Journal of Machine Learning Research 14 (2013) 899-925 Submitted 12/11; Revised 10/12; Published 4/13

Truncated Power Method for Sparse Eigenvalue Problems

Xiao-Tong Yuan XTYUAN1980@GMAIL.COM
Tong Zhang TZHANG@STAT.RUTGERS.EDU
Department of Statistics
Rutgers University
New Jersey, 08816, USA

Editor: Hui Zou

Abstract

This paper considers the sparse eigenvalue problem, which is to extract dominant (largest) sparse eigenvectors with at most k non-zero components. We propose a simple yet effective solution called *truncated power method* that can approximately solve the underlying nonconvex optimization problem. A strong sparse recovery result is proved for the truncated power method, and this theory is our key motivation for developing the new algorithm. The proposed method is tested on applications such as sparse principal component analysis and the densest k-subgraph problem. Extensive experiments on several synthetic and real-world data sets demonstrate the competitive empirical performance of our method.

Keywords: sparse eigenvalue, power method, sparse principal component analysis, densest k-subgraph

1. Introduction

Given a $p \times p$ symmetric positive semidefinite matrix A, the *largest k-sparse eigenvalue* problem aims to maximize the quadratic form $x^\top Ax$ with a sparse unit vector $x \in \mathbb{R}^p$ with no more than k non-zero elements:

$$\lambda_{\max}(A,k) = \max_{x\in\mathbb{R}^p} x^\top Ax, \qquad \text{subject to } \|x\| = 1, \quad \|x\|_0 \leq k, \tag{1}$$

where $\|\cdot\|$ denotes the ℓ_2-norm, and $\|\cdot\|_0$ denotes the ℓ_0-norm which counts the number of non-zero entries in a vector. The sparsity is controlled by the values of k and can be viewed as a design parameter. In machine learning applications, for example, principal component analysis, this problem is motivated from the following perturbation formulation of matrix A:

$$A = \bar{A} + E, \tag{2}$$

where A is the empirical covariance matrix, $\bar{A}$ is the true covariance matrix, and E is a random perturbation due to having only a finite number of empirical samples. If we assume that the largest eigenvector $\bar{x}$ of $\bar{A}$ is sparse, then a natural question is to recover $\bar{x}$ from the noisy observation A when the error E is “small”. In this context, the problem (1) is also referred to as sparse principal component analysis (sparse PCA).

In general, problem (1) is non-convex. In fact, it is also NP-hard because it can be reduced to the subset selection problem for ordinary least squares regression (Moghaddam et al., 2006), which is known to be NP-hard. Various researchers have proposed approximate optimization methods: some are based on greedy procedures (e.g., Moghaddam et al., 2006; Jolliffe et al., 2003; d'Aspremont et al., 2008), and some others are based on various types of convex relaxation or reformulation (e.g., d'Aspremont et al., 2007; Zou et al., 2006; Journée et al., 2010). Statistical analysis of sparse PCA has also received significant attention. Under the high dimensional single spike model, Johnstone (2001) proved the consistency of PCA using a subset of features corresponding to the largest sample variances. Under the same single spike model, Amini and Wainwright (2009) established conditions for recovering the non-zero entries of eigenvectors using the convex relaxation method of d'Aspremont et al. (2007). However, these results were concerned with variable selection consistency under a relatively simple and specific example with limited general applicability. More recently, Paul and Johnstone (2012) studied an extension called multiple spike model, and proposed an augmented sparse PCA method for estimating each of the leading eigenvectors and investigated the rate of convergence of their procedure in the high dimensional setting. In another recent work that is independent of ours, Ma (2013) analyzed an iterative thresholding method for recovering the sparse principal subspace. Although it also focused on the multiple spike covariance model, the procedures and techniques considered there are closely related to the method studied in this paper. In addition, Shen et al. (2013) analyzed the consistency of the sparse PCA method of Shen and Huang (2008), and Cai et al. (2012) analyzed the optimal convergence rate of sparse PCA and introduced an adaptive procedure for estimating the principal subspace.

This paper proposes and analyzes a computational procedure called *truncated power iteration method* that approximately solves (1). This method is similar to the classical power method, with an additional truncation operation to ensure sparsity. We show that if the true matrix $\bar{A}$ has a sparse (or approximately sparse) dominant eigenvector $\bar{x}$, then under appropriate assumptions, this algorithm can recover $\bar{x}$ when the spectral norm of sparse submatrices of the perturbation E is small. Moreover, this result can be proved under relative generality without restricting ourselves to the rather specific spike covariance model. Therefore our analysis provides strong theoretical support for this new method, and this differentiates our proposal from previous studies. We have applied the proposed method to sparse PCA and to the densest k-subgraph finding problem (with proper modification). Extensive experiments on synthetic and real-world large-scale data sets demonstrate both the competitive sparse recovering performance and the computational efficiency of our method.

It is worth mentioning that the truncated power method developed in this paper can also be applied to the *smallest k-sparse eigenvalue* problem given by:

$$\lambda_{\min}(A,k) = \min_{x\in\mathbb{R}^p} x^\top A x, \qquad \text{subject to } \|x\| = 1, \quad \|x\|_0 \le k,$$

which also has many applications in machine learning.

1.1 Notation

Let $\mathbb{S}^p = \{A \in \mathbb{R}^{p\times p} \mid A = A^\top\}$ denote the set of symmetric matrices, and $\mathbb{S}^p_+ = \{A \in \mathbb{S}^p, A \succeq 0\}$ denote the cone of symmetric, positive semidefinite matrices. For any $A \in \mathbb{S}^p$, we denote its eigenvalues by $\lambda_{\min}(A) = \lambda_p(A) \le \cdots \le \lambda_1(A) = \lambda_{\max}(A)$. We use $\rho(A)$ to denote the spectral norm of A, which is $\max\{|\lambda_{\min}(A)|, |\lambda_{\max}(A)|\}$, and define

$$\rho(A,s) := \max\{|\lambda_{\min}(A,s)|, |\lambda_{\max}(A,s)|\}. \tag{3}$$

The i-th entry of vector x is denoted by $[x]_i$ while $[A]_{ij}$ denotes the element on the i-th row and j-th column of matrix A. We denote by A_k any $k\times k$ principal submatrix of A and by A_F the principal submatrix of A with rows and columns indexed in set F. If necessary, we also denote A_F as the restriction of A on the rows and columns indexed in F. Let $\|x\|_p$ be the ℓ_p-norm of a vector x. In particular, $\|x\|_2 = \sqrt{x^\top x}$ denotes the Euclidean norm, $\|x\|_1 = \sum_{i=1}^{d} |[x]_i|$ denotes the ℓ_1-norm, and $\|x\|_0 = \#\{j : [x]_j \neq 0\}$ denotes the ℓ_0-norm. For simplicity, we also denote the ℓ_2 norm $\|x\|_2$ by $\|x\|$. In the rest of the paper, we define $Q(x) := x^\top Ax$. We let $\mathrm{supp}(x) := \{j : [x]_j \neq 0\}$ denote the support set of vector x. Given an index set F, we define

$$x(F) := \arg\max_{x \in \mathbb{R}^p} x^\top Ax, \qquad \text{subject to } \|x\| = 1, \quad \mathrm{supp}(x) \subseteq F.$$

Finally, we denote by $I_{p\times p}$ the $p \times p$ identity matrix.

1.2 Paper Organization

The remaining of this paper is organized as follows: §2 describes the truncated power iteration algorithm that approximately solves problem (1). In §3 we analyze the solution quality of the proposed algorithm. §4 evaluates the relevance of our theoretical prediction and the practical performance of the proposed algorithm in applications of sparse PCA and the densest k-subgraph finding problems. We conclude this work and discuss potential extensions in §5.

2. Truncated Power Method

Since $\lambda_{\max}(A,k)$ equals $\lambda_{\max}(A_k^*)$ where A_k^* is the $k \times k$ principal submatrix of A with the largest eigenvalue, one may solve (1) by exhaustively enumerating all subsets of $\{1,\ldots,p\}$ of size k in order to find A_k^*. However, this procedure is impractical even for moderate sized k since the number of subsets is exponential in k.

2.1 Algorithm

Therefore in order to solve the spare eigenvalue problem (1) more efficiently, we consider an iterative procedure based on the standard power method for eigenvalue problems, while maintaining the desired sparsity for the intermediate solutions. The procedure, presented in Algorithm 1, generates a sequence of intermediate k-sparse eigenvectors $x_0, x_1, \ldots$ from an initial sparse approximation x_0. At each time stamp t, the intermediate vector x_{t-1} is multiplied by A, and then the entries are truncated to zeros except for the largest k entries. The resulting vector is then normalized to unit length, which becomes x_t. The cardinality k is a free parameter in the algorithm. If no prior knowledge of sparsity is available, then we have to tune this parameter, for example, through cross-validation. Note that our theory does not require choosing k precisely (see Theorem 4), and thus the tuning is not difficult in practice. At each iteration, the computational complexity is in $O(kp + p)$ which is $O(kp)$ for matrix-vector product Ax_{t-1} and $O(p)$[1] for selecting k largest elements from the obtained vector of length p to get F_t.

Definition 1 *Given a vector x and an index set F, we define the truncation operation Truncate(x,F) to be the vector obtained by restricting x to F, that is*

$$[\mathit{Truncate}(x,F)]_i = \begin{cases} [x]_i & i \in F \\ 0 & \mathit{otherwise} \end{cases}.$$

[1] Our actual implementation employs sorting for simplicity, which has a slightly worse complexity of $O(p \ln p)$ instead of $O(p)$.

Algorithm 1: Truncated Power (TPower) Method

```
Input      : matrix $A \in \mathbb{S}^p$, initial vector $x_0 \in \mathbb{R}^p$
Output     : $x_t$
Parameters : cardinality $k \in \{1, ..., p\}$
Let $t = 1$.
repeat
    Compute $x'_t = Ax_{t-1}/\|Ax_{t-1}\|$.
    Let $F_t = \mathrm{supp}(x'_t, k)$ be the indices of $x'_t$ with the largest $k$ absolute values.
    Compute $\hat{x}_t = \mathrm{Truncate}(x'_t, F_t)$.
    Normalize $x_t = \hat{x}_t/\|\hat{x}_t\|$.
    $t \leftarrow t + 1$.
until Convergence;
```

Remark 2 *Similar to the behavior of traditional power method, if $A \in \mathbb{S}^p_+$, then TPower tries to find the (sparse) eigenvector of A corresponding to the largest eigenvalue. Otherwise, it may find the (sparse) eigenvector with the smallest eigenvalue if $-\lambda_p(A) > \lambda_1(A)$. However, this situation is easily detectable because it can only happen when $\lambda_p(A) < 0$. In such case, we may restart TPower with A replaced by an appropriately shifted version $A + \tilde{\lambda} I_{p\times p}$.*

2.2 Convergence

We now show that when A is positive semidefinite, TPower converges. This claim is a direct consequence of the following proposition.

Proposition 3 *If all $2k \times 2k$ principal submatrix A_{2k} of A are positive semidefinite, then the sequence $\{Q(x_t)\}_{t\geq 1}$ is monotonically increasing, where x_t is obtained from the TPower algorithm.*

Proof Observe that the iterate x_t in TPower solves the following constrained linear optimization problem:

$$x_t = \arg\max_{\|x\|=1, \|x\|_0 \leq k} L(x; x_{t-1}), \qquad L(x; x_{t-1}) := \langle 2Ax_{t-1}, x - x_{t-1}\rangle.$$

Clearly, $Q(x) - Q(x_{t-1}) = L(x; x_{t-1}) + (x - x_{t-1})^\top A(x - x_{t-1})$. Since $\|x_t - x_{t-1}\|_0 \leq 2k$ and each $2k \times 2k$ principal submatrix of A is positive semidefinite, we have $(x_t - x_{t-1})^\top A(x_t - x_{t-1}) \geq 0$. It follows that $Q(x_t) - Q(x_{t-1}) \geq L(x_t; x_{t-1})$. By the definition of x_t as the maximizer of $L(x; x_{t-1})$ over x (subject to $\|x\| = 1$ and $\|x\|_0 \leq k$), we have $L(x_t; x_{t-1}) \geq L(x_{t-1}; x_{t-1}) = 0$. Therefore $Q(x_t) - Q(x_{t-1}) \geq 0$, which proves the desired result. ■

3. Sparse Recovery Analysis

We consider the general noisy matrix model (2), and are especially interested in the high dimensional situation where the dimension p of A is large. We assume that the noise matrix E is a dense $p \times p$ matrix such that its sparse submatrices have small spectral norm $\rho(E, s)$ (see (3)) for s in the same order of k. We refer to this quantity as *restricted perturbation error*. However, the spectral norm of the full matrix perturbation error $\rho(E)$ can be large. For example, if the original covariance is corrupted by an additive standard Gaussian iid noise vector, then $\rho(E,s) = O(\sqrt{s \log p/n})$, which grows linearly in $\sqrt{s}$, instead of $\rho(E) = O(\sqrt{p/n})$, which grows linearly in $\sqrt{p}$. The main advantage of the sparse eigenvalue formulation (1) over the standard eigenvalue formulation is that the estimation error of its optimal solution depends on $\rho(E,s)$ with respectively a small $s = O(k)$ rather than $\rho(E)$. This linear dependency on sparsity k instead of the original dimension p is analogous to similar results for sparse regression (or compressive sensing). In fact, the restricted perturbation error considered here is analogous to the idea of restricted isometry property (RIP) considered by Candes and Tao (2005).

The purpose of the section is to show that if matrix $\bar{A}$ has a unique sparse (or approximately sparse) dominant eigenvector, then under suitable conditions, TPower can (approximately) recover this eigenvector from the noisy observation A.

Assumption 1 *Assume that the largest eigenvalue of $\bar{A} \in \mathbb{S}^p$ is $\lambda = \lambda_{\max}(\bar{A}) > 0$ that is non-degenerate, with a gap $\Delta\lambda = \lambda - \max_{j>1} |\lambda_j(\bar{A})|$ between the largest and the remaining eigenvalues. Moreover, assume that the eigenvector $\bar{x}$ corresponding to the dominant eigenvalue λ is sparse with cardinality $\bar{k} = \|\bar{x}\|_0$.*

We want to show that under Assumption 1, if the spectral norm $\rho(E,s)$ of the error matrix is small for an appropriately chosen $s > \bar{k}$, then it is possible to approximately recover $\bar{x}$. Note that in the extreme case of $s = p$, this result follows directly from the standard eigenvalue perturbation analysis (which does not require Assumption 1).

We now state our main result as below, which shows that under appropriate conditions, the TPower method can recover the sparse eigenvector. The final error bound is a direct generalization of standard matrix perturbation result that depends on the full matrix perturbation error $\rho(E)$. Here this quantity is replaced by the restricted perturbation error $\rho(E,s)$.

Theorem 4 *We assume that Assumption 1 holds. Let $s = 2k + \bar{k}$ with $k \geq \bar{k}$. Assume that $\rho(E,s) \leq \Delta\lambda/2$. Define*

$$\gamma(s) := \frac{\lambda - \Delta\lambda + \rho(E,s)}{\lambda - \rho(E,s)} < 1, \qquad \delta(s) := \frac{\sqrt{2}\rho(E,s)}{\sqrt{\rho(E,s)^2 + (\Delta\lambda - 2\rho(E,s))^2}}.$$

If $|x_0^\top \bar{x}| \geq \theta + \delta(s)$ for some $\|x_0\|_0 \leq k$, $\|x_0\| = 1$, and $\theta \in (0,1)$ such that

$$\mu = \sqrt{(1 + 2((\bar{k}/k)^{1/2} + \bar{k}/k))(1 - 0.5\theta(1+\theta)(1-\gamma(s)^2))} < 1, \tag{4}$$

then we either have

$$\sqrt{1 - |x_0^\top \bar{x}|} < \sqrt{10}\delta(s)/(1-\mu), \tag{5}$$

or for all $t \geq 0$

$$\sqrt{1 - |x_t^\top \bar{x}|} \leq \mu^t \sqrt{1 - |x_0^\top \bar{x}|} + \sqrt{10}\delta(s)/(1-\mu). \tag{6}$$

Remark 5 *We only state our result with a relatively simple but easy to understand quantity $\rho(E,s)$, which we refer to as restricted perturbation error. It is analogous to the RIP concept (Candes and Tao, 2005), and is also directly comparable to the traditional full matrix perturbation error $\rho(E)$. While it is possible to obtain sharper results with additional quantities, we intentionally keep the theorem simple so that its consequence is relatively easy to interpret.*

Remark 6 *Although we state the result by assuming that the dominant eigenvector $\bar{x}$ is sparse, the theorem can also be adapted to certain situations that $\bar{x}$ is only approximately sparse. In such case, we simply let $\bar{x}'$ be a $\bar{k}$ sparse approximation of $\bar{x}$. If $\bar{x}' - \bar{x}$ is sufficiently small, then $\bar{x}'$ is the dominant eigenvector of a symmetric matrix $\bar{A}'$ that is close to $\bar{A}$; hence the theorem can be applied with the decomposition $A = \bar{A}' + E'$ where $E' = E + A - \bar{A}'$.*

Note that we did not make any attempt to optimize the constants in Theorem 4, which are relatively large. Therefore in the discussion, we shall ignore the constants, and focus on the main message Theorem 4 conveys. If $\rho(E,s)$ is smaller than the eigen-gap $\Delta\lambda/2 > 0$, then $\gamma(s) < 1$ and $\delta(s) = O(\rho(E,s))$. It is easy to check that for any $k \geq \bar{k}$, if $\gamma(s)$ is sufficiently small then the requirement (4) can be satisfied for a sufficiently small θ of the order $(\bar{k}/k)^{1/2}$. It follows that under appropriate conditions, as long as we can find an initial x_0 such that

$$|x_0^\top \bar{x}| \geq c(\rho(E,s) + (\bar{k}/k)^{1/2})$$

for some constant c, then $1 - |x_t^\top \bar{x}|$ converges geometrically until

$$\|x_t - \bar{x}\| = O(\rho(E,s)).$$

This result is similar to the standard eigenvector perturbation result stated in Lemma 10 of Appendix A, except that we replace the spectral error $\rho(E)$ of the full matrix by $\rho(E,s)$ that can be significantly smaller when $s \ll p$. To our knowledge, this is the first sparse recovery result for the sparse eigenvalue problem in a relatively general setting. This theorem can be considered as a strong theoretical justification of the proposed TPower algorithm that distinguishes it from earlier algorithms without theoretical guarantees. Specifically, the replacement of the full matrix perturbation error $\rho(E)$ with $\rho(E,s)$ gives the theoretical insights on why TPower works well in practice.

To illustrate our result, we briefly describe a consequence of the theorem under the single spike covariance model of Johnstone (2001) which was investigated by Amini and Wainwright (2009). We assume that the observations are p dimensional vectors

$$x_i = \bar{x} + \varepsilon,$$

for $i = 1, \ldots, n$, where $\varepsilon \sim N(0, I_{p\times p})$. For simplicity, we assume that $\|\bar{x}\| = 1$. The true covariance is

$$\bar{A} = \bar{x}\bar{x}^\top + I_{p\times p},$$

and A is the empirical covariance

$$A = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top.$$

Let $E = A - \bar{A}$, then random matrix theory implies that with large probability,

$$\rho(E,s) = O(\sqrt{s \ln p/n}).$$

Now assume that $\max_j |\bar{x}_j|$ is sufficiently large. In this case, we can run TPower with a starting point $x_0 = e_j$ for some vector e_j (where e_j is the vector of zeros except the j-th entry being one) so that $|e_j^\top \bar{x}| = |\bar{x}_j|$ is sufficiently large, and the assumption for the initial vector $|x_0^\top \bar{x}| \geq c(\rho(E,s) +$

$(\bar{k}/k)^{1/2})$ is satisfied with $s = O(\bar{k})$. We may run TPower with an appropriate initial vector to obtain an approximate solution x_t of error

$$\|x_t - \bar{x}\| = O(\sqrt{\bar{k}\ln p/n}).$$

This bound is optimal (Cai et al., 2012). Note that our results are not directly comparable to those of Amini and Wainwright (2009), which studied support recovery. Nevertheless, it is worth noting that if $\max_j |\bar{x}_j|$ is sufficiently large, then our result becomes meaningful when $n = O(\bar{k}\ln p)$; however their result requires $n = O(\bar{k}^2 \ln p)$ to be meaningful, although this is for the pessimistic case of $\bar{x}$ having equal nonzero values of $1/\sqrt{\bar{k}}$. Based on a similar spike covariance model, Ma (2013) independently presented and analyzed an iterative thresholding method for recovering sparse orthogonal principal components, using ideas related to what we present in this paper.

Finally we note that if we cannot find an initial vector with large enough value $|x_0^\top \bar{x}|$, then it may be necessary to take a relatively large k so that the requirement $|x_0^\top \bar{x}| \geq c(\rho(E,s) + (\bar{k}/k)^{1/2})$ is satisfied. With such a k, $\rho(E,s)$ may be relatively large and hence the theorem indicates that x_t may not converge to $\bar{x}$ accurately. Nevertheless, as long as $|x_t^\top \bar{x}|$ converges to a value that is not too small (e.g., can be much larger than $|x_0^\top \bar{x}|$), we may reduce k and rerun the algorithm with a k-sparse truncation of x_t as initial vector together with the reduced k. In this two stage process, the vector found from the first stage (with large k) is truncated and normalized, and then used as the initial value of the second stage (with small k). Therefore we may also regard it as an initialization method for TPower. Specially, in the first stage we may run TPower with $k = p$ from arbitrary initialization. In this stage, TPower reduces to the classic power method which outputs the dominant eigenvector x of A. Let $F = \mathrm{supp}(x,k)$ be the indices of x with the largest k absolute values and $x_0 := \mathrm{Truncate}(x,F)/\|\mathrm{Truncate}(x,F)\|$. Let $\theta = x^\top \bar{x} - (\bar{k}/k)^{1/2}\sqrt{1-(x^\top \bar{x})^2} - \delta(s)$. It is implied by Lemma 12 in Appendix A that $x_0^\top \bar{x} \geq \theta + \delta(s)$. Obviously, if $\theta(1+\theta) \geq 8(\bar{k}/k)/((1+4\bar{k}/k)(1-\gamma(s)^2))$, then x_0 will be an initialization suitable for Theorem 1. From this initialization, we can obtain a better solution using the TPower method. In practice, one may use other methods to obtain an approximate x_0 to initialize TPower, not necessarily restricted to running TPower with larger k.

4. Experiments

In this section, we first show numerical results (in §4.1) that confirm the relevance of our theoretical predictions. We then illustrate the effectiveness of TPower method when applied to sparse principal component analysis (sparse PCA) (in §4.2) and the densest k-subgraph (DkS) finding problem (in §4.3). The Matlab code for reproducing the experimental results reported in this section is available from `https://sites.google.com/site/xtyuan1980/publications`.

4.1 Simulation Study

In this experiment, we illustrate the performance of TPower using simulated data. Theorem 4 implies that under appropriate conditions, the estimation error $\sqrt{1 - x_t^\top \bar{x}}$ is proportional to $\delta(s)$. By definition, $\delta(s)$ is an increasing function with respect to perturbation error $\rho(E,s)$ and a decreasing function with respect to the gap $\Delta\lambda$ between the largest eigenvalue and the remaining eigenvalues. We will verify the results of Theorem 4 by applying TPower to the following single spike model with true covariance

$$\bar{A} = \beta \bar{x}\bar{x}^\top + I_{p \times p}$$

and empirical covariance

$$A = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top,$$

where $x_i \sim \mathcal{N}(0, \bar{A})$. For the true covariance matrix $\bar{A}$, its dominant eigenvector is $\bar{x}$ with eigenvalue $\beta + 1$, and its eigenvalue gap is $\Delta\lambda = \beta$. For this model, with large probability we have $\rho(E, s) = O(\sqrt{s \ln p/n})$. Therefore, for fixed dimensionality p, the error bound is relevant to the triplet $\{n, \beta, k\}$. In this study, we consider a setup with $p = 1000$, and $\bar{x}$ is a $\bar{k}$-sparse uniform random vector with $\bar{k} = 20$ and $\|\bar{x}\| = 1$. We are interested in the following two cases:

1. Cardinality k is tuned and fixed: we will study how the estimation error is affected by sample size n and eigen-gap β.
2. Cardinality k is varying: for fixed sample size n and eigen-gap β, we will study how the estimation error is affected by cardinality k in the algorithm.

4.1.1 ON INITIALIZATION

Theorem 4 suggests that TPower can benefit from a good initial vector x_0. We initialize x_0 by using the warm-start strategy suggested at the end of §3. In our implementation, this strategy is specialized as follows: we sequentially run TPower with cardinality $\{8k, 4k, 2k, k\}$, using the (truncated) output from the previous running as the initial vector for the next running. This initialization strategy works satisfactory in our numerical experiments.

4.1.2 TEST I: CARDINALITY k IS TUNED AND FIXED

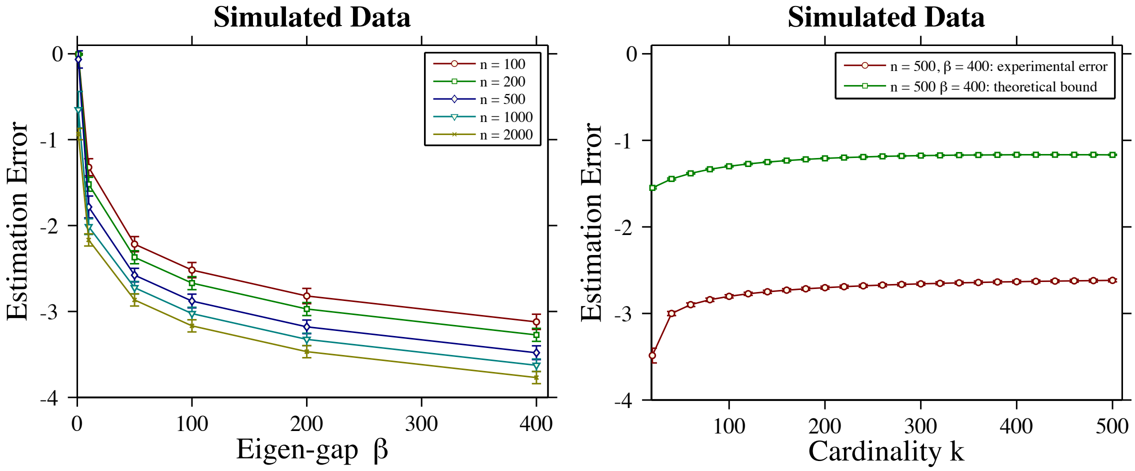
In this case, we test with $n \in \{100, 200, 500, 1000, 2000\}$ and $\beta \in \{1, 10, 50, 100, 200, 400\}$. For each pair $\{n, \beta\}$, we generate 100 empirical covariance matrices and employ the TPower to compute a k-sparse eigenvector $\hat{x}$. For each empirical covariance matrix A, we also generate an independent empirical covariance matrix A_{val} to select k from the candidate set $\mathcal{K} = \{5, 10, 15, ..., 50\}$ by maximizing the following criterion:

$$\hat{k} = \arg\max_{k \in \mathcal{K}} \hat{x}(k)^\top A_{val} \hat{x}(k),$$

where $\hat{x}(k)$ is the output of TPower for A under cardinality k. For different pairs (n, β), the tuned values of k could be different. For example, for $(n, \beta) = (100, 1)$, $k = 10$ will be selected; while for $(n, \beta) = (100, 10)$, $k = 20$ will be selected. Note that Theorem 4 does not require an accurate estimation of k. Figure 1(a) shows the estimation error curves as functions of β under various n. It can be observed that for any fixed n, the estimation error decreases as eigen-gap β increases; and for any fixed β, the estimation error decreases as sample size n increases. This is consistent with the prediction of Theorem 4.

4.1.3 TEST II: CARDINALITY k IS VARYING

In this case, we fix sample size $n = 500$ and eigen-gap $\beta = 400$, and test the values of $k \in \{20..., 500\}$ that are at least as large as the true sparsity $\bar{k} = 20$. We generate 100 empirical covariance matrices and employ the TPower to compute a k-sparse eigenvector. Figure 1(b) shows the estimation error



(a) Estimation error *vs.* eigen-gap β (under various sample size *n*).

(b) Estimation error bound *vs.* cardinality *k* (with $n = 500$ and $\beta = 400$). Both theoretical and empirical curves are plotted.

Figure 1: Estimation error curves on the simulated data. For better viewing, please see the original pdf file.

curves as functions of k. It can be observed that the estimation error becomes larger as k increases. This is consistent with the prediction of Theorem 4. For a fixed k, provided that the conditions are satisfied in Theorem 4, we can also calculate the theoretical estimation error bound $\sqrt{10}\delta(s)/(1-\mu)$. The curve of the theoretical bound is plotted in the same figure. As predicted by Theorem 4, the theoretical bound curve dominates the empirical error curve. Similar observations are also made for other fixed pairs $\{n, \beta\}$.

4.2 Sparse PCA

Principal component analysis (PCA) is a well established tool for dimensionality reduction and has a wide range of applications in science and engineering where high dimensional data sets are encountered. Sparse principal component analysis (sparse PCA) is an extension of PCA that aims at finding sparse vectors (loading vectors) capturing the maximum amount of variance in the data. In recent years, various researchers have proposed various approaches to directly address the conflicting goals of explaining variance and achieving sparsity in sparse PCA. For instance, greedy search and branch-and-bound methods were investigated by Moghaddam et al. (2006) to solve small instances of sparse PCA exactly and to obtain approximate solutions for larger scale problems. d'Aspremont et al. (2008) proposed the use of greedy forward selection with a certificate of optimality. Another popular technique for sparse PCA is regularized sparse learning. Zou et al. (2006) formulated sparse PCA as a regression-type optimization problem and imposed the Lasso penalty (Tibshirani, 1996) on the regression coefficients. The DSPCA algorithm of d'Aspremont et al. (2007) is an ℓ_1-norm based semidefinite relaxation for sparse PCA. Shen and Huang (2008) resorted to the singular value decomposition (SVD) to compute low-rank matrix approximations of the data matrix under various sparsity-inducing penalties. Mairal et al. (2010) proposed an online learning method for matrix decomposition with sparsity regularization. More recently, Journée et al. (2010) studied a generalized power method to solve sparse PCA with a certain dual reformulation of the problem. Similar power-truncation-type methods were also considered by Witten et al. (2009) and Ma (2013).

Given a sample covariance matrix, $\Sigma \in \mathbb{S}_+^p$ (or equivalently a centered data matrix $D \in \mathbb{R}^{n\times p}$ with n rows of p-dimensional observations vectors such that $\Sigma = D^\top D$) and the target cardinality k, following the literature (Moghaddam et al., 2006; d'Aspremont et al., 2007, 2008), we formulate sparse PCA as:

$$\hat{x} = \arg\max_{x\in\mathbb{R}^p} x^\top \Sigma x, \qquad \text{subject to } \|x\| = 1, \|x\|_0 \leq k. \tag{7}$$

The TPower method proposed in this paper can be directly applied to solve the above problem. One advantage of TPower for Sparse PCA is that it directly addresses the constraint on cardinality k. To find the top m rather than the top one sparse loading vectors, a common approach in the literature (d'Aspremont et al., 2007; Moghaddam et al., 2006; Mackey, 2008) is to use the *iterative deflation* method for PCA: subsequent sparse loading vectors can be obtained by recursively removing the contribution of the previously found loading vectors from the covariance matrix. Here we employ a projection deflation scheme proposed by Mackey (2008), which deflates a vector $\hat{x}$ using the formula:

$$\Sigma' = (I_{p\times p} - \hat{x}\hat{x}^\top)\Sigma(I_{p\times p} - \hat{x}\hat{x}^\top).$$

Obviously, Σ' remains positive semidefinite. Moreover, Σ' is rendered left and right orthogonal to $\hat{x}$.

4.2.1 Connection With Existing Sparse PCA Methods

In the setup of sparse PCA, TPower is closely related to GPower (Journée et al., 2010) and sPCA-rSVD (Shen and Huang, 2008) which share the same spirit of thresholding iteration to make the loading vectors sparse. Indeed, GPower and sPCA-rSVD are identical except for the initialization and post-processing phases (see, e.g., Journée et al., 2010). TPower is most closely related to the GPower$_{\ell_0}$ (Journée et al., 2010, Algorithm 3) in the sense that both are characterized by rank-1 approximation and alternate optimization with hard-thresholding. Indeed, given a data matrix $D \in \mathbb{R}^{n\times p}$, GPower$_{\ell_0}$ solves the following ℓ_0-norm regularized rank-1 approximation problem:

$$\min_{x\in\mathbb{R}^p, z\in\mathbb{R}^n} \|D - zx^\top\|_F^2 + \gamma\|x\|_0, \qquad \text{subject to } \|z\| = 1.$$

GPower$_{\ell_0}$ is essentially a coordinate descent procedure which iterates between updating x and z. Given x_{t-1}, the update of z_t is $z_t = Dx_{t-1}/\|Dx_{t-1}\|$. Given z_t, the update of x_t is a hard-thresholding operation which selects those entries in $D^\top z_t = D^\top Dx_{t-1}/\|Dx_{t-1}\|$ with squared values greater than γ and then normalize the vector after truncation. From the viewpoint of rank-1 approximation, it can be shown that TPower optimizes the following cardinality constrained problem:

$$\min_{x\in\mathbb{R}^p, z\in\mathbb{R}^n} \|D - zx^\top\|_F^2, \qquad \text{subject to } \|z\| = 1,\ \|x\| = 1,\ \|x\|_0 \leq k.$$

Indeed, based on the fact that $z = Dx/\|Dx\|$ is optimal at any x, the above problem is identical to the formulation (7). To update x_t, TPower selects the top k entries of $D^\top Dx_{t-1}$ and then normalize the truncated vector. Therefore, we can see that TPower and GPower$_{\ell_0}$ differs in the thresholding manner: the former selects the top k entries in $D^\top Dx_{t-1}$ while the latter preserves those entries in

$D^\top Dx_{t-1}$ with squared values greater than $\gamma\|Dx_{t-1}\|^2$. Another rank-1 approximation formulation was considered by Witten et al. (2009) with ℓ_1-norm ball constraint:

$$\min_{x\in\mathbb{R}^p, z\in\mathbb{R}^n} \|D - zx^\top\|_F^2, \qquad \text{subject to } \|z\| = 1,\ \|x\| = 1,\ \|x\|_1 \le c.$$

Its minimization procedure, called Projected Matrix Decomposition (PMD), alternates between the update of x and the update of z; where the update of x is a soft-thresholding operation.

Our method is also related to the Iterative Thresholding Sparse PCA (ITSPCA) method (Ma, 2013) which concentrates on recovering a sparse subspace of dimension m under the spike model. In particular, when $m = 1$, ITSPCA reduces to a power method with thresholding. However, TPower differs from ITSPCA in the following two aspects. First, the truncation strategy is different: we truncate the vector by preserving the top k largest absolute entries and setting the remaining entries to zeros, while ITSPCA truncates the vector by setting entries below a fixed threshold to zeros. Second, the analysis is different: TPower is analyzed under the matrix perturbation theory and thus is deterministic, while the analysis of ITSPCA focused on the convergence rate under the stochastic multiple spike model.

TPower is essentially a greedy selection method for solving problem (1). In this viewpoint, it is related to PathSPCA (d'Aspremont et al., 2008) which is a forward greedy selection procedure. PathSPCA starts from the empty set and at each iteration it selects the most relevant variable and adds it to the current variable set; it then re-estimates the leading eigenvector on the augmented variable set. Both TPower and PathSPCA output sparse solutions with exact cardinality k.

4.2.2 RESULTS ON TOY DATA SET

To illustrate the sparse recovering performance of TPower, we apply the algorithm to a synthetic data set drawn from a sparse PCA model. We follow the same procedure proposed by Shen and Huang (2008) to generate random data with a covariance matrix having sparse eigenvectors. To this end, a covariance matrix is first synthesized through the eigenvalue decomposition $\Sigma = VDV^\top$, where the first m columns of $V \in \mathbb{R}^{p\times p}$ are pre-specified sparse orthogonal unit vectors. A data matrix $X \in \mathbb{R}^{n\times p}$ is then generated by drawing n samples from a zero-mean normal distribution with covariance matrix Σ, that is $X \sim \mathcal{N}(0, \Sigma)$. The empirical covariance $\hat{\Sigma}$ matrix is then estimated from data X as the input for TPower.

Consider a setup with $p = 500$, $n = 50$, and the first $m = 2$ dominant eigenvectors of Σ are sparse. Here the first two dominant eigenvectors are specified as follows:

$$[v_1]_i = \begin{cases} \frac{1}{\sqrt{10}}, & i = 1, \ldots, 10 \\ 0, & \text{otherwise} \end{cases}, \quad [v_2]_i = \begin{cases} \frac{1}{\sqrt{10}}, & i = 11, \ldots, 20 \\ 0, & \text{otherwise} \end{cases}.$$

The remaining eigenvectors v_j for $j \ge 3$ are chosen arbitrarily, and the eigenvalues are fixed at the following values:

$$\begin{cases} \lambda_1 = 400, \\ \lambda_2 = 300, \\ \lambda_j = 1, \quad j = 3, \ldots, 500. \end{cases}$$

We generate 500 data matrices and employ the TPower method to compute two unit-norm sparse loading vectors $u_1, u_2 \in \mathbb{R}^{500}$, which are hopefully close to v_1 and v_2. Our method is compared on this data set with a greedy algorithm PathPCA (d'Aspremont et al., 2008), two power-iteration-type methods GPower (Journée et al., 2010) and PMD (Witten et al., 2009), two sparse regression based methods SPCA (Zou et al., 2006) and online SPCA (oSPCA) (Mairal et al., 2010), and the standard PCA. For GPower, we test its two block versions GPower$_{\ell_1,m}$ and GPower$_{\ell_0,m}$ with ℓ_1-norm and ℓ_0-norm penalties, respectively. Here we do not directly compare to two representative sparse PCA algorithms sPCA-rSVD (Shen and Huang, 2008) and DSPCA (d'Aspremont et al., 2007) because the former is shown to be identical to GPower up to initialization and post-processing phases (Journée et al., 2010), while the latter is suggested by the authors as a secondary choice after PathSPCA. All tested algorithms were implemented in Matlab 7.12 running on a desktop. We use the two-stage warm-start strategy for initialization. Similar to the empirical study in the previous section, we tune the cardinality parameter k on independently generated validation matrices.

In this experiment, we regard the true model to be successfully recovered when both quantities $|v_1^\top u_1|$ and $|v_2^\top u_2|$ are greater than 0.99. Table 1 lists the recovering results by the considered methods. It can be observed that TPower, PathPCA, GPower, PMD and oSPCA all successfully recover the ground truth sparse PC vectors with high rate of success. SPCA frequently fails to recover the spares loadings on this data set. The potential reason is that SPCA is initialized with the ordinary principal components which in many random data matrices are far away from the truth sparse solution. Traditional PCA always fails to recover the sparse PC loadings on this data set. The success of TPower and the failure of traditional PCA can be well explained by our sparse recovery result in Theorem 4 (for TPower) in comparison to the traditional eigenvector perturbation theory in Lemma 10 (for traditional PCA), which we have already discussed in §3. However, the success of other methods suggests that it might be possible to prove sparse recovery results similar to Theorem 4 for some of these alternative algorithms. The running time of these algorithms on this data is listed in the last column of Table 1. It can be seen that TPower is among the top efficient solvers.

Algorithms	Parameter	$\lvert v_1^\top u_1\rvert$	$\lvert v_2^\top u_2\rvert$	Prob. of succ.	CPU (in ms)
TPower	$k = 10$	.9998 (.0001)	.9997 (.0002)	1	6.14 (0.76)
PathSPCA	$k = 10$	.9998 (.0001)	.9997 (.0002)	1	77.42 (2.95)
GPower$_{\ell_1,m}$	$\gamma = 0.8$	.9997 (.0016)	.9996 (.0022)	0.99	6.22 (0.30)
GPower$_{\ell_0,m}$	$\gamma = 0.8$	.9997 (.0016)	.9991 (.0117)	0.99	6.07 (0.30)
PMD	$c = 3.0$	.9998 (.0001)	.9997 (.0002)	1	11.97 (0.48)
oSPCA	$\lambda = 3$	.9929 (.0434)	.9923 (.0483)	0.97	24.74 (1.20)
SPCA	$\lambda_1 = 10^{-3}$	.9274 (.0809)	.9250 (.0810)	0.25	799.99 (50.62)
PCA	–	.9146 (.0801)	.9086 (.0790)	0	3.87 (1.59)

Table 1: The quantitative results on a synthetic data set. The values $|v_1^\top u_1|$, $|v_2^\top u_2|$, CPU time (in ms) are in format of mean (std) over 500 running.

4.2.3 RESULTS ON PITPROPS DATA

The PitProps data set (Jeffers, 1967), which consists of 180 observations with 13 measured variables, has been a standard benchmark to evaluate algorithms for sparse PCA (see, e.g., Zou et al., 2006; Shen and Huang, 2008; Journée et al., 2010). Following these previous studies, we also consider to compute the first six sparse PCs of the data. In Table 2, we list the total cardinality and the proportion of adjusted variance (Zou et al., 2006) explained by six components computed with TPower, PathSPCA (d'Aspremont et al., 2008), GPower, PMD, oSPCA and SPCA. From these results we can see that on this relatively simple data set, TPower, PathSPCA and GPower perform quite similarly and are slightly better than PMD, oSPCA and SPCA.

Table 3 lists the six extracted PCs by TPower with cardinality setting 6-2-1-2-1-1. We can see that the important variables associated with the six PCs are exclusive except for the variable "ringb" which is simultaneously selected by PC1 and PC4. The variable "diaknot" is excluded from all the six PCs. The same loadings are also extracted by both PathSPCA and GPower under the parameters listed in Table 2.

Method	Parameters	Total cardinality	Prop. of explained variance
TPower	cardinalities: 7-2-4-3-5-4	25	0.8887
TPower	cardinalities: 6-2-1-2-1-1	13	0.7978
PathSPCA	cardinalities: 7-2-4-3-5-4	25	0.8834
PathSPCA	cardinalities: 6-2-1-2-1-1	13	0.7978
GPower$_{\ell_1,m}$	$\gamma = 0.22$	26	0.8438
GPower$_{\ell_1,m}$	$\gamma = 0.50$	13	0.7978
PMD	$c = 1.50$	25	0.8244
PMD	$c = 1.10$	13	0.7309
oSPCA	$\lambda = 0.2$	27	0.8351
oSPCA	$\lambda = 0.4$	12	0.6625
SPCA	see Zou et al. (2006)	18	0.7580

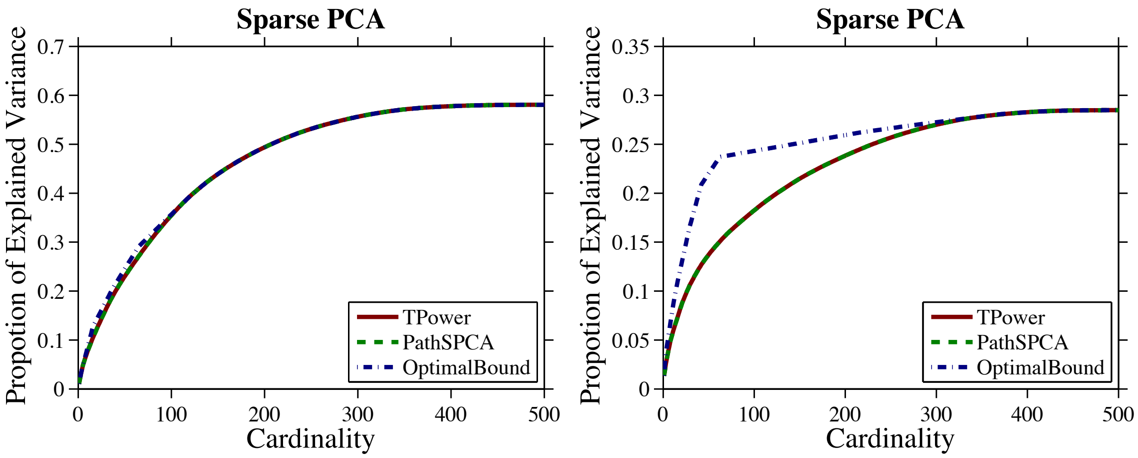
Table 2: The quantitative results on the PitProps data set. The result of SPCA is taken from Zou et al. (2006).

PCs	x_1 topd	x_2 length	x_3 moist	x_4 testsg	x_5 ovensg	x_6 ringt	x_7 ringb	x_8 bowm	x_9 bowd	x_{10} whorls	x_{11} clear	x_{12} knots	x_{13} diaknot
PC1	.4444	.4534	0	0	0	0	.3779	.3415	.4032	.4183	0	0	0
PC2	0	0	.7071	.7071	0	0	0	0	0	0	0	0	0
PC3	0	0	0	0	1.000	0	0	0	0	0	0	0	0
PC4	0	0	0	0	0	.8569	.5154	0	0	0	0	0	0
PC5	0	0	0	0	0	0	0		0	0	1.000	0	0
PC6	0	0	0	0	0	0	0	0	0	0	0	1.000	0

Table 3: The extracted six PCs by TPower on PitProps data set with cardinality setting 6-2-1-2-1-1. Note that in this setting, the extracted significant loadings are non-overlapping except for "ringb". And the variable "diaknot" is excluded from all the six PCs.

4.2.4 RESULTS ON BIOLOGICAL DATA

We have also evaluated the performance of TPower on two gene expression data sets, one is the Colon cancer data from Alon et al. (1999), the other is the Lymphoma data from Alizadeh et al. (2000). Following the experimental setup of d'Aspremont et al. (2008), we consider the 500 genes with the largest variances. We plot the variance versus cardinality tradeoff curves in Figure 2, to-



(a) Colon Cancer

(b) Lymphoma

Figure 2: The variance versus cardinality tradeoff curves on two gene expression data sets. For better viewing, please see the original pdf file.

gether with the result from PathSPCA and the upper bounds of optimal values from d'Aspremont et al. (2008). Note that our method performs almost identical to the PathSPCA which is demonstrated to have optimal or very close to optimal solutions in many cardinalities. The computational time of the two methods on both data sets is comparable and is less than two seconds.

4.2.5 SUMMARY

To summarize this group of experiments on sparse PCA, the basic finding is that TPower performs quite competitively in terms of the tradeoff between explained variance and representation sparsity. The performance is comparable or superior to leading methods such as PathSPCA and GPower. It is observed that TPower, PathSPCA and GPower outperform PMD, oSPCA and SPCA on the benchmark data Pitprops. It is not surprising that TPower and GPower behave similarly because both are power-truncation-type method (see the previous §4.2.1). While strong theoretical guarantee can be established for the TPower method, it remains open to show that PathSPCA and GPower have a similar sparse recovery performance.

4.3 Densest k-Subgraph Finding

As another concrete application, we show that with proper modification, TPower can be applied to the densest k-subgraph finding problem. Given an undirected graph $G = (V, E)$, $|V| = n$, and integer $1 \leq k \leq n$, the densest k-subgraph (DkS) problem is to find a set of k vertices with maximum average degree in the subgraph induced by this set. In the weighted version of DkS we are also given nonnegative weights on the edges and the goal is to find a k-vertex induced subgraph of maximum average edge weight. Algorithms for finding DkS are useful tools for analyzing networks. In particular, they have been used to select features for ranking (Geng et al., 2007), to identify cores of communities (Kumar et al., 1999), and to combat link spam (Gibson et al., 2005).

It has been shown that the DkS problem is NP hard for bipartite graphs and chordal graphs (Corneil and Perl, 1984), and even for graphs of maximum degree three (Feige et al., 2001). A large body of algorithms have been proposed based on a variety of techniques including greedy algorithms (Feige et al., 2001; Asahiro et al., 2002; Ravi et al., 1994), linear programming (Billionnet and Roupin, 2004; Khuller and Saha, 2009), and semidefinite programming (Srivastav and Wolf, 1998; Ye and Zhang, 2003). For general k, the algorithm developed by Feige et al. (2001) achieves the best approximation ratio of $O(n^{\varepsilon})$ where $\varepsilon < 1/3$. Ravi et al. (1994) proposed 4-approximation algorithms for weighted DkS on complete graphs for which the weights satisfy the triangle inequality. Liazi et al. (2008) has presented a 3-approximation algorithm for DkS for chordal graphs. Recently, Jiang et al. (2010) proposed to reformulate DkS as a 1-mean clustering problem and developed a 2-approximation to the reformulated clustering problem. Moreover, based on this reformulation, Yang (2010) proposed a $1+\varepsilon$-approximation algorithm with certain exhaustive (and thus expensive) initialization procedure. In general, however, Khot (2006) showed that DkS has no polynomial time approximation scheme (PTAS), assuming that there are no sub-exponential time algorithms for problems in NP.

Mathematically, DkS can be restated as the following binary quadratic programming problem:

$$\max_{\pi \in \mathbb{R}^n} \pi^\top W \pi, \qquad \text{subject to } \pi \in \{1,0\}^n, \|\pi\|_0 = k, \tag{8}$$

where W is the (non-negative weighted) adjacency matrix of G. If G is an undirected graph, then W is symmetric. If G is directed, then W could be asymmetric. In this latter case, from the fact that $\pi^\top W \pi = \pi^\top \frac{W+W^\top}{2} \pi$, we may equivalently solve Problem (8) by replacing W with $\frac{W+W^\top}{2}$. Therefore, in the following discussion, we always assume that the affinity matrix W is symmetric (or G is undirected).

4.3.1 THE TPOWER-DKS ALGORITHM

We propose the TPower-DkS algorithm as an adaptation of TPower to the DkS problem. The process generates a sequence of intermediate vectors $\pi_0, \pi_1, \ldots$ from a starting vector π_0. At each step t the vector π_{t-1} is multiplied by the matrix W, then π_t is set to be the indicator vector of the top k entries in $W\pi_{t-1}$. The TPower-Dks is outlined in Algorithm 2. The convergence of this algorithm can be justified using the same arguments of bounding optimization as described in §2.2.

```
Algorithm 2: Truncated Power Method for DkS (TPower-DkS)
  Input       : $W \in \mathbb{S}^n_+$,, initial vector $\pi_0 \in \mathbb{R}^n$
  Output      : $\pi_t$
  Parameters  : cardinality $k \in \{1, \ldots, n\}$
  Let $t = 1$.
  repeat
      Compute $\pi'_t = W\pi_{t-1}$.
      Identify $F_t = \mathrm{supp}(\pi'_t, k)$ the index set of $\pi'_t$ with top $k$ values.
      Set $\pi_t$ to be 1 on the index set $F_t$, and 0 otherwise.
      $t \leftarrow t+1$.
  until Convergence;
```

Remark 7 *By relaxing the constraint $\pi \in \{0,1\}^n$ to $\|\pi\| = \sqrt{k}$, we may convert the densest k-subgraph problem* (8) *to the standard sparse eigenvalue problem* (1) *(up to a scaling) and then directly apply TPower (in Algorithm 1) for solution. Our numerical experience shows that such a relaxation strategy also works satisfactory in practice, although is slightly inferior to TPower-DkS (in Algorithm 2) which directly addresses the original problem.*

Remark 8 *As aforementioned that the DkS problem is generally NP-hard. The quality of its approximate solution can be measured by the approximation ratio defined as the output objective to the optimal objective. Recently, Jiang et al. (2010) proposed to reformulate DkS as a 1-mean clustering problem and developed a* 2*-approximation to the reformulated clustering problem. Moreover, based on this reformulation, Yang (2010) proposed a* $1+\varepsilon$*-approximation algorithm with certain exhaustive (and thus expensive) initialization procedure. Provided that W is positive semidefinite with equal diagonal elements, trivial derivation shows that TPower-DkS is identical to the method of Jiang et al. (2010). Therefore, the approximation ratio results from Jiang et al. (2010); Yang (2010) can be shared by TPower-DkS in this restricted case.*

Note that in Algorithm 2 we require that W is positive semidefinite. The motivation of this requirement is to guarantee the convexity of the objective in problem (8), and thus the convergence of Algorithm 2 can be justified by the similar arguments in §2.2. In many real-world DkS problems, however, it is often the case that the affinity matrix W is not positive semidefinite. In this case, the objective is non-convex and thus the monotonicity of TPower-DkS does not hold. However, this complication can be circumvented by instead running the algorithm with the shifted quadratic function:

$$\max_{\pi\in\mathbb{R}^n} \pi^\top (W + \tilde{\lambda} I_{p\times p})\pi, \qquad \text{subject to } \pi \in \{0,1\}^n, \|\pi\|_0 = k.$$

where $\tilde{\lambda} > 0$ is large enough such that $\tilde{W} = W + \tilde{\lambda} I_{p\times p} \in \mathbb{S}^n_+$. On the domain of interest, this change only adds a constant term to the objective function. The TPower-DkS, however, produces a different sequence of iterates, and there is a clear tradeoff. If the second term dominates the first term (say by choosing a very large $\tilde{\lambda}$), the objective function becomes approximately a squared norm, and the algorithm tends to terminate in very few iterations. In the limiting case of $\tilde{\lambda} \to \infty$, the method will not move away from the initial iterate. To handle this issue, we propose to gradually increase $\tilde{\lambda}$ during the iterations and we do so only when the monotonicity is violated. To be precise, if at a time instance t, $\pi_t^\top W \pi_t < \pi_{t-1}^\top W \pi_{t-1}$, then we add $\tilde{\lambda} I_{p\times p}$ to W with a gradually increased $\tilde{\lambda}$ by repeating the current iteration with the updated matrix until $\pi_t^\top (W + \tilde{\lambda} I_{p\times p})\pi_t \geq \pi_{t-1}^\top (W + \lambda I_{p\times p})\pi_{t-1}$,[2] which implies $\pi_t^\top W \pi_t \geq \pi_{t-1}^\top W \pi_{t-1}$.

4.3.2 On Initialization

Since TPower-DkS is a monotonically increasing procedure, it guarantees to improve the initial point π_0. Basically, any existing approximation DkS method, for example, greedy algorithms (Feige et al., 2001; Ravi et al., 1994), can be used to initialize TPower-DkS. In our numerical experiments, we observe that by simply setting π_0 as the indicator vector of the vertices with the top k (weighted) degrees, our method can achieve very competitive results on all the real-world data sets we have tested on.

[2] Note that the inequality $\pi_t^\top (W + \tilde{\lambda} I_{p\times p})\pi_t \geq \pi_{t-1}^\top (W + \tilde{\lambda} I_{p\times p})\pi_{t-1}$ is deemed to be satisfied when $\tilde{\lambda}$ is large enough, for example, when $W + \tilde{\lambda} I_{p\times p} \in \mathbb{S}^n_+$.

4.3.3 RESULTS ON WEB GRAPHS

We have tested TPower on four page-level web graphs: cnr-2000, amazon-2008, ljournal-2008, hollywood-2009, from the WebGraph framework provided by the Laboratory for Web Algorithms.[3] We treated each directed arc as an undirected edge. Table 4 lists the statistics of the data sets used in the experiment.

Graph	Nodes ($\|V\|$)	Total Arcs ($\|E\|$)	Average Degree
cnr-2000	325,557	3,216,152	9.88
amazon-2008	735,323	5,158,388	7.02
ljournal-2008	5,363,260	79,023,142	14.73
hollywood-2009	1,139,905	113,891,327	99.91

Table 4: The statistics of the web graph data sets.

We compare our TPower-DkS method with two greedy methods for the DkS problem. One greedy method is proposed by Ravi et al. (1994) which is referred to as Greedy-Ravi in our experiments. The Greedy-Ravi algorithm works as follows: it starts from a heaviest edge and repeatedly adds a vertex to the current subgraph to maximize the weight of the resulting new subgraph; this process is repeated until k vertices are chosen. The other greedy method is developed by Feige et al. (2001, Procedure 2) which is referred as Greedy-Feige in our experiments. The procedure works as follows: let S denote the $k/2$ vertices with the highest degrees in G; let C denote the $k/2$ vertices in the remaining vertices with largest number of neighbors in S; return $S \cup C$.

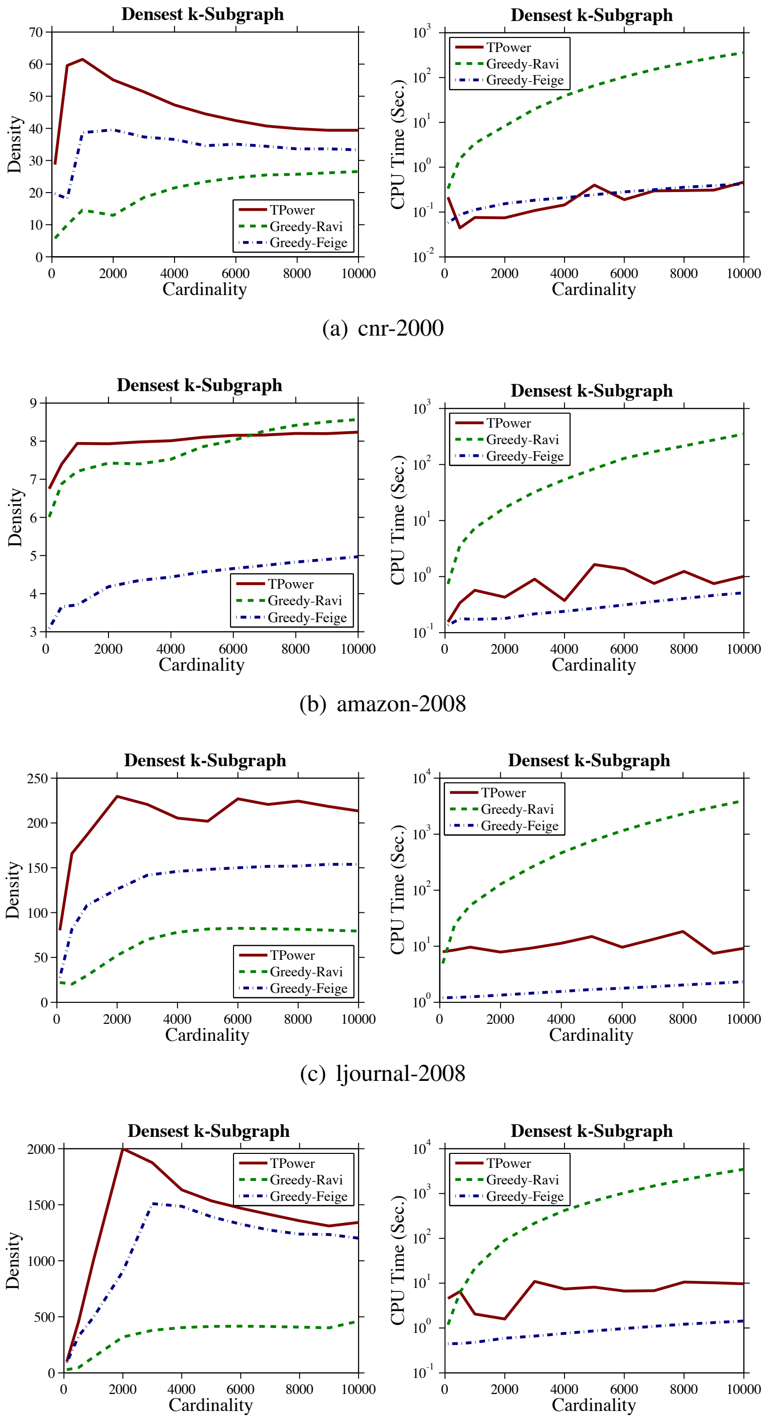
Figure 3 shows the density value $\pi^\top W\pi/k$ and CPU time versus the cardinality k. From the density curves we can observe that on cnr-2000, ljournal-2008 and hollywood-2009, TPower-DkS consistently outputs denser subgraphs than the two greedy algorithms, while on amazon-2008, TPower-DkS and Greedy-Ravi are comparable and both are better than Greedy-Feige. For CPU running time, it can be seen from the right column of Figure 3 that Greedy-Feige is the fastest among the three methods while TPower-DkS is only slightly slower. This is due to the fact that TPower-DkS needs iterative matrix-vector products while Greedy-Feige only needs a few degree sorting operations. Although TPower-DkS is slightly slower than Greedy-Feige, it is still quite efficient. For example, on hollywood-2009 which has hundreds of millions of arcs, for each k, Greedy-Feige terminates within about 1 second while TPower terminates within about 10 seconds. The Greedy-Ravi method is however much slower than the other two on all the graphs when k is large.

4.3.4 RESULTS ON AIR-TRAVEL ROUTINE

We have applied TPower-DkS to identify subsets of American and Canadian cities that are most easily connected to each other, in terms of estimated commercial airline travel time. The graph[4] is of size $|V| = 456$ and $|E| = 71,959$: the vertices are 456 busiest commercial airports in United States and Canada, while the weight w_{ij} of edge e_{ij} is set to the inverse of the mean time it takes to travel from city i to city j by airline, including estimated stopover delays. Due to the headwind

[3] These four data sets are publicly available at `http://lae.dsi.unimi.it/datasets.php`.

[4] The data is available at `www.psi.toronto.edu/affinitypropogation`.



(a) cnr-2000

(b) amazon-2008

(c) ljournal-2008

(d) hollywood-2009

Figure 3: Identifying densest k-subgraph on four web graphs. Left: density curves as a function of cardinality. Right: CPU time (in second) curves as a function of cardinality. For better viewing, please see the original pdf file.

effect, the transit time can depend on the direction of travel; thus 36% of the weight are asymmetric. Figure 4(a) shows a map of air-travel routine.

As in the previous experiment, we compare TPower-DkS to Greedy-Ravi and Greedy-Feige on this data set. For all the three considered algorithms, the densities of k-subgraphs under different k values are shown in Figure 4(b), and the CPU running time curves are given in Figure 4(c). From the former figure we observe that TPower-DkS consistently outperforms the other two greedy algorithms in terms of the density of the extracted k-subgraphs. From the latter figure we can see that TPower-DkS is slightly slower than Greed-Feige but much faster than Greedy-Ravi. Figure 4(d)$\sim$4(f) illustrate the densest k-subgraph with $k = 30$ output by the three algorithms. In each of these three subgraph, the red dot indicates the representing city with the largest (weighted) degree. Both TPower-DkS and Greedy-Feige reveal 30 cities in east US. The former takes *Cleveland* as the representing city while the latter *Cincinnati*. Greedy-Ravi reveals 30 cities in west US and CA and takes *Vancouver* as the representing city. Visual inspection shows that the subgraph recovered by TPower-DkS is the densest among the three.

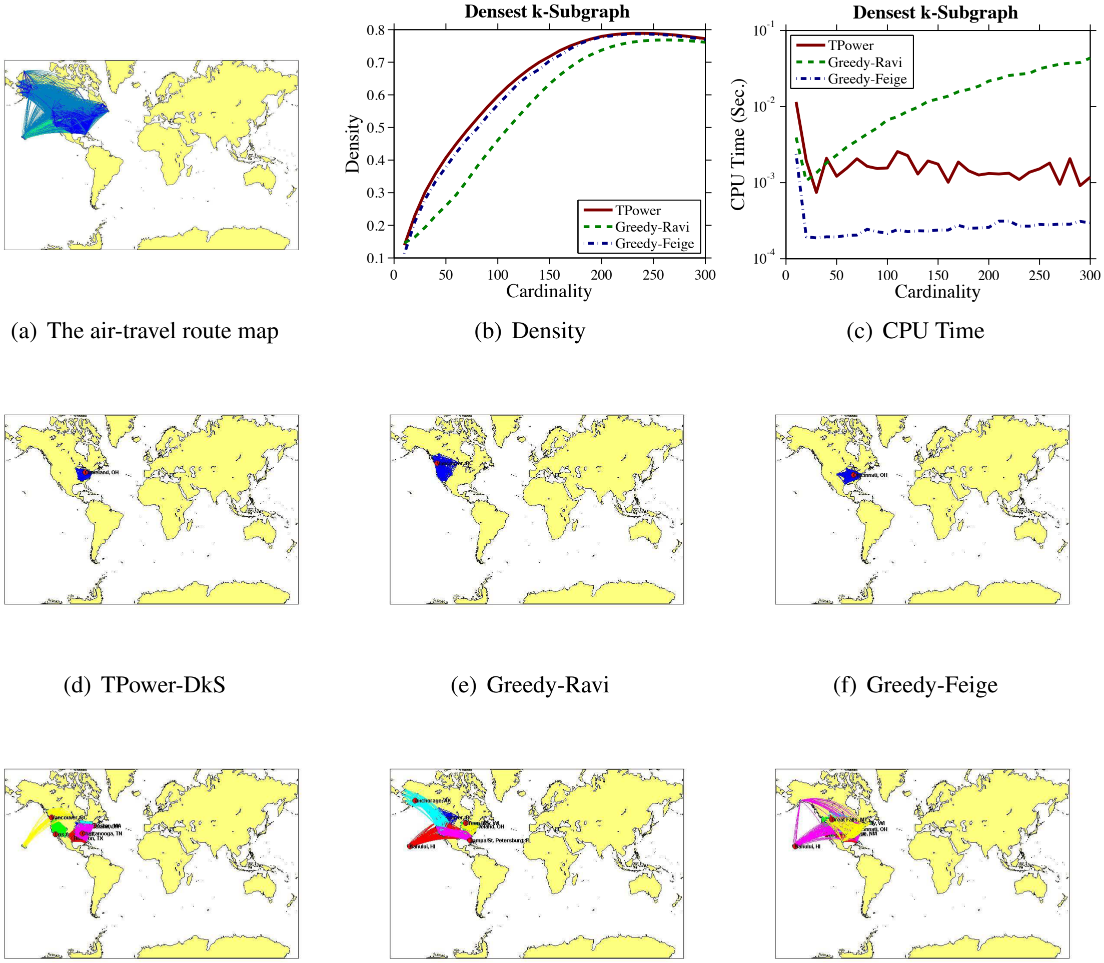
After discovering the densest k-subgraph, we can eliminate their nodes and edges from the graph and then apply the algorithms on the reduced graph to search for the next densest subgraph. This sequential procedure can be repeated to find multiple densest k-subgraphs. Figure 4(g)$\sim$4(i) illustrate sequentially estimated six densest 30-subgraphs by the three considered algorithms. Again, visual inspection shows that our method outputs more geographically compact subsets of cities than the other two. As a quantitative result, the total densities of the six subgraphs discovered by the three algorithms are: 1.14 (TPower-DkS), 0.90 (Greedy-Feige) and 0.99 (Greedy-Ravi), respectively.

5. Conclusion

The sparse eigenvalue problem has been widely studied in machine learning with applications such as sparse PCA. TPower is a truncated power iteration method that approximately solves the non-convex sparse eigenvalue problem. Our analysis shows that when the underlying matrix has sparse eigenvectors, under proper conditions TPower can approximately recover the true sparse solution. The theoretical benefit of this method is that with appropriate initialization, the reconstruction quality depends on the restricted matrix perturbation error at size s that is comparable to the sparsity $\bar{k}$, instead of the full matrix dimension p. This explains why this method has good empirical performance. To our knowledge, this is one of the first theoretical results of this kind, although our empirical study suggests that it might be possible to prove related sparse recovery results for some other algorithms we have tested. We have applied TPower to two concrete applications: sparse PCA and the densest k-subgraph finding problem. Extensive experimental results on synthetic and real-world data sets validate the effectiveness and efficiency of the TPower algorithm. To summarize, simply combing power iteration with hard-thresholding truncation provides an accurate and scalable computational method for the sparse eigenvalue problem.

Acknowledgments

The work is supported by NSF grants DMS-1007527, IIS-1016061, and IIS-1250985.



(a) The air-travel route map

(b) Density

(c) CPU Time

(d) TPower-DkS

(e) Greedy-Ravi

(f) Greedy-Feige

(g) TPower-DkS

(h) Greedy-Ravi

(i) Greedy-Feige

Figure 4: Identifying densest *k*-subgraph of air-travel routing. Top row: Route map, and the density and CPU time evolving curves. Middle row: The densest 30-subgraph discovered by the three considered algorithms. Bottom row: Sequentially discovered six densest 30-subgraphs by the three considered algorithms. For better viewing, please see the original pdf file.

Appendix A. Proof Of Theorem 4

Our proof employs several technical tools including the perturbation theory of symmetric eigenvalue problem (Lemma 9 and Lemma 10), the convergence analysis of traditional power method (Lemma 11), and the error analysis of hard-thresholding operation (Lemma 12).

We state the following standard result from the perturbation theory of symmetric eigenvalue problem (see, e.g., Golub and Loan, 1996).

Lemma 9 *If B and $B+U$ are $p\times p$ symmetric matrices, then $\forall 1\le k\le p$,*

$$\lambda_k(B)+\lambda_p(U)\le\lambda_k(B+U)\le\lambda_k(B)+\lambda_1(U),$$

where $\lambda_k(B)$ denotes the k-th largest eigenvalue of matrix B.

Lemma 10 *Consider set F such that $supp(\bar{x})\subseteq F$ with $|F|=s$. If $\rho(E,s)\le\Delta\lambda/2$, then the ratio of the second largest (in absolute value) to the largest eigenvalue of sub matrix A_F is no more than $\gamma(s)$. Moreover,*

$$\|\bar{x}^\top - x(F)\|\le\delta(s):=\frac{\sqrt{2}\rho(E,s)}{\sqrt{\rho(E,s)^2+(\Delta\lambda-2\rho(E,s))^2}}.$$

Proof We may use Lemma 9 with $B=\bar{A}_F$ and $U=E_F$ to obtain

$$\lambda_1(A_F)\ge\lambda_1(\bar{A}_F)+\lambda_p(E_F)\ge\lambda_1(\bar{A}_F)-\rho(E_F)\ge\lambda-\rho(E,s)$$

and $\forall j\ge 2$,

$$|\lambda_j(A_F)|\le|\lambda_j(\bar{A}_F)|+\rho(E_F)\le\lambda-\Delta\lambda+\rho(E,s).$$

This implies the first statement of the lemma.

Now let $x(F)$, the largest eigenvector of A_F, be $\alpha\bar{x}+\beta x'$, where $\|\bar{x}\|_2=\|x'\|_2=1$, $\bar{x}^\top x'=0$ and $\alpha^2+\beta^2=1$, with eigenvalue $\lambda'\ge\lambda-\rho(E,s)$. This implies that

$$\alpha A_F\bar{x}+\beta A_F x'=\lambda'(\alpha\bar{x}+\beta x'),$$

implying

$$\alpha x'^\top A_F\bar{x}+\beta x'^\top A_F x'=\lambda'\beta.$$

That is,

$$|\beta|=|\alpha|\frac{x'^\top A_F\bar{x}}{\lambda'-x'^\top A_F x'}\le|\alpha|\frac{|x'^\top A_F\bar{x}|}{\lambda'-x'^\top A_F x'}=|\alpha|\frac{|x'^\top E_F\bar{x}|}{\lambda'-x'^\top A_F x'}\le t|\alpha|,$$

where $t=\rho(E,s)/(\Delta\lambda-2\rho(E,s))$. This implies that $\alpha^2(1+t^2)\ge\alpha^2+\beta^2=1$, and thus $\alpha^2\ge 1/(1+t^2)$. Without loss of generality, we may assume that $\alpha>0$, because otherwise we can replace $\bar{x}$ with $-\bar{x}$. It follows that

$$\|x(F)-\bar{x}\|^2=2-2x(F)^\top\bar{x}=2-2\alpha\le 2\frac{\sqrt{1+t^2}-1}{\sqrt{1+t^2}}\le\frac{2t^2}{1+t^2}.$$

This implies the desired bound. ■

The following result measures the progress of untruncated power method.

Lemma 11 *Let y be the eigenvector with the largest (in absolute value) eigenvalue of a symmetric matrix A, and let $\gamma < 1$ be the ratio of the second largest to largest eigenvalue in absolute values. Given any x such that $\|x\| = 1$ and $y^\top x > 0$; let $x' = Ax/\|Ax\|$, then*

$$|y^\top x'| \geq |y^\top x|[1 + (1-\gamma^2)(1-(y^\top x)^2)/2].$$

Proof Without loss of generality, we may assume that $\lambda_1(A) = 1$ is the largest eigenvalue in absolute value, and $|\lambda_j(A)| \leq \gamma$ when $j > 1$. We can decompose x as $x = \alpha y + \beta y'$, where $y^\top y' = 0$, $\|y\| = \|y'\| = 1$, and $\alpha^2 + \beta^2 = 1$. Then $|\alpha| = |x^\top y|$. Let $z' = Ay'$, then $\|z'\| \leq \gamma$ and $y^\top z' = 0$. This means $Ax = \alpha y + \beta z'$, and

$$\begin{aligned} |y^\top x'| =& \frac{|y^\top Ax|}{\|Ax\|} = \frac{|\alpha|}{\sqrt{\alpha^2 + \beta^2\|z'\|^2}} \geq \frac{|\alpha|}{\sqrt{\alpha^2+\beta^2\gamma^2}} \\ =& \frac{|y^\top x|}{\sqrt{1-(1-\gamma^2)(1-(y^\top x)^2)}} \\ \geq& |y^\top x| \; [1 + (1-\gamma^2)(1-(y^\top x)^2)/2]. \end{aligned}$$

The last inequality is due to $1/\sqrt{1-z} \geq 1 + z/2$ for $z \in [0,1)$. This proves the desired bound. ■

The following lemma quantifies the error introduced by the truncation step in TPower.

Lemma 12 *Consider $\bar{x}$ with $supp(\bar{x}) = \bar{F}$ and $\bar{k} = |\bar{F}|$. Consider y and let $F = supp(y,k)$ be the indices of y with the largest k absolute values. If $\|\bar{x}\| = \|y\| = 1$, then*

$$|Truncate(y,F)^\top \bar{x}| \geq |y^\top \bar{x}| - (\bar{k}/k)^{1/2} \min\left[\sqrt{1-(y^\top\bar{x})^2}, (1 + (\bar{k}/k)^{1/2})\; (1-(y^\top\bar{x})^2)\right].$$

Proof Without loss of generality, we assume that $y^\top \bar{x} = \Delta > 0$. We can also assume that $\Delta > \sqrt{\bar{k}/(\bar{k}+k)}$ because otherwise the right hand side is smaller than zero, and thus the result holds trivially.

Let $F_1 = \bar{F} \setminus F$, and $F_2 = \bar{F} \cap F$, and $F_3 = F \setminus \bar{F}$. Now, let $\bar{\alpha} = \|\bar{x}_{F_1}\|$, $\bar{\beta} = \|\bar{x}_{F_2}\|$, $\alpha = \|y_{F_1}\|$, $\beta = \|y_{F_2}\|$, and $\gamma = \|y_{F_3}\|$. let $k_1 = |F_1|$, $k_2 = |F_2|$, and $k_3 = |F_3|$. It follows that $\alpha^2/k_1 \leq \gamma^2/k_3$. Therefore

$$\Delta^2 \leq [\bar{\alpha}\alpha + \bar{\beta}\beta]^2 \leq \alpha^2 + \beta^2 \leq 1 - \gamma^2 \leq 1 - (k_3/k_1)\alpha^2.$$

This implies that

$$\alpha^2 \leq (k_1/k_3)(1-\Delta^2) \leq (\bar{k}/k)(1-\Delta^2) < \Delta^2, \tag{9}$$

where the second inequality follows from $\bar{k} \leq k$ and the last inequality follows from the assumption $\Delta > \sqrt{\bar{k}/(\bar{k}+k)}$. Now by solving the following inequality for $\bar{\alpha}$

$$\alpha\bar{\alpha} + \sqrt{1-\alpha^2}\sqrt{1-\bar{\alpha}^2} \geq \alpha\bar{\alpha} + \beta\bar{\beta} \geq \Delta$$

under the condition $\Delta > \alpha \geq \alpha\bar{\alpha}$, we obtain that

$$\bar{\alpha} \leq \alpha\Delta + \sqrt{1-\alpha^2}\sqrt{1-\Delta^2} \leq \min\left[1, \alpha + \sqrt{1-\Delta^2}\right] \leq \min\left[1, (1+(\bar{k}/k)^{1/2})\sqrt{1-\Delta^2}\right], \tag{10}$$

where the second inequality follows from the Cauchy-Schwartz inequality and $\Delta \leq 1$, $\sqrt{1-\alpha^2} \leq 1$, while the last inequality follows from (9). Finally,

$$\begin{aligned} |y^\top \bar{x}| - |\text{Truncate}(y,F)^\top \bar{x}| \quad &\leq \quad |(y-\text{Truncate}(y,F))^\top \bar{x}| \\ &\leq \quad \alpha\bar{\alpha} \leq (\bar{k}/k)^{1/2} \min \left[\sqrt{1-(y^\top \bar{x})^2}, (1+(\bar{k}/k)^{1/2})\,(1-(y^\top \bar{x})^2) \right], \end{aligned}$$

where the last inequality follows from (9) and (10). This leads to the desired bound. ■

Next is our main lemma, which says each step of sparse power method improves eigenvector estimation.

Lemma 13 *Assume that $k \geq \bar{k}$. Let $s = 2k + \bar{k}$. If $|x_{t-1}^\top \bar{x}| > \theta + \delta(s)$, then*

$$\sqrt{1-|\hat{x}_t^\top \bar{x}|} \leq \mu \sqrt{1-|x_{t-1}^\top \bar{x}|} + \sqrt{10}\delta(s).$$

Proof Let $F = F_{t-1} \cup F_t \cup \text{supp}(\bar{x})$. Consider the following vector

$$\tilde{x}_t' = A_F x_{t-1}/\|A_F x_{t-1}\|, \tag{11}$$

where A_F denotes the restriction of A on the rows and columns indexed by F. We note that replacing x_t' with $\tilde{x}_t'$ in Algorithm 1 does not affect the output iteration sequence $\{x_t\}$ because of the sparsity of x_{t-1} and the fact that the truncation operation is invariant to scaling. Therefore for notation simplicity, in the following proof we will simply assume that x_t' is redefined as $x_t' = \tilde{x}_t'$ according to (11).

Without loss of generality and for simplicity, we may assume that $x_t'^\top x(F) \geq 0$ and $x_{t-1}^\top \bar{x} \geq 0$, because otherwise we can simply do appropriate sign changes in the proof. We obtain from Lemma 11 that

$$x_t'^\top x(F) \geq x_{t-1}^\top x(F)\ [1 + (1-\gamma(s)^2)(1-(x_{t-1}^\top x(F))^2)/2].$$

This implies that

$$\begin{aligned} [1 - x_t'^\top x(F)] \leq & [1 - x_{t-1}^\top x(F)]\ [1 - (1-\gamma(s)^2)(1 + x_{t-1}^\top x(F))(x_{t-1}^\top x(F))/2] \\ \leq & [1 - x_{t-1}^\top x(F)]\ [1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)], \end{aligned}$$

where in the derivation of the second inequality, we have used Lemma 10 and the assumption of the lemma that implies $x_{t-1}^\top x(F) \geq x_{t-1}^\top \bar{x} - \delta(s) \geq \theta$. We thus have

$$\|x_t' - x(F)\| \leq \|x_{t-1} - x(F)\|\ \sqrt{1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)}.$$

Therefore using Lemma 10, we have

$$\|x_t' - \bar{x}\| \leq \|x_{t-1} - \bar{x}\|\ \sqrt{1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)} + 2\delta(s).$$

This is equivalent to

$$\sqrt{1-|x_t'^\top \bar{x}|} \leq \sqrt{1-|x_{t-1}^\top \bar{x}|}\sqrt{1 - 0.5\theta(1+\theta)(1-\gamma(s)^2)} + \sqrt{2}\delta(s).$$

Next we can apply Lemma 12 and use $k \geq \bar{k}$ to obtain

$$\begin{aligned}\sqrt{1-|\hat{x}_t^\top \bar{x}|} &\leq \sqrt{1-|x_t'^\top \bar{x}|+((\bar{k}/k)^{1/2}+\bar{k}/k)(1-|x_t'^\top \bar{x}|^2)} \\ &\leq \sqrt{1-|x_t'^\top \bar{x}|}\sqrt{1+2((\bar{k}/k)^{1/2}+\bar{k}/k)} \\ &\leq \mu\sqrt{1-|x_{t-1}{}^\top \bar{x}|}+\sqrt{10}\delta(s).\end{aligned}$$

This proves the second desired inequality. ■

We are now in the position to prove Theorem 4.

Proof of Theorem 4:
Let us distinguish the following two complementary cases:
Case I: $\theta+\delta(s) > 1-10\delta(s)^2/(1-\mu)^2$. In this case, we have that $x_0^\top \bar{x} \geq \theta+\delta(s) > 1-10\delta(s)^2/(1-\mu)^2$ which implies the inequality (5).
Case II: $\theta+\delta(s) \leq 1-10\delta(s)^2/(1-\mu)^2$. In this case, we first prove by induction that for all $t \geq 0$, $x_t^\top \bar{x} \geq \theta+\delta(s)$. This is obviously hold for $t=0$. Assume that $|x_{t-1}^\top \bar{x}| \geq \theta+\delta(s)$. Let us further distinguish the following two cases:

(a) $\sqrt{1-|x_{t-1}^\top \bar{x}|} \geq \sqrt{10}\delta(s)/(1-\mu)$. From Lemma 13 we obtain that

$$\sqrt{1-|x_t^\top \bar{x}|} \leq \sqrt{1-|\hat{x}_t^\top \bar{x}|} \leq \mu\sqrt{1-|x_{t-1}^\top \bar{x}|}+\sqrt{10}\delta(s) \leq \sqrt{1-|x_{t-1}^\top \bar{x}|},$$

where the first inequality follows from $|x_t^\top \bar{x}| = |\hat{x}_t^\top \bar{x}|/\|\hat{x}_t\| \geq |\hat{x}_t^\top \bar{x}|$. This implies $|x_t^\top \bar{x}| \geq |x_{t-1}^\top \bar{x}| \geq \theta+\delta(s)$.

(b) $\sqrt{1-|x_{t-1}^\top \bar{x}|} < \sqrt{10}\delta(s)/(1-\mu)$. Based on the previous argument we have

$$\sqrt{1-|x_t^\top \bar{x}|} \leq \mu\sqrt{1-|x_{t-1}^\top \bar{x}|}+\sqrt{10}\delta(s) < \sqrt{10}\delta(s)/(1-\mu),$$

which implies that $|x_t^\top \bar{x}| > 1-10\delta(s)^2/(1-\mu)^2 \geq \theta+\delta(s)$.

In both cases (a) and (b), we have $|x_t^\top \bar{x}| \geq \theta+\delta(s)$ and this finishes the induction. Therefore, by recursively applying Lemma 13 we have that for all $t \geq 0$

$$\sqrt{1-|x_t^\top \bar{x}|} \leq \mu^t\sqrt{1-|x_0^\top \bar{x}|}+\sqrt{10}\delta(s)/(1-\mu),$$

which is inequality (6). This completes the proof.

References

A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, and A. Rosenwald. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

A. 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. *Cell Biology*, 96:6745–6750, 1999.

A. A. Amini and M. J. Wainwright. High-dimensional analysis of semidefinite relaxiation for sparse principal components. *Annals of Statistics*, 37:2877–2921, 2009.

Y. Asahiro, R. Hassin, and K. Iwama. Complexity of finidng dense subgraphs. *Discrete Applied Mathematics*, 211(1-3):15–26, 2002.

A. Billionnet and F. Roupin. A deterministic algorithm for the densest k-subgraph problem using linear programming. Technical report, Technical Report, No. 486, CEDRIC, CNAM-IIE, Paris, 2004.

T. Cai, Z. Ma, and Y. Wu. Sparse pca: Optimal rates and adaptive estimation. 2012. URL `arxiv.org/pdf/1211.1309v1.pdf`.

E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51:4203–4215, 2005.

D. G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9:27–39, 1984.

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

A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

U. Feige, G. Kortsarz, and D. Peleg. The dense *k*-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

X. Geng, T. Liu, T. Qin, and H. Li. Feature selection for ranking. In *Proceedings of the 30th Annual International ACM SIGIR Conference (SIGIR'07)*, 2007.

D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05)*, pages 721–732, 2005.

G. H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

J. Jeffers. Two case studies in the application of principal components. *Applied Statistics*, 16(3): 225–236, 1967.

P. Jiang, J. Peng, M. Heath, and R. Yang. Finding densest k-subgraph via 1-mean clustering and low-dimension approximation. Technical report, 2010.

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

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.

M. Journée, Y. Nesterov, P. Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

S. Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

S. Khuller and B. Saha. On finding dense subgraphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pages 597–608, 2009.

R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the 8th World Wide Web Conference (WWW'99)*, pages 403–410, 1999.

M. Liazi, I. Milis, and V. Zissimopoulos. A constant approximation algorithm for the densest k-subgraph problem on chordal graphs. *Information Processing Letters*, 108(1):29–32, 2008.

Z. Ma. Sparse principal component analysis and iterative thresholding. *Annals of Statistics*, to appear, 2013.

L. Mackey. Deflation methods for sparse pca. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS'08)*, 2008.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:10–60, 2010.

B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse lda. In *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*, pages 641–648, 2006.

D. Paul and I.M. Johnstone. Augmented sparse principal component analysis for high dimensional data. 2012. URL `arxiv.org/pdf/1202.1242v1.pdf`.

S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42:299–310, 1994.

D. Shen, H. Shen, and J.S. Marron. Consistency of sparse pca in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, 115:317–333, 2013.

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.

A. Srivastav and K. Wolf. Finding dense subgraphs with semidefinite programming. In *Proceedings of International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98)*, pages 181–191, 1998.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

R. Yang. New approximation methods for solving binary quadratic programming problem. Technical report, Master Thesis, Department of Industrial and Enterprise Systems Engineering, University of Illnois at Urbana-Champaign, 2010.

Y. Y. Ye and J. W. Zhang. Approximation of dense-n/2-subgraph and the complement of min-bisection. *Journal of Global Optimization*, 25:55–73, 2003.

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