Faster Eigenvector Computation via Shift-and-Invert Preconditioning *

Dan Garber
Toyota Technological Institute at Chicago
dgarber@ttic.edu

Elad Hazan
Princeton University
ehazan@cs.princeton.edu

Chi Jin
UC Berkeley
chijin@eecs.berkeley.edu

Sham M. Kakade
University of Washington
sham@cs.washington.edu

Cameron Musco
MIT
cnmusco@mit.edu

Praneeth Netrapalli
Microsoft Research, New England
praneeth@microsoft.com

Aaron Sidford
Microsoft Research, New England
asid@microsoft.com

Abstract

We give faster algorithms and improved sample complexities for estimating the top eigenvector of a matrix $\mathbf{\Sigma}$ – i.e. computing a unit vector x such that $x^\top \mathbf{\Sigma} x \geq (1-\epsilon)\lambda_1(\mathbf{\Sigma})$:

- **Offline Eigenvector Estimation:** Given an explicit $\mathbf{A} \in \mathbb{R}^{n \times d}$ with $\mathbf{\Sigma} = \mathbf{A}^\top \mathbf{A}$, we show how to compute an ϵ approximate top eigenvector in time $\widetilde{O}\left(\left[\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right] \cdot \log 1/\epsilon\right)$ and $\widetilde{O}\left(\left[\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}}\right] \cdot \log 1/\epsilon\right)$. Here $\text{nnz}(\mathbf{A})$ is the number of nonzeros in $\mathbf{A}$, $\text{sr}(\mathbf{A}) \stackrel{\text{def}}{=} \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_2^2}$ is the stable rank, gap is the relative eigengap, and $\tilde{O}(\cdot)$ hides log factors in d and gap. By separating the gap dependence from the $\text{nnz}(\mathbf{A})$ term, our first runtime improves upon the classical power and Lanczos methods. It also improves prior work using fast subspace embeddings [AC09, CW13] and stochastic optimization [Sha15c], giving significantly better dependencies on $\text{sr}(\mathbf{A})$ and ϵ. Our second running time improves these further when $\text{nnz}(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}$.
- **Online Eigenvector Estimation:** Given a distribution $\mathcal{D}$ with covariance matrix $\mathbf{\Sigma}$ and a vector x_0 which is an $O(\text{gap})$ approximate top eigenvector for $\mathbf{\Sigma}$, we show how to refine to an ϵ approximation using $O\left(\frac{\text{v}(\mathcal{D})}{\text{gap}\cdot\epsilon}\right)$ samples from $\mathcal{D}$. Here $\text{v}(\mathcal{D})$ is a natural notion of variance. Combining our algorithm with previous work to initialize x_0, we obtain improved sample complexity and runtime results under a variety of assumptions on $\mathcal{D}$.

We achieve our results using a general framework that we believe is of independent interest. We give a robust analysis of the classic method of *shift-and-invert* preconditioning to reduce eigenvector computation to *approximately* solving a sequence of linear systems. We then apply fast stochastic variance reduced gradient (SVRG) based system solvers to achieve our claims. We believe our results suggest the general effectiveness of shift-and-invert based approaches and imply that further computational gains may be reaped in practice.

*This paper combines work first appearing in [GH15] and [JKM+15]

1 Introduction

Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, computing the top eigenvector of $\mathbf{A}^\top \mathbf{A}$ is a fundamental problem in numerical linear algebra, applicable to principal component analysis [Jol02], spectral clustering and learning [NJW02, VW04], pagerank computation, and many other graph computations [PBMW99, Kor03, Spi07]. For instance, a degree-k principal component analysis is nothing more than performing k leading eigenvector computations. Given the ever-growing size of modern datasets, it is thus a key challenge to come up with more efficient algorithms for this basic computational primitive.

In this work we provide improved algorithms for computing the top eigenvector, both in the *offline* case, when $\mathbf{A}$ is given explicitly and in the *online* or *statistical* case where we access samples from a distribution $\mathcal{D}$ over $\mathbb{R}^d$ and wish to estimate the top eigenvector of the covariance matrix $\mathbb{E}_{a \sim \mathcal{D}} \left[aa^\top\right]$. In the offline case, our algorithms are the fastest to date in a wide and meaningful regime of parameters. Notably, while the running time of most popular methods for eigenvector computations is a product of the size of the dataset (i.e. number of non-zeros in $\mathbf{A}$) and certain spectral characteristics of $\mathbf{A}$, which both can be quite large in practice, we present running times that actually split the dependency between these two quantities, and as a result may yield significant speedups. In the online case, our results yield improved sample complexity bounds and allow for very efficient *streaming* implementations with memory and processing-time requirements that are proportional to the size of a single sample.

On a high-level, our algorithms are based on a robust analysis of the classic idea of *shift-and-invert* preconditioning [Saa92], which allows us to efficiently reduce eigenvector computation to *approximately* solving a *short* sequence of *well-conditioned* linear systems in $\lambda \mathbf{I} - \mathbf{A}^\top \mathbf{A}$ for some shift parameter $\lambda \approx \lambda_1(\mathbf{A})$. We then apply state-of-the-art stochastic gradient methods to approximately solve these linear systems.

1.1 Our Approach

The well known power method for computing the top eigenvector of $\mathbf{A}^\top \mathbf{A}$ starts with an initial vector x and repeatedly multiplies by $\mathbf{A}^\top \mathbf{A}$, eventually causing x to converge to the top eigenvector. For a random start vector, the power method converges in $O(\log(d/\epsilon)/\text{gap})$ iterations, where $\text{gap} = (\lambda_1 - \lambda_2)/\lambda_1$, λ_i denotes the i^{th} largest eigenvalue of $\mathbf{A}^\top \mathbf{A}$, and we assume a high-accuracy regime where $\epsilon < \text{gap}$. The dependence on this gap ensures that the largest eigenvalue is significantly amplified in comparison to the remaining values.

If the eigenvalue gap is small, one approach is replace $\mathbf{A}^\top \mathbf{A}$ with a preconditioned matrix – i.e. a matrix with the same top eigenvector but a much larger gap. Specifically, let $\mathbf{B} = \lambda \mathbf{I} - \mathbf{A}^\top \mathbf{A}$ for some shift parameter λ. If $\lambda > \lambda_1$, we can see that the smallest eigenvector of $\mathbf{B}$ (the largest eigenvector of $\mathbf{B}^{-1}$) is equal to the largest eigenvector of $\mathbf{A}^\top \mathbf{A}$. Additionally, if λ is close to λ_1, there will be a constant gap between the largest and second largest values of $\mathbf{B}^{-1}$. For example, if $\lambda = (1 + \text{gap})\lambda_1$, then we will have $\lambda_1\left(\mathbf{B}^{-1}\right) = \frac{1}{\lambda - \lambda_1} = \frac{1}{\text{gap} \cdot \lambda_1}$ and $\lambda_2\left(\mathbf{B}^{-1}\right) = \frac{1}{\lambda - \lambda_2} = \frac{1}{2 \cdot \text{gap} \cdot \lambda_1}$.

This constant factor gap ensures that the power method applied to $\mathbf{B}^{-1}$ converges to the top eigenvector of $\mathbf{A}^\top \mathbf{A}$ in just $O(\log(d/\epsilon))$ iterations. Of course, there is a catch – each iteration of this *shifted-and-inverted power method* must solve a linear system in $\mathbf{B}$, whose condition number is proportional $\frac{1}{\text{gap}}$. For small gap, solving this system via iterative methods is more expensive.

Fortunately, linear system solvers are incredibly well studied and there are many efficient iterative algorithms we can adapt to apply $\mathbf{B}^{-1}$ approximately. In particular, we show how to accelerate the iterations of the shifted-and-inverted power method using variants of Stochastic Variance Re-

duced Gradient (SVRG) [JZ13]. Due to the condition number of $\mathbf{B}$, we will not entirely avoid a $\frac{1}{\text{gap}}$ dependence, however, we can separate this dependence from the input size $\text{nnz}(\mathbf{A})$.

Typically, stochastic gradient methods are used to optimize convex functions that are given as the sum of many convex components. To solve a linear system $(\mathbf{M}^\top\mathbf{M})x = b$ we minimize the convex function $f(x) = \frac{1}{2}x^\top(\mathbf{M}^\top\mathbf{M})x - b^\top x$ with components $\psi_i(x) = \frac{1}{2}x^\top\left(m_i m_i^\top\right)x - \frac{1}{n}b^\top x$ where m_i is the i^{th} row of $\mathbf{M}$. Such an approach can be used to solve systems in $\mathbf{A}^\top\mathbf{A}$, however solving systems in $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ requires more care. We require an analysis of SVRG that guarantees convergence even when some of our components are *non-convex*. We give a simple analysis for this setting, generalizing recent work in the area [SS15, CR15].

Given fast approximate solvers for $\mathbf{B}$, the second main piece of our algorithmic framework is a new error bound for the shifted-and-inverted power method, showing that it is robust to approximate linear system solvers, such as SVRG. We give a general analysis, showing exactly what accuracy each system must be solved to, allowing for faster implementations using linear solvers with weaker guarantees. Our proofs center around the potential function: $G(x) \stackrel{\text{def}}{=} \left\|\mathbf{P}_{v_1^\perp}x\right\|_{\mathbf{B}} / \left\|\mathbf{P}_{v_1}x\right\|_{\mathbf{B}}$, where $\mathbf{P}_{v_1}$ and $\mathbf{P}_{v_1^\perp}$ are the projections onto the top eigenvector and its complement respectively. This function resembles tangent based potential functions used in previous work [HP14] except that we use the $\mathbf{B}$ norm rather than the ℓ_2 norm. For the exact power method, this is irrelevant – progress is identical in both norms (see Lemma 38 of the Appendix). However, $\|\cdot\|_{\mathbf{B}}$ is a natural norm for measuring the progress of linear system solvers for $\mathbf{B}$, so our potential function makes it possible to extend analysis to the case when $\mathbf{B}^{-1}x$ is computed up to error ξ with bounded $\|\xi\|_{\mathbf{B}}$.

1.2 Our Results

Our algorithmic framework described above offers several advantageous. We obtain improved running times for computing the top eigenvector in the offline model. In Theorem 16 we give an algorithm running in time $O\left(\left[\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}\,\mathbf{A}}{\text{gap}^2}\right]\cdot\left[\log\frac{1}{\epsilon} + \log^2\frac{d}{\text{gap}}\right]\right)$, where $\text{sr}(\mathbf{A}) = \|\mathbf{A}\|_F^2 / \|\mathbf{A}\|_2^2 \leq \text{rank}(\mathbf{A})$ is the stable rank and $\text{nnz}(\mathbf{A})$ is the number of non-zero entries. Up to log factors, our runtime is in many settings proportional to the input size $\text{nnz}(\mathbf{A})$, and so is very efficient for large matrices. In the case when $\text{nnz}(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}$ we apply the results of [FGKS15b, LMH15] to provide an accelerated runtime of $O\left(\left[\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}}\right]\cdot\left[\log\frac{d}{\text{gap}}\log\frac{1}{\epsilon} + \log^3\frac{d}{\text{gap}}\right]\right)$, shown in Theorem 17. Finally, in the case when $\epsilon > \text{gap}$, our results easily extend to give gap-free bounds (Theorems 35 and 36), identical to those shown above but with gap replaced by ϵ. Note that our offline results hold for any $\mathbf{A}$ and require no initial knowledge of the top eigenvector. In Section 6 we discuss how to estimate the parameters λ_1, gap, with modest additional runtime cost.

Our algorithms return an approximate top eigenvector x with $x^\top\mathbf{A}^\top\mathbf{A}x \geq (1-\epsilon)\lambda_1$. By choosing error $\epsilon\cdot\text{gap}$, we can ensure that x is actually close to v_1 – i.e. that $|x^\top v_1| \geq 1-\epsilon$. Further, we obtain the same asymptotic runtime since $O\left(\log\frac{1}{\epsilon\cdot\text{gap}} + \log^2\frac{d}{\text{gap}}\right) = O\left(\log\frac{1}{\epsilon} + \log^2\frac{d}{\text{gap}}\right)$. We compare our runtimes with previous work in Table 1.

In the online case, in Theorem 25, we show how to improve an $O(\text{gap})$ approximation to the top eigenvector to an ϵ approximation with constant probability using $O\left(\frac{\text{v}(\mathcal{D})}{\text{gap}\cdot\epsilon}\right)$ samples where $\text{v}(\mathcal{D})$ is a natural variance measure. Our algorithm is based on the streaming SVRG algorithm of [FGKS15a]. It requires just $O(d)$ amortized time per sample, uses just $O(d)$ space, and is easily parallelized. We can apply our result in a variety of regimes, using existing algorithms to obtain

the initial $O(\text{gap})$ approximation and our algorithm to improve. As shown in Table 2, this gives improved runtimes and sample complexities over existing work. Notably, we give better asymptotic sample complexity than known matrix concentration results for general distributions, and give the first streaming algorithm that is asymptotically optimal in the popular Gaussian spike model.

Overall, our robust shifted-and-inverted power method analysis gives new understanding of this classical technique. It gives a means of obtaining provably accurate results when each iteration is implemented using fast linear system solvers with weak accuracy guarantees. In practice, this reduction between approximate linear system solving and eigenvector computation shows that optimized regression libraries can be leveraged for faster eigenvector computation in many cases. Furthermore, in theory we believe that the reduction suggests computational limits inherent in eigenvector computation as seen by the often easier-to-analyze problem of linear system solving. Indeed, in Section 7, we provide evidence that in certain regimes our statistical results are optimal.

1.3 Previous Work

Offline Eigenvector Computation

Due to its universal applicability, eigenvector computation in the offline case is extremely well studied. Classical methods, such as the QR algorithm, take roughly $O(nd^2)$ time to compute a full eigendecomposition. This can be accelerated to $O(nd^{\omega-1})$, where $\omega < 2.373$ is the matrix multiplication constant [Wil12, LG14], however this is still prohibitively expensive for large matrices. Hence, faster iterative methods are often employed, especially when only the top eigenvector (or a few of the top eigenvectors) is desired.

As discussed, the popular power method requires $O\left(\frac{\log(d/\epsilon)}{\text{gap}}\right)$ iterations to converge to an ϵ approximate top eigenvector. Using Chebyshev iteration, or more commonly, the Lanczos method, this bound can be improved to $O\left(\frac{\log(d/\epsilon)}{\sqrt{\text{gap}}}\right)$ [Saa92], giving total runtime of $O\left(\text{nnz}(\mathbf{A}) \cdot \frac{\log(d/\epsilon)}{\sqrt{\text{gap}}}\right)$. When $\epsilon > \text{gap}$, the gap terms in these runtimes can be replaced by ϵ. While we focus on the high-precision regime when $\epsilon < \text{gap}$, we also give gap-free bounds in Section 8.

Unfortunately, if $\text{nnz}(\mathbf{A})$ is very large and gap is small, the above runtimes can still be quite expensive, and there is a natural desire to separate the $\frac{1}{\sqrt{\text{gap}}}$ dependence from the $\text{nnz}(\mathbf{A})$ term. One approach is to use random subspace embedding matrices [AC09, CW13] or fast row sampling algorithms [CLM+15], which can be applied in $O(\text{nnz}(\mathbf{A}))$ time and yield a matrix $\tilde{\mathbf{A}}$ which is a good spectral approximation to the original. The number of rows in $\tilde{\mathbf{A}}$ depends only on the stable rank of $\mathbf{A}$ and the error of the embedding – hence it can be significantly smaller than n. Applying such a subspace embedding and then computing the top eigenvector of $\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}$ requires runtime $O\left(\text{nnz}(\mathbf{A}) + \text{poly}(\text{sr}(\mathbf{A}), \epsilon, \text{gap})\right)$, achieving the goal of reducing runtime dependence on the input size $\text{nnz}(\mathbf{A})$. Unfortunately, the dependence on ϵ is significantly suboptimal – such an approach cannot be used to obtain a linearly convergent algorithm. Further, the technique does not extend to the online setting, unless we are willing to store a full subspace embedding of our sampled rows.

Another approach, which we follow more closely, is to apply stochastic optimization techniques, which iteratively update an estimate to the top eigenvector, considering a random row of $\mathbf{A}$ with each update step. Such algorithms naturally extend to the online setting and have led to improved dependence on the input size for a variety of problems [Bot10]. Using variance-reduced stochastic gradient techniques, [Sha15c] achieves runtime $O\left(\left(\text{nnz}(\mathbf{A}) + \frac{dr^2n^2}{\text{gap}^2\lambda_1^2}\right) \cdot \log(1/\epsilon)\log\log(1/\epsilon)\right)$ for approximately computing the top eigenvector of a matrix with constant probability. Here r is an

upper bound on the squared row norms of $\mathbf{A}$. In the *best case*, when row norms are uniform, this runtime can be simplified to $O\left(\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})^2}{\text{gap}^2}\right) \cdot \log(1/\epsilon)\log\log(1/\epsilon)\right)$.

The result in [Sha15c] makes an important contribution in separating input size and gap dependencies using stochastic optimization techniques. Unfortunately, the algorithm requires an approximation to the eigenvalue gap and a starting vector that has a constant dot product with the top eigenvector. In [Sha15b] the analysis is extended to a random initialization, however loses polynomial factors in d. Furthermore, the dependencies on the stable rank and ϵ are suboptimal – we improve them to $\text{sr}(\mathbf{A})$ and $\log(1/\epsilon)$ respectively, obtaining true linear convergence.

Algorithm	**Runtime**
Power Method	$O\left(\text{nnz}(\mathbf{A})\frac{\log(d/\epsilon)}{\text{gap}}\right)$
Lanczos Method	$O\left(\text{nnz}(\mathbf{A})\frac{\log(d/\epsilon)}{\sqrt{\text{gap}}}\right)$
Fast Subspace Embeddings [CW13] Plus Lanczos	$O\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\max\{\text{gap}^{2.5}\epsilon,\epsilon^{2.5}\}}\right)$
SVRG [Sha15c] (assuming bounded row norms, warm-start)	$O\left(\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})^2}{\text{gap}^2}\right) \cdot \log(1/\epsilon)\log\log(1/\epsilon)\right)$
Theorem 16	$O\left(\left[\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right] \cdot \left[\log\frac{1}{\epsilon} + \log^2\frac{d}{\text{gap}}\right]\right)$
Theorem 17	$O\left(\left[\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}}\right] \cdot \left[\log\frac{d}{\text{gap}}\log\frac{1}{\epsilon} + \log^3\frac{d}{\text{gap}}\right]\right)$

Table 1: Comparision to previous work on Offline Eigenvector Estimation. We give runtimes for computing a unit vector x such that $x^\top \mathbf{A}^\top \mathbf{A} x \geq (1-\epsilon)\lambda_1$ in the regime $\epsilon = O(\text{gap})$.

Online Eigenvector Computation

While in the offline case the primary concern is computation time, in the online, or statistical setting, research also focuses on minimizing the number of samples that are drawn from $\mathcal{D}$ in order to achieve a given accuracy. Especially sought after are results that achieve asymptotically optimal accuracy as the sample size grows large.

While the result we give in Theorem 25 works for any distribution parameterized by a variance bound, in this section, in order to more easily compare to previous work, we normalize $\lambda_1 = 1$ and assume we have the row norm bound $\|a\|_2^2 \leq O(d)$ which then gives us the variance bound $\left\|\mathbb{E}_{a\sim\mathcal{D}}\left[(aa^\top)^2\right]\right\|_2 = O(d)$. Additionally, we compare runtimes for computing some x such that $|x^\top v_1| \geq 1-\epsilon$, as this is the most popular guarantee studied in the literature. Theorem 25 is easily extended to this setting as obtaining x with $x^T\mathbf{A}\mathbf{A}^\top x \geq (1-\epsilon\cdot\text{gap})\lambda_1$ ensures $|x^\top v_1| \geq 1-\epsilon$. Our algorithm requires $O\left(\frac{d}{\text{gap}^2\epsilon}\right)$ samples to find such a vector under the assumptions given above.

The simplest algorithm in this setting is to take n samples from $\mathcal{D}$ and compute the leading eigenvector of the empirical estimate $\widehat{\mathbb{E}}[aa^\top] = \frac{1}{n}\sum_{i=1}^n a_i a_i^\top$. By a matrix Bernstein bound, such as inequality of Theorem 6.6.1 of [Tro15], $O\left(\frac{d\log d}{\text{gap}^2\epsilon}\right)$ samples is enough to insure $\left\|\widehat{\mathbb{E}}[aa^\top] - \mathbb{E}[aa^\top]\right\|_2 \leq \sqrt{\epsilon}\cdot\text{gap}$. By Lemma 37 in the Appendix, this gives that, if x is set to the top eigenvector of $\widehat{\mathbb{E}}[aa^\top]$ it will satisfy $|x^\top v_1| \geq 1-\epsilon$. x can be approximated with any offline eigenvector algorithm.

A large body of work focuses on improving this simple algorithm, under a variety of assumptions on $\mathcal{D}$. A common focus is on obtaining *streaming algorithms*, in which the storage space is just $O(d)$ - proportional to the size of a single sample. In Table 2 we give a sampling of results in this

area. All listed results rely on distributional assumptions at least as strong as those given above.

Note that, in each setting, we can use the cited algorithm to first compute an $O(\text{gap})$ approximate eigenvector, and then refine this approximation to an ϵ approximation using $O\left(\frac{d}{\text{gap}^2\epsilon}\right)$ samples by applying our streaming SVRG based algorithm. This allows us to obtain improved runtimes and sample complexities. To save space, we do not include our improved runtime bounds in Table 2, however they are easy to derive by adding the runtime required by the given algorithm to achieve $O(\text{gap})$ accuracy, to $O\left(\frac{d^2}{\text{gap}^2\epsilon}\right)$ – the runtime required by our streaming algorithm.

The bounds given for the simple matrix Bernstein based algorithm described above, Krasulina/Oja's Algorithm [BDF13], and SGD [Sha15a] require no additional assumptions, aside from those given at the beginning of this section. The streaming results cited for [MCJ13] and [HP14] assume a is generated from a Gaussian spike model, where $a_i = \sqrt{\lambda_1}\gamma_i v_1 + Z_i$ and $\gamma_i \sim \mathcal{N}(0,1)$, $Z_i \sim \mathcal{N}(0, I_d)$. We note that under this model, the matrix Bernstein results improve by a $\log d$ factor and so match our results in achieving asymptotically optimal convergence rate. The results of [MCJ13] and [HP14] sacrifice this optimality in order to operate under the streaming model. Our work gives the best of both works – a streaming algorithm giving asymptotically optimal results.

The streaming Alecton algorithm [SRO15] assumes $\mathbb{E}\left\|aa^\top \mathbf{W} aa^\top\right\| \leq O(1)\text{tr}(\mathbf{W})$ for any symmetric $\mathbf{W}$ that commutes with $\mathbb{E}aa^\top$. This is strictly stronger than our assumption that $\left\|\mathbb{E}_{a\sim\mathcal{D}}\left[(aa^\top)^2\right]\right\|_2 = O(d)$.

Algorithm	**Sample Size**	**Runtime**	**Streaming?**	**Our Sample Complexity**
Matrix Bernstein plus Lanczos (explicitly forming sampled matrix)	$O\left(\frac{d\log d}{gap^2\epsilon}\right)$	$O\left(\frac{d^3\log d}{gap^2\epsilon}\right)$	$\times$	$O\left(\frac{d\log d}{gap^3} + \frac{d}{gap^2\epsilon}\right)$
Matrix Bernstein plus Lanczos (iteratively applying sampled matrix)	$O\left(\frac{d\log d}{gap^2\epsilon}\right)$	$O\left(\frac{d^2\log d\cdot\log(d/\epsilon)}{gap^{2.5}\epsilon}\right)$	$\times$	$O\left(\frac{d\log d}{gap^3} + \frac{d}{gap^2\epsilon}\right)$
Memory-efficient PCA [MCJ13, HP14]	$O\left(\frac{d\log(d/\epsilon)}{gap^3\epsilon}\right)$	$O\left(\frac{d^2\log(d/\epsilon)}{gap^3\epsilon}\right)$	$\surd$	$O\left(\frac{d\log(d/\text{gap})}{gap^4} + \frac{d}{gap^2\epsilon}\right)$
Alecton [SRO15]	$O(\frac{d\log(d/\epsilon)}{gap^2\epsilon})$	$O(\frac{d^2\log(d/\epsilon)}{gap^2\epsilon})$	$\surd$	$O(\frac{d\log(d/\text{gap})}{gap^3} + \frac{d}{gap^2\epsilon})$
Krasulina / Oja's Algorithm [BDF13]	$O(\frac{d^{c_1}}{gap^2\epsilon^{c_2}})$	$O(\frac{d^{c_1+1}}{gap^2\epsilon^{c_2}})$	$\surd$	$O(\frac{d^{c_1}}{gap^{2+c_2}} + \frac{d}{gap^2\epsilon})$
SGD [Sha15a]	$O(\frac{d^3\log(d/\epsilon)}{\epsilon^2})$	$O(\frac{d^4\log(d/\epsilon)}{\epsilon^2})$	$\surd$	$O\left(\frac{d^3\log(d/\text{gap})}{\text{gap}^2} + \frac{d}{\text{gap}^2\epsilon}\right)$

Table 2: Summary of existing work on Online Eigenvector Estimation and improvements given by our results. Runtimes are for computing a unit vector x such that $|x^\top v_1| \geq 1-\epsilon$. For each of these results we can obtain improved running times and sample complexities by running the algorithm to first compute an $O(\text{gap})$ approximate eigenvector, and then running our algorithm to obtain an ϵ approximation using an additional $O\left(\frac{d}{\text{gap}^2\epsilon}\right)$ samples, $O(d)$ space, and $O(d)$ work per sample.

1.4 Paper Organization

Section 2 Review problem definitions and parameters for our runtime and sample bounds.

Section 3 Describe the shifted-and-inverted power method and show how it can be implemented using approximate system solvers.

Section 4 Show how to apply SVRG to solve systems in our shifted matrix, giving our main runtime results for offline eigenvector computation.

Section 5 Show how to use an online variant of SVRG to run the shifted-and-inverted power method, giving our main sampling complexity and runtime results in the statistical setting.

Section 6 Show how to efficiently estimate the shift parameters required by our algorithms.

Section 7 Give a lower bound in the statistical setting, showing that our results are asymptotically optimal for a wide parameter range.

Section 8 Give gap-free runtime bounds, which apply when $\epsilon >$ gap.

2 Preliminaries

We bold all matrix variables. We use $[n] \stackrel{\text{def}}{=} \{1, ..., n\}$. For a symmetric positive semidefinite (PSD) matrix $\mathbf{M}$ we let $\|x\|_{\mathbf{M}} \stackrel{\text{def}}{=} \sqrt{x^\top \mathbf{M} x}$ and $\lambda_1(\mathbf{M}), ..., \lambda_d(\mathbf{M})$ denote its eigenvalues in decreasing order. We use $\mathbf{M} \preceq \mathbf{N}$ to denote the condition that $x^\top \mathbf{M} x \leq x^\top \mathbf{N} x$ for all x.

2.1 The Offline Problem

We are given a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ with rows $a^{(1)}, ..., a^{(n)}$ and wish to compute an approximation to the top eigenvector of $\mathbf{\Sigma} \stackrel{\text{def}}{=} \mathbf{A}^\top \mathbf{A}$. Specifically, for error parameter ϵ we want a unit vector x such that $x^\top \mathbf{\Sigma} x \geq (1 - \epsilon)\lambda_1(\mathbf{\Sigma})$.

2.2 The Statistical Problem

We have access to an oracle returning independent samples from a distribution $\mathcal{D}$ on $\mathbb{R}^d$ and wish to compute the top eigenvector of $\mathbf{\Sigma} \stackrel{\text{def}}{=} \mathbb{E}_{a \sim \mathcal{D}} \left[aa^\top\right]$. Again, for error parameter ϵ we want to return a unit vector x such that $x^\top \mathbf{\Sigma} x \geq (1 - \epsilon)\lambda_1(\mathbf{\Sigma})$.

2.3 Problem Parameters

We parameterize the running times and sample complexities of our algorithms in terms of several natural properties of $\mathbf{A}$, $\mathcal{D}$, and $\mathbf{\Sigma}$. Let $\lambda_1, ..., \lambda_d$ denote the eigenvalues of $\mathbf{\Sigma}$ in decreasing order and $v_1, ..., v_d$ denote their corresponding eigenvectors. We define the *eigenvalue gap* by $\text{gap} \stackrel{\text{def}}{=} \frac{\lambda_1 - \lambda_2}{\lambda_1}$.

We use the following additional parameters for the offline and statistical problems respectively:

- **Offline Problem**: Let $\text{sr}(\mathbf{A}) \stackrel{\text{def}}{=} \sum_i \frac{\lambda_i}{\lambda_1} = \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_2^2}$ denote the stable rank of $\mathbf{A}$. Note that we always have $\text{sr}(\mathbf{A}) \leq \text{rank}(\mathbf{A})$. Let $\text{nnz}(\mathbf{A})$ denote the number of non-zero entries in $\mathbf{A}$.
- **Online Problem**: Let $\text{v}(\mathcal{D}) \stackrel{\text{def}}{=} \frac{\left\|\mathbb{E}_{a \sim \mathcal{D}}\left[\left(aa^\top\right)^2\right]\right\|_2}{\left\|\mathbb{E}_{a \sim \mathcal{D}}(aa^\top)\right\|_2^2} = \frac{\left\|\mathbb{E}_{a \sim \mathcal{D}}\left[\left(aa^\top\right)^2\right]\right\|_2}{\lambda_1^2}$ denote a natural upper bound on the variance of $\mathcal{D}$ in various settings. Note that $\text{v}(\mathcal{D}) \geq 1$.

3 Algorithmic Framework

Here we develop our robust shift-and-invert framework. In Section 3.1 we provide a basic overview of the framework and in Section 3.2 we introduce the potential function we use to measure progress of our algorithms. In Section 3.3 we show how to analyze the framework given access to an exact linear system solver and in Section 3.4 we strengthen this analysis to work with an inexact linear system solver. Finally, in Section 3.5 we discuss initializing the framework.

3.1 Shifted-and-Inverted Power Method Basics

We let $\mathbf{B}_\lambda \stackrel{\text{def}}{=} \lambda \mathbf{I} - \mathbf{\Sigma}$ denote the shifted matrix that we will use in our implementation of the shifted-and-inverted power method. As discussed, in order for $\mathbf{B}_\lambda^{-1}$ to have a large eigenvalue gap, λ should be set to $(1 + c \cdot \text{gap})\lambda_1$ for some constant $c \geq 0$. Throughout this section we assume that we have a crude estimate of λ_1 and gap and fix λ to be a value satisfying $\left(1 + \frac{\text{gap}}{150}\right)\lambda_1 \leq \lambda \leq \left(1 + \frac{\text{gap}}{100}\right)\lambda_1$. (See Section 6 for how we can compute such a λ). For the remainder of this section we work with such a fixed value of λ and therefore for convenience denote $\mathbf{B}_\lambda$ as $\mathbf{B}$.

Note that $\lambda_i\left(\mathbf{B}^{-1}\right) = \frac{1}{\lambda_i(\mathbf{B})} = \frac{1}{\lambda - \lambda_i}$ and so $\frac{\lambda_1(\mathbf{B}^{-1})}{\lambda_2(\mathbf{B}^{-1})} = \frac{\lambda - \lambda_2}{\lambda - \lambda_1} \geq \frac{\text{gap}}{\text{gap}/100} = 100$. This large gap will ensure that, assuming the ability to apply $\mathbf{B}^{-1}$, the power method will converge very quickly. In the remainder of this section we develop our error analysis for the shifted-and-inverted power method which demonstrates that approximate application of $\mathbf{B}^{-1}$ in each iteration in fact suffices.

3.2 Potential Function

Our analysis of the power method focuses on the objective of maximizing the Rayleigh quotient, $x^\top \mathbf{\Sigma} x$ for a unit vector x. Note that as the following lemma shows, this has a direct correspondence to the error in maximizing $|v_1^\top x|$:

Lemma 1 (Bounding Eigenvector Error by Rayleigh Quotient)**.** *For a unit vector x let $\epsilon = \lambda_1 - x^\top \mathbf{\Sigma} x$. If $\epsilon \leq \lambda_1 \cdot \text{gap}$ then*

$$\left|v_1^\top x\right| \geq \sqrt{1 - \frac{\epsilon}{\lambda_1 \cdot \text{gap}}}.$$

Proof. Among all unit vectors x such that $\epsilon = \lambda_1 - x^\top \mathbf{\Sigma} x$, a minimizer of $\left|v_1^\top x\right|$ has the form $x = (\sqrt{1 - \delta^2})v_1 + \delta v_2$ for some δ. We have

$$\epsilon = \lambda_1 - x^\top \mathbf{\Sigma} x = \lambda_1 - \lambda_1(1 - \delta^2) - \lambda_2 \delta^2 = (\lambda_1 - \lambda_2)\delta^2.$$

Therefore by direct computation,

$$\left|v_1^\top x\right| = \sqrt{1 - \delta^2} = \sqrt{1 - \frac{\epsilon}{\lambda_1 - \lambda_2}} = \sqrt{1 - \frac{\epsilon}{\lambda_1 \cdot \text{gap}}}\,.$$

□

In order to track the progress of our algorithm we use a more complex potential function than just the Rayleigh quotient error, $\lambda_1 - x^\top \mathbf{\Sigma} x$. Our potential function G is defined for $x \neq 0$ by

$$G(x) \stackrel{\text{def}}{=} \frac{\left\|\mathbf{P}_{v_1^\perp} x\right\|_{\mathbf{B}}}{\|\mathbf{P}_{v_1} x\|_{\mathbf{B}}}$$

where $\mathbf{P}_{v_1}$ and $\mathbf{P}_{v_1^\perp}$ are the projections onto v_1 and the subspace orthogonal to v_1 respectively. Equivalently, we have that:

$$G(x) = \frac{\sqrt{\|x\|_{\mathbf{B}}^2 - \left(v_1^\top \mathbf{B}^{1/2} x\right)^2}}{\left|v_1^\top \mathbf{B}^{1/2} x\right|} = \frac{\sqrt{\sum_{i \geq 2} \frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}}}{\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}}}. \tag{1}$$

where $\alpha_i = v_i^\top x$.

When the Rayleigh quotient error $\epsilon = \lambda_1 - x^\top \mathbf{\Sigma} x$ of x is small, we can show a strong relation between ϵ and $G(x)$. We prove this in two parts. We first give a technical lemma, Lemma 2, that we will use several times for bounding the numerator of G. We then prove the connection in Lemma 3.

Lemma 2. *For a unit vector x and $\epsilon = \lambda_1 - x^\top \mathbf{\Sigma} x$ if $\epsilon \leq \lambda_1 \cdot \text{gap}$ then*

$$\epsilon \leq x^\top \mathbf{B} x - (v_1^\top \mathbf{B} x)(v_1^\top x) \leq \epsilon \left(1 + \frac{\lambda - \lambda_1}{\lambda_1 \cdot \text{gap}}\right).$$

Proof. Since $\mathbf{B} = \lambda \mathbf{I} - \mathbf{\Sigma}$ and since v_1 is an eigenvector of $\mathbf{\Sigma}$ with eigenvalue λ_1 we have

$$\begin{aligned} x^\top \mathbf{B} x - (v_1^\top \mathbf{B} x)(v_1^\top x) &= \lambda \|x\|_2^2 - x^\top \mathbf{\Sigma} x - (\lambda v_1^\top x - v_1^\top \mathbf{\Sigma} x)(v_1^\top x) \\ &= \lambda - \lambda_1 + \epsilon - (\lambda v_1^\top x - \lambda_1 v_1^\top x)(v_1^\top x) \\ &= (\lambda - \lambda_1)\left(1 - (v_1^\top x)^2\right) + \epsilon. \end{aligned}$$

Now by Lemma 1 we know that $|v_1^\top x| \geq \sqrt{1 - \frac{\epsilon}{\lambda_1 \cdot \text{gap}}}$, giving us the upper bound. Furthermore, since trivially $\left|v_1^\top x\right| \leq 1$ and $\lambda - \lambda_1 > 0$, we have the lower bound. □

Lemma 3 (Potential Function to Rayleigh Quotient Error Conversion)**.** *For a unit vector x and $\epsilon = \lambda_1 - x^\top \mathbf{\Sigma} x$ if $\epsilon \leq \frac{1}{2}\lambda_1 \cdot \text{gap}$, we have:*

$$\frac{\epsilon}{\lambda - \lambda_1} \leq G(x)^2 \leq \left(1 + \frac{\lambda - \lambda_1}{\lambda_1 \cdot \text{gap}}\right)\left(1 + \frac{2\epsilon}{\lambda_1 \cdot \text{gap}}\right)\frac{\epsilon}{\lambda - \lambda_1}.$$

Proof. Since v_1 is an eigenvector of $\mathbf{B}$, we can write $G(x)^2 = \frac{x^\top \mathbf{B} x - (v_1^\top \mathbf{B} x)(v_1^\top x)}{(v_1^\top \mathbf{B} x)(v_1^\top x)}$. Lemmas 1 and 2 then give us:

$$\frac{\epsilon}{\lambda - \lambda_1} \leq G(x)^2 \leq \left(1 + \frac{\lambda - \lambda_1}{\lambda_1 \cdot \text{gap}}\right)\frac{\epsilon}{(\lambda - \lambda_1)\left(1 - \frac{\epsilon}{\lambda_1 \cdot \text{gap}}\right)}.$$

Since $\epsilon \leq \frac{1}{2}\lambda_1 \cdot \text{gap}$, we have $\frac{1}{1 - \frac{\epsilon}{\lambda_1 \cdot \text{gap}}} \leq 1 + \frac{2\epsilon}{\lambda_1 \cdot \text{gap}}$. This proves the lemma. □

3.3 Power Iteration

Here we show that the shifted-and-inverted power iteration in fact makes progress with respect to our objective function given an exact linear system solver for $\mathbf{B}$. Formally, we show that applying $\mathbf{B}^{-1}$ to a vector x decreases the potential function $G(x)$ geometrically.

Theorem 4. *Let x be a unit vector with $\langle x, v_1\rangle \neq 0$ and let $\widetilde{x} = \mathbf{B}^{-1}x$, i.e. the power method update of $\mathbf{B}^{-1}$ on x. Then, under our assumption on λ, we have:*

$$G(\widetilde{x}) \leq \frac{\lambda_2\left(\mathbf{B}^{-1}\right)}{\lambda_1\left(\mathbf{B}^{-1}\right)} G(x) \leq \frac{1}{100} G(x).$$

Note that $\widetilde{x}$ may no longer be a unit vector. However, $G(\widetilde{x}, v_1) = G(c\widetilde{x}, v_1)$ for any scaling parameter c, so the theorem also holds for $\widetilde{x}$ scaled to have unit norm.

Proof. Writing x in the eigenbasis, we have $x = \sum_i \alpha_i v_i$ and $\widetilde{x} = \sum_i \alpha_i \lambda_i\left(\mathbf{B}^{-1}\right) v_i$. Since $\langle x, v_1\rangle \neq 0$, $\alpha_1 \neq 0$ and by the equivalent formulation of $G(x)$ given in (1):

$$G(\widetilde{x}) = \frac{\sqrt{\sum_{i\geq 2} \alpha_i^2 \lambda_i(\mathbf{B}^{-1})}}{\sqrt{\alpha_1^2 \lambda_1(\mathbf{B}^{-1})}} \leq \frac{\lambda_2\left(\mathbf{B}^{-1}\right)}{\lambda_1\left(\mathbf{B}^{-1}\right)} \cdot \frac{\sqrt{\sum_{i\geq 2} \frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}}}{\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}}} = \frac{\lambda_2\left(\mathbf{B}^{-1}\right)}{\lambda_1\left(\mathbf{B}^{-1}\right)} \cdot G(x) \ .$$

Recalling that $\frac{\lambda_1(\mathbf{B}^{-1})}{\lambda_2(\mathbf{B}^{-1})} = \frac{\lambda - \lambda_2}{\lambda - \lambda_1} \geq \frac{\text{gap}}{\text{gap}/100} = 100$ yields the result. □

The challenge in using the above theorem, and any traditional analysis of the shifted-and-inverted power method, is that we don't actually have access to $\mathbf{B}^{-1}$. In the next section we show that the shifted-and-inverted power method is robust – we still make progress on our objective function even if we only approximate $\mathbf{B}^{-1}x$ using a fast linear system solver.

3.4 Approximate Power Iteration

We are now ready to prove our main result. We show that each iteration of the shifted-and-inverted power method makes constant factor expected progress on our potential function assuming we:

1. Start with a sufficiently good x and an approximation of λ_1
2. Can apply $\mathbf{B}^{-1}$ approximately using a system solver such that the function error (i.e. distance to $\mathbf{B}^{-1}x$ in the $\mathbf{B}$ norm) is sufficiently small *in expectation.*
3. Can estimate Rayleigh quotients over $\mathbf{\Sigma}$ well enough to only accept updates that do not hurt progress on the objective function too much.

This third assumption is necessary since the second assumption is quite weak. An expected progress bound on the linear system solver allows, for example, the solver to occasionally return a solution that is entirely orthogonal to v_1, causing us to make unbounded backwards progress on our potential function. The third assumption allows us to reject possibly harmful updates and ensure that we still make progress in expectation. In the offline setting, we can access $\mathbf{A}$ and are able to compute Rayleigh quotients exactly in time $\text{nnz}(\mathbf{A})$ time. However, we only assume the ability to estimate quotients since in the online setting we only have access to $\mathbf{\Sigma}$ through samples from $\mathcal{D}$.

Our general theorem for the approximate power iteration, Theorem 5, assumes that we can solve linear systems to some absolute accuracy in expectation. This is not completely standard. Typically, system solver analysis assumes an initial approximation to $\mathbf{B}^{-1}x$ and then shows a relative progress bound – that the quality of the initial approximation is improved geometrically in each iteration of the algorithm. In Corollary 6 we show how to find a coarse initial approximation

to $\mathbf{B}^{-1}x$, in fact just approximating $\mathbf{B}^{-1}$ with $\frac{1}{x^\top \mathbf{B} x}x$. Using this approximation, we show that Theorem 5 actually implies that traditional system solver relative progress bounds suffice.

Note that in both claims we measure error of the linear system solver using $\|\cdot\|_{\mathbf{B}}$. This is a natural norm in which geometric convergence is shown for many linear system solvers and directly corresponds to the function error of minimizing $f(w) = \frac{1}{2}w^\top \mathbf{B} w - w^\top x$ to compute $\mathbf{B}^{-1}x$.

Theorem 5 (Approximate Shifted-and-Inverted Power Iteration – Warm Start)**.** *Let $x = \sum_i \alpha_i v_i$ be a unit vector such that $G(x) \leq \frac{1}{\sqrt{10}}$. Suppose we know some shift parameter λ with $\left(1 + \frac{\text{gap}}{150}\right)\lambda_1 < \lambda \leq \left(1 + \frac{\text{gap}}{100}\right)\lambda_1$ and an estimate $\widehat{\lambda}_1$ of λ_1 such that $\frac{10}{11}(\lambda - \lambda_1) \leq \lambda - \widehat{\lambda}_1 \leq \lambda - \lambda_1$. Furthermore, suppose we have a subroutine $\text{solve}(\cdot)$ such that on any input x*

$$\mathbb{E}\left[\left\|\text{solve}\left(x\right) - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}\right] \leq \frac{c_1}{1000}\sqrt{\lambda_1(\mathbf{B}^{-1})},$$

for some $c_1 < 1$, and a subroutine $\widehat{\text{quot}}(\cdot)$ that on any input $x \neq 0$

$$\left|\widehat{\text{quot}}\left(x\right) - \text{quot}(x)\right| \leq \frac{1}{30}\left(\lambda - \lambda_1\right) \quad \textit{for all nonzero} \quad x \in \mathbb{R}^d.$$

where $\text{quot}(x) \stackrel{\text{def}}{=} \frac{x^\top \mathbf{\Sigma} x}{x^\top x}$.

Then the following update procedure:

$$\textit{Set } \widehat{x} = \text{solve}\left(x\right),$$

$$\textit{Set } \widetilde{x} = \begin{cases} \widehat{x} & \textit{if } \begin{cases} \widehat{\text{quot}}\left(\widehat{x}\right) \geq \widehat{\lambda}_1 - \left(\lambda - \widehat{\lambda}_1\right)/6 \textit{ and} \\ \|\widehat{x}\|_2 \geq \frac{2}{3}\frac{1}{\lambda - \widehat{\lambda}_1} \end{cases} \\ x & \textit{otherwise,} \end{cases}$$

satisfies the following:

- $G(\widetilde{x}) \leq \frac{1}{\sqrt{10}}$ *and*
- $\mathbb{E}\left[G(\widetilde{x})\right] \leq \frac{3}{25}G(x) + \frac{c_1}{500}$.

That is, not only do we decrease our potential function by a constant factor in expectation, but we are guaranteed that the potential function will never increase beyond $1/\sqrt{10}$.

Proof. The first claim follows directly from our choice of $\widetilde{x}$ from x and $\widehat{x}$. If $\widetilde{x} = x$, it holds trivially by our assumption that $G(x) \leq \frac{1}{\sqrt{10}}$. Otherwise, $\widetilde{x} = \widehat{x}$ and we know that

$$\begin{aligned}\lambda_1 - \text{quot}\left(\widehat{x}\right) \leq \widehat{\lambda}_1 - \text{quot}\left(\widehat{x}\right) &\leq \widehat{\lambda}_1 - \widehat{\text{quot}}\left(\widehat{x}\right) + \left|\widehat{\text{quot}}\left(\widehat{x}\right) - \text{quot}\left(\widehat{x}\right)\right| \\ &\leq \frac{\lambda - \widehat{\lambda}_1}{6} + \frac{\lambda - \lambda_1}{30} \leq \frac{\lambda - \lambda_1}{5} \leq \frac{\lambda_1 \cdot \text{gap}}{500}.\end{aligned}$$

The claim then follows from Lemma 3 as

$$\begin{aligned}G(\widehat{x})^2 &\leq \left(1 + \frac{\lambda - \lambda_1}{\lambda_1 \cdot \text{gap}}\right)\left(1 + \frac{2\left(\lambda_1 - \text{quot}\left(\widehat{x}\right)\right)}{\lambda_1 \cdot \text{gap}}\right)\frac{\lambda_1 - \text{quot}\left(\widehat{x}\right)}{\lambda - \lambda_1} \\ &\leq \frac{101}{100} \cdot \frac{251}{250} \cdot \frac{\left(\frac{\lambda_1 \cdot \text{gap}}{500}\right)}{\left(\frac{\lambda_1 \cdot \text{gap}}{150}\right)} \leq \frac{1}{\sqrt{10}}.\end{aligned}$$

All that remains is to show the second claim, that $\mathbb{E}\left[G(\widetilde{x})\right] \leq \frac{3}{25}G(x) + \frac{4c_1}{1000}$. Let $\mathcal{F}$ denote the event that we accept our iteration and set $x = \widehat{x} = \text{solve}\left(x\right)$. That is:

$$\mathcal{F} \stackrel{\text{def}}{=} \left\{\widehat{\text{quot}}\left(\widehat{x}\right) \geq \widehat{\lambda}_1 - \frac{\lambda - \widehat{\lambda}_1}{6}\right\} \cup \left\{\|\widehat{x}\|_2 \geq \frac{2}{3}\frac{1}{\lambda - \widehat{\lambda}_1}\right\}.$$

Using our bounds on $\widehat{\lambda}_1$ and $\widehat{\text{quot}}\left(\cdot\right)$, we know that $\widehat{\text{quot}}\left(x\right) \leq \text{quot}(x) + (\lambda - \lambda_1)/30$ and $\lambda - \widehat{\lambda}_1 \leq \lambda - \lambda_1$. Therefore, since $-1/6 - 1/30 \geq -1/2$ we have

$$\mathcal{F} \subseteq \left\{\text{quot}\left(\widehat{x}\right) \geq \lambda_1 - \left(\lambda - \lambda_1\right)/2\right\} \cup \left\{\|\widehat{x}\|_2 \geq \frac{2}{3}\frac{1}{\lambda - \lambda_1}\right\},$$

We will complete the proof in two steps. First we let $\xi \stackrel{\text{def}}{=} \widehat{x} - \mathbf{B}^{-1}x$ and show that assuming $\mathcal{F}$ is true then $G(\widehat{x})$ and $\|\xi\|_{\mathbf{B}}$ are linearly related, i.e. expected error bounds on $\|\xi\|_{\mathbf{B}}$ correspond to expected error bounds on $G(\widehat{x})$. Second, we bound the probability that $\mathcal{F}$ does not occur and bound error incurred in this case. Combining these yields the result.

To show the linear relationship in the case where $\mathcal{F}$ is true, first note Lemma 1 shows that in this case $\left|v_1^\top \frac{\widehat{x}}{\|\widehat{x}\|_2}\right| \geq \sqrt{1 - \frac{\lambda_1 - \text{quot}(\widehat{x})}{\lambda_1 \cdot \text{gap}}} \geq \frac{3}{4}$. Consequently,

$$\|\mathbf{P}_{v_1}\widehat{x}\|_{\mathbf{B}} = \left|v_1^\top \widehat{x}\right| \sqrt{\lambda - \lambda_1} = \left|v_1^\top \frac{\widehat{x}}{\|\widehat{x}\|_2}\right| \cdot \|\widehat{x}\| \sqrt{\lambda - \lambda_1} \geq \frac{3}{4} \cdot \frac{2}{3}\frac{1}{\sqrt{\lambda - \lambda_1}} = \frac{\sqrt{\lambda_1(\mathbf{B}^{-1})}}{2}.$$

However,

$$\left\|\mathbf{P}_{v_1^\perp}\widehat{x}\right\|_{\mathbf{B}} \leq \left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} + \left\|\mathbf{P}_{v_1^\perp}\xi\right\|_{\mathbf{B}} \leq \left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} + \|\xi\|_{\mathbf{B}}$$

and by Theorem 4 and the definition of G we have

$$\left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} = \left\|\mathbf{P}_{v_1}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} \cdot G(\mathbf{B}^{-1}x) \leq \left(\left|\langle x, v_1\rangle\right| \sqrt{\lambda_1(\mathbf{B}^{-1})}\right) \cdot \frac{G(x)}{100}.$$

Taking expectations, using that $|\langle x, v_1\rangle| \leq 1$, and combining these three inequalities yields

$$\mathbb{E}\left[G(\widehat{x})|\mathcal{F}\right] = \mathbb{E}\left[\frac{\left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}}}{\left\|\mathbf{P}_{v_1}\mathbf{B}^{-1}x\right\|_{\mathbf{B}}}\middle|\mathcal{F}\right] \leq \frac{G(x)}{50} + 2\frac{\mathbb{E}\left[\|\xi\|_{\mathbf{B}}|\mathcal{F}\right]}{\sqrt{\lambda_1(\mathbf{B}^{-1})}} \tag{2}$$

So, conditioning on making an update and changing x (i.e. $\mathcal{F}$ occurring), we see that our potential function changes exactly as in the exact case (Theorem 4) with additional additive error due to our inexact linear system solve.

Next we upper bound $\mathbb{P}\left[\mathcal{F}\right]$ and use it to compute $\mathbb{E}\left[\|\xi\|_{\mathbf{B}}|\mathcal{F}\right]$. We will show that

$$\mathcal{G} \stackrel{\text{def}}{=} \left\{\|\xi\|_{\mathbf{B}} \leq \frac{1}{100} \cdot \sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)}\right\} \subseteq \mathcal{F}$$

which then implies by Markov inequality that

$$\mathbb{P}\left[\mathcal{F}\right] \geq \mathbb{P}\left[\|\xi\|_{\mathbf{B}} \leq \frac{1}{100} \cdot \sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)}\right] \geq 1 - \frac{\mathbb{E}\left[\|\xi\|_{\mathbf{B}}\right]}{\frac{1}{100} \cdot \sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)}} \geq \frac{9}{10}, \tag{3}$$

where we used the fact that $\mathbb{E}[\|\xi\|_{\mathbf{B}}] \leq \frac{c_1}{1000}\sqrt{\lambda_1(\mathbf{B}^{-1})}$ for some $c_1 < 1$.

Let us now show that $\mathcal{G} \subseteq \mathcal{F}$. Suppose $\mathcal{G}$ is occurs. We can bound $\|\widehat{x}\|_2$ as follows:

$$\begin{aligned}\|\widehat{x}\|_2 &\geq \left\|\mathbf{B}^{-1}x\right\|_2 - \|\xi\|_2 \geq \left\|\mathbf{B}^{-1}x\right\| - \sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)}\|\xi\|_{\mathbf{B}} \\ &\geq |\alpha_1|\,\lambda_1\left(\mathbf{B}^{-1}\right) - \frac{1}{100}\cdot\lambda_1\left(\mathbf{B}^{-1}\right) \\ &= \frac{1}{\lambda-\lambda_1}\left(|\alpha_1| - \frac{1}{100}\right) \geq \frac{3}{4}\frac{1}{\lambda-\lambda_1}, \end{aligned} \tag{4}$$

where we use Lemmas 2 and 3 to conclude that $|\alpha_1| \geq \sqrt{1-\frac{1}{10}}$. We now turn to showing the Rayleigh quotient condition required by $\mathcal{F}$. In order to do this, we first bound $\widehat{x}^\top\mathbf{B}\widehat{x} - \left(v_1^\top\mathbf{B}\widehat{x}\right)\left(v_1^\top\widehat{x}\right)$ and then use Lemma 2. We have:

$$\begin{aligned}\sqrt{\widehat{x}^\top\mathbf{B}\widehat{x} - \left(v_1^\top\mathbf{B}\widehat{x}\right)\left(v_1^\top\widehat{x}\right)} &= \left\|\mathbf{P}_{v_1^\perp}\widehat{x}\right\|_{\mathbf{B}} \leq \left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} + \left\|\mathbf{P}_{v_1^\perp}\xi\right\|_{\mathbf{B}} \\ &\leq \sqrt{\sum_{i\geq 2}\alpha_i^2\lambda_i\left(\mathbf{B}^{-1}\right)} + \frac{1}{100}\cdot\sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)} \\ &\leq \sqrt{\lambda_2\left(\mathbf{B}^{-1}\right)} + \frac{1}{100}\cdot\sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)} \leq \frac{1}{9}\sqrt{\lambda-\lambda_1},\end{aligned}$$

where we used the fact that $\lambda_2\left(\mathbf{B}^{-1}\right) \leq \frac{1}{100}\lambda_1\left(\mathbf{B}^{-1}\right)$ since $\lambda \leq \lambda_1 + \frac{\text{gap}}{100}$ in the last step. Now, using Lemma 2 and the bound on $\|\widehat{x}\|_2$, we conclude that

$$\begin{aligned}\widehat{\lambda}_1 - \widehat{\text{quot}}\left(\widehat{x}\right) &\leq \lambda_1 - \text{quot}\left(\widehat{x}\right) + \left|\text{quot}\left(\widehat{x}\right) - \widehat{\text{quot}}\left(\widehat{x}\right)\right| + \widehat{\lambda}_1 - \lambda_1 \\ &\leq \frac{\widehat{x}^\top\mathbf{B}\widehat{x} - \left(v_1^\top\mathbf{B}\widehat{x}\right)\left(v_1^\top\widehat{x}\right)}{\|\widehat{x}\|_2^2} + \frac{\lambda-\lambda_1}{30} + \frac{\lambda-\lambda_1}{11} \\ &\leq \frac{1}{81\left(\lambda-\lambda_1\right)}\cdot\frac{16}{9}\left(\lambda-\lambda_1\right)^2 + \frac{\lambda-\lambda_1}{8} \\ &\leq \left(\lambda-\lambda_1\right)/6 \leq \left(\lambda-\widehat{\lambda}_1\right)/4. \end{aligned} \tag{5}$$

Combining (4) and (5) shows that $\mathcal{G} \subseteq \mathcal{F}$ there by proving (3).

Using this and the fact that $\|\cdot\|_{\mathbf{B}} \geq 0$ we can upper bound $\mathbb{E}\left[\|\xi\|_{\mathbf{B}}|\mathcal{F}\right]$ as follows:

$$\mathbb{E}\left[\|\xi\|_{\mathbf{B}}|\mathcal{F}\right] \leq \frac{1}{\mathbb{P}\left[\mathcal{F}\right]}\cdot\mathbb{E}\left[\|\xi\|_{\mathbf{B}}\right] \leq \frac{c_1}{900}\cdot\sqrt{\lambda_1(\mathbf{B}^{-1})}$$

Plugging this into (2), we obtain:

$$\mathbb{E}\left[G(\widehat{x})|\mathcal{F}\right] \leq \frac{1}{50}G(x) + \frac{2\mathbb{E}\left[\|\xi\|_{\mathbf{B}}|\mathcal{F}\right]}{\sqrt{\lambda_1(\mathbf{B}^{-1})}} \leq \frac{1}{50}\cdot G(x) + \frac{2c_1}{900}.$$

We can now finally bound $\mathbb{E}\left[G(\widetilde{x})\right]$ as follows:

$$\begin{aligned}\mathbb{E}\left[G(\widetilde{x})\right] &= \mathbb{P}\left[\mathcal{F}\right]\cdot\mathbb{E}\left[G(\widehat{x})|\mathcal{F}\right] + \left(1-\mathbb{P}\left[\mathcal{F}\right]\right)G(x) \\ &\leq \frac{9}{10}\left(\frac{1}{50}\cdot G(x) + \frac{2c_1}{900}\right) + \frac{1}{10}G(x) = \frac{3}{25}G(x) + \frac{2c_1}{1000}.\end{aligned}$$

This proves the theorem. □

Corollary 6 (Relative Error Linear System Solvers)**.** *For any unit vector x, we have:*

$$\left\|\frac{1}{x^\top \mathbf{B} x}x - \mathbf{B}^{-1}x\right\|_{\mathbf{B}} \leq \alpha_1 \sqrt{\lambda_1(\mathbf{B}^{-1})} \cdot G(x) = \lambda_1\left(\mathbf{B}^{-1}\right)\sqrt{\sum_{i\geq 2}\frac{\alpha_i^2}{\lambda_i\left(\mathbf{B}^{-1}\right)}}, \tag{6}$$

where $x = \sum_i \alpha_i v_i$ is the decomposition of x along v_i. Therefore, instantiating Theorem 5 with $c_1 = \alpha_1 G(x)$ gives $\mathbb{E}[G(\widetilde{x})] \leq \frac{4}{25}G(x)$ as long as:

$$\mathbb{E}\left[\left\|\text{solve}\,(x) - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}\right] \leq \frac{1}{1000}\left\|\frac{1}{\lambda - x^\top \mathbf{\Sigma} x}x - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}.$$

Proof. Since $\mathbf{B}$ is PSD we see that if we let $f(w) = \frac{1}{2}w^\top \mathbf{B} w - w^\top x$, then the minimizer is $\mathbf{B}^{-1}x$. Furthermore note that $\frac{1}{x^\top \mathbf{B} x} = \arg\min_\beta f(\beta x)$ and therefore

$$\begin{aligned}
\left\|\frac{1}{x^\top \mathbf{B} x}x - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}^2 &= x^\top \mathbf{B}^{-1}x - \frac{1}{x^\top \mathbf{B} x} = 2\left[f\left(\frac{x}{x^\top \mathbf{B} x}\right) - f(\mathbf{B}^{-1}x)\right] \\
&= 2\left[\min_\beta f(\beta x) - f(\mathbf{B}^{-1}x)\right] \leq 2\left[f(\lambda_1\left(\mathbf{B}^{-1}\right)x) - f(\mathbf{B}^{-1}x)\right] \\
&= \lambda_1\left(\mathbf{B}^{-1}\right)^2 x^\top \mathbf{B} x - 2\lambda_1\left(\mathbf{B}^{-1}\right)x^\top x + x^\top \mathbf{B}^{-1}x \\
&= \sum_{i=1}^{d}\left|v_i^\top \mathbf{B}^{\frac{1}{2}}x\right|^2 (\lambda_1\left(\mathbf{B}^{-1}\right) - \lambda_i\left(\mathbf{B}^{-1}\right))^2 \leq \lambda_1\left(\mathbf{B}^{-1}\right)^2 \sum_{i\geq 2}\left|v_i^\top \mathbf{B}^{\frac{1}{2}}x\right|^2 \\
&= \lambda_1\left(\mathbf{B}^{-1}\right)^2 \sum_{i\geq 2}\frac{\alpha_i^2}{\lambda_i\left(\mathbf{B}^{-1}\right)},
\end{aligned}$$

which proves (6). Consequently

$$\frac{c_1}{1000}\sqrt{\lambda_1(\mathbf{B}^{-1})} = \frac{1}{1000}\alpha_1 G(x)\sqrt{\lambda_1(\mathbf{B}^{-1})} \geq \frac{1}{1000}\left\|\frac{1}{x^\top \mathbf{B} x}x - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}$$

which with Theorem 5 then completes the proof. □

3.5 Initialization

Theorem 5 and Corollary 6 show that, given a good enough approximation to v_1, we can rapidly refine this approximation by applying the shifted-and-inverted power method. In this section, we cover initialization. That is, how to obtain a good enough approximation to apply these results.

We first give a simple bound on the quality of a randomly chosen start vector x_0.

Lemma 7 (Random Initialization Quality)**.** *Suppose $x \sim \mathcal{N}(0, \mathbf{I})$, and we initialize x_0 as $\frac{x}{\|x\|_2}$, then with probability greater than $1 - O\left(\frac{1}{d^{10}}\right)$, we have:*

$$G(x_0) \leq \sqrt{\kappa(\mathbf{B}^{-1})}d^{10.5} \leq 15\frac{1}{\sqrt{\text{gap}}} \cdot d^{10.5}$$

where $\kappa(\mathbf{B}^{-1}) = \lambda_1(\mathbf{B}^{-1})/\lambda_d(\mathbf{B}^{-1})$.

Proof.

$$G(x_0) = G(x) = \frac{\left\|\mathbf{P}_{v_1^\perp} x\right\|_{\mathbf{B}}}{\left\|\mathbf{P}_{v_1} x\right\|_{\mathbf{B}}} = \frac{\sqrt{\|x\|_{\mathbf{B}}^2 - \left(v_1^\top \mathbf{B}^{1/2} x\right)^2}}{\left|v_1^\top \mathbf{B}^{1/2} x\right|} = \frac{\sqrt{\sum_{i\geq 2} \frac{(v_i^\top x)^2}{\lambda_i(\mathbf{B}^{-1})}}}{\sqrt{\frac{(v_1^\top x)^2}{\lambda_1(\mathbf{B}^{-1})}}},$$

$$\leq \sqrt{\kappa(\mathbf{B}^{-1})} \cdot \frac{\sqrt{\sum_{i\geq 2}(v_i^\top x)^2}}{\left|v_1^\top x\right|}$$

Since $\{v_i^\top x\}_i$ are independent standard normal Gaussian variables. By standard concentration arguments, with probability greater than $1 - e^{-\Omega(d)}$, we have $\sqrt{\sum_{i\geq 2}(v_i^\top x)^2} = O(\sqrt{d})$. Meanwhile, $v_1^\top x$ is just a one-dimensional standard Gaussian. It is easy to show $\mathbb{P}\left(\left|v_1^\top x\right| \leq \frac{1}{d^{10}}\right) = O\left(\frac{1}{d^{10}}\right)$, which finishes the proof. □

We now show that we can rapidly decrease our initial error to obtain the required $G(x) \leq \frac{1}{\sqrt{10}}$ bound for Theorem 5.

Theorem 8 (Approximate Shifted-and-Inverted Power Method – Burn-In)**.** *Suppose we initialize x_0 as in Lemma 7 and suppose we have access to a subroutine* $\mathrm{solve}(\cdot)$ *such that*

$$\mathbb{E}\left[\left\|\mathrm{solve}(x) - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}\right] \leq \frac{1}{3000\kappa(\mathbf{B}^{-1})d^{21}} \cdot \left\|\frac{1}{\lambda - x^\top \mathbf{\Sigma} x} x - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}$$

where $\kappa(\mathbf{B}^{-1}) = \lambda_1(\mathbf{B}^{-1})/\lambda_d(\mathbf{B}^{-1})$. Then the following procedure,

$$x_t = \mathrm{solve}(x_{t-1}) \,/\, \|\mathrm{solve}(x_{t-1})\|_2$$

after $T = O\left(\log d + \log \kappa(\mathbf{B}^{-1}))\right)$ iterations satisfies:

$$G(x_T) \leq \frac{1}{\sqrt{10}},$$

with probability greater than $1 - O(\frac{1}{d^{10}})$.

Proof. As before, we first bound the numerator and denominator of $G(\widehat{x})$ more carefully as follows:

$$\begin{aligned}
\textbf{Numerator:} \quad \left\|\mathbf{P}_{v_1^\perp}\widehat{x}\right\|_{\mathbf{B}} &\leq \left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} + \left\|\mathbf{P}_{v_1^\perp}\xi\right\|_{\mathbf{B}} \leq \left\|\mathbf{P}_{v_1^\perp}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} + \|\xi\|_{\mathbf{B}} \\
&= \sqrt{\textstyle\sum_{i\geq 2}\left(v_i^T B^{-1/2}x\right)^2} + \|\xi\|_{\mathbf{B}} = \sqrt{\textstyle\sum_{i\geq 2}\alpha_i^2\lambda_i\left(\mathbf{B}^{-1}\right)} + \|\xi\|_{\mathbf{B}}, \\
\textbf{Denominator:} \quad \|\mathbf{P}_{v_1}\widehat{x}\|_{\mathbf{B}} &\geq \left\|\mathbf{P}_{v_1}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} - \|\mathbf{P}_{v_1}\xi\|_{\mathbf{B}} \geq \left\|\mathbf{P}_{v_1}\mathbf{B}^{-1}x\right\|_{\mathbf{B}} - \|\xi\|_{\mathbf{B}} \\
&= \left|v_i^T\mathbf{B}^{-1/2}x\right| - \|\xi\|_{\mathbf{B}} = \alpha_1\sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)} - \|\xi\|_{\mathbf{B}}
\end{aligned}$$

We now use the above estimates to bound $G(\widehat{x})$.

$$G(\widehat{x}) \leq \frac{\sqrt{\sum_{i\geq 2}\alpha_i^2\lambda_i\left(\mathbf{B}^{-1}\right)} + \|\xi\|_{\mathbf{B}}}{\alpha_1\sqrt{\lambda_1\left(\mathbf{B}^{-1}\right)} - \|\xi\|_{\mathbf{B}}} \leq \frac{\lambda_2\left(\mathbf{B}^{-1}\right)\sqrt{\sum_{i\geq 2}\frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}} + \|\xi\|_{\mathbf{B}}}{\lambda_1\left(\mathbf{B}^{-1}\right)\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}} - \|\xi\|_{\mathbf{B}}}$$

$$= G(x)\frac{\lambda_2\left(\mathbf{B}^{-1}\right) + \|\xi\|_{\mathbf{B}}\,/\sqrt{\sum_{i\geq 2}\frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}}}{\lambda_1\left(\mathbf{B}^{-1}\right) - \|\xi\|_{\mathbf{B}}\,/\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}}}$$

By Lemma 7, we know with at least probability $1 - O(\frac{1}{d^{10}})$, we have $G(x_0) \leq \sqrt{\kappa(\mathbf{B}^{-1})}d^{10.5}$.

Conditioned on high probability result of $G(x_0)$, we now use induction to prove $G(x_t) \leq G(x_0)$. It trivially holds for $t = 0$. Suppose we now have $G(x) \leq G(x_0)$, then by the condition in Theorem 8 and Markov inequality, we know with probability greater than $1 - \frac{1}{100\sqrt{\kappa(\mathbf{B}^{-1})}d^{10.5}}$ we have:

$$\begin{aligned}
\|\xi\|_{\mathbf{B}} \leq & \frac{1}{30\sqrt{\kappa(\mathbf{B}^{-1})}d^{10.5}} \cdot \left\| \frac{1}{\lambda - x^\top \mathbf{\Sigma} x} x - \mathbf{B}^{-1}x \right\|_{\mathbf{B}} \\
\leq & \frac{1}{30} \cdot \left\| \frac{1}{\lambda - x^\top \mathbf{\Sigma} x} x - \mathbf{B}^{-1}x \right\|_{\mathbf{B}} \min\left\{1, \frac{1}{G(x_0)}\right\} \\
\leq & \frac{1}{30} \cdot \left\| \frac{1}{\lambda - x^\top \mathbf{\Sigma} x} x - \mathbf{B}^{-1}x \right\|_{\mathbf{B}} \min\left\{1, \frac{1}{G(x)}\right\} \\
\leq & \frac{\lambda_1\left(\mathbf{B}^{-1}\right) - \lambda_2\left(\mathbf{B}^{-1}\right)}{4} \min\left\{ \sqrt{\sum_{i\geq 2} \frac{\alpha_i^2}{\lambda_i\left(\mathbf{B}^{-1}\right)}}, \sqrt{\frac{\alpha_1^2}{\lambda_1\left(\mathbf{B}^{-1}\right)}} \right\}
\end{aligned}$$

The last inequality uses Corollary 6 with the fact that $\lambda_2\left(\mathbf{B}^{-1}\right) \leq \frac{1}{100}\lambda_1\left(\mathbf{B}^{-1}\right)$. Therefore, we have: We will have:

$$G(\widehat{x}) \leq \frac{\lambda_1\left(\mathbf{B}^{-1}\right) + 3\lambda_2\left(\mathbf{B}^{-1}\right)}{3\lambda_1\left(\mathbf{B}^{-1}\right) + \lambda_2\left(\mathbf{B}^{-1}\right)} \times G(x) \leq \frac{1}{2}G(x)$$

This finishes the proof of induction.

Finally, by union bound, we know with probability greater than $1 - O(\frac{1}{d^{10}})$ in $T = O(\log d + \log \kappa(\mathbf{B}^{-1}))$ steps, we have:

$$G(x_T) \leq \frac{1}{2^T}G(x_0) \leq \frac{1}{\sqrt{10}}$$

□

4 Offline Eigenvector Computation

In this section we show how to instantiate the framework of Section 3 in order to compute an approximate top eigenvector in the offline setting. As discussed, in the offline setting we can trivially compute the Rayleigh quotient of a vector in $\text{nnz}(\mathbf{A})$ time as we have explicit access to $\mathbf{A}^\top\mathbf{A}$. Consequently the bulk of our work in this section is to show how we can solve linear systems in $\mathbf{B}$ efficiently in expectation, allowing us to apply Corollary 6 of Theorem 5.

In Section 4.1 we first show how Stochastic Variance Reduced Gradient (SVRG) [JZ13] can be adapted to solve linear systems of the form $\mathbf{B}x = b$. If we wanted, for example, to solve a linear system in a positive definite matrix like $\mathbf{A}^\top\mathbf{A}$, we would optimize the objective function $f(x) = \frac{1}{2}x^\top\mathbf{A}^\top\mathbf{A}x - b^\top x$. This function can be written as the sum of n *convex components*, $\psi_i(x) = \frac{1}{2}x^\top\left(a_i a_i^\top\right)x - \frac{1}{n}b^\top x$. In each iteration of traditional gradient descent, one computes the full gradient of $f(x_i)$ and takes a step in that direction. In stochastic gradient methods, at each iteration, a single component is sampled, and the step direction is based only on the gradient of the sampled component. Hence, we avoid a full gradient computation at each iteration, leading to runtime gains.

Unfortunately, while we have access to the rows of $\mathbf{A}$ and so can solve systems in $\mathbf{A}^\top\mathbf{A}$, it is less clear how to solve systems in $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$. To do this, we will split our function into components of the form $\psi_i(x) = \frac{1}{2}x^\top\left(w_i\mathbf{I} - a_i a_i^\top\right)x - \frac{1}{n}b^\top x$ for some set of weights w_i with $\sum_{i\in[n]} w_i = \lambda$.

Importantly, $(w_i \mathbf{I} - a_i a_i^\top)$ may not be positive semidefinite. That is, we are minimizing a sum of functions which is convex, but consists of non-convex components. While recent results for minimizing such functions could be applied directly [SS15, CR15] here we show how to obtain stronger results by using a more general form of SVRG and analyzing the specific properties of our function (i.e. the variance).

Our analysis shows that we can make constant factor progress in solving linear systems in $\mathbf{B}$ in time $O\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right)$. If $\frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2} \leq \text{nnz}(\mathbf{A})$ this gives a runtime proportional to the input size – the best we could hope for. If not, we show in Section 4.2 that it is possible to *accelerate* our system solver, achieving runtime $\tilde{O}\left(\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}}\right)$. This result uses the work of [FGKS15b, LMH15] on accelerated approximate proximal point algorithms.

With our solvers in place, in Section 4.3 we pull our results together, showing how to use these solvers in the framework of Section 3 to give faster running times for offline eigenvector computation.

4.1 SVRG Based Solver

Here we provide a sampling based algorithm for solving linear systems in $\mathbf{B}$. In particular we provide an algorithm for solving the more general problem where we are given a strongly convex function that is a sum of possibly non-convex functions that obey smoothness properties. We provide a general result on bounding the progress of an algorithm that solves such a problem by non-uniform sampling in Theorem 9 and then in the remainder of this section we show how to bound the requisite quantities for solving linear systems in $\mathbf{B}$.

Theorem 9 (SVRG for Sums of Non-Convex Functions)**.** *Consider a set of functions, $\{\psi_1, \psi_2, ...\psi_n\}$, each mapping $\mathbb{R}^d \to \mathbb{R}$. Let $f(x) = \sum_i \psi_i(x)$ and let $x^{\text{opt}} \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^d} f(x)$. Suppose we have a probability distribution p on $[n]$, and that starting from some initial point $x_0 \in \mathbb{R}^d$ in each iteration k we pick $i_k \in [n]$ independently with probability p_{i_k} and let*

$$x_{k+1} := x_k - \frac{\eta}{p_i}\left(\bigtriangledown \psi_i(x_k) - \bigtriangledown \psi_i(x_0)\right) + \eta \bigtriangledown f(x_0)$$

for some η. If f is μ-strongly convex and if for all $x \in \mathbb{R}^d$ we have

$$\sum_{i \in [n]} \frac{1}{p_i} \left\| \bigtriangledown \psi_i(x) - \bigtriangledown \psi_i(x^{\text{opt}}) \right\|_2^2 \leq 2\overline{S}\left[f(x) - f(x^{\text{opt}})\right], \tag{7}$$

where $\overline{S}$ is a variance parameter, then for all $m \geq 1$ we have

$$\mathbb{E}\left[\frac{1}{m}\sum_{k \in [m]} f(x_k) - f(x^{\text{opt}})\right] \leq \frac{1}{1 - 2\eta\bar{S}}\left[\frac{1}{\mu\eta m} + 2\eta\overline{S}\right] \cdot \left[f(x_0) - f(x^{\text{opt}})\right]$$

Consequently, if we pick η to be a sufficiently small multiple of $1/\bar{S}$ then when $m = O(\overline{S}/\mu)$ we can decrease the error by a constant multiplicative factor in expectation.

Proof. We first note that $\mathbb{E}_{i_k}[x_{k+1} - x_k] = \eta \bigtriangledown f(x_k)$. This is, in each iteration, in expectation, we

make a step in the direction of the gradient. Using this fact we have:

$$\begin{aligned}\mathbb{E}_{i_k}\left\|x_{k+1}-x^{\text{opt}}\right\|_2^2 &= \mathbb{E}_{i_k}\left\|(x_{k+1}-x_k)+(x_k-x^{\text{opt}})\right\|_2^2 \\ &= \left\|x_k-x^{\text{opt}}\right\|_2^2 - 2\mathbb{E}_{i_k}(x_{k+1}-x_k)^\top(x_k-x^{\text{opt}}) + \mathbb{E}_{i_k}\left\|x_{k+1}-x_k\right\|_2^2 \\ &= \left\|x_k-x^{\text{opt}}\right\|_2^2 - 2\eta\bigtriangledown f(x_k)^\top\left(x_k-x^{\text{opt}}\right) \\ &+ \sum_{i\in[n]}\eta^2 p_i\left\|\frac{1}{p_i}\left(\bigtriangledown\psi_i(x_k)-\bigtriangledown\psi_i(x_0)\right)+\bigtriangledown f(x_0)\right\|_2^2\end{aligned}$$

We now apply the fact that $\|x+y\|_2^2 \leq 2\|x\|_2^2 + 2\|y\|_2^2$ to give:

$$\begin{aligned}&\sum_{i\in[n]} p_i\left\|\frac{1}{p_i}\left(\bigtriangledown\psi_i(x_k)-\bigtriangledown\psi_i(x_0)\right)+\bigtriangledown f(x_0)\right\|_2^2 \\ &\quad\leq \sum_{i\in[n]} 2p_i\left\|\frac{1}{p_i}\left(\bigtriangledown\psi_i(x_k)-\bigtriangledown\psi_i(x^{\text{opt}})\right)\right\|_2^2 + \sum_{i\in[n]} 2p_i\left\|\frac{1}{p_i}\left(\bigtriangledown\psi_i(x_0)-\bigtriangledown\psi_i(x^{\text{opt}})\right)-\bigtriangledown f(x_0)\right\|_2^2.\end{aligned}$$

Then, using that $\bigtriangledown f(x^{\text{opt}})=0$ by optimality, that $\mathbb{E}\|x-\mathbb{E}x\|_2^2 \leq \mathbb{E}\|x\|_2^2$, and (7) we have:

$$\begin{aligned}&\sum_{i\in[n]} p_i\left\|\frac{1}{p_i}\left(\bigtriangledown\psi_i(x_k)-\bigtriangledown\psi_i(x_0)\right)+\bigtriangledown f(x_0)\right\|_2^2 \\ &\quad\leq \sum_{i\in[n]}\frac{2}{p_i}\left\|\bigtriangledown\psi_i(x_k)-\bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 + \sum_{i\in[n]} 2p_i\left\|\frac{1}{p_i}\left(\bigtriangledown\psi_i(x_0)-\bigtriangledown\psi_i(x^{\text{opt}})\right)-\left(\bigtriangledown f(x_0)-\bigtriangledown f(x^{\text{opt}})\right)\right\|_2^2 \\ &\quad\leq \sum_{i\in[n]}\frac{2}{p_i}\left\|\bigtriangledown\psi_i(x_k)-\bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 + \sum_{i\in[n]} 2p_i\left\|\frac{1}{p_i}\bigtriangledown\psi_i(x_0)-\bigtriangledown\psi_i(x^{\text{opt}}))\right\|_2^2 \\ &\quad\leq 4\overline{S}\left[f(x_k)-f(x^{\text{opt}})+f(x_0)-f(x^{\text{opt}})\right]\end{aligned}$$

Since $f(x^{\text{opt}})-f(x_k)\geq\bigtriangledown f(x_k)^\top(x^{\text{opt}}-x_k)$ by the convexity of f, these inequalities imply

$$\begin{aligned}\mathbb{E}_{i_k}\left\|x_{k+1}-x^{\text{opt}}\right\|_2^2 &\leq \left\|x_k-x^{\text{opt}}\right\|_2^2 - 2\eta\left[f(x_k)-f(x^{\text{opt}})\right] + 4\eta^2\overline{S}\left[f(x_k)-f(x^{\text{opt}})+f(x_0)-f(x^{\text{opt}})\right] \\ &= \left\|x_k-x^{\text{opt}}\right\|_2^2 - 2\eta(1-2\eta S)\left(f(x_k)-f(x^{\text{opt}})\right) + 4\eta^2\bar{S}\left(f(x_0)-f(x^{\text{opt}})\right)\end{aligned}$$

Rearranging, we have:

$$2\eta(1-2\eta S)\left(f(x_k)-f(x^{\text{opt}})\right) \leq \left\|x_k-x^{\text{opt}}\right\|_2^2 - \mathbb{E}_{i_k}\left\|x_{k+1}-x^{\text{opt}}\right\|_2^2 + 4\eta^2\bar{S}\left(f(x_0)-f(x^{\text{opt}})\right).$$

And summing over all iterations and taking expectations we have:

$$\mathbb{E}\left[2\eta(1-2\eta\bar{S})\sum_{k\in[m]} f(x_k)-f(x^{\text{opt}})\right] \leq \left\|x_0-x^{\text{opt}}\right\|_2^2 + 4m\eta^2\bar{S}\left[f(x_0)-f(x^{\text{opt}})\right].$$

Finally, we use that by strong convexity, $\left\|x_0-x^{\text{opt}}\right\|_2^2 \leq \frac{2}{\mu}\left(f(x_0)-f(x^{\text{opt}})\right)$ to obtain:

$$\mathbb{E}\left[2\eta(1-2\eta\bar{S})\sum_{k\in[m]} f(x_k)-f(x^{\text{opt}})\right] \leq \frac{2}{\mu}\left[f(x_0)-f(x^{\text{opt}})\right] + 4m\eta^2\bar{S}\left[f(x_0)-f(x^{\text{opt}})\right]$$

and thus

$$\mathbb{E}\left[\frac{1}{m}\sum_{k\in[m]} f(x_k) - f(x^{\text{opt}})\right] \leq \frac{1}{1-2\eta\bar{S}}\left[\frac{1}{\mu\eta m} + 2\eta\bar{S}\right]\cdot\left[f(x_0) - f(x^{\text{opt}})\right]$$

□

Theorem 9 immediately yields a solver for $\mathbf{B}x = b$. Finding the minimum norm solution to this system is equivalent to minimizing $f(x) = \frac{1}{2}x^\top\mathbf{B}x - b^\top x$. If we take the common approach of applying a smoothness bound for each ψ_i along with a strong convexity bound on $f(x)$ we obtain:

Lemma 10 (Simple Variance Bound for SVRG)**.** *Let*

$$\psi_i(x) \stackrel{\text{def}}{=} \frac{1}{2}x^\top\left(\frac{\lambda\left\|a_i\right\|_2^2}{\left\|\mathbf{A}\right\|_F^2}\mathbf{I} - a_i a_i^\top\right)x - \frac{1}{n}b^\top x$$

so we have $\sum_{i\in[n]}\psi_i(x) = f(x) = \frac{1}{2}x^\top\mathbf{B}x - b^\top x$*. Setting* $p_i = \frac{\|a_i\|_2^2}{\|\mathbf{A}\|_F^2}$ *for all* i*, we have*

$$\sum_{i\in[n]}\frac{1}{p_i}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 = O\left(\frac{\left\|\mathbf{A}\right\|_F^4}{\lambda-\lambda_1}\left[f(x) - f(x^{\text{opt}})\right]\right)$$

Proof. We first compute, for all $i \in [n]$

$$\bigtriangledown\psi_i(x) = \left(\frac{\lambda\left\|a_i\right\|_2^2}{\left\|\mathbf{A}\right\|_{\text{F}}^2}\mathbf{I} - a_i a_i^\top\right)x - \frac{1}{n}b. \tag{8}$$

We have that each ψ_i is $\frac{\lambda\|a_i\|_2^2}{\|\mathbf{A}\|_{\text{F}}^2} + \|a_i\|^2$ smooth with respect to $\|\cdot\|_2$. Specifically,

$$\begin{aligned}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2 &= \left\|\left(\frac{\lambda\left\|a_i\right\|_2^2}{\left\|\mathbf{A}\right\|_{\text{F}}^2}\mathbf{I} - a_i a_i^\top\right)(x - x^{\text{opt}})\right\|_2 \\ &\leq \left(\frac{\lambda\left\|a_i\right\|_2^2}{\left\|\mathbf{A}\right\|_{\text{F}}^2} + \left\|a_i\right\|^2\right)\left\|x - x^{\text{opt}}\right\|_2.\end{aligned}$$

Additionally, $f(x)$ is $\lambda_d(\mathbf{B}) = \lambda - \lambda_1$ strongly convex so we have $\left\|x - x^{\text{opt}}\right\|_2^2 \leq \frac{2}{\lambda-\lambda_1}\left[f(x) - f(x^{\text{opt}})\right]$ and putting all this together we have

$$\begin{aligned}\sum_{i\in[n]}\frac{1}{p_i}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 &\leq \sum_{i\in[n]}\frac{\left\|\mathbf{A}\right\|_F^2}{\left\|a_i\right\|_2^2}\cdot\left\|a_i\right\|_2^4\left(\frac{\lambda}{\left\|\mathbf{A}\right\|_F^2} + 1\right)^2\cdot\frac{2}{\lambda-\lambda_1}\left[f(x) - f(x^{\text{opt}})\right] \\ &= O\left(\frac{\left\|\mathbf{A}\right\|_F^4}{\lambda-\lambda_1}\left[f(x) - f(x^{\text{opt}})\right]\right)\end{aligned}$$

where the last step uses that $\lambda \leq 2\lambda_1 \leq 2\left\|\mathbf{A}\right\|_F^2$ so $\frac{\lambda}{\|\mathbf{A}\|_F^2} \leq 2$. □

Assuming that $\lambda = (1 + c \cdot \text{gap})\lambda_1$ for some constant c, the above bound means that we can make constant progress on our linear system by setting $m = O(\overline{S}/\mu) = O\left(\frac{\|\mathbf{A}\|_F^4}{(\lambda-\lambda_1)^2}\right) = O\left(\frac{\text{sr}(\mathbf{A})^2}{\text{gap}^2}\right)$. This dependence on stable rank matches the dependence given in [Sha15c] (see discussion in Section 1.3), however we can show that it is suboptimal. We show to improve the bound to $O\left(\frac{\text{sr}(\mathbf{A})}{\text{gap}^2}\right)$ by using a better variance analysis. Instead of bounding each $\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2$ term using the smoothness of ψ_i, we more carefully bound the sum of these terms.

Lemma 11. *(Improved Variance Bound for SVRG) For $i \in [n]$ let*

$$\psi_i(x) \stackrel{\text{def}}{=} \frac{1}{2}x^\top\left(\frac{\lambda\left\|a_i\right\|_2^2}{\|\mathbf{A}\|_F^2}\mathbf{I} - a_i a_i^\top\right)x - \frac{1}{n}b^\top x$$

so we have $\sum_{i\in[n]}\psi_i(x) = f(x) = \frac{1}{2}x^\top\mathbf{B}x - b^\top x$. Setting $p_i = \frac{\|a_i\|_2^2}{\|\mathbf{A}\|_F^2}$ for all i, we have for all x

$$\sum_{i\in[n]}\frac{1}{p_i}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 \leq \frac{4\lambda_1\|\mathbf{A}\|_{\text{F}}^2}{\lambda-\lambda_1}\cdot\left[f(x) - f(x^{\text{opt}})\right].$$

Proof. Using the gradient computation in (8) we have

$$\begin{aligned}
\sum_{i\in[n]}\frac{1}{p_i}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 &= \sum_{i\in[n]}\frac{\|\mathbf{A}\|_{\text{F}}^2}{\|a_i\|_2^2}\left\|\left(\frac{\lambda\|a_i\|_2^2}{\|\mathbf{A}\|_{\text{F}}^2}\mathbf{I} - a_i a_i^\top\right)(x - x^{\text{opt}})\right\|_2^2 \\
&= \sum_{i\in[n]}\frac{\lambda^2\|a_i\|_2^2}{\|\mathbf{A}\|_{\text{F}}^2}\left\|x - x^{\text{opt}}\right\|_2^2 - 2\sum_{i\in[n]}\lambda\left\|x - x^{\text{opt}}\right\|_{a_i a_i^\top}^2 \\
&+ \sum_{i\in[n]}\frac{\|\mathbf{A}\|_{\text{F}}^2}{\|a_i\|^2}\left\|x - x^{\text{opt}}\right\|_{\|a_i\|_2^2 a_i a_i^\top}^2 \\
&= \lambda^2\left\|x - x^{\text{opt}}\right\|_2^2 - 2\lambda\left\|x - x^{\text{opt}}\right\|_{\mathbf{\Sigma}}^2 + \|\mathbf{A}\|_{\text{F}}^2\left\|x - x^{\text{opt}}\right\|_{\mathbf{\Sigma}}^2. \\
&\leq \lambda\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 + \|\mathbf{A}\|_{\text{F}}^2\left\|x - x^{\text{opt}}\right\|_{\mathbf{\Sigma}}^2. \qquad (9)
\end{aligned}$$

Now since

$$\mathbf{\Sigma} \preceq \lambda_1\mathbf{I} \preceq \frac{\lambda_1}{\lambda-\lambda_1}\mathbf{B}$$

we have

$$\begin{aligned}
\sum_{i\in[n]}\frac{1}{p_i}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\text{opt}})\right\|_2^2 &\leq \left(\frac{\lambda(\lambda-\lambda_1) + \|\mathbf{A}\|_{\text{F}}^2\cdot\lambda_1}{\lambda-\lambda_1}\right)\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 \\
&\leq \left(\frac{2\|\mathbf{A}\|_{\text{F}}^2\lambda_1}{\lambda-\lambda_1}\right)\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2
\end{aligned}$$

where in the last inequality we just coarsely bound $\lambda(\lambda-\lambda_1) \leq \lambda_1\|\mathbf{A}\|_F^2$. Now since $\mathbf{B}$ is full rank, $\mathbf{B}x^{\text{opt}} = b$, we can compute:

$$\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 = x^\top\mathbf{B}x - 2b^\top x + b^\top x^{\text{opt}} = 2[f(x) - f(x^{\text{opt}})]. \qquad (10)$$

The result follows. □

Plugging the bound in Lemma 11 into Theorem 9 we have:

Theorem 12. *(Offline SVRG-Based Solver) Let* $\overline{S} = \frac{2\lambda_1 \|\mathbf{\Sigma}\|_{\mathrm{F}}^2}{\lambda - \lambda_1}$, $\mu = \lambda - \lambda_1$. *The iterative procedure described in Theorem 9 with* $f(x) = \frac{1}{2}x^\top \mathbf{B}x - b^\top x$, $\psi_i(x) = \frac{1}{2}x^\top \left(\frac{\lambda \|a_i\|_2^2}{\|\mathbf{\Sigma}\|_F^2}\mathbf{I} - a_i a_i^\top\right)x - b^\top x$, $p_i = \frac{\|a_i\|_2^2}{\|\mathbf{\Sigma}\|_F^2}$, $\eta = 1/(8\overline{S})$ *and* m *chosen uniformly at random from* $[64\overline{S}/\mu]$ *returns a vector* x_m *such that*

$$\mathbb{E}\left\|x_m - x^{\mathrm{opt}}\right\|_{\mathbf{B}}^2 \leq \frac{1}{2}\left\|x_0 - x^{\mathrm{opt}}\right\|_{\mathbf{B}}^2.$$

Further, assuming $\left(1 + \frac{\mathrm{gap}}{150}\right)\lambda_1 < \lambda \leq \left(1 + \frac{\mathrm{gap}}{100}\right)\lambda_1$, *this procedure runs in time* $O\left(\mathrm{nnz}(\mathbf{A}) + \frac{d \cdot \mathrm{sr}(\mathbf{A})}{\mathrm{gap}^2}\right)$.

Proof. Lemma 11 tells us that

$$\sum_{i\in[n]} \frac{1}{p_i}\left\|\bigtriangledown\psi_i(x) - \bigtriangledown\psi_i(x^{\mathrm{opt}})\right\|_2^2 \leq 2\overline{S}\left[f(x) - f(x^{\mathrm{opt}})\right].$$

Further $f(x) = \frac{1}{2}x^\top \mathbf{B}x - b^\top x$ is $\lambda_d(\mathbf{B})$-strongly convex and $\lambda_d(\mathbf{B}) = \lambda - \lambda_1 = \mu$. Plugging this into Theorem 9 and using (10) which shows $\left\|x - x^{\mathrm{opt}}\right\|_{\mathbf{B}}^2 = 2[f(x) - f(x^{\mathrm{opt}})]$ we have, for m chosen uniformly from $[64\overline{S}/\mu]$:

$$\mathbb{E}\left[\frac{1}{64\overline{S}/\mu}\sum_{k\in[64\overline{S}/\mu]} f(x_k) - f(x^{\mathrm{opt}})\right] \leq 4/3 \cdot [1/8 + 1/8] \cdot \left[f(x_0) - f(x^{\mathrm{opt}})\right]$$

$$\mathbb{E}\left[f(x_m) - f(x^{\mathrm{opt}})\right] \leq \frac{1}{2}\left[f(x_0) - f(x^{\mathrm{opt}})\right]$$

$$\mathbb{E}\left\|x_m - x^{\mathrm{opt}}\right\|_{\mathbf{B}}^2 \leq \frac{1}{2}\left\|x_0 - x^{opt}\right\|_{\mathbf{B}}^2.$$

The procedure requires $O\left(\mathrm{nnz}(\mathbf{A})\right)$ time to initially compute $\bigtriangledown f(x_0)$, along with each p_i and the step size η which depend on $\|\mathbf{A}\|_F^2$ and the row norms of $\mathbf{A}$. Each iteration then just requires $O(d)$ time to compute $\bigtriangledown\psi_i(\cdot)$ and perform the necessary vector operations. Since there are at most $[64\overline{S}/\mu] = O\left(\frac{\lambda_1\|\mathbf{A}\|_F^2}{(\lambda-\lambda_1)^2}\right)$ iterations, our total runtime is

$$O\left(\mathrm{nnz}(\mathbf{A}) + d \cdot \frac{\lambda_1 \|\mathbf{A}\|_F^2}{(\lambda - \lambda_1)^2}\right) = O\left(\mathrm{nnz}(\mathbf{A}) + \frac{d \cdot \mathrm{sr}(\mathbf{A})}{\mathrm{gap}^2}\right).$$

Note that if our matrix is uniformly sparse - i.e. all rows have sparsity at most d_s, then the runtime is actually at most $O\left(\mathrm{nnz}(\mathbf{A}) + \frac{d_s \cdot \mathrm{sr}(\mathbf{A})}{\mathrm{gap}^2}\right)$. □

4.2 Accelerated Solver

Theorem 12 gives a linear solver for $\mathbf{B}$ that makes progress in expectation and which we can plug into Theorems 5 and 8. However, we first show that the runtime in Theorem 12 can be accelerated in some cases. We apply a result of [FGKS15b], which shows that, given a solver for a regularized version of a convex function $f(x)$, we can produce a fast solver for $f(x)$ itself. Specifically:

Lemma 13 (Theorem 1.1 of [FGKS15b]). *Let $f(x)$ be a μ-strongly convex function and let $x^{\text{opt}} \stackrel{\text{def}}{=} \arg\min_{x\in\mathbb{R}^d} f(x)$. For any $\gamma > 0$ and any $x_0 \in \mathbb{R}^d$, let $f_{\gamma,x_0}(x) \stackrel{\text{def}}{=} f(x) + \frac{\gamma}{2}\|x - x_0\|_2^2$. Let $x^{\text{opt}}_{\gamma,x_0} \stackrel{\text{def}}{=} \arg\min_{x\in\mathbb{R}^d} f_{\gamma,x_0}(x)$.*

Suppose that, for all $x_0 \in \mathbb{R}^d$, $c > 0$, $\gamma > 0$, we can compute a point x_c such that

$$\mathbb{E} f_{\gamma,x_0}(x_c) - f_{\gamma,x_0}(x^{\text{opt}}_{\gamma,x_0}) \leq \frac{1}{c}\left[f_{\gamma,x_0} - f_{\gamma,x_0}(x^{\text{opt}}_{\gamma,x_0})\right]$$

in time $\mathcal{T}_c$. Then given any x_0, $c > 0$, $\gamma > 2\mu$, we can compute x_1 such that

$$\mathbb{E} f(x_1) - f(x^{\text{opt}}) \leq \frac{1}{c}\left[f(x_0) - f(x^{\text{opt}})\right]$$

in time $O\left(\mathcal{T}_{4\left(\frac{2\gamma+\mu}{\mu}\right)^{3/2}\sqrt{\lceil\gamma/\mu\rceil}\log c}\right)$.

We first give a new variance bound on solving systems in $\mathbf{B}$ when a regularizer is used. The proof of this bound is very close to the proof given for the unregularized problem in Lemma 11.

Lemma 14. *For $i \in [n]$ let*

$$\psi_i(x) \stackrel{\text{def}}{=} \frac{1}{2}x^\top\left(\frac{\lambda\|a_i\|_2^2}{\|\mathbf{A}\|_F^2}\mathbf{I} - a_i a_i^\top\right)x - \frac{1}{n}b^\top x + \frac{\gamma\|a_i\|_2^2}{2\|\mathbf{A}\|_F^2}\|x - x_0\|_2^2$$

so we have $\sum_{i\in[n]}\psi_i(x) = f_{\gamma,x_0}(x) = \frac{1}{2}x^\top\mathbf{B}x - b^\top x + \frac{\gamma}{2}\|x - x_0\|_2^2$. Setting $p_i = \frac{\|a_i\|_2^2}{\|\mathbf{A}\|_F^2}$ for all i, we have for all x

$$\sum_{i\in[n]}\frac{1}{p_i}\left\|\nabla\psi_i(x) - \nabla\psi_i(x^{\text{opt}}_{\gamma,x_0})\right\|_2^2 \leq \left(\frac{\gamma^2 + 12\lambda_1\|\mathbf{A}\|_F^2}{\lambda - \lambda_1 + \gamma}\right)\left[f_{\gamma,x_0}(x) - f_{\gamma,x_0}(x^{\text{opt}}_{\gamma,x_0})\right]$$

Proof. We have for all $i \in [n]$

$$\nabla\psi_i(x) = \left(\frac{\lambda\|a_i\|_2^2}{\|\mathbf{A}\|_{\mathrm{F}}^2}\mathbf{I} - a_i a_i^\top\right)x - \frac{1}{n}b + \frac{\gamma\|a_i\|_2^2}{2\|\mathbf{A}\|_F^2}(x - 2x_0) \tag{11}$$

Plugging this in we have:

$$\sum_{i\in[n]}\frac{1}{p_i}\left\|\nabla\psi_i(x) - \nabla\psi_i(x^{\text{opt}}_{\gamma,x_0})\right\|_2^2 = \sum_{i\in[n]}\frac{\|\mathbf{A}\|_{\mathrm{F}}^2}{\|a_i\|_2^2}\left\|\left(\frac{\lambda\|a_i\|_2^2}{\|\mathbf{A}\|_{\mathrm{F}}^2}\mathbf{I} - a_i a_i^\top\right)(x - x^{\text{opt}}_{\gamma,x_0}) + \frac{\gamma\|a_i\|_2^2}{2\|\mathbf{A}\|_F^2}(x - x^{\text{opt}}_{\gamma,x_0})\right\|_2^2$$

For simplicity we now just use the fact that $\|x + y\|_2^2 \leq 2\|x\|_2^2 + 2\|y\|_2^2$ and apply our bound from equation (9) to obtain:

$$\begin{aligned}
\sum_{i\in[n]}\frac{1}{p_i}\left\|\nabla\psi_i(x) - \nabla\psi_i(x^{\text{opt}}_{\gamma,x_0})\right\|_2^2 &\leq 2\lambda^2\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_2^2 - 4\lambda\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_{\mathbf{\Sigma}}^2 + 2\|\mathbf{\Sigma}\|_{\mathrm{F}}^2\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_{\mathbf{\Sigma}}^2 \\
&\quad + 2\sum_{i\in[n]}\frac{\|a_i\|_2^2}{\|\mathbf{A}\|_{\mathrm{F}}^2}\frac{\gamma^2}{4}\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_2^2 \\
&\leq \left(2\lambda^2 + \gamma^2/2 + 2\lambda_1\|\mathbf{A}\|_F^2 - 4\lambda_1\lambda\right)\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_2^2 \\
&\leq \left(\gamma^2/2 + 6\lambda_1\|\mathbf{A}\|_F^2\right)\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_2^2
\end{aligned}$$

Now, $f_{\gamma,x_0}(\cdot)$ is $\lambda - \lambda_1 + \gamma$ strongly convex, so

$$\left\|x - x^{\text{opt}}_{\gamma,x_0}\right\|_2^2 \leq \frac{2}{\lambda - \lambda_1 + \gamma}\left[f_{\gamma,x_0}(x) - f_{\gamma,x_0}(x^{\text{opt}}_{\gamma,x_0})\right].$$

So overall we have:

$$\sum_{i\in[n]} \frac{1}{p_i}\left\|\nabla\psi_i(x) - \nabla\psi_i(x^{\text{opt}}_{\gamma,x_0})\right\|_2^2 \leq \left(\frac{\gamma^2 + 12\lambda_1\left\|\mathbf{A}\right\|_F^2}{\lambda - \lambda_1 + \gamma}\right)\left[f_{\gamma,x_0}(x) - f_{\gamma,x_0}(x^{\text{opt}}_{\gamma,x_0})\right]$$

□

We can now use this variance bound to obtain an accelerated solver for $\mathbf{B}$. We assume $\text{nnz}(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}$, as otherwise, the unaccelerated solver in Theorem 12 runs in $O(\text{nnz}(\mathbf{A}))$ time and cannot be accelerated further.

Theorem 15 (Accelerated SVRG-Based Solver)**.** *Assuming* $\left(1 + \frac{\text{gap}}{150}\right)\lambda_1 < \lambda \leq \left(1 + \frac{\text{gap}}{100}\right)\lambda_1$ *and* $nnz(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}$*, applying the iterative procedure described in Theorem 9 along with the acceleration given by Lemma 13 gives a solver that returns* x *with*

$$\mathbb{E}\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 \leq \frac{1}{2}\left\|x_0 - x^{\text{opt}}\right\|_{\mathbf{B}}^2.$$

in time $O\left(\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}} \cdot \log\left(\frac{d}{\text{gap}}\right)\right)$.

Proof. Following Theorem 12, the variance bound of Lemma 14 means that we can make constant progress in minimizing $f_{\gamma,x_0}(x)$ in $O\left(\text{nnz}(\mathbf{A}) + dm\right)$ time where $m = O\left(\frac{\gamma^2 + 12\lambda_1\|\mathbf{\Sigma}\|_F^2}{(\lambda - \lambda_1 + \gamma)^2}\right)$. So, for $\gamma \geq 2(\lambda - \lambda_1)$ we can make $4\left(\frac{2\gamma + (\lambda - \lambda_1)}{\lambda - \lambda_1}\right)^{3/2}$ progress, as required by Lemma 13 in time $O\left((\text{nnz}(\mathbf{A}) + dm) \cdot \log\left(\frac{\gamma}{\lambda - \lambda_1}\right)\right)$ time. Hence by Lemma 13 we can make constant factor expected progress in minimizing $f(x)$ in time:

$$O\left(\left(\text{nnz}(\mathbf{A}) + d\frac{\gamma^2 + 12\lambda_1\left\|\mathbf{A}\right\|_F^2}{(\lambda - \lambda_1 + \gamma)^2}\right)\log\left(\frac{\gamma}{\lambda - \lambda_1}\right)\sqrt{\frac{\gamma}{\lambda - \lambda_1}}\right)$$

By our assumption, we have $\text{nnz}(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2} = \frac{d\lambda_1\|\mathbf{A}\|_F^2}{(\lambda - \lambda_1)^2}$. So, if we let $\gamma = \Theta\left(\sqrt{\frac{d\lambda_1\|\mathbf{A}\|_F^2}{\text{nnz}(\mathbf{A})}}\right)$ then using a sufficiently large constant, we have $\gamma \geq 2(\lambda - \lambda_1)$. We have $\frac{\gamma}{\lambda - \lambda_1} = \Theta\left(\sqrt{\frac{d\lambda_1\|\mathbf{A}\|_F^2}{\text{nnz}(\mathbf{A})\lambda_1^2\text{gap}^2}}\right) = \Theta\left(\sqrt{\frac{d\,\text{sr}(\mathbf{A})}{\text{nnz}(\mathbf{A})\text{gap}^2}}\right)$ and can make constant expected progress in minimizing $f(x)$ in time:

$$O\left(\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}} \cdot \log\left(\frac{d}{\text{gap}}\right)\right).$$

□

4.3 Shifted-and-Inverted Power Method

Finally, we are able to combine the solvers from Sections 4.1 and 4.2 with the framework of Section 3 to obtain faster algorithms for top eigenvector computation.

Theorem 16 (Shifted-and-Inverted Power Method With SVRG)**.** *Let* $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ *for* $\left(1+\frac{\text{gap}}{150}\right)\lambda_1 \leq \lambda \leq \left(1+\frac{\text{gap}}{100}\right)\lambda_1$ *and let* $x_0 \sim \mathcal{N}(0,\mathbf{I})$ *be a random initial vector. Running the inverted power method on* $\mathbf{B}$ *initialized with* x_0*, using the SVRG solver from Theorem 12 to approximately apply* $\mathbf{B}^{-1}$ *at each step, returns* x *such that with probability* $1 - O\left(\frac{1}{d^{10}}\right)$*,* $x^\top\mathbf{\Sigma}x \geq (1-\epsilon)\lambda_1$ *in total time*

$$O\left(\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right)\cdot\left(\log^2\left(\frac{d}{\text{gap}}\right) + \log\left(\frac{1}{\epsilon}\right)\right)\right).$$

Note that by instantiating the above theorem with $\epsilon' = \epsilon\cdot\text{gap}$, and applying Lemma 1 we can find a unit vector x such that $|v_1^\top x| \geq 1-\epsilon$ in the same asymptotic running time (an extra $\log(1/\text{gap})$ term is absorbed into the $\log^2(d/\text{gap})$ term).

Proof. By Theorem 8, if we start with $x_0 \sim \mathcal{N}(0,\mathbf{I})$ we can run $O\left(\log\left(\frac{d}{\text{gap}}\right)\right)$ iterations of the inverted power method, to obtain x_1 with $G(x_1) \leq \frac{1}{\sqrt{10}}$ with probability $1 - O\left(\frac{1}{d^{10}}\right)$. Each iteration requires applying an linear solver that decreases initial error in expectation by a factor of $\frac{1}{\text{poly}(d,1/\text{gap})}$. Such a solver is given by applying the solver in Theorem 12 $O\left(\log\left(\frac{d}{\text{gap}}\right)\right)$ times, decreasing error by a constant factor in expectation each time. So overall in order to find x_1 with $G(x_1) \leq \frac{1}{\sqrt{10}}$, we require time $O\left(\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right)\cdot\log^2\left(\frac{d}{\text{gap}}\right)\right)$.

After this initial 'burn-in' period we can apply Corollary 6 of Theorem 5, which shows that running a single iteration of the inverted power method will decrease $G(x)$ by a constant factor in expectation. In such an iteration, we only need to use a solver that decreases initial error by a constant factor in expectation. So we can perform each inverted power iteration in this stage in time $O\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right)$.

With $O\left(\log\left(\frac{d}{\epsilon}\right)\right)$ iterations, we can obtain x with $\mathbb{E}\left[G(x)^2\right] = O\left(\frac{\epsilon}{d^{10}}\right)$ So by Markov's inequality, we have $G(x)^2 = O(\epsilon)$, giving us $x^T\mathbf{\Sigma}x \geq (1-O(\epsilon))\lambda_1$ by Lemma 3. Union bounding over both stages gives us failure probability $O\left(\frac{1}{d^{10}}\right)$, and adding the runtimes from the two stages gives us the final result. Note that the second stage requires $O\left(\log\left(\frac{d}{\epsilon}\right)\right) = O(\log d + \log(1/\epsilon))$ iterations to achieve the high probability bound. However, the $O(\log d)$ term is smaller than the $O\left(\log^2\left(\frac{d}{\text{gap}}\right)\right)$ term, so is absorbed into the asymptotic notation.

□

We can apply an identical analysis using the accelerated solver from Theorem 15, obtaining the following runtime which beats Theorem 16 whenever $\text{nnz}(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}$:

Theorem 17 (Shifted-and-Inverted Power Method Using Accelerated SVRG)**.** *Let* $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ *for* $\left(1+\frac{\text{gap}}{150}\right)\lambda_1 \leq \lambda \leq \left(1+\frac{\text{gap}}{100}\right)\lambda_1$ *and let* $x_0 \sim \mathcal{N}(0,\mathbf{I})$ *be a random initial vector. Assume that* $\text{nnz}(\mathbf{A}) \leq \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}$*. Running the inverted power method on* $\mathbf{B}$ *initialized with* x_0*, using the accelerated SVRG solver from Theorem 15 to approximately apply* $\mathbf{B}^{-1}$ *at each step, returns* x *such that with probability* $1 - O\left(\frac{1}{d^{10}}\right)$*,* $|v_1^\top x| \geq 1-\epsilon$ *in total time*

$$O\left(\left(\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}}\right)\cdot\left(\log^3\left(\frac{d}{\text{gap}}\right) + \log\left(\frac{d}{\text{gap}}\right)\log\left(\frac{1}{\epsilon}\right)\right)\right).$$

5 Online Eigenvector Computation

Here we show how to apply the shifted-and-inverted power method framework of Section 3 to the online setting. This setting is more difficult than the offline case. As there is no canonical matrix $\mathbf{A}$, and we only have access to the distribution $\mathcal{D}$ through samples, in order to apply Theorem 5 we must show how to both estimate the Rayleigh quotient (Section 5.1) as well as solve the requisite linear systems in expectation (Section 5.2).

After laying this ground work, our main result is given in Section 5.3. Ultimately, the results in this section allow us to achieve more efficient algorithms for computing the top eigenvector in the statistical setting as well as improve upon the previous best known sample complexity for top eigenvector computation. As we show in Section 7 the bounds we provide in this section are in fact tight for general distributions.

5.1 Estimating the Rayleigh Quotient

Here we show how to estimate the Rayleigh quotient of a vector with respect to $\mathbf{\Sigma}$. Our analysis is standard – we first approximate the Rayleigh quotient by its empirical value on a batch of k samples and prove using Chebyshev's inequality that the error on this sample is small with constant probability. We then repeat this procedure $O(\log(1/p))$ times and output the median. By a Chernoff bound this yields a good estimate with probability $1-p$. The formal statement of this result and its proof comprise the remainder of this subsection.

Theorem 18 (Online Rayleigh Quotient Estimation)**.** *Given $\epsilon \in (0,1]$, $p \in [0,1]$, and unit vector x set $k = \lceil 4\,\mathrm{v}(\mathcal{D})\epsilon^{-2} \rceil$ and $m = O(\log(1/p))$. For all $i \in [k]$ and $j \in [m]$ let $a_i^{(j)}$ be drawn independently from $\mathcal{D}$ and set $R_{i,j} = x^\top a_i^{(j)} (a_i^{(j)})^\top x$ and $R_j = \frac{1}{k}\sum_{i\in[k]} R_{i,j}$. If we let z be median value of the R_j then with probability $1-p$ we have $\left|z - x^\top \mathbf{\Sigma} x\right| \leq \epsilon\lambda_1$.*

Proof.

$$\begin{aligned}
\mathrm{Var}_{a\sim\mathcal{D}}(x^\top aa^\top x) &= \mathbb{E}_{a\sim\mathcal{D}}(x^\top aa^\top x)^2 - (\mathbb{E}_{a\sim\mathcal{D}} x^\top aa^\top x)^2 \\
&\leq \mathbb{E}_{a\sim\mathcal{D}} \|a\|_2^2\, x^\top aa^\top x - (x^\top \mathbf{\Sigma} x)^2 \\
&\leq \left\| \mathbb{E}_{a\sim\mathcal{D}} \|a\|_2^2\, aa^\top \right\|_2 = \mathrm{v}(\mathcal{D})\lambda_1^2
\end{aligned}$$

Consequently, $\mathrm{Var}(R_{i,j}) \leq \mathrm{v}(\mathcal{D})\lambda_1^2$, and since each of the $a_i^{(j)}$ were drawn independently this implies that we have that $\mathrm{Var}(R_j) \leq \mathrm{v}(\mathcal{D})\lambda_1^2/k$. Therefore, by Chebyshev's inequality

$$\mathbb{P}\left[|R_j - \mathbb{E}[R_j]| \geq 2\sqrt{\frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{k}} \right] \leq \frac{1}{4}.$$

Since $\mathbb{E}[R_j] = x^\top \mathbf{\Sigma} x$ and since we defined k appropriately this implies that

$$\mathbb{P}\left[\left| R_j - x^\top \mathbf{\Sigma} x \right| \geq \epsilon\lambda_1 \right] \leq \frac{1}{4}. \tag{12}$$

The median z satisfies $|z - x^\top \mathbf{\Sigma} x| \leq \epsilon$ as more than half of the R_j satisfy $|R_j - x^\top \mathbf{\Sigma} x| \leq \epsilon$. This happens with probability $1-p$ by Chernoff bound, our choice of m and (12). □

5.2 Solving the Linear system

Here we show how to solve linear systems in $\mathbf{B} = \lambda\mathbf{I} - \mathbf{\Sigma}$ in the streaming setting. We follow the general strategy of the offline algorithms in Section 4, replacing traditional SVRG with the streaming SVRG algorithm of [FGKS15a]. Similarly to the offline case we minimize $f(x) = \frac{1}{2}x^\top \mathbf{B}x - b^\top x$ and define for all $a \in \text{supp}(\mathcal{D})$,

$$\psi_a(x) \stackrel{\text{def}}{=} \frac{1}{2}x^\top(\lambda\mathbf{I} - aa^\top)x - b^\top x. \tag{13}$$

insuring that $f(x) = \mathbb{E}_{a\sim\mathcal{D}}\psi_a(x)$..

The performance of streaming SVRG [FGKS15a] is governed by three regularity parameters. As in the offline case, we use the fact that $f(\cdot)$ is μ-strongly convexity for $\mu = \lambda - \lambda_1$ and we require a smoothness parameter, denoted $\overline{S}$, that satisfies:

$$\forall x \in \mathbb{R}^d \quad : \quad \mathbb{E}_{a\sim\mathcal{D}} \left\|\bigtriangledown\psi_a(x) - \bigtriangledown\psi_a(x^{\text{opt}})\right\|_2^2 \leq 2\overline{S}\left[f(x) - f(x^{\text{opt}})\right] \ . \tag{14}$$

Furthermore, we require an upper bound the variance, denoted σ^2, that satisfies:

$$\mathbb{E}_{a\sim\mathcal{D}}\frac{1}{2}\left\|\bigtriangledown\psi_a(x^{\text{opt}})\right\|^2_{(\bigtriangledown^2 f(x^{\text{opt}}))^{-1}} \leq \sigma^2 \ . \tag{15}$$

With the following two lemmas we bound these parameters.

Lemma 19 (Streaming Smoothness)**.** *The smoothness parameter* $\overline{S} \stackrel{\text{def}}{=} \lambda + \frac{\text{v}(\mathcal{D})\lambda_1^2}{\lambda-\lambda_1}$ *satisfies* (14)*.*

Proof. Our proof is similar to the one for Lemma 10.

$$\begin{aligned}
\mathbb{E}_{a\sim\mathcal{D}}\left\|\bigtriangledown\psi_a(x) - \bigtriangledown\psi_a(x^{\text{opt}})\right\|_2^2 &= \mathbb{E}_{a\sim\mathcal{D}}\left\|(\lambda\mathbf{I} - aa^\top)(x - x^{\text{opt}})\right\|_2^2 \\
&= \lambda^2\left\|x - x^{\text{opt}}\right\|_2^2 - 2\lambda\mathbb{E}_{a\sim\mathcal{D}}\left\|x - x^{\text{opt}}\right\|^2_{aa^\top} + \mathbb{E}_{a\sim\mathcal{D}}\left\|aa^\top(x - x^{\text{opt}})\right\|_2^2 \\
&\leq \lambda^2\left\|x - x^{\text{opt}}\right\|_2^2 - 2\lambda\left\|x - x^{\text{opt}}\right\|^2_{\mathbf{\Sigma}} + \left\|\mathbb{E}_{a\sim\mathcal{D}}\left\|a\right\|_2^2 aa^\top\right\|_2 \cdot \left\|x - x^{\text{opt}}\right\|_2^2 \\
&\leq \lambda\left\|x - x^{\text{opt}}\right\|^2_{\mathbf{B}} + \text{v}(\mathcal{D})\lambda_1^2\left\|x - x^{\text{opt}}\right\|_2^2.
\end{aligned}$$

Since f is $\lambda - \lambda_1$-strongly convex, $\left\|x - x^{\text{opt}}\right\|_2^2 \leq \frac{2}{\lambda-\lambda_1}[f(x) - f(x^{\text{opt}})]$. Furthermore, since direct calculation reveals, $2[f(x) - f(x^{\text{opt}})] = \left\|x - x^{\text{opt}}\right\|^2_{\mathbf{B}}$, the result follows. □

Lemma 20 (Streaming Variance)**.** *The variance parameter* $\sigma^2 \stackrel{\text{def}}{=} \frac{\text{v}(\mathcal{D})\lambda_1^2}{\lambda-\lambda_1}\left\|x^{\text{opt}}\right\|_2^2$ *satisfies* (15)*.*

Proof. We have

$$\begin{aligned}
\mathbb{E}_{a\sim\mathcal{D}}\frac{1}{2}\left\|\bigtriangledown\psi_a(x^{\text{opt}})\right\|^2_{(\bigtriangledown^2 f(x^{\text{opt}}))^{-1}} &= \mathbb{E}_{a\sim\mathcal{D}}\frac{1}{2}\left\|\left(\lambda\mathbf{I} - aa^\top\right)x^{\text{opt}} - b\right\|^2_{\mathbf{B}^{-1}} \\
&= \mathbb{E}_{a\sim\mathcal{D}}\frac{1}{2}\left\|\left(\lambda\mathbf{I} - aa^\top\right)x^{\text{opt}} - \mathbf{B}x^{\text{opt}}\right\|^2_{\mathbf{B}^{-1}} \\
&= \mathbb{E}_{a\sim\mathcal{D}}\frac{1}{2}\left\|\left(\mathbf{\Sigma} - aa^\top\right)x^{\text{opt}}\right\|^2_{\mathbf{B}^{-1}}.
\end{aligned}$$

Applying $\mathbb{E}\left\|a-\mathbb{E}a\right\|_2^2=\mathbb{E}\left\|a\right\|_2^2-\left\|\mathbb{E}a\right\|_2^2$ gives:

$$\mathbb{E}_{a\sim\mathcal{D}}\left\|\left(\mathbf{\Sigma}-aa^\top\right)x^{\text{opt}}\right\|_{\mathbf{B}^{-1}}^2=\mathbb{E}_{a\sim\mathcal{D}}\left\|x^{\text{opt}}\right\|_{aa^\top\mathbf{B}^{-1}aa^\top}^2-\left\|x^{\text{opt}}\right\|_{\mathbf{\Sigma}\mathbf{B}^{-1}\mathbf{\Sigma}}^2\leq\mathbb{E}_{a\sim\mathcal{D}}\left\|x^{\text{opt}}\right\|_{aa^\top\mathbf{B}^{-1}aa^\top}^2.$$

Furthermore, since $\mathbf{B}^{-1}\preceq\frac{1}{\lambda-\lambda_1}\mathbf{I}$ we have

$$\mathbb{E}_{a\sim\mathcal{D}}aa^\top\mathbf{B}^{-1}aa^\top\preceq\frac{1}{\lambda-\lambda_1}\mathbb{E}_{a\sim\mathcal{D}}(aa^\top)^2\preceq\left(\frac{\left\|\mathbb{E}_{a\sim\mathcal{D}}(aa^\top)^2\right\|_2}{\lambda-\lambda_1}\right)\mathbf{I}=\left(\frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{\lambda-\lambda_1}\right)\mathbf{I}.$$

Combining these three equations yields the result. □

With the regularity parameters bounded we can apply the streaming SVRG algorithm of [FGKS15a] to solve systems in $\mathbf{B}$. We encapsulate the core iterative step of Algorithm 1 of [FGKS15a] as follows:

Definition 21 (Streaming SVRG Step)**.** *Given $x_0\in\mathbb{R}^d$ and $\eta,k,m>0$ we define a* streaming SVRG step*, $x=\text{ssvrg\_iter}(x_0,\eta,k,m)$ as follows. First we take k samples $a_1,...,a_k$ from $\mathcal{D}$ and set $g=\frac{1}{k}\sum_{i\in[k]}\psi_{a_i}$ where ψ_{a_i} is as defined in (13). Then for $\widetilde{m}$ chosen uniformly at random from $\{1,...,m\}$ we draw $\widetilde{m}$ additional samples $\widetilde{a}_1,...,\widetilde{a}_{\widetilde{m}}$ from $\mathcal{D}$. For $t=0,...,\widetilde{m}-1$ we let*

$$x_{t+1}:=x_t-\frac{\eta}{L}\left(\bigtriangledown\psi_{\widetilde{a}_t}(x_t)-\bigtriangledown\psi_{\widetilde{a}_t}(x_0)+\bigtriangledown g(x_0)\right)$$

and return $x_{\widetilde{m}}$ as the output.

The accuracy of the above iterative step is proven in Theorem 4.1 of [FGKS15a], which we include, using our notation below:

Theorem 22 (Theorem 4.1 of [FGKS15a] [1])**.** *Let $f(x)=\mathbb{E}_{a\sim\mathcal{D}}\psi_a(x)$ and let μ, $\overline{S}$, σ^2 be the strong convexity, smoothness, and variance bounds for $f(x)$. Then for any distribution over x_0 we have that $x:=\text{ssvrg\_iter}(x_0,\eta,k,m)$ has $\mathbb{E}[f(x)-f(x^{\text{opt}})]$ upper bounded by*

$$\frac{1}{1-4\eta}\left[\left(\frac{\overline{S}}{\mu m\eta}+4\eta\right)\left[\mathbb{E}f(x_0)-f(x^{\text{opt}})\right]+\frac{1+2\eta}{k}\left(\sqrt{\frac{\overline{S}}{\mu}\cdot\left[\mathbb{E}f(x_0)-f(x^{\text{opt}})\right]}+\sigma\right)^2\right].$$

Using Theorem 22 we can immediately obtain the following guarantee for solve system in $\mathbf{B}$:

Corollary 23 (Streaming SVRG Solver - With Initial Point)**.** *Let $\mu=\lambda-\lambda_1$, $\overline{S}=\lambda+\frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{\lambda-\lambda_1}$, and $\sigma^2=\frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{\lambda-\lambda_1}\left\|x^{\text{opt}}\right\|_2^2$. Let $c_2,c_3\in(0,1)$ be any constants and set $\eta=\frac{c_2}{8}$, $m=\left\lceil\frac{\overline{S}}{\mu c_2^2}\right\rceil$, and $k=\max\left\{\left\lceil\frac{\overline{S}}{\mu c_2}\right\rceil,\left\lceil\frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{(\lambda-\lambda_1)^2c_3}\right\rceil\right\}$. If to solve $\mathbf{B}x=b$ for unit vector b with initial point x_0, we use the iterative procedure described in Definition 21 to compute $x=\text{ssvrg\_iter}(x_0,\eta,k,m)$ then:*

$$\mathbb{E}\left\|x-x^{\text{opt}}\right\|_{\mathbf{B}}^2\leq22c_2\cdot\left\|x_0-x^{\text{opt}}\right\|_{\mathbf{B}}^2+10c_3\lambda_1(\mathbf{B}^{-1}).$$

Further, the procedure requires $O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2}\left[\frac{1}{c_2^2}+\frac{1}{c_3}\right]\right)$ samples from $\mathcal{D}$.

[1] Note that Theorem 4.1 in [FGKS15a] has an additional parameter of α, which bounds the Hessian of $f(x^{\text{opt}})$ in comparison to the Hessian everywhere else. In our setting this parameter is 1 as $\bigtriangledown^2f(y)=\bigtriangledown^2f(z)$ for all y and z.

Proof. Using the inequality $(x+y)^2 \le 2x^2 + 2y^2$ we have that

$$\left(\sqrt{\frac{\overline{S}}{\mu} \cdot \mathbb{E}[f(x_0) - f(x^{\text{opt}})]} + \sigma\right)^2 \le \frac{2\overline{S}}{\mu} \cdot \mathbb{E}[f(x_0) - f(x^{\text{opt}})] + 2\sigma^2$$

Additionally, since b is a unit vector, we know that $\left\|x^{\text{opt}}\right\|_2^2 = \left\|\mathbf{B}^{-1}b\right\|_2^2 \le \frac{1}{(\lambda-\lambda_1)^2}$. Using equation (10), i.e. that $\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 = 2[f(x) - f(x^{\text{opt}})]$ for all x, we have by Theorem 22:

$$\begin{aligned}\mathbb{E}\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 &\le \frac{1}{1 - c_2/2}\left[\left(8c_2 + \frac{c_2}{2} + \frac{4+c_2}{2} \cdot c_2\right) \cdot \left\|x_0 - x^{\text{opt}}\right\|_{\mathbf{B}}^2 + \frac{4+c_2}{4k} \cdot \frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{(\lambda-\lambda_1)^3}\right] \\ &\le 22c_2 \cdot \left\|x_0 - x^{\text{opt}}\right\|_{\mathbf{B}}^2 + \frac{10c_3}{\lambda - \lambda_1} = 22c_2 \cdot \left\|x_0 - x^{\text{opt}}\right\|_{\mathbf{B}}^2 + 10c_3\lambda_1(\mathbf{B}^{-1}).\end{aligned}$$

Since $1/(\lambda - \lambda_1) = \lambda_1(\mathbf{B}^{-1})$ we see that $\mathbb{E}\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2$ is as desired. All that remains is to bound the number of samples we used.

Now the number of samples used to compute x is clearly at most $m + k$ Now

$$m = \frac{\overline{S}}{\mu c_2^2} = O\left(\frac{\lambda}{c_2^2(\lambda - \lambda_1)} + \frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{c_2^2(\lambda-\lambda_1)^2}\right) = O\left(\frac{1}{c_2^2\text{gap}} + \frac{\mathrm{v}(\mathcal{D})}{c_2^2\text{gap}^2}\right).$$

. However since gap < 1 and $\mathrm{v}(\mathcal{D}) \ge 1$ this simplifies to $m = O\left(\frac{\mathrm{v}}{c_2^2\text{gap}^2}\right)$. Next to bound k we can ignore the $\left\lceil\frac{\overline{S}}{\mu c_2}\right\rceil$ term since this was already included in our bound of m and just bound $\frac{\mathrm{v}(\mathcal{D})\lambda_1^2}{c_3(\lambda-\lambda_1)^2} = O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2 c_3}\right)$ yielding our desired sample complexity. □

Whereas in the offline case, we could ensure that our initial error $\left\|x_0 - x^{\text{opt}}\right\|_{\mathbf{B}}^2$ is small by simply scaling by the Rayleigh quotient (Corollary 6) in the online case estimating the Rayleigh quotient to sufficient accuracy would require too many samples. Instead, here simply show how to simply apply Corollary 23 iteratively to solve the desired linear systems to absolute accuracy without an initial point. Ultimately, due to the different error dependences in the online case this guarantee suffices and the lack of an initial point is not a bottleneck.

Corollary 24 (Streaming SVRG Solver)**.** *There is a streaming algorithm that iteratively applies the solver of Corollary 23 to solve* $\mathbf{B}x = b$ *for unit vector* b *and returns a vector* x *that satisfies* $\mathbb{E}\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 \le 10c\lambda_1(\mathbf{B}^{-1})$ *using* $O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2 \cdot c}\right)$ *samples from* $\mathcal{D}$*.*

Proof. Let $x_0 = 0$. Then $\left\|x_0 - x^{\text{opt}}\right\|_{\mathbf{B}}^2 = \left\|\mathbf{B}^{-1}b\right\|_{\mathbf{B}}^2 \le \lambda_1(\mathbf{B}^{-1})$ since b is a unit vector. If we apply Corollary 23 with $c_2 = \frac{1}{44}$ and $c_3 = \frac{1}{20}$, then we will obtain x_1 with $\mathbb{E}\left\|x_1 - x^{\text{opt}}\right\|_{\mathbf{B}}^2 \le \frac{1}{2}\lambda_1(\mathbf{B}^{-1})$. If we then double c_3 and apply the solver again we obtain x_2 with $\mathbb{E}\left\|x_1 - x^{\text{opt}}\right\|_{\mathbf{B}}^2 \le \frac{1}{4}\lambda_1(\mathbf{B}^{-1})$. Iterating in this way, after $\log(1/c)$ iterations we will have the desired guarantee: $\mathbb{E}\left\|x - x^{\text{opt}}\right\|_{\mathbf{B}}^2 \le 10c\lambda_1(\mathbf{B}^{-1})$. Our total sample cost in each iteration is, by Corollary 23, $O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2}\left[\frac{1}{44^2} + \frac{1}{c_3}\right]\right)$. Since we double c_3 each time, the cost corresponding to the $\frac{1}{c_3}$ terms is dominated by the last iteration when we have $c_3 = O(c)$. So our overall sample cost is just:

$$O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2}\left[\frac{1}{c} + \log(1/c)\right]\right) = O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2 \cdot c}\right).$$

□

5.3 Online Shifted-and-Inverted Power Method

We now apply the results in Section 5.1 and Section 5.2 to the shifted-and-inverted power method framework of Section 3 to give our main result in the online setting, an algorithm that quickly refines a coarse approximation to v_1 into a finer approximation.

Theorem 25 (Online Shifted-and-Inverted Power Method – Warm Start)**.** *Let* $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ *for* $\left(1 + \frac{\text{gap}}{150}\right)\lambda_1 \leq \lambda \leq \left(1 + \frac{\text{gap}}{100}\right)\lambda_1$ *and let* x_0 *be some vector with* $G(x_0) \leq \frac{1}{\sqrt{10}}$*. Running the shifted-and-inverted power method on* $\mathbf{B}$ *initialized with* x_0*, using the streaming SVRG solver of Corollary 24 to approximately apply* $\mathbf{B}^{-1}$ *at each step, returns* x *such that* $x^\top\mathbf{\Sigma}x \geq (1-\epsilon)\lambda_1$ *with constant probability for any target* $\epsilon < \text{gap}$*. The algorithm uses* $O(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}\cdot\epsilon})$ *samples and amortized* $O(d)$ *time per sample.*

We note that by instantiating Theorem 25, with $\epsilon' = \epsilon \cdot \text{gap}$, and applying Lemma 1 we can find x such that $|v_1^\top x| \geq 1 - \epsilon$ with constant probability in time $O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2\cdot\epsilon}\right)$.

Proof. By Lemma 3 it suffices to have $G^2(x) = O(\frac{\epsilon}{\text{gap}})$ or equivalently $G(x) = O(\sqrt{\epsilon/\text{gap}})$. In order to succed with constant probability it suffices to have $\mathbb{E}[G(x)] = O(\sqrt{\epsilon/\text{gap}})$ with constant probability. Since we start with $G(x_0) \leq \frac{1}{\sqrt{10}}$, we can achieve this using $\log(\text{gap}/\epsilon)$ iterations of the approximate shifted-and-inverted power method of Theorem 5. In each iteration i we choose the error parameter for Theorem 5 to be $c_1(i) = \frac{1}{\sqrt{10}} \cdot \left(\frac{1}{5}\right)^i$. Consequently,

$$\mathbb{E}[G(x_i)] \leq \frac{3}{25}G(x_{i-1}) + \frac{4}{1000}\frac{1}{\sqrt{10}} \cdot \left(\frac{1}{5}\right)^i$$

and by induction $\mathbb{E}[G(x_i)] \leq \frac{1}{5^i}\frac{1}{\sqrt{10}}$. We halt when $(\frac{1}{5})^i = O(\sqrt{\epsilon/\text{gap}})$ and hence $c_1(i) = O(\sqrt{\epsilon/\text{gap}})$.

In order to apply Theorem 5 we need a subroutine $\widehat{\text{quot}}(x)$ that lets us approximate $\text{quot}(x)$ to within an additive error $\frac{1}{30}(\lambda - \lambda_1) = O(\text{gap} \cdot \lambda_1)$. Theorem 18 gives us such a routine, requiring $O\left(\frac{\mathrm{v}(\mathcal{D})\log\log(\text{gap}/\epsilon)}{\text{gap}^2}\right) = O(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}\cdot\epsilon})$ samples to succeed with probability $1 - O\left(\frac{1}{\log(\text{gap}/\epsilon)}\right)$ (since $\epsilon < \text{gap}$). Union bounding, the estimation succeeds in all rounds with constant probability.

By Corollary 24 with $c = \Theta(c_1(i)^2)$ the cost for solving each linear system solve is $O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}^2 c_1(i)^2}\right)$. Since $c_1(i)$ multiplies by a constant factor with each iteration the cost over all $O(\log(\text{gap}/d\epsilon)$ iterations is just a truncated geometric series and is proportional to cost in the last iteration, when $c = \Theta\left(\frac{\epsilon}{\text{gap}}\right)$. So the total cost for solving the linear systems is $O\left(\frac{\mathrm{v}(\mathcal{D})}{\text{gap}\cdot\epsilon}\right)$. Adding this to the number of samples for the Rayleigh quotient estimation yields the result. □

6 Parameter Estimation for Offline Eigenvector Computation

In Section 4, in order to invoke Theorems 5 and 8 we assumed knowledge of some λ with $(1 + c_1 \cdot \text{gap})\lambda_1 \leq \lambda \leq (1 + c_2 \cdot \text{gap})\lambda_1$ for some small constant c_1 and c_2. Here we show how to estimate this parameter using Algorithm 1, incurring a modest additional runtime cost.

In this section, for simplicity we initially assume that we have oracle access to compute $\mathbf{B}_\lambda^{-1}x$ for any given x, and any $\lambda > \lambda_1$. We will then show how to achieve the same results when we

Algorithm 1 Estimating the eigenvalue and the eigengap

Input: $\mathbf{A} \in \mathbb{R}^{n\times d}$, α

$w = [w_1, w_2] \leftarrow \mathcal{N}(0,1)^{d\times 2}$

$t \leftarrow O(\alpha \log d)$

$\left[\widetilde{\lambda}_1^{(0)}, \widetilde{\lambda}_2^{(0)}\right] \leftarrow eigEstimate\left(\left(\mathbf{A}^T\mathbf{A}\right)^t w\right)$

$\overline{\lambda}^{(0)} \leftarrow (1+\frac{1}{2})\widetilde{\lambda}_1^{(0)}$

$i \leftarrow 0$

while $\overline{\lambda}^{(i)} - \widetilde{\lambda}_1^{(i)} < \frac{1}{10}\left(\overline{\lambda}^{(i)} - \widetilde{\lambda}_2^{(i)}\right)$ **do**

  $i \leftarrow i+1$

  $w = [w_1, w_2] \leftarrow \mathcal{N}(0,1)^{d\times 2}$

  $\left[\widehat{\lambda}_1^{(i)}, \widehat{\lambda}_2^{(i)}\right] \leftarrow eigEstimate\left(\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-t} w\right)$

  $\left[\widetilde{\lambda}_1^{(i)}, \widetilde{\lambda}_2^{(i)}\right] \leftarrow \left[\overline{\lambda}^{(i-1)} - \frac{1}{\widehat{\lambda}_1^{(i)}}, \overline{\lambda}^{(i-1)} - \frac{1}{\widehat{\lambda}_2^{(i)}}\right]$

  $\overline{\lambda}^{(i)} \leftarrow \frac{1}{2}\left(\widetilde{\lambda}_1^{(i)} + \overline{\lambda}^{(i-1)}\right)$

end while

Output: λ

can only compute $\mathbf{B}_\lambda^{-1}x$ approximately. We use a result of [MM15] that gives gap free bounds for computing eigenvalues using the power method. The following is a specialization of Theorem 1 from [MM15]:

Theorem 26. *For any $\epsilon > 0$, any matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$ with eigenvalues $\lambda_1, ..., \lambda_d$, and $k \leq d$, let $\mathbf{W} \in \mathbb{R}^{d\times k}$ be a matrix with entries drawn independently from $\mathcal{N}(0,1)$. Let $eigEstimate(\mathbf{Y})$ be a function returning for each i, $\tilde{\lambda}_i = \tilde{v}_i^\top \mathbf{M}\tilde{v}_i$ where $\tilde{v}_i$ is the i^{th} largest left singular vector of $\mathbf{Y}$. Then setting $[\tilde{\lambda}_1, ..., \tilde{\lambda}_k] = eigEstimate\left(\mathbf{M}^t\mathbf{W}\right)$, for some fixed constant c and $t = c\alpha\log d$ for any $\alpha > 1$, with probability $1 - \frac{1}{d^{10}}$, we have for all i:*

$$|\tilde{\lambda}_i - \lambda_i| \leq \frac{1}{\alpha}\lambda_{k+1}$$

Throughout the proof, we assume α is picked to be some large constant - e.g. $\alpha > 100$. Theorem 26 implies:

Lemma 27. *Conditioning on the event that Theorem 26 holds for all iterates i, then the iterates of Algorithm 1 satisfy:*

$$0 \leq \lambda_1 - \widetilde{\lambda}_1^{(0)} \leq \frac{1}{\alpha}\lambda_1 \quad \textit{and} \quad \frac{1}{2}\left(1 - \frac{3}{\alpha}\right)\lambda_1 \leq \overline{\lambda}^{(0)} - \lambda_1 \leq \frac{1}{2}\lambda_1, \quad \textit{and,}$$

$$0 \leq \lambda_1 - \widetilde{\lambda}_1^{(i)} \leq \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(i-1)} - \lambda_1\right) \quad \textit{and} \quad \frac{1}{2}\left(1 - \frac{1}{\alpha-1}\right)\left(\overline{\lambda}^{(i-1)} - \lambda_1\right) \leq \overline{\lambda}^{(i)} - \lambda_1 \leq \frac{1}{2}\left(\overline{\lambda}^{(i-1)} - \lambda_1\right).$$

Proof. The proof can be decomposed into two parts:

Part I (Lines 3-4): Theorem 26 tells us that $\widetilde{\lambda}_1^{(0)} \geq \left(1 - \frac{1}{\alpha}\right)\lambda_1$. This means that we have

$$0 \leq \lambda_1 - \widetilde{\lambda}_1^{(0)} \leq \frac{1}{\alpha}\lambda_1 \text{ and } \frac{1}{2}\left(1 - \frac{3}{\alpha}\right)\lambda_1 \leq \overline{\lambda}^{(0)} - \lambda_1 \leq \frac{1}{2}\lambda_1.$$

Part II (Lines 5-6): Consider now iteration i. We now apply Theorem 26 to the matrix $\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-1}$. The top eigenvalue of this matrix is $\left(\overline{\lambda}^{(i-1)} - \lambda_1\right)^{-1}$. This means that we have $\left(1 - \frac{1}{\alpha}\right)\left(\overline{\lambda}^{(i-1)} - \lambda_1\right)^{-1} \leq \widehat{\lambda}_1^{(i)} \leq \left(\overline{\lambda}^{(i-1)} - \lambda_1\right)^{-1}$, and hence we have,

$$0 \leq \lambda_1 - \widetilde{\lambda}_1^{(i)} \leq \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(i-1)} - \lambda_1\right) \text{ and } \frac{1}{2}\left(1 - \frac{1}{\alpha - 1}\right)\left(\overline{\lambda}^{(i-1)} - \lambda_1\right) \leq \overline{\lambda}^{(i)} - \lambda_1 \leq \frac{1}{2}\left(\overline{\lambda}^{(i-1)} - \lambda_1\right).$$

This proves the lemma. □

Lemma 28. *Recall we denote* $\lambda_2 \overset{\text{def}}{=} \lambda_2\left(\mathbf{A}^T\mathbf{A}\right)$ *and* $\text{gap} \overset{\text{def}}{=} \frac{\lambda_1 - \lambda_2}{\lambda_1}$. *Then conditioning on the event that Theorem 26 holds for all iterates* i, *the iterates of Algorithm 1 satisfy* $\left|\lambda_2 - \widetilde{\lambda}_2^{(i)}\right| \leq \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(i-1)} - \lambda_2\right)$, *and* $\overline{\lambda}^{(i)} - \widetilde{\lambda}_2^{(i)} \geq \frac{\text{gap}\lambda_1}{4}$.

Proof. Since $\left(\overline{\lambda}^{(i-1)} - \lambda_2\right)^{-1}$ is the second eigenvalue of the matrix $\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-1}$, Theorem 26 tells us that

$$\left(1 - \frac{1}{\alpha}\right)\left(\overline{\lambda}^{(i-1)} - \lambda_2\right)^{-1} \leq \widehat{\lambda}_2^{(i)} \leq \left(1 + \frac{1}{\alpha}\right)\left(\overline{\lambda}^{(i-1)} - \lambda_2\right)^{-1}.$$

This immediately yields the first claim. For the second claim, we notice that

$$\begin{aligned}
\overline{\lambda}^{(i)} - \widetilde{\lambda}_2^{(i)} &= \overline{\lambda}^{(i)} - \lambda_2 + \lambda_2 - \widetilde{\lambda}_2^{(i)} \\
&\overset{(\zeta_1)}{\geq} \overline{\lambda}^{(i)} - \lambda_2 - \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(i-1)} - \lambda_2\right) \\
&= \overline{\lambda}^{(i)} - \lambda_1 - \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(i-1)} - \lambda_1\right) + \left(1 - \frac{1}{\alpha - 1}\right)(\lambda_1 - \lambda_2) \\
&\overset{(\zeta_2)}{\geq} \frac{1}{2}\left(1 - \frac{3}{\alpha - 1}\right)\left(\overline{\lambda}^{(i-1)} - \lambda_1\right) + \left(1 - \frac{1}{\alpha - 1}\right)(\lambda_1 - \lambda_2) \geq \frac{\text{gap}\lambda_1}{4},
\end{aligned}$$

where (ζ_1) follows from the first claim of this lemma, and (ζ_2) follows from Lemma 27. □

We now state and prove the main result in this section:

Theorem 29. *Suppose* $\alpha > 100$, *and after* T *iterations, Algorithm 1 exits. Then with probability* $1 - \frac{\left\lceil \log \frac{10}{\text{gap}} \right\rceil + 1}{d^{10}}$, *we have* $T \leq \left\lceil \log \frac{10}{\text{gap}} \right\rceil + 1$, *and:*

$$\left(1 + \frac{\text{gap}}{120}\right)\lambda_1 \leq \overline{\lambda}^{(T)} \leq \left(1 + \frac{\text{gap}}{8}\right)\lambda_1$$

Proof. By union bound, we know with probability $1 - \frac{\left\lceil \log \frac{10}{\text{gap}} \right\rceil + 1}{d^{10}}$, Theorem 26 will hold for all iterates where $i \le \left\lceil \log \frac{10}{\text{gap}} \right\rceil + 1$.

Let $\bar{i} = \left\lceil \log \frac{10}{\text{gap}} \right\rceil$, suppose the algorithm has not exited yet after $\bar{i}$ iterations, then since $\overline{\lambda}^{(i)} - \lambda_1$ decays geometrically, we have $\overline{\lambda}^{(\bar{i})} - \lambda_1 \le \frac{\text{gap}\lambda_1}{10}$. Therefore, Lemmas 27 and 28 imply that $\overline{\lambda}^{(\bar{i}+1)} - \widetilde{\lambda}_1^{(\bar{i}+1)} \le \left(\frac{1}{2} + \frac{1}{\alpha - 1}\right)\left(\overline{\lambda}^{(\bar{i})} - \lambda_1\right) \le \frac{\text{gap}\lambda_1}{15}$, and

$$\begin{aligned}\overline{\lambda}^{(\bar{i}+1)} - \widetilde{\lambda}_2^{(\bar{i}+1)} &\ge \overline{\lambda}^{(\bar{i}+1)} - \lambda_2 - \left|\lambda_2 - \widetilde{\lambda}_2^{(\bar{i}+1)}\right| \ge \lambda_1 - \lambda_2 - \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(\bar{i})} - \lambda_2\right) \\ &= \text{gap}\lambda_1 - \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(\bar{i})} - \lambda_1 + \lambda_1 - \lambda_2\right) \ge \frac{3}{4}\text{gap}\lambda_1\end{aligned}$$

This means that the exit condition on Line 6 must be triggered in $\bar{i} + 1$ iteration, proving the first part of the lemma.

For upper bound, by Lemmas 27, 28 and exit condition we know:

$$\begin{aligned}\overline{\lambda}^{(T)} - \lambda_1 \le \overline{\lambda}^{(T)} - \widetilde{\lambda}_1^{(T)} &\le \frac{1}{10}(\overline{\lambda}^{(T)} - \widetilde{\lambda}_2^{(T)}) \le \frac{1}{10}\left(\overline{\lambda}^{(T)} - \lambda_2 + \left|\lambda_2 - \widetilde{\lambda}_2^{(T)}\right|\right) \\ &\le \frac{1}{10}\left(\overline{\lambda}^{(T)} - \lambda_2 + \frac{1}{\alpha - 1}(\overline{\lambda}^{(T-1)} - \lambda_2)\right) \\ &= \frac{1}{10}\left(\frac{\alpha}{\alpha - 1}\text{gap}\lambda_1 + (\overline{\lambda}^{(T)} - \lambda_1) + \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(T-1)} - \lambda_1\right)\right) \\ &\le \frac{1}{10}\left(\frac{\alpha}{\alpha - 1}\text{gap}\lambda_1 + \frac{\alpha}{\alpha - 2}\left(\overline{\lambda}^{(T)} - \lambda_1\right)\right)\end{aligned}$$

Since $\alpha > 100$, this directly implies $\overline{\lambda}^{(T)} - \lambda_1 \le \frac{\text{gap}}{8}\lambda_1$.

For lower bound, since as long as the Algorithm 1 does not exists, by Lemmas 28, we have $\overline{\lambda}^{(T-1)} - \widetilde{\lambda}_1^{(T-1)} \ge \frac{1}{10}\left(\overline{\lambda}^{(T-1)} - \widetilde{\lambda}_2^{(T-1)}\right) \ge \frac{\text{gap}\lambda_1}{40}$, and thus:

$$\begin{aligned}\overline{\lambda}^{(T-1)} - \lambda_1 = \overline{\lambda}^{(T-1)} - \widetilde{\lambda}_1^{(T-1)} - (\lambda_1 - \widetilde{\lambda}_1^{(T-1)}) &\ge \frac{\text{gap}\lambda_1}{40} - \frac{1}{\alpha - 1}\left(\overline{\lambda}^{(T-1)} - \lambda_1\right) \\ &\ge \frac{\text{gap}\lambda_1}{40} - \frac{2}{\alpha - 2}\left(\overline{\lambda}^{(T)} - \lambda_1\right) \ge \frac{\text{gap}\lambda_1}{50}\end{aligned}$$

By Lemma 27, we know $\overline{\lambda}^{(T)} - \lambda_1 \ge \frac{1}{2}(1 - \frac{1}{\alpha - 1}(\overline{\lambda}^{(T-1)} - \lambda_1)) > \frac{\text{gap}}{120}\lambda_1$ □

Note that, although we proved the upper bound and lower bound in Theorem 29 with specific constants coefficient $\frac{1}{8}$ and $\frac{1}{120}$, this analysis can easily be extended to any smaller constants by modifying the constant in the exit condition, and choosing α larger. Also in the failure probability

$$1 - \frac{\left\lceil \log \frac{10}{\text{gap}} \right\rceil + 1}{d^{10}},$$

the term d^{10} can be replaced by any $\text{poly}(d)$ by adjusting the constant in setting $t \leftarrow O(\alpha \log d)$ in Algorithm 1. Assuming $\log \frac{1}{\text{gap}} < \text{poly}(d)$, thus gives that Theorem 29 returns a correct result with high probability.

Finally, we can also bound the runtime of algorithm 1, when we use SVRG based approximate linear system solvers for $\mathbf{B}_\lambda$.

Theorem 30. *With probability* $1 - O(\frac{1}{d^{10}} \log \frac{1}{\text{gap}})$*, Algorithm 1 runs in time*

$$O\left(\left[\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right] \cdot \log^3\left(\frac{d}{\text{gap}}\right)\right)$$

.

Proof. By Theorem 29, we know only $O(\log 1/\text{gap})$ iterations of the algorithm are needed. In each iteration, the runtime is dominated by running *eigEstimate* $\left(\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-t} w\right)$, which is dominated by computing $\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-t} w$. Since $t = O(\log d)$, it's easy to verify that: to make Theorem 26 hold, we only need to approximate $\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-1} w$ up to accuracy $\text{poly}(\text{gap}/d)$. By Theorem 12, we know this approximation can be calculated in time

$$O\left(\left[\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})\lambda_1^2}{(\overline{\lambda}^{(i-1)} - \lambda_1)^2}\right] \cdot \log\left(\frac{d}{\text{gap}}\right)\right)$$

. Combining Theorem 29 with Lemma 27, we know $\overline{\lambda}^{(i-1)} - \lambda_1 \geq \overline{\lambda}^{(T)} - \lambda_1 \geq \frac{\text{gap}}{120}$, thus approximately solving $\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-1} w$ can be done in time $\tilde{O}\left(\text{nnz}(\mathbf{A}) + \frac{d\,\text{sr}(\mathbf{A})}{\text{gap}^2}\right)$. Finally, since the runtime of Algorithm 1 is dominated by repeating this subroutine $t \times T = O(\log d \cdot \log(1/\text{gap}))$ times, we finish the proof. □

Note that we can accelerate the runtime of Algorithm 1 to $\tilde{O}\left(\frac{\text{nnz}(\mathbf{A})^{3/4}(d\,\text{sr}(\mathbf{A}))^{1/4}}{\sqrt{\text{gap}}}\right)$, by simply replacing the base solver for $\left(\overline{\lambda}^{(i-1)}\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)^{-1} w$ with the accelerated solver in Theorem 15.

7 Lower Bounds

Here we show that our online eigenvector estimation algorithm (Theorem 25) is asymptotically optimal - as sample size grows large it achieves optimal accuracy as a function of sample size. We rely on the following lower bound for eigenvector estimation in the Gaussian spike model:

Lemma 31 (Lower bound for Gaussian Spike Model [BJNP13])**.** *Suppose data is generated as*

$$a_i = \sqrt{\lambda}\iota_i v^\star + Z_i \tag{16}$$

where $\iota_i \sim \mathcal{N}(0,1)$*, and* $Z_i \sim \mathcal{N}(0, I_d)$*. Let* $\hat{v}$ *be some estimator of the top eigenvector* $v^\star$*. Then, there is some universal constant* c_0*, so that for* n *sufficiently large, we have:*

$$\inf_{\hat{v}} \max_{v^\star \in \mathbb{S}^{d-1}} \mathbb{E}\left\|\hat{v} - v^\star\right\|_2 \geq c_0 \frac{(1+\lambda)d}{\lambda^2 n}$$

Theorem 32. *Consider the problem of estimating the top eigenvector v_1 of $\mathbb{E}_{a\sim\mathcal{D}}aa^\top$, where we observe n i.i.d samples from unknown distribution $\mathcal{D}$. If gap < 0.9, then there exists some universal constant c, such that for any estimator $\hat{v}$ of top eigenvector, there always exists some hard distribution $\mathcal{D}$ so that for n sufficiently large:*

$$\mathbb{E}\left\|\hat{v}-v_1\right\|_2^2 \geq c\frac{\mathrm{v}(\mathcal{D})}{\mathrm{gap}^2 n}$$

Proof. Suppose the claim of theorem is not true, then there exist some estimator $\hat{v}$ so that

$$\mathbb{E}\left\|\hat{v}-v_1\right\|_2^2 < c'\frac{\mathrm{v}(\mathcal{D})}{\mathrm{gap}^2 n}$$

holds for all distribution $\mathcal{D}$, and for any fixed constant c' when n is sufficiently large.

Let distribution $\mathcal{D}$ be the Gaussian Spike Model specified by Eq.(16), then by calculation, it's not hard to verify that:

$$\mathrm{v}(\mathcal{D}) = \frac{\left\|\mathbb{E}_{a\sim\mathcal{D}}\left[\left(aa^\top\right)^2\right]\right\|_2}{\left\|\mathbb{E}_{a\sim\mathcal{D}}(aa^\top)\right\|_2^2} = \frac{d+2+3\lambda}{1+\lambda}$$

Since we know gap $= \frac{\lambda}{1+\lambda} < 0.9$, this implies $\lambda < 9$, which gives $\mathrm{v}(\mathcal{D}) < \frac{d+29}{1+\lambda} < \frac{30d}{1+\lambda}$. Therefore, we have that:

$$\mathbb{E}\left\|\hat{v}-v^\star\right\|_2^2 < c'\frac{\mathrm{v}(\mathcal{D})}{\mathrm{gap}^2 n} < 30c'\frac{(1+\lambda)d}{\lambda^2 n}$$

holds for all $v^\star \in \mathbb{S}^{d-1}$. Choose $c' = \frac{c_0}{30}$ in Lemma 31 we have a contradiction. □

$\left\|\hat{v}-v_1\right\|_2^2 = 2-2\hat{v}^\top v_1$, so this bound implies that- to obtain $|\hat{v}^\top v_1| \geq 1-\epsilon$, we need $\frac{\mathrm{v}(\mathcal{D})}{\mathrm{gap}^2 n} = O(\epsilon)$ so $n = \Theta\left(\frac{\mathrm{v}(\mathcal{D})}{\mathrm{gap}^2\epsilon}\right)$. This exactly matches the sample complexity given by Theorem 25.

8 Gap-Free Bounds

In this section we demonstrate that our techniques can easily be extended to obtain gap-free runtime bounds, for the regime when $\epsilon \geq$ gap. In many ways these bounds are actually much easier to achieve than the gap dependent bounds since they require less careful error analysis.

Let ϵ be our error parameter and m be the number of eigenvalues of Σ that are $\geq (1-\epsilon/2)\lambda_1$. Choose $\lambda = \lambda_1 + \epsilon/100$. We have $\lambda_1(\mathbf{B}^{-1}) = \frac{100}{\epsilon\lambda_1}$. For $i > m$ we have $\lambda_i(\mathbf{B}^{-1}) < \frac{2}{\epsilon\lambda_1}$. $\kappa(\mathbf{B}^{-1}) \leq \frac{100}{\epsilon}$.

Let $\mathbf{V}_b$ have columns equal to all *bottom* eigenvectors with eigenvalues $\lambda_i < (1-\epsilon/2)\lambda_1$. Let $\mathbf{V}_t$ have columns equal to the m remaining *top* eigenvectors. We define a simple modified potential:

$$\bar{G}(x) \stackrel{\text{def}}{=} \frac{\left\|\mathbf{P}_{\mathbf{V}_b}x\right\|_{\mathbf{B}}}{\left\|\mathbf{P}_{v_1}x\right\|_{\mathbf{B}}} = \frac{\sqrt{\sum_{i>m}\frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}}}{\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}}}$$

We have the following Lemma connecting this potential function to eigenvalue error:

Lemma 33. *For unit x, if $\bar{G}(x) \leq c\sqrt{\epsilon}$ for sufficiently small constant c then $\lambda_1 - x^\top\mathbf{\Sigma}x \leq \epsilon\lambda_1$.*

Proof.

$$\bar{G}(x) \geq \frac{\|\mathbf{P}_{\mathbf{V}_b} x\|_2}{\|\mathbf{P}_{v_1} x\|_2} \geq \frac{\|\mathbf{P}_{\mathbf{V}_b} x\|_2}{\|\mathbf{P}_{\mathbf{V}_t} x\|_2}$$

So if $\bar{G}(x) \leq c\sqrt{\epsilon}$ then $\|\mathbf{P}_{\mathbf{V}_t} x\|_2^2 c^2 \epsilon \geq \|\mathbf{P}_{\mathbf{V}_b} x\|_2^2$ and since $\|\mathbf{P}_{\mathbf{V}_t} x\|_2^2 + \|\mathbf{P}_{\mathbf{V}_b} x\|_2^2 = 1$, this gives $\|\mathbf{P}_{\mathbf{V}_t} x\|_2^2 \geq \frac{1}{1+c^2\epsilon}$. So we have $x^T \mathbf{\Sigma} x \geq \mathbf{P}_{\mathbf{V}_t} x^T \mathbf{\Sigma} x \mathbf{P}_{\mathbf{V}_t} \geq \frac{(1-\epsilon/2)\lambda_1}{1+c^2\epsilon} \geq 1-\epsilon$ for small enough c, giving the lemma. □

We now follow the proof of Lemma 8, which is actually much simpler in the gap-free case.

Theorem 34 (Approximate Shifted-and-Inverted Power Method – Gap-Free)**.** *Suppose we randomly initialize x_0 as in Lemma 7 and suppose we have access to a subroutine* solve $(\cdot)$ *such that*

$$\mathbb{E}\left[\left\|\text{solve}\,(x) - \mathbf{B}^{-1}x\right\|_{\mathbf{B}}\right] \leq \frac{\epsilon^3}{3000d^{21}}\sqrt{\lambda_d(\mathbf{B}^{-1})}$$

Then the following procedure,

$$x_t = \text{solve}\,(x_{t-1}) \,/\, \|\text{solve}\,(x_{t-1})\|$$

after $T = O\left(\log d/\epsilon\right)$ iterations satisfies:

$$\bar{G}(x_T) \leq c\sqrt{\epsilon},$$

with probability greater than $1 - O(\frac{1}{d^{10}})$.

Proof. By Lemma 7, we know with at least probability $1 - O(\frac{1}{d^{10}})$, we have $\bar{G}(x_0) \leq G(x_0) \leq \sqrt{\kappa(\mathbf{B}^{-1})}d^{10.5} = \frac{100d^{10.5}}{\epsilon}$. We want to show by induction that at iteration i we have $\bar{G}(x_i) \leq \frac{1}{2^i} \cdot \frac{100d^{10.5}}{\epsilon}$, which will give us the lemma if we set $T = \log_2\left(\frac{100d^{10.5}}{c\epsilon^{1.5}}\right) = O(\log(d/\epsilon))$.

Let $\widehat{x} = \text{solve}\,(x)$ and $\xi = \widehat{x} - \mathbf{B}^{-1}x$. Following Lemma 8 we have:

$$\begin{aligned}
\|\mathbf{P}_{\mathbf{V}_b}(\widehat{x})\|_{\mathbf{B}} &\leq \left\|\mathbf{P}_{\mathbf{V}_b}\left(\mathbf{B}^{-1}x\right)\right\|_{\mathbf{B}} + \|\mathbf{P}_{\mathbf{V}_b}(\xi)\|_{\mathbf{B}} \leq \left\|\mathbf{P}_{\mathbf{V}_b}\left(\mathbf{B}^{-1}x\right)\right\|_{\mathbf{B}} + \|\xi\|_{\mathbf{B}} \\
&= \sqrt{\sum_{i>m} \alpha_i^2 \lambda_i(\mathbf{B}^{-1})} + \|\xi\|_{\mathbf{B}} \\
&\leq \lambda_{m+1}(\mathbf{B}^{-1})\left(\sqrt{\sum_{i>m}\frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}} + \frac{\epsilon^3}{3000d^{21}\sqrt{\lambda_{m+1}(\mathbf{B}^{-1})}}\right) \\
&\leq 2\lambda_{m+1}(\mathbf{B}^{-1})\max\left\{\sqrt{\sum_{i>m}\frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}}, \frac{\epsilon^3}{3000d^{21}\sqrt{\lambda_{m+1}(\mathbf{B}^{-1})}}\right\}
\end{aligned}$$

and

$$\begin{aligned}
\|\mathbf{P}_{v_1}(\widehat{x})\|_{\mathbf{B}} &\geq \left\|\mathbf{P}_{v_1}\left(\mathbf{B}^{-1}x\right)\right\|_{\mathbf{B}} - \|\mathbf{P}_{v_1}(\xi)\|_{\mathbf{B}} \geq \left\|\mathbf{P}_{v_1}\left(\mathbf{B}^{-1}x\right)\right\|_{\mathbf{B}} - \|\xi\|_{\mathbf{B}} \\
&= \sqrt{\alpha_1^2\lambda_1(\mathbf{B}^{-1})} - \|\xi\|_{\mathbf{B}} \\
&\geq \lambda_1(\mathbf{B}^{-1})\sqrt{\frac{\alpha_1^2 - \frac{\epsilon^6}{(3000d^{21})^2}}{\lambda_1(\mathbf{B}^{-1})}}.
\end{aligned}$$

Initially, we have with high probability, by the argument in Lemma 7, $\alpha_1 \geq \frac{1}{d^{10}}$ so we have $\|\mathbf{P}_{v_1}(\widehat{x})\|_{\mathbf{B}} \geq \frac{\lambda_1(\mathbf{B}^{-1})}{2}\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}}$. This also holds by induction in each iteration.

Let $\hat{\alpha}_1 = |v_1^\top \widehat{x}|/\|\widehat{x}\|_2$. $\|\mathbf{P}_{v_1}(\widehat{x})\|_{\mathbf{B}}^2 = \frac{\hat{\alpha}_1^2\|\hat{x}\|_2^2}{\lambda_1(\mathbf{B}^{-1})}$ so we have

$$\hat{\alpha}_1^2 \geq \frac{\lambda_1(\mathbf{B}^{-1})^2}{\|\hat{x}\|_2^2}\left(\alpha_1^2 - \frac{\epsilon^6}{(3000d^{21})^2}\right)$$

and since $\|\hat{x}\|_2^2 \leq 2\left(\left\|\mathbf{B}^{-1}x\right\|_2^2 + 2\|\xi\|_2^2\right) \leq \lambda_1(\mathbf{B}^{-1})^2 + 2\frac{\epsilon^6}{(3000d^{21})^2} \leq \lambda_1(\mathbf{B}^{-1})^2\left(2 + 2\frac{\epsilon^6}{(3000d^{21})^2}\right)$ we have:

$$\hat{\alpha}_1^2 \geq \frac{1}{2.1}\left(\alpha_1^2 - \frac{\epsilon^6}{(3000d^{21})^2}\right) \geq \frac{1}{3}\alpha_1^2.$$

So over all $\log_2\left(\frac{100d^{10.5}}{c\epsilon^{1.5}}\right)$ iterations, we always have $\hat{\alpha}_1^2 \geq \frac{1}{d^{10}}\cdot\left(\frac{c\epsilon^{1.5}}{100d^{10.5}}\right)^{\log_2 3}$ and so $\frac{\epsilon^6}{(3000d^{21})^2} << 1/2\alpha_1^2$. Combining the above bounds:

$$\begin{aligned}\bar{G}(\widehat{x}) &\leq \frac{2\lambda_{m+1}\left(\mathbf{B}^{-1}\right)}{\lambda_1\left(\mathbf{B}^{-1}\right)/2}\cdot\frac{\max\left\{\sqrt{\sum_{i>m}\frac{\alpha_i^2}{\lambda_i(\mathbf{B}^{-1})}}, \frac{\epsilon^3}{3000d^{21}\sqrt{\lambda_{m+1}(\mathbf{B}^{-1})}}\right\}}{\sqrt{\frac{\alpha_1^2}{\lambda_1(\mathbf{B}^{-1})}}} \\ &\leq \frac{4}{50}\max\left\{\bar{G}(x), O(\sqrt{\epsilon})\right\}.\end{aligned}$$

This is enough to give the Theorem. □

Finally, we combine Theorem 34 with the SVRG based solvers of Theorem 12 and 15 to obtain:

Theorem 35 (Gap-Free Shifted-and-Inverted Power Method With SVRG)**.** *Let* $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ *for* $\lambda = \left(1 + \frac{\epsilon}{100}\right)$ *and let* $x_0 \sim \mathcal{N}(0, \mathbf{I})$ *be a random initial vector. Running the inverted power method on* $\mathbf{B}$ *initialized with* x_0*, using the SVRG solver from Theorem 12 to approximately apply* $\mathbf{B}^{-1}$ *at each step, returns* x *such that with probability* $1 - O\left(\frac{1}{d^{10}}\right)$, $x^\top\mathbf{\Sigma}x \geq (1-\epsilon)\lambda_1$ *in time*

$$O\left(\left(\mathrm{nnz}(\mathbf{A}) + \frac{d\,\mathrm{sr}(\mathbf{A})}{\epsilon^2}\right)\cdot\log^2\left(\frac{d}{\epsilon}\right)\right).$$

Theorem 36 (Accelerated Gap-Free Shifted-and-Inverted Power Method With SVRG)**.** *Let* $\mathbf{B} = \lambda\mathbf{I} - \mathbf{A}^\top\mathbf{A}$ *for* $\lambda = \left(1 + \frac{\epsilon}{100}\right)$ *and let* $x_0 \sim \mathcal{N}(0, \mathbf{I})$ *be a random initial vector. Running the inverted power method on* $\mathbf{B}$ *initialized with* x_0*, using the SVRG solver from Theorem 15 to approximately apply* $\mathbf{B}^{-1}$ *at each step, returns* x *such that with probability* $1 - O\left(\frac{1}{d^{10}}\right)$, $x^\top\mathbf{\Sigma}x \geq (1-\epsilon)\lambda_1$ *in total time*

$$O\left(\frac{\mathrm{nnz}(\mathbf{A})^{3/4}(d\,\mathrm{sr}(\mathbf{A}))^{1/4}}{\sqrt{\epsilon}}\cdot\log^3\left(\frac{d}{\epsilon}\right)\right).$$

9 Acknowledgements

Sham Kakade acknowledges funding from the Washington Research Foundation for innovation in Data-intensive Discovery.

References

[AC09] Nir Ailon and Bernard Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

[BDF13] Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 3174–3182, 2013.

[BJNP13] Aharon Birnbaum, Iain M Johnstone, Boaz Nadler, and Debashis Paul. Minimax bounds for sparse PCA with noisy high-dimensional data. *Annals of Statistics*, 41(3):1055, 2013.

[Bot10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010.

[CLM+15] Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 181–190, 2015.

[CR15] Dominik Csiba and Peter Richtárik. Primal method for ERM with flexible mini-batching schemes and non-convex losses. *arXiv:1506.02227*, 2015.

[CW13] Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 81–90, 2013.

[FGKS15a] Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Proceedings of the 28th Annual Conference on Computational Learning Theory (COLT)*, pages 728–763, 2015.

[FGKS15b] Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[GH15] Dan Garber and Elad Hazan. Fast and simple PCA via convex optimization. *arXiv:1509.05647*, 2015.

[HP14] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2861–2869, 2014.

[JKM+15] 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. *arXiv:1510.08896*, 2015.

[Jol02] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 315–323, 2013.

[Kor03] Yehuda Koren. On spectral graph drawing. In *Computing and Combinatorics*, pages 496–508. Springer, 2003.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 296–303. ACM, 2014.

[LMH15] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. *arXiv:1506.02186*, 2015.

[MCJ13] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming PCA. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 2886–2894, 2013.

[MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

[NJW02] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 849–856, 2002.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: bringing order to the Web. 1999.

[Saa92] Yousef Saad. *Numerical methods for large eigenvalue problems*. SIAM, 1992.

[Sha15a] Ohad Shamir. Convergence of stochastic gradient descent for PCA. *arXiv:1509.09002*, 2015.

[Sha15b] Ohad Shamir. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. *arXiv:1507.08788*, 2015.

[Sha15c] Ohad Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 144–152, 2015.

[Spi07] Daniel A Spielman. Spectral graph theory and its applications. In *null*, pages 29–38. IEEE, 2007.

[SRO15] Christopher D Sa, Christopher Re, and Kunle Olukotun. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2332–2341, 2015.

[SS15] Shai Shalev-Shwartz. SDCA without duality. *arXiv:1502.06177*, 2015.

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *arXiv:1501.01571*, 2015.

[VW04] Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

[Wil12] Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 887–898, 2012.

A Appendix

Lemma 37 (Eigenvector Estimation via Spectral Norm Matrix Approximation)**.** *Let* $\mathbf{A}^\top\mathbf{A}$ *have top eigenvector* 1*, top eigenvector* v_1 *and eigenvalue gap* gap*. Let* $\mathbf{B}^\top\mathbf{B}$ *be some matrix with* $\left\|\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B}\right\|_2 \leq O(\sqrt{\epsilon}\cdot\text{gap})$*. Let* x *be the top eigenvector of* $\mathbf{B}^\top\mathbf{B}$*. Then:*

$$|x^\top v_1| \geq 1 - \epsilon \ .$$

Proof. We can any unit vector y as $y = c_1 v_1 + c_2 v_2$ where v_2 is the component of x orthogonal to v_1 and $c_1^2 + c_2^2 = 1$. We know that

$$\begin{aligned} v_1^\top\mathbf{B}^\top\mathbf{B}v_1 &= v_1^\top\mathbf{A}^\top\mathbf{A}v_1 - v_1^T(\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B})v_1 \\ 1 - \sqrt{\epsilon}\text{gap} \leq v_1^\top\mathbf{B}^\top\mathbf{B}v_1 &\leq 1 + \sqrt{\epsilon}\text{gap} \end{aligned}$$

Similarly we can compute:

$$\begin{aligned} v_2^\top\mathbf{B}^\top\mathbf{B}v_2 &= v_2^\top\mathbf{A}^\top\mathbf{A}v_2 - v_2^T(\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B})v_2 \\ 1 - \text{gap} - \sqrt{\epsilon}\text{gap} \leq v_2^\top\mathbf{B}^\top\mathbf{B}v_2 &\leq 1 - \text{gap} + \sqrt{\epsilon}\text{gap}. \end{aligned}$$

and

$$\begin{aligned} |v_1^\top\mathbf{B}^\top\mathbf{B}v_2| &= |v_1^\top\mathbf{A}^\top\mathbf{A}v_2 - v_1^T(\mathbf{A}^\top\mathbf{A} - \mathbf{B}^\top\mathbf{B})v_2| \\ &\leq \sqrt{\epsilon}\text{gap}. \end{aligned}$$

We have $x^\top\mathbf{B}\mathbf{B}^\top x = c_1^2(v_1^\top\mathbf{B}^\top\mathbf{B}v_1) + c_2^2(v_2^\top\mathbf{B}^\top\mathbf{B}v_2) + 2c_1c_2\cdot v_2^\top\mathbf{B}^\top\mathbf{B}v_1$.
We want to bound $c_1 \geq 1-\epsilon$ so $c_1^2 \geq 1 - O(\epsilon)$. Since x is the top eigenvector of $\mathbf{B}\mathbf{B}^\top$ we have:

$$\begin{aligned} x^\top\mathbf{B}\mathbf{B}^\top x &\geq v_1^\top\mathbf{B}\mathbf{B}^\top v_1 \\ c_2^2(v_2^\top\mathbf{B}^\top\mathbf{B}v_2) + 2c_2v_2^\top\mathbf{B}^\top\mathbf{B}v_1 &\geq (1-c_1^2)v_1^\top\mathbf{B}\mathbf{B}^\top v_1 \\ 2\sqrt{1-c_1^2}\sqrt{\epsilon}\text{gap} &\geq (1-c_1^2)\left(v_1^\top\mathbf{B}\mathbf{B}^\top v_1 - v_2^\top\mathbf{B}\mathbf{B}^\top v_2\right) \\ \frac{1}{\sqrt{1-c_1^2}} &\geq \frac{(1-2\sqrt{\epsilon})\text{gap}}{2\sqrt{\epsilon}\text{gap}} \\ \frac{1}{1-c_1^2} &\geq \frac{1-5\sqrt{\epsilon}}{4\epsilon} \end{aligned}$$

This means we need have $1 - c_1^2 \leq O(\epsilon)$ meaning $c_1^2 \geq 1 - O(\epsilon)$ as desired. □

Lemma 38 (Inverted Power Method progress in ℓ_2 and $\mathbf{B}$ norms)**.** *Let x be a unit vector with $\langle x, v_1\rangle \neq 0$ and let $\widetilde{x} = \mathbf{B}^{-1}w$, i.e. the power method update of $\mathbf{B}^{-1}$ on x. Then, we have both:*

$$\frac{\left\|\mathbf{P}_{v_1^\perp}\widetilde{x}\right\|_{\mathbf{B}}}{\|\mathbf{P}_{v_1}\widetilde{x}\|_{\mathbf{B}}} \leq \frac{\lambda_2(\mathbf{B}^{-1})}{\lambda_1(\mathbf{B}^{-1})} \cdot \frac{\left\|\mathbf{P}_{v_1^\perp}x\right\|_{\mathbf{B}}}{\|\mathbf{P}_{v_1}x\|_{\mathbf{B}}} \tag{17}$$

and

$$\frac{\left\|\mathbf{P}_{v_1^\perp}\widetilde{x}\right\|_{2}}{\|\mathbf{P}_{v_1}\widetilde{x}\|_{2}} \leq \frac{\lambda_2(\mathbf{B}^{-1})}{\lambda_1(\mathbf{B}^{-1})} \cdot \frac{\left\|\mathbf{P}_{v_1^\perp}x\right\|_{2}}{\|\mathbf{P}_{v_1}x\|_{2}} \tag{18}$$

Proof. (17) was already shown in Lemma 4. We show (18) similarly.

Writing x in the eigenbasis of $\mathbf{B}^{-1}$, we have $x = \sum_i \alpha_i v_i$ and $\widetilde{x} = \sum_i \alpha_i \lambda_i \left(\mathbf{B}^{-1}\right) v_i$. Since $\langle x, v_1\rangle \neq 0$, $\alpha_1 \neq 0$ and we have:

$$\frac{\left\|\mathbf{P}_{v_1^\perp}\widetilde{x}\right\|_{2}}{\|\mathbf{P}_{v_1}\widetilde{x}\|_{2}} = \frac{\sqrt{\sum_{i\geq 2}\alpha_i^2\lambda_i^2(\mathbf{B}^{-1})}}{\sqrt{\alpha_1^2\lambda_1^2(\mathbf{B}^{-1})}} \leq \frac{\lambda_2\left(\mathbf{B}^{-1}\right)}{\lambda_1\left(\mathbf{B}^{-1}\right)} \cdot \frac{\sqrt{\sum_{i\geq 2}\alpha_i^2}}{\sqrt{\alpha_1^2}} = \frac{\lambda_2\left(\mathbf{B}^{-1}\right)}{\lambda_1\left(\mathbf{B}^{-1}\right)} \cdot \frac{\left\|\mathbf{P}_{v_1^\perp}x\right\|_{2}}{\|\mathbf{P}_{v_1}x\|_{2}}.$$

□