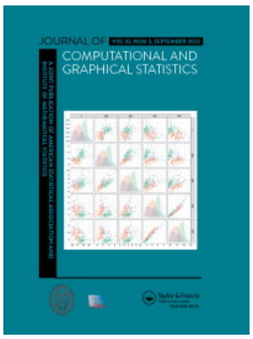

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

A New Basis for Sparse Principal Component Analysis

Fan Chen & Karl Rohe

To cite this article: Fan Chen & Karl Rohe (08 Sep 2023): A New Basis for Sparse Principal Component Analysis, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2023.2256502

To link to this article: https://doi.org/10.1080/10618600.2023.2256502

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2023, VOL. 00, NO. 0, 1–14
https://doi.org/10.1080/10618600.2023.2256502

A New Basis for Sparse Principal Component Analysis

Fan Chen and Karl Rohe

Department of Statistics, University of Wisconsin–Madison, Madison, WI

ABSTRACT

Previous versions of sparse principal component analysis (PCA) have presumed that the eigen-basis (a $p \times k$ matrix) is approximately sparse. We propose a method that presumes the $p \times k$ matrix becomes approximately sparse after a $k \times k$ rotation. The simplest version of the algorithm initializes with the leading k principal components. Then, the principal components are rotated with an $k \times k$ orthogonal rotation to make them approximately sparse. Finally, soft-thresholding is applied to the rotated principal components. This approach differs from prior approaches because it uses an orthogonal rotation to approximate a sparse basis. One consequence is that a sparse component need not to be a leading eigenvector, but rather a mixture of them. In this way, we propose a new (rotated) basis for sparse PCA. In addition, our approach avoids "deflation" and multiple tuning parameters required for that. Our sparse PCA framework is versatile; for example, it extends naturally to a two-way analysis of a data matrix for simultaneous dimensionality reduction of rows and columns. We provide evidence showing that for the same level of sparsity, the proposed sparse PCA method is more stable and can explain more variance compared to alternative methods. Through three applications—sparse coding of images, analysis of transcriptome sequencing data, and large-scale clustering of social networks, we demonstrate the modern usefulness of sparse PCA in exploring multivariate data. An R package, `epca`, and the supplementary materials for this article are available online.

ARTICLE HISTORY
Received January 2022
Accepted July 2023

KEYWORDS
Column sparsity; Dimensionality reduction; Independent component analysis; Orthogonal rotation; Sparse matrix decomposition

1. Introduction

Principal component analysis (PCA), introduced in the early 20th century (Pearson 1901; Hotelling 1933), is one of the most prevalent tools in exploratory multivariate data analysis. PCA projects higher-dimensional data into a lower-dimensional space that is spanned by some uncorrelated principal components (PCs), with the vast majority of the variance in the data kept. It is, however, commonly conceived that PCs are difficult to interpret (e.g., Jeffers 1967), as each PC is a linear combination of many, if not all, original variables. To remedy such disadvantage, sparse PCA estimates "sparse" PCs, each of which consists of a small subset of original variables (Zou and Xue 2018).

Sparse PCA is originally formulated as an optimization problem over the loading coefficients with a cardinality constraint. Such nonconvex constraint results in an NP-hard problem in the strong sense (Tillmann and Pfetsch 2014). In order to circumvent the obstacle, various methods have been proposed, such as the iconic regression-based approach by Zou, Hastie, and Tibshirani (2006), a convex relaxation to semidefinite programming (d'Aspremont et al. 2007), the penalized matrix decomposition framework of Witten, Tibshirani, and Hastie (2009), and the generalized power method due to Journée et al. (2010). More recently, theoretical developments of sparse PCA have covered the consistency (Johnstone and Lu 2009; Shen, Shen, and Marron 2013), variable selection properties (Amini and Wainwright 2009), rates of convergence, the minimaxity over some Gaussian or sub-Gaussian classes (Vu and Lei 2013; Cai, Ma, and Wu 2013), and the statistical-computational tradeoffs under the restricted covariance concentration condition (Berthet and Rigollet 2013; Wang, Berthet, and Samworth 2016).

Despite the extensive literature of sparse PCA, there are two enigmas. First, sparse PCA often explains far less variance in the data than PCA does (Figure 1). While this may appear to be a tradeoff for sparsity, our results show that a substantial improvement is possible. Second, the most common formulations of sparse PCA rely on a matrix deflation after estimating each component. The deflation entails complications of multiple tuning parameters, nonorthogonality, and sub-optimality (Mackey 2008). Identifiability and consistency present more subtle issues; there is no reason to assume a priori distinct eigenvalues or that the gaps between the eigenvalues are small (Vu et al. 2013). Estimating the subspace spanned by multiple sparse PCs at once overcomes this dilemma (Vu et al. 2013).

There are two distinct notions of subspace sparsity: row sparsity and column sparsity (Vu and Lei 2013). Contemporary approaches to sparse PCA primarily focus on row sparsity, which implies that the eigenvectors of the covariance matrix themselves are sparse (e.g., Moghaddam, Weiss, and Avidan 2006). The second notion, column sparsity, is an alternative. A column sparse subspace "is one which has some orthogonal basis con-

CONTACT Fan Chen *fan.chen@wisc.edu* Department of Statistics, University of Wisconsin–Madison, Madison, WI.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JCGS*.

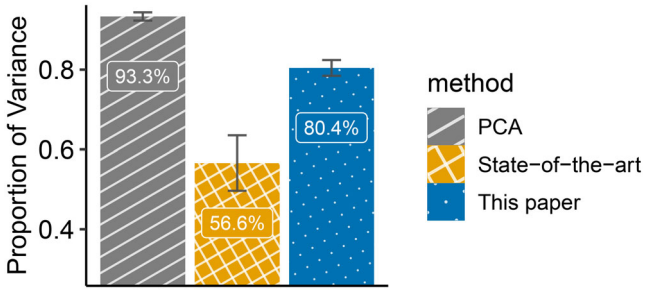

Figure 1. Comparison of the proportion of variance explained (PVE) by the 16 PCs estimated by PCA (gray), GPower (yellow, see Journée et al. (2010)), and the proposed sparse PCA method (blue). For each method, an error bar (based on the three-sigma rule) depicts the variation of PVE over 30 repeats of experiments. More details about the simulated data and settings (e.g., sparsity constraints) are described in Section 4.1

sisting of sparse vectors. This means that the choice of basis is crucial; the existence of a sparse basis is an implicit assumption behind the frequent use of rotation techniques by practitioners to help interpret principal components" (Vu and Lei 2013). Row sparsity is the most prevalent notion of sparsity used in contemporary sparse PCA, yet it does not appear to describe many contemporary parametric multivariate models; conversely, many contemporary parametric models in multivariate statistics can be estimated with the sparse PCA approaches that can identify column sparsity (Rohe and Zeng 2020).

In high-dimensional regression, sparse penalties such as the Lasso resolve an invariance; there is an entire space of solutions b which exactly interpolate the data $Y = Xb$ and presuming that the solution b is sparse can make the solution unique. Interestingly, there is no analogue to "sparsity resolving an invariance" for the estimation of row sparse subspace, but there is a very clear analogue in estimating column sparse subspace; the basis is determined by the one that provides the most sparse representation of data.

1.1. Our Contributions

In this work, we propose a new method, sparse component analysis (SCA), to estimate multiple PCs that are column sparse. The column sparsity is achieved by allowing an orthogonal rotation to PCs prior to imposing any sparsity constraints. The algorithm is motivated by two facts. First, an orthogonal rotation does not affect the total variance explained by a given set of PCs. Second, by choosing the orthogonal rotation carefully, PCs can be aligned closely with the coordinate axes, making them approximately sparse (Figure 2). This technique has been commonly adapted in factor analysis, a close cousin of PCA (Thurstone 1931; Kaiser 1960; Jolliffe 1995). For example, the varimax rotation (Kaiser 1958) is a popular choice in the psychology literature. SCA incorporates the orthogonal rotation and sparsity constraints to find the sparse and orthogonal basis in a subspace (i.e., column sparse PCs). We show in Proposition 1 (Section 2.1.2) that

column sparse PCs can explain more variance in the data than row sparse PCs.

We validated this with numerical experiments. Additionally, the simulations suggest that SCA is more stable and robust across tuning parameters than existing sparse PCA methods. Our framework of SCA generalizes naturally to a two-way analysis of a data matrix for simultaneous row and column dimensionality reductions. For this, we introduce a low-rank matrix approximation method called sparse matrix approximation (SMA). The SMA builds on the penalized matrix decomposition previously proposed by Witten, Tibshirani, and Hastie (2009). Furthermore, the SMA provides a unified view of sparse PCA and other modern multivariate data analysis, including sparse independent component analysis (see, e.g., Comon 1994). Finally, we demonstrate our sparse PCA methods with various high-dimensional data applications, including sparse coding of images, blind source separation, analysis of single-cell transcriptome data, and large-scale clustering of social networks. We find compelling evidence for the practical use of our approach, despite concerns about the consistency of PCA in high-dimensions.

1.2. Organization

The rest of this article goes as follows. Section 2 describes the methods. Section 3 compares SCA to existing methods. Section 4 compares different sparse PCA methods using simulated data. Section 5 applies SCA to several high-dimensional datasets. Section 6 concludes the article with some discussions.

1.3. Notations

In this article, we discuss the *entrywise* matrix norm only. For any matrix $A \in \mathbb{R}^{m\times n}$, its entrywise ℓ_p-norm is defined as $\|A\|_{p,p} = (\sum_{i=1}^m \sum_{j=1}^n |A_{ij}|^p)^{1/p}$. For simplicity, we also use the notation $\|A\|_p$ for entrywise norm, rather than the norm induced by a vector norm. In particular, the Frobenius norm (or the Hilbert-Schmidt norm) is then an alias of entrywise ℓ_2-norm, $\|A\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2} = \|A\|_2$. Throughout, the following sets of matrices are frequently considered. $\mathcal{U}(n) = \{U \in \mathbb{R}^{n\times n} \mid U^{\mathrm{T}}U = UU^{\mathrm{T}} = I_n\}$ denotes all orthogonal (unitary) matrices in $\mathbb{R}^n$. $\mathcal{V}(n,k) = \{V \in \mathbb{R}^{n\times k} \mid V^{\mathrm{T}}V = I_k\}$ represents the Stiefel manifold in $\mathbb{R}^n$, and $\mathcal{B}(n,k) = \{V \in \mathbb{R}^{n\times k} \mid V^{\mathrm{T}}V \preceq I_k\}$ is its convex hull (Gallivan and Absil 2010).

2. The Methods

We present a new formulation of sparse PCA as follows. After revisiting PCA, we give the new formulation (2) (Section 2.1) and elaborate how it represents column sparsity (Section 2.1.1) and how it outperforms a row sparsity based method (Section 2.1.2). Next, we present an iterative algorithm to compute sparse PCA (Section 2.2). Lastly, we apply the column sparsity concept to a more general matrix decomposition method (Section 2.3).

Consider the data matrix $X \in \mathbb{R}^{n\times p}$ of n observations (or samples) on p variables. Without loss of generality, we assume that each column of X is centered (i.e., mean-zero) unless otherwise noted. Throughout this article, we presume the number

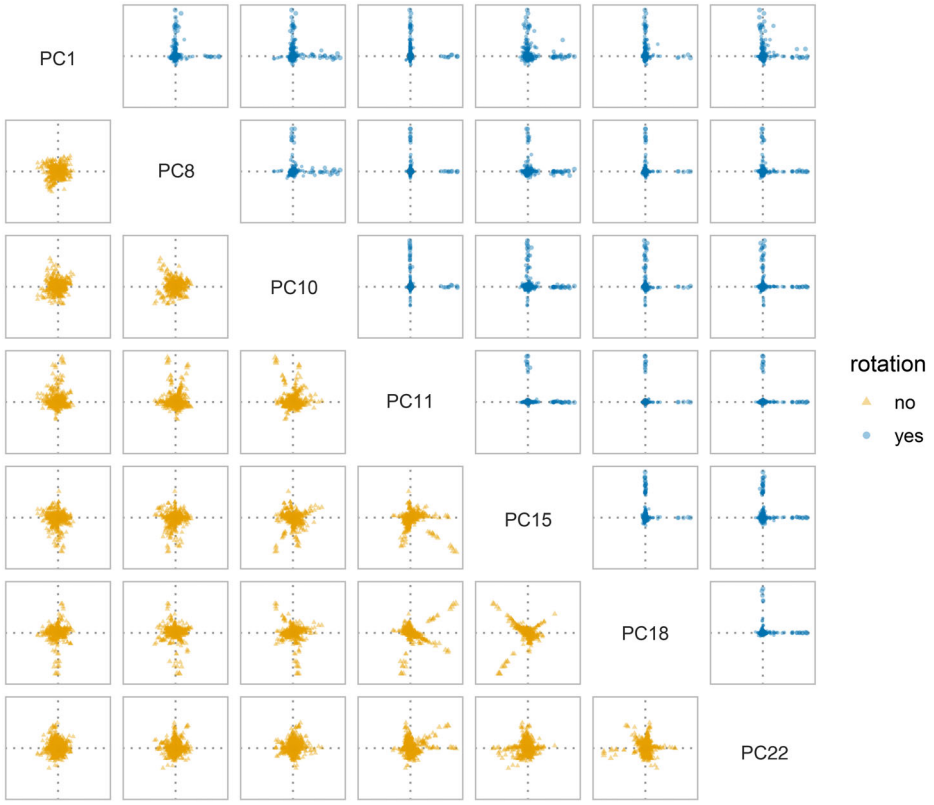

Figure 2. Loadings of seven principal components (PCs) from a large scale social network matrix. Each (off-diagonal) panel shows the loadings of two PCs on the original variables (displayed as points). The lower-triangular panels (yellow) depict the PCs before a rotation. The upper-triangular panels (blue) display the PCs after an orthogonal rotation. The PCs before and after the rotation have no special or corresponding relationship. In each panel, two perpendicular dotted lines (gray) indicate the coordinate axes. See Section 5.3 for details about the data analyzed.

of underlying PCs, k, is known (see, e.g., Chen et al. (2021) for a separated work on estimating k from data using "cross-validated eigenvalues"). PCA finds k uncorrelated linear transformations of the original variables such that after the linear transformations, the most variance is kept. That is,

$$\underset{Y}{\text{maximize}} \quad \|XY\|_{\text{F}} \quad \text{subject to} \quad Y \in \mathcal{V}(p,k), \tag{1}$$

where the feasible set is the Stiefel manifold, $\mathcal{V}(p,k)$. The jth PC is the linear combination of original variables whose coefficients are in the jth columns of Y. The coefficients are often called *loadings* (or loading coefficients). Note that loadings are usually nonzero (i.e., Y is usually not sparse). The transformed data $S = XY \in \mathbb{R}^{n\times k}$ contains the *scores*. That is, S_{ij} is the score of the ith sample on the jth PC.

In PCA, PCs are often defined sequentially. That is, in order to find the kth PCs, we fix the previous $k-1$ PCs and solve (1); repeat this for $k = 1, 2, \ldots$ in order. Such definition ensures the first k PCs together always explain the most variance in the data. By contrast, for sparse PCA, we reason in the following that it is sufficient to solve the optimization problem for all PCs at once. Note first that the solution to (1) is a subspace, because if Y^* is an optimizer of (1), then for any orthogonal matrix $R \in \mathcal{U}(k)$, Y^*R is also an optimizer. The solution to (1) being a rotation-invariant subspace is desirable because it allows a sparsity-enabling orthogonal rotation to any given solution. Importantly, such rotation exists under the assumption of *column sparsity* (see Section 2.1.1 and Vu and Lei 2013). We thereby propose a new method for sparse PCA.

2.1. Sparse Component Analysis

For sparse PCA, we impose an ℓ_1-norm constraint[1] on the loadings and formulate the following minimization of matrix reconstruction error (MRE)[2]:

$$\begin{aligned} &\underset{Z,B,Y}{\text{minimize}} \quad \left\|X - ZBY^{\text{T}}\right\|_{\text{F}} \\ &\text{subject to} \quad Z \in \mathcal{V}(n,k),\ Y \in \mathcal{V}(p,k),\ \|Y\|_1 \le \gamma, \end{aligned} \tag{2}$$

where $\gamma > 0$ is the sparsity controlling parameter, and the columns of Y are PC loadings. ZBY^{T} is an approximation of X.

The fundamental difference between formulation (2) and previous sparse PCA formulations is that the middle B matrix is not necessarily diagonal. Compared to the diagonal B case, this added flexibility has two merits—(i) it allows PCs to be column sparse and (ii) it allows sparse PCs to explain more variance in the data.

2.1.1. SCA Presumes Column Sparsity

Our formulation (2) presumes the PCs are column sparse. That is, given the subspace of ordinary PCs, there exists an orthogonal rotation, such that after the rotation, the PCs are approximately sparse.

Let UDV^{T} be the low-rank singular value decomposition (SVD) of X, where $U \in \mathcal{V}(n,k)$ and $V \in \mathcal{V}(p,k)$ contain singular

[1] The ℓ_1-norm constraint could be replaced by other sparsity constraints, for example, the ℓ_0-norm analogue.

[2] MRE depicts the unexplained variation in the data, akin to the sum of squares error in regression.

vectors, and $D \in \mathbb{R}^{k\times k}$ is a diagonal matrix with the diagonal entries in decreasing order, and $k \leq \min\{n, p\}$ is the rank. For any two orthogonal matrices $O, R \in \mathcal{U}(k)$, define $Z = UO$, $B = O^{\mathrm{T}}DR$, and $Y = VR$. With these definitions,

$$X \approx UDV^{\mathrm{T}} = (UO)(O^{\mathrm{T}}DR)(VR)^{\mathrm{T}} = ZBY^{\mathrm{T}}.$$

As such, ZBY^{T} approximates X as well as UDV^{T}. In particular, the middle B matrix is not diagonal because it absorbs the orthogonal matrices (O and R). Z and Y are orthogonally rotated from U and V, and both matrices still have orthogonal columns. Hence, by imposing an ℓ_1-norm constraint on Y to make it approximately sparse, we presume that there exists at least one orthogonal basis for the column space of V (i.e., the eigenvectors' subspace), which is not necessarily the original coordinate basis, such that the PCs are sparse under that basis.

Remark 1. The formulation of SCA does not explicitly defines an ordering for sparse PCs. This is because permuting the columns of Y, which can be absorbed by the orthogonal matrix R, does not change the approximation of ZBY^{T}. As such, the solution to (2) is not unique. In practice (see Section 4.1), we sort sparse PCs by the explained variance (EV) of individual PCs, which is defined as $\|Xy\|_2^2$, where $y \in \mathbb{R}^p$ contains the loadings of a PC. As such, the first sparse PC explains the most variation in the data, and the second PC the second most, etc.

2.1.2. Column Sparsity versus Row Sparsity

Column sparsity does not assume the loadings of ordinary PCs (i.e., singular vectors of X) to be already approximately sparse; they only need to be so after some orthogonal rotations. By contrast (or more strictly), row sparse PCA presumes that the loadings of ordinary PCs are by themselves approximately sparse (i.e., the singular vectors align closely with the natural coordinate axes already).

In SCA, the nondiagonal middle B matrix facilitates the more general formulation of column sparse PCA. Specially, if B is restricted to diagonal, the formulation reduces to row sparse PCA.[3] The next proposition compares column and row sparse PCA in terms of MRE (the proof is simple and provided in Appendix A for completeness).

Proposition 1 (Comparison of row and column sparsity). Let $X \in \mathbb{R}^{n\times p}$ be any matrix. Suppose $S_Z \subseteq \mathbb{R}^{n\times k}$ and $S_Y \subseteq \mathbb{R}^{p\times k}$ are the feasible sets for Z and Y, respectively, where $k \leq \min(n, p)$. Then, subject to $Z \in S_Z$, $Y \in S_Y$, and D is diagonal, it holds that

$$\min_{Z,B,Y} \left\|X - ZBY^{\mathrm{T}}\right\|_{\mathrm{F}} \leq \min_{Z,D,Y} \left\|X - ZDY^{\mathrm{T}}\right\|_{\mathrm{F}}.$$

In particular, the inequality is strict if S_Z and S_Y are defined in (2).

Recall that MRE reflects the unexplained variance in the data. Under the same constraints in (2), the left-hand side of the inequality corresponds to the MRE objective of column sparse PCA, and the right-hand-side row sparse one. Proposition 1 says that the solution to column sparse PCA has an optimal MRE strictly less than that of row sparse PCA. In other words, column sparse PCA can capture more variance in the data than row sparse PCA.

Remark 2. From a parametric perspective, SCA explains more variance because it uses $k^2 - k$ more parameters in the B matrix. Relative to the total number of parameters, this is typically a small increase; the Z and Y matrices contain roughly $(n + p)k$ parameters, and typically k is much smaller than $n+p$. Whether these additional parameters in B are statistically justified must be addressed in a case-by-case basis. In our limited experience with these techniques, the additional parameters are easily justified because the proportion of variance explained dramatically increases (see Section 4.1); the output becomes more stable against initializations, perturbations, and tuning parameters (see Section 4.2); and the estimated factors are easily interpretable (see Sections 5.2 and 5.3).

2.2. An Algorithm for SCA

To solve SCA, the following lemma translates (2) into an equivalent and more convenient form (the proof can be found in Appendix A).

Lemma 1 (Bilinear form of SCA). Solving the minimization in (2) is equivalent to solving the following maximization problem,

$$\begin{aligned} \underset{Z,Y}{\text{maximize}} \ \left\|Z^{\mathrm{T}}XY\right\|_{\mathrm{F}} \ \text{subject to} \ Z \in \mathcal{V}(n,k), \\ Y \in \mathcal{V}(p,k), \ \|Y\|_1 \leq \gamma. \end{aligned} \tag{3}$$

In particular, for the optimizer in (2), $B = Z^{\mathrm{T}}XY$.

Due to the non-convexity of ℓ_2-equality constraints ($Z \in \mathcal{V}(n,k)$ and $Y \in \mathcal{V}(p,k)$), the feasible set in (3) is not convex in general. We replace the feasible set with its convex hull using some ℓ_2-inequality constraints for simplicity,

$$\begin{aligned} \underset{Z,Y}{\text{maximize}} \ \left\|Z^{\mathrm{T}}XY\right\|_{\mathrm{F}} \ \text{subject to} \ Z \in \mathcal{B}(n,k), \\ Y \in \mathcal{B}(p,k), \ \|Y\|_1 \leq \gamma. \end{aligned} \tag{4}$$

Due to the Karush-Kuhn-Tucker conditions (see, e.g., Nocedal and Wright 2006), one could expect the solution to fall on the boundary (i.e., $Z \in \mathcal{V}(n,k)$, $Y \in \mathcal{V}(p,k)$, and $\|Y\|_1 = \gamma$) so long as the sparsity parameters are chosen such that $k \leq \gamma \leq k\sqrt{p}$[4].

Algorithm 2 describes an iterative algorithm that computes sparse PCs as formulated in (4). The input includes a data matrix X, the desired number of sparse PCs k, and optionally the sparsity controlling parameters γ. The algorithm outputs the loadings of k sparse PCs. In our experiences, a default value of $\gamma = \sqrt{pk}$ appears to generate robust and interpretable sparse PCs (see, e.g., Section 4.2). We discuss a data-driven method of tuning the sparsity parameters in supplementary section S1. In general, Algorithm 2 does not necessarily converge to a global optimum for (4); however, our empirical studies indicate that

[3] This restricted formulation is essentially a low-rank SVD with an additional sparsity constraint on the right singular vectors.

[4] This is for the set $\{Y \in \mathbb{R}^{p\times k} \mid \|Y\|_1 = \gamma\}$ to intersect with the Stiefel manifold $\mathcal{V}(p,k)$.

the algorithm does converge to interpretable factors for appropriate choices of the sparsity parameters. Note that each iteration results in a decrease in the objective.

The SCA algorithm initializes $Z \in \mathcal{V}(n,k)$ and $Y \in \mathcal{V}(p,k)$ with the top k left and right singular vectors of X respectively. Once initialized, the algorithm alternatively updates Z and Y; fixing one and optimizing the other until convergence. The iteration is because the objective function is bilinear in Z and Y, allowing for fast updates. Specifically, with Y fixed, (4) takes the form

$$\underset{Z}{\text{maximize}} \quad \left\|Z^{\mathrm{T}}XY\right\|_{\mathrm{F}} \quad \text{subject to} \quad Z \in \mathcal{B}(n,k). \tag{5}$$

With Z fixed, (4) takes the form

$$\underset{Y}{\text{maximize}} \quad \left\|Z^{\mathrm{T}}XY\right\|_{\mathrm{F}} \quad \text{subject to} \quad Y \in \mathcal{B}(p,k),\ \|Y\|_1 \le \gamma. \tag{6}$$

Input: $A \in \mathbb{R}^{p\times k}$,
sparsity parameter γ (optional, default to $\sqrt{pk}$)
Procedure PRS(A)**:**
$\tilde{Y} \leftarrow$ left singular vectors of A
$Y^* \leftarrow$ rotate $\tilde{Y}$ with varimax // Section 2.2.3
$\hat{Y} \leftarrow$ soft-threshold Y^* with parameter γ
Output: $\hat{Y}$

Algorithm 1: Polar-Rotate-Shrink (PRS)

Input: Data matrix X and a number of components k
Procedure SCA (X, k)**:**
Initialize $\hat{Z}$ and $\hat{Y}$ with the top k left and right singular vectors of X
repeat
$\hat{Y} \leftarrow$ PRS($X^{\mathrm{T}}\hat{Z}$) // Algorithm 1
$\hat{Z} \leftarrow$ polar($X\hat{Y}$) // Lemma 2
until *convergence*
Output: Sparse loadings $\hat{Y}$

Algorithm 2: Sparse Component Analysis (SCA)

2.2.1. Update Z fixing Y

The update of Z fixing Y in (5) is algebraic. The following lemma provides a set of solutions to (5), which is extended from Theorem 7.3.2 in Horn and Johnson (1985) (the proof is included in Appendix A for completeness).

Lemma 2 (Maximization without sparsity constraint). Given a full-rank matrix $X \in \mathbb{R}^{n\times p}$, with $p \le n$, let the singular values of X be σ_i for $i = 1, 2, \ldots, p$. Then,

$$\max_{Y\in\mathcal{V}(n,p)} \left\|X^{\mathrm{T}}Y\right\|_{\mathrm{F}} = \sum_{i=1}^{p}\sigma_i$$

with the maximizer $Y^* = \text{polar}(X)$, up to any orthogonal rotation from the right. Here, $\text{polar}(X) = X(X^{\mathrm{T}}X)^{-1/2}$ is the *polar* of X.

Due to Lemma 2, the SCA algorithm updates Z with the polar of XY, $\hat{Z} = \text{polar}(XY)$, which can be computed in $\mathcal{O}(nk)$ time (Journée et al. 2010).

2.2.2. Update Y Fixing Z

To update Y fixing Z, we start by solving the nonsparse version of (6) (i.e., remove the sparsity constraint $\|Y\|_1 \le \gamma$),

$$\underset{Y}{\text{maximize}} \quad \left\|Z^{\mathrm{T}}XY\right\|_{\mathrm{F}} \quad \text{subject to} \quad Y \in \mathcal{B}(p,k). \tag{7}$$

Let $\tilde{Y} = \text{polar}(X^{\mathrm{T}}Z)$. Then, $\tilde{Y}$ is one element in the subspace of the solutions to (7). Before imposing the sparsity constraint, we look for an orthogonal rotation R to $\tilde{Y}$ to minimize $\|\tilde{Y}R\|_1$. However, $\|Y\|_1$ is not a smooth function of Y if it contains at least one zero entry, entailing the complications of defining sub-gradients. Alternatively, the SCA algorithm minimizes a smoother criterion based on the $\ell_{4/3}$ norm:

$$\underset{R}{\text{minimize}} \quad \left\|\tilde{Y}R\right\|_{\frac{4}{3}} \quad \text{subject to} \quad R \in \mathcal{U}(k). \tag{8}$$

This sub-problem leads to the varimax rotation (see Section 2.2.3) that is widely applied in factor analysis (Kaiser 1958). We denote $Y^* = \tilde{Y}R^*$ to be the orthogonally rotated solution to (7), where R^* is the solution to (8). Finally, considering the ℓ_1-norm sparsity constraint, we apply the element-wise soft-thresholding of Y^* with the sparsity parameter γ, which is defined as (Donoho 1995; Tibshirani 1996)

$$\left[T_\gamma(Y^*)\right]_{ij} = \text{sign}(Y^*_{ij}) \cdot \left(|Y^*_{ij}| - t\right)_+, \tag{9}$$

where $t > 0$ is the threshold determined by the equation $\left\|T_\gamma(Y^*)\right\|_1 = \gamma$, and x_+ equals x if $x > 0$ or 0 otherwise. We discuss several properties of soft-thresholding in Supplementary Section S2. In summary, the update of Y given Z consists of three steps that we call "Polar-Rotate-Shrink" (PRS, Algorithm 1)—first, compute a solution to the unconstrained problem (7); second, rotate with varimax; third, soft-threshold all of the elements.[5]

2.2.3. Orthogonal Rotations: Varimax and Quartimax

For any matrix $A \in \mathbb{R}^{p\times k}$, the *varimax criterion* is defined as the sum of column (sample) variance of squared elements (A_{ij}^2) (Kaiser 1958):

$$C_{\text{varimax}}(A) = \sum_{j=1}^{k}\left[\frac{1}{p}\sum_{i=1}^{p}A_{ij}^4 - \frac{1}{p^2}\left(\sum_{i=1}^{p}A_{ij}^2\right)^2\right].$$

For a fixed matrix $Y \in \mathbb{R}^{p\times k}$, the *varimax rotation* seeks an orthogonal rotation $R \in \mathbb{R}^{k\times k}$ to maximize the varimax criterion evaluated at YR,

$$\underset{R}{\text{maximize}} \quad C_{\text{varimax}}(YR) \quad \text{subject to} \quad R \in \mathcal{U}(k). \tag{10}$$

It is commonly used in factor analysis for producing nearly sparse and interpretable loadings of PCs, especially in the psychology literature. The varimax rotation is easy to compute; for example, the base function `varimax` in R implements a gradient projection algorithm of it (Bernaards and Jennrich 2005).

[5] More investigation is needed in order to understand the statistical properties of PRS. For example, in a recent paper (Rohe and Zeng 2020), we showed that PCA with the varimax rotation is a consistent estimator for a broad class of modern factor models, that includes the degree corrected stochastic block model (Karrer and Newman 2011).

Jennrich (2001) showed that the gradient projection algorithm converges to a local optimum from any starting point and enjoys geometric (or linear) convergence rate.

The varimax criterion naturally links to the $\ell_{4/3}$-norm objective function in (8). Since $Y \in \mathcal{V}(p,k)$, the columns of Y have unit length. Hence, $\sum_{i=1}^{p} Y_{ij}^2 = 1$, and the varimax criterion reduces to a simpler form (also known as the *quartimax* criterion as introduced by Carroll (1953)) up to an additive constant:

$$C_{\text{quartimax}}(Y) = \sum_{i=1}^{p}\sum_{j=1}^{k} Y_{ij}^4 = \|Y\|_4^4,$$

which is the ℓ_4-norm of Y to the power of 4. Next, by the Hölder's inequality (using the Hölder conjugates 4/3 and 4) and the power mean inequality (and that $\|Y\|_{\text{F}} = \sqrt{k}$), $\|Y\|_{\frac{4}{3}}\|Y\|_4 \geq \|Y\|_1 \geq \|Y\|_F = \sqrt{k}$. This implies that maximizing the varimax criterion is the dual problem of minimizing the $\ell_{4/3}$-norm objective. Hence, to update Y in the algorithm of SCA, we invoke the varimax rotation in (10) as a proxy of (8).

Remark 3. Besides varimax, we experimented the orthogonal rotation that directly minimizes the ℓ_1 norm, which we call the "*absmin*" rotation:

$$\underset{R}{\text{minimize}} \quad \|YR\|_1 \quad \text{subject to} \quad R \in \mathcal{U}(k). \tag{11}$$

However, the objective function is not smooth at those R where YR contains at least one zero element; this posts challenges to solving (11). For example, we tried a gradient projection algorithm using the gradient direction $Y^{\text{T}}\,\text{sign}(YR)$, where $\text{sign}(\cdot)$ is the element-wise sign function, yet the algorithm hardly converges. It is worth noting that in our limited experiments, where we used the absmin rotation but only allowed 15 iterations of this gradient projection algorithm, we obtained marginally better solutions, in terms of explained variance, than using the varimax rotation (see Section 4.1). It is of future interest to investigate alternative orthogonal rotations that are easy to compute and can generate approximately sparse structure.

2.3. Sparse Matrix Approximation

In the SCA algorithm above, a sparsity constraint can also be applied to Z, in addition to Y. We call this sparse matrix approximation (SMA). We define SMA as the solution to a matrix reconstruction error minimization problem:

$$\begin{aligned} \underset{Z,B,Y}{\text{minimize}} \quad & \left\|X - ZBY^{\text{T}}\right\|_{\text{F}} \\ \text{subject to} \quad & Z \in \mathcal{B}(n,k),\ \mathcal{P}_1(Z) \leq \gamma_z, \\ & Y \in \mathcal{B}(p,k),\ \mathcal{P}_2(Y) \leq \gamma_y, \end{aligned} \tag{12}$$

where $\gamma_z > 0$ and $\gamma_y > 0$ are the sparsity controlling parameters, and $\mathcal{P}_1$ and $\mathcal{P}_2$ are some *penalty* functions that promote sparsity. If γ_Z is so large that $\mathcal{P}_1(Z) \leq \gamma_z$ is always satisfied, then (12) is equivalent to SCA. Similar to Lemma 1, we transform (12) into an equivalent and more convenient form (the proof is almost identical to that of Lemma 1 thus is omitted),

$$\begin{aligned} \underset{Z,Y}{\text{maximize}} \quad & \left\|Z^{\text{T}}XY\right\|_{\text{F}} \\ \text{subject to} \quad & Z \in \mathcal{B}(n,k),\ \mathcal{P}_1(Z) \leq \gamma_z, \\ & Y \in \mathcal{B}(p,k),\ \mathcal{P}_2(Y) \leq \gamma_y. \end{aligned} \tag{13}$$

The two criteria in (12) and (13) are equivalent if and only if $B = Z^{\text{T}}XY$. We interpret B as the "score" of SMA, since the solution to (12) maximizes the sum of squares of its elements, $\sum_{i,j} B_{ij}^2$. It is also worth noting that the squared matrix reconstruction error equals to $\|X\|_{\text{F}}^2 - \|B\|_{\text{F}}^2$ (see the proof of Lemma 1).

Since SMA is a simple extension from SCA, we extend Algorithm 2 for SMA in Algorithm 3, where we apply PRS to Z in addition to Y. The output includes the estimated Z, B, and Y.

Input: data matrix $X \in \mathbb{R}^{n\times p}$ and the approximation rank k
Procedure SMA (X, k)**:**
  Initialize $\hat{Z}$ and $\hat{Y}$ with the top k left and right singular vectors of X
  repeat
    $\hat{Z} \leftarrow \texttt{PRS}(X\hat{Y})$ `// Algorithm 1`
    $\hat{Y} \leftarrow \texttt{PRS}(X^{\text{T}}\hat{Z})$ `// Algorithm 1`
  until *convergence*
  $\hat{B} \leftarrow \hat{Z}^{\text{T}}X\hat{Y}$
Output: $\hat{Z}$, $\hat{B}$, and $\hat{Y}$

Algorithm 3: Sparse Matrix Approximation (SMA) with $\mathcal{P}_1(A) = \mathcal{P}_2(A) = \|A\|_1$.

We highlight that SMA generalizes the popular penalized matrix decomposition (PMD) proposed by Witten, Tibshirani, and Hastie (2009), which is also similar to the method of Shen and Huang (2008). The PMD also approximates a data matrix $X \in \mathbb{R}^{n\times p}$ by the product of three matrices, ZDY^{T}, where $Z \in \mathcal{V}(n,k)$ and $Y \in \mathcal{V}(p,k)$ are presumed sparse, and $D \in \mathbb{R}^{k\times k}$ is a diagonal matrix whose diagonal entries are in decreasing order, and k is the rank of the matrix approximation. For sparsity, PMD applies penalty functions to Z and Y, leading to the matrix reconstruction error minimization formulation of PMD:[6]

$$\begin{aligned} \underset{U,D,V}{\text{minimize}} \quad & \left\|X - ZDY^{\text{T}}\right\|_{\text{F}} \\ \text{subject to} \quad & Z \in \mathcal{B}(n,k),\ \mathcal{P}_1(Z) \leq \gamma_z, \\ & Y \in \mathcal{B}(p,k),\ \mathcal{P}_2(Y) \leq \gamma_y, \\ & D \text{ is diagonal}, \end{aligned}$$

where $\gamma_z, \gamma_y > 0$ are parameters that control the sparsity of Z and Y, and $\mathcal{P}_1$ and $\mathcal{P}_2$ are some convex penalty function (e.g. ℓ_1-norm).

The single difference between SMA and PMD is the diagonal constraint on the middle matrix. In this way, SMA generalizes PMD, because, SMA estimates $k^2 - k$ more parameters in B than PMD (see Remark 2). Proposition 1 suggests that the reconstruction error of SMA is less or equal to that of PMD (see also Remark 4 in Appendix A). Algorithmically, in order to compute PMD, Witten, Tibshirani, and Hastie (2009) proposed to find the solution by sequentially maximizing B_{ii} for $i = 1, 2, \ldots, k$ (recall that $B = Z^{\text{T}}XY$). By contrast, solving the SMA in (13) amounts to maximizing the entirety of the score matrix, that is, $\|B\|_{\text{F}}$.

[6]The article originally considers the PMD with $k = 1$. The PMD finds multiple factors sequentially using a deflation technique.

3. Connections to Existing Methods

As mentioned in Section 1.1, SCA is related to factor analysis in that they both use a rotation. One key difference is that the sparsity constraint in SCA creates actual zeros. In this section, we compare SCA with several existing methods of sparse PCA. Then, we introduce two existing data processing techniques that are related to SCA.

3.1. Existing Sparse PCA Methods

The formulation of SCA is akin to multiple existing sparse PCA formulations. However, the possibility of orthogonal rotations has not been explored thoroughly, despite the plethora of available methods. In this section, we elucidate these connections and point to some differences.

SPCA (Zou, Hastie, and Tibshirani 2006) SPCA is motivated to maximize the explained variance in the data (Jolliffe, Trendafilov, and Uddin 2003). The formulation of SPCA minimizes a "residual sum of squares plus penalties" type of criterion,

$$\underset{U,V}{\text{minimize}} \quad \left\|X - XVU^{\mathrm{T}}\right\|_{\mathrm{F}}^2 + \lambda_1 \|V\|_{\mathrm{F}}^2 + \textstyle\sum_{j=1}^{k} \lambda_{2,j} \left\|v_j\right\|_1 \\ \text{subject to} \quad U \in \mathcal{V}(p,k),$$

where v_j is the jth column of $V \in \mathbb{R}^{p\times k}$ containing the sparse loadings of the jth PC, and λ_1 and $\lambda_{2,j}$ are tuning parameters. In this formulation, the first and the third terms are not invariant to orthogonal rotations (on V). Specially, the first term $\left\|X - XVU^{\mathrm{T}}\right\|_{\mathrm{F}}^2$ is minimized when V corresponds to the k ordinary PCs. Based on this, Zou, Hastie, and Tibshirani (2006) shows that the algorithm of SPCA searches for a sparse approximation of the ordinary PCs, yet without sparsity-enabling orthogonal rotations (i.e., it assumes row sparsity).

SPC (Witten, Tibshirani, and Hastie 2009) SPC finds one sparse PC at a time,

$$\underset{u,v}{\text{maximize}} \ u_i^{\mathrm{T}} X v_i \ \text{ subject to } \ \|u_i\|_2 = 1, \\ \|v_i\|_2 = 1, \ \|v_i\|_1 \le \gamma, \tag{14}$$

where $v_i \in \mathbb{R}^p$ contains the loadings of the ith sparse PC, for $1 \le i \le k$. When $k = 1$, our formulation of SCA in (3) takes the same form as the SPC formulation, where an orthogonal rotation is unnecessary. When $k > 1$, however, SPC searches for sparse PCs sequentially and does not rotate PCs, unlike SCA, which computes k sparse PCs simultaneously. SPC is similar to the rSVD proposed by Shen and Huang (2008) and the TPower proposed by Yuan and Zhang (2013) in that all the three methods rely on a deflation technique for multiple PCs. This technique entails complications of, for example, non-orthogonality and sub-optimality (Mackey 2008). More generally, these methods can each be viewed as a special case of the following GPower formulation.

GPower (Journée et al. 2010) GPower has a "block version" that computes multiple sparse PCs simultaneously by considering a linear combination of individual sparse PCA (as formulated in SPC),

$$\underset{U,V}{\text{maximize}} \ \sum_{j=1}^{k} \mu_j u_j^{\mathrm{T}} X v_j - \sum_j \lambda_j \left\|v_j\right\|_1 \\ \text{subject to} \ \ U \in \mathcal{B}(n,k), \ V \in \mathcal{V}(p,k),$$

where V contains the PC loadings, and u_j and v_j are the jth column of U and V respectively, and μ_j is the weight for the jth sparse PC, and λ_j is the sparsity tuning parameter for the jth sparse PC. The algorithm of GPower fundamentally deals with sparse PCs individually, which prohibits orthogonal rotations (on V).

SPCArt (Hu et al. 2016) SPCArt is the first (to our knowledge) sparse PCA method that concerns orthogonal rotations in its formulation. It searches for sparse PCs by directly approximating the singular vectors (as opposed to minimizing the reconstruction error or maximizing the explained variance),

$$\underset{Y,R}{\text{minimize}} \ \|V - YR\|_{\mathrm{F}}^2 + \lambda \|Y\|_1 \\ \text{subject to} \ \ Y \in \mathcal{V}(p,k), \ R \in \mathcal{U}(k),$$

where $V \in \mathcal{V}(p,k)$ contains the top k singular vectors of X, and Y contains the sparse loadings. Conceptually, introducing an orthogonal rotation (R) allows a larger searching space for Y. However, the algorithm of SPCArt does not specifically update R to promote sparsity (e.g., minimize $\|Y\|_1$ as in SCA); instead, SPCArt simply computes R so as to align the polar of V and Y (i.e., $\hat{R} = \text{polar}(Y^{\mathrm{T}}V)$). As such, the performance of SPCArt could be sensitive to the initialization of Y. Empirically, SPCArt yields results that are nearly comparable to the GPower based method, as concluded by the authors.

3.2. Sparse Coding and Independent Component Analysis

Sparse coding concerns low-rank representations of individual samples. We view it as a variant of PCA, where we presume the component scores to be sparse. Recall that the scores are the representations of individual data points in $\mathbb{R}^k$, where k is the number of PCs. In particular, presuming sparse scores implies that each data point is correlated with only a small subset of PCs. Sparse coding is useful to generate simple representations of individual date points, and the basis of such representations (i.e., PCs) usually provide scientific insights. For example, sparse coding of natural images recovers the common understanding of how the primary visual cortex in mammalian perceives scenes (see, e.g., Section 5.1).

The SCA algorithm can be used to solve sparse coding. This is because, similar to SCA, sparse coding can be viewed as a special case of the SMA problem. To see this, simply omit the sparsity constraint on Y in (12),

$$\underset{Z,B,Y}{\text{minimize}} \quad \left\|X - ZBY^{\mathrm{T}}\right\|_{\mathrm{F}} \\ \text{subject to} \quad Z \in \mathcal{B}(n,k), \ Y \in \mathcal{B}(p,k), \ \mathcal{P}_1(Z) \le \gamma_z$$

Here, Z contains the sparse scores, and BY^{T} contains the basis of sparse coding. To solve sparse coding, we apply the SCA algorithm (Algorithm 2) to the transposed data matrix, X^{T}. In doing this, the output of the algorithm is actually an estimate of sparse component scores for the original data matrix.

More broadly, independent component analysis (ICA) is widely applied for sparse coding in the signal processing literature. Despite the different motivations, sparse PCA on a transposed data matrix appears to perform very similarly to sparse ICA on the original data. We elaborate on this in Supplementary Section S3 and apply SCA to blind source separation of images.

4. Simulation Studies

In this section, we compare several sparse PCA methods using simulated data. Specifically, we focused on (a) their ability of explaining variance in the data, (b) the robustness against varying sparsity parameters, and (c) the computational speed. We selected SPCA, SPC, GPower, the SPCAvRP method recently proposed by Gataric, Wang, and Samworth (2020), SCA, and another variant of SCA which deploys the absmin rotation (SCA-absmin, see Remark 3 of Section 2.2.3). For SCA and SCA-absmin, we implemented the algorithms in R.[7] For SPCA, SPC, and SPCAvRP, we invoked the original R packages `elasticnet`, `PMA`, and `SPCAvRP`, respectively. The implementation of GPower (in `MATLAB`) was obtained from the authors' website. For all the iterative methods, we specified maximum number of iterations to 1000 and the stopping (convergence) criterion to 10^{-5}. Overall, our numerical experiments showed that the SCA algorithm converges faster and produces more robust sparse PCs that capture a larger amount of variance in the data.

4.1. Proportion of Variance Explained

In this simulation, we compared the abilities of sparse PCA methods in explaining variance in the data. To this end, we simulated 30 data matrices with $n = 100$ observations and $p = 100$ variables from the following low-rank generative model:

$$X = SY^{\mathrm{T}} + E,$$

where $S \in \mathbb{R}^{100\times 16}$ contains the component scores, and $Y \in \mathbb{R}^{100\times 16}$ contains the loadings of sparse PCs, and $E \in \mathbb{R}^{100\times 100}$ is some noise. To generate S, we randomly sampled $U \in \mathcal{V}(100, 16)$ and $V \in \mathcal{U}(16)$ and set $S = U\Sigma V^{\mathrm{T}}$, where Σ is a diagonal matrix with the diagonals $\sigma_l = 10 - \sqrt{l}$ for $l = 1, 2, \ldots, 16$. To simulate a sparse Y, we took a random element from $\mathcal{V}(100, 16)$, then soft-threshold its elements with sparsity parameter $\gamma = 20$ (i.e., T_{20} as defined in (9)).[8] Note that, it is unnecessary to re-scale the columns of loadings to unit length, because the column of S can absorb these scalars. Lastly, the elements in E were drawn independently from the normal distribution, $E_{ij} \sim \mathrm{N}(0, 0.1^2)$.

We applied the six sparse PCA methods to each simulated data matrix X with $k = 2, 4, 6, \ldots, 16$. For each k, we imposed the same ℓ_1-norm constraint on the sparse loadings for all methods. Specifically, for SCA, and SPC, we directly configured the sparsity controlling parameters to $2.5k$.[9] As for SPCA, GPower and SPCAvRP, to ensure a fair comparison, we tuned the parameters using binary search such that the returned loadings all have the same ℓ_1 norm of $2.5k$. To evaluate sparse PCs, we define the cumulative proportion of variance explained (PVE) by the first k sparse PCs as $\|X_Y\|_{\mathrm{F}}^2$, where $X_Y = XY(Y^{\mathrm{T}}Y)^{-1}Y^{\mathrm{T}}$ (Shen and Huang 2008). Note that the PVE by sparse PCs is upper bounded by that of ordinary PCs (no sparsity constraint). Therefore, we also applied PCA to X for comparison. Figure 3 displays the mean PVE for different PCA methods, varying the requested number of PCs from 2 to 16. It can be seen that SPCAvRP and SPCA explained less than half of the PVE by PCA, and that GPower and SPC both exhibited some improvements over SPCA. For GPower, we tested both the single-unit and the block versions, but the block version often converged to a defective solution with some columns decaying to all zeros. This happened when the number of targeted PCs went above five in this simulation. As such, we display only the single-unit version of the results. Overall, SCA performed the best among sparse PCA methods and were the closest to PCA. In addition, the SCA algorithm converged with fewer iterations than the other sparse PCA methods (see Table 1 for a comparison when $k = 16$). We also observed that using the varimax rotation (SCA), the algorithm was more computationally efficient than using the absmin rotation (SCA-absmin).

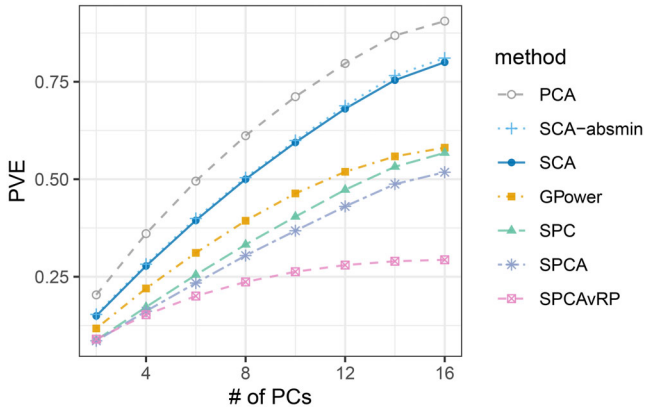

Figure 3. Comparisons of sparse PCA methods using simulated data. The proportion of variance explained (PVE) by sparse principal components (PCs) with the number of targeted PCs varying from 2 to 16.

Table 1. Comparison of the computational efficiency of sparse PCA methods.

Method	# of iterations	Mean run time (s)	Environment
SCA	10 ∼ 65 (all PCs)	0.96	R
SPC	25 ∼ 1000 (each PC)	1.21	R
GPower	30 ∼ 150 (each PC)	0.19	MATLAB
SPCA	470 ∼ 920 (all PCs)	56.30	R
SPCAvRP	/	28.67	R
SCA-absmin	/	23.5	R

NOTE: Each method is tasked to find 16 PCs on a single CPU (2.50GHz). SPCAvRPs is not iterative (yet is parallelizable), hence, the number of iterations is not applicable. The absmin rotation is less efficient, so we halted the algorithm of SCA-absmin after the 15th iteration.

[7] We provide an R package `epca`, for **e**xploratory **p**rincipal **c**omponent **a**nalysis, which implements SCA and SMA with various algorithmic options. The package is available from CRAN (*https://CRAN.R-project.org/package=epca*).

[8] We also experimented with $\gamma = \sqrt{pk} = 40$. The results are comparable.

[9] The coefficient 2.5 is calculated from $\lambda/16$, assuming that the 16 sparse PCs have equally distributed ℓ_1-norm.

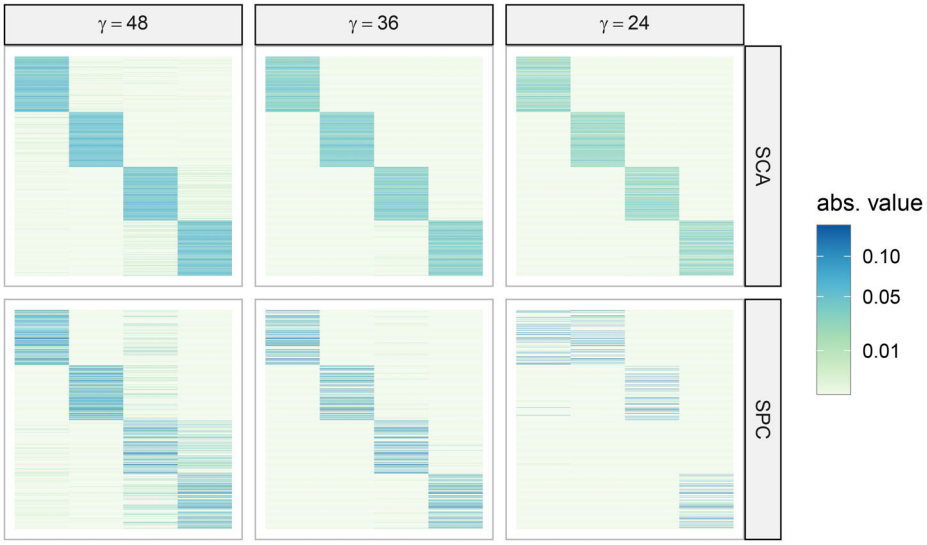

Figure 4. Comparisons of SCA and SPC using simulated network data. Heat maps of the loadings (900 × 4 matrices) returned by SCA and SPC using three different sparsity parameters ($\gamma = 24, 36, 48$). In each heat map, rows correspond to nodes, which are grouped by the true community membership, and each column corresponds to one sparse PC. The color shade indicates the absolute of loadings.

4.2. Robustness Against Tuning Parameters

This simulation study investigates the robustness of sparse PCA to the choice of sparsity parameters. For this, we applied sparse PCA to detect communities in networks (or graph partitioning) (see, e.g., Fortunato 2010), using the graph adjacency matrix (see the definition below) as input. This application is possible thanks to the recent consistency results (Rohe and Zeng 2020) showing that under the stochastic block model (SBM, see for example Holland, Laskey, and Leinhardt 1983), the support of each sparse PC estimates the membership (indicator) of one community. Hence, we could evaluate sparse PCs by examining their support.

We simulated 30 undirected graphs with $n = 900$ nodes and four equally sized blocks from the SBM. Under the SBM, the edge between node i and j is sampled from the Bernoulli distribution, Bernoulli$(B_{z(i),z(j)})$, where $z(i) \in \{1, 2, 3, 4\}$ is the membership of node i, and

$$B = 0.05 \times \begin{bmatrix} 0.6 & 0.2 & 0.1 & 0.1 \\ 0.2 & 0.7 & 0.05 & 0.05 \\ 0.1 & 0.05 & 0.6 & 0.25 \\ 0.1 & 0.05 & 0.25 & 0.6 \end{bmatrix}$$

is the block connectivity matrix. Under this setting, the expected number of edges connected to each node is 45. For each simulated graph, we defined the adjacency matrix $A \in \{0, 1\}^{n \times n}$ with $A_{ij} = 1$ if and only if i and j are connected.

We applied SCA, SPC, and GPower[10] to each of the 30 simulated adjacency matrices with $k = 4$. We varied the sparsity parameter γ to take value in $\{18, 24, 36, 48, 60, 66\}$. For SPC, we required each of the four PCs to have ℓ_1 norm $\gamma/4$. As for GPower, we tuned its parameters such that the returned loading matrix has the ℓ_1 norm of γ. Figure 4 depicts the estimated loadings returned by SCA and SPC. On the left two columns of panels ($\gamma = 48$ and 36), the supports of the four sparse PCs were well separated and indicated block memberships. This suggested that we could use the loadings to cluster nodes and quantitatively assessed the quality of sparse PCA methods. Specifically, we assigned node i to cluster j if Y_{ij} is the largest absolute value in the ith row of Y, that is, $|Y_{ij}| > |Y_{il}|$ for all $l \neq j$. In the case of ties or all-zero rows, the cluster label is randomly assigned. For each estimate, let $C \in \{1, 2, 3, 4\}^n$ contain the assigned cluster labels and $C^* \in \{1, 2, 3, 4\}^n$ contain the true labels. Define the accuracy as

$$\text{Accuracy}(C, C^*) = \max_{\pi \in \mathcal{P}(4)} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(\pi\left(C_i\right) = C_i^*\right) \right\},$$

where $\mathcal{P}(4)$ contains all the possible permutation functions of the set $\{1, 2, 3, 4\}$, and $\mathbb{1}(x)$ is the indicator function of x. We used the accuracy to assess the quality of the sparse PCA solutions. Figure 5 depicts the accuracy of the three methods with varying sparsity parameters. It can be seen that the performance of GPower and SCA were less affected by the changing of sparsity

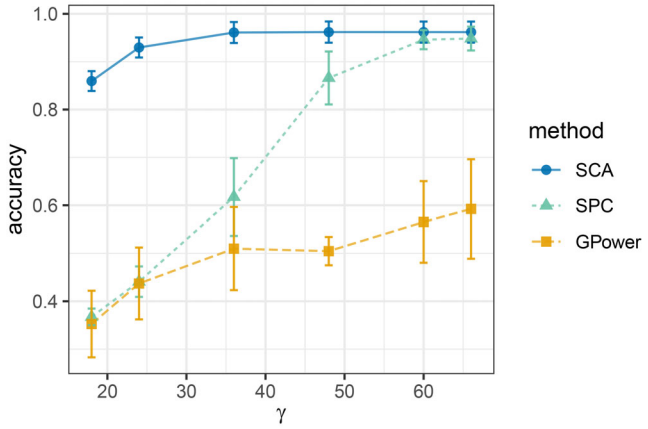

Figure 5. Comparisons of sparse PCA methods using simulated network data. The accuracy of SCA, GPower, and SPC in community detection using various sparsity parameters (γ). Each point indicates the mean accuracy across 30 replicates, and the error bar indicates the standard deviation of the evaluated accuracy.

[10]Since SPCA and SPCAvRP performs worse than SPC and GPower (Zou and Xue 2018), we excluded the two methods in this simulation for simplicity.

Table 2. Comparison of the SPC objective values, $\sum_{i=1}^{4}(u_i^{\mathrm{T}}Av_i)^2$ (see (14)), evaluated using the output of the SCA and SPC algorithms with various sparsity parameter (γ).

	$\gamma = 18$	$\gamma = 24$	$\gamma = 36$	$\gamma = 48$	$\gamma = 60$	$\gamma = 66$
Using SCA solution	191.47	323.36	**1135.03**	**1906.25**	**2554.86**	**2783.73**
Using SPC solution	**544.81**	**705.01**	1029.04	1195.91	1334.67	1423.95

Bold values represent the p-value < 0.001, based on the paired t-test of 30 replicates.

parameter, while SPC was profoundly influenced. As γ became smaller, SPC quickly lost its power in community detection, suggesting that SPC is more sensitive to the choices of tuning parameter. Although less sensitive to the change in γ, GPower produced poorer estimation of sparse PCs, with the accuracy slightly better than random guesses (accuracy = 0.25). Overall, SCA yielded higher accuracy with smaller deviation compared to the others, suggesting that SCA is less dependent on the choice of sparsity parameters.

In this example, SCA outperforms SPC because it finds a better optimization solution. This comparison could be made difficult by the fact that they have different objective functions. However, in this case, even though SCA is optimizing a different objective function, it outperforms SPC at *optimizing the SPC objective function*. Table 2 lists the objective values of SPC (14) evaluated using the solutions of the SCA and SPC algorithms with various γ. When $\gamma \in \{36, 48, 60, 66\}$, the SCA algorithm outputs a solution that achieves a higher value of the SPC objective, suggesting that the SPC algorithm is likely to return local optima.

5. Applications

In this section, we applied SCA to real data. The first application is the sparse coding of natural images. It illustrates the utility of sparse PCA as independent component analysis. Supplementary Section S3.1 contains another application of SCA to blind source separation of images. Next, we demonstrate the ability of SCA in handling high-dimensional problems (i.e., $p > n$) through a transcriptome sequencing dataset and a targeted sample of Twitter friendship network. These datasets are of large scale. To our knowledge, no other current implementations of sparse PCA can efficiently handle a large matrix at the scale. As such, we will restrict our discussion to SCA.

5.1. Sparse Coding of Images

Low-level visual layers, such as retina, the lateral geniculate nucleus, and the primary visual cortex (V1) are shared processing components in mammalian. The receptive fields in the V1 can be characterized as being spatially localized, oriented and bandpass (i.e., selective to structure at different spatial scales). To understand V1, one line of research focuses on finding sparse and linearly independent codes for natural images, which provides an efficient representation for later stages of processing (Field 1994; Olshausen and Field 1996; Bell and Sejnowski 1997). This type of research is based on the hypothesis of sparse coding, that is, any perceived scenes can be synthesized via the linear combination of some small subsets of basis images (Lee et al. 2006; Gregor and LeCun 2010)). In this application, we show that sparse PCA produces a set of bases for natural images that resembles those found in Olshausen and Field (1996).

We used 10 natural images from Olshausen and Field (1996), each of which contains 512 × 512 pixels. We followed the same whitening process as described by the authors. Next, we randomly sampled a total of 12,000 small image patches the ten images, where each patch contains 16 × 16 pixels. This was followed by a centering step that subtracts each pixel by the mean of all 256 pixels. We vectorized each patch of image and put them into the rows of a data matrix, $X \in \mathbb{R}^{n\times p}$, where $n = 12{,}000$ and $p = 256$. Finally, we applied SCA to the transposed data matrix, X^{T} (note that this is sparse coding). For this exploratory analysis, we set $k = 49$ to find 49 sparse PCs (the same result holds for various selections of k) with the default sparsity parameter, $\gamma = \sqrt{pk}$. In particular, for the varimax rotation, we normalized the rows to unit length rescaled them afterward, as recommended by Kaiser (1958). In the output of SCA, the estimated scores $S \in \mathbb{R}^{p\times k}$ contains the basis images, and the estimated sparse loadings $Y \in \mathbb{R}^{n\times k}$ encodes how the basis images are linearly combined to form each image patch (i.e., Y contains the linear coefficients).

Figure 6 displays the 49 image bases returned by PCA and SCA, where each image represents one column of S (transformed into a 16×16 array). For SCA, all of the basis images appeared to exhibit simple patterns, such as lines and edges. As for PCA, the oriented structure in the first few basis images does not arise as a result of the oriented structures in natural images, yet more likely because of the existence of those components with low spatial frequency (Field 1987).

5.2. Analysis of Single-Cell Gene Expression Data

Single-cell transcriptome sequencing (scRNA-seq) provides high-throughput transcriptome expression quantification at individual cell level. It has been widely used across biological disciplines. For example, patterns of gene expression can be identified through clustering analysis. This helps uncover the existence of rare cell types within a cell population that have never been seen (Plasschaert et al. 2018; Montoro et al. 2018). In this application, we aimed to use SCA to extract the sparse PCs of genes that characterize some known cell types.

For this application, we used the human pancreatic islet cell data from Baron et al. (2016). We removed the genes that do not exhibit variation across all cells (i.e., zero standard deviation) and removed the cell types that contain fewer than 100 cells. This resulted in a data matrix $X \in \mathbb{R}^{n\times p}$ of $n = 8451$ cells across nine cell types and $p = 17{,}499$ genes, with X_{ij} measuring the expression level of gene j in cell i. X is sparse; it contains 10.8% nonzero elements. We applied SCA on X to find $k = 9$ sparse gene PCs. We set the sparsity parameter to $\gamma = \log(pk) \approx 12$, as we aimed for particularly sparse PCs (i.e., each PC is consist of a small number of genes). The algorithm took about 5 min (24 iterations) to complete on a single processor (3.3GHz). As a result, each column of the loading matrix contains a small

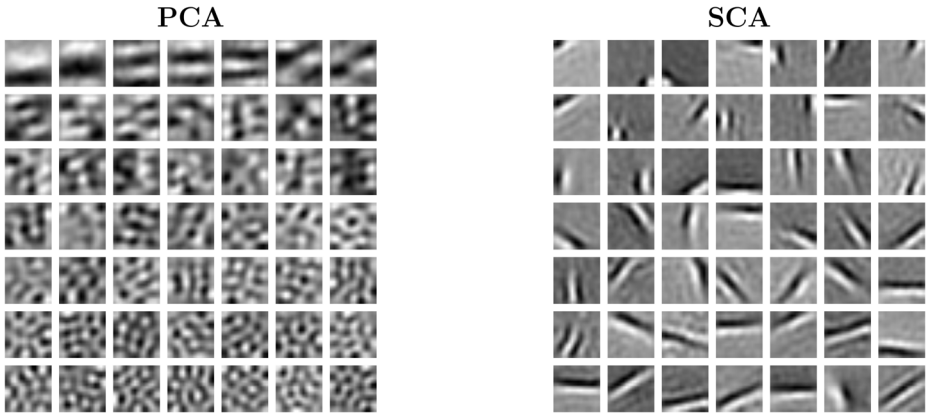

Figure 6. Sparse image encoding using PCA (left) and SCA (right). For both method, shown are the 49 image bases (i.e., component scores) extracted from natural images. Each image basis is in 16×16 pixel.

Table 3. Sparse gene PCs estimated by SCA.

PC	# of genes	Gene name(s)
1	1	INS
2	1	SST
3	1	GCG
4	8	CTRB2, REG1A, REG1B, REG3A, SPINK1 ...
5	15	CELA3A, CPA1, CTRB1, PRSS1, PRSS2 ...
6	1	IAPP
7	1	PPY
8	3	CLU, GNAS, TTR
9	61	ACTG1, EEF1A1, FTH1, FTL, TMSB4X ...

NOTE: For each gene PC, the number of genes (i.e., the number of nonzeros in the loadings) and the top five genes according to the absolute loadings are reported.

number of nonzero elements, suggesting that most of the gene PCs consist of one or a few genes. Table 3 lists the names of these genes for each PCs. For example, the PC 2 consists of only one gene, SST. Despite the simple structure of PCs, these PCs picked up informative gene markers for individual cell types. To see this, we calculated the scores for each cell using the 9 PCs (That is, each cell gets 9 scores, each of which corresponds to one of the nine PCs.) Figure 7 displays the box plots of the scores stratified by cell type. For example, the expression of the SST gene (which solely composes the 2nd PC) identifies the "delta" cells. This result highlights the power of scRNA-seq in capturing cell-type specific information and suggests the applicability of our methods to high-dimensional biological data.

5.3. Clustering of Twitter Friendship Network

This application serves in a grand efforts of ours to study political communication on social media, like Twitter. The information on Twitter is organized so that users primarily read the tweets of their "friends." In order to select content, a user can freely "follow" (and "unfollow") any other accounts, and we call these other accounts the friends of it. Thanks to this design, the communication on Twitter can be contextualized by the friendship network. As such, we hypothesize that user's community membership in the network offers the context of user's opinion expression on social media (Zhang, Chen, and Rohe 2022; Zhang, Chen, and Lukito 2022; Zhang et al. 2022). To study the hypothesis, a key step is to cluster Twitter accounts using their friendship network. In this section, we demonstrate large-scale network clustering using sparse PCA.

For this application, we collected a targeted sample from the Twitter friendship network in August 2018 (Chen, Zhang, and Rohe 2020). In this sample, there are $n = 193{,}120$ Twitter accounts who follow a total of $p = 1{,}310{,}051$ accounts, after filtering out the accounts with few followers or followings. We defined the graph adjacency matrix $A \in \{0,1\}^{n\times p}$ with $A_{ij} = 1$ if and only if account i follows account j.[11] This resulted in a sparse A with about 0.02% entries being 1. We applied SMA to A with $k = 100$ and default sparsity parameters. This analysis was computationally tractable; one iteration of the SMA algorithm took about 54 min on a single processor (2.5GHz), thanks to the efficient algorithm that computes the sparse SVD (Baglama and Reichel 2005). Figure 2 displays seven example columns of Y. Using the output $Z \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{p\times k}$ from SMA, the clusters of Twitter accounts were determined as follows (same as in Section 4.2): the ith row account of A was assigned to the lth row cluster if Z_{il} was the greatest in the ith row of Z, that is, $|Z_{il}| \geq |Z_{il'}|$ for all $l' = 1, 2, \ldots, k$, and the jth column account of A was assigned to the lth column cluster if Y_{jl} was the greatest in the jth row of Y, $|Y_{jl}| \geq |Y_{jl'}|$ for all $l' = 1, 2, \ldots, k$. Upon detailed evaluation of these clusters, we showed that our clustering of Twitter accounts formed homogeneous, connected, and stable social groups (Zhang, Chen, and Rohe 2022). For example, we found that a user is more likely to retweet the content that originated from another member in the same clusters (p-value $< 10^{-16}$ in a χ^2 test). More interestingly, the estimated row clusters and column clusters are matched (Rohe, Qin, and Yu 2016), that is, the kth row cluster tends to follow the accounts in the kth column cluster. To illustrate this, we quantified the number of followings from the row clusters to the corresponding column clusters. Figure 8 displays the results for 50 selected clusters that are related to U.S. politics. It can be seen that the number of

[11] The columns of A are not centered nor scaled. One alternative is to use the normalized version of A. For example, define the regularized graph Laplacian as $L \in \mathbb{R}^{n\times p}$ with $L_{ij} = A_{ij}/\sqrt{(r_i + \bar{r})(c_j + \bar{c})}$, where $r_i = \sum_j A_{ij}$ is the sum of the ith row of A, $c_j = \sum_i A_{ij}$ is the sum the jth column of A. Here, $\bar{r}$ and $\bar{c}$ are the means of r_i's and c_j's respectively. (Zhang and Rohe 2018).

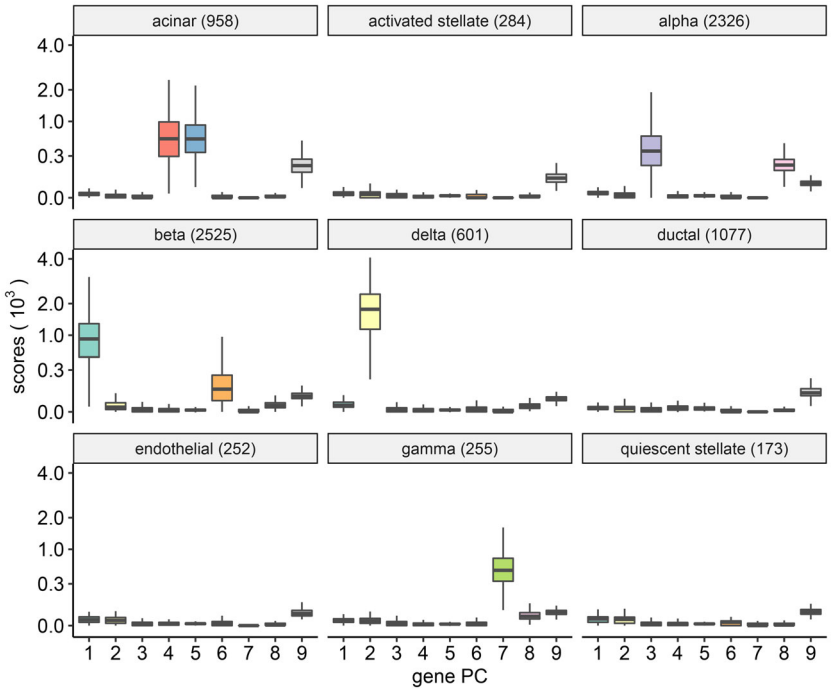

Figure 7. Scores of sparse gene principal components (PCs) stratified by cell types. Each panel displays one of nine cell types with the names of cell types and the number of cells reported on the top strips. For each cell type, a box depicts the component scores for nine sparse gene PCs.

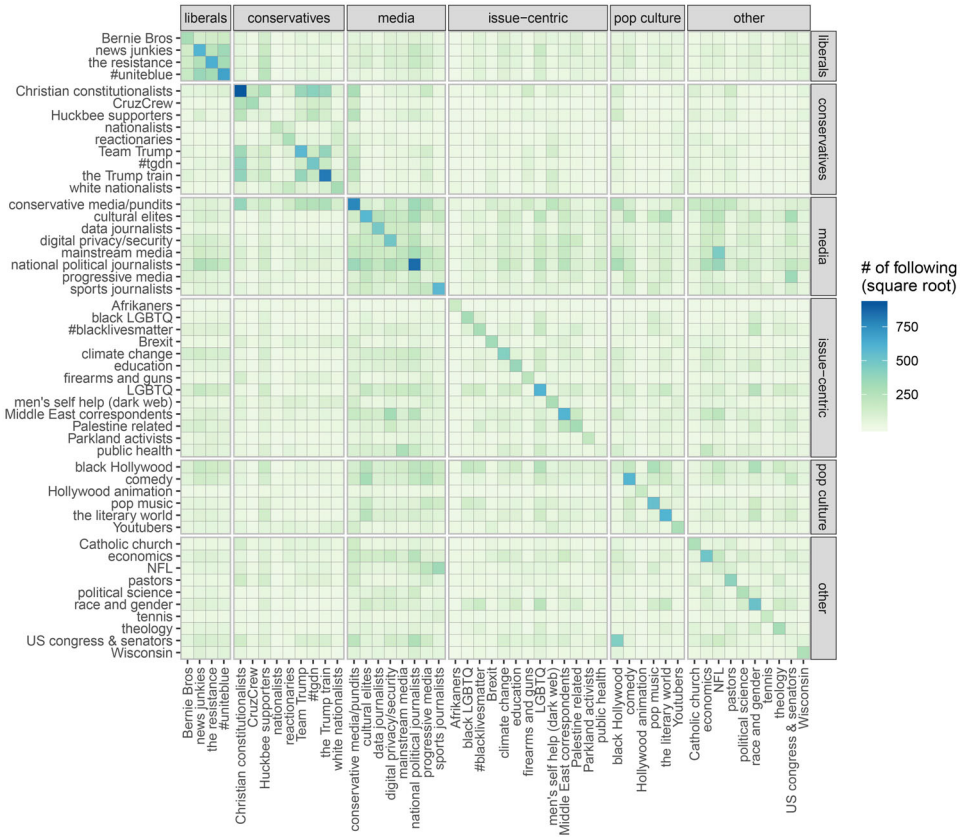

Figure 8. Heat map of friend counts between row and column clusters of Twitter accounts. Each row and column corresponds to a cluster. The row and column panels indicate cluster category, with the category names shown in the top and right strips. The color shades indicate the number of followings from the row cluster to the column cluster, after the square root transformation.

followings between each paired row and column clusters (i.e., the diagonals in Figure 8) showed marked enrichment. These results suggest the efficacy of our methods for analysis of social network data.

6. Discussions

In this article, we introduced SCA, a new method for sparse PCA, and SMA, an extension for two-way matrix analysis. SCA differs from the existing sparse PCA methods in that it estimates column sparse PCs, that is PCs that are sparse in an orthogonally rotated basis. This is particularly useful when the singular vectors of a data matrix (or the eigenvectors of the covariance matrix) are not readily sparse. We demonstrated that it explains more variance in the data than the state-of-the-art methods of sparse PCA. In addition, the algorithm is also stable and robust against a wide choices of tuning parameters. In practice, SCA is advantageous when multiple PCs are desired because it does not require the deflation.

Supplementary Materials

The supplementary materials contain: (1) additional discussion on choosing the sparsity parameters and a data-driven cross-validation framework for it, (2) further discussion on the soft thresholding step of SCA algorithm, (3) a comparison between SCA and independent component analysis, with a data example.

Acknowledgments

We thank Sündüz Keleş, Sébastien Roch, Po-Ling Loh, Michael A Newton, Yini Zhang, Muzhe Zeng, Alex Hayes, E Auden Krauska, Jocelyn Ostrowski, Daniel Conn, and Shan Lu for all the helpful discussions.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

The authors gratefully acknowledge National Science Foundation grant DMS-1612456 and DMS-1916378 and Army Research Office grant W911NF-15-1-0423.

ORCID

Fan Chen http://orcid.org/0000-0003-4508-6023

Appendix A. Technical proofs

Proof of Proposition 1. Since D is diagonal, and B can be any matrix including diagonal, the inequality result holds. Furthermore, given any fixed Z and Y subject to the constraints in (2) (i.e., Y's columns are not the leading eigenvectors of X), the maximizer on the left-hand-side is $B^* = Z^{\mathrm{T}}XY$ which is not diagonal.[12] Hence, the inequality is strict. □

Proof of Lemma 1. We rewrite the objective function:

$$\begin{aligned}\left\|X - ZBY^{\mathrm{T}}\right\|_{\mathrm{F}}^2 &= \mathrm{tr}\left[\left(X - ZBY^{\mathrm{T}}\right)^{\mathrm{T}}\left(X - ZBY^{\mathrm{T}}\right)\right] \\ &= \|X\|_{\mathrm{F}}^2 - 2\,\mathrm{tr}\left(X^{\mathrm{T}}ZBY^{\mathrm{T}}\right) + \mathrm{tr}\left(B^{\mathrm{T}}B\right) \\ &= \|X\|_{\mathrm{F}}^2 - \mathrm{tr}\left[B^{\mathrm{T}}\left(2Z^{\mathrm{T}}XY - B\right)\right].\end{aligned}$$

For fixed Z and Y, take the derivative of B and set it to zero. We have the optimizer $B^* = Z^{\mathrm{T}}XY$ and the squared optimal value is $\|X\|_{\mathrm{F}}^2 - \left\|Z^{\mathrm{T}}XY\right\|_{\mathrm{F}}^2$. Recognizing that $\|X\|_{\mathrm{F}}^2$ is determined, the desired formulation (13) follows. □

Remark 4 (Minimal matrix reconstruction error of PMD). If B is constrained to a diagonal matrix in (12), then the squared minimal value is $\|X\|_{\mathrm{F}}^2 - \sum_{i=1}^k d_i^2$, where $d_i = \left[Z^{\mathrm{T}}XY\right]_{ii}$ for $i = 1, 2, ..., k$.

Proof. From the proof of Lemma 1, we have

$$\left\|X - ZDY^{\mathrm{T}}\right\|_{\mathrm{F}}^2 = \|X\|_{\mathrm{F}}^2 - \mathrm{tr}\left[D^{\mathrm{T}}\left(2Z^{\mathrm{T}}XY - D\right)\right].$$

Then, take the derivative of D and set it to zero. This yields the solution $\hat{D} = \mathrm{diag}(d_i)$, where $d_i = \left[U^{\mathrm{T}}XV\right]_{ii}$. Finally, plugging-in the maximizer $\hat{D}$ gives the claimed optimal value. Note that $\sum_{i=1}^k d_i^2 \leq \left\|U^{\mathrm{T}}XV\right\|_{\mathrm{F}}^2$. □

Proof of Lemma 2. Suppose the low-rank SVD of $C \in \mathbb{R}^{p\times k}$ is UDV^{T}, where $U \in \mathcal{V}(p,k)$, $V \in \mathcal{U}(k)$, and $D \in \mathbb{R}^{k\times k}$ is diagonal. Then,

$$\left\|C^{\mathrm{T}}X\right\|_{\mathrm{F}}^2 = \mathrm{tr}\left(X^{\mathrm{T}}CC^{\mathrm{T}}X\right) = \mathrm{tr}\left(X^{\mathrm{T}}UD^2U^{\mathrm{T}}X\right).$$

The trace quadratic form is maximized at $X^* = UR$, for any orthogonal matrix $R \in \mathcal{U}(k)$. In particular, when $R = V$, $X^* = \mathrm{polar}(C)$. □

References

Amini, A. A., and Wainwright, M. J. (2009), "High-Dimensional Analysis of Semidefinite Relaxations for Sparse Principal Components," *The Annals of Statistics*, 37, 2877–2921. [1]

Baglama, J., and Reichel, L. (2005), "Augmented Implicitly Restarted Lanczos Bidiagonalization Methods," *SIAM Journal on Scientific Computing*, 27, 19–42. [11]

Baron, M., Veres, A., Wolock, S. L., Faust, A. L., Gaujoux, R., Vetere, A., Ryu, J. H., Wagner, B. K., Shen-Orr, S. S., Klein, A. M., et al. (2016), "A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-Cell Population Structure," *Cell Systems*, 3, 346–360. [10]

Bell, A. J., and Sejnowski, T. J. (1997), "The "Independent Components" of Natural Scenes are Edge Filters," *Vision Research*, 37, 3327–3338. [10]

Bernaards, C. A., and Jennrich, R. I. (2005), "Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis," *Educational and Psychological Measurement*, 65, 676–696. [5]

Berthet, Q., and Rigollet, P. (2013), "Optimal Detection of Sparse Principal Components in High Dimension," *The Annals of Statistics*, 41, 1780–1815. [1]

Cai, T. T., Ma, Z., and Wu, Y. (2013), "Sparse PCA: Optimal Rates and Adaptive Estimation," *The Annals of Statistics*, 41, 3074–3110. [1]

Carroll, J. B. (1953), "An Analytical Solution for Approximating Simple Structure in Factor Analysis," *Psychometrika*, 18, 23–38. [6]

Chen, F., Zhang, Y., and Rohe, K. (2020), "Targeted Sampling from Massive Block Model Graphs with Personalized PageRank," *Journal of the Royal Statistical Society*, Series B, 82, 99–126. [11]

Chen, F., Roch, S., Rohe, K., and Yu, S. (2021), "Estimating Graph Dimension with Cross-Validated Eigenvalues," arXiv preprint arXiv:2108.03336. [3]

Comon, P. (1994), "Independent Component Analysis, A New Concept?" *Signal Processing*, 36, 287–314. [2]

d'Aspremont, A., Ghaoui, L. E., Jordan, M. I., and Lanckriet, G. R. G. (2007), "A Direct Formulation for Sparse PCA Using Semidefinite Programming," *SIAM Review*, 49, 434–448. [1]

Donoho, D. L. (1995), "De-Noising by Soft-Thresholding," *IEEE Transactions on Information Theory*, 41, 613–627. [5]

[12] Generally, $B^* = \left(Z^{\mathrm{T}}Z\right)^{-1}Z^{\mathrm{T}}XY\left(Y^{\mathrm{T}}Y\right)^{-1}$ if Z and Y are full-rank, or $B^* = \left(Z^{\mathrm{T}}Z\right)^{+}Z^{\mathrm{T}}XY\left(Y^{\mathrm{T}}Y\right)^{+}$ if either Z or Y is singular, where A^+ is the Moore–Penrose inverse of matrix A.

Field, D. J. (1987), "Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells," *Journal of the Optical Society of America A*, 4, 2379–2394. [10]

Field, D. J. (1994), "What is the Goal of Sensory Coding?" *Neural Computation*, 6, 559–601. [10]

Fortunato, S. (2010), "Community Detection in Graphs," *Physics Reports*, 486, 75–174. [9]

Gallivan, K. A., and Absil, P. A. (2010), "Note on the Convex Hull of the Stiefel Manifold," Florida State University. [2]

Gataric, M., Wang, T., and Samworth, R. J. (2020), "Sparse Principal Component Analysis via Axis-Aligned Random Projections," *Journal of the Royal Statistical Society*, Series B, 82, 329–359. DOI:10.1111/rssb.12360 [8]

Gregor, K., and LeCun, Y. (2010), "Learning Fast Approximations of Sparse Coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 399–406, Madison, WI, USA, Omnipress. [10]

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983), "Stochastic Blockmodels: First Steps," *Social Networks*, 5, 109–137. [9]

Horn, R. A., and Johnson, C. R. (1985), *Matrix Analysis*, Cambridge, UK: Cambridge University Press. [5]

Hotelling, H. (1933), "Analysis of a Complex of Statistical Variables Into Principal Components," *Journal of Educational Psychology*, 24, 417–441. [1]

Hu, Z., Pan, G., Wang, Y., and Wu, Z. (2016), "Sparse Principal Component Analysis via Rotation and Truncation," *IEEE Transactions on Neural Networks and Learning Systems*, 27, 875–890. [7]

Jeffers, J. N. R. (1967), "Two Case Studies in the Application of Principal Component Analysis," *Journal of the Royal Statistical Society*, Series C, 16, 225–236. [1]

Jennrich, R. I. (2001), "A Simple General Procedure for Orthogonal Rotation," *Psychometrika*, 66, 289–306. [6]

Johnstone, I. M., and Lu, A. Y. (2009), "On Consistency and Sparsity for Principal Components Analysis in High Dimensions," *Journal of the American Statistical Association*, 104, 682–693. [1]

Jolliffe, I. T. (1995), "Rotation of Principal Components: Choice of Normalization Constraints," *Journal of Applied Statistics*, 22, 29–35. [2]

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003), "A Modified Principal Component Technique based on the LASSO," *Journal of Computational and Graphical Statistics*, 12, 531–547. [7]

Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. (2010), "Generalized Power Method for Sparse Principal Component Analysis," *Journal of Machine Learning Research*, 11, 517–553. [1,2,5,7]

Kaiser, H. F. (1958), "The Varimax Criterion for Analytic Rotation in Factor Analysis," *Psychometrika*, 23, 187–200. [2,5,10]

——— (1960), "The Application of Electronic Computers to Factor Analysis," *Educational and Psychological Measurement*, 20, 141–151. [2]

Karrer, B., and Newman, M. E. J. (2011), "Stochastic Blockmodels and Community Structure in Networks," *Physical Review E*, 83, 016107. [5]

Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006), "Efficient Sparse Coding Algorithms," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pp. 801–808, Cambridge, MA, MIT Press. [10]

Mackey, L. (2008), "Deflation Methods for Sparse PCA," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pp. 1017–1024, Red Hook, NY: Curran Associates Inc. [1,7]

Moghaddam, B., Weiss, Y., and Avidan, S. (2006), "Generalized Spectral Bounds for Sparse LDA," in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 641–648, New York: Association for Computing Machinery. [1]

Montoro, D. T., Haber, A. L., Biton, M., Vinarsky, V., Lin, B., Birket, S. E., Yuan, F., Chen, S., Leung, H. M., Villoria, J, et al. (2018), "A Revised Airway Epithelial Hierarchy Includes CFTR-Expressing Ionocytes," *Nature*, 560, 319–324. [10]

Nocedal, J., and Wright, S. (2006), *Numerical Optimization* (2nd ed.), New York: Springer. [4]

Olshausen, B. A., and Field, D. J. (1996), "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images," *Nature*, 381, 607. [10]

Pearson, K. (1901), "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, 559–572. [1]

Plasschaert, L. W., Žilionis, R., Choo-Wing, R., Savova, V., Knehr, J., Roma, G., Klein, A. M., and Jaffe, A. B. (2018), "A Single-Cell Atlas of the Airway Epithelium Reveals the CFTR-Rich Pulmonary Ionocyte," *Nature*, 560, 377–381. [10]

Rohe, K., and Zeng, M. (2020), "Vintage Factor Analysis with Varimax Performs Statistical Inference," arXiv:2004.05387. [2,5,9]

Rohe, K., Qin, T., and Yu, B. (2016), "Co-clustering Directed Graphs to Discover Asymmetries and Directional Communities," *Proceedings of the National Academy of Sciences*, 113, 12679–12684. [11]

Shen, D., Shen, H., and Marron, J. S. (2013), "Consistency of Sparse PCA in High Dimension, Low Sample Size Contexts," *Journal of Multivariate Analysis*, 115, 317–333. [1]

Shen, H., and Huang, J. Z. (2008), "Sparse Principal Component Analysis via Regularized Low Rank Matrix Approximation," *Journal of Multivariate Analysis*, 99, 1015–1034. [6,7,8]

Thurstone, L. L. (1931), "Multiple Factor Analysis," *Psychological Review*, 38, 406. [2]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [5]

Tillmann, A. M., and Pfetsch, M. E. (2014), "The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing," *IEEE Transactions on Information Theory*, 60, 1248–1259. [1]

Vu, V. Q., and Lei, J. (2013), "Minimax Sparse Principal Subspace Estimation in High Dimensions," *The Annals of Statistics*, 41, 2905–2947. [1,2,3]

Vu, V. Q., Cho, J., Lei, J., and Rohe, K. (2013), "Fantope Projection and Selection: A Near-Optimal Convex Relaxation of Sparse PCA," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pp. 2670–2678, Red Hook, NY, USA, 2013. Curran Associates Inc. [1]

Wang, T., Berthet, Q., and Samworth, R. J. (2016), "Statistical and Computational Trade-Offs in Estimation of Sparse Principal Components," *The Annals of Statistics*, 44, 1896–1930. [1]

Witten, D. M., Tibshirani, R., and Hastie, T. (2009), "A Penalized Matrix Decomposition, with Applications to Sparse Principal Components and Canonical Correlation Analysis," *Biostatistics*, 10, 515–534. [1,2,6,7]

Yuan, X.-T., and Zhang, T. (2013), "Truncated Power Method for Sparse Eigenvalue Problems," *Journal of Machine Learning Research*, 14, 899–925. [7]

Zhang, Y., and Rohe, K. (2018), "Understanding Regularized Spectral Clustering via Graph Conductance," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 10654–10663, Red Hook, NY, USA, 2018. Curran Associates Inc. [11]

Zhang, Y., Chen, F., and Lukito, J. (2022), "Network Amplification of Politicized Information and Misinformation about Covid-19 by Conservative Media and Partisan Influencers on Twitter," *Political Communication*, 40, 24–47. [11]

Zhang, Y., Chen, F., and Rohe, K. (2022), "Social Media Public Opinion as Flocks in a Murmuration: Conceptualizing and Measuring Opinion Expression on Social Media," *Journal of Computer-Mediated Communication*, 27, zmab021. [11]

Zhang, Y., Yue, Z., Yang, X., Chen, F., and Kwak, N. (2022), "How a Peripheral Ideology Becomes Mainstream: Strategic Performance, Audience Reaction, and News Media Amplification in the Case of Qanon Twitter Accounts," *New Media & Society*, 14614448221137324. [11]

Zou, H., and Xue, L. (2018), "A Selective Overview of Sparse Principal Component Analysis," *Proceedings of the IEEE*, 106, 1311–1320. [1,9]

Zou, H., Hastie, T., and Tibshirani, R. (2006), "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, 15, 265–286. [1,7]