ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Liuzzi, F. Rinaldi

A FIRST-ORDER METHOD FOR ℓ_0-PENALIZED PROBLEMS WITH SIMPLE CONSTRAINTS

R. 16, 2012

Giampaolo Liuzzi – Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche, Viale Manzoni 30, 00185 Rome, Italy. `email: giampaolo.liuzzi@iasi.cn`

Francesco Rinaldi – Dipartimento di Matematica, Università degli Studi di Padova, Via Trieste 63, 35121 Padova, Italy. `email: rinaldi@dis.uniroma1.it`.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

This paper is concerned about the definition of a first-order method for the solution of ℓ_0-penalized problems with simple constraints. We propose a reduced dimension Frank-Wolfe algorithm and show that the subproblem related with computation of the FW direction can be solved analitically at least for some sets of simple constraints. The proposed method is applied to the numerical solution of two practical optimization problems, namely, the sparse principal component analysis and the sparse reconstruction of noisy signals. In both cases, the reported numerical performances and comparisons with state-of-the-art solvers show efficiency of the proposed method.

Key words: Sparse problems, Frank-Wolfe method, SPCA, Noisy signals

1. Introduction

In this paper we consider the following class of problems

$$\begin{aligned} &\min_{x\in\Re^n} g(x) + \lambda\|x\|_0 \\ &s.t. \quad x \in C, \end{aligned} \tag{1}$$

where λ is a positive parameter, $g(x)$ is a continuously differentiable function, C is a closed convex and $\|x\|_0$ is the zero-norm of x, that is,

$$\|x\|_0 = card(\{i : x_i \neq 0\}).$$

In what follows, we require the set C to satisfy the assumption below.

Assumption 1. *Set C is described by "simple" constraints. In particular:*

(a) a spherical constraint, i.e. $C = \{x \in R^n : \ \|x\|^2 \leq 1\}$;

(b) bound constaints on the variables, i.e. $C = \{x \in R^n : -M \leq x \leq M\}$.

As shown in [2, 20, 21], since the zero-norm is a discontinuous and nonconvex function, dealing with Problem (1) is a very difficult task, meaning that, as it can be shown, Problem (1) is an NP-hard combinatorial problem.

Despite our assumption on set C, Problem (1) is still sufficiently general to encompass different practical optimization problems like, for instance,

- the Sparse Principal Component Analysis (SPCA) [10, 15, 29] and
- the Sparse reconstruction of noisy signals (SRNS) [13, 18].

In the paper, following the ideas proposed in [17, 22, 27], we choose to deal with Problem (1) by replacing the zero-norm $\|x\|_0$ with a suitable smooth concave approximating function $h(y) : \Re^n \to \Re$, thus obtaining the following problem.

$$\begin{aligned} &\min_{x\in\Re^n} f(x) = g(x) + \lambda h(y) \\ &s.t. \quad x \in C, \\ &\qquad\ -y \leq x \leq y. \end{aligned} \tag{2}$$

As proposed in [17, 22, 27], possible expressions for $h(y)$ are the following:

$$h(y) \ = \ \sum_{i=1}^{n}(1 - e^{-\alpha y_i}), \quad \alpha > 0; \tag{3}$$

$$h(y) \ = \ \sum_{i=1}^{n}\ln(\epsilon + y_i), \quad \epsilon > 0; \tag{4}$$

$$h(y) \ = \ \sum_{i=1}^{n}(y_i + \epsilon)^p, \quad \epsilon > 0, \ 0 < p < 1; \tag{5}$$

$$h(y) \ = \ \sum_{i=1}^{n} -(y_i + \epsilon)^{-p}, \quad \epsilon > 0, \ p \geq 1. \tag{6}$$

We then solve Problem (7) employing the Frank-Wolfe (FW) algorithm proposed in [21]. To this aim, under Assumption 1, we show that the FW direction can be computed analytically thus making the algorithms very efficient and competitive.
The paper is organized as follows. In Section 2 we present the general Frank-Wolfe method that we use to solve the approximating problem (2). In Section 3, we show how to analitically compute the FW direction for problems satisfying Assumption 1. In Section 4 we briefly present the SPCA problem. Section 5 is devoted to the SRNS problem. In Sections 6 and 7 we report some numerical results of the proposed method and comparison with two well-known codes for SPCA and SRNS, namely GPower [15] and SpaRSA [18] respectively, which show the viability and efficiency of the proposed approach. Finally, in Section 8 we draw some conclusions.

2. The Frank-Wolfe reduced dimension algorithm

The Frank-Wolfe algorithm is a well-known and widely-used method 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 [11]. At each step of the algorithm the objective function is linearized and a step is taken along a feasible descent direction. Recently, the approach has been successfully used for finding sparse solutions to problems with convex constraints (see e.g. [17, 22, 21, 27]).
In this section, we describe an efficient version of the Frank-Wolfe algorithm for solving problem (2) and recall some theoretical results about its global convergence (see [21] for further details and proofs). The motivation for using this algorithm as a local minimizer is twofold:

1) the optimization problem to be solved at each step of the algorithm has linear objective function and a closed convex feasible set described by simple constraints. So that, its solution can be computed analitically.

2) it is possibile to reduce the problem dimension at each step of the algorithm, thus obtaining significant savings in terms of computational time.

In order to ease the description of the algorithm we restate problem 2 as follows:

$$\begin{aligned} \min_{x\in\Re^n} \; & f(x) = g(x) + \lambda h(x) \\ s.t. \quad & x \in C, \\ & x_i \geq 0, \; i \in I \subseteq \{1, \dots, n\}, \end{aligned} \tag{7}$$

we denote by Ω the feasible set of the above problem.
Then, here is the outline of the algorithm:

As we can easily see, at each iteration the problem (8) is equivalent to a problem of dimension $n - |I^k|$ and $I^k \subseteq I^{k+1}$, then the problems to be solved are of nonincreasing dimensions. This yields obvious advantages in terms of computational time. We report here the main result about the global convergence of the FW-RD Algorithm to a stationary point [21].

Proposition 2.1. *Let $\{x^k\}$ be a sequence generated by the FW-RD Algorithm*

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

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

Algorithm 1 Frank-Wolfe - Reduced Dimension (FW-RD) Algorithm

```
Require: $x^0 \in \Omega$.
  for $k = 0, 1, \ldots,$ do
    Set $I^{x^k} = \{i \in I : x_i^k = 0\}$ and $\Omega^{x^k} = \{x \in \Omega : x_i = 0, \forall\ i \in I^{x^k}\}$
    Obtain solution $\overline{x}^k$ by solving the following problem:
        $\overline{x}^k = \arg \min_{x \in \Omega^{x^k}} \nabla f(x^k)^T (x - x^k)$        (8)
    if $\nabla f(x^k)^T (\overline{x}^k - x^k) = 0$ then STOP
    else define a feasible descent direction
        $d^k = \overline{x}^k - x^k$
      and generate a new feasible vector
        $x^{k+1} = x^k + \alpha^k d^k$
      with $\alpha^k \in (0, 1]$ a suitably chosen stepsize.
    end if
  end for
```

(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*

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

Suppose there exists a value S *such that* $h_i^{'}(0) \geq S\ \forall\ x_i = 0$ *with* $i \in I$, *then every limit point* $\bar{x}$ *of* $\{x^k\}$ *is a stationary point.*

We notice that the assumption of Proposition 2.1 holds for suitable values of the parameters of the smooth concave approximating functions (3)–(6); so that Algorithm FW-RD can be applied. 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, the stepsize is not computed exactly, but it is replaced by a stepsize α^k satisfying some termination criteria.

2. **Armijo Rule:** In this case, 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:** According to this choice, a fixed stepsize

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

is used. This rule can be adopted when the objective function has some particular properties (e.g. concavity). Anyway, if we rescale or redefine appropriately the direction d^k, we can always use a constant stepsize.

3. Analytical Solution of the Frank-Wolfe Subproblem

In this section, under assumption 1, we show how to analytically compute the solution of the Frank-Wolfe subproblem. In the next proposition, we give the analytical solution for the FW subroblem when C satisfies assumption 1 (a).

Proposition 3.1. *Let $C = \{x \in \Re^n : \ \|x\|^2 \leq 1\}$. The problem*

$$\begin{aligned} \min \quad & c_x' x + c_y' y \\ s.t. \quad & x'x \leq 1, \\ & -y \leq x \leq y. \end{aligned} \tag{9}$$

admits the following solution:

$$x^\star = \begin{cases} 0 & if \ \forall \ i \ |(c_x)_i| \leq (c_y)_i \\ \frac{\tilde{x}}{\|\tilde{x}\|} & otherwise \end{cases} \qquad y^\star = |x^\star|$$

where

$$\tilde{x}_i = \begin{cases} 0 & |(c_x)_i| \leq (c_y)_i \\ sgn[(c_x)_i](c_y)_i - (c_x)_i & |(c_x)_i| > (c_y)_i \end{cases} \qquad i = 1, \ldots, n. \tag{10}$$

Proof. The KKT conditions for Problem (17) are the following:

$$\begin{aligned} c_x + 2\mu x + \sigma - \rho &= 0, && \text{(11a)} \\ c_y - \sigma - \rho &= 0, && \text{(11b)} \\ \mu(\|x\|^2 - 1) &= 0, \quad \mu \geq 0, && \text{(11c)} \\ \sigma'(x - y) &= 0, \quad \sigma \geq 0, && \text{(11d)} \\ \rho'(x + y) &= 0, \quad \rho \geq 0, && \text{(11e)} \\ \|x\|^2 &\leq 1, && \text{(11f)} \\ x - y &\leq 0, && \text{(11g)} \\ -x - y &\leq 0. && \text{(11h)} \end{aligned}$$

We say that $(\bar{x}, \bar{y}) \in \Re^{2n}$ is a KKT pair for problem (17) when multipliers $\bar{\mu} \in \Re$, $\bar{\sigma} \in \Re^n$ and $\bar{\rho} \in \Re^n$ exist such that $(\bar{x}, \bar{y}, \bar{\mu}, \bar{\sigma}, \bar{\rho})$ satisfy (11). Let us consider the two cases:

1. $|(c_x)_i| \leq (c_y)_i$ for all $i = 1, \ldots, n$. It is easy to see that the tuple $(x^\star, y^\star, \mu^\star, \sigma^\star, \rho^\star)$ where

$$x^\star = 0, \quad y^\star = 0, \quad \mu^\star = 0,$$

$$\rho_i^\star = \frac{(c_y)_i + (c_x)_i}{2}, \quad \text{for all } i = 1, \ldots, n,$$

$$\sigma_i^\star = \frac{(c_y)_i - (c_x)_i}{2}, \quad \text{for all } i = 1, \ldots, n,$$

satisfy the KKT conditions.

2. $|(c_x)_i| > (c_y)_i$, for at least an index $i \in \{1, \ldots, n\}$. In this case, it can be seen that the tuple $(x^\star, y^\star, \mu^\star, \sigma^\star, \rho^\star)$ where

$$x^\star = \frac{\tilde{x}}{\|\tilde{x}\|}, \quad y^\star = |x^\star|, \quad \mu^\star = \frac{\|\tilde{x}\|}{2},$$

with $\tilde{x}$ is given by (10), and

$$\sigma_i^\star = \begin{cases} (c_y)_i & \text{if } (c_x)_i < -(c_y)_i < 0 \\ 0 & \text{if } (c_x)_i > \quad (c_y)_i > 0 \\ \frac{(c_y)_i - (c_x)_i}{2} & |(c_x)_i| \leq (c_y)_i, \end{cases} \qquad \rho_i^\star = \begin{cases} 0 & \text{if } (c_x)_i < -(c_y)_i < 0 \\ (c_y)_i & \text{if } (c_x)_i > \quad (c_y)_i > 0 \\ \frac{(c_y)_i + (c_x)_i}{2} & |(c_x)_i| \leq (c_y)_i, \end{cases}$$

satisfy the KKT conditions.

The proof follows by considering that the gradients of the active constraints at $(x^\star, y^\star)$ are linearly independent. □

Now, we report an analogous result for the problem with bound constraints.

Proposition 3.2. *Let $C = \{x \in \Re^n : -M \leq x \leq M\}$. The problem*

$$\begin{array}{ll} \min & c_x' x + c_y' y \\ s.t. & -y \leq x \leq y, \\ & -M \leq x \leq M, \end{array} \tag{12}$$

admits $(x^\star, y^\star)$ as a solution, where

$$x_i^\star = \begin{cases} 0 & \text{if } |(c_x)_i| \leq (c_y)_i \\ -M sgn[(c_x)_i] & \text{otherwise} \end{cases} \qquad y_i^\star = |x_i^\star|,$$

for $i = 1, \ldots, n$.

Proof. The KKT conditions for Problem (23) are the following:

$$c_x + \sigma - \rho + r - s = 0, \tag{13a}$$
$$c_y - \sigma - \rho = 0, \tag{13b}$$
$$\sigma'(x - y) = 0, \quad \sigma \geq 0, \tag{13c}$$
$$\rho'(x + y) = 0, \quad \rho \geq 0, \tag{13d}$$
$$r'(x - M) = 0, \quad r \geq 0, \tag{13e}$$
$$s'(-x - M) = 0, \quad s \geq 0, \tag{13f}$$
$$x - y \leq 0, \tag{13g}$$
$$-x - y \leq 0. \tag{13h}$$

We say that $(\bar{x}, \bar{y}) \in \Re^{2n}$ is a KKT pair for problem (23) when multipliers $\bar{\sigma} \in \Re^n$, $\bar{\rho} \in \Re^n$, $\bar{r} \in \Re^n$ and $\bar{s} \in \Re^n$ exist such that $(\bar{x}, \bar{y}, \bar{\sigma}, \bar{\rho}, \bar{r}, \bar{s})$ satisfy (13). Let us consider the two cases:

1. $|(c_x)_i| \leq (c_y)_i$ for all $i = 1, \dots, n$. It is easy to see that the tuple $(x^\star, y^\star, \sigma^\star, \rho^\star, r^\star, s^\star)$ where
$$x^\star = 0, \quad y^\star = 0, \quad r^\star = 0, \quad s^\star = 0,$$
$$\rho_i^\star = \frac{(c_y)_i + (c_x)_i}{2}, \quad \text{for all } i = 1, \dots, n,$$
$$\sigma_i^\star = \frac{(c_y)_i - (c_x)_i}{2}, \quad \text{for all } i = 1, \dots, n,$$
satisfy the KKT conditions.

2. If $|(c_x)_j| > (c_y)_j$, for at least an index $j \in \{1, \dots, n\}$, then, for $i = 1, \dots, n$, we consider the following cases:

   - $(c_x)_i < -(c_y)_i < 0$. In this case, it can be seen that
   $$\sigma_i^\star = (c_y)_i, \quad \rho_i^\star = 0, \quad s_i^\star = 0, \quad r_i^\star = -((c_x)_i + (c_y)_i), \quad x_i^\star = M, \quad y_i^\star = M;$$
   - $(c_x)_i > (c_y)_i > 0$. In this case, it can be seen that
   $$\sigma_i^\star = 0, \quad \rho_i^\star = (c_y)_i, \quad s_i^\star = (c_x)_i - (c_y)_i, \quad r_i^\star = 0, \quad x_i^\star = -M, \quad y_i^\star = M;$$
   - $|(c_x)_i| \leq (c_y)_i$. In this case, it can be seen that
   $$\sigma_i^\star = \frac{(c_y)_i - (c_x)_i}{2}, \quad \rho_i^\star = \frac{(c_y)_i + (c_x)_i}{2}, \quad s_i^\star = 0, \quad r_i^\star = 0, \quad , x_i^\star = 0, \quad y_i^\star = 0.$$

The proof follows by considering that the gradients of the active constraints at $(x^\star, y^\star)$ are linearly independent. □

4. Sparse PCA

In this section we consider the SPCA problem and its solution via the proposed FW-RD Algorithm. First we shall briefly recall PCA. PCA is a well-established tool for data processing and analysis which allows to reduce high dimensional data to a smaller dimension. Given a real matrix $A \in \Re^{p \times n}$ which encodes p samples of n variables or features, PCA aims at finding a few linear combinations of the variables, the principal components, which are orthogonal to each other and explain as much of the variance in the data as possible. If the rows of matrix A are of zero mean, then the classical PCA problem can be formulated by using the scaled covariance matrix $Q = A'A$ as follows
$$\begin{aligned} x^* = \ & \arg\max \ x'Qx \\ & s.t. \ x'x \leq 1. \end{aligned} \tag{14}$$

The solution vector x^* is said the *loading vector* or the (first) Principal Component (PC) of the data, that is the component that explains the maximum amount of variance in the data. x^* is the eigenvector of Q corresponding to the maximum eigenvalue. Hence, computing all of the PC's of Q amounts to computing the Singular Value Decomposition (SVD) of Q. Usually, the PC's of Q, there including x^*, will have many non-zero components.

Sparse PCA aims at finding the PC's of the covariance matrix by minimizing, at the same time, the number of their non-zero components. In [9] a term penalizing the zero-norm of x is introduced in Problem (14) thus obtaining the following formulation of the SPCA problem:

$$\begin{aligned} x^* = \ & \arg\max \ x'Qx - \lambda\|x\|_0 \\ & s.t. \ x'x \leq 1. \end{aligned} \tag{15}$$

While PCA is numerically easy, Sparse PCA is a hard combinatorial problem. In fact, in [19] it is shown that the subset selection problem for ordinary least squares, which is NP-hard [20], can be reduced to a sparse generalized eigenvalue problem, of which sparse PCA is a particular intance. Hence, researchers are studying ways to make Problem 15 computationally tractable.
A simple approach to Problem (15) consists in solving PCA by neglecting the zero-norm term and then to threshold the loadings with small absolute value to zero [4]. More systematic approaches to the problem appeared in recent years, with various researchers proposing the use of nonconvex algorithms (e.g., SCoTLASS in [14], SLRA in [28] or D.C. based methods [25]) which find modified principal components with zero loadings. The SPCA algorithm in [29] is based on the representation of PCA as a regression-type optimization problem thus allowing for the application of the LASSO [26]. All the mentioned approaches and algorithms require solving non convex problems. Recently in [10] an l1 based semidefinite relaxation for the sparse PCA problem has been proposed.
In this section we propose solution of Problem (15) via the FW-RD method. To this purpose, following [21] and by adding some auxiliary variables, we substitute the zero-norm of vector x in the definition of Problem (15) with a concave separable function thus obtaining the problem

$$\begin{aligned} (x^*, y^*) = \ & \arg\max \ x'Qx - \lambda\sum_{i=1}^{n}\log(\epsilon + y_i) \\ & s.t. \ \ x'x \leq 1, \\ & \qquad -y \leq x \leq y, \end{aligned} \tag{16}$$

where, in particular, we add variables y_i for each $i \in \{1, \dots, n\}$. We note that above problem (16) has the form of Problem (7).
At every iteration of the FW-RD method, we have to solve the following subproblem:

$$\begin{aligned} (x^*, y^*) = \ & \arg\min \ -(Qx^k)'x + \lambda\sum_{i=1}^{n}\frac{y_i}{\epsilon + y_i^k} = c_x'x + c_y'y \\ & s.t. \ \ x'x \leq 1, \\ & \qquad -y \leq x \leq y, \end{aligned} \tag{17}$$

where x^k is the current iterate. In order to ease the understanding of the algorithm, we do not take into account the fact that at iteration k some of the variables can be fixed to zero.
With reference to Problem (17), we know that $(c_y)_i = \frac{\lambda}{\epsilon + y_i^k} > 0$, for all $i = 1, \dots, n$. Then, the FW-RD algorithm described in Section 2 can be specialized by considering that:

- solution of the Frank-Wolfe subproblem is computed as described in Proposition 3.1;
- since the problem (16) is a concave programming problem, the stepsize α^k is fixed to 1.

5. Sparse reconstruction of noisy signals

Many problems in signal/image processing and statistics can be formulated as that of finding a sparse approximate solution to a large scale underdetermined linear system. A widely-studied problem in this context is the sparse representation of signals (see, e.g., [3, 8]). Various media types (i.e. imagery, video and audio) can be sparsely represented using transform-domain methods, and in fact various relevant problems dealing with these media can be easily viewed as the problem of finding sparse solutions to a linear undertermined or ill-conditioned system. In practice, given a dictionary $A \in R^{m\times n}$ of elementary signals and a real noisy signal b, the goal is finding a sparse representation x of signal b in terms of the dictionary A. A quite standard approach consists in solving an ℓ_1 regularized least-squares problem having the following form:

$$\min_{x\in R^n} \frac{1}{2}\|Ax-b\|^2 + \lambda\|x\|_1 \tag{18}$$

The ℓ_1-norm term promotes sparse solutions by forcing small components of the solution vector x to be zero. Problem (18) is strictly related to the *Least Absolute Shrinkage and Selection Operator*, a widely-studied problem in statistics, described for the first time by Tibshirani in [26]:

$$\begin{aligned} &\min_{x\in R^n} \|Ax-b\|_2^2 \\ &s.t. \quad \|x\|_1 \le \tau, \end{aligned} \tag{19}$$

where τ is a nonnegative real parameter regulating the sparsity of the solution. The Basis Pursuit [8] problem:

$$\begin{aligned} &\min_{x\in R^n} \|x\|_1 \\ &s.t. \quad Ax = b, \end{aligned} \tag{20}$$

is also related to Problem (18). Another interesting application of Problem (18) is Compressed Sensing [5, 6, 7]. The idea behind Compressed Sensing is that of encoding a large sparse signal using a relatively small number of linear measurements, and minimizing the ℓ_1-norm in order to decode the signal. In the last decades, Problem (18) has become increasingly popular and various algorithms have been proposed for efficiently solving it (see e.g. [1, 13, 16, 18]). An alternative way to formulate the problem of reconstructing a noisy signal by elementary signals is the following:

$$x^\star = \arg\min \frac{1}{2}\|Ax-b\|^2 + \lambda\|x\|_0 \tag{21}$$

In this section we propose solution of Problem (21) via the FW-RD method. To this purpose, following [21] and by adding some auxiliary variables, we substitute the zero-norm of vector x in the definition of Problem (15) with a concave separable function thus obtaining the problem

$$\begin{aligned} (x^*, y^*) = \; &\arg\min \; \tfrac{1}{2}\|Ax-b\|^2 + \lambda\sum_{i=1}^{n}(1-e^{-\alpha y_i}) \\ &s.t. \; -y \le x \le y, \end{aligned} \tag{22}$$

where, in particular, we add variables y_i for each $i \in \{1,\dots,n\}$. We note that above problem (22) has the form of Problem (7).

As done in Section 4, at every iteration of the FW-RD method, we have to solve the following linear subproblem:

$$\begin{aligned} (x^*, y^*) = \arg\min\ & (Ax^k - b)'Ax + \lambda \sum_{i=1}^{n} \alpha e^{-\alpha y_i^k} y_i = c_x' x + c_y' y \\ s.t.\ & -y \le x \le y, \\ & -M \le x \le M, \end{aligned} \tag{23}$$

where x^k is the current iterate. The last set of constraints ($-M \le x \le M$), when $M > 0$, makes the feasible region of Problem (23) compact. With reference to Problem (23), we know that $(c_y)_i = \lambda \alpha e^{-\alpha y_i^k} > 0$, for all $i = 1, \dots, n$.
Then, the FW-RD algorithm described in section 2 can be specialized by considering that:

- solution of the Frank-Wolfe subproblem is computed as described in Proposition 3.2;
- Stepsize α^k is chosen by means of an Armijo rule.

6. Numerical results on sparse PCA

In this section we report the results obtained by testing our method on two different classes of sparse PCA problems. More precisely as in [15], first we experiment on random data (with an underlying sparse PCA model). Then we consider some real datasets related to the analysis of gene expressions [27]. Further, we compare our method with the method for sparse PCA proposed in [15], namely GPower$_{\ell_0}$.
All the numerical experiments have been conducted using Matlab 7.12 (R2011a) on an Intel core 2 duo with 4GB RAM and running Linux version 2.6.38.

6.1. Random data drawn from a sparse PCA model

In order to generate random data with a covariance matrix having sparse eigenvectors, we follow the procedure proposed in [24]. Let $\Sigma = VDV'$ be a covariance matrix, where the first m columns of $V \in \Re^{n \times n}$ are pre-specified sparse orthonormal vectors. Then, a data matrix $A \in \Re^{p \times n}$ is generated by using a zero-mean normal distribution with covariance matrix Σ, that is, $A \sim N(0, \Sigma)$.
We consider different pairs (p, n) and for each of them we generate 100 data matrices following [15]. We then use Algorithm FW-RD and GPower$_{\ell_0}$ to compute two unit-norm sparse PC's of Q. This can be done by using a standard so-called deflation scheme like that used in [10]. In particular, let z_1 be the computed solution of Problem (15), then z_2 can be obtained by solving again Problem (15) with

$$Q = (A - Az_1 z_1')'(A - Az_1 z_1').$$

In Table 1, we report the obtained results.

For every pair (p, n) we provide the average of the scalar products and computing times. Furthermore, in the column labelled "succ." we report the percentage of problems where the two pre-specified eigenvectors are successfully identified, that is when $|z_1' v_1|$ and $|z_2' v_2|$ are both greater than 0.99. As we can see from the table, the FW-RD Algorithms is competitive with GPower ℓ_0 in terms of CPU time and it is better in terms of success rate.

		GPower ℓ_0					FW-RD				
p	n	$\|z_1'z_2\|$	$\|z_1'v_1\|$	$\|z_2'v_2\|$	succ.	time	$\|z_1'z_2\|$	$\|z_1'v_1\|$	$\|z_2'v_2\|$	succ.	time
10	100	6.05e-03	7.43e-01	7.28e-01	37%	1.21e-02	1.50e-02	7.32e-01	7.10e-01	39%	1.00e-02
11	110	6.40e-03	7.25e-01	7.10e-01	37%	1.31e-02	1.59e-02	7.24e-01	7.05e-01	41%	1.01e-02
12	120	5.92e-03	7.02e-01	6.90e-01	35%	1.35e-02	1.66e-02	6.93e-01	6.74e-01	36%	1.03e-02
13	130	6.86e-03	7.56e-01	7.46e-01	43%	1.21e-02	1.38e-02	7.45e-01	7.27e-01	45%	1.06e-02
14	140	6.38e-03	7.47e-01	7.37e-01	44%	1.24e-02	1.37e-02	7.40e-01	7.22e-01	44%	1.19e-02
15	150	5.72e-03	7.32e-01	7.23e-01	40%	1.30e-02	1.12e-02	7.24e-01	7.08e-01	43%	1.12e-02
16	160	5.35e-03	7.60e-01	7.52e-01	47%	1.27e-02	1.14e-02	7.42e-01	7.31e-01	50%	1.10e-02
17	170	5.40e-03	7.44e-01	7.37e-01	46%	1.32e-02	1.39e-02	7.48e-01	7.34e-01	47%	1.18e-02
18	180	5.31e-03	7.70e-01	7.63e-01	48%	1.31e-02	1.24e-02	7.66e-01	7.53e-01	49%	1.16e-02
19	190	5.42e-03	7.76e-01	7.70e-01	48%	1.30e-02	1.09e-02	7.73e-01	7.60e-01	49%	1.18e-02
20	200	4.74e-03	7.74e-01	7.69e-01	45%	1.37e-02	1.37e-02	7.79e-01	7.64e-01	49%	1.18e-02

Table 1: Performance of GPower ℓ_0 and FW-RD Algorithm on Random data.

6.2. Analysis of gene expressions data

DNA microarrays allow to provide the expression level of tens of thousands of genes across several hundreds of experiments thus constituting the source of a huge quantity of data. The interpretation of all these data is a challenging topic and calls for the use of advanced analitical tools. For more details and inshights on microarrays and gene expression data, we refer to [23] and the references therein. Below we report results on two particular datasets [27] and precisely (a) colon cancer, (b) brown yeast and (c) lymphoma.

In the colon cancer dataset we have the expression of 2000 genes in 62 (22 normal and 40 colon cancer) tissue samples. The goal is that of determining the relevant genes to discriminate between cancerous and normal tissues. In Figure 1 we report the proportion of adjusted variance versus the cardinality of the extracted set of discriminating genes (the so-called trade-off curve) both for FW-RD and GPower. As it can be seen, for this example our method is comparable with GPower.

In the brown yeast dataset we have a total of 208 genes that have to be discriminated based on 79 gene expression data corresponding to different experimental settings. In Figure 2 we report the trade-off curves of the two methods (GPower and FW-RD), from which we can see that FW-RD outperforms GPower for small cardinalities, that is when the underlying combinatorial problem is harder to solve.

In the lymphoma problem the gene expression of 96 samples is measured with microarrays to give 4026 features, 61 of the samples are in classes DLCL, FL or CLL (malignant) and 35 are labelled normal. As in the case of colon cancer data, the goal here is that of determining the relevant genes in discrimination. In Figure 3, we report the trade-off curves of the two methods (GPower and FW-RD). We can see, once again, that FW-RD outperforms GPower for small cardinalities.

7. Numerical results on sparse representation of noisy signals

We compare FW-RD algorithm on a set of sparse signal reconstruction problems with the SpaRSA code [18]. We generated matrix A and vector b according to five different basic compressed sensing scenarios, like those described in [13, 12]. In practice, we first randomly generate matrix A (according to one of the given scenarios), then we choose $b = Ax^\star + \nu$, where ν is a

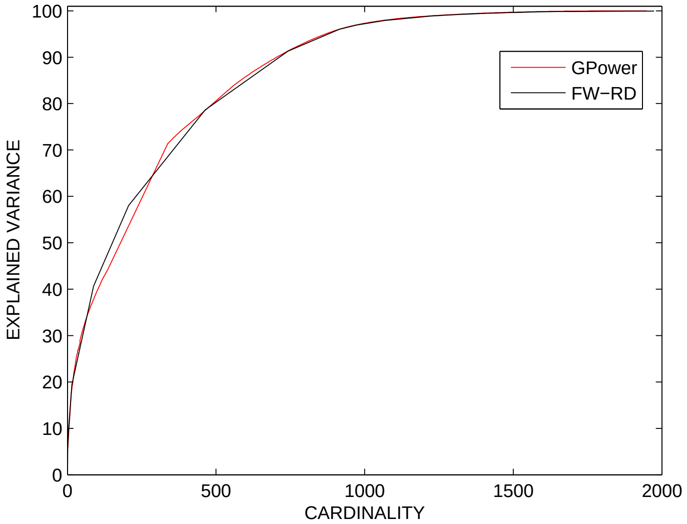

Figure 1: Trade-off curves for gene expressions in colon cancer dataset.

Gaussian white vector with variance 10^{-4} and $x^\star$ is a vector with T randomly placed ±1 spikes and zeroes in the other components.

In Table 2, we report the experimentation and comparison between FW-RD algorithm and SpaRSA. n, m and T denote, respectively, the number of columns, rows of the matrix A and the number of spikes of the sparse solution $x^\star$. $Prob$ denotes the compressed sensing scenario used, namely, partial discrete cosine transform ($Prob = 0$), random ±1 Bernoulli ($Prob = 1$), partial Hadamard ($Prob = 2$), normally distributed random ($Prob = 3$) and scaled normally distributed random ($Prob = 4$) matrix. In the columns labelled time and MSE we report, respectively, the CPU computing time and the mean squared error of the reconstructions with respect to $x^\star$. Every table row reports the avarage results over 10 runs of the algorithms. All the numerical experiments have been conducted using Matlab 7.12 (R2011a) on an Intel core 2 duo with 4GB RAM and running Linux version 2.6.38.

As we can easily see by taking a look at the table, FW-RD outperforms SpaRSA in terms of CPU time in all the scenarios but one (scenario 4), where SpaRSA is slightly better. Furthermore, the FW-RD algorithm gives better results in terms of MSE in all the scenarios but one (scenario 0), where the two methods are comparable. We finally want to notice that the cost in terms of CPU time for the FW-RD algorithm does not depend on the scenario chosen and it does not grow that much even when the size of matrix A is quite large.

8. Conclusions

In this paper we consider a class of ℓ_0-penalized problems with simple constraints and propose a first-order method for their solution. Despite the assumption made on the constraint set C, the considered class of problems is still sufficiently general to encompass many significant applicative problems. We propose a Frank-Wolfe method for the solution of the problem and show that the FW subproblem can be solved analytically which is beneficial to the overall algorithm efficiency. To show the effectiveness and efficiency of the proposed approach, we present numerical results

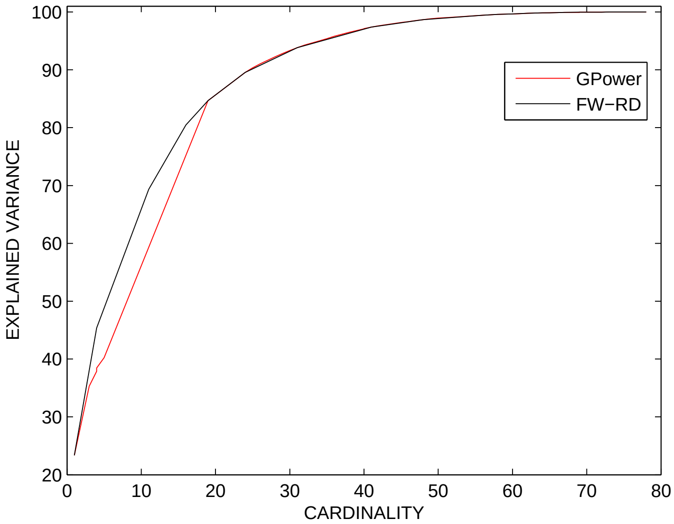

Figure 2: Trade-off curves for gene expressions in yeast dataset.

and comparison with other well-know software packages. In particular, we consider two numerical problems: (a) sparse PCA problems, for which we report comparison with the GPower method [15] and (b) sparse reconstruction of noisy signals problems, for which we report comparison with the SpaRSA method [18]. For the latter class of problem we tested our code on both random generated problems and biological problems. Our method performs well and compare quite favorably with GPower on random generated problems and is slightly superior on biological data. For the former calss of problems (reconstruction of noisy signals), we run our experimentation on five different signal scenario and outperformed SpaRSA both in terms of CPU time and reconstruction error.
To conclude, the results confirm viability of the proposed method and its efficiency when compared with state-of-the-art software packages for the solution of special classes of sparse problems. In this regard, we point out that our method is more general than both GPower and SpaRSA, since it allows to solve a more general class of sparse problems than that addressed by the two competing solvers.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-threshold algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*, 2:183–202, 2009.

[2] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74:121–140, 1996.

[3] A.M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *Siam Review*, 51(1):34–81, 2009.

[4] J. Cadima and I. Jolliffe. Loadings and correlations in the interpretation of principal components. *Journal of Applied Statistics*, 22:203–214, 1995.

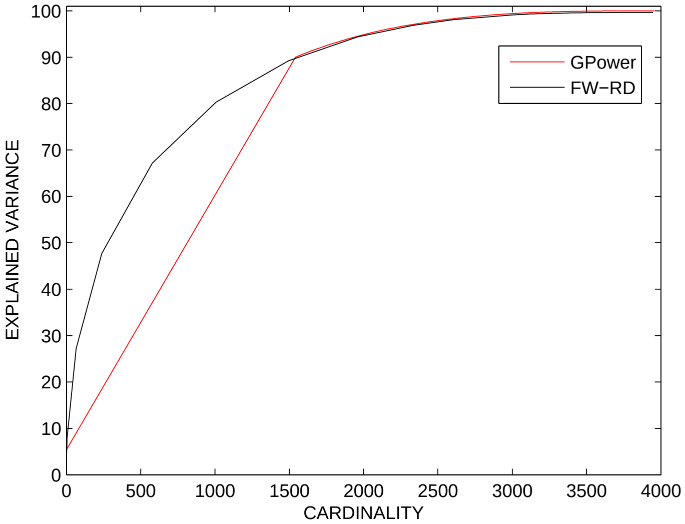

Figure 3: Trade-off curves for gene expressions in lymphoma dataset.

[5] E. Candès and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Comput. Math.*, 6(2):227 – 254, 2006.

[6] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory*, 52(2):489 – 509, 2006.

[7] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[8] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition basis pursuit. *SIAM Rev.*, 43:129–159, 2001.

[9] A. D'Aspremont, F.R. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of machine learning research*, 9:1269–1294, 2008.

[10] A. D'Aspremont, L. El Ghaoui, N.I. Jordan, and G.R.G. Lanckriet. A direct formulation for sparce pca using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[11] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[12] E.T. Hale, W. Yin, and Y. Zhang. `http://www.caam.rice.edu/~optimization/l1/fpc/`.

[13] E.T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[14] I.T. Jolliffe, N.T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

	SpaRSA		FW-RD	
Prob	time	MSE	time	MSE
	$n = 8192; T = 100; m = 2048$			
0	5.484e+00	8.517e-02	2.213e+00	8.829e-02
1	1.643e+01	2.451e-02	2.002e+00	3.461e-05
2	2.279e+00	2.250e-02	2.038e+00	2.153e-03
3	1.727e+01	2.592e-02	2.009e+00	3.668e-05
4	4.760e-01	2.605e-02	9.910e-01	4.862e-05
	$n = 4096; T = 50; m = 1024;$			
0	6.458e+00	9.194e-02	2.337e+00	1.188e-01
1	1.618e+01	2.465e-02	2.176e+00	3.481e-05
2	2.326e+00	2.303e-02	2.258e+00	3.245e-05
3	1.846e+01	2.517e-02	2.198e+00	3.554e-05
4	4.800e-01	2.564e-02	1.011e+00	2.158e-03
	$n = 2048; T = 25; m = 512;$			
0	2.199e+00	9.644e-02	3.550e-01	1.176e-01
1	2.665e+00	2.323e-02	4.990e-01	4.324e-03
2	8.560e-01	2.925e-02	5.440e-01	2.881e-05
3	2.681e+00	2.465e-02	5.930e-01	3.481e-05
4	1.450e-01	2.429e-02	2.250e-01	4.436e-05

Table 2: Performance of SparSA and FW-RD Algorithm on randomly generated SRNS problems.

[15] M. Journeé, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

[16] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[17] O.L. Mangasarian. Machine learning via polyhedral concave minimization. In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, *Applied Mathematics and Parallel Computing Festschrift for Klaus Ritter*, pages 175–188. Physica-Verlag, Germany, 1996.

[18] Figueiredo M.A.T., Nowak R.D., and S.J. Wright. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[19] B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse lda. In *Proceedings ICML*, 2006.

[20] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[21] F. Rinaldi. Concave programming for finding sparse solutions to problems with convex constraints. *Accepted for publication on Optimization Methods and Software.*

[22] F. Rinaldi, F. Schoen, and M. Sciandrone. Concave programming for minimizing the zero-norm over polyhedral sets. *Computational Optimization and Applications*, 46(3):467–486, 2010.

[23] Alessandra Riva, Anne-Sophie Carpentier, Bruno Torrsani, and Alain Hnaut. Comments on selected fundamental aspects of microarray analysis. *Computational Biology and Chemistry*, 29(5):319 – 336, 2005.

[24] H. Shen and J.Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

[25] B.K. Sriperumbudur, D.A. Torres, and G.R.G. Lanckriet. Sparse eigen methods by dc programming. In *Proceedings of the 24th international conference on Machine learning*, pages 831–838, 2007.

[26] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal statistical society, series B*, 58(1):267–288, 1996.

[27] J. Weston, A. Elisseef, and B. Schölkopf. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.

[28] Z. Zhang, H. Zha, and H. Simon. Low rank approximations with sparse factors i: basic algorithms and error analysis. *SIAM journal on matrix analysis and its applications*, 23(3):706–727, 2002.

[29] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.