The Sparse PCA Problem: Optimality Conditions and Algorithms

Amir Beck, Yakov Vaisbourd

Abstract

Sparse principal component analysis (PCA) addresses the problem of finding a linear combination of the variables in a given data set with a sparse coefficients vector that maximizes the variability of the data. This model enhances the ability to interpret the principal components, and is applicable in a wide variety of fields including genetics and finance, just to name a few. We suggest a necessary coordinate-wise-based optimality condition, and show its superiority over the stationarity-based condition that is commonly used in the literature, and which is the basis for many of the algorithms designed to solve the problem. We devise algorithms that are based on the new optimality condition, and provide numerical experiments that support our assertion that algorithms which are guaranteed to converge to stronger optimality condition, perform better than algorithms that converge to points satisfying weaker optimality conditions.

Principal component analysis (PCA) is a well known data-analytic technique that linearly transforms a given set of data to some equivalent representation. This transformation is defined in such a manner that any variable in the new representation, called a *principal component* (PC), expresses most of the variance in the data which is not expressed by the PCs that preceded it. The linear combination defining each of the PCs is given by a coefficients (also termed *loadings*) vector. In terms of the covariance (or correlation) matrix of the data, the coefficients vector of the k-th PC is the eigenvector that corresponds to the k-th largest eigenvalue [9]. One major drawback of PCA is that commonly the coefficients vectors are dense, i.e. each PC is a linear combination of much, if not most, of the original variables, which causes a difficulty in interpreting the obtained PCs. This disadvantage encouraged a wide interest in the sparsity constrained version of PCA, which imposes an additional constraint, enforcing the coefficients vector not to exceed some predetermined

sparsity level s. Thus, for example, the first sparse coefficients vector is determined by solving

$$\text{(SPCA)} \qquad \begin{array}{ll} \max_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{s.t.} & \|\mathbf{x}\|_2 \leq 1, \\ & \|\mathbf{x}\|_0 \leq s, \end{array}$$

where the l_0-norm[1] is defined as $\|\mathbf{x}\|_0 = |\{i : x_i \neq 0\}|$ and $\mathbf{A}$ is the covariance matrix. Consecutive vectors are found by resolving problem (SPCA) after applying a deflation scheme [5, 14]. Enforcing sparsity on the coefficients vector is commonly acceptable in some applications. For example, in the exploration of micro-array gene expression patterns, PCA is utilized in order to classify different tissues according to their gene expression. It is also desirable that such discrimination can be executed by utilizing only a small subset of the genes, thus encouraging sparse solutions [15]. The desire to obtain interpretable coefficient vectors is not the only reason to favor a sparse coefficients vector. For example, some financial applications will prefer sparse solutions in order to reduce transaction costs [6]. Clearly, incorporating an additional sparsity constraint will provide a PC that, generally, does not explain all of the variance which is explained by the regular PC; nevertheless, in such applications, this sacrifice is acceptable with respect to the obtained benefits. The l_0-constrained formulation given in (SPCA) is merely one of several alternative formulations considered in the literature. The common alternatives, which are the result of relaxation, penalization or both, are given in Table 1.

Table 1: Alternative formulations for sparse PCA.

Description	Mathematical Formulation
l_1-constrained	$\max\left\{\mathbf{x}^T\mathbf{A}\mathbf{x} : \|\mathbf{x}\|_1 \leq \alpha, \|\mathbf{x}\|_2 \leq 1, \mathbf{x} \in \mathbb{R}^n\right\}$
l_0-penalized	$\max\left\{\mathbf{x}^T\mathbf{A}\mathbf{x} - \gamma\|\mathbf{x}\|_0 : \|\mathbf{x}\|_2 \leq 1, \mathbf{x} \in \mathbb{R}^n\right\}$
l_1-penalized	$\max\left\{\mathbf{x}^T\mathbf{A}\mathbf{x} - \gamma\|\mathbf{x}\|_1 : \|\mathbf{x}\|_2 \leq 1, \mathbf{x} \in \mathbb{R}^n\right\}$

The sparse PCA problem is difficult non-convex problem, and can be optimally solved only for small scale problems by performing exhaustive or a branch and bound search over all possible support sets [16]. Thus, in order to handle large scale problems, the algorithms

[1] We note that the l_0-norm is not actually a norm since it does not satisfy the absolute homogeneity property.

proposed in the literature are seeking to find an approximate solution. One of the first methods, suggested by Cadima and Jolliffe [4], is to threshold the $n - s$ smallest, in absolute value, elements of the dominant eigenvector. Unfortunately, this remarkably simple approach is known to frequently provide poor results. In [16] Moghaddam et al. proposed several greedy methods. An advantage of these methods is that they produce a full path of solutions (i.e. a solution for each of the values of sparsity level up to s), but the necessity to perform a large amount of eigenvalue computations at each step render them quite computationally expensive. In [7], d'Aspremont et al. proposed an approximate greedy approach that obviates the necessity to perform most of the eigenvalue computations by evaluating a lower bound on the eigenvalues, which results in a substantial reduction of computation time. Another approach presented by d'Aspremont et al. in [7], and earlier in [5], is to consider a semidefinite programming formulation with a rank constraint for some of the models of PCA given in Table 1. These equivalent formulations are still hard non-convex problems, and thus a relaxed model is solved and an approximate solution is derived for the original problem. The algorithms used to solve the SDP relaxations are not applicable for large scale problems, rendering this approach as non-scalable. In [10], encouraged by the LASSO approach suggested for regression [21], Jolliffe et al. proposed the l_1-constrained formulation under the name SCoTLass (simplified component technique LASSO), which is a relaxation of (SPCA). In practice, the numerical study was conduced on the l_1-penalized version by implementing the projected gradient algorithm. An alternating minimization scheme to solve the l_1-constrained singular value decomposition (SVD) was proposed in [22]. Another work that addressed the l_1-constrained formulation was motivated by the expectation maximization algorithm for probabilistic PCA [19]. Even though the work addressed the l_1-constrained formulation, the sequence generated by the method in [19] is guaranteed to be s-sparse. Penalized versions were also considered extensively. In [24] Zou et al. formulated the sparse PCA as a regression-type model, where the i-th principal component was approximated by the linear combination of the original variables. A LASSO (l_1) and ridge (l_2) penalties are imposed on the coefficients vector forming the elastic net model that generalizes the LASSO [23] and an alternating minimization algorithm, called SPCA, was proposed. In [18] Shen and Huang proposed several iterative schemes to solve the penalized versions via regularized SVD. These methods were considered further in [11], where a gradient scheme was proposed and a convergence analysis that was missing in [18] was also provided.

Recently, Luss and Teboulle showed in [13] that the seemingly different methods pro-

posed in [11, 18, 19, 20, 22, 24] are some particular realizations of the conditional gradient algorithm with unit step-size. The work [13] proposed a unified algorithmic framework which they refer to as ConGradU and established convergence results showing that the algorithm produces a point satisfying some necessary first order optimality criteria. Some novel schemes are provided. One of them addresses directly the l_0-constrained formulation of sparse PCA.

As already mentioned, none of the methods listed above can guarantee to produce an optimal solution. In addition, there does not seem to be a verifiable necessary and sufficient global optimality condition for (SPCA), and hence there is no efficient way to check if a given vector is the global optimal solution. Therefore, the comparison of the methods in the literature is based solely on numerical experiments without providing any theoretical justification for the advantage of a certain method over the others. However, most of the algorithms just listed will produce a solution that satisfies some necessary optimality condition. In a recent work, Beck and Eldar [2] employed some of the aforesaid conditions in order to provide an insight regarding the success of the corresponding algorithms. Under the framework of minimizing a continuously differentiable function subject to a sparsity constraint, several necessary optimality conditions were presented. The relations between the different optimality conditions were established, showing that some of the conditions are stronger (that is, more restrictive) than others. An extension problems over sparse symmetric sets was considered in [3]. In this paper we adopt this methodology in order to establish an equivalent hierarchy for two necessary optimality conditions for (SPCA). The first condition that we consider is a first order condition that was originally presented in [13]. We will refer to it as the *complete (co) stationarity* condition. Much of the existing algorithms in the literature are actually guaranteed to converge to a co-stationary point. The second condition, which we call *coordinate-wise (CW) maximality*, is a generalization of one of the conditions considered in [2], and it essentially states that the function value cannot be improved by making changes of at most two vectors.

In the following section we will explicitly define the conditions under consideration. In Section 2, we will establish the relation between the conditions, showing that the CW-maximality condition is stronger (that is, more restrictive) than co-stationarity. In Section 3 we will introduce algorithms that produce points satisfying the aforementioned conditions and finally, in Section 4, we will provide a numerical study on simulated and real life data that supports our assertion that algorithms that correspond to stronger conditions are more likely to provide better results.

1 Necessary Optimality Conditions

Throughout the paper we consider the following sparsity constrained problem:

$$\text{(P)} \qquad \begin{array}{ll} \max & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in S, \end{array}$$

where f is a continuously differentiable convex function over $\mathbb{R}^n$ and

$$S = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \leq 1, \|\mathbf{x}\|_0 \leq s\}.$$

In this section, we will present two necessary optimality conditions for the general model (P). Although our main motivation is to study the sparse PCA problem, we will nonetheless consider the general model (P), since our results are also applicable in this general setting.

Prior to presenting the optimality conditions, we will introduce in the following subsection some notation and definitions that will be used in our analysis.

1.1 Notation and Definitions

A subvector of given vector $\mathbf{x} \in \mathbb{R}^n$ corresponding to a set of indices $T \subseteq \{1, 2, \ldots, n\}$ is denoted by $\mathbf{x}_T$, and the principal submatrix of $\mathbf{A}$ consisting of the rows and columns corresponding to T is denoted by $\mathbf{A}_T$. Similarly, we will denote the subvector of the gradient $\nabla f(\mathbf{x})$ corresponding to the indices in T by $\nabla_T f(\mathbf{x})$. The sign of a given $\alpha \in \mathbb{R}$ is denoted by sgn(α) and is equal to 1 for $\alpha \geq 0$ and -1 for $\alpha < 0$. The support set of some arbitrary vector $\mathbf{x}$ will be denoted by $I_1(\mathbf{x}) = \{i : x_i \neq 0\}$ and its complement by $I_0(\mathbf{x}) = \{i : x_i = 0\}$. For a given vector $\mathbf{x} \in \mathbb{R}^n$ and an integer $s \in \{1, 2, \ldots, n-1\}$, we will define $M_s(\mathbf{x})$ to be the s-th largest absolute value component in $\mathbf{x}$. For such $\mathbf{x}$ and s, we will define the sets $I_>(\mathbf{x}, s), I_=(\mathbf{x}, s)$ and $I_<(\mathbf{x}, s)$ as follows:

$$\begin{aligned} I_>(\mathbf{x}, s) &\equiv \{i : |x_i| > M_s(\mathbf{x})\}, \\ I_=(\mathbf{x}, s) &\equiv \begin{cases} \{i : |x_i| = M_s(\mathbf{x})\} & \|\mathbf{x}\|_0 \geq s, \\ \emptyset & \|\mathbf{x}\|_0 < s, \end{cases} \\ I_<(\mathbf{x}, s) &\equiv \begin{cases} \{i : |x_i| < M_s(\mathbf{x})\} & \|\mathbf{x}\|_0 \geq s, \\ \{i : x_i = 0\} & \|\mathbf{x}\|_0 < s. \end{cases} \end{aligned}$$

We will also define $I_{\geq}(\mathbf{x}, s) = I_{>}(\mathbf{x}, s) \cup I_{=}(\mathbf{x}, s)$ and $I_{\leq}(\mathbf{x}, s) = I_{<}(\mathbf{x}, s) \cup I_{=}(\mathbf{x}, s)$. Obviously, the sets $I_{>}(\mathbf{x}, s), I_{=}(\mathbf{x}, s)$ and $I_{<}(\mathbf{x}, s)$ form a partition of $\{1, 2, \ldots, n\}$. Also, when $\|\mathbf{x}\|_0 < s$, we have that $I_{>}(\mathbf{x}, s) = I_1(\mathbf{x}), I_{=}(\mathbf{x}) = \emptyset$ and $I_{<}(\mathbf{x}, s) = I_0(\mathbf{x})$.
The sets defined above posses some convenient and elementary properties which are given in Lemma 1.1 below. Since all the properties stated in the lemma are rather simple consequences of the definition of the sets $I_{>}(\mathbf{x}, s), I_{=}(\mathbf{x}, s), I_{<}(\mathbf{x}, s)$, the proof is omitted.

Lemma 1.1. *1. If $\mathbf{x} \neq \mathbf{0}$, then $I_{\geq}(\mathbf{x}, s) \neq \emptyset$.*

2. If $|I_{\geq}(\mathbf{x}, s)| < s$ then $x_j = 0$ for all $j \in I_{<}(\mathbf{x}, s)$.

3. For any $i \in I_{>}(\mathbf{x}, s), j \in I_{=}(\mathbf{x}, s)$ and $k \in I_{<}(\mathbf{x}, s)$, it holds that $|x_i| > |x_j| > |x_k|$.

We will frequently use the notation

$$R_s(\mathbf{x}) \equiv \{T : I_{>}(\mathbf{x}, s) \subseteq T \subseteq I_{\geq}(\mathbf{x}, s), |T| = \min\{s, |I_{\geq}(\mathbf{x}, s)|\}\}$$

for the set containing all the subsets of indices corresponding to the nonzero s largest in absolute value components of a given vector $\mathbf{x}$. When $\|\mathbf{x}\|_0 < s$, there are less than s nonzero elements in $\mathbf{x}$, and the above definition actually amounts to $R_s(\mathbf{x}) = \{I_1(\mathbf{x})\}$. However, when $\|\mathbf{x}\|_0 \geq s$, there might be more than one set of indices corresponding to the s largest absolute value components of $\mathbf{x}$. For example, consider the vector $\mathbf{x} = (3, 2, 1, 1, 1, 0, 0)^T$ and the sparsity level $s = 3$. Then,

$$R_3(\mathbf{x}) = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}\}.$$

On the other hand, in the following examples, the set contains a single subset:

$$R_3((0, -5, 4, -3, 2, 0)^T) = \{\{2, 3, 4\}\}, R_3((0, 0, 4, -3, 0, 0)^T) = \{\{3, 4\}\}.$$

The *hard thresholding* operator maps a vector $\mathbf{x} \in \mathbb{R}^n$ to the set of vectors that are generated by keeping the s largest absolute value components of $\mathbf{x}$ and setting all the others to zeros. This operator, which we denote by H_s, is formally defined by

$$H_s(\mathbf{x}) \equiv \bigcup_{T \in R_s(\mathbf{x})} \{\mathbf{y} : \mathbf{y}_T = \mathbf{x}_T, \mathbf{y}_{\bar{T}} = \mathbf{0}\}.$$

Thus, for example

$$H_3((3, 2, 1, 1, 1, 0, 0)^T) = \{(3, 2, 1, 0, 0, 0, 0)^T, (3, 2, 0, 1, 0, 0, 0)^T, (3, 2, 0, 0, 1, 0, 0)^T\}.$$

1.2 Complete (co) - Stationarity

The first condition that we consider was presented for the sparse PCA problem in [13]. We refer to it as the complete (co) stationarity condition.

Definition 1.1 (co-stationarity). *Let* $\mathbf{x}$ *be a feasible solution of* (P). *Then* $\mathbf{x}$ *is called a co-stationary point of* (P) *over* S *if it satisfies:*

$$\langle \nabla f(\mathbf{x}), \mathbf{v} - \mathbf{x} \rangle \leq 0 \qquad \forall \mathbf{v} \in S.$$

This is probably the most elementary first order condition for constrained differentiable optimization problems. The work [13] provided a unified framework for several algorithms designed to solve different formulations of sparse PCA (see Table 1). Actually, [13] considered the co-stationarity condition over a general nonempty compact set instead of S, and for this general case, the following proposition, which was originally established in [17] was recalled. This result follows from the convexity of the objective function.

Proposition 1.1. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a continuous differentiable convex function over* $\mathbb{R}^n$, *and let* C *be a nonempty compact set. If* $\mathbf{x}$ *is a global maximum of* f *over* C, *then* $\mathbf{x}$ *is a co-stationary point over* C, *meaning that* $\langle \nabla f(\mathbf{x}), \mathbf{v} - \mathbf{x} \rangle \leq 0$ *for any* $\mathbf{v} \in C$.

1.3 CW-Maximality

The second necessary optimality condition that we will consider is coordinate-wise maximality. This optimality condition is in fact a type of a local optimality condition, stating that a given point $\mathbf{x}$ is a minimizer over a neighborhood consisting of all feasible points that are different by at most two coordinates. We will denote the corresponding neighborhood by

$$S_2(\mathbf{x}) = \{\mathbf{z} : \|\mathbf{z} - \mathbf{x}\|_0 \leq 2, \mathbf{z} \in S\}.$$

The formal definition of a CW-maximum point follows.

Definition 1.2 (CW-maximum point). *Let* $\mathbf{x}$ *be a feasible solution of* (P). *Then* $\mathbf{x}$ *is called a coordinate-wise (CW) maximum point of* (P) *if* $f(\mathbf{x}) \geq f(\mathbf{z})$ *for every* $\mathbf{z} \in S_2(\mathbf{x})$.

Obviously, CW-maximality, by its definition, is a necessary optimality condition.

Proposition 1.2. *Let* $\mathbf{x}$ *be an optimal solution to* (P). *Then* $\mathbf{x}$ *is an CW-maximum point.*

2 Optimality Conditions Hierarchy

Our main result in this section is that CW-maximality is a stronger (that is, more restrictive) optimality condition than co-stationarity. This result also has an impact on the performance of the corresponding algorithms in the sense that, loosely speaking, algorithms that are only guaranteed to converge to a co-stationary point are less likely to produce the optimal solution of the problem than algorithms that are guaranteed to converge to a CW-maximal point. In Section 4 we will show that the numerical results support this assertion.

2.1 Technical Preliminaries

We will begin by providing some auxiliary technical results that will be used in order to establish the main result. Lemma 2.1 is a trivial result that follows directly from the Cauchy-Schwarz inequality (see also Lemma 4.1 in [13]).

Lemma 2.1. *Suppose that* $\mathbf{0} \neq \mathbf{q} \in \mathbb{R}^d$ *and* $\rho > 0$. *Then the optimal solution of the optimization problem*

$$\text{(QCLP)} \qquad \begin{array}{ll} \max_{\mathbf{x}\in\mathbb{R}^d} & \mathbf{q}^T\mathbf{x} \\ s.t. & \|\mathbf{x}\|_2 \leq \rho, \end{array}$$

is given by $\mathbf{x}^* = \rho\frac{\mathbf{q}}{\|\mathbf{q}\|_2}$ *with the optimal value of* $\rho\|\mathbf{q}\|_2$.

The following simple lemma is an extension of Proposition 4.3 from [13].

Lemma 2.2. *Assume that* $\mathbf{0} \neq \mathbf{p} \in \mathbb{R}^n$. *Then the set of optimal solutions of the optimization problem*

$$\text{(S-QCLP)} \qquad \begin{array}{ll} \max_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{p}^T\mathbf{x} \\ s.t. & \|\mathbf{x}\|_0 \leq s, \\ & \|\mathbf{x}\|_2 \leq 1, \end{array}$$

is given by

$$\mathbf{X}^*(\mathbf{p}, s) = \left\{ \frac{\mathbf{x}}{\|\mathbf{x}\|_2} : \mathbf{x} \in H_s(\mathbf{p}) \right\},$$

with the optimal value of $\|\mathbf{p}_T\|_2$ *where* $T \in R_s(\mathbf{p})$.

Proof. We can write (S-QCLP) as

$$\max_{\substack{T \subseteq \{1,\ldots,n\} \\ |T| \leq s}} \max_{\mathbf{x} \in \mathbb{R}^n} \left\{ \mathbf{p}^T \mathbf{x} : \|\mathbf{x}\|_2 \leq 1, I_1(\mathbf{x}) \subseteq T \right\}. \tag{1}$$

According to Lemma 2.1, for each $T \subseteq \{1,\ldots,n\}$ satisfying $|T| \leq s$, the optimal value of the inner optimization problem is $\|\mathbf{p}_T\|_2$, and if $\mathbf{p}_T \neq \mathbf{0}$, then a solution $\mathbf{x}^*$ to the inner optimization problem is given by

$$\mathbf{x}^*_T = \frac{\mathbf{p}_T}{\|\mathbf{p}_T\|_2}, \ \mathbf{x}^*_{\bar{T}} = \mathbf{0}. \tag{2}$$

The problem (1) thus reduces to

$$\max_{\substack{T \subseteq \{1,\ldots,n\} \\ |T| \leq s}} \|\mathbf{p}_T\|_2 \tag{3}$$

Obviously, when $\|\mathbf{p}\|_0 \geq s$, the optimal solutions of the latter problem are all the sets containing the indices of components corresponding to the s largest absolute values in $\mathbf{p}$, and when $\|\mathbf{p}\|_0 < s$, the unique optimal solution is $I_1(\mathbf{p})$. Thus, the set of all optimal solutions of (3) is $R_s(\mathbf{p})$. Noting that $\mathbf{p}_T \neq \mathbf{0}$ for any $T \in R_s(\mathbf{p})$, we conclude that the optimal solutions of (S-QCLP) are given by (2) with T being any set in $R_s(\mathbf{p})$, which are exactly the members of $X^*(\mathbf{p}, s)$. □

Our final technical lemma states if a given vector $\tilde{\mathbf{x}}$ is *not* an optimal solution of the problem of maximizing a linear function over the unit norm, then there must be two indices $i \neq j$ for which the subvector $\tilde{\mathbf{x}}^*_{\{i,j\}}$ is also *not* an optimal solution for the problem restricted to the the variables x_i, x_j (while fixing all the other variables). This lemma is rather simple, but will play a key role in the proof of the main result.

Lemma 2.3. *Let $\mathbf{q} \in \mathbb{R}^d$ and $\rho > 0$. Suppose that $\tilde{\mathbf{x}}$ satisfies $\|\tilde{\mathbf{x}}\|_2 \leq \rho$, and that it is <u>not</u> an optimal solution to* (QCLP). *Then there exist indices $i, j (i \neq j)$ such that $\tilde{\mathbf{x}}_{\{i,j\}}$ is <u>not</u> the optimal solution of*

$$\text{(2-QCLP}_{\{i,j\}}\text{)} \qquad \begin{array}{ll} \max_{\mathbf{x}_{\{i,j\}} \in \mathbb{R}^2} & \mathbf{q}^T_{\{i,j\}} \mathbf{x}_{\{i,j\}} \\ \text{s.t.} & \|\mathbf{x}_{\{i,j\}}\|_2 \leq \left(\rho^2 - \sum_{l \neq i,j} \tilde{x}_l^2\right)^{1/2} \end{array}$$

Proof. Since $\tilde{\mathbf{x}}$ is not the optimal solution to (QCLP), we obtain that $\mathbf{q} \neq \mathbf{0}$ (since otherwise, if $\mathbf{q} = \mathbf{0}$, all feasible points are also optimal). Thus, the set $I_1(\mathbf{q})$ is nonempty. We will split the analysis into two cases.

- If $\|\tilde{\mathbf{x}}\|_2 < \rho$, then take any $i \in I_1(\mathbf{q})$ and $j \neq i$, and we can write

$$\|\tilde{\mathbf{x}}_{\{i,j\}}\|_2 < \left(\rho^2 - \sum_{l \neq i,j} \tilde{x}_l^2\right)^{1/2},$$

which together with $\mathbf{q}_{\{i,j\}} \neq \mathbf{0}$ (since $i \in I_1(\mathbf{q})$) implies that $\tilde{\mathbf{x}}_{\{i,j\}}$ is not the optimal solution of (2-QCLP$_{\{i,j\}}$), since we have by Lemma 2.1, that the constraint at the optimal solution must be active.

- If, on the other hand, $\|\tilde{\mathbf{x}}\|_2 = \rho$, then assume in contradiction that for each $i \neq j$ the vector $\tilde{\mathbf{x}}_{\{i,j\}}$ is the optimal solution of (2-QCLP$_{\{i,j\}}$). Take some $i \in I_1(\mathbf{p})$. For any $j \in I_0(\mathbf{p})$, we know that $\tilde{\mathbf{x}}_{\{i,j\}}$ is the optimal solution of (2-QCLP$_{\{i,j\}}$) and thus, according to Lemma 2.1 (employed on the problem (2-QCLP$_{\{i,j\}}$)), it must in particular satisfy $\tilde{x}_j = 0$, that is, $j \in I_0(\mathbf{x})$. To summarize,

$$\tilde{x}_j = 0 \text{ for any } j \in I_0(\mathbf{p}). \tag{4}$$

Now, for any $j \in I_1(\mathbf{p})$, according to Lemma 2.1, $\tilde{\mathbf{x}}_{\{i,j\}}$ must satisfy

$$\tilde{x}_i = \frac{p_i}{\|(p_i, p_j)^T\|_2}(\tilde{x}_i^2 + \tilde{x}_j^2)^{1/2}, \tag{5}$$

where here we used the fact that $\rho^2 - \sum_{l \neq i,j} \tilde{x}_l^2 = \tilde{x}_i^2 + \tilde{x}_j^2$. Squaring both sides of (5), we obtain that it is equivalent to $p_j^2 \tilde{x}_i^2 = p_i^2 \tilde{x}_j^2$, and hence

$$\tilde{x}_j^2 = \frac{p_j^2}{p_i^2}\tilde{x}_i^2 \qquad \text{for any } j \in I_1(\mathbf{p}).$$

By (4), $\tilde{x}_j = 0$ whenever $j \in I_0(\mathbf{p})$, and we can therefore write

$$\tilde{x}_j^2 = \frac{p_j^2}{p_i^2}\tilde{x}_i^2, \qquad j = 1, 2, \ldots, n.$$

Summing over $j = 1, 2, \ldots, n$, and using the fact that $\|\tilde{\mathbf{x}}\|_2^2 = \rho^2$, it follows that

$$\sum_{j=1}^{n} \tilde{x}_i^2 \frac{p_j^2}{p_i^2} = \rho^2,$$

implying that

$$\tilde{x}_i^2 = \rho^2 \frac{p_i^2}{\|\mathbf{p}\|_2^2},$$

which combined with the fact that that $\text{sgn}(\tilde{x}_i) = \text{sgn}(\tilde{p}_i)$ (see (5)), yields

$$\tilde{x}_i = \rho \frac{p_i}{\|\mathbf{p}\|_2}.$$

Since we actually proved the latter for an arbitrary $i \in I_1(\mathbf{p})$, and since $\tilde{x}_i = 0$ for any $i \in I_0(\mathbf{p})$ (see (4)), it follows that

$$\mathbf{x} = \rho \frac{\mathbf{p}}{\|\mathbf{p}\|_2},$$

in contradiction to the assumption that $\tilde{\mathbf{x}}$ is not an optimal solution of (QCLP).

□

The following corollary is a direct consequence of Lemmas 2.2 and 2.3.

Corollary 2.1. *Let $\tilde{\mathbf{x}} \in S$. If $\tilde{\mathbf{x}}$ is not an optimal solution to (S-QCLP) and $I_1(\tilde{\mathbf{x}}) \subseteq T$ for some $T \in R_s(\mathbf{p})$, then there exist indices $i, j \in T (i \neq j)$ such that $\tilde{\mathbf{x}}_{\{i,j\}}$ is not an optimal solution of (2-QCLP$_{\{i,j\}}$).*

Proof. Assume that $|T| = k$. Since $\tilde{\mathbf{x}}$ is not an optimal solution of (S-QCLP), it follows by Lemma 2.2 that $\tilde{\mathbf{x}}_T \neq \frac{\mathbf{p}_T}{\|\mathbf{p}_T\|}$, which implies that $\tilde{\mathbf{x}}_T$ is not the optimal solution of the restricted problem

$$\min_{\mathbf{y} \in \mathbb{R}^k} \left\{ \mathbf{p}_T^T \mathbf{y} : \|\mathbf{y}\|_2 \leq \rho \right\}.$$

Therefore, invoking Lemma 2.3 with $d = k, \mathbf{q} = \mathbf{p}_T$, it follows that there exist indices $i, j \in T (i \neq j)$ such that $\tilde{\mathbf{x}}_{i,j}$ is not an optimal solution of (2-QCLP$_{\{i,j\}}$). □

2.2 Co-Stationarity vs. CW-Maximality

The main result of this paper is given in the following theorem, which establishes the superiority of the CW-maximality condition over the co-stationarity condition.

Theorem 2.1. *Let $\mathbf{x}$ be a CW-maximum point of problem (SPCA). Then $\mathbf{x}$ is a co-stationary point of (SPCA).*

Proof. Let $\mathbf{x}$ be a CW-maximum point of (SPCA). Assume by contradiction that $\mathbf{x}$ is not a co-stationary point. This means that there exists a vector $\mathbf{v} \in S$ such that

$$\nabla f(\mathbf{x})^T (\mathbf{v} - \mathbf{x}) > 0. \tag{6}$$

We will show that we can find a vector $\mathbf{z} \in S_2(\mathbf{x})$ such that

$$\nabla f(\mathbf{x})^T(\mathbf{z}-\mathbf{x}) > 0. \tag{7}$$

This will imply a contradiction to the CW-maximality of $\mathbf{x}$ by the following simple argument: since f is a convex function, we have

$$f(\mathbf{z}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{z}-\mathbf{x}),$$

which combined with (7) implies that

$$f(\mathbf{z}) > f(\mathbf{x}),$$

which is an obvious contradiction to the CW-maximality of $\mathbf{x}$.

Since $\mathbf{x}$ satisfies (6), we obviously have $\nabla f(\mathbf{x}) \neq \mathbf{0}$. Let $X^*(\nabla f(\mathbf{x}), s)$ be the set of optimal solutions of (S-QCLP) with $\mathbf{p} = \nabla f(\mathbf{x})$ and let $\mathbf{x}^* \in X^*(\nabla f(\mathbf{x}), s)$ be some particular solution. Then

$$\nabla f(\mathbf{x})^T\mathbf{x}^* \geq \nabla f(\mathbf{x})^T\mathbf{v} > \nabla f(\mathbf{x})^T\mathbf{x},$$

and thus $\mathbf{x} \notin X^*(\nabla f(\mathbf{x}), s)$.
Suppose there exists some l for which $\nabla_l f(\mathbf{x}) \cdot x_l < 0$ (and in particular $l \in I_1(\mathbf{x})$). Define $\mathbf{z}$ as:

$$j = 1, \ldots, n \qquad z_j = \begin{cases} -x_l & j = l, \\ x_j & otherwise. \end{cases}$$

$\mathbf{z} \in S_2(\mathbf{x})$ and $\nabla f(\mathbf{x})^T(\mathbf{z}-\mathbf{x}) > 0$ since

$$\nabla f(\mathbf{x})^T(\mathbf{z}-\mathbf{x}) = -2 \cdot \nabla_l f(\mathbf{x}) \cdot x_l > 0.$$

We have thus shown in this case the desired contradiction. From now on , we will therefore consider the case where $\nabla_i f(\mathbf{x}) \cdot x_i \geq 0$ for all $i = 1, \ldots, n$.
Consider the following cases:

1. $I_1(\mathbf{x}) \not\subseteq I_{\geq}(\nabla f(\mathbf{x}), s)$.
   Obviously, there is some $h \in I_1(\mathbf{x}) \cap I_{<}(\nabla f(\mathbf{x}), s)$.
   We will consider the following subcases:

1.1. If $|I_{\geq}(\nabla f(\mathbf{x}), s)| < s$, then $\nabla_h f(\mathbf{x}) = 0$ (by Lemma 1.1, part 2), and since $\nabla f(\mathbf{x}) \neq \mathbf{0}$, we conclude, using Lemma 1.1 (part 1), that there is some $l \in I_{\geq}(\nabla f(\mathbf{x}), s)$. Define $\mathbf{z}$ as:

$$j = 1, \ldots, n \qquad z_j = \begin{cases} \operatorname{sgn}\left(\nabla_l f(\mathbf{x})\right) \cdot (x_h^2 + x_l^2)^{1/2} & j = l, \\ 0 & j = h, \\ x_j & otherwise. \end{cases}$$

Obviously $\mathbf{z} \in S_2(\mathbf{x})$, and in addition $\nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) > 0$ since

$$\begin{aligned} \nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) &= \nabla_l f(\mathbf{x}) \cdot \operatorname{sgn}\left(\nabla_l f(\mathbf{x})\right) \cdot (x_h^2 + x_l^2)^{1/2} \\ &\quad -\nabla_l f(\mathbf{x}) \cdot x_l \\ &= \left|\nabla_l f(\mathbf{x})\right| \cdot (x_h^2 + x_l^2)^{1/2} - \nabla_l f(\mathbf{x}) \cdot x_l \\ &= \left|\nabla_l f(\mathbf{x})\right| \cdot (x_h^2 + x_l^2)^{1/2} - \left|\nabla_l f(\mathbf{x})\right| \cdot |x_l| \qquad (\nabla_l f(\mathbf{x}) \cdot x_l \geq 0) \\ &= \left|\nabla_l f(\mathbf{x})\right| \cdot \left((x_h^2 + x_l^2)^{1/2} - |x_l|\right) > 0 \qquad (\nabla_l f(\mathbf{x}) \neq 0, x_h \neq 0)\,. \end{aligned}$$

1.2. If $|I_{\geq}(\nabla f(\mathbf{x}), s)| \geq s$, then there is some $l \in I_{\geq}(\nabla f(\mathbf{x}), s)$ such that $l \notin I_1(\mathbf{x})$. Otherwise $I_{\geq}(\nabla f(\mathbf{x}), s) \subseteq I_1(\mathbf{x})$, and since $|I_{\geq}(\nabla f(\mathbf{x}), s)| \geq s$ and $|I_1(\mathbf{x})| \leq s$, we have that $I_{\geq}(\nabla f(\mathbf{x}), s) = I_1(\mathbf{x})$, contradicting our assumption that $I_1(\mathbf{x}) \not\subseteq I_{\geq}(\nabla f(\mathbf{x}), s)$.
We will define $\mathbf{z}$ as:

$$j = 1, \ldots, n \qquad z_j = \begin{cases} \operatorname{sgn}\left(\nabla_l f(\mathbf{x})\right) \cdot |x_h| & j = l, \\ 0 & j = h, \\ x_j & otherwise. \end{cases}$$

Clearly, $\mathbf{z} \in S_2(\mathbf{x})$. In addition, $\nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) > 0$ since:

$$\begin{aligned} \nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) &= \nabla_l f(\mathbf{x}) \cdot \operatorname{sgn}\left(\nabla_l f(\mathbf{x})\right) \cdot |x_h| \\ &\quad -\nabla_h f(\mathbf{x}) \cdot x_h \\ &= \left|\nabla_l f(\mathbf{x})\right| \cdot |x_h| - \left|\nabla_h f(\mathbf{x})\right| \cdot |x_h| \qquad (\nabla_h f(\mathbf{x}) \cdot x_h \geq 0) \\ &= \left(\left|\nabla_l f(\mathbf{x})\right| - \left|\nabla_h f(\mathbf{x})\right|\right) \cdot |x_h| > 0, \end{aligned}$$

where the last inequality holds since $x_h \neq 0$ and the indices l and h are such that $l \in I_{\geq}(\nabla f(\mathbf{x}), s)$ and $h \in I_{<}(\nabla f(\mathbf{x}), s)$, thus according to Lemma 1.1 (part 3) $\left|\nabla_l f(\mathbf{x})\right| > \left|\nabla_h f(\mathbf{x})\right|$.

2. $I_1(\mathbf{x}) \subseteq I_{\geq}(\nabla f(\mathbf{x}), s)$
Now we will consider the following subcases:

2.1. If $I_1(\mathbf{x}) \subseteq T$ for some $T \in R_s(\nabla f(\mathbf{x}))$, then since $\mathbf{x} \notin X^*(\nabla f(\mathbf{x}), s)$, it follows that according to Corollary 2.1, there exist indices $h, l \in T$ such that

$$\hat{\mathbf{x}} = \underset{\mathbf{y}\in\mathbb{R}^2}{\operatorname{argmax}} \left\{ \nabla_{\{h,l\}} f(\mathbf{x})^T \mathbf{y} : \|\mathbf{y}\|^2 \leq 1 - \sum_{i\neq h,l} x_i^2 \right\}$$

satisfies

$$\nabla_{\{h,l\}} f(\mathbf{x})^T \hat{\mathbf{x}} > \nabla_{\{h,l\}} f(\mathbf{x})^T \mathbf{x}_{\{h,l\}}. \tag{8}$$

Since $|T| \leq s$ and $\|\hat{\mathbf{x}}\|_2^2 \leq 1 - \sum_{i\neq h,l} x_i^2$, the vector

$$j = 1, \ldots, n \qquad z_j = \begin{cases} \hat{x}_1 & j = h, \\ \hat{x}_2 & j = l, \\ x_j & otherwise, \end{cases}$$

is in $S_2(\mathbf{x})$, and satisfies by (8) that $\nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) > 0$.

2.2. If $I_1(\mathbf{x}) \not\subseteq T$ for all $T \in R_s(\nabla f(\mathbf{x}))$, then:

- Take $h \in I_1(\mathbf{x})$ such that $h \notin T$ for some $T \in R_s(\nabla f(\mathbf{x}))$. Since $I_1(\mathbf{x}) \subseteq I_{\geq}(\nabla f(\mathbf{x}), s)$, it follows that $h \in I_{\geq}(\nabla f(\mathbf{x}), s)$. Also, since $I_{>}(\nabla f(\mathbf{x}), s) \subseteq T$ and $h \notin T$, we have that $h \notin I_{>}(\nabla f(\mathbf{x}), s)$, implying that $h \in I_{=}(\nabla f(\mathbf{x}), s)$. Thus, $h \in I_{=}(\nabla f(\mathbf{x}), s) \cap I_1(\mathbf{x})$.
- $I_{>}(\nabla f(\mathbf{x}), s) \not\subseteq I_1(\mathbf{x})$. To show this, note that otherwise, $I_{>}(\nabla f(\mathbf{x}), s) \subseteq I_1(\mathbf{x})$, and since $I_1(\mathbf{x}) \subseteq I_{\geq}(\nabla f(\mathbf{x}), s)$ and $|I_1(\mathbf{x})| \leq s$, we obtain that $|I_1(\mathbf{x})| \leq \min\{s, |I_{\geq}(\nabla f(\mathbf{x}), s)|\}$, implying that $I_1(\mathbf{x}) \subseteq T$ for some $T \in R_s(\nabla f(\mathbf{x}))$ in contradiction to our assumption. Thus, there exists some $l \in I_{>}(\nabla f(\mathbf{x}), s)$ such that $l \notin I_1(\mathbf{x})$.

Define $\mathbf{z}$ as:

$$j = 1, \ldots, n \qquad z_j = \begin{cases} \operatorname{sgn}(\nabla_l f(\mathbf{x})) \cdot |x_h| & j = l, \\ 0 & j = h, \\ x_j & otherwise. \end{cases}$$

Clearly, $\mathbf{z} \in S_2(\mathbf{x})$. Also, $\nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) > 0$ since

$$\begin{aligned} \nabla f(\mathbf{x})^T(\mathbf{z} - \mathbf{x}) &= \nabla_l f(\mathbf{x}) \cdot \operatorname{sgn}(\nabla_l f(\mathbf{x})) \cdot |x_h| \\ &\quad -\nabla_h f(\mathbf{x}) \cdot x_h \\ &= \left|\nabla_l f(\mathbf{x})\right| \cdot |x_h| - \left|\nabla_h f(\mathbf{x})\right| \cdot |x_h| \qquad (\nabla_h f(\mathbf{x}) \cdot x_h \geq 0) \\ &= \left(\left|\nabla_l f(\mathbf{x})\right| - \left|\nabla_h f(\mathbf{x})\right|\right) \cdot |x_h| > 0. \end{aligned}$$

where the last inequality holds since $x_h \neq 0$ and the indices l and h are such that $l \in I_{>}(\nabla f(\mathbf{x}), s)$ and $h \in I_{=}(\nabla f(\mathbf{x}), s)$, and thus according to Lemma 1.1 (part 3) $\left|\nabla_l f(\mathbf{x})\right| > \left|\nabla_h f(\mathbf{x})\right|$.

We have thus arrived at a contradiction, and the desired implication is established. □

In order to show that the reverse implication is not valid, that is, that co-stationary points are not necessarily CW-maximal points, we present an example of a problem instance and a co-stationary point that is not a CW-maximal point.

Example 2.1. For any $n > s > 0$, we consider problem (SPCA) with a diagonal matrix $\mathbf{A}$ whose entries on the main diagonal are given by the vector $\mathbf{a}$ defined by

$$\mathbf{a} = \begin{pmatrix} 2 \cdot \mathbf{1}_{n-s} \\ 0.5 \cdot \mathbf{1}_s \end{pmatrix},$$

where for a given positive integer m, $\mathbf{1}_m$ and $\mathbf{0}_m$ are the vectors of size m with all entries equals to ones or zeros respectively. We also define

$$\mathbf{x} = \begin{pmatrix} \mathbf{0}_{n-s} \\ s^{-0.5} \cdot \mathbf{1}_s \end{pmatrix} \qquad \text{and} \qquad \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{0}_{n-s-1} \\ s^{-0.5} \\ 0 \\ s^{-0.5} \cdot \mathbf{1}_{s-1} \end{pmatrix}.$$

It easy to see that $\mathbf{x}, \tilde{\mathbf{x}} \in S$ and that $\mathbf{A} \succ \mathbf{0}$, since it is a diagonal matrix with positive diagonal elements. The gradient of f is given by:

$$\nabla f(\mathbf{x}) = 2\mathbf{A}\mathbf{x} = \begin{pmatrix} \mathbf{0}_{n-s} \\ s^{-0.5} \end{pmatrix}.$$

For any $\mathbf{v} \in S$:

$$\langle \nabla f(\mathbf{x}), \mathbf{v} - \mathbf{x} \rangle = \sum_{i=n-s+1}^{n} s^{-0.5}(v_i - s^{-0.5})$$

$$= s^{-0.5} \left(\sum_{i=n-s+1}^{n} v_i - s^{0.5} \right) \leq s^{-0.5} \left(\|\mathbf{v}\|_1 - s^{0.5} \right) \leq 0,$$

where the last inequality holds since $\|\mathbf{v}\|_1 \leq \sqrt{\|\mathbf{v}\|_0}\|\mathbf{v}\|_2 \leq \sqrt{s}$. Hence, $\mathbf{x}$ is co-stationary. The vector $\tilde{\mathbf{x}}$ satisfies $\tilde{\mathbf{x}} \in S_2(\mathbf{x})$ and since:

$$f(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^T \mathbf{A} \tilde{\mathbf{x}} = (s-1) \cdot (2s)^{-1} + 2s^{-1} = (s+3) \cdot (2s)^{-1} > s \cdot (2s)^{-1} = \mathbf{x}^T \mathbf{A} \mathbf{x} = f(\mathbf{x}),$$

it follows that $\mathbf{x}$ is not a CW-maximum point.

2.3 Support Optimality

Theorem 2.1 establishes the relationship between the two stationarity conditions considered up to this point: co-stationarity and CW-maximality. A third condition, proposed in [16], that we will refer to as *support optimality* (SO) is given in the following definition.

Definition 2.1 (Support Optimality). *A vector $\mathbf{x}^* \in S$ is called a* **support optimal (SO) point** *of (P) if it is an optimal solution of the optimization problem*

$$\text{(SO)} \qquad \begin{array}{ll} \max_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) \\ s.t. & \|\mathbf{x}\|_2 \leq 1, \\ & I_1(\mathbf{x}) \subseteq T, \end{array}$$

for some $T \subseteq \{1, 2, \ldots, n\}$ such that $|T| = s$. A vector $\mathbf{x}^*$ is called **generalized support optimal (GSO) point** *if it satisfies the above condition, but with the relaxed assumption $1 \leq |T| \leq s$ instead of $|T| = s$*

This condition is remarkably weak and cannot be used exclusively to derive a reasonable algorithm. Nevertheless, this condition is not totally futile. In Section 4 we will adopt the variational normalization strategy suggested in [16], stating that for each sparse solution obtained by any technique, it is reasonable to replace this solution with the SO point that correspond to the same support. This modification does not debilitate the relationships just established.

We will conclude this section with an example that demonstrates the potential benefit of employing algorithms that produce a point that satisfies stronger necessary optimality conditions. Consider the pit-prop data, which consists of 13 variables measuring various physical properties of 180 pitprops. This data set was suggested originally in [8], and since then was extensively used as a benchmark example for sparse PCA, see for example [10, 11, 16]. We consider a sparsity level $s = 4$, and combined with the fact that there are 13 variables, we obtain $\binom{13}{4} = 715$ SO points. Out of this set of points, 28 satisfy the co-stationarity condition and only 2 satisfy the CW-maximality condition. The following table presents the support sets of each of the co-stationarity points along with their function values.

Table 2: The supports of the co-stationary points for the pit prop data.

#	Support	CW-maximum	Value	#	Support	CW-maximum	Value
1	{1,2,9,10}	*	2.937	15	{5,6,7,10}		2.337
2	{1,2,7,10}		2.883	16	{7,8,10,12}		2.314
3	{1,2,7,9}		2.859	17	{7,8,10,13}		2.302
4	{1,2,8,9}		2.797	18	{5,6,7,13}		2.28
5	{1,2,8,10}		2.759	19	{3,4,6,7}		2.209
6	{1,2,6,7}		2.697	20	{4,5,6,7}		2.196
7	{2,7,9,10}		2.696	21	{7,10,12,13}		2.136
8	{2,6,7,10}		2.592	22	{3,4,8,12}		1.995
9	{1,6,7,10}		2.587	23	{3,4,10,12}		1.992
10	{1,2,3,4}	*	2.563	24	{3,10,11,12}		1.609
11	{7,8,9,10}		2.549	25	{3,5,12,13}		1.516
12	{6,7,9,10}		2.522	26	{1,5,12,13}		1.414
13	{6,7,10,13}		2.459	27	{2,5,12,13}		1.408
14	{6,7,8,10}		2.444	28	{3,5,11,13}		1.382

Since the number of CW-maximum points is significantly smaller than the number of co-stationary points, it is much more probable that the optimal solution will be found by an algorithm that produces CW-maximum points than an algorithm that produces co-stationary points.

3 Algorithms

In this section we will present the conditional gradient and the CW-based algorithms that produce co-stationarity and CW-maximum points, respectively, for the sparse PCA problem. For simplicity of presentation, we will present the algorithms under the assumption that the covariance matrix is given. Nevertheless, each of the algorithms can be applied directly on the data matrix with an appropriate adjustment. In addition, although all the algorithms are described in the context of the sparse PCA problem, they can be easily converted to algorithms that solve the general problem (P).

3.1 Conditional Gradient

The conditional gradient algorithm with unit step-size (ConGradU) was suggested in [13] as a unified algorithmic framework for several algorithms, which were proposed for sparse PCA or one of its penalized or relaxed versions. Each of these algorithms can be reduced to a simple iterative scheme of the following form:

$$\mathbf{x}^{k+1} = \frac{S(\mathbf{A}\mathbf{x}^k)}{\|S(\mathbf{A}\mathbf{x}^k)\|_2},$$

where $S : \mathbb{R}^n \to \mathbb{R}^n$ is a simple operator given as an analytical formula or can be efficiently computed. It is stated in Theorem 3.4 in [13] that for a convex function over a nonempty compact set, the function values of any sequence generated by ConGradU will be monotonically increasing and that it will converge to a co-stationary point. We will not expand our discussion on most of the schemes that ConGradU encompasses, and will only consider the l_0-constrained formulation for which ConGradU is as follows.

ConGradU Algorithm for (SPCA)
Input: $\mathbf{A}$ - covariance/correlation matrix, s - sparsity level.
Output: $\mathbf{x}$ - a co-stationary point of (SPCA).
Initialization: Take $\mathbf{x}^0$.
General step: $(k = 0, 1, \dots)$

$$\begin{aligned} \mathbf{y}^k &\in H_s(\mathbf{A}\mathbf{x}^k) \\ \mathbf{x}^{k+1} &= \frac{\mathbf{y}^k}{\|\mathbf{y}^k\|_2}, \end{aligned}$$

If *stopping criteria*, STOP and set $\mathbf{x} \leftarrow \mathbf{x}^{k+1}$.

Notice that in order to find a vector in $H_s(\mathbf{z})$, all we need to do is to set to zero all the $n - s$ entries of $\mathbf{z}$ with the smallest absolute values, which can be done in $O(n)$ time. Thus, the total complexity per iteration is $O(s \cdot n)$ (taking into account the matrix-vector multiplication). An extensive numerical study on simulated and real life data was presented in [13]. It was empirically shown that all the examined versions of ConGradU provide similar performance and complexities. Thus, as a result of its simplicity and the lack of parameter tuning requirements, the l_0-constrained version just presented (in which $S = H_s$) is preferred over other variants in which S is chosen differently, for the case when the desired sparsity level is known.

3.2 Coordinate Wise Based Methods: GCW and PCW

In [2] several algorithms that produce a CW-minimum point were considered. These block coordinate descent type algorithms perform at each iteration an optimization step with respect to one or two variables, while keeping the rest fixed. The coordinates that need to be altered are chosen to be the ones that produce the maximal decrease among all possible alternatives or by applying an index selection strategy based on a local first order information. We adopt this approach and present similar algorithms for the sparse PCA problem. At each iteration of a CW-based algorithm applied to (P), at most two variables will be updated. We can categorize each of the iterations according to whether the support is altered or not. Block coordinate algorithms suffer from a major drawback - a slow convergence rate. In order to reduce the effect of this displeasing characteristic, we will replace the point obtained at each step with a GSO point that corresponds to the same support. This modification allows us to bypass the large amount of iterations that should have been devoted for optimizing the variables with respect to a fixed support.

Regarding the step when the support is altered, we consider two cases. For the case $\|\mathbf{x}^k\|_0 = s$, in order to assure that the sparsity constraint remains satisfied, any inclusion of some index $j \in I_0(\mathbf{x}^k)$ in the support is accompanied with an exclusion of some index $i \in I_1(\mathbf{x}^k)$. Combined with the fact that $\|\mathbf{x}^k\|_2 = 1$ for each $k > 0$ (since $\mathbf{x}^k$ is an SO point), we obtain that for such a pair of indices, if we set the value of the element that corresponds to i to be equal to zero, the optimal value of the element that corresponds to j will be equal in absolute value to x_i^k, and there are only two alternative solutions that should be examined for the two possible values of the sign. Hence, the comparison of each possible pair of indices $i \in I_1(\mathbf{x}^k)$ and $j \in I_0(\mathbf{x}^k)$ is straightforward and is given by a closed analytical formula.

For the case $\|\mathbf{x}^k\|_0 < s$, since f is a convex function we have for any $\mathbf{y}$

$$f(\mathbf{y}) - f(\mathbf{x}^k) \geq \nabla f(\mathbf{x}^k)^T(\mathbf{y} - \mathbf{x}^k) \tag{9}$$

and according to Lemma 2.2 if there is some $j \in I_0(\mathbf{x}^k)$ such that $|\nabla f_j(\mathbf{x}^k)| > 0$, then setting

$$\mathbf{y} = \underset{z}{\operatorname{argmax}} \left\{ \nabla f(\mathbf{x}^k)^T(\mathbf{z} - \mathbf{x}^k) : \mathbf{z} \in S \right\}$$

implies that $\mathbf{y} \neq \mathbf{x}^k$ and that the right-hand side of (9) is strictly positive. Thus, choosing the index to be included into the support to be $j_k = \operatorname{argmax}\{|\nabla_j f(\mathbf{x}^k)| : j \in I_0(\mathbf{x}^k)\}$ will guarantee increase in the objective function if $|\nabla_{j_k} f(\mathbf{x}^k)| > 0$. This strategy is pretty

elementary and clearly inferior to the greedy and the approximate greedy approaches suggested in [16] and [7]. Nevertheless, it is computationally cheaper and, as opposed to the greedy methods just mentioned, the algorithms that we consider do not terminate at the moment that a solution with a full support is obtained, thus choosing the best alternative in this situation is less significant. Another important empirical observation that we should keep in mind is that if the initial support satisfies $|T| = s$, then the condition $\|\mathbf{x}^k\|_0 < s$ will probably be false for all k in any reasonable practical scenario.
Below we present the Greedy CW PCA (GCW) algorithm.

The Greedy CW PCA (GCW) Algorithm
Input: $\mathbf{A}$ - covariance/correlation matrix, s - sparsity level.
Output: x - a CW-maximum point of (SPCA).
Initialization: Take $T \in \{1, 2, \ldots, n\}$ such that $1 \leq |T| \leq s$ and set $k = 0$.
General step:

1. Compute

   (10) $$\mathbf{x}_T^k \in \underset{\mathbf{x}\in\mathbb{R}^{|T|}}{\operatorname{argmax}} \left\{\mathbf{x}^T \mathbf{A}_T \mathbf{x} : \|\mathbf{x}\|_2 = 1\right\}, \mathbf{x}_{\bar{T}}^k = \mathbf{0}.$$

2. If $\|\mathbf{x}^k\|_0 < s$, then compute

   $$j_k \in \underset{j\in I_0(\mathbf{x}^k)}{\operatorname{argmax}} \left\{|\nabla_j f(\mathbf{x}^k)|\right\}.$$

   If $|\nabla_{j_k} f(\mathbf{x}^k)| > 0$, then set

   $$\begin{aligned} T &= I_1(\mathbf{x}^k) \cup \{j_k\} \\ k &= k + 1 \end{aligned}$$

   and return to 1.

3. For every $i \in I_1(\mathbf{x}^k)$ and $j \in I_0(\mathbf{x}^k)$ compute

   $$f_{i,j} = \max_{\sigma\in\{-1,1\}} \left\{f\left(\mathbf{x}^k - x_i^k \mathbf{e}_i + \sigma|x_i^k|\mathbf{e}_j\right)\right\}.$$

   Let $(i_k, j_k) = \operatorname{argmax}\left\{f_{i,j} : i \in I_1(\mathbf{x}^k), j \in I_0(\mathbf{x}^k)\right\}$. If $f_{i_k,j_k} > f(\mathbf{x}^k)$, then set

   $$\begin{aligned} T &= (T \setminus \{i_k\}) \cup \{j_k\} \\ k &= k + 1 \end{aligned}$$

and return to 1.

Otherwise, STOP and set $\mathbf{x} \leftarrow \mathbf{x}^{k+1}$.

Note that the the sequence generated by the GCW algorithm is comprised of GSO points. We can assume that the optimal solution that we pick in the solution of problem (10) is uniquely defined by T. Under this setting, there is a finite number of GSO points that we consider. Adding to this the fact that the GCW algorithm generates a strictly increasing sequence of function values, we conclude that it is terminated in a finite number of iterations. In addition, by the way the algorithm is defined, when its output is a point with a full support, then it is necessarily a CW-maximum point.
The following table summarizes the steps that require the bulk of the computational effort.

Table 3: Computational complexity of GCW.

Step	Complexity
Solving the maximum eigenvalue problem	$O(s^3)$
Computing the gradient	$O(s \cdot n)$
Computing $f_{i,j}$ for each possible swap	$O\left(s \cdot (n-s)\right)$

Thus, the theoretical complexity of the algorithm, assuming that $s \ll n$, is $O(s \cdot n)$. Practically, most of the computation time is consumed in computing $f_{i,j}$ for each possible swap. Even though the complexity of this step is of the same order as the computation of the gradient, the actual number of operations is notably higher. This observation encourages us to consider the following variation of GCW, which we name *the Partial CW PCA (PCW)* algorithm.

The Partial CW PCA (PCW) Algorithm
Input: $\mathbf{A}$ - covariance/correlation matrix, s - sparsity level.
Output: $\mathbf{x}$ - a CW-maximum point of (SPCA).
Initialization: Take $T \in \{1, 2, \ldots, n\}$ such that $1 \leq |T| \leq s$ and set $k = 0$.
General step:

1. Compute

$$\mathbf{x}_T^k = \underset{\mathbf{x}\in\mathbb{R}^{|T|}}{\operatorname{argmax}} \left\{ \mathbf{x}^T \mathbf{A}_T \mathbf{x} : \|\mathbf{x}\|_2 = 1 \right\}, \mathbf{x}_{\bar{T}}^k = \mathbf{0}.$$

2. If $\|\mathbf{x}^k\|_0 < s$, then compute

$$j_k \in \underset{j\in I_0(\mathbf{x}^k)}{\operatorname{argmax}} \left\{ |\nabla_j f(\mathbf{x}^k)| \right\}.$$

   If $|\nabla_{j_k} f(\mathbf{x}^k)| > 0$, then set

$$\begin{aligned} T &= I_1(\mathbf{x}^k) \cup \{j_k\} \\ k &= k+1 \end{aligned}$$

   and return to 1.

3. Set $R = I_1(\mathbf{x}^k)$.
   While $|R| > 0$

   Set $i_k \in \operatorname{argmin} \left\{ |x_i^k| : i \in R \right\}$ and for each $j \in I_0(\mathbf{x}^k)$ compute

$$f_{i_k,j} = \max_{\sigma\in\{-1,1\}} \left\{ f\left(\mathbf{x}^k - x_{i_k}^k \mathbf{e}_{i_k} + \sigma |x_{i_k}^k| \mathbf{e}_j\right) \right\}.$$

   Let $j_k \in \operatorname{argmax} \left\{ f_{i_k,j} : j \in I_0(\mathbf{x}^k) \right\}$.
   If $f_{i_k,j_k} > f(\mathbf{x}^k)$, then set

$$\begin{aligned} T &= (T \setminus \{i_k\}) \cup \{j_k\} \\ k &= k+1 \end{aligned}$$

   and return to 1.

   Otherwise, set $R = R \setminus \{i_k\}$.

   STOP and set $\mathbf{x} \leftarrow \mathbf{x}^{k+1}$.

Before termination, PCW will perform the computation of all possible $f_{i,j}$,thus assuring the convergence to a CW-maximum point, given that the output is of a full support. For the general step, the amount of computation will significantly decrease on the expense

of finding the indices that provide the maximal increase in the function value. Nevertheless, the empirical study suggests that PCW provides similar results as GCW with respect to function values in a fraction of the time, as demonstrated in Section 4.

4 Numerical Results

We will illustrate the effectiveness of the algorithms proposed in the previous section on simulated and a gene expression datasets. We compared the results with the following alternative algorithms: the novel l_0-constrained version of ConGradU [13], the expectation maximization [19], approximate greedy [7] and thresholding [4]. The MATLAB implementation of ConGradU was kindly provided by the authors, for all the other alternative algorithms we used a MATLAB implementation available on the authors' web-pages. For the thresholding algorithm and the algorithms proposed in this paper, we used a MATLAB implementation which is available in the following URL:

http://tx.technion.ac.il/~yakovv/packages/CW_PCA.zip

Whenever an initialization is required, we set the initial point to be the solution of the thresholding method. For each of the algorithms, we extracted the sparsity pattern (the set of indices of the nonzero elements). The actual output vector is determined by solving problem (10), where T is the generated sparsity pattern. The experiments were conduced on a PC with a 3.40GHz processor with 16GB RAM.

4.1 Random Data

The covariance matrix $\mathbf{A}$ is given by $\mathbf{A} = \mathbf{D}^T\mathbf{D}$, where $\mathbf{D}$ is the so-called "data matrix". Each entry in the data matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$ was randomly generated according to the Gaussian distribution with zero mean and variance $1/m$ ($D_{i,j} \sim \mathcal{N}(0, 1/m)$). We considered data matrices with $n = 2000, 5000, 10,000$ and $50,000$ variables. The number of observations is set to $m = 150$ for all matrices. The sparsity levels considered are $s = 5, 10, \ldots, 250$, and for each sparsity level we generated 100 realizations. We will measure the effectiveness of the algorithms according to the average proportion of variability explained by the algorithm with respect to the largest eigenvalue of the data covariance matrix (i.e. $\mathbf{x}^T\mathbf{A}\mathbf{x}/\lambda_1(\mathbf{A})$ where $\mathbf{x}$ is the solution and $\lambda_1(\mathbf{A})$ is the largest eigenvalue of $\mathbf{A}$).

4.1.1 GCW vs. PCW

First, we would like to compare the effectiveness and performance of the CW-based algorithms proposed in the previous section: GCW and PCW. We conducted the comparison based on data matrices with $2,000$ variables and the results are given in Figure 1.

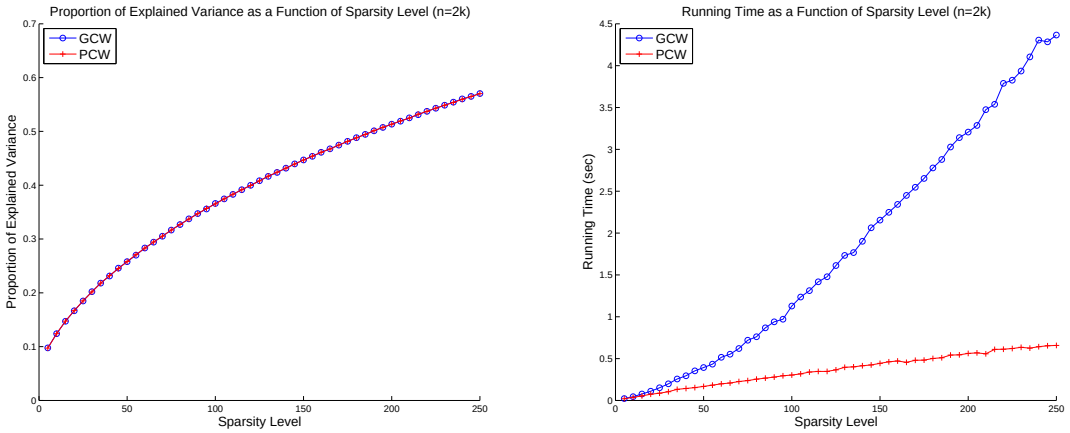

Figure 1: *GCW vs. PCW - The proportion of explained variability is given in the left figure and the computation time is given in the right one. The plot in both figures are given as a function of the sparsity level.*

We can clearly see that both methods achieve similar results with respect to the function values, while PCW achieves these results in a fraction of the time. Thus, in the remaining numerical study we will omit GCW. Although the partial version remarkably reduces the computation time, it is still not competitive for very large-scale problems when a full path of solutions is required. Thus, for such cases, we will also examine the effect of initializing PCW with the solution of the previous run (with the smaller sparsity level), and we will refer to such a continuation scheme as PCW$_{cont}$.

4.1.2 PCW vs. Alternative Methods

We will now compare the effectiveness and performance of PCW with respect to the alternative algorithms mentioned earlier. The setting for this set of experiments is the same as the one described in the previous example, but with problems with $n = 5,000,\ 10,000$ and $50,000$ variables. Figure 2 provides the proportion of explained variability as a function of the sparsity level.

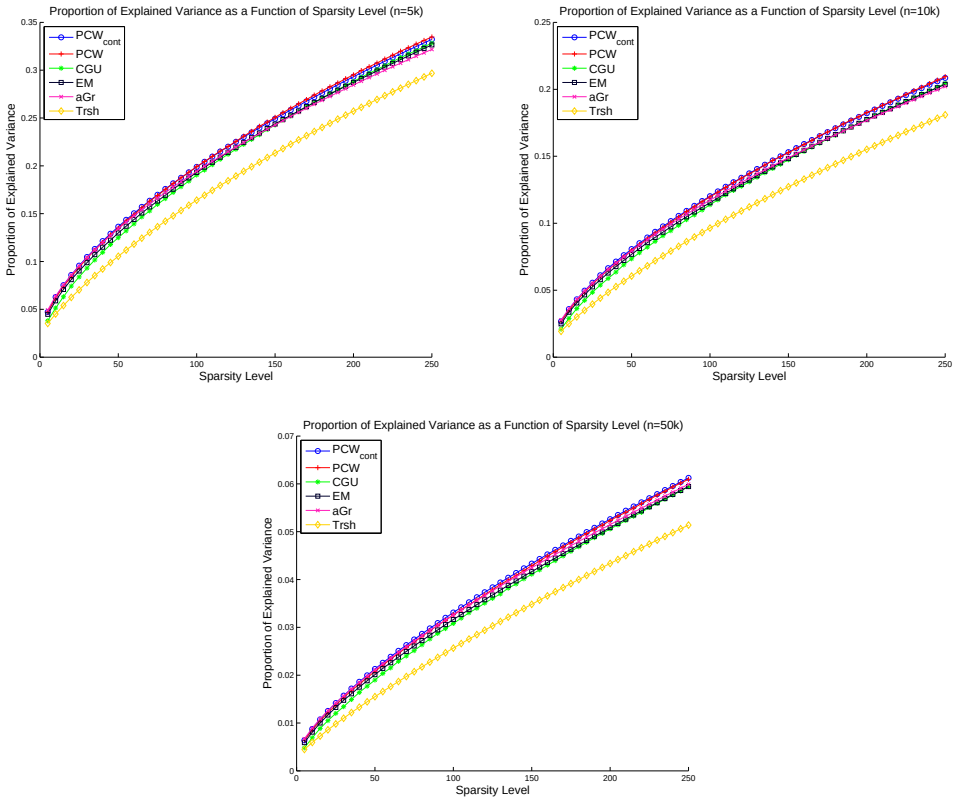

Figure 2: *PCW vs. Others - The proportion of explained variability as a function of the sparsity level for* $n = 5000, 10,000$ *and* $50,000$ *are given in the upper left, upper right and bottom figures, respectively.*

For small sparsity levels (< 50) most of the algorithms provide similar results, but as the sparsity level is increased, the CW algorithms becomes superior to all the other methods. This advantage is not achieved without a price. In Figure 3 we provide the cumulative computation time of the algorithms (the cumulative time is considered since the approximate greedy algorithm provides a full set of solutions).

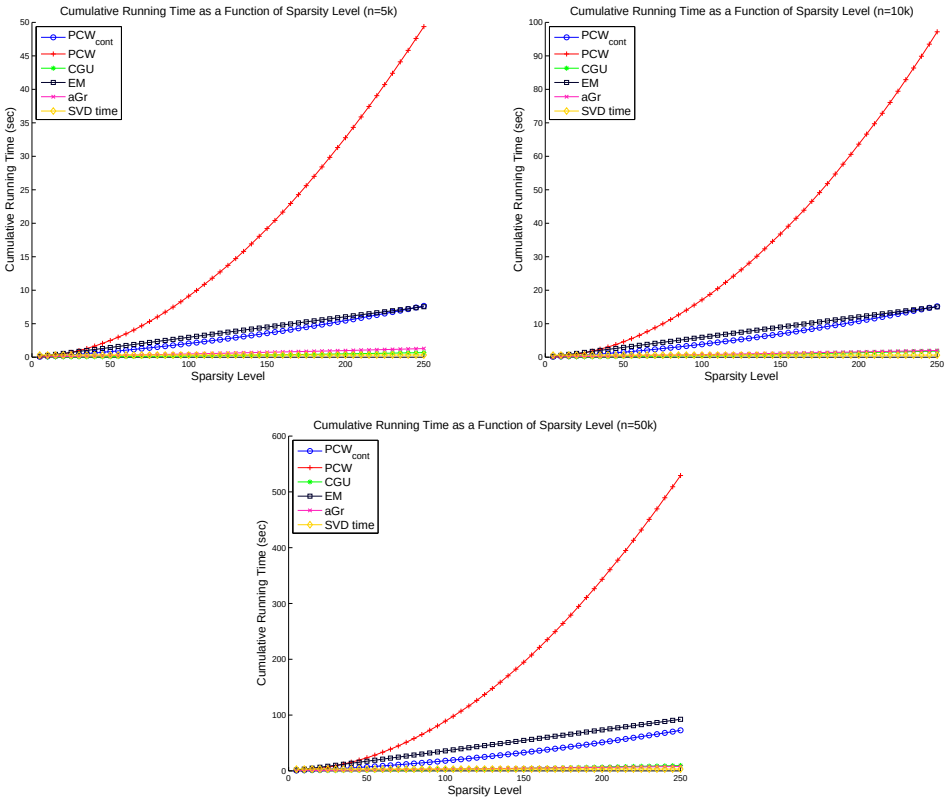

Figure 3: *PCW vs. Alternative Methods - The cumulative computation times as a function of the sparsity level for* $n = 5,000, 10,000$ *and* $50,000$ *are given in the upper left, upper right and bottom, respectively. The SVD time is the time required for computing the principal eigenvector of the covariance matrix that corresponds to the generated data, which is used in order to find the thresholding solution, and in order to initialize the CW and ConGradU algorithms.*

Even though PCW has greatly decreased the computation time with respect to GCW, it still requires a notably higher amount of computation time with respect to the alternative algorithms. The scheme we referred as PCW$_{cont}$ achieves similar results to PCW with respect to the function value. Regarding the running time, this scheme is competitive to the EM algorithm and requires somewhat more computational effort than the ConGradU and approximate greedy algorithms, thus providing a reasonable approach when a full set of solutions is required.

4.2 Gene Expression Dataset

Sparse PCA is extensively utilized in the identification of the genes that reflect the changes in the gene expression patterns during different biological states, thus contributing to the diagnosis and research of certain diseases such as cancer. Figure 4 illustrates the proportion of explained variability and the cumulative running time for a Leukemia data set [1]. This data set is composed from gene expression profiles of 72 patients with 12582 genes. The data set is normalized such that it has zero mean and unit variance.

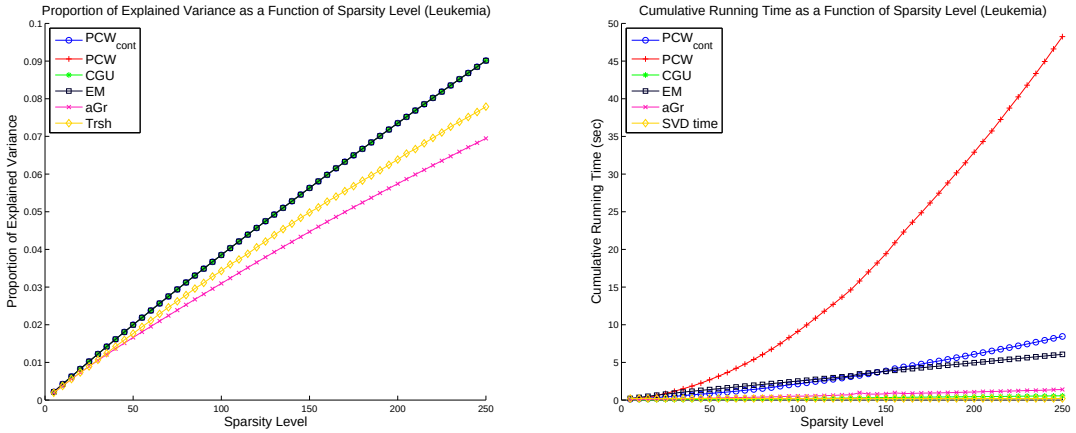

Figure 4: *Leukemia Gene Expression Data - The proportion of explained variability is given in the left figure and the cumulative computation time is given in the right one. The plot in both figures are given as a function of the sparsity level.*

Most of the algorithms under consideration provide similar results with respect to the explained variability, which might indicate that this problem is, in a sense, rather easy to solve. We conducted similar experiments for additional 20 gene expression data sets from the GeneChip oncology database [12] that is publicly available in:

`http://compbio.dfci.harvard.edu/compbio/tools/gcod`

While commonly, all the algorithms provided similar results, we can still see in Figure 5 that PCW yields the best solution (with respect to the function value) more times than the alternative algorithms, and consequently it obtains the smallest mean error with respect to the best solution.

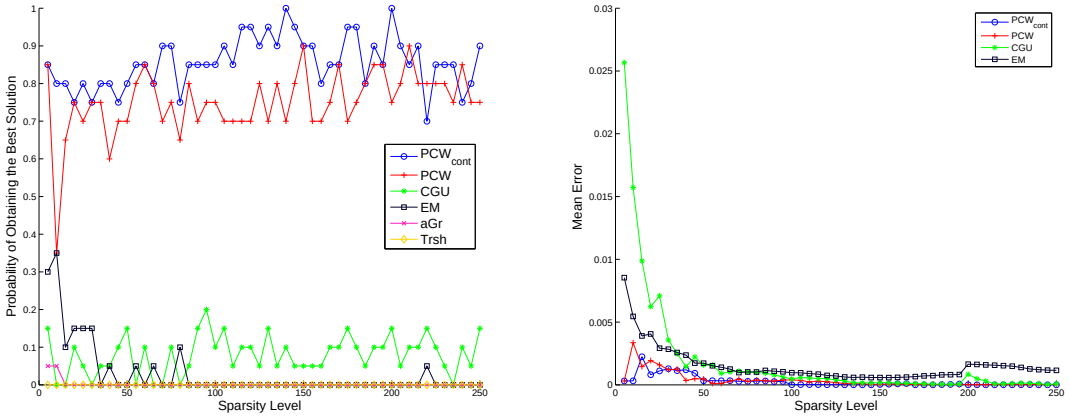

Figure 5: *Gene Expression Data - The left figure illustrates for each sparsity level the proportion of the number of data sets for which each algorithm obtained the best solution. The right figure illustrates for each sparsity level the mean error with respect to the best solution (the approximate greedy and thresholding algorithms were disregarded since both of them provide relative poor results).*

References

[1] Scott A. Armstrong, Jane E. Staunton, Lewis B. Silverman, Rob Pieters, Monique L. den Boer, Mark D. Minden, Stephen E. Sallan, Eric S. Lander, Todd R. Golub, and Stanley J. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature genetics*, 30(1):41–47, January 2002.

[2] Amir Beck and Yonina C. Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM J. Optim.*, 23(3):1480–1509, 2013.

[3] Amir Beck and Nadav Hallak. On the minimization over sparse symmetric sets. 2014. submitted for publication.

[4] Jorge Cadima and Ian T. Jolliffe. Loading and correlations in the interpretation of principle compenents. *Journal of Applied Statistics*, 22(2):203–214, 1995.

[5] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation of sparse PCA using semidefinite programming. *SIAM Review*, 49(3), 2007.

[6] Alexandre D'Aspremont. Identifying small mean-reverting portfolios. *Quantitative Finance*, 11(3):351–364, 2011.

[7] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *J. Mach. Learn. Res.*, 9:1269–1294, June 2008.

[8] J. N. R. Jeffers. Two case studies in the application of principal component analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 16(3):pp. 225–236, 1967.

[9] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

[10] Ian T. Jolliffe, Nickolay T. Trendafilov, and Mudassir Uddin. A Modified Principal Component Technique Based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, September 2003.

[11] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11:517–553, March 2010.

[12] Fenglong Liu, Joseph White, Corina Antonescu, Daniel Gusenleitner, and John Quackenbush. Gcod - genechip oncology database. *BMC Bioinformatics*, 12(1):46, 2011.

[13] Ronny Luss and Marc Teboulle. Conditional gradient algorithms for rank-one matrix approximations with a sparsity constraint. *SIAM Review*, 55(1):65–98, 2013.

[14] Lester W. Mackey. Deflation methods for sparse pca. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1017–1024. Curran Associates, Inc., 2009.

[15] J. Misra, W. Schmitt, D. Hwang, L. L. Hsiao, S. Gullans, G. Stephanopoulos, and G. Stephanopoulos. Interactive exploration of microarray gene expression patterns in a reduced dimensional space. *Genome research*, 12(7):1112–1120, July 2002.

[16] Baback Moghaddam, Yair Weiss, and Shai Avidan. Spectral bounds for sparse pca: Exact and greedy algorithms. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 915–922. MIT Press, Cambridge, MA, 2006.

[17] R.T. Rockafellar. *Convex Analysis*. Princeton mathematical series. Princeton University Press, 1970.

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

[19] Christian D. Sigg and Joachim M. Buhmann. Expectation-maximization for sparse and non-negative pca. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 960–967, New York, NY, USA, 2008. ACM.

[20] Bharath K. Sriperumbudur, David A. Torres, and Gert R. Lanckriet. A majorization-minimization approach to the sparse generalized eigenvalue problem. *Mach. Learn.*, 85(1-2):3–39, October 2011.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[22] Daniela M. Witten, Trevor Hastie, and Robert Tibshirani. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 2009.

[23] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[24] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:2006, 2004.