

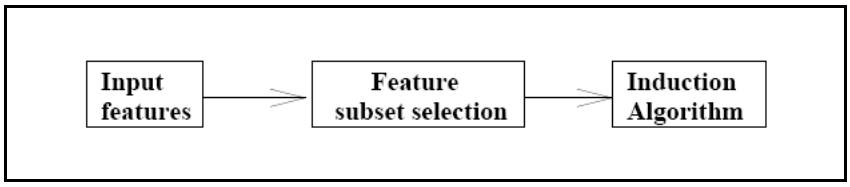

Figure 2.2.1. The filter approach to feature selection.

Given a set of data D, in filters we usually calculate a *scoring function* $F(S)$ that estimates how relevant is the subset S for classification. Using these relevance indices with single features x_j with $j = 1, \ldots, n$, a ranking order may be easily established:

$$F(x_{j_1}) \leq F(x_{j_2}) \leq \cdots \leq F(x_{j_n}) \ .$$

Those features having a low rank are discarded. Due to its simplicity, variable ranking is included as a selection mechanism in various feature selection algorithms [5, 90, 33].
A well-known method for variable ranking is the Relief Algorithm [49]. It assigns a weight to each feature, which denotes the relevance of the feature as represented in the training set. Examples from the training set are randomly selected and the relevance is updated based on the difference between the selected example and the two nearest examples of the same and opposite class. Since each feature relevance is calculated independently of other features, relief is not able to find redundant features [49].
The FOCUS algorithm [1] exhaustively examines all subsets of features, selecting the minimal subset sufficient for classification. FOCUS may not be able to find irrelevant features when noise is present in the training set, or if the training set is not representative of future data [46].

Another filter method based upon information theory is introduced in [51]. This method attempts to compute a subset of features such that the probability distribution of the class is close to the distribution of the class given by the full set of features. Distance used is the KL-distance or equivalently, cross-entropy.

2.2.2 Wrapper Methods

The Wrapper approach, described by Kohavi and John in [46], is a simple and powerful tool for variable selection which consists in using the prediction performance of a given learning machine to determine the usefulness of a subset of variables, see Figure 2.2.2. This approach basically conducts a search in the space of possible parameters. An exhaustive search can be performed only if the number of variables is sufficiently small. Hence, in order to save time we need to use efficient search strategies such as branch and bound, greedy selection, simulated annealing, genetic algorithms. Search strategies will be described in the next section. Performance evaluation is usually done using a validation set or by cross-validation.
Wrappers are often considered a costly method as they seem to require a large amount of computation. When good and inexpensive searches, such as the greedy ones, are adopted, this is not necessarily true. Since the learning machine can be represented as a black box, wrappers are a very simple to use technique for feature selection.

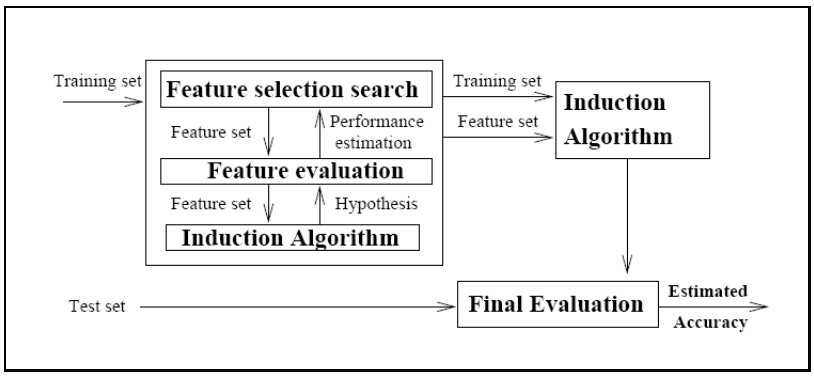

Figure 2.2.2. The wrapper approach to feature selection.

2.2.3 Search Strategies

A search strategy establishes the order in which the variable subsets are evaluated. The simplest strategy is the exhaustive search: every possible subset is evaluated and the best one is chosen. Unfortunately this approach is affordable only when the number of variable is quite small. When exhaustive search is not possible, searching only a part of the subset would be a good strategy. When a certain subset size d is given and the evaluation function is monotonic (i.e. the addition of a variable never makes a subset worse), search may be done by means of the branch and bound algorithm [64]. The strategy is based on the fact that once a subset S having a dimension greater than d has been evaluated, if the score for S is worse than the score obtained using the currently best known subset S' of size d, then there is no need to evaluate the subsets of S, because the score of those subsets will never exceed the score of S'. The algorithm has an exponential worst case complexity and this makes the approach infeasible when a large number of variables is given.
Greedy selection methods [22, 62] represent a computationally advantageous class of search strategies. There are two types of greedy searches:

1) *forward selection*: it begins with the empty set of features and progressively includes new variables to make the set larger;

2) *backward elimination*: it refers to a search that starts with the full set of features and progressively eliminates the less promising ones.

Simulated annealing is a stochastic algorithm introduced by Kirkpatrick et al. [50] for the general search of the minimum of a function. It is based on how physical matter cools down and freezes, ending up in a structure that minimizes the energy of the body. Siedlecki and Sklanski in [77] first suggested this stochastic approach as a search strategy. In simulated annealing, the search starts with an initial random subset in a high "temperature". At each step, a small random change is introduced to the subset. if the subset obtained is better, the change is accepted. If the subset is worse, the change is accepted or refused depending on the "temperature": in a high temperature, a worse subset is more likely to be accepted than in a low temperature. Genetic algorithms [61] represent another family of stochastic optimization methods. The main difference between the two algorithms is that simulated annealing has only a subset in memory, while genetic algorithms keep a set

of them. In genetic algorithms a solution is usually called chromosome and a set of chromosomes is called a population. A new population is usually obtained by retaining some chromosomes in the old population and creating new chromosomes by means of a manipulation of old chromosomes. The better a chromosome is, the higher is the probability to be selected to the new population or as a parent in a genetic operation. This method is well-suited for problems with a large number of variables [52].

2.2.4 Embedded Methods

The main difference between embedded methods and other feature selection methods is in the way the feature selection is combined with learning. Filter methods are completely independent from learning. Wrappers use a learning machine to evaluate the quality of a subset of features without taking into account knowledge about the structure of the classification function. Differently from filters and wrappers, embedded methods incorporate feature selection as a part of the training process. The embedded approach can be formally described as follows:

$$\begin{array}{ll} \min\limits_{w \in R^n} & g(w, X, Y) \\ s.t. & s(w) \leq s_0 \end{array} \tag{2.2}$$

where g is a function measuring the performance of the selected learning machine, described by the vector of parameters w, on the given training data (X, Y), and s is an approximation of the zero norm. Problem (2.2) can be converted in a problem of the form

$$\min_{w \in R^n} \quad g(w, X, Y) + \lambda s(w) \tag{2.3}$$

with $\lambda > 0$. Some embedded methods iteratively add or remove features from the data to approximate a solution of the problem (2.2). Iterative methods can be divided in two classes:

1. *Forward selection methods* [15, 12, 67]: these methods start with one or a few features selected according to some criteria and iteratively add more features until a stopping criterion is satisfied;

2. *Backward elimination methods* [39, 54, 70]: methods belonging to this class start with the full set of features and iteratively remove one or more features.

Weston et al. [89] proposed a method of using SVMs for feature selection based on choosing the scaling factors which minimize a bound. Feature selection is performed by scaling the input parameters by a vector $\sigma \in [0,1]^n$. The larger is the value of σ_i the more useful is the i-th feature. Thus the problem become that of choosing the best kernel of the form:

$$k_\sigma(x, x') = k(\sigma * x, \sigma * x')$$

where $*$ is the element-wise multiplication.
Some embedded methods can be reformulated as a minimization problem of type (2.3). These methods will be deeply described in the next section.

2.3 Feature Selection as an Optimization Problem

In feature selection methods for classification, the task is to discriminate between two given sets in an n-dimensional feature space using as few of the given features as possible. This problem can be addressed in a simple way by means of linear classification models. The goal, in this case, is to construct a separating plane that gives good performance on the training set, while using a minimum number of problem features. When dealing with linear models, feature selection can be directly enforced on the parameters of the model. This can be made by adding a sparsity term to the objective function that the model minimizes.
In the first part of the section we formulate the feature selection problem as a mathematical program with a parametric objective function which attempts to construct a separating plane in a feature space of as small dimension as possible. We also show a fundamental result about the complexity of feature selection for linearly separable data sets.
In the second part of the section various methods of solution for the mathematical programming formulation of the problem are described: replacement of the sparsity term in the objective function with the ℓ_1-norm; approximation of the sparsity term by a sigmoid function, by a concave exponential function, and by a logarithmic function; a bilinear function minimization over a polyhedral set.

2.3.1 Feature Selection using Linear Models

Let us consider two nonempty finite point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n consisting of m and k points respectively. The point sets $\mathcal{A}$ and $\mathcal{B}$ are represented by the matrices $A \in R^{m \times n}$ and $B \in R^{k \times n}$, where each point of a set is represented as a row of the matrix. In the feature selection problem, we want to construct a separating plane:

$$P = \{x \mid x \in R^n,\ x^T w = \gamma\} \tag{2.4}$$

with normal $w \in R^n$ and distance

$$\frac{|\gamma|}{\|w\|_2}$$

to the origin, while suppressing as many of the components of w as possible. The separating plane P determines two open halfspaces :

- $\{x \mid x \in R^n,\ x^T w > \gamma\}$ containing mostly points belonging to $\mathcal{A}$;
- $\{x \mid x \in R^n,\ x^T w < \gamma\}$ containing mostly points belonging to $\mathcal{B}$.

Therefore, we want to satisfy the following inequalities:

$$Aw > e\gamma,\ Bw < e\gamma \tag{2.5}$$

to the extent possible. A normalized version of these inequalities is

$$Aw > e\gamma + e,\ Bw < e\gamma - e\ . \tag{2.6}$$

Conditions (2.5) and (2.6) can be satisfied if and only if the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ are disjoint (i.e. the two sets are linearly separable). Amaldi and Kann [3] established that the feature selection problem for linearly separable data sets is hard to approximate:

Theorem 2.3.1. *Let $DTIME(p^{polylog\ p})$ be the class of problems whose instances of size s can be solved in deterministic time $O(s^{polylog\ s})$, with polylog s any polynomial in log s.*
Assuming $NP \nsubseteq DTIME(p^{polylog\ p})$, the feature selection problem is not approximable within a factor of $2^{log^{1-\epsilon} p}$ for any $\epsilon > 0$, where p is the number of examples.

As shown in Theorem 2.3.1, designing close-to minimum size networks in terms of nonzero weights is very hard even for linearly separable training sets that are performable by the simplest type of networks.

2.3.2 A Mathematical Programming Formulation of the Feature Selection Problem

In real-world applications we hardly find linearly separable data sets. Thus we try to satisfy (2.6), in some approximate sense, by minimizing the norm of the average violation of those inequalities:

$$\min_{w,\gamma} f(w,\gamma) = \min_{w,\gamma} \frac{1}{m}\|(-Aw + e\gamma + e)_+\|_1 + \frac{1}{k}\|(Bw - e\gamma + e)_+\|_1 \quad (2.7)$$

The reasons for choosing the ℓ_1-norm in (2.7) are:

(i) Problem (2.7) is equivalent to a linear programming problem (2.8) with many important theoretical properties;

(ii) the ℓ_1-norm is less sensitive to outliers (i.e. those occurring when the underlying data distributions have pronounced tails).

The formulation (2.7) is equivalent to the following linear programming formulation proposed in [6]:

$$\begin{array}{ll} \min\limits_{w,\gamma,y,z} & \dfrac{e^T y}{m} + \dfrac{e^T z}{k} \\ s.t. & -Aw + e\gamma + e \le y \\ & Bw - e\gamma + e \le z \\ & y \ge 0, \quad z \ge 0 \end{array} \quad (2.8)$$

This linear programming problem, or equivalently formulation (2.7), defines a separating plane P that approximately satisfies the conditions (2.6). Since in feature selection the goal is suppressing as many elements of w as possible, a feature selection term must be introduced:

$$\begin{array}{lll} \min\limits_{w,\gamma,y,z} & (1-\lambda)(\dfrac{e^T y}{m} + \dfrac{e^T z}{k}) + \lambda\|w\|_0 & \\ s.t. & -Aw + e\gamma + e \le y & \lambda \in [0,1) \\ & Bw - e\gamma + e \le z & \\ & y \ge 0, \quad z \ge 0 & \end{array} \quad (2.9)$$

This problem is equivalent to the following parametric program:

$$\begin{array}{lll} \min\limits_{w,\gamma,y,z,v} & (1-\lambda)(\frac{e^T y}{m}+\frac{e^T z}{k})+\lambda\sum\limits_{i=1}^{n} s(v_i) & \\ s.t. & -Aw+e\gamma+e\leq y & \lambda\in[0,1) \\ & Bw-e\gamma+e\leq z & \\ & -v\leq w\leq v & \\ & y\geq 0, \quad z\geq 0 & \end{array} \tag{2.10}$$

where $s : R \to R^+$ is the *step function* such that $s(t) = 1$ for $t > 0$ and $s(t) = 0$ for $t \leq 0$. This is the **fundamental feature selection (FS) problem**, as defined in [57]. The parameter λ is used to balance two objectives:

1. the number of misclassified training data;
2. the number of nonzero elements of vector w.

The feature selection problem (2.10) will be solved for a value of λ for which the generalization ability of the classifier is maximized. Usually, this will be achieved in a feature space of reduced dimensionality, that is $\|w\|_0 < n$. As the step function in (2.10) is discontinuous, it is typically replaced by a smooth function, such as:

- sigmoid function;
- concave exponential function;
- logarithm function.

In order to make the problem (2.9) tractable, it is also possible to replace the ℓ_0-norm with another function such as the ℓ_1-norm, or to reformulate it as a linear program with equilibrium constraints.

2.3.3 Norms and their Duals

We introduce here the concept of dual norm (See [73, 44] for further details).

Definition 2.3.1. (Support Function) *Let S be a nonempty set in R^n. The function $\sigma_S : R^n \to R \cup \{+\infty\}$ defined by*

$$R^n \ni x \to \sigma_S(x) = \sup\{s^T x : s \in S\}$$

is called the support *of S.*

Definition 2.3.2. (Gauge) *Let C be a closed convex set in R^n containing the origin. The function γ_C defined by*

$$\gamma_C(x) = \inf\{\lambda > 0 \;\; x \in \lambda C\}$$

is called the gauge *of C.*

It is possible to prove that

$$\{x \in R^n \;:\; \gamma_C(x) \leq 1\} = C.$$

Let $\|\cdot\|$ be an arbitrary norm on R^n. It is a positive (except at 0) closed sublinear function and its sublevel-set:

$$B = \{x \;\in\; R^n, \;\; : \|x\| \leq 1\} \tag{2.11}$$

is very interesting. In fact, the unit ball associated with the norm is a symmetric, convex, compact set containing the origin as interior point; $\|\cdot\|$ represents the *gauge* of B. Let us consider the set whose support function is $\|\cdot\|$:

$$B^* = \{s \;\in\; R^n, \;\; : \; s^T x \leq \|x\| \; for \; all \; x \; in \; R^n\}. \tag{2.12}$$

It easy to check that B^* is also symmetric, convex, compact; and it contains the origin as an interior point.
Now, we can generate two more closed sublinear functions: the support function of B and the gauge of B^*. It turns out that we then obtain the same function, which actually is a norm, denoted by $\|\cdot\|^*$: this is the so called *dual norm* of $\|\cdot\|$. It is support of B and gauge of B^*.

Theorem 2.3.2. *Let B and B^* defined as in (1) and (2), where $\|\cdot\|$ is a norm of R^n. The support function of B and the gauge of B^* are the same function $\|\cdot\|^*$ defined by:*

$$\|s\|^* = \max\{s^T x \;:\; \|x\| \leq 1\}. \tag{2.13}$$

Furthermore, we have:

$$\|x\| = \max\{s^T x \;:\; \|s\|^* \leq 1\}. \tag{2.14}$$

Proof. See Proposition 3.2.1 [44].

Using theorem (2.3.2) we can show the following result:

Corollary 2.3.1. *The ∞-norm:*

$$\|x\|_\infty = max\ \{|x_1|, \ldots, |x_n|\} \tag{2.15}$$

is the dual of the ℓ_1-norm:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|. \tag{2.16}$$

Proof. Let us consider the dual of the ℓ_1-norm. By using (2.13), we can define it as follows:

$$\|s\|^* = \max\{s^T x \ : \ \|x\|_1 \leq 1\}. \tag{2.17}$$

We need to maximize the scalar product $s^T x$, which can be written as follows:

$$s^T x = s_1 x_1 + \cdots + s_n x_n. \tag{2.18}$$

The following inequality holds for (2.18):

$$s^T x = s_1 x_1 + \cdots + s_n x_n \leq |s_1||x_1| + \cdots + |s_n||x_n|. \tag{2.19}$$

Let s_M be the component of s vector with the largest absolute value. By considering the constraint of (2.17), we have:

$$|s_1||x_1| + \cdots + |s_n||x_n| \leq |s_M|(|x_1| + \cdots + |x_n|) \leq |s_M|. \tag{2.20}$$

Thus we can write:

$$\max\{s^T x \ : \ \|x\|_1 \leq 1\} \leq |s_M| \tag{2.21}$$

Then, the optimal solution is x vector with:

$$x_i = 0 \quad i = 1, \ldots, n, \ i \neq M \tag{2.22}$$

$$x_M = \begin{cases} 1 & if\ s_M \geq 0 \\ -1 & if\ s_M < 0 \end{cases} \tag{2.23}$$

This is equivalent to:

$$\max\ \{|s_1|, \ldots, |s_n|\} = \|x\|_\infty \tag{2.24}$$

Let us show now :

$$\|x\|_1 = \max\ \{s^T x\ :\ \|s\|_\infty \leq 1\} \tag{2.25}$$

Our problem is maximizing $s^T x$, subject to constraint

$$\|s\|_\infty \leq 1\ \equiv\ max\ \{|s_1|, \ldots, |s_n|\} \leq 1. \tag{2.26}$$

In order to solve (2.25), we then set s_i as follows:

$$s_i = \begin{cases} 1 & if\ x_i \geq 0 \\ -1 & if\ x_i < 0 \end{cases} \quad i = 1, \ldots, n. \tag{2.27}$$

This is equivalent to write:

$$|x_1| + \cdots + |x_n| = \sum_{i=1}^{n} |x_i| = \|x\|_1 \tag{2.28}$$

Therefore, we obtained that ∞-norm is the dual of 1-norm. □

2.3.4 ℓ_1-norm based Approach

When an appropriate norm is used for measuring the distance between the two parallel bounding planes for the sets being separated, SVMs does indirectly suppress components of the normal vector w. In the SVM approach the term $\frac{\|w\|'}{2}$ is added to the objective function of (2.10) the same way the term $\|w\|_0$ is added in (2.9). Here, $\|\cdot\|'$ is the dual of some norm on R^n used to measure the distance between the two bounding planes.
The following theorem [59] gives an explicit form for the projection of an arbitrary point on a plane using a general norm in order to measure the distance between the point and its projection:

Theorem 2.3.3. *Let q be any point in R^n not on the plane:*

$$P = \{x \mid w^T x = \gamma\},\ 0 \neq w \in R^n,\ \gamma \in R\ .$$

A projection $p(q) \in P$ using a general norm $\|\cdot\|$ on R^n is given by:

$$p(q) = q - \frac{w^T q - \gamma}{\|w\|'} y(w), \tag{2.29}$$

where $\|\cdot\|'$ is the dual norm of $\|\cdot\|$ and:

$$y(w) \in \arg\max_{\|y\|=1} w^T y\ . \tag{2.30}$$

The distance between q and its projection $p(q)$ is

$$\|q - p(q)\| = \frac{|w^T q - \gamma|}{\|w\|'}. \tag{2.31}$$

The distance between the two bounding planes is given in the following corollary:

Corollary 2.3.2. *Let $\|\cdot\|$ be a norm in R^n. The distance d between the two bounding planes $w^T x = \gamma + 1$ and $w^T x = \gamma - 1$ is equal to $\frac{2}{\|w\|'}$.*

Proof. Consider two points q_1 and q_2 belonging to $w^T x = \gamma + 1$ and $w^T x = \gamma - 1$ respectively. Suppose these points have the same projection on P:

$$p(q_1) = p(q_2) = \bar{p}\ .$$

By theorem 2.3.3, the distance d between the two planes is obtained as follows:

$$d = \|q_1 - q_2\| = \|q_1 - \bar{p} - (q_2 - \bar{p})\| =$$
$$= \frac{|w^T q_1 - \gamma - (w^T q_2 - \gamma)| \cdot \|y(w)\|}{\|w\|'} = \frac{2}{\|w\|'}\ .$$

□

The separating plane P (2.4) lies halfway between the two bounding planes $w^T x = \gamma + 1$ and $w^T x = \gamma - 1$ and, as shown in corollary 2.3.2, the distance between these planes is exactly $\frac{2}{\|w\|'}$. Hence, the term $\frac{\|w\|'}{2}$ is added to the

objective function of (2.10) in order to drive up the distance between the two planes, thus getting a better separation. The following mathematical programming formulation for SVM is given:

$$\begin{aligned} \min_{w,\gamma,y,z} \quad & (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \frac{\lambda}{2}\|w\|' \\ s.t. \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & y \geq 0, \quad z \geq 0 \end{aligned} \tag{2.32}$$

with $\lambda \in [0, 1)$. If the ∞-norm is used to measure the distance between the planes, the dual norm is the ℓ_1-norm which leads to the following linear programming problem [10]:

$$\begin{aligned} \min_{w,\gamma,y,z,s} \quad & (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \frac{\lambda}{2}e^T s \\ s.t. \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & -s \leq w \leq s \\ & y \geq 0, \quad z \geq 0 \end{aligned} \tag{2.33}$$

with $\lambda \in [0, 1)$. The main difference with respect to a classical SVM is the use of the ∞-norm instead of the ℓ_2-norm to measure the distance between the two bounding planes and the change of the quadratic regularization term $\|w\|_2^2$ with $\|w\|_1$ in (2.32). This new term induces big differences in the outcome of the optimization. In fact, the solutions obtained by means of (2.33) are much sparser than those obtained by using the classical SVM approach [10].

2.3.5 Approximating the l_0-norm by the Standard Sigmoid Function

In [57], Mangasarian proposed a continuous approximation of the step function in (2.10) using the standard sigmoid function:

$$v(t) = (e + \varepsilon^{-\alpha t})^{-1} \ . \tag{2.34}$$

This leads to the following mathematical programming problem:

$$\begin{aligned} \min_{w,\gamma,y,z,v} \quad & (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \lambda \sum_{i=1}^{n} (1+\varepsilon^{-\alpha v_i})^{-1} \\ s.t. \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & -v \leq w \leq v \\ & y \geq 0, \quad z \geq 0 \end{aligned} \tag{2.35}$$

with $\lambda \in [0,1)$. As the objective of this problem is neither concave nor convex, it must be solved by means of a nonlinear optimization code.

2.3.6 A Concave Exponential Approximation of the ℓ_0-norm

Another way to solve problem 2.10 is replacing the step function with a concave exponential function [57]. Thus the following concave programming problem is obtained:

$$\begin{aligned} \min_{w,\gamma,y,z,v} \quad & (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \lambda \sum_{i=1}^{n} (1-\varepsilon^{-\alpha v_i}) \\ s.t. \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & -v \leq w \leq v \\ & y \geq 0, \quad z \geq 0 \end{aligned} \tag{2.36}$$

with $\lambda \in [0,1)$. The replacement of (2.10) by the smooth concave problem (2.36) is well-motivated (see [11]) both from a theoretical and a computational point of view:

- for sufficiently high values of the parameter α there exists a vertex solution of (2.36) which provides a solution of the original problem (2.10), and in this sense the approximating problem (2.36) is equivalent to the given nonsmooth problem (2.10), as already shown in theorem 1.2.4;

- because its objective function is a differentiable concave function, and has a vertex solution, it is possible to use the Successive Linear Approximation (SLA) algorithm. This algorithm is guaranteed to converge

to a vertex stationary point of (2.36) in a finite number of iterations (this convergence result was proved for a general class of concave programming problems, See theorem 1.2.5); thus the algorithm requires the solution of a finite sequence of linear programs for computing a stationary point of (2.36), and this may be quite advantageous from a computational point of view.

Here is an outline of the SLA algorithm for feature selection, where T is the feasible set of the problem (2.36):

SLA Algorithm for Feature Selection (Concave Exponential Approximation)

Initialization. Choose $\lambda \in [0, 1)$. Start with a random (w^0, γ^0). Set $y_0 = (-Aw^0 + e\gamma^0 + e)_+$, $z_0 = (Bw^0 - e\gamma^0 + e)_+$ and $v^0 = |w^0|$.

1. Having $(w^k, \gamma^k, y^k, z^k, v^k)$ determine $(w^{k+1}, \gamma^{k+1}, y^{k+1}, z^{k+1}, v^{k+1})$ by solving the linear program:

$$\min_{(w,\gamma,y,z,v)\in T} (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \lambda\alpha(\varepsilon^{-\alpha v^k})^T(v - v^k) \qquad (2.37)$$

2. If $(w^k, \gamma^k, y^k, z^k, v^k) \in T$ and

$$(1-\lambda)(\frac{e^T(y^{k+1} - y^k)}{m} + \frac{e^T(z^{k+1} - z^k)}{k}) + \lambda\alpha(\varepsilon^{-\alpha v^k})^T(v^{k+1} - v^k) = 0$$

then STOP.

3. Go to step **1**.

By Theorem 1.2.5 we have the following finite termination result for this version of the SLA algorithm:

Corollary 2.3.3. *The SLA algorithm for problem (2.36) generates a finite sequence $(w^k, \gamma^k, y^k, z^k, v^k)$ with strictly decreasing objective function values and terminates at a step $\bar{k}$ satisfying the minimum principle necessary optimality condition:*

$$(1-\lambda)(\tfrac{e^T(y-y^{\bar{k}})}{m} + \tfrac{e^T(z-z^{\bar{k}})}{k}) + \lambda\alpha(\varepsilon^{-\alpha v^{\bar{k}}})^T(v - v^{\bar{k}}) \geq 0, \forall\ (w, \gamma, y, z, v) \in T.$$

2.3.7 A Logarithmic Approximation of the ℓ_0-norm

A similar concave optimization-based approach has been proposed in [90], where the idea is that of using the logarithm function instead of the step function, and this leads to a concave smooth problem of the form:

$$\begin{aligned} \min_{w,\gamma,y,z,v} \quad & (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \lambda \sum_{i=1}^{n} \log(\epsilon + v_i) \\ s.t. \quad & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & -v \leq w \leq v \\ & y \geq 0, \quad z \geq 0 \end{aligned} \tag{2.38}$$

with $\lambda \in [0, 1)$ and $0 < \epsilon \ll 1$. Similarly to [**?**], the SLA algorithm has been applied to solve (2.38). We now state the version of SLA algorithm using a logarithmic approximation of the ℓ_0-norm; with T we indicate the feasible set of the problem (2.38):

SLA Algorithm for Feature Selection (Logarithmic Approximation)

Initialization. Choose $\lambda \in [0, 1)$. Start with a random (w^0, γ^0).
Set $y_0 = (-Aw^0 + e\gamma^0 + e)_+$, $z_0 = (Bw^0 - e\gamma^0 + e)_+$ and $v^0 = |w^0|$.

1. Having $(w^k, \gamma^k, y^k, z^k, v^k)$ determine $(w^{k+1}, \gamma^{k+1}, y^{k+1}, z^{k+1}, v^{k+1})$ by solving the linear program:

$$\min_{(w,\gamma,y,z,v)\in T} (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \lambda \sum_{i=1}^{n} \frac{v_i - v_i^k}{\epsilon + v_i^k} \tag{2.39}$$

2. If $(w^k, \gamma^k, y^k, z^k, v^k) \in T$ and

$$(1-\lambda)(\frac{e^T(y^{k+1} - y^k)}{m} + \frac{e^T(z^{k+1} - z^k)}{k}) + \lambda \sum_{i=1}^{n} \frac{v_i - v_i^k}{\epsilon + v_i^k} = 0$$

then STOP.

3. Go to step **1**.

By Theorem 1.2.5 we have for this version of the SLA algorithm the same finite termination result we obtained in the previous section:

Corollary 2.3.4. *The SLA algorithm for problem (2.38) generates a finite sequence* $(w^k, \gamma^k, y^k, z^k, v^k)$ *with strictly decreasing objective function values and terminates at a step* $\bar{k}$ *satisfying the minimum principle necessary optimality condition:*

$$(1-\lambda)(\frac{e^T(y-y^{\bar{k}})}{m}+\frac{e^T(z-z^{\bar{k}})}{k})+\lambda\sum_{i=1}^{n}\frac{v_i-v_i^k}{\epsilon+v_i^k}\geq 0,\forall\ (w,\gamma,y,z,v)\in T.$$

Formulation (2.38) is practically motivated by the fact that, due to the form of the logarithm function, it is better to increase one variable v_i while setting to zero another one rather than doing some compromise between both, and this should facilitate the computation of a sparse solution. A relation of (2.38) with the minimization of the zero-norm has been given in [**?**].

2.3.8 Feature Selection as a Linear Program with Equilibrium Constraints

By means of the following lemma [57], it is possible to give an exact reformulation of problem (2.10) as a linear program with equilibrium constraints (LPEC)[55, 57]:

Lemma 2.3.1. *Let* $a \in R^m$*. Consider the following problem:*

$$(r,u)=\arg\min_{r,u}\{e^Tr|0\leq r\perp u-a\geq 0, 0\leq u\perp -r+e\geq 0\}\ . \qquad (2.40)$$

Then $r=a_*,\ u=a_+$*.*

Proof. The constraints considered in the minimization problem constitute the Karush-Kuhn-Tucker condition for the dual programs:

$$\max_r\{a^Tr|0\leq r\leq e\}\quad \min_u\{e^Tu\mid u\geq a,\ u\geq 0\} \qquad (2.41)$$

These two problems are solved by

$$r_i=\begin{cases} 0 \text{ for } a_i<0\\ r_i\in[0,1] \text{ for } a_i=0,\ u=a_+\\ 1 \text{ for } a_i>0\end{cases} \qquad (2.42)$$

As the objective function $e^T r$ is minimized in (2.40), $r_i = 0$ for $a_i = 0$, thus giving $r = a_*$. □

Using this lemma, problem (2.10) can be rewritten in the equivalent form of a linear program with equilibrium constraints:

$$\begin{array}{ll} \min\limits_{w,\gamma,y,z,v} & (1-\lambda)(\dfrac{e^T y}{m} + \dfrac{e^T z}{k}) + \lambda e^T r \\ \\ s.t. & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & -v \leq w \leq v \\ & y \geq 0, \quad z \geq 0 \\ & 0 \leq r \perp u - v \geq 0 \\ & 0 \leq u \perp -r + e \geq 0 \end{array} \tag{2.43}$$

with $\lambda \in [0, 1)$. As problem (2.43) contains complementarity constraints, and the general linear complementarity problem is NP-complete [18], it is NP-hard in general. In order to make this problem tractable, the complementarity terms $r^T(u - v)$ and $u^T(-r + e)$ are moved into the objective function as a positive penalty term

$$-r^T v + e^T u$$

with penalty parameter $\mu \in (0, 1)$. The new formulation of the problem is:

$$\begin{array}{ll} \min\limits_{w,\gamma,y,z,v,u,r} & (1-\mu)((1-\lambda)(\dfrac{e^T y}{m} + \dfrac{e^T z}{k}) + \lambda e^T r) + \mu(-r^T v + e^T u) \\ \\ s.t. & -Aw + e\gamma + e \leq y \\ & Bw - e\gamma + e \leq z \\ & -v \leq w \leq v \\ & y \geq 0, \quad z \geq 0 \\ & 0 \leq r, \quad u - v \geq 0 \\ & 0 \leq u, \quad -r + e \geq 0 \end{array} \tag{2.44}$$

It is easy to see that the step function is modeled exactly when the penalty term $-r^T v + e^T u = 0$. This parametric bilinear program can be easily solved by means of an algorithm requiring the solution of a finite succession

of linear problems and terminating at a stationary point [7]. The algorithm can be applied as follows:

Bilinear Algorithm for Feature Selection

Initialization. Choose $\lambda \in [0,1)$, $\mu \in (0,1)$. Start with a point $(w^0, \gamma^0, y^0, z^0, v^0, r^0, u^0)$ belonging to the feasible set of (2.44).

1. Having $(w^k, \gamma^k, y^k, z^k, v^k, r^k, u^k)$ determine the next iterate by solving the two linear programs:

$$r^{k+1} \in \arg\min_{r}\{(1-\mu)\lambda e^T r - \mu {v^k}^T r \mid 0 \le r \le e\}$$

$(w^{k+1}, \gamma^{k+1}, y^{k+1}, z^{k+1}, v^{k+1}, r^{k+1}, u^{k+1})$ solution of

$$\min_{w,\gamma,y,z,v,u} \quad (1-\mu)(1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \mu(-{r^{k+1}}^T v + e^T u)$$

$$\begin{aligned} s.t. \quad & -Aw + e\gamma + e \le y \\ & Bw - e\gamma + e \le z \\ & -v \le w \le v \\ & y \ge 0, \quad z \ge 0 \\ & 0 \le u, \quad u - v \ge 0 \end{aligned}$$

2.Stop when:

$$(1-\mu)((1-\lambda)(\frac{e^T(y^{k+1}-y^k)}{m} + \frac{e^T(z^{k+1}-z^k)}{k}) + \lambda e^T(r^{k+1}-r^k)) +$$

$$+\mu(-{r^{k+1}}^T v^{k+1} + {r^k}^T v^k + e^T(u^{k+1}-u^k)) = 0$$

The parameter μ is chosen as the smallest one in $(0,1)$ such that the following complementarity condition holds at termination for the stationary point $(w^{\bar{k}}, \gamma^{\bar{k}}, y^{\bar{k}}, z^{\bar{k}}, v^{\bar{k}}, r^{\bar{k}}, u^{\bar{k}})$ found by means of the bilinear algorithm:

$$-{r^{\bar{k}}}^T v^{\bar{k}} + e^T u^{\bar{k}} = {r^{\bar{k}}}^T(u^{\bar{k}} - v^{\bar{k}}) + {u^{\bar{k}}}^T(r^{\bar{k}} + e) = 0 \ .$$

The following theorem shows that the algorithm terminates at a stationary point in a finite number of steps [7]:

Theorem 2.3.4. *For a fixed $\lambda \in [0,1)$ and $\mu \in (0,1)$, the bilinear algorithm for feature selection terminates in a finite number of steps at a stationary point $(w^{\bar{k}}, \gamma^{\bar{k}}, y^{\bar{k}}, z^{\bar{k}}, v^{\bar{k}}, r^{\bar{k}}, u^{\bar{k}})$ satisfying the following necessary optimality condition:*

$$(1-\mu)((1-\lambda)(\frac{e^T(y-y^k)}{m} + \frac{e^T(z-z^k)}{k}) + \lambda e^T(r-r^{k+1}))+$$

$$+\mu(-{r^{k+1}}^T(v-v^k) + (r-r^{k+1})^T v^k + e^T(u-u^k)) \geq 0$$

for all feasible points $(w, \gamma, y, z, v, r, u)$.

Chapter 3

Concave Programming for Minimizing the Zero-Norm over Polyhedral Sets

Given a non empty polyhedral set, we consider the problem of finding a vector belonging to it and having the minimum number of nonzero components, i.e., a feasible vector with minimum zero-norm. This combinatorial optimization problem is NP-Hard and arises in various fields such as machine learning, pattern recognition, signal processing. We introduce two new smooth approximations of the zero-norm function, where the approximating functions are separable and concave. We first formally prove the equivalence between the approximating problems and the original nonsmooth problem. To this aim, we preliminarly state in a general setting theoretical conditions sufficient to guarantee the equivalence between pairs of problems. Moreover we also define an effective and efficient version of the Frank-Wolfe algorithm for the minimization of concave separable functions over polyhedral sets in which variables which are null at an iteration remain zero for all the following ones, with significant savings in computational time, and we prove the global convergence of the method. Finally, we report the numerical results on test problems showing both the usefulness of the new concave formulations and the efficiency in terms of computational time of the implemented minimization algorithm.

3.1 The Zero-Norm Problem

In this section a formal description of the general problem of finding a vector belonging to a given polyhedral set and having as few nonzero components as possible will be given. As the problem is NP-complete, various approximate formulations, which can be used in order to make the problem tractable, will be described.

3.1.1 General Formulation

Given a polyhedral set, we consider the problem of finding a vector belonging to it and having the minimum number of nonzero components. Formally, the problem is

$$\begin{aligned} &\min_{x \in R^n} \|x\|_0 \\ &x \in P \end{aligned} \tag{3.1}$$

where $P \subset R^n$ is a non empty polyhedral set. This combinatorial optimization problem is NP-Hard [3], and arises in various fields such as machine learning (see, e.g., [40]), pattern recognition (see, e.g., [87]), signal processing (see, e.g., [26]).

3.1.2 Concave Approximations of the Zero-Norm

In order to make the problem tractable, the simplest approach can be that of replacing the zero-norm, which is a nonconvex discontinuous function, by the ℓ_1 norm thus obtaining the linear programming problem

$$\begin{aligned} &\min_{x,y \in R^n} \sum_{i=1}^{n} y_i \\ &x \in P \\ &-y_i \leq x_i \leq y_i \quad i = 1, \ldots, n, \end{aligned} \tag{3.2}$$

which can be efficiently solved even when the dimension of the problem is very large. Under suitable assumptions on the polyhedral set P (defined by an underdetermined linear system of equations) it is possible to prove that a solution of (3.1) can be obtained by solving (3.2) (see, e.g., [38]). However, these assumptions may be not satisfied in many cases, and some experiments concerning machine learning problems and reported in [10] show that a concave optimization-based approach performs better than that based on the employment of the ℓ_1 norm.
In order to illustrate the idea underlying the concave approach, we observe that the objective function of problem (3.1) can be written as follows

$$\|x\|_0 = \sum_{i=1}^{n} s(|x_i|)$$

where $s : R \to R^+$ is the *step function* such that $s(t) = 1$ for $t > 0$ and $s(t) = 0$ for $t \leq 0$. The nonlinear approach experimented in [10] was originally proposed in [57], and is based on the idea of replacing the discontinuous step function by a continuously differentiable concave function $v(t) = 1 - e^{-\alpha t}$, with $\alpha > 0$, thus obtaining a problem of the form

$$\begin{aligned} &\min_{x,y \in R^n} \sum_{i=1}^{n} (1 - e^{-\alpha y_i}) \\ &x \in P \\ &-y_i \leq x_i \leq y_i \quad i = 1, \ldots, n. \end{aligned} \tag{3.3}$$

The replacement of (3.1) by the smooth concave problem (3.3) is well-motivated see [57]) both from a theoretical and a computational point of view:

- for sufficiently high values of the parameter α there exists a vertex solution of (3.3) which provides a solution of the original problem (3.1), and in this sense the approximating problem (3.3) is equivalent to the given nonsmooth problem (3.1);

- the Frank-Wolfe algorithm [34] with unitary stepsize is guaranteed to converge to a vertex stationary point of (3.3) in a finite number of iterations (this convergence result was proved for a general class of concave

programming problems); thus the algorithm requires the solution of a finite sequence of linear programs for computing a stationary point of (3.3), and this may be quite advantageous from a computational point of view.

A similar concave optimization-based approach has been proposed in [90], where the idea is that of using the logarithm function instead of the step function, and this leads to a concave smooth problem of the form

$$\begin{aligned} &\min_{x,y \in R^n} \sum_{i=1}^{n} \ln(\epsilon + y_i) \\ &x \in P \\ &-y_i \leq x_i \leq y_i \quad i = 1, \ldots, n, \end{aligned} \tag{3.4}$$

with $0 < \epsilon \ll 1$. Formulation (3.4) is practically motivated by the fact that, due to the form of the logarithm function, it is better to increase one variable y_i while setting to zero another one rather than doing some compromise between both, and this should facilitate the computation of a sparse solution. A relation of (3.4) with the minimization of the zero-norm has been given in [90], and similarly to [57], the Frank-Wolfe algorithm with unitary stepsize has been applied to solve (3.4), and good computational results have been obtained.
In section 3.2 we derive new results on the equivalence, in a sense to be made more precise later, between a specific optimization problem and a parametrized family of problems. These results allow us to derive, within a general framework, results about two previously known families of approximations schemes for the zero-norm problem. Then we introduce two new families of approximation problems for which, thanks to the theory developed in section 3.2, it is possible to obtain convergence results. In section 3.3.6 after a brief review of the well known Frank-Wolfe method, we derive some new theoretical results having an important impact on the computational efficiency of the method when applied to concave optimization over polyhedra. In particular we prove that once the algorithm sets a variable to zero, it will not change this variable any more, thus allowing for a dimensionality reduction which greatly increments the speed of the procedure. We formally prove the global convergence of this modified

version of the Frank-Wolfe method. Finally, in section 3.4 we report the numerical results on test problems showing both the usefulness of the new concave formulations and the efficiency in terms of computational time of the implemented minimization algorithm.

3.2 Results on the Equivalence between Problems

In the first part of this section we state general conditions sufficient to ensure that a problem depending on a vector of parameters is equivalent to a given (unspecified) problem. Then we define two concave smooth problems depending on some parameters, and we show (using the general results) that these problems, for suitable values of their parameters, are equivalent to the original nonsmooth problem (3.1).

3.2.1 General Results

Consider the problem

$$\min_{x \in T} g(x) \tag{3.5}$$

where $g : R^n \to R$, $T \subseteq R^n$, and assume that it admits solutions. Let $G^\star$ be the set of such solutions.
Let $f(\cdot, u) : R^n \to R$ be a function depending on a vector of parameters $u \in U \subseteq R^m$. For any $u \in U$, consider the following problem

$$\min_{x \in T} f(x, u) \tag{3.6}$$

Assumption 3.2.1. *There exists a finite set $S^\star \subset R^n$ having the property that, for any $u \in U$, a point $x(u) \in S^\star$ exists such that*

$$x(u) \in \arg\min_{x \in T} f(x, u). \tag{3.7}$$

Theorem 3.2.1. *Let $\{u^k\} \subset U$ be an infinite sequence such that for every $\tilde{x} \in T \setminus G^\star$ and every $x^\star \in G^\star$, for all but finitely many indices k we have:*

$$f(\tilde{x}, u^k) > f(x^\star, u^k). \tag{3.8}$$

Then, under Assumption 3.2.1*, there exists a finite index* $\bar{k}$ *such that, for any* $k \geq \bar{k}$*, problem* (3.6)*, with* $u = u^k$*, has a solution* x^k *that also solves the original problem* (3.5)*.*

Proof. Let $x^\star \in G^\star$ be a solution of (3.5). In order to prove the thesis, by contradiction let us assume that there exists a subsequence $\{u^k\}_K$ such that, for all $k \in K$, denoting by x^k a point in $S^\star$ such that

$$x^k \in \arg\min_{x \in T} f(x, u^k), \tag{3.9}$$

we have

$$g(x^k) > g(x^\star). \tag{3.10}$$

Since $S^\star$ is finite, we can extract a further subsequence such that $x^k = \bar{x}$ for all $k \in K$, and hence, from (3.10), we can write

$$g(\bar{x}) > g(x^\star) \ . \tag{3.11}$$

Thus $\bar{x} \in T \setminus G^\star$ and, as a consequence,

$$f(\bar{x}, u^k) > f(x^\star, u^k) \tag{3.12}$$

for all k sufficiently large. But this contradicts (3.9). □

Using the above theorem we can state the next proposition.

Proposition 3.2.1. *Let* $\{u^k\} \subset U$ *be an infinite sequence such that*

$$\lim_{k \to \infty} \frac{f(\tilde{x}, u^k) - f(x^\star, u^k)}{a + |f(x^\star, u^k)|} = C \cdot [g(\tilde{x}) - g(x^\star)] \quad \forall \tilde{x} \in T, x^\star \in G^\star \tag{3.13}$$

with $a \geq 0$ *and* $C > 0$*. Under Assumption* 3.2.1*, there exists a finite index* $\bar{k}$ *such that, for any* $k \geq \bar{k}$*, problem* (3.6)*, with* $u = u^k$*, has a solution* x^k *that also solves the original problem* (3.5)*.*

Proof. If $\tilde{x} \in T \setminus G^\star$ then the right hand side in (3.13) is strictly positive. From this it follows that, for k large enough, also $f(\tilde{x}, u^k) - f(x^\star, u^k)$ will be strictly positive. □

As immediate consequence of Proposition 3.2.1 we have the following result.

Corollary 3.2.1. *Let $\{u^k\} \subset U$ be an infinite sequence such that*

$$\lim_{k\to\infty} f(x, u^k) = g(x) \quad \forall x \in T. \tag{3.14}$$

Under Assumption 3.2.1*, there exists a finite index $\bar{k}$ such that, for any $k \geq \bar{k}$, problem* (3.6)*, with $u = u^k$, has a solution x^k that also solves the original problem* (3.5)*.*

Under additional assumptions on the feasible set T and on the objective function $f(x, u)$ we can prove the following results.

Proposition 3.2.2. *Suppose that the feasible set T is a polyhedral set and that it admits a vertex. Assume that, for any $u \in U$, the objective function of* (3.6) *is concave, continuously differentiable, and bounded below on T. Let $\{u^k\} \subset U$ be an infinite sequence such that*

$$\lim_{k\to\infty} \frac{f(\tilde{x}, u^k) - f(x^\star, u^k)}{a + |f(x^\star, u^k)|} = C \cdot [g(\tilde{x}) - g(x^\star)] \quad \forall\ \tilde{x} \in T, x^\star \in G^\star \tag{3.15}$$

with $a \geq 0$ and $C > 0$. There exists a finite index $\bar{k}$ such that, for any $k \geq \bar{k}$, problem (3.6)*, with $u = u^k$, has a solution x^k that also solves the original problem* (3.5)*.*

Proof. Let $S^\star$ be the set of vertices of T. Since the objective function of (3.23) is concave, continuously differentiable, and bounded below on T, it follows that $S^\star$ satisfies Assumption 3.2.1, and hence the thesis follows from Proposition 3.2.1. □

Corollary 3.2.2. *Suppose that the feasible set T is a polyhedral set and that it admits a vertex. Assume that, for any $u \in U$, the objective function of* (3.6) *is concave, continuously differentiable, and bounded below on T. Let $\{u^k\} \subset U$ be an infinite sequence such that*

$$\lim_{k\to\infty} f(x, u^k) = g(x) \quad \forall x \in T. \tag{3.16}$$

There exists a finite index $\bar{k}$ such that, for any $k \geq \bar{k}$, problem (3.6)*, with $u = u^k$, has a solution x^k that also solves the original problem* (3.5)*.*

3.2.2 Concave Formulations Equivalent to the Zero-Norm Problem

For our convenience we rewrite the zero-norm problem as follows:

$$\begin{aligned} &\min_{x\in R^n, y\in R^n} \|y\|_0 \\ &x \in P \\ &-y_i \leq x_i \leq y_i \quad i = 1, \ldots, n \end{aligned} \tag{3.17}$$

We state the following assumption.

Assumption 3.2.2. *The polyhedral set P has at least a vertex.*

We denote by T the feasible set of (3.17), i.e.,

$$T = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in R^{2n} : x \in P,\ -y_i \leq x_i \leq y_i \quad i = 1, \ldots, n \right\}. \tag{3.18}$$

Assumption 3.2.2 implies that the polyhedral feasible set T has at least a vertex.
We introduce two concave formulations related to the ideas developed in [57] and [90], respectively.

Formulation I

$$\begin{aligned} &\min_{x\in R^n, y\in R^n} \sum_{i=1}^{n} (y_i + \epsilon)^p \\ &x \in P \\ &-y_i \leq x_i \leq y_i \quad i = 1, \ldots, n \end{aligned} \tag{3.19}$$

with $0 < p < 1$, and $0 < \epsilon$.

We observe that:

- given p and ϵ, the objective function is concave, continuously differentiable, bounded below on the feasible set set;
- $\lim_{p\to 0} \sum_{i=1}^{n} y_i^p = \|y\|_0$, so that the objective function can be view as a smooth approximation of the zero-norm.

The following proposition shows the equivalence between the approximating problem (3.19) and the zero-norm problem (3.17).

Proposition 3.2.3. *There exist values $\bar{p} > 0$, $\bar{\epsilon} > 0$, $\bar{\gamma} > 0$ such that, for any pair $(p, \epsilon)^T \in R^2_+$ and satisfying*

$$\begin{aligned} p &\leq \bar{p} \\ \epsilon &\leq \bar{\epsilon} \\ \epsilon^p &\leq \bar{\gamma}, \end{aligned} \tag{3.20}$$

problem (3.19) *has a vertex solution $(x(p,\epsilon), y(p,\epsilon))^T$ which is also solution of the original problem* (3.17).

Proof. In order to prove the thesis, assume by contradiction that there exists a sequence $\{(p^k, \epsilon^k, \gamma^k)^T\}$ converging to $(0,0,0)^T$, with

$$(\epsilon^k)^{p^k} \leq \gamma^k, \tag{3.21}$$

and such that, any vertex solution of (3.19), with $p = p^k$ and $\epsilon = \epsilon^k$, is not a solution of (3.17).

Set $z = (x,y)^T$, $u = (p,\epsilon)^T$, $g(z) = \|y\|_0$, $f(z,u) = \sum_{i=1}^{n}(y_i + \epsilon)^p$. Problems (3.17) and (3.19) can be written as follows

$$\min_{z \in T} g(z) \tag{3.22}$$

$$\min_{z \in T} f(z,u) \tag{3.23}$$

where T is defined in (3.18). From (3.21), as $\gamma^k \to 0$, we can write

$$\lim_{k \to \infty} (\epsilon^k)^{p^k} = 0. \tag{3.24}$$

Let $\{u^k\} = \left\{ \left(p^k, \epsilon^k\right)^T \right\}$ be the sequence convergent to $(0,0)^T$ and satisfying condition (3.24). Since for any $y \in R^+$ we have

$$\lim_{k \to \infty} \left(y_i + \epsilon^k\right)^{p^k} = \begin{cases} 1 & \text{if } y_i > 0 \\ 0 & \text{if } y_i = 0 \end{cases}$$

we obtain

$$\lim_{k\to\infty} f(z, u^k) = g(z) \quad \forall z \in T. \tag{3.25}$$

For any $u \in U$ the objective function of (3.23) is concave, continuously differentiable, and bounded below on T, so that, recalling (3.25), the assumptions of Corollary 3.2.2 hold and hence, for any k sufficiently large there exists a vertex solution $\left(x^k, y^k\right)^T$ which is also a solution of (3.22), in contradiction with our initial assumption. □

Formulation II

$$\begin{aligned} &\min_{x\in R^n, y\in R^n} -\sum_{i=1}^{n}(y_i+\epsilon)^{-p} \\ &x \in P \\ &-y_i \le x_i \le y_i \quad i = 1, \dots, n \end{aligned} \tag{3.26}$$

with $1 \le p$, and $0 < \epsilon$.

We observe that:

- given p and ϵ, the objective function is concave, continuously differentiable, bounded below on the feasible set set;
- similarly to the logarithm functions appearing in problem (3.4), the functions $-(y_i + \epsilon)^{-p}$ favor sparse vectors rather than points having many small nonzero components; indeed, when a variable is set to zero the decrease of the function is strong compared to the increase for a larger value of another variable;
- differently from the logarithm functions of problem (3.4), the functions $-(y_i + \epsilon)^{-p}$ are bounded above for positive values of the independent variables, and this may be a useful additional feature for finding sparse solutions.

The equivalence between problem (3.26) and the original problem (3.17) is formally proved below.

Proposition 3.2.4. *Assume that problem* (3.17) *admits a solution* $y^\star$ *such that* $\|y^\star\|_0 < n$. *There exists a value* $\bar{\epsilon} > 0$ *such that, for any* $\epsilon \in (0, \bar{\epsilon}]$, *problem* (3.26) *has a vertex solution* $(x(\epsilon), y(\epsilon))^T$ *which is also solution of the original problem* (3.17).

Proof. In order to prove the thesis, assume by contradiction that there exists a sequence $\{\epsilon^k\}$ converging to zero and such that, any vertex solution of (3.26), with $\epsilon = \epsilon^k$, is not a solution of (3.17).

Set $z = (x, y)^T$, $u = \epsilon$, $g(z) = \|y\|_0$, $f(z, u) = -\sum_{i=1}^{n}(y_i + u)^{-p}$. Problems (3.17) and (3.26) can be written as follows

$$\min_{z \in T} g(z) \tag{3.27}$$

$$\min_{z \in T} f(z, u) \tag{3.28}$$

where T is defined in (3.18). Let $\{u^k\} = \{\epsilon^k\}$ be the sequence convergent to 0. For any $z \in T$ we have

$$f(z,u) = -\sum_{i:y_i=0} u^{-p} - \sum_{i:y_i \neq 0} (y_i + u)^{-p} = -(n - \|y\|_0)u^{-p} - \sum_{i:y_i \neq 0} (y_i + u)^{-p},$$

so that, recalling that $u^k \to 0$ for $k \to \infty$, we can write for each $\tilde{z} \in T$ and for each $z^\star \in G^\star$ (being $G^\star$ the set of optimal solutions for problem (3.17))

$$\begin{aligned}
&\lim_{k\to\infty} \frac{f(\tilde{z}, u^k) - f(z^\star, u^k)}{|f(z^\star, u^k)|} \\
&= \lim_{k\to\infty} \frac{-(n - \|\tilde{y}\|_0)(u^k)^{-p} - \sum_{i:\tilde{y}_i \neq 0} (\tilde{y}_i + u^k)^{-p} + (n - \|y_i^\star\|_0)(u^k)^{-p} + \sum_{i:y_i^\star \neq 0} (y_i^\star + u^k)^{-p}}{|-(n - \|y^\star\|_0)(u^k)^{-p} - \sum_{i:y_i^\star \neq 0} (y_i^\star + u^k)^{-p}|} \\
&= \frac{\|\tilde{y}\|_0 - \|y^\star\|_0}{n - \|y^\star\|_0} = C \cdot [g(\tilde{z}) - g(z^\star)]
\end{aligned} \tag{3.29}$$

For any $u \in U = R_+$ the objective function of (3.28) is concave, continuously differentiable, and bounded below on T, so that, recalling (3.29), the assumptions of Proposition 3.2.2 hold (by setting a equal to zero) and hence, for any k sufficiently large there exists a vertex solution $\left(x^k, y^k\right)^T$ which is also a solution of (3.27), in contradiction with our initial assumption. □

We terminate the section by showing that the general results allow us to prove the equivalence between the smooth concave problems (3.3) and (3.4)

and the given nonsmooth problem (3.17). We remark that the equivalence between (3.3) and (3.17) was formally proved in [57], while the equivalence between (3.4) and (3.17) was not formally proved.

Proposition 3.2.5. *There exists a value $\bar{\alpha} > 0$ such that, for any $\alpha \geq \bar{\alpha}$, problem* (3.3) *has a vertex solution $(x(\alpha), y(\alpha))^T$ which is also solution of the original problem* (3.17)*.*

Proof. In order to prove the thesis, assume by contradiction that there exists a sequence $\{\alpha^k\}$ such that $\alpha^k \to \infty$, and any vertex solution of (3.3), with $\alpha = \alpha^k$, is not a solution of (3.17).

Set $z = (x, y)^T$, $u = \alpha$, $g(z) = \|y\|_0$, $f(z, u) = \sum_{i=1}^{n}(1 - e^{-uy_i})$ and consider the problems

$$\min_{z \in T} g(z) \tag{3.30}$$

$$\min_{z \in T} f(z, u) \tag{3.31}$$

where T is defined in (3.18). Let $\{u^k\} = \{\alpha^k\}$ be the sequence convergent to $+\infty$. Since for any $y \in R^+$ we have

$$\lim_{k \to \infty} \left(1 - e^{-u^k y_i}\right) = \begin{cases} 1 & \text{if } y_i > 0 \\ 0 & \text{if } y_i = 0 \end{cases}$$

we obtain

$$\lim_{k \to \infty} f(z, u^k) = g(z) \quad \forall z \in T. \tag{3.32}$$

For any $u \in U = R_+$ the objective function of (3.31) is concave, continuously differentiable, and bounded below on T, so that, recalling (3.32), the assumptions of Corollary 3.2.2 hold and hence, for any k sufficiently large there exists a vertex solution $\left(x^k, y^k\right)^T$ which is also a solution of (3.30), in contradiction with our initial assumption. □

Proposition 3.2.6. *Assume that problem* (3.17) *admits a solution $y^\star$ such that $\|y^\star\|_0 < n$. There exists a value $\bar{\epsilon} > 0$ such that, for any $\epsilon \in (0, \bar{\epsilon}]$, problem* (3.4) *has a vertex solution $(x(\epsilon), y(\epsilon))^T$ which is also solution of the original problem* (3.17)*.*

Proof. In order to prove the thesis, assume by contradiction that there exists a sequence $\{\epsilon^k\}$ such that $\epsilon^k \to 0$, and any vertex solution of (3.4), with $\epsilon = \epsilon^k$, is not a solution of (3.17).

Set $z = (x, y)^T$, $u = \epsilon$, $g(z) = \|y\|_0$, $f(z, u) = \sum_{i=1}^{n} \log(y_i + u)$ and consider the problems

$$\min_{z \in T} g(z) \tag{3.33}$$

$$\min_{z \in T} f(z, u) \tag{3.34}$$

where T is defined in (3.18). Let $\{u^k\} = \{\epsilon^k\}$ be the sequence convergent to 0. For any $z \in T$ we have

$$f(z,u) = \sum_{i:y_i=0} \log u + \sum_{i:y_i \neq 0} \log(y_i + u) = (n - \|y\|_0) \log u + \sum_{i:y_i \neq 0} \log(y_i + u),$$

so that, recalling that $u^k \to 0$ for $k \to \infty$, we can write for each $\tilde{z} \in T, z^\star \in G^\star$ (being $G^\star$ the set of optimal solutions for problem (3.17))

$$\begin{aligned}
&\lim_{k\to\infty} \frac{f(\tilde{z}, u^k) - f(z^\star, u^k)}{|f(z^\star, u^k)|} \\
&= \lim_{k\to\infty} \frac{(n - \|\tilde{y}\|_0) \log u^k + \sum_{i:\tilde{y}_i \neq 0} \log(\tilde{y}_i + u^k) - (n - \|y^\star\|_0) \log u^k - \sum_{i:y_i^\star \neq 0} \log(y_i^\star + u^k)}{|(n - \|y^\star\|_0) \log u^k + \sum_{i:y_i^\star \neq 0} \log(y_i^\star + u^k)|} \\
&= \frac{\|\tilde{y}\|_0 - \|y^\star\|_0}{n - \|y^\star\|_0} = C \cdot [g(\tilde{z}) - g(z^\star)]
\end{aligned} \tag{3.35}$$

For any $u \in U = R_+$ the objective function of (3.34) is concave, continuously differentiable, and bounded below on T, so that, recalling (3.35), the assumptions of Proposition 3.2.2 hold (by setting a equal to zero) and hence for any k sufficiently large there exists a vertex solution $\left(x^k, y^k\right)^T$ which is also a solution of (3.33), in contradiction with our initial assumption. □

For easier reference, in Figure 3.1 we report the graphs of the four concave functions we have analyzed.

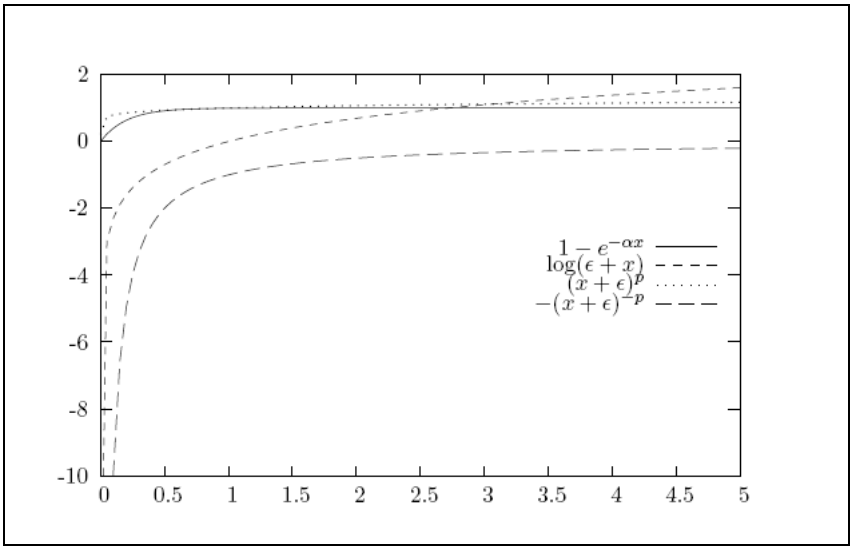

Figure 3.1: Graph of functions (3.3) with $\alpha = 5$, (3.4) with $\varepsilon = 10^{-9}$, (3.19) with $\varepsilon = 10^{-9}, p = 0.1$, (3.26) with $\varepsilon = 10^{-9}, p = 1$

3.3 The Frank-Wolfe Algorithm

The FrankWolfe algorithm is a well-known algorithm in operations research. It was originally proposed by Marguerite Frank and Phil Wolfe in 1956 as a procedure for solving quadratic programming problems with linear constraints. At each step the objective function is linearized and then a step is taken along a feasible descent direction.
We first describe the algorithm and give some results about its convergence to a stationary point. Then we analyze the case of minimizing a concave function over a polyhedral set. In the last part of this section, we propose a new efficient version of the Frank-Wolfe algorithm for minimizing a concave separable function over a polyhedral set.

3.3.1 A General Framework

Let us consider the constrained optimization problem:

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in C \end{array} \tag{3.36}$$

We make the following assumptions:

1. C is a non-empty compact convex set of R^n;
2. $f(x)$ is a continuously differentiable function over C.

Herein, we describe the Conditional Gradient Method for solving problem (3.36); this method is also known as the Frank-Wolfe Method [34].
Given a feasible point $x^k \in C$, a *feasible direction at* x^k that satisfies the descent condition $\nabla f(x^k)^T(x - x^k) < 0$ can be generated by solving the optimization problem:

$$\begin{array}{ll} \min & \nabla f(x^k)^T(x - x^k) \\ s.t. & x \in C \end{array} \tag{3.37}$$

As we assumed compactness of set C, a solution $\overline{x}^k \in C$ is guaranteed to exist by Weierstrass Theorem. Therefore $\overline{x}^k$ can be defined as follows:

$$\overline{x}^k = \arg\min_{x \in C} \nabla f(x^k)^T(x - x^k)$$

and two different cases can occur:

$$\nabla f(x^k)^T(\overline{x}^k - x^k) \begin{cases} = 0 & (a) \\ < 0 & (b) \end{cases} \tag{3.38}$$

When case (a) occurs, we have that minimum principle necessary optimality conditions are satisfied:

$$0 = \nabla f(x^k)^T(\overline{x}^k - x^k) \leq \nabla f(x^k)^T(x - x^k) \ \forall x \in C$$

and $\overline{x}^k$ is a stationary point.

When case (b) occurs, the algorithm does not stop at iteration k; a feasible direction at x^k, which is also a descent direction, can then be defined

$$d^k = \overline{x}^k - x^k$$

and a new feasible vector is generated according to

$$x_{k+1} = x^k + \alpha^k d^k$$

where $\alpha^k \in (0, 1]$.

3.3.2 Stepsize rules

We report some of the most popular rules for choosing the stepsize α^k:

1. **Minimization Rule**
   Here α^k is the value obtained by minimizing the function along the direction d^k,
   $$f(x^k + \alpha^k d^k) = \min f(x^k + \alpha d^k) \ .$$
   Minimization rule is typically implemented by means of line search algorithms. In practice, minimizing stepsize is not computed exactly, and it is replaced by a stepsize α^k satisfying some termination criteria.

2. **Armijo Rule**
   Here fixed scalars $\triangle^k$, δ and γ, with $\delta \in (0, 1)$ and $\gamma \in (0, 1/2)$, are chosen, and $\alpha^k = \delta^{m^k} \triangle^k$, where m^k is the first nonnegative integer m for which
   $$f(x^k + \alpha d^k) \leq f(x^k) + \gamma \alpha \nabla f(x^k)^T d^k \ .$$
   The stepsizes $\delta^m \triangle^k$, $m = 1, 2, \ldots$, are tried successively until the above inequality is satisfied for $m = m^k$.

3. **Constant Stepsize**
   Here a fixed stepsize
   $$\alpha^k = 1, \quad k = 0, 1, \ldots$$
   is used. The choice is not as simple or as restrictive as it may seem. In fact, a constant unit stepsize can always be used in a feasible direction method by defining appropriately the direction d^k.

3.3.3 Convergence Analysis

The following Proposition provides an analysis of convergence behavior of the Frank-Wolfe Algorithm.

Proposition 3.3.1. *Let $\{x^k\}$ be a sequence generated by the Frank-Wolfe Algorithm*

$$x^{k+1} = x^k + \alpha^k d^k.$$

Assume that method used for choosing stepsize α^k satisfies the following conditions:

(i) $f(x^{k+1}) < f(x^k)$, *with* $\nabla f(x^k) \neq 0$;

(ii) if $\nabla f(x^k) \neq 0 \ \ \forall\ k$, *then we have*

$$\lim_{k\to\infty} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0\ .$$

Then every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point.

Proof. As we assumed compactness of C, a limit point $\bar{x} \in C$ exists and the norm of vector d^k is bounded above

$$\|d^k\| = \|\bar{x}^k - x^k\| \leq \|\bar{x}^k\| + \|x^k\|\ .$$

We can now define a subsequence $\{x_k\}_K$ such that

$$\lim_{k\to\infty, k\in K} x^k = \bar{x}, \quad \lim_{k\to\infty, k\in K} d^k = \bar{d}\ .$$

By using hypothesis (ii), we obtain

$$\nabla f(\bar{x})^T \bar{d} = 0.$$

Let ^{k} be a direction generated by the Frank-Wolfe method; we have

$$\nabla f(x^k)^T d^k \leq \nabla f(x^k)^T (x - x^k),\ \forall\ x \in C\ .$$

By taking the limit as $k \in K$, $k \to \infty$,

$$0 = \nabla f(\bar{x})^T \bar{d} \leq \nabla f(\bar{x})^T (x - \bar{x}),\ \forall\ x \in C\ .$$

It follows that every limit point is a stationary point. □

3.3.4 Convergence Results with Concave Differentiable Functions

In this section we consider the problem (3.36), where we assume that:

1. C is a non-empty compact convex set of R^n;
2. $f(x)$ is a differentiable concave function over C.

The Frank-Wolfe Algorithm remains basically the same. It is possible to show convergence when:

a) a fixed stepsize $\alpha^k = s$ with $s \in (0, 1]$ is chosen;

b) a variable stepsize $\alpha^k \in (\bar{\alpha}, 1]$ with $\bar{\alpha} > 0$ is chosen.

The following proposition shows convergence of the Frank-Wolfe Algorithm with stepsize $\alpha^k = s$ and $s \in (0, 1]$ when a concave function is minimized over a compact convex set:

Proposition 3.3.2. *Let $\{x^k\}$ be a sequence generated by the Frank-Wolfe algorithm*

$$x^{k+1} = x^k + \alpha^k d^k \ ,$$

where a constant stepsize is chosen

$$\alpha^k = s, \ k = 0, 1, \ldots$$

with $s \in (0, 1)$. Then every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point.

Proof. we have from concavity of f:

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T (x^{k+1} - x^k) < f(x^k) \ .$$

Note that since $\{f(x^k)\}$ is monotonically decreasing, $\{f(x^k)\}$ either converges to a finite value or diverges to $-\infty$.
Let $\bar{x}$ be a limit point of $\{x^k\}$; since f is continuous $f(\bar{x})$ is a limit point of $\{f(x^k)\}$, so it follows that the entire sequence converges to $f(\bar{x})$. Therefore, we obtain

$$f(x^k) - f(x^{k+1}) \to 0$$

From concavity of f:

$$f(x^k) - f(x^{k+1}) \geq -\alpha^k \nabla f(x^k)^T d^k \ .$$

Since α^k is a constant stepsize, we have that

$$\nabla f(x^k)^T d^k \to 0 \ .$$

By Proposition 3.3.1 it follows that every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point. □

Let us consider the case when stepsize is variable over a finite interval. A convergence result similar to the ones shown for constant stepsize can be obtained:

Proposition 3.3.3. *Let $\{x^k\}$ be a sequence generated by the Frank-Wolfe Algorithm*

$$x^{k+1} = x^k + \alpha^k d^k \ ,$$

where a variable stepsize is chosen

$$\alpha^k \in (\bar{\alpha}, 1], \ k = 0, 1, \ldots$$

with $\bar{\alpha} > 0$. Then every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point.

Proof. It is a verbatim repetition of the proof of Proposition 3.3.2
From concavity of f, we have:

$$f(x^k) - f(x^{k+1}) \geq -\alpha^k \nabla f(x^k)^T d^k \ .$$

Since $\alpha^k \in (\bar{\alpha}, 1]$ and $\bar{\alpha} > 0$, we have that

$$\nabla f(x^k)^T d^k \to 0 \ .$$

By Proposition 3.3.1 it follows that every limit point $\bar{x}$ of $\{x^k\}$ is a stationary point. □

3.3.5 The Case of a Concave Function over a Polyhedral Convex Set

Let us consider the problem

$$\begin{array}{l} \min f(x) \\ x \in P \end{array} \tag{3.39}$$

where $P \subset R^n$ is a non empty polyhedral set, $f : R^n \to R$ is a concave, continuously differentiable function, bounded below on P.
Herein, we describe a modified version of Frank-Wolfe algorithm, which considers assumptions made above. By the concavity of f and its boundedness from below on P, we have:

$$-\infty < \inf_{x \in P} f(x) - f(x^k) \leq f(x) - f(x^k) \leq \nabla f(x^k)^T (x - x^k), \ \forall \ x \in P \ .$$

It follows that $\nabla f(x^k)^T (x - x^k)$ is bounded below on P for any $x^k \in \mathbb{R}^n$. Therefore, the linear program (3.37) is solvable, even if $x^k \notin P$. Thus, we have:

$$\overline{x}^k = \arg\min_{x \in C} \nabla f(x^k)^T (x - x^k) \ .$$

We note that because $x^k \in P$ for $k = 1, 2 \ldots$, it follows that only two cases can occur:

$$\nabla f(x^k)^T (\overline{x}^k - x^k) \begin{cases} = 0 & (a) \\ \\ < 0 & (b) \end{cases} .$$

When case (a) occurs, we have that minimum principle necessary optimality conditions are satisfied:

$$0 = \nabla f(x^k)^T (\overline{x}^k - x^k) \leq \nabla f(x^k)^T (x - x^k) \ \forall \ x \in P$$

and the algorithm terminates (provided $x^k \in P$, which may not be the case if $x^k = x^0 \notin P$).

When case (b) occurs, the algorithm does not stop at iteration k; a feasible direction at x^k, which is also a descent direction, can then be defined

$$d^k = \overline{x}^k - x^k$$

and we have from concavity of f:

$$f(\overline{x}^k) \leq f(x^k) + \nabla f(x^k)^T (\overline{x}^k - x^k) < f(x^k) \ .$$

Therefore, we obtain $f(\overline{x}^k) < f(x^k)$, and set $x^{k+1} = \overline{x}^k$, which is equivalent to choose a constant stepsize $\alpha^k = 1$.

The Frank-Wolfe algorithm with unitary stepsize can be described as follows.

Frank-Wolfe - Unitary Stepsize (FW1) Algorithm

1. Let $x^0 \in R^n$ be the starting point;

2. For $k = 0, 1, \ldots,$

   if $x^k \notin \arg\min_{x \in P} \nabla f(x^k)^T x$ then compute a vertex solution x^{k+1} of

$$\min_{x \in P} \nabla f(x^k)^T x \tag{3.40}$$

   else exit.

The algorithm involves only the solution of linear programming problems, and the following result, see [57], shows that the algorithm generates a finite sequence and that it terminates at a stationary point.

Proposition 3.3.4. *The Frank-Wolfe algorithm with unitary stepsize converges to a vertex stationary point of problem* (3.39) *in a finite number of iterations.*

Proof. By Corollary 1.2.3 it follows that f has its minimum at a vertex of the feasible region P. Since P has a finite number of vertices, $\{f(x^k)\}$ is strictly decreasing and f(x) is bounded below on P, a vector $x^s \in P$, such that

$$\nabla f(x^s)^T (x - x^s) \geq 0, \ \forall \ x \in P$$

must be generated after a finite number of steps. □

3.3.6 A New Version of the Frank-Wolfe Algorithm for Concave Separable Functions

Now consider the problem

$$\begin{aligned} &\min f(x) = \sum_{j=1}^{n} f_j(x_j) \\ &x \in P \\ &x_i \geq 0,\ i \in I \subseteq \{1, \ldots, n\} \end{aligned} \tag{3.41}$$

where $f_j : R \to R$, for $j = 1, \ldots, n$ are concave, continuously differentiable functions. We assume that f is bounded below on P.
We observe that problem (3.41) includes as particular cases the concave programming problems presented in the preceding section.
The next proposition shows that, under suitable conditions on the concave functions f_j, the algorithm does not change a nonnegative variable once that it has been fixed to zero.

Proposition 3.3.5. *Let $\{x^0, x^1, \ldots, x^h\}$ be any finite sequence generated by the Frank-Wolfe algorithm with unitary stepsize. There exists a value M such that, if $i \in I$ and $f_i'(0) \geq M$, then we have that*

$$x_i^k = 0 \quad \text{implies} \quad x_i^{k+1} = \ldots = x_i^h = 0.$$

Proof. At each iteration k of the Frank-Wolfe algorithm the linear problem to be solved is

$$\begin{aligned} &\min \sum_{j:x_j^k \neq 0} f_j'(x_j^k)\ x_j + \sum_{j \notin I : x_j^k = 0} f_j'(0) x_j + \sum_{j \in I : x_j^k = 0} f_j'(0) x_j \\ &x \in P \\ &x_i \geq 0,\ i \in I \subseteq \{1, \ldots, n\} \end{aligned} \tag{3.42}$$

Let x^{k+1} be a vertex solution of (3.42). For any $i \in I$ such that $x_i^k = 0$, by (ii) of Proposition 3.6.1 it follows that there exists a value M^k such that, if $f_i'(0) \geq M^k$, then we have $x_i^{k+1} = 0$. Thus, if $i \in I$, $x_i^k = 0$ and $f_i'(0) \geq M^k$, then we obtain

$$x_i^{k+1} = x_i^k = 0.$$

Letting

$$M = \max_{0 \leq k \leq h} \{M^k\},$$

and assuming

$$f_i'(0) \geq M$$

the thesis follows by induction. □

On the basis of Proposition 3.3.5 we can define the following version of the Frank-Wolfe algorithm with unitary stepsize, where the linear problems to be solved are of reduced dimension. We denote by Ω the feasible set of problem (3.41), i.e.,

$$\Omega = \{x \in R^n : x \in P, x_i \geq 0, i \in I\}.$$

Frank-Wolfe - Unitary Stepsize - Reduced Dimension (FW1-RD) Algorithm

1. Let $x^0 \in R^n$ be the starting point;

2. For $k = 0, 1, \ldots,$

   let $I^k = \{i \in I : x_i^k = 0\}$, $P^k = \{x \in \Omega : x_i = 0 \ \forall i \in I^k\}$

   if $x^k \notin \arg\min_{x \in P^k} \nabla f(x^k)^T x$ then compute a vertex solution x^{k+1} of

   $$\min_{x \in P^k} \nabla f(x^k)^T x \tag{3.43}$$

   else exit.

Note that the linear programming problem (3.43) is equivalent to a linear problem of dimension $n - |I^k|$, and that $I^k \subseteq I^{k+1}$, so that the linear problems to be solved are of nonincreasing dimensions. This yields obvious advantages (shown in the next section) in terms of computational time. We formally prove the finite convergence of the algorithm at a stationary point.

Proposition 3.3.6. *There exists a value M such that, if $f_j'(0) \geq M$ for $j = 1, \ldots, n$, then Algorithm* FW1-RD *converges to a vertex stationary point of problem* (3.41) *in a finite number of iterations.*

Proof. Since f is a concave differentiable function and is bounded below on Ω, we can write

$$-\infty < \inf_{x\in\Omega} \; f(x) - f(x^k) \leq f(x) - f(x^k) \leq \nabla f(x^k)^T(x - x^k), \; \forall \; x \in \Omega.$$

Therefore, as $P^k \subseteq \Omega$, it follows that $\nabla f(x^k)^T x$ is bounded below on the polyhedral set P^k and hence problem (3.43) admits a vertex solution x^{k+1}, so that Step 2 is well-defined.
We observe that the number of polyhedral sets P^k is finite and hence the number of vertex points generated by the algorithm is finite.
Now we show that $x^k \notin \arg\min_{x\in P^k} \; \nabla f(x^k)^T x$ implies $f(x^{k+1}) < f(x^k)$. Indeed, in this case we have $\nabla f(x^k)^T(x^{k+1} - x^k) < 0$, and hence, recalling the assumptions on f, we can write

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) < f(x^k). \tag{3.44}$$

Since the number of points visited by the algorithm is finite, from (3.44) we get that the algorithm terminates in a finite number k of iterations with a point $x^k \in \arg\min_{x\in P^k} \; \nabla f(x^k)^T x$. We prove that x^k is a stationary point. Indeed, x^k is a vertex solution of

$$\begin{array}{l} \min \displaystyle\sum_{j:x_j^k\neq 0} f_j'(x_j^k)\; x_j + \sum_{j\notin I^k : x_j^k = 0} f_j'(0)\; x_j \\ x \in \Omega \\ x_i = 0, \; i \in I^k \end{array}$$

and by (i) of Proposition 3.6.1 it follows that there exists a value M such that, if $f_j^{'}(0) \geq M$ then x^k is a solution of

$$\begin{array}{l} \min \displaystyle\sum_{j:x_j^k\neq 0} f_j'(x_j^k)\; x_j + \sum_{j\notin I^k : x_j^k = 0} f_j'(0)\; x_j + \sum_{j\in I^k : x_j^k = 0} f_j'(0) x_j \\ x \in \Omega \end{array} \tag{3.45}$$

Therefore we have

$$\nabla f(x^k)^T x^k \leq \nabla f(x^k)^T x \quad \forall x \in \Omega,$$

and this proves that x^k is a stationary point of problem (3.41). □

Concerning the separable concave objective functions of problems (3.3), (3.4), (3.19), (3.26), we have for $j = 1, \ldots, n$

- $f_j(y_j; \alpha) = 1 - e^{-\alpha y_j}$ and $f_j^{'}(0) = \alpha$;
- $f_j(y_j; \epsilon) = \ln(y_j + \epsilon)$ and $f_j^{'}(0) = 1/\epsilon$;
- $f_j(y_j; \epsilon, p) = (y_j + \epsilon)^p$ and $f_j^{'}(0) = p(\epsilon)^{p-1}$ with $0 < p < 1$;
- $f_j(y_j; \epsilon, p) = -(y_j + \epsilon)^{-p}$ and $f_j^{'}(0) = p(\epsilon)^{-p-1}$ with $1 \leq p$;

Therefore, the assumption of Proposition 3.3.6 holds for suitable values of the parameters of the above concave functions, so that Algorithm FW1-RD can be applied to solve problems (3.3), (3.4), (3.19), (3.26). The results obtained on computational experiments will be presented in the next section.

3.4 Computational experiments

In order to show both the usefulness of the new concave formulations and the efficiency in terms of computational time of the new version of the Frank-Wolfe algorithm for concave separable functions we report in this section numerical results on various test problems.

3.4.1 Feature Selection Problems

In our computational experiments we have considered feature selection problems of linear classification models. Given two linearly separable sets of points in a $n-$dimensional feature space, the problem is that of finding the hyperplane that separates the two sets and utilizes as few of the feature as possible. Formally, given two linearly separable sets

$$S1 = \{u^i \in R^n,\ i = 1, \ldots, p\} \qquad S2 = \{v^j \in R^n,\ j = 1, \ldots, q\},$$

the problem is

$$\begin{array}{l} \min\limits_{w \in R^n, \theta \in R} \|w\|_0 \\ \\ w^T u^i + \theta \geq 1 \quad i = 1, \ldots, p \\ w^T v^j + \theta \leq -1 \quad j = 1, \ldots, q \end{array} \tag{3.46}$$

Thus, according to the notation adopted in the chapter, the problems we used in our experiments take the form

$$\begin{aligned} &\min_{x \in R^n, \theta \in R} \|x\|_0 \\ &A \begin{pmatrix} x \\ \theta \end{pmatrix} \geq e \end{aligned} \tag{3.47}$$

where $A \in R^{m \times (n+1)}$, $e \in R^m$ is a vector of ones.
We remark that the aim of the experiments has been that of evaluating the effectiveness of various formulations in finding sparse vectors (possibly the sparsest vectors) belonging to polyhedral sets. As said above, the class of problems (3.47) considered in the experimentation derives from a specific machine learning problem, that is the feature selection problem of linear classifier models. Such a machine learning problem would require to investigate other important issues concerning, for instance, the generalization capability of the linear classifier model determined. This aspect will not be considered here, since it deserves particular attention and will be the object of the next chapter.
We observe that the mixed integer linear programming problem

$$\begin{aligned} &\min_{x \in R^n, \theta \in R, \delta \in \{0,1\}^n} \sum_{i=1}^{n} \delta_i \\ &A \begin{pmatrix} x \\ \theta \end{pmatrix} \geq e \\ &-M\delta_i \leq x_i \leq M\delta_i \quad i = 1, \ldots, n \\ &\delta_i \in \{0, 1\} \quad i = 1, \ldots, n \end{aligned} \tag{3.48}$$

is equivalent to problem (3.47) for sufficiently high values of M. Thus, for relatively small dimensional test problems we can determine an optimal solution of (3.47) by solving (3.48) by means of an exact method.

3.4.2 Test problems

P-random. For several values of n and m we randomly generated the matrix A. In particular, each instance of (3.47) was generated as follows: we

randomly defined an hyperplane in a n_1-dimensional space, and we randomly determined m_1 points u^i in an half-space (corresponding to labels +1) and other m_2 points v^j in the other half-space (corresponding to labels -1), for a total number of $m = m_1 + m_2$ points. We added to each of these vectors a number n_2 of random components, thus obtaining two linearly separable sets, of cardinality m_1 and m_2 respectively, in the space of dimension $n = n_1 + n_2$. In this way, the resulting problem (3.47) had the optimal objective function value less or equal than $n_1 < n$.

Colon cancer [2]. The colon cancer dataset contains 22 normal and 40 colon cancer tissues described by 2000 genes expression values extracted from DNA microarray data.

Catalysis. In Catalysis Dataset targets represent the presence (or absence) of catalytic activity of a protein. Inputs are gene expression levels of the genes encoding those proteins. This version of the database was prepared for the Pascal 2004 Evaluating Predictive Uncertainty Challenge. The data are available at
`http://predict.kyb.tuebingen.mpg.de/pages/home.php`.

Nova. This dataset consists of 1754 articles collected from 20 different newsgroups. There are 499 articles related to politics or religion topics and 1255 articles related to other topics. Input variables use a bag-of-words representation with a vocabulary of approximately 17000 words. This version of the database was prepared for the WCCI 2006 challenge on performance prediction. The data are available at
`http://clopinet.com/isabelle/Projects/modelselect/`.

3.4.3 Experiments and Implementation Details

For each problem we performed experiments using:

- formulation (3.2), denoted by ℓ_1;
- formulation (3.3), denoted by *exp*, with $\alpha = 5$;
- formulation (3.4), denoted by *log*, with $\epsilon = 10^{-9}$;
- formulation (3.19), denoted by *Formulation I*, with $\epsilon = 10^{-9}$ and $p = 0.001$;

- formulation (3.26), denoted *Formulation II*, with $\epsilon = 10^{-9}$ and $p = 1$.

We applied the Frank-Wolfe algorithm for solving the instances of (3.3), while we used Algorithm FW1-RD, that is the version of the Frank-Wolfe algorithm presented in the preceding section, for solving problems (3.4), (3.19), (3.26). The reason for which we employed the standard version of the Frank-Wolfe algorithm, instead of Algorithm FW1-RD, for solving the instances of (3.3) is that the chosen value $\alpha = 5$, suggested in [10], did not seem sufficiently high to ensure that the assumptions of Proposition 3.3.6 were satisfied. We used 100 random initial points for all the problems.
The instances of problem (3.48) were solved by means of CPLEX (8.0). Algorithms FW1 and FW1-RD were implemented in C using GLPK (4.9) as solver of the linear programming problems. The experiments were carried out on Intel Pentium 4 3.2 GHz 512 MB RAM.

3.4.4 Results

The results obtained on P-random problems and on the other three test problems are shown in tables 1 and 2 respectively, where we report

- the number m of constraints, the number n of variables;
- for formulation ℓ_1, the zero-norm of the optimal solution attained;
- for each nonlinear concave formulation:
  - the average of the zero-norm value of the stationary points determined;
  - the best zero-norm value of those stationary points;
  - percentage of runs where the best zero-norm value was attained.

In Table 1, which concerns relatively small dimensional problems, we also report the optimal value $\|x^*\|_0$ determined by solving (3.48).
From Table 1 we can observe that the best results are obtained by means of *Formulation II*. Indeed, in seven problems over ten, a simple multi-start strategy applied to *Formulation II* allowed us to attain the certificated optimal solution. We may note that the results obtained by means of formulations *log* and *Formulation I* are comparable, and clearly better than those corresponding to formulations ℓ_1 and *exp*.

The results obtained on problems Colon cancer, Catalysis, and Nova are reported in Table 2, where we can observe that the multi-start strategy applied to the nonlinear concave formulations performed clearly better than the approach based on the minimization of the ℓ_1 norm. Furthermore, we can note that the best results on problem Colon Cancer were obtained by *exp* and *Formulation I*, the best results on problem Catalysis were obtained by *Formulation II*, while the best results on problem Nova were obtained by *log* and *Formulation I*.

Summarizing, the computational experiments confirm the validity of the concave-based approach for the minimization of the zero-norm over a polyhedral set, and show that the concave formulations here proposed are valid alternatives to known formulations. Indeed, *Formulation I* and *Formulation II* attained the best results in 3 tests over 13 and 9 tests over 13 respectively. We remark that a wider availability of efficient formulations is important since it can facilitate the search of sparse enough solutions for different classes of problems.

Finally, in order to assess the differences in terms of computational time between the standard Frank-Wolfe (FW1) algorithm and the version of the algorithm presented in the preceding section and denoted by Algorithm FW1-RD, we report in Table 3 the results obtained by the two algorithms on the three benchmark problems using *Formulation I*. As we might expect, the differences are remarkable and show the usefulness of Algorithm FW1-RD. Further experiments not here reported and performed using the other concave formulations point out the same differences between the two algorithms in terms of computational time. In all the tests we did not detect differences between the two algorithms in terms of computed solution.

P-random	m	n	$\|\|x^*\|\|_0$	l_1	**Exp**	**Log**	**Form. I**	**Form. II**
1	20	10	2	3	3.0/3/100	2.1/**2**/93	2.1/**2**/93	2.0/**2**/97
2	20	10	3	4	4.0/4/100	3.6/**3**/66	3.6/**3**/66	3.9/**3**/45
3	40	20	3	8	8.0/8/100	6.3/4/9	6.2/4/9	5.7/**3**/6
4	40	20	4	10	10.0/10/100	7.7/5/1	7.7/5/1	6.5/5/15
5	60	30	6	12	12.0/12/100	10.0/8/2	10.0/8/2	8.8/**6**/3
6	60	30	7	14	13.9/13/3	10.9/8/1	11.0/8/1	9.7/**7**/6
7	80	40	6	14	14.0/14/100	10.4/7/1	10.4/7/1	9.4/**6**/3
8	80	40	9	24	23.4/22/14	16.4/12/1	16.4/12/1	14.1/11/4
9	100	50	8	19	19.0/19/100	15.1/11/1	15.2/11/1	13.0/**8**/2
10	100	50	10	28	28.0/28/100	18.5/14/3	18.5/14/2	16.0/12/6

Table 3.1: Comparison on P-random problems (average zero-norm value/best zero-norm value/percentage of best values attained).

Problem	m	n	l_1	**Exp**	**Log**	**Form. I**	**Form. II**
Colon Cancer	62	2000	57	8.5/**6**/10	13.8/7/1	13.7/7/1	9.4/**6**/3
Catalysis	873	617	422	199.3/184/1	222.1/201/1	221.0/201/1	189.8/**173**/1
Nova	1754	16969	448	168.5/147/2	127.0/**105**/1	126.7/**105**/1	131.9/114/1

Table 3.2: Comparison on three benchmark problems (average zero-norm value/best zero-norm value/percentage of best values attained).

3.5 Conclusions

In this chapter we have considered the general hard problem of minimizing the zero-norm over polyhedral sets, which arises in different important fields, such as machine learning and signal processing. Following the concave optimization-based approach, we have proposed two new smooth concave formulations and we have formally proved the equivalence of these and other formulations with the original nonsmooth problem. The main contributions of this work are both theoretical and computational. From the theoretical point of view, we have been able to introduce some general results on approximability for concave optimization problems and we obtained an

Problem	FW1	FW1-RD
Colon Cancer	225	24
Catalysis	2776	465
Nova	10448	1003

Table 3.3: Comparison using Formulation I between the two versions of the Frank-Wolfe algorithm in terms of CPU-time (seconds).

important characterization of the behaviour of the Frank-Wolfe algorithm which has, as we could confirm in computational experiments, a dramatic influence on the efficiency of the method. The computational evidence we report suggests a speed-up in the range 5 to 10 when using the variable fixing variant of the Franke-Wolfe method in place of the traditional one. This very high speed-up might prove to be extremely beneficial when multiple runs of the algorithm are performed, e.g. in a Multistart method. Apart from the great improvement in efficiency, the computational experiments also evidenced that the new formulations are valid alternatives to known formulations, as in most cases they allowed us to compute highly sparse solutions. We remark that a wider availability of efficient formulations is important since it can facilitate the search of sparse enough solutions for different classes of problems.

3.6 Appendix

We report here a known result (and its proof) that we have used to derive some new convergence results of the Frank-Wolfe method and of the modified version we have presented.

Proposition 3.6.1. *Consider the linear programming problems*

$$\begin{array}{l} \min c^T x \\ Ax \geq b \\ Hx = d \end{array} \tag{3.49}$$

$$\begin{array}{l} \min c^T x + Me^T z \\ Ax + Qz \geq b \\ Hx + Sz = d \\ z \geq 0 \end{array} \tag{3.50}$$

where $c \in R^n$, $e \in R^{n_z}$ *is a vector of ones,* $b \in R^m$, $d \in R^p$, $A \in R^{m \times n}$, $H \in R^{p \times n}$, $Q \in R^{m \times n_z}$, $S \in R^{p \times n_z}$. *Assume that problem* (3.49) *admits a solution* $x^\star$. *Then, there exists a value* M_0 *such that for all* $M \geq M_0$ *we have that*

(i) *the vector* $(x^\star, 0)^T$ *is a solution of* (3.50)*;*

(ii) *if* $(\bar{x}, \bar{z})^T$ *is a solution of* (3.50)*, then* $\bar{z} = 0$ *and* $\bar{x}$ *is a solution of* (3.49)*.*

Proof. (i) Since $x^\star$ is a solution of problem (3.49) we have that the dual problem

$$\begin{array}{l} \max b^T \lambda + d^T \mu \\ A^T \lambda + H^T \mu = c \\ \lambda \geq 0 \end{array} \tag{3.51}$$

admits a solution $(\lambda^\star, \mu^\star)^T \in R^{m+p}$ and we have

$$c^T x^\star = b^T \lambda^\star + d^T \mu^\star. \tag{3.52}$$

Now consider problem (3.50) and its dual

$$\begin{aligned} &\max b^T\lambda + d^T\mu \\ &\\ &A^T\lambda + H^T\mu = c \\ &Q^T\lambda + S^T\mu \leq Me \\ &\lambda \geq 0 \end{aligned} \tag{3.53}$$

The vector $(x^\star, 0)^T$ is a feasible point for (3.50), and for M sufficiently large the vector $(\lambda^\star, \mu^\star)^T$ is a feasible point for (3.53). Thus, from (3.52) the assertion is proved. Furthermore, we can also conclude that $(\lambda^\star, \mu^\star)^T$ is a solution of (3.53) for M sufficiently large.

(ii) By contradiction let us assume that there exist a sequence of positive scalars $\{M^k\}$, with $M^k \to \infty$ for $k \to \infty$, and a corresponding sequence of vectors $\{(x^k, z^k)^T\}$ such that $z^k \neq 0$, and $(x^k, z^k)^T$ is solution of (3.50) when $M = M^k$. We can then define an infinite subset K such that, for all $k \in K$

- we have $z_i^k > 0$ for some index $i \in \{1, \ldots, n_z\}$;
- the vector $(\lambda^\star, \mu^\star)^T$ is a solution of (3.53) when $M = M^k$.

Then, using the complementarity conditions we can write

$$\left(e_i^T Q^T \lambda^\star + e_i^T S^T \mu^\star - M^k\right) = 0 \qquad \forall k \in K,$$

which contradicts the fact that $M^k \to \infty$. □

Chapter 4

Concave Programming Methods for Feature Selection and Sparse Approximation of Signals

In this chapter new methods for feature selection and sparse approximation of signals are described. Since both of these problems can be modelled as a search for a sparse solution to a linear system, new approaches can be developed using the ideas described in Chapter 3.
Concerning feature selection problems, which are of great importance in machine learning, a new algorithm has been described. It combines the concave optimization-based approach (to eliminate irrelevant features) with linear Support Vector Machines (to guarantee predictive capability). An extensive computational experience is performed on several datasets in order to show the efficiency of the proposed feature selection technique.
A concave approach for finding sparse representations of noisy signals (based on FW1-RD algorithm) is also proposed. In order to show both the usefulness of the proposed concave formulations and the efficiency of the new method, various problems concerning sparse approximation of noisy signals are solved.

4.1 Feature Selection Combining Linear Support Vector Machines and Concave Optimization

Feature selection is a crucial task in a wide range of machine learning problems. Given an unknown process that generates data represented by vectors of an Euclidean space and corresponding label values, the feature selection problem consists in selecting a subset of relevant features (components of the vectors) that permit to build a reliable model of the underlying process. As we already said in Chapter 2, feature selection can have various motivations: improving the generalization performance of the predictive model, reducing the computational time to construct the model, better understanding of the underlying process.
In this section we introduce a new method for feature selection combining linear SVMs and concave minimization. We first give a brief overview of the problem we deal with. Then, we describe our feature selection approach. Finally, we show results obtained on various datasets.

4.1.1 Feature Selection for Linear Classification Models

We focus on feature selection problems of two-class linear models, which are common in several applications (see, e.g., [16]). We assume that the unknown process generates vector data belonging to two classes, and we suppose that the process can be modelled by a linear machine defined by a decision function of the form

$$y(x) = sgn(w^T x + b), \tag{4.1}$$

where $x \in R^n$ is the input vector, $w \in R^n$ is the vector of weights, $b \in R$ is the threshold, $sgn : R \to \{-1, 1\}$ is the *sign function* such that $s(t) = 1$ for $t \geq 0$ and $s(t) = -1$ for $t < 0$. We denote by $H(w, b)$ the separating hyperplane associated to the decision function (4.1). In order to model the process, a finite set (the training set) of data

$$TS = \{(x^i, y^i),\ x^i \in R^n,\ y^i \in \{-1, 1\},\ i = 1, \ldots, N\}$$

is available, where the label y^i denotes the class of the vector x^i.
We are interested in finding the relevant features of the input space R^n,

namely we want to reduce the dimensionality of the data by selecting those variables that permits to model the unknown process by a linear classifier in a subspace of R^n. Our motivation lies in the fact that, as said above, the detection of the relevant features provides a better understanding of the underlying phenomenon, and this can be of great interest in important fields, such as medicine and biology. For instance, feature selection in data concerning healthy patients and patients affected by a given pathology may help to better understand the considered pathology from a medical point of view.

4.1.2 A brief review of Linear Support Vector Machines

Consider the training set

$$TS = \{(x^i, y^i),\ x^i \in R^n,\ y^i \in \{-1, 1\},\ i = 1, \ldots, N\}$$

and assume it is linearly separable, that is, there exists a separating hyperplane

$$H(w, b) = \{x \in R^n : w^T x + b = 0\}$$

such that

$$\begin{aligned} w^T x^i + b \geq 1 \quad & \forall x^i :\ y^i = 1 \\ w^T x^i + b \leq -1 \quad & \forall x^i :\ y^i = -1 \end{aligned} \tag{4.2}$$

The *margin* $\rho(w, b)$ of a separating hyperplane $H(w, b)$ is the distance from the hyperplane to the closest training points, i.e.,

$$\rho(w, b) = \min_{x^i, i=1,\ldots,N} \frac{|w^T x^i + b|}{\|w\|}.$$

Support Vector Machine approach picks out, among the linear classifiers, the optimum separating hyperplane (i.e. the hyperplane having maximum margin). The basic training principle of SVM, motivated by the statistical learning theory [86], is that the expected classification error for unseen test samples is minimized, so that SVM defines a good predictive model.

The optimum hyperplane can be determined by solving the following quadratic programming

$$\begin{aligned} & \min_{w \in R^n, b \in R} \ \frac{1}{2}\|w\|^2 \\ & y^i(w^T x^i + b) \geq 1 \quad i = 1, \ldots, N. \end{aligned} \tag{4.3}$$

In the case that the training set is not linearly separable, the system of inequalities (4.2) does not admit solution. By introducing slack variables ξ^i, for $i = 1, \ldots, N$, the SVM classifier is determined by solving

$$\begin{aligned} &\min_{w \in W, b \in R, \xi \in R^N} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi^i \\ &y^i(w^T x^i + b) \geq 1 - \xi^i \quad i = 1, \ldots, N \\ &\xi^i \geq 0 \quad i = 1, \ldots, N, \end{aligned} \tag{4.4}$$

where the term $\sum_{i=1}^{N}\xi^i$ is an upper bound on the training error. The regularization parameter $C > 0$ trades off margin size and training error, and is usually determined by standard cross-validation tuning procedures. More specifically, discrete values of C are defined, and for each value a $k-$fold cross validation on the training set is performed. The value of C which gives the lowest cross validation error rate is selected. Recently, some algorithms have been developed for efficiently solve large-scale problems of the form (4.4) (see [31] and the references therein).

We observe that SVM provides a good linear classifier, but we can not expect that the obtained separating hyperplane utilizes as few of the features as possible, since the minimization of the Euclidean norm favorites solutions with many small nonzero components. We will induce sparsity by suitably exploiting the concave approach described in Chapter 3.

4.1.3 A new Algorithm for Feature Selection

In feature selection two objectives have to be considered:

(a) the goodness (to be maximized) of the data fitting of the linear classifier modelling the process;

(b) the number of input variables (to be minimized) of the classifier.

A reasonable and often adopted approach is that of formalizing the feature selection problem as an optimization problem whose objective function consists of two terms, the first one related to the error on the training data (its

inverse gives a measure of the goodness of the training data fitting), the second one to the sparsity of the solution to be determined. The second term is also introduced to prevent *overtraining* (the machine learns too specifically the training data), a phenomenon that could lead to a linear classifier with poor generalization capabilities. The general formulation (see Chapter 2) takes the form

$$\begin{aligned} \min_{w,b,\xi} \quad & (1-\lambda) f(\xi) + \lambda g(w) \\ \text{s.t.} \quad & y^i(w^T x^i + b) \geq 1 - \xi^i \quad i = 1, \ldots, N \\ & \xi^i \geq 0, \quad i = 1, \ldots, N, \end{aligned} \tag{4.5}$$

where $\lambda \in [0, 1)$ is a parameter, ξ^i for $i = 1, \ldots, N$ are slack variables, the first term f in the objective function is a measure of the classification error on the training data, the second term g penalizes nonzero components of w. In this formulation, the term f is the sum (possibly weighted) of the slack variables ξ^i, and hence measures the (average) training error. Concerning the term g, a good choice is that of using a concave smooth approximation of the so called zero-norm of w, denoted by $\|w\|_0$, and indicating the number of nonzero components of w. The concave-based approach performs well as feature selection technique, that is, the classifiers obtained select a small number of features, still granting good prediction accuracy.
In this section we present a feature selection strategy that combines SVM and the concave optimization-based approach. Differently from the approach described above, where the two objectives (a) and (b) are embedded in the objective function of formulation (4.5) and assessed by means of parameter λ, we sequentially and iteratively operate on the two separated objectives (a) and (b). Our strategy is essentially a two-step strategy, with the two steps described as follows:

1) *LSVM step*: a good linear classifier (according to the statistical learning theory [86]) is computed in a given subspace (which is the original input space at the beginning) by means of linear SVM;

2) *Concave optimization step*: a new sparse linear classifier, not "too far" from SVM solution, is determined using concave optimization.

In practice, starting from the SVM solution in a given subspace, we perform a single iteration of the Frank-Wolfe method applied to a problem whose ob-

jective function is a concave approximation of the zero norm (that favorites sparse solutions), and the constraints are the linear inequalities related to the training data correctly classified by SVM. Again, the nonzero components of the computed separating hyperplane define a subspace of lower dimension, and we repeat the two-phase procedure in this subspace.
In the sequel we refer to an unspecified concave function of the form

$$F(z) = \sum_{i=1}^{n} f_i(z_i),$$

where $f_i : R \to R$ are concave smooth functions, aimed to approximate the zero-norm problem. Specific concave functions that can be practically employed are the objective functions of formulations (3.3)-(3.26).
Formally, the Feature Selection algorithm, based on the combination of SVM with Concave Programming, is reported below.

Algorithm FS-SVMCP

Let $W = R^n$.

1. Solve SVM problem

$$\begin{aligned} &\min_{w \in W, b \in R, \xi \in R^N} \frac{1}{2}\|w\|^2 + C\sum_{i \in I} \xi^i \\ &y^i(w^T x^i + b) \geq 1 - \xi^i \quad i = 1, \ldots, N \\ &\xi^i \geq 0 \quad i = 1, \ldots, N \end{aligned} \tag{4.6}$$

   and let $(w^\star, b^\star, \xi^\star)$ be the solution obtained.

2. Set $I = \{1, \ldots, N\}/\{i : (\xi^\star)^i \geq 1\}$, $z_i^\star = |w_i^\star|$ for $i = 1, \ldots, n$, and compute a vertex solution $(\hat{w}, \hat{b}, \hat{z})^T$ of

$$\begin{aligned} &\min_{w \in W, b \in R, z \in R^N} \nabla F(z^\star)^T z \\ &y^i(w^T x^i + b) \geq 1 \quad i \in I \\ &-z_h \leq w_h \leq z_h \quad h = 1, \ldots, n \end{aligned} \tag{4.7}$$

If $\|\hat{w}\|_0 < \|w^\star\|_0$ then remove features corresponding to null components of $\hat{w}$, i.e., set

$$W = \{w \in R^n : w_h = 0,\ \forall h : \hat{w}_h = 0\},$$

and go to step 1, otherwise let $H(w^\star, b^\star)$ be the separating hyperplane and exit.

At Step 1 a good classifier in a given subspace of R^n (defined by W) is determined by means of the standard SVM technique (as said in Section 2, the value of parameter C can be computed by means of a cross-validation tuning procedure). The misclassified training points x^i are those corresponding to slack variables $(\xi^*)^i$ greater than 1. The index set I, defined at Step 2, identifies the well-classified training points.
At Step 2, starting from the point $(w^\star, b^\star, z^\star)^T$ provided by the SVM classifier, a single iteration of FW1 Algorithm is applied to the problem

$$\begin{aligned} &\min_{w \in W, b \in R, z \in R^N} \quad F(z) \\ &y^i(w^T x^i + b) \geq 1 \quad i \in I \\ &-z_h \leq w_h \leq z_h \quad h = 1, \ldots, n, \end{aligned} \tag{4.8}$$

where, as said above, the set I identifies the inequalities corresponding to the training points well-classified by SVM classifier. The inequalities defined by the set I should induce the good behavior of SVM classifier in terms of separation, in other words, the constraints defined by the set I impose that the hyperplane provides a decision function yielding, on the training data, the same outputs of the SVM classifier. The aim of the single-iteration of Step 2 is to determine a separating hyperplane that utilizes fewer features and is not too far from the SVM solution. The nonzero components of the separating hyperplane so determined define a subspace of lower dimension, and we repeat the two-phase procedure in this subspace. The algorithm terminates whenever no dimension reduction is obtained by the concave minimization phase.

4.1.4 Computational experiments

We describe the numerical results obtained by Algorithm FS-SVMCP on nine data sets (See next paragraph for further information) usually employed in linear classifier testing.

Implementation details of the algorithm

At Step 1, SVM problem (4.6) has been solved by *LIBLINEAR A Library for Large Linear Classification*, developed by the Machine Learning Group at National Taiwan University (see [31] for the details). At each iteration, the parameter C has been determined by a standard cross-validation procedure. At Step 2, we have used the following concave zero-norm approximation corresponding to the objective function of formulation (3.26):

$$F(z) = -\sum_{i=1}^{n}(z_i + \epsilon)^{-p} \tag{4.9}$$

with $\epsilon = 10^{-6}$ and $p = 1$. At each iteration, the linear programming problem (4.7) has been solved using *GLPK* (4.9). The experiments were carried out on Intel Pentium 4 3.2 GHz 512 MB RAM.

Experiments and Results

For each problem, we randomly divided the available data into training set and test set, thus generating ten instances. We grouped the nine problems into two sets:

- the first set is made of four well-known problems in Bioinformatics (Breast Cancer, Colon Cancer, Leukemia, Lymphoma) having many features (of the order of thousands) and a small number of training and test data (of the order of tens);

- the second set is made of five miscellaneous problems (Ionosphere, Sonar, Musk, Sylva, Gina) with number n of features ranking from 34 to 970, and number N of training data ranking from 180 to 13056.

The results obtained on the two sets of test problems are shown in tables 4.1 and 4.2 respectively, where for each problem we report

- the number **Tr** of training data, the number **Ts** of test data, the number **n** of features;

DataSet	**Tr**	**Ts**	**n**	**SVM**		**FS-SVMCP**		
				TR%	**TS%**	**TR%**	**TS%**	$\|w^*\|_0$
Breast Cancer	40	4	7129	100%	95.00%	100%	90.00%	10.10
Colon Cancer	50	12	2000	100%	88.33%	100%	85.00%	9.30
Leukemia	58	14	7129	100%	98.57%	100%	97.14%	7.30
Lymphoma	85	11	4026	100%	97.27%	100%	96.36%	10.50

Table 4.1: Results obtained by Algorithm FS-SVMCP on the first set of test problems

DataSet	**Tr**	**Ts**	**n**	**SVM**		**FS-SVMCP**		
				TR%	**TS%**	**TR%**	**TS%**	$\|w^*\|_0$
Ionosphere	300	51	34	100%	94.51%	99.83%	93.92%	15.90
Sonar	180	28	60	91.22%	79.99%	89.73%	81.07%	21.60
Musk	430	46	166	96.28%	89.77%	94.95%	88.91%	48.2
Sylva	13086	1308	216	98.15%	97.95%	98.85%	98.70%	17.9
Gina	3153	315	970	93.85%	85.17%	91.95%	85.78%	156.10

Table 4.2: Results obtained by Algorithm FS-SVMCP on the second set of test problems

- the average (on the ten instances) classification accuracy on the training set (**TR%**) and on the test set (**TS%**) obtained by the standard linear SVM classifier (**SVM**);

- the average (on the ten instances) classification accuracy on the training set (**TR%**), on the test set (**TS%**), and the number of selected features $\|w^\star\|_0$ obtained by Algorithm FS-SVMCP (**FS-SVMCP**).

he results of tables 4.1 and 4.2 clearly show the effectiveness of the proposed feature selection technique. Indeed, we can note that, for each problem, the algorithm is able:

(a) to detect a very small number (in comparison with the number n of original features) of relevant features;

(b) to obtain a good linear classifier (in the reduced dimension space) as model of the process underlying the data (see the classification

accuracies on the training and test sets, and observe that they are comparable with those of the standard SVM classifier operating in the original input space);

Note that the data sets of the first table are very different (in the structure) from those of the second table, and that the performance of the proposed algorithm on the two sets of problems are very similar. This points out a good robustness of the method (thanks to its simplicity), which is useful whenever different kind of applications must be tackled. As previously remarked, the method involves convex quadratic and linear programming problems (which can be efficiently solved by available solvers) and this makes it possible its application, as shown by the experiments, to large dimensional problems. Further experiments not here reported point out that the adoption at Step 2 of other concave functions in place of (4.9) yields results similar to those reported in tables 1 and 2. From the computational experience we get that the proposed feature selection methodology could be advantageously employed to detect the relevant variables of an unknown process (that can be modelled by a linear classifier) and hence to better understand it. This can be of great interest in important fields, such as medicine and biology.

4.1.5 Test problems

Breast Cancer [88]. The duke breast-cancer dataset contains 44 breast cancer tissues described by 7129 genes expression values extracted from DNA microarray data. The data are available at
`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Colon Cancer [2]. The colon cancer dataset contains 22 normal and 40 colon cancer tissues described by 2000 genes expression values extracted from DNA microarray data. The data are available at
`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Leukemia [37]. The leukemia dataset contains information on gene-expression in samples from human acute myeloid (AML) and acute lymphoblastic leukemias (ALL). It contains 25 ALL examples and 47 AML examples described by 7129 genes. The data are available at
`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Lymphoma The gene expression of 96 samples is measured with microarray to give 4026 features; 61 of the samples are malignant and 35 are normal. The data are available at
`http://llmpp.nih.gov/lymphoma/`.

Ionosphere The Ionosphere dataset describes a binary classification task where radar signals target two types of electrons in the ionosphere: those that show some structure (good) and those that do not (bad). The dataset contains 351 samples described by 34 attributes. The data are available at
`http://archive.ics.uci.edu/ml/datasets/`.

Sonar The dataset contains 111 sonar signals bounced off a metal cylinder and 97 bounced off a roughly cylindrical rock. Each pattern is a set of 60 numbers in the range 0.0 to 1.0. The data are available at
`http://archive.ics.uci.edu/ml/datasets/`.

Musk This dataset contains 476 conformations of molecules. The goal is to learn to predict whether new molecules will be musks or non-musks. Each sample is described by a set of 166 features. The data are available at
`http://archive.ics.uci.edu/ml/datasets/`.

Sylva The task of Sylva is to classify forest cover types. This is a two-class classification problem with 216 input variables. Each pattern is composed of 4 records: 2 true records matching the target and 2 records picked at random. Thus a half of the features are distracters. This version of the database was prepared for the WCCI 2006 challenge on performance prediction. The data are available at
`http://clopinet.com/isabelle/Projects/modelselect/`.

Gina The task of Gina is handwritten digit recognition. We chose the problem of separating the odd numbers from even numbers. We use 2-digit numbers. Only the unit digit is informative for that task, therefore at least a half of the features are distracters. This version of the database was prepared for the WCCI 2006 challenge on performance prediction. The data are available at
`http://clopinet.com/isabelle/Projects/modelselect/`.

4.2 Sparse Approximation of Signals

As we have already said in Chapter 1, the goal in sparse approximation is that of approximating a given input signal by means of a linear combination of elementary signals. These elementary signals, usually called atoms, do belong to a large, linearly dependent collection. A preference for linear combinations involving only a few elementary signals is obtained by penalizing nonzero coefficients. A well-known penalty function is the number of elementary signals used in the approximation. Obviously the choice we make about the specified collection, the linear model and the sparsity criterion must be justified by the domain of the problem we deal with.
In this section we propose an approach based on the *Frank-Wolfe - Unitary Stepsize - Reduced Dimension* (FW1-RD) algorithm described in Chapter 3.

4.2.1 A Concave Approach for Sparse Approximation of Signals

The sparse approximation problem is formally described as follows:

$$\min f(x) = \|Ax - b\|_1 + C \ \|x\|_0, \tag{4.10}$$

where A is the dictionary of elementary signals, b is the target signal, and $C > 0$ is a regularization parameter which trades off reconstruction error and sparsity of the optimal solution. Problem (4.10) can be rewritten as

$$\begin{aligned} \min_{x,z \in R^n \ y \in R^m} \quad & \sum_{i=1}^{m} y_i + C \ g(z) \\ s.t. \quad & -y \leq Ax - b \leq y \\ & -z \leq x \leq z \end{aligned} \tag{4.11}$$

with $g(z)$ a concave approximation of the zero norm. As we want to minimize a concave function over a polyhedral set, the FW1-RD Algorithm can be used for for solving the problem.

4.2.2 Experiments and Implementation Details

The set of experiments is performed on synthetic data built as described in [28]. The dictionary A is comprised of L random orthonormal bases, meaning that $K = LN$ is the length of the coefficients vector x. We denote x_0 as the original coefficients vector, which is built by randomly choosing a few non-zero elements (either 1 or -1) with probability γ. The clean signal b is then given by $b = Ax_0$, while its noisy version is $\tilde{b} = b + n$, where n is a white-Gaussian pseudo-random noise with variance σ_n^2 .
We tested our algorithm on 3 different randomly generated problems having the following features:

- synthetic signals of length $N = 64$;
- random dictionaries comprised of $L = 4$ orthonormal bases;
- probability $\gamma = 0.03$;
- variance $\sigma_n^2 = \{0.01,\ 0.05,\ 0.1\}$.

For each problem we performed experiments using:

- formulation (3.19), denoted by *Formulation I*, with $\epsilon = 10^{-9}$ and $p = 0.001$;
- formulation (3.26), denoted *Formulation II*, with $\epsilon = 10^{-6}$ and $p = 1$.

We run the FW1-RD Algorithm over 10 random initial points and selected the solution having the smallest reconstruction error, defined as:

$$err(x) = \|Ax - b\|_1 \ .$$

Algorithm FW1-RD were implemented in C using GLPK (4.9) as solver of the linear programming problems. The experiments were carried out on Intel Pentium 4 3.2 GHz 512 MB RAM.
We make a comparison between our method and the PCD-SESOP method proposed in [28]. This algorithm is for linear least squares problems with non-quadratic regularization, i.e., for unconstrained minimization problems of the form

$$\min f(x) \quad = \|Ax - b\|_2^2 + \rho(x),$$

where $\rho(x)$ is a general regularizer.
Figure 4.1, 4.5, 4.9 show the original solution obtained respectively with σ_n^2 equal to 0.01, 0.05, 0.1; Figure 4.2, 4.6, 4.10 show the reconstructed solution generated by means of PCD-SeSOP; 4.3, 4.7, 4.11 show the reconstructed solution generated by FW1-RD using $FormulationI$; Figure 4.4, 4.8, 4.12 show the reconstructed solution generated by FW1-RD using $FormulationII$. These preliminary results highlight the effectiveness of our method in finding sparse solutions, especially when dealing with noisy signals. However, methods for the sparse approximation of signals in presence of noise deserve more attention and should be the object of a deeper study.

4.3 Conclusions

In this chapter we have tackled two challenging tasks: feature selection and sparse approximation of signals. We proposed a new approach for feature selection based on concave programming and linear SVMs. The results obtained on various datasets seem to show the effectiveness of the method. In fact, the number of features selected by our algorithm is very small if compared with the original number of features. Furthermore, the accuracy of the classifier trained using the data projected on the selected features is comparable with the accuracy obtained by training a classifier on the original data. The absence of significant deterioration in classification performance basically means that the feature selected by our algorithm are among the relevant ones. The results obtained on sparse approximation using the FW1-RD algorithm highlight how helpful can be a concave approach in searching a good representation of a signal, especially in presence of noise.

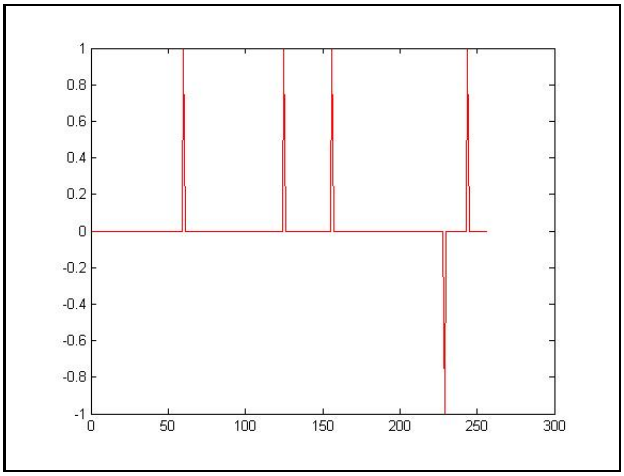

Figure 4.1: Original solution ($\sigma_n^2 = 0.01$).

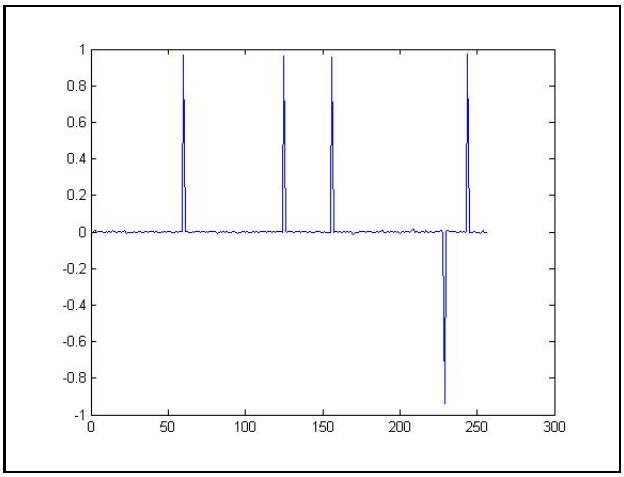

Figure 4.2: PCD-SeSOP solution ($\sigma_n^2 = 0.01$).

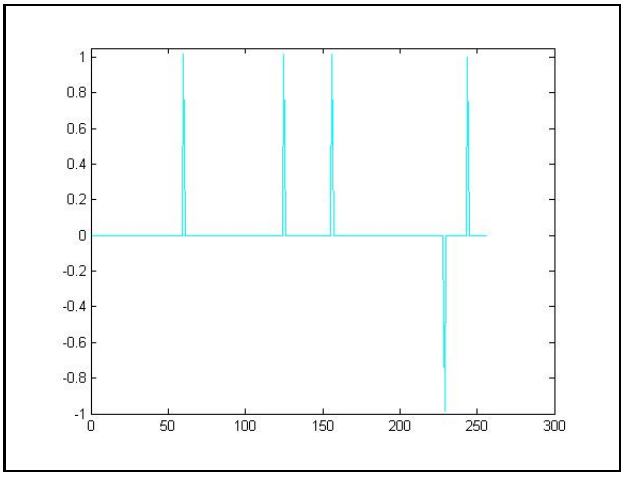

Figure 4.3: FW1-RD solution using $FormulationI$ ($\sigma_n^2 = 0.01$).

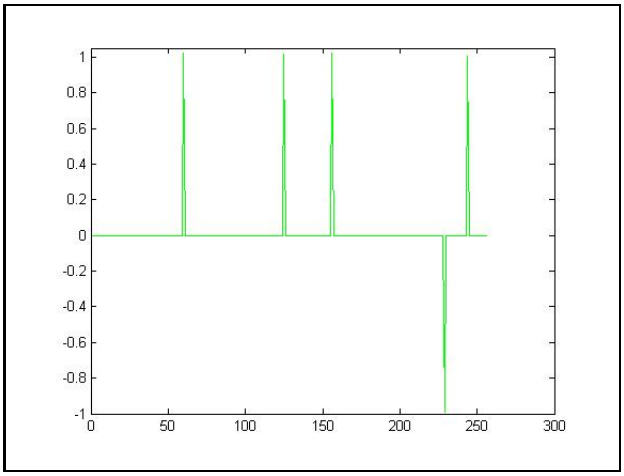

Figure 4.4: FW1-RD solution using $FormulationII$ ($\sigma_n^2 = 0.01$).

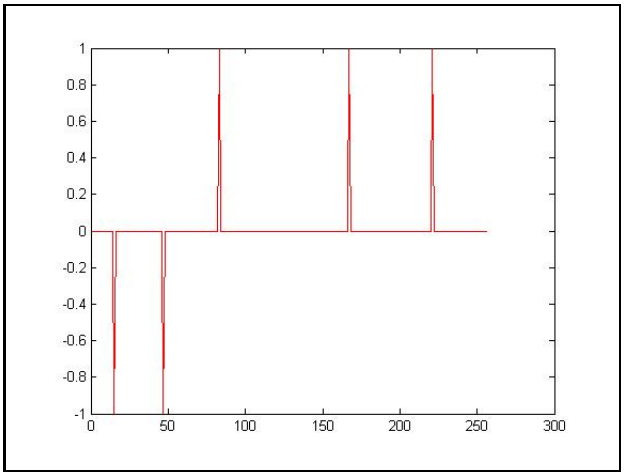

Figure 4.5: Original solution ($\sigma_n^2 = 0.05$).

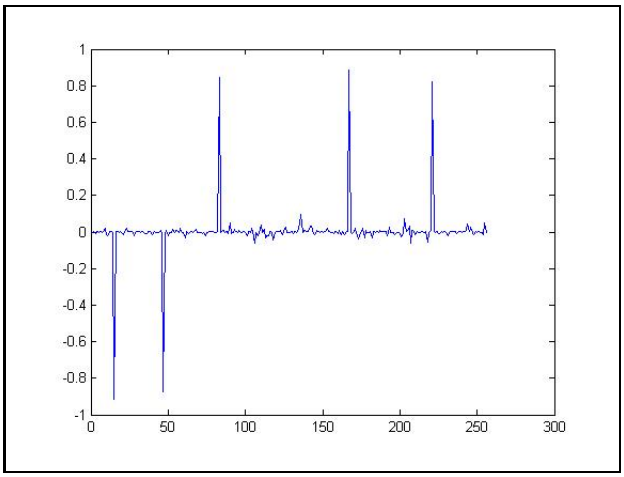

Figure 4.6: PCD-SeSOP solution ($\sigma_n^2 = 0.05$).

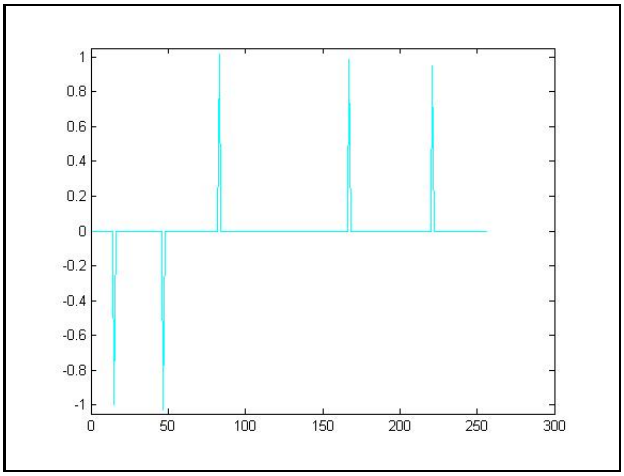

Figure 4.7: FW1-RD solution using $FormulationI$ ($\sigma_n^2 = 0.05$).

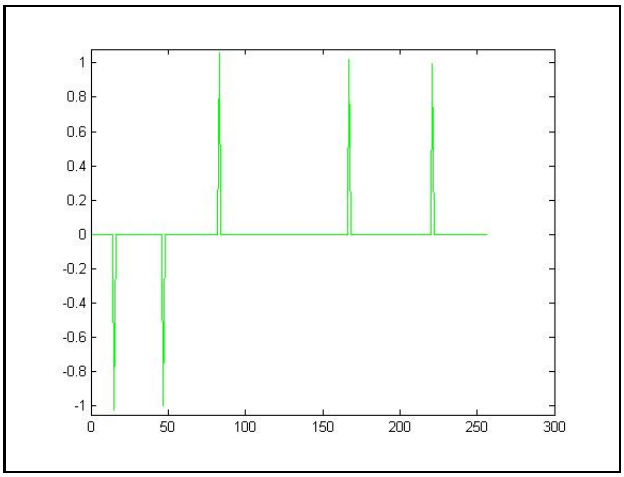

Figure 4.8: FW1-RD solution using $FormulationII$ ($\sigma_n^2 = 0.05$).

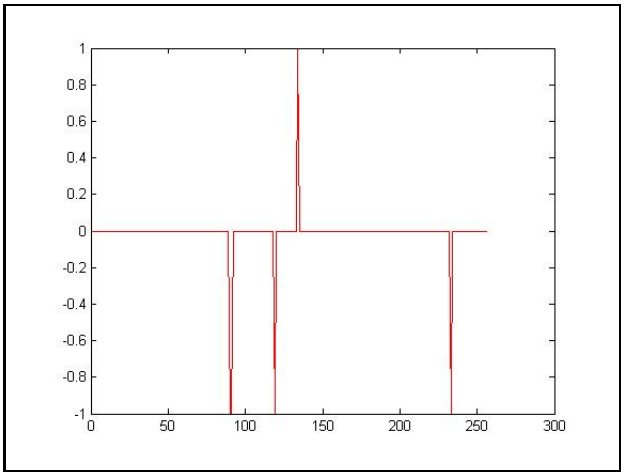

Figure 4.9: Original solution ($\sigma_n^2 = 0.1$).

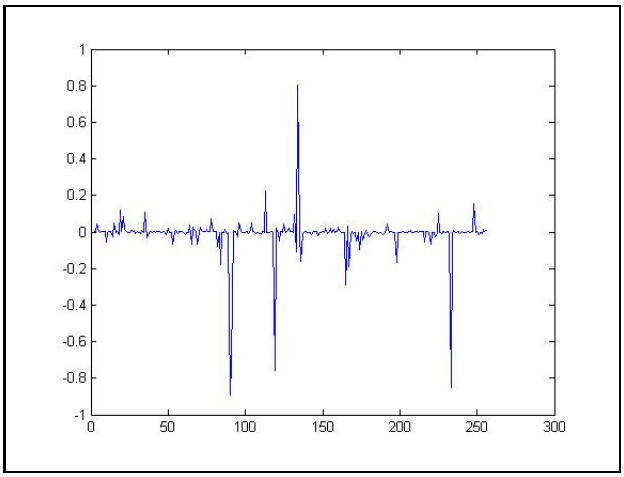

Figure 4.10: PCD-SeSOP solution ($\sigma_n^2 = 0.1$).

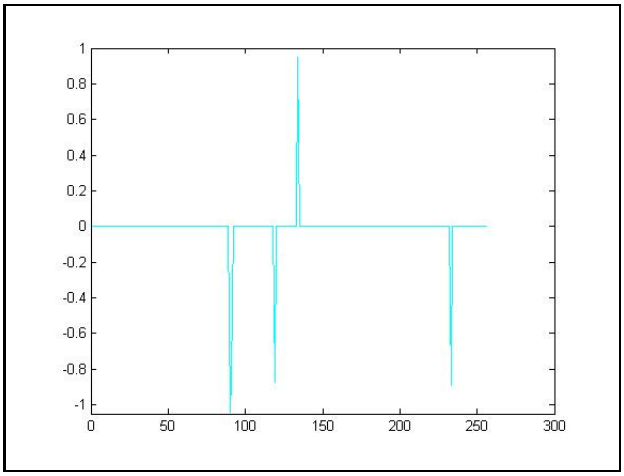

Figure 4.11: FW1-RD solution using $FormulationI$ ($\sigma_n^2 = 0.1$).

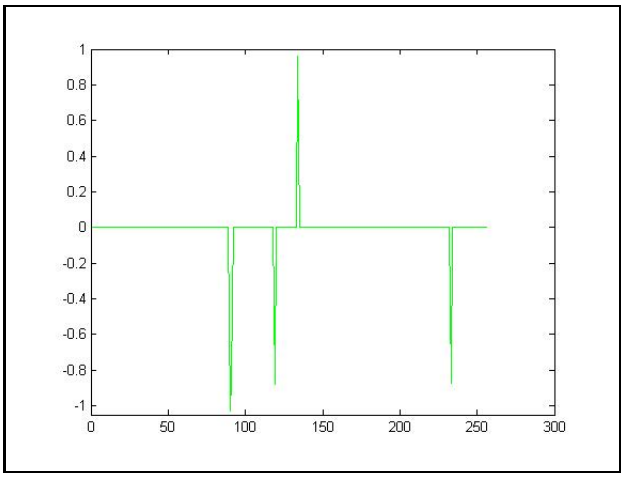

Figure 4.12: FW1-RD solution using $FormulationII$ ($\sigma_n^2 = 0.1$).

Chapter 5

Exact Methods for Global Optimization of Separable Concave Functions over Polyhedral Sets: Challenges and Future Perspectives

The hard task of minimizing a separable concave function over a polyhedral set, which has first examined by researchers more than forty years ago, represents one of the earliest branches of nonlinear programming to be explored in depth. Exact global methods that effectively solve this problem are largely used nowadays.
These global techniques can be very useful when searching for sparse solutions over a linear system. In fact, by combining local approaches described in previous chapters with global optimization methods, we might find solutions much more sparser than those obtained by means of local approaches only.
The problem when dealing with exact global methods is that they are very time and memory consuming as the dimensions of the problem get large. Then some ad-hoc techniques must be used to speed up the algorithms.
In this chapter we first describe convex envelopes and their properties. Then we make an overview of the well-known Branch and Bound algorithm.

5.1 Convex Envelopes

The convex envelopes are a powerful tool widely used in global optimization to solve non-convex mathematical problems. A convex envelope of a function f on a subset S of R^n is defined as the tightest convex underestimating function of f on S. In this section we report some well known properties of convex envelopes [29, 43, 73, 83].

5.1.1 Properties of Convex Envelopes

Definition 5.1.1. *Given a set $S \subseteq R^n$ a **convex underestimator** of a function $f : S \to R$ is a function g having the following properties:*

(a) $g(x)$ is convex;

(b) $g(x) \leq f(x) \quad \forall \; x \in S$.

Definition 5.1.2. *Given a set $S \subseteq R^n$ the **convex envelope** of a function $f : S \to R$ is a function g having the following properties:*

(a) $g(x)$ is a convex underestimator of f;

(b) $g(x) \geq h(x) \quad \forall \; x \in S \quad$ and $\quad \forall \; h$ convex underestimator of f.

Definition 5.1.3. *Given a set $S \subseteq R^n$ the **epigraph** $epi(f)$ of a function $f : S \to R$ is:*

$$epi(f) = \{(x, r) \in S \times R : \; r \geq f(x)\} \; . \tag{5.1}$$

Definition 5.1.4. *Given a set $S \subseteq R^n$ the convex envelope g of a function $f : S \to R$ can be expressed as follows:*

$$g(x) = \inf\{ \; r : (x, r) \in conv(epi(f)) \; \} \; . \tag{5.2}$$

By applying Caratheodory's Theorem to (2) $g(x)$ can be defined as

$$\begin{aligned} g(x) = \quad & \inf \sum_{i=1}^{n+1} \lambda_i f(x_i) \\ s.t. \quad & \sum_{i=1}^{n+1} \lambda_i x_i = x \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & x_i \in S, \quad \lambda_i \geq 0, \quad i = 1, \ldots, n+1 \ . \end{aligned} \tag{5.3}$$

When S is compact and f is lower semicontinuous , the set $conv(epi(f))$ is a closed and convex set, and the convex envelope admits a more precise characterization:

$$\begin{aligned} g(x) = \quad & \min \sum_{i=1}^{n+1} \lambda_i f(x_i) \\ s.t. \quad & \sum_{i=1}^{n+1} \lambda_i x_i = x \\ & \sum_{i=1}^{n+1} \lambda_i = 1 \\ & x_i \in S, \quad \lambda_i \geq 0, \quad i = 1, \ldots, n+1 \ . \end{aligned} \tag{5.4}$$

If S is finite and $conv(S)$ is full-dimensional, $conv(epi(f))$ is a full dimensional polyhedron and the convex envelope is:

$$g(x) = \max_{i \in I} \ a_i^T x + b_i \tag{5.5}$$

with I a finite set of indices. The following propositions show some useful properties of convex envelopes.

Proposition 5.1.1. *Let $f_1,\ f_2 : S \to R$ be two functions defined on a convex set $S \subset R^n$. Let g_1 and g_2 be their convex envelopes. Let us define the sum of these two function as*

$$f_0(x) = f_1(x) + f_2(x)$$

and g_0 its convex envelope. Then

$$g_1(x) + g_2(x) \leq g_0(x) \quad \forall\ x \in S \tag{5.6}$$

and if f_2 is affine, then

$$g_1(x) + g_2(x) = g_0(x) \quad \forall\ x \in S\ . \tag{5.7}$$

Proof. By definition of convex envelope we have for all $x \in S$

$$g_0(x) \leq f_1(x) + f_2(x)$$

and

$$g_1(x) + g_2(x) \leq f_1(x) + f_2(x)\ .$$

Let us suppose, by contraddiction, there exists a point $x \in S$ such that

$$g_0(x) \leq g_1(x) + g_2(x)\ .$$

This contradicts (b) of Definition 5.1.2. Hence, we obtain

$$g_0(x) \geq g_1(x) + g_2(x) \quad \forall x \in S\ .$$

Let us now consider the convex function

$$g_0(x) - g_2(x)\ .$$

By using definition of convex envelope and affine function, we have for all $x \in S$

$$f_1(x) \geq g_1(x) \geq g_0(x) - f_2(x) = g_0(x) - g_2(x)$$

and

$$g_1(x) + g_2(x) \geq g_0(x)$$

which we can combine to (5.6) to have

$$g_1(x) + g_2(x) = g_0(x) \quad \forall\ x \in S\ .$$

□

Proposition 5.1.2.
(i) Let $f_1,\ f_2 : S \to R$ be two functions defined on a convex set $S \subset R^n$ such that

$$f_1(x) \leq f_2(x) \quad \forall\ x \in S\ .$$

Let g_1 and g_2 be their convex envelopes. Then

$$g_1(x) \leq g_2(x) \quad \forall\ x \in S\ . \tag{5.8}$$

(ii) Let $f_1 : S \to R$ be a function defined over a set $S \subset R^n$ with convex envelope g_1. Let $f_2 : T \to R$ be a function defined over a set $T \subset S$ such that

$$f_2(x) = f_1(x) \quad \forall\ x \in T\ . \tag{5.9}$$

with convex envelope g_2. Then

$$g_1(x) \leq g_2(x) \quad \forall\ x \in T\ . \tag{5.10}$$

Proof. By definition of convex envelope we have for all $x \in S$

$$g_1(x) \leq f_1(x)$$

and

$$g_2(x) \leq f_2(x)\ .$$

Let us suppose, by contradiction, there exists a point $\tilde{x} \in S$ such that

$$g_1(\tilde{x}) \geq g_2(\tilde{x})\ .$$

Thus we obtain

$$f_2(\tilde{x}) \geq f_1(\tilde{x}) \geq g_1(\tilde{x}) \geq g_2(\tilde{x})$$

but this contradicts (b) of Definition 5.1.2.

By definition 5.1.4 we have

$$g_1(x) = \inf\{\ r : (x, r) \in conv(epi(f_1))\ \}$$

and

$$g_2(x) = \inf\{\ r : (x, r) \in conv(epi(f_2))\ \}\ .$$

It is easy to see that

$$conv(epi(f_1)) \supseteq conv(epi(f_2))$$

and

$$g_1(x) \leq g_2(x) \quad \forall\, x \in T\ .$$

□

Proposition 5.1.3. *if $f : S \to R$ is lower semicontinuous on the compact convex set $S \subset R^n$, then the convex envelope g is such that*

(i) $\min_{x\in S}\ g(x) = \min_{x\in S}\ f(x)$*;*

(ii) $\arg\min_{x\in S}\ g(x) \supseteq \arg\min_{x\in S}\ f(x)$*;*

(iii) $g(x) = f(x)$ *at extreme points of* S*.*

Proof. Let $x^* \in \arg\min_{x\in S}\ f(x)$. Let us suppose, by contradiction, there exists a point $\tilde{x} \in S$ satisfying the following relation:

$$g(\tilde{x}) < f(x^*)$$

it is possible to define a constant function

$$h(x) = f(x^*) \quad \forall x \in S$$

which satisfies properties (a) and (b) of Definition 5.1.1 and such that

$$h(\tilde{x}) = f(x^*) > g(\tilde{x})$$

but this contradicts (b) of Definition 5.1.2.

Thus we have

$$g(x) \geq \min f(x) \quad \forall\, x \in S$$

and by using property (b) of Definition 5.1.1

$$g(x^*) = f(x^*)\ .$$

Hence (i) and (ii) are both proved.

An extreme point $\bar{x}$ cannot be represented by a convex combination of points in S different from itself, then by using (5.4):

$$g(\bar{x}) = f(\bar{x})$$

and (iii) is proved. □

5.1.2 Necessary and Sufficient Conditions of Poliedrality of Convex Envelopes

Finding the convex envelope of a function is a hard task in general. Here we report necessary and sufficient conditions for a convex envelope to be a polyhedral function and illustrate the way these conditions can be used to construct convex envelopes [72].

Definition 5.1.5. *A function $f(x)$ is said to be a **polyhedral function** if it is the pointwise maximum of a finite set of affine functions $h_i(x)$:*

$$f(x) = \max\{h_i(x)\ i = 1, \ldots, m\}\ . \tag{5.11}$$

Definition 5.1.6. *A lower semicontinuous function $f : P \to R$ defined over a polytope P is said to be a **vertex-polyhedral function** if is such that*

$$\{x :\ (x, f(x)) \in vert(epi(f))\} = vertP\ .$$

Definition 5.1.7. *Let $f : P \to R$ be a lower semicontinuous function on a polytope P. Let g be the convex envelope of f. Set $X(f)$ is said to be the **generating set** of f, if*

$$X(f) = \{x : (x, f(x)) \in vert(epi(g))\}\ . \tag{5.12}$$

Definition 5.1.8. *Let $f : P \to R$ be a lower semicontinuous function on a polytope P. The convex envelope of f is said to be a **vertex-polyhedral convex envelope** if the generating set $X(f)$ coincides with the set of vertices of P:*

$$X(f) = vertP\ , \tag{5.13}$$

which is equivalent to say that its convex envelope on P polytope coincides with the convex envelope of the following function

$$f^{\infty}(x) = \begin{cases} f(x) & x \in vertP \\ \infty & x \notin vertP \end{cases}\ .$$

It is self-evident that convex envelope of f^{∞} is vertex-polyhedral.

Proposition 5.1.4. *Let $f : P \to R$ be a lower semicontinuous function on a polytope P. Let g and g^{∞} be the convex envelopes of f and f^{∞}. The convex envelope g is vertex-polyhedral if and only if*

$$f(x) \geq g^{\infty}(x) \quad \forall\, x \in P\ .$$

Proof.

Necessity. From vertex-polyhedrality of convex envelope g and definition 5.1.1 we have that

$$f(x) \geq g(x) = g^{\infty}(x) \quad \forall\, x \in P\ .$$

Sufficiency. As $g^{\infty}(x)$ is a convex function, from Definition 5.1.2:

$$g(x) \geq g^{\infty}(x) \quad \forall\, x \in P\ .$$

From (i) of Proposition 5.1.2 we have that

$$g^{\infty}(x) \geq g(x) \quad \forall\, x \in P\ ,$$

then

$$g^{\infty}(x) = g(x) \quad \forall\, x \in P$$

which means that $g(x)$ is vertex-polyhedral. □

Remark A vertex-polyhedral function is obviously polyhedral but the converse cannot be always guaranteed (e.g. the function $f(x) = |x|$ on $P = [-1,\ 1] \subset R$).

However, in [72] the following result has been proved:

Proposition 5.1.5. *Let $f : P \to R$ be continuously differentiable over the polytope P. The convex envelope g of f is polyhedral if and only if it is vertex-polyhedral.*

The convex envelope vertex-polyhedrality for a function f is a very important property. In fact, if convex envelope of f is vertex-polyhedral, then it can be explicitly calculated and expressed as the maximum of a finite number of affine functions.

Corollary 5.1.1. *Let $f_i : P \to R \;\; i = 1, \ldots, n$ be continuously differentiable on the polytope P. Let the convex envelopes of all these functions and their sum $f_0(x) = \sum_{i=1}^{n} f_i(x)$ be polyhedral. Let g_0 be the convex envelope of f_0 and g_i the convex envelope of f_i. Then*

$$g_0(x) = \sum_{i=1}^{n} g_i(x)$$

if and only if the generating set of $\sum_{i=1}^{n} g_i(x)$ coincides with the vertices of P:

$$X\big(\sum_{i=1}^{n} g_i(x)\big) = vertP \; . \tag{5.14}$$

Proposition 5.1.6. *Let $f : P \to R$ be lower semicontinuous on the polytope P and for every point $\tilde{x} \in int(P)$ there exists a set*

$$L = \{x : \; x = \alpha \; x_1 + (1 - \alpha) \; x_2, \; 0 \leq \alpha \leq 1, \; x_1, x_2 \in P\} \tag{5.15}$$

such that $\tilde{x} \in int(L)$ and f is concave on L. Then g the convex envelope of f is a vertex-polyhedral function.

Proof. Let us suppose, by contradiction, there exists $\tilde{x} \in X(f) \backslash vertP$. We can find two points $x_1, x_2 \in P$ such that

$$\tilde{x} = \alpha \; x_1 + (1 - \alpha) \; x_2, \; 0 < \; \alpha \; < 1$$

and by concavity of f, we obtain

$$f(\tilde{x}) \geq \alpha \; f(x_1) + (1 - \alpha) \; f(x_2) \; .$$

Hence, $(\tilde{x}, f(\tilde{x})) \notin vert(epi(g))$ and $\tilde{x} \notin X(f)$. $\square$

It might be useful to have a criterion to check if an affine function $h(x)$ is an element of the convex envelope of a function $f(x)$.

Lemma 5.1.1. *Let f(x) be continuously differentiable on P polytope with vertices* V_i $i = 1, \ldots, k$ *and let g the convex envelope of f be polyhedral. Then it is possible to define* $g(x)$ *as follows*

$$g(x) = \min \{\sum_{i=1}^{k} \lambda_i f(V_i) : \ \sum_{i=1}^{k} \lambda_i V_i = x; \ \sum_{i=1}^{k} \lambda_i = 1, \ \lambda_i \geq 0, \ i = 1, \ldots, k\} \ . \tag{5.16}$$

Proof. By our hypothesis g is polyhedral, then from Proposition 5.1.5 we have that

$$X(f) = vertP \ .$$

We can write the convex envelope as follows

$$g(x) = \min\{\alpha : \ (x, \alpha) \in F\}$$

where F is the convex hull of vertices $(V_i, f(V_i))$ $i = 1, \ldots, k$ of P.
This is equivalent to write

$$g(x) = \min \{\sum_{i=1}^{k} \lambda_i f(V_i) : \ \sum_{i=1}^{k} \lambda_i V_i = x; \ \sum_{i=1}^{k} \lambda_i = 1, \ \lambda_i \geq 0, \ i = 1, \ldots, k\} \ .$$

□

Lemma 5.1.2. *Let f(x) be continuously differentiable on P polytope and let g the convex envelope of f be polyhedral. Let an affine function* $h(x)$ *be such that*

(a) $h(x) \leq f(x)$ *for all* $x \in vertP$*;*

(b) $h(V_i) = f(V_i)$ *with* V_i $i = 1, \ldots, n+1$ *affinely independent vertices of P.*

Then $h(x)$ *belongs to the polyhedral description of g and*

$$g(x) = h(x)$$

for any $x \in Conv\{V_i \ i = 1, \ldots, n+1\}$.

Proof. By using Lemma 5.1.1 we have that

$$\begin{aligned} g(x) = \quad & \min \sum_{i=1}^{k} \lambda_i f(\xi_i) \\ s.t. \quad & \sum_{i=1}^{k} \lambda_i \xi_i = x \\ & \sum_{i=1}^{k} \lambda_i = 1 \\ & \xi_i \in P, \quad \lambda_i \geq 0, \quad i = 1, \ldots, n+1 \ . \end{aligned}$$

with $\xi_i \;\; i = 1, \ldots, k$ vertices of P.

As $h(x)$ is an affine function

$$h(x) = \sum_{i=1}^{k} \lambda_i h(\xi_i)$$

and from (a) we have

$$h(x) = \sum_{i=1}^{k} \lambda_i h(\xi_i) \leq \sum_{i=1}^{k} \lambda_i f(\xi_i) = g(x) \quad \forall \ x \in P \ .$$

Let $\tilde{x} \in Conv\{V_i \ i = 1, \ldots, n+1\}$, then from (b)

$$h(\tilde{x}) = \sum_{i=1}^{n+1} \lambda_i h(V_i) = \sum_{i=1}^{n+1} \lambda_i f(V_i) \geq g(\tilde{x}) \quad \forall \ x \in S$$

with $\tilde{x} = \sum_{i=1}^{n+1} \lambda_i V_i, \;\; \sum_{i=1}^{n+1} \lambda_i = 1 \ , \ \lambda_i \geq 0, \;\; i = 1, \ldots, n+1 \ .$

Thus we obtain $h(\tilde{x}) = g(\tilde{x}) \;\; . \ \square$

Proposition 5.1.7. *Let f(x) be continuously differentiable on P polytope and let g the convex envelope of f be polyhedral. Let* $\{h_i(x)\}$ $i = 1, \ldots, m$ *be a collection of affine functions which respect conditions of Lemma 5.1.2. Then the function*

$$\psi(x) = \max\{h_i(x)\ i = 1, \ldots, m\}$$

coincides with $g(x)$ *if and only if*

(i) $X(\psi) = vertP$;

(ii) $\forall\ \xi \in vertP$ *there exists an index* $i \in \{1, \ldots, m\}$ *such that* $\psi_i(\xi) = f(\xi)$.

Proof. Necessary part follows immediately from definition of g. In order to prove the sufficient part, by using property (a) of Lemma 5.1.2 and condition (ii), we obtain:

$$\psi(\xi) = f(\xi) \quad \forall\ \xi \in vertP\ .$$

By condition (i) and Proposition 5.1.5 we have

$$vert(epi(\psi)) = vert(epi(g))$$

and by using Lemma 5.1.1:

$$\psi(x) = g(x) \quad \forall\ x \in P\ .$$

□

5.1.3 Convex Envelopes of Concave Functions

Concave programming represents one of the most interesting and tough class of problems in global optimization. Although finding a convex envelope of an arbitrary concave function is an hard task in general, some useful results for functions defined over special sets have been obtained [43].

Proposition 5.1.8. *Let* $f(x)$ *be a concave function over P polytope with vertices* V_i $i = 1, \ldots, k$.
Then the convex envelope g of f can be defined as

$$g(x) = \min\{\sum_{i=1}^{k} \lambda_i f(V_i) :\ \sum_{i=1}^{k} \lambda_i V_i = x;\ \sum_{i=1}^{k} \lambda_i = 1,\ \lambda_i \geq 0,\ i = 1, \ldots, k\}\ . \tag{5.17}$$

Proof. By our hypothesis f is a concave function over P polytope, then from Proposition 5.1.6 we have that the convex envelope g is a polyhedral function and

$$X(f) = vertP \ .$$

We can write the convex envelope as follows

$$g(x) = \min\{\alpha : \ (x, \alpha) \in F\}$$

where F is the convex hull of vertices $(V_i, f(V_i))$ $i = 1, \ldots, k$ of P.
This is equivalent to write

$$g(x) = \min\{\sum_{i=1}^{k} \lambda_i f(V_i) : \ \sum_{i=1}^{k} \lambda_i V_i = x, \ \sum_{i=1}^{k} \lambda_i = 1, \ \lambda_i \geq 0, \ i = 1, \ldots, k\} \ .$$

□

Proposition 5.1.9. *Let $f(x)$ be a concave function over an n-simplex with vertices $V_1, \ldots, V_{n+1}$.*
Then the convex envelope g of f is an affine function

$$g(x) = a^T x + b, \ a \in R^n, \ b \in R \tag{5.18}$$

which a and b determined by solving the following system of linear equation

$$f(V_i) = a^T V_i + b, \ i = 1, \ldots, n+1 \ . \tag{5.19}$$

Proof. By our hypothesis a vector x has a unique representation as a convex combination of the $n+1$ vertices of the simplex:

$$\sum_{i=1}^{n+1} \lambda_i V_i = x, \ \sum_{i=1}^{n+1} \lambda_i = 1, \ \lambda_i \geq 0, \ i = 1, \ldots, n+1 \ .$$

By using Proposition 5.1.8 and Proposition 5.1.3, we have

$$g(x) = \sum_{i=1}^{n+1} \lambda_i f(V_i) = \sum_{i=1}^{n+1} \lambda_i g(V_i)$$

which means $g(x)$ is an affine function.

It is easy to see that $g(x)$ coincides with the function obtained by solving system (5.19). □

Corollary 5.1.2. *Let $f : S \to R$ be a concave function over a closed interval $S = [l,\ u] \subset R$.*
Then the convex envelope g of f is the segment passing through the points $(l, f(l))$ and $(u, f(u))$.

Corollary 5.1.3. *Let $f_i : S_i \to R$ $i = 1, \ldots, n$ be concave functions over $S_i = [l_i,\ u_i] \subset R$ and let g_i be their convex envelope. Let $f_0 : S \to R$ be the sum of functions f_i defined over the box $S = S_1 \times \cdots \times S_n$. Then the convex envelope of f_0 is defined as follows*

$$g_0(x) = \sum_{i=1}^{n} g_i(x_i)$$

with $g_i(x)$ affine function of the single variable x_i that agrees with f_i at the endpoints of S_i.

5.2 Branch and Bound Methods

Branch and Bound is probably the most popular approach for solving global minimization problems. In this technique, the feasible set (or a relaxation) is split into parts (branching phase) over which lower bounds of the objective function value are determined (bounding phase). The main feature of Branch and Bound is its ability to delete (fathom) subsets of the original feasible set during the iteration process.
In this section a general framework is first presented, then a Branch-and-Bound algorithm for separable concave problems is described. Finally, various techniques for speeding up the branch-and-bound algorithm are reported.

5.2.1 Branch and Bound: A General Framework

We consider the following problem (P):

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in X \end{array} \tag{5.20}$$

where X is a subset of R^n, and assume the minimum of f over X exists and is finite.

Definition 5.2.1. *Let X be a subset of R^n and I be a finite set of indices. A set $\{X_i \mid i \in I\}$ of subsets of X is said to be a* partition *of X if*

$$X = \cup_{i \in I} X_i;$$

$$X_i \cap X_j = \partial X_i \cap \partial X_j \quad \forall\, i, j \in I,\ i \neq j$$

In order to give a flavor of how Branch and Bound works, we describe here the scheme of a general prototype algorithm [75, 43]:

Branch and Bound :

Initialization: $M_0 = X$, $PX_0 = \{M_0\}$, $L_0 = L(M_0)$ and $U_0 = U(M_0)$
If $U_0 = L_0$ then STOP. U_0 is an optimum for P;

Iteration k

1. *Partitioning Step*: Construct a partition PM_{k-1} of the considered subset M_{k-1} and add it to the list of partition elements $PX_k = (PX_{k-1} \backslash M_{k-1}) \cup PM_{k-1}$ to be further explored;
2. *Bounding Step*: For each subregion $M \in PM_{k-1}$ determine lower and upper bounds $L(M)$ and $U(M)$, such that

$$L(M) \leq f(x) \leq U(M),\ \forall\, x \in M;$$

3. *Global Bounding Step*: Set L_k and U_k according to the following formulas:

$$L_k = \min\{L(M) \mid M \in\ PX_k\}, \quad U_k = \min\{U(M) \mid M \in\ PX_k\};$$

4. *Fathoming Step*: Remove each $M \in PX_k$ from PX_k for which:

$$L(M) \geq U_k;$$

5. *Termination and Selection Step*:
If $U_k = L_k$ then STOP. U_k is an optimum for P;
Otherwise,

Select a subregion M_k from the list of partition elements PX_k
Set $k = k + 1$
Go to STEP 1.

The Branch-and-Bound process is usually represented as a tree, where the nodes and the branches respectively correspond to the bounding and the partitioning steps.

Definition 5.2.2. *A Branch-and-Bound algorithm is* finite *if $L_k = U_k$ for some $k < \infty$.*

This is equivalent to say that a globally optimal solution is obtained after a finite number k of steps of the algorithm. When there is no finite termination, one needs to analyze the limit behavior of the algorithm.

Definition 5.2.3. *A Branch-and-Bound algorithm is* convergent *if*

$$\lim_{k\to\infty} |U_k - L_k| = 0.$$

The convergence of the Branch and Bound algorithm depends on the choice of three crucial operations:

1. Partitioning;
2. Bounding;
3. Selection.

Definition 5.2.4. (See Definition IV.4 [43]) *A bounding operation is called* consistent *if at every step any unfathomed partition element can be further refined, and if any infinitely decreasing sequence $\{M_{i_q}\}$ of successively refined partition elements satisfies*

$$\lim_{q\to\infty} L(M_{i_q}) = \lim_{q\to\infty} U_{i_q}. \tag{5.21}$$

As U_{i_q} is not necessarily attained at M_{i_q}, the relation (5.21) is difficult to verify in practice. Then it is easier to show that the following consistency condition holds:

$$\lim_{q\to\infty} L(M_{i_q}) = \lim_{q\to\infty} U(M_{i_q}).$$

Definition 5.2.5. (See Definition IV.6 [43]) *The selection operation is said to be* bounding improving *if at least every finite number of iteration M_k satisfies*

$$L(M_k) = \min\{L(M) \mid M \in PX_k\} = L_k.$$

Equivalently, we can say that at least one element where the actual lower bound is attained is selected for further partition in a finite number of steps of the prototype algorithm.

Theorem 5.2.1. (See Theorem IV.3 [43]) *Suppose in the prototype Branch-and-Bound algorithm that*

(i) the bounding operation is consistent;

(ii) the selection operation is bounding improving.

Then the procedure is convergent:

$$L = \lim_{k\to\infty} L_k = \min_{x\in X} f(x) = \lim_{k\to\infty} U_k = U.$$

5.2.2 Branch-and-Bound Algorithm for Separable Concave Problems

The problem of minimizing a separable concave function represents one of the earliest branches of nonlinear programming to be explored. In [78] Shectman and Sahinidis introduced a Branch-and-Bound algorithm that finds the global minimum of the problem in a finite number of iterations. In this paragraph we give a detailed description of the algorithm.
Let us consider the following problem:

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in X \cap B \end{array} \tag{5.22}$$

with

1. $X = \{x \in R^n \mid a_i^T x \leq b_i,\ i = 1, \ldots, m\}$;
2. $B = \prod_{j=1}^{n} B_j$, $B_j = [l_j, u_j]$ and $l_j,\ u_j \in R \cup \{-\infty, +\infty\}$;

3. $f(x) = \sum_{j=1}^{n} f_j(x_j)$ and $f_j : R \to R$ concave and bounded on B_j;
4. $X \cap B$ bounded.

The algorithm combines the standard Branch-and-Bound procedure with *domain reduction* techniques (we give in the next paragraph a detailed description of this class of accelerating devices). The algorithm basically follows the general scheme showed in the previous paragraph:

Branch and Bound for Concave Separable Problems:

Initialization: $S_0 = B$, $PB_0 = \{B\}$, $L_0 = L(S_0)$ and $U_0 = U(S_0)$.
If $U_0 = L_0$ then STOP. U_0 is an optimum for P;
Choose an integer value $2 \leq N < \infty$;

Iteration k

1. *Partitioning Step*: Construct a partition PS_{k-1} of the considered subset S_{k-1} and add it to the list of partition elements $PB_k = (PB_{k-1} \backslash S_{k-1}) \cup PS_{k-1}$ to be further explored;
2. *Bounding Step*: For each subregion $S \in PS_{k-1}$ determine lower and upper bounds $L(S)$ and $U(S)$, such that

$$L(S) \leq f(x) \leq U(S), \ \forall \ x \in X \cap S;$$

3. *Global Bounding Step*: Set L_k and U_k according to the following formulas:

$$L_k = \min\{L(S) \mid S \in \ PB_k\}, \quad U_k = \min\{U(S) \mid S \in \ PB_k\};$$

4. *Fathoming Step*: Remove each $S \in PB_k$ from PB_k for which:

$$L(S) \geq U_k;$$

5. *Termination and Selection Step*:
If $U_k = L_k$ then STOP. U_k is an optimum for P;
Otherwise,

Select a subregion S_k from the list of partition elements PB_k
Set $k = k + 1$

Go to STEP 1.

Here we report an in-depth description of the critical operations:

1. *Preprocessing*: For any variable x_j unrestricted from below (i.e. $l_j = -\infty$), replace l_j in B_j with the solution of the following linear programming problem:

$$\begin{array}{ll} \min & x_j \\ s.t. & x \in X \cap B \end{array} \tag{5.23}$$

and for any variable x_j unrestricted from above (i.e. $u_j = +\infty$), replace u_j in B_j with the solution of the following linear programming problem:

$$\begin{array}{ll} \max & x_j \\ s.t. & x \in X \cap B. \end{array} \tag{5.24}$$

2. *Bounding*: The bounds on the rectangular subregion S are determined by solving a linear programming relaxation of the original problem. For each concave term f_j we construct a linear underestimator g_j intersecting f_j at the current bounds l_j^S and u_j^S of x_j. The function g_j so described represents the convex envelope of f_j over $[l_j^S, u_j^S]$. We also know (see previous section) that the convex envelope of a separable concave function over a rectangular set S is the sum of the convex envelopes of its individual terms f_j over the set S. Then we have

$$g(x) = \sum_{i=1}^{n} g_j(x_j).$$

BOUNDING RULE
Let w^S be a basic optimal solution of the relaxed problem:

$$\begin{array}{ll} \min & g(x) \\ s.t. & x \in X \cap S. \end{array} \tag{5.25}$$

A lower bound is given as follows

$$L(S) = g(w^S),$$

and an upper bound may be obtained by evaluating

$$U(S) = f(w^S).$$

3. *Selection*: At each step the procedure selects from the list of open subproblems a subproblem having the least lower bound.

   SELECTION RULE
   select any S from the list of partition elements PB_k such that

   $$L(S) = L_k.$$

4. *Branching*: The subset S_k selected for partitioning is replaced by two new elements, thus obtaining a binary tree of the problem. If the level of the problem in the tree is a multiple of N, the algorithm selects the longest edge of the selected subset and bisects it. This measure ensures the finiteness of the algorithm.
   Otherwise, the partitioning rule selects and bisects the edge that corresponds to a variable most responsible for the gap between the objective function $f(w^{S_k})$ and the underestimator $g(w^{S_k})$.
   Furthermore, when the best current solution lies within S_k , it substitutes the midpoint in the branching step. This operation is also made to guarantee the finiteness of the procedure.
   We indicate with j' the index of the partitioning variable, p represents the partitioning point, N is a parameter, $D(S_k)$ gives the level of the tree related to subset S_k, $\tilde{x}$ is the best current solution.

   PARTITIONING RULE

   if $D(S_k) \mod N = 0$ **then**
   $j' \in \arg\max(u_{j'}^{S_k} - l_{j'}^{S_k})$
   $p = (u_{j'}^{S_k} - l_{j'}^{S_k})/2.$
   else
   $j' \in \arg\max[f_j(w_j^{S_k}) - g_j(w_j^{S_k})]$
   if $\tilde{x} \in S_k$ and $\tilde{x}_{j'} \in]l_{j'}^{S_k}, u_{j'}^{S_k}[$ **then**
   $p = \tilde{x}_{j'}$
   else
   $p = (u_{j'}^{S_k} - l_{j'}^{S_k})/2$
   endif
   endif

Basically, what we do is splitting the domain $S_k = \prod_{j=1}^{n}[l_j^{S_k}, u_j^{S_k}]$ into two subdomains:

$$[l_{j'}^{S_k}, p] \prod_{j \neq j'} [l_j^{S_k}, u_j^{S_k}]$$

and

$$[p, u_{j'}^{S_k}] \prod_{j \neq j'} [l_j^{S_k}, u_j^{S_k}].$$

The following theorem (see [78]) shows that the algorithm has a much stronger property than convergence, namely finiteness.

Theorem 5.2.2. *The algorithm determines a solution of problem (5.22) in a finite number of steps.*

5.2.3 Acceleration Techniques

In this paragraph we describe some devices normally used to speed up the Branch-and-Bound algorithm. Techniques similar to the ones given below have been adopted in integer programming [81] and in concave programming [82, 41, 53, 75, 79].
Consider the following LP relaxation:

$$\begin{array}{ll} \min & g(x) \\ s.t. & x \in X \cap S. \end{array} \tag{5.26}$$

Let w^S be an optimal solution of the problem, let $L^S = g(w^S)$, and let U be the upper bound on the global solution.

1. **Linear Parametric Programming Acceleration Techniques:**

   T1. *Generation of valid inequalities by linear parametric programming*: These methods, widely discussed in [35], consist in perturbing a particular constraint from its current value $\sum_{i=1}^{n} a_{ij} w_j^S$ by a quantity μ_i. Let $L^\pi(\mu_i)$ be the function representing the optimum value obtained for the problem when a quantity μ_i is added to the i-th constraint. When dealing with LP problems, this function is well known to be convex.
   As μ_i is decreased from zero, let l_i^π be the value $(\sum_{j=1}^{n} a_{ij} w_j^S) + \mu_i$

such that $L^\pi(\mu_i) = U$ or the perturbed problem becomes infeasible. Similarly, as μ_i is increased from zero, let u_i^π be the value $(\sum_{j=1}^n a_{ij} w_j^S) + \mu_i$ such that $L^\pi(\mu_i) = U$ or the perturbed problem becomes infeasible. From convexity of $L^\pi(\mu_i)$, for any feasible value $(\sum_{j=1}^n a_{ij} w_j^S) + \mu_i$ lower than l_i^π or greater than u_i^π, we have

$$L^\pi(\mu_i) > U.$$

Hence, the two new inequalities:

$$\sum_{j=1}^{n} a_{ij} x_j \geq l_i^\pi$$

and

$$\sum_{j=1}^{n} a_{ij} x_j \leq u_i^\pi$$

are valid for the considered subproblem.

T2. *Bound Tightening by linear parametric programming*: Similarly to the previous technique, we consider a variable x_j, to be perturbed from its current value w_j^S by λ_j. As λ_j is decreased from zero, let l_j^π be the value $w_j^S + \lambda_j$ such that $L^\pi(\mu_i) = U$ or the perturbed problem becomes infeasible. Similarly, as λ_j is increased from zero, let u_j^π be the value $w_j^S + \lambda_j$ such that $L^\pi(\mu_i) = U$ or the perturbed problem becomes infeasible. From convexity of $L^\pi(\mu_i)$, for any feasible value $w_j^S + \lambda_j$ lower than l_j^π or greater than u_j^π, we have

$$L^\pi(\mu_i) > U.$$

Hence, the two new inequalities:

$$x_j \geq l_j^\pi$$

and

$$x_j \leq u_j^\pi$$

are valid for the considered subproblem.

2. **Sensitivity Analysis-based Acceleration Devices:**
The general global optimization problem considered here is

$$\begin{array}{ll} \min & f(x) \\ s.t. & h(x) \leq 0 \quad x \in X \end{array} \tag{5.27}$$

where $f : X \to R$, $h : X \to R^m$ and $X \subseteq R^n$.
The relaxation of (5.27) to be solved at the root node of a branch-and-bound tree is

$$\begin{array}{ll} \min & g(x) \\ s.t. & \bar{h}(x) \leq 0 \quad x \in \bar{X} \end{array} \tag{5.28}$$

where $f : \bar{X} \to R$, $\bar{h} : \bar{X} \to R^m$ and $\bar{X} \subseteq R^n$. For any feasible x in the original problem we have

$$g(x) \leq f(x) \quad and \quad \bar{h}(x) \leq h(x).$$

A perturbed version of (5.28) is

$$\begin{array}{lll} \varphi(y) & = \min & g(x) \\ & s.t. & \bar{h}(x) \leq y \quad x \in \bar{X}. \end{array} \tag{5.29}$$

The following theorem [63] shows an important property of the perturbation function $\varphi(y)$ considered above:

Theorem 5.2.3. *Assume relaxed problem (5.28) has a finite optimum at $\bar{x}$ with value $\varphi(0)$ and lagrange multipliers μ. Then the hyperplane:*

$$z(y) = \varphi(0) - \mu^T y$$

is a supporting hyperplane of the graph of $\varphi(y)$ at $y = 0$, i.e.

$$\varphi(y) \geq \varphi(0) - \mu^T y \quad \forall y \in R^m$$

The following result can be derived from theorem 5.2.3:

Theorem 5.2.4. *If relaxed problem (5.28) is convex with optimal value $\varphi(0) = L$, constraint i is active at the optimum and the lagrange multiplier $\mu_i > 0$ then, if U is an upper bound of the original problem (5.27) the constraint:*

$$\bar{h}_i(x) \geq -(U - L)/\mu_i \tag{5.30}$$

is valid for problem (5.27), i.e. does not exclude any feasible solution with value better than U.

Proof. Problem (5.29) can be seen as a convex relaxation of the perturbed nonconvex problem

$$\begin{array}{rll} \Phi(y) & = \min & f(x) \\ & s.t. & h(x) \leq y \quad x \in X. \end{array} \tag{5.31}$$

then we have $\varphi(y) \leq \Phi(y)$. Let $y = e_i y_i$; From theorem 5.2.3 we have

$$L - \mu^T e_i y_i \leq \varphi(y) \leq \Phi(y).$$

By requiring that $\Phi(y) \leq U$, we obtain

$$L - \mu^T e_i y_i \leq U.$$

Finally, consider only non-positive values for y. Since $\bar{h}_i(x) \leq y_i$ is active for $y_i = 0$, then it will also be active for $y_i \leq 0$. Then we can substitute y_i with $\bar{h}_i(x)$ and deduce

$$L - \mu_i \bar{h}_i(x) \leq U.$$

□

These general results can be specialized to the case we have described in the previous pages, and the following techniques can be obtained:

T1. *Generating mirror inequalities by means of sensitivity analysis with respect to b_i*: Consider the linear constraint

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

that is active at w^s with the lagrange multiplier $\mu_i > 0$. Then we can deduce:

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i - (U - L)/\mu_i$$

is valid for the considered subproblem.

T2. *Tightening bounds by means of sensitivity analysis*: Consider the upper bound

$$x_j \leq u_j$$

that is active at w^s with the lagrange multiplier $\mu_i > 0$. Then we can deduce:

$$x_j \geq u_j - (U - L)/\mu_i$$

is valid for the considered subproblem. Consider the lower bound

$$x_j \geq l_j$$

that is active at w^s with the lagrange multiplier $\mu_i > 0$. Then we can deduce:

$$x_j \leq l_j + (U - L)/\mu_i$$

is valid for the considered subproblem.

3. **Domain Reduction using Probing to induce Marginal Values**: The generation of valid inequalities is based upon the sets of constraints that are active at the solution w^s of the considered relaxed subproblem. Valid inequalities can be also derived from constraints not active at w^s by probing at the bounds. In practice, we temporarily fix these variables at their bounds and solve the partially restricted relaxed problem. This lead to the following techniques:

T1. *Generating mirror inequalities by probing the slack domain of i*: Consider the linear constraint

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

that is not active at w^s. Solve subproblem after fixing

$$\sum_{j=1}^{n} a_{ij} x_j = b_i$$

(i.e by adding $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$). Let Z be the optimal solution of this partially restricted problem. If the lagrange multiplier $\mu_i > 0$ for constraint $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$, then we can deduce:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i + (U - Z)/\mu_i$$

is valid for the considered subproblem.

T2. *Tightening bounds by probing the existing domain of variable x_i*: Consider the upper bound

$$x_j \leq u_j$$

that is not active at w^s. Solve subproblem after fixing

$$x_j = u_j$$

(i.e by adding $x_j \geq u_j$). Let Z be the optimal solution of this partially restricted problem. If the lagrange multiplier $\mu_i > 0$ for constraint $x_j \geq u_j$, then we can deduce:

$$x_j \leq u_j + (U - Z)/\mu_i$$

is valid for the considered subproblem. Consider the lower bound

$$x_j \geq l_j$$

that is not active at w^s. Solve subproblem after fixing

$$x_j = l_j$$

(i.e by adding $x_j \leq l_j$). Let Z be the optimal solution of this partially restricted problem. If the lagrange multiplier $\mu_i > 0$ for constraint $x_j \leq l_j$, then we can deduce:

$$x_j \geq l_j - (U - Z)/\mu_i$$

is valid for the considered subproblem.

4. **Acceleration Devices using no Dual Information**:

T1. *Optimality-based Tightening*: This method uses current upper and lower bounds on the global solution to generate constraints that may trim-off inferior portions of S.

T2. *Feasibility-based Tightening*: This method generates constraints that cut-off infeasible portions of the solution space.

See [78] for further details.

5.3 Future Perspectives

Significant progress has been made in global optimization, but there is clearly still a lot of work that need to be done.
First, when dealing with large scale concave problems we need efficient convex envelopes that give a good lower bound (i.e. a lower bound that makes the gap narrow enough). So the crucial task of finding tight lower bounds of a separable concave function over a polyhedral set deserves a deeper study.
Second, Branch-and-Bound algorithms for separable concave programming problems should be further developed and investigated, as well as implemented using parallel techniques.
Third, Computational strategies for speeding up the Branch-and-Bound algorithm (e.g. domain reduction techinques) need to be studied in depth and implanted effectively.
Finally, a great deal of work should be devoted to the development of efficient local optimization methods for finding useful upper bounds (i.e. feasible points not far from the optimal solution). In fact, combining effective local methods with exact global approaches might lead to fast algorithms able to close the gap in reasonable time.
Anyway, results obtained in global optimization are very encouraging, so there is reason for optimism in the future of this area.

Bibliography

[1] H.Almuallin and T.G.Deitterich, *Learning with many irrelevant features*, In Proceedings of the Ninth National Conference on Artificial Intelligence, pages 547–552,1991.

[2] U. Alon, N. Barkai, DA Notterman, K. Gish, S. Ybarra, D. Mack, AJ. Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays*, Proceedings of the National Academy of Science, 96, pp. :6745–6750, 1999.

[3] E. Amaldi, V. Kann, *On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems*, Theoretical Computer science, 209, pp. 237–260, 1998.

[4] E. Amaldi, *On the Complexity of Designing Compact Perceptrons and Some Consequences*, Electronic Proc. Fifth Int. Symposium on A.I. and Mahematics, Fort Lauderdale, Florida, E. Boros and R. Greiner, eds., RUTCOR, 1999.

[5] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, *Distributional word clusters vs. words for text categorization.*, JMLR, 3:1183–1208, 2003.

[6] Bennett, K. P., and Mangasarian, O. L., *Neural Network Training via Linear Programming*, Advances in Optimization and Parallel Computing, Edited by P. M. Pardalos, North Holland, Amsterdam, Holland, pp. 56-67, 1992.

[7] Bennett, K. P., and Mangasarian, O. L., *Bilinear separation of two sets in n-space*, Computational Optimization and Applications, 2:207–227, 1993.

[8] C.M.Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, NewYork, 1995.

[9] A.Blumer,A.Ehrenfeucht,D.Hausler,andM.K.Warmuth, *Occam's razor*, Information Processing Letters, 24:377–380, 1987.

[10] P. S. Bradley, O. L. Mangasarian, *Feature selection via concave minimization and support vector machines*, Machine Learning Proceedings of the Fifteenth International Conference(ICML '98), J. Shavlik, editor, Morgan Kaufmann, San Francisco, California, pp. 82–90, 1998.

[11] P. S. Bradley, O. L. Mangasarian and J. B. Rosen, *Parsimonious Least Norm Approximation*, Computational Optimization and Applications 11(1), October 1998, 5–21.

[12] L.Breiman J.H.Friedman R.A.Olshen C.J.Stone, *Classification and Regression Trees*, Wadsworth and Brooks/Cole, Monterey, California, 1984.

[13] D. S. Broomhead and D. Lowe, *Multivariate functional interpolation and adaptive networks* " Complex Systems " 2, pp. 321 – 355, 1988.

[14] A.M. Bruckstein, D. L. Donoho, M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, to appear in Siam Review.

[15] Chen, S., Billings, S. A. and Luo, W., *Orthogonal least squares methods and their application to non-linear system identification*, International Journal of Control, 50 . pp. 1873–1896, 1989.

[16] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. *Coordinate Descent Method for Large-scale L2-loss Linear SVM.* Journal of Machine Learning Research (2008). To appear.

[17] S.S. Chen, D.L. Donoho, M.A. Saunders, *Atomic decomposition basis pursuit*, SIAM Rev., 43, pp. 129–159, 2001.

[18] S.-J., Chung, *NP-completeness of the linear complementarity problem*, Journal of Optimization Theory and Applications,60:393-399,1989.

[19] P. Combettes, V. Wajs. *Signal recovery by proximal forward-backward splitting*. SIAM Journal on Multiscale Modeling and Simulation, vol. 4, pp. 11681200, 2005.

[20] Dai Y. H., Fletcher R., *"New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds"*. Mathematical Programming, Ser. A 106, 403–421 (2006).

[21] I. Daubechies, M. Defriese, C. De Mol. *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*. Communications on Pure and Applied Mathematics, vol. LVII, pp. 1413 1457, 2004.

[22] P.A. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice Hall, London, 1982.

[23] D.L. Donoho and P.B. Starck, *Uncertainty principles and signal recovery*, SIAM Journal on Applied Mathematics, 49(3):906–931, June, 1989.

[24] D.L. Donoho and X. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. On Information Theory, 47(7):2845 2862, 1999.

[25] D.L. Donoho, M. Elad, *Optimal Sparse Representation in General (Nonorthogoinal) Dictionaries via L1 Minimization*, the Proc. Nat. Aca. Sci., Vol. 100, pp. 2197–2202, March 2003.

[26] D.L. Donoho, M. Elad, V.N. Temlyakov, *Stable recovery of sparse overcomplete representation in the presence of noise*, IEEE Transactions on Information Theory, 52, pp. 6–18, 2006.

[27] M. Elad and A.M. Bruckstein, *A generalized uncertainty principle and sparse representation in pairs of bases*, IEEE Trans. On Information Theory, 48:2558–2567, 2002.

[28] M. Elad, B. Matalon, M. Zibulevsky. *Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization.* Applied Comput. Harmon Anal 23, pp. 346–367, 2007.

[29] J. Falk. *Lagrange multipliers and nonconvex programs.* SIAM Journal of Control 7 (1969). 534-545.

[30] Fan R.-E., Chen P.-H., and Lin C.-J. , *"Working set selection using the second order information for training SVM"*. Journal of Machine Learning Research, 6:1889–1918, 2005.

[31] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. *LIBLINEAR: A library for large linear classification.* Journal of Machine Learning Research (2008), to appear.

[32] M. Figueiredo, R. Nowak, S. Wright. *Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems.* IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing, vol. 1, no. 4, pp. 586–598, 2007.

[33] G. Forman. *An extensive empirical study of feature selection metrics for text classification.* JMLR, 3:1289–1306, 2003.

[34] M. Frank, P. Wolfe, *An algorithm for quadratic programming*, Naval Research Logistics Quarterly, 3, pp. 95–110, 1956.

[35] Gal, T., *Postoptimal Analysis, Parametric Programming and Related Topics*, 1995. second ed. de Gruyter, Berlin.

[36] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[37] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.* Science, 286(5439):531, 1999.

[38] R. Gribonval, M. Nielsen, *Sparse representation in union of bases*, IEEE Trans. on Information Theory, 49, pp. 3320–3325, 2003.

[39] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. *Gene selection for cancer classification using support vector machines.* Machine Learning, 46:389–422, 2002.

[40] I. Guyon, A. Elisseeff, *An introduction to variable and feature selection*, Journal of Machine Learning Research, 3, pp. 1157–1182, 2003.

[41] P. Hansen, B. Jaumard and S.J. Lu, *An Analytical approach to global optimization*, Mathematical Programming 52 (1991), pp. 227-254.

[42] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice-Hall, 1999.

[43] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, 2nd revised edition, Springer-Verlag: Berlin, 1993.

[44] Hiriart-Urruty, J.B., Lemaréchal, C., *"Fundamentals of Convex Analysis"*, Springer Verlag, 2001.

[45] Joachims T., *"Making large-scale SVM learning practical"*. In Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C.J.C. Burges, and A. J. Smola, Eds. Cambridge, MA:MIT Press, 1998.

[46] G.H.John, R.Kohavi, K.P. Pfleger, *"Irelevant features and the subset selection problem"*. In Procedings of the 11th International Conference on Machine Learning, pages121–129, San Matteo, CA, 1994. MorganKaufmann.

[47] L.A. Karlovitz, *Construction of nearest points in the ℓ_p, p even and ℓ_1 norms*, Journal of Approximation Theory, 3:123–127,1970.

[48] Keerthi S. S., Shevade S., Bhattacharyya C. and Murthy K. *"Improvements to Platts SMO algorithm for SVM classifier design"*, Neural Comput., vol. 13, pp. 637–649, 2002.

[49] K. Kira and L. Rendell, *"A practical approach to feature selection"*, In D. Sleeman and P. Edwards, editors, International Conference on Machine Learning, pages 368-377, Aberdeen, July 1992. Morgan Kaufmann.

[50] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi, *Optimization by Simulated Annealing*, Science, New Series, Vol. 220, No. 4598. (May 13, 1983), pp. 671–680.

[51] D. Koller and M. Sahami, *"Toward optimal feature selection"*, In 13th International Conference on Machine Learning, pages 284–292, July 1996.

[52] M. Kudo and J. Sklansky, *"Comparison of algorithms that select features for pattern classifiers"*, Pattern Recognition, vol. 33, no. 1, pp. 25–41, 2000.

[53] B. W. Lamar, *"An Improved Branch and Bound Algorithm for Minimum Concave Cost Network Flow Problems"*, Journal of Global Optimisation, 3, 1994, p. 261-287.

[54] Y. Le Cun, J. S. Denker, S. A. Solla, *"Optimal brain damage"*, Advances in Neural Information Processing Systems, 1990.

[55] Z. Q. Luo, J. S. Pang, D. Ralph, *"Mathematical Programs with Equilibrium Constraints"*, Cambridge University Press, New York, 1996.

[56] S. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Processing, 41(12):3397–3415, 1993.

[57] O.L. Mangasarian, *Machine learning via polyhedral concave minimization*, in "Applied Mathematics and Parallel Computing – Festschrift for Klaus Ritter", H. Fischer, B. Riedmueller, S. Schaeffler, editors, Physica-Verlag, Germany, pp. 175–188, 1996.

[58] O.L. Mangasarian, *Solution of General Linear Complementarity Problems via Nondifferentiable Concave Minimization*, Acta Mathematica Vietnamica, 22(1), 1997, 199–205.

[59] O.L. Mangasarian, *Arbitrary-Norm Separating Plane.*, Mathematical Programming Technical Report 97-07r, May 1997, Revised September 1998. Operations Research Letters 24, 1999, 15-23.

[60] Mangasarian O. L. and Musicant D. R. *"Successive Overrelaxation for Support Vector Machines"*. IEEE Transaction on Neural Networs, Vol. 10, No. 5, September 1999.

[61] Z. Michalewicz *"Genetic algorithms + data STRUCTURES = evolution programs"*. Springer, 1992.

[62] A. Miller *"Subset Selection in Regression"*. Chapman and Hall , 1990.

[63] M. Minoux *"Mathematical programming: theory and algorithms"*. Wiley 1986.

[64] Narendra P.M., Fukunaga K. *"A Branch and Bound Algorithm for Feature Subset Selection"*. IEEE Trans. Computer , vol 26, pp 917–922, September 1977.

[65] B. K. Natarajan, *Sparse Approximate Solutions to Linear Systems*, SIAM J. Comput. 24(2): 227–234 (1995).

[66] Osuna E., Freund R. and Girosi F. *"Support vector machines: Training and applications"*. A.I. Memo 1602, MIT A.I. Lab., 1997

[67] S. Perkins, K. Lacker, J. Theiler *"Grafting: Fast, Incremental Feature Selection by Gradient Descent in Function Space"*. Journal of Machine Learning Research 3: 1333-1356 (2003).

[68] Platt J. C. *"Fast training of support vector machines using sequential minimal optimization"*. In Bernhard Scholkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, 1998. MIT Press.

[69] Powell, M. J. D. *"Radial basis functions for multivariable interpolation: A review."*. In Algorithms for Approximation, J. C. Mason and M. G. Cox, Eds. Oxford University Press, Oxford, UK, 1987, pp. 143-167.

[70] Rakotomamonjy, A. *"Variable selection using SVM-based criteria"*. J. Mach. Learning Res. v3. 1357-1370, 2003.

[71] B.D. Rao, K. Engan, S.F. Cotter, J. Palmer, and K. Kreutz-Delgado, *Subset selection in noise based on diversity measure minimization*, IEEE Trans. on Signal Processing, 51(3):760–770, 2003.

[72] A. D. Rikun, *A Convex Envelope Formula for Multilinear Functions*, Journal of Global Optimization, v.10 n.4, p.425-437, June 1997.

[73] T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[74] D.E.Rumelhart, G.E.Hinton and R.J.Williams, *Learning representations by back-propagating errors* Nature, (323):533–536, 1986.

[75] Ryoo, H. S. and Sahinidis, N. V., *A branch-and-reduce approach to global optimization* A, Journal of Global Optimization 8, 107-139, 1996.

[76] Scheinberg K. *"An Efficient Implementation of an Active Set Methods for SVMs"*. Journal of Machine Learning Research 7 (2006) 2237–2257.

[77] W. Siedlecki and J. Sklanski *"On automatic feature selection"*. Int. J. Pattern Recognit. Artif. Intell., vol. 2, no. 2, pp. 197-220, 1988.

[78] Shectman, J. P. and N. V. Sahinidis *"A finite algorithm for global minimization of separable concave programs"*. Journal of Global Optimization, 12(1), 1-36, 1998.

[79] Sherali, H.D. and C.H. Tuncbilek *" A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique"*. Journal of Global Optimization 2, 101-112.

[80] M. Stone, *"A Cross-validatory choice and assessment of statistical predictions"*. Journal of the Royal Statistical Society,36:111–147,1974.

[81] Suhl U.H., R. Szymanski, *"Supernode Processing of Mixed-Integer Models"*. Computational Optimization and Applications, 3, 317-331, 1994.

[82] Thakur, L.S., *"Domain Contraction in Nonconvex Programming: Minimizing a Concave Quadratic Objective Over a Polyhedron"*. Mathematics Of Operations Research, Vol. 16, No. 2, pp. 390-407, 1991.

[83] Tawarmalani, M. and Sahinidis, N. V., *"Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications"*. Kluwer Academic Publishers, Dordrecht, Vol. 65 in "Nonconvex Optimization And Its Applications" series, 2002.

[84] Tibshirani, R., *Regression shrinkage and selection via the lasso*, J. Royal. Statist. Soc B., Vol. 58, No. 1, pages 267–288.

[85] P. Tseng and S. Yun, *A Block-Coordinate Gradient Descent Method for Linearly Constrained Nonsmooth Separable Optimization*, January 2008, to appear in J. Optim. Theory Appl.

[86] Vapnik, V.N. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

[87] R.E. Warmack, R.C. Gonzalez, *An algorithm for optimal solution of linear inequalities and its application to pattern recognition*, IEEE Transactions on Computers, 22, pp. 1065–1075, 1973.

[88] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, Jr., J. R. Marks, and J. R. Nevins. *Predicting the clinical status of human breast cancer by using gene expression profiles.* Proceedings of the National Academy of Sciences, 98:11462-11467, 2001.

[89] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, *Feature selection for SVMs.*, NIPS 13, 2000.

[90] J. Weston, A. Elisseef, B. Schölkopf, *Use of the zero-norm with linear models and kernel model*, Journal of Machine Learning Research, 3, pp. 1439–1461, 2003.