Learning Sparse PCA with Stabilized ADMM Method on Stiefel Manifold

Mingkui Tan, Zhibin Hu, Yuguang Yan, Jiezhang Cao, Dong Gong, and Qingyao Wu

Abstract—Sparse principal component analysis (SPCA) produces principal components with sparse loadings, which is very important for handling data with many irrelevant features and also critical to interpret the results. To deal with orthogonal constraints, most previous approaches address SPCA with several components using techniques such as deflation technique and convex relaxations. However, the deflation technique usually suffers from suboptimal solutions due to poor approximations. On the other hand, the convex relaxations are often computationally expensive. To address the above issues, in this paper, we propose to address SPCA over the Stiefel manifold directly, and develop a stabilized Alternating Direction Method of Multipliers (SADMM) to handle the nonconvex orthogonal constraints. Compared to traditional ADMM, the proposed SADMM method converges well with a wide range of parameters and obtains a better solution. We also theoretically study the convergence property of the proposed SADMM method. Furthermore, most existing methods ignore an inherent drawback of SPCA — the importance of different components is not considered when doing feature selection, which often makes the selected features non-optimal. To address this, we further propose a two-stage method which considers the importance of different components to select the most important features. Empirical studies on both synthetic and real-world datasets show that the proposed algorithms achieve better performance compared to existing state-of-the-art methods.

Index Terms—Sparse principal component analysis, Feature selection, Stiefel manifold, ADMM

1 Introduction

Principal Component Analysis (PCA) [33] [45] [55] is one of the most important tools for data analysis [18] [15] and dimension reduction [14] [44]. Given a multivariate dataset, PCA aims to find a set of orthogonal loadings to transform possibly correlated features into a set of *principal components* that are linearly uncorrelated. However, the principal components obtained by PCA are linear combinations of *all* original features, which leads to poor interpretability of the results [31]. Another main flaw of the classical PCA is that in many real-world datasets, there can be many noisy features, which may seriously contaminate the covariance matrix and dramatically affect the accuracy of loading vectors. Regarding these drawbacks, the sparse PCA (SPCA) problem, which seeks to find principal components with sparse loading vectors, has attracted great attention in machine learning, data mining and signal processing communities [2], [31], [39], [43], [50], [59].

In the past decade, many attempts have been made to address SPCA problem. In general, there are two points should be considered—the balance between sparsity of loadings and variance [31], [43], [58], and the orthogonality between loadings [27]. Some existing algorithms may obtain loading vectors with the leading one being highly dense while others being highly sparse [27], [32]. However, in the context of feature selection, it is unreasonable to use the leading one to explain the variance and use others to achieve the sparsity. Besides, in practice, the orthogonality [28], [37] between loadings in SPCA is easy to lose when pursuing sparsity [23], [25] on loadings. Here, the orthogonality indicates the independence of the loadings. It is meaningless that the loadings are very close, and thus handling the orthogonal constraints is important.

In practice, given a covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, a direct SPCA estimator can be formulated as the following nonconvex problem [31]:

$$\max_{\mathbf{V}\in\mathbb{R}^{d\times r}} \ \mathrm{tr}(\mathbf{V}^{\mathsf{T}}\mathbf{\Sigma}\mathbf{V}) - \lambda||\mathbf{V}||_1, \ \text{s.t.} \ \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I}, \tag{1}$$

where r is the number of principal components, $\mathbf{V} \in \mathbb{R}^{d\times r}$ denotes r corresponding loadings, and $\lambda > 0$ is a trade-off parameter. Here, the ℓ_1-regularizer $||\mathbf{V}||_1$ is applied to induce sparse solutions. A variety of methods have been proposed to address the above problem. Unfortunately, due to its nonconvex nature, Problem (1) is difficult to address. Most approaches thus extend power methods [56] or apply convex relaxations [53] to address SPCA. However, these methods can be suboptimal because of poor approximations. Some approaches deal with SPCA by applying deflation techniques [41], which find multiple sparse components one after another by iteratively solving Problem (1) with $r = 1$. However, these methods are suboptimal, since the optimal loadings when the dimension equals r may not be coincident with the optimal loadings when the dimension increases to $r + 1$. Moreover, these methods may neglect the interactions between multiple components, thus obtaining suboptimal solutions.

Different from deflation based methods, block algorithms find multiple sparse principal components jointly. Relying on convex relations, these block algorithms are guaranteed to stay away from most of the local minima [53]. However, the convex relaxations are often expensive to address and the computational complexity of these methods is very high, which make them unsuitable for high-dimensional data. Besides, some block algorithms suffer from the imbalance of sparsity among loadings, like one version of GPower [32], in which the leading loading is often highly dense and takes

- *M. Tan, Z. Hu, Y. Yan, J. Cao, and Q. Wu are with the South China University of Technology, China. E-mail: mingkuitan@scut.edu.cn; huzhibinscut@gmail.com; yan.yuguang@mail.scut.edu.cn; caojiezhang@gmail.com; qyw@scut.edu.cn.*
- *D. Gong is with the University of Adelaide, Australia. E-mail: dong.gong@adelaide.edu.au.*
- *M. Tan, Z. Hu and Y. Yan are the co-first authors; D. Gong and Q. Wu are the co-corresponding authors.*

account of the most variance while other loadings are enforced to achieve sparsity.

Besides the above issues, there is one inherent drawback in the SPCA model in (1) — the importance of different components is inherently the same when considering feature selection using the regularizer $||\mathbf{V}||_1$. For example, we consider a case of 2 loadings (i.e., $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$) and assume that only the first component (namely $\mathbf{v}_1$) is a true component and $\mathbf{v}_2$ is a redundant one. Due to the orthogonal constraint $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$, it follows that $||\mathbf{v}_1||_2 = ||\mathbf{v}_2||_2 = 1$. Without loss of generality, suppose $\mathbf{v}_2$ is mistakenly considered, then $\mathbf{v}_2$ has equal importance to $\mathbf{v}_1$ in the regularizer $||\mathbf{V}||_1$ (even though $\mathbf{v}_2$ is a redundant component). In this case, at least one feature related to $\mathbf{v}_2$ shall be considered due to the constraint $||\mathbf{v}_2||_2 = 1$. This feature is apparently a redundant feature since $\mathbf{v}_2$ is redundant. In practice, some redundant features shall be mistakenly included if some redundant components are mistakenly considered.

To address the above issues, instead of directly handling the orthogonal constraints, we propose to perform the optimization over the Stiefel manifold [16]. Specifically, we propose a stabilized Alternating Direction Method of Multipliers (SADMM) algorithm for addressing SPCA by exploiting the geometric structure of Stiefel manifold. Since the loading vectors are updated on the Stiefel manifold, the orthogonal constraint always holds. Moreover, in contrast to conventional ADMM methods which may converge slowly when the algorithm parameters are not well chosen, the proposed SADMM method converges well with a wide range of parameters. Furthermore, our proposed algorithm does not rely on the estimation of the covariance matrix, thus enjoying good scalability to high-dimensional data.

The main contributions of this paper are concluded as follows.

- We propose a stabilized ADMM (SADMM) method over the Stiefel manifold to efficiently solve the SPCA problem by splitting the penalty term into two terms, each of which is associated with one block in ADMM. We provide a theoretical analysis of the convergence property for the proposed algorithm. In particular, the proposed method converges well with a wide range of parameters.
- We take the importance of different components into consideration and develop a two-stage SADMM method to select the most important features. Since the features related to a component with larger singular value are often more important, the two-stage SADMM method helps to select more important features with more discriminative power or larger variance.
- Extensive experiments on several real-world datasets demonstrate the effectiveness of the proposed SADMM method and the two-stage SADMM method.

The rest of the paper is organized as follows. In Section 2, we review some important related work. In Section 3, we present the notations used in the paper. In Section 4, we introduce the proposed methods for SPCA and provide some theoretical results. In Section 5, we present the experimental results on both synthetic and real-world datasets. We conclude this work in Section 6.

2 Related Studies

During the past decade, a variety of methods have been proposed to handle the SPCA problem. There are two critical points in the SPCA problem — pursuiting the sparse representation [24], [38], [42] of loadings and holding the orthogonal constraint [10], [35] between loadings.

Some methods solved the optimization problem through matrix approximation. Jolliffe et al. [31] solved the LASSO based problem via a projected gradient descent approach to find the modified principle components with zero loadings. Shen and Huang [47] adapted the SVD to PCA and obtained the principle components through solving a low-rank approximation problem under some sparsity-inducing penalties. Zou et al. [58] formulated the SPCA as a regression-type optimization problem with LASSO (elastic net) penalty

$$\min_{\mathbf{V},\mathbf{U}\in\mathbb{R}^{d\times r}} \|\mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{V}^\mathsf{T}\|_F^2 + \lambda\|\mathbf{U}\|_F^2 + \sum_{i=1}^{r}\gamma_i\|\mathbf{v}_i\|_1, \quad \text{s.t. } \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}.$$

And then, they obtained the top-r principle components by iteratively solving two subproblems w.r.t. $\mathbf{U}$ and $\mathbf{V}$, respectively.

In [30], the authors proposed a two-stage method which uses a diagonal thresholding based preprocessing step to select relevant variables and then obtains principle components. Ma et al. [40] proposed to estimate the sparse principal subspace based on iterative thresholding. Moghaddam et al. [43] studied a discrete spectral formulation for SPCA and provided a greedy branch-and-bound search approach. Grbovic et al. [21] introduced two types of grouping constraints into the SPCA problem to ensure the reliability of the resulting groups.

Some methods solved SPCA relying on power methods. Journee et al. [32] proposed a generalized power method (GPower) to solve the single-unit ($r = 1$) or block ($r > 1$) SPCA problems with ℓ_0-norm or ℓ_1-norm regularizers. Yuan and Zhang [57] proposed an efficient SPCA approach based on a truncated power method (TPower). In [26], the single component SPCA problem was formulated as a nonlinear eigenproblem and solved via an inverse power method.

Given a covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, the convex SDP relaxation [13], [52] for SPCA can be formulated as:

$$\max_{\mathbf{W}\in\mathbb{R}^{d\times d}} \operatorname{tr}(\mathbf{\Sigma}\mathbf{W}) - \lambda\|\mathbf{W}\|_1, \text{s.t. } \operatorname{tr}(\mathbf{W}) = 1, 0 \preceq \mathbf{W} \preceq \mathbf{I}_d,$$

where $\mathbf{W}$ is a projector to the principle subspace (e.g., a reparametrization of $\mathbf{V}\mathbf{V}^\mathsf{T}$ where $\mathbf{V}$ is the loading matrix), $0 \preceq \mathbf{W} \preceq \mathbf{I}_d$ means the eigenvalues of $\mathbf{W}$ should be in $[0, 1]$, and $\lambda > 0$ is a trade-off parameter. In [13], the authors proposed a method called DSPCA to relax the SPCA problem as a convex SDP problem with an ℓ_1-norm constraint. d'Asprenont et al. [12] proposed a convex relaxation based greedy algorithm (PathSPCA) to compute a full set of solutions. Amni et al. [4] provided the theoretical optimality guarantees of the SDP relaxation based on the spiked covariance model in [29]. Considering the drawbacks of deflation in DSPCA, Vu et al. [52] proposed a Fantope projection and selection (FPS) method, which formulated the SPCA problem as an SDP and solved it in the sparse principle subspace directly. Gu et al. [22] proposed a convex estimator and a nonconvex estimator based on the SDP relaxation with novel regularizations. Wang et al. [53] proposed a two-stage method which consists of sparse orthogonal iteration pursuit as a main stage and a SDP relaxation estimator for initialization. Asteris et al. [5] studied the connection between SPCA and the bipartite maximum weight matching problem, and proposed a bipartite matching based SPCA approach. Bouveyron et al. [7] used Bayesian Variable Selection to obtain several sparse components with the same sparsity pattern. Ma et al. [39] focused on high-dimensional problem and proposed a convex estimator based on the Sum-of-Squares relaxation. In [11], the authors summarized and analysed the existing SPCA methods that using ℓ_1-norm to pursuit feature selection. Nevertheless, there are some major differences between our proposed method and these mentioned methods, which are summarized as follows: 1) These methods adopt deflation techniques to find multiple sparse components, which neglect the interactions between multiple components. Our proposed method find multiple sparse components jointly. 2) These methods neglect the importance of different components when considering feature selection, our proposed method address this drawback to select more important features.

3 Notations

Matrices and vectors are denoted by upper-case and lower-case boldface characters, respectively. A set is denoted by calligraphic letter. Given an integer $m \in \mathbb{Z}^+$, let $[m] = \{1, ..., m\}$. Given an index set $\mathcal{I} \subseteq [m]$, let $\mathbf{A}(:, \mathcal{I})$ denote the columns of $\mathbf{A}$ regarding $\mathcal{I}$. Let the superscript $^\mathsf{T}$ denote the transpose of a vector/matrix, $\mathbf{0}$ be a vector/matrix with all zeros, and $\mathrm{diag}(\mathbf{v})$ be a diagonal matrix with diagonal elements being $\mathbf{v}$, $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}(\mathbf{A}\mathbf{B}^\mathsf{T})$ be the inner product of $\mathbf{A}$ and $\mathbf{B}$, where $\mathrm{tr}(\cdot)$ is the trace operator. Let $\|\mathbf{v}\|_p$ be the ℓ_p-norm of a vector $\mathbf{v}$. The ℓ_1-norm of a matrix $\mathbf{Z}$ is defined as $\|\mathbf{Z}\|_1 = \sum_{ij} |Z_{ij}|$. Also, we let $||\mathbf{Z}||_{2,1} = \sum_i ||\mathbf{Z}_{i\cdot}||_2$ be the $\ell_{2,1}$ norm of matrix $\mathbf{Z}$. The operator $\max(\mathbf{u}, \mathbf{v})$ operates on each dimension. Let $\mathbf{Z} = \mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$ be the SVD of any matrix $\mathbf{Z} \in \mathbb{R}^{n \times d}$. The Frobenius norm of $\mathbf{Z}$ is defined as $\|\mathbf{Z}\|_F = \|\boldsymbol{\sigma}\|_2$. For a convex function $\Omega(\mathbf{Z})$, we denote by $\partial\Omega(\mathbf{Z})$ its subdifferential at $\mathbf{Z}$. Lastly, given a square matrix $\mathbf{A}$, we define $\mathrm{sym}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\mathsf{T})$ and $\mathrm{skew}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} - \mathbf{A}^\mathsf{T})$.

4 SPCA on the Stiefel Manifold

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$ be a dataset of n instances with d features. In this paper, we seek to learn SPCA by solving the following optimization problem:

$$\min_{\mathbf{V} \in \mathbb{R}^{d \times r}} \ \frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^\mathsf{T}||_F^2 + \lambda\Omega(\mathbf{V}), \ \text{ s.t., } \ \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}, \quad (2)$$

where λ is a trade-off parameter and $\Omega(\mathbf{V})$ is some sparsity-inducing regularizer. Here, we consider the ℓ_1-norm regularizer on $\mathbf{V}$, namely $\Omega(\mathbf{V}) = ||\mathbf{V}||_1$, which encourages $\mathbf{V}$ to be sparse. One may also apply $\ell_{2,1}$-norm on $\mathbf{V}$ to encourage row sparsity of $\mathbf{V}$. Note that, $\mathbf{V}$ lies on the Stiefel manifold

$$\mathrm{St}_d^r = \mathcal{M}_r = \{\mathbf{V} \in \mathbb{R}^{d \times r} | \mathbf{V}^\top\mathbf{V} = \mathbf{I}\},$$

which is a set of d by r orthogonal matrices. Besides, $\mathcal{M}_r$ is compact, and its dimension is $(dr - \frac{1}{2}r(r+1))$. Since $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$, it follows that

$$\frac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^\mathsf{T}||_F^2 = \frac{1}{2}||\mathbf{X}||_F^2 - \frac{1}{2}||\mathbf{X}\mathbf{V}||_F^2. \quad (3)$$

In this paper, we first consider solving the following problem:

$$\min_{\mathbf{V} \in \mathcal{M}_r} -\frac{1}{2}||\mathbf{X}\mathbf{V}||_F^2 + \lambda||\mathbf{V}||_1. \quad (4)$$

Here, the Stiefel manifold $\mathcal{M}_r$ is nonconvex, so Problem (4) is hard to address. The non-smooth term $||\mathbf{V}||_1$ makes the problem even more challenging to address. To address this problem, we propose a novel Alternating Direction Method of Multipliers (ADMM) on the Stiefel manifold.

Algorithm 1 ADMM for SPCA

```
Require: Input data $\mathbf{X}$, $r$, $\lambda$, $\rho_0$, $\rho_{\max}$ and $\beta \in (1, 2)$.
Initialize $\mathbf{V}_0, \boldsymbol{\Upsilon}_0$ and $\boldsymbol{\Omega}_0 = \mathbf{0}$.
For $t = 0, ..., T-1$
    Compute
    $\mathbf{V}_{t+1} = \arg\min_{\mathbf{V} \in \mathcal{M}_r} \mathcal{L}_{\rho_t}(\mathbf{V}, \boldsymbol{\Upsilon}_t, \boldsymbol{\Omega}_t)$.
    Compute
    $\boldsymbol{\Upsilon}_{t+1} = \arg\min_{\boldsymbol{\Upsilon}} \mathcal{L}_{\rho_t}(\mathbf{V}_{t+1}, \boldsymbol{\Upsilon}, \boldsymbol{\Omega}_t)$.
    Update $\boldsymbol{\Omega}_{t+1} = \boldsymbol{\Omega}_t + \rho_t(\mathbf{V}_{t+1} - \boldsymbol{\Upsilon}_{t+1})$.
    Update $\rho_{t+1} = \min(\beta\rho_t, \rho_{\max})$.
End For
```

The ADMM method has been widely applied in distributed optimization and statistic learning areas. Following [8], to decouple the non-smooth $\|\mathbf{V}\|_1$ from the smooth term, we introduce an auxiliary variable matrix $\boldsymbol{\Upsilon}$ and add an additional equality constraint $\mathbf{V} = \boldsymbol{\Upsilon}$ into (4). We then transform Problem (4) into the following equivalent:

$$\min_{\mathbf{V} \in \mathcal{M}_r} \ -\frac{1}{2}||\mathbf{X}\mathbf{V}||_F^2 + \lambda||\boldsymbol{\Upsilon}||_1, \ \text{ s.t., } \ \mathbf{V} = \boldsymbol{\Upsilon}. \quad (5)$$

By introducing a dual variable matrix $\boldsymbol{\Omega} \in \mathbb{R}^{d \times r}$ for the equality constraint, we obtain an augmented Lagrangian:

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{V}, \boldsymbol{\Upsilon}, \boldsymbol{\Omega}) = &- \frac{1}{2}||\mathbf{X}\mathbf{V}||_F^2 + \lambda||\boldsymbol{\Upsilon}||_1 \\ &+ \langle \boldsymbol{\Omega}, \mathbf{V} - \boldsymbol{\Upsilon} \rangle + \frac{\rho}{2}||\mathbf{V} - \boldsymbol{\Upsilon}||_F^2, \end{aligned} \quad (6)$$

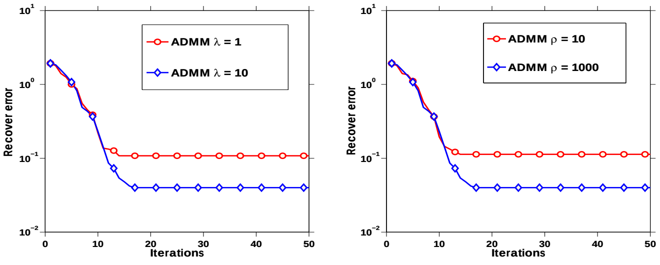
where $\rho > 0$ is a penalty parameter. By iteratively minimizing $\mathcal{L}_\rho(\cdot)$ w.r.t. $\mathbf{V}$ and $\boldsymbol{\Upsilon}$, the general ADMM algorithm is conducted in Algorithm 1.

In Algorithm 1, the parameter ρ increases monotonically, which aims to reduce the gap between $\mathbf{V}$ and $\boldsymbol{\Upsilon}$ gradually. The minimization over $\mathbf{V}$ is performed on the Stiefel manifold $\mathcal{M}_r$, which will be depicted in detail in the following section. Given $\mathbf{V}$, the optimization of $\mathcal{L}_\rho(\mathbf{V}, \boldsymbol{\Upsilon}, \boldsymbol{\Omega})$ over $\boldsymbol{\Upsilon}$ can be written as

$$\min_{\boldsymbol{\Upsilon}} \ \lambda||\boldsymbol{\Upsilon}||_1 + \frac{\rho}{2}||\mathbf{V} - \boldsymbol{\Upsilon} + \frac{1}{\rho}\boldsymbol{\Omega}||_F^2, \quad (7)$$

which has a closed-form solution $\boldsymbol{\Upsilon}^*$. Specifically, let $\mathbf{Z} := \mathbf{V} + \frac{1}{\rho}\boldsymbol{\Omega}$, we can compute $\boldsymbol{\Upsilon}^*$ by a simple shrinkage operation

$$\Upsilon_{ij}^* := \max(|Z_{ij}| - \frac{\lambda}{\rho}, 0) \ \mathrm{sign}(Z_{ij}). \quad (8)$$



(a) Convergence w.r.t. λ (b) Convergence w.r.t. ρ

Fig. 1. Convergence of Algorithm 1 on synthetic data $\mathbf{X} \in \mathbb{R}^{1000 \times 1000}$.

In Algorithm 1, the shrinkage operation in Eq. (8) is sensitive to the value of λ/ρ, which may incur convergence issues. Specifically, since ρ increases monotonically, λ/ρ varies w.r.t. iterations. As a result, Algorithm 1 may be hard to converge if λ and ρ_0 are not well adjusted.

To illustrate the above problem, we conduct a synthetic experiment on a toy dataset with the detailed setting being provided in Section 5.1.1. The results are shown in Fig. 1. We observe that when we change the best setting ($\lambda = 10$ and $\rho = 1000$), the convergence behavior of Algorithm 1 is sensitive to the value of λ/ρ. If λ/ρ is not appropriately chosen, Algorithm 1 may be far away from the promising solution.

4.1 Stabilized ADMM for SPCA

In this section, we propose a stabilized ADMM to address the convergence issue incurred by the value of λ/ρ. To this end, we split the augmented Lagrangian function $\mathcal{L}_\rho(\mathbf{V}, \mathbf{\Upsilon}, \mathbf{\Omega})$ into two parts by introducing the following two functions:

$$\begin{aligned}\mathcal{A}_\rho(\mathbf{V}, \mathbf{\Upsilon}, \mathbf{\Omega}) = &-\frac{1}{2}||\mathbf{X}\mathbf{V}||_F^2 \\ &+ \langle \mathbf{\Omega}, \mathbf{V} - \mathbf{\Upsilon} \rangle + \frac{\rho}{2}||\mathbf{V} - \mathbf{\Upsilon}||_F^2,\end{aligned} \tag{9}$$

and

$$\begin{aligned}\mathcal{B}_\gamma(\mathbf{V}, \mathbf{\Upsilon}, \mathbf{\Omega}) =&\lambda||\mathbf{\Upsilon}||_1 \\ &+ \langle \mathbf{\Omega}, \mathbf{V} - \mathbf{\Upsilon} \rangle + \frac{\gamma}{2}||\mathbf{V} - \mathbf{\Upsilon}||_F^2.\end{aligned} \tag{10}$$

Following the conventional ADMM scheme, we propose a stabilized ADMM scheme in Algorithm 2. Specifically, at each iteration, $\mathbf{V}$ and $\mathbf{\Upsilon}$ are updated by minimizing $\mathcal{A}_\rho(\cdot)$ and $\mathcal{B}_\gamma(\cdot)$, respectively. $\mathbf{\Omega}$ is updated relying on the parameter γ only.

Letting $\mathbf{Z} := \mathbf{V} + \frac{1}{\gamma}\mathbf{\Omega}$, similar to Algorithm 1, we can update $\mathbf{\Upsilon}$ via the following shrinkage operation:

$$\Upsilon_{ij}^{t+1} := \max(|Z_{ij}| - \frac{\lambda}{\gamma}, 0)\ \mathrm{sign}(Z_{ij}). \tag{11}$$

In the proposed Algorithm 2, when updating $\mathbf{V}$ by minimizing $\mathcal{A}_\rho(\cdot)$, we can keep the parameter ρ increasing monotonically to make the residual of the equality constraint decrease fast. On the other hand, when updating $\mathbf{\Upsilon}$, we use a fixed γ to obtain a fixed shrinkage operation, which makes the update of $\mathbf{\Upsilon}$ more stable. As a result, although ρ keeps monotonically increasing, the thresholding for shrinkage λ/γ keeps fixed, leading to more stable and faster convergence. Meanwhile, thanks to the increasing parameter ρ in the penalty term in Eq. (9), the residual of the equality constraint still converges fast.

Algorithm 2 Stabilized ADMM (SADMM) for SPCA

```
Require: Input data $\mathbf{X}$, $r$, $\lambda$, $\rho_0$, $\rho_{\max}$, $\gamma$ and $\beta \in (1, 2)$.
Initialize $\mathbf{V}_0, \mathbf{\Upsilon}_0$ and $\mathbf{\Omega}_0 = \mathbf{0}$.
For $t = 0, ..., T-1$
    Compute
    $\mathbf{V}_{t+1} = \arg\min_{\mathbf{V} \in \mathcal{M}_r}\ \ \mathcal{A}_{\rho_t}(\mathbf{V}, \mathbf{\Upsilon}_t, \mathbf{\Omega}_t)$.
    Compute
    $\mathbf{\Upsilon}_{t+1} = \arg\min_{\mathbf{\Upsilon}}\ \ \mathcal{B}_\gamma(\mathbf{V}_{t+1}, \mathbf{\Upsilon}, \mathbf{\Omega}_t)$.
    Update $\mathbf{\Omega}_{t+1} = \mathbf{\Omega}_t + \gamma(\mathbf{V}_{t+1} - \mathbf{\Upsilon}_{t+1})$.
    Update $\rho_{t+1} = \min(\beta\rho_t, \rho_{\max})$.
End For
```

By exploiting the geometric structure of Riemannian manifold (i.e., Stiefel manifold), SADMM can effectively handle the orthogonal constraint in SPCA. For convenience of presentation, hereafter we denote SADMM for SPCA on the Stiefel manifold as StSPCA.

Initialization of $\mathbf{V}$. In our case, by setting $\rho = 0$ and $\mathbf{\Omega}_0 = \mathbf{0}$ at the beginning of ADMM, the minimization of $\mathcal{A}_\rho(\mathbf{V}, \mathbf{\Upsilon}, \mathbf{\Omega})$ is reduced to: $\min_{\mathbf{V} \in \mathcal{M}_r} -\frac{1}{2}||\mathbf{X}\mathbf{V}||_F^2$. The optimal solution to this problem, denoted by $\mathbf{V}_0$, can be trivially obtained by a truncated SVD of rank r on $\mathbf{X}$, i.e., $[\widetilde{\mathbf{U}}, \widetilde{\mathbf{S}}, \widetilde{\mathbf{V}}] = \mathrm{svds}(\mathbf{X}, r)$. After that, we initialize $\mathbf{V}_0$ by $\mathbf{V}_0 = \widetilde{\mathbf{V}}$. In experiments, we empirically study the effects on different initializations of $\mathbf{V}$. We first adopt random initialization and truncated SVD to initialize $\mathbf{V}$. Then we run the proposed algorithm based on the obtained $\mathbf{V}$. Interestingly, even with random initializations, our method still shows a competitive convergence behavior.

4.2 Conjugate Gradient on Riemannian Manifold

The optimization of $\mathcal{A}_\rho(\mathbf{V}, \mathbf{\Upsilon}, \mathbf{\Omega})$ w.r.t. $\mathbf{V}$ is performed on the Stiefel manifold $\mathcal{M}_r$. To equip the optimization on manifolds, we need to introduce some geometries over the Stiefel manifold $\mathcal{M}_r$. First, the tangent space to $\mathcal{M}_r$ at $\mathbf{V}$ (i.e., the set of all tangent vectors to $\mathcal{M}_r$ at $\mathbf{V}$) , denoted by $T_{\mathbf{V}}\mathcal{M}_r$, is defined as

$$T_{\mathbf{V}}\mathcal{M}_r := \left\{\mathbf{Z} \in \mathbb{R}^{d \times r} :\ \mathbf{Z}^{\mathsf{T}}\mathbf{V} + \mathbf{V}^{\mathsf{T}}\mathbf{Z} = \mathbf{0}\right\}.$$

On the tangent space $T_{\mathbf{V}}\mathcal{M}_r$ at any $\mathbf{V} \in \mathcal{M}_r$, if we introduce the standard inner product as a metric $\langle \mathbf{Y}, \mathbf{Z} \rangle_{\mathbf{V}} := \mathrm{tr}(\mathbf{Y}^{\mathsf{T}}\mathbf{Z})$, $\forall \mathbf{Y}, \mathbf{Z} \in T_{\mathbf{V}}\mathcal{M}_r$, then $\mathcal{M}_r$ can be viewed as a Riemannian submanifold of $\mathbb{R}^{d \times r}$. Given a smooth function $f(\mathbf{V})$ on $\mathcal{M}_r$, let $\mathbf{G} = \nabla f$ be the gradient of f in the Euclidean space. Then, the Riemannian gradient of f at $\mathbf{V}$, denoted by $\mathrm{grad}f(\mathbf{V})$, is given as the orthogonal projection of $\mathbf{G}$ onto the tangent space:

$$\mathrm{grad}f(\mathbf{V}) = P_{T_{\mathbf{V}}\mathcal{M}_r}(\mathbf{G}), \tag{12}$$

where

$$P_{T_{\mathbf{V}}\mathcal{M}_r}(\mathbf{Z}) = \mathbf{Z} - \mathbf{V}\ \ \mathrm{sym}(\mathbf{V}^{\mathsf{T}}\mathbf{Z}). \tag{13}$$

Different from the optimization in the Euclidean space, the search direction on a manifold should follow a path on the manifold [1]. At the kth iteration, based on the Riemannian gradient $\mathrm{grad}f(\mathbf{V}_k)$ at $\mathbf{V}_k$, we can construct the search direction $\boldsymbol{\zeta}_k = -\mathrm{grad}f(\mathbf{V}_k)$, which corresponds to the steepest descent direction in the Euclidean space and is known to be slow in convergence. Hence, we address this by applying a Nonlinear Conjugate Gradient (NCG) method, which has shown promising performance and fast convergence speed [34], [51]. Algorithm 3 summarizes the NCG method. The convergence threshold is set to $1e-5$ and the maximum number is set to 30.

Following NCG in the Euclidean space, the conjugate search $\boldsymbol{\zeta}_k$ direction can be calculated by $\boldsymbol{\zeta}_k = -\mathrm{grad}f(\mathbf{V}_k) + \beta_k\boldsymbol{\zeta}_{k-1}$. However, since $\mathrm{grad}f(\mathbf{V}_k) \in T_{\mathbf{V}_k}\mathcal{M}_r$, $\mathrm{grad}f(\mathbf{V}_{k-1}) \in T_{\mathbf{V}_{k-1}}\mathcal{M}_r$, and $\boldsymbol{\zeta}_{k-1}$ are in different tangent spaces of the manifold, the above equation is not applicable on Riemannian manifolds. To address this, we need to introduce the *Vector Transport* $\mathcal{T}_{\mathbf{X} \to \mathbf{Y}}(\boldsymbol{\zeta}_{\mathbf{X}})$, which transports $\boldsymbol{\zeta}_{\mathbf{X}}$ from one tangent space $T_{\mathbf{X}}\mathcal{M}_r$ to another tangent space $T_{\mathbf{Y}}\mathcal{M}_r$. Here, $\mathcal{T}_{\mathbf{X} \to \mathbf{Y}}(\boldsymbol{\zeta}_{\mathbf{X}})$ can be computed by $P_{T_{\mathbf{Y}}\mathcal{M}_r}(\boldsymbol{\zeta}_{\mathbf{X}})$. After that, we can compute the conjugate search direction by

$$\boldsymbol{\zeta}_k = -\mathrm{grad}f(\mathbf{V}_k) + \beta_k\mathcal{T}_{\mathbf{V}_{k-1} \to \mathbf{V}_k}(\boldsymbol{\zeta}_{k-1}),$$

where β_k can be calculated by the Fletcher-Reeves rule [17].

Once the search direction $\boldsymbol{\zeta}$ is computed, we can move $\mathbf{V}$ alongh the direction $\boldsymbol{\zeta}$ with the step size θ. The step size is chosen according to the strong Wolfe conditions, and the details can be found in the supplementary material. After that, in order to make $\mathbf{V}$ stay on the manifold, we map it on the manifold $\mathcal{M}_r$ by the following *Retraction* operation:

$$R_{\mathbf{V}}(\theta\boldsymbol{\zeta}) = \mathrm{qf}(\mathbf{V} + \theta\boldsymbol{\zeta}), \tag{14}$$

where $\mathrm{qf}(\mathbf{A})$ denotes the Q factor of the QR decomposition of $\mathbf{A}$.

Once a search direction $\boldsymbol{\zeta}$ is computed, we can move $\mathbf{V}$ along the direction $\boldsymbol{\zeta}$ and to stay on the manifold by a *Retraction* mapping on $\mathcal{M}_r$. Specifically, let $\xi = \theta\boldsymbol{\zeta}$, the retraction at $\mathbf{V}$ along the direction $\boldsymbol{\zeta}$ with the step size θ can be computed by

$$R_{\mathbf{V}}(\xi) = \mathrm{qf}(\mathbf{V} + \xi), \tag{15}$$

where $\mathrm{qf}(\mathbf{A})$ denotes the Q factor of the QR decomposition of $\mathbf{A}$. Here, the strong Wolfe conditions are used to choose θ_k, and the details can be found in the supplementary material.

Algorithm 3 Nonlinear Conjugate Gradient for Updating $\mathbf{V}$.

```
1: Initialize $\mathbf{V}_0$ and $\boldsymbol{\zeta}_0 = \mathbf{0}$. Let $k = 1$.
2: Compute the Riemannian gradient $\text{grad}f(\mathbf{V}_k)$ (by (12)).
3: Compute a conjugate direction by $\boldsymbol{\zeta}_k = -\text{grad}f(\mathbf{V}_k) + \beta_k \mathcal{T}_{\mathbf{V}_{k-1} \to \mathbf{V}_k}(\boldsymbol{\zeta}_{k-1})$.
4: Choose a step size $\theta_k$ satisfying the strong Wolfe conditions, and set $\mathbf{V}_{k+1} = R_{\mathbf{V}_k}(\theta_k \boldsymbol{\zeta}_k)$.
5: Stop if the stopping conditions are achieved; otherwise, let $k = k + 1$ and go to step 2.
```

4.3 Two-Stage SADMM for SPCA

In most existing SPCA methods, the importance of different components is not considered when inducing the sparsity. For example, in Problem (4), assuming $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r]$, the ℓ_1-regularizer $||\mathbf{V}||_1$ treats each component equally. Due to the orthogonal constraint $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$, we have $||\mathbf{v}_1||_2 = ||\mathbf{v}_2||_2 = ...||\mathbf{v}_r||_2 = 1$. In practice, some components (e.g., $\mathbf{v}_r$) might be redundant if r is not well estimated. If a redundant component is mistakenly considered (e.g., $\mathbf{v}_r$), it will have equal importance to leading component $\mathbf{v}_1$ in the regularization. As a result, some redundant features would be mistakenly chosen, since at least one feature related to $\mathbf{v}_r$ will be selected due to the constraint $||\mathbf{v}_r||_2 = 1$.

Algorithm 4 Two-Stage SADMM for SPCA (StSPCA-2S)

```
Require: Input data $\mathbf{X}$, $r$, $\lambda$, $\rho_0$, $\gamma$ and $K$.
1: Stage I: Call Algorithm 2 to obtain $\mathbf{V}$. Compute the feature score $\mathbf{s} = \sum_{i=1}^{r} \sigma_i |\mathbf{v}_i|$. Choose $K$ features with the $K$ largest values in $\mathbf{s}$.
2: Stage II: Perform Algorithm 2 with those selected features to obtain $\mathbf{V}_\mathrm{s}^*$. Let $\mathbf{V}^* = \mathbf{0}$ and update $\mathbf{V}^*$ with $\mathbf{V}_\mathrm{s}$.
```

To address the above issue, we here propose a two-stage SADMM method in Algorithm 4, which takes the importances of both components and features into consideration. To this end, in the first stage, we conduct Algorithm 2 to obtain $\mathbf{V}$, and then select important features from loadings; in the second stage, we perform Algorithm 2 with the important features selected in the first stage.

Specifically, given $\mathbf{V}$ obtained in the first stage, for the component $\mathbf{v}_i$, we calculate the corresponding singular value $\sigma_i = \|\mathbf{X}\mathbf{v}_i\|_2$ to measure the importance of the component $\mathbf{v}_i$. Considering that the magnitude of $|v_{ij}|$ indicates the importance of the feature j w.r.t. the component $\mathbf{v}_i$, we compute the score s_j as follows to measure the importance of the feature j based on all the components:

$$s_j = \sum_{i=1}^{r} \sigma_i |v_{ij}|. \tag{16}$$

Based on this, we select K most important features with the K largest values of s_j, where $j \in \{1, \ldots, n\}$. As a result, the selected features are obtained by considering the importance of different components.

4.4 Convergence Analysis

We analyze the convergence of the proposed SADMM in Algorithm 2. To this end, we firstly show the convergence property of step 3 of Algorithm 2 in the following theorem.

Theorem 1. *Let $\{\mathbf{V}_k\}$ be the sequence generated by the NCG method with the strong Wolfe line search, where $0 < c_1 < c_2 < 1/2$, then we have $\lim_{k\to\infty} \inf ||\text{grad}f(\mathbf{V}_k)|| = 0$.*

The proof can be found in Appendices A and B.

Now we are ready to analyze the convergence of SADMM in Algorithm 2. Let $\varphi(\mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^\mathsf{T}\|_F^2$ and $g(\boldsymbol{\Upsilon}) = \lambda\|\boldsymbol{\Upsilon}\|_1$. Note that $\varphi(\mathbf{V})$ is smooth but non-convex, and $g(\boldsymbol{\Upsilon})$ is non-smooth. Theorem 2 characterizes the cluster point of the sequence generated by Algorithm 2, and shows the global convergence of Algorithm 2.

Theorem 2. *(Global subsequential convergence)*
(i) Suppose that $\|\mathbf{X}^\mathsf{T}\mathbf{X}\|_F \leq \rho$, any convergent subsequence $\{(\mathbf{V}_{t_i}, \boldsymbol{\Upsilon}_{t_i}, \boldsymbol{\Omega}_{t_i})\}$ of the sequence $\{(\mathbf{V}_t, \boldsymbol{\Upsilon}_t, \boldsymbol{\Omega}_t)\}$ generated by stabilized ADMM converges to a cluster point $(\mathbf{V}^*, \boldsymbol{\Upsilon}^*, \boldsymbol{\Omega}^*)$, i.e., $\lim_{t\to\infty} \|\mathbf{V}_{t+1} - \mathbf{V}_t\|_F^2 + \|\boldsymbol{\Upsilon}_{t+1} - \boldsymbol{\Upsilon}_t\|_F^2 + \|\boldsymbol{\Omega}_{t+1} - \boldsymbol{\Omega}_t\|_F^2 = 0$.
(ii) Any cluster point $(\mathbf{V}^*, \boldsymbol{\Upsilon}^*, \boldsymbol{\Omega}^*)$ of a sequence $\{(\mathbf{V}_t, \boldsymbol{\Upsilon}_t, \boldsymbol{\Omega}_t)\}$ generated by Algorithm 2 is a stationary point of problem (4).

The detailed proof can be found in Appendix C.

4.5 Complexity Analysis

Our proposed method has good scalability for large-scale problems. Specifically, it only needs to perform rank-r truncated SVD one time for the initialization, which takes $O(mnr)$ complexity by PROPACK. The vector transport operation also takes $O(dr^2)$ complexity. Additionally, the QR decomposition of $d \times r$ matrices is required for the retraction, of which the complexity is $(2dr^2 - 2r^3/3)$ flops with fast Givens QR method [19].

5 Experiments

We verify the performance of proposed StSPCA and StSPCA-2S on both synthetic and real-world datasets. To thoroughly compare with the deflation technique, following [41], we also apply StSPCA to handle multiple components by conducting StSPCA with $r = 1$ iteratively, which is denoted by StSPCA-D.

Baselines and Parameters Setting. Several state-of-the-art SPCA methods are adopted as the baselines, including GPower0, GPower1, GPower0-B and GPower1-B [32], GRQI [36], IPM [26], Manopt [6], TPower [57], and Wang [53]. Among them, Wang, GPower0-B and GPower1-B deal with multiple components jointly, while the others rely on deflation techniques. Some methods mentioned in the related studies are not compared due to the absence of the source code.

All experiments are conducted in Matlab on a server with two Intel(R) Xeon E5-2620 v3 CPUs (2.40 GHz) and 128 GB memory. During comparison, we use grid search to determine the parameters γ and ρ in the range $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. For StSPCA and StSPCA-D, we vary λ in $(10^{-2}, 10^{-1}, 10^0, 10^1, 10^2)$ to control the selected features. For StSPCA-2S, we vary the parameter K to control the selected features. We repeat the experiments 10 times, and report the average results. For fair comparisons, no parallel technique is used for all the methods.

Real-world datasets. We use seven real-world datasets, which are commonly used for evaluating SPCA models. Table 1 shows the statistics of the real-world datasets.

TABLE 1
Details of Datasets

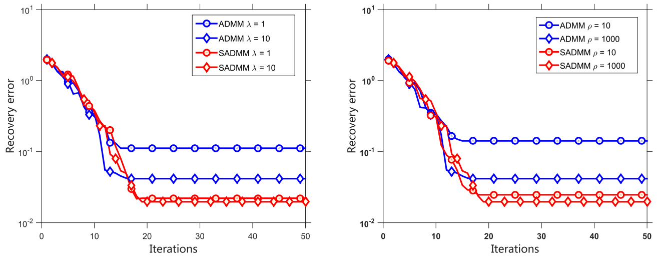
Dataset	#Instances	#Features	Reference
Colon	62	2,000	[3]
Prostate	102	2,135	[49]
Dlbcl	77	2,647	[48]
Leukemia	72	7,129	[20]
Duke	42	7,129	[54]
GCM	198	11,370	[46]
Breast	128	47,293	[54]
News20	15935	62,061	[9]

5.1 Experiments on Synthetic DataSets

Generation of synthetic data. We generate synthetic data $\{\mathbf{x}_i\}_{i=1}^{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is a covariance matrix. Following [22], [32], we generate $\boldsymbol{\Sigma} = \mathbf{V}_g \mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}_g^{\mathsf{T}}$, where $\boldsymbol{\sigma}$ denotes the vector of eigenvalues and $\mathbf{V}_g$ denotes the ground-truth sparse loading vectors. We choose the first 2 columns of $\mathbf{V}_g$ as sparse orthogonal vectors by setting the first 10 elements of the first column and the second 10 elements of the second column to be $1/\sqrt{10}$, and others to be 0. The rest columns of $\mathbf{V}_g$ are randomly sampled from a uniform distribution. For eigenvalues in $\boldsymbol{\sigma}$, we set $\sigma_1 = 400$ and $\sigma_2 = 300$, and the rests to be 1. *In other words, there are 20 relevant features, and the first two components are much more important.* Since the ground-truth loadings $\mathbf{V}_g$ is known, we use the recovery error $\epsilon = ||\mathbf{V}\mathbf{V}^{\mathsf{T}} - \mathbf{V}_g(:, 1:2)\mathbf{V}_g(:, 1:2)^{\mathsf{T}}||_F^2$ as the comparison metric, where $\mathbf{V}$ is the recovered loadings.

5.1.1 Comparison of ADMM and SADMM for SPCA

Firstly, we show the convergence behavior of ADMM for SPCA under different settings of ρ and λ. See Fig. 1, the best performance is achieved under the setting $\lambda = 10$ and $\rho = 1000$. Note that when we change $\lambda = 1$ (Fig. 1(a)) or $\rho = 10$ (Fig. 1(b)), the algorithm will be far away from the optimal result, which indicates the sensitivity of the shrinkage operation λ/ρ. Then we compare the convergence behavior of ADMM and SADMM for SPCA under different values of the parameters. From Figs. 2(a) and 2(b), we can observe that SADMM with different parameters converges to similar solutions, while ADMM produces a large difference in convergence. We conclude that SADMM achieves a better solution and is more stable than ADMM.

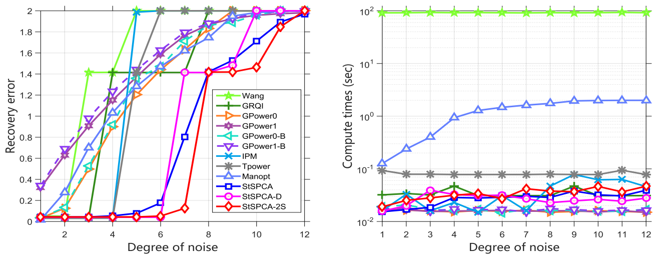


(a) Convergence w.r.t. λ (b) Convergence w.r.t. ρ

Fig. 2. Convergence comparison of ADMM and SADMM. Results on synthetic data $\mathbf{X} \in \mathbb{R}^{1000\times1000}$.

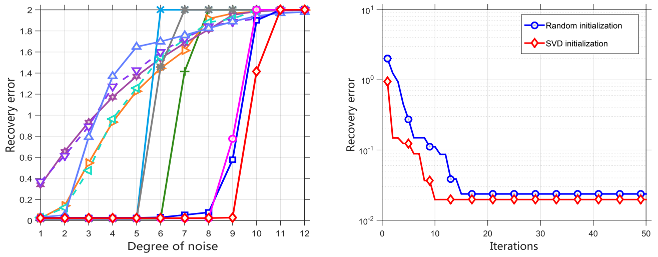
5.1.2 Comparison on Different Noises

In this experiment, we increase the strength of noise by multiplying irrelevant features by a noise factor from 1 to 12. We report the result on synthetic data $\mathbf{X} \in \mathbb{R}^{100\times1000}$ in Fig. 3 and $\mathbf{X} \in \mathbb{R}^{500\times5000}$ in Fig. 4(a). From Fig. 3(a), we observe that the GPower based methods and Manopt perform relatively worse than other methods; while the recovery errors of GRQI, IPM, TPower and Wang increases rapidly when the noise degree is around 4. On the contrary, the proposed StSPCA-2S, StSPCA and StSPCA-D perform well even when the noise factor is 6, which is much better than others. From Fig. 3(b), GPower based methods are fastest and our proposed methods have acceptable computational cost. The same observations can also be obtained from Fig. 4(a) (we omit the result of Wang, since it requires too much running time). This indicates that the proposed methods can extract important features when the data have many noisy features and are more stable to noises. Lastly, StSPCA-2S achieves the best performance, which demonstrates the effect of the proposed two-stage strategy.



(a) Recovery error (b) Running time

Fig. 3. Recovery performance on synthetic data $\mathbf{X} \in \mathbb{R}^{100\times1000}$.



(a) (b)

Fig. 4. (a): Recovery performance on synthetic data $\mathbf{X} \in \mathbb{R}^{500\times5000}$; (b): Convergence of different initialization methods of $\mathbf{V}$ for SADMM.

5.1.3 Comparison of Different Initialization Methods

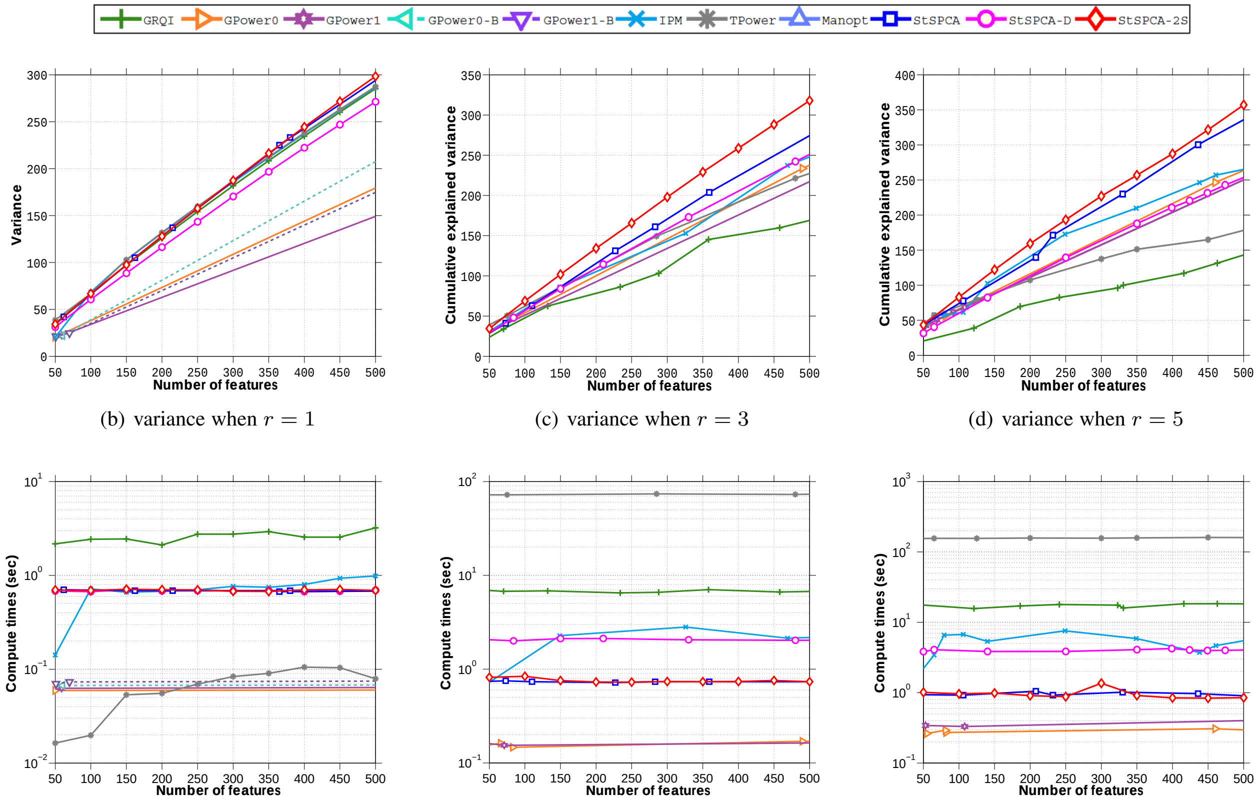
We study the convergence of the proposed method with different initialization strategies — the SVD initialization introduced in SADMM and random initialization. We report the results on the synthetic data $\mathbf{X} \in \mathbb{R}^{1000\times1000}$. Fig. 4(b) shows that even with random initialization, the proposed method still achieves a comparable result. This demonstrates that our proposed optimization methods on nonlinear manifolds are guaranteed to converge. Moreover, the proposed initialization helps to achieve faster convergence and a better solution. Furthermore, we have theoretically shown that the complexity of the proposed initialization strategy is low in Sec. 4.5. Thus, our proposed method can converge faster to a better solution with small additional complexity.

5.2 Results on Real-World DataSets

We then conduct experiments on several real-world datasets. Note that the results for some methods are absent since these methods require more than 10,000 seconds. Firstly, we compare the proposed methods with baselines on first five datasets. TABLE

TABLE 2
Total cumulative variance captured by k = 3 components on various datasets.

Method	Colon	Prostate	Dlbcl	Duke	Leukemia
GPower-0	1.713e+03	1.477e+04	4.153e+10	1.541e+03	1.899e+03
GPower-1	1.707e+03	1.457e+04	4.084e+10	1.573e+03	1.871e+03
GRQI	1.724e+03	1.653e+04	3.531e+10	1.462e+03	1.875e+03
IPM	1.638e+03	1.682e+04	4.028e+10	1.647e+03	1.921e+03
TPower	1.605e+03	1.597e+04	3.863e+10	1.497e+03	1.639e+03
Wang	2.173e+03	2.062e+04	-	-	-
StSPCA	2.116e+03	2.096e+04	4.408e+10	1.838e+03	2.391e+03
StSPCA-D	1.741e+03	1.672e+04	4.015e+10	1.627e+03	2.138e+03
StSPCA-2S	**2.354e+03**	**2.185e+04**	**4.768e+10**	**1.951e+03**	**2.508e+03**



(b) variance when $r = 1$ (c) variance when $r = 3$ (d) variance when $r = 5$

(e) running time when $r = 1$ (f) running time when $r = 3$ (g) running time when $r = 5$

Fig. 5. Results on the GCM dataset.

2 lists the total cumulative variance captured by $r = 3$ sparse components, where each one activates 50 features. Total cumulative variance represents the sum of variances of each extracted sparse principal component. Since GPower0-B, GPower1-B and Manopt obtain loading vectors with the leading one being highly dense (close to PCA) while the others being highly sparse, these methods may suffer from imbalance of sparsity among loadings. Therefore, we omit the results of them when $r > 1$. From TABLE 2, we observe that compared with deflation based methods, the block based methods (Wang, StSPCA and StSPCA-2S) achieve better performance, which proves that block based methods outperform deflation based methods when extracting multiple sparse components. Moreover, our proposed methods StSPCA and StSPCA-2S outperform other baselines in most cases, which proves the effectiveness of our proposed algorithms. In addition, StSPCA-2S achieves the highest values in all cases, which indicates that it is meaningful to consider the importance of different components.

In addition, we also compare our methods with baselines under different numbers of components and features on three larger datasets, which includes tens of thousands of instances and features. We extract different numbers of sparse principal components (i.e., $r \in \{1, 3, 5\}$) and features (varying from 50 to 500). We draw several observations as follows.

- **Effects on the number of components and features.** When $r = 1$, all the compared methods achieve the comparable performance. Our proposed algorithms achieve the highest values in most cases and are among the fastest algorithms. When $r = 3$ or 5, StSPCA-2S, and StSPCA outperform the other baselines, which indicates that the methods finding multiple components jointly are more effective compared to the deflation-based methods. Moreover, we can observe that our proposed algorithms can extract more important features. In particular, StSPCA-2S shows that considering the importance of different sparse components is more beneficial to feature extraction.
- **Comparison of StSPCA variants.** Compared with StSPCA-

D, StSPCA and StSPCA-2S show that block based methods are beneficial to extract multiple sparse components. Besides, StSPCA-2S achieves the better performance than StSPCA when $r = 3$ or 5, which is consistent with the observation on the synthetic data. This further demonstrates the effectiveness of our proposed two-stage method in Algorithm 4.

- **Comparison on efficiency.** GPower based methods are fastest in all cases, but they exhibit the worst performance in variance. The methods of TPower, GRQI, IPM take more running times when increasing the number of components. In addition, another efficient block based method Wang requires too much running time. Conversely, our proposed methods have acceptable computational cost while achieving the best performance in variance.

6 Conclusions

In this paper, we have proposed a stabilized ADMM (SADMM) to address the SPCA problem on the *Stiefel* manifold. The global convergence of the proposed algorithm has been theoretically analyzed. Based on SADMM, a two-stage method has also been proposed for SPCA. Extensive experiments on both synthetic and real-world datasets demonstrate superior performance compared to other methods in terms of interpretable variance and computational efficiency.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (NSFC) (No. 61602185 and 61876208), key project of NSFC (No. 61836003), Program for Guangdong Introducing Innovative and Enterpreneurial Teams 2017ZT07X183, Guangdong Provincial Scientific and Technological Funds under Grants (No. 2018B010107001, 2018B010108002 and 2017B090901008), Tencent AI Lab Rhino-Bird Focused Research Program (No. JR201902), CCF-Tencent Open Research Fund (No. RAGR20190103), Pearl River S&T Nova Program of Guangzhou 201806010081.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Y. Ait-Sahalia and D. Xiu. Principal component analysis of high-frequency data. *Journal of the American Statistical Association*, pages 1–17, 2018.

[3] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96(12):6745–6750, 1999.

[4] A. A. Amini and M. J. Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. *Annals of Statistics*, 37(5B):2877–2921, 2009.

[5] M. Asteris, D. Papailiopoulos, A. Kyrillidis, and A. G. Dimakis. Sparse pca via bipartite matchings. In *Neural Information Processing Systems (NIPS)*, pages 766–774, 2015.

[6] N. Boumal, B. Mishra, P.-A. Absil, R. Sepulchre, et al. Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research (JMLR)*, 15(1):1455–1459, 2014.

[7] C. Bouveyron, P. Latouche, and P.-A. Mattei. Bayesian variable selection for globally sparse probabilistic pca. *arXiv:1605.05918*, 2016.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[9] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[10] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, pages 854–863. JMLR. org, 2017.

[11] J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research (JMLR)*, 16(1):2859–2900, 2015.

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

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

[14] C. Ding, X. He, H. Zha, and H. D. Simon. Adaptive dimension reduction for clustering high dimensional data. In *IEEE International Conference on Data Mining (ICDM)*, pages 147–154. IEEE, 2002.

[15] N. Duntsch and G. Gediga. Modal-style operators in qualitative data analysis. In *IEEE International Conference on Data Mining (ICDM)*, pages 155–162. IEEE, 2002.

[16] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[17] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

[18] Y. Gil, D. Garijo, V. Ratnakar, R. Mayani, R. Adusumilli, H. Boyce, A. Srivastava, and P. Mallick. Towards continuous scientific data analysis and hypothesis evolution. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 4406–4414, 2017.

[19] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[20] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[21] M. Grbovic, C. R. Dance, and S. Vucetic. Sparse principal component analysis with constraints. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 935–941, 2012.

[22] Q. Gu, Z. Wang, and H. Liu. Sparse pca with oracle property. In *Neural Information Processing Systems (NIPS)*, pages 1529–1537, 2014.

[23] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan. Feature selection based on structured sparsity: A comprehensive study. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 28(7):1490–1507, 2017.

[24] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji. Discriminant sparse neighborhood preserving embedding for face recognition. *Pattern Recognition (PR)*, 45(8):2884–2893, 2012.

[25] J. Gui, D. Tao, Z. Sun, Y. Luo, X. You, and Y. Y. Tang. Group sparse multiview patch alignment framework with view consistency for image classification. *IEEE Transactions on Image Processing (TIP)*, 23(7):3126–3137, 2014.

[26] M. Hein and T. Bühler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca. In *Neural Information Processing Systems (NIPS)*, pages 847–855, 2010.

[27] Z. Hu, G. Pan, Y. Wang, and Z. Wu. Sparse principal component analysis via rotation and truncation. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 27(4):875–890, 2016.

[28] Z. Huang and L. Van Gool. A riemannian network for spd matrix learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[29] I. M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics*, pages 295–327, 2001.

[30] I. M. Johnstone and A. Y. Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 2012.

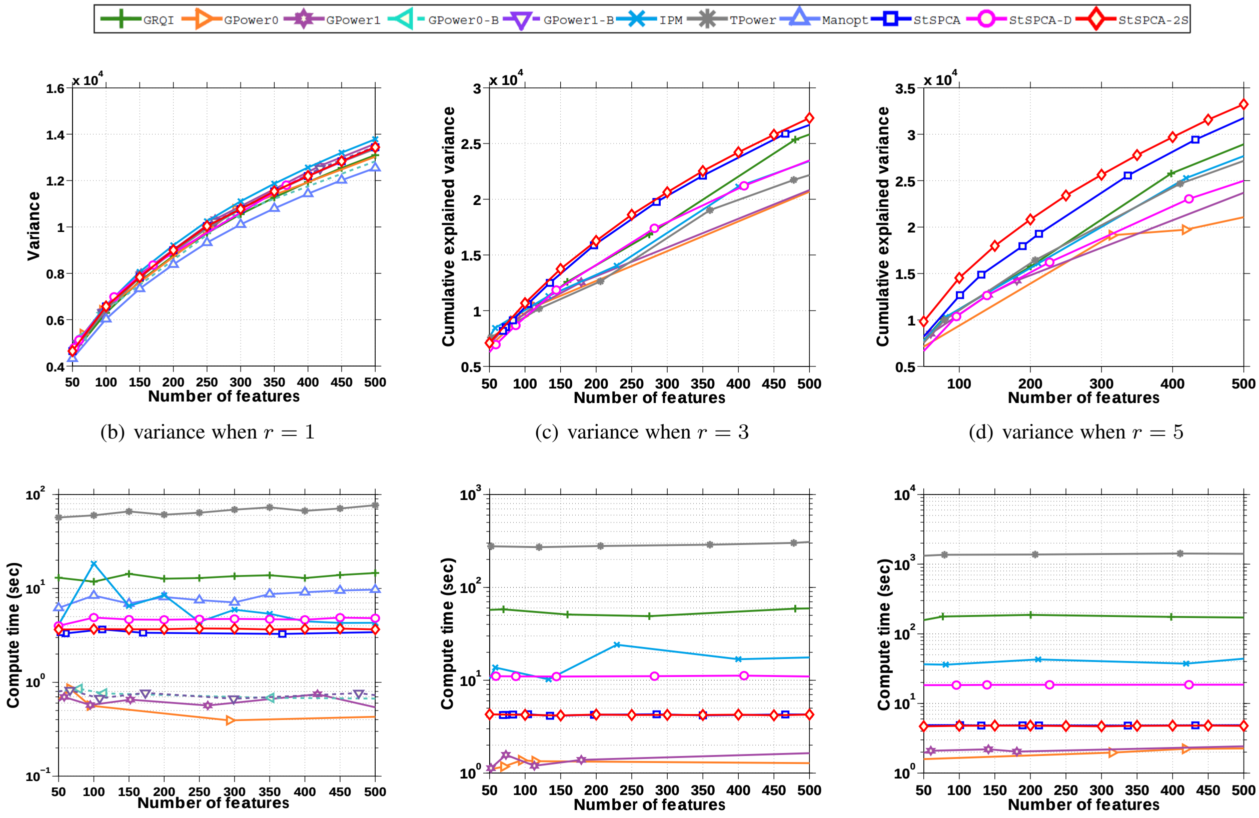
[31] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, 12(3):531–547, 2003.

[32] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research (JMLR)*, 11:517–553, 2010.

[33] Z. Kang, C. Peng, and Q. Cheng. Robust pca via nonconvex rank approximation. In *IEEE International Conference on Data Mining (ICDM)*, pages 211–220. IEEE, 2015.

[34] H. Kasai and B. Mishra. Low-rank tensor completion: a riemannian manifold preconditioning approach. In *International Conference on Machine Learning (ICML)*, pages 1012–1021, 2016.

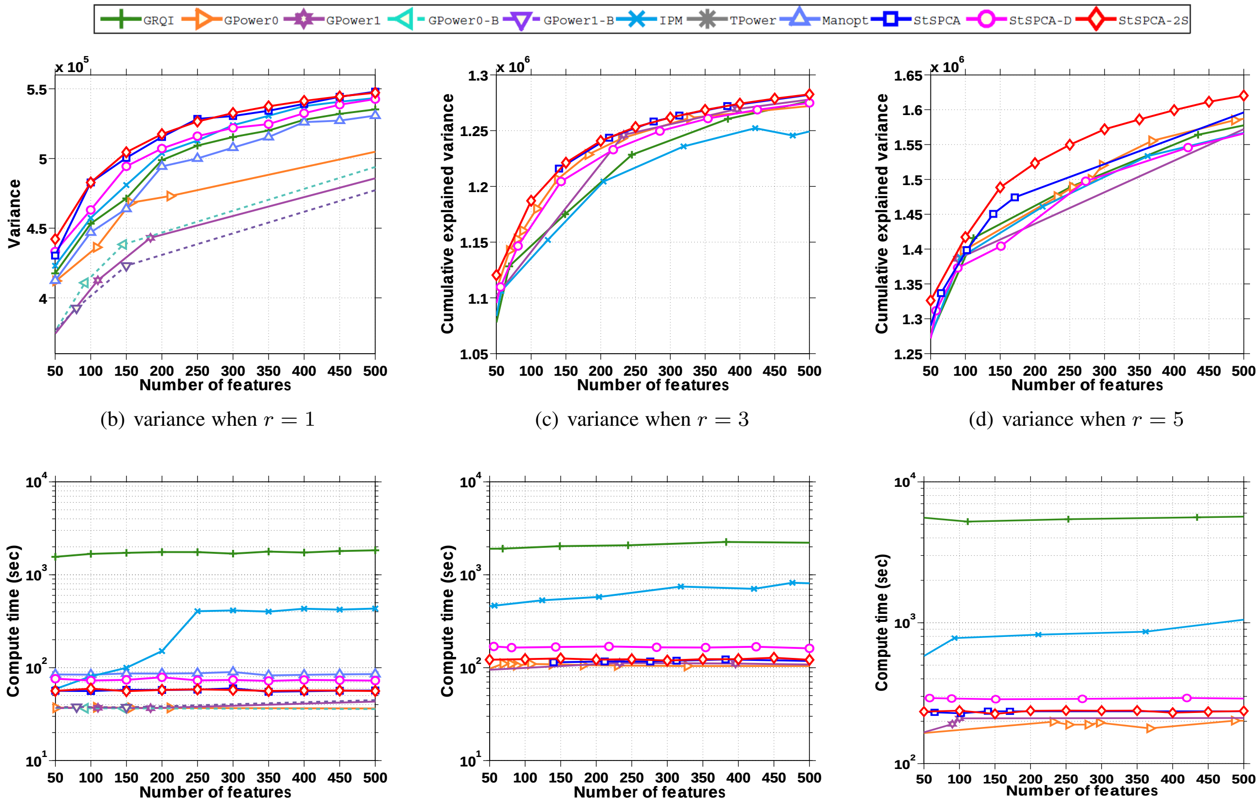
[35] J. Kovaevi, A. Chebira, et al. An introduction to frames. *Foundations and Trends® in Signal Processing*, 2(1):1–94, 2008.



(b) variance when $r = 1$ (c) variance when $r = 3$ (d) variance when $r = 5$

(e) running time when $r = 1$ (f) running time when $r = 3$ (g) running time when $r = 5$

Fig. 6. Results on the Breast dataset.



(b) variance when $r = 1$ (c) variance when $r = 3$ (d) variance when $r = 5$

(e) running time when $r = 1$ (f) running time when $r = 3$ (g) running time when $r = 5$

Fig. 7. Results on the News20 dataset.

[36] V. Kuleshov. Fast algorithms for sparse principal component analysis based on rayleigh quotient iteration. In *International Conference on Machine Learning (ICML)*, pages 1418–1425, 2013.

[37] H. Liu, R. Ji, J. Wang, and C. Shen. Ordinal constraint binary coding for approximate nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[38] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei. Face recognition via weighted sparse representation. *Journal of Visual Communication and Image Representation*, 24(2):111–116, 2013.

[39] T. Ma and A. Wigderson. Sum-of-squares lower bounds for sparse pca. In *Neural Information Processing Systems (NIPS)*, pages 1612–1620, 2015.

[40] Z. Ma et al. Sparse principal component analysis and iterative thresholding. *Annals of Statistics*, 41(2):772–801, 2013.

[41] L. W. Mackey. Deflation methods for sparse pca. In *Neural Information Processing Systems (NIPS)*, pages 1017–1024, 2009.

[42] J.-X. Mi, D. Lei, and J. Gui. A novel method for recognizing face with partial occlusion via sparse representation. *Optik*, 124(24):6786–6789, 2013.

[43] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse pca: Exact and greedy algorithms. In *Neural Information Processing Systems (NIPS)*, pages 915–922, 2005.

[44] N. Nori, D. Bollegala, and H. Kashima. Multinomial relation prediction in social data: A dimension reduction approach. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 12, pages 115–121, 2012.

[45] C. Peng, Z. Kang, and Q. Cheng. A fast factorization-based approach to robust pca. In *IEEE International Conference on Data Mining (ICDM)*, pages 1137–1142. IEEE, 2016.

[46] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, et al. Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS*, 98(26):15149–15154, 2001.

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

[48] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

[49] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.

[50] K. Sjöstrand, L. H. Clemmensen, R. Larsen, G. Einarsson, and B. K. Ersbøll. Spasm: A matlab toolbox for sparse statistical modeling. *Journal of Statistical Software*, 84(10), 2018.

[51] B. Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[52] V. Q. Vu, J. Cho, J. Lei, and K. Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. In *Neural Information Processing Systems (NIPS)*, pages 2670–2678, 2013.

[53] Z. Wang, H. Lu, and H. Liu. Tighten after relax: Minimax-optimal sparse pca in polynomial time. In *Neural Information Processing Systems (NIPS)*, pages 3383–3391, 2014.

[54] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS*, 98(20):11462–11467, 2001.

[55] K. Yang and C. Shahabi. A pca-based kernel for kernel pca on multivariate time series. In *Proceedings of ICDM 2005 workshop on temporal data mining: algorithms, theory and applications held in conjunction with the fifth IEEE international conference on data mining (ICDM05)*, pages 149–156, 2005.

[56] X.-T. Yuan and T. Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research (JMLR)*, 14(Apr):899–925, 2013.

[57] X.-T. Yuan and T. Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research (JMLR)*, 14(1):899–925, 2013.

[58] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[59] H. Zou and L. Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.

Mingkui Tan is currently a professor with the School of Software Engineering at South China University of Technology. He received his Bachelor Degree in Environmental Science and Engineering in 2006 and Master degree in Control Science and Engineering in 2009, both from Hunan University in Changsha, China. He received the Ph.D. degree in Computer Science from Nanyang Technological University, Singapore, in 2014. From 2014-2016, he worked as a Senior Research Associate on computer vision in the School of Computer Science, University of Adelaide, Australia. His research interests include machine learning, sparse analysis, deep learning and large-scale optimization.

Zhibin Hu is currently a PhD candidate in the School of Software Engineering, South China University of Technology, China. He received the B.S. degree in Computer Science from South China Normal University. His research interests include data mining, large-scale machine learning, and deep learning.

Yuguang Yan received the Ph.D. and B.S. degrees in software engineering from the South China University of Technology, China, in 2019 and 2013, respectively. His current research interests include transfer learning, multi-label classification, and online learning.

Jiezhang Cao is a master of the School of Software Engineering, South China University of Technology, China. He received the B.S. degree in statistics from the Guangdong University of Technology, China, 2017. His current research interests include machine learning, generative model.

Dong Gong is a postdoctoral researcher at The University of Adelaide. He received the Ph.D. and B.S. degrees in computer science from the Northwestern Polytechnical University, Xi'an, China, in 2018 and 2012, respectively. His current research interests include machine learning and optimization techniques and their applications in image processing and computer vision.

Qingyao Wu is currently a Professor with the School of Software Engineering, South China University of Technology, China. He received the B.S. degree in software engineering from the South China University of Technology, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 2007, 2009, and 2013, respectively. He was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore, from 2014 to 2015. His current research interests include machine learning, data mining, vision and language understanding.