Adaptive Power Method: Eigenvector Estimation from Sampled Data

Seiyun Shin SEIYUNS2@ILLINOIS.EDU
University of Illinois at Urbana-Champaign

Han Zhao HANZHAO@ILLINOIS.EDU
University of Illinois at Urbana-Champaign

Ilan Shomorony ILANS@ILLINOIS.EDU
University of Illinois at Urbana-Champaign

Editors: Shipra Agrawal and Francesco Orabona

Abstract

Computing the dominant eigenvectors of a matrix A has many applications, such as principal component analysis, spectral embedding, and PageRank. However, in general, this task relies on the complete knowledge of the matrix A, which can be too large to store or even infeasible to observe in many applications, e.g., large-scale social networks. Thus, a natural question is how to accurately estimate the eigenvectors of A when only partial observations can be made by sampling entries from A. To this end, we propose the Adaptive Power Method (APM), a variant of the well-known power method. At each power iteration, APM adaptively selects a subset of the entries of A to observe based on the current estimate of the top eigenvector. We show that APM can estimate the dominant eigenvector(s) of A with squared error at most ϵ by observing roughly $O(n\epsilon^{-2}\log^2(n/\epsilon))$ entries of an $n \times n$ matrix. We present empirical results for the problem of eigenvector centrality computation on two real-world graphs and show that APM significantly outperforms a non-adaptive estimation algorithm using the same number of observations. Furthermore, in the context of eigenvector centrality, APM can also adaptively allocate the observation budget to selectively refine the estimate of nodes with high centrality scores in the graph.

Keywords: Adaptive sampling, Eigenvector estimation, Noisy power method

1. Introduction

Computing the dominant eigenvectors of a matrix A is a central task in countless applications. Examples include dimensionality reduction techniques such as principal component analysis and low-rank approximation, spectral clustering, matrix completion, topic modeling, and many other data science problems. For instance, given a search query from a user, Google's PageRank algorithm shows the relevant websites by ranking the relevance of web pages via eigenvector computations on their link structures (Brin and Page, 1998; Page et al., 1998). In addition, in light of the ongoing growth of web-based services (e.g., Facebook), a key challenge for viral marketing in social networks is to identify influencers in social networks (Kiss and Bichler, 2008; De Valck et al., 2009), which can be accomplished through the notion of *eigenvector centrality*. The centrality is given by the principal eigenvector's entries, indicating an importance score of the corresponding entity/node: the higher the score the greater the level of influence within the network. Computing the eigenvector centrality of the nodes in a network also finds applications in computational biology, where the relationships between biological entities such as genes, proteins and metabolites can be modeled as

biological networks such as gene regulatory networks, protein interaction networks, and metabolic networks (Wuchty and Stadler, 2003; Junker and Schreiber, 2011).

There are several well-known algorithms that find the principal eigenvector of a matrix A. One important example is the *power method*. Starting with a randomly drawn normalized vector $\boldsymbol{x}_0$, in the ℓth iteration, the algorithm updates the current eigenvector estimate by computing $A\boldsymbol{x}_{\ell-1}$ and normalizing it, until convergence. The power method computes the dominant eigenvector(s) of an $n \times n$ matrix A, with squared error at most ϵ in $O(\log(n/\epsilon))$ iterations (Van Loan and Golub, 1996; Parlett, 1998). To compute the dominant eigenvectors, the power method relies on the complete knowledge of all entries of A. However, there are many practical applications where observing all entries of A may be prohibitive, particularly in high-dimensional or big data settings. For example, there are at least 4.26 billion web pages (De Kunder, 2022) and the current number of social media users is over 4.59 billion (Statista, 2022). The adjacency matrix corresponding to these networks is not explicitly stored anywhere and must be learned through node/edge queries, making an eigenvector computation fairly non-trivial. Access to the entries of a matrix may also be limited in situations where it is physically difficult to query pairwise relationships, such as in biological networks. For example, the mapping of the neural network circuitry of living organisms (e.g., the connectome of *C. elegans* (Cook et al., 2019)) requires physical probing of the connectivity between neurons. In such cases, it may be too costly to probe all pairwise relationships to build a complete adjacency matrix. Identifying the most useful observations for the subsequent network analysis is thus crucial.

Motivated by settings where one can only access a limited number of the entries of A, we study two related questions: (1) Is it possible to accurately estimate the top eigenvectors of a matrix A from a carefully chosen small set of its entries? (2) Can adaptive sampling be used to improve the estimation accuracy? Notice that sampling entries of A in an adaptive manner can be beneficial as the previously chosen entries may provide information about which entries are more important for the eigenvector estimation problem. Hence our goal is to develop adaptive sampling and estimation algorithms with high eigenvector estimation accuracy, while minimizing the sample complexity.

Since the power method is an iterative algorithm for eigenvector computation, it provides a natural starting point to develop an adaptive algorithm that samples elements of A in a sequential manner. Based on this idea, we propose an algorithm called Adaptive Power Method (APM), shown in a simplified form as Algorithm 1. The APM computes the dominant eigenvector of a symmetric matrix with binary entries $A \in \{0,1\}^{n\times n}$ (such as a graph adjacency matrix) using adaptively sampled entries. Generalizations to asymmetric real-valued (bounded) matrices and to the computation of the dominant k eigenvectors are straightforward. The algorithm starts with an initial random and normalized vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ and iteratively updates the current eigenvector estimate by computing $A^{(\ell)}\boldsymbol{x}_{\ell-1}$, with the sampled and scaled matrix $A^{(\ell)}$ at the ℓth iteration. The key is to adaptively construct the matrix $A^{(\ell)}$ so that $A^{(\ell)}\boldsymbol{x}$ is an unbiased estimate of $A\boldsymbol{x}$. Using concentration inequalities we can then derive a high-probability bound on $\|A^{(\ell)}\boldsymbol{x} - A\boldsymbol{x}\|$, which can be combined with a result called the Noisy Power Method (Hardt and Price, 2014) to establish convergence guarantees for the APM. Here we highlight one useful informal result to illustrate our more general results. See Theorem 2 for our formal convergence guarantee.

Theorem 1 (Informal) *Suppose the spectral gap between the largest singular value of A and the second largest one is sufficiently large. If we run Algorithm 1 for $L = O\left(\log(n/\epsilon)\right)$ iterations and a budget of $B = O(n\epsilon^{-2}\log^2(n/\epsilon))$ samples,* APM *returns the top eigenvector of A with squared error at most ϵ.*

Algorithm 1 ADAPTIVE POWER METHOD (APM) (SIMPLIFIED)

Input: Symmetric matrix $A \in \{0,1\}^{n\times n}$, budget B, number of iterations L, threshold $c > 0$
Output: Approximate top eigenvector $\boldsymbol{x}_L \in \mathbb{R}^n$

1: Draw a random vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ with $\|\boldsymbol{x}_0\| = 1$
2: **for** $\ell = 1, \ldots, L$ **do**
3: Draw $I_{ij}^{(\ell)} \sim \mathrm{Bern}(p_j^{(\ell)})\ \forall (i,j) \in [n] \times [n]$ independently, where

$$p_j^{(\ell)} = \min\left[\frac{B}{nL} \cdot \max\left(x_{(\ell-1)j}^2, c^2\right), 1\right]$$

4: $[A^{(\ell)}]_{ij} \leftarrow \frac{a_{ij}}{p_j^{(\ell)}} I_{ij}^{(\ell)}$
5: $\boldsymbol{y}_\ell \leftarrow A^{(\ell)} \boldsymbol{x}_{\ell-1}$
6: $\boldsymbol{x}_\ell \leftarrow \frac{\boldsymbol{y}_\ell}{\|\boldsymbol{y}_\ell\|}$
7: **end for**

Hence, APM can estimate the top eigenvector of A by accessing only $O(n\epsilon^{-2}\log^2(n/\epsilon))$ out of the n^2 entries of A (all of which are needed in the standard power method). Furthermore, we propose a refinement over APM called $\mathrm{APM}_{\min \mathrm{Var}}$ (Algorithm 2), which further optimizes the allocation of the sample budget across the different rows of A in a non-uniform way. The non-uniform allocation of samples across rows is chosen so as to minimize the total variance of the estimator $A^{(\ell)}\boldsymbol{x}$ of $A\boldsymbol{x}$. Specifically, we demonstrate that the minimum variance can be achieved by allocating the budget in a way that is proportional to the variance of the corresponding eigenvector coordinate estimate. This non-uniform allocation of samples across rows is shown to outperform the basic APM for a variety of choices of the input budget B. Moreover, we observe that this non-uniform budget allocation produces eigenvector estimates with smaller errors on the largest entries. This can be thought of as a way to selectively refine the estimate of the top entries of the principal eigenvector, which is particularly useful in the context of eigenvector centrality estimation, where one is typically more interested in identifying nodes with high centrality scores.

From numerical evaluations using benchmark datasets from Hu et al. (2020), we corroborate our theoretical claims and the algorithms' practicality by demonstrating significant performance improvements over a non-adaptive baseline that uses the same number of observations.

Brief summary of related work: There is an extensive literature on a wide variety of eigenvector computation problems under different constraints. This includes the eigenvector computation with an incomplete knowledge on the underlying matrix (1) using query models that return vectors (Garber et al., 2016; Simchowitz et al., 2017), (2) entry-wise sampling of the matrix (Kamath et al., 2020; Ruggeri and De Bacco, 2019, 2020; Shomorony and Avestimehr, 2014), and (3) with the full matrix in addition to a noise perturbation (Mitliagkas et al., 2013; Hardt and Roth, 2013; Hardt and Price, 2014; Musco and Musco, 2015; Liu et al., 2015; Balcan et al., 2016; Xu and Li, 2019, 2020, 2021, 2022). A detailed discussion of related works is presented in Section 5.

Notation: We focus on the setting where a symmetric matrix $A = (a_{ij}) \in \{0,1\}^{n\times n}$ is fixed and unknown, and hence needs to be sampled in an entrywise manner. We write $A_{i:}$ and $A_{:j}$ to indicate

the ith row and jth column of the matrix A respectively. We use σ_i, $i = 1, \dots, n$, for the ith largest singular value of the matrix A. We also let $[n] := \{1, 2, \dots, n\}$.

Throughout the paper, we denote vectors by a lowercase bold letter (e.g., $\boldsymbol{x} \in \mathbb{R}^n$) and assume them to be in column form. In particular, $\boldsymbol{u}_1$ denotes the top eigenvector of A. Also, $(\cdot)^\top$ denotes the transpose. We indicate the $i$the entry of $\boldsymbol{x}$ with either $[\boldsymbol{x}]_i$ or x_i, and we use $[\boldsymbol{x}]_S$ to denote the concatenation of x_is for all $i \in S$. We denote by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \sum_i x_i y_i \in \mathbb{R}$ the inner product between $\boldsymbol{x}$ and $\boldsymbol{y}$. Unless otherwise mentioned, we use $\| \cdot \|$ for the ℓ_2-norm. Lastly, we write $\mathrm{Bern}(p)$ for a Bernoulli distribution with parameter p.

Outline: In Section 2, we propose the Adaptive Power Method (APM) for the top eigenvector computation. We extend our algorithm via variance reduction techniques in Section 3. In Section 4, we numerically validate the theoretical findings and the practicality afforded by the two adaptive schemes. In Section 5, we discuss related works. We conclude and discuss future works in Section 6. Detailed proofs are presented in the Appendix.

2. Adaptive Power Method

In this section, we introduce the Adaptive Power Method (APM) and provide theoretical guarantees for its sample complexity. As shown in Algorithm 1, the algorithm follows an iterative paradigm inspired by the power method to estimate the principal eigenvector of a matrix A. The algorithm starts with an initial random and normalized vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ and iteratively updates the current eigenvector estimate by computing $A^{(\ell)}\boldsymbol{x}_{\ell-1}$, for a matrix $A^{(\ell)}$ carefully constructed from a small number of sampled entries from A. At the ℓth iteration, Algorithm 1 draws sampling indicator variables $I_{ij}^{(\ell)} \sim \mathrm{Bern}(p_j)$ for all i, j independently with

$$p_j = \min \left[\frac{B}{nL} \cdot \max \left(x_{(\ell-1)j}^2, c^2 \right), 1 \right] \tag{1}$$

and builds the matrix $A^{(\ell)}$ with $[A^{(\ell)}]_{ij} = \frac{a_{ij}}{p_j} I_{ij}^{(\ell)}$. Note that the sampling procedure is adaptive in the sense that the algorithm decides whether to sample a_{ij} depending on the value of $x_{(\ell-1)j}$: the larger value $|x_{(\ell-1)j}|$ is, there exists a higher chance for a_{ij} to be sampled. Since a larger $|x_{(\ell-1)j}|$ contributes more to the computation of $A\boldsymbol{x}_{\ell-1}$ in each iteration, our adaptive sampling effectively allocates the budget in a way such that only the a_{ij}s that contribute significantly to the matrix-vector product will be sampled with high probabilities.

Unbiased estimate of $A\boldsymbol{x}_{\ell-1}$: Algorithm 1 produces an unbiased estimate of $A\boldsymbol{x}_{\ell-1}$ $\forall \ell \in [L]$. This can be readily seen from the entrywise analysis that $\forall (\ell, i) \in [L] \times [n]$,

$$\mathbb{E}\left[[A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i \right] = \sum_{j=1}^{n} p_j \cdot \frac{1}{p_j} a_{ij} x_{(\ell-1)j} = \sum_{j=1}^{n} a_{ij} x_{(\ell-1)j} = [A\boldsymbol{x}_{\ell-1}]_i. \tag{2}$$

Thresholding procedure: We note that the expression for p_j in (1) performs a thresholding on $x_{(\ell-1)j}^2$: if $|x_{(\ell-1)j}| > c$, $p_j = \frac{B}{nL} \cdot x_{(\ell-1)j}^2$; otherwise, $p_j = \frac{B}{nL} \cdot c^2$. This thresholding prevents $\frac{1}{p_j} a_{ij}$ from becoming too large, which allows us to apply Bernstein's inequality when bounding the error of our estimator $A^{(\ell)}\boldsymbol{x}_{\ell-1}$ in Lemma 3. See Appendix C for details. However, this thresholding comes at the cost of an excess sampling of entries of A that contribute little to the eigenvector

estimation (i.e., entries where $|x_{(\ell-1)j}|$ is small). This will make the expected number of samples be $(1+c^2n)\cdot B$ in the worst case. We resolve this by setting c to be small so as to make this additional cost negligible, as explained next.

Number of observations: We claim that the expected number of sampled entries per row and per iteration is at most $\frac{B}{nL}(1+c^2n)$. This can be readily seen from

$$\sum_{j=1}^{n}\mathbb{E}\left[I_{ij}^{(\ell)}\right]=\sum_{j=1}^{n}p_j\le\sum_{j=1}^{n}\frac{B}{nL}\cdot\left(x_{(\ell-1)j}^2+c^2\right)=\frac{B}{nL}\left(1+c^2n\right),\quad\forall(\ell,i)\in[L]\times[n],\quad(3)$$

where we use the fact that $\|\boldsymbol{x}_{\ell-1}\|=1$ in the last step. This implies that the expected total number of sampled entries is at most $(1+c^2n)\cdot B$, since the total number of observations that the APM exploits is $\sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n}I_{ij}^{(\ell)}$. If we set $c\le\sqrt{\frac{\alpha}{n}}$ for some small $\alpha>0$, the expected budget is at most $(1+\alpha)B$. Furthermore, using concentration inequalities, we can show that, with high probability, the actual number of samples is close to $(1+\alpha)B$. This is proved in Appendix B.

Reusing previous samples: Notice that Algorithm 1 does not have memory of past sampled entries and may sample the same entries of A multiple times. A natural improvement is then to use the past sampled entries and to sample entries only among the unsampled ones. This corresponds to setting $p_{ij}^{(\ell)}=1$ if $I_{ij}^{(\ell-1)}=1$ before sampling entries according to equation (1), as shown in the modified Algorithm 1 in Appendix A. Notice that once $I_{ij}^{(\ell)}$ is set to 1 for some ℓ, it remains equal to 1 for subsequent iterations, meaning that the same previously sampled a_{ij} can be reused. The empirical results in Section 4 show that exploiting past sampled entries yields a better performance in practice, particularly when the ratio of the budget to the total number of entries $\frac{B}{n^2}$ is not vanishing. Note that this modification does not violate the unbiasedness in (2). See Appendix A for the detailed description of the algorithm including the memory of the past sampled entries.

Our main theoretical result is a convergence guarantee for APM with respect to ℓ_2 error. As it turns out, this convergence guarantee can be established even for Algorithm 1, without reusing previous samples, and we present this in Theorem 2. Intuitively, the reutilization of previous samples in Algorithm 1 can only improve performance.

Theorem 2 *Suppose Algorithm 1 is run for* $L=O\left(\frac{\sigma_1}{\sigma_1-\sigma_2}\log\left(\frac{n\tau}{\epsilon}\right)\right)$ *iterations with a budget of* $B=\frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$ *samples, and* $c=\frac{\sqrt{\alpha}(\sigma_1-\sigma_2)}{n^{3/2}}$. *Then the eigenvector estimate* $\boldsymbol{x}_L$ *satisfies*

$$\left\|\boldsymbol{u}_1-\langle\boldsymbol{u}_1,\boldsymbol{x}_L\rangle\,\boldsymbol{x}_L\right\|\le\epsilon,\quad(4)$$

with probability at least $1-n^{-\Omega(1)}-\tau^{-\Omega(1)}$.

Generalization to real-valued matrices A: Our techniques can be readily extended to obtain similar results for a real-valued matrix A, as long as the entries are all bounded (i.e., $a_{ij}\in[-W,W]$, $\forall(i,j)$, where $W<\infty$). The main difference in the analysis will be that the error in the estimate of Ax will be increased by a factor of W. This will affect the two upper bounds in Proposition 7 (having an additional factor of W for the norm and W^2 for the second moment). Hence, as long as W is a constant (i.e., is not a function of n), it will not affect the sample complexity. We also believe that it may be possible to obtain similar results even when the entries of A are not

bounded, but we will need some kind of tail bound on the entries (such as subgaussianity) to establish the error bounds.

The regime where Theorem 2 provides the most meaningful gains in terms of sample complexity is when the spectral gap satisfies $\sigma_1 - \sigma_2 = \Theta(n)$ (and $\sigma_1 = \Theta(n)$). In this regime, the number of iterations required in Theorem 2 is $L = O(\log(n/\epsilon))$ and the expected number of samples used is upper bounded by $(1+\alpha)B = (1+\alpha) \cdot \frac{\sigma_1}{(\sigma_1-\sigma_2)^3} \frac{n^3}{\epsilon^2} \log^2\left(\frac{n\tau}{\epsilon}\right) = O\left(n\epsilon^{-2}\log^2(n/\epsilon)\right)$. This recovers the (informal) Theorem 1. Notice that the standard power method would require n^2 samples to compute $\boldsymbol{u}_1$. We also note that even if $\sigma_1 - \sigma_2$ is sublinear in n, as long as $\frac{\sigma_1}{(\sigma_1-\sigma_2)^3} = o\left(\frac{1}{n\log^2 n}\right)$, there are still sample complexity gains. The proof of Theorem 2 is based on the following key lemma.

Lemma 3 *If Algorithm 1 is run for $L = O\left(\frac{\sigma_1}{\sigma_1-\sigma_2}\log\left(\frac{n\tau}{\epsilon}\right)\right)$ iterations with a budget of $B = \frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$, for some fixed parameters τ and $\epsilon < 1/2$, the algorithm satisfies*

1. $\left\|A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}\right\| \le \frac{\epsilon}{5}(\sigma_1 - \sigma_2)$
2. $\left|(A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1})^\top \boldsymbol{u}_1\right| \le \frac{1}{5\tau\sqrt{n}}(\sigma_1 - \sigma_2)$

for all $\ell \in [L]$, with probability at least $1 - n^{-\Omega(1)}$.

Establishing conditions 1 and 2 requires a careful bounding of the error of the estimator $A^{(\ell)}\boldsymbol{x}_{\ell-1}$ using Bernstein's inequality. The proof is provided in Appendix C. By letting $\boldsymbol{n}_\ell = A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}$, and viewing $A^{(\ell)}\boldsymbol{x}_{\ell-1} = A\boldsymbol{x}_{\ell-1} + \boldsymbol{n}_\ell$, we are in the exact setting considered in the Noisy Power Method (Hardt and Price, 2014), which implies the following lemma.

Lemma 4 *If conditions 1 and 2 hold for some τ and $\epsilon < 1/2$ for all iterations in Algorithm 1, the output $\boldsymbol{x}_L$ after $L = O\left(\frac{\sigma_1}{\sigma_1-\sigma_2}\log\left(\frac{n\tau}{\epsilon}\right)\right)$ iterations satisfies $\left\|\boldsymbol{u}_1 - \langle \boldsymbol{u}_1, \boldsymbol{x}_L\rangle \boldsymbol{x}_L\right\| \le \epsilon$, with probability at least $1 - \tau^{-\Omega(1)} - e^{-\Omega(n)}$.*

Proof Starting with a randomly chosen normalized vector $\boldsymbol{x}_0 \in \mathbb{R}^n$, the APM iteratively updates the current eigenvector estimate $\boldsymbol{x}_{\ell-1}$ at ℓth iteration ($\ell \in [L]$), by computing $A^{(\ell)}\boldsymbol{x}_{\ell-1}$ and normalizing it. Defining $\boldsymbol{n}_\ell = A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}$, one can view the APM as the noisy power method proposed by Hardt and Price (2014), where the product $A\boldsymbol{x}_{\ell-1}$ is computed with an additive perturbation $\boldsymbol{n}_\ell$. Theorem 2.4 followed by Corollary 1.1 in Hardt and Price (2014) completes the proof. ■

Proof of Theorem 2 By the union bound, Lemmas 3 and 4 imply that equation (4) holds with probability at least $1 - n^{-\Omega(1)} - \tau^{-\Omega(1)} - e^{-\Omega(n)} = 1 - n^{-\Omega(1)} - \tau^{-\Omega(1)}$ with a budget $B = \frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$. ■

3. Adaptive Power Method with Variance Minimization

As stated in the previous section, Algorithm 1 produces an unbiased estimate of $A\boldsymbol{x}_{\ell-1}$ for all iteration steps $\ell \in [L]$. Hence, one way to improve the algorithm's performance is to reduce the estimator variance $\mathrm{Var}(A^{(\ell)}\boldsymbol{x}_{\ell-1})$. Intuitively, this would also reduce the final mean square error (MSE) $\mathbb{E}\|\boldsymbol{u}_1 - \boldsymbol{x}_L\|^2$, since $\boldsymbol{u}_1 \approx A^{(\ell)}\boldsymbol{x}_{\ell-1}/\|A^{(\ell)}\boldsymbol{x}_{\ell-1}\|$ and $A\boldsymbol{x}_{\ell-1} \approx \sigma_1\boldsymbol{u}_1$, and thus $\mathbb{E}\|\boldsymbol{u}_1 - \boldsymbol{x}_L\|^2 \approx$

Algorithm 2 ADAPTIVE POWER METHOD WITH MINIMUM VARIANCE ($\text{APM}_{\min \text{Var}}$)

Input: Symmetric matrix $A \in \{0,1\}^{n\times n}$, budget B, number of iterations L
Output: Approximate top eigenvector $\boldsymbol{x}_L \in \mathbb{R}^n$

1: Draw a random vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ with $\|\boldsymbol{x}_0\| = 1$
2: **for** $\ell = 1, \ldots, L$ **do**
3: Draw $I_{ij}^{(\ell)} \sim \text{Bern}(p_{ij}^{(\ell)})\ \forall (i,j) \in [n] \times [n]$ independently, according to equation (6), except $p_{ij}^{(1)} \leftarrow \frac{B}{nL} \cdot x_{0j}^2$
4: $[A^{(\ell)}]_{ij} \leftarrow \frac{a_{ij}}{p_j} I_{ij}^{(\ell)}$
5: $\boldsymbol{y}_\ell \leftarrow A^{(\ell)} \boldsymbol{x}_{\ell-1}$
6: $\boldsymbol{x}_\ell \leftarrow \frac{\boldsymbol{y}_\ell}{\|\boldsymbol{y}_\ell\|}$
7: **end for**

$\frac{1}{\sigma_1}\mathbb{E}\,\|A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}\|^2$. In order to reduce the estimator variance $\text{Var}(A^{(\ell)}\boldsymbol{x}_{\ell-1})$, instead of specifying one sampling probability p_j for all entries in the jth column, we consider a sampling probability $p_{ij}^{(\ell)}$ that can differ for each row i. Dropping the superscript (ℓ) for conciseness, we have

$$[A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i = \sum_{j=1}^n \frac{a_{ij}x_{(\ell-1)j}}{p_{ij}} I_{ij}; \quad \mathbb{E}\left[[A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i\right] = \sum_{j=1}^n a_{ij}x_{(\ell-1)j};$$
$$\text{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i) = \sum_{j=1}^n \frac{a_{ij}^2 x_{(\ell-1)j}^2}{p_{ij}^2}\,\text{Var}\left(I_{ij}\right) = \sum_{j=1}^n \frac{a_{ij}^2 x_{(\ell-1)j}^2}{p_{ij}}(1-p_{ij}); \tag{5}$$

where we used the fact that $\text{Var}(I_{ij}) = p_{ij}(1-p_{ij})$. Now the question boils down to choosing p_{ij} for all (i,j) so as to minimize the total variance given the fixed expected budget B. In Appendix D, we show that this constrained optimization problem can be solved by setting

$$p_{ij}^{(\ell)} = \frac{x_{(\ell-1)j}^2}{\beta^{(\ell)}} \cdot \left(\frac{-\|[A\boldsymbol{x}_{\ell-1}]_i\|^2 + \sqrt{\|[A\boldsymbol{x}_{\ell-1}]_i\|^4 + 4\beta^{(\ell)}\|A_{i:}\|^2}}{2}\right). \tag{6}$$

Here $\|A_{i:}\|^2 := \sum_j a_{ij}^2$ denotes the squared ℓ_2 norm of ith row of A, and $\beta^{(\ell)}$ is a normalization constant chosen so that the expected budget per iteration is B/L; i.e., $\sum_{i=1}^n \sum_{j=1}^n (p_{ij}^{(\ell)})^2 = B/L$.

Variance adaptivity across rows: In Appendix D, we also show that, given the choice of p_{ij} in (6), the expected number of samples in row i is given by $\text{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)/\beta^{(\ell)}$. This implies that if the variance of the ith coordinate of $A^{(\ell)}\boldsymbol{x}_{\ell-1}$ is large, the algorithm samples entries in the ith row of A with higher probability than in other rows. This agrees with the intuition that if the variance from row i is large, one needs to sample more entries in $A_{i:}$ to make the estimation accurate. We also observe that this connects to the idea of Stratified Monte Carlo Sampling introduced in Carpentier et al. (2015) that samples entries proportionally to the stratum (e.g., the level of variability).

Estimating $\|A_{i:}\|$ and $\|[A\boldsymbol{x}_{\ell-1}]_i\|$: The optimal choice of p_{ij} in (6) requires knowledge of $\|A_{i:}\|$ and $\|[A\boldsymbol{x}_{\ell-1}]_i\|$, but these quantities cannot be computed without full knowledge of A. Hence, we

replace these quantities with their estimates based on the samples obtained in the $(\ell-1)$th iteration:

$$\widehat{\|A_{i:}\|^2} = \frac{n}{\sum_j I_{ij}^{(\ell-1)}} \sum_{j: I_{ij}^{(\ell-1)}=1} a_{ij}^2 \tag{7}$$

$$\widehat{\|[A\boldsymbol{x}_{\ell-1}]_i\|} = \|[A^{(\ell-1)}\boldsymbol{x}_{\ell-1}]_i\|. \tag{8}$$

By replacing the true quantities $\|A_{i:}\|$ and $\|[A\boldsymbol{x}_{\ell-1}]_i\|$ in (6) with these estimates, we can approximately compute the sampling probabilities $p_{ij}^{(\ell)}$. The normalizing constant $\beta^{(\ell)}$ can then be chosen so that the expected number of samples is satisfied; i.e., $\sum_{i=1}^n \sum_{j=1}^n (p_{ij}^{(\ell)})^2 = B/L$.

Our adaptive scheme with the variance minimization, which we refer to as $\text{APM}_{\text{min Var}}$, is summarized in Algorithm 2. As shown in the next section, $\text{APM}_{\text{min Var}}$ outperforms the first APM for a variety of choices of the input budget B. In particular, we empirically observe that this non-uniform allocation produces lower errors on the top-k entries of the principal eigenvector.

4. Empirical Results

In this section, we corroborate our theoretical claims with numerical experiments to validate the practicality and performance of the two Adaptive Power Methods in a non-asymptotic setting. To this end, we consider the problem of eigenvector centrality computation on real-world graphs, taking the adjacency matrix A as an input. Notice that the centrality of a node corresponds to its corresponding entry in the principal eigenvector.

Dataset: We use the two benchmark datasets from the Open Graph Benchmark (OGB) (Hu et al., 2020), which includes a collection of realistic, large-scale, and diverse benchmark datasets for machine learning on graphs:

1. ogbl-ddi dataset, representing drug-drug interactions
2. ogbl-biokg dataset, representing drug-drug, drug-protein, and protein-protein interactions

In particular, ogbl-ddi dataset is a dense graph, consisting of $4,267$ nodes and $2,135,822$ edges. The graph is homogeneous, unweighted, and undirected, where each node represents an FDA-approved or experimental drug; an edge represents interactions between drugs. These interactions are interpreted as the difference between the joint effect of taking the two drugs together and the expected effect when drugs act independently of each other. Hence evaluating interactive nodes through the computation of the eigenvector centrality and ranking them can potentially play a role in drug development.

The second dataset, called ogbl-biokg dataset, is a Knowledge Graph, created from a biomedical data. The graph includes 5 types of entities: diseases, proteins, drugs, side effects, and protein functions. All relations are modeled as directed edges. In such case, we consider the eigenvector centrality for $A^\intercal A$ or $AA^\intercal$ instead in order to make the matrix symmetric. Then the top eigenvector designates the importance of nodes in terms of in-edges and out-edges in the graph respectively. Note, however, that sampling entries of $A^\intercal A$ or $AA^\intercal$ do not correspond to directly querying pairwise relationships between nodes in the graph. In addition, since the dataset involves heterogeneous interactions in a variety of scales, we construct a reduced homogeneous graph, consisting of $1,500$ nodes and $23,131$ edges and focusing on interactions between drugs and proteins. In contrast to the first graph, we highlight that the constructed graph is sparse. As in the previous case, analyzing the interactions among entities allows us to get better insights into predictions related to human biology.

Performance metrics: We measure the effectiveness of the algorithms with three performance metrics:

1. ℓ_2 error between the true top eigenvector and the top eigenvector estimate, along the direction of true eigenvector: Denoting $\boldsymbol{x}_L$ be the final estimate, the error is given by $\left\|\boldsymbol{u}_1 - \langle \boldsymbol{u}_1, \boldsymbol{x}_L \rangle \boldsymbol{x}_L\right\|$.

2. ℓ_2 error on the top-k entries: In order to measure how well the algorithm can produce the approximate top-k entries of the principal eigenvector, we next focus on the same ℓ_2 error on the top-k entries. Define $\mathcal{I}_k := \{i \in [n] : \psi(u_{1i}) \in [k]\}$, where ψ is any ranking scheme that returns a set of k indices in $[n]$, and $[k]$ denotes the (unordered) set of the first k indices. The top-k ℓ_2 error is then given by $\|[\boldsymbol{u}_1]_{\mathcal{I}_k} - \left\langle [\boldsymbol{u}_1]_{\mathcal{I}_k}, [\boldsymbol{x}_L]_{\mathcal{I}_k} \right\rangle [\boldsymbol{x}_L]_{\mathcal{I}_k}\|$.

3. Jaccard similarity (JS): For two sets A and B, the Jaccard similarity between them, $\mathrm{JS}(A, B)$, is defined as the size of their intersection divided by the size of their union. Note that it is bounded between 0 and 1; in addition, $\mathrm{JS}(A, B) = 0$ if and only if $A \cap B = \emptyset$ and $\mathrm{JS}(A, B) = 1$ if and only if $A = B$. We refer the interested reader to Leskovec et al. (2020), Chapter 3, for a detailed review of the topic. Particularly, we focus on the Jaccard similarity on the top-k entries, given by $\mathrm{JS}(\mathcal{I}_k, \hat{\mathcal{I}}_k) = |\mathcal{I}_k \cap \hat{\mathcal{I}}_k| / |\mathcal{I}_k \cup \hat{\mathcal{I}}_k|$. Here $\hat{\mathcal{I}}_k := \{i \in [n] : \psi(x_{Li}) \in [k]\}$ denotes the top-k entries of the final eigenvector estimate $\boldsymbol{x}_L$.

Evaluation methods: With the aforementioned performance metrics, we evaluate the two proposed schemes with the two baselines described below.

1. APM (Algorithm 1)

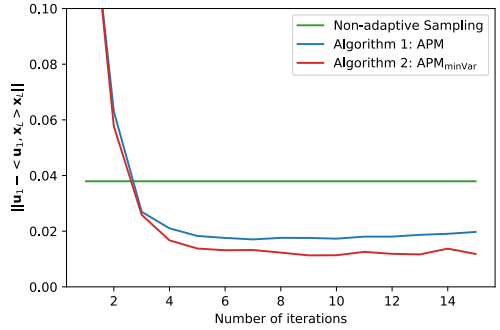
2. $\mathrm{APM}_{\min \mathrm{Var}}$ (Algorithm 2)

3. Non-adaptive estimation algorithm that samples B entries *uniformly at random* in the adjacency matrix and computes the top eigenvector with sampled entries

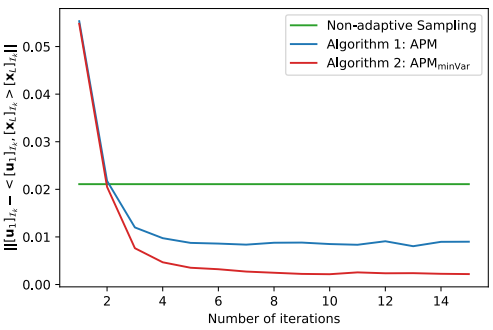
4. Power method exploiting the full adjacency matrix

While implementing the first two algorithms, we include sample reuse procedures described in Appendix A. Note that the third method serves as a baseline and the last one can be understood as the oracle method. To check whether or not our theoretical convergence guarantee for fixed budget B holds (as L increases) in practice, we first compare our algorithms with the non-adaptive estimation algorithm. Specifically, we evaluate the ℓ_2 error with respect to the number of iterations L. We next measure the top-k ℓ_2 error of the first three methods for fixed k ($k = 100$ or $k = 500$), with respect to L. We then evaluate ℓ_2 error on the top-k elements ($k = 1, 2, \ldots, n$) of the four algorithms. To this end, we fix the number of iterations of the algorithms to be $L = 10$. For the computation of the Jaccard similarity, we run the four algorithms for $L = 10$ iterations, to evaluate $\mathrm{JS}_k(\mathcal{I}_k, \hat{\mathcal{I}}_k)$ over $k \in [n]$. For these two evaluations, we set the standard power method as another baseline for the performance using the full adjacency matrix.

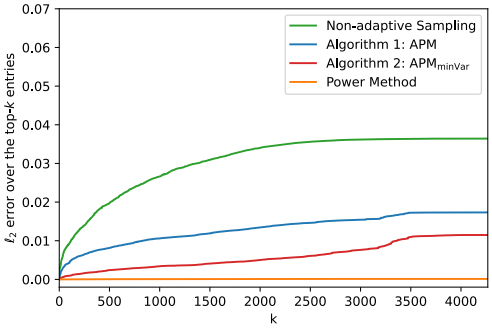
Implementation details: While conducting the first two experiments for a variety range of budget B, we face stability issues if the budget is low and the number of iterations run is large. In this case, the expected number of sampled entries per row per iteration $\frac{B}{nL}$ becomes very low, yielding a large variance in the performance. As a heuristic way to stabilize the performance, we add few more



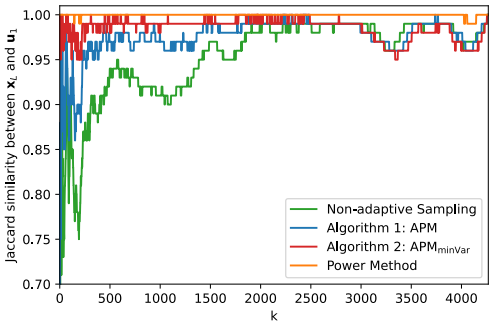
(*a*) ℓ_2 error



(*b*) top-500 ℓ_2 error of $\boldsymbol{u}_1$



(*c*) ℓ_2 error over the top-k entries

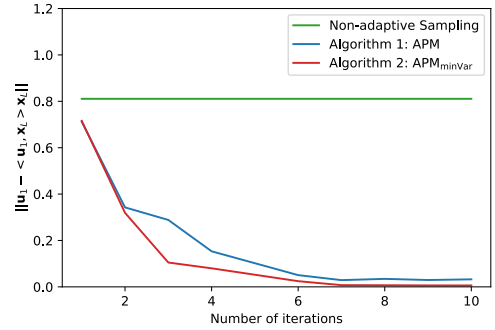


(*d*) Jaccard similarity over the top-k entries

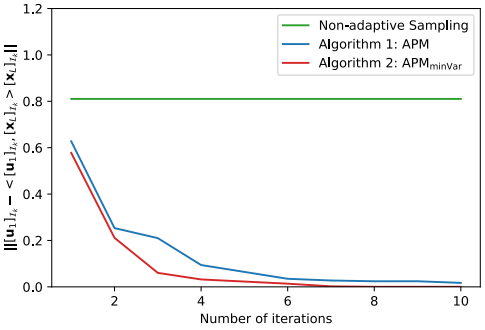
Figure 1: All plots are the results with observing 50% of the entire ogbl-ddi dataset. (a) shows the ℓ_2 error between the top eigenvector estimate and the true one. (b) shows the same type of ℓ_2 error, but on the top-500 entries of the true top eigenvector. (c) shows the ℓ_2 error over the top-k entries. (d) shows the Jaccard similarity over the top-k entries.

power iterations (without further sampling) at the end of the proposed algorithms. When the budget is large enough, the variance of the performance is small, and additional power iterations can lead to performance degradation. Hence in this case, we remove additional power iteration steps.

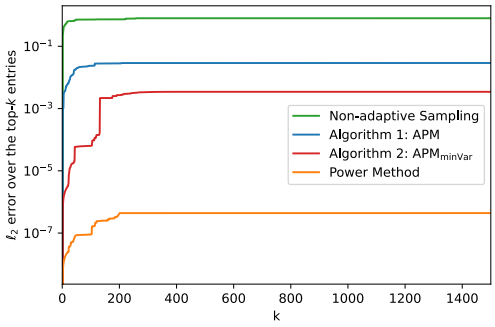
Results and analysis: In Figure 1, we plot the error reductions afforded by adaptivity for the case of sampling $B = 0.5n^2$ (i.e., 50% budget) entries of the ogbl-ddi dataset. Specifically, figure 1(*a*) represents the ℓ_2 error between the eigenvector estimate and the true one, where there exists 54.36% error reduction of APM over the non-adaptive estimation algorithm. In addition, we observe a further significant error decrease for $\text{APM}_{\text{min Var}}$ over the non-adaptive scheme and over APM. In particular, the improvement was projected to be 70.06% and 40.05% respectively. Figure 1(*b*) represents the ℓ_2 error on the top-500 entries of the true principal eigenvector. With the same budget constraint, we observe much larger gains for the two proposed schemes over the non-adaptive one. Specifically, the error decrease for APM and $\text{APM}_{\text{min Var}}$ over the non-adaptive scheme are increased to be 61.83% and 89.99% respectively. We also observe that the error decrease of 75.63%, for $\text{APM}_{\text{min Var}}$ over APM. Note that the idea of non-uniform allocation of samples across the



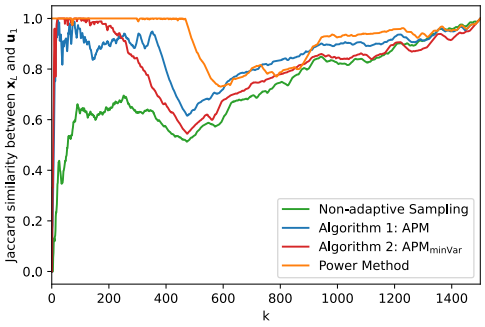
(*a*) ℓ_2 error



(*b*) top-100 ℓ_2 error of $\boldsymbol{u}_1$



(*c*) ℓ_2 error over the top-k entries



(*d*) Jaccard similarity over the top-k entries

Figure 2: All plots are the results with observing 5% of the entire ogbl-biokg dataset. (a) shows the ℓ_2 error between the top eigenvector estimate and the true one. (b) shows the top-100 ℓ_2 error. (c) shows the ℓ_2 error over the top-k entries. (d) shows the Jaccard similarity over the top-k entries.

different rows of A in $\text{APM}_{\text{min Var}}$ plays a key role in reducing the top-k error significantly. From the two plots, we additionally corroborate our theoretical findings from the two plots that the errors converge as the number of iterations increases. Figure 1(*c*) and figure 1(*d*) represent ℓ_2 error and the Jaccard similarity over the top-k entries. In particular, we use a simple moving average method with a window size of 4 for plotting the Jaccard similarity over top-k entries.

In Figure 2, we plot the error decrease afforded by adaptivity with sampling $B = 0.05n^2$ (i.e., 5% budget) entries of the ogbl-biokg dataset. For this sparse graph dataset, we observe larger performance improvements. In particular, the decrease in errors for APM and $\text{APM}_{\text{min Var}}$ over the non-adaptive scheme are 97.03% and 99.39% respectively. Notice that 100% decrease indicates the acquirement of zero error. In addition, the improvement of $\text{APM}_{\text{min Var}}$ over APM is 79.6%. For the case of the error over the top-100 entries, we highlight that there exist significant error reductions of 98.07%, 99.99%, and 99.26% respectively. Referring to the log-scale error plot in figure 2(*c*), the performance of $\text{APM}_{\text{min Var}}$ is close to that of the standard power method that exploits the full adjacency matrix. Compared to the results for the ogbl-ddi dataset, we empirically observe that these improvements are noticeable when the budget constraint is low. As the budget increases, both

schemes work very well and their performance behaves similarly. Furthermore, recall that there is a procedure of estimating $\|A_{i:}\|^2$ in $\text{APM}_{\min \text{Var}}$. We observe that the row norm estimation still works well even if the budget is low in the sparse graph. This is in contrast to the dense graph case that there needs to be a moderate amount of budget to guarantee the estimation of $\|A_{i:}\|^2$, in order for $\text{APM}_{\min \text{Var}}$ to perform reasonably well.

Remark 5 (Comparison between APM vs. Non-adaptive Scheme) *Although we demonstrate the superiority of* APM *compared to the non-adaptive estimation algorithm through experiments, we note that the non-adaptive scheme requires a budget of* $B = \Omega\left(\max\left\{\frac{(1+\epsilon)^2}{\epsilon^2} \cdot \frac{n^3}{(\sigma_1-\sigma_2)^2}, n\log n\right\}\right)$ *samples in theory, indicating a better dependency on* n *(by a* $\log n$ *factor). We provide a proof sketch of the sample complexity of the non-adaptive scheme in Appendix E.*

5. Related Work

To the best of our knowledge, the most relevant works on eigenvector estimation based on incomplete knowledge of the matrix are based on query models that return vectors. For example, given a distribution $\mathcal{D}$ with covariance matrix $\Sigma = \mathbb{E}_{\boldsymbol{a}\sim\mathcal{D}}[\boldsymbol{a}\boldsymbol{a}^\top] \in \mathbb{R}^{n\times n}$, Garber et al. (2016) propose another variant of the power method for estimating the top eigenvector when the learner has access to an oracle that returns independent samples $\boldsymbol{a} \in \mathbb{R}^n$ from $\mathcal{D}$. In particular, they show that the sample complexity required to obtain an ϵ-approximate top eigenvector is $O(n\epsilon^{-1}\log(n/\epsilon))$. There are four key distinctions relative to our case. First, the learner in Garber et al. (2016) samples a vector $\boldsymbol{a} \in \mathbb{R}^n$ (which is of dimension n) from $\mathcal{D}$, and hence one can view the effective sample complexity to be $O(n^2\epsilon^{-1}\log(n/\epsilon))$; in our work, we show the sample complexity of $O(n\epsilon^{-2}\log^2(n/\epsilon))$. Second, the way of sampling is different: we assume that the underlying matrix A is fixed and the learner performs an entry-wise observation and acquires the exact value of the entry of A. Third, the matrix has to be symmetric positive definite (pd) in Garber et al. (2016), whereas the matrix of interest A in our case is not necessarily pd, and can also be asymmetric. Lastly, instead of ℓ_2 norm as the performance metric for representing errors, the authors use a norm with symmetric positive definite matrix; particularly, their goal is to obtain $\boldsymbol{x}$ such that $\boldsymbol{x}\Sigma\boldsymbol{x} \geq (1-\epsilon)\sigma_1(\Sigma)$. Another work by Simchowitz et al. (2017) develops a lower bound for the problem of computing the top eigenvector of a symmetric $n \times n$ matrix A. In particular, they consider a query model in which a learner receives noiseless responses of the form $\boldsymbol{x} = A\boldsymbol{y}$, for some adaptively drawn vector $\boldsymbol{y}$, for T queries. It is shown that $T = \Omega(\log n)$ queries, and hence $\Omega(n\log n)$ effective samples are necessary for any adaptive, randomized algorithm that finds a normalized vector $\hat{\boldsymbol{u}}_1$ satisfying $\hat{\boldsymbol{u}}_1^\top A\hat{\boldsymbol{u}}_1 \geq (1-\epsilon)\|A\|_{op}$ for some small $\epsilon > 0$. Such a model allows the sampling of a full column of A (by choosing y to be one-hot), but individual entries cannot be sampled as in our setting.

For the case of performing entrywise observations on data, motivated by applications in crowdsourcing and computational biology, Kamath et al. (2020) consider the problem of identifying the k largest entries of the leading left singular vector $\boldsymbol{u}_1$ in a rank-one matrix. An adaptive spectral estimation algorithm based on multi-armed bandits is proposed to find the k largest entries of $\boldsymbol{u}_1$, by roughly observing $O(n\log n)$ entries of a rank-one matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. In addition, Ruggeri and De Bacco (2019, 2020) develop a sampling method that selects the best node from a set of non-sampled nodes in an online fashion, to estimate eigenvector centrality. From the perspective of graph signal processing, Shomorony and Avestimehr (2014) also develop efficient online algorithms

for finding the smallest subset of nodes for a given cut-off frequency, where it designates the highest frequency component of a given signal that guarantees recovery from the samples.

There have also been works on estimating dominant eigenvectors from noisy observations of A using variants of the power method. This captures various forms of noise, including missing entries, sampling errors, approximation error, privacy constraints, or adversarial attacks (Mitliagkas et al., 2013; Hardt and Roth, 2013; Hardt and Price, 2014; Musco and Musco, 2015; Liu et al., 2015; Balcan et al., 2016; Xu and Li, 2019, 2020, 2021, 2022). In particular, the noisy power method by Hardt and Price (2014) is a meta algorithm that obtains the dominant eigenspace with high probability, under noise-corrupted matrix-vector multiplications. As with the noiseless setting, the convergence rate of the noisy power method is inversely proportional to and largely depends on the consecutive spectral gap $\sigma_k - \sigma_{k+1}$. The key distinction relative to our setting is that they introduce an additive perturbation noise but have access to the full matrix A, whereas our algorithms focus on adaptively sampling entries of A to improve the eigenvector estimation accuracy.

In terms of adaptive algorithms for partially observing a matrix, methods have been proposed (Li et al., 2013; Cohen et al., 2015; Musco and Musco, 2017) to adaptively query entries of a matrix based on the importance of rows. Rather than trying to estimate eigenvectors, the goal in these works is to approximate the leverage scores of the matrix. Nevertheless, their algorithms are similar to our adaptive scheme in the sense that they sample parts of the matrix based on current estimates (i.e., leverage score sampling using leverage score estimates) and refine the estimates iteratively.

6. Discussion and Conclusion

We studied the problem of eigenvector estimation when the matrix A is unknown but a limited number of its entries can be adaptively observed. Our new algorithm, Adaptive Power Method, can adaptively choose entries from A to observe based on the current eigenvector estimate. We also provided a theoretical convergence guarantee for estimating the dominant eigenvector with small error, which implies that only a small fraction of the entries of A needs to be sampled to ensure estimation accuracy. On two real-world benchmark datasets, we demonstrated the performance improvements of the two proposed schemes over the non-adaptive schemes via numerical simulations.

Several extensions of our results are possible. First, a generalization to the estimation of the top-k eigenvectors of A can be readily obtained since the Noisy Power Method (Hardt and Price, 2014) generalizes for the top-k eigenvectors. Another direction for future work is to establish a convergence guarantee for $\text{APM}_{\min \text{Var}}$. We believe an analysis similar to that of Algorithm 1 should be possible for this extension based on the following steps, although it would likely be much longer. First, we assume that the values of $\|[Ax]_i\|$ and $\|A_{i:}\|^2$ in equation (6) are known exactly and use a similar analysis as that in Appendix C, based on Bernstein's inequality, to bound the error of the estimator for Ax. Second, we would obtain high probability bounds on our estimates for $\|[Ax]_i\|$ and $\|A_{i:}\|^2$, and analyze the impact of only knowing $\|[Ax]_i\|$ and $\|A_{i:}\|^2$ within some bounded error. All these steps should be feasible, although cumbersome.

Based on the empirical observation that $\text{APM}_{\min \text{Var}}$ achieves better estimation accuracy on the largest entries of the eigenvectors, another interesting direction is to study the problem of only accurately estimating the top-k entries of the top eigenvector.

Lastly, we point out that it may be possible to reduce the sample complexity beyond the $(n\epsilon^{-2}\log^2(n/\epsilon))$ that we achieved. In particular, There are several possible ways to improve the factor $\frac{n^3\sigma_1}{(\sigma_1-\sigma_2)^3}$ in the sample complexity. This factor comes from (1) using the convergence

guarantee from Hardt and Price (2014) and (2) using Bernstein's inequality to bound the norm of the perturbation and the inner product between the perturbation and the top eigenvector. This analysis is presented in step (vi) in Proposition 8. In order to improve this factor, we would need to improve (1) or (2). Even if we stick to the same algorithm, one should be able to obtain better convergence guarantee results for the algorithm in Appendix A, which reuses past samples. Furthermore, there have been recent results that improve the convergence guarantees of noisy power methods by improving from $\sigma_1 - \sigma_2$ to $O(\sigma_1 - \sigma_q)$ in (Balcan et al., 2016) and to $\tilde{O}(\sqrt{\sigma_1 - \sigma_q})$ in (Xu and Li, 2022), where $q \geq 2$. We believe we can adapt those results in order to improve our query complexity. We may also be able to use other iterative algorithms like the Lanczos algorithm to improve the query complexity.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. The work of I.S. was supported in part by the National Science Foundation (NSF) under Grant CCF-2046991. H.Z. would like to thank the support from a Facebook Research Award and Amazon AWS Cloud Credit.

References

Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An improved gap-dependency analysis of the noisy power method. In *Conference on Learning Theory*, pages 284–309. PMLR, 2016.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

Alexandra Carpentier, Remi Munos, and András Antos. Adaptive strategy for stratified monte carlo sampling. *J. Mach. Learn. Res.*, 16:2231–2271, 2015.

Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190, 2015.

Steven J Cook, Travis A Jarrell, Christopher A Brittin, Yi Wang, Adam E Bloniarz, Maksim A Yakovlev, Ken CQ Nguyen, Leo T-H Tang, Emily A Bayer, Janet S Duerr, et al. Whole-animal connectomes of both Caenorhabditis elegans sexes. *Nature*, 571(7763):63–71, 2019.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Maurice De Kunder. The size of the indexed World Wide Web, 2022. URL https://www.worldwidewebsize.com/.

Kristine De Valck, Gerrit H Van Bruggen, and Berend Wierenga. Virtual communities: A marketing perspective. *Decision support systems*, 47(3):185–203, 2009.

Dan Garber, Elad Hazan, Chi Jin, Cameron Musco, Praneeth Netrapalli, Aaron Sidford, et al. Faster eigenvector computation via shift-and-invert preconditioning. In *International Conference on Machine Learning*, pages 2626–2634. PMLR, 2016.

Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016.

Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

Moritz Hardt and Aaron Roth. Beyond worst-case analysis in private singular vector computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 331–340, 2013.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs, 2020. URL https://ogb.stanford.edu/.

Björn H Junker and Falk Schreiber. *Analysis of biological networks*. John Wiley & Sons, 2011.

Govinda Kamath, Tavor Baharav, and Ilan Shomorony. Adaptive learning of rank-one models for efficient pairwise sequence alignment. *Advances in Neural Information Processing Systems*, 33: 7513–7525, 2020.

Christine Kiss and Martin Bichler. Identification of influencers—measuring influence in customer networks. *Decision Support Systems*, 46(1):233–253, 2008.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

Mu Li, Gary L Miller, and Richard Peng. Iterative row sampling. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 127–136. IEEE, 2013.

Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 171–178, 2015.

Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming PCA. *Advances in neural information processing systems*, 26, 2013.

Cameron Musco and Christopher Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. *Advances in neural information processing systems*, 28, 2015.

Cameron Musco and Christopher Musco. Recursive sampling for the Nyström method. *Advances in neural information processing systems*, 30, 2017.

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.

Beresford N Parlett. *The symmetric eigenvalue problem*. SIAM, 1998.

Nicolò Ruggeri and Caterina De Bacco. Sampling on networks: estimating eigenvector centrality on incomplete networks. In *International Conference on Complex Networks and Their Applications*, pages 90–101. Springer, 2019.

Nicolò Ruggeri and Caterina De Bacco. Sampling on networks: estimating spectral centrality measures and their impact in evaluating other relevant network measures. *Applied Network Science*, 5(1):1–29, 2020.

Ilan Shomorony and Salman Avestimehr. Sampling large data on graphs. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 933–936. IEEE, 2014.

Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. On the gap between strict-saddles and true convexity: An $\Omega(\log d)$ lower bound for eigenvector approximation. *arXiv preprint arXiv:1704.04548*, 2017.

Statista. Number of social media users worldwide from 2018 to 2022, with forecasts from 2023 to 2027, 2022. URL https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.

Stefan Wuchty and Peter F Stadler. Centers of complex networks. *Journal of theoretical biology*, 223(1):45–53, 2003.

Zhiqiang Xu and Ping Li. Towards practical alternating least-squares for CCA. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhiqiang Xu and Ping Li. A practical Riemannian algorithm for computing dominant generalized Eigenspace. In *Conference on Uncertainty in Artificial Intelligence*, pages 819–828. PMLR, 2020.

Zhiqiang Xu and Ping Li. On the Riemannian Search for Eigenvector Computation. *J. Mach. Learn. Res.*, 22:249–1, 2021.

Zhiqiang Xu and Ping Li. Faster Noisy Power Method. In *International Conference on Algorithmic Learning Theory*, pages 1138–1164. PMLR, 2022.

Appendix A. Description of Adaptive Power Method with Sample Reuse

Algorithm 1 ADAPTIVE POWER METHOD (APM) (WITH SAMPLE REUSE)

Input: Symmetric matrix $A \in \{0,1\}^{n\times n}$, budget B, number of iterations L, threshold $c > 0$
Output: Approximate top eigenvector $\boldsymbol{x}_L \in \mathbb{R}^n$

1: Draw a random vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ with $\|\boldsymbol{x}_0\| = 1$
2: **for** $\ell = 1, \ldots, L$ **do**
3: Draw $I_{ij}^{(\ell)} \sim \mathrm{Bern}(p_{ij}^{(\ell)})\ \forall (i,j) \in [n]\times[n]$ independently, where

$$p_{ij}^{(\ell)} = \begin{cases} 1, & \text{if } I_{ij}^{(\ell-1)} = 1; \\ \min\left[\frac{B}{nL}\cdot \max\left(\frac{x_{(\ell-1)j}^2}{M}, c^2\right), 1\right], & \text{else.} \end{cases}$$

4: $[A^{(\ell)}]_{ij} \leftarrow \frac{a_{ij}}{p_{ij}^{(\ell)}} I_{ij}^{(\ell)}$
5: $\boldsymbol{y}_\ell \leftarrow A^{(\ell)}\boldsymbol{x}_{\ell-1}$
6: $\boldsymbol{x}_\ell \leftarrow \frac{\boldsymbol{y}_\ell}{\|\boldsymbol{y}_\ell\|}$
7: $M \leftarrow \sum_{j: I_{ij}^{(\ell)}=0} x_{\ell j}^2$
8: **end for**

Appendix B. High Probability Guarantee for Number of Samples Used

In this section, we show that APM uses at most $(1 + c^2 n + o(1))B$ samples with high probability. Notice that if $\sigma_1 - \sigma_2 = \Theta(n)$ and we set $c = \frac{\sqrt{\alpha}(\sigma_1-\sigma_2)}{n^{3/2}} = O(\sqrt{\frac{\alpha}{n}})$ for some small $\alpha > 0$, then the number of observation becomes at most $(1 + \alpha + o(1))B$ with high probability.

Lemma 6 *If we run Algorithm 1 for L steps, with an expected budget of $B = \omega\left(n\sqrt{L}\log\left(1/\delta\right)\right)$, then the algorithm samples at most $(1 + c^2 n + o(1))B$ entries of a $n \times n$ matrix A in total, with probability at least $1 - \delta$.*

Proof Since $I_{ij}^{(\ell)} \sim \mathrm{Bern}(p_j^{(\ell)})$, $\forall(\ell, i, j) \in [L]\times[n]\times[n]$, where

$$p_j^{(\ell)} = \min\left[\frac{B}{nL}\cdot\max\left(\frac{x_{(\ell-1)j}^2}{M}, c^2\right), 1\right],$$

one can readily see that the total number of sampled entries is $\sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n} I_{ij}^{(\ell)}$. By Hoeffding's inequality, we observe that

$$\Pr\left(\left|\sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(I_{ij}^{(\ell)} - \mathbb{E}\left[I_{ij}^{(\ell)}\right]\right)\right| \geq \eta\right) \leq 2\exp\left(-\frac{2\eta^2}{n^2 L}\right).$$

This implies that if we set $\eta := n\sqrt{\frac{L}{2}}\log(\frac{2}{\delta})$, APM samples at most

$$\sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\left[I_{ij}^{(\ell)}\right] + n\sqrt{\frac{L}{2}}\log\left(\frac{2}{\delta}\right) \tag{9}$$

entries, with probability at least $1-\delta$.

What is remaining is to bound the expected number of sampled entries. Referring to equation (3) in Section 2, we see that

$$\sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\left[I_{ij}^{(\ell)}\right] \le \sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n}p_j^{(\ell)} = (nL)\cdot\frac{B}{nL}\left(1+c^2n\right) = \left(1+c^2n\right)\cdot B. \tag{10}$$

From equations (9) and (10), we see that with probability at least $1-\delta$, the number of entries that APM exploits is:

$$\begin{aligned}&\sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\left[I_{ij}^{(\ell)}\right] + n\sqrt{\frac{L}{2}}\log\left(\frac{2}{\delta}\right)\\ &\le \left(1+c^2n\right)\cdot B + n\sqrt{\frac{L}{2}}\log\left(\frac{2}{\delta}\right).\end{aligned} \tag{11}$$

Hence if $B = \omega\left(n\sqrt{L}\log\left(1/\delta\right)\right)$, then the algorithm samples at most $(1+c^2n+o(1))B$ entries of a $n\times n$ matrix A in total, with probability at least $1-\delta$. ■

Notice that when $L = O\left(\frac{\sigma_1}{\sigma_1-\sigma_2}\log\left(\frac{n\tau}{\epsilon}\right)\right)$ and $B = \frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$, as in the statement of Theorem 2, and if $\sigma_1-\sigma_2 = \Theta(n)$, we indeed have $B = \omega\left(n\sqrt{L}\log\left(1/\delta\right)\right)$, and the lemma above implies that we sample at most $(1+c^2n+o(1))B$ entries with probability $1-\delta$.

Appendix C. Proof of Lemma 3

We show that conditions 1 and 2 hold via Bernstein's inequality. We will use the following statement of Bernstein's inequality: if $W_1,\dots,W_n$ are independent zero-mean random variables such that $|W_i|\le M$ with probability 1. Then for all $t>0$,

$$\Pr\left(\left|\sum_{i=1}^{n}W_i\right|\ge t\right) \le 2\exp\left(-\frac{\frac{1}{2}t^2}{\sum_{i=1}^{n}\mathbb{E}\left[W_i^2\right]+\frac{1}{3}Mt}\right). \tag{12}$$

To apply this inequality, we first provide the following proposition which will serve as key ingredients for proving the lemma.

Proposition 7 *Let $W_{ij}^{(\ell)} := [A^{(\ell)}]_{ij}x_{(\ell-1)j} - a_{ij}x_{(\ell-1)j}$. Then for all $(\ell,i,j)\in[L]\times[n]\times[n]$, the following two hold:*

1. $\left|W_{ij}^{(\ell)}\right| \le \max\left(1, \frac{nL}{cB}\right)$

2. $\mathbb{E}\left[\left|W_{ij}^{(\ell)}\right|^2\right] \leq \frac{nL}{B}$

Proof Since $[A^{(\ell)}]_{ij} = \frac{a_{ij}}{p_j^{(\ell)}} I_{ij}^{(\ell)}$, we bound $\left|W_{ij}^{(\ell)}\right|$ as:

$$\begin{aligned}
\left|W_{ij}^{(\ell)}\right| &= \left|\frac{a_{ij}}{p_j^{(\ell)}} I_{ij}^{(\ell)} x_{(\ell-1)j} - a_{ij} x_{(\ell-1)j}\right| = a_{ij}|x_{(\ell-1)j}| \left|\frac{1}{p_j^{(\ell)}} I_{ij}^{(\ell)} - 1\right| \\
&\leq |x_{(\ell-1)j}| \max\left(\frac{1}{p_j^{(\ell)}}, 1\right) = \frac{|x_{(\ell-1)j}|}{p_j^{(\ell)}} \\
&= \frac{|x_{(\ell-1)j}|}{\min\left[\frac{B}{nL} \cdot \max\left(x_{(\ell-1)j}^2, c^2\right), 1\right]} \\
&\leq \frac{|x_{(\ell-1)j}|}{\min\left[\frac{B}{nL} \cdot c|x_{(\ell-1)j}|, 1\right]} \\
&= \max\left(\left|x_{(\ell-1)j}\right|, \frac{nL}{cB}\right) \leq \max\left(1, \frac{nL}{cB}\right),
\end{aligned} \tag{13}$$

where in the first inequality we used the fact that $a_{ij} \in \{0, 1\}$ so $a_{ij} \leq 1$, and in the second-to-last inequality we used the fact that $\max(a^2, b^2) \geq |ab|$.

We also bound the the second moment of $W_{ij}^{(\ell)}$ as

$$\begin{aligned}
\mathbb{E}\left[\left|W_{ij}^{(\ell)}\right|^2\right] &= \mathbb{E}\left[\left([A^{(\ell)}]_{ij} x_{(\ell-1)j} - a_{ij} x_{(\ell-1)j}\right)^2\right] \\
&= \mathbb{E}\left[[A^{(\ell)}]_{ij}^2 x_{(\ell-1)j}^2 - 2a_{ij}[A^{(\ell)}]_{ij} x_{(\ell-1)j}^2 + a_{ij}^2 x_{(\ell-1)j}^2\right] \\
&= \frac{a_{ij}^2}{p_j^2} \mathbb{E}\left[I_{ij}^{(\ell)}\right] x_{(\ell-1)j}^2 - 2\frac{a_{ij}}{p_j} \mathbb{E}\left[I_{ij}^{(\ell)}\right] x_{(\ell-1)j}^2 + a_{ij}^2 x_{(\ell-1)j}^2 \\
&= \frac{a_{ij}}{p_j} x_{(\ell-1)j}^2 - a_{ij} x_{(\ell-1)j}^2 = a_{ij} x_{(\ell-1)j}^2 \left(\frac{1}{p_j} - 1\right) \\
&\leq \frac{x_{(\ell-1)j}^2}{p_j} \leq \frac{x_{(\ell-1)j}^2}{\min\left[\frac{B}{nL} x_{(\ell-1)j}^2, 1\right]} \leq \max\left(\frac{nL}{B}, 1\right) = \frac{nL}{B},
\end{aligned} \tag{14}$$

where we used the facts that $\mathbb{E}[I_{ij}^{(\ell)}] = p_j$, $a_{ij} \in \{0, 1\}$ and $|x_{(\ell-1)j}| \leq 1$; and the last equality holds for any reasonable budget $B \geq nL$ (which is required for us to have at least one sample per row per iteration). This completes the proof. ■

Now we are ready to prove that conditions 1 and 2 hold. For conciseness, define $Z_i^{(\ell)} := [A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i - [A\boldsymbol{x}_{\ell-1}]_i$. Then one can readily see that $Z_i^{(\ell)} = \sum_{j=1}^n W_{ij}^{(\ell)}$. Starting with the union

bound, we obtain

$$
\begin{aligned}
&\Pr\left(\exists \ell \in [L] : \left\|A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}\right\| \geq \frac{\epsilon}{5}(\sigma_1-\sigma_2)\right)\\
&= \Pr\left(\bigcup_{\ell=1}^{L}\left\{\left\|A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}\right\| \geq \frac{\epsilon}{5}(\sigma_1-\sigma_2)\right\}\right)\\
&\leq \Pr\left(\bigcup_{\ell=1}^{L}\bigcup_{i=1}^{n}\left\{\left|[A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i - [A\boldsymbol{x}_{\ell-1}]_i\right| \geq \frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right\}\right)\\
&\leq nL\cdot\Pr\left(\left|[A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i - [A\boldsymbol{x}_{\ell-1}]_i\right| \geq \frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right)\\
&= nL\cdot\Pr\left(\left|Z_i^{(\ell)}\right| \geq \frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right)\\
&= nL\cdot\Pr\left(\left|\sum_{j=1}^{n} W_{ij}^{(\ell)}\right| \geq \frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right)\\
&\overset{(i)}{\leq} 2nL\cdot\exp\left(-C_1\log\left(\frac{n}{\epsilon}\right)\right)\\
&\overset{(ii)}{=} 2nC'\cdot\frac{\sigma_1}{\sigma_1-\sigma_2}\cdot\log\left(\frac{n\tau}{\epsilon}\right)\cdot\exp\left(-C_1\log\left(\frac{n}{\epsilon}\right)\right)\\
&\leq 2\exp\left((1-C_1)\log\left(\frac{n}{\epsilon}\right)+\log\left(2C'\cdot\frac{\sigma_1}{\sigma_1-\sigma_2}\cdot\log\left(\frac{n\tau}{\epsilon}\right)\right)\right)\\
&\leq 2\exp\left((2-C_1)\log\left(\frac{n}{\epsilon}\right)+o(\log n)\right)\\
&= n^{-\Omega(1)},
\end{aligned}
$$

where (i) follows from Proposition 8 below that holds under the setting $B = \frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$; and (ii) follows from setting $L = C'\cdot\frac{\sigma_1}{\sigma_1-\sigma_2}\cdot\log\left(\frac{n\tau}{\epsilon}\right)$ for some constant $C'>0$.

For sufficiently large n, we therefore conclude that as long as we choose $B = \frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$ and set τ large enough to guarantee $C_1 > 2$, the following holds

$$\left\|A^{(\ell)}\boldsymbol{x}_{\ell-1} - A\boldsymbol{x}_{\ell-1}\right\| \leq \frac{\epsilon}{5}(\sigma_1-\sigma_2), \quad \forall \ell \in [L],$$

with probability at least $1 - n^{-\Omega(1)}$.

Proposition 8 *If* $c = \frac{\sqrt{\alpha}(\sigma_1-\sigma_2)}{n^{3/2}}$ *and* $B = \frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$, *for each* $\ell \in [L]$,

$$\Pr\left(\left|\sum_{j=1}^{n} W_{ij}^{(\ell)}\right| \geq \frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right) \leq 2\exp\left(-C_1\log\left(\frac{n}{\epsilon}\right)\right).$$

Proof From Bernstein's inequality described in equation (12), we obtain:

$$
\begin{aligned}
&\Pr\left(\left|\sum_{j=1}^{n} W_{ij}^{(\ell)}\right| \geq \frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right) \\
&\overset{(iii)}{\leq} 2\exp\left(-\frac{1}{2}\cdot\frac{\left(\frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right)^2}{\sum_{j=1}^{n}\frac{nL}{B}+\frac{\epsilon(\sigma_1-\sigma_2)}{15\sqrt{n}}\cdot\max\left(1,\frac{nL}{cB}\right)}\right) \\
&\overset{(iv)}{=} 2\exp\left(-\frac{1}{2}\cdot\frac{\left(\frac{\epsilon}{5\sqrt{n}}(\sigma_1-\sigma_2)\right)^2}{\frac{n^2L}{B}+\frac{\epsilon(\sigma_1-\sigma_2)}{15\sqrt{n}}\cdot\max\left(1,\frac{n^{5/2}L}{\sqrt{\alpha}(\sigma_1-\sigma_2)B}\right)}\right) \\
&\overset{(v)}{\leq} 2\exp\left(-\frac{\epsilon^2}{50}\cdot\frac{\frac{(\sigma_1-\sigma_2)^2}{n}}{\max\left[\frac{n^2L}{B}+\frac{\epsilon(\sigma_1-\sigma_2)}{15\sqrt{n}},\frac{n^2L}{C''B}\right]}\right) \\
&= 2\exp\left(-\frac{\epsilon^2}{50}\cdot\min\left[C''\cdot\frac{B}{L}\cdot\frac{(\sigma_1-\sigma_2)^2}{n^3},\frac{1}{\frac{n^3L}{B(\sigma_1-\sigma_2)^2}+\frac{\epsilon\sqrt{n}}{15(\sigma_1-\sigma_2)}}\right]\right) \\
&\overset{(vi)}{=} 2\exp\left(-\frac{\epsilon^2}{50}\cdot C''\cdot\frac{B}{L}\cdot\frac{(\sigma_1-\sigma_2)^2}{n^3}\right) \\
&\overset{(vii)}{\leq} 2\exp\left(-C_1\log\left(\frac{n}{\epsilon}\right)\right) \quad \text{for some } C_1>2,
\end{aligned}
$$

where (iii) follows from equation (13) and equation (14) shown in Proposition 7; (iv) follows from the fact that the choice of $c=\frac{\sqrt{\alpha}(\sigma_1-\sigma_2)}{n^{3/2}}$, $B=\frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$, L described in (ii); (v) follows from setting $C''=(1+\frac{15\sqrt{\alpha}}{\epsilon})C'$; (vi) follows from the fact that the regime of interest would be $\sigma_1-\sigma_2=\Omega(\sqrt{n}\log n)$ so that $B=\frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)=O(n^2)$; and (vii) follows by setting $B=\frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$ and τ large enough to guarantee $C_1>2$. This completes the proof. ■

In order to show the condition 2, we view $(A^{(\ell)}\boldsymbol{x}_{\ell-1}-A\boldsymbol{x}_{\ell-1})^\top\boldsymbol{u}_1$ as the sum of n^2 independent random variables. Similar to what we have shown before, applying Bernstein's inequality yields:

$$
\begin{aligned}
&\Pr\left(\exists\ell\in[L]:\left|\left(A^{(\ell)}\boldsymbol{x}_{\ell-1}-A\boldsymbol{x}_{\ell-1}\right)^\top\boldsymbol{u}_1\right|\geq\frac{1}{5\tau\sqrt{n}}(\sigma_1-\sigma_2)\right) \\
&\leq L\Pr\left(\exists\ell\in[L]:\left|\sum_{i=1}^{n}\sum_{j=1}^{n}W_{ij}^{(\ell)}u_{1i}\right|\geq\frac{1}{5\tau\sqrt{n}}(\sigma_1-\sigma_2)\right) \\
&\overset{(viii)}{\leq} 2L\exp\left(-\frac{1}{50\tau^2}\cdot\frac{\frac{(\sigma_1-\sigma_2)^2}{n}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{u_{1i}^2nL}{B}+\frac{1}{3}\cdot\frac{nL}{cB}\cdot\frac{1}{5\tau\sqrt{n}}(\sigma_1-\sigma_2)}\right)
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(ix)}{=} 2L \exp\left(-\frac{1}{50\tau^2}\cdot\frac{\frac{(\sigma_1-\sigma_2)^2}{n}}{\frac{n^2L}{B}+\frac{(\sigma_1-\sigma_2)\sqrt{n}}{15\tau}\cdot\frac{L}{cB}}\right)\\
&= 2L\exp\left(-\frac{1}{50\tau^2}\cdot\frac{B}{L}\cdot\left(\frac{(\sigma_1-\sigma_2)^2}{n^3}+\frac{15\tau c(\sigma_1-\sigma_2)}{n^{3/2}}\right)\right)\\
&= 2L\exp\left(-\frac{1+15\tau\sqrt{\alpha}}{50\tau^2}\cdot B\cdot\frac{(\sigma_1-\sigma_2)^3}{\sigma_1 n^3\log\left(\frac{n\tau}{\epsilon}\right)}\right)\\
&\overset{(x)}{\leq} 2\exp\left(-C_2\log\left(\frac{n\tau}{\epsilon}\right)+o(\log n)\right)\quad \text{for some } C_2>0\\
&= n^{-\Omega(1)},
\end{aligned}
$$

where $(viii)$ follows from setting $c=\frac{\sqrt{\alpha}(\sigma_1-\sigma_2)}{n^{3/2}}$, from Proposition 7 and the regime of interest described in (vi) and thus $\frac{nL}{cB}>1\geq|x_{(\ell-1)j}|$; (ix) follows from the fact that $\|\boldsymbol{u}_1\|_2=1$; and (x) follows by setting $B=\frac{\sigma_1}{(\sigma_1-\sigma_2)^3}\frac{n^3}{\epsilon^2}\log^2\left(\frac{n\tau}{\epsilon}\right)$. Therefore, we see that for sufficiently large n, the following holds

$$
\left|(A^{(\ell)}\boldsymbol{x}_{\ell-1}-A\boldsymbol{x}_{\ell-1})^\top\boldsymbol{u}_1\right|\leq\frac{1}{5\tau\sqrt{n}}(\sigma_1-\sigma_2),\quad\forall\ell\in[L],
$$

with probability at least $1-n^{-\Omega(1)}$. Applying union bound techniques, we obtain the desired result. This completes the proof.

Appendix D. Detailed Derivation of Equation (6)

The rationale behind our new scheme is as follows. Decomposing the total budget-per-row into $\frac{B}{L}=\sum_i B_i^{(\ell)}$, where B_i denotes the expected budget allocated to row i, the key observation is by Cauchy-Schwarz inequality that

$$
\begin{aligned}
\mathrm{Var}(A^{(\ell)}\boldsymbol{x}_{\ell-1})&=\sum_{i=1}^n\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)=\frac{1}{B/L}\sum_{i=1}^n\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)\cdot\frac{B}{L}\\
&=\frac{L}{B}\left(\sum_{i=1}^n\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)\right)\cdot\left(\sum_{i=1}^n B_i\right)\\
&\geq\frac{L}{B}\sum_{i=1}^n\left(\sqrt{\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)}\cdot\sqrt{B_i}\right)^2,
\end{aligned}
$$

and the equality holds if and only if $B_i\propto\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i$. Notice that, if we have $p_{ij}^{(\ell)}=\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)\cdot\frac{x_{(\ell-1)j}^2}{\beta}$ for all i, for some normalization constant β, we obtain

$$
B_i^{(\ell)}=\sum_j p_{ij}^{(\ell)}=\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)\sum_j\frac{x_{(\ell-1)j}^2}{\beta}=\frac{\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)}{\beta},
$$

and the Cauchy-Schwarz bound above is achieved with equality. Plugging $p_{ij}^{(\ell)} = \mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i) \cdot \frac{x_{(\ell-1)j}^2}{\beta}$ into the formula for $\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)$ in (5), we obtain

$$\begin{aligned}\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i) =& \sum_{j=1}^{n} \frac{a_{ij}^2 x_{(\ell-1)j}^2}{p_{ij}^{(\ell)}}(1-p_{ij}^{(\ell)}) \\ =& \sum_{j=1}^{n} \frac{a_{ij}^2}{\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)} \left(\beta - x_{(\ell-1)j}^2 \,\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)\right) \\ =& \frac{\beta}{\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)} \sum_{j=1}^{n} a_{ij}^2 - \sum_{j=1}^{n} a_{ij}^2 x_{(\ell-1)j}^2\end{aligned}$$

If we let $y = \mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)$ and let $\|A_{i:}\|^2 = \sum_j a_{ij}^2$ be the squared ℓ_2 norm of ith row of A, the above can be written as the equation

$$y = \frac{\beta}{y}\|A_{i:}\|^2 - \|[A\boldsymbol{x}_{\ell-1}]_i\|^2 \;\Leftrightarrow\; y^2 + \|[A\boldsymbol{x}_{\ell-1}]_i\|^2 y - \beta\|A_{i:}\|^2 = 0,$$

which yields the solution

$$y = \mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i) = \frac{-\|[A\boldsymbol{x}_{\ell-1}]_i\|^2 + \sqrt{\|[A\boldsymbol{x}_{\ell-1}]_i\|^4 + 4\beta\|A_{i:}\|^2}}{2}.$$

We conclude that, if we had access to $\|[A\boldsymbol{x}_{\ell-1}]_i\|$ and $\|A_{i:}\|$ (which are not known, as A is not known), the optimal choice of p_{ij}s at the ℓth iteration would be

$$p_{ij}^{(\ell)} = \frac{x_{(\ell-1)j}^2}{\beta^{(\ell)}} \cdot \left(\frac{-\|[A\boldsymbol{x}_{\ell-1}]_i\|^2 + \sqrt{\|[A\boldsymbol{x}_{\ell-1}]_i\|^4 + 4\beta^{(\ell)}\|A_{i:}\|^2}}{2}\right),$$

where $\beta^{(\ell)}$ is chosen so that the expected budget per iteration is B/L. This completes the derivation of the equation (6). Notice that the result holds for all $i \in [n]$. Also notice that the resulting expected number of samples in row i is

$$\sum_j p_{ij}^{(\ell)} = \frac{-\|[A\boldsymbol{x}_{\ell-1}]_i\|^2 + \sqrt{\|[A\boldsymbol{x}_{\ell-1}]_i\|^4 + 4\beta^{(\ell)}\|A_{i:}\|^2}}{2\beta^{(\ell)}} = \frac{\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)}{\beta^{(\ell)}},$$

which is proportional to $\mathrm{Var}([A^{(\ell)}\boldsymbol{x}_{\ell-1}]_i)$ as expected.

Appendix E. Sample Complexity of the Non-adaptive Scheme

In this section, we provide a proof sketch of the sample complexity needed for the non-adaptive estimation algorithm that samples entries of A *uniformly at random*. Specifically, we will see that the non-adaptive scheme can estimate the top eigenvector (which we will denote by $\hat{\boldsymbol{u}}_1$) with squared error at most ϵ by observing $B = \Omega\left(\max\left\{\frac{(1+\epsilon)^2}{\epsilon^2} \cdot \frac{n^3}{(\sigma_1-\sigma_2)^2}, n\log n\right\}\right)$ entries of A.

Our proof sketch consists of two parts. We initially resort to the Davis-Kahan theorem (Davis and Kahan, 1970) to bound the ℓ_2 error between the top eigenvector and the estimate with respect to

the operator norm of the difference between A and the estimated matrix (denoted by $\hat{A}$). We will then use a result by Hajek et al. (2016, Theorem 5) to bound the operator norm $\|\hat{A} - A\|_{\text{op}}$, which will yield the desired result.

Suppose the algorithm sets the sampling probability to be $p = \frac{B}{n^2}$ to construct the estimated matrix $\hat{A}$ of A, where $\hat{A}_{ij} \sim \frac{n^2}{B} \cdot a_{ij} \cdot \text{Bern}(\frac{B}{n^2})$. Then one can readily see that $\mathbb{E}[\hat{A}] = A$. Denoting the top eigenvector of $\hat{A}$ as $\hat{\boldsymbol{u}}_1$, we observe from the Davis-Kahan theorem that

$$\left\|\boldsymbol{u}_1 - \langle \boldsymbol{u}_1, \hat{\boldsymbol{u}}_1 \rangle \hat{\boldsymbol{u}}_1\right\| \le \frac{\left\|\hat{A} - \mathbb{E}[\hat{A}]\right\|_{\text{op}}}{\max\left\{\sigma_1(A) - \sigma_2(\hat{A}), \sigma_1(\hat{A}) - \sigma_2(A)\right\}} \le \frac{\left\|\hat{A} - \mathbb{E}[\hat{A}]\right\|_{\text{op}}}{\sigma_1(A) - \sigma_2(A) - \left\|\hat{A} - \mathbb{E}[\hat{A}]\right\|_{\text{op}}}.$$

Here $\|\cdot\|_{\text{op}}$ denotes the operator norm (i.e., $\|A\|_{\text{op}} := \sup_{\|w\|=1} \|Aw\|$). Now, the result by Hajek et al. (2016, Theorem 5) implies that $\|\hat{A} - A\|_{\text{op}} \le c\sqrt{np} \cdot \frac{1}{p}$ for some $c > 0$, with probability at least $1 - n^{-c}$, under the condition that $\frac{B}{n^2} = \Omega(\frac{\log n}{n})$. We note that the $\frac{1}{p}$ factor is due to the scaling when constructing $\hat{A}$. Using this, we conclude that with probability at least $1 - n^{-c}$,

$$\begin{aligned}
\left\|\boldsymbol{u}_1 - \langle \boldsymbol{u}_1, \hat{\boldsymbol{u}}_1 \rangle \hat{\boldsymbol{u}}_1\right\| &\le \frac{\left\|\hat{A} - A\right\|_{\text{op}}}{\sigma_1(A) - \sigma_2(A) - \left\|\hat{A} - A\right\|_{\text{op}}} \\
&\le \frac{c\sqrt{\frac{B}{n}} \cdot \frac{n^2}{B}}{\sigma_1(A) - \sigma_2(A) - c\sqrt{\frac{B}{n}} \cdot \frac{n^2}{B}} \\
&\le \epsilon,
\end{aligned}$$

provided that $B = \Omega\left(\max\left\{\frac{(1+\epsilon)^2}{\epsilon^2} \cdot \frac{n^3}{(\sigma_1 - \sigma_2)^2}, n \log n\right\}\right)$.