Even Faster SVD Decomposition
Yet Without Agonizing Pain

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Princeton University

Yuanzhi Li
yuanzhil@cs.princeton.edu
Princeton University

first circulated on May 20, 2016

Abstract

We study $k$-SVD that is to obtain the first $k$ singular vectors of a matrix $A$ approximately. Recently, a few breakthroughs have been discovered on $k$-SVD: Musco and Musco [14] provided the first gap-free theorem for the block Krylov method, Shamir [16] discovered the first variance-reduction stochastic method, and Bhojanapalli et al. [3] provided the fastest $O(\text{nnz}(A) + \text{poly}(1/\varepsilon))$-type of algorithm using alternating minimization.

In this paper, put forward a new framework for SVD and improve the above breakthroughs. We obtain faster gap-free convergence rate outperforming [14], we obtain the first accelerated and stochastic method outperforming [16]. In the $O(\text{nnz}(A) + \text{poly}(1/\varepsilon))$ running-time regime, we outperform [3] in certain parameter regimes without even using alternating minimization.

1 Introduction

The singular value decomposition (SVD) of a rank-$r$ matrix $A \in \mathbb{R}^{d \times n}$ corresponds to decomposing $A = V \Sigma U^\top$ where $V \in \mathbb{R}^{d \times r}$, $U \in \mathbb{R}^{n \times r}$ are two column orthonormal matrices, and $\Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_r\} \in \mathbb{R}^{r \times r}$ is a non-negative diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. The columns of $V$ (resp. $U$) are called the left (resp. right) singular vectors of $A$ and the diagonal entries of $\Sigma$ are called the singular values of $A$. SVD is one of the most fundamental tools used in machine learning, computer vision, statistics, and operations research, and is essentially equivalent to principal component analysis (PCA) up to column averaging.

A rank $k$ partial SVD (or $k$-SVD for short) is to find the top $k$ left (resp. right) singular vectors of $A$, or equivalently, the first $k$ columns of $V$ (resp. $U$). Denoting by $V_k \in \mathbb{R}^{d \times k}$ the first $k$ columns of $V$, and $U_k$ the first $k$ columns of $U$, one can define $A_k^* \overset{\text{def}}{=} V_k V_k^\top A = V_k \Sigma_k U_k^\top$ where $\Sigma_k = \text{diag}\{\sigma_1, \ldots, \sigma_k\}$. Under this notation, $A_k^*$ is the the best rank-$k$ approximation of matrix $A$ in terms of minimizing $\|A - A_k\|$ among all rank $k$ matrices $A_k$. Here, the norm can be any Schatten-$q$ norm for $q \in [1, \infty]$, including spectral norm ($q = \infty$) and Frobenius norm ($q = 2$), therefore making $k$-SVD a very powerful tool in terms of information retrieval, data de-noising, or even data compression.

Traditional algorithms to compute SVD essentially run in time $O(nd \min\{d, n\})$, which is usually very expensive for big-data scenarios. As for $k$-SVD, defining $\text{gap} \overset{\text{def}}{=} (\sigma_k - \sigma_{k-1})/\sigma_k$ to be the relative $k$-th eigengap of matrix $A$, the famous subspace power method or block Krylov method [9] solves $k$-SVD in time $O(\text{gap}^{-1} \cdot k \cdot \text{nnz}(A) \cdot \log(1/\varepsilon))$ or $O(\text{gap}^{-0.5} \cdot k \cdot \text{nnz}(A) \cdot 1/\varepsilon))$ respectively if ignoring lower order terms. Here, $\text{nnz}(A)$ is the number of non-zero elements in matrix $A$, and the more precise running times are stated in Table 1.
Recently, there are breakthroughs to compute $k$-SVD faster, from three incomparable perspectives.

The first breakthrough is the work of Musco and Musco [14] for providing running times that do not depend on properties (i.e., the eigengap) of $A$ for block Krylov method. As highlighted in [14], providing gap-free results was an open question for decades and is a more reliable goal for practical purposes. Specifically, they proved that the same block Krylov method converges in time $\tilde{O}\left(\frac{\text{nnz}(A)}{\varepsilon^{1/2}} + \frac{k^2 d}{\varepsilon} + \frac{k^3}{\varepsilon^{3/2}}\right)$, where $\varepsilon$ is the multiplicative approximation error.\footnote{In this paper, we use $\tilde{O}$ notations to hide possible logarithmic factors on $1/\text{gap}, 1/\varepsilon, n, d, k$ and potentially also on $\sigma_1/\sigma_{k+1}$.}

The second breakthrough is the work of Shamir [16] for providing a fast stochastic $k$-SVD algorithm. In a stochastic setting, we assume $A$ is given in form $A^\top = \frac{1}{n} \sum_{i=1}^n a_i a_i^\top$ and each $a_i \in \mathbb{R}^d$ has Euclidean norm at most 1. (1.1) Instead of using the entire matrix $AA^\top$ when applying (subspace) power method, one can use a rank-1 copy $a_i a_i^\top$ where $i$ is chosen uniformly at random. While this stochastic approach improves the per-iteration running time for obvious reason, one needs to carefully introduce ad-hoc variance-reduction techniques in order to make the algorithm suitable for small $\varepsilon$. We state Shamir’s running time in Table 1. Unfortunately, Shamir’s result is (1) not gap-free; (2) not accelerated (i.e., does not match the $\text{gap}^{-0.5}$ dependence comparing to block Krylov); and (3) requires a very accurate warm-start that in principle can take a very long time to compute.

The third breakthrough is to obtain running times of the form $\tilde{O}(\text{nnz}(A) + \text{poly}(k, 1/\varepsilon) \cdot (n + d))$ [3, 4], see Table 2. We call them NNZ running times. To obtain such results, one usually needs sub-sampling on the matrix and thus incurs a poor dependence on $\varepsilon$. For this reason, to make the best use of NNZ type of results, one usually focuses on improving the dependence on $1/\varepsilon$. To this end, Bhojanapalli et al. [3] provide a $1/\varepsilon^2$ result using alternating minimization. Since $1/\varepsilon^2$ also shows up in the sampling complexity, we believe the quadratic dependence on $\varepsilon$ is tight among NNZ types of algorithms.

All the cited results rely on ad-hoc non-convex optimization techniques together with matrix algebra, which make their final proofs complicated. Furthermore, Shamir’s result [16] only works if a very accurate (i.e., $1/\text{poly}(d)$-accurate) warm start is given, and the time needed to find such a warm start remains unclear.

In this paper, we develop a new algorithmic framework to solve $k$-SVD. It not only improves the aforementioned breakthroughs, but also relies only on simple convex analysis so do not require a warm start.

**Other Related Work.** Some authors focus on the streaming model of 1-SVD [10, 12]. These algorithms are slower than off-line methods. We also acknowledge that, unlike $k$-SVD, accelerated stochastic methods were previously known for 1-SVD [7, 8].

### 1.1 Overview of Our Result and the Settlement of an Open Question

Our algorithmic framework is extremely simple: instead of computing all $k$ singular vectors together like all recent breakthroughs, we find singular vectors one at a time, for $k$ iterations in total.

A naive implementation of this idea results in a running time that depends on the intermediate eigengaps between all first $k$ singular values [13, 16]. The situation becomes worse if singular values form clusters, and even worse if one wants to obtain a gap-free result. To the best of our
| Paper             | Running time                                                                 | GF? | Stoc? | Acc? |
|-------------------|-------------------------------------------------------------------------------|-----|-------|------|
| subspace PM [14]  | \(\tilde{O}\left(\frac{\text{nnz}(A)}{\epsilon} + \frac{k^2d}{\epsilon}\right)\) | yes | no    | no   |
|                   | \(\tilde{O}\left(\frac{\text{nnz}(A)}{\text{gap}} + \frac{k^2d}{\text{gap}}\right)\) | no  |       |      |
| block Krylov [14] | \(\tilde{O}\left(\frac{\text{nnz}(A)}{\sqrt{\epsilon/2}} + \frac{k^2d}{\epsilon} + \frac{k^3}{\sqrt{\epsilon/2}}\right)\) | yes | no    | yes  |
|                   | \(\tilde{O}\left(\frac{\text{nnz}(A)}{\text{gap}/\sqrt{2}} + \frac{k^2d}{\text{gap}} + \frac{k^3}{\text{gap}/\sqrt{2}}\right)\) | no  |       |      |
| LazySVD           | \(\tilde{O}\left(\frac{\text{nnz}(A)}{\sqrt{\epsilon/2}} + \frac{k^2d}{\epsilon}\right)\) | yes | no    | yes  |
| Corollary 4.3 and 4.4 | \(\tilde{O}\left(\frac{\text{nnz}(A)}{\text{gap}/\sqrt{2}} + \frac{k^2d}{\text{gap}}\right)\) | no  |       |      |
| Shamir [16]       | \(\tilde{O}(\text{kn}d + \frac{k^4d}{\sigma^4 \text{gap}})\) (local convergence only) | no  | yes   | no   |
| LazySVD           | \(\tilde{O}(\text{kn}d + \frac{k^3\sqrt{d}}{\sigma_k^2 \text{gap} \epsilon^{1/2}})\) | yes | yes   | yes  |
| Corollary 4.3 and 4.4 | \(\tilde{O}(\text{kn}d + \frac{k^3\sqrt{d}}{\epsilon \sigma_k^2 \text{gap}^{1/2}})\) | no  |       |      |

All GF results above provide \((1 + \epsilon)\|\Delta\|_2\) spectral and \((1 + \epsilon)\|\Delta\|_F\) Frobenius guarantees.

Table 1: Performance comparison among direct methods. Define \(\text{gap} = (\sigma_k - \sigma_{k+1})/\sigma_k \in [0, 1]\). GF = Gap Free; Stoc = Stochastic; Acc = Accelerated. Stochastic results in this table are assuming \(\|a_i\|_2 \leq 1\) following (1.1).

Knowledge, running time depending on intermediate gaps is the only known result behind this type of algorithms, and even thought necessary by some authors [13]. Furthermore, Musco and Musco [14] explicitly stated it as an open question to design “small-block” or even “single-vector” algorithms like ours to obtain a better running time (i.e., time independent of intermediate gaps).

In this paper, we answer this open question in full. We carefully specify the single-vector computation routine, and provide novel analyses where the convergence neither depends on intermediate eigengaps, nor on the \(k\)-th eigengap. This yields our gap-free result. We also obtain gap-dependent results for free because all gap-free results imply gap-dependent ones. As for how to find individual singular vectors in each of the \(k\) iterations, we use the recent advances of shift-and-inverse preconditioning technique developed in [7, 8], and reduce the problem to convex optimization that can be solved either with accelerated gradient descent (AGD) or accelerated SVRG. The former leads to faster accelerated \(k\)-SVD algorithm outperforming block Krylov, and the latter leads to faster accelerated and stochastic \(k\)-SVD algorithm outperforming Shamir. See Table 1 for a detailed comparison.

In terms of NNZ running time, somewhat surprisingly, we show if one sub-samples \(A\) and then applies our new \(k\)-SVD algorithm, the resulting running time becomes \(\tilde{O}(\text{nnz}(A) + \text{poly}(k, 1/\epsilon) \cdot d)\) where the polynomial dependence on \(\epsilon\) is quadratic. This improves upon [3] in certain (but sufficiently interesting) parameter regimes, see Table 2, but completely avoids the use of alternating minimization.

Finally, besides the running time advantages above, our algorithm also works when \(k\) is not known to the algorithm, as opposed to block power methods or Krylov which need to know \(k\) in advance. We call our algorithm LazySVD and discuss its running time formally in the subsequent sections.
2 Preliminaries

Given a matrix $A$ we denote by $\|A\|_2$ and $\|A\|_F$ respectively the spectral and Frobenius norms of $A$. For $q \geq 1$, we denote by $\|A\|_q$, the Schatten-$q$ norm of $A$. We write $A \succeq B$ if $A, B$ are symmetric and $A - B$ is positive semi-definite (PSD). We denote by $\lambda_k(M)$ the $k$-th largest eigenvalue of a symmetric matrix $M$, and $\sigma_k(A)$ the $k$-th largest singular value of a rectangular matrix $A$. It is clear that $\lambda_k(AA^\top) = (\sigma_k(A))^2$. We often denote by $\sigma_1 \geq \cdots \sigma_d \geq 0$ the singular values of $A \in \mathbb{R}^{d \times n}$, by $\lambda_1 \geq \cdots \lambda_d \geq 0$ the eigenvalues of $M = AA^\top \in \mathbb{R}^{d \times d}$. (Although $A$ may have fewer than $d$ singular values for instance when $n < d$, if this happens, we append zeros in the end.) We denote by $A_k^*$ the best rank-$k$ approximation of $A$.

We use $\perp$ to denote the orthogonal complement of a matrix. More specifically, given a column orthonormal matrix $U \in \mathbb{R}^{d \times k}$, we define $U \perp \defeq \{x \in \mathbb{R}^d \mid U^\top x = 0\}$. For notational simplicity, we sometimes also denote $U \perp$ as a $d \times (d - k)$ matrix consisting of some basis of $U \perp$.

**Theorem 2.1** (approximate matrix inverse). Given $d \times d$ matrix $M \succeq 0$, and constants $\lambda, \delta > 0$ satisfying $\lambda I - M \succeq \delta I$, one can minimize the quadratic $f(x) \defeq x^\top (\lambda I - M) x - b^\top x$ in order to invert $(\lambda I - M)^{-1}b$. Suppose the desired accuracy is $\|x - (\lambda I - M)^{-1}b\| \leq \varepsilon$. Then,

- Accelerated gradient descent (AGD) produces such an output $x$ in $O\left(\frac{\lambda^{1/2}}{\delta^{1/4}} \log \frac{1}{\varepsilon} \right)$ iterations, each requiring $O(d)$ time plus the time needed to multiply $M$ with a vector.
- If $M$ is given in the form $M = \frac{1}{n} \sum_{i=1}^n a_i a_i^\top$ and $\|a_i\|_2 \leq 1$, then accelerated SVRG (see for instance [1]) produces such an output $x$ in time $O\left(\max\{nd, \frac{n^{3/4}d^{1/4}}{\delta^{1/2}}\} \log \frac{1}{\varepsilon} \right)$.

3 Shift-and-Inverse PCA, Revisited

In this section, we define AppxPCA, the (multiplicative-)approximation algorithm for computing the leading eigenvector of a symmetric matrix using the shift-and-inverse routine [7, 8]. Our pseudo-code in Algorithm 1 is a modification of Algorithm 5 that appeared in [7]. Since we need a more refined running time statement in this paper in terms of multiplicative error guarantees, and since the stated proof in [7] is anyways only a sketched one, we choose to carefully reprove a similar result of [7] (details in Appendix A) and state the following theorem:

**Theorem 3.1** (AppxPCA). Let $M \in \mathbb{R}^{d \times d}$ be a symmetric matrix with eigenvalues $1 \geq \lambda_1 \geq \cdots \geq \lambda_d \geq 0$ and corresponding eigenvectors $u_1, \ldots, u_d$. With probability at least $1 - p$, AppxPCA produces
Algorithm 1 \texttt{AppxPCA}(\mathcal{A}, M, \delta_x, \varepsilon, p)

\textbf{Input:} \mathcal{A}, an approximate matrix inversion method; \(M \in \mathbb{R}^{d \times d}\), a symmetric matrix satisfying  
\(0 \preceq M \preceq I; \delta_x \in (0, 0.5]\), a multiplicative error; \(\varepsilon \in (0, 1]\), a numerical accuracy parameter; and \(p \in (0, 1]\), the confidence parameter. \diamond \ \text{running time only logarithmically depends on } 1/\varepsilon \text{ and } 1/p.

1: \(m_1 \leftarrow \left[4 \log \left(\frac{288d}{\delta_x^2} \right)\right], m_2 \leftarrow \left\lceil \log \left(\frac{26d}{\varepsilon \delta_x^2} \right)\right\rceil\);  
\quad \diamond \ m_1 = T_{PM}(8, 1/32, p) \text{ and } m_2 = T_{PM}(2, \varepsilon/4, p) \text{ using definition in Lemma A.1.}

2: \(\bar{\varepsilon}_1 \leftarrow \frac{1}{64m_1}(\delta_x^2)^{m_1} \) and \(\bar{\varepsilon}_2 \leftarrow \frac{1}{8m_2}(\delta_x^2)^{m_2}\).

3: \(\tilde{w}_0 \leftarrow \text{a random unit vector}; s \leftarrow 0; \lambda(0) \leftarrow 1 + \delta_x;\)

4: \textbf{repeat}
5: \quad s \leftarrow s + 1;
6: \quad \textbf{for } t = 1 \textbf{ to } m_1 \textbf{ do}
7: \quad \quad \text{Apply } \mathcal{A} \text{ to find } \tilde{w}_t \text{ satisfying } \|\tilde{w}_t - (\lambda^{(s-1)}I - M)^{-1}\tilde{w}_{t-1}\| \leq \bar{\varepsilon}_1;
8: \quad \textbf{end for}
9: \quad w \leftarrow \tilde{w}_{m_1}/\|\tilde{w}_{m_1}\|;
10: \quad \text{Apply } \mathcal{A} \text{ to find } v \text{ satisfying } \|v - (\lambda^{(s-1)}I - M)^{-1}w\| \leq \bar{\varepsilon}_1;
11: \quad \Delta^{(s)} \leftarrow \frac{1}{2}w^\top(v - \bar{\varepsilon}_1) \text{ and } \lambda^{(s)} \leftarrow \lambda^{(s-1)} - \Delta^{(s)} / 2;
12: \quad \textbf{until } \Delta^{(s)} \leq \frac{\delta_x \lambda^{(s)}}{3}\)
13: \quad f \leftarrow s;
14: \quad \textbf{for } t = 1 \textbf{ to } m_2 \textbf{ do}
15: \quad \quad \text{Apply } \mathcal{A} \text{ to find } \tilde{w}_t \text{ satisfying } \|\tilde{w}_t - (\lambda^{(f)}I - M)^{-1}\tilde{w}_{t-1}\| \leq \bar{\varepsilon}_2;
16: \quad \textbf{end for}
17: \quad \textbf{return } w \leftarrow \tilde{w}_{m_2}/\|\tilde{w}_{m_2}\|.

an output \(w\) satisfying  
\[
\sum_{i \in [d], \lambda_i \leq (1 - \delta_x)\lambda_1} (w^\top u_i)^2 \leq \varepsilon \quad \text{and} \quad w^\top Mw \geq (1 - \delta_x)(1 - \varepsilon)\lambda_1.
\]

Furthermore, the total number of oracle calls to \(\mathcal{A}\) is \(O(\log(1/\delta_x)m_1 + m_2)\), and each time we call \(\mathcal{A}\) we have  
\[
\frac{\lambda^{(s)}}{\lambda_{\min}(\lambda^{(s)}I - M)} \leq \frac{12}{\delta_x} \quad \text{and} \quad \frac{1}{\lambda_{\min}(\lambda^{(s)}I - M)} \leq \frac{12}{\delta_x \lambda_1}.
\]

The stated conditions \[
\frac{\lambda^{(s)}}{\lambda_{\min}(\lambda^{(s)}I - M)} \leq \frac{12}{\delta_x} \quad \text{and} \quad \frac{1}{\lambda_{\min}(\lambda^{(s)}I - M)} \leq \frac{12}{\delta_x \lambda_1}
\]
immediately imply the following running time of \texttt{AppxPCA} owing to Theorem 2.1:

\textbf{Corollary 3.2.}

\begin{itemize}
  \item If \(\mathcal{A}\) is AGD, the running time of \texttt{AppxPCA} is \(\widetilde{O}\left(\frac{1}{\delta_x^{1/2}}\right)\) multiplied with \(O(d)\) plus the time needed to multiply \(M\) with a vector.
  \item If \(M = \frac{1}{n} \sum_{i=1}^{n} a_i a_i^\top\) where each \(\|a_i\|_2 \leq 1\), and \(\mathcal{A}\) is accelerated SVRG, then the total running time of \texttt{AppxPCA} is \(\widetilde{O}\left(\max\{nd, \frac{n^{3/4}d}{\lambda_1^{1/2} \delta_x^{1/2}}\}\right)\).
\end{itemize}

\section{Our Main Algorithm and Theorems}

Our algorithm \texttt{LazySVD} is formally stated in Algorithm 2. It simply applies \(k\) times \texttt{AppxPCA}, each time with a multiplicative error factor \(\delta_x/2\), and projects the matrix \(M\) into the orthogonal space with respect to the obtained leading eigenvector.
Algorithm 2 LazySVD(A, M, k, δ, ε_pca, p)

Input: A, an approximate matrix inversion method; M ∈ R^{d×d}, a matrix satisfying 0 ⪯ M ⪯ I; k ∈ [d], the desired rank; δ ∈ (0, 1), a multiplicative error; ε_pca ∈ (0, 1), a numerical accuracy parameter; and p ∈ (0, 1), a confidence parameter.

1: M_0 ← M;
2: V_0 = [];
3: for s = 1 to k do
4: v'_s ← AppxPCA(A, M_{s-1}, δ, ε_pca, p/k);
5: v_s ← ((I - V_{s-1} V_{s-1}^\top) v'_s) / ∥(I - V_{s-1} V_{s-1}^\top) v'_s∥;
6: V_s ← [V_{s-1}, v_s];
7: M_s ← (I - v_s v_s^\top) M_{s-1} (I - v_s v_s^\top);
8: end for
9: return V_k.

4.1 Our Main Theorems

We state our main approximation and running time theorems of LazySVD below, and then provide corollaries to translate them into gap-dependent and gap-free results for k-SVD.

Theorem 4.1 (approximation). Let M ∈ R^{d×d} be a symmetric matrix with eigenvalues 1 ≥ λ_1 ≥ ⋯ ≥ λ_d ≥ 0 and corresponding eigenvectors u_1, ⋯, u_d. Let k ∈ [d], let δ, p ∈ (0, 1), and let ε_pca ≤ poly(ε, δ, 1/λ_{k+1}^{3/2}) Then, LazySVD outputs a (column) orthonormal matrix V_k = (v_1, ⋯, v_k) ∈ R^{d×k} which, with probability at least 1 - p, satisfies all of the following properties. (Denote by M_k = (I - V_k V_k^\top) M (I - V_k V_k^\top).)

(a) Core lemma: ∥V_k^\top U∥ ≤ ε, where U = (u_j, ⋯, u_d) is the (column) orthonormal matrix and j is the smallest index satisfying λ_j ≤ (1 - δ/2)∥M_{k-1}∥2.

(b) Spectral norm guarantee: λ_{k+1} ≤ ∥M_k∥2 ≤ 1/δ/2.

(c) Rayleigh quotient guarantee: (1 - δ)λ_k ≤ v_k^\top M v_k ≤ 1/δ/2 λ_k.

(d) Schatten-q norm guarantee: for every q ≥ 1, we have ∥M_k∥q ≤ (1+δ/2)^2 (∑_{i=k+1}^d λ_i^q)^{1/q}.

We defer the proof of Theorem 4.1 to the appendix, but highlight the main ideas and techniques behind the proof in Section 4.3. Below we state the running time of LazySVD.

Theorem 4.2 (running time). LazySVD can be implemented to run in time

- O(kn\sqrt{(M+k^2d)/δ}) if A is AGD and M ∈ R^{d×d} is given explicitly;
- O(kn\sqrt{(M+k^2d)/δ}) if A is AGD and M is given as M = A A^\top where A ∈ R^{d×n}; or
- O(k n d + kn^{3/4}/λ_k^{1/4}d^{1/2}) if A is accelerated SVRG and M = 1/n \sum_{i=1}^n a_i a_i^\top where each ∥a_i∥2 ≤ 1.

Above, the O notation hides logarithmic factors with respect to k, d, 1/δ, 1/p, 1/λ_1, λ_1/λ_k.

The detailed specifications of ε_pca can be found in the appendix where we restate the theorem more formally. To provide the simplest proof, we have not tightened the polynomial factors in the theoretical upper bound of ε_pca because the running time depends only logarithmic on 1/ε_pca.
Proof of Theorem 4.2. We call $k$ times \texttt{AppxPCA}, and each time we can feed $M_{s-1} = (I - V_{s-1}V_{s-1}^\top)M(I - V_{s-1}V_{s-1}^\top)$ implicitly into \texttt{AppxPCA} thus the time needed to multiply $M_{s-1}$ with a $d$-dimensional vector is $O(dk + \text{nnz}(M))$ or $O(dk + \text{nnz}(A))$. Here, the $O(dk)$ overhead is due to the projection of a vector into $V_{s-1}$. This proves the first two running times using Corollary 3.2.

To obtain the third running time, when we compute $M_s$ from $M_{s-1}$, we explicitly project $a_i' \leftarrow (I - v_i v_i^\top)a_i$ for each vector $a_i$, and feed the new $a_1', \ldots, a_n'$ into \texttt{AppxPCA}. Now the running time follows from the second part of Corollary 3.2 together with the fact that $\|M_{s-1}\|_2 \geq \|M_k\|_2 \geq \lambda_k$.

4.2 Our Gap-Dependent and Gap-Free Results

Our main theorems imply the following corollaries (proved in Appendix C.1 for completeness).

Corollary 4.3 (Gap-dependent $k$-SVD). Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $1 \geq \sigma_1 \geq \cdots \sigma_d \geq 0$ and the corresponding left singular vectors $u_1, \ldots, u_d \in \mathbb{R}^d$. Let $\text{gap} = \frac{\sigma_k - \sigma_{k+1}}{\sigma_k}$ be the relative gap. For fixed $\epsilon, p > 0$, consider the output

$$V_k \leftarrow \text{LazySVD} \left( A, AA^\top, k, \text{gap}, O \left( \frac{\epsilon^4 \text{gap}^2}{k^4 (\sigma_1/\sigma_k)^4} \right), p \right).$$

Then, defining $W = (u_{k+1}, \ldots, u_d)$, we have with probability at least $1 - p$:

$$V_k \text{ is a rank-} k \text{ (column) orthonormal matrix with } \|V_k^\top W\|_2 \leq \epsilon.$$

Our running time is $\tilde{O}\left( \frac{\text{nnz}(A) + k^2 d}{\sqrt{\epsilon \text{gap}}} \right)$, or time $\tilde{O}\left( kn d + \frac{kn^{3/4} d}{\sqrt{\epsilon \text{gap}}} \right)$ in the stochastic setting (1.1).

Above, both running times depend only logarithmically on $1/\epsilon$ so enjoy linear convergence rates.

Corollary 4.4 (Gap-free $k$-SVD). Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $1 \geq \sigma_1 \geq \cdots \sigma_d \geq 0$. For fixed $\epsilon, p > 0$, consider the output

$$(v_1, \ldots, v_k) = V_k \leftarrow \text{LazySVD} \left( A, AA^\top, k, \frac{\epsilon}{3}, O \left( \frac{\epsilon^6}{k^4 d^2 (\sigma_1/\sigma_{k+1})^2} \right), p \right).$$

Then, defining $A_k = V_k V_k^\top A$ which is a rank $k$ matrix, we have with probability at least $1 - p$:

1. Spectral norm guarantee: $\|A - A_k\|_2 \leq (1 + \epsilon)\|A - A_k^*\|_2$;
2. Frobenius norm guarantee: $\|A - A_k\|_F \leq (1 + \epsilon)\|A - A_k^*\|_F$; and
3. Rayleigh quotient guarantee: $\forall i \in [k]$, $|v_i^\top AA^\top v_i - \sigma_i^2| \leq \epsilon \sigma_i^2$.

Running time is $\tilde{O}\left( \frac{\text{nnz}(A) + k^2 d}{\sqrt{\epsilon}} \right)$, or time $\tilde{O}\left( kn d + \frac{kn^{3/4} d}{\sqrt{\epsilon \text{gap}}} \right)$ in the stochastic setting (1.1).

Remark 4.5. The spectral and Frobenius guarantees we adopted are standard. It was observed that the spectral guarantee is more desirable than the Frobenius one in practice [14]. In fact, our algorithm implies for all $q \geq 1$, $\|A - A_k\|_S_q \leq (1 + \epsilon)\|A - A_k^*\|_S_q$ where $\|\cdot\|_S_q$ is the Schatten-$q$ norm. Rayleigh-quotient types of guarantee were introduced by Musco and Musco [14] for a more refined comparison. They showed that the block Krylov method satisfies $|v_i^\top AA^\top v_i - \sigma_i^2| \leq \epsilon \sigma_k^2$, which is slightly stronger than ours. However, these two guarantees are not much different in practice as we evidenced in our experiments.
4.3 Ideas and Techniques Behind Our Theorems

For sake of demonstrating the idea, we focus on the case when there is a (known) relative gap \(\sigma_k - \sigma_{k+1}/\sigma_k \in [0,1]\) between the \(k\)-th and the \((k+1)\)-th singular values of \(A\). Note that \texttt{LazySVD} can be equivalently viewed as follows: at iteration \(s\), \texttt{LazySVD} starts with a (column) orthonormal matrix \(V_{s-1} \in \mathbb{R}^{d \times (s-1)}\). It finds an approximate leading eigenvector \(v\) of \(M_{s-1} = (I - V_{s-1}V_{s-1}^\top)AA^\top (I - V_{s-1}V_{s-1}^\top)\), which is also approximately the leading eigenvector of \(M\) in the space \(V_{s-1}^\perp\). Then, \texttt{LazySVD} appends \(V_s \leftarrow [V_{s-1}, v]\) and continues to the next iteration.

**Obtain Faster Running Time.** Ideally, if each \(v\) were exactly the leading eigenvector of \(M_s\), then our final \(V_k\) would become exactly the top \(k\) singular vectors of \(A\). However, computing exact eigenvectors is too slow, so the main challenge is to tolerate as much error as possible to compute each \(v\), in order to tradeoff for a faster running time.

It was previously a folklore that one can approximately compute each \(v\) to a good precision so that the running time depends on all intermediate gaps \(\frac{\sigma_i}{\sigma_{i+1}}\) for \(i = 1, \ldots, k\). This is too slow, although thought to be somewhat necessary by some authors [13]. A weaker alternative is to compute each \(v\) to an additive precision so \(v^\top M_{s-1}v \geq \sigma_s^2 - \text{gap} \cdot \sigma_k^2\). However, this remains too slow because the running time would polynomially depend on \(\sigma_1/\sigma_k\) rather than \(1/\text{gap}\).

In \texttt{LazySVD} we tolerate the \(s\)-th leading eigenvalue computation to suffer from a multiplicative error \(\text{gap}\) —or loosely speaking, to satisfy \(v^\top M_{s-1}v \geq (1 - \text{gap})\sigma_s^2\). This turns out to imply our declared running time in Table 1 if one can prove the correctness of our algorithm, that is, if one can prove \(\|M_{s-1}\|_2 \approx \sigma_s^2\).

**Obtain Correctness.** Our main idea is to use the fact that each vector \(v\) “approximately” lies in the span of the top \(k\) eigenvectors of \(M = AA^\top\).

Notice if each \(v\) were perfectly lying in the span of the top \(k\) eigenvectors of \(M\), we would be able to claim —using the Cauchy interlacing theorem— that the \((k+1-s)\)-th largest eigenvalue of matrix \(M_s\) would be never larger than \(\sigma_{k+1}^2\). In other words, the difference between the largest and the \((k+1-s)\)-th eigenvalue of \(M_s\) would be at least \(\sigma_s^2 - \sigma_{k+1}^2\). For this reason, one could continue to apply a “gap-multiplicative error” algorithm to obtain the next leading eigenvector \(v\), and this \(v\) would “almost completely” lie in the span of the top \((k-s)\) eigenvectors of \(M_s\) and the span of the top \(k\) eigenvectors of \(M\). Repeating this argument for \(k\) times, we could have obtained the top \(k\) singular vectors of \(A\) exactly, up to rotation.

Our main technique contribution is to extend the above argument into an approximate setting, and to propagate error “moderately” across iterations. While a naive bound could easily blow up the error exponentially with respect to \(k\), we provide non-trivial analysis to show that the error grows at most linearly in \(k\). This step of our proof essentially consists of two parts.

Part 1: we apply a modified theorem of [7] to show that, in each iteration, the vector \(v\) obtained from \texttt{AppxPCA} only correlates with the last \(d-k\) eigenvectors of \(M_s\) by a polynomially small factor.

Part 2: we develop a gap-free variant of the Wedin theorem for matrices [18], which translates Part 1 into two statements. On one hand, the last \(d-k\) eigenvectors of \(M_s = (I - vv^\top)M_{s-1}(I - vv^\top)\) approximately lie in the span of the last \(d-k\) eigenvectors of \(M_{s-1}\). On the other hand, the \((k+1-s)\)-th eigenvalue of \(M_s\) is roughly the same as the \((k+2-s)\)-th eigenvalue of \(M_{s-1}\). Recursively applying this \(s\) times, we conclude that the last \(d-k\) eigenvectors of \(M_s\) approximately lie in the span of the last \(d-k\) eigenvectors of \(M_0 = M\), and the \((k+1-s)\)-th eigenvalue of \(M_s\) is roughly the \((k+1)\)-th eigenvalue of \(M_0 = M\) which is \(\sigma_{k+1}^2\). These two properties together ensure the correctness and ensure that we can safely move to the next iteration.
5 NNZ running time

In this section, we translate our results in the previous section into the $O(\text{nnz}(A) + \text{poly}(k, 1/\varepsilon)(n + d))$ running-time statements. The idea is surprisingly simple: we sample either random columns of $A$, or random entries of $A$, and then apply LazySVD to compute the $k$-SVD. Such translation directly gives either $1/\varepsilon^{2.5}$ results if AGD is used as the convex subroutine and either column or entry sampling is used, or a $1/\varepsilon^2$ result if accelerated SVRG and column sampling are used together.

Due to space limitation, we only informally state our theorem and defer all the details to Appendix D.

Theorem 5.1 (informal). Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. For every $\varepsilon \in (0, 1/2)$, one can apply LazySVD with appropriately chosen $\delta_x$ on a “carefully sampled version” of $A$. Then, the resulting matrix $V \in \mathbb{R}^{d \times k}$ can satisfy

- spectral norm guarantee: $\|A - VV^\top A\|_2 \leq \|A - A_k^*\|_2 + \varepsilon \|A - A_k^*\|_F$,
- Frobenius norm guarantee: $\|A - VV^\top A\|_F \leq (1 + \varepsilon)\|A - A_k^*\|_F$.

The total running time depends on (1) whether column or entry sampling is used, (2) which matrix inversion routine $A$ is used, and (3) whether spectral or Frobenius guarantee is needed. We list our deduced results in Table 2 and the formal statements can be found in Theorem D.4, D.6, and D.9.

Remark 5.2. The main purpose of our NNZ results is to demonstrate the strength of LazySVD in terms of improving the $\varepsilon$ dependency. We have not tried very hard, and believe it possible, to improve the polynomial dependence with respect to $k$ or $(\sigma_1/\sigma_{k+1})$. Also, somewhat surprisingly, in our analysis Frobenius norms become harder to minimize as opposed to spectral norms; this is in contrast to known literatures where usually Frobenius results are easier to prove [3].

6 Experiment

In this section we demonstrate the practicality of our SVD decomposition framework. We implement an iterative algorithm to compute approximate leading eigenvector $k$ times, and compare it to block power method or block Krylov method. Notice that in theory, the best worse-cast complexity for approximate leading eigenvector computation is obtained by AppxPCA on top of AGD or accelerated SVRG. However, in practice, Lanczos method runs much faster than these shift-and-inverse based methods and therefore we adopt Lanczos method as the method of choice to replace AppxPCA.

Datasets. We use datasets SNAP/amazon0302, SNAP/email-enron, and news20 that were also used by Musco and Musco [14], as well as an additional but famous dataset RCV1. The first two can be found on the SNAP website [11] and the last two can be found on the LibSVM website [6]. The four datasets give rise sparse matrices of dimensions $257570 \times 262111$, $35600 \times 16507$, $11269 \times 53975$, and $20242 \times 47236$ respectively.

Implemented Algorithms. For the block Krylov method, it is a well-known issue that the Lanczos type of three-term recurrence update is numerically unstable. This is why Musco and Musco [14] only used the stable variant of block Krylov which requires an orthogonalization of each $n \times k$ matrix with respect to all previously obtained $n \times k$ matrices. This greatly improves the numerical stability albeit sacrificing running time. We implement both these algorithms. In sum, we have implemented:

---

4 This is the best known spectral guarantee one can obtain using NNZ running time [3]. It is an open question whether the stricter $\|A - VV^\top A\|_2 \leq (1 + \varepsilon)\|A - A_k^*\|_2$ type of spectral guarantee is possible.
We make the following observations:

- **PM**: block power method for $T$ iterations.
- **Krylov**: stable block Krylov method for $T$ iterations [14].
- **Krylov(unstable)**: the three-term recurrence implementation of block Krylov for $T$ iterations.
- **LazySVD**: $k$ calls of the vanilla Lanczos method, and each call runs $T$ iterations.

**A Fair Running-Time Comparison.** For a fixed integer $T$, the four methods go through the dataset (in terms of multiplying $A$ with column vectors) the same number of times. However, since LazySVD does not need block orthogonalization (as needed in PM and Krylov) and does not need a $(T k)$-dimensional SVD computation in the end (as needed in Krylov), the running time of LazySVD is clearly much faster for a fixed value $T$. We therefore compare the performances of the four methods in terms of running time rather than $T$.

We programmed the four algorithms using the same programming language with the same sparse-matrix implementation. We tested them single-threaded on the same Intel i7-3770 3.40GHz personal computer. As for the final low-dimensional SVD decomposition step at the end of the PM or Krylov method (which is not needed for our LazySVD), we used a third-party library that is built upon the x64 Intel Math Kernel Library so the time needed for such SVD is maximally reduced.

**Performance Metrics.** We compute four metrics on the output $V = (v_1, \ldots, v_k) \in \mathbb{R}^{n \times k}$:

- **Fnorm**: relative Frobenius norm error: $(\| A - V V^\top A \|_F - \| A - A_k^* \|_F)/\| A - A_k^* \|_F$.
- **spectral**: relative spectral norm error: $(\| A - V V^\top A \|_2 - \| A - A_k^* \|_2)/\| A - A_k^* \|_2$.
- **rayleigh(last)**: Rayleigh quotient error relative to $\sigma_{k+1}$: $\max_{j=1}^{k} |\sigma_j^2 - v_j^\top A A^\top v_j|/\sigma_{k+1}^2$.
- **rayleigh**: relative Rayleigh quotient error: $\max_{j=1}^{k} |\sigma_j^2 - v_j^\top A A^\top v_j|/\sigma_j^2$.

The first three metrics were also used by Musco and Musco [14]. We added the fourth one because our theory only predicted convergence with respect to the fourth but not the third metric. However, we observe that in practice they are not much different from each other.

**Our Results.** We study four datasets each with $k = 10, 20, 30$ and with the four performance metrics, totaling 48 plots. Due to space limitation, we only select six representative plots out of 48 and include them in Figure 1. (The full plots can be found in Figure 2, 3, 4 and 5 in the appendix.) We make the following observations:

- **LazySVD** outperforms its three competitors almost universally.
• Krylov(unstable) outperforms Krylov for small value $T$; however, it is less useful for obtaining accurate solutions due to its instability. (The dotted green curves even go up if $T$ is large.)

• Subspace power method performs the slowest unsurprisingly due to its lack of acceleration.

Acknowledgements

We thank Ohad Shamir for listening to our ideas and pointing us to some of the existing results. Part of this work is supported by a Microsoft Research Grant, no. 0518584.
APPENDIX

A Proof Details for Theorem 3.1: Convergence of AppxPCA

A.1 Inexact Power Method

Consider the classical power method that is to start with a random unit vector \( w_0 \) and apply \( w_t \leftarrow Mw_{t-1}/\|Mw_{t-1}\| \) iteratively.

Lemma A.1 (Exact Power Method). Let \( M \) be a PSD matrix with eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_d \) and the corresponding eigenvectors \( u_1, \ldots, u_d \). Fix an error tolerance \( \varepsilon > 0 \), parameter \( \kappa \geq 1 \), and failure probability \( p > 0 \), define

\[
T_{\text{PM}}(\kappa, \varepsilon, p) = \left\lceil \frac{\kappa}{2} \log \left( \frac{9d}{p^2 \varepsilon} \right) \right\rceil
\]

Then, with probability at least \( 1 - p \) it holds that \( \forall t \geq T_{\text{PM}}(\kappa, \varepsilon, p) \):

\[
\sum_{i \in [d], \lambda_i \leq (1 - 1/\kappa) \lambda_1} (w_t^\top u_i)^2 \leq \varepsilon \quad \text{and} \quad w_t^\top Mw_t \geq (1 - 1/\kappa - \varepsilon) \lambda_1 .
\]

The probability of success depends only on the random variable \( (w_0^\top u_1)^2 \).

Proof. For each \( i \in [d] \),

\[
(w_t^\top u_i)^2 = \frac{(M^t w_0)^\top u_i}{\|M^t w_0\|^2} = \frac{(w_0^\top M^t u_i)^2}{w_0^\top M^{2t} u_0} = \frac{\lambda_i^{2t} (w_0^\top u_i)^2}{\sum_{j=1}^d \lambda_j^{2t} (w_0^\top u_j)^2} = \frac{(w_0^\top u_i)^2}{\sum_{j=1}^d \left( \frac{\lambda_j}{\lambda_1} \right)^{2t} (w_0^\top u_j)^2}
\]

Since \( w_0 \) is a random unit vector, according to for instance Lemma 5 of [2], it holds with probability at least \( 1 - p \) that \( (w_0^\top u_1)^2 \geq \frac{p^2}{9d} \). Substituting this into the above inequality, we conclude that with probability at least \( 1 - p \), for all \( i \in [d] \), we have

\[
(w_t^\top u_i)^2 \leq (w_0^\top u_i) \cdot \frac{9d}{p^2} \left( 1 - \frac{\lambda_i}{\lambda_1} \right)^{2t} \leq (w_0^\top u_i) \cdot \frac{9d}{p^2} \cdot \exp \left(-2 \frac{\lambda_i - \lambda_1}{\lambda_1} t \right)
\]

As a result, for every \( t \geq T_{\text{PM}}(\kappa, \varepsilon, p) \), and every \( i \) such that \( \lambda_i \leq (1 - 1/\kappa) \lambda_1 \) (which implies \( \lambda_i - \lambda_1 \geq 1/\kappa \)), we have

\[
(w_t^\top u_i)^2 \leq \varepsilon \cdot (w_0^\top u_i)^2
\]

Summing them up we have

\[
\sum_{i \in [d], \lambda_i \leq (1 - 1/\kappa) \lambda_1} (w_t^\top u_i)^2 \leq \varepsilon \sum_{i \in [d]} (w_0^\top u_i)^2 = \varepsilon .
\]

This finishes the proof of the first bound. To prove the second bound, we compute that

\[
w_t^\top Mw_t \geq \sum_{i=1}^d \lambda_i (w_t^\top u_i)^2 \geq \sum_{i \in [d], \lambda_i > (1 - 1/\kappa) \lambda_1} \lambda_i (w_t^\top u_i)^2 \geq (1 - 1/\kappa) \lambda_1 \cdot \sum_{i \in [d], \lambda_i > (1 - 1/\kappa) \lambda_1} (w_t^\top u_i)^2 \geq (1 - 1/\kappa)(1 - \varepsilon) \lambda_1 \geq (1 - 1/\kappa - \varepsilon) \lambda_1 .\]

□
Lemma A.2 (Lemma 4.1 of [7]). Let $M$ be a PSD matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$. Fix an accuracy parameter $\tilde{\varepsilon} > 0$, and consider two update sequences

$$\hat{w}_0^* = w_0, \quad \forall t \geq 1: \hat{w}_t^* \leftarrow M\hat{w}_{t-1}^*$$

$$\hat{w}_0 = w_0, \quad \forall t \geq 1: \hat{w}_t \text{ satisfies } \|\hat{w}_t - M\hat{w}_{t-1}\| \leq \tilde{\varepsilon},$$

Then, defining $w_t = \hat{w}_t/\|\hat{w}_t\|$ and $w_t^* = \hat{w}_t^*/\|\hat{w}_t^*\|$, it satisfies

$$\|w_t - w_t^*\| \leq \tilde{\varepsilon} \cdot \Gamma(M,t),$$

where

$$\Gamma(M,t) \overset{\text{def}}{=} \frac{2}{\lambda_d^2} \begin{cases} t, & \text{if } \lambda_1 = 1; \\ \left(\lambda_1^s - 1\right)/(\lambda_1 - 1), & \text{if } \lambda_1 \neq 1. \end{cases}$$

and we have $\Gamma(M,t) \leq 2t \cdot \max\{1, \lambda_1^t\}$.  

Theorem A.3 (Inaccurate Power Method). Let $M$ be a PSD matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ and the corresponding eigenvectors $u_1, \ldots, u_d$. With probability at least $1 - p$ it holds that, for every $\varepsilon \in (0,1)$ and every $t \geq T_{\text{PM}}(\kappa, \varepsilon/4, p)$, if $w_t$ is generated by the power method with per-iteration error $\tilde{\varepsilon} = \varepsilon/4t(M,t)$, then

$$\sum_{j \in [d], \lambda_j \leq (1-1/\kappa)\lambda_1} (w_t^\top u_j)^2 \leq \varepsilon \quad \text{and} \quad w_t^\top Mw_t \geq (1-1/\kappa - \varepsilon)\lambda_1.$$

Proof. Denoting by $w_t^*$ the output of power method with exact updates (with the same starting vector $w_0$ following Lemma A.2)

$$\sum_{j \in [d], \lambda_j \leq (1-1/\kappa)\lambda_1} (w_t^\top u_j)^2 \leq \sum_{j \in [d], \lambda_j \leq (1-1/\kappa)\lambda_1} (\langle w_t^*, u_j \rangle + \langle w_t - w_t^*, u_j \rangle)^2$$

$$\leq \sum_{j \in [d], \lambda_j \leq (1-1/\kappa)\lambda_1} 2(\langle w_t^*, u_j \rangle)^2 + 2(\langle w_t - w_t^*, u_j \rangle)^2$$

$$\leq \frac{\varepsilon}{2} + 2 \sum_{j \in [d]} (w_t - w_t^*)^\top u_j u_j^\top (w_t - w_t^*) = \frac{\varepsilon}{2} + 2\|w_t - w_t^*\|^2 \leq \varepsilon.$$

Above, the first inequality is because $(a + b)^2 \leq 2a^2 + 2b^2$, the second inequality is due to the definition of $w_t^*$ and Lemma A.1, and the last inequality is because $\|w_t - w_t^*\| \leq \tilde{\varepsilon} \cdot \Gamma(M,t) = \frac{\varepsilon}{4}$ which implies $2\|w_t - w_t^*\|^2 \leq \varepsilon^2/8 < \varepsilon/2$.

This finishes the proof of the first bound. The proof of the second bound is identical to the last paragraph of the proof of Lemma A.1. 

A.2 Proof of Theorem A.3

Lemma A.4. With probability at least $1 - p$, it holds that (where $\lambda_1$ is the largest eigenvalue of $M$.)

(a) $\tilde{\varepsilon}_1 \leq \frac{1}{32t((\lambda^{(s+1)} - 1)\lambda_1)}$ for each iteration $s$;

(b) $\tilde{\varepsilon}_2 \leq \frac{1}{4t((\lambda^{(s+1)} - 1)\lambda_1)}$;

(c) $0 \leq \frac{1}{2}(\lambda^{(s+1)} - 1) - \lambda_1 \leq \Delta^{(s)} \leq \lambda^{(s-1)} - \lambda_1$ and $\frac{1}{2}(\lambda^{(s+1)} - 1) - \lambda_1 \leq \lambda^{(s)} - \lambda_1$ for each iteration $s$;

and

(d) $\lambda^{(f)} - \lambda_1 \in \left[\frac{\delta_x \lambda_1}{12}, \delta_x \lambda_1\right]$ when the repeat-until loop is over.
Proof. We denote by $A^{(s)} \overset{\text{def}}{=} (\lambda^{(s)}I - M)^{-1}$ for notational simplicity. Below we prove all the items by induction for a specific iteration $s \geq 2$ assuming that the items of the previous $s - 1$ iterations are true. The base case of $s = 1$ can be verified similar to the general arguments below but requiring some non-trivial notational changes. We omitted the proof of the base case $s = 1$ in this paper.

(a) Recall that

$$
\Gamma(A^{(s-1)}, t) \leq 2t \cdot \frac{\max\{1, \lambda_{\max}(A^{(s-1)})^t\}}{\lambda_{\min}(A^{(s-1)})^t}
$$

On one hand, we have $\lambda_{\max}(A^{(s-1)}) = \frac{1}{\lambda^{(s-1)} - \lambda_1} \leq \frac{2}{\lambda^{(s-2)} - \lambda_1} \leq \frac{2}{\Delta^{(s-1)}}$ using Lemma A.4.c of the previous iteration. Combining this with the termination criterion $\Delta^{(s-1)} \geq \frac{\delta_x}{2} \lambda^{(s-1)}$, we have $\lambda_{\max}(A^{(s-1)}) \leq \frac{6}{\delta_x \lambda^{(s-1)}}$. On the other hand, we have $\lambda_{\min}(A^{(s-1)}) = \frac{1}{\lambda^{(s-1)} - \lambda_1} \geq \frac{1}{\lambda^{(s-1)}}$.

Combining the two bounds we conclude that $\Gamma(A^{(s-1)}, t) \leq 2t(6/\delta_x)^t$. It is now obvious that $\bar{\varepsilon}_1 \leq \frac{1}{32\Gamma(A^{(s-1)}, m_1)}$ is satisfied because $\bar{\varepsilon}_1 = \frac{1}{64m_1} \left(\frac{\delta_x}{6}\right)^{m_1}$.

(b) The same analysis as in the proof of Lemma A.4.a suggests that $\Gamma(A^{(f)}, t) \leq 2t(6/\delta_x)^t$. This immediately yields $\bar{\varepsilon}_2 \leq \frac{\varepsilon}{4\Gamma(A^{(f)}, m_2)}$ because $\bar{\varepsilon}_2 = \frac{\varepsilon}{8m_2} \left(\frac{\delta_x}{6}\right)^{m_2}$.

(c) Because Lemma A.4.a holds for the current iteration $s$ we can apply Theorem A.3 (with $\varepsilon = 1/8$ and $\kappa = 8$) and get

$$
w^\top A^{(s-1)} w \geq \frac{3}{4} \lambda_{\max}(A^{(s-1)})
$$

By the definition of $v$ in AppxPCA and the Cauchy-Schwartz inequality it holds that

$$
w^\top v = w^\top A^{(s-1)} w + w^\top (v - A^{(s-1)} w) \in [w^\top A^{(s-1)} w - \bar{\varepsilon}_1, w^\top A^{(s-1)} w + \bar{\varepsilon}_1]
$$

Combining the above two equations we have

$$w^\top v - \bar{\varepsilon}_1 \in \left[\frac{3}{4} \lambda_{\max}(A^{(s-1)}) - 2\bar{\varepsilon}_1, \lambda_{\max}(A^{(s-1)})\right] \subseteq \left[\frac{1}{2} \lambda_{\max}(A^{(s-1)}), \lambda_{\max}(A^{(s-1)})\right] = \left[\frac{1}{2}, 1\right] \cdot \frac{1}{\lambda^{(s-1)} - \lambda_1}.$$

In other words, $\Delta^{(s)} \overset{\text{def}}{=} \frac{1}{2} \cdot \frac{1}{w^\top v - \bar{\varepsilon}_1} \in \left[\frac{1}{2} (\lambda^{(s-1)} - \lambda_1), \lambda^{(s-1)} - \lambda_1\right]$.

At the same time, our update rule $\lambda^{(s)} = \lambda^{(s-1)} - \Delta^{(s)}/2$ ensures that $\lambda^{(s)} - \lambda_1 = \lambda^{(s-1)} - \lambda_1 - \Delta^{(s)}/2 \geq \lambda^{(s-1)} - \lambda_1 - \frac{\Delta^{(s)}/2}{\lambda^{(s-1)} - \lambda_1} = \frac{1}{2} (\lambda^{(s-1)} - \lambda_1)$.

(d) The upper bound holds because $\lambda^{(f)} - \lambda_1 = \lambda^{(f-1)} - \frac{\Delta^{(f)}}{2} - \lambda_1 \leq \frac{3}{2} \Delta^{(f)} \leq \frac{\delta_x}{2} \lambda^{(f)}$ where the first inequality follows from Lemma A.4.c of this last iteration, and the second inequality follows from our termination criterion $\Delta^{(f)} \leq \frac{\delta_x}{3} \lambda^{(f)}$. Simply rewriting this inequality we have $\lambda^{(f)} - \lambda_1 \leq \frac{\delta_x}{1 - \delta_x/2} \lambda_1 \leq \delta_x \lambda_1$.

The lower bound is because using Lemma A.4.c (of this and the previous iteration) we have $\lambda^{(f)} - \lambda_1 \geq \frac{1}{4} (\lambda^{(f-2)} - \lambda_1) \geq \frac{\Delta^{(f-1)}}{4} \geq \frac{\delta_x}{3} \lambda^{(f-1)} \geq \frac{\delta_x}{12} \lambda^{(f)}$. Here, inequality $\oplus$ is because $\Delta^{(f-1)} > \frac{\delta_x}{3} \lambda^{(f-1)}$ due to the termination criterion.
Finally since the success of Theorem A.3 only depends on the randomness of $\hat{w}_0$, we have that with probability at least $1 - p$ that all the above items are satisfied.

**Proof of Theorem 2.1.** It follows from Theorem A.3 (with $\kappa = 2$) that, letting $\mu_i = 1/(\lambda(f) - \lambda_i)$ be the $i$-th largest eigenvalue of the matrix $(\lambda(f)I - M)^{-1}$, then

$$\sum_{i \in [d]} (w^\top u_i)^2 \leq \varepsilon$$

Note that if an index $i \in [d]$ satisfies $\lambda_1 - \lambda_i \geq \delta_x \lambda_1$, then we must have $\lambda_1 - \lambda_i \geq \lambda(f) - \lambda_1$ owing to $\lambda(f) - \lambda_1 \leq \delta_x \lambda_1$ from Lemma A.4.d. This further implies that $2(\lambda(f) - \lambda_1) \leq \lambda(f) - \lambda_i$ and therefore $\mu_1/2 \geq \mu_i$. In sum, we also have

$$\sum_{i \in [d], \lambda_i \leq (1 - \delta_x)\lambda_1} (w^\top u_i)^2 \leq \varepsilon.$$

On the other hand,

$$w^\top M w = \sum_{i=1}^{d} \lambda_i (w^\top u_i)^2 \geq \sum_{i \in [d], \lambda_i \geq (1 - \delta_x)\lambda_1} \lambda_i (w^\top u_i)^2 \geq (1 - \delta_x)\lambda_1 \cdot \sum_{i \in [d], \lambda_i \geq (1 - \delta_x)\lambda_1} (w^\top u_i)^2 \geq (1 - \delta_x)(1 - \varepsilon)\lambda_1.$$

The number of oracle calls to $A$ is determined by the number of iterations in the repeat-until loop. It is easy to verify that there are at least $O(\log(1/\delta_x))$ such iterations, so the total number of oracle calls to $A$ is only $O(\log(1/\delta_x)m_1 + m_2)$.

In addition, each time we call $A$ we have

$$\frac{\lambda^{(s)}}{\lambda_{\min}(\lambda^{(s)}I - M)} \leq \frac{\lambda^{(s)}}{\lambda_1 - \lambda_1} \leq \frac{\lambda^{(s)}}{\Delta_{s+1}} \leq \frac{\lambda^{(s)}}{\lambda^{(s)}I - M} \leq \frac{\lambda^{(s)}}{\lambda^{(s)} - \lambda_1} \leq \frac{\lambda^{(s)}}{\Delta_{s+1}} \leq \frac{\lambda^{(s)}}{\lambda^{(s)}I - M} \leq \frac{\lambda^{(s)}}{\lambda_1} \leq \frac{12}{\delta_x}$$

if $s = 0$ then we have $\frac{\lambda^{(s)}}{\lambda^{(s)} - \lambda_1} \leq \frac{1 + \delta_x}{\delta_x}$ because $\lambda_1 \leq 1$. If $s \leq f - 2$ then we have $\frac{\lambda^{(s)}}{\lambda^{(s)} - \lambda_1} \leq \frac{\lambda^{(s)}}{\Delta_{s+1}} \leq \frac{\lambda^{(s)}}{\lambda^{(s)}I - M} \leq \frac{\lambda^{(s)}}{\lambda^{(s)} - \lambda_1} \leq \frac{\lambda^{(s)}}{\Delta_{s+1}} \leq \frac{\lambda^{(s)}}{\lambda^{(s)}I - M} \leq \frac{\lambda^{(s)}}{\lambda_1} \leq \frac{12}{\delta_x}$.

Finally, we have $\frac{1}{\lambda_{\min}(\lambda^{(s)}I - M)} = \lambda^{(s)} / \lambda^{(s)}I - M \leq \lambda^{(s)} / \lambda_1 \leq \frac{12}{\delta_x}$ where the last inequality follows from $\lambda^{(s)} \geq \lambda_1$.

**B Lemmas Needed for Proving Our Main Theorem**

In this section we provide some necessary lemmas on matrices that shall become essential for our proof of Theorem 4.1.

**Proposition B.1.** Let $A, B$ be two (column) orthonormal matrix such that for $\eta \geq 0$,

$$A^\top BB^\top A \succeq (1 - \eta)I$$

Then we have: there exists a matrix $Q, \|Q\|_2 \leq 1$ such that

$$\|A - BQ\|_2 \leq \sqrt{\eta}$$
Proof. Since $A^\top A = I$ and $A^\top B B^\top A \succeq (1 - \eta)I$, we know that $A^\top B^\perp (B^\perp)^\top A \preceq \eta I$. By the fact that

$$A = (B B^\top + B^\perp (B^\perp)^\top) A = B B^\top A + B^\perp (B^\perp)^\top A$$

we can let $Q = B^\top A$ and obtain

$$\|A - B Q\|_2 \leq \|B^\perp (B^\perp)^\top A\|_2 \leq \sqrt{\eta}.$$  

\[\square\]

### B.1 Approximate Projection Lemma

**Lemma B.2.** Let $M$ be a PSD matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ and the corresponding eigenvectors $u_1, \ldots, u_d \in \mathbb{R}^d$. For every $k \geq 1$, define $U^\perp = (u_1, \ldots, u_k) \in \mathbb{R}^{d \times k}$ and $U = (u_{k+1}, \ldots, u_d) \in \mathbb{R}^{d \times (d-k)}$. For every $\varepsilon \in (0, \frac{1}{2})$, let $V_s \in \mathbb{R}^{d \times s}$ be an orthogonal matrix such that $\|V^\top_s U\|_2 \leq \varepsilon$, define $Q_s \in \mathbb{R}^{d \times s}$ to be an arbitrary orthogonal basis of the column span of $U^\perp (U^\perp)^\top V_s$, then we have:

$$\left\| \left( I - Q_s Q_s^\top \right) M \left( I - Q_s Q_s^\top \right) - \left( I - V_s V_s^\top \right) M \left( I - V_s V_s^\top \right) \right\|_2 \leq 13\lambda_1 \varepsilon .$$

**Proof of Lemma B.2.** Since $Q_s$ is an orthogonal basis of the column span of $U^\perp (U^\perp)^\top V_s$, there is a matrix $R \in \mathbb{R}^{s \times s}$ such that

$$Q_s = U^\perp (U^\perp)^\top V_s R$$

Using the fact that $Q_s^\top Q_s = I$, we have:

$$(U^\perp (U^\perp)^\top V_s R)^\top (U^\perp (U^\perp)^\top V_s R) = I \implies R^\top V_s^\top U^\perp (U^\perp)^\top V_s R = I .$$

By the fact that $V_s^\top V_s = I$ and $U^\perp (U^\perp)^\top + U U^\top = I$, we can rewrite the above equality as:

$$R^\top \left( I - V_s^\top U U^\top V_s \right) R = I \quad \text{(B.1)}$$

From our lemma assumption, we have: $\|V^\top_s U\|_2 \leq \varepsilon$, which implies $0 \preceq V_s^\top U U^\top V_s \preceq \varepsilon^2 I$. Putting this into (B.1), we obtain:

$$I \preceq R^\top R \preceq \frac{1}{1 - \varepsilon^2} I \preceq (1 + \frac{\varepsilon^2}{3}) I$$

The above inequality directly implies that $I \preceq R R^\top \preceq (1 + \frac{\varepsilon^2}{3}) I$. Therefore,

$$\|Q_s Q_s^\top - V_s V_s^\top\|_2 = \|U^\perp (U^\perp)^\top V_s R R^\top V_s^\top U^\perp (U^\perp)^\top - V_s V_s^\top\|_2$$

$$\leq \left\| U^\perp (U^\perp)^\top V_s R R^\top V_s^\top U^\perp (U^\perp)^\top - U^\perp (U^\perp)^\top V_s V_s^\top (U^\perp (U^\perp)^\top + U U^\top) \right\|_2$$

$$\leq \left\| U^\perp (U^\perp)^\top V_s (R R^\top - I) V_s^\top (U^\perp)^\top \right\|_2 + \left\| U U^\top V_s V_s^\top U U^\top \right\|_2 + 2 \left\| U^\perp (U^\perp)^\top V_s V_s^\top U U^\top \right\|_2$$

$$\leq \|R R^\top - I\|_2 + \|U^\top V_s V_s^\top U\|_2 + 2 \|V_s^\top U U^\top V_s\|^{1/2}$$

$$\leq \frac{4}{3} \varepsilon^2 + \varepsilon^2 + 2 \varepsilon < \frac{19}{6} \varepsilon .$$
Finally, we have
\[
\left\| \left( I - Q_s Q_s^T \right) M \left( I - Q_s Q_s^T \right) - \left( I - V_s V_s^T \right) M \left( I - V_s V_s^T \right) \right\|_2 \\
\leq 2 \left\| \left( Q_s Q_s^T - V_s V_s^T \right) M \right\|_2 + \left\| \left( Q_s Q_s^T - V_s V_s^T \right) M Q_s Q_s^T \right\|_2 + \left\| \left( Q_s Q_s^T - V_s V_s^T \right) M V_s V_s^T \right\|_2 \\
\leq \frac{19 \times 4}{6} \lambda_1 \varepsilon < 13 \lambda_1 \varepsilon .
\]

\[\square\]

### B.2 Gap-Free Wedin Theorem

**Lemma B.3** (Gap free Wedin Theorem). For \( \varepsilon \geq 0 \), let \( A, B \) be two PSD matrices such that \( \| A - B \|_2 \leq \varepsilon \). For every \( \mu \geq 0 \), \( \tau > 0 \), let \( U \) be column orthonormal matrix consisting of eigenvectors of \( A \) with eigenvalue \( \leq \mu \), let \( V \) be column orthonormal matrix consisting of eigenvectors of \( B \) with eigenvalue \( \geq \mu + \tau \), then we have:

\[
\| U^T V \| \leq \frac{\varepsilon}{\tau}.
\]

**Proof of Lemma B.3.** We write \( A \) and \( B \) in terms of eigenvalue decomposition:

\[ A = U \Sigma U^T + U' \Sigma' U'^T \quad \text{and} \quad B = V \bar{\Sigma} V^T + V' \bar{\Sigma}' V'^T , \]

where \( U' \) is orthogonal to \( U \) and \( V' \) is orthogonal to \( V \). Letting \( R = A - B \), we obtain:

\[
\Sigma U^T = U^T A = U^T (B + R) \\
\implies \Sigma U^T V = U^T B V + U^T R V = U^T V \bar{\Sigma} + U^T R V \\
\implies \Sigma U^T V \bar{\Sigma}^{-1} = U^T V + U^T R V \bar{\Sigma}^{-1} .
\]

Taking spectral norm on both sides, we obtain:

\[
\| \Sigma \|_2 \| U^T V \|_2 \| \bar{\Sigma}^{-1} \|_2 \geq \| \Sigma U^T V \bar{\Sigma}^{-1} \|_2 \geq \| U^T V \|_2 - \| U^T R V \bar{\Sigma}^{-1} \|_2 .
\]

This can be simplified to

\[
\frac{\mu}{\mu + \tau} \| U^T V \|_2 \geq \| U^T V \|_2 - \frac{\varepsilon}{\mu + \tau} ,
\]

and therefore we have \( \| U^T V \|_2 \leq \frac{\varepsilon}{\tau} \) as desired. \[\square\]

### B.3 Projected Power Method

**Lemma B.4.** Let \( M \in \mathbb{R}^{d \times d} \) be a PSD matrix with eigenvalues \( \lambda_1 \geq \cdots \geq \lambda_d \) and corresponding eigenvectors \( u_1, \ldots, u_d \). Define \( U = (u_1, \ldots, u_d) \in \mathbb{R}^{d \times (d-j)} \) to be the matrix consisting of all eigenvectors with eigenvalue \( \leq \mu \). Let \( v \in \mathbb{R}^d \) be a unit vector such that \( \| v^T U \|_2 \leq \varepsilon \leq 1/2 \), and define

\[
M' = \left( I - vv^T \right) M \left( I - vv^T \right)
\]

Then, denoting \( V_2, V_1, v \in \mathbb{R}^{d \times d} \) as the unitary matrix consisting of (column) eigenvectors of \( M' \) with descending eigenvalues, where \( V_1 \) consists of eigenvectors with eigenvalue \( \leq \mu + \tau \), then there exists a matrix \( Q, \| Q \|_2 \leq 1 \) such that

\[
\| U - V_1 Q \|_2 \leq \sqrt{\frac{169 \lambda_1^2 \varepsilon^2}{\tau^2} + \varepsilon^2}
\]

17
Proof of Lemma B.4. Using Lemma B.2, let \( q = \frac{U⊥(U⊥)⊤v}{∥U⊥(U⊥)⊤v∥_2} \) be the projection of \( v \) to \( U⊥ \), we know that
\[
\left\| \left( I - qq^⊤ \right) M \left( I - qq^⊤ \right) - \left( I - vv^⊤ \right) M \left( I - vv^⊤ \right) \right\|_2 \leq 13λ_1ε.
\]

Denote \( (I - qq^⊤) M (I - qq^⊤) \) as \( M'' \). We know that \( u_{j+1}, \ldots, u_d \) are still eigenvectors of \( M'' \) with eigenvalue \( λ_{j+1}, \ldots, λ_d \).

Apply Lemma B.3 on \( A = M'', U = V_1 \), \( B = M', V = V_2 \), we obtain:
\[
∥U∥_2 ≤ \frac{13λ_1ε}{τ}.
\]

This implies that
\[
U^T V_1 V_1^T U = I - U^T V_2 V_2^T U - U^T vv^T U \succeq \left( 1 - \frac{169λ_1^2ε^2}{τ^2} - ε^2 \right) I,
\]
where the inequality uses the assumption \( ∥v^T U∥_2 ≤ ε \).

Apply Proposition B.1 to \( A = U \) and \( B = V_1 \), we conclude that there exists a matrix \( Q \), \( ∥Q∥_2 ≤ 1 \) such that
\[
∥U - V_1 Q∥_2 ≤ \sqrt{\frac{169λ_1^2ε^2}{τ^2} + ε^2}.
\]

C Proof Details for Theorem 4.1: Our Main Theorem

In this section we prove Theorem 4.1 formally.

Theorem 4.1 (restatement). Let \( M ∈ ℝ^{d×d} \) be a symmetric matrix with eigenvalues \( 1 ≥ λ_1 ≥ \cdots ≥ λ_d ≥ 0 \) and corresponding eigenvectors \( u_1, \ldots, u_d \). Let \( k ∈ [d] \), and \( δ_x, ε_{pca}, p ∈ (0,1) \). Then, LazySVD outputs a (column) orthonormal matrix \( V_k = (v_1, \ldots, v_k) ∈ ℝ^{d×k} \) which, with probability at least \( 1 - p \), satisfies all of the following properties.

(Denote by \( M_k = (I - V_k V_k^T) M (I - V_k V_k^T) \).)

(a) Core lemma: if \( ε_{pca} ≤ \frac{ε^2 δ_x^2}{2σ_k^4(λ_1/λ_k)^2} \), then \( ∥V_k^T U∥_2 ≤ ε \), where \( U = (u_{j+1}, \ldots, u_d) \) is the (column) orthonormal matrix and \( j \) is the smallest index satisfying \( λ_j ≤ (1 - δ_x)∥M_{k-1}∥_2 \).

(b) Spectral norm guarantee: if \( ε_{pca} ≤ \frac{δ_x^6}{2σ_k^4 k^3 (λ_1/λ_k)^2} \), then \( λ_{k+1} ≤ ∥M_k∥_2 ≤ \frac{λ_{k+1}}{1 - δ_x} ≤ λ_k \).

(c) Rayleigh quotient guarantee: if \( ε_{pca} ≤ \frac{δ_x^6}{2σ_k^4 k^3 (λ_1/λ_{k+1})^2} \), then \( 1 - δ_x)λ_k ≤ v_k^T M v_k ≤ \frac{1}{1 - δ_x}λ_k \).

(d) Schatten-q norm guarantee: for every \( q ≥ 1 \), if \( ε_{pca} ≤ \frac{δ_x^6}{2σ_k^4 d^q (λ_1/λ_{k+1})^2} \), then \( ∥M_k∥_{s_q} ≤ \frac{(1+δ_x)^2}{(1-δ_x)^2} \left( \sum_{i=k+1}^d λ_i^q \right)^{1/q} \).

Proof of Theorem 4.1 Let \( V_s = (v_1, \ldots, v_s) \), so we can write
\[
M_s = (I - V_s V_s^T) M (I - V_s V_s^T) = (I - v_s v_s^T) M_{s-1} (I - v_s v_s^T)
\]

(a) Define \( \tilde{λ} = ∥M_{k-1}∥_2 ≥ λ_k \).

Note that all column vectors in \( V_s \) are automatically eigenvectors of \( M_s \) with eigenvalues zero.

For analysis purpose only, let \( W_s \) be the column matrix of eigenvectors in \( V_s^⊥ \) of \( M_s \) that have eigenvalues in the range \( [0, (1 - δ_x + τ_s)\tilde{λ}] \), where \( τ_s \overset{\text{def}}{=} \frac{s}{2k} δ_x \). We now show that for every \( s = 0, \ldots, k \), there exists a matrix \( Q_s \) such that \( ∥U - W_s Q_s∥_2 \) is small and \( ∥Q_s∥_2 ≤ 1 \). We will do this by induction.
In the base case: since \( \tau_0 = 0 \), we have \( W_0 = U \) by the definition of \( U \). We can therefore define \( Q_0 \) to be the identity matrix.

For every \( s = 0, 1, \ldots, k - 1 \), suppose there exists a matrix \( Q_s \) with \( \| Q_s \|_2 \leq 1 \) that satisfies \( \| U - W_s Q_s \|_2 \leq \eta_s \) for some \( \eta_s > 0 \), we construct \( Q_{s+1} \) as follows.

First we observe that \( \text{AppxPCA} \) outputs a unit vector \( v'_{s+1} \) satisfying \( \| v'_{s+1} W_s \|_2^2 \leq \varepsilon_{\text{pca}} \) and \( \| v'_{s+1} V_s \|_2^2 \leq \varepsilon_{\text{pca}} \) with probability at least \( 1 - p/k \). This follows from Theorem 3.1 because \( [0, (1 - \delta_x + \tau_s) \hat{\lambda}] \subseteq [0, (1 - \delta_x/2) \hat{\lambda}] \), together with the fact that \( \| M_s \|_2 \geq \| M_{k-1} \|_2 \geq \hat{\lambda} \). Now, since \( v_{s+1} \) is the projection of \( v'_{s+1} \) into \( V_s \), we have

\[
\| v'_{s+1} W_s \|_2^2 \leq \frac{\| v'_{s+1} W_s \|_2^2}{\|(I - V_s V_s^T) v'_{s+1}\|_2^2} = \frac{\| v'_{s+1} W_s \|_2^2}{1 - \| V_s^T v'_{s+1} \|_2^2} \leq \frac{\varepsilon_{\text{pca}}}{1 - \varepsilon_{\text{pca}}} < 1.5\varepsilon_{\text{pca}} .
\] (C.1)

Next we apply Lemma B.4 with \( M = M_s, M' = M_{s+1}, U = W_s, V = W_{s+1}, v = v_{s+1}, \mu = (1 - \delta_x + \tau_s) \hat{\lambda} \), and \( \tau = (\tau_{s+1} - \tau_s) \hat{\lambda} \). We obtain a matrix \( Q_s \), \( \| Q_s \|_2 \leq 1 \) such that \(^5\)

\[
\| W_s - W_{s+1} Q_s \|_2 \leq \sqrt{169(\lambda_1/\hat{\lambda})^2 \cdot 1.5\varepsilon_{\text{pca}}} + \varepsilon_{\text{pca}} < \frac{32\lambda_1 k \sqrt{\varepsilon_{\text{pca}}}}{\lambda_k \delta_x} ,
\]

and this implies that

\[
\| W_{s+1} Q_s - U \|_2 \leq \| W_{s+1} Q_s Q_s - W_s Q_s \|_2 + \| W_s Q_s - U \|_2 \leq \eta_s + \frac{32\lambda_1 k \sqrt{\varepsilon_{\text{pca}}}}{\lambda_k \delta_x} .
\]

Let \( Q_{s+1} = \widetilde{Q}_s Q_s \) we know that \( \| Q_{s+1} \|_2 \leq 1 \) and

\[
\| W_{s+1} Q_{s+1} - U \|_2 \leq \| Q_{s+1} - U \|_2 \leq \eta_{s+1} \overset{\text{def}}{=} \eta_s + \frac{32\lambda_1 k \sqrt{\varepsilon_{\text{pca}}}}{\lambda_k \delta_x} .
\]

Therefore, after \( k \)-iterations of LazySVD, we obtain:

\[
\| W_k Q_k - U \|_2 \leq \eta_k = \frac{32\lambda_1 k^2 \sqrt{\varepsilon_{\text{pca}}}}{\lambda_k \delta_x} .
\]

Multiply \( U^T \) from the left, we obtain \( \| U^T W_k Q_k - I \|_2 \leq \eta_k \). Since \( \| Q_k \|_2 \leq 1 \), we must have \( \sigma_{\min}(U^T W_k) \geq 1 - \eta_k \) (here \( \sigma_{\min} \) denotes the smallest singular value). Therefore,

\[
U^T W_k W_k^T U \succeq (1 - \eta_k)^2 I .
\]

Since \( V_k \) and \( W_k \) are orthogonal of each other, we have

\[
U^T V_k V_k^T U \preceq U^T (I - W_k W_k^T) U \leq I - (1 - \eta_k)^2 I \leq 2\eta_k I .
\]

Therefore,

\[
\| V_k^T U \|_2 \leq 8 \frac{(\lambda_1/\lambda_k)^{1/2} \varepsilon_{\text{pca}}^{1/4}}{\delta_2^{1/2}} \leq \varepsilon .
\]

^5Technically speaking, to apply Lemma B.4 we need \( U = W_s \) to consist of all eigenvectors of \( M_s \) with eigenvalues \( \leq \mu \). However, we only defined \( W_s \) to be eigenvectors of \( M_s \) with eigenvalues \( \leq \mu \) that are also orthogonal to \( V_s \). It is straightforward to verify that the same result of Lemma B.4 remains true because \( v_{s+1} \) is orthogonal to \( V_s \).
(b) The statement is obvious when \( k = 0 \). For every \( k \geq 1 \), the lower bound is obvious. We prove the upper bound by contradiction. Suppose that \( \|M_k\|_2 > \frac{\lambda_{k+1}}{1-\delta_x} \). Then, since \( \|M_{k-1}\|_2 \geq \|M_k\|_2 \) and therefore \( \lambda_{k+1}, \ldots, \lambda_d < (1-\delta_x)\|M_{k-1}\|_2 \), we can apply Theorem 4.1.a of the current \( k \) to deduce that \( \|V_k^TU_{>k}\|_2 \leq \varepsilon \) where \( U_{>k} \triangleq (u_{k+1}, \ldots, u_d) \). We now apply Lemma B.2 with \( V_s = V_k \) and \( U = U_{>k} \), we obtain a matrix \( Q_k \in \mathbb{R}^{d \times k} \) whose columns are spanned by \( u_1, \ldots, u_k \) and satisfy

\[
\left\| \left( I - Q_k Q_k^\top \right) M \left( I - Q_k Q_k^\top \right) - \left( I - V_k V_k^\top \right) M \left( I - V_k V_k^\top \right) \right\|_2 < 16\lambda_1 \varepsilon .
\]

However, our assumption says that the second matrix \( (I - V_k V_k^\top) M (I - V_k V_k^\top) \) has spectral norm at least \( \lambda_{k+1}/(1-\delta_x) \), but we know that \( (I - Q_k Q_k^\top) M (I - Q_k Q_k^\top) \) has spectral norm exactly \( \lambda_{k+1} \) due to the definition of \( Q_k \). Therefore, we must have \( \frac{\lambda_{k+1}}{1-\delta_x} - \lambda_{k+1} \leq 16\lambda_1 \varepsilon \) due to triangle inequality.

In other words, by selecting \( \varepsilon \) in Theorem 4.1.a to satisfy \( \varepsilon \leq \frac{\delta_x}{16\lambda_1/\lambda_{k+1}} \) (which is satisfied by our assumption on \( \varepsilon_{\text{pca}} \)), we get a contradiction so can conclude that \( \|M_k\|_2 \leq \frac{\lambda_{k+1}}{1-\delta_x} \).

(c) We compute that

\[
v_k^\top M v_k = v_k^\top M_{k-1} v_k \overset{\#}{\geq} \frac{v_k^\top M_{k-1} v_k'}{\|I - V_{k-1} V_{k-1}^\top\|_2^2} \geq \frac{v_k^\top M_{k-1} v_k'}{1 - \varepsilon_{\text{pca}}} \geq (1 - \frac{\delta_x}{2})\|M_{k-1}\|_2 \geq (1 - \delta_x)\|M_{k-1}\|_2 .
\]

Above, \( \# \) is because \( v_k \) is the projection of \( v_k' \) into \( V_{k-1}^\perp \); \( \# \) is because \( \|V_{k-1}^\perp v_k'\|_2 \leq \varepsilon_{\text{pca}} \) following the same reason as (C.1) and \( \# \) is owing to Theorem 3.1. Next, since \( \|M_{k-1}\|_2 \geq \lambda_k \), we automatically have \( v_k^\top M v_k = v_k^\top M_{k-1} v_k \leq \|M_{k-1}\|_2 \leq \frac{\lambda_k}{1-\delta_x} \) where the last inequality is owing to Theorem 4.1.b.

(d) Since \( \|V_k^\top U\|_2 \leq \varepsilon_c \triangleq 8(\lambda_1/\lambda_k)^{1/2}\|k_{\text{pca}}/\delta_x\|^{1/4} \) from the analysis of Theorem 4.1.a, we can apply Lemma B.2 to obtain a (column) orthogonal matrix \( Q_k \in \mathbb{R}^{d \times k} \) such that

\[
\|M'_k - M_k\|_2 \leq 16\lambda_1 \varepsilon_c , \quad \text{where } M'_k \overset{\text{def}}{=} (I - Q_k Q_k^\top) M (I - Q_k Q_k^\top) .
\]

Suppose \( U = (u_{d-p+1}, \ldots, u_d) \) is of dimension \( d \times p \), that is, there are exactly \( p \) eigenvalues of \( M \) that are \( \leq (1 - \delta_x)\|M_{k-1}\|_2 \). Then, the definition of \( Q_k \) in Lemma B.2 tells us \( U^\top Q_k = 0 \) so \( M'_k \) agrees with \( M \) on all the eigenvalues and eigenvectors \( \{(\lambda_j, u_j)\}_{j=d-p+1}^d \) because an index \( j \) satisfies \( \lambda_j \leq (1 - \delta_x)\|M_{k-1}\|_2 \) if and only if \( j \in \{d-p+1, d-p+2, \ldots, d\} \).

Denote by \( \mu_1, \ldots, \mu_{d-k} \) the eigenvalues of \( M'_k \) excluding the \( k \) zero eigenvalues in subspace \( Q_k \), and assume without loss of generality that \( \{\mu_1, \ldots, \mu_p\} = \{\lambda_{d-p+1}, \ldots, \lambda_d\} \). Then,

\[
\|M'_k\|_S_q = \sum_{i=1}^{d-k} \mu_i^q = \sum_{i=1}^p \mu_i^q + \sum_{i=p+1}^{d-k} \mu_i^q = \sum_{i=d-p+1}^d \lambda_i^q + \sum_{i=p+1}^{d-k} \mu_i^q \overset{\#}{\leq} \sum_{i=d-p+1}^d \lambda_i^q + (d-k-p)(\|M_k\|_2 + 16\lambda_1 \varepsilon_c)^q
\]

\[
\overset{\#}{\leq} \sum_{i=d-p+1}^d \lambda_i^q + (d-k-p)\left(\frac{\lambda_{k+1}}{(1-\delta_x)} + 16\lambda_1 \varepsilon_c\right)^q
\]

20
Above, Θ is because each \( \mu_i \) is no greater than \( \|M'_k\|_2 \), and Θ is owing to (C.2), and Θ is because of Theorem 4.1. Let

\[
\|M'_k\|_S^q \leq \sum_{i=d-p+1}^{d} \lambda_i^q + (d-k-p) (1 + \delta_X)^q (1 - \delta_X)^{2q} \lambda_{k+1}^q
\]

\[
\leq \sum_{i=d-p+1}^{d} \lambda_i^q + \frac{(1 + \delta_X)^q}{(1 - \delta_X)^{2q}} \sum_{i=k+1}^{d-p} \lambda_i^q \sum_{i=k+1}^{d} \lambda_i^q
\]

Above, Θ is because for each eigenvalue \( \lambda_i \) where \( i \in \{k+1, k+2, \ldots, d-p\} \), we have

\[
\lambda_i > (1 - \delta_X)\|M_{k-1}\|_2 \geq (1 - \delta_X)\lambda_k \geq (1 - \delta_X)\lambda_{k+1}
\]

Finally, using (C.2) again we have

\[
\|M_k\|_S^q \leq \|M'_k\|_S^q + \|M_k - M'_k\|_S^q \leq \|M'_k\|_S^q + d^{1/p}\|M_k - M'_k\|_2
\]

As long as \( \varepsilon_c \leq \frac{\delta_X \lambda_{k+2}}{16d^{1/p} \lambda_1} \), we have

\[
\|M_k\|_S^q \leq \left( \frac{1 + \delta_X}{1 - \delta_X} \right)^{1/q} \left( \sum_{i=k+1}^{d} \lambda_i^q \right)^{1/q}
\]

as desired. Finally, we note that \( \varepsilon_c \leq \frac{\delta_X \lambda_{k+1}}{16d^{1/p} \lambda_1} \) is satisfied with our assumption on \( \varepsilon_{\text{pca}} \).

\[\square\]

### C.1 Proofs of Corollary 4.3 and Corollary 4.4

We first note that since LazySVD outputs \( v_i \) one by one, although we have only stated Theorem 4.1 for the last iteration \( k \), the claimed properties (a)-(d) hold for all intermediate iterations \( s = 1, \ldots, k \).

**Proof of Corollary 4.3 from Theorem 4.1.** By Theorem 4.1.a, we have: \( \|V_k^T U\|_2 \leq \varepsilon \) where \( U = (u_j, \ldots, u_d) \) is a (column) orthonormal matrix and \( j \) is the smallest index satisfying \( \lambda_j \leq (1 - \delta_X)\|M_{k-1}\|_2 \). Since it satisfies \( \|M_{k-1}\|_2 \geq \lambda_k \), we have

\[
\lambda_{k+1} = \sigma_{k+1}^2 = \sigma_k^2 (1 - \text{gap})^2 = \lambda_k (1 - \text{gap})^2 \leq \lambda_k (1 - \delta_X) \leq (1 - \delta_X)\|M_{k-1}\|_2
\]

where the first inequality is because our choice of \( \delta_X = \text{gap} \). Therefore, \( j \) must be equal to \( k + 1 \) according to its definition, and we conclude \( \|V_k^T W\|_2 \leq \varepsilon \).

The running time of the algorithm comes directly from Theorem 4.2 by putting in the parameters.

**Proof of Corollary 4.4 from Theorem 4.1.** Denote

\[
M_k = (I - V_k V_k^T) M (I - V_k V_k^T) = (I - V_k V_k^T) A A^T (I - V_k V_k^T)
\]

According to Theorem 4.1, we have:

\[
\|M_k\|_S^q \leq \left( \frac{1 + \delta_X}{1 - \delta_X} \right)^2 \|M - M_k\|_S^q \quad \forall q \geq 1 \text{ and } q = \infty
\]
where $M^*_k = (A^*_k)(A^*_k)^\top$ is the rank-$k$ SVD of $M$, and recall $\delta_x = \frac{\epsilon}{5}$. For the spectral norm guarantee, we take $q = \infty$ and compute

$$
\|A - A_k\|_2 = \|(I - V_kV_k^\top)A\|_2 = \sqrt{\|(I - V_kV_k^\top)AA^\top(I - V_kV_k^\top)\|_2}
$$

$$
= \sqrt{\|M_k\|_2} = \sqrt{\|M_k\|_{\infty}} \leq \frac{1 + \delta_x}{1 - \delta_x} \sqrt{\|M - M^*_k\|_{\infty}}
$$

$$
\leq (1 + \varepsilon)\sqrt{\|M - M^*_k\|_{\infty}} = (1 + \varepsilon)\sqrt{\|M - M^*_k\|_2} = (1 + \varepsilon)\sigma_{k+1} = (1 + \varepsilon)\|A - A_k\|_2 .
$$

For the Frobenius norm guarantee, we take $q = 1$ and compute

$$
\|A - A_k\|_F = \sqrt{\text{Tr}[(I - V_kV_k^\top)AA^\top(I - V_kV_k^\top)]} = \sqrt{\|M_k\|_s} \leq \frac{1 + \delta_x}{1 - \delta_x} \sqrt{\|M - M^*_k\|_s}
$$

$$
\leq (1 + \varepsilon)\sqrt{\|M - M^*_k\|_s} = (1 + \varepsilon)\sqrt{\sum_{i=k+1}^d \sigma_i^2} = (1 + \varepsilon)\|A - A_k\|_F .
$$

The Rayleigh quotient guarantees directly follow from Theorem 4.1.c. The running time of the algorithm comes directly from Theorem 4.2 by putting in the parameters. □

D Proof Details for Our NNZ Running-Time Results

We state and prove a simple proposition first, and then divide this sections into three subsections: Section D.1 deals with column sampling together with the spectral-norm guarantee; Section D.2 deals with column sampling together with the Frobenius-norm guarantee; and Section D.3 deals with entry-wise sampling together with the spectral-norm guarantee.

**Proposition D.1.** Let $A, A' \in \mathbb{R}^{d \times n}$ be two matrices with $d \leq n$, and $\eta \geq 0$ be an non-negative real. Suppose $\|A - A'\|_2 \leq \eta$, then for every $k \in [d]$, $\sigma_k(A') - \eta \leq \sigma_k(A) \leq \sigma_k(A') + \eta$

**Proof of Proposition D.1.** By symmetry it is enough to show that $\sigma_k(A') \leq \sigma_k(A) + \eta$.

Let $v_1, \ldots, v_d$ be the (left) singular vectors of $A$ in decreasing order of the corresponding singular values, and let $S_k$ be the space spanned by $v_k, \ldots, v_d$. Then, for every $x \in S_k$ that has $\|x\|_2 = 1$,

$$
\|x^\top A'\|_2 \leq \|x^\top (A - A')\|_2 + \|x^\top A\|_2 \leq \sigma_k(A) + \eta .
$$

Recall that the Courant-Fischer theorem says that

$$
\sigma_k(A') = \min_{S, \dim(S) = d-k+1} \max_{x \in S, \|x\|_2 = 1} \|x^\top A'\|_2 .
$$

Take $S = S_k$, we immediately obtain $\sigma_k(A') \leq \sigma_k(A) + \eta$. □

D.1 Column Sampling with Spectral-Norm Guarantee

We first state a concentration bound on column sampling (which is easily provable using for instance [17, Theorem 6.6.1]):

**Lemma D.2** (column sampling). Let $A \in \mathbb{R}^{d \times n}$ be a matrix and $A_i \in \mathbb{R}^d$ be the $i$-th column of $A$. Setting $p_i \defeq \|A_i\|_2^2/\|A\|_F^2$ for each $i$, and define random rank-1 matrix $R = \frac{1}{p_i}A_iA_i^\top$ with probability
$p_i$ for each $i$. For every $m \geq 1$, define $\overline{R}_m$ to be the average of $m$ independent copies of $R$, that is, $\overline{R}_m \overset{\text{def}}{=} \frac{1}{m} \sum_{t=1}^{m} R_t$ where each $R_t$ is drawn from $R$. Then, for every $\eta, \delta > 0$, if

$$m \geq \frac{8\|A\|_F^2 \|A\|_2^2 \log \frac{1}{\delta}}{\eta^2} + \frac{8\|A\|_F^2 \log \frac{1}{\delta}}{\eta},$$

we have that with probability $1 - \delta$, it satisfies $\|\overline{R}_m - AA^T\|_2 \leq \eta$.

The next lemma translates the approximate solution on the column sampled matrix into a spectral guarantee on the original matrix.

**Lemma D.3.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix and define $\overline{R}_m$ as in Lemma D.2. For every $k \in [d - 1]$, every $\varepsilon > 0$, every $p \in (0, 1)$, and every $\delta_X \in (0, 1)$, if $m \geq \frac{32k \log(1/p) \sigma_1(A)^4}{\varepsilon^2 \sigma_{k+1}(A)^4}$ and one obtains an $\delta_X$-approximate $k$-SVD of $\overline{R}_m$ in terms of spectral norm, that is

a column orthogonal matrix $V \in \mathbb{R}^{d \times k}$ such that $\|(I - VV^T)\overline{R}_m(I - VV^T)\|_2 \leq \frac{\lambda_{k+1}(\overline{R}_m)}{1 - \delta_X}$.

Then, with probability at least $1 - p$, this matrix $V$ satisfies

$$\|A - VV^T A\|_2 \leq \frac{\|A - A_k^*\|_2 + 2\varepsilon \|A - A_k^*\|_F}{1 - \delta_X}. $$

**Proof of Lemma D.3.** If we let $\eta \overset{\text{def}}{=} \varepsilon^2 \|A - A_k^*\|_F^2 + \varepsilon \|A - A_k^*\|_F \|A - A_k^*\|_2$, we can compute

$$\frac{8\|A\|_F^2 \|A\|_2^2 \log \frac{1}{p}}{\eta^2} + \frac{8\|A\|_F^2 \log \frac{1}{p}}{\eta} \leq \frac{8\|A\|_F^2 \|A\|_2^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} + \frac{8\|A\|_F^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} \leq \frac{8\|A\|_F^2 \|A - A_k^*\|_2^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} + \frac{8\|A\|_F^2 \log \frac{1}{p}}{\varepsilon^2 \|A - A_k^*\|_F^2} \leq \frac{32k \log(1/p) \sigma_1(A)^4}{\varepsilon^2 \sigma_{k+1}(A)^4} \leq m$$

Therefore, according to Lemma D.2, with probability at least $1 - p$, it satisfies

$$\|AA^T - \overline{R}_m\|_2 \leq \eta = \varepsilon^2 \|A - A_k^*\|_F^2 + \varepsilon \|A - A_k^*\|_F \|A - A_k^*\|_2.$$

This further implies that, owing to $\|I - VV^T\|_2 \leq 1$,

$$\|(I - VV^T)AA^T(I - VV^T)\|_2 \leq \|(I - VV^T)\overline{R}_m(I - VV^T)\|_2 + \|AA^T - \overline{R}_m\|_2 \leq \frac{\lambda_{k+1}(\overline{R}_m)}{1 - \delta_X} + \eta \leq \frac{\lambda_{k+1}(AA^T) + 2\eta}{1 - \delta_X} \leq \frac{\sigma_{k+1}(A)^2 + 2\varepsilon^2 \|A - A_k^*\|_F^2 + 2\varepsilon \|A - A_k^*\|_F \|A - A_k^*\|_2}{1 - \delta_X} \leq \frac{(\|A - A_k^*\|_2 + 2\varepsilon \|A - A_k^*\|_F)^2}{1 - \delta_X}. $$
Therefore,
\[ \| (I - V V^\top) A \|_2 = \sqrt{\| (I - V V^\top) A A^\top (I - V V^\top) \|_2} \leq \frac{\| A - A_k^* \|_2 + 2\varepsilon \| A - A_k^* \|_F}{1 - \delta_x}. \]

Using the previous lemma, it is not hard to deduce our main result of this sub-section.

**Theorem D.4.** Let \( A \in \mathbb{R}^{d \times n} \) be a matrix with singular values \( \sigma_1 \geq \cdots \geq \sigma_d \geq 0 \). For every \( \varepsilon \in (0, 1/2) \), let \( \tilde{R}_m \) be the subsampled version of \( A \) as defined in Lemma D.2 with \( m = \Omega \left( \frac{k \log(1/\varepsilon) \sigma_{k+1}^4}{\varepsilon^2 \sigma_k} \right) \).

Then, one can call LazySVD with appropriately chosen \( \delta_x \) to produce a matrix \( V_k \in \mathbb{R}^{d \times k} \) satisfying
\[ \| A - V_k V_k^\top A \|_2 \leq \| A - A_k^* \|_2 + O(\varepsilon) \| A - A_k^* \|_F, \tag{D.1} \]

and the total running time is \( O(\text{nnz}(A)) + \tilde{O}(k^2 d (\sigma_1/\sigma_{k+1})^4 \varepsilon^2) \) if AGD is used as the approximate matrix inversion algorithm \( A \). Furthermore, if \( \varepsilon \leq \frac{\| A - A_k^* \|_2}{2 \| A - A_k^* \|_F} \), then one can use accelerated SVRG as \( A \) and improve the running time to \( O(\text{nnz}(A)) + \tilde{O}(k^2 d (\sigma_1/\sigma_{k+1})^4 \varepsilon^2) \).

**Proof of Theorem D.4.** Let us define \( M \defeq \tilde{R}_m/\|A\|_F^2 \), and we can write \( M = \frac{1}{m} \sum_{i=1}^m a_i a_i^\top \) where each \( a_i \) has Euclidean norm at most 1 due to our definition of \( \tilde{R}_m \) in Lemma D.2. We pass this matrix \( M \) as the input \( M \) to LazySVD, and it satisfies \( \| M \|_2 \leq \text{Tr}(M) \leq 1 \).

Before specifying the parameter choices for \( \delta_x \) and the algorithm choice for \( A \), we notice that for sufficiently small \( \varepsilon \), Theorem 4.1.b implies \( \| (I - V_k V_k^\top) \tilde{R}_m (I - V_k V_k^\top) \|_2 \leq \frac{1}{1 - \delta_x} \lambda_{k+1}(\tilde{R}_m) \) where \( V_k \) is the output matrix from LazySVD. Applying Lemma D.3, we have
\[ \| A - V_k V_k^\top A \|_2 \leq \frac{\| A - A_k^* \|_2 + 2\varepsilon \| A - A_k^* \|_F}{1 - \delta_x}. \]

Now there are two cases. If we use AGD as the method \( A \), then we can choose \( \delta_x = \varepsilon \). In such a case it is easy to see that \( \frac{\| A - A_k^* \|_2 + 2\varepsilon \| A - A_k^* \|_F}{1 - \delta_x} \leq \| A - A_k^* \|_2 + 4\varepsilon \| A - A_k^* \|_F \) so the guarantee (D.1) is satisfied. The total running time is \( O(\text{nnz}(A)) \) to sample \( \{a_1, \ldots, a_m\} \) plus \( \tilde{O}(kmd/\sqrt{\varepsilon}) \) to perform LazySVD (see Theorem 4.2).

If we use accelerated SVRG as the method \( A \), then we obtain a better dependence on \( \varepsilon \) as follows. Suppose \( \varepsilon \leq \frac{\| A - A_k^* \|_2}{2 \| A - A_k^* \|_F} \), and we choose \( \delta_x \leq \varepsilon \) \( \frac{\| A - A_k^* \|_2}{\| A - A_k^* \|_F} \leq \| A - A_k^* \|_2 + O(\varepsilon) \| A - A_k^* \|_F \) so the guarantee (D.1) is satisfied. As for the running time, in addition to \( O(\text{nnz}(A)) \) for sampling, we need (using Theorem 4.2 again)
\[ \tilde{O}(kmd + \frac{km^{3/4}d}{\lambda_k(M)^{1/4} \delta_x^{1/2}}) = \tilde{O}(kmd + \frac{km^{3/4}d}{\lambda_k(M)^{1/4} \delta_x^{1/2}}) = \tilde{O}\left(kmd + \frac{km^{3/4}d}{\sigma_k(A)^2 \delta_x^{1/2}} \left( \frac{\| A \|^2_F}{\sigma_k(A)^2} \right)^{1/4} \right) \]
\[ = \tilde{O}\left(kmd + \frac{km^{3/4}d}{\varepsilon^{1/2}} \left( \frac{\sigma_k(A)^2 \| A \|^2_F}{\sigma_k(A)^2 \| A - A_k^* \|_F^2} \right)^{1/4} \right) \]

Since it can be verified (similar to the proof of Lemma D.3) \( \frac{\sigma_{k+1}(A)^2 \| A \|^2_F}{\sigma_k(A)^2 \| A - A_k^* \|_F^2} \leq 1 + \frac{\sigma_1(A)^2}{\sigma_k(A)^2} \cdot k \), we conclude that the above running time becomes
\[ \tilde{O}\left(kmd + \frac{km^{3/4}d}{\varepsilon^{1/2}} \left( \frac{\sigma_1(A)^2}{\sigma_k(A)^2} \cdot k \right)^{1/4} \right) = \tilde{O}\left(\frac{k^2 d (\sigma_1/\sigma_{k+1})^4 \varepsilon^2}{\varepsilon^2} \right) \]
where the last equality follows from the definition of \( m \).

\( \square \)
D.2 Column Sampling with Frobenius-Norm Guarantee

The next lemma translates the approximate solution on the column sampled matrix into a Frobenius-norm guarantee on the original matrix.

**Lemma D.5.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix and define $\overline{R}_m$ as in Lemma D.2. For every $k \in [d - 1]$, every $\varepsilon > 0$, every $p \in (0, 1)$, and every $\delta_x \in (0, 1)$, if $m \geq \frac{128k^3 \log(1/p)\sigma_1(A)^4}{\varepsilon^2 \sigma_{k+1}(A)^4}$ and one obtains a $\delta_x$-approximate $k$-SVD of $\overline{R}_m$ in terms of Rayleigh quotient, that is, a column orthonormal matrix $V = (v_1, \ldots, v_k) \in \mathbb{R}^{d \times k}$ such that

$$ \forall i \in [k], \quad |v_i^T \overline{R}_m v_i - \lambda_i(\overline{R}_m)| \leq \delta_x \lambda_i(\overline{R}_m) . $$

Then, with probability at least $1 - p$, this matrix $V$ satisfies

$$ \| (I - VV^T) A \|_F^2 \leq \delta_x \| A_k^* \|_F^2 + (1 + \varepsilon) \| A - A_k^* \|_F^2 . $$

**Proof of Lemma D.5.** Using similar arguments as in the proof of Lemma D.3, one can deduce that our choice of $m$ ensures

$$ \| AA^T - \overline{R}_m \|_2 \leq \eta \equiv \frac{\varepsilon}{2k} \| A - A_k^* \|_F^2 . $$

Which implies that $| \lambda_i(\overline{R}_m) - \sigma_i^2 | \leq \eta$ due to Proposition D.1. Finally,

$$ \| (I - VV^T) A \|_F^2 = \text{Tr}[(I - VV^T)AA^T(I - VV^T)] = \text{Tr}[AA^T - VAA^TV] $$

$$ = \sum_{i=1}^d \sigma_i(A)^2 - \sum_{i=1}^k v_i^T AA^T v_i \leq \sum_{i=1}^d \sigma_i(A)^2 + k\eta - \sum_{i=1}^k v_i^T \overline{R}_m v_i $$

$$ \leq \sum_{i=1}^d \sigma_i(A)^2 + k\eta - \sum_{i=1}^k (1 - \delta_x)(\sigma_i^2(A) - \eta) \leq \sum_{i=1}^k \delta_x \sigma_i(A)^2 + 2k\eta + \sum_{i=k+1}^d \sigma_i(A)^2 $$

$$ = \delta_x \| A_k^* \|_F^2 + \| A - A_k^* \|_F^2 + 2k\eta = \delta_x \| A_k^* \|_F^2 + (1 + \varepsilon) \| A - A_k^* \|_F^2 . \square $$

Using the previous lemma, it is not hard to deduce our main result of this sub-section.

**Theorem D.6.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. For every $\varepsilon \in (0, 1/2)$, let $\overline{R}_m$ be the subsampled version of $A$ as defined in Lemma D.2 with $m = \Omega\left(\frac{k^4 (\log(1/\varepsilon)) \sigma_1^4}{\varepsilon^2 \sigma_{k+1}^4}\right)$. Then, one can call **LazySVD** with appropriately chosen $\delta_x$ to produce a matrix $V_k \in \mathbb{R}^{d \times k}$ satisfying

$$ \| A - V_k V_k^T A \|_2 \leq (1 + O(\varepsilon)) \| A - A_k^* \|_F , $$

and the total running time is $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^4 d (\sigma_1/\sigma_{k+1})^5}{\varepsilon^{2\alpha}}\right)$ if AGD is used as the approximate matrix inversion algorithm $A$, or $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^4 d (\sigma_1/\sigma_{k+1})^4}{\varepsilon^2}\right)$ if $A$ is accelerated SVRG.

**Proof.** One can choose $\delta_x = \varepsilon \| A - A_k^* \|_F^2 / \| A_k^* \|_F^2 \geq \frac{\varepsilon^2 \sigma_{k+1}^2}{k \sigma_k^2}$ as the parameter of **LazySVD** and the proof is completely analogous to that of Theorem D.4. \square
D.3 Entry-Wise Sampling with Spectral-Norm Guarantee

We first state a concentration bound on entry-wise sampling:

**Lemma D.7** (entry-wise sampling [5]). Let $A \in \mathbb{R}^{d \times n}$ be a matrix. Define random single-entry matrix $R = \frac{1}{p_{i,j}}A_{i,j}$ where $(i,j)$ is selected from $[d] \times [m]$ each with probability $p_{i,j} \overset{\text{def}}{=} A_{i,j}^2/\|A\|_F^2$. For every $m \geq 1$, define $R_s$ to be the average of $s$ independent copies of $R$, that is, $R_s = \frac{1}{s} \sum_{t=1}^s R_t$ where each $R_t$ is drawn from $R$. Then, for every $\eta, p \in (0,1)$, if

$$s \geq \frac{28(d + n) \log(2/p)\|A\|_F^2}{\eta^2},$$

we have with probability $1 - p$, it satisfies $\|A - R_s\|_2 \leq \eta$.

The next lemma translates the approximate solution on the entry-wise sampled matrix into a spectral guarantee on the original matrix.

**Lemma D.8.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix and define $R_s$ as in Lemma D.7. For every $k \in [d - 1]$, every $\varepsilon > 0$, every $p \in (0,1]$, and every $\delta_x \in (0, 1)$, if $s \geq \frac{56k(d+n)\ln(2/p)\sigma_1(A)^2}{\varepsilon^2 \sigma_{k+1}(A)^2}$ and one obtains a matrix $R'$ satisfying

$$\|R' - R_s\|_2 \leq \frac{\sigma_{k+1}(R_s)}{1 - \delta_x}.$$

Then, with probability at least $1 - p$, we also have

$$\|A - R'\|_2 \leq \frac{\|A - A_k^*\|_2 + 2\varepsilon\|A - A_k^*\|_F}{1 - \delta_x}. \quad \Box$$

**Proof of Lemma D.8.** We first compute that

$$s \geq \frac{56k(d + n) \ln(2/p)\sigma_1(A)^2}{\varepsilon^2 \sigma_{k+1}(A)^2} \geq \frac{28(d + n) \ln(2/p)\sigma_1(A)^2}{\varepsilon^2 \sigma_{k+1}(A)^2} + \frac{28d \ln(2/p)}{\varepsilon^2} \geq \frac{28(d + n) \ln(2/p)\|A_k^*\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2} + \frac{28(d + n) \ln(2/p)\|A - A_k^*\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2} = \frac{28(d + n) \ln(2/p)(\|A_k^*\|_F^2 + \|A - A_k^*\|_F^2)}{\varepsilon^2 \|A - A_k^*\|_F^2} = \frac{28(d + n) \ln(2/p)\|A\|_F^2}{\varepsilon^2 \|A - A_k^*\|_F^2}.$$

Owing to Lemma D.7, with probability $1 - p$, it satisfies $\|A - R_s\|_2 \leq \eta \overset{\text{def}}{=} \varepsilon \|A - A_k^*\|_F$. Using Proposition D.1, we know that $\sigma_{k+1}(R_s) \leq \sigma_{k+1}(A) + \eta$, which implies that

$$\|R' - R_s\|_2 \leq \frac{\sigma_{k+1}(A) + \eta}{1 - \delta_x}. \quad \Box$$

Finally,

$$\|R' - A\|_2 \leq \|R' - R_s\|_2 + \|A - R_s\|_2 \leq \frac{\sigma_{k+1}(A) + \eta}{1 - \delta_x} + \eta \leq \frac{\sigma_{k+1}(A) + 2\varepsilon\|A - A_k^*\|_F}{1 - \delta_x}. \quad \Box$$
Using the previous lemma, it is not hard to deduce our main result of this sub-section.

**Theorem D.9.** Let $A \in \mathbb{R}^{d \times n}$ be a matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. For every $\varepsilon \in (0, 1/2)$, let $\overline{A}_s$ be the entry-sampled version of $A$ as defined in Lemma D.7 with $s = \Omega\left(\frac{k(k+d+n)\log(1/\varepsilon)}{\varepsilon^2\sigma_{k+1}^2}\right)$. Then, one can call LazySVD with appropriately chosen $\delta_{\times}$ to produce a matrix $V_k \in \mathbb{R}^{d \times k}$ satisfying

$$
\|A - V_k V_k^T A\|_2 \leq \|A - A_k^*\|_2 + O(\varepsilon)\|A - A_k^*\|_F,
$$

and the total running time is $O(\text{nnz}(A)) + \tilde{O}\left(\frac{k^2(n+d)(\sigma_1/\sigma_{k+1})^2}{\varepsilon^2}\right)$ if AGD is used as the approximate matrix inversion algorithm $A$.

**Proof.** One can choose $\delta_{\times} = \varepsilon$ and a completely analogous proof as that of Theorem D.4 gives us the desired spectral guarantee. The running time follows from Theorem 4.2 because $\text{nnz}(\overline{A}_s) = s$. \qed

**References**

[1] Zeyuan Allen-Zhu and Yang Yuan. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In ICML, 2016.

[2] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, 56(2), 2009.

[3] Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter Low-rank Approximation via Sampling the Leveraged Element. In SODA, pages 902–920, 2015.

[4] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In STOC, pages 81–90, 2013.

[5] Petros Drineas and Anastasios Zouzias. A Note on Element-wise Matrix Sparsification via a Matrix-valued Bernstein Inequality. *ArXiv e-prints*, abs/1006.0407, January 2011.

[6] Rong-En Fan and Chih-Jen Lin. LIBSVM Data: Classification, Regression and Multi-label. Accessed: 2015-06.

[7] Dan Garber and Elad Hazan. Fast and simple PCA via convex optimization. *ArXiv e-prints*, September 2015.

[8] Daniel Garber, Elad Hazan, Chi Jin, Sham M. Kakade, Cameron Musco, Praneeth Netrapalli, and Aaron Sidford. Robust shift-and-invert preconditioning: Faster and more sample efficient algorithms for eigenvector computation. In ICML, 2016.

[9] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The JHU Press, 4th edition, 2012.

[10] Prateek Jain, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming PCA: Matching Matrix Bernstein and Near-Optimal Finite Sample Guarantees for Oja’s Algorithm. *ArXiv e-prints*, abs/1602.06929, February 2016.

[11] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. [http://snap.stanford.edu/data](http://snap.stanford.edu/data), June 2014.
[12] Chris J. Li, Mengdi Wang, Han Liu, and Tong Zhang. Near-Optimal Stochastic Approximation for Online Principal Component Estimation. ArXiv e-prints, abs/1603.05305, March 2016.

[13] Ren-Cang Li and Lei-Hong Zhang. Convergence of the block lanczos method for eigenvalue clusters. Numerische Mathematik, 131(1):83–113, 2015.

[14] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In NIPS, pages 1396–1404, 2015.

[15] Ohad Shamir. A Stochastic PCA and SVD Algorithm with an Exponential Convergence Rate. In icml, ICML 2015, pages 144—153, 2015.

[16] Ohad Shamir. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In ICML, 2016.

[17] Joel A. Tropp. An Introduction to Matrix Concentration Inequalities. ArXiv e-prints, abs/1501.01571, January 2015.

[18] P. A. Wedin. Perturbation bounds in connection with singular value decomposition. BIT Numerical Mathematics, 12(1):99–111, 1972.
Figure 2: Performance on dataset email. Relative error (y-axis) vs. running time (x-axis).
Figure 3: Performance on dataset Amazon. Relative error ($y$-axis) vs. running time ($x$-axis).
Figure 4: Performance on dataset news20. Relative error (y-axis) vs. running time (x-axis).
Figure 5: Performance on dataset rcv1. Relative error (y-axis) vs. running time (x-axis).