

# Sparse PCA With Multiple Components

Ryan Cory-Wright

Department of Analytics, Marketing and Operations, Imperial College Business School, London, UK  
ORCID: 0000-0002-4485-0619  
r.cory-wright@imperial.ac.uk

Jean Pauphilet

Management Science and Operations, London Business School, London, UK  
ORCID: 0000-0001-6352-0984  
jpauphilet@london.edu

Sparse Principal Component Analysis (sPCA) is a cardinal technique for obtaining combinations of features, or principal components (PCs), that explain the variance of high-dimensional datasets in an interpretable manner. This involves solving a sparsity and orthogonality constrained convex maximization problem, which is extremely computationally challenging. Most existing works address sparse PCA via methods—such as iteratively computing one sparse PC and deflating the covariance matrix—that do not guarantee the orthogonality, let alone the optimality, of the resulting solution when we seek multiple mutually orthogonal PCs. We challenge this status by reformulating the orthogonality conditions as rank constraints and optimizing over the sparsity and rank constraints simultaneously. We design tight semidefinite relaxations to supply high-quality upper bounds, which we strengthen via additional second-order cone inequalities when each PC’s individual sparsity is specified. Further, we derive a combinatorial upper bound on the maximum amount of variance explained as a function of the support. We exploit these relaxations and bounds to propose exact methods and rounding mechanisms that, together, obtain solutions with a bound gap on the order of 0%–15% for real-world datasets with  $p = 100$ s or  $1000$ s of features and  $r \in \{2, 3\}$  components. Numerically, our algorithms match (and sometimes surpass) the best performing methods in terms of fraction of variance explained and systematically return PCs that are sparse and orthogonal. In contrast, we find that existing methods like deflation return solutions that violate the orthogonality constraints, even when the data is generated according to sparse orthogonal PCs. Altogether, our approach solves sparse PCA problems with multiple components to certifiable (near) optimality in a practically tractable fashion.

*Key words:* Sparse Principal Component Analysis; Semidefinite Optimization; Practical Tractability

---

## 1. Introduction

Principal Component Analysis, or PCA, initially proposed by Pearson (1901), is one of the most popular techniques used by data practitioners to reduce the dimension of a dataset (see also Hotelling 1933, Eckart and Young 1936). Given a normalized and centered data matrix  $\mathbf{A} \in \mathbb{R}^{n \times p}$ , and its sample covariance matrix  $\mathbf{\Sigma} := \frac{1}{n-1} \mathbf{A} \mathbf{A}^\top$ , they identify the top  $r$  principal components ( $r \ll p$ ) of  $\mathbf{\Sigma}$  by solving:

$$\max_{\mathbf{U} \in \mathbb{R}^{p \times r}} \langle \mathbf{U} \mathbf{U}^\top, \mathbf{\Sigma} \rangle \text{ s.t. } \mathbf{U}^\top \mathbf{U} = \mathbb{I}, \quad (1)$$

and subsequently project the data matrix  $\mathbf{A}$  onto the principal components  $\mathbf{U}$ . As described by Hotelling (1933), the principal components of  $\mathbf{\Sigma}$  correspond to its  $r$  leading eigenvectors and can efficiently be obtained via a greedy procedure where, at each iteration, the leading eigenvector of  $\mathbf{\Sigma}$ ,  $\mathbf{u}$ , is computed (e.g., by solving (1) with  $r = 1$ ) and then the matrix  $\mathbf{\Sigma}$  is updated (or *deflated*) to eliminate the influence of  $\mathbf{u}$ . PCA is now a cardinal unsupervised learning paradigm that is practically useful across a range of fields, including pattern recognition (Naikal et al. 2011), sequence classification (Tan et al. 2014), factor models in finance (Fan et al. 2016), and variable selection in statistics (Wang et al. 2023).

Despite efficient modern implementations (Udell et al. 2016, Tropp et al. 2017), PCA suffers from at least two limitations. First, it generates components that are dense linear combination of the original features and hence uninterpretable (Rudin et al. 2022). Second, it yields inconsistent estimates in high-dimensional settings where  $p/n \rightarrow \alpha > 0$  (Johnstone and Lu 2009). Accordingly, several authors (such as Jolliffe et al. 2003, d’Aspremont et al. 2007) have proposed sparse PCA, namely augmenting Problem (1) with a sparsity constraint. When  $r = 1$ , sparse PCA can be formulated as the following optimization problem:

$$\max_{\mathbf{u} \in \mathbb{R}^p} \langle \mathbf{\Sigma}, \mathbf{u}\mathbf{u}^\top \rangle \text{ s.t. } \|\mathbf{u}\|_2^2 = 1, \|\mathbf{u}\|_0 \leq k, \quad (2)$$

where  $\|\mathbf{u}\|_0$  denotes the cardinality of  $\mathbf{u}$  or the size of its support:  $\|\mathbf{u}\|_0 = |\text{supp}(\mathbf{u})| = |\{j : \mathbf{u}_j \neq 0\}|$ . From a statistical recovery perspective, assuming the data is generated according to a ‘true’ covariance matrix of the form  $\mathbf{\Sigma}^* = \beta \mathbf{v}\mathbf{v}^\top + \mathbf{I}_p$ , for some  $\mathbf{v}$  with  $\|\mathbf{v}\|_0 \leq k$  and some  $\beta > 0$ , Amini and Wainwright (2008) show that an exhaustive search algorithm can reliably identify the support of  $\mathbf{v}$  provided that the number of samples scales at the rate of  $k \log(p)$  ( $n \gtrsim k \log(p)$ , in short), which constitutes a significant improvement over the traditional PCA formulation that requires  $n \gtrsim p$ . On the other hand, they show that no method can succeed when  $n \lesssim k \log(p)$ , because of information theoretic limitations. Berthet and Rigollet (2013) analyze the support recovery ability of polynomial-time algorithms and show that no polynomial-time algorithm can succeed when  $n \lesssim k^2$ . Hence, there exists a regime,  $k \log(p) \lesssim n \lesssim k^2$ , where exhaustive search successfully detects the support of  $\mathbf{v}$ , while no polynomial-time algorithm can.

This gap motivated the development of tailored discrete optimization algorithms to efficiently implement exhaustive search for sparse PCA with a single PC. Indeed, Problem (2) can be formulated as a mixed-integer semidefinite optimization problem and solved via global optimization techniques such as branch-and-bound (Berk and Bertsimas 2019), or branch-and-cut (Bertsimas et al. 2022b). Confirming the statistical theory, certifiably optimal methods for (2) are typically and often significantly more accurate than polynomial-time methods. For example, Berk and Bertsimas (2019) compared the performance of seven popular sparse PCA heuristics across four UCI

datasets ( $k \in \{5, 10\}$ ) and observed that no heuristic found an optimal solution across all datasets, with some methods being routinely 20% suboptimal or more. On the other hand, exact methods found the optimal solution in all cases on the same benchmark.

Unfortunately, no consensual formulation extends the formulation of Problem (2) to  $r > 1$ , and there are no practically relevant algorithms with optimality guarantees that successfully address this extension. Indeed, most algorithmic work for sparse PCA with one PC cannot be readily generalized to the case with  $r > 1$ . This is because, in the multiple component case, the sparsity constraint in (2) causes sparse principal components to no longer be eigenvectors of  $\Sigma$  and deflation methods to no longer lead to orthogonal, let alone optimal, PCs (Mackey 2008). In other words, the conventional PCA wisdom that multiple components can be computed one by one fails as soon as sparsity is required. This observation calls for new methods for sparse PCA with multiple PCs that optimize the components simultaneously. In response, in this paper, we propose a generic optimization formulation that extends Problem (2) to  $r > 1$ , reformulate it as a sparsity and rank constrained optimization problem, and design certifiably (near) optimal techniques to solve it.

### 1.1. A Generic Formulation for Sparse PCA with Multiple PCs

Perhaps the most natural extension of sparse PCA to multiple principal components, and the one which we advocate in this paper, is to augment Problem (1) with a constraint on the number of non-zero entries in the matrix  $\mathbf{U}$ ,

$$\|\mathbf{U}\|_0 = |\text{supp}(\mathbf{U})| = |\{(i, j) \in [p] \times [r] : U_{i,j} \neq 0\}| \leq k.$$

This gives a formulation which enforces two desirable properties on the matrix  $\mathbf{U}$ : orthogonality ( $\mathbf{U}^\top \mathbf{U} = \mathbb{I}$ ), as present in the prototypical formulation of PCA (see, e.g., Horn and Johnson 1985); and sparsity, to address interpretability and accuracy concerns (c.f. Rudin et al. 2022).

Formally, introducing a binary matrix  $\mathbf{Z}$  to encode the support of  $\mathbf{U}$ , we consider the problem:

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in \{0,1\}^{p \times r} \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k}} \max_{\mathbf{U} \in \mathbb{R}^{p \times r}} \quad & \langle \mathbf{U} \mathbf{U}^\top, \Sigma \rangle \\ \text{s.t.} \quad & \mathbf{U}^\top \mathbf{U} = \mathbb{I}, U_{i,t} = 0 \text{ if } Z_{i,t} = 0, \forall i \in [p], \forall t \in [r], \end{aligned} \tag{3}$$

where  $pr > k > r$  in order that the problem is well-posed and non-trivial.

Alternatively, instead of imposing an overall sparsity budget ( $k$ ) on the entire matrix  $\mathbf{U}$ , we could restrict the size of the support of each column of  $\mathbf{U}$  separately, i.e., impose  $\|\mathbf{U}_t\|_0 \leq k_t \forall t \in [r]$  and adapt (3) accordingly. Indeed, optimizing this formulation over all integer combinations  $k_t$  which sum to  $k$  is equivalent to solving (3). We address both modeling options in this paper. We remark that although Problem (3) is a very natural extension of Problem (2), we are not aware

of any works that explicitly formulate sparse PCA with multiple components as an orthogonality constrained problem with logical constraints; the formulation (3) is, to our knowledge, new.

From a generative model perspective, Problem (3) is consistent with a spiked covariance model (see, e.g., Amini and Wainwright 2008, d’Aspremont et al. 2008), where the true covariance matrix  $\Sigma^*$  can be decomposed as the sum of a sparse and low-rank term plus some noise:

$$\Sigma^* = \sum_{t \in [r]} \beta_t \mathbf{v}_t \mathbf{v}_t^\top + \mathbf{N}, \quad (4)$$

where  $\mathbf{v}_t$  are sparse vectors that may have non-overlapping, partially overlapping, or completely overlapping support, and  $\mathbf{N}$  is a noise matrix. This generative model is referred to as the spiked Wishart model when  $\mathbf{N} = \mathbf{I}_p$ , and the spiked Wigner model when  $\mathbf{N}$  is drawn from the Gaussian orthogonal ensemble (see Ding et al. 2023, for a general theory of both families of assumptions).

## 1.2. Literature Review

To identify the extent to which the state-of-the-art for sparse PCA could be improved, we now review methods that have been proposed to approximately solve sparse PCA with multiple PCs.

*Methods for sparse PCA with  $r = 1$ :* Several polynomial-time algorithms have been proposed to obtain high-quality solutions to (2), including greedy heuristics (d’Aspremont et al. 2008),  $\ell_1$  relaxations (Zou et al. 2006, d’Aspremont et al. 2007, Dey et al. 2022a), linear regression-based estimators (Bresler et al. 2018, Behdin and Mazumder 2021), or thresholding techniques (Johnstone and Lu 2009, Deshpande and Montanari 2014b). Note that the covariance thresholding method of Deshpande and Montanari (2014b) provably recovers the support in the spiked Wishart model whenever  $n \gtrsim k^2$ , which is the best achievable rate for polynomial-time methods (Berthet and Rigollet 2013). Over the past decade, various authors including Gally and Pfetsch (2016), Bertsimas et al. (2022b), Li and Xie (2020), Kim et al. (2022) have shown that Problem (2) can be recast as a mixed-integer semidefinite optimization (MISDO) problem and have derived both high-quality solutions and valid dual bounds using this discrete optimization lens.

*Deflation methods for sparse PCA:* It is well known that an optimal solution to Problem (1) can be obtained via a greedy deflation procedure. Consequently, Mackey (2008) propose a sparse extension of this deflation scheme where, at each iteration, a sparse PC is computed (e.g., by solving (2) or a relaxation) and  $\Sigma$  is updated by projecting out the eigenspace modeled by  $\mathbf{u}$ :  $\Sigma_{\text{new}} = (\mathbb{I} - \mathbf{u}\mathbf{u}^\top)\Sigma(\mathbb{I} - \mathbf{u}\mathbf{u}^\top)$ . Empirically, this method often performs reasonably well (see Berk and Bertsimas 2019, Section 5.3), particularly when Problem (2) is solved to global optimality; see also Hein and Bühler (2010), Bühler (2014) for a related deflation-based scheme. However, unlike in the traditional case, deflation need not return an optimal solution to (3)—see Asteris et al. (2015) for a simple four-dimensional example. Actually, deflation need not even return a feasible solution, since

the orthogonality constraint is not explicitly imposed by the method, and is therefore often violated in practice. Because they cannot guarantee the feasibility (i.e., orthogonality) of the returned PCs, we consider deflation-based methods as heuristics methods for our sparse PCA problem with multiple PCs. Furthermore, because of their iterative nature, deflation-based procedures can easily control the sparsity of each component but usually struggle to enforce global sparsity.

*Methods for generic sparse PCA:* A second approach for solving Problem (3) is to apply a heuristic which (approximately) optimizes all  $r$  PCs simultaneously, rather than sequentially. Among others, Zou et al. (2006) propose an alternating minimization scheme for an  $\ell_1$  relaxation of Problem (3), Journée et al. (2010) propose an iterative conditional gradient method to identify a local optimum of Problem (3) without the orthogonality constraints, Lu and Zhang (2012) apply an augmented Lagrangian method which solves an  $\ell_1$  relaxation of (3), Vu et al. (2013) solve a semidefinite relaxation of Problem (3)'s  $\ell_1$  relaxation, and Benidis et al. (2016) adopt a minorization-maximization approach which also solves Problem (3) approximately. Unfortunately, these approaches are often suboptimal if  $r = 1$ , and only provide candidate solutions, with no indication on the optimality gap. Indeed, none of these approaches explicitly control both the sparsity and orthogonality constraints, and therefore none of the methods reviewed here are guaranteed to return feasible solutions to (3). We should note that the covariance thresholding method of Deshpande and Montanari (2014b) can be applied to the  $r > 1$  case and can return PCs that are asymptotically orthogonal. However, for a given dataset, it cannot guarantee the orthogonality of the returned solution, as we observe empirically in Sections 5.2 and 5.3.

*Row-Sparsity:* Motivated by tractability concerns, another line of work studies a special case of (3), namely row-sparse principal component analysis or principal component analysis with global support. This formulation replaces the sparsity constraint on  $\mathbf{U}$  with one requiring  $\mathbf{U}$  has at most  $k$  non-zero rows, as advocated by Boutsidis et al. (2011), Probel and Tropp (2011), Vu and Lei (2013). This rewrites sparse PCA as performing a top- $r$  SVD on a subset of  $k$  rows of  $\mathbf{\Sigma}$ , i.e.,

$$\max_{\mathbf{z} \in \{0,1\}^p: \mathbf{e}^\top \mathbf{z} \leq k} \max_{\mathbf{U} \in \mathbb{R}^{p \times r}} \langle \mathbf{U}\mathbf{U}^\top, \mathbf{\Sigma} \rangle \text{ s.t. } \mathbf{U}^\top \mathbf{U} = \mathbb{I}_{r \times r}, U_{i,t} = 0 \text{ if } z_i = 0 \forall i \in [p], \forall t \in [r]. \quad (5)$$

Problem (5) is a special case of Problem (3) where each PC has the same support, which corresponds to constraining  $\mathbf{Z}$  such that  $z_{i,t} = z_{j,t}$ ,  $\forall t \in [r]$  in (3), and dividing the “ $k$ ” in (3) by  $r$ . This restriction is advantageous from a computational perspective, but disadvantageous from a statistical one. Indeed, because of the global support assumption, Problem (5) can be reformulated as a mixed-integer semidefinite problem (see Bertsimas et al. 2022b, Li and Xie 2021, Bertsimas and Kitane 2023, for derivations) and solved via relax-and-round (Dey et al. 2022b, Li and Xie 2021) or branch-and-bound (Del Pia 2023) strategies. However, from a generative model perspective (4), it is equivalent to making the very strong assumption that all leading eigenvectors  $\mathbf{v}_t$  have

the same sparsity pattern. Indeed, if one PC  $\mathbf{v}_t$  is sparser than  $k$  ( $\|\mathbf{v}_t\|_0 \leq k_t < k$ ) or two PCs are partially disjoint then Problem (5) will nonetheless estimate that the support of each  $\mathbf{u}_t$  is of size  $k$ . Therefore, Problem (5) is vulnerable to consistently making false discoveries in the identification of relevant features.

*Disjoint Sparsity:* Another relevant special case of sparse PCA with multiple PCs is when the supports of all  $r$  columns of  $\mathbf{U}$  are assumed mutually disjoint, as originally proposed by Asteris et al. (2015). This gives rise to the formulation:

$$\max_{\substack{\mathbf{Z} \in \{0,1\}^{p \times r}, \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k, \mathbf{Z}\mathbf{e} \leq \mathbf{e}}} \max_{\mathbf{U} \in \mathbb{R}^{p \times r}} \langle \mathbf{U}\mathbf{U}^\top, \boldsymbol{\Sigma} \rangle \text{ s.t. } \mathbf{U}^\top \mathbf{U} = \mathbb{I}_{r \times r}, U_{i,t} = 0 \text{ if } Z_{i,t} = 0 \forall i \in [p], t \in [r], \quad (6)$$

where  $\mathbf{E}$  denotes a matrix of all ones of the appropriate dimension.

Note that (6) is a special case of (3) where we additionally require that  $\sum_{t \in [r]} Z_{i,t} \leq 1 \forall i \in [p]$ , i.e., that each feature can be included in at most one PC. Interestingly, this restriction allows (6) to be recast as a MISDO and solved as such (c.f. Bertsimas et al. 2022b).

An obvious criticism of formulation (6) is that, in practice, we may wish to include a feature in multiple PCs. Therefore, (6) is best thought of as a special case of sparse PCA. Indeed, if the true generative model involves vectors  $\mathbf{v}_t$  with partially overlapping supports, then (6) could not recover them. Nonetheless, as we explore in our numerical experiments (Section 5), disjoint solutions often perform well when  $k$  is very small relative to  $p$ . This is because, in high-dimensional settings, there are often several disjoint submatrices which are near-optimal in the rank-one case, and selecting the leading PCs from each of them is often a reasonable approach in practice.

### 1.3. Contributions and Structure

To our knowledge, no existing algorithm solves sparse PCA problems with multiple components and obtains certificates of optimality, except in the aforementioned special cases of row-sparsity or disjoint support. Accordingly, we undertake a detailed study of Problem (3) in its full generality.

The main contributions of the paper are twofold. First, we propose a novel reformulation of Problem (3) as a mixed-integer low-rank problem. Based on this reformulation, we derive tight, yet tractable, semidefinite and second-order cone relaxations of (3) and propose valid inequalities that strengthen the relaxations. Second, by leveraging our convex relaxations and a combinatorial upper bound that generalizes the Gershgorin circle theorem to encompass multiple components, we propose rounding techniques and an alternating minimization scheme to obtain high-quality solutions. By combining strong relaxations and good solutions, we obtain bound gaps of 5% on real-world instances of sparse PCA with multiple PCs where  $p = 100$ s or  $1000$ s. We provide the first reformulation of the sparse PCA problem (3) that is amenable to certifiably optimal algorithms and admits tractable relaxations, while demonstrating its benefit compared with state-of-the-art

heuristics. For example, on the `pitprops` dataset, with 6 PCs, we explain 81% of the variance with an overall sparsity of 24 while previous studies could only explain less than 70% of the variance with twice as many variables (see Section 5.2). The structure of the paper is as follows:

- In Section 2.1, we prove that rank constraints successfully model orthogonality constraints. Leveraging this result, we reformulate Problem (3) as a sparsity and rank constrained optimization problem and derive a semidefinite relaxation. We propose valid inequalities which strengthen the relaxation in Section 2.2, and additional valid inequalities which hold if we restrict the support of each PC separately in Section 2.3. For larger instances ( $p \geq 100$  in our experiments), we also propose a tractable second-order cone relaxation in Section EC.4.2.
- In Section 3, we generalize the Gershgorin Circle Theorem to multiple PCs with sparsity constraints, and demonstrate that this generalization gives rise to a combinatorial upper bound on Problem (3)'s objective value as a function of the support of the PCs.
- In Section 4.1, we develop an exact formulation of Problem (3) in the original space of decision variables, which can be addressed via global branch-and-bound solvers, and propose valid inequalities which improve the tightness of the formulation. In Section 4.2, we develop a greedy rounding strategy which rounds an optimal solution to one of the relaxations from Section 2 into  $r$  sparse PCs with disjoint support (hence, orthogonal). Moreover, we use the combinatorial upper bound derived in Section 3 to improve the rounding procedure by breaking the symmetry that may otherwise arise in our semidefinite relaxations. Finally, in Section 4.3, we consider a Lagrangian relaxation of Problem (3) and design an alternating minimization strategy to obtain near-orthogonal solutions by iteratively solving a sequence of sparse PCA problems with  $r = 1$ . Compared with existing deflation-based techniques, we explicitly penalize the orthogonality violation at each iteration, with an increasing penalty parameter, thus converging towards an orthogonal and typically high-quality solution.
- In Section 5, we investigate the quality of our semidefinite relaxations and feasible methods and observe empirically that they collectively give small bound gaps in practice. First, we invoke our convex relaxations to obtain high-quality upper bounds on UCI datasets in Section 5.1. We also compare our approximate methods with existing methods from the literature. On both synthetic and UCI datasets, we demonstrate in Sections 5.2-5.3 that our approaches, particularly the alternating minimization scheme developed in Section 4.3, explain more of the variance in the data and exhibit a lower false discovery/higher true discovery rate for a given sparsity budget. Finally, in Section 5.4, we explore the relationship between the symmetry in the size of each PCs sparsity budget and the variance explained, and demonstrate that the best asymmetric sparsity budget tends to explain more variance than a symmetric sparsity budget with the same total sparsity. For instance, on the `Geographical` dataset with 3 PCs, sparsity budgets of (12, 12, 6) explain 6% more variance than (10, 10, 10), although both have a total sparsity of 30.

## 1.4. Preliminaries and Notation

We let nonbold face characters such as  $u$  denote scalars, lowercase bold-faced characters such as  $\mathbf{u}$  denote vectors, uppercase bold-faced characters such as  $\mathbf{U}$  denote matrices, and calligraphic uppercase characters such as  $\mathcal{Z}$  denote sets. If  $\mathbf{U}$  is a matrix then  $\mathbf{U}_t$  denotes the  $t$ th column vector of  $\mathbf{U}$ , and  $U_{i,t}$  denotes the  $(i,t)$ th entry of  $\mathbf{U}$ . We let  $[p]$  denote the set of running indices  $\{1, \dots, p\}$ . We let  $\mathbf{e}$  denote a vector of all 1's,  $\mathbf{0}$  denote a vector of all 0's, and  $\mathbb{I}$  denote the identity matrix, with dimension implied by the context.

We also use an assortment of matrix operators. We let  $\langle \cdot, \cdot \rangle$  denote the Euclidean inner product between two matrices,  $\|\cdot\|_F$  denote the Frobenius norm, and  $\mathcal{S}_+^p$  denote the  $p \times p$  positive semidefinite cone; see Horn and Johnson (1985) for a general theory of matrix operators.

Further, we use some basic properties of orthogonal projection matrices. Let  $\mathcal{Y}_n := \{\mathbf{Y} \in \mathcal{S}^n : \mathbf{Y}^2 = \mathbf{Y}\}$  denote the set of  $n \times n$  projection matrices and  $\mathcal{Y}_n^k := \{\mathbf{Y} \in \mathcal{Y}_n : \text{tr}(\mathbf{Y}) \leq k\}$  denote projection matrices with rank at most  $k$ : note that  $\text{Rank}(\mathbf{Y}) = \text{tr}(\mathbf{Y})$  for any projection matrix  $\mathbf{Y}$ . Among others, the convex hulls of  $\mathcal{Y}_n$  and  $\mathcal{Y}_n^k$  are well-studied, as we now remind the reader:

LEMMA 1. (*Theorem 3 of Overton and Womersley 1992*)  $\text{Conv}(\mathcal{Y}_n) = \{\mathbf{P} : 0 \preceq \mathbf{P} \preceq \mathbb{I}\}$  and  $\text{Conv}(\mathcal{Y}_n^k) = \{\mathbf{P} : 0 \preceq \mathbf{P} \preceq \mathbb{I}, \text{tr}(\mathbf{P}) \leq k\}$ . Moreover, the extreme points of  $\text{Conv}(\mathcal{Y}_n)$  are  $\mathcal{Y}_n$ , and the extreme points of  $\text{Conv}(\mathcal{Y}_n^k)$  are  $\mathcal{Y}_n^k$ .

Finally, we repeatedly reference two results on the convex hulls of convex quadratic functions under logical constraints:

LEMMA 2. (*Lemma 4 of Günlük and Linderoth 2010*) The convex closure of the set

$$\mathcal{S} = \left\{ (\mathbf{x}, \mathbf{z}, t) \in \mathbb{R}^n \times \{0, 1\}^n \times \mathbb{R} : t \geq \sum_{i=1}^n x_i^2, \mathbf{e}^\top \mathbf{z} \leq k, x_i = 0 \text{ if } z_i = 0, \forall i \in [n] \right\}$$

is given by

$$\mathcal{S}^c = \left\{ (\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}, t) \in \mathbb{R}^n \times [0, 1]^n \times \mathbb{R}^n \times \mathbb{R} : t \geq \sum_{i=1}^n \theta_i, \mathbf{e}^\top \mathbf{z} \leq k, \theta_i z_i \geq x_i^2 \forall i \in [n] \right\}.$$

The above result is sometimes known as a perspective reformulation, since we strengthen the quadratic constraint  $t \geq \sum_i x_i^2$  by replacing  $x_i^2$  with its perspective  $z_i(x_i/z_i)^2$ .

LEMMA 3. (*Theorem 1 of Atamtürk and Gomez 2019*) The convex closure of the set

$$\mathcal{T} = \{(\mathbf{x}, \mathbf{z}, t) \in \mathbb{R}^n \times \{0, 1\}^n \times \mathbb{R} : t \geq (\mathbf{e}^\top \mathbf{x})^2, \mathbf{e}^\top \mathbf{z} \leq k, x_i = 0 \text{ if } z_i = 0, \forall i \in [n]\},$$

is given by

$$\mathcal{T}^c = \{(\mathbf{x}, \mathbf{z}, t) \in \mathbb{R}^n \times [0, 1]^n \times \mathbb{R} : t \cdot \min(1, \mathbf{e}^\top \mathbf{z}) \geq (\mathbf{e}^\top \mathbf{x})^2, \mathbf{e}^\top \mathbf{z} \leq k\}.$$

Lemma 3 is extremely useful when a single continuous variable depends upon multiple indicator variables, as occurs in certain substructures of our reformulations of Problem (3).



## 2. Exact Formulations and Their Relaxations

In this section, we reformulate Problem (3) as a mixed-integer low-rank problem, study its semidefinite and second-order cone relaxations, and propose valid inequalities for strengthening them.

### 2.1. An Extended Formulation and Its Semidefinite Relaxation

Our sparse PCA formulation (3) is a mixed-integer quadratic optimization problem that exhibits three primary sources of difficulty. First, as typical in PCA problems, Problem (3) maximizes a convex quadratic function in the decision variable  $\mathbf{U}$ . Second, there is a sparsity constraint, which is computationally challenging to model, although recent evidence suggests that sparsity constraints need not imply intractability (see, e.g., Bertsimas et al. 2020). Finally, and more consequentially, there is an orthogonality constraint. To our knowledge, existing generic non-convex solvers such as **Gurobi** cannot optimize over such orthogonality constraints at a scale of  $p = 100$ s of features.

To address the three aforementioned difficulties, we now derive an orthogonality-free reformulation in five steps. First, we introduce the matrix  $\mathbf{Y} = \mathbf{U}\mathbf{U}^\top \preceq \mathbb{I}$ , thus linearizing the non-convex objective. Second, we introduce rank-one matrices  $\mathbf{Y}^t$  to model the outer products of each column of  $\mathbf{U}$ ,  $\mathbf{U}_t$ , with itself,  $\mathbf{U}_t\mathbf{U}_t^\top$ . Third, we reassign the indicator variable  $Z_{i,t}$  to model whether  $Y_{i,j}^t$ , rather than  $U_{i,t}$ , is non-zero. Fourth, by letting  $\mathbf{Y} = \sum_{t=1}^r \mathbf{Y}^t$ , we observe that we can omit the matrix  $\mathbf{U}$  (and the constraints involving  $\mathbf{U}$ ) without altering the set of feasible  $\mathbf{Y}$ 's. Finally, we use the fact that  $\mathbf{Y}_{i,j}^t$  is only supported on indices  $i, t$  where  $Z_{i,t} > 0$  to strengthen the constraint  $\mathbf{Y} \preceq \mathbb{I}$  to  $\mathbf{Y} \preceq \text{Diag}(\min(\mathbf{e}, \sum_t \mathbf{Z}_t))$ . Formally, we have:

**THEOREM 1.** *Problem (3) attains the same optimal objective value as the problem:*

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in \{0,1\}^{p \times r}: \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k}} \max_{\mathbf{Y} \in \mathcal{S}^p, \mathbf{Y}^t \in \mathcal{S}_+^p} \langle \mathbf{Y}, \boldsymbol{\Sigma} \rangle \quad \text{s.t.} \quad & \mathbf{Y} \preceq \text{Diag} \left( \min \left( \mathbf{e}, \sum_t \mathbf{Z}_t \right) \right), \mathbf{Y} = \sum_{t=1}^r \mathbf{Y}^t, \\ & \text{tr}(\mathbf{Y}^t) = 1, \quad \forall t \in [r], \quad Y_{i,j}^t = 0 \text{ if } Z_{i,t} = 0, \quad \forall t \in [r], i, j \in [p], \\ & \text{Rank}(\mathbf{Y}^t) = 1, \quad \forall t \in [r]. \end{aligned} \quad (7)$$

**REMARK 1.** We do not explicitly require  $\mathbf{Y} \succeq \mathbf{0}$ , since  $\mathbf{Y}$  is the sum of positive semidefinite matrices  $\mathbf{Y}^t$ . Indeed, omitting  $\mathbf{Y} \succeq \mathbf{0}$  substantially improves the tractability of the relaxations of (7).

The proof of Theorem 1 requires an intermediate result (Proposition 1). Proposition 1 shows that by imposing rank-one constraints on each  $\mathbf{Y}^t$ , the condition that the  $\mathbf{Y}^t$ 's are mutually orthogonal can be reformulated as a linear semidefinite constraint. Independently, Proposition 1 is crucial for designing our main algorithm in Section 4.3 (proof deferred to Section EC.1).

**PROPOSITION 1.** *Consider  $r$  matrices,  $\mathbf{Y}^t \in \mathcal{S}_+^p$ , such that  $\text{tr}(\mathbf{Y}^t) = 1$  and  $\text{Rank}(\mathbf{Y}^t) = 1$ . Then,  $\sum_{t \in [r]} \mathbf{Y}^t \preceq \mathbb{I}$  if and only if  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = 0 \quad \forall t, t' \in [r] : t \neq t'$ .*

*Proof of Theorem 1* It suffices to show that for any feasible solution to (3), we can construct a feasible solution to Problem (7) with an equal or greater payoff, and vice versa.

- Let  $(\mathbf{U}, \mathbf{Z})$  be a solution to Problem (3). Then, since  $U_{i,t}$  can only be non-zero if  $Z_{i,t} = 1$  and  $\mathbf{U}^\top \mathbf{U} \preceq \mathbb{I}$ , it follows that  $\mathbf{U}\mathbf{U}^\top \preceq \text{Diag}(\mathbf{e}, \sum_t \mathbf{Z}_t)$ . Therefore,  $(\mathbf{Y} := \mathbf{U}\mathbf{U}^\top, \mathbf{Y}^t := \mathbf{U}_t \mathbf{U}_t^\top, \mathbf{Z})$  is a feasible solution to (7) with an equal cost.
- Let  $(\mathbf{Y}, \mathbf{Y}^t, \mathbf{Z})$  denote a feasible solution to Problem (7). Then, since each  $\mathbf{Y}^t$  is symmetric and rank-one, we can decompose  $\mathbf{Y}^t$  as  $\mathbf{Y}^t = \mathbf{U}_t \mathbf{U}_t^\top$  for a vector  $\mathbf{U}_t$  such that  $U_{i,t} = 0$  if  $Z_{i,t} = 0$ , and concatenate these vectors  $\mathbf{U}_t$  into a matrix  $\mathbf{U}$  such that  $(\mathbf{U}, \mathbf{Z})$  has the same cost in (3) as  $(\mathbf{Y}, \mathbf{Y}^t, \mathbf{Z})$  does in (7). Therefore, it remains to show that  $\mathbf{U}^\top \mathbf{U} = \mathbb{I}$ . To see this, observe that  $\mathbf{Y} \preceq \mathbb{I}$  implies  $(\mathbf{U}_t^\top \mathbf{U}_{t'})^2 = \langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = 0$  if  $t \neq t'$  by Proposition 1.  $\square$

Theorem 1 provides a formulation that is less compact than (3) but contains rank constraints rather than orthogonality constraints. Therefore, it is amenable to exact approaches for addressing sparsity (Bertsimas et al. 2021) and rank (Bertsimas et al. 2022a) constraints.

Furthermore, this formulation already provides valid upper bounds on the objective of (3) by relaxing the rank and sparsity constraints:

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in [0,1]^{p \times r}: \langle \mathbf{E}, \mathbf{Z} \rangle \leq k \\ \mathbf{Y} \in \mathcal{S}_+^p, \mathbf{Y}^t \in \mathcal{S}_+^p \\ \forall t \in [k]}} \max_{\mathbf{Y}, \Sigma} \langle \mathbf{Y}, \Sigma \rangle \text{ s.t. } & \mathbf{Y} = \sum_{t=1}^k \mathbf{Y}^t, \mathbf{Y} \preceq \text{Diag} \left( \mathbf{e}, \sum_t \mathbf{Z}_t \right), \\ & \text{tr}(\mathbf{Y}^t) = 1, \forall t \in [r], |Y_{i,j}^t| \leq M_{i,j} Z_{i,t}, \forall i, j \in [p], t \in [r], \end{aligned} \quad (8)$$

where  $M_{i,i} = 1$  and  $M_{i,j} = 1/2$  if  $i \neq j$  is an upper bound on  $|Y_{i,j}^t|$ , since  $\mathbf{Y}^t$  was, before relaxing the rank constraint, a rank one matrix (c.f. Bertsimas et al. 2022b).

## 2.2. Valid Inequalities for Strengthening the Extended Formulation

In this section, we propose valid inequalities which allow us to improve the quality of the convex relaxation (8) introduced in the previous section. Formally, we have the following result:

**THEOREM 2.** *Let  $\mathcal{P}_{\text{Strengthened}}$  denote the optimal value of the following problem:*

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in [0,1]^{p \times r}: \langle \mathbf{E}, \mathbf{Z} \rangle \leq k \\ \mathbf{Y} \in \mathcal{S}_+^p, \mathbf{Y}^t \in \mathcal{S}_+^p \\ \mathbf{w} \in [0,1]^p}} \max_{\mathbf{Y}, \Sigma} \langle \mathbf{Y}, \Sigma \rangle & \quad (9) \\ \text{s.t. } & \mathbf{Y} \preceq \text{Diag}(\mathbf{w}), \mathbf{Y} = \sum_{t=1}^k \mathbf{Y}^t, \text{tr}(\mathbf{Y}^t) = 1, \mathbf{w} \leq \mathbf{Z}\mathbf{e} \\ & |Y_{i,j}^t| \leq M_{i,j} Z_{i,t} \quad \forall i, j \in [p], t \in [k], \\ & \sum_{j=1}^p Y_{i,j}^t{}^2 \leq Y_{i,i}^t Z_{i,t} \quad \forall i \in [p], t \in [r], \\ & \sum_{j=1}^p Y_{i,j}^2 \leq r Y_{i,i} w_i \quad \forall i \in [p], \end{aligned}$$

$$\begin{aligned} \left( \sum_{j=1}^p |Y_{i,j}| \right)^2 &\leq kY_{i,i}w_i && \forall i \in [p], \\ \sum_{i \in [p]: i \neq j} Y_{i,j}^2 &\leq (k-r+1)w_j(w_j - Y_{j,j}), && \forall j \in [p], \end{aligned}$$

where  $w_i$  models  $\min(1, \sum_{t=1}^r Z_{i,t})$ , and the last four constraints are rotated second-order cone constraints (see, e.g., Alizadeh and Goldfarb 2003).

Further, let  $\mathcal{P}_{\text{Relax}}$  denote the optimal value of Problem (8), and  $\mathcal{P}_{\text{Exact}}$  denote the optimal objective value of Problem (7). Then, for any covariance matrix  $\Sigma$  and any sparsity and rank  $(k, r)$ :

$$\mathcal{P}_{\text{Exact}} \leq \mathcal{P}_{\text{Strengthened}} \leq \mathcal{P}_{\text{Relax}}.$$

*Proof of Theorem 2* The first inequality holds by verifying that any feasible solution to (7) is also a solution to (9) with equal cost. Indeed,  $\sum_{j=1}^p Y_{i,j}^2 \leq Y_{i,i}Z_{i,t}$  follows from aggregating the  $2 \times 2$  minor constraints on  $\mathbf{Y}^t$ ,  $\sum_{j=1}^p Y_{i,j}^2 \leq rY_{i,i}w_i$  follows from aggregating the  $2 \times 2$  minor constraints on  $\mathbf{Y}$  and invoking Lemma 3;  $(Y_{i,j}^t)^2 \leq Y_{i,i}^t Y_{j,j}^t$  and invoking Lemma 2;  $\sum_{j=1}^p Y_{i,j}^2 \leq Y_{i,i}Z_{i,t}$  follows by invoking Cauchy-Schwarz to derive  $\left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq kY_{i,i}$  and invoking Lemma 3;  $\sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k-r+1)w_j(w_j - Y_{j,j})$  follows by aggregating the  $2 \times 2$  minors of  $\text{Diag}(\mathbf{w}) - \mathbf{Y}$  and invoking Lemma 3. We provide a full derivation of these valid inequalities in Section EC.2.

The second inequality holds by observing that (8) is a relaxation of (9).  $\square$

We remark that of the four groups of valid inequalities introduced in Problem (9), the first inequality has been previously stated in the case of sparse PCA with one component by Bertsimas and Cory-Wright (2020), Bertsimas et al. (2022b), Li and Xie (2020), but the three other groups of valid inequalities are, to our knowledge, new.

### 2.3. Strong Inequalities with a Per-Component Sparsity Budget

In this section, we consider specifying a sparsity budget  $k_t$  for each component  $\mathbf{Y}^t$ , in addition to an overall sparsity budget  $k$  (with  $k \leq \sum_{t=1}^r k_t$ ) The primary motivation for this additional assumption is that, as we observe numerically in Section 5, specifying  $k_t$  in addition to  $k$  often yields substantially tighter convex relaxations. This assumption is also common in the literature (e.g., Hein and Bühler 2010, Berk and Bertsimas 2019).

Formally, we have the following result (proof deferred to Section EC.3):

PROPOSITION 2. *Suppose that  $\sum_{i \in [p]} Z_{i,t} \leq k_t$  in Problem (7). Then, the following inequalities hold:*

$$\left( \sum_{j=1}^p |Y_{i,j}^t| \right)^2 \leq k_t Y_{i,i}^t Z_{i,t} \quad \forall i \in [p], \forall t \in [r], \quad (10)$$

$$\sum_{i \in [p]: i \neq j} (Y_{i,j}^t)^2 \leq (k_t - 1) Z_{j,t} (Z_{j,t} - Y_{j,j}^t) \quad \forall j \in [p]. \quad (11)$$

Interestingly, as we observe in Section 5.1, combining Problem (9) with Constraints (10)-(11) often yields much tighter upper bounds than (9) alone, even if we take the worst-case upper bound over all feasible splits  $k_t$ 's which sum to  $k$ .

Now that we introduced the sparsity of each PC,  $k_t$ , we can further tighten our semidefinite relaxations by leveraging valid inequalities obtained in the case of a single PC. For example, for each component  $t \in [r]$ , Kim et al. (2022) observe that the feasible set of all  $k_t$ -sparse components  $\{\mathbf{u} \in \mathbb{R}^p : \|\mathbf{u}\|_0 \leq k_t\}$  is permutation and sign invariant, i.e., for any feasible vector  $\mathbf{u}$ , any vector obtained by permuting or changing the sign of the coordinates of  $\mathbf{u}$  is also feasible. Based on this observation, they propose a lifted formulation for sparse PCA with a single PC, which to the best of our knowledge, leads to the strongest known relaxation for sparse PCA with  $r = 1$  which can be solved in polynomial time. For the sake of concision, we denote  $(\mathbf{Y}^t, \mathbf{Z}^t) \in \mathcal{T}(k_t)$  the set of valid inequalities comprised in their ‘‘T-relaxation’’ (see Problem (EC.5) in Section EC.4.1 for an explicit formulation) and consider the following relaxation:

$$\begin{aligned}
 & \max_{\substack{\mathbf{Z} \in [0,1]^{p \times r}, \mathbf{Y} \in \mathcal{S}_+^p, \mathbf{Y}^t \in \mathcal{S}_+^p \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k, \\ \mathbf{w} \in [0,1]^p}} \max_{\mathbf{Y}, \Sigma} \langle \mathbf{Y}, \Sigma \rangle & (12) \\
 \text{s.t. } & \mathbf{Y} \preceq \text{Diag}(\mathbf{w}), \mathbf{Y} = \sum_{t=1}^k \mathbf{Y}^t, \text{tr}(\mathbf{Y}^t) = 1, \mathbf{w} \leq \mathbf{Z}\mathbf{e}, \\
 & \sum_{j=1}^p Y_{i,j}^2 \leq r Y_{i,i} w_i & \forall i \in [p], \\
 & \left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i} w_i & \forall i \in [p], \\
 & \sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k - r + 1) w_j (w_j - Y_{j,j}) & \forall j \in [p], \\
 & (\mathbf{Y}^t, \mathbf{Z}^t) \in \mathcal{T}(k_t) & \forall t \in [r].
 \end{aligned}$$

REMARK 2 (STRENGTH). The relaxation of Problem (12) dominates that of Problem (9), even when (9) is strengthened with the inequalities proposed in (10)-(11). Indeed, Kim et al. (2022, Theorem 13) can be extended to show that (10)-(11) are redundant in (12). However, this result only holds after including the inequalities we introduced in Section 2.2 which capture notions of orthogonality between the  $\mathbf{Y}^t$ 's in (12). Without these inequalities, neither (12) nor (9) would dominate the other relaxation.

Unfortunately, (12) cannot scale beyond  $p \approx 100$ , at least with current technology, due to the presence of multiple semidefinite matrices and constraints. To solve instances with  $p > 100$  features, we develop a more tractable, albeit less tight, version of the relaxation of (12) in Section EC.4.2, by using second-order cone approximations of the cone of semidefinite matrices and eigenvector cuts, as presented by Bertsimas and Cory-Wright (2020) and references therein.

### 3. A Valid Bound via a Generalization of the Gershgorin Circle Theorem

The Gershgorin circle theorem (c.f. Horn and Johnson 1985, Chapter 6) bounds the largest eigenvalue of a matrix  $\Sigma \in \mathcal{S}_+^p$  via the combinatorial function:

$$\lambda_{\max}(\Sigma) \leq \max_{i \in [p]} \sum_{j \in [p]} |\Sigma_{i,j}|. \quad (13)$$

For sparse PCA with a single PC, several authors (Berk and Bertsimas 2019, Bertsimas et al. 2022b) have leveraged this result to derive an upper bound on (2) that depends on the support of the sparse PC. Moreover, they observed that integrating this upper bound within mixed-integer algorithms for solving (2) to optimality significantly accelerates these algorithms in practice. Motivated by their observations, we now bound the objective value of (3) as a function of the support matrix  $\mathbf{Z}$ .

One naive upper bound is to apply the circle theorem to each PC separately and bound the objective of (3) via the sum of the largest eigenvalues of each sub-matrix of  $\Sigma$  induced by a column of  $\mathbf{Z}$ . However, this approach is too conservative for non-disjoint PCs, since it does not take into account any information about overlap between the support of each PC. Indeed, with fully overlapping support, this approach bounds the sum of the  $r$  largest eigenvalues of the relevant submatrix by  $r$  times the largest eigenvalue of the relevant submatrix. Instead, the valid inequalities we derive in this section rely on a new and non-trivial bound on the variance collectively explained by  $r$  orthogonal PCs. Incidentally, our result leads to the following bound of the sum of the  $r$  largest eigenvalues of a semidefinite matrix  $\Sigma$ :

$$\sum_{t \in [r]} \lambda_t(\Sigma) \leq \max_{\mu \in \{0,1\}^p: \mathbf{e}^\top \mu \leq r} \sum_{i,j \in [p]} \mu_i |\Sigma_{i,j}|, \quad (14)$$

which strictly generalizes (13) and could be of independent interest to the linear algebra community.

Formally, we derive the following generalization of the circle theorem bound, which holds with multiple PCs and a fixed but arbitrary support pattern  $\mathbf{Z} \in \{0,1\}^{p \times r} : \sum_{i \in [p]} Z_{i,t} \geq 1 \ \forall t \in [r]$ . Subsequently, we derive a mixed-integer linear representation of this bound:

**THEOREM 3.** *For any feasible support pattern  $\mathbf{Z} \in \{0,1\}^{p \times r} : \sum_{i \in [p]} Z_{i,t} \geq 1 \ \forall t \in [r]$ , an upper bound on the objective value attained by any matrix  $\mathbf{U}$  such that  $U_{i,t} = 0$  if  $Z_{i,t} = 0 \ \forall i \in [p], \forall t \in [r]$  in Problem (3) is given by:*

$$\max_{\substack{\mu \in \{0,1\}^{p \times r}: \\ \sum_{i \in [p]} \mu_{i,t} = 1, \forall t \in [r], \\ \sum_{t \in [r]} \mu_{i,t} \leq 1 \forall i \in [p]}} \sum_{i,j \in [p]} \sum_{t \in [r]} \mu_{i,t} Z_{i,t} Z_{j,t} |\Sigma_{i,j}|. \quad (15)$$

**REMARK 3.** Taking  $\mathbf{Z} = \mathbf{E}$  in Theorem 3 yields (14). If  $r = 1$ , this bound is equivalent to the circle theorem, and if  $r = p$  it is equivalent to the (known) fact that  $\sum_{t \in [p]} \lambda_t(\Sigma) = \text{tr}(\Sigma) \leq \sum_{i,j \in [p]} |\Sigma_{i,j}|$ . More generally, it shows that the eigenvalues of a positive semidefinite matrix are majorized by the absolute column sums (see Marshall and Olkin 1979, for a general theory of majorization).

When the supports of each PC are disjoint, (15)'s upper bound is equivalent to applying the circle theorem to each PC separately. Alternatively, with fully overlapping support, it reduces to bounding the variance explained by the largest  $r$  column sums of the submatrix selected by  $\mathbf{Z}$ . In the case with partially overlapping support, it systematically interpolates between these bounds.

*Proof of Theorem 3* Fix  $\mathbf{Z} \in \{0, 1\}^{p \times r}$  in Problem (3). Then, it follows directly from Theorem 1 that an upper bound on the objective value attained by any orthogonal matrix  $\mathbf{U}$  such that  $U_{i,t} = 0$  if  $Z_{i,t} = 0$  is given by the following maximization problem:

$$\max_{\mathbf{Y}^t \in \mathcal{S}_+^p \ \forall t \in [r]} \sum_{t \in [r]} \langle \mathbf{Y}^t, \mathbf{\Sigma} \rangle \text{ s.t. } \text{tr}(\mathbf{Y}^t) = 1 \ \forall t \in [r], \sum_{t \in [r]} \mathbf{Y}^t \preceq \mathbb{I}, Y_{i,j}^t = 0 \text{ if } Z_{i,t} = 0 \ \forall i \in [p], t \in [r].$$

To obtain a non-trivial mixed-integer linear representable upper bound as a function of the support pattern  $\mathbf{Z}$ , we now relax this problem. First, we observe that  $Y_{i,j}^t = 0$  if  $Z_{i,t} = 0$  and therefore we can replace  $\mathbf{\Sigma}$  in the objective with  $\text{Diag}(\mathbf{Z}^t) \mathbf{\Sigma} \text{Diag}(\mathbf{Z}^t)$  without loss of generality, where  $\text{Diag}(\mathbf{Z}^t)$  is a diagonal matrix with on-diagonal entries specified by the  $t$ th column of  $\mathbf{Z}$ . Further relaxing the problem by omitting the logical constraints then gives the following semidefinite upper bound:

$$\max_{\mathbf{Y}^t \in \mathcal{S}_+^p \ \forall t \in [r]} \sum_{t \in [r]} \langle \mathbf{Y}^t, \text{Diag}(\mathbf{Z}^t) \mathbf{\Sigma} \text{Diag}(\mathbf{Z}^t) \rangle \text{ s.t. } \text{tr}(\mathbf{Y}^t) = 1 \ \forall t \in [r], \sum_{t \in [r]} \mathbf{Y}^t \preceq \mathbb{I}.$$

Moreover, strong duality holds between this problem and its dual problem, namely:

$$\min_{\mathbf{U} \in \mathcal{S}_+^p, \mathbf{s} \in \mathbb{R}^r} \text{tr}(\mathbf{U}) + \mathbf{e}^\top \mathbf{s} \text{ s.t. } \mathbf{U} + s_t \mathbb{I} \succeq \text{Diag}(\mathbf{Z}^t) \mathbf{\Sigma} \text{Diag}(\mathbf{Z}^t) \ \forall t \in [r].$$

To obtain a linear, rather than semidefinite, upper bound from this problem, we restrict  $\mathbf{U}$  to be a diagonal matrix and  $\mathbf{U} + s_t \mathbb{I} - \text{Diag}(\mathbf{Z}^t) \mathbf{\Sigma} \text{Diag}(\mathbf{Z}^t)$  to be contained within the cone of diagonally dominant matrices, which is an inner approximation of the positive semidefinite cone (see also Barker and Carlson 1975, Ahmadi et al. 2017, for detailed studies of this inner approximation). This gives the following upper bound:

$$\min_{\mathbf{u} \in \mathbb{R}_+^p, \mathbf{s}} \mathbf{e}^\top \mathbf{u} + \mathbf{e}^\top \mathbf{s} \text{ s.t. } u_i + s_t \geq \sum_{j \in [p]} Z_{i,t} Z_{j,t} |\Sigma_{i,j}| \ \forall i \in [p], \forall t \in [r].$$

Finally, we invoke strong duality and use the fact that some (binary) extreme point in the dual problem must be dual-optimal, to verify that the above problem attains the same value as:

$$\max_{\substack{\boldsymbol{\mu} \in \{0,1\}^{p \times r}: \\ \sum_{i \in [p]} \mu_{i,t} = 1 \ \forall t \in [r], \\ \sum_{t \in [r]} \mu_{i,t} \leq 1 \ \forall i \in [p]}} \sum_{i,j \in [p]} \sum_{t \in [r]} \mu_{i,t} Z_{i,t} Z_{j,t} |\Sigma_{i,j}|. \quad \square$$

Observe that the proof of Theorem 3 involves invoking strong duality and taking a finitely generated inner approximation of the positive semidefinite cone. This is quite different to existing proofs of the Gershgorin circle theorem, which usually leverage properties of eigenvectors and therefore cannot easily be generalized. Thus, our proof technique could also be useful in other contexts, e.g., in sparse canonical correlation analysis (Witten et al. 2009).

To represent Theorem 3's upper bound as a mixed-integer linear system, we introduce the auxiliary variables  $\rho_{i,t} \forall i \in [p], t \in [r]$  to model the column sum  $\sum_{j \in [p]} Z_{j,t} |\Sigma_{i,j}|$  if  $\mu_{i,t} = 1$  and equal 0 if  $\mu_{i,t} = 0$ . This allows us to represent Theorem 3's upper bound via the following system,

$$\begin{aligned}
\theta &= \sum_{i,t} \rho_{i,t}, \\
\rho_{i,t} &= \begin{cases} \sum_{j \in [p]} Z_{j,t} |\Sigma_{i,j}| & \text{if } \mu_{i,t} = 1 \\ \rho_{i,t} = 0 & \text{if } \mu_{i,t} = 0 \end{cases} \quad \forall i \in [p], t \in [r], \\
\sum_{i \in [p]} \mu_{i,t} &= 1 \quad \forall t \in [r], \\
\sum_{t \in [r]} \mu_{i,t} &\leq 1 \quad \forall i \in [p], \\
\mu_{i,t} &\leq Z_{i,t} \quad \forall i \in [p], t \in [r], \\
\mu_{i,t} &\in \{0, 1\} \quad \forall i \in [p], t \in [r],
\end{aligned} \tag{16}$$

where  $\theta$  is an upper bound on the largest objective value obtainable with a given support pattern  $\mathbf{Z}$ , and we omit the term  $Z_{i,t}$  from the partial sum  $\sum_{j \in [p]} Z_{j,t} |\Sigma_{i,j}|$  by imposing the constraint  $\mu_{i,t} \leq Z_{i,t}$  because if  $Z_{i,t} = 0$  then the  $(i,t)$ th column sum is zero and can be omitted from the bound without loss of generality.

In the above system, we replace the maximization term in Problem (15) with the requirement that there exists some  $\boldsymbol{\rho}, \boldsymbol{\mu}$  such that the above system is feasible, because feasibility at value  $\theta$  implies that  $\theta$  is indeed no larger than (15)'s upper bound. Therefore, we can implement this system of inequalities within an optimization problem by requiring that  $\langle \boldsymbol{\Sigma}, \mathbf{Y} \rangle \leq \sum_{i \in [p], t \in [r]} \rho_{i,t}$ .

In the next section, we propose utilizing the above bound by combining it with a greedy rounding mechanism (Section 4.2). We subsequently observe in numerical experiments that this improves the performance of the rounding mechanism (Section EC.5.4), which reveals that in addition to being theoretically interesting, the combinatorial bound derived in this section is practically useful.

## 4. Algorithmic Strategies

In this section, we propose three numerical strategies that provide high-quality solutions to Problem (3). First, in Section 4.1, we propose a formulation of Problem (3) which is amenable to existing spatial branch-and-bound codes. Second, in Section 4.2, we propose a rounding mechanism that converts a solution to one of the convex relaxations from Section 2 into a certifiably near-optimal solution to Problem (3). Moreover, we observe that it can be used to warm-start the branch-and-bound scheme proposed in Section 4.1. Finally, in Section 4.3, we propose an iterative deflation heuristic that exploits the relative maturity of sparse PCA technology in the rank-one case to obtain high-quality PCs in the rank- $r$  case for  $r > 1$ .

#### 4.1. Exact Non-Convex Formulation with Warmstart and Presolving

The sparse PCA problem with multiple PCs, either in its original formulation (3) or its equivalent reformulation (7), can be seen as a non-convex mixed-integer quadratically constrained problem— for (7), the rank constraints can be encoded as non-convex quadratic constraints  $(\mathbf{Y}^t)^2 = \mathbf{Y}^t$  or  $\mathbf{Y}^t = \mathbf{U}_t \mathbf{U}_t^\top$ . However, current non-convex MIQCP solvers cannot handle SDP variables and constraints. Accordingly, we devise a strategy to solve our sparse PCA problem exactly using formulation (3).

Since feasible solutions are extremely challenging for spatial branch-and-bound solvers to recover when quadratic equality constraints are imposed exactly, especially when these solutions are irrational or of exponential size (c.f. Ramana 1997, Bienstock et al. 2023), we relax the constraint  $\mathbf{U}^\top \mathbf{U} = \mathbb{I}$  to require that it is satisfied to within an elementwise tolerance of  $\epsilon$ . This gives:

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in \{0,1\}^{p \times r}: \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k}} \max_{\mathbf{U} \in \mathbb{R}^{p \times r}} \quad & \langle \mathbf{U} \mathbf{U}^\top, \boldsymbol{\Sigma} \rangle \\ \text{s.t.} \quad & \|\mathbf{U}^\top \mathbf{U} - \mathbb{I}\|_\infty \leq \epsilon, \\ & U_{i,t} = 0 \text{ if } Z_{i,t} = 0, \quad \forall i \in [p], t \in [r]. \end{aligned} \tag{17}$$

We set  $\epsilon = 10^{-4}/r^2$  so that the total constraint violation does not exceed  $10^{-4}$ . Problem (17) is a non-convex quadratically constrained mixed-integer problem with  $pr$  continuous variables,  $pr$  binaries, and  $r^2$  quadratic constraints.

In addition, we strengthen Problem (17) with valid inequalities derived from the  $\ell_1$  relaxation of sparse PCA, as explored by Dey et al. (2022b,a). Indeed, if the sparsity of each PC,  $k_t$ , is specified a priori, we have the valid inequalities

$$\|\mathbf{U}_t\|_1 \leq \sqrt{k_t}, \quad \forall t \in [r]. \tag{18}$$

Moreover, if  $k$  is specified but  $k_t$  is not, we instead impose the second-order cone inequalities

$$\|\mathbf{U}_t\|_1^2 \leq \sum_{i \in [p]} Z_{i,t}, \quad \forall t \in [r], \tag{19}$$

which allows us to model  $k_t = \sum_{j=1}^p Z_{j,t}$  in a tractable fashion.

In practice, this approach allows MIQCP solvers to solve Problem (7) to optimality for  $pr < 100$  (Section 5.2) and obtain high-quality solutions at larger problem sizes. Note that we avoid mixing both sets of inequalities, as we observed in some preliminary numerical experiments that this sometimes induces numerical instability. To further improve the performance of branch-and-bound, we accelerate its convergence by using the solution generated by Algorithm 1 as a warm-start everywhere except where explicitly stated otherwise.

As spatial branch-and-bound technology improves over time, we believe that it should be possible to solve Problem (17) exactly at larger problem sizes. Indeed, recent works, e.g., Dong and Luo



(2018) and Gupta et al. (2023), solve some quadratically constrained problems with up to 50 variables to optimality using custom branch-and-bound solvers, and Gupta et al. (2023) reports that **Gurobi**’s off-the-shelf QCQP solver has achieved a machine independent speedup factor of 67.5 in less than two years, which suggests that larger instances of (20)–(17) may soon be in reach.

In Section EC.5.4, we consider the possibility of improving our non-convex formulation by combining it with the combinatorial upper bound derived in Section 3. Unfortunately, introducing this bound does more harm than good, because it introduces more decision variables into the formulation, and the cost of introducing these variables outweighs the benefits of having a tighter combinatorial bound. Therefore, we do not consider utilizing this bound within branch-and-bound elsewhere in the paper. However, the combinatorial bound is useful in practice when combined with a rounding scheme, as we discuss in the next section.

#### 4.2. Feasible Solutions from the Relaxation via Greedy Disjoint Rounding

In this section, we develop a rounding mechanism that converts an optimal solution to a convex relaxation (see Section 2) into a high-quality feasible solution. Historically, a useful strategy for similar integer optimization problems has been to (a) solve a convex relaxation in  $(\mathbf{Z}, \mathbf{Y})$ , (b) greedily round  $\mathbf{Z}^*$ , the solution to the relaxation, to obtain a feasible binary matrix  $\hat{\mathbf{Z}}$  that is close to  $\mathbf{Z}^*$ , and (c) resolve for  $\mathbf{U}$  under the constraints  $U_{i,t} = 0$  if  $\hat{Z}_{i,t} = 0$  (c.f. Bertsimas et al. 2022b).

Unlike in the case with a single PC, observe that the “resolve” step (c) is non-trivial. Namely, solving for  $\mathbf{U}$  for some arbitrary and fixed sparsity pattern  $\hat{\mathbf{Z}}$ , i.e., solving

$$\max_{\mathbf{U} \in \mathbb{R}^{p \times r}} \langle \mathbf{U}\mathbf{U}^\top, \boldsymbol{\Sigma} \rangle \text{ s.t. } \mathbf{U}^\top \mathbf{U} = \mathbb{I}, U_{i,t} = 0 \text{ if } \hat{Z}_{i,t} = 0, \forall i \in [p], t \in [r], \quad (20)$$

cannot be done in closed form in general. When  $\hat{\mathbf{Z}}$  corresponds to fully overlapping or fully disjoint supports, however, we can obtain the solution of (20) via an eigenvalue decomposition of the corresponding submatrix/submatrices.

Therefore, we propose in Algorithm 1 a relax-round-and-resolve strategy where the rounding step (b) generates a solution with fully disjoint supports, i.e., where  $\sum_{t \in [r]} \hat{Z}_{i,t} \leq 1, \forall i \in [p]$ .

The rounding step involves a weight parameter  $\lambda$  that controls the importance of obtaining a solution that is close to the solution of the relaxation ( $\lambda = 0$  corresponding to greedy rounding) vs. obtaining a solution that maximizes the upper-bound (16). Numerically, we find that having  $\lambda > 0$  is particularly useful when the individual sparsity budgets  $k_t$  are identical, in order to break the degeneracy of our relaxation (in this case, the relaxation (12) returns a solution with  $Z_{i,t}^* = Z_{i,t'}^*, \forall t, t'$ ). In particular, in Section EC.5.4, we observe on the `pitprops` dataset that across  $r \in \{2, 3, \dots, 6\}$  and  $k_t \in \{2, 4, \dots, 10\}$ , setting  $\lambda = 1$  improves the average objective value obtained by rounding by 20.14% compared to setting  $\lambda = 0$  before initiating the rounding step. Accordingly, we set  $\lambda = 1$  throughout our numerical experiments, except where explicitly stated otherwise.

Since  $\hat{\mathbf{Z}}$  encodes for disjoint supports, any matrix  $\mathbf{U}$  with support  $\hat{\mathbf{Z}}$  automatically satisfies the orthogonality constraint. We can obtain  $\mathbf{U}$  solution of (20) by solving for each PC independently. For each  $t \in [r]$ , we consider the submatrix of  $\mathbf{\Sigma}$  over the indices  $\{i : \hat{Z}_{i,t} = 1\}$ , extract its leading eigenvector via SVD, and pad it with zeros to construct  $\mathbf{U}_t$ . Although the restriction to disjoint support is not without loss of optimality, we will observe numerically in Section 5 that disjoint solutions are not particularly suboptimal for Problem (3) when  $k$  and  $r$  are small relative to  $p$ —an observation already made by Asteris et al. (2015).

We close this section by remarking that Algorithm 1 could be adapted to exploit any sparsity structure which makes Problem (20) tractable; for instance, fully overlapping support, or a combination of fully overlapping and fully disjoint supports.

---

**Algorithm 1** A disjoint greedy rounding method for Problem (7)

---

**Require:** Covariance matrix  $\mathbf{\Sigma}$ , rank parameter  $r$ , sparsity parameter  $k$ , weight  $\lambda$

Compute  $\mathbf{Z}^*$  solution of (12) or (EC.6)

Construct  $\hat{\mathbf{Z}} \in \{0, 1\}^{p \times r}$  solution of

$$\begin{aligned} \max_{\mathbf{Z} \in \{0, 1\}^{p \times r}} \langle \mathbf{Z}, \mathbf{Z}^* \rangle + \lambda \theta \quad \text{s.t.} \quad & \sum_{i=1}^p Z_{i,t} \geq 1, \quad \forall t \in [r], \langle \mathbf{e}, \mathbf{Z} \rangle \leq k, \\ & \sum_{t=1}^r Z_{i,t} \leq 1, \quad \forall i \in [p], \\ & \theta \text{ satisfying (16)}. \end{aligned}$$

Compute  $\mathbf{U}$  solution of (20) via SVD

**return**  $\mathbf{Z}, \mathbf{U}$ .

---

### 4.3. An Iterative Deflation Heuristic

In this section, we propose a local improvement technique which identifies near-optimal and near-feasible solutions to Problem (3). The technique is based on the theory of Lagrangian relaxations (see Geoffrion 1974), which argues that if a non-convex problem is decomposable as a sum of easier (but still non-convex) subproblems with a coupling constraint, a good strategy is often to penalize the coupling constraint in the objective and iteratively solve the non-convex subproblems with different penalty multipliers on the coupling constraint. For example, Lu and Zhang (2012) propose an augmented Lagrangian method for solving an  $\ell_1$  relaxation of the sparse PCA, where the sparsity constraints are replaced by an  $\ell_1$  penalty in the objective.

To apply this perspective to our sparse PCA problem with multiple PCs, we first rewrite (7) without the variable  $\mathbf{Y} = \sum_{t \in [r]} \mathbf{Y}^t$  as:

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in \{0,1\}^{p \times r}: \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k}} \max_{\mathbf{Y}^t \in \mathcal{S}_+^p} \sum_{t \in [r]} \langle \mathbf{Y}^t, \boldsymbol{\Sigma} \rangle \text{ s.t. } & \sum_{t \in [r]} \mathbf{Y}^t \preceq \mathbb{I} \\ & \text{tr}(\mathbf{Y}^t) = 1, \forall t \in [r], \\ & Y_{i,j}^t = 0 \text{ if } Z_{i,t} = 0, \forall t \in [r], i, j \in [p], \\ & \text{Rank}(\mathbf{Y}^t) = 1, \forall t \in [r]. \end{aligned} \quad (21)$$

Hence, Problem (21) is the sum of  $r$  rank-1 sparse PCA problems coupled via linear constraints on their respective supports  $\mathbf{Z}_t$  and the semidefinite constraint  $\sum_{t \in [r]} \mathbf{Y}^t \preceq \mathbb{I}$ . Therefore, if the coupling constraint can be handled appropriately then this problem can be addressed via scalable methods for rank-one sparse PCA, as reviewed in the introduction. To this end, we assume that a sparsity budget  $k_t$  is imposed for each component as in Section 2.3, i.e.,  $\sum_{i \in [p]} Z_{i,t} \leq k_t, \forall t \in [r]$ , and invoke Proposition 1 to replace the orthogonality constraint with  $r(r-1)/2$  bilinear scalar constraints  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = 0, \forall t' \neq t$ . Finally, since  $\langle \mathbf{X}, \mathbf{W} \rangle \geq 0$  for any positive semidefinite matrices  $\mathbf{X}, \mathbf{W}$  of the same size, we replace  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = 0, \forall t' \neq t$  with  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle \leq 0, \forall t' \neq t$  without loss of generality. Therefore, for any penalty  $\lambda > 0$  we have the following valid Lagrangian relaxation:

$$\begin{aligned} \max_{\substack{\mathbf{Z} \in \{0,1\}^{p \times r}: \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k}} \max_{\mathbf{Y}^t \in \mathcal{S}_+^p} \sum_{t \in [r]} \langle \mathbf{Y}^t, \boldsymbol{\Sigma} \rangle - \lambda \sum_{t, t' \in [r]: t \neq t'} \langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle \text{ s.t. } & \text{tr}(\mathbf{Y}^t) = 1, \forall t \in [r], \\ & Y_{i,j}^t = 0 \text{ if } Z_{i,t} = 0, \forall t \in [r], i, j \in [p], \\ & \text{Rank}(\mathbf{Y}^t) = 1, \forall t \in [r], \\ & \sum_{i \in [p]} Z_{i,t} \leq k_t, \forall t \in [r]. \end{aligned} \quad (22)$$

Given an index  $t$  and a sparsity budget  $k_t$ , optimizing for  $\mathbf{Y}^t$  (with all other  $\mathbf{Y}_{t'}, t' \neq t$ , and  $\lambda$  fixed) is equivalent to finding the leading  $k_t$ -sparse PC of the matrix  $\boldsymbol{\Sigma} - \lambda \sum_{t' \neq t} \mathbf{Y}^{t'}$ . Since  $\text{tr}(\mathbf{Y}^t) = 1$ , we can add a constant term  $\lambda_{\text{offset}} \text{tr}(\mathbf{Y}^t)$  to the objective without impacting the optimal solution. Accordingly, we pick  $\lambda_{\text{offset}} > 0$  to be sufficiently large that the matrix  $\boldsymbol{\Sigma} - \lambda \sum_{t' \neq t} \mathbf{Y}^{t'} + \lambda_{\text{offset}} \mathbb{I}$  is positive semidefinite, as required by most sparse PCA algorithms for  $r = 1$ . In our implementation, we solve these subproblems approximately using the truncated power method of Yuan and Zhang (2013). Algorithm 2 proceeds by optimizing for each  $\mathbf{Y}^t$  sequentially and then iteratively increasing the penalty parameter  $\lambda > 0$  to improve the orthogonality of the resulting PCs.

Regarding the penalty parameter  $\lambda$ , we increase it progressively via update rules of the form  $\lambda_{\ell+1} \leftarrow \lambda_{\ell} + \alpha_{\ell} \delta_{\ell}$ . Typically, one can take  $\delta_{\ell}$  equal to the gradient of the Lagrangian with respect to  $\lambda$ ,  $\sum_{t \neq t'} \langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle$ , or as a scaling factor between the two concurrent objectives,  $\sum_{t \in [r]} \langle \mathbf{Y}^t, \boldsymbol{\Sigma} \rangle / \sum_{t, t' \in [r]: t \neq t'} \langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle$ . In our implementation, we use the former during the first

**Algorithm 2** Lagrangian Alternating Minimization for Problem (3)**Require:** Matrix  $\Sigma$ , rank parameter  $r$ , sparsity parameters  $k_1, \dots, k_r$ , number of iterations  $L$ **Require:** Update scheme  $\{\lambda_\ell\}_{\ell \in [L]}$  $\ell \leftarrow 1$ **repeat** $t \leftarrow 1$ **repeat**

$$\lambda_{\text{offset}} \leftarrow \epsilon - \lambda_{\min} \left( \Sigma - \lambda_\ell \sum_{t' \in [r]: t' \neq t} \mathbf{Y}_{t'} \right)$$

Compute  $\mathbf{Y}^t$  (approximate) solution of

$$\max_{\substack{\mathbf{z} \in \{0,1\}^p \\ \mathbf{e}^\top \mathbf{z} \leq k_t}} \max_{\mathbf{Y} \in \mathcal{S}_+^p} \left\langle \Sigma - \lambda_\ell \sum_{t' \in [r]: t' \neq t} \mathbf{Y}_{t'} + \lambda_{\text{offset}} \mathbb{I}, \mathbf{Y} \right\rangle \text{ s.t. } \text{tr}(\mathbf{Y}) = 1, \text{Rank}(\mathbf{Y}) = 1, \\ Y_{i,j} = 0 \text{ if } z_i = 0 \ \forall i, j \in [p].$$

 $t \leftarrow t + 1$ **until**  $t > r$  $\ell \leftarrow \ell + 1$ **until**  $\ell > L$ Return  $\mathbf{Y} = \sum_{t \in [r]} \mathbf{Y}^t$ 

iterations ( $\ell \leq \lfloor 0.15L \rfloor$ ), and the latter (which is typically larger) at later stages. We fix  $\alpha_\ell = 0.01$  for  $\ell \leq \lfloor 0.75L \rfloor$ , and 0.05 otherwise. Finally, for each PC  $t \in [r]$ , we set the initial value  $\lambda_0^t$  to the value of  $\langle \mathbf{Y}^t, \Sigma \rangle$  at the first iteration.

## 5. Numerical Results

In this section, we evaluate the algorithmic strategies derived in the previous two sections, implemented in Julia 1.9 using JuMP.jl 1.12.0, Gurobi version 10.0.0 to solve all non-convex quadratically constrained problems, and Mosek 10.1.11 to solve all conic relaxations. For the sake of conciseness, we defer full details of our experimental setup to Section EC.5.1.

For the purpose of averaging results across datasets with different  $p$ 's, we report the proportion of variance explained whenever we report an objective value. For a correlation matrix, this corresponds to dividing by  $p$ , the number of features. We make our code freely available on [GitHub](https://github.com/ryancorywright/MultipleComponentsSoftware) at [github.com/ryancorywright/MultipleComponentsSoftware](https://github.com/ryancorywright/MultipleComponentsSoftware).

*Description of Data Sources:* We perform experiments on eleven datasets from the UCI database in Sections 5.1-5.2 and 5.4, and experiments on synthetic data in Section 5.3 (see therein for details).

Of the eleven datasets (described in detail in Section EC.5.2), six datasets are overdetermined (meaning  $n > p$ ), while five datasets are underdetermined (meaning  $p > n$ ).

### 5.1. Performance of Upper Bounds

In this section, we compare the upper bounds from our extended formulation and permutation invariant semidefinite relaxations—(9) with (10)-(11), hereafter “Extended-Ineq”, and (12), hereafter “Perm-Ineq”—against the bound obtained from the exact formulation (17) strengthened with the  $\ell_1$  valid inequalities (18)-(19) after a time limit of 7200s, hereafter “Branch-and-Bound”. We also explore the scalability of these formulations, and their conic relaxations.

*Benchmarking on Pitprops Data* We first compare the bounds generated by each method—in terms of proportion of correlation explained—on the UCI `pitprops` dataset ( $p = 13$ ) as we vary  $r \in \{2, 3\}$  and  $k$ , in Table 1. We consider both imposing an overall sparsity budget alone (denoted by “ $k, -$ ”) and imposing a separate budget for each PC, denoted by “ $k, (k_1, \dots, k_t)$ ”. We set the `Gurobi` parameters `FuncPieceError` and `FuncPieceLength` to  $10^{-6}$  and  $10^{-5}$  respectively (their minimum possible values; see `Gurobi` API’s documentation), and report the final upper bound. Note that we do not provide `Gurobi` with a warm-start for this set of experiments.

On the instances presented in Table 1, we observe that branch-and-bound on our non-convex formulation (17) terminates within minutes and provides the tightest (i.e., smallest) upper bound. In comparison, on the same instances, the conic relaxations terminate in less than a second, while providing upper bounds—especially, for “Perm-Ineq”—often only weaker at the third decimal.

However, further experiments on the same dataset show that `Gurobi`’s upper bound does not scale as well as our conic relaxations when the number of PCs  $r$  increases (see Tables EC.4-EC.5 in Section EC.5.4). For  $r \in \{4, 5, 6\}$ , `Gurobi`’s upper bound after ten minutes routinely exceeds 1 (which is a trivial upper bound). On the other hand, the upper bound from the Perm-Ineq relaxation is typically accurate to the first two decimal places on these instances. Accordingly, in the rest of this paper, we only use our convex relaxations to certify the quality of a solution.

Regarding the two semidefinite relaxations “Extended-Ineq” and “Perm-Ineq”, we remind the reader that Perm-Ineq cannot compute a bound if we specify an overall sparsity budget  $k$  only. Extended-Ineq’s bound for a given  $k$  is much weaker than the worst-case bound over all possible allocations of  $k_t$ ’s that sum to  $k$ . This can be explained by the strong second-order cone inequalities (10)-(11) that can be added in the latter case. Therefore, time permitting, we recommend computing the lower bound by solving the relaxations for all possible allocations of  $k_t$  and taking the worst-case bound. Furthermore, when individual sparsity budgets  $k_t$  are given, we observe that Perm-Ineq provides uniformly and sometimes significantly tighter bounds than Extended-Ineq. Accordingly, in the rest of the paper, we only consider instances of Problem (3) where we know both  $k$  and  $k_t$  and consider Perm-Ineq and its second-order cone relaxations, but not Extended-Ineq.

Rank ( $r$ )	Sparsity ( $k, k_t$ )	Extended-Ineq		Perm-Ineq		Branch-and-Bound		
		UB	T(s)	UB	T(s)	UB	Nodes	T(s)
2	4, -	0.297	20.28	-	-	<b>0.295</b>	5,300	13.95
2	4, (1, 3)	<b>0.267</b>	20.58	<b>0.267</b>	1.97	<b>0.267</b>	1,600	4.03
2	4, (2, 2)	<b>0.295</b>	0.37	<b>0.295</b>	0.47	<b>0.295</b>	3,600	12.2
2	6, -	0.384	0.78	-	-	<b>0.371</b>	32,800	28.00
2	6, (1, 5)	<b>0.339</b>	0.44	<b>0.339</b>	0.25	<b>0.339</b>	10,300	6.15
2	6, (2, 4)	<b>0.371</b>	0.47	<b>0.371</b>	0.57	<b>0.371</b>	9,900	24.01
2	6, (3, 3)	0.361	0.42	<b>0.360</b>	0.45	<b>0.360</b>	18,300	34.80
2	8, -	0.451	0.75	-	-	<b>0.435</b>	67,200	52.85
2	8, (1, 7)	<b>0.384</b>	0.46	<b>0.384</b>	0.4	<b>0.384</b>	332,500	38.79
2	8, (2, 6)	<b>0.435</b>	0.43	<b>0.435</b>	0.48	<b>0.435</b>	9,700	18.93
2	8, (3, 5)	0.420	0.52	<b>0.418</b>	0.65	<b>0.418</b>	36,300	56.20
2	8, (4, 4)	0.412	0.48	0.408	0.54	<b>0.404</b>	109,500	195.6
2	10, -	0.490	0.72	-	-	<b>0.458</b>	611,000	391.6
2	10, (1, 9)	<b>0.395</b>	0.43	<b>0.395</b>	0.3	<b>0.395</b>	5,333,000	585.4
2	10, (2, 8)	<b>0.457</b>	0.53	<b>0.457</b>	0.5	<b>0.457</b>	13,200	14.61
2	10, (3, 7)	0.461	0.41	0.459	0.5	<b>0.458</b>	34,300	37.22
2	10, (4, 6)	0.458	0.44	0.455	0.6	<b>0.451</b>	364,600	306.5
2	10, (5, 5)	0.453	0.55	0.449	0.45	<b>0.439</b>	1,814,900	775.5
3	6, -	0.443	25.76	-	-	<b>0.435</b>	201,300	324.2
3	6, (1, 1, 4)	<b>0.380</b>	42.92	<b>0.380</b>	5.02	<b>0.380</b>	9,000	11.60
3	6, (1, 2, 3)	<b>0.412</b>	0.94	<b>0.412</b>	1.54	<b>0.412</b>	13,600	10.49
3	6, (2, 2, 2)	<b>0.435</b>	0.68	<b>0.435</b>	2.65	<b>0.435</b>	22,400	27.36
3	9, -	0.570	2.10	-	-	<b>0.539</b>	2,334,600	> 7200
3	9, (1, 1, 7)	<b>0.461</b>	0.80	<b>0.461</b>	1.21	<b>0.461</b>	41,300	10.58
3	9, (1, 2, 6)	<b>0.512</b>	0.84	<b>0.512</b>	0.59	<b>0.512</b>	15,200	8.08
3	9, (1, 3, 5)	0.497	0.71	<b>0.495</b>	0.80	<b>0.495</b>	95,900	27.18
3	9, (1, 4, 4)	0.489	0.65	0.485	0.54	<b>0.481</b>	329,500	142.6
3	9, (2, 2, 5)	<b>0.539</b>	0.78	<b>0.539</b>	0.73	<b>0.539</b>	70,400	52.83
3	9, (2, 3, 4)	0.532	0.77	0.531	0.98	<b>0.530</b>	161,400	171.0
3	9, (3, 3, 3)	0.520	0.88	0.512	0.58	<b>0.511</b>	735,900	680.0

**Table 1** Performance of upper bounds on the pitprops dataset ( $p = 13$ ), as we vary the overall sparsity ( $k$ ), the number of PCs ( $r$ ) and the allocation of a sparsity budget to the different PCs. We denote the best performing solution (least upper bound) in bold. Note that all results are normalized by dividing by the trace of  $\Sigma$ , i.e.,  $p$ , the number of features, to report results in terms of the proportion of variance explained.

*Benchmarking on Larger-Scale Datasets* We now investigate the scalability of the relaxation Perm-Ineq and its second-order cone relaxations on larger UCI datasets. Table EC.2 (see Section EC.5.3) compares the performance of the original semidefinite formulation (12), its second-order cone relaxation (EC.6) and up to 50 PSD cuts (SOC-Cuts) as in Remark EC.1, and the second-order cone relaxation alone (EC.6) (SOC). Note that if  $p > 1000$  we only include sets of second-order cone constraints with fewer than  $O(p^2)$  members, to avoid excessively memory-intensive problems.

We observe that the PSD relaxation can be solved within a few minutes for  $p \approx 50$  but quickly requires a prohibitive amount of memory and time at higher dimensions. The SOC-Cuts relaxation scales up to  $p \approx 300$  and provides a high-quality upper bound, within 1 – 2% of PSD. Without the additional cuts, the SOC relaxation alone is weak when  $(k, r)$  are large relative to  $p$ . For example,

for the `pitprops` dataset, when  $k_t = (10, 10, 10)$ , SOC returns an upper bound of 1.007 while the proportion of correlation explain can trivially not exceed 1. However, SOC is still preferable in high-dimensional settings, where the semidefinite formulation cannot be solved via an interior point method due to excessive memory requirements, and the formulation SOC-Cuts provides a small improvement in the upper bound at the price of significantly more runtime.

## 5.2. Performance of Feasible Methods

In this section, we numerically evaluate the quality of the three methods developed in Section 4 in terms of their ability to recover approximately orthogonal and high-quality principal components on real-world datasets. We first compare different implementation variants of our methods and validate that they are capable of recovering feasible and near-optimal solutions to small-scale sparse PCA problems with multiple PCs. We then compare our algorithms with four state-of-the-art techniques on eleven UCI datasets. Since Algorithm 2 and all four benchmarked algorithms from the literature require that the sparsity of each PC is specified separately, we consider this formulation in this section (and fix  $k_t = k/r$  for concision).

*Benchmarking on Pitprops Data:* We first investigate the performance of different variants of the three methods presented in Section 4 on the `pitprops` dataset, in order to select the best-performing variants to use for our remaining experiments. We consider: Branch-and-bound with/without the combinatorial bound developed in Section 3 and with/without a warm-start (from Algorithm 1 with  $\lambda = 1$ ); Algorithm 1 with  $\lambda = 0$  (so that it disregards the combinatorial bound developed in Section 3) and with  $\lambda = 1$ ; and Algorithm 2. Algorithm 1 involves solving a convex relaxation from Section 2. We impose a two hour time limit for branch-and-bound. Based on the scalability results presented in the previous section, we use different convex relaxations depending on the dimensionality of the problem. Namely, for Algorithm 1, we use the full semidefinite relaxation (12) if  $p \leq 50$ , the SOC relaxation (EC.6) with 50 PSD cuts if  $p \leq 200$ , and the SOC relaxation (EC.6) otherwise.

Detailed results are reported in Tables EC.3–EC.5. Based on these results, we find that branch-and-bound with a warm-start but without the combinatorial bound (16) performs best among the four branch-and-bound implementations considered, and Algorithm 1 performs best with  $\lambda = 1$ . We only consider these variants of branch-and-bound and Algorithm 1 in the rest of the paper.

It is worth noting that, on the same dataset, Lu and Zhang (2012) extensively compared the performance of existing sparse PCA methods with  $r = 6$  PCs. All six methods they benchmarked required an overall sparsity of around 45–60 to explain less than 70% of the variance. The best performing method could explain 69.55% of the variance with an overall sparsity of 46 (Lu and Zhang 2012, Table 11). They concluded that “there do not exist six highly sparse, nearly orthogonal

and uncorrelated PCs while explaining most of variance”. As reported in Tables EC.3–EC.5, with  $r = 6$  PCs and  $k_t = 2$ , hence an overall sparsity of 12, solutions returned by *any* of our three main methods explain 73%–75% of the variance. Algorithm 2 even provides a solution that explains 81% of the variance with an overall sparsity of  $6 \times 4 = 24$ . That is to say, what was previously considered by the community to be impossible can be done with the techniques in this paper, in seconds.

*Benchmarking on Larger-Scale Datasets* We now investigate the performance of Algorithm 1, Algorithm 2, and branch-and-bound on eleven UCI datasets, whose dimension range from  $p = 13$  (`pitprops`) to  $p = 1300$  (`micromass`). We compare them with four state-of-the-art methods from the literature. Namely,

- The branch-and-bound method of Berk and Bertsimas (2019) for optimally computing one sparse PC, combined with the deflation scheme of Mackey (2008) to obtain multiple PCs, implemented in `Julia` and made available at [github.com/lauren897/Optimal-SPCA](https://github.com/lauren897/Optimal-SPCA). According to Berk and Bertsimas (2019), this method outperformed four others across three UCI datasets ( $r = 3, k = 5$ ).
- The deflation method of Hein and Bühler (2010), using the custom deflation method developed in Bühler (2014), implemented in `Matlab` and made publicly available at [github.com/tbuehler/sparsePCA](https://github.com/tbuehler/sparsePCA), using default parameters. This approach was found by Berk and Bertsimas (2019, Table 9) to be second-best of the methods in their comparison.
- The Lasso-inspired method of Zou et al. (2006), using the `spca` function in the `elasticnet` package version 1.3, using default parameters. This approach is perhaps the most commonly used one in practice, since it is distributed via the ubiquitous `elasticnet` package.
- The covariance thresholding method of Deshpande and Montanari (2014b), building upon the works of Krauthgamer et al. (2015), which relies on applying a soft-thresholding operation first on the entries of the covariance matrix and then on its  $r$  leading eigenvectors. We implemented this method natively in `Julia` and release it as part of our codebase.

We report summary results in Table 2 (see also Tables EC.6–EC.12 in Section EC.5.5 for instance-wise results). We remind the reader that the methods from the literature we benchmark against do not provide any upper bound; we need Algorithm 1 to compute optimality gaps. Also, when the returned solution violates the orthogonality condition, its objective value is not necessarily a valid bound on the objective of (3) and the reported gap may be an optimistic estimate.

First, we observe that our methods are the only one to return PCs that are systematically orthogonal (with an average orthogonality violation  $< 10^{-3}$ ). Among them, we observe that Algorithm 2 and Algorithm 1 perform the best overall, with an average fraction of correlation explained of 0.199 and 0.195 respectively (8.84% and 9.64% optimality gap respectively). Excluding a minority of small-scale instances with  $p \leq 34$ , branch-and-bound is largely dominated by Algorithms 1 and



Method	Obj.	Rel. gap (%)	Viol.	T(s)
Algorithm 1	0.195	9.64%	0.000	2588
Algorithm 2	0.199	8.81%	0.000	61.38
Branch-and-bound	0.187	14.67%	0.000	> 7200
Berk and Bertsimas (2019)	0.205	3.66%	0.031	25.57
Deshpande and Montanari (2014b)	0.197	8.20%	0.094	12.05
Hein and Bühler (2010)	0.180	23.24%	0.023	0.20
Zou et al. (2006)	0.049	80.52%	1.458	4.26

**Table 2** Average performance of methods across the 11 UCI datasets described in Table EC.1 with  $k \in \{5, 10, 20\}, r \in \{2, 3\} : k \leq p$ . Note that the average conic upper bound across these instances is 0.213.

2, both in terms of objective value and computational time. However, our conclusions should be revisited as global non-convex solvers further improve.

Of the remaining methods, the method of Berk and Bertsimas (2019) performs the next best, in terms of explaining a large proportion of the correlation (0.205) while not violating feasibility significantly (0.031). Finally, the methods of Hein and Bühler (2010), Deshpande and Montanari (2014b), Zou et al. (2006) arguably perform less well, because they repeatedly violate the orthogonality constraint, and explain less correlation than Algorithm 2 on average.

Finally, we remark that no one method performs best on every instance. The method of Berk and Bertsimas (2019) performs best on instances where  $k$  is small relative to  $p$  and where disjoint solutions are nearly optimal, in which case the orthogonality constraint can essentially be ignored (avg. relative gap of 3.46% and avg. constraint violation of 0.002 over instances where  $p \geq 101$ ). Algorithm 2, on the other hand, performs best on instances where  $k$  is large relative to  $p$  and the orthogonality constraint is important to account for (avg. relative gap of 3.93% and avg. constraint violation of 0.000 over instances where  $p \leq 54$ ). These results suggest that both  $k$  and the amount of overlap between the optimal PCs impact the performance of each method, and motivate a comparison on synthetic data, where we control the ground truth, in the next section.

### 5.3. Statistical Recovery on Synthetic Data

To evaluate the support recovery ability of each method, we now compare the performance of Algorithms 1 and 2 against the same four methods from the literature on synthetic data. We use a spiked Wishart model with multiple spikes, as in Deshpande and Montanari (2014a), Ding et al. (2023), to generate data. Namely, we consider an underlying covariance matrix of the form  $\Sigma = \mathbf{I}_p + \beta \mathbf{x}_1 \mathbf{x}_1^\top + \beta \mathbf{x}_2 \mathbf{x}_2^\top$ , where  $\beta = 2$  is the signal-to-noise ratio. The vectors  $\mathbf{x}_1, \mathbf{x}_2 \in \{-1, 0, 1\}^p$  are random  $k_{\text{true}}$ -sparse orthogonal vectors. We also control the proportion of overlap between the supports of  $\mathbf{x}_1$  and  $\mathbf{x}_2$ ,  $q \in [0, 1]$  ( $q = 0$  corresponds to disjoint support while  $q = 1$  corresponds to row-sparsity). In our experiments, we take  $p = 50$ ,  $k_{\text{true}} = 20$  and vary  $q \in \{0.1, 0.5, 0.9\}$ . Finally, we sample  $n$  observations from a multivariate centered normal distribution with covariance matrix

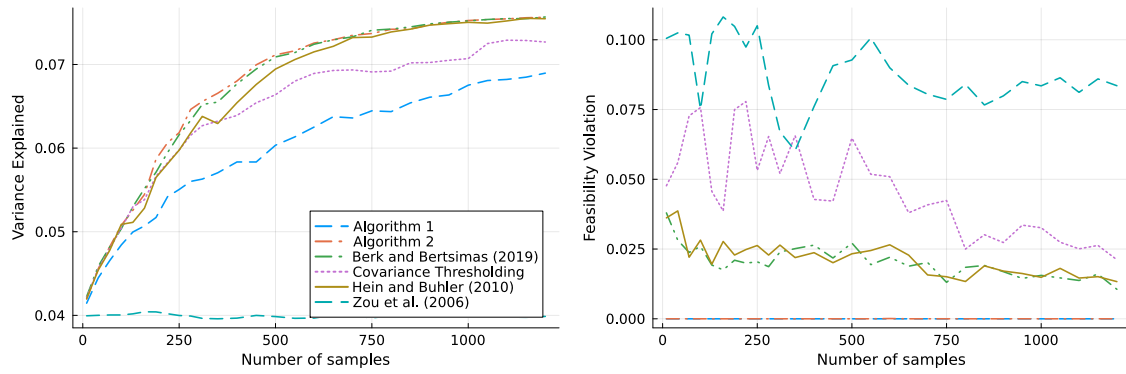
$\Sigma$  and construct the empirical covariance matrix  $\hat{\Sigma}$ . We investigate the performance of different methods as  $n$  increases (so that the empirical covariance matrix converges to the underlying truth,  $\Sigma$ ). We impose a limit of 100 iterations for Algorithm 2.

For each algorithm, we compute the fraction of variance explained and the feasibility violation (i.e., the inner product between the two PCs computed). We average these performance metrics over 20 random instances and report them in Figure 1. The method of Zou et al. (2006) is clearly dominated by all other methods since it explains a significantly lower fraction of the variance, while returning the least orthogonal vectors. In terms of objective value (left panel), we observe that Algorithm 2, Berk and Bertsimas (2019), Hein and Bühler (2010) perform almost identically, followed closely by covariance thresholding. They all explain a larger fraction of the variance than Algorithm 1. However, we observe on the right panel that Algorithms 1 and 2 are the only methods to return orthogonal PCs, across all values of  $n$ . In addition, the gap between the methods (and especially the gap between the four best performing methods) seems to shrink as  $q$  increases, i.e., when the overlap between the support increases.

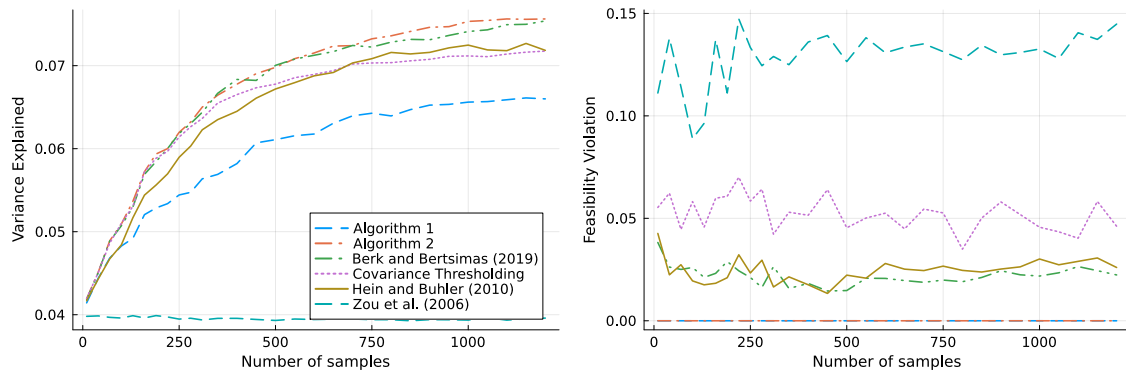
Since the data is synthetically generated, we can also evaluate the ability to recover the true support. For two  $k_{true}$ -sparse candidate PCs  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , we measure how well  $\text{supp}(\mathbf{u}_1) \cup \text{supp}(\mathbf{u}_2)$  recovers the support of  $\text{supp}(\mathbf{x}_1) \cup \text{supp}(\mathbf{x}_2)$  in terms of accuracy and false detection rate:

$$A := \frac{|S \cap S^*|}{|S^*|},$$

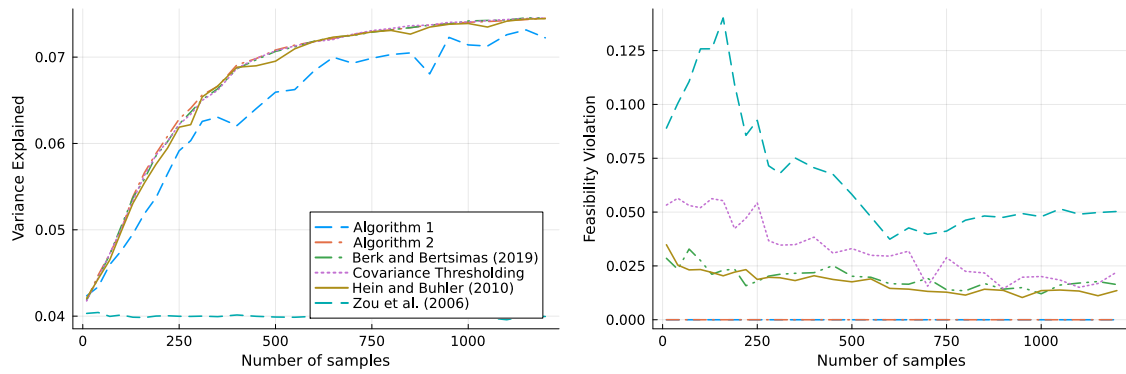
with  $S = \text{supp}(\mathbf{u}_1) \cup \text{supp}(\mathbf{u}_2)$  and  $S^* = \text{supp}(\mathbf{x}_1) \cup \text{supp}(\mathbf{x}_2)$ . This definition of support recovery corresponds to the one used in statistical studies for sparse PCA with multiple PCs (e.g., Deshpande and Montanari 2014b). Figure 2 (left panel) reports the value of  $A$  for different algorithms, including Algorithms 1 and 2, as  $n$  increases. Since we do not explicitly control for the overlap between the returned PCs (except for Algorithm 1), methods might differ in the size of the support they return,  $|S|$ . Hence, to allow for a fair comparison, we also report  $|S|$  in the right panel of Figure 2. Regarding  $A$ , we observe that Algorithm 1 detects a noticeably higher fraction of the true features than other methods, which is not surprising given the fact that it returns disjoint supports. For the remaining methods, their relative performance is aligned with their performance in terms of fraction of variance explained (Figure 1, left panel). It is interesting to observe that Algorithm 1 does not systematically return a support of size  $2k_{true}$ . As the number of observations increases, it detects that some of these features are not needed and sets more coordinates to zero than what is encoded in the binary variable  $\mathbf{Z}$ , leading to a support size  $|S|$  closer to  $|S^*|$ .



(a) Support overlap  $q = 0.1$

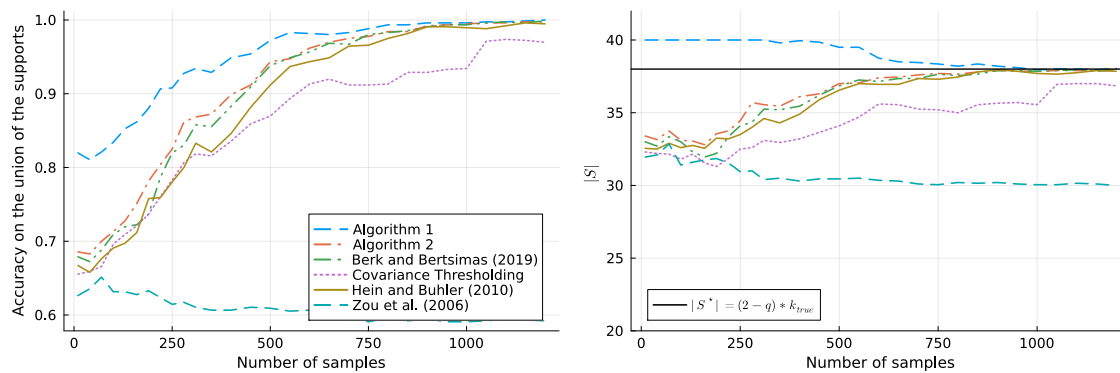
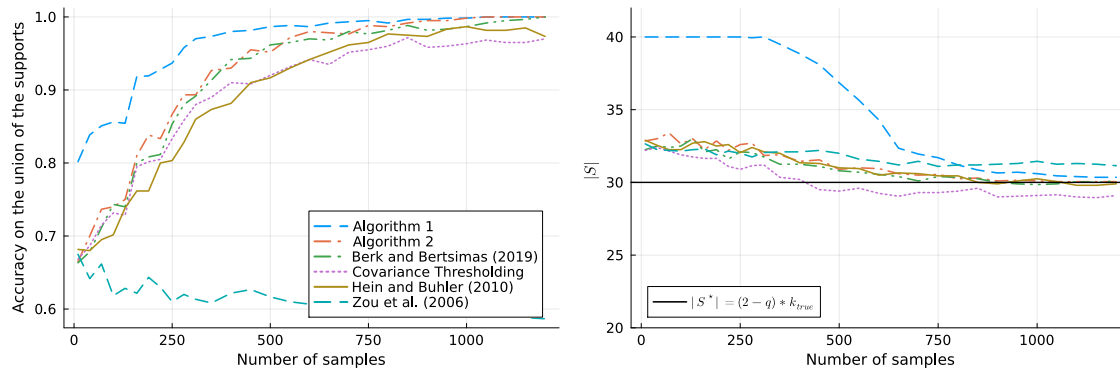
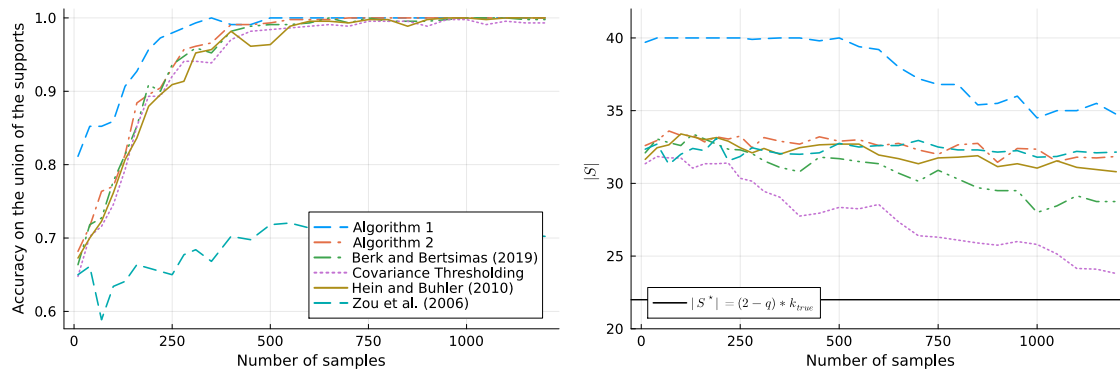


(b) Support overlap  $q = 0.5$



(c) Support overlap  $q = 0.9$

**Figure 1** Variance explained (left panel) and feasibility violation (right panel) on synthetic instances of sparse PCA with two 20-sparse PCs with partially overlapping support. Results are averaged over 20 replications.

(a) Support overlap  $q = 0.1$ (b) Support overlap  $q = 0.5$ (c) Support overlap  $q = 0.9$ 

**Figure 2** Accuracy (left panel) and joint support size (right panel) for the recovery of  $\text{supp}(\mathbf{x}_1) \cup \text{supp}(\mathbf{x}_2)$ , on synthetic instances of sparse PCA with two 20-sparse PCs with partially overlapping support. Results are averaged over 20 replications.

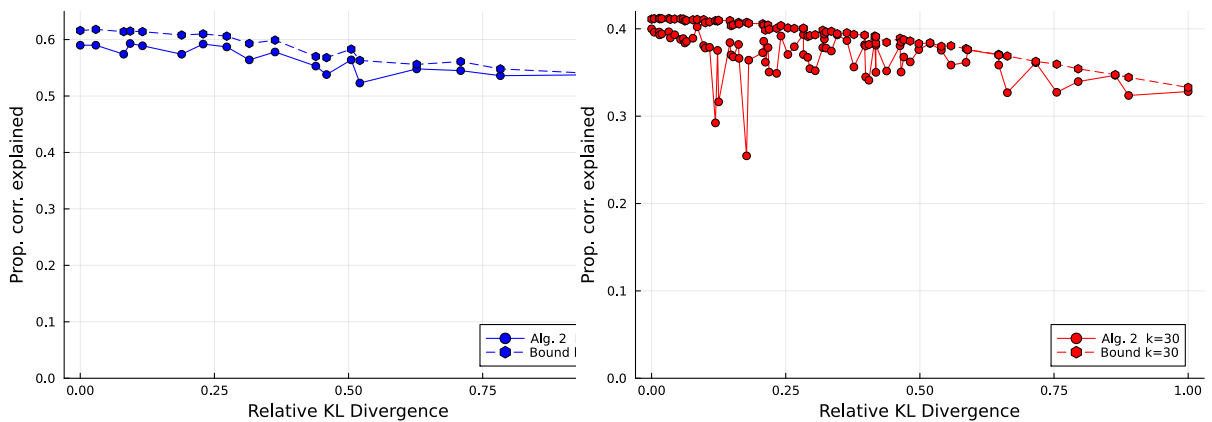
### 5.4. Specifying the Sparsity Pattern: The Benefits of Asymmetry

While Problem (3) only requires a bound on the total sparsity, thus allowing flexibility on how this budget is allocated across PCs, the worst-case semidefinite upper bound over all sparsity patterns  $(k_1, \dots, k_r) : \sum_{t \in [r]} k_t = k$  is often significantly tighter than the semidefinite relaxation of (3) with a sparsity budget of  $k$  alone, as demonstrated in Section 5.1. Moreover, Algorithm 2, which as demonstrated in Sections 5.2–5.3 is currently the best performing method for obtaining feasible solutions to Problem (3), requires that  $(k_1, \dots, k_t)$  are individually specified. Collectively, these observations suggest that it may be necessary to enumerate all allocations of the sparsity budget  $k$ , which could be expensive. In our experiments, as is often done in practice, we restricted our search to symmetric allocations. In this section, we revisit the symmetry assumption, investigate when it is justified, and study the relative benefits of asymmetric sparsity budget allocations in terms of obtaining equally sparse sets of PCs that explain more variance.

We consider the `pitprops`, `ionosphere`, `geographical` and `communities` UCI datasets with a fixed number of PCs  $r = 3$  and a given overall sparsity budget  $k \in \{15, 30\}$ . Accordingly, in Figures 3 and EC.1 of Section EC.5.6, we depict the relationship between the proportion of correlation explained in the data for each possible allocation of the sparsity budget  $(k_1, k_2, k_3) : k_1 + k_2 + k_3 = k, p \geq k_1 \geq k_2 \geq k_3 \geq 1$ , as computed by Algorithm 2 with a limit of 200 iterations and the same setup as in Section 5.2 (and the corresponding upper bound computed by Algorithm 1), against the relative asymmetry in the sparsity budget, as measured by

$$\frac{\text{KL}((k_1, k_2, k_3) || (k/3, k/3, k/3))}{\max_{p \geq k_1 \geq k_2 \geq k_3 : k_1 + k_2 + k_3 = k} \text{KL}((k_1, k_2, k_3) || (k/3, k/3, k/3))},$$

where  $KL(p||q) := \sum_i p_i \log(p_i/q_i)$  denotes the KL divergence.



**Figure 3** Symmetry of sparsity budget allocation vs. proportion of correlation in the dataset explained for the `pitprops`  $k = 15$  (left) and `ionosphere`  $k = 30$  (right) where  $r = 3$ . Note that we normalize the KL divergence for  $k = 15$  and  $k = 30$  separately.

We observe a general trend that more symmetric sparsity budget allocations tend to explain more of the correlation in the data (both in terms of actual correlation explained by a solution from Algorithm 2 and in terms of the upper bound). This suggests that, when time is a concern, requiring that all PCs are equally sparse is a reasonable approach.

Table 3 compares the quality of Algorithm 2’s solution (a) when all PCs have the sparsity budget of  $k/r$  and (b) the maximum possible correlation explained over all feasible allocations of the sparsity budget  $k$  (computed by enumerating all possible allocations of the sparsity budget—18 such allocations for  $k = 15, p \geq k$  and 74 allocations for  $k = 30, p \geq 30$ ), together with the upper bound on the proportion of correlation explained obtained in each case. We observe that in several instances a perfectly symmetric allocation of the sparsity budget yields the highest quality solution, and in all instances a perfectly symmetric allocation is within 7% in the worst-case (and within 1.75% in average) of the best solution. In the enumerated case, we also compute the optimality gap between the worst-case upper bound over all sparsity budget allocations, and the best solution found, and observe that on average it is less than 1% over the instances considered. Note that this is a different gap to the one reported in Section 5.2, where the upper bound is computed after assuming that all PCs are equally sparse.

Dataset	$p$	$r$	$k$	Symmetric			Enumerated				Improvement (%)	
				UB	Obj.	Viol.	UB	$k_t$	Obj.	Rel. gap (%)		Viol.
Pitprops	13	3	15	0.616	0.590	0.000	0.618	(6, 6, 3)	0.593	4.06%	0.000	0.47%
			30	0.652	0.650	0.000	0.652	(10, 10, 10)	0.650	0.19%	0.000	0%
Ionosphere	34	3	15	0.297	0.286	0.000	0.299	(7, 6, 2)	0.299	0%	0.001	4.48%
			30	0.411	0.400	0.000	0.412	(15, 8, 7)	0.402	2.34%	0.000	0.60%
Geographical	68	3	15	0.221	0.221	0.000	0.221	(5, 5, 5)	0.221	0%	0.000	0.00%
			30	0.410	0.389	0.000	0.420	(12, 12, 6)	0.415	1.07%	0.000	6.29%
Communities	101	3	15	0.141	0.141	0.000	0.142	(6, 5, 4)	0.142	0.02%	0.000	0.48%
			30	0.246	0.243	0.000	0.247	(11, 10, 9)	0.247	0.02%	0.000	1.71%

**Table 3** Comparison of symmetric and enumerated solution computed by Algorithm 2 for a given sparsity budget  $k_{\text{total}}$  and given number of PCs ( $r = 3$ ). We report the largest upper bound over all sparsity budget allocations as our enumerated upper bound, and report the relative optimality gap between the best solution and the worst-case bound. On average, considering asymmetric sparsity budget allocations improves the proportion of correlation explained by 1.65%.

In summary, more symmetric allocations of the sparsity budget tend to perform better on average. Therefore, for a given sparsity budget  $k$ , a reasonable strategy could be to (a) run Algorithm 2 to compute a perfectly symmetric allocation, (b) compute an upper bound using (12) across all possible allocations, and (c) run Algorithm 2 only on the asymmetric allocations for which the upper bound from (b) allows for a significant potential improvement upon the symmetric solution.

## 5.5. Summary and Guidelines From Numerical Experiments

In summary, our main findings from our numerical experiments are as follows:

- The sparsity constraint on  $\mathbf{U}$  can either be imposed via an overall sparsity budget of  $\|\mathbf{U}\|_0 \leq k$  or with component-specific sparsity constraints  $\|\mathbf{U}_t\|_0 \leq k_t$ . As reflected in Section 5.1, constraining each column separately yields substantially tighter conic relaxations. Moreover, some of the best performing algorithms for obtaining feasible solutions, including our Algorithm 2, explicitly require a constraint on the non-zero entries in each column. Therefore, constraining the sparsity of each column separately should be preferred wherever feasible.
- If practitioners have an overall sparsity budget but are agnostic about the sparsity of each column, a reasonable strategy is to require that all columns are equally sparse (i.e., set  $k_t = k/r$ ), as shown in Section 5.4. In our experiments, considering asymmetrically sparse sets of PCs increases the amount of variance explained by around 2% on average, at the price of increasing the total runtime by an order of magnitude.
- Under a separate sparsity constraint on each column of  $\mathbf{U}$ , our proposed combination of solving a conic relaxation and either rounding to obtain a disjoint solution or running Algorithm 2 yields certifiably near optimal solutions within minutes (resp. hours) for problems with hundreds (resp. thousands) of features. Moreover, it substantially outperforms existing methods for sparse PCA with multiple PCs in terms of obtaining higher-quality solutions (Section 5.2) with a lower false discovery rate (Section 5.3). Therefore, it should be considered as a viable and more accurate alternative for sparse PCA problems with multiple PCs.

## 6. Conclusion

In this paper, we studied the problem of selecting a set of mutually orthogonal sparse principal components and proposed techniques which, for the first time, allow this problem to be solved to certifiable (near) optimality with 100s or 1000s of features in minutes or hours. In particular, we proposed a strong semidefinite relaxation (Section 2) which provides high-quality upper bounds on the amount of variance explainable by any set of sparse and mutually orthogonal components, we derived a new combinatorial upper bound on the objective value that depends only on the sparsity pattern (Section 3) and a suite of numerically efficient algorithms (Section 4) which, as demonstrated in Section 5, recover components that nearly match these upper bounds. This contributes towards an ever-growing body of work demonstrating that computationally challenging non-convex optimization problems can often be solved to provable (near) optimality in practice.

## Acknowledgments

We are grateful to the associate editor and two anonymous referees for valuable suggestions that improved the manuscript.

## References

- Ahmadi AA, Dash S, Hall G (2017) Optimization over structured subsets of positive semidefinite matrices via column generation. *Discrete Optimization* 24:129–151.
- Alizadeh F, Goldfarb D (2003) Second-order cone programming. *Mathematical Programming* 95(1):3–51.
- Amini AA, Wainwright MJ (2008) High-dimensional analysis of semidefinite relaxations for sparse principal components. *2008 IEEE International Symposium on Information Theory*, 2454–2458 (IEEE).
- Asteris M, Papailiopoulos D, Kyriillidis A, Dimakis AG (2015) Sparse PCA via bipartite matchings. *Advances in Neural Information Processing Systems* 28.
- Atamtürk A, Gomez A (2019) Rank-one convexification for sparse regression. *arXiv preprint arXiv:1901.10334* .
- Barker G, Carlson D (1975) Cones of diagonally dominant matrices. *Pacific Journal of Mathematics* 57(1):15–32.
- Behdin K, Mazumder R (2021) Sparse PCA: A new scalable estimator based on integer programming. *arXiv preprint arXiv:2109.11142* .
- Benidis K, Sun Y, Babu P, Palomar DP (2016) Orthogonal sparse PCA and covariance estimation via procrustes reformulation. *IEEE Transactions on Signal Processing* 64(23):6211–6226.
- Berk L, Bertsimas D (2019) Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation* 11(3):381–420.
- Berthet Q, Rigollet P (2013) Optimal detection of sparse principal components in high dimension. *The Annals of Statistics* 41(4):1780–1815.
- Bertsimas D, Cory-Wright R (2020) On polyhedral and second-order cone decompositions of semidefinite optimization problems. *Operations Research Letters* 48(1):78–85.
- Bertsimas D, Cory-Wright R, Pauphilet J (2021) A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization* 31(3):2340–2367.
- Bertsimas D, Cory-Wright R, Pauphilet J (2022a) Mixed-projection conic optimization: A new paradigm for modeling rank constraints. *Operations Research* 70(6):3321–3344.
- Bertsimas D, Cory-Wright R, Pauphilet J (2022b) Solving large-scale sparse PCA to certifiable (near) optimality. *Journal of Machine Learning Research* 23(13):1–35.
- Bertsimas D, Kitane DL (2023) Sparse PCA: A geometric approach. *Journal of Machine Learning Research* 24:32–1.
- Bertsimas D, Pauphilet J, Van Parys B (2020) Sparse regression: Scalable algorithms and empirical performance. *Statistical Science* 35(4):555–578.
- Bienstock D, Pia AD, Hildebrand R (2023) Complexity, exactness, and rationality in polynomial optimization. *Mathematical Programming* 197(2):661–692.
- Boutsidis C, Drineas P, Magdon-Ismaïl M (2011) Sparse features for PCA-like linear regression. *Advances in Neural Information Processing Systems* 24.
- Bresler G, Park SM, Persu M (2018) Sparse PCA from sparse linear regression. *Advances in Neural Information Processing Systems* 31.
- Bühler T (2014) *A flexible framework for solving constrained ratio problems in machine learning*. Ph.D. thesis, Saarland University.
- d’Aspremont A, Bach F, El Ghaoui L (2008) Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research* 9(7).



- d'Aspremont A, El Ghaoui L, Jordan MI, Lanckriet GR (2007) A direct formulation for sparse PCA using semidefinite programming. *SIAM Review* 49(3):434–448.
- Del Pia A (2023) Sparse PCA on fixed-rank matrices. *Mathematical Programming* 198(1):139–157.
- Deshpande Y, Montanari A (2014a) Information-theoretically optimal sparse PCA. *2014 IEEE International Symposium on Information Theory*, 2197–2201 (IEEE).
- Deshpande Y, Montanari A (2014b) Sparse PCA via covariance thresholding. *Advances in Neural Information Processing Systems* 27.
- Dey SS, Mazumder R, Wang G (2022a) Using  $\ell_1$ -relaxation and integer programming to obtain dual bounds for sparse PCA. *Operations Research* 70(3):1914–1932.
- Dey SS, Molinaro M, Wang G (2022b) Solving sparse principal component analysis with global support. *Mathematical Programming* 1–39.
- Ding Y, Kunisky D, Wein AS, Bandeira AS (2023) Subexponential-time algorithms for sparse PCA. *Foundations of Computational Mathematics* 1–50.
- Dong H, Luo Y (2018) Compact disjunctive approximations to nonconvex quadratically constrained programs. *arXiv preprint arXiv:1811.08122* .
- Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. *Psychometrika* 1(3):211–218.
- Fan J, Liao Y, Wang W (2016) Projected principal component analysis in factor models. *Annals of Statistics* 44(1):219.
- Gally T, Pfetsch ME (2016) Computing restricted isometry constants via mixed-integer semidefinite programming. *Optimization Online* .
- Geoffrion AM (1974) Lagrangean relaxation for integer programming. *Approaches to Integer Programming*, 82–114 (Springer).
- Günlük O, Linderoth J (2010) Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical Programming* 124(1):183–205.
- Gupta SD, Van Parys BP, Ryu EK (2023) Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods. *Mathematical Programming* .
- Hein M, Bühler T (2010) An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. *Advances in Neural Information Processing Systems* 23.
- Horn RA, Johnson CR (1985) *Matrix analysis* (Cambridge University Press, New York).
- Hotelling H (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24(6):417.
- Jeffers JN (1967) Two case studies in the application of principal component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 16(3):225–236.
- Johnstone IM, Lu AY (2009) On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association* 104(486):682–693.
- Jolliffe IT, Trendafilov NT, Uddin M (2003) A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics* 12(3):531–547.
- Journée M, Nesterov Y, Richtárik P, Sepulchre R (2010) Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research* 11(2).
- Kim J, Tawarmalani M, Richard JPP (2022) Convexification of permutation-invariant sets and an application to sparse principal component analysis. *Mathematics of Operations Research* 47(4):2547–2584.

- Krauthgamer R, Nadler B, Vilenchik D (2015) Do semidefinite relaxations solve sparse pca up to the information limit? *The Annals of Statistics* 43(3):1300–1322.
- Li Y, Xie W (2020) Exact and approximation algorithms for sparse PCA. *arXiv preprint arXiv:2008.12438* .
- Li Y, Xie W (2021) Beyond symmetry: Best submatrix selection for the sparse truncated SVD. *arXiv preprint arXiv:2105.03179* .
- Lu Z, Zhang Y (2012) An augmented Lagrangian approach for sparse principal component analysis. *Mathematical Programming* 135(1):149–193.
- Mackey L (2008) Deflation methods for sparse PCA. *Advances in Neural Information Processing Systems* 21.
- Marshall AW, Olkin I (1979) *Inequalities: theory of majorization and its applications* (Academic, New York).
- Naikal N, Yang AY, Sastry SS (2011) Informative feature selection for object recognition via sparse PCA. *2011 International Conference on Computer Vision*, 818–825 (IEEE).
- Overton ML, Womersley RS (1992) On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications* 13(1):41–45.
- Pearson K (1901) Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2(11):559–572.
- Probel CJ, Tropp JA (2011) Large-scale PCA with sparsity constraints. Technical report, California Institute of Technology.
- Ramana MV (1997) An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming* 77(1):129–162.
- Reuther A, Kepner J, Byun C, Samsi S, Arcand W, Bestor D, Bergeron B, Gadepally V, Houle M, Hubbell M, Jones M, Klein A, Milechin L, Mullen J, Prout A, Rosa A, Yee C, Michaleas P (2018) Interactive supercomputing on 40,000 cores for machine learning and data analysis. *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 1–6 (IEEE).
- Rudin C, Chen C, Chen Z, Huang H, Semenova L, Zhong C (2022) Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys* 16:1–85.
- Tan KM, Petersen A, Witten D (2014) Classification of RNA-seq data. *Statistical Analysis of Next Generation Sequencing Data*, 219–246 (Springer).
- Tropp JA, Yurtsever A, Udell M, Cevher V (2017) Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications* 38(4):1454–1485.
- Udell M, Horn C, Zadeh R, Boyd S (2016) Generalized low rank models. *Foundations and Trends® in Machine Learning* 9(1):1–118.
- Vu VQ, Cho J, Lei J, Rohe K (2013) Fantope projection and selection: A near-optimal convex relaxation of sparse pca. *Advances in Neural Information Processing Systems* 26.
- Vu VQ, Lei J (2013) Minimax sparse principal subspace estimation in high dimensions. *The Annals of Statistics* 41(6):2905–2947.
- Wang J, Dey SS, Xie Y (2023) Variable selection for kernel two-sample tests. *arXiv preprint arXiv:2302.07415* .
- Witten DM, Tibshirani R, Hastie T (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics* 10(3):515–534.
- Yuan XT, Zhang T (2013) Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research* 14(4).

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

## Supplementary Material

### EC.1. Proof of Proposition 1

*Proof of Proposition 1* We decompose each matrix  $\mathbf{Y}^t$  into  $\mathbf{Y}^t = \mathbf{u}_t \mathbf{u}_t^\top$  with  $\|\mathbf{u}_t\|^2 = \text{tr}(\mathbf{Y}^t) = 1$ . Hence, for any pair  $(t, t')$ ,  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = \mathbf{u}_t^\top \mathbf{Y}^{t'} \mathbf{u}_t = (\mathbf{u}_t^\top \mathbf{u}_{t'})^2 \geq 0$ .

( $\Rightarrow$ ) If  $\mathbf{Y} := \sum_{t' \in [r]} \mathbf{Y}^{t'} \preceq \mathbb{I}$ , then, for any  $t \in [r]$ ,  $\mathbf{u}_t^\top \mathbf{Y} \mathbf{u}_t \leq \|\mathbf{u}_t\|^2 = 1$ . However,  $\mathbf{u}_t^\top \mathbf{Y} \mathbf{u}_t = 1 + \sum_{t' \neq t} \langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle$ . Hence, for all  $t' \neq t$ , we must have  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = 0$ .

( $\Leftarrow$ ) If  $\langle \mathbf{Y}^t, \mathbf{Y}^{t'} \rangle = 0$  for all  $t' \neq t$ , then  $\{\mathbf{u}_t\}_{t \in [r]}$  is an orthonormal family that can be completed to form an orthonormal basis  $\{\mathbf{u}_t\}_{t \in [p]}$ . For any  $t \in [p]$ ,  $t' \in [r]$ ,  $\mathbf{u}_t^\top \mathbf{Y}^{t'} \mathbf{u}_t = 1$  if  $t = t'$ , 0 otherwise so for any  $t \in [p]$ ,  $\mathbf{u}_t^\top \mathbf{Y} \mathbf{u}_t \leq \|\mathbf{u}_t\|^2$  and  $\mathbf{Y} = \sum_{t' \in [r]} \mathbf{Y}^{t'} \preceq \mathbb{I}$ .  $\square$

### EC.2. A derivation of some valid inequalities

In this section, we derive the valid inequalities introduced in Theorem 2 from first principles. We proceed in two ways. First, we derive inequalities which hold for each  $\mathbf{Y}^t$  separately. Second, we observe that these inequalities can be generalized to also apply for  $\mathbf{Y} = \sum_{t \in [r]} \mathbf{Y}^t$ , and also derive new inequalities which reflect the interaction of the sparsity and rank constraints.

*Rank-One Valid Inequalities* First, inspired by Bertsimas and Cory-Wright (2020), we observe that in a feasible solution to Problem (8) the  $2 \times 2$  minors of  $\mathbf{Y}^t$  are certainly non-negative, i.e.,

$$(Y_{i,j}^t)^2 \leq Y_{i,i}^t Y_{j,j}^t, \quad \forall i, j \in [p].$$

These constraints are implied by  $\mathbf{Y}^t \succeq \mathbf{0}$  and hence redundant in-and-of-themselves. However, we can sum over all such constraints  $i \in [p]$  and use  $\text{tr}(\mathbf{Y}^t) = 1$ , to obtain the constraint

$$\sum_{i \in [p]} (Y_{i,j}^t)^2 \leq Y_{j,j}^t, \quad \forall j \in [p].$$

This constraint is a sum of redundant constraints and hence redundant. However, we can strengthen it, by noting that it is a separable convex quadratic inequality under logical constraints. Indeed, by Lemma 2, its convex closure under the logical constraints  $Y_{i,j}^t = 0$  if  $Z_{i,t} = 0$  is given by:

$$\sum_{j=1}^p (Y_{i,j}^t)^2 \leq Y_{i,i}^t Z_{i,t}, \quad \forall i \in [p], t \in [k]. \quad (\text{EC.1})$$

*Rank-r Valid Inequalities* In the same spirit as in the rank one case, we can obtain strong valid inequalities by summing the  $2 \times 2$  minors of  $Y_{i,i} = \sum_{t=1}^r Y_{i,i}^t$ . Indeed, since  $\mathbf{Y}$  is positive semidefinite, summing its  $2 \times 2$  minors implies that:

$$\sum_{j=1}^p (Y_{i,j})^2 \leq r Y_{i,i}.$$

Moreover, since  $Y_{i,j} = \sum_{t=1}^r \mathbf{Y}_{i,j}^t$  is a rank-one quadratic under logical constraints  $Y_{i,j}^t = 0$  if  $Z_{i,t} = 0$ , invoking Lemma 3 reveals that the convex closure of this quadratic constraint under these logical constraints is given by the strengthened inequality:

$$\sum_{j=1}^p (Y_{i,j})^2 \leq r Y_{i,i} \min \left( 1, \sum_{t=1}^r Z_{i,t} \right), \quad \forall i \in [p]. \quad (\text{EC.2})$$

Second, in any feasible solution we have:

$$|Y_{i,j}| \leq \sum_{t=1}^r |Y_{i,j}^t| = \sum_{t=1}^r |U_{i,t}| |U_{j,t}|.$$

Let us denote by  $k_t$  the sparsity of the  $t$ th column of  $\mathbf{U}$ ,  $\mathbf{U}_t$ . Then, it is well known that  $\|\mathbf{U}_t\|_1 \leq \sqrt{k_t}$ .

Therefore:

$$\sum_{j=1}^p |Y_{i,j}| \leq \sum_{j=1}^p \left( \sum_{t=1}^r |U_{i,t}| |U_{j,t}| \right) \leq \sum_{t=1}^r \sqrt{k_t} |U_{i,t}|.$$

Next, squaring both sides and invoking the Cauchy-Schwarz inequality reveals that

$$\left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq \left( \sum_{t=1}^r U_{i,t}^2 \right) \left( \sum_{t=1}^r k_t \right) = k Y_{i,i}.$$

Finally, noting that the expression  $\left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i}$  is a convex quadratic under logical constraints  $Y_{i,j}^t = 0$  if  $Z_{i,t} = 0$  and invoking Lemma 3 to obtain its convex closure yields the strengthened second-order cone inequality

$$\left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i} \min \left( 1, \sum_{t \in [r]} Z_{i,t} \right), \quad \forall i \in [p], t \in [k]. \quad (\text{EC.3})$$

Third, in the same spirit, the  $2 \times 2$  minors of  $\mathbf{Y} \preceq \text{Diag} \left( \min \left( \mathbf{e}, \sum_{t \in [r]} \mathbf{Z}_t \right) \right)$  are

$$\left( \min \left( 1, \sum_{t \in [r]} Z_{i,t} \right) - Y_{i,i} \right) \left( \min \left( 1, \sum_{t \in [r]} Z_{j,t} \right) - Y_{j,j} \right) \geq \left( \min \left( 1, \sum_{t \in [r]} Z_{i,t} \right) \delta_{i,j} - Y_{i,j} \right)^2,$$

where  $\delta_{i,j} = \mathbf{1}\{i=j\}$  is an indicator denoting whether  $i=j$ . Summing these constraints over all indices  $i \neq j$  and using  $k-r+1$  as an upper bound on  $\sum_{i \in [p]: i \neq j} \sum_{t \in [r]} Z_{i,t} - Y_{i,i}$  then yields

$$(k-r+1) \left( \min \left( 1, \sum_{t \in [r]} Z_{j,t} \right) - Y_{j,j} \right) \geq \sum_{i \in [p]: i \neq j} Y_{i,j}^2, \quad \forall j \in [p].$$

Finally, we recognize the right hand side as a sum of rank-one quadratic terms  $(\sum_{t=1}^r Y_{i,j}^t)^2$  under logical constraints  $Y_{i,j}^t = 0$  if  $Z_{j,t} = 0$  and invoke Lemma 3 to obtain the convex closure, giving:

$$(k-r+1) \min \left( 1, \sum_{t \in [r]} Z_{j,t} \right) \left( \min \left( 1, \sum_{t \in [r]} Z_{j,t} \right) - Y_{j,j} \right) \geq \sum_{i \in [p]: i \neq j} Y_{i,j}^2 \quad \forall j \in [p]. \quad (\text{EC.4})$$

The result then follows by introducing a vector  $\mathbf{w}$  such that  $w_i$  models  $\min(1, \sum_{t \in [r]} Z_{i,t})$  via  $\mathbf{w} \in [0, 1]^p$ ,  $\mathbf{w} \leq \mathbf{Z}\mathbf{e}$ , and noting that we can replace  $\mathbf{Y} \preceq \text{Diag}(\min(\mathbf{e}, \mathbf{Z}\mathbf{e}))$  with  $\mathbf{Y} \preceq \text{Diag}(\mathbf{w})$ .

### EC.3. Proof of Proposition 2

*Proof of Proposition 2* First, let us observe that if  $\sum_{i \in [p]} Z_{i,t} \leq k$  and  $\mathbf{Y}^t = \mathbf{U}_t \mathbf{U}_t^\top$  is a rank-one matrix such that  $\|\mathbf{U}\|_2 = 1$  then we have  $\|\mathbf{U}\|_1 \leq \sqrt{k_t}$  by norm equivalence. Therefore

$$\sum_{j=1}^p |Y_{i,j}^t| \leq \sum_{j=1}^p |U_{i,t}| |U_{j,t}| \leq \sqrt{k_t} |U_{i,t}|.$$

Squaring both sides of this inequality then yields

$$\left( \sum_{j=1}^p |Y_{i,j}^t| \right)^2 \leq k_t Y_{i,i}^t,$$

and combining Lemma 2 with this inequality yields (10).

Second, in the same spirit, since  $\mathbf{U}_t \mathbf{U}_t^\top$  is only supported on indices where  $\mathbf{Z}_t$  is non-zero, we have that  $\mathbf{Y}^t \preceq \text{Diag}(\mathbf{Z}_t)$ . This constraint implies the following  $2 \times 2$  minors are non-negative

$$(Z_{i,t} - Y_{i,i}^t)(Z_{j,t} - Y_{j,j}^t) \geq (\delta_{i,j} - Y_{i,j}^t)^2, \quad \forall i, j \in [p],$$

where  $\delta_{i,j} = 1$  if  $i = j$  and 0 otherwise. Summing these inequalities over indices  $i \neq j$  and setting  $k_t - 1$  as a valid upper bound on  $\sum_{i \in [p]: i \neq j} Z_{i,t} - Y_{i,i}^t$  whenever  $Z_{j,t} = 1$  (as  $Y_{i,j}^t = 0$  if  $Z_{j,t} = 0$ ) gives

$$(k_t - 1)(Z_{j,t} - Y_{j,j}^t) \geq \sum_{i \in [p]: i \neq j} Y_{i,j}^t{}^2, \quad \forall j \in [p].$$

Finally, using Lemma 2 to take the convex closure of this inequality under the logical constraints  $Y_{i,j}^t = 0$  if  $Z_{j,t} = 0$  gives Equation (11).  $\square$

### EC.4. Complete Formulations for the Semidefinite and Second-Order Cone Relaxations

In this section, we provide the complete formulation for our SDP relaxation (12) as well as its second-order cone approximation.

#### EC.4.1. Semidefinite Relaxation

In Section 2.3, we proposed a semidefinite relaxation, (12), in the case where a sparsity budget for each PC,  $k_t$ , is provided. In particular, this relaxation involves valid inequalities that Kim et al. (2022) have derived in the single-PC case. To the best of our knowledge, their formulation leads to the strongest known relaxation for sparse PCA with  $r = 1$  which can be solved in polynomial time. We note however that invoking a fixed but sufficiently large level of the sum-of-squares hierarchy may give tighter relaxations, although we do not write these relaxations down as they involve very large semidefinite constraints and are therefore intractable in practice (see also Dey et al. (2022a) for an NP-hard relaxation that uses the  $\ell_1$  norm).

In (12), we concisely denoted  $(\mathbf{Y}^t, \mathbf{Z}^t) \in \mathcal{T}(k_t)$  the set of valid inequalities involved in Kim et al. (2022)’s “T-relaxation”. We now elicit the constraints involved in the set  $\mathcal{T}(k_t)$  and provide the

complete formulation of the SDP relaxation (12). For each  $t \in [r]$ , we introduce an additional variable  $\mathbf{F}^t$  to capture the entry-wise absolute value of  $\mathbf{Y}^t$ , and an additional matrix  $\mathbf{G}^t$  which contains a sorted version of  $\mathbf{F}^t$ . We obtain:

$$\begin{aligned}
& \max_{\substack{\mathbf{Z} \in [0,1]^{p \times r}: \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k, \\ \mathbf{w} \in [0,1]^p}} \max_{\substack{\mathbf{Y} \in \mathcal{S}_+^p, \mathbf{Y}^t, \mathbf{F}^t, \mathbf{G}^t \in \mathcal{S}_+^p, \\ \mathbf{T}^t \in \mathbb{R}_+^{p \times p}, \\ \mathbf{r}^{t,D} \in \mathbb{R}^{p-1}, \mathbf{t}^{t,D} \in \mathbb{R}_+^{p \times p-1}}} \langle \mathbf{Y}, \boldsymbol{\Sigma} \rangle & \quad (\text{EC.5}) \\
\text{s.t. } & \mathbf{Y} \preceq \text{Diag}(\mathbf{w}), \mathbf{Y} = \sum_{t=1}^k \mathbf{Y}^t, \mathbf{w} \leq \mathbf{Z}\mathbf{e}, \\
& \sum_{j=1}^p Y_{i,j}^2 \leq r Y_{i,i} w_i & \forall i \in [p], \\
& \left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i} w_i & \forall i \in [p], \\
& \sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k-r+1) w_j (w_j - Y_{j,j}) & \forall j \in [p], \\
& \pm \mathbf{Y}^t \leq \mathbf{F}^t & \forall t \in [r], \\
& G_{i,1}^t \geq G_{i,2}^t \geq \dots \geq G_{i,k_t}^t & \forall i \in [k_t], \forall t \in [r], \\
& G_{i,j}^t = 0 & \forall i > k_t \text{ or } j > k_t, \forall t \in [r], \\
& \text{tr}(\mathbf{Y}^t) = \text{tr}(\mathbf{G}^t) = \text{tr}(\mathbf{F}^t) = 1 & \forall t \in [r], \\
& \langle \mathbf{E}, \mathbf{G}^t \rangle = \langle \mathbf{E}, \mathbf{F}^t \rangle & \forall t \in [r], \\
& \sum_{i=1}^j G_{i,i}^t \geq j r_j^{D,t} + \sum_{j=1}^n t_{i,j}^{t,D} & \forall j \in [p-1], t \in [r], \\
& Y_{i,i}^t \leq r_j^{t,D} + t_{i,j}^{t,D} & \forall i \in [p], j \in [p-1], t \in [r], \\
& (F_{i,j}^t)^2 \leq T_{i,j}^t T_{j,i}^t, T_{i,i}^t = F_{i,i}^t & \forall i \in [p], j \in [i-1], t \in [r], \\
& \sum_{j \in [p]} T_{i,j}^t = Z_{i,t}, \sum_{i \in [p]} T_{i,j}^t = k_t F_{j,j}^t & \forall i \in [p], \forall j \in [p], t \in [r], \\
& 0 \leq T_{i,j}^t \leq F_{i,j}^t & \forall i, j \in [p], t \in [r].
\end{aligned}$$

The additional variables  $\mathbf{r}^{t,D}$  and  $\mathbf{t}^{t,D}$  are introduced to enforce coupling constraints between the diagonal entries of  $\mathbf{F}^t$  and  $\mathbf{G}^t$  (Kim et al. 2022, eq. 44), while  $\mathbf{T}^t$  allows to couple  $\mathbf{F}^t$  with the binary variables  $\mathbf{Z}$  (Kim et al. 2022, eq. 50). In contrast to Kim et al. (2022), we explicitly require that each  $\mathbf{F}^t$  is positive semidefinite (rather than that its  $2 \times 2$  minors are), in order to obtain a stronger relaxation; we consider the  $2 \times 2$  minors when developing a more tractable relaxation in the next section.

### EC.4.2. A Second-Order Cone Relaxation for High-Dimensional Settings

Unfortunately, (12) cannot scale beyond  $p = 100$ , at least with current technology, due to the presence of multiple semidefinite matrices and constraints.

We now develop a more tractable, albeit less tight, version of the relaxation of (12) which scales to  $p > 100$  features. Namely, we replace all semidefinite constraints of the form  $\mathbf{X} \in \mathcal{S}_+^p$  with the non-negativity of their  $2 \times 2$  minors,  $X_{i,i}X_{j,j} \geq X_{i,j}^2 \ \forall i, j \in [p]$ , as presented by Bertsimas and Cory-Wright (2020) and references therein. This gives the following second-order cone relaxation of (12):

$$\begin{aligned}
& \max_{\substack{\mathbf{Z} \in [0,1]^{p \times r}: \\ \langle \mathbf{E}, \mathbf{Z} \rangle \leq k, \mathbf{w} \in [0,1]^p}} \quad \max_{\substack{\mathbf{Y} \in \mathcal{S}^p, \mathbf{Y}^t, \mathbf{F}_t, \mathbf{G}_t \in \mathcal{S}^p, \\ \mathbf{T}_t \in \mathbb{R}_+^{p \times p} \ \forall t \in [k], \\ \mathbf{r}^{t,D} \in \mathbb{R}^{p-1}, \mathbf{t}^{t,D} \in \mathbb{R}^{p \times p-1}}} \langle \mathbf{Y}, \boldsymbol{\Sigma} \rangle \tag{EC.6} \\
\text{s.t.} \quad & \mathbf{Y} = \sum_{t=1}^k \mathbf{Y}^t, \ \text{tr}(\mathbf{Y}^t) = 1, \ \mathbf{w} \leq \mathbf{Z}\mathbf{e} \quad \forall t \in [r], \\
& Y_{i,j}^t{}^2 \leq Y_{i,i}^t Y_{j,j}^t \quad \forall i, j \in [p], \forall t \in [r], \\
& (\delta_{i,j} - Y_{i,j})^2 \leq (w_i - Y_{i,i})(w_j - Y_{j,j}) \quad \forall i, j \in [p], \forall t \in [r], \\
& \sum_{j=1}^p Y_{i,j}^2 \leq r Y_{i,i} w_i \quad \forall i \in [p] \\
& \left( \sum_{j=1}^p |Y_{i,j}| \right)^2 \leq k Y_{i,i} w_i, \ \pm \mathbf{Y}^t \leq \mathbf{F}_t \quad \forall i \in [p], \forall t \in [r], \\
& \sum_{i \in [p]: i \neq j} Y_{i,j}^2 \leq (k - r + 1) w_j (w_j - Y_{j,j}) \quad \forall j \in [p], \\
& Y_{i,i}^t \leq t_{i,j}^{t,D} + r_j^{t,D} \quad \forall i \in [p], j \in [p-1], t \in [r], \\
& \text{tr}(\mathbf{Y}^t) = \text{tr}(\mathbf{G}^t) = \text{tr}(\mathbf{F}_t) = 1 \quad \forall t \in [r], \\
& \langle \mathbf{E}, \mathbf{G}_t - \mathbf{F}_t \rangle = 0 \quad \forall t \in [r], \\
& G_{i,j}^t{}^2 \leq G_{i,i}^t G_{j,j}^t \quad \forall i, j \in [p], \forall t \in [r], \\
& G_{i,1}^t \geq G_{i,2}^t \geq \dots \geq G_{i,k_t}^t \quad \forall i \in [k_t], \forall t \in [r], \\
& G_{i,j}^t = 0 \quad \forall i > k_t \text{ or } j > k_t \ \forall t \in [r], \\
& \sum_{i=1}^j G_{i,i}^t \geq j r_j^{D,t} + \sum_{j=1}^n t_{i,j}^{t,D} \quad \forall j \in [p-1], t \in [r], \\
& F_{i,j}^t{}^2 \leq T_{i,j}^t T_{j,i}^t, \ T_{i,i}^t = F_{i,i}^t \quad \forall i \in [p], j \in [i-1], t \in [r], \\
& \sum_{j \in [p]} T_{i,j}^t = Z_{i,t}, \ \sum_{i \in [p]} T_{i,j}^t = k_t F_{j,j}^t \quad \forall i \in [p], \forall j \in [p], t \in [r], \\
& 0 \leq T_{i,j}^t \leq F_{i,j}^t, F_{i,j}^t{}^2 \leq F_{i,i}^t F_{j,j}^t \quad \forall i, j \in [p], \forall t \in [r].
\end{aligned}$$



REMARK EC.1. We can further improve the second-order cone relaxation (EC.6) without compromising its tractability by iteratively solving (EC.6) and imposing linear cuts of the form

$$\langle \mathbf{X}, \mathbf{v}\mathbf{v}^\top \rangle \geq \mathbf{0},$$

for each matrix  $\mathbf{X}$  which is positive semidefinite in (12), where  $\mathbf{v}$  is a trailing eigenvector of  $\mathbf{X}$  in the most recent solution to the relaxation, as presented in Bertsimas and Cory-Wright (2020). In our numerical experiments (Section 5), we consider a version of this scheme where we iteratively impose one such cut corresponding to the most negative eigenvector in the matrices  $\mathbf{Y}^t$ ,  $\mathbf{G}^t$ ,  $\mathbf{F}^t$ , and  $\text{Diag}(\mathbf{w}) - \mathbf{Y}$ , and perform up to 50 iterations of this scheme, stopping early if the most negative eigenvalue in all constraints is  $-10^{-4}$  or larger.

## EC.5. Supplementary Numerical Results

This section provides supplementary results supporting the numerical experiments performed in Section 5.

### EC.5.1. Description of the Experimental Setup

All experiments were performed on MIT’s supercloud cluster (Reuther et al. 2018), which hosts Intel Xeon Platinum 8260 processors and Intel Xeon Gold 6248 processors. For experiments where  $p < 100$ , we use Platinum processors with a budget of 32 GB RAM, for experiments where  $p \in [100, 250]$ , use Platinum processors with a budget of 100 GB RAM, while for experiments where  $p > 250$ , we use Gold processors with a budget of 370 GB RAM.

We also implement some existing algorithmic strategies from the literature, to provide a baseline for the performance of our methods. To abide by software licensing restrictions, all existing strategies from the literature were benchmarked using a MacBook Pro laptop with a 2.9GHz 6-Core Intel i9 CPU, using 16 GB DDR4 RAM. Therefore, runtimes are not directly comparable across strategies, although hardware differences should not cause significant deviations in runtime.

### EC.5.2. Description of the Data Sources

We perform experiments on eleven datasets from the frequently used UCI database in Sections 5.1-5.2 and 5.4. Of the eleven datasets, six datasets are overdetermined (meaning  $n > p$ ), while five datasets are underdetermined (meaning  $p < n$ ). Moreover, many existing works on sparse PCA report results on similar datasets. For instance, the pitprops dataset was also considered by Jolliffe et al. (2003), Zou et al. (2006), Journée et al. (2010) among others, and three of the datasets studied by Berk and Bertsimas (2019) are included within our suite of datasets. Thus, our experimental setup is broadly representative of both the underdetermined and overdetermined regimes, and the literature. For completeness, we summarize the datasets we benchmark on and their dimensionality in Table EC.1.

Dataset	$p$	$n$
Pitprops	13	180
Wine	13	178
Ionosphere	34	351
Lung (Lung cancer)	54	32
Geographical (Geographical Original of Music)	68	1059
Communities (Communities and Crime)	101	1994
Arrhythmia	274	452
Voice (LSVT Voice Rehabilitation)	310	126
Gait (Gait Classification)	320	48
Gastro (Gastrointestinal Lesions in Regular Colonoscopy)	466	152
Micromass	1300	931

**Table EC.1** Summary of the 11 datasets in our library, where  $n$  denotes the number of observations and  $p$  the number of features. For conciseness, the names of certain datasets are abbreviated throughout. For these datasets, we first state the abbreviation used, followed by their full names in brackets. Further, we report the dimensionality of each dataset after preprocessing to remove all features with missing values. All datasets can be found in the UCI database, except the pitprops dataset, which is due to Jeffers (1967) and distributed via the R package ElasticNet.

### EC.5.3. Performance of Conic Relaxations

Dataset	Dim. ( $p$ )	Rank ( $r$ )	Sparsity ( $k, k_t$ )	PSD		SOC-Cuts		SOC	
				UB	T(s)	UB	T(s)	UB	T(s)
Pitprops	13	2	10, (5, 5)	0.449	17.81	0.450	19.83	0.524	0.22
			20, (10, 10)	0.507	0.59	0.516	16.33	0.672	0.23
			15, (5, 5, 5)	0.616	0.87	0.619	29.53	0.761	0.42
			30, (10, 10, 10)	0.652	0.72	0.678	26.61	1.007	0.46
Wine	13	2	10, (5, 5)	0.458	0.57	0.459	18.32	0.529	0.25
			20, (10, 10)	0.554	0.55	0.56	16.25	0.722	0.22
			15, (5, 5, 5)	0.632	0.79	0.634	25.49	0.762	0.52
			30, (10, 10, 10)	0.665	0.74	0.689	26.66	1.083	0.57
Ionosphere	34	2	10, (5, 5)	0.209	8.22	0.209	118.9	0.221	1.45
			20, (10, 10)	0.305	8.35	0.31	113.5	0.363	1.53
		2	40, (20, 20)	0.378	9.84	0.391	89.17	0.504	1.85
			15, (5, 5, 5)	0.297	12.74	0.298	272.1	0.331	4.39
		3	30, (10, 10, 10)	0.411	11.93	0.419	206.8	0.545	4.49
			60, (20, 20, 20)	0.464	13.38	0.495	291.8	0.757	5.95
Geographical	68	2	10, (5, 5)	0.147	81.93	0.147	67.86	0.147	17.45
			20, (10, 10)	0.294	59.82	0.294	53.28	0.294	19.32
		2	40, (20, 20)	0.432	101.1	0.433	652.9	0.567	13.08
			15, (5, 5, 5)	0.221	93.60	0.221	82.31	0.221	28.63
		3	30, (10, 10, 10)	0.410	79.02	0.41	2292	0.441	29.76
			60, (20, 20, 20)	0.520	111.4	0.529	1241	0.852	29.02
Communities	101	2	20, (10, 10)	0.169	307.7	0.169	2583	0.175	44.83
			40, (20, 20)	0.263	373.9	0.268	2431	0.286	40.13
		3	15, (5, 5, 5)	0.141	441.6	0.141	2122	0.144	80.3
			30, (10, 10, 10)	0.245	564.2	0.246	4142	0.262	65.25
			60, (20, 20, 20)	0.378	447.3	0.386	4581	0.429	54.71
Arrhythmia	274	2	10, (5, 5)	-	-	0.031	36830	0.031	54.71
			20, (10, 10)	-	-	0.052	37140	0.055	456.0
		2	40, (20, 20)	-	-	0.080	39940	0.086	445.8
			15, (5, 5, 5)	-	-	0.046	71670	0.047	671.8
		3	30, (10, 10, 10)	-	-	0.076	74840	0.083	812.2
			60, (20, 20, 20)	-	-	0.118	69390	0.129	803.7
Micromass	1300	2	10, (5, 5)	-	-	-	-	0.008	1089
			20, (10, 10)	-	-	-	-	0.015	13620
		2	40, (20, 20)	-	-	-	-	0.027	9213
			15, (5, 5, 5)	-	-	-	-	0.012	7953
		3	30, (10, 10, 10)	-	-	-	-	0.023	19640
			60, (20, 20, 20)	-	-	-	-	0.043	18630

**Table EC.2** Performance of bounds across UCI datasets. All bounds are normalized by dividing by  $p = \text{tr}(\Sigma)$ , i.e., the number of features, to report in terms of the proportion of correlation explained. The notation “-” denotes that an instances could not be solved using the provided memory budget, namely 32 GB for instances where  $p \leq 101$ , 100 GB for instances where  $p \in [102, 250]$ , 370 GB for instances where  $p > 250$ , and a time budget of 2 days.

#### EC.5.4. Preliminary Experiments With Pitprops Dataset

We now provide instancewise results for different variants of our methods on the `pitprops` dataset. In particular, we consider invoking the valid inequalities (16) derived in Section 3 to improve branch-and-bound further. When we do so, we also invoke a branching callback each time we expand a node, to determine whether the subtree rooted at this node can improve upon the incumbent

solution. This is justified by the fact that, at each node, some variables  $Z_{i,t}$  have been fixed to 0, some have been fixed to 1, and some have not been fixed. Accordingly, we can compute an upper bound on any solution with the same fixed variables by relaxing the orthogonality constraint and applying the Gershgorin circle theorem to each component separately; see Bertsimas et al. (2022b, Section 2.4) for a discussion of this callback in the rank-one case. In particular, if the Gershgorin bound for a given subtree is weaker than an incumbent solution, then this subtree does not contain any optimal solutions and we can prune it from our search tree.

$r$	$k_t$	Alg. 1 $\lambda = 0$				Alg. 1 $\lambda = 1$			Alg. 2		
		UB	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
2	2	0.295	<b>0.295</b>	0	20.64	<b>0.295</b>	0	0.43	<b>0.295</b>	0	7.54
2	4	0.408	0.378	0	20.80	<b>0.404</b>	0	0.52	0.400	0	4.94
2	6	0.477	0.437	0	20.62	0.443	0	0.62	0.452	0	6.77
2	8	0.501	0.375	0	20.93	0.446	0	0.640	<b>0.476</b>	0	7.83
2	10	0.507	0.463	0	21.01	0.464	0	0.72	<b>0.500</b>	0	6.36
3	2	0.435	<b>0.435</b>	0	20.81	<b>0.435</b>	0	0.71	0.424	0	9.98
3	4	0.572	0.525	0	21.03	<b>0.555</b>	0	1.01	0.551	0	7.24
3	6	0.641	0.463	0	23.03	0.569	0	0.92	<b>0.608</b>	0	10.21
3	8	0.652	0.580	0	20.94	0.569	0	1.06	<b>0.638</b>	0	8.88
3	10	0.652	0.392	0	20.81	0.569	0	1.39	<b>0.650</b>	0	11.43
4	2	0.554	<b>0.554</b>	0	21.22	<b>0.554</b>	0	0.97	<b>0.554</b>	0	11.05
4	4	0.704	0.470	0	21.07	<b>0.657</b>	0	3.25	0.657	0.003	13.81
4	6	0.737	0.537	0	22.22	0.644	0	2.07	<b>0.697</b>	0.002	12.14
4	8	0.737	0.553	0	22.71	0.644	0	3.48	<b>0.720</b>	0	11.84
4	10	0.737	0.508	0	21.03	0.644	0	3.16	<b>0.736</b>	0	11.22
5	2	0.657	0.455	0	20.87	0.648	0	1.57	0.647	0	12.23
5	4	0.795	0.586	0	21.27	0.709	0	8.12	<b>0.743</b>	0	15.6
5	6	0.807	0.538	0	23.47	0.713	0	16.54	<b>0.779</b>	0.016	13.95
5	8	0.807	0.563	0	21.02	0.713	0	24.78	<b>0.800</b>	0.004	17.94
5	10	0.807	0.525	0	21.97	0.713	0	7.39	<b>0.807</b>	0.001	15.01
6	2	0.749	0.581	0	21.45	<b>0.749</b>	0	4.87	0.746	0	12.61
6	4	0.866	0.576	0	21.23	<b>0.780</b>	0	54.2	0.807	0.035	23.08
6	6	0.870	0.617	0	23.45	0.780	0	12.54	<b>0.839</b>	0.044	15.63
6	8	0.870	0.628	0	21.69	0.780	0	8.45	0.849	0.011	17.20
6	10	0.870	0.664	0	21.12	0.780	0	86.88	0.866	0.006	25.91
Avg		0.668	0.508	0	21.46	0.610	0	9.85	<b>0.649</b>	0.005	12.42

**Table EC.3** Performance of Algorithms 1 and 2 on the pitprops dataset ( $p = 13$ ) using the experimental setup laid out in Section 5.2. We denote the best-performing solution (in terms of the proportion of variance explained minus the total orthogonality constraint violation) in bold. We report the semidefinite upper bound obtained from solving Problem (12) as part of our analysis of Algorithm 1 with  $\lambda = 0$ , but do not report it as part of our analysis of Algorithm 1 with  $\lambda = 1$  to avoid redundancy. Note that  $k_t$  denotes the sparsity of each individual component, meaning a set of  $r$  PCs have a collective sparsity budget of  $k_t r$ , and that all objective values are reported in terms of the proportion of variance explained by dividing by  $p$ , the number of features.

$r$	$k_t$	Branch-and-Bound					Branch-and-Bound with (15)				
		UB	Obj.	Viol.	Nodes	T(s)	UB	Obj.	Viol.	Nodes	T(s)
2	2	0.295	<b>0.295</b>	0	5100	10.9	0.295	<b>0.295</b>	0	3557	19.42
2	4	0.404	<b>0.404</b>	0	99800	59.49	0.404	<b>0.404</b>	0	109485	238.48
2	6	0.514	<b>0.456</b>	0	1405600	> 600	0.521	0.452	0	1038059	> 600
2	8	0.595	0.467	0	1266600	> 600	0.604	0.465	0	1413722	> 600
2	10	0.633	0.486	0	640900	> 600	0.635	0.488	0	848164	> 600
3	2	0.435	<b>0.435</b>	0	22400	17.92	0.435	<b>0.435</b>	0	17924	176.79
3	4	0.717	0.530	0	542700	> 600	0.753	0.524	0	222894	> 600
3	6	0.846	0.551	0	518100	> 600	0.879	0.560	0	387403	> 600
3	8	0.933	0.585	0	564200	> 600	0.935	0.570	0	612382	> 600
3	10	0.962	0.627	0	383100	> 600	0.963	0.595	0	455231	> 600
4	2	0.566	<b>0.554</b>	0	783600	> 600	0.774	<b>0.554</b>	0	36000	> 600
4	4	1.089	0.636	0	269900	> 600	1.114	0.610	0	203231	> 600
4	6	1.213	0.642	0	268300	> 600	1.226	0.617	0	201271	> 600
4	8	1.267	0.648	0	251800	> 600	1.266	0.699	0	308274	> 600
4	10	1.289	0.713	0	163200	> 600	1.289	0.714	0	266105	> 600
5	2	0.946	0.641	0	702700	> 600	1.073	0.603	0	44795	> 600
5	4	1.430	0.697	0	215200	> 600	1.468	0.652	0	113951	> 600
5	6	1.555	0.692	0	199400	> 600	1.555	0.686	0	152744	> 600
5	8	1.592	0.761	0	191100	> 600	1.610	0.714	0	156767	> 600
5	10	1.616	0.801	0	147500	> 600	1.620	0.804	0	108311	> 600
6	2	1.323	0.702	0	323700	> 600	1.430	0.698	0	43880	> 600
6	4	1.822	0.761	0	139900	> 600	1.807	0.711	0	93572	> 600
6	6	1.903	0.771	0	114000	> 600	1.907	0.776	0	43754	> 600
6	8	1.932	<b>0.846</b>	0	141700	> 600	1.946	0.833	0	62657	> 600
6	10	1.947	<b>0.868</b>	0	31200	> 600	1.947	0.864	0	68429	> 600
Avg		1.113	0.623	0	375700	533.28	1.138	0.613	0	280500	547.62

**Table EC.4** Performance of branch-and-bound without warmstart on the pitprops dataset ( $p = 13$ ) using the experimental setup laid out in Section 5.2, except we use a time limit of 600s for branch-and-bound. We report the performance of branch-and-bound with and without the upper bound developed in Section 5.1 separately. The column “UB” reports the upper bound obtained by the branch-and-bound scheme at the time limit. We use > 600 to denote an instance where branch-and-bound terminates at the 600s time limit. We denote the best-performing solution (in terms of the proportion of variance explained minus the total orthogonality constraint violation) in bold

(cont.).

$r$	$k_t$	Branch-and-Bound (warm-start)					Branch-and-Bound with (15) (warm-start)				
		UB	Obj.	Viol.	Nodes	T(s)	UB	Obj.	Viol.	Nodes	T(s)
2	2	0.295	<b>0.295</b>	0	6400	10.90	0.295	<b>0.295</b>	0	143	19.42
	4	0.404	<b>0.404</b>	0	75200	59.49	0.404	<b>0.404</b>	0	88735	238.5
	6	0.521	0.453	0	1051700	> 600	0.524	0.445	0	744000	> 600
	8	0.598	0.463	0	826700	> 600	0.604	0.462	0	1447055	> 600
	10	0.633	0.489	0	686100	> 600	0.636	0.487	0	687944	> 600
3	2	0.435	<b>0.435</b>	0	48600	17.92	0.435	<b>0.435</b>	0	2490	176.8
	4	0.716	0.536	0	491800	> 600	0.752	0.524	0	240207	> 600
	6	0.863	0.560	0	470900	> 600	0.855	0.567	0	598767	> 600
	8	0.935	0.578	0	383000	> 600	0.937	0.566	0	239381	> 600
	10	0.963	0.603	0	297600	> 600	0.963	0.595	0	355350	> 600
4	2	0.554	<b>0.554</b>	0	604900	> 600	0.554	<b>0.544</b>	0	29467	> 600
	4	1.107	0.627	0	264300	> 600	1.126	0.646	0	136767	> 600
	6	1.208	0.633	0	250300	> 600	1.220	0.625	0	247504	> 600
	8	1.273	0.676	0	225600	> 600	1.271	0.651	0	216736	> 600
	10	1.289	0.702	0	210000	> 600	1.291	0.712	0	165362	> 600
5	2	0.907	<b>0.656</b>	0	279800	> 600	1.132	0.616	0	28514	> 600
	4	1.419	0.718	0	251000	> 600	1.475	0.680	0	141932	> 600
	6	1.549	0.699	0	225900	> 600	1.555	0.677	0	229664	> 600
	8	1.609	0.743	0	223700	> 600	1.603	0.743	0	200306	> 600
	10	1.619	0.800	0	126300	> 600	1.621	0.794	0	135142	> 600
6	2	1.285	<b>0.749</b>	0	253200	> 600	1.647	<b>0.749</b>	0	11818	> 600
	4	1.821	0.778	0	89900	> 600	1.859	0.737	0	141476	> 600
	6	1.891	0.780	0	185600	> 600	1.906	0.768	0	126387	> 600
	8	1.927	0.833	0	124300	> 600	1.942	0.856	0	91093	> 600
	10	1.948	0.867	0	43200	> 600	1.948	0.866	0	82759	> 600
Avg		1.111	0.625	0	307800	> 600	1.150	0.618	0	255600	> 600

**Table EC.5** Performance of branch-and-bound with warmstart on the pitprops dataset ( $p = 13$ ) using the experimental setup laid out in Section 5.2, except we use a time limit of 600s for branch-and-bound. We report the performance of branch-and-bound with and without the upper bound developed in Section 5.1 separately. We use > 600 to denote an instance where branch-and-bound terminates at the 600s time limit. We denote the best-performing solution (in terms of the proportion of variance explained minus the orthogonality violation) in bold (cont.).

We observe that including the combinatorial upper bound developed in Section 3 within the branch-and-bound scheme does more harm than good, and therefore we do not consider combining the combinatorial bound with branch-and-bound elsewhere.

Furthermore, we observe that the upper bound returned by branch-and-bound outperforms the semidefinite upper bound from Problem (12) for the smallest combinations of  $r$  and  $k$ , but rapidly becomes worse as  $k$  and  $r$  increases, to the extent that it is unable to provide an upper bound better than the trivial bound of 1 for the largest combinations of  $r$  and  $k$ . This suggests that the upper bound from branch-and-bound is not practically useful for larger problem instances, and therefore we do not consider branch-and-bound beyond this dataset as a method for obtaining upper bounds.

### **EC.5.5. Instance-Wise Results on Larger UCI Datasets**

Next, we provide an instance-by-instance account of the results summarized in Table 2, in Tables EC.6–EC.12. Subsequently, we report the instancewise duality gap between the upper bound obtained by Algorithm 1 and the objective value obtained by each method, in Tables EC.12–EC.15

Dataset	$p$	$r$	$k_t$	Alg. 1				Alg. 2			Branch-and-bound		
				UB	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Pitprops	13	2	5	0.449	0.429	0	21.27	0.433	0	5.01	<b>0.439</b>	0	11291
		2	10	0.507	0.464	0	0.82	<b>0.500</b>	0	4.1	0.498	0	> 7200
		3	5	0.616	0.568	0	1.11	<b>0.582</b>	0	6.48	0.555	0	> 7200
		3	10	0.652	0.569	0	1.57	<b>0.650</b>	0	6.41	0.618	0	> 7200
Wine	13	2	5	0.458	0.429	0	0.73	0.446	0	2.19	<b>0.448</b>	0	1073
		2	10	0.554	0.508	0	0.78	<b>0.544</b>	0	4.03	0.537	0	> 7200
		3	5	0.632	0.577	0	1.66	<b>0.613</b>	0	3.56	0.576	0	> 7200
		3	10	0.665	0.580	0	1.43	<b>0.660</b>	0	6.47	0.640	0	> 7200
Ionosphere	34	2	5	0.209	0.202	0	7.78	<b>0.204</b>	0	3.18	0.202	0	> 7200
		2	10	0.305	0.275	0	8.44	<b>0.285</b>	0	3.95	<b>0.285</b>	0	> 7200
		2	20	0.378	0.357	0	10.84	<b>0.360</b>	0	7.64	0.344	0	> 7200
		3	5	0.297	<b>0.289</b>	0	13.54	0.279	0	6.72	<b>0.289</b>	0	> 7200
		3	10	0.411	<b>0.402</b>	0	15.01	0.397	0	11.89	0.375	0	> 7200
		3	20	0.464	0.408	0	60.95	<b>0.458</b>	0	12.17	0.383	0	> 7200
Lung	54	2	5	0.119	<b>0.119</b>	0	42.17	0.110	0	3.81	0.113	0	> 7200
		2	10	0.176	0.175	0	28.12	0.171	0	2.06	0.168	0	> 7200
		2	20	0.234	<b>0.222</b>	0	36.74	0.217	0	4.12	0.185	0	> 7200
		3	5	0.173	<b>0.172</b>	0	90.07	0.160	0	4.45	0.169	0	> 7200
		3	10	0.249	0.226	0	55.09	0.240	0	4.31	0.188	0	> 7200
		3	20	0.324	0.271	0	55.49	<b>0.303</b>	0	9.51	0.213	0	> 7200
Geography	68	2	5	0.147	0.144	0	83.2	<b>0.147</b>	0	2.99	0.145	0	> 7200
		2	10	0.294	0.290	0	51.02	<b>0.294</b>	0	1.81	0.292	0	> 7200
		2	20	0.433	<b>0.395</b>	0	942.1	0.376	0	6.32	0.327	0	> 7200
		3	5	0.221	0.216	0	80.63	<b>0.221</b>	0	2.58	0.215	0	> 7200
		3	10	0.410	0.345	0	2680	0.348	0	5.39	0.355	0	> 7200
		3	20	0.529	<b>0.444</b>	0	1237	0.345	0	11.51	0.352	0	> 7200
Communities	101	2	5	0.095	<b>0.095</b>	0	1088	0.078	0	3.73	<b>0.095</b>	0	> 7200
		2	10	0.169	<b>0.169</b>	0	2601	0.159	0	4.81	0.160	0	> 7200
		2	20	0.268	0.244	0	2427	0.244	0	6.26	0.198	0	> 7200
		3	5	0.141	<b>0.141</b>	0	2571	0.119	0	7.63	<b>0.141</b>	0	> 7200
		3	10	0.246	0.244	0	4219	0.243	0	8.92	0.205	0	> 7200
		3	20	0.386	0.306	0	4609	<b>0.370</b>	0	8.28	0.300	0	> 7200
Arrhythmia	274	2	5	0.031	0.030	0	503.7	0.027	0	16.57	0.027	0	> 7200
		2	10	0.055	0.049	0	493.5	0.047	0	24.72	0.044	0	> 7200
		2	20	0.086	0.071	0	688.1	0.071	0.002	49.41	0.059	0	> 7200
		3	5	0.047	0.045	0	1411	0.039	0	27.99	0.044	0	> 7200
		3	10	0.083	0.072	0	958.9	0.067	0	38.16	0.065	0	> 7200
		3	20	0.129	0.102	0	1241	0.105	0	28.96	0.068	0	> 7200

**Table EC.6** Performance of our three algorithms proposed in Section 4 on UCI datasets.  $k_t$  denotes the sparsity of each individual component, meaning a set of  $r$  PCs have a collective sparsity budget of  $k_t r$ . Note that all objective values are reported in terms of the proportion of correlation explained by dividing by  $p$ , the number of features.



Dataset	$p$	$r$	$k_t$	Alg. 1				Alg. 2			Branch-and-bound		
				UB	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Voice	310	2	5	0.032	<b>0.032</b>	0	699.4	<b>0.032</b>	0	21.14	<b>0.032</b>	0	> 7200
		2	10	0.064	<b>0.064</b>	0	741.2	0.063	0	21.99	<b>0.064</b>	0	> 7200
		2	20	0.127	<b>0.127</b>	0	630.6	0.124	0	20.97	<b>0.109</b>	0	> 7200
		3	5	0.048	<b>0.048</b>	0	1225	0.047	0	29.94	<b>0.048</b>	0	> 7200
		3	10	0.096	<b>0.096</b>	0	1372	0.093	0	34.52	<b>0.096</b>	0	> 7200
		3	20	0.191	<b>0.190</b>	0	1195	0.183	0	31.06	0.155	0	> 7200
Gait	320	2	5	0.031	<b>0.030</b>	0	854.6	0.028	0	19.40	0.028	0	> 7200
		2	10	0.057	<b>0.054</b>	0	840.7	0.050	0	24.64	0.047	0	> 7200
		2	20	0.103	<b>0.096</b>	0	918.1	0.081	0	24.51	0.067	0	> 7200
		3	5	0.046	<b>0.045</b>	0	1548	0.041	0	27.80	<b>0.045</b>	0	> 7200
		3	10	0.085	0.081	0	1437	0.077	0	34.99	0.060	0	> 7200
		3	20	0.154	<b>0.135</b>	0	1306	0.121	0	34.11	0.111	0	> 7200
Gastro	466	2	5	0.021	<b>0.021</b>	0	3557	<b>0.021</b>	0	452.9	<b>0.021</b>	0	> 7200
		2	10	0.043	<b>0.043</b>	0	4242	<b>0.043</b>	0	38.95	<b>0.043</b>	0	> 7200
		2	20	0.086	<b>0.086</b>	0	4485	0.085	0	535.4	<b>0.086</b>	0	> 7200
		3	5	0.032	<b>0.032</b>	0	4974	<b>0.032</b>	0	59.97	<b>0.032</b>	0	> 7200
		3	10	0.064	<b>0.064</b>	0	5859	<b>0.064</b>	0	73.39	<b>0.064</b>	0	> 7200
		3	20	0.129	<b>0.128</b>	0	6602	0.121	0	529.8	<b>0.128</b>	0	> 7200
Micromass	1300	2	5	0.008	0.004	0	13870	0.006	0	170.5	0.004	0	> 7200
		2	10	0.015	0.008	0	10440	0.011	0	158.6	0.011	0	6100
		2	20	0.027	0.017	0	9515	0.019	0	440.3	0.018	0	6826
		3	5	0.012	0.009	0	19640	0.010	0	238.0	0.006	0	> 7200
		3	10	0.023	0.010	0	17820	0.018	0	208.5	0.009	0	> 7200
		3	20	0.043	0.015	0	18370	0.030	0	205.9	0.010	0	> 7200
Avg				0.213	0.195	0.000	2588	0.199	0.000	61.38	0.187	0.000	> 7200

**Table EC.7** Performance of our three algorithms proposed in Section 4 on UCI datasets (cont).  $k_t$  denotes the sparsity of each individual component, meaning a set of  $r$  PCs have a collective sparsity budget of  $k_t r$ . Note that all objective values are reported in terms of the proportion of correlation explained by dividing by  $p$ , the number of features.

Dataset	$p$	$r$	$k_t$	Berk and Bertsimas (2019)			Hein and Bühler (2010)			Zou et al. (2006)		
				Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Pitprops	13	2	5	0.421	0.168	1.67	0.418	0	0.11	0.177	1.341	0.12
			10	0.502	0.008	0.14	0.502	0.008	0.01	0.139	1.827	0.22
			3	0.592	0.675	0.08	0.575	0.166	0.02	0.169	3.462	0.04
			3	0.648	0.073	0.07	0.647	0.084	0	0.181	3.771	0.36
Wine	13	2	5	<b>0.448</b>	0	0.04	0.422	0.004	0.01	0.127	0.315	0.04
			10	0.545	0.020	0.04	0.545	0.02	0	0.068	0.731	0.06
			3	0.610	0.019	0.06	0.559	0.092	0	0.225	2.830	0.05
			3	0.654	0.059	0.06	0.655	0.093	0	0.232	2.771	0.32
Ionosphere	34	2	5	<b>0.205</b>	0	0.08	0.153	0	0.08	0.078	0	0.02
			10	<b>0.289</b>	0	0.30	0.288	0	0.01	0.106	0	0.04
		2	20	0.369	0.058	4.45	<b>0.370</b>	0.010	0.17	0.147	0.305	0.12
			3	0.291	0	0.14	0.227	0	0.02	0.097	1.666	0.07
		3	10	0.392	0.109	0.38	0.365	0.255	0.01	0.100	1.909	0.12
			3	0.449	0.183	0.27	0.451	0.037	0.03	0.111	2.111	4.51
Lung	54	2	5	<b>0.119</b>	0	0.43	0.107	0	0.34	0.040	0.587	0.04
			10	<b>0.176</b>	0	0.10	0.170	0	0.03	0.044	0.639	0.12
		2	20	0.220	0.008	0.44	0.184	0	0.05	0.044	0.908	0.63
			3	<b>0.172</b>	0	0.10	0.149	0	0.03	0.061	2.755	0.15
		3	10	<b>0.243</b>	0	0.11	0.234	0.113	0.05	0.054	1.593	0.20
			3	0.300	0.219	0.16	0.261	0.081	0.04	0.044	1.703	1.20
Geography	68	2	5	<b>0.147</b>	0	0.09	0.097	0	0.01	0.034	1.793	0.41
			10	<b>0.294</b>	0	0.08	0.164	0	0	0.068	1.939	0.66
		2	20	<b>0.395</b>	0	5.95	0.316	0.135	0.04	0.062	1.754	0.53
			3	<b>0.221</b>	0	0.13	0.122	0	0.01	0.061	2.720	0.57
		3	10	<b>0.389</b>	0	0.18	0.192	0	0.01	0.054	4.021	0.90
			3	0.484	0.273	23.66	0.387	0.261	0.06	0.090	5.009	1.40
Communities	101	2	5	<b>0.095</b>	0	0.73	0.093	0	0	0.032	0.576	0.05
			10	<b>0.169</b>	0	1.68	0.154	0	0	0.029	0.605	0.18
		2	20	<b>0.258</b>	0	120	<b>0.258</b>	0	0.07	0.027	0.090	1.49
			3	<b>0.141</b>	0	1.18	0.129	0	0.01	0.050	1.854	0.29
		3	10	<b>0.245</b>	0	2.85	0.181	0	0.02	0.044	1.504	1.76
			3	0.361	0.058	180.1	0.350	0.064	0.02	0.043	1.869	5.27
Arrhythmia	274	2	5	<b>0.031</b>	0	2.81	0.012	0	0.02	0.007	1.799	0.71
			10	<b>0.052</b>	0	61.25	0.011	0	0.03	0.007	1.143	1.08
		2	20	<b>0.077</b>	0	120.0	0.043	0.005	0.06	0.006	1.140	4.62
			3	<b>0.046</b>	0	5.35	0.016	0	0.02	0.012	1.076	0.53
		3	10	<b>0.074</b>	0	121.6	0.018	0	0.05	0.012	0.876	3.82
			3	<b>0.109</b>	0	180.0	0.074	0.005	0.07	0.012	0.694	10.65

**Table EC.8** Performance of the methods by Berk and Bertsimas (2019), Hein and Bühler (2010), and Zou et al. (2006) on UCI datasets.

Dataset	$p$	$r$	$k_t$	Berk and Bertsimas (2019)			Hein and Bühler (2010)			Zou et al. (2006)		
				Obj.	Viol.	T(s)	Obj.	Viol.	T(s)	Obj.	Viol.	T(s)
Voice	310	2	5	<b>0.032</b>	0	1.04	<b>0.032</b>	0	0.05	0.006	0.874	0.71
			10	<b>0.064</b>	0	1.09	<b>0.064</b>	0	0.04	0.006	0.907	2.68
		3	5	<b>0.048</b>	0	1.72	0.039	0	0.16	0.009	1.834	0.94
			10	<b>0.096</b>	0	1.56	0.069	0	0.08	0.012	0.242	12.8
		3	20	<b>0.190</b>	0	1.22	0.187	0	0.12	0.021	2.121	26.18
Gait	320	2	5	<b>0.030</b>	0	0.93	0.027	0	0.02	0.006	1.071	1.75
			10	<b>0.055</b>	0	0.61	0.051	0	0.05	0.004	1.054	1.27
		3	5	<b>0.045</b>	0	1.02	0.041	0	0.06	0.01	1.16	4.81
			10	<b>0.082</b>	0	1.64	0.070	0	0.06	0.009	1.821	5
		3	20	<b>0.135</b>	0	1.94	0.095	0	0.15	0.008	1.477	9.49
Gastro	466	2	5	<b>0.021</b>	0	2.71	0.020	0	0.05	0.007	1.154	1.14
			10	<b>0.043</b>	0	1.42	0.039	0	0.08	0.007	0.178	1.62
		3	5	<b>0.032</b>	0	2.79	0.029	0	0.05	0.006	2.303	1.59
			10	<b>0.064</b>	0	3.51	0.053	0	0.12	0.007	1.826	3.33
		3	20	<b>0.128</b>	0	2.81	0.085	0	0.24	0.008	1.139	46.05
Micromass	1300	2	5	<b>0.008</b>	0	45.4	0.004	0	0.77	0.002	0.014	18.05
			10	<b>0.014</b>	0	120.2	0.007	0	1.09	0.002	0.323	41.97
		3	5	<b>0.011</b>	0	71.97	0.005	0	1.54	0.002	3.004	24.30
			10	<b>0.020</b>	0	180.3	0.008	0	1.80	0.002	2.301	31.89
		3	20	<b>0.034</b>	0	180.3	0.013	0	2.64	0.002	1.252	2.02
Avg				0.205	0.031	25.57	0.180	0.023	0.19	0.049	1.458	4.95

**Table EC.9** Performance of the methods by Berk and Bertsimas (2019), Hein and Bühler (2010), and Zou et al. (2006) on UCI datasets (cont.).

Dataset	$p$	$r$	$k_t$	Deshpande and Montanari (2014b)		T(s)
				Obj.	Viol.	
Pitprops	13	2	5	0.422	0.226	1.00
		2	10	0.501	0.104	0.00
		3	5	0.592	0.661	0.00
		3	10	0.644	0.214	0.00
Wine	13	2	5	0.434	0.211	0.02
		2	10	0.545	0.021	0
		3	5	0.546	0.510	0
		3	10	0.656	0.228	0
Ionosphere	34	2	5	0.203	0	0.12
		2	10	<b>0.287</b>	0	0.02
		2	20	0.368	0.011	0.02
		3	5	0.276	0	0.02
		3	10	0.357	0.143	0.02
		3	20	0.447	0.166	0.02
Lung	54	2	5	0.117	0	0.45
		2	10	0.154	0.291	0.04
		2	20	0.217	0.495	0.03
		3	5	0.164	0	0.03
		3	10	0.200	0.558	0.03
		3	20	0.290	0.591	0.03
Geography	68	2	5	0.130	0	0.09
		2	10	0.294	0	0.07
		2	20	0.395	0	0.08
		3	5	0.184	0	0.08
		3	10	0.378	0	0.07
		3	20	0.462	0.073	0.08
Communities	101	2	5	0.089	0	0.19
		2	10	0.158	0	0.23
		2	20	0.249	0.145	0.19
		3	5	0.140	0	0.19
		3	10	0.208	0.138	0.20
		3	20	0.317	0.669	0.19
Arrhythmia	274	2	5	0.030	0	1.79
		2	10	0.051	0.113	1.84
		2	20	0.074	0.019	1.88
		3	5	0.039	0	1.97
		3	10	0.072	0	2.35
		3	20	0.103	0.243	2.23

**Table EC.10** Performance of the method of Deshpande and Montanari (2014b) on UCI datasets.

Dataset	$p$	$r$	$k_t$	Deshpande and Montanari (2014b)		
				Obj.	Viol.	T(s)
Voice	310	2	5	<b>0.032</b>	0	60.46
		2	10	0.063	0	54.59
		2	20	0.124	0	39.19
		3	5	0.047	0	1.42
		3	10	0.093	0	1.75
		3	20	0.184	0	1.42
Gait	320	2	5	<b>0.030</b>	0	1.45
		2	10	0.054	0	1.45
		2	20	0.089	0	1.42
		3	5	0.042	0	1.85
		3	10	0.078	0	1.65
		3	20	0.128	0	3.04
Gastro	466	2	5	<b>0.021</b>	0	6.59
		2	10	0.042	0	4.33
		2	20	0.083	0	3.67
		3	5	0.031	0	3.26
		3	10	0.061	0	3.12
		3	20	0.117	0	3.11
Micromass	1300	2	5	0.007	0	75.38
		2	10	0.012	0	147.9
		2	20	0.022	0	98.34
		3	5	0.010	0	71.05
		3	10	0.018	0	71.88
		3	20	0.033	0	73.00
Avg				0.197	0.094	12.04

**Table EC.11** Performance of the method of Deshpande and Montanari (2014b) on UCI datasets. (cont.).

Dataset	$p$	$r$	$k_t$	Algorithm 1	Algorithm 2	Branch-and-Bound	Berk and Bertsimas (2019)
Pitprops	13	2	5	4.29%	3.55%	2.07%	6.05%
		2	10	8.60%	1.40%	1.83%	1.14%
		3	5	7.81%	5.65%	9.89%	3.93%
		3	10	12.65%	0.36%	5.16%	0.58%
Wine	13	2	5	6.43%	2.72%	2.18%	2.18%
		2	10	8.28%	1.89%	3.13%	1.58%
		3	5	8.64%	2.93%	8.83%	3.50%
		3	10	12.81%	0.83%	3.82%	1.63%
Ionosphere	34	2	5	2.91%	2.30%	2.91%	1.49%
		2	10	9.68%	6.54%	6.25%	5.23%
		2	20	5.44%	4.79%	8.97%	2.22%
		3	5	2.77%	6.15%	2.79%	1.99%
		3	10	2.34%	3.34%	8.7%	4.57%
		3	20	12.03%	1.27%	17.42%	3.11%
Lung	54	2	5	0%	7.88%	5.44%	0%
		2	10	0.16%	2.65%	4.20%	0.12%
		2	20	4.97%	7.07%	20.83%	6.03%
		3	5	0.72%	7.41%	2.55%	0.72%
		3	10	9.20%	3.66%	24.75%	2.75%
		3	20	16.09%	6.26%	34.09%	7.29%
Geography	68	2	5	2.06%	0%	1.36%	0%
		2	10	1.35%	0%	0.75%	0%
		2	20	8.79%	13.15%	24.43%	8.77%
		3	5	2.28%	0%	2.74%	0%
		3	10	15.87%	15.19%	13.5%	5.09%
		3	20	16.15%	34.86%	33.56%	8.60%
Communities	101	2	5	0.15%	17.79%	0.16%	0.15%
		2	10	0.05%	5.88%	5.08%	0.05%
		2	20	8.98%	9.00%	26.04%	3.74%
		3	5	0.28%	16.01%	0.30%	0.28%
		3	10	0.60%	1.12%	16.59%	0.33%
		3	20	20.71%	4.07%	22.13%	6.29%

**Table EC.12** Instancewise relative optimality gap between the upper bound obtained by Algorithm 1 and the objective value obtained by each feasible method (for branch-and-bound, note that the upper bound used differs from the upper bound computed by Gurobi). Note that certain methods return infeasible solutions on some instances, and when this occurs, the optimality gap may be small (or even negative) partly because of this infeasibility.

Dataset	$p$	$r$	$k_t$	Algorithm 1	Algorithm 2	Branch-and-Bound	Berk and Bertsimas (2019)
Arrhythmia	274	2	5	5.02%	12.88%	14.13%	0.92%
			10	11.31%	14.32%	20.19%	5.91%
		3	5	4.49%	16.55%	6.54%	1.64%
			10	13.02%	18.52%	21.12%	10.59%
		3	20	20.64%	18.52%	47.48%	15.03%
Voice	310	2	5	0.01%	1.55%	0.01%	0.01%
			10	0.86%	2.53%	0.86%	0.17%
		3	5	0.01%	2.09%	0.01%	0.01%
			10	0.28%	3.84%	0.28%	0.27%
		3	20	0.65%	4.13%	19.07%	0.81%
Gait	320	2	5	1.14%	8.03%	9.91%	1.82%
			10	4.92%	11.12%	17.32%	2.56%
		3	5	1.72%	11.05%	1.73%	1.72%
			10	4.98%	10.13%	29.87%	3.91%
		3	20	12.42%	21.77%	27.79%	12.40%
Gastro	466	2	5	0.05%	0.12%	0.05%	0%
			10	0.01%	0.24%	0.01%	0.01%
		3	5	0.04%	0.13%	0.04%	0%
			10	0.02%	0.17%	0.02%	0.02%
		3	20	0.27%	6.25%	0.28%	0.22%
Micromass	1300	2	5	49.96%	17.44%	50.67%	1.42%
			10	47.65%	28.42%	23.94%	8.63%
		3	5	20.57%	14.85%	44.73%	1.57%
			10	55.31%	22.90%	58.40%	10.38%
		3	20	65.44%	29.13%	76.96%	21.06%

**Table EC.13** Instancewise relative optimality gap between the upper bound obtained by Algorithm 1 and the objective value obtained by each feasible method (for branch-and-bound, note that the upper bound used differs from the upper bound computed by Gurobi). Note that certain methods return infeasible solutions on some instances, and when this occurs, the optimality gap may be small partly because of this infeasibility.

Dataset	$p$	$r$	$k_t$	Hein and Bühler (2010)	Zou et al. (2006)	Deshpande and Montanari (2014a)
Pitprops	13	2	5	6.92%	60.58%	6.01%
		2	10	1.14%	72.59%	1.34%
		3	5	6.65%	72.51%	4.03%
		3	10	0.70%	72.18%	1.17%
Wine	13	2	5	7.86%	72.36%	5.38%
		2	10	1.58%	87.8%	1.61%
		3	5	11.56%	64.46%	13.57%
		3	10	1.59%	65.08%	1.36%
Ionosphere	34	2	5	26.57%	62.51%	2.72%
		2	10	5.34%	65.35%	5.82%
		2	20	2.17%	61.14%	2.51%
		3	5	23.56%	67.29%	6.95%
		3	10	11.23%	75.63%	13.14%
		3	20	2.76%	76.07%	3.70%
Lung	54	2	5	10.57%	66.76%	1.9%
		2	10	3.20%	74.86%	12.26%
		2	20	21.24%	81.18%	7.44%
		3	5	13.94%	64.88%	5.14%
		3	10	6.15%	78.46%	19.75%
		3	20	19.19%	86.39%	10.39%
Geography	68	2	5	33.96%	77.21%	11.46%
		2	10	44.23%	77.04%	0%
		2	20	26.98%	85.70%	8.84%
		3	5	44.58%	72.51%	16.71%
		3	10	53.06%	86.88%	7.74%
		3	20	26.98%	83.02%	12.72%
Communities	101	2	5	2.31%	66.78%	6.81%
		2	10	8.46%	82.53%	6.46%
		2	20	3.85%	89.96%	6.92%
		3	5	8.77%	64.79%	0.63%
		3	10	26.24%	81.88%	15.29%
		3	20	9.39%	88.81%	17.78%

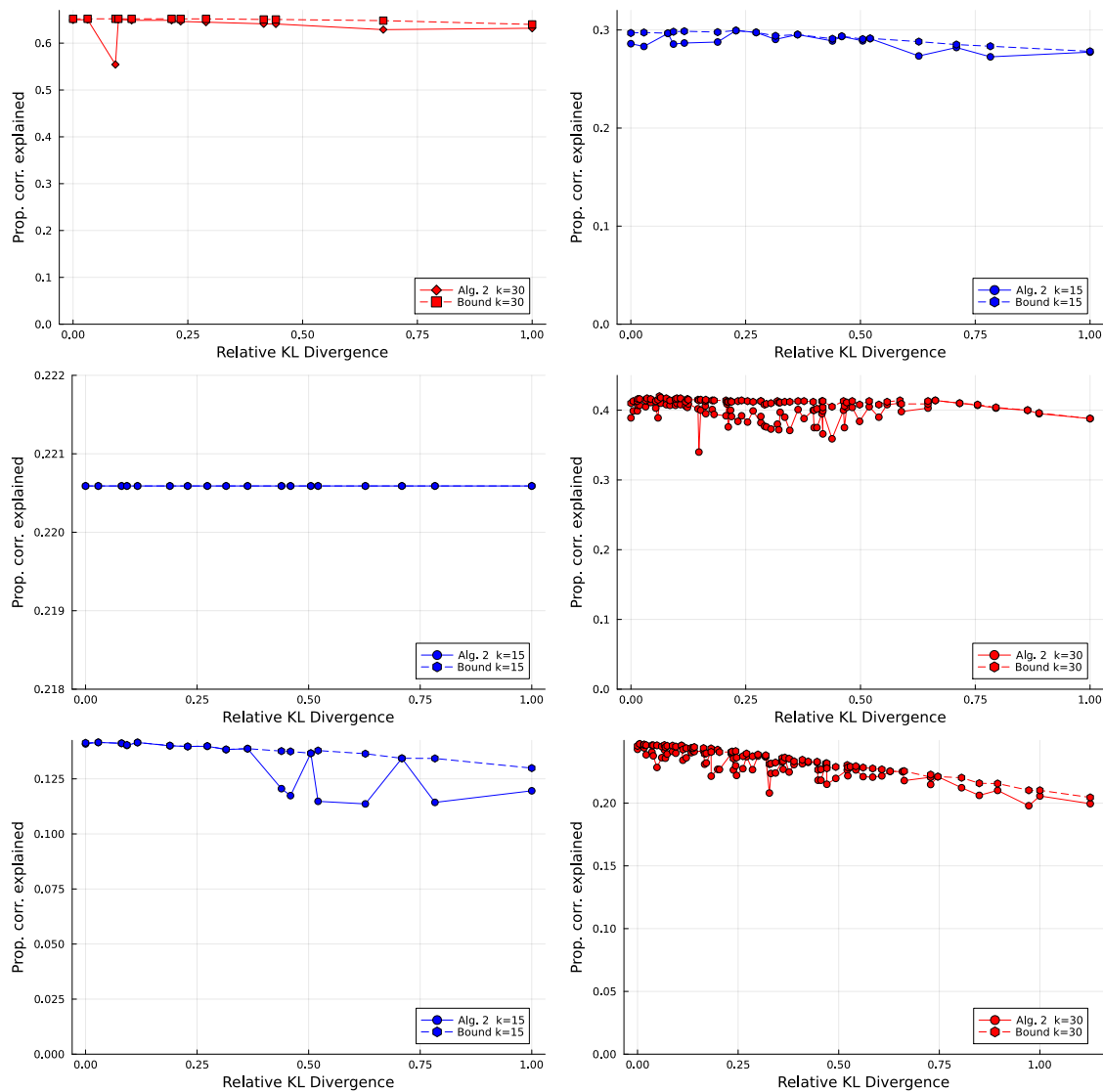
**Table EC.14** Instancewise relative optimality gap between the upper bound obtained by Algorithm 1 and the objective value obtained by each feasible method (cont).



Dataset	$p$	$r$	$k_t$	Hein and Bühler (2010)	Zou et al. (2006)	Deshpande and Montanari (2014a)
Arrhythmia	274	2	5	61.03%	78.56%	4.76%
			10	80.17%	87.44%	6.95%
		2	20	50.39%	92.57%	13.43%
			3	5	66.20%	74.80%
		10		78.13%	85.67%	12.45%
		20	42.49%	90.51%	20.33%	
Voice	310	2	5	0.62%	82.64%	0.71%
			10	0.84%	90.37%	2.27%
		2	20	0.13%	95.14%	3.05%
			3	5	19.64%	81.79%
		10		28.23%	87.4%	3.89%
		20	2.43%	88.96%	3.84%	
Gait	320	2	5	13.13%	81.89%	2.75%
			10	9.47%	92.97%	4.47%
		2	20	22.45%	94.82%	13.3%
			3	5	10.19%	77.63%
		10		17.54%	88.96%	8.04%
		20	38.17%	94.77%	17.11%	
Gastro	466	2	5	5.76%	66.34%	1.16%
			10	10.14%	84.56%	2.19%
		2	20	11.81%	94.45%	3.62%
			3	5	10.15%	80.11%
		10		17.35%	88.44%	5.13%
		20	33.91%	94.08%	9.20%	
Micromass	1300	2	5	41.78%	80.07%	11.94%
			10	50.11%	89.54%	17.45%
		2	20	57.05%	93.89%	18.99%
			3	5	54.53%	82.63%
		10		63.80%	91.41%	22.09%
		20	70.69%	95.07%	21.71%	

**Table EC.15** Instancewise relative optimality gap between the upper bound obtained by Algorithm 1 and the objective value obtained by each feasible method (cont).

### EC.5.6. Instance-Wise Plots of Symmetry vs. Proportion of Correlation Explained



**Figure EC.1** Symmetry of sparsity budget allocation vs. proportion of correlation in the dataset explained for pitprops  $k = 30$  (top left), ionosphere  $k = 15$  (top right), geographical  $k = 15$  (middle left), geographical  $k = 30$  (middle right), communities  $k = 15$  (bottom left), and communities  $k = 30$  (bottom right). Note that we normalize the KL divergence for  $k = 15$  and  $k = 30$  separately.