Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation

Konstantinos Benidis, Ying Sun, Prabhu Babu, and Daniel P. Palomar, *Fellow, IEEE*

***Abstract*—The problem of estimating sparse eigenvectors of a symmetric matrix has attracted a lot of attention in many applications, especially those with a high dimensional dataset. While classical eigenvectors can be obtained as the solution of a maximization problem, existing approaches formulate this problem by adding a penalty term into the objective function that encourages a sparse solution. However, the vast majority of the resulting methods achieve sparsity at the expense of sacrificing the orthogonality property. In this paper, we develop a new method to estimate dominant sparse eigenvectors without trading off their orthogonality. The problem is highly nonconvex and hard to handle. We apply the minorization–maximization framework, wherein we iteratively maximize a tight lower bound (surrogate function) of the objective function over the Stiefel manifold. The inner maximization problem turns out to be a rectangular Procrustes problem, which has a closed-form solution. In addition, we propose a method to improve the covariance estimation problem when its underlying eigenvectors are known to be sparse. We use the eigenvalue decomposition of the covariance matrix to formulate an optimization problem wherein we impose sparsity on the corresponding eigenvectors. Numerical experiments show that the proposed eigenvector extraction algorithm outperforms existing algorithms in terms of support recovery and explained variance, whereas the covariance estimation algorithms improve the sample covariance estimator significantly.**

***Index Terms*—Covariance estimation, minorization-maximization, procrustes, sparse PCA, stiefel manifold.**

I. Introduction

PRINCIPAL Component Analysis (PCA) is a popular technique for data analysis and dimensionality reduction [2]. It has been used in various fields of engineering and science with a large number of applications such as machine learning, financial asset trading, face recognition, and gene expression data analysis. Given a data matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ with $\text{rank}(\boldsymbol{A}) = r$, PCA finds sequentially orthogonal unit vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$, such that the variance of $\boldsymbol{A}\boldsymbol{v}_i$, which essentially is the projection of the data on the direction $\boldsymbol{v}_i$, for $i = 1, \ldots, r$, is maximized. The directions $\boldsymbol{v}_i$ are known as principal component (PC) loadings, while $\boldsymbol{A}\boldsymbol{v}_i$ are the corresponding principal components (PCs). The PC loadings are effectively the right singular vectors of $\boldsymbol{A}$ or the eigenvectors of the sample covariance matrix $\boldsymbol{S} = \frac{1}{n}\boldsymbol{A}^T\boldsymbol{A}$.

Manuscript received January 29, 2016; revised June 18, 2016; accepted August 10, 2016. Date of publication September 1, 2016; date of current version October 4, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsung-Hui Chang. This work was supported by the Hong Kong RGC 16206315 Research Grant. This work has been presented in part at the International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, March 20–25, 2016 [1].

The authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: kbenidis@ust.hk; ysunac@ust.hk; eeprabhubabu@ust.hk; palomar@ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2016.2605073

PCA has many optimal properties that made it so widely used. First, it captures the directions of maximum variance of the data, thus enabling us to compress the data with minimum information loss. Further, these directions are orthogonal to each other, i.e., they form an orthonormal basis. Finally, the PCs are uncorrelated which aids further statistical analysis. On the other hand, a particular disadvantage of PCA is that the PCs are usually linear combinations of all variables, i.e., the eigenvectors of $\boldsymbol{S}$ are dense. Even if the underlying covariance matrix from which the samples are generated indeed has sparse eigenvectors, we do not expect to get a sparse result due to estimation error. Further, in many applications the PCs have an actual physical meaning. Thus, a sparse eigenvector could significantly help the interpretability of the result.

Many different techniques have been proposed in this direction during the last two decades. In one of the first approaches, Jolliffe used various rotating techniques to obtain sparse loading vectors [3]. He showed though that it is impossible to preserve both the orthogonality of the loadings and the uncorrelatedness of the rotated components. In the same year, Cadima and Jolliffe suggested to simply set to zero all the elements that their absolute value is smaller than a threshold [4]. In [5], the authors propose the SCoTLASS algorithm which maximizes the Rayleigh quotient of the covariance matrix, while sparsity is enforced with the Lasso penalty [6]. Many recent approaches are based on reformulations or convex relaxations. For example in [7], Zou et al. formulate the sparse PCA problem as a ridge regression problem, while sparsity is imposed again using the Lasso penalty. In [8], d'Aspremont et al. form a semidefinite program (SDP) after a convex relaxation of the sparse PCA problem, leading to the DSPCA algorithm. In [9], the authors propose a greedy algorithm accompanied with a certificate of optimality. Low rank approximation of the data matrix is considered in [10], under sparsity penalties, while in [11], Journeé et al. reformulated the problem as an alternating optimization problem, resulting in the GPower algorithm. This algorithm turns out to be identical to the rSVD algorithm in [10], except for the initialization and the post-processing phases. Similar power-type truncation methods were considered in [12], [13]. In [14], the authors propose an iterative thresholding sparse PCA (ITSPCA) algorithm based on the QR decomposition. This algorithm is similar to the orthogonal iteration method with an additional truncation step to enforce sparsity. As the authors state though, convergence of the algorithm is not guaranteed. Finally, in [15], the sparse generalized eigenvalue problem is considered only for the first principal component, where the minorization-maximization (MM) framework is used.

Apart from the typical sparse PCA problem, several variations have been considered. Zass et al. impose sparsity on the eigenvectors while restricting all the elements of the eigenvectors to be non-negative [16], while in [17] a sparse PCA method is proposed with the additional constraint that the supports of

the eigenvectors are non-overlapping. Although useful, these extensions are not in the focus of this paper.

In the vast majority of the aforementioned algorithms, apart from the fact that the PCs are correlated, the orthogonality property of the loadings is also sacrificed for sparse solutions. The only two exceptions are: 1) the SCoTLASS algorithm that is suboptimal in the sense that it finds a sparse basis sequentially, rather than jointly, and 2) the ITSPCA algorithm proposed in [14] that we will use as a benchmark along with the GPower method.

The advantages of an orthogonal basis are well known. To begin with, an orthonormal basis can be extremely useful since it can reduce the potential computational cost of any processing procedure; this may not seem much for vector spaces of small dimension but it is invaluable for high dimensional vector spaces or function spaces. As an example, consider the solution of a linear system via Gaussian elimination. It requires $O(m^3)$ operations for a non-orthogonal basis, compared to $O(m)$ operations if the basis is orthogonal, where m is the dimension. Apart from all potential computational gains, orthogonality is a property much needed in various applications. Consider for example the problem of locating a moving object over time, the well known object tracking problem in computer vision community. Among other works, in [18] the authors use a set of features for object representation. They extract basic object models as subsets of this feature set. Sparse PCA is used so the models can capture the tracking object's varying appearance, while at the same time maintain a small number of features. One of the conditions for the object models to be good in terms of tracking performance and efficiency is that they are complementary to each other. This property is achieved through the orthogonality of the sparse eigenvectors.

Another issue in many contemporary applications is that the number of features m in the corresponding datasets is extremely large, while in many cases the number of samples n is limited. It is well know by now that the sample covariance $\boldsymbol{S}$ can be a very poor estimate of the population covariance matrix $\boldsymbol{\Sigma}$ if the number of samples is restricted. Since the population covariance matrix $\boldsymbol{\Sigma}$ is unknown, the classical PCA estimates the leading population eigenvectors by using the sample covariance matrix $\boldsymbol{S}$, which coincides with the maximum likelihood estimator (MLE) if $n \geq m$ and under the assumption that the samples are independent and identically distributed (i.i.d.), drawn from an m-dimensional Gaussian distribution. Many methods have been proposed to improve the covariance estimation in different settings and for different applications, e.g., for some representative works see [19]–[27] and references therein. However, in many cases we expect sparsity in the eigenvectors. For example, in the well-known protein-folding problem in bioinformatics, the underlying eigenvectors are expected to be sparse by nature as the number of protein positions in contact in a 3D fold is very small compared to the total number of positions in the protein [28]. Nevertheless, none of the existing methods has considered to combine the prior information of sparsity in the eigenvectors with the covariance estimation, especially in low sample settings where the estimation is poor.

In this paper we focus and solve the two aforementioned problems: 1) the orthogonal sparse eigenvector extraction and 2) the joint covariance estimation with sparse eigenvectors. First, we apply the MM framework on the sparse PCA problem which results in solving a sequence of rectangular Procrustes problems. With this approach, we obtain sparse results but with the orthogonality property retained. Then, we consider low sample settings where the population covariance matrices are known to have sparse eigenvectors. We formulate a covariance estimation problem where we impose sparsity on the eigenvectors. We propose two methods, i.e., alternating and joint estimation of the eigenvalues and eigenvectors, based on the MM framework. Both methods reduce to an iterative closed-form update with bounded iterations for the eigenvalues and a sequence of Procrustes problems for the eigenvectors, which maintain their orthogonality.

Throughout the paper we consider real-valued matrices for simplicity. However, all the results hold for complex-valued matrices with trivial modifications: in the complex-valued case $|x_i|$ denotes the modulus of x_i rather than the absolute value, while we should replace the transpose operation (i.e., $(\cdot)^T$) with the conjugate transpose operation (i.e., $(\cdot)^H$). Finally, we do not assume direct access to the data matrix $\boldsymbol{A}$. Nevertheless, all the formulations hold if either the data matrix $\boldsymbol{A}$ or the sample covariance matrix $\boldsymbol{S}$ is provided.

The rest of the paper is organized as follows: In Section II we first formulate the sparse eigenvector extraction and the covariance estimation problems. Then, we give a short review of the MM framework which will be the main tool to tackle both of the aforementioned problems. Finally we present the Procrustes problem since the solution of both our problems involve certain Procrustes reformulations. In Section III we present the solution of the sparse eigenvector extraction problem. In Section IV we consider the problem of joint covariance estimation with sparse eigenvectors and we propose two algorithms to iteratively minimize the associated objective function. In Section V we provide a convergence analysis of the proposed algorithms, while in Section VI we present an acceleration scheme that improves the convergence speed of the algorithms. Further, we provide a parameter selection analysis. Finally, in Section VII we present numerical experiments on artificial and real data, while we conclude the paper in Section VIII.

Notation: $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{R}$ denotes the real field, $\mathbb{R}^m$ ($\mathbb{R}^m_+$) the set of (non-negative) real vectors of size m, and $\mathbb{R}^{n\times m}$ the set of real matrices of size $n \times m$. Vectors are denoted by bold lower case letters and matrices by bold capital letters i.e., $\boldsymbol{x}$ and $\boldsymbol{X}$, respectively. The i-th entry of a vector is denoted by x_i, the i-th column of matrix $\boldsymbol{X}$ by $\boldsymbol{x}_i$, and the (i-th,j-th) element of a matrix by x_{ij}. A size m vector of ones is denoted by $\mathbf{1}_m$, while $\boldsymbol{I}_m$ denotes the identity matrix of size m. vec$(\cdot)$ denotes the vectorized form of a matrix. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose of a matrix, respectively, and Tr$(\cdot)$ its trace. Diag$(\boldsymbol{X})$ is a column vector consisting of all the diagonal elements of $\boldsymbol{X}$ and diag$(\boldsymbol{x})$ is a diagonal matrix formed with $\boldsymbol{x}$ at its principal diagonal. Given a vector $\boldsymbol{x} \in \mathbb{R}^{mn}$, $[\boldsymbol{x}]_{m\times n}$ is an $m \times n$ matrix such that vec$([\boldsymbol{x}]_{m\times n}) = \boldsymbol{x}$. $\|\boldsymbol{x}\|_0$ denotes the number of nonzero elements of a vector $\boldsymbol{x} \in \mathbb{R}^m$ and $\|\boldsymbol{X}\|_F$ the Frobenius norm of matrix $\boldsymbol{X}$. $\boldsymbol{S} \succcurlyeq 0$ means that the symmetric matrix $\boldsymbol{S}$ is positive semidefinite, while $\lambda^{(\boldsymbol{S})}_{\max}$ denotes its maximum eigenvalue. $\boldsymbol{X} \otimes \boldsymbol{Y}$ is the Kronecker product of the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. card$(\mathcal{A})$ denotes the cardinality of the set $\mathcal{A}$, $\mathcal{A} \bigcup \mathcal{B}$ the union of the sets $\mathcal{A}$ and $\mathcal{B}$, and $\mathcal{A} \setminus \mathcal{B}$ their difference. $[i : j]$ with $i \leq j$, denotes the set of all integers between (and including) i and j.

II. Problem Statement and Background

A. Sparse Eigenvector Extraction

Given a data matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, encoding n samples of dimension m, we can extract the leading eigenvector of the scaled sample covariance matrix $\boldsymbol{S} = \boldsymbol{A}^T\boldsymbol{A}$ by solving the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{maximize}} \quad & \boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u} \\ \text{subject to} \quad & \boldsymbol{u}^T\boldsymbol{u} = 1. \end{aligned} \tag{1}$$

In order to get a sparse result, we can include a regularization term in the objective that imposes sparsity, i.e.,

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{maximize}} \quad & \boldsymbol{u}^T\boldsymbol{S}\boldsymbol{u} - \rho\|\boldsymbol{u}\|_0 \\ \text{subject to} \quad & \boldsymbol{u}^T\boldsymbol{u} = 1, \end{aligned} \tag{2}$$

where ρ is a regularization parameter. Problem (2) can be generalized to extract multiple eigenvectors as follows:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \text{Tr}\left(\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\boldsymbol{D}\right) - \sum_{i=1}^{q}\rho_i\|\boldsymbol{u}_i\|_0 \\ \text{subject to} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_q. \end{aligned} \tag{3}$$

Here, q is the number of eigenvectors we wish to estimate, $\boldsymbol{U} \in \mathbb{R}^{m\times q}$, and $\boldsymbol{D} \succcurlyeq 0$ is a diagonal matrix giving weights to the different eigenvectors. In the case where $q = m$, $\boldsymbol{D}$ should be different from the (scaled) identity matrix since the first term reduces to a constant and $\boldsymbol{U}^\star = \boldsymbol{P}_m$, where $\boldsymbol{P}_m$ is a permutation matrix of size m. Many variations of this formulation have been considered for the extraction of multiple sparse eigenvectors, e.g., see [8], [11].

The optimization problem of (3) involves the maximization of a non-concave discontinuous objective function over a non-convex set, thus the problem is too hard to deal with directly. In order to handle the discontinuity of the ℓ_0-norm of (3), we approximate it by a continuous function $g_p(x)$, where $p > 0$ is a parameter that controls the approximation. Following [15], we consider an even function defined on $\mathbb{R}$, which is differentiable everywhere except at 0, concave and monotone increasing on $[0, +\infty)$, with $g_p(0) = 0$. Among the functions that satisfy the aforementioned criteria, in this paper we choose the function

$$g_p(x) = \frac{\log\,(1 + |x|/p)}{\log\,(1 + 1/p)}, \tag{4}$$

with $0 < p \leq 1$. This function is also used to replace the ℓ_1-norm in [29], and leads to the iteratively reweighted ℓ_1-norm minimization algorithm.

The function $g_p(\cdot)$ is not smooth which may cause an optimization algorithm to get stuck at a non-differentiable point [30]. To deal with the non-smoothness of $g_p(\cdot)$ we use a smoothened version, based on Nesterov's smooth minimization technique presented in [31] and following the results of [15], which is defined as:

$$g_p^\epsilon(x) = \begin{cases} \dfrac{x^2}{2\epsilon(p+\epsilon)\log\,(1+1/p)}, & |x| \leq \epsilon, \\ \dfrac{\log\left(\frac{p+|x|}{p+\epsilon}\right) + \frac{\epsilon}{2(p+\epsilon)}}{\log\,(1+1/p)}, & |x| > \epsilon, \end{cases} \tag{5}$$

with $0 < p \leq 1$ and $0 < \epsilon \ll 1$. This leads to the following approximate problem:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \text{Tr}\left(\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\boldsymbol{D}\right) - \sum_{j=1}^{q}\rho_j\sum_{i=1}^{m} g_p^\epsilon\left(u_{ij}\right) \\ \text{subject to} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_q. \end{aligned} \tag{6}$$

The problem presented in [15], is a special case of the above optimization problem, with $q = 1$. Nevertheless, it is not possible to follow the same procedure as in [15] to solve the problem due to the orthogonality constraint. Instead, we tackle this problem using the MM algorithm, which results in solving a sequence of rectangular Procrustes problems that have a closed-form solution based on singular value decomposition.

B. Covariance Estimation

We first consider a typical covariance estimation problem. We assume that the random variable $\boldsymbol{x} \in \mathbb{R}^m$ follows a zero mean Gaussian distribution with covariance $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. Given $n \geq m$ i.i.d. samples $\boldsymbol{x}_i$, with $i = 1, \ldots, n$, our goal is to estimate $\boldsymbol{\Sigma}$. The maximum likelihood estimator (MLE) of $\boldsymbol{\Sigma}$ is given by the solution of the following problem:

$$\begin{aligned} \underset{\boldsymbol{\Sigma}}{\text{minimize}} \quad & \log\det\left(\boldsymbol{\Sigma}\right) + \text{Tr}\left(\boldsymbol{S}\boldsymbol{\Sigma}^{-1}\right) \\ \text{subject to} \quad & \boldsymbol{\Sigma} \succcurlyeq 0, \end{aligned} \tag{7}$$

where $\boldsymbol{S}$ is the sample covariance matrix, i.e.,

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T. \tag{8}$$

The above problem is not convex but it can be easily transformed into a convex one by setting $\boldsymbol{\Psi} = \boldsymbol{\Sigma}^{-1}$. With this transformation we get:

$$\begin{aligned} \underset{\boldsymbol{\Psi}}{\text{minimize}} \quad & -\log\det\left(\boldsymbol{\Psi}\right) + \text{Tr}\left(\boldsymbol{S}\boldsymbol{\Psi}\right) \\ \text{subject to} \quad & \boldsymbol{\Psi} \succcurlyeq 0. \end{aligned} \tag{9}$$

The optimal solution of this problem is $\boldsymbol{\Psi} = \boldsymbol{S}^{-1}$, thus, the MLE of the covariance matrix is $\boldsymbol{\Sigma} = \boldsymbol{S}$, which is simply the sample covariance matrix.

Now, we consider the case where the population covariance matrix $\boldsymbol{\Sigma}$ is composed by sparse eigenvectors. We would like to estimate $\boldsymbol{\Sigma}$ taking into account the sparsity information. Thus, we need to reformulate the covariance estimation problem in terms of eigenvalues and eigenvectors. Further, we add a cardinality penalty on the first q principal eigenvector. Notice though that we estimate all m eigenvectors and not only the q principal ones since it is a covariance estimation and not an eigenvector extraction problem. Consider the eigenvalue decomposition of $\boldsymbol{\Psi}$, i.e., $\boldsymbol{\Psi} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, with $\boldsymbol{U}, \boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda}) \succcurlyeq 0$.

Then, we can formulate our problem as follows:

$$\begin{aligned} \underset{\boldsymbol{U},\boldsymbol{\Lambda}}{\text{minimize}} \quad & -\log\det\left(\boldsymbol{\Lambda}\right)+\text{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\right)+\sum_{i=1}^{q}\rho_i\|\boldsymbol{u}_i\|_0 \\ \text{subject to} \quad & \boldsymbol{\Lambda}\succcurlyeq 0, \\ & \lambda_i\leq\lambda_{i+1}, \quad i=1,\ldots,q-1, \\ & \lambda_q\leq\lambda_{q+i}, \quad i=1,\ldots,m-q, \\ & \boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_m. \end{aligned} \tag{10}$$

Let us first make some comments on the above problem. We penalize the cardinality of the first $q\leq m$ principal eigenvectors where each of them is associated with a different sparsity inducing parameter ρ_i. Thus, we need to keep the order of the first q eigenvectors intact. We succeed this by imposing ordering to the corresponding eigenvalues. Notice also that the principal eigenvector corresponds to the smallest eigenvalue of $\boldsymbol{\Psi}$ since $\boldsymbol{\Psi}=\boldsymbol{\Sigma}^{-1}$. It will be useful in the following to expand the sparsity term and include all eigenvectors by setting the redundant sparsity inducing parameters to zero, i.e., $\rho_i=0$ for $i=q+1,\ldots,m$. Again, we approximate the ℓ_0-norm by a differentiable function $g_p^{\epsilon}(\cdot)$, given by (5). This leads to the following approximate problem:

$$\begin{aligned} \underset{\boldsymbol{U},\boldsymbol{\Lambda}}{\text{minimize}} \quad & -\log\det\left(\boldsymbol{\Lambda}\right)+\text{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\right) \\ & +\sum_{j=1}^{m}\rho_j\sum_{i=1}^{m}g_p^{\epsilon}\left(u_{ij}\right) \\ \text{subject to} \quad & \boldsymbol{\Lambda}\succcurlyeq 0, \\ & \lambda_i\leq\lambda_{i+1}, \quad i=1,\ldots,q-1, \\ & \lambda_q\leq\lambda_{q+i}, \quad i=1,\ldots,m-q, \\ & \boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_m. \end{aligned} \tag{11}$$

Although in (11) we have approximated the objective of (10) with a continuous and differentiable function, the problem still remains too hard to deal with directly since it involves the minimization of a non-convex function over a non-convex set.

C. Shrinkage

In the case where the number of samples is lower than the dimension of the problem, i.e., when $n<m$, the sample covariance matrix $\boldsymbol{S}$ is low rank. As a result, all the covariance estimation problems that were presented are unbounded below.

We can overcome this problem by shrinking the sample covariance matrix towards an identity matrix [20], [32], i.e.,

$$\boldsymbol{S}_{\text{sh}}=(1-\delta)\boldsymbol{S}+\delta\boldsymbol{I}_m, \tag{12}$$

with $0<\delta\leq 1$. With this technique we bound the minimum eigenvalue of $\boldsymbol{S}_{\text{sh}}$ by δ, the matrix becomes full rank and the optimization problems are now well defined. We present the effect of shrinkage in the estimation of $\boldsymbol{\Sigma}$ in Section VII.

D. Review of the MM Framework

The minorization-maximization (if we maximize) or majorization-minimization (if we minimize) algorithm is a way to handle optimization problems that are too difficult to face directly [33]. Consider a general optimization problem

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{maximize}} \quad & f\left(\boldsymbol{x}\right) \\ \text{subject to} \quad & \boldsymbol{x}\in\mathcal{X}, \end{aligned}$$

where $\mathcal{X}$ is a closed set. We say that the function $f(\boldsymbol{x})$ is majorized at a given point $\boldsymbol{x}^{(k)}$ by the surrogate function $g(\boldsymbol{x}|\boldsymbol{x}^{(k)})$ if the following properties are satisfied:

- $f(\boldsymbol{x}^{(k)})=g(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k)})$,
- $f(\boldsymbol{x})\geq g(\boldsymbol{x}|\boldsymbol{x}^{(k)})$, $\forall\boldsymbol{x}\in\mathcal{X}$,
- $\nabla f(\mathbf{x}^{(k)})=\nabla g(\mathbf{x}^{(k)}|\mathbf{x}^{(k)})$.

Then, $\boldsymbol{x}$ is iteratively updated (with k denoting iterations) as:

$$\boldsymbol{x}^{(k+1)}=\arg\max_{\boldsymbol{x}\in\mathcal{X}} g\left(\boldsymbol{x}|\boldsymbol{x}^{(k)}\right). \tag{13}$$

It can be seen easily that $f(\boldsymbol{x}^{(k)})\leq f(\boldsymbol{x}^{(k+1)})$ holds. The majorization-minimization algorithm works in an equivalent way, such that in each update $f(\boldsymbol{x}^{(k)})\geq f(\boldsymbol{x}^{(k+1)})$ holds.

In practice, it is not a trivial task to find a surrogate function such that the maximizer of the minorization (or minimizer of the majorization) function of the objective can be easily found or even have a closed-form solution. The following lemma will be useful for the MM algorithms that will be derived throughout this paper:

Lemma 1: On the set $\left\{\boldsymbol{U}\in\mathbb{R}^{m\times q}|\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}_q\right\}$, the function $\sum_{j=1}^{q}\rho_j\sum_{i=1}^{m}g_p^{\epsilon}(u_{ij})$ is majorized at $\boldsymbol{U}_0$ by $2\text{Tr}(\boldsymbol{H}^T\boldsymbol{U})+c$, where

$$\boldsymbol{H}=\left[\text{diag}\left(\boldsymbol{w}-\boldsymbol{w}_{\max}\otimes\mathbf{1}_m\right)\boldsymbol{u}_0\right]_{m\times q}, \tag{14}$$

$$c=2\left(\mathbf{1}_q^T\boldsymbol{w}_{\max}\right)-\boldsymbol{u}_0{}^T\text{diag}\left(\boldsymbol{w}\right)\boldsymbol{u}_0. \tag{15}$$

The weights $\boldsymbol{w}\in\mathbb{R}_+^{mq}$ are given by

$$w_i=\begin{cases}\dfrac{\rho_i}{2\epsilon(p+\epsilon)\log\left(1+1/p\right)}, & |u_{0,i}|\leq\epsilon, \\ \dfrac{\rho_i}{2\log\left(1+1/p\right)|u_{0,i}|\left(|u_{0,i}|+p\right)}, & |u_{0,i}|>\epsilon,\end{cases} \tag{16}$$

where $\boldsymbol{u}_0=\text{vec}(\boldsymbol{U}_0)$, and $\boldsymbol{w}_{\max}\in\mathbb{R}_+^q$, with $w_{\max,i}$ being the maximum weight that corresponds to $\boldsymbol{u}_{0,i}$.

Proof: See Appendix A. ■

E. Procrustes Problems

Consider the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{maximize}} \quad & \text{Tr}\left(\boldsymbol{Y}^T\boldsymbol{X}\right) \\ \text{subject to} \quad & \boldsymbol{X}^T\boldsymbol{X}=\boldsymbol{I}_q, \end{aligned} \tag{17}$$

where $\boldsymbol{X},\boldsymbol{Y}\in\mathbb{R}^{m\times q}$. Notice that problem (17) is equivalent to

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \|\boldsymbol{X}-\boldsymbol{Y}\|_F^2 \\ \text{subject to} \quad & \boldsymbol{X}^T\boldsymbol{X}=\boldsymbol{I}_q, \end{aligned} \tag{18}$$

which is a Procrustes problem.

Lemma 2: For $m=q$ $(m>q)$, problem (17) can be transformed into an orthogonal (rectangular) Procrustes problem and its optimal solution is $\boldsymbol{X}^{\star}=\boldsymbol{V}_L\boldsymbol{V}_R^T$, where $\boldsymbol{V}_L,\boldsymbol{V}_R$ are the left and right singular vectors of the matrix $\boldsymbol{Y}$, respectively [34], [35, Proposition 7].

III. Sparse PCA

In this section we return to the sparse eigenvector extraction problem as formulated in (6). In the following, we apply the MM algorithm and derive a tight lower bound (surrogate function), $g(\boldsymbol{U}|\boldsymbol{U}^{(k)})$, for the objective function of (6), denoted by $f(\boldsymbol{U})$, at the $(k+1)$-th iteration.

Proposition 1: The function $f(\boldsymbol{U})$ is lowerbounded at $\boldsymbol{U}^{(k)}$ by the surrogate function

$$g\left(\boldsymbol{U}|\boldsymbol{U}^{(k)}\right) = 2\text{Tr}\left(\left(\boldsymbol{G}^{(k)} - \boldsymbol{H}^{(k)}\right)^T \boldsymbol{U}\right) + const, \quad (19)$$

where

$$\boldsymbol{G}^{(k)} = \boldsymbol{S}\boldsymbol{U}^{(k)}\boldsymbol{D}, \quad (20)$$

$$\boldsymbol{H}^{(k)} = \left[\text{diag}\left(\boldsymbol{w}^{(k)} - \boldsymbol{w}^{(k)}_{\max} \otimes \boldsymbol{1}_m\right)\boldsymbol{u}^{(k)}\right]_{m\times q}, \quad (21)$$

$\boldsymbol{u}^{(k)} = \text{vec}(\boldsymbol{U}^{(k)})$, and $const$ denotes an optimization irrelevant constant. Equality is achieved when $\boldsymbol{U} = \boldsymbol{U}^{(k)}$.

Proof: The first term of the objective is convex so a lower bound can be constructed by its first order Taylor expansion:

$$\text{Tr}\left(\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\boldsymbol{D}\right) \geq 2\text{Tr}\left(\left(\boldsymbol{S}\boldsymbol{U}^{(k)}\boldsymbol{D}\right)^T \boldsymbol{U}\right) + c_1, \quad (22)$$

where $c_1 = -\text{Tr}({\boldsymbol{U}^{(k)}}^T\boldsymbol{S}\boldsymbol{U}^{(k)}\boldsymbol{D})$ is a constant. For the second term, using the results from Lemma 1 it is straightforward to show that it is lowerbounded by the function $-2\text{Tr}(\boldsymbol{H}^{(k)}\boldsymbol{U}) - c_2$, where $\boldsymbol{H}^{(k)}$ is given by (21) and $c_2 = 2(\boldsymbol{1}_q^T\boldsymbol{w}_{\max}) - {\boldsymbol{u}^{(k)}}^T\text{diag}(\boldsymbol{w})\boldsymbol{u}^{(k)}$ is a constant. ■

Now, we drop the constants and the optimization problem of every MM iteration takes the following form:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \text{Tr}\left(\left(\boldsymbol{G}^{(k)} - \boldsymbol{H}^{(k)}\right)^T \boldsymbol{U}\right) \\ \text{subject to} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_q. \end{aligned} \quad (23)$$

Proposition 2: The optimal solution of the optimization problem (23) is $\boldsymbol{U}^\star = \boldsymbol{V}_L\boldsymbol{V}_R^T$, where $\boldsymbol{V}_L \in \mathbb{R}^{m\times q}$ and $\boldsymbol{V}_R \in \mathbb{R}^{q\times q}$ are the left and right singular vectors of the matrix $(\boldsymbol{G}^{(k)} - \boldsymbol{H}^{(k)})$, respectively.

Proof: The proof comes directly from Lemma 2. ■

In Algorithm 1 we summarize the above iterative procedure. We will refer to it as IMRP. Since the algorithm does not perform any hard thresholding, the resulting eigenvectors do not have zero elements but rather very small values. To this end, we can set to zero all the values that are below a threshold and obtain sparse eigenvectors. As it will be shown in the numerical experiments, the effect of this thresholding on the orthogonality of the eigenvectors is negligible.

A. Computational Complexity of IMRP

In this section we study the computational complexity of IMRP. In every iteration, first we need to compute the matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ which involve some matrix multiplications. Then, we need to perform an SVD and finally a matrix multiplication for the variable update. The matrices $\boldsymbol{G}, \boldsymbol{H}$ can be computed in $O(mqn)$ and $O(mq)$ operations, respectively. The complexity of the SVD is $O(mq^2)$, while the last matrix multiplication can be computed in $O(mq^2)$ operations. Thus, the complexity of the algorithm in every iteration is $O(mqn + mq^2)$.

Algorithm 1: IMRP - Iterative Minimization of Rectangular Procrustes for the Sparse Eigenvector Problem (6).

1: Set $k = 0$, choose $\boldsymbol{U}^{(0)} \in \{\boldsymbol{U}|\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_q\}$
2: **repeat**:
3: Compute $\boldsymbol{G}^{(k)}, \boldsymbol{H}^{(k)}$ with (20)-(21)
4: Compute $\boldsymbol{V}_L, \boldsymbol{V}_R$, the left and right singular vectors of $(\boldsymbol{G}^{(k)} - \boldsymbol{H}^{(k)})$, respectively
5: $\boldsymbol{U}^{(k+1)} = \boldsymbol{V}_L\boldsymbol{V}_R^T$
6: $k \leftarrow k+1$
7: **until** convergence
8: **return** $\boldsymbol{U}^{(k)}$

In general, in high dimensional problems we are interested in extracting a low dimensional subspace that contains most of the data information. Further, it is common in the high dimensional datasets that the number of samples is limited or significantly lower than the dimension of the problem, i.e., $m \gg n$. In these cases, IMRP is scalable to very high dimensions as will be shown in Section VII.

B. Explained Variance

In the ordinary PCA the principal components are uncorrelated while the corresponding loadings are orthogonal. If we denote by $\boldsymbol{Y}$ the ordinary principal components, the total explained variance can be calculated as $\text{Tr}(\boldsymbol{Y}^T\boldsymbol{Y})$. If the principal components are correlated though, computing the total variance this way will overestimate the true explained variance. An approach to overcome this issue was first suggested in [7] (and adopted in [11]), where the authors introduced the notion of adjusted variance. The idea is to remove the correlations of the principal components sequentially. This can be done efficiently by the QR decomposition: if $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ is a data matrix and $\boldsymbol{U} \in \mathbb{R}^{m\times q}$ are the q estimated loadings, then the adjusted variance is simply

$$\text{AdjVar}\left(\boldsymbol{U}\right) = \text{Tr}\left(\boldsymbol{R}^2\right), \quad (24)$$

where $\boldsymbol{A}\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{R}$, is the QR decomposition of $\boldsymbol{A}\boldsymbol{U}$. The explained variance percentage can be then computed as $\text{AdjVar}(\boldsymbol{U})/\text{AdjVar}(\boldsymbol{U}_{\text{PCA}})$, where $\boldsymbol{U}_{\text{PCA}}$ are the first q eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$.

As mentioned in [10], in the above approach the lack of orthogonality in the loadings is not addressed. Thus, a new approach was proposed: when the loading vectors are not orthogonal we should not consider separate projections of the data matrix onto each of them. Instead, we should project the data matrix onto the q-dimensional subspace, i.e., $\boldsymbol{A}_q = \boldsymbol{A}\boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T$. Then, the total variance is simply $\text{Tr}(\boldsymbol{A}_q^T\boldsymbol{A}_q)$ and the cumulative percentage of explained variance (CPEV) can be computed as

$$\text{CPEV} = \text{Tr}\left(\boldsymbol{A}_q^T\boldsymbol{A}_q\right)/\text{Tr}\left(\boldsymbol{A}^T\boldsymbol{A}\right). \quad (25)$$

In this paper we adopt the second approach and compute the explained variance using (25).

IV. Sparse Eigenvectors in Covariance Estimation

In this section we return to the problem of covariance estimation with sparse eigenvectors. We consider the formulation (11), i.e.,

$$\begin{aligned}\underset{\boldsymbol{U},\boldsymbol{\Lambda}}{\text{minimize}} \quad & -\log\det\left(\boldsymbol{\Lambda}\right)+\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{T}\right)\\ & +\sum_{j=1}^{m}\rho_j\sum_{i=1}^{m}g_p^{\epsilon}\left(u_{ij}\right)\\ \text{subject to} \quad & \boldsymbol{\Lambda}\succcurlyeq 0,\\ & \lambda_i\leq\lambda_{i+1}, \quad i=1,\ldots,q-1,\\ & \lambda_q\leq\lambda_{q+i}, \quad i=1,\ldots,m-q,\\ & \boldsymbol{U}^{T}\boldsymbol{U}=\boldsymbol{I}_m.\end{aligned}$$

To deal with this problem, we propose two methods based on the MM framework. In Section IV-A we perform an alternating optimization of the eigenvalues and eigenvectors, while in Section IV-B we estimate them jointly.

A. Alternating Optimization Using the MM Framework

We begin with the optimization problem (11) which is highly non-convex. We tackle it by alternating optimization of $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$. For fixed $\boldsymbol{U}$ the optimization problem over $\boldsymbol{\lambda}$ can be written in the following form:

$$\begin{aligned}\underset{\boldsymbol{\lambda}}{\text{minimize}} \quad & -\sum_{i=1}^{m}\log\lambda_i+\sum_{i=1}^{m}z_i\lambda_i\\ \text{subject to} \quad & \lambda_i\leq\lambda_{i+1}, \quad i=1,\ldots,q-1,\\ & \lambda_q\leq\lambda_{q+i}, \quad i=1,\ldots,m-q,\end{aligned} \tag{26}$$

where we have dropped the positive semidefinite constraint of $\boldsymbol{\Lambda}=\mathrm{diag}(\boldsymbol{\lambda})$ since it is implicit form the log function, and $\boldsymbol{z}=\mathrm{Diag}(\boldsymbol{U}^{T}\boldsymbol{S}\boldsymbol{U})\geq\boldsymbol{0}$, since $\boldsymbol{S}\succcurlyeq\boldsymbol{0}$.

The optimization problem (26) is convex thus we can use any standard solver to obtain the optimal solution. However, since the problem has to be solved several times during the MM procedure the computational cost can be significant, especially for high dimensions. To this end, we propose an iterative closed-form update of the parameter $\boldsymbol{z}$ that will allow us to obtain the optimal solution for $\boldsymbol{\lambda}$ with a lower complexity. This algorithm terminates in at most $\min(2q-1,m-1)$ steps and is scalable to very high dimensions as will be shown in Section VII.

We start from the corresponding unconstrained version of problem (26) whose solution is

$$\boldsymbol{\lambda}^{(0)}=\frac{1}{\boldsymbol{z}^{(0)}}, \tag{27}$$

where $\boldsymbol{z}^{(0)}=\boldsymbol{z}$. If this solution is feasible then it is the optimal one. Else, we need to update $\boldsymbol{z}$. In every iteration, all the non-overlapping blocks of z_i's that satisfy certain conditions need to be updated in parallel. In the k-th iteration we distinguish three different cases:

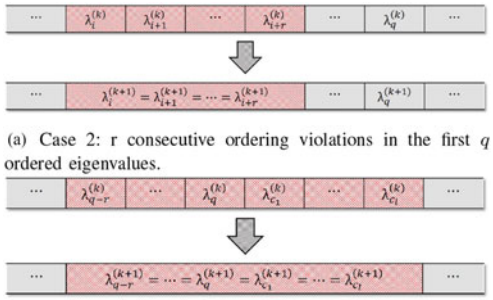
Case 1: $\boldsymbol{\lambda}^{(k)}=1/\boldsymbol{z}^{(k)}$ satisfies all the constraints of problem (26). Then the optimal solution is $\boldsymbol{\lambda}^{\star}=\boldsymbol{\lambda}^{(k)}$.

Case 2: $\boldsymbol{\lambda}^{(k)}=1/\boldsymbol{z}^{(k)}$ violates $r\geq 1$ consecutive ordering constraints of the first q eigenvalues (first constraint set of (26)). For any such block violation we need to update $\boldsymbol{z}^{(k)}$.

Case 3: $\boldsymbol{\lambda}^{(k)}=1/\boldsymbol{z}^{(k)}$ violates $r+l\geq 1$ consecutive ordering constraints, with $r\geq 0$ and $l\geq 1$, including the last $r+1$ ordered and a set of l unordered eigenvalues (second constraint set of (26)). Since we do not impose ordering on the $m-q$ last eigenvalues, any of them could violate the inequality with λ_q and not only the neighboring ones. Thus, we use the indices $c_1,\ldots,c_l$, with $c_i>q$, for $i=1,\ldots,l$, and $c_i\in\mathcal{C}$, with $\mathcal{C}$ the set of indices of the eigenvalues that violate the inequality constraints with λ_q. We further denote by $\mathcal{A}\subseteq\mathcal{C}$, with $\mathrm{card}(\mathcal{A})=p<l$, the set of indices given by

$$\mathcal{A}=\left\{c_i\middle|z_{c_i}^{(k)}\geq\frac{1}{r+l-i+1}\left(\sum_{s=0}^{r}z_{q-s}^{(k)}+\sum_{s=0}^{l-i-1}z_{c_{l-s}}^{(k)}\right)\right\}. \tag{28}$$

For any such block violation we need to update $\boldsymbol{z}^{(k)}$.



(a) Case 2: r consecutive ordering violations in the first q ordered eigenvalues.

(b) Case 3: consecutive ordering violations in the last $r+1$ ordered and a set of l unordered eigenvalues.

Fig. 1. Visual illustration of the two constraint violation cases.

In Fig. 1 we illustrate the two possible cases of constraint violations, i.e., cases 2 and 3. We observe that for any block violation, all the involved eigenvalues will become equal in the next iteration. The procedure is summarized in Table I. We will refer to it as AOCE$_\lambda$ hereafter.

Proposition 3: The iterative-closed form update procedure given in Table I converges to the solution of problem (26).

Proof: See Appendix B. ■

Now, for fixed $\boldsymbol{\Lambda}$ the problem over $\boldsymbol{U}$ becomes:

$$\begin{aligned}\underset{\boldsymbol{U}}{\text{minimize}} \quad & \mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{T}\right)+\sum_{j=1}^{q}\rho_j\sum_{i=1}^{m}g_p^{\epsilon}\left(u_{ij}\right)\\ \text{subject to} \quad & \boldsymbol{U}^{T}\boldsymbol{U}=\boldsymbol{I}_m.\end{aligned} \tag{29}$$

For the second term we can use the same bound as the one for problem (6). However, we cannot linearize the first term as previously since the linear approximation is a lower and not an upper bound of a convex function.

To minimize the objective function we apply the MM algorithm and derive a tight upper bound, $g_{\text{alt}}(\boldsymbol{U}|\boldsymbol{U}^{(k)})$, for the objective function of (29), denoted by $f_{\text{alt}}(\boldsymbol{U})$, at the $(k+1)$-th iteration.

Proposition 4: The function $f_{\text{alt}}(\boldsymbol{U})$ is upper bounded at $\boldsymbol{U}^{(k)}$ by the surrogate function

$$g_{alt}\left(\boldsymbol{U}|\boldsymbol{U}^{(k)}\right)=2Tr\left(\left(\boldsymbol{G}_{\text{alt}}^{(k)}+\boldsymbol{H}^{(k)}\right)^{T}\boldsymbol{U}\right)+const, \tag{30}$$

where

$$\boldsymbol{G}_{\text{alt}}^{(k)}=\left[\left(\boldsymbol{\Lambda}\otimes\left(\boldsymbol{S}-\lambda_{\max}^{(\boldsymbol{S})}\boldsymbol{I}_m\right)\right)\boldsymbol{u}^{(k)}\right]_{m\times m}, \tag{31}$$

$$\boldsymbol{H}^{(k)}=\left[\mathrm{diag}\left(\boldsymbol{w}^{(k)}-\boldsymbol{w}_{\max}^{(k)}\otimes\boldsymbol{1}_m\right)\boldsymbol{u}^{(k)}\right]_{m\times m}, \tag{32}$$

$\boldsymbol{u}^{(k)}=\mathrm{vec}(\boldsymbol{U}^{(k)})$, and $const$ denotes an optimization irrelevant constant. Equality is achieved when $\boldsymbol{U}=\boldsymbol{U}^{(k)}$.

TABLE I
UPDATES AND OPTIMAL SOLUTION OF THE ITERATIVE PROCEDURE THAT SOLVES THE OPTIMIZATION PROBLEM (26)

	Case 1	Case 2		Case 3
Conditions	$z_i^{(k)} \geq z_{i+1}^{(k)}, i \in [1:q-1]$	$z_{j-1}^{(k)} > z_j^{(k)}$ if $j > 1$		$z_{q-r-1}^{(k)} > z_{q-r}^{(k)}$
		$z_i^{(k)} \leq z_{i+1}^{(k)}, \quad i \in [j : j+k-1]$		$z_i^{(k)} \leq z_{i+1}^{(k)}, \quad i \in [q-r : q-1]$
	$z_q^{(k)} \geq z_{q+i}^{(k)}, i \in [1:m-q]$	if $j+k < q$	if $j+k = q$	$z_q^{(k)} \leq z_{c_i}^{(k)}, \quad i \in [1:k]$
		$z_{j+k}^{(k)} > z_{j+k+1}^{(k)}$	$z_q^{(k)} > z_{q+i}^{(k)}, i \in [1:m-q]$	$z_q^{(k)} > z_{q+i}^{(k)}, \quad i \in [1:m-q] \setminus \mathcal{C}$
Block Updates	-	$z_i^{(k+1)} = \frac{1}{k+1} \sum_{i=0}^{k} z_{j+i}^{(k)}, \ i \in [j : j+k]$		$z_i^{(k+1)} = z_i^{(k)}, \quad i \in \mathcal{C} \setminus \mathcal{A}$
				$z_i^{(k+1)} = \frac{1}{r+p+1} \sum_{i=0}^{r} z_{q-i}^{(k)} + \sum_{i=1}^{p} z_{a_i}^{(k)}, \ i \in [q-r:q] \bigcup \mathcal{A}$
Solution	$\boldsymbol{\lambda}^\star = \frac{1}{\boldsymbol{z}^{(k)}}$	-		-

Proof: For the first term of the objective it holds that

$$\mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\right) = \boldsymbol{u}^T(\boldsymbol{\Lambda} \otimes \boldsymbol{S})\boldsymbol{u}, \tag{33}$$

where $\boldsymbol{u} = \mathrm{vec}(\boldsymbol{U})$. In a similar manner as in the proof of Lemma 1, it is easy to show that the following holds:

$$\boldsymbol{u}^T(\boldsymbol{\Lambda} \otimes \boldsymbol{S})\boldsymbol{u} \leq 2\mathrm{Tr}({\boldsymbol{G}_{\text{alt}}^{(k)}}^T \boldsymbol{U}) + c_3, \tag{34}$$

where $\boldsymbol{G}_{\text{alt}}^{(k)} = [(\boldsymbol{\Lambda} \otimes (\boldsymbol{S} - \lambda_{\max}^{(\boldsymbol{S})} \boldsymbol{I}_m))\boldsymbol{u}^{(k)}]_{m \times m}$ and $c_3 = 2\lambda_{\max}^{(\boldsymbol{S})} \boldsymbol{1}^T \boldsymbol{\lambda} - {\boldsymbol{u}^{(k)}}^T (\boldsymbol{\Lambda} \otimes \boldsymbol{S})\boldsymbol{u}^{(k)}$ is a constant. For the second term it is straightforward from Lemma 1 that an upper bound is the function $2\mathrm{Tr}({\boldsymbol{H}^{(k)}}^T \boldsymbol{U}) + c_4$, with $\boldsymbol{H}^{(k)} = [\mathrm{diag}(\boldsymbol{w}^{(k)} - \boldsymbol{w}_{\max}^{(k)} \otimes \boldsymbol{1}_m)\boldsymbol{u}^{(k)}]_{m \times m}$ and $c_4 = \boldsymbol{1}_m^T \boldsymbol{w}_{\max}^{(k)} - {\boldsymbol{u}^{(k)}}^T \mathrm{diag}(\boldsymbol{w}^{(k)} - \boldsymbol{w}_{\max}^{(k)} \otimes \boldsymbol{1}_m)\boldsymbol{u}^{(k)}$ a constant. ■

Now, we drop the constants and the optimization problem of every MM iteration takes the following form:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{minimize}} \quad & \mathrm{Tr}\left(\left(\boldsymbol{G}_{\text{alt}}^{(k)} + \boldsymbol{H}^{(k)}\right)^T \boldsymbol{U}\right) \\ \text{subject to} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_m. \end{aligned} \tag{35}$$

Proposition 5: The optimal solution of the optimization problem (35) is $\boldsymbol{U}^\star = \boldsymbol{V}_L \boldsymbol{V}_R^T$, where $\boldsymbol{V}_L \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V}_R \in \mathbb{R}^{m \times m}$ are the left and right singular vectors of the matrix $-(\boldsymbol{G}_{\text{alt}}^{(k)} + \boldsymbol{H}^{(k)})$, respectively.

Proof: The proof comes directly from Lemma 2. ■

In Algorithm 2 we summarize the above iterative procedure. We will refer to it as AOCE.

B. Joint Optimization Using the MM Framework

Let us consider again the formulation (11) with the variable transformation $\boldsymbol{\Xi} = \boldsymbol{\Lambda}^{-1}$. The optimization problem becomes:

$$\begin{aligned} \underset{\boldsymbol{U},\boldsymbol{\Xi}}{\text{minimize}} \quad & \log\det(\boldsymbol{\Xi}) + \mathrm{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Xi}^{-1}\boldsymbol{U}^T\right) \\ & + \sum_{j=1}^{m} \rho_j \sum_{i=1}^{m} g_p^{\epsilon}(u_{ij}) \\ \text{subject to} \quad & \boldsymbol{\Xi} \succcurlyeq 0, \\ & \xi_i \geq \xi_{i+1}, \quad i = 1, \ldots, q-1, \\ & \xi_q \geq \xi_{q+i}, \quad i = 1, \ldots, m-q, \\ & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_m. \end{aligned} \tag{36}$$

Here $\boldsymbol{U}, \boldsymbol{\Xi} \in \mathbb{R}^{m \times m}$, with $\boldsymbol{\Xi} = \mathrm{diag}(\boldsymbol{\xi}) \succcurlyeq 0$. Now, we derive a tight upper bound, $g_{\text{jnt}}(\boldsymbol{U}, \boldsymbol{\Xi} | \boldsymbol{U}^{(k)}, \boldsymbol{\Xi}^{(k)})$, for the objective function of (36), denoted by $f_{\text{jnt}}(\boldsymbol{U}, \boldsymbol{\Xi})$, at the $(k+1)$-th iteration.

Proposition 6: The function $f_{\text{jnt}}(\boldsymbol{U}, \boldsymbol{\Xi})$ is upper bounded at $(\boldsymbol{U}^{(k)}, \boldsymbol{\Xi}^{(k)})$ by the surrogate function

$$g_{\text{jnt}}\left(\boldsymbol{U}, \boldsymbol{\Xi} | \boldsymbol{U}^{(k)}, \boldsymbol{\Xi}^{(k)}\right) = g_\xi(\boldsymbol{\Xi}) + g_u(\boldsymbol{U}) + const, \tag{37}$$

where

$$g_\xi(\boldsymbol{\Xi}) = \log\det(\boldsymbol{\Xi}) + \mathrm{Tr}\left(\boldsymbol{G}_{\text{jnt}}^{(k)}\boldsymbol{\Xi}\right) + \lambda_{\max}^{(\boldsymbol{S})} \mathrm{Tr}\left(\boldsymbol{\Xi}^{-1}\right), \tag{38}$$

Algorithm 2: AOCE - Alternating Optimization for Covariance Estimation for the Problem (11).

1: Set $k = 0$, choose $\boldsymbol{U}^{(0)} \in \{\boldsymbol{U} | \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_q\}$
2: **repeat**:
3: Compute $\boldsymbol{\lambda}^{(k+1)}$ from Proposition 3
4: Compute $\boldsymbol{G}_{\text{alt}}^{(k)}, \boldsymbol{H}^{(k)}$ with (31)-(32)
5: Compute $\boldsymbol{V}_L, \boldsymbol{V}_R$, the left and right singular vectors of $-(\boldsymbol{G}_{\text{alt}}^{(k)} + \boldsymbol{H}^{(k)})$, respectively
6: $\boldsymbol{U}^{(k+1)} = \boldsymbol{V}_L \boldsymbol{V}_R^T$
7: $k \leftarrow k+1$
8: **until** convergence
9: **return** $\boldsymbol{U}^{(k)}, \boldsymbol{\lambda}^{(k)}$

with

$$\boldsymbol{G}_{\text{jnt}}^{(k)} = -\left(\boldsymbol{\Xi}^{(k)}\right)^{-1} \boldsymbol{U}^{(k)^T} \left(\boldsymbol{S} - \lambda_{\max}^{(\boldsymbol{S})} \boldsymbol{I}_m\right) \boldsymbol{U}^{(k)} \left(\boldsymbol{\Xi}^{(k)}\right)^{-1} \tag{39}$$

and

$$g_u(\boldsymbol{U}) = 2Tr\left(\boldsymbol{H}_{\text{jnt}}^{(k)^T} \boldsymbol{U}\right), \tag{40}$$

with

$$\boldsymbol{H}_{\text{jnt}}^{(k)} = \boldsymbol{H}^{(k)} + \left(\boldsymbol{S} - \lambda_{\max}^{(\boldsymbol{S})} \boldsymbol{I}_m\right) \boldsymbol{U}^{(k)} \left(\boldsymbol{\Xi}^{(k)}\right)^{-1}. \tag{41}$$

The term $\boldsymbol{H}^{(k)}$ is given by (32), while $const$ denotes an optimization irrelevant constant.

Proof: Based on Lemma 1 we can upper bound the third term of the objective with the function $2\text{Tr}(\boldsymbol{H}^{(k)^T}\boldsymbol{U}) + c_4$, with $\boldsymbol{H}^{(k)}$ given by (32).

The second term of the objective function of (36), denoted by f, is jointly convex on $\boldsymbol{U}, \boldsymbol{\Xi} = \text{diag}(\boldsymbol{\xi})$. One way to establish convexity of f is via its epigraph using the Schur complement:

$$\text{epi}(f) = \left\{ (\boldsymbol{U}, \boldsymbol{\xi}, t) \,\middle|\, \text{diag}(\boldsymbol{\xi}) \succ \boldsymbol{0}, \begin{bmatrix} \text{diag}(\boldsymbol{\xi} \otimes \boldsymbol{1}_m) & \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{u}}^T & t \end{bmatrix} \succcurlyeq \boldsymbol{0} \right\},$$

where $\tilde{\boldsymbol{u}} = \text{vec}(\boldsymbol{S}^{1/2}\boldsymbol{U})$. Without loss of generality we have assumed that all the eigenvalues ξ_i are strictly positive. The last condition is a linear matrix inequality in $(\boldsymbol{U}, \boldsymbol{\xi}, t)$, and therefore $\text{epi}(f)$ is convex.

We can subtract the maximum eigenvalue of the sample covariance matrix $\boldsymbol{S}$ and therefore create a jointly concave term. An upper bound to this term is its first order Taylor expansion. It can be shown that

$$\begin{aligned} \text{Tr}\left(\boldsymbol{S}\boldsymbol{U}\boldsymbol{\Xi}^{-1}\boldsymbol{U}^T\right) \leq\ & 2\text{Tr}\left(\boldsymbol{F}^{(k)^T}\boldsymbol{U}\right) + \text{Tr}\left(\boldsymbol{G}_{\text{jnt}}^{(k)}\boldsymbol{\Xi}\right) \\ & + \lambda_{\max}^{(\boldsymbol{S})} \text{Tr}\left(\boldsymbol{\Xi}^{-1}\right) + c_5 \end{aligned} \tag{42}$$

where $\boldsymbol{F}^{(k)} = (\boldsymbol{S} - \lambda_{\max}^{(\boldsymbol{S})}\boldsymbol{I}_m)\boldsymbol{U}^{(k)}(\boldsymbol{\Xi}^{(k)})^{-1}$ and $\boldsymbol{G}_{\text{jnt}}^{(k)} = -(\boldsymbol{\Xi}^{(k)})^{-1}\boldsymbol{U}^{(k)^T}\boldsymbol{F}^{(k)}$. The constant c_5 is given by $c_5 = -\text{Tr}(\boldsymbol{F}^{(k)}\boldsymbol{U}^{(k)^T}) - \text{Tr}(\boldsymbol{G}_{\text{jnt}}^{(k)}\boldsymbol{\Xi}^{(k)})$.

We observe that now the variables are decoupled. Thus, by combining the upper bounds for the second and the third term we can derive the functions $g_u(\cdot)$ and $g_\xi(\cdot)$, with $\boldsymbol{H}_{\text{jnt}}^{(k)} = \boldsymbol{H}^{(k)} + \boldsymbol{F}^{(k)}$. ■

Now, in every MM iteration we need to solve the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{U},\boldsymbol{\Xi}}{\text{minimize}} \quad & g_\xi(\boldsymbol{\Xi}) + g_u(\boldsymbol{U}) \\ \text{subject to} \quad & \boldsymbol{\Xi} \succcurlyeq 0, \\ & \xi_i \geq \xi_{i+1}, \quad i = 1, \ldots, q-1, \\ & \xi_q \geq \xi_{q+i}, \quad i = 1, \ldots, m-q, \\ & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_m. \end{aligned} \tag{43}$$

Since the variables are decoupled we can optimize each one of them separately. The optimization problem for $\boldsymbol{\Xi}$ becomes:

$$\begin{aligned} \underset{\boldsymbol{\xi}}{\text{minimize}} \quad & \sum_{i=1}^{m} \left(\log \xi_i + \alpha_i \xi_i + \lambda_{\max}^{(\boldsymbol{S})} \frac{1}{\xi_i}\right) \\ \text{subject to} \quad & \xi_i \geq \xi_{i+1}, \quad i = 1, \ldots, q-1, \\ & \xi_q \geq \xi_{q+i}, \quad i = 1, \ldots, m-q, \end{aligned} \tag{44}$$

where $\boldsymbol{\alpha} = \text{Diag}(\boldsymbol{G}_{\text{jnt}}^{(k)})$.

The above problem is not convex. We can make it convex though with the following simple variable transformation:

$$\boldsymbol{\phi} = \frac{1}{\boldsymbol{\xi}}. \tag{45}$$

Now, the problem becomes

$$\begin{aligned} \underset{\boldsymbol{\phi}}{\text{minimize}} \quad & \sum_{i=1}^{m} \left(-\log \phi_i + \alpha_i \frac{1}{\phi_i} + \lambda_{\max}^{(\boldsymbol{S})} \phi_i\right) \\ \text{subject to} \quad & \phi_i \leq \phi_{i+1}, \quad i = 1, \ldots, q-1, \\ & \phi_q \leq \phi_{q+i}, \quad i = 1, \ldots, m-q, \end{aligned} \tag{46}$$

which is in a convex form. Similar to the alternating optimization case, the problem (46) does not have a closed form solution and the computational cost increases significantly in high dimensions. Again, we can find an iterative closed form update of the parameter $\boldsymbol{\alpha}$ that will provide the optimal solution.

We start from the corresponding unconstrained problem whose solution is

$$\boldsymbol{\phi}^{(0)} = \frac{1 + \sqrt{1 + 4\lambda_{\max}^{(\boldsymbol{S})}\boldsymbol{\alpha}^{(0)}}}{2\lambda_{\max}^{(\boldsymbol{S})}}, \tag{47}$$

where $\boldsymbol{\alpha}^{(0)} = \boldsymbol{\alpha}$. We can distinguish the same three cases as for problem (26), where the set $\mathcal{A}$ now is given by

$$\mathcal{A} = \left\{ c_i \,\middle|\, \alpha_{c_i}^{(k)} \leq \frac{1}{r + l - i + 1} \left(\sum_{s=0}^{r} \alpha_{q-s}^{(k)} + \sum_{s=0}^{l-i-1} \alpha_{c_{l-s}}^{(k)}\right) \right\}. \tag{48}$$

The procedure is summarized in Table II. We will refer to it as JOCE$_\phi$ hereafter.

Proposition 7: The iterative closed-form update procedure given in Table II converges to the solution of problem (46).

Proof: The proof of Proposition 7 follows the same steps as the proof of Proposition 3, thus it is omitted. ■

Having obtained the optimal $\boldsymbol{\phi}^\star$, it is easy to retrieve $\boldsymbol{\xi}^\star$ from (45).

The optimization problem for $\boldsymbol{U}$ is the following:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{minimize}} \quad & \text{Tr}\left(\boldsymbol{H}_{\text{jnt}}^{(k)^T}\boldsymbol{U}\right) \\ \text{subject to} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_m. \end{aligned} \tag{49}$$

Proposition 8: The optimal solution of the optimization problem (49) is $\boldsymbol{U}^\star = \boldsymbol{V}_L\boldsymbol{V}_R^T$, where $\boldsymbol{V}_L \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V}_R \in \mathbb{R}^{m\times m}$ are the left and right singular vectors of the matrix $-\boldsymbol{H}_{\text{jnt}}^{(k)}$, respectively.

Proof: The proof comes directly from Lemma 2. ■

In Algorithm 3 we summarize the above iterative procedure. We will refer to it as JOCE.

TABLE II
UPDATES AND OPTIMAL SOLUTION OF THE ITERATIVE PROCEDURE THAT SOLVES THE OPTIMIZATION PROBLEM (46)

	Case 1	Case 2		Case 3
Conditions	$\alpha_i^{(k)} \le \alpha_{i+1}^{(k)}, i \in [1:q-1]$	$\alpha_{j-1}^{(k)} < \alpha_j^{(k)}$ if $j > 1$		$\alpha_{q-r-1}^{(k)} < \alpha_{q-r}^{(k)}$
		$\alpha_i^{(k)} \ge \alpha_{i+1}^{(k)}, \quad i \in [j : j+k-1]$		$\alpha_i^{(k)} \ge \alpha_{i+1}^{(k)}, \quad i \in [q-r : q-1]$
	$\alpha_q^{(k)} \le \alpha_{q+i}^{(k)}, i \in [1:m-q]$	if $j+k < q$	if $j+k = q$	$\alpha_q^{(k)} \ge \alpha_{c_i}^{(k)}, \quad i \in [1:k]$
		$\alpha_{j+k}^{(k)} < \alpha_{j+k+1}^{(k)}$	$\alpha_q^{(k)} < \alpha_{q+i}^{(k)}, i \in [1:m-q]$	$\alpha_q^{(k)} < \alpha_{q+i}^{(k)}, \quad i \in [1:m-q] \setminus \mathcal{C}$
Block Updates	-	$\alpha_i^{(k+1)} = \frac{1}{k+1} \sum_{i=0}^{k} \alpha_{j+i}^{(k)}, \ i \in [j : j+k]$		$\alpha_i^{(k+1)} = \alpha_i^{(k)}, \quad i \in \mathcal{C} \setminus \mathcal{A}$
				$\alpha_i^{(k+1)} = \frac{1}{r+p+1} \sum_{i=0}^{r} \alpha_{q-i}^{(k)} + \sum_{i=1}^{p} \alpha_{a_i}^{(k)}, \ i \in [q-r:q] \bigcup \mathcal{A}$
Solution	$\boldsymbol{\phi}^\star = \frac{1+\sqrt{1+4\lambda_{\max}^{(\boldsymbol{S})} \boldsymbol{\alpha}^{(k)}}}{2\lambda_{\max}^{(\boldsymbol{S})}}$	-		-

Algorithm 3: JOCE - Joint Optimization for Covariance Estimation for the Problem (36).

1: Set $k = 0$, choose $\boldsymbol{U}^{(0)} \in \{\boldsymbol{U} | \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_q\}$
2: **repeat**:
3: Compute $\boldsymbol{\phi}^{(k+1)}$ from Proposition 7
4: Compute $\boldsymbol{H}_{\text{jnt}}^{(k)}$ with (41)
5: Compute $\boldsymbol{V}_L$, $\boldsymbol{V}_R$, the left and right singular vectors of $-\boldsymbol{H}_{\text{jnt}}^{(k)}$, respectively
6: $\boldsymbol{U}^{(k+1)} = \boldsymbol{V}_L \boldsymbol{V}_R^T$
7: $k \leftarrow k+1$
8: **until** convergence
9: Set $\boldsymbol{\xi} = \frac{1}{\boldsymbol{\phi}^{(k)}}$
10: **return** $\boldsymbol{U}^{(k)}, \boldsymbol{\xi}$

V. CONVERGENCE ANALYSIS

In this section we establish the convergence of the proposed algorithms. Our proof hinges on the proofs of SUM and BSUM in [36]. Denote the unitary constraint set $\mathcal{U} \triangleq \{\boldsymbol{U} \in \mathbb{R}^{m \times q} | \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_q\}$, then it can be expressed as

$$\mathcal{U} = \left\{ \boldsymbol{U} \left| \begin{array}{ll} h_i(\boldsymbol{U}) = 0, & \forall i = 1, \ldots, q \\ h_{ij}(\boldsymbol{U}) = 0, & \forall i < j \le q \end{array} \right. \right\}, \tag{50}$$

where $h_i(\boldsymbol{U}) \triangleq \boldsymbol{u}_i^T \boldsymbol{u}_i$, and $h_{ij}(\boldsymbol{U}) \triangleq \boldsymbol{u}_i^T \boldsymbol{u}_j$.

Lemma 3: Linear independence constraint qualification (LICQ) holds everywhere on set $\mathcal{U}$.

Proof: Observe that $\text{vec}(\nabla h_i(\boldsymbol{U})) = [\boldsymbol{0}; 2\boldsymbol{u}_i; \boldsymbol{0}]$ and $\text{vec}(\nabla g_{ij}(\boldsymbol{U})) = [\boldsymbol{0}; \boldsymbol{u}_j; \boldsymbol{0}, \boldsymbol{u}_i; \boldsymbol{0}]$. Partition the gradient vectors into blocks of length m, we can see that $\boldsymbol{u}_i$ appears at the i-th block only in vector $\text{vec}(\nabla h_i(\boldsymbol{U}))$. Consequently, $\text{vec}(\nabla h_i(\boldsymbol{U}))$ cannot be expressed as a linear combination of the rest gradient vectors, since $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q\}$ are linearly independent on set $\mathcal{U}$. ■

Proposition 9: The iterates generated by IMRP converge to the set of KKT points of Problem (6).

Proof: Let $\bar{\boldsymbol{U}}$ be a convergent point of the sequence $(\boldsymbol{U}^{(k)})_{k \in \mathbb{N}}$, following the proof of Theorem 1 in [36] it is not hard to arrive at

$$g\left(\bar{\boldsymbol{U}} | \bar{\boldsymbol{U}}\right) \ge g\left(\boldsymbol{U} | \bar{\boldsymbol{U}}\right), \ \forall \boldsymbol{U} \in \mathcal{U}, \tag{51}$$

meaning $\bar{\boldsymbol{U}}$ is a global mimimizer of the problem

$$\begin{array}{ll} \underset{\boldsymbol{U}}{\text{maximize}} & g\left(\boldsymbol{U} | \bar{\boldsymbol{U}}\right) \\ \text{subject to} & \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_q. \end{array} \tag{52}$$

Since LICQ holds on $\mathcal{U}$ (cf. Lemma 3), $\bar{\boldsymbol{U}}$ satisfies the KKT conditions of Problem (52), i.e.,

$$\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_q, \tag{53}$$

$$\nabla g\left(\bar{\boldsymbol{U}} | \bar{\boldsymbol{U}}\right) - 2\text{Tr}\left(\bar{\boldsymbol{U}} \boldsymbol{\Psi}^T\right) = \boldsymbol{0}, \tag{54}$$

where $\boldsymbol{\Psi}$ is the Lagrange multiplier.

Replacing $\nabla g(\bar{\boldsymbol{U}}, \bar{\boldsymbol{U}})$ in (54) by $\nabla f_{\text{jnt}}(\bar{\boldsymbol{U}})$ we conclude that $\bar{\boldsymbol{U}}$ is a KKT point of Problem (6). Since $\mathcal{U}$ is a compact set, the rest of the proof follows from Corollary 1 in [36]. ■

Note that LICQ also holds on set

$$\mathcal{S} = \left\{ \boldsymbol{\xi} \left| \begin{array}{ll} \boldsymbol{\xi} \ge \boldsymbol{0}, & \\ \xi_i \ge \xi_{i+1}, & i = 1, \ldots, q-1, \\ \xi_q \ge \xi_{q+i}, & i = 1, \ldots, m-q. \end{array} \right. \right\} \tag{55}$$

By the same reasoning, it can be proved that the iterates generated by Algorithm JOCE converges to the set of KKT points of Problem (36).

Next, we show the convergence of AOCE for solving (11), where $\boldsymbol{U}$ and $\boldsymbol{\lambda}$ are updated alternately.

Proposition 10: The iterates generated by Algorithm AOCE converge to the set of KKT points of Problem (11).

Proof: First, it can be verified that the level set $\{(\boldsymbol{\Lambda}, \boldsymbol{U}) | f_{\text{alt}}(\boldsymbol{\Lambda}, \boldsymbol{U}) \le f_{\text{alt}}(\boldsymbol{\Lambda}^{(0)}, \boldsymbol{U}^{(0)})\}$ is compact. Similar as before, it can be shown that LICQ holds on the constraint set. Furthermore, Problem (26) has a unique solution since the objective function is strictly convex.

Let $(\bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}})$ be a limit point of the sequence $(\bar{\boldsymbol{\Lambda}}^{(k)}, \bar{\boldsymbol{U}}^{(k)})_{k \in \mathbb{N}}$, following the proof of Theorem 2(b) in [36], it can be shown that

$$g\left(\bar{\boldsymbol{\Lambda}} | \bar{\boldsymbol{\Lambda}} \bar{\boldsymbol{U}}\right) \le g\left(\boldsymbol{\Lambda} | \bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}}\right), \ \forall \boldsymbol{\Lambda} \in \mathcal{X}, \tag{56}$$

$$g\left(\bar{\boldsymbol{U}} | \bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}}\right) \le g\left(\boldsymbol{U} | \bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}}\right), \ \forall \boldsymbol{U} \in \mathcal{U}, \tag{57}$$

where

$$\mathcal{X} = \left\{ \boldsymbol{\Lambda} \left| \begin{array}{ll} \boldsymbol{\Lambda} \succcurlyeq 0, & \\ \lambda_i \leq \lambda_{i+1}, & i = 1, \ldots, q-1, \\ \lambda_q \leq \lambda_{q+i}, & i = 1, \ldots, m-q. \end{array} \right. \right\} \tag{58}$$

Similar to (50), we express the set $\mathcal{X}$ as $\mathcal{X} = \{\boldsymbol{\Lambda} | \ell_i(\boldsymbol{\Lambda}) \leq 0, i = 1, \ldots, 2m\}$. Then, (56) implies that $\bar{\boldsymbol{\Lambda}}$ satisfies

$$\begin{aligned} 0 \leq \mu_i \perp \ell_i(\boldsymbol{\Lambda}) \leq 0, \forall i = 1, \ldots, 2m, \\ \nabla g\left(\bar{\boldsymbol{\Lambda}} | \bar{\boldsymbol{\Lambda}}, \bar{\mathbf{U}}\right) + \sum_{i=1}^{2m} \mu_i \nabla \ell_i\left(\bar{\boldsymbol{\Lambda}}\right) = 0, \end{aligned} \tag{59}$$

where μ_i is the Lagrange multiplier which corresponds to the constraint $\ell_i(\boldsymbol{\Lambda}) \leq 0$. Eq. (57) implies

$$\begin{aligned} \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_q, \\ \nabla g\left(\bar{\boldsymbol{U}} | \bar{\boldsymbol{U}}\right) + 2\mathrm{Tr}\left(\bar{\boldsymbol{U}} \boldsymbol{\Psi}^T\right) = \mathbf{0}, \end{aligned} \tag{60}$$

where $\boldsymbol{\Psi}$ is the Lagrange multiplier.

Since $\nabla g(\bar{\boldsymbol{\Lambda}} | \bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}}) = \nabla_{\boldsymbol{\Lambda}} f_{\text{alt}}(\bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}})$ and $\nabla g(\bar{\boldsymbol{U}} | \bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}}) = \nabla_{\boldsymbol{U}} f_{\text{alt}}(\bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}})$, putting together (59) and (60) reveals that $(\bar{\boldsymbol{\Lambda}}, \bar{\boldsymbol{U}})$ is a KKT point of the original problem (11). ■

VI. Acceleration and Parameter Selection

A. Acceleration

The derivation of all the proposed algorithms is based on the minorization-majorization framework. In order to obtain surrogate functions that can be easily solved in closed-form we need to minorize twice many terms of the original functions. This can possibly lead to loose bounds which is associated with slow convergence. Thus, in this section we describe an acceleration scheme, called SQUAREM, that can improve significantly the convergence speed of the proposed algorithms.

SQUAREM was originally proposed in [37] to accelerate EM algorithms. Since MM is a generalization of EM and the update rule of MM is just a fixed-point iteration like EM, we can easily apply the SQUAREM acceleration method to MM algorithms with minor modifications.

We denote by $F_{\text{IMRP}}(\cdot)$ the fixed-point iteration map of the IMRP algorithm, i.e., $\boldsymbol{U}^{(k+1)} = F_{\text{IMRP}}(\boldsymbol{U}^{(k)})$, and by $\text{IMRP}(\boldsymbol{U}^{(k)})$ the value of the objective function of (6) at the k-th iteration. The general SQUAREM method can cause two possible problems to the MM algorithms. First, the updated point may violate the constraints. To solve this issue we can project to the feasible set which in our case is the Stiefel manifold. The projection is just a Procrustes problem with a closed form solution. The second problem is that the acceleration may violate the ascend property of the MM algorithm. For this reason, a backtracking step is adopted halving the distance of the step-length γ and -1. As $\gamma \to -1$, $\text{IMRP}(\boldsymbol{U}^{(k+1)}) \geq \text{IMRP}(\boldsymbol{U}^{(k)})$ is guaranteed to hold. The accelerated IMRP is summarized in Algorithm 4.

We have presented the accelerated version only for the IMRP algorithm. A similar technique may also be used for the AOCE and JOCE algorithms. In particular, we can replace the part that correspond to the eigenvector estimation with acceleration steps similar to the ones in Algorithm 4. This procedure is straightforward and therefore the details are omitted. In the rest of the paper, when we refer to an algorithm we will mean the accelerated version.

Algorithm 4: Accelerated IMRP.

Set $k = 0$, choose $\boldsymbol{U}^{(0)} \in \{\boldsymbol{U} | \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_q\}$
repeat:
  $\boldsymbol{U}_1 = F_{\text{IMRP}}(\boldsymbol{U}^{(k)})$
  $\boldsymbol{U}_2 = F_{\text{IMRP}}(\boldsymbol{U}_1)$
  $\boldsymbol{R} = \boldsymbol{U}_1 - \boldsymbol{U}^{(k)}$
  $\boldsymbol{V} = \boldsymbol{U}_2 - \boldsymbol{U}_1 - \boldsymbol{R}$
  Compute the step-length $\gamma = -\frac{\|\boldsymbol{R}\|_F}{\|\boldsymbol{V}\|_F}$
  $\boldsymbol{U} = \boldsymbol{U}^{(k)} - 2\gamma \boldsymbol{R} + \gamma^2 \boldsymbol{V}$
  $\boldsymbol{U} = \boldsymbol{U}_l \boldsymbol{U}_r^T$, where $\boldsymbol{U}_l, \boldsymbol{U}_r$ are the left and right singular vectors of $\boldsymbol{U}$, respectively (projection)
  while $\text{IMRP}(\boldsymbol{U}) < \text{IMRP}(\boldsymbol{U}^{(k)})$
    $\gamma \leftarrow (\gamma - 1)/2$
    $\boldsymbol{U} = \boldsymbol{U}^{(k)} - 2\gamma \boldsymbol{R} + \gamma^2 \boldsymbol{V}$
    $\boldsymbol{U} = \boldsymbol{U}_l \boldsymbol{U}_r^T$ (projection)
  end while
  $\boldsymbol{U}^{(k+1)} = \boldsymbol{U}$
  $k \leftarrow k + 1$
until convergence
return $\boldsymbol{U}^{(k)}$

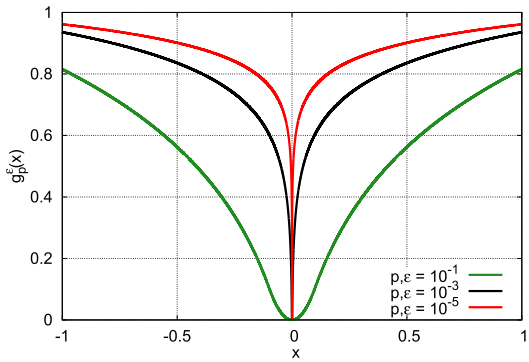


Fig. 2. Approximation of the ℓ_0 norm via the $g_p^\epsilon(\cdot)$ function for different values of p, ϵ.

B. ℓ_0 Approximation Parameters

In all proposed algorithms we have used the function $g_p^\epsilon(\cdot)$ given in (5) as a smooth proxy of the ℓ_0 norm. The smaller the parameters p, ϵ are, the better the approximation of the ℓ_0 norm is. This can be seen in Fig. 2, where we have depicted the function $g_p^\epsilon(\cdot)$ for decreasing values of p, ϵ.

In practice, as $p, \epsilon \to 0$, it is likely that the algorithm will get stuck in an undesirable local minimum [15], [29]. A good strategy (that we adopt) is a sequentially decreasing scheme, i.e., first solve the problem for large (and fixed) values of p, ϵ and then decrease their values and solve the problem again using the previous solution as an initial point.

C. Sparsity Inducing Parameters

Every eigenvector is associated with a different sparsity inducing parameter ρ_i. Following [11], we set the range of these parameters to $[0, \rho_w^i \|\boldsymbol{C}\|_{2,1}^2]$. Here, $\|\boldsymbol{C}\|_{2,1}^2$ is the operator norm induced by $\|\cdot\|_2$ and $\|\cdot\|_1$ that is equivalent to the maximum ℓ_2 norm of the columns of the data matrix $\boldsymbol{C}$. The specific weight of each ρ_i is defined as $\rho_w^i = (\lambda_i d_i)/(\lambda_1 d_1)$, where λ_i is the i-th largest eigenvalue of the sample covariance matrix $\boldsymbol{S}$ and d_i is the i-th diagonal element of matrix $\boldsymbol{D}$.

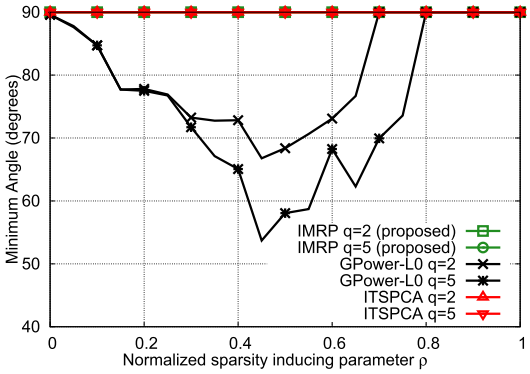


Fig. 3. Minimum angle vs normalized regularization parameter.

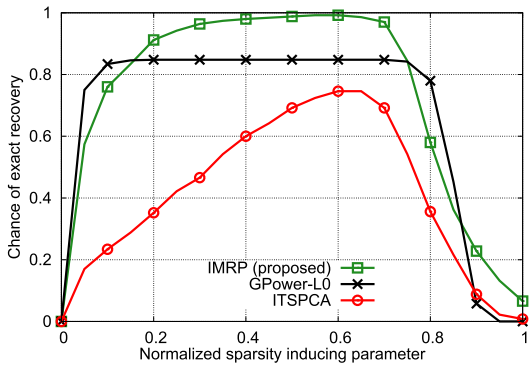


Fig. 4. Chance of exact recovery vs normalized regularization parameter.

VII. Numerical Experiments

All the experiments were performed on a PC with a 3.20 GHz i5-4570 CPU and 8 GB RAM.

A. Random Data Drawn from a Sparse PCA Model

In the first experiment we compare the performance of the proposed IMRP algorithm with the benchmark algorithms GPower$_{\ell_0}$ proposed in [11] and ITSPCA proposed in [14]. Note that all four GPower algorithms that are proposed in [11] have very similar performance in terms of chance of recovery and percentage of explained variance. Thus, it is sufficient to consider only one of them.

We first examine the orthogonality of the estimated sparse eigenvectors. We define the angle between eigenvectors i, j as:

$$\theta_{ij} = \min\left(|\arccos\left(\boldsymbol{v}_i^T \boldsymbol{v}_j\right)|, 180^o - |\arccos\left(\boldsymbol{v}_i^T \boldsymbol{v}_j\right)|\right). \quad (61)$$

We consider a setup with dimension $m = 500$ and $n = 50$ samples. We construct 100 covariance matrices $\boldsymbol{\Sigma}$ through their eigenvalue decomposition $\boldsymbol{\Sigma} = \boldsymbol{V}\text{diag}(\boldsymbol{\lambda})\boldsymbol{V}^T$, where the first $k = 5$ columns of $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ are of the following form:

$$\begin{cases} v_{ij} \neq 0, & \text{for } i = 1,\ldots,10,\ j = 1,\ldots,5, \\ v_{ij} = 0, & \text{otherwise,} \end{cases} \quad (62)$$

where the non-zero values are such that the eigenvectors are orthonormal. The remaining eigenvectors are generated randomly, satisfying the orthogonality property. The eigenvalues are set to be $\lambda_i = 100(k - i + 1)$ for $i = 1,\ldots,5$, and the rest are set to one.

For each of the covariance matrix $\boldsymbol{\Sigma}$, we randomly generate 50 data matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ by drawing n samples from a zero-mean normal distribution with covariance matrix $\boldsymbol{\Sigma}$, i.e., i.e., $\boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, for $i = 1,\ldots,n$. Then we employ the two algorithms to compute the first two and the first five sparse eigenvectors. In Fig. 3, the minimum angle between any two eigenvectors, i.e., $\min_{i,j}(\theta_{i,j})$ for a wide range of the regularization parameter ρ is depicted. It is clear that the proposed IMRP algorithm (after thresholding) and ITSPCA are orthogonal[1] for any choice of ρ, while for the GPower$_{\ell_0}$ algorithm the are cases that the estimated eigenvectors have angle less than 55^o. For large values of ρ, GPower$_{\ell_0}$ gives orthogonal results since the sparsity level is high and the estimated eigenvectors do not have overlapping support.

Now, to illustrate the sparse recovering performance of our algorithm we generate synthetic data as in [11], [13], [15]. To this end, we construct a covariance matrix $\boldsymbol{\Sigma}$ through the eigenvalue decomposition $\boldsymbol{\Sigma} = \boldsymbol{V}\text{diag}(\boldsymbol{\lambda})\boldsymbol{V}^T$, where the first q columns of $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ have a pre-specified sparse structure. We consider a setup with $m = 500$, $n = 50$ and $q = 2$. We set the first two orthonormal eigenvectors to be

$$\begin{cases} v_{i1} = \frac{1}{\sqrt{10}}, & \text{for } i = 1,\ldots,10, \\ v_{i1} = 0, & \text{otherwise,} \end{cases} \quad \begin{cases} v_{i2} = \frac{1}{\sqrt{10}}, & \text{for } i = 11,\ldots,20, \\ v_{i2} = 0, & \text{otherwise.} \end{cases} \quad (63)$$

The remaining eigenvectors are generated randomly, satisfying the orthogonality property. We set the eigenvalues to be $\lambda_1 = 400$, $\lambda_2 = 300$ and $\lambda_i = 1$ for $i = 3,\ldots,500$. The parameters of the IMRP algorithm are chosen according to Section VI with $\boldsymbol{d} = [1, 0.5]$.

We randomly generate 500 data matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ by drawing n samples from a zero-mean normal distribution with covariance matrix $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, for $i = 1,\ldots,n$. Then, we employ the two algorithms to compute the two leading sparse eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{R}^{500}$. We consider a successful recovery when both quantities $|\boldsymbol{u}_1^T \boldsymbol{v}_1|$ and $|\boldsymbol{u}_2^T \boldsymbol{v}_2|$ are greater than 0.99.

The chance of successful recovery over a wide range of the regularization parameters ρ_i is plotted in Fig. 4. It is clear that the proposed IMRP algorithm achieves a significantly higher chance of exact recovery for a wide range of the parameters, while the ITSPCA algorithm that also preserves orthogonality exhibits a degraded performance.

B. Gene Expression Data

In this subsection we compare the performance of the two algorithms on the gene expression dataset collected in the breast cancer study by Bild *et al.* [38]. The dataset contains 158 samples over 12,625 genes. We consider the 2,000 genes with the largest variances and we estimate the first 5 eigenvectors.

Notice that due to the orthogonality constraints, increasing the cardinality does not necessarily mean that the CPEV will increase. To this end, for a fixed cardinality, we depict the maximum variance being explained from the sparse eigenvectors up to this cardinality. Thus, the CPEV for cardinality i, denoted as

[1]Orthogonality in the sense that $|\boldsymbol{u}_i^T \boldsymbol{u}_j| \leq \epsilon$, with $i \neq j$, where in the worst case ϵ is in the order of the selected threshold t. For example, for $t = 10^{-12}$, the inner product $|\boldsymbol{u}_i^T \boldsymbol{u}_j|$ is effectively zero for all practical purposes.

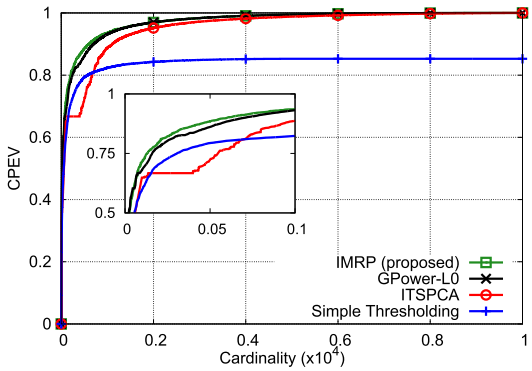


Fig. 5. CPEV vs cardinality.

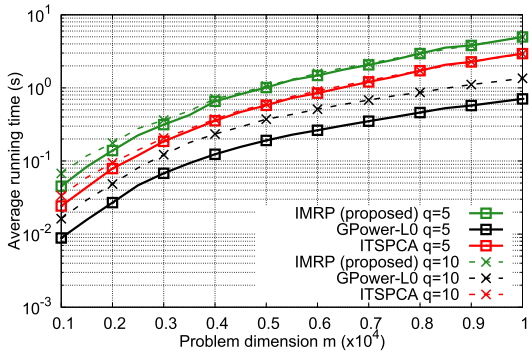


Fig. 6. Average running time of IMRP, GPower$_{\ell 0}$ and ITSPCA with increasing dimension m. We estimate $q = 5$ and $q = 10$ eigenvectors.

CPEV$_i$, is being post-processed as follows:

$$\text{CPEV}_i = \max(\text{CPEV}_i, \text{CPEV}_{i-1}). \tag{64}$$

In Fig. 5 we illustrate the cumulative percentage of explained variance, computed by Eq. (25) and post-processed by (64), versus the total cardinality of the estimated eigenvectors for the IMRP, ITSPCA and GPower$_{\ell_0}$ algorithms. For maximum cardinality the percentage of explained variance becomes 1 for all algorithms. For small cardinalities, IMRP dominates all the other algorithms, while ITSPCA performs worse than IMRP and GPrower$_{\ell 0}$ in most of the range of cardinalities. For comparison we have also included the simple thresholding scheme which first computes the regular principal component and then keeps a required number of entries with the largest absolute values.

C. Computational Complexity

In this experiment we compare the computational complexity of IMRP, GPower$_{\ell 0}$ and ITSPCA algorithms. We consider the setup of Section VII-A where we create a covariance matrix through its eigenvalue decomposition. We predefine the first $k = 10$ eigenvectors in a similar manner as in (63), while we set the eigenvalues to be $\lambda_i = 100(k - i + 1)$ for $i = 1, \ldots, 10$, and we fix the rest to one. For a given dimension m we randomly generate 100 data matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, where $n = 0.1\, m$. The sparsity inducing parameters of each algorithm are chosen such that the solutions of all the algorithms exhibit similar cardinalities.

The average running time over the 100 independent trials for each dimension m is shown in Fig. 6. We considered the estimation of the first $q = 5$ and $q = 10$ eigenvectors of the above model. We observe that although the GPower$_{\ell 0}$ is faster than IMRP and ITSPCA, it is significantly affected by the number of estimated eigenvectors q. On the other hand, IMRP and ITSPCA seem to be affected by q only in small dimensions. In practice, although slightly slower, IMRP can be used for very high dimensional problems, especially since it increases the estimation performance. For example, as shown in Fig. 6, for dimension $m = 10^4$ the average running time of IMRP is less that 5 seconds when simulated on our computing system.

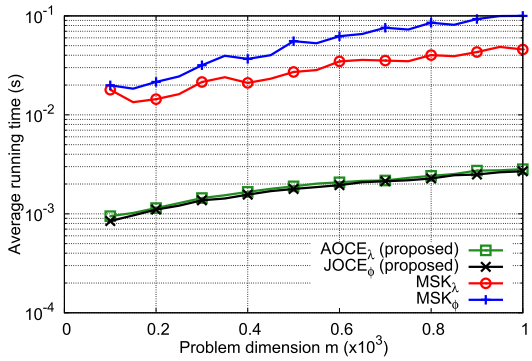


Fig. 7. Average running time of AOCE$_\lambda$, JOCE$_\phi$, MSK$_\lambda$ and MSK$_\phi$ with increasing dimension m.

D. Covariance Estimation

The main idea behind the AOCE and JOCE algorithms presented in Section IV is to estimate a covariance matrix by estimating its eigenvalues and eigenvectors. Consequently, one step of these algorithms corresponds to the eigenvalue estimation problem which involves the optimization of a separable convex objective function with ordering constraints. Although these problems are convex and can be solved directly with a solver, we presented two iterative algorithms, AOCE$_\lambda$ and JOCE$_\phi$, with a closed-form update. In this section we examine the benefit of these algorithms in terms of average running time compared to the MOSEK solver.

For a given dimension m we randomly generate 500 parameters of each optimization problem, i.e., $\boldsymbol{z}$ for AOCE$_\lambda$ and $\boldsymbol{\alpha}$, $\lambda_{\max}^{(\boldsymbol{S})}$ for JOCE$_\phi$. Each parameter z_i, α_i, for $i = 1, \ldots, m$, is generated based on a uniform distribution in the interval $(0, 1)$, while $\lambda_{\max}^{(\boldsymbol{S})}$ is uniformly distributed in the interval $(100, 200)$. Fig. 7 illustrates the average running time of the algorithms AOCE$_\lambda$, JOCE$_\phi$ and the corresponding MOSEK implementations of the problems (26) and (46), denoted as MSK$_\lambda$ and MSK$_\phi$, respectively. It is clear that the proposed algorithms are more than one order of magnitude faster.

Now, we examine the covariance estimation performance of the proposed AOCE and JOCE algorithms. In this experiment we consider again the setting and data generation process of Section VII-A, with the only difference that we reduce the dimension to $m = 200$. We compute the relative mean square error (RelMSE) for each algorithm, defined as

$$\text{RelMSE}(\hat{\boldsymbol{S}}) = 1 - \frac{\text{MSE}(\hat{\boldsymbol{S}})}{\text{MSE}\,(\boldsymbol{S})}, \tag{65}$$

where $\text{MSE}(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{\Sigma}\|_F^2$, while $\hat{\boldsymbol{S}}$ is the estimated covariance matrix from the two algorithms and $\boldsymbol{S}$ is the sample covariance matrix.

In Fig. 8 we observe that AOCE outperforms JOCE when a low number of samples is available, while after one point the

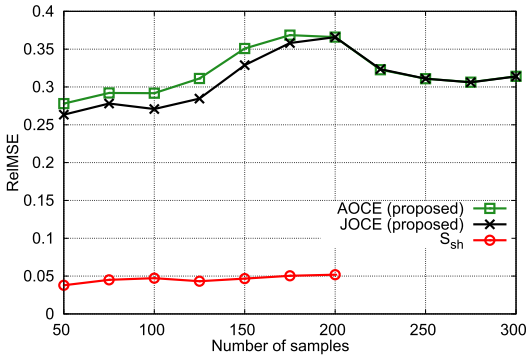


Fig. 8. RelMSE vs number of samples.

two algorithms have the same performance. Both the algorithms improve significantly the estimation of the covariance matrix. For example, when $n = m$, the improvement is approximately 35%. For $n \leq m$, instead of $\boldsymbol{S}$ we use $\boldsymbol{S}_{\text{sh}}$ as defined in (12). The parameter δ is chosen based on a grid search. For this case, in order to show that the improvement in estimation is not due to shrinkage, we include the RMSE for $\boldsymbol{S}_{\text{sh}}$. It is clear from the plot that the improvement from shrinkage is approximately 5%. This explains the slight estimation improvement of AOCE and JOCE for $n \leq m$.

VIII. Conclusions

In this paper, we proposed a new algorithm (IMRP) for the sparse eigenvector extraction problem. The algorithm is derived based on the minorization-majorization method that was applied after a smooth approximation of the sparsity inducing ℓ_0-norm. Unlike the vast majority of the state of the art methods, the sparse eigenvectors obtained by our proposed method maintain their orthogonality property. Further, we consider the problem of covariance estimation where the underlying structure of its eigenvectors is sparse. We formed a covariance estimation problem using the eigenvalue decomposition and we imposed sparsity on some of the principal eigenvectors to improve the estimation performance. We have proposed two algorithms (AOCE and JOCE) based on the MM framework to efficiently solve the above problem. Numerical experiments have shown that IMRP outperforms existing algorithms while AOCE and JOCE improve significantly the estimation of the covariance matrix when its eigenvectors have a sparse structure.

Appendix A
Proof of Lemma 1

Proof: Following the same approach as [15], we can bound the function $\sum_{i=1}^{q} \rho_i \sum_{j=1}^{m} g_p^{\epsilon}(u_{ij})$ with a weighted quadratic one. Based on the results of [15] and by incorporating the sparsity parameters ρ_i to the corresponding weights, it holds that

$$\sum_{i=1}^{q} \rho_i \sum_{j=1}^{m} g_p^{\epsilon}\left(u_{ij}\right) \leq \text{vec}(\boldsymbol{U})^T \,\text{diag}\left(\boldsymbol{w}\right) \text{vec}(\boldsymbol{U}),$$

with the weights $\boldsymbol{w} \in \mathbb{R}_+^{mq}$ given by (16). Now, the idea is to create a concave term and linearize it since the linear approximation of a concave function is an upper bound of the function. We define $\boldsymbol{w}_{\max} \in \mathbb{R}_+^{q}$, with $w_{\max,i}$ being the maximum weight that corresponds to the i-th eigenvector. For convenience we further define $\boldsymbol{u} = \text{vec}(\boldsymbol{U})$, $\boldsymbol{W}_d = \text{diag}(\boldsymbol{w})$ and $\boldsymbol{W}_m = \text{diag}(\boldsymbol{w}_{\max} \otimes \boldsymbol{1}_m)$. Now, we can bound the weighted quadratic function as follows:

$$\begin{aligned}
\boldsymbol{u}^T \boldsymbol{W}_d \boldsymbol{u} &= \boldsymbol{u}^T \left(\boldsymbol{W}_d - \boldsymbol{W}_m\right) \boldsymbol{u} + \boldsymbol{u}^T \boldsymbol{W}_m \boldsymbol{u} \\
&= \boldsymbol{u}^T \left(\boldsymbol{W}_d - \boldsymbol{W}_m\right) \boldsymbol{u} + \boldsymbol{1}_m^T \boldsymbol{w}_{\max} \\
&\leq \boldsymbol{u}_0^T \left(\boldsymbol{W}_d - \boldsymbol{W}_m\right) \boldsymbol{u}_0 + 2\boldsymbol{u}_0^T \left(\boldsymbol{W}_d - \boldsymbol{W}_m\right) \\
&\quad \times \left(\boldsymbol{u} - \boldsymbol{u}_0\right) + \boldsymbol{1}_m^T \boldsymbol{w}_{\max} \\
&= 2\boldsymbol{u}_0^T \left(\boldsymbol{W}_d - \boldsymbol{W}_m\right) \boldsymbol{u} - \boldsymbol{u}_0^T \boldsymbol{W}_d \boldsymbol{u}_0 \\
&\quad + 2\left(\boldsymbol{1}_m^T \boldsymbol{w}_{\max}\right) \\
&= 2\text{Tr}\left(\boldsymbol{H}^T \boldsymbol{U}\right) + 2\left(\boldsymbol{1}_m^T \boldsymbol{w}_{\max}\right) - \boldsymbol{u}_0^T \boldsymbol{W}_d \boldsymbol{u}_0,
\end{aligned}$$

where $\boldsymbol{H} = [(\boldsymbol{W}_d - \boldsymbol{W}_m)\boldsymbol{u}_0]_{m \times q}$. ■

Appendix B
Proof of Proposition 3

Proof: For convenience, in all the proofs we drop the superscript (k) that denotes the current iteration. We denote the updates of $\boldsymbol{z}$ by $\bar{\boldsymbol{z}}$, i.e., if $\boldsymbol{z} = \boldsymbol{z}^{(k)}$ then $\bar{\boldsymbol{z}} = \boldsymbol{z}^{(k+1)}$. The Lagrangian of the optimization problem (26) is

$$\begin{aligned}
L(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}) = &-\sum_{i=1}^{m} \log \lambda_i + \sum_{i=1}^{m} z_i \lambda_i + \sum_{i=1}^{q-1} \mu_i(\lambda_i - \lambda_{i+1}) \\
&+ \sum_{i=1}^{m-q} \nu_{q+i}(\lambda_q - \lambda_{q+i}),
\end{aligned} \tag{66}$$

with $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, $\boldsymbol{\mu} \in \mathbb{R}_+^{q-1}$ and $\boldsymbol{\nu} \in \mathbb{R}_+^{m-q}$. Now, we can derive the following Karush-Kuhn-Tucker (KKT) conditions [39]:

$$-\frac{1}{\lambda_1} + z_1 + \mu_1 = 0, \tag{67}$$

$$-\frac{1}{\lambda_i} + z_i + \mu_i - \mu_{i-1} = 0, \quad i = 2, \ldots, q-1, \tag{68}$$

$$-\frac{1}{\lambda_q} + z_q - \mu_{q-1} + \sum_{i=1}^{m-q} \nu_i = 0, \tag{69}$$

$$-\frac{1}{\lambda_{q+i}} + z_{q+i} - \nu_{q+i} = 0, \quad i = 1, \ldots, m-q, \tag{70}$$

$$\lambda_i - \lambda_{i+1} \leq 0, \quad i = 1, \ldots, q-1, \tag{71}$$

$$\lambda_q - \lambda_{q+i} \leq 0, \quad i = 1, \ldots, m-q, \tag{72}$$

$$\mu_i \geq 0, \quad i = 1, \ldots, q-1, \tag{73}$$

$$\nu_{q+i} \geq 0, \quad i = 1, \ldots, m-q, \tag{74}$$

$$\mu_i(\lambda_i - \lambda_{i+1}) = 0, \quad i = 1, \ldots, q-1, \tag{75}$$

$$\nu_{q+i}(\lambda_q - \lambda_{q+i}) = 0, \quad i = 1, \ldots, m-q. \tag{76}$$

As a first result we can state the following lemma:

Lemma 4: The solution of the KKT system (67)-(76) is $\lambda_i = \frac{1}{z_i}$, for $i = 1, \ldots, m$, if the following conditions hold:

$$z_i \geq z_{i+1}, \quad i = 1, \ldots, q-1, \tag{77}$$

$$z_q \geq z_{q+i}, \quad i = 1, \ldots, m-q. \tag{78}$$

In this case all the Lagrange multipliers are zero.

Proof: It is straightforward that if inequalities (77) and (78) hold, then the solutions of the primal and dual variables given in the above lemma satisfy all equations. Since the problem is convex, this solution is the optimal. ■

We can interpret Lemma 4 as follows: if the unconstrained problem has an optimal solution that is inside the feasible region of the constrained problem, then it is also the optimal solution of the constrained problem. Now, if the conditions of Lemma 4 do not hold, the solution of the unconstrained problem will violate a set of inequality constraints. We can distinguish two different types of violations.

(a) Violations in the first q eigenvalues: Here, we consider the case where the solution of the unconstrained problem violates the ordering constraints of the first q eigenvalues (Case 2 of Table I). In this case, we need to update the parameters $\boldsymbol{z}$ according to the following Lemma:

Lemma 5: For any block of r consecutive inequality violations between the first q eigenvalues, i.e., $\forall j, r$, with $j + r \leq q$, that the following conditions hold

$$z_{j-1} > z_j, \quad \text{if } j > 1, \tag{79}$$

$$z_i \leq z_{i+1}, \quad i = j, \ldots, j + r - 1, \tag{80}$$

$$\begin{cases} z_{j+r} > z_{j+r+1}, & \text{if } j + r < q, \\ z_q > z_{q+i}, \quad i = 1, \ldots, m - q, & \text{if } j + r = q, \end{cases} \tag{81}$$

where at least one inequality of (80) is strict, the update of the corresponding block of $\boldsymbol{z}$ is

$$\bar{z}_i = \frac{1}{r+1} \sum_{s=0}^{r} z_{j+s}, \quad i = j, \ldots, j + r. \tag{82}$$

The new KKT system with the updated parameters has the same solution as the original one.

Proof: See Appendix C. ■

(b) Violations including a set of the last $m - q$ eigenvalues: Since we do not impose ordering on the $m - q$ last eigenvalues, any of them could violate the inequality with λ_q and not only the neighboring ones. Thus, we use the indices $c_1, \ldots, c_k$, with $c_i > q$, for $i = 1, \ldots, l$, and $c_i \in \mathcal{C}$, with $\mathcal{C}$ the set of indices of the eigenvalues that violate the inequality constraints with λ_q. We further denote by $\mathcal{A} \subseteq \mathcal{C}$ the set of indices of the active dual variables $\boldsymbol{\nu}$, i.e., $a_i \in \mathcal{A}$ if $\nu_{a_i} > 0$. We assume that $\text{card}(\mathcal{A}) = p \leq l$. For this type of violations (Case 3 of Table I), the solution is given from the following lemma:

Lemma 6: For any block of $r + l$ consecutive inequality violations between the last $r + 1$ ordered and a set of l unordered eigenvalues, i.e., $\forall r, l$, that the following conditions hold

$$z_{q-r-1} > z_{q-r}, \tag{83}$$

$$z_{q-i} \leq z_{q-i+1}, \quad i = 1, \ldots, r, \tag{84}$$

$$z_q \leq z_{c_i}, \quad i = 1, \ldots, l, \tag{85}$$

$$z_q > z_i, \quad i \in [q + 1 : m] \setminus \mathcal{C}, \tag{86}$$

where at least one inequality of (85) is strict, the update of the corresponding block of $\boldsymbol{z}$ is

$$\begin{cases} \bar{z}_i = \frac{1}{r+p+1} \left(\sum_{s=0}^{r} z_{q-s} + \sum_{s=1}^{p} z_{a_s} \right), & i \in [q - r : q] \bigcup \mathcal{A}, \\ \bar{z}_i = z_i, & i \in \mathcal{C} \setminus \mathcal{A}. \end{cases} \tag{87}$$

The set $\mathcal{A}$ is given by

$$\mathcal{A} = \left\{ c_i \middle| z_{c_i} \geq \frac{1}{r + l - i + 1} \left(\sum_{s=0}^{r} z_{q-s} + \sum_{s=0}^{l-i-1} z_{c_{l-s}} \right) \right\}. \tag{88}$$

The new KKT system with the updated parameters has the same solution as the original one.

Proof: See Appendix D. ■

After applying Lemma 5 and/or 6, the new KKT system, apart from equivalent to the original, it further has the exact same form. Thus, we can apply Lemmas 4-6 to the updated system of equations, until we obtain the optimal solution. Since, the original KKT system has m primal and $m - 1$ dual variables and in every iteration we effectively remove at least one primal and one dual variable (see Appendix D), we need at most $m - 1$ iterations. However, it is easy to see that the algorithm finishes in at most $\min(2q - 1, m - 1)$ steps. ■

Appendix C
Proof of Lemma 5

Proof: First, we will prove that when an inequality is violated, then the corresponding eigenvalues become equal. Assume that $z_k < z_{k+1}$, with $j \leq k < k + 1 \leq j + r$. The KKT conditions for this pair are:

$$-\frac{1}{\lambda_k} + z_k + \mu_k - \mu_{k-1} = 0, \tag{89}$$

$$-\frac{1}{\lambda_{k+1}} + z_{k+1} + \mu_{k+1} - \mu_k = 0, \tag{90}$$

$$\lambda_k - \lambda_{k+1} \leq 0, \tag{91}$$

$$\mu_k \geq 0, \tag{92}$$

$$\mu_k(\lambda_k - \lambda_{k+1}) = 0. \tag{93}$$

If we subtract the first two equations we get:

$$2\mu_k = z_{k+1} - z_k + \frac{1}{\lambda_k} - \frac{1}{\lambda_{k+1}} + \mu_{k+1} + \mu_{k-1}. \tag{94}$$

The right hand side of the above equation is strictly positive since $z_{k+1} - z_k > 0$, $\frac{1}{\lambda_k} - \frac{1}{\lambda_{k+1}} \geq 0$ and $\mu_{k+1}, \mu_{k-1} \geq 0$. Thus, $\mu_k > 0$ and from (93) it holds that $\lambda_k = \lambda_{k+1}$. In a similar manner, and using that $\mu_k > 0$, it is easy to prove that $\mu_i > 0$, with $i = j, \ldots, j + r - 1$, which means that $\lambda_j = \cdots = \lambda_{j+r}$.

Having proved the equality of the eigenvalues and that $\boldsymbol{\mu}_{[j:j+r-1]} > \mathbf{0}$, it is straightforward that the primal feasibility, dual feasibility and complementary slackness are trivially satisfied for this block. Further, the $r + 1$ equations of the partial derivative of the Lagrangian reduce to

$$-\frac{1}{\lambda_i} + \bar{z}_i + \frac{1}{r+1}(\mu_{j+r} - \mu_{j-1}) = 0, \quad i = j, \ldots, j + r, \tag{95}$$

with $\bar{z}_i$ given by (82). We can treat (95) as only one equation since it is repeated $r+1$ times. Effectively, we have removed r primal and r dual variables. It is clear that every solution of the reduced set of KKT conditions, is a solution for the original set of KKT conditions. ■

APPENDIX D
PROOF OF LEMMA 6

Proof: We write the KKT conditions for the corresponding block in the following form:

$$-\frac{1}{\lambda_i}+z_i+\mu_i-\mu_{i-1} = 0,$$
$$i = q-r,\ldots,q-1, \quad (96)$$
$$-\frac{1}{\lambda_q}+z_q-\mu_{q-1}+\sum_{i=1}^{m-q}\nu_{q+i} = 0, \quad (97)$$
$$-\frac{1}{\lambda_{a_i}}+z_{a_i}-\nu_{a_i} = 0, \quad a_i\in\mathcal{A}, \quad (98)$$
$$-\frac{1}{\lambda_{d_i}}+z_{d_i}-\nu_{d_i} = 0, \quad d_i\in\mathcal{C}\setminus\mathcal{A}, \quad (99)$$
$$\lambda_i-\lambda_{i+1} \le 0,$$
$$i = q-r,\ldots,q-1, \quad (100)$$
$$\lambda_q-\lambda_{a_i} \le 0, \quad a_i\in\mathcal{A} \quad (101)$$
$$\lambda_q-\lambda_{d_i} \le 0, \quad d_i\in\mathcal{C}\setminus\mathcal{A}, \quad (102)$$
$$\mu_i \ge 0,$$
$$i = q-r,\ldots,q-1, \quad (103)$$
$$\nu_{q+a_i} \ge 0, \quad a_i\in\mathcal{A}, \quad (104)$$
$$\nu_{q+d_i} \ge 0, \quad d_i\in\mathcal{C}\setminus\mathcal{A}, \quad (105)$$
$$\mu_i(\lambda_i-\lambda_{i+1}) = 0,$$
$$i = q-r,\ldots,q-1, \quad (106)$$
$$\nu_{a_i}(\lambda_q-\lambda_{a_i}) = 0, \quad a_i\in\mathcal{A}, \quad (107)$$
$$\nu_{d_i}(\lambda_q-\lambda_{d_i}) = 0, \quad a_i\in\mathcal{C}\setminus\mathcal{A}. \quad (108)$$

As in the proof of Lemma 5, it is easy to show that $\mu_i > 0$, for $i=q-r,\ldots,q-1$. This means that $\lambda_{q-r}=\cdots=\lambda_q$. Further, assuming that we know the set $\mathcal{A}$, since $\nu_{a_i}>0$, from complementary slackness we get that $\lambda_q=\lambda_{a_i}, \forall a_i\in\mathcal{A}$.

Again, having proved the equality of the eigenvalues, and that $\boldsymbol{\mu}_{[q-r:q-1]}>\mathbf{0}$, $\nu_{a_i}>0$ for $a_i\in\mathcal{A}$, it is straightforward that equations (100), (101), (103), (104), (106) and (107) are trivially satisfied.

The equations (96)-(99) reduce to

$$-\frac{1}{\lambda_i}+\bar{z}_i+\frac{1}{r+p+1}\left(\sum_{d_i\in\mathcal{C}\setminus\mathcal{A}}\nu_{d_i}-\mu_{q-r-1}\right)=0, \quad (109)$$

for $i\in[q-r:q]\cup\mathcal{A}$ and

$$-\frac{1}{\lambda_{d_i}}+z_{d_i}-\nu_{d_i}=0, \quad (110)$$

for $d_i\in\mathcal{C}\setminus\mathcal{A}$, where $\bar{z}_i$ is given by (87). Assuming that $\text{card}(\mathcal{A})=p$, we can treat (109) as only one equation since it is repeated $r+p+1$ times. Effectively, we have removed $r+p$ primal and $r+p$ dual variables. It is clear that every solution of the reduced set of KKT conditions, is a solution for the original set of KKT conditions.

Now, we will prove that the indices of the active dual variables $\boldsymbol{\nu}$ for this iteration are given by (88).

We consider the case where $z_q\le z_{c_1}\le\cdots\le z_{c_l}$, where at least one inequality is strict. We assume that we know the active set of this and any further iteration. First, we will prove by contradiction that $c_l\in\mathcal{A}$.

Assume that $c_l\notin\mathcal{A}$. Since $\bar{z}_q$ will be the average of z_i's that are less or equal to z_{c_l}, with at least one z_i strictly smaller, it holds that $\bar{z}_q<z_{c_l}$. Now, by adding (97) and (98), and subtracting the partial derivative of the Lagrangian corresponding to c_l, we get:

$$-\frac{1}{\lambda_q}+\frac{1}{\lambda_{c_l}}+\bar{z}_q-z_{c_l}-\mu_{q-r-1}=0. \quad (111)$$

The last equation implies that $\mu_{q-r-1}<0$ should hold which is not valid. Thus, $c_l\in\mathcal{A}$ holds.

Having proved that $c_l\in\mathcal{A}$ and that $\boldsymbol{\mu}_{[q-r:q-1]}>\mathbf{0}$, if the average of $z_{[q-r:q-1]}$ and z_{c_l} is less or equal to $z_{c_{l-1}}$, following the same arguments we can show that $c_{l-1}\in\mathcal{A}$. Generalizing this result, $c_i\in\mathcal{A}$ if the following condition is true:

$$z_{c_i}\ge\frac{1}{r+l-i+1}\left(\sum_{s=0}^{r}z_{q-s}+\sum_{s=0}^{l-i-1}z_{c_{l-s}}\right) \quad (112)$$

Assuming that $\text{card}(\mathcal{A})=p$, the above results states that only the p largest indices of $\mathcal{C}$ will belong in the active set $\mathcal{A}$, i.e., $c_i\in\mathcal{A}$, for $i=l-p+1,\ldots,l$. Thus, in order to find the active set, we need to find all the indices that $z_{c_i}\ge\bar{z}_q$ is true, where $\bar{z}_q$ is the average of $z_{[q-r:q]}$ and $z_{[c_{i+1}:c_k]}$, as given in (87). ■

REFERENCES

[1] K. Benidis, Y. Sun, P. Babu, and D. P. Palomar, "Orthogonal sparse eigenvectors: A procrustes problem," in *Proc. IEEE Int. Conf. on Acous., Speech and Signal Proc. (ICASSP)*, 2016, pp. 4683–4686.

[2] I. T. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.

[3] I. T. Jolliffe, "Rotation of principal components: choice of normalization constraints," *J Appl. Statist.*, vol. 22, no. 1, pp. 29–35, 1995.

[4] J. Cadima and I. T. Jolliffe, "Loading and correlations in the interpretation of principle compenents," *J. Appl. Statist.*, vol. 22, no. 2, pp. 203–214, 1995.

[5] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the LASSO," *J. Comput. Graph. Statist.*, vol. 12, no. 3, pp. 531–547, 2003.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc. Ser. B (Methodological)*, vol. 58, pp. 267–288, 1996.

[7] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, 2006.

[8] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *SIAM Rev.*, vol. 49, no. 3, pp. 434–448, Jul. 2007.

[9] A. d'Aspremont, F. Bach, and L. E. Ghaoui, "Optimal solutions for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 9, pp. 1269–1294, Jun. 2008.

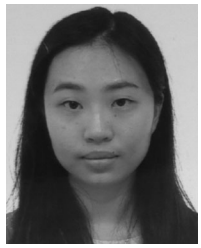
[10] H. Shen and J. Z. Huang, "Sparse principal component analysis via regularized low rank matrix approximation," *J. Multivariate Anal.*, vol. 99, no. 6, pp. 1015–1034, Jul. 2008.

[11] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 517–553, Mar. 2010.

[12] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, vol. 10, pp. 515–534, Jul. 2009.

[13] X.-T. Yuan and T. Zhang, "Truncated power method for sparse eigenvalue problems," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 899–925, 2013.
[14] Z. Ma, "Sparse principal component analysis and iterative thresholding," *Ann. Statist.*, vol. 41, no. 2, pp. 772–801, 2013.
[15] J. Song, P. Babu, and D. P. Palomar, "Sparse generalized eigenvalue problem via smooth optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1627–1642, Apr. 2015.
[16] R. Zass and A. Shashua, "Nonnegative sparse PCA," in *Proc. Adv. Neural Inf. Process Syst.*, 2006, pp. 1561–1568.
[17] M. Asteris, D. Papailiopoulos, A. Kyrillidis, and A. G. Dimakis, "Sparse pca via bipartite matchings," in *Proc. 28th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 766–774.
[18] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1269–1276.
[19] S. Tadjudin and D. Landgrebe, "Covariance estimation for limited training samples," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 1998, vol. 5, pp. 2688–2690.
[20] O. Ledoit and M. Wolf, "Honey, I shrunk the sample covariance matrix," *J. Portfolio Manag.*, vol. 30, no. 4, pp. 110–119, Jun. 2004.
[21] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivariate Anal.*, vol. 88, no. 2, pp. 365–411, Feb. 2004.
[22] P. J. Bickel and E. Levina, "Regularized estimation of large covariance matrices," *Ann. Statist.*, vol. 36, pp. 199–227, 2008.
[23] Y. Sun, P. Babu, and D. P. Palomar, "Regularized robust estimation of mean and covariance matrix under heavy-tailed distributions," *IEEE Trans. Signal Process.*, vol. 63, no. 12, pp. 3096–3109, Jun. 2015.
[24] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
[25] E. Levina, A. Rothman, and J. Zhu, "Sparse estimation of large covariance matrices via a nested Lasso penalty," *Ann. Appl. Statist.*, vol. 2, no. 1, pp. 245–263, 2008.
[26] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, "First-order methods for sparse covariance selection," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 56–66, Jan. 2008.
[27] A. Kyrillidis, R. K. Mahabadi, Q. Tran-Dinh, and V. Cevher, "Scalable sparse covariance estimation via self-concordance," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1946–1952.
[28] S. Cocco, R. Monasson, and M. Weigt, "From principal component to direct coupling analysis of coevolution in proteins: Low-eigenvalue modes are needed for structure prediction," *PLoS Comput. Biol.*, vol. 9, no. 8, 2013, Art. no. e1003176.
[29] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5/6, pp. 877–905, Dec. 2008.
[30] M. A. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization–minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2980–2991, Dec. 2007.
[31] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.
[32] Y. I. Abramovich, "A controlled method for adaptive optimization of filters using the criterion of maximum signal-to-noise ratio," *Radio Eng. Electron. Phys.*, vol. 26, pp. 87–95, 1982.
[33] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
[34] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[35] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 635–650, Mar. 2002.
[36] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.
[37] R. Varadhan and C. Roland, "Simple and globally convergent methods for accelerating the convergence of any EM algorithm," *Scand. J. Statist.*, vol. 35, no. 2, pp. 335–353, 2008.
[38] A. H. Bild *et al.*, "Oncogenic pathway signatures in human cancers as a guide to targeted therapies," *Nature*, vol. 439, no. 7074, pp. 353–357, Jan. 2006.
[39] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Konstantinos Benidis received the M.Eng. degree from the School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece, in 2011, and the M.Sc. degree in information and communication technologies from the Polytechnic University of Catalonia, Barcelona, Spain, in 2013. He is currently working toward the Ph.D. degree in the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. His research interests include convex optimization and efficient algorithms, with applications in signal processing, financial engineering, and machine learning.



Ying Sun received the B.E. degree in electronic information from Huazhong University of Science and Technology, Wuhan, China, in 2011, and the Ph.D. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Hong Kong, in 2016. She is currently a Postdoctoral in the School of Industrial Engineering, Purdue University, West Lafayette, IN, USA. Her research interests include statistical signal processing, optimization algorithms, and machine learning.

Prabhu Babu received the Ph.D. degree in electrical engineering from Uppsala University, Uppsala, Sweden, in 2012. From 2013 to 2016, he was a Postdoctorial Fellow with Hong Kong University of Science and Technology. He is currently in the Centre for Applied Research in Electronics, Indian Institute of Technology Delhi, New Delhi, India.

Daniel P. Palomar (S'99–M'03–SM'08–F'12) received the electrical engineering and Ph.D. degrees from the Technical University of Catalonia, Barcelona, Spain, in 1998 and 2003, respectively.

Since 2006, he has been a Professor in the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong. Since 2013, he has been a Fellow of the Institute for Advance Study, HKUST. He had previously held several research appointments, namely, at King's College London, London, U.K.; Stanford University, Stanford, CA, USA; Telecommunications Technological Center of Catalonia, Barcelona, Spain; Royal Institute of Technology, Stockholm, Sweden; University of Rome "La Sapienza", Rome, Italy; and Princeton University, Princeton, NJ, USA. His current research interests include applications of convex optimization theory, game theory, and variational inequality theory to financial systems, big data systems, and communication systems.

Dr. Palomar was the recepient of the 2004/06 Fulbright Research Fellowship, the 2004 Young Author Best Paper Award by the IEEE Signal Processing Society, the 2002/03 Best Ph.D. Prize in Information Technologies and Communications by the Technical University of Catalonia, the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in Information Technologies and Communications by the Epson Foundation, and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation.

He is a Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING 2016 Special Issue on "Financial Signal Processing and Machine Learning for Electronic Trading," an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE TRANSACTIONS ON SIGNAL PROCESSING, and a Guest Editor of the *IEEE Signal Processing Magazine* 2010 Special Issue on "Convex Optimization for Signal Processing," IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2008 Special Issue on "Game Theory in Communication Systems," and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2007 Special Issue on "Optimization of MIMO Transceivers for Realistic Communication Networks."