A DC Programming Approach for Sparse Eigenvalue Problem

Mamadou Thiao MAMADOU.THIAO@INSA-ROUEN.FR
Laboratory of Mathematics LMI, INSA de Rouen, 76801 Saint-Etienne-du-Rouvray Cedex, FRANCE

Tao Pham Dinh PHAM@INSA-ROUEN.FR
Laboratory of Mathematics LMI, INSA de Rouen, 76801 Saint-Etienne-du-Rouvray Cedex, FRANCE

Hoai An Le Thi LETHI@UNIV-METZ.FR
Laboratory of Theoretical and Applied Computer Science, UFR MIM, Metz University, 57045 Metz, FRANCE

Abstract

We investigate the sparse eigenvalue problem which arises in various fields such as machine learning and statistics. Unlike standard approaches relying on approximation of the l_0-norm, we work with an equivalent reformulation of the problem at hand as a DC program. Our starting point is the eigenvalue problem to which a constraint for sparsity requirement is added. The obtained problem is first formulated as a mixed integer program, and exact penalty techniques are used to equivalently transform the resulting problem into a DC program, whose solution is assumed by a customized DCA. Computational results for sparse principal component analysis are reported, which show the usefulness of our approach that compares favourably with some related standard methods using approximation of the l_0-norm.

1. Introduction

Eigenvalue problem is a popular problem and has many applications in science and engineering. One of the uses of eigenvalue problem is the principal component analysis (PCA) in statistics which is a powerful and popular tool for factor analysis and modeling of data. The main aim in PCA is to extract principal components corresponding to directions of maximal variance in data, each principal component being a linear combination of the input variables. Generally, in practice each principal component given by PCA contains *all* the input variables (i.e. all the coefficients of the input variables in the linear combination are typically non-zero), what raises problem of interpretation and human understanding in several cases. To by-pass this problem several techniques, called sparse PCA techniques, were developed in the literature to obtain principal components with a small nonzero coefficients that explain most of the variance present in the data.

The first work we are aware concerning sparse PCA is that of (Cadima & Jolliffe, 1995) who proposed a simple axis rotation and component thresholding for subset selection. Later (Jolliffe et al., 2003) proposed SCoTLASS by enforcing a sparsity constraint on the principal directions by bounding their l_1-norm, leading to a nonconvex procedure. Recent years have witnessed a flurry of research on algorithms and theory for sparse PCA, (Zou et al., 2006) proposed SPCA, a l_1-penalized regression algorithm for PCA using least regression, convex relaxed solutions leading to semidefinite programs (like DSPCA) are proposed by (d'Aspremont et al., 2007; 2008). (Moghaddam et al., 2005) proposed GSPA, a combinatorial optimization algorithm based on bidirectional greedy search, (Sriperumbudur et al., 2007) proposed DC-PCA, a DC programming algorithm (DCA) obtained by penalizing the approximation term proposed by (Weston et al., 2003).

In this paper, we propose a new solution for the sparse eigenvalue problem using DC programming, which does not approximate the l_0-norm as usually, but uses it completely. Our approach sheds a new light on the use of l_0-norm and is particularly interesting for related problems for which approximating l_0-norm is not satisfactory. We first formulate the problem as a mixed integer program and by using an appropriate penalty function, we show that the problem can be reformu-

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

lated as a DC program (minimization of a DC function over a closed convex set) by exact penalty techniques in DC programming. The resulting DC program is handled by the DCA which consists of solving a sequence of quadratically constrained linear programs (QCLP) with a complexity $O(n^2)$ for each (QCLP). DC Programming and DCA were first introduced, in their preliminary form, by Pham Dinh Tao in 1985, and have been extensively developed since 1994 by Le Thi Hoai An and Pham Dinh Tao. It becomes now classic and increasingly popular (see e.g. (Pham Dinh & Le Thi, 1997; 1998; Le Thi et al., 1999; Le Thi & Pham Dinh, 2001; 2005) and http://lita.sciences.univ-metz.fr/∼lethi/). DCA was applied successfully to many large scale (smooth and nonsmooth) nonconvex programs in various domains of applied sciences for which it proved to be very robust and very efficient(see e.g. (Le Thi & Pham Dinh, 2003; Weber et al., 2006; Pham Dinh et al., 2008)). For the sparse eigenvalue problem, we specialize a suitable DCA, taking into account its specific structure. Computational experiments demonstrate that DCA gives correct solution on the artificial data proposed in (Zou et al., 2006). On the well known standard pit props (Jeffers, 1967), benchmark dataset, very used to estimate the performances of the methods for the principal component analysis because of its lack of sparseness and subsequent difficulty of interpretation, DCA performs better in term of sparsity, explained variance and practical use than SPCA, DSPCA and DC-PCA. Finally, DCA gives better results than SPCA and DC-PCA while estimating its performance in high-dimension data by using the well known colon tumor data (Alon et al., 1999).

The paper is organized as follows: In the following two sections we present the variational formulations for the eigenvalue and sparse eigenvalue problems respectively and state different reformulations in order to get appropriate equivalent DC programs for the original problems . We outline the DC programming and DCA in section 4, while section 5 is devoted to the description of the customized DCA for solving the related DC program. Some extensions are reported in section 6. Computational experiments on the datasets mentioned previously are reported in the last section where we analyse the performance of DCA with related standard methods using approximation of the l_0-norm.

Notation. In this paper, $e = (1, ..., 1)^T \in R^n$, and $e^1, ..., e^n$ are the standard basis vectors of R^n. I denotes the $n \times n$ identity matrix, $S^n = \{X \in M_n(R) : X = X^T\}$, $S^n_+ = \{X \in S^n : X$ positive semidefinite$\}$. $\lambda_{\max}(A)$ denotes the maximal eigenvalue of A and $\|x\|_0 := card\{i \in \{1, ..., n\} : x_i \neq 0\}$.

2. Eigenvalue Problem

The variational formulation for the eigenvalue problem is given by

$$\max\left\{x^T Ax : x^T x = 1\right\}, \tag{1}$$

where $x \in R^n$, and $A = (A_{ij})_{i,j=1,...,n}$ a n-by-n symmetric real matrix. (1) is nonconvex, however efficient methods exist which can find a global solution in polynomial time.

In the principal component analysis (PCA) setting, the goal is to extract the r leading eigenvectors of the sample covariance matrix, A_0 symmetric positive semidefinite, as its eigenvectors are equivalent to the loadings of the first r principal components. Usually Hotelling deflation method (White, 1958; Saad, 1998) is used to sequentially extract these eigenvectors. On the t-th iteration, we extract the leading eigenvector of A_{t-1},

$$x_t \in \arg\max\left\{x^T A_{t-1} x : x^T x = 1\right\},$$

$A_t = A_{t-1} - x_t x_t^T A_{t-1} x_t x_t^T$, and the $(t+1)$-st leading eigenvector of A_0 is the leading eigenvector of A_t.

3. Sparse Eigenvalue Problem

The variational formulation for the sparse eigenvalue problem is given by

$$\max\left\{x^T Ax : x^T x = 1, \|x\|_0 \leq k\right\}, \tag{2}$$

where $k \in N$, $1 \leq k < n$. It is nonconvex, discontinuous, combinatorial and NP-hard. Because of the difficulty to directly handle the cardinality constraint, usually relaxed problems are considered. In SCoTLASS (Jolliffe et al., 2003) the l_1-approximation for the l_0-norm is used. Convex relaxations using semidefinite relaxation are proposed in the literature as the DSPCA of (d'Aspremont et al., 2007). There exists another methods in which the cardinality constraint is absorbed into the objective function as the SPCA of (Zou et al., 2006) and the DC-PCA of (Sriperumbudur et al., 2007). All these methods propose solutions of approximated problems which are not equivalent to the original problem. In this work we propose an exact reformulation of (2) by directly minimizing a quadratic objective function over closed convex constraints.

Without loss of generality we assume that $A \in S^n_+$, $A \neq 0$ afterward, indeed if $A \notin S^n_+$, then we choose $\mu > 0$ such that $\mu I + A \in S^n_+$ and we consider the problem $\max\left\{x^T(\mu I + A)x : x^T x = 1, \|x\|_0 \leq k\right\}$, which is equivalent to (2).

By replacing the quadratic equality constraint $x^T x =$

1 in (2) by the inequality constraint $x^T x \leq 1$, we obtain the following problem

$$\max\left\{x^T Ax : x^T x \leq 1, \|x\|_0 \leq k\right\}, \tag{3}$$

which is equivalent to (2) by the following proposition.

Proposition 1 (2) *and* (3) *are equivalent.*

Proof It suffices to show that $\bar{x}^T\bar{x} = 1$, for any solution $\bar{x}$ of (3). Assume by contradiction that $\bar{x}^T\bar{x} < 1$. As $A \in S_+^n$ and $A \neq 0$, we have $\bar{x}^T A\bar{x} \geq \max_i A_{ii} > 0$, thus $\bar{x} \neq 0$ and $\tilde{x} := \frac{\bar{x}}{\sqrt{\bar{x}^T\bar{x}}}$ is a feasible point of (3). $\tilde{x}^T A\tilde{x} = \frac{\bar{x}^T A\bar{x}}{\bar{x}^T\bar{x}} > \bar{x}^T A\bar{x}$ in contradiction with the fact that $\bar{x}$ is a solution of (3). □

A mixed-integer formulation for (3) is given by

$$\begin{cases} \max_{(x,u)} & x^T Ax \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u \leq k, u \in \{0,1\}^n. \end{cases} \tag{4}$$

In the aim of writing (4) under a continuous formulation, we define

$$q(u) := \sum_{j=1}^{n} u_j(1 - u_j) = e^T u - u^T u. \tag{5}$$

Property 1 *q is a finite nonnegative concave function and $u \in \{0,1\}^n \Leftrightarrow (q(u) \leq 0, u \in [0,1]^n)$.*

3.0.1. Formulation 1

Let $t > 0$ and let us consider the penalty program

$$\begin{cases} \min_{(x,u)} & f_t(x,u) = -x^T Ax + tq(u) \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u \leq k, u \in [0,1]^n. \end{cases} \tag{6}$$

The following proposition shows the equivalence between (3) and (6).

Proposition 2 *Let*

$$c := \max_{j=1,...,n}\left[2\sum_{k=1,k\neq j}^{n} |A_{kj}| + A_{jj}\right],$$
$$t > t_0(A) := 2\max\left\{c, \lambda_{\max}(A) - \max_j A_{jj}\right\},$$

(3) *and* (6) *are equivalent in the following sense:*

- *if $\bar{x}$ is a solution of* (3)*, then there exists $\bar{u}$ such that $(\bar{x}, \bar{u})$ is a solution of* (6)*.*
- *if $(\bar{x}, \bar{u})$ is a solution of* (6)*, then $\bar{x}$ is a solution of* (3)*.*

Proof It suffices to show that $q(\bar{u}) = 0$, for any solution $(\bar{x}, \bar{u})$ of (6). Assume by contradiction that $q(\bar{u}) > 0$. We point out how to compute a feasible point (x, u) of (6) such that $f_t(\bar{x}, \bar{u}) > f_t(x, u)$, in contradiction with the fact that $(\bar{x}, \bar{u})$ is a solution of (6). Put $J := \{j \in \{1, ...n\} : 0 < \bar{u}_j < 1 - \bar{u}_j\}$ and $I := \{j \in \{1, ...n\} : 0 < 1 - \bar{u}_j \leq \bar{u}_j\}$ and consider the following cases:
Case $J \neq \varnothing$. Choose $j_0 \in J$ and put $x_j := \bar{x}_j, u_j := \bar{u}_j, \forall j \neq j_0$ and $x_{j_0} = u_{j_0} := 0$.
Case $J = \varnothing$, and $I \neq \varnothing$. If $e^T\bar{u} < k$, then choose $i_0 \in I$ and $\epsilon > 0$ such that $e^T\bar{u} + \epsilon \leq k$ and $\bar{u}_{i_0} + \epsilon \leq 1$ and put $x := \bar{x}$, $u_i := \bar{u}_i, \forall i \neq i_0$ and $u_{i_0} := \bar{u}_{i_0} + \epsilon$,
else choose i such that $A_{ii} = \max_j A_{jj}$ and put $(x, u) := (e^i, e^i)$. □

From formulation 1 we derive a second formulation by replacing the constraint $e^T u \leq k$ by $e^T u = k$ and by removing $e^T u$ of the objective function.

3.0.2. Formulation 2

Consider the problem

$$\begin{cases} \min_{(x,u)} & f_t^1(x,u) := -x^T Ax - tu^T u \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u = k, u \in [0,1]^n, \end{cases} \tag{7}$$

with $t > t_0(A)$.

Proposition 3 (3) *and* (7) *are equivalent in the same sense as in the proposition 2.*

Proof It suffices to show that (6) has a solution $(\bar{x}, \bar{u})$ that satisfies $e^T\bar{u} = k$. Let $(\tilde{x}, \tilde{u})$ be a solution of (6), thus we have $e^T\tilde{u} \leq k$ and $\tilde{u} \in \{0,1\}^n$. To construct these solution $(\bar{x}, \bar{u})$, we set $\bar{x} := \tilde{x}$, $\bar{u}_j := \tilde{u}_j, \forall j$ such that $\tilde{u}_j = 1$ and complete the rest of components of $\bar{u}$ by 0 or 1 to obtain $e^T\bar{u} = k$. □

Even if the constraints are convex, the objective functions of (6) and (7) are nonconvex (concave). The particular structure of the objective functions suggest us to use the DC programming, which is one of the optimization tools to solve this type of problems, that we will introduce in the next section.

4. D.C. Programming and DCA

Let $\Gamma_0(R^n)$ denote the convex cone of all lower semi-continuous proper convex functions on R^n. The vector space of DC functions, $DC(R^n) = \Gamma_0(R^n) - \Gamma_0(R^n)$, is quite large to contain many real-life objective functions and is closed under all operations usually considered in optimization. Consider the standard DC program

$$(P_{dc}) \qquad \alpha = \inf\left\{f(x) := g(x) - h(x) : x \in R^n\right\},$$

where $g, h \in \Gamma_0(R^n)$. A DC program (P_{dc}) is called polyhedral DC program when either g or h is polyhedral convex function(i.e. the pointwise supremum of a finite collection of affine functions). Note that a polyhedral convex function is almost always differentiable, say, it is differentiable everywhere except on a set of measure zero.
Let C be a nonempty closed convex set. Then, the problem

$$\inf\{F(x) := g(x) - h(x) : x \in C\}, \tag{8}$$

can be transformed into an unconstrained DC program by using the indicator function of C ($\chi_C(x) = 0$ if $x \in C, +\infty$ otherwise), i.e.,

$$\inf\{f(x) := \phi(x) - h(x) : x \in R^n\}, \tag{9}$$

where $\phi := g + \chi_C$ is in $\Gamma_0(R^n)$.
Let

$$g^*(y) := \sup\left\{y^T x - g(x) : x \in R^n\right\} \tag{10}$$

be the conjugate function of g. The dual problem of (P_{dc}) is defined by

$$(D_{dc}) \qquad \beta = \inf\{h^*(y) - g^*(y) : y \in R^n\}. \tag{11}$$

Under the natural convention in DC programming that is $+\infty - (+\infty) = +\infty$, and by using the fact that every function $h \in \Gamma_0(R^n)$ is characterized as a pointwise supremum of affine functions, more precisely

$$h(x) := \sup\left\{y^T x - h^*(y) : y \in R^n\right\}, \tag{12}$$

it can be proved that $\alpha = \beta$. There is a perfect symmetry between primal and dual DC programs: the dual of (D_{dc}) is (P_{dc}).

Recall that, for $\theta \in \Gamma_0(R^n)$ and $x_0 \in$, dom$\theta := \{x \in R^n : \theta(x) < +\infty\}$, the subdifferential of θ at x_0, denoted by $\partial\theta(x_0)$, is defined as

$$\partial\theta(x_0) := \left\{y \in R^n : \theta(x) \geq \theta(x_0) + (x - x_0)^T y : \forall x \in R^n\right\} \tag{13}$$

which is a closed convex subset of R^n. It generalizes the derivative in the sense that θ is differentiable at x_0 if and only if $\partial\theta(x_0)$ is reduced to a singleton which is exactly $\{\nabla\theta(x_0)\}$.
The necessary local optimality condition for the primal DC program, (P_{dc}), is

$$\partial h(x^*) \subset \partial g(x^*). \tag{14}$$

The condition (14) is also sufficient for many important classes of DC programs, for example, for DC polyhedral programs, or when the function f is locally convex at x^* (Pham Dinh & Le Thi, 1997; 1998; Le Thi et al., 1999; Le Thi & Pham Dinh, 2001; 2005).

A point that x^* verifies the generalized Kuhn-Tucker condition

$$\partial h(x^*) \cap \partial g(x^*) \neq \varnothing \tag{15}$$

is called a critical point of $g - h$. It follows that if h is polyhedral convex, then a critical point of $g - h$ is almost always a local solution to (P_{dc}).
The transportation of global solutions between (P_{dc}) and (D_{dc}) is expressed by:

Property 2

$$\left[\bigcup_{y^* \in \mathcal{D}} \partial g^*(y^*)\right] \subset \mathcal{P}, \qquad \left[\bigcup_{x^* \in \mathcal{P}} \partial h(x^*)\right] \subset \mathcal{D}, \tag{16}$$

where $\mathcal{P}$ and $\mathcal{D}$ denote the solution sets of (P_{dc}) and (D_{dc}) respectively. The first inclusion becomes equality if h is subdifferentiable in $\mathcal{P} \subset$ dom$g$, and the second inclusion becomes equality if g^* is subdifferentiable in $\mathcal{D} \subset$ domh^*. Under certain technical conditions, this property also holds for the local solutions of (P_{dc}) and (D_{dc}). For example see (Pham Dinh & Le Thi, 1997; 1998; Le Thi et al., 1999; Le Thi & Pham Dinh, 2001; 2005) for more informations.

Property 3 *Let x^* be a local solution to (P_{dc}) and let $y^* \in \partial h(x^*)$. If g^* is differentiable at y^* then y^* is a local solution to (D_{dc}). Similarly, let y^* be a local solution to (D_{dc}) and let $x^* \in \partial g^*(y^*)$. If h is differentiable at x^* then x^* is a local solution to (P_{dc}).*

Based on local optimality conditions and duality in DC programming, the DC Algorithm (DCA) consists in constructing of two sequences $\left\{x^l\right\}$ and $\left\{y^l\right\}$ of trial solutions of the primal and dual programs, respectively, such that the sequences $\left\{g(x^l) - h(x^l)\right\}$ and $\left\{h^*(y^l) - g^*(y^l)\right\}$ are decreasing, and $\left\{x^l\right\}$ (resp.$\left\{y^l\right\}$) converges to a primal feasible solutions $\tilde{x}$ (resp. a dual feasible solution $\tilde{y}$) satisfying the local optimality condition and

$$\tilde{x} \in \partial g^*(\tilde{y}), \qquad \tilde{y} \in \partial h(\tilde{x}). \tag{17}$$

The DCA then yields the next simple scheme:

$$y^l \in \partial h(x^l); \qquad x^{l+1} \in \partial g^*(y^l). \tag{18}$$

In other words, these two sequences $\left\{x^l\right\}$ and $\left\{y^l\right\}$ are determined in the way that x^{l+1} and y^{l+1} are solutions of the convex primal program (P_l) and dual program (D_{l+1}), respectively. These are defined as

$$(P_l) \min\left\{g(x) - h(x^l) - (x - x^l)^T y^l : x \in R^n\right\}, \tag{19}$$

$$(D_{l+1}) \min\left\{h^*(y) - g^*(y^l) - (y - y^l)^T x^{l+1} : y \in R^n\right\}. \tag{20}$$

At each iteration, the DCA performs a double linearization with use of the subgradients of h and g^*. In fact, at each iteration, one replaces in the primal DC program, (P_{dc}), the second component h by its affine minorization $h_l(x) := h(x^l) + (x - x^l)^T y^l$ to construct the convex program (P_l) whose the solution set is nothing but $\partial g^*(y^l)$. Likewise, the second DC component g^* of the dual DC program, (D_{dc}), is replaced by its affine minorization $g_l^*(y) := g^*(y^l) + (y - y^l)^T x^{l+1}$ to obtain the convex program (D_{l+1}) whose $\partial h(x^{l+1})$ is the solution set. One sees thus the DCA works with the convex DC components g and h but not with the DC function f itself. Moreover, a DC function f has infinitely many DC decompositions which have crucial impacts on the performance of the DCA in terms of speed of convergence, robustness, efficiency, and globality of computed solutions. Convergence properties of the DCA and its theoretical basis are described in (Pham Dinh & Le Thi, 1997; 1998; Le Thi et al., 1999; Le Thi & Pham Dinh, 2001; 2005). However, it is worthwhile to summarize the following properties for the shake of completeness :

- DCA is a descent method (without line search). The sequences $(g(x^l)-h(x^l))$ and $(h^*(y^l)-g^*(y^l))$ are decreasing such that

$$g(x^{l+1})-h(x^{l+1}) \leq h^*(y^l)-g^*(y^l) \leq g(x^l)-h(x^l). \quad (21)$$

- If $g(x^{l+1}) - h(x^{l+1}) = g(x^l) - h(x^l)$, then x^l is a critical point of $g - h$ and y^l is a critical point of $h^* - g^*$. In this case, DCA terminates at l^{th} iteration.

- If the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^l\}$ and $\{y^l\}$ are bounded, then every limit point $\tilde{x}$(resp. $\tilde{y}$) of the sequence $\{x^l\}$ (resp. $\{y^l\}$) is a critical point of $g - h$ (resp. $h^* - g^*$).

- DCA has a linear convergence for general DC programs. Especially, for polyhedral DC programs the sequences $\{x^l\}$ and $\{y^l\}$ contain finitely many elements and the algorithm convergences to a solution in a finite number of iterations.

We shall apply all DC enhancement features to solve (7).

5. Sparse eigenvalue by D.C. Programming and DCA

We consider a new approach based on DC programming and DCA to solve (3). The DCA requires a reformulation of the problem so that the objective function is represented by the difference of two convex functions. Then the original problem becomes a DC program in which the DC function is minimized over a convex set. In this section, we introduce a DC reformulation and then present the corresponding DCA.

Algorithm 1 DCA for sparse eigenvalue problem

Input: $A \in S_+^n$, $1 \leq k < n$, $t > 0$, $(x^0, u^0) \in R^n \times R_+^n$ and ϵ the tolerance
Output: (x, u)
Initialize $l := 0$
repeat
  $X^l := 2Ax^l$, $U^l := 2tu^l$
  Compute (x^{l+1}, u^{l+1}) solution of (25)
until $\left|f_t^1(x^{l+1}, u^{l+1}) - f_t^1(x^l, u^l)\right| \leq \epsilon$
$x := x^l$ and $u := u^l$.

According to the previous section a DC formulation of (7) is given by

$$\min\{G(x,u) - H(x,u) : (x,u) \in R^n \times R^n\}, \quad (22)$$

where $G(x,u) := \chi_C(x,u)$, $H(x,u) := x^T Ax + tu^T u$ and C is the feasible set of (7). Then performing DCA for problem (22) amounts to computing the two sequences $\{(x^l, u^l)\}$ and $\{(X^l, U^l)\}$ defined by

$$(X^l, U^l) \in \partial H(x^l, u^l), (x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l). \quad (23)$$

In other words, we have to compute the subdifferentials ∂H and ∂G^*.

$$(X^l, U^l) \in \partial H(x^l, u^l) \Leftrightarrow X^l = 2Ax^l, U^l = 2tu^l, \quad (24)$$

and $\partial G^*(X^l, U^l)$ is the solution set of the following convex program

$$\min\left\{-(X^l)^T x - (U^l)^T u : (x,u) \in C\right\}, \quad (25)$$

of which a solution can be computed in polynomial complexity $O(n^2)$ by using the KKT conditions (Thiao et al., 2009).

The algorithm 1 summarizes the DCA applied to (7) and the following proposition shows that for $k = n$, our DCA algorithm is reduced to the *power iteration algorithm*.

Proposition 4 *Let $k = n$. Then our DCA algorithm is reduced to the power method for eigenvalue computation.*

Proof As $e^T u^l = k = n$ and $u^l \in [0,1]^n$, we have $u^l = e$ for all $l \geq 1$, thus (25) is reduced to $\min\left\{-(Ax^l)^T x : x^T x \leq 1\right\}$, and by applying the KKT conditions to this problem we obtain $x^{l+1} = Ax^l / \left\|Ax^l\right\|_2$. □

6. Extensions

In this section we present exact penalty techniques for some sparse eigenvalue formulations. First, consider the formulation proposed in (El Ghaoui, 2006), in which the l_0-norm term is absorbed in the objective function

$$\max\left\{x^TAx - \rho\|x\|_0 : x^Tx = 1\right\}, \tag{26}$$

where $\rho > 0$. By a similar reasoning, as in the section 3, we show that (26) is equivalent to

$$\begin{cases} \min_{(x,u)} & -x^TAx + \rho e^Tu + tq(u) \\ s.t. & x^Tx \leq 1, |x_j| \leq u_j, \\ & j = 1, ..., n, u \in [0,1]^n, \end{cases} \tag{27}$$

in the sense as in proposition 2 for all

$$t > 2\max\left\{\rho, \max_{j=1,...,n}\left[2\sum_{k=1,k\neq j}^{n}|A_{kj}| + A_{jj}\right]\right\}.$$

Second, even if we used the quadratic concave penalty function $q(u) = \sum_{j=1}^{n} u_j(1-u_j)$ in our previous reformulations (6) and (27), another polyhedral concave penalty function $p(u) := \sum_{j=1}^{n}\min(u_j, 1-u_j)$ could be used to obtain similar reformulations.

7. Experiments and Results

In this section, we illustrate the effectiveness of the proposed method in the context of sparse principal component analysis. We present experiments on an artificial data and different real-life datasets.

In the sparse PCA setting, usually, the sparse eigenvectors of the covariance matrix A are obtained by applying algorithm on the sequence of deflated matrices with the same (or different) k depending on the sparsity requirement. Here we use an orthogonalized projection deflation technique (see (Mackey, 2008) for more details in deflation techniques for sparse PCA): $A_0 = A$, $v_i = (I - V_{i-1}V_{i-1}^T)x_i / \left\|(I - V_{i-1}V_{i-1}^T)x_i\right\|$, $A_i = (I - v_iv_i^T)A_{i-1}(I - v_iv_i^T)$, where $v_1 = x_1$, and v_1,...,v_{i-1} form the columns of V_{i-1}. Since v_1,...,v_{i-1} form an orthonormal basis for the space spanned by x_1,...,x_{i-1} ((x_1, u_1),...,(x_{i-1}, u_{i-1}) are the output of the Algorithm DCA with A_0,...,A_{i-1}), we have $\mathcal{P}_{i-1} = V_{i-1}V_{i-1}^T$, the orthogonal projection. The cummulative variance is then calculated as $\sum_i v_i^TA_iv_i$.

7.1. Artificial Data

We consider the simulation example proposed in (Zou et al., 2006), in this example three hidden factors are created:

$$V_1 \sim \mathcal{N}(0, 290), V_2 \sim \mathcal{N}(0, 300),$$
$$V_3 = -0.3V_1 + 0.925V_2 + \epsilon, \epsilon \sim \mathcal{N}(0, 1),$$

Table 1. Loadings and explained variance for the first two principal components of the artificial example for DCA.

	PC	X_1	X_2	X_3	X_4	X_5	X_6
	1	0	0	0	0	.5	.5
DCA	2	.5	.5	.5	.5	0	0

	PC	X_7	X_8	X_9	X_{10}	Explained variance
	1	.5	.5	0	0	40.9%
DCA	2	0	0	0	0	39.5%

with V_1, V_2 and ϵ independent. Afterward, 10 observed variables are generated as follows:

$$X_i = V_j + \epsilon_i^j, \epsilon_i^j \sim \mathcal{N}(0, 1),$$

with $j = 1$ for $i = 1, ..., 4$, $j = 2$ for $i = 5, ..., 8$, and $j = 3$ for $i = 9, 10$ and ϵ_i^j independent for $j = 1, 2, 3$, $i = 1, ..., 10$.

The variance of the three underlying factors is 290, 300 and 283.8, respectively. The number of variables associated with the three factors are 4, 4 and 2. Therefore V_2 and V_1 are almost equally important, and they are much more important than V_3. The first two principal components together explain more than 99% of the total variance. These facts suggest that we only need to consider two derived variables with sparse representations. Ideally, the first derived variable should recover the factor V_2 only using (X_5, X_6, X_7, X_8), and the second derived variable should recover the factor V_1 only using (X_1, X_2, X_3, X_4).

Quite as DSPCA, SPCA and SCoTLASS (see (d'Aspremont et al., 2007)) and by taking $k = 4$ for the first two principal components, our DCA algorithm finds the correct sparse principal components of the two first principal components and the results are summarized in Table 1.

7.2. Pit Props Data

The pit props dataset (Jeffers, 1967) has become a standard benchmark example to test sparse PCA algorithms. The first six principal components (PCs) capture 87% of the total variance and so all these other methods compare their explanatory power using six sparses principal components. As it was shown in (Sriperumbudur et al., 2007) that DC-PCA gave a better result than DSPCA and SPCA with 13 non-zero loadings and 77.1% of the total variance for the six first sparse principal components, we are going to use it for our comparison. Table 2 shows the first 3 PCs for SPCA, DSPCA, DC-PCA and the first 6 PCs for our DCA algorithm for sparse PCA. With the same cardinality pattern $(k_1, k_2, k_3, k_4, k_5, k_6) := (6, 2, 2, 1, 1, 1)$ with 13 non-zero loadings, our DCA captures almost the same variance (77.05%). We observe that all of the principal components $C_1, ..., C_6$ generated by DCA

Table 2. Pit Props: Loadings for the first three principal components for SPCA, DSPCA, and DC-PCA and the six PCs for our DCA algorithm. SPCA, DSPCA, and DC-PCA loadings are taken from (Zou et al., 2006), (d'Aspremont et al., 2007) and (Sriperumbudur et al., 2007) respectively.

	PC	x_1	x_2	x_3	x_4	x_5	x_6	x_7
SPCA	C_1	-.477	-.476	0	0	.177	0	-.250
	C_2	0	0	.785	.620	0	0	0
	C_3	0	0	0	0	.640	.589	.492
DSPCA	C_1	-.560	-.583	0	0	0	0	-.263
	C_2	0	0	.707	.707	0	0	0
	C_3	0	0	0	0	0	-.793	-.610
DC-PCA	C_1	.449	.459	0	0	0	0	.374
	C_2	0	0	.707	.707	0	0	0
	C_3	0	0	0	0	0	.816	.578
DCA	C_1	-.444	-.453	0	0	0	0	-.379
	C_2	0	0	.707	.707	0	0	0
	C_3	0	0	0	0	.694	.721	0
	C_4	0	0	0	0	0	0	0
	C_5	0	0	0	0	0	0	0
	C_6	0	0	0	0	0	0	0

	PC	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
SPCA	C_1	-.344	-.416	-.400	0	0	0
	C_2	-.021	0	0	0	.013	0
	C_3	0	0	0	0	0	-.015
DSPCA	C_1	-.099	-.371	-.362	0	0	0
	C_2	0	0	0	0	0	0
	C_3	0	0	0	0	0	.012
DC-PCA	C_1	.332	.403	.419	0	0	0
	C_2	0	0	0	0	0	0
	C_3	0	0	0	0	0	0
DCA	C_1	-.341	-.403	-.419	0	0	0
	C_2	0	0	0	0	0	0
	C_3	0	0	0	0	0	0
	C_4	0	0	0	0	0	1
	C_5	0	0	0	1	0	0
	C_6	0	0	0	0	-1	0

Table 3. Pits Props: variation of the explained variance following k for the first principal component.

k	1	2	3	4	5	6	7
VARIANCE %	7.69	15.03	19.04	22.56	26.20	29.00	30.74

k	8	9	10	11	12	13
VARIANCE %	31.30	31.83	32.10	32.37	32.44	32.45

satisfy the orthonormality property ($C_i^T C_i = 1$ and $C_i^T C_j = 0, \forall i \neq j$) which is not the case for DC-PCA. Another advantage is that in DCA the sparsity requirement is explicitly mentioned and that in DC-PCA it is difficult to set the regularization parameter to attain a given sparsity. Table 3 shows the variation of the explained variance following k for the first principal component.

7.3. Colon Cancer Data

The colon cancer data (Alon et al., 1999) consist of 62 tissue samples (22 normal and 40 cancerous) with the gene expression profiles of $n = 2000$ genes extracted from DNA micro-array data. We consider its first 5 principal components which explain 70% of the total variance. The figure 1 shows that DCA gives much better results than SPCA and DC-PCA. DC-PCA explains only 62% of cumulative variance with more than

Table 4. Colon Cancer: variation of the explained variance following k for the first principal component.

k	200	400	600	800	1000	1200	1400
VARIANCE %	7.70	14.35	20.22	25.51	30.25	34.41	38.11

k	1600	1800	2000
VARIANCE %	41.30	43.76	44.96

6000 for the cumulative cardinality, whereas our DCA algorithm explains 65.94% of cumulative variance with 5000 for cumulative cardinality with the cardinality pattern $(k_1, k_2, k_3, k_4, k_5) := (1800, 800, 800, 800, 800)$. In Table 4 we represent the variation of the explained variance following k for the first principal component.

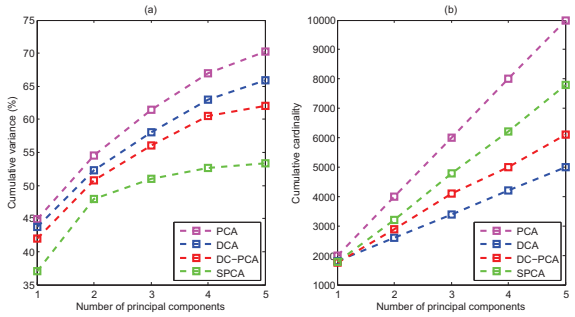

Figure 1. Colon Cancer (a) cumulative variance (b) cumulative cardinality for the first 5 sparse principal components.

8. Summary and Future Work

We have proposed a sparse eigenvalue algorithm using the DC programming and DCA. Our method differs substantially from previous works and approaches in sparse PCA. A difference that begins with the *exact reformulation* of the problem at hand as a DC program (minimization of a DC function over a convex closed set), that is original knowing that, because of the discontinuity of the the l_0-norm, standard methods must resort to its approximation (without having equivalence between the problem and its approximated one). The simplicity of the reformulations and the better results provided by DCA on different well-known data for testing methods for sparse PCA, show that DCA is efficient and promising for sparse PCA. We have also proposed some reformulations which could allow various approaches.

We are extending this technique to other problems involving l_0-norm and investigating global optimization techniques.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues. *Cell Biology*, 96:6745–6750, 1999.

Cadima, J. and Jolliffe, I. T. Loadings and correlations in the interpretation of principal components. *Applied Statistics*, pp. 203–214, 1995.

d'Aspremont, A., El Ghaoui, L., Jordan, M. I., and Lanckriet, G. R. G. A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

d'Aspremont, A., Bach, F., and Ghaoui, L. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

El Ghaoui, L. On the quality of a semidefinite programming bound for sparse principal component analysis. *arXive.org*, 2006.

Jeffers, J. Two case studies in the application of principal components. *Applied Statistics*, 16:225–236, 1967.

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

Le Thi, H. A. and Pham Dinh, T. The dc (difference of convex functions) programming and dca revisited with dc models of real world non convex optimization problems. *Annals of Operations Research*, 133: 23–46, 2005.

Le Thi, H.A. and Pham Dinh, T. A continuous approach for globally solving linearly constrained quadratic zero-one programming problems. *Optimization*, 50(1-2):93–120, 2001.

Le Thi, H.A. and Pham Dinh, T. Large scale molecular optimization from distance matrices by a dc optimization approach. *SIAM Journal on Optimization*, 4(1):77–116, 2003.

Le Thi, H.A., Pham Dinh, T., and Le Dung, M. Exact penalty in dc programming. *Vietnam Journal of Mathematics*, 27(2):169–179, 1999.

Mackey, L. Deflation methods for sparse pca. In *Proceedings of the Conference Neural Information Processing Systems (NIPS 2008)*, 2008.

Moghaddam, B., Weiss, Y., and Avidan, S. Spectral bounds for sparse pca: Exact and greedy algorithms. In *Proceedings of the Conference Neural Information Processing Systems (NIPS 2005)*, 2005.

Pham Dinh, T. and Le Thi, H.A. Convex analysis approaches to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22 (1):287–367, 1997.

Pham Dinh, T. and Le Thi, H.A. D.c. optimization algorithms for solving the trust region subproblem. *SIAM J. Optim.*, pp. 476–505, 1998.

Pham Dinh, T., Le Thi, H.A., and Akoa, F. Combining dca and interior point techniques for large-scale nonconvex quadratic programming. *Optimization, Methods and Softwares*, 23(4):609–629, 2008.

Saad, Y. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Automat. Contr.*, 33:290–297, 1998.

Sriperumbudur, B. K., Torres, D. A., and Lanckriet, G.R.G. Sparse eigen methods by dc programming. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pp. 831–838, 2007.

Thiao, M., Pham Dinh, T., and Le Thi, H.A. Solutions of a linear program with an additional euclidean unit ball constraint by a customized polynomial algorithm. Technical report, Laboratory of Mathematics, LMI, Insa of Rouen, Saint-Etienne-du-Rouvray cedex, France, 2009.

Weber, S., Nagy, A., Schüle, T., Schnörr, C., and Kuba, A. A benchmark evaluation of large-scale optimization approaches to binary tomography. In *Proceedings of the Conference on Discrete Geometry on Computer Imagery (DGCI 2006)*, volume 4245, 2006.

Weston, J., Elisseeff, A., Schölkopf, B., and M., Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.

White, P. The computation of eigenvalues and eigenvectors of a matrix. *Journal of the Society for Industrial and Applied Mathematics*, 6(4):393–437, 1958.

Zou, H., Hastie, T., and Tibshirani, R. Sparse principal component analysis. *J. Comput. Graphical Statist.*, 15:265–286, 2006.