2024 60th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3315-4103-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ALLERTON63246.2024.10735322

The Power Method with Row Erasures

Nicholas Sacco*, Viveck Cadambe†, and Bobak Nazer*
*Department of Electrical and Computer Engineering, Boston University
†School of Electrical and Computer Engineering, Georgia Tech

***Abstract*—This paper considers the task of estimating the principal eigenvector of a positive semi-definite matrix using the power method subjected to random row erasures at each iteration. This can be used to model applications where large matrix operations are distributed across multiple servers, some of which may fail to respond at each iteration. We analyze the simple strategy of only updating coordinates corresponding to non-erased rows, and demonstrate that, presuming a good initialization, the power method with erasures still converges exponentially fast to the principal eigenvector. The rate of convergence is governed by a modified spectral gap, which is a function of the original spectral gap and the fraction of erased rows. Accompanying numerical results validate our bounds, and demonstrate that, in certain regimes, our approach outperforms techniques such as coded computation and Oja's algorithm.**

I. Introduction

In many modern scientific and engineering applications, we must work with very large data matrices as part of the inference process. Due to memory and computational constraints, it is quite natural to distribute the required matrix operations across multiple servers. However, it is often the case that one or more servers will encounter significant delays in completing their assigned jobs [1]. Of course, one can always simply wait for these "stragglers" or, alternatively, reassign their jobs to other servers.

Another possibility is to preemptively introduce some form of erasure coding so that delayed computations can be reliably reconstructed from completed ones. Specifically, for the important special case of matrix multiplication, recent efforts on *coded computation* [2]–[5] have delineated the information-theoretic limits for recovering from the erasures introduced by stragglers, as well as code constructions that can efficiently approach these limits.

In this paper, we focus on the specific problem of recovering the principal eigenvector of a positive semi-definite (PSD) matrix via the power method. Let $A \in \mathbb{R}^{n\times n}$ be a PSD matrix with eigendecomposition

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^\top \tag{1}$$

where $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ are the eigenvalues, presumed to be sorted into decreasing magnitude order, $\lambda_1 > \cdots > \lambda_n$, and $v_1, \dots, v_n \in \mathbb{R}^n$ are the corresponding eigenvectors. Starting with initial vector $v_0 \in \mathbb{R}^n$, the power method refines its estimate of the principal eigenvector v_1 at each iteration by first multiplying by A and then normalizing,

$$x^{(t)} = A\hat{v}^{(t-1)} \ , \qquad \hat{v}^{(t)} = \frac{x^{(t)}}{\|x^{(t)}\|} \ . \tag{2}$$

It is well-known that the power method converges exponentially fast to the principal eigenvector. Specifically, the sine-squared error is upper bounded as a simple function of the spectral gap

$$\sin^2\theta_t = 1 - \langle v_1, \hat{v}^{(t)}\rangle^2 \le \left(\frac{\lambda_2}{\lambda_1}\right)^{2t} \tan^2\theta_0 \ . \tag{3}$$

where $\tan^2\theta_0 = \frac{1-\alpha_1^2}{\alpha_1^2}$, and $\alpha_1^2 = \langle v_1, v_0\rangle^2$. We assume that, in a distributed implementation, the required row-vector products for a single iteration are distributed across multiple servers. Erasures (due to stragglers) can be concisely described via a diagonal projection matrix $\mathcal{P}^{(t)}$ with 1's in the entries corresponding to erased rows and 0's otherwise. Overall, the servers collectively return $(I - \mathcal{P}^{(t)})A\hat{v}^{(t-1)}$, rather than the desired $A\hat{v}^{(t-1)}$. As discussed above, coded computation could be used to recover from these erasures at each iteration. The aim of this paper is to investigate the simpler possibility of proceeding with the next iteration while only updating the available coordinates. Intuitively, this recursion should also recover v_1, provided that all coordinates are sampled sufficiently often.

Consider the following modified power method for making progress in the presence of erasures:

$$x^{(t)} = \mathcal{P}^{(t)}x^{(t-1)} + (I - \mathcal{P}^{(t)})A\hat{v}^{(t-1)} \ , \tag{4}$$

$$\hat{v}^{(t)} = \frac{x^{(t)}}{\|x^{(t)}\|} \ . \tag{5}$$

Ideally, we would like to establish that this recursion rapidly converges to v_1, using an upper bound of similar form to (3). This paper makes progress towards this goal by providing theoretical guarantees for the recursion

$$x^{(t)} = \lambda_1\mathcal{P}^{(t)}\hat{v}^{(t-1)} + (I - \mathcal{P}^{(t)})A\hat{v}^{(t-1)} \tag{6}$$

$$\hat{v}^{(t)} = \frac{x^{(t)}}{\|x^{(t)}\|} \tag{7}$$

where we presume that an oracle provides knowledge of the principal eigenvalue (but not the eigenvector).[1] Note that this choice is intended to mimic the scale of the $x^{(t-1)} = \|x^{(t-1)}\|\hat{v}^{(t-1)}$ memory term in (4), since we expect $\|x^{(t-1)}\|$ to converge to λ_1.

We also demonstrate empirically that the performance of these two recursions are very close, and our future work will aim to extend our theoretical guarantees to (6). Overall, our results suggest that, for certain iterative algorithms involving matrix products, it may be more efficient to proceed with the next iteration in the midst of erasures, rather than utilize additional resources for coded computation to recover from these erasures at each iteration.

II. Related Work

There has been considerable recent interest in developing efficient methods for high-dimensional matrix computations [6]. Owing to space limitations, here we briefly summarize recent works related to handling erasures. As discussed above, coded computation strategies [2]–[5], [7], construct an erasure code on top of the linear computations assigned to the servers, and can recover from erasures up to a threshold. Recent work has also considered approximate reconstruction [8].

Variants on the *noisy power method* [9]–[13] encompass settings where the desired update $A\hat{v}^{(t-1)}$ is perturbed by some noise vector, which can be chosen to model erasures as well as independent additive noise. However, the theoretical guarantees require tight bounds on the norm of the noise and its projections onto certain subspaces. Here, we are able to derive faster convergence rates by focusing on the special structure of row erasures. Recent work has also considered adaptively subsampling the data matrix at each iteration of the power method. [13]

Oja's algorithm [14] is a well-known variation on the power method that is adapted to online (or streaming) principal component analysis. A line of recent work [15]–[17] has developed convergence guarantees for Oja's algorithm for a broad range of update configurations, essentially those where the expected value of the update is equal to the full update. Although these bounds include our setting of row erasures as a special case, we are able to derive sharper bounds by taking advantage of the additional structure induced by row erasures.

Approximate message passing is a powerful framework for the analysis of recursive algorithms applied to high-dimensional matrices (see, e.g., [18] for a survey). Very recent work [19] (by the last author and others) has proposed *linear operator approximate message passing* (OpAMP) as a framework that includes row erasures as a special case. (In fact, our preliminary results served as a motivation for OpAMP.) While OpAMP provides very precise guarantees, they rely on distributional assumptions on the data matrix (e.g., that it is drawn from the Gaussian orthogonal ensemble) whereas our results depend only on the spectral gap.

[1] For certain applications, exact knowledge of λ_1 can be viewed as perfect knowledge of the effective signal-to-noise ratio.

III. Main Results

Throughout the paper, we will assume that, at each iteration, exactly m out of n rows are erased. For a subset $\mathcal{K} \subset [n]$, let

$$\mathcal{P}_{\mathcal{K}} = \sum_{k\in\mathcal{K}} e_k e_k^\top \tag{8}$$

be the projection matrix onto the coordinates in $\mathcal{K}$ where e_k is the k^{th} standard basis vector. We further assume that the erasure coordinates are chosen uniformly at each iteration and independently across iterations. Specifically, let

$$\mathcal{D}_m = \big\{\tilde{A}_{\mathcal{K}} : \tilde{A}_{\mathcal{K}} = A + \mathcal{P}_{\mathcal{K}}(\lambda_1 I - A), \quad \mathcal{K} \subset [n], |\mathcal{K}| = m\big\} \tag{9}$$

be the set of all matrix updates with m coordinates erased. We can concisely express the recursion (6) as

$$x^{(t)} = A_t\hat{v}^{(t-1)} \qquad \hat{v}^{(t)} = \frac{x^{(t)}}{\|x^{(t)}\|} \tag{10}$$

where we assume that $A_1, A_2, \ldots$ are drawn i.i.d. $\mathrm{Unif}(\mathcal{D}_m)$. Our main result demonstrates that, given a good initialization, this recursion converges to v_1 exponentially fast according to a modified spectral gap that is a function of the number of erasures.

Theorem 1. Assume that the initialization $v_0 \in \mathbb{R}^n$ is selected such that

$$\tan^2\theta_0 \le \left(1 - \Delta^2\right)\left(\frac{\delta(1-c)^2(n-1)}{mv_*^2}\right) \tag{11}$$

where $v_* = \max_i |v_{1,i}|$ is the largest coordinate in the principal eigenvector and $\Delta = \frac{\lambda_2}{\lambda_1}$. Further assume that, for some $m \le n$, the update matrices $A_1, A_2, \ldots$ are generated i.i.d. $\mathrm{Unif}(\mathcal{D}_m)$ and that $\hat{v}^{(t)}$ is generated from the recursion (10). Then, with probability at least $1 - 2\delta$, we have

$$\sin^2\theta_t = 1 - \langle v_1, \hat{v}^{(t)}\rangle^2 \le \frac{1}{\delta c^2}\tilde{\Delta}^{2t}\tan^2\theta_0 \tag{12}$$

where

$$\tilde{\Delta}^2 = \left(\frac{\lambda_2}{\lambda_1}\right)^2 + \frac{m}{n}\left(1 - \left(\frac{\lambda_2}{\lambda_1}\right)^2\right). \tag{13}$$

Remark 1. In cases where the initialization condition (11) is not satisfied, we can use a "warm start" with a few erasure-free iterations from (1) to decrease the initial error. For example, if we assume that v_1 is drawn uniformly over the unit sphere, then approximately $\log n$ erasure-free iterations would suffice. Our future work will seek to relax the initialization condition.

To build intuition for why the modified spectral gap appears, we first consider the impact of a single update in expectation. Using Lemma 8, it follows that

$$\mathbb{E}\left[A_t\right] = A + \frac{m}{n}(\lambda_1 I - A) \tag{14}$$

$$= \sum_{i=1}^{n} \left(\lambda_i + \frac{m}{n}(\lambda_1 - \lambda_i)\right) \, v_i v_i^\top \; . \tag{15}$$

Thus, the average matrix $\mathbb{E}\left[A_t\right]$ shares the same basis as A, but with a new spectrum $\left\{\lambda_i + \frac{m}{n}(\lambda_1 - \lambda_i)\right\}$.

IV. Second-Moment Bounds

For our analysis, it will be convenient to defer normalization until the last step. Let $Z_t = A_t A_{t-1} \cdots A_2 A_1$ be the product of the effective matrix updates. The sine squared error is written as:

$$\sin^2 \theta_t = \frac{v_0^\top Z_t^T U_\perp Z_t v_0}{v_0^\top Z_t^T Z_t v_0} \tag{16}$$

$$\leq \frac{v_0^\top Z_t^T U_\perp Z_t v_0}{v_0^\top Z_t^T v_1 v_1^\top Z_t v_0} \tag{17}$$

Where $U_\perp = \sum_{i=2}^{n} v_i v_i^\top$ projects onto the orthogonal complement of v_1. If there are no erasures, this definition is exactly (3). Define random variables W_t, Y_t as:

$$W_t = v_1^\top Z_t v_0 \tag{18}$$

$$Y_t = \|U_\perp Z_t v_0\| \tag{19}$$

Thus, $\sin^2 \theta_t \leq \frac{Y_t^2}{W_t^2}$. Below, we bound the second moments $\mathbb{E}\left[Y_t^2\right]$ and $\mathbb{E}\left[W_t^2\right]$.

Lemma 1. Assume the update matrices $Z_t = A_t A_{t-1} \cdots A_1$ are generated i.i.d. Unif $(\mathcal{D}_m)$. Then:

$$\mathbb{E}[Y_t^2] \leq \tilde{\lambda}_2^{2t}(1 - \alpha_1^2) \tag{20}$$

where $\tilde{\lambda}_2^2 = \lambda_2^2 + \frac{m}{n}\left(\lambda_1^2 - \lambda_2^2\right)$, and $\alpha_1^2 = \langle v_1, v_0\rangle^2$.
Proof: The random product Z_t is decomposed as $Z_t = A_t Z_{t-1}$, where A_t is the effective matrix update at iteration t and Z_{t-1} describes the history over the first $t-1$ iterations. Conditioning on Z_{t-1} and using the independence between A_t and Z_{t-1}:

$$\begin{aligned}\mathbb{E}\left[Y_t^2 \middle| \, Z_{t-1}\right] &= \mathbb{E}\left[v_0^\top Z_{t-1}^T A_t^T U_\perp A_t Z_{t-1} v_0 \;\middle|\; Z_{t-1}\right] \\ &= v_0^\top Z_{t-1}^T \mathbb{E}\left[A_t^T U_\perp A_t\right] Z_{t-1} v_0 \end{aligned} \tag{21}$$

Applying Lemma 3 with $Q = U_\perp$:

$$\begin{aligned}\mathbb{E}\left[A_t^T U_\perp A_t\right] &= \omega_0 A U_\perp A + 2\omega_\ell \lambda_1 A U_\perp \\ &\quad + \omega_2(\lambda_1 I - A) U_\perp (\lambda_1 I - A) \\ &\quad + \omega_1(\lambda_1 I - A)\left(\sum_{\ell=1}^{n} P_\ell U_\perp P_\ell\right)(\lambda_1 I - A)\end{aligned}$$

where $P_\ell = e_\ell e_\ell^\top$. Using Lemma 10 ($\sum_{\ell=1}^{n} P_\ell U_\perp P_\ell) \preceq u_* I$, where $u_* = \max_\ell (e_\ell^\top U_\perp e_\ell) \leq 1$. Thus:

$$\begin{aligned}\mathbb{E}\left[A_t^T U_\perp A_t\right] &\preceq \omega_0 A U_\perp A + 2\omega_\ell \lambda_1 A U_\perp \\ &\quad + \omega_2(\lambda_1 I - A)^2 + \omega_1(\lambda_1 I - A)^2 \\ &\preceq \omega_0 A U_\perp A + 2\omega_\ell \lambda_1 A U_\perp \\ &\quad + (\omega_1 + \omega_2)(\lambda_1 I - A)^2 \end{aligned} \tag{22}$$

Writing the full matrix decomposition in the $\{v_i\}$ basis:

$$\begin{aligned}\mathbb{E}\left[A_t^T U_\perp A_t\right] &\preceq \sum_{i=2}^{n} \left[\omega_0 \lambda_i^2 + 2\omega_\ell \lambda_1 \lambda_i\right] v_i v_i^\top \\ &\quad + \sum_{i=2}^{n} \left[(\omega_1 + \omega_2)(\lambda_1 - \lambda_i)^2\right] v_i v_i^\top\end{aligned}$$

Simplifying using (41): $\omega_1 + \omega_2 = \frac{m}{n}$:

$$\begin{aligned}\mathbb{E}\left[A_t^T U_\perp A_t\right] &\preceq \sum_{i=2}^{n} \left[\left(1 - 2\frac{m}{n}\right)\lambda_i^2 + 2\frac{m}{n}\lambda_1 \lambda_i\right] v_i v_i^\top \\ &\quad + \sum_{i=2}^{n} \left(\frac{m}{n}\right)\left(\lambda_1^2 - 2\lambda_1 \lambda_i + \lambda_i^2\right) v_i v_i^\top \\ &\preceq \sum_{i=2}^{n} \left[\lambda_i^2 + \frac{m}{n}\left(\lambda_1^2 - \lambda_i^2\right)\right] v_i v_i^\top \end{aligned} \tag{23}$$

Let $\tilde{\lambda}_i^2 = \lambda_i^2 + \frac{m}{n}\left(\lambda_1^2 - \lambda_i^2\right)$. Substituting (23) into (21):

$$\mathbb{E}\left[Y_t^2 \middle| \, Z_{t-1}\right] \leq v_0^\top Z_{t-1}^T \left\{\sum_{i=2}^{n} \tilde{\lambda}_i^2 \, v_i v_i^\top\right\} Z_{t-1} v_0 \; .$$

Since $\tilde{\lambda}_i^2 \leq \tilde{\lambda}_2^2$ for $i \geq 2$, we have

$$\begin{aligned}\mathbb{E}\left[Y_t^2 \middle| \, Z_{t-1}\right] &\leq \tilde{\lambda}_2^2 \left[v_0^\top Z_{t-1}^T \left\{\sum_{i=2}^{n} v_i v_i^\top\right\} Z_{t-1} v_0\right] \\ &\leq \tilde{\lambda}_2^2 \left[v_0^\top Z_{t-1}^T U_\perp Z_{t-1} v_0\right] \; . \end{aligned} \tag{24}$$

Computing the total expectation, we obtain the recursion:

$$\begin{aligned}\mathbb{E}\left[Y_t^2\right] &= \mathbb{E}\left[\mathbb{E}\left[Y_t^2 \middle| \, Z_{t-1}\right]\right] \\ &\leq \tilde{\lambda}_2^2 \cdot \mathbb{E}\left[v_0^\top Z_{t-1}^T U_\perp Z_{t-1} v_0\right] \\ &\leq \tilde{\lambda}_2^2 \, \mathbb{E}\left[Y_{t-1}^2\right] \end{aligned} \tag{25}$$

Unravelling the full recursion $t-1$ more times, and using $\mathbb{E}\left[Y_0^2\right] = v_0^\top U_\perp v_0 = (1 - \alpha_1^2)$, we arrive at the claim.

Lemma 2. Assume the update matrices $Z_t =$

$A_t A_{t-1} \cdots A_1$ are generated i.i.d. Unif$(\mathcal{D}_m)$. Then:

$$\mathbb{E}\left[W_t^2\right] \leq \lambda_1^{2t}\left[\alpha_1^2 + \frac{m v_*^2}{(n-1)(1-\Delta^2)}(1-\alpha_1^2)\right] \quad (26)$$

where $\Delta^2 = \frac{\lambda_2^2}{\lambda_1^2}$, $v_* = \max_i |v_{1,i}|$, and $\alpha_1^2 = \langle v_1, v_0\rangle^2$.

Proof: For simplicity, we can factor out a λ_1 from each A_τ in the product $Z_t = A_t A_{t-1} \cdots A_2 A_1$. Then:

$$\begin{aligned}\mathbb{E}\left[W_t^2\right] &= \mathbb{E}\left[v_0^\top Z_t^T v_1 v_1^\top Z_t v_0\right] \\ &= \lambda_1^{2t}\, \mathbb{E}\left[v_0^\top \bar{Z}_t^T v_1 v_1^\top \bar{Z}_t v_0\right] \\ &= \lambda_1^{2t}\, \mathbb{E}\left[\bar{W}_t^2\right] \end{aligned} \quad (27)$$

where $\bar{W}_t = v_0^\top \bar{Z}_t^T v_1 v_1^\top \bar{Z}_t v_0$, $\bar{Z}_t = \bar{A}_t \bar{A}_{t-1} \cdots \bar{A}_2 \bar{A}_1$, $\bar{A}_\tau = \frac{A_\tau}{\lambda_1}$, and $\bar{A} = \frac{A}{\lambda_1}$. To bound $\mathbb{E}\left[\bar{W}_t^2\right]$, we proceed in a similar manner to the Y_t second moment bound. Decomposing the matrix product $\bar{Z}_t = \bar{A}_t \bar{Z}_{t-1}$, conditioning on $\bar{Z}_{t-1}$, and using the independence between $\bar{A}_t$ and $\bar{Z}_{t-1}$:

$$\begin{aligned}\mathbb{E}\left[\bar{W}_t^2 \middle| \bar{Z}_{t-1}\right] &= \mathbb{E}\left[v_0^\top \bar{Z}_{t-1}^T \bar{A}_t^T v_1 v_1^\top \bar{A}_t \bar{Z}_{t-1} v_0 \;\middle|\; \bar{Z}_{t-1}\right] \\ &= v_0^\top \bar{Z}_{t-1}^T \Big\{\mathbb{E}\left[\bar{A}_t^T v_1 v_1^\top \bar{A}_t\right]\Big\} \bar{Z}_{t-1} v_0 \end{aligned} \quad (28)$$

Applying Corollary 1 with $Q = v_1 v_1^\top$:

$$\begin{aligned}\mathbb{E}\left[\bar{A}_t^T v_1 v_1^\top \bar{A}_t\right] &= \omega_0 \bar{A} v_1 v_1^\top \bar{A} + 2\omega_\ell \bar{A} v_1 v_1^\top \\ &+ \omega_2 (I-\bar{A}) v_1 v_1^\top (I-\bar{A}) \\ &+ \omega_1 (I-\bar{A}) \left(\sum_{\ell=1}^n P_\ell v_1 v_1^\top P_\ell\right)(I-\bar{A})\end{aligned}$$

By definition, $\bar{A} v_1 = v_1$, and $(I-\bar{A})v_1 = 0$. Using (41) to expand the other ω_i constants:

$$\begin{aligned}\mathbb{E}\left[\bar{A}_t^T v_1 v_1^\top \bar{A}_t\right] &= \left(1 - 2\frac{m}{n}\right) v_1 v_1^\top + 2\frac{m}{n} v_1 v_1^\top \\ &+ \omega_1 (I-\bar{A}) \left(\sum_{\ell=1}^n P_\ell v_1 v_1^\top P_\ell\right)(I-\bar{A})\end{aligned}$$

Define matrix B as:

$$B = (I-A)\left(\sum_{\ell=1}^n P_\ell v_1 v_1^\top P_\ell\right)(I-A) \quad (29)$$

Then:

$$\mathbb{E}\left[\bar{A}_t^T v_1 v_1^\top \bar{A}_t\right] = v_1 v_1^\top + \omega_1 B \quad (30)$$

Substituting (30) into (28):

$$\begin{aligned}\mathbb{E}\left[\bar{W}_t^2 \middle| \bar{Z}_{t-1}\right] &= v_0^\top \bar{Z}_{t-1}^T v_1 v_1^\top \bar{Z}_{t-1} v_0 \\ &+ \omega_1 v_0^\top \bar{Z}_{t-1}^T B \bar{Z}_{t-1} v_0 \end{aligned} \quad (31)$$

Computing the total expectation:

$$\begin{aligned}\mathbb{E}\left[\bar{W}_t^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\bar{W}_t^2 \middle| \bar{Z}_{t-1}\right]\right] \\ &= \mathbb{E}\left[v_0^\top \bar{Z}_{t-1}^T v_1 v_1^\top \bar{Z}_{t-1} v_0\right] \\ &+ \omega_1 \mathbb{E}\left[v_0^\top \bar{Z}_{t-1}^T B \bar{Z}_{t-1} v_0\right] \\ &= \mathbb{E}\left[\bar{W}_{t-1}^2\right] + \omega_1 \mathbb{E}\left[R_t\right]\end{aligned} \quad (32)$$

where the residual term $\mathbb{E}[R_t]$ is defined as:

$$\mathbb{E}[R_t] = v_0^\top \mathbb{E}\left[\bar{Z}_{t-1}^T B \bar{Z}_{t-1}\right] v_0 \quad (33)$$

Unravelling the recursion $t-1$ more times and using $\mathbb{E}\left[\bar{W}_0\right] = v_0^\top v_1 v_1^\top v_0 = \alpha_1^2$, we obtain the intermediate result:

$$\mathbb{E}\left[\bar{W}_t^2\right] = \alpha_1^2 + \omega_1 \sum_{\tau=1}^t \mathbb{E}\left[R_\tau\right] \quad (34)$$

Using Lemma 5, to bound each residual term $\mathbb{E}\left[R_\tau\right]$:

$$\mathbb{E}\left[\bar{W}_t^2\right] \leq \alpha_1^2 + \omega_1 v_*^2 \sum_{\tau=1}^t v_0^\top B^{(\tau-1)} v_0 \quad (35)$$

where $B^{(\tau)}$ is a PSD matrix with eigenvalues $\left\{\eta_i^{(\tau)}\right\}$, $1 \leq i \leq n$ and corresponding eigenvectors $v_1,\ v_2, \dots v_n \in \mathbb{R}^n$, the same eigenvectors as A. The eigenvalues are defined according to the recursion:

$$\eta_i^{(t)} = c_i \eta_i^{(t-1)} + \omega_1 \eta_*^{(t-1)} (1-\Delta_i)^2,\ 2 \leq i \leq n \quad (36)$$

where $\eta_1^{(t)} = 0$ for all t, $\eta_*^{(t)} = \max_i \eta_i^{(t)}$, and the recursion is initialized with $\eta_i^{(0)} = (1-\Delta_i)^2$, $\eta_*^{(0)} = 1$. The c_i term is defined as:

$$c_i = \omega_0 \Delta_i^2 + 2\omega_\ell \Delta_i + \omega_2 (1-\Delta_i)^2 \quad (37)$$

Using the eigendecomposition (36) for each $B^{(\tau)}$ term:

$$\begin{aligned}\mathbb{E}\left[\bar{W}_t^2\right] &\leq \alpha_1^2 + \omega_1 v_*^2 \sum_{\tau=1}^t v_0^\top B^{(\tau-1)} v_0 \\ &\leq \alpha_1^2 + \omega_1 v_*^2 \sum_{\tau=1}^t v_0 \left\{\sum_{i=2}^n \eta_i^{(\tau-1)} v_i v_i^\top\right\} v_0 \\ &\leq \alpha_1^2 + \omega_1 v_*^2 \sum_{i=2}^n \alpha_i^2 \left\{\sum_{\tau=0}^{t-1} \eta_i^{(\tau-1)}\right\}\end{aligned}$$

where $\alpha_i^2 = \langle v_i, v_0\rangle^2$. Using Lemma 6, to bound the sum of η_i:

$$\begin{aligned}\mathbb{E}\left[\bar{W}_t^2\right] &\leq \alpha_1^2 + \omega_1 v_*^2 \sum_{i=2}^n \alpha_i^2 \left(\frac{n}{(n-m)(1-\Delta^2)}\right) \\ &\leq \alpha_1^2 + \frac{n\omega_1 v_*^2}{(n-m)(1-\Delta^2)}(1-\alpha_1^2)\end{aligned}$$

Using the definition of $\omega_1 = \left(\frac{n-m}{n-1}\right)\left(\frac{m}{n}\right)$ to simplify:

$$\mathbb{E}\left[\bar{W}_t^2\right] \leq \alpha_1^2 + \frac{m v_*^2}{(n-1)(1-\Delta^2)}(1-\alpha_1^2) \quad (38)$$

Substituting (38) into (27) yields the claim.

V. Proof of Theorem 1

Using Lemma 1 and Markov's inequality, we construct an upper bound on Y_t:

$$\mathbb{P}\left[Y_t \geq ka\right] \leq \frac{\mathbb{E}\left[Y_t^2\right]}{k^2a^2} \leq \frac{\tilde{\lambda}_2^{2t}(1-\alpha_1^2)}{k^2a^2}$$

Taking $a = \tilde{\lambda}_2^t\sqrt{(1-\alpha_1^2)}$ and $k = \sqrt{\delta^{-1}}$ yields $Y_t \leq \tilde{\lambda}_2^t\sqrt{\delta^{-1}(1-\alpha_1^2)}$ with probability at least $1-\delta$. Next, using Lemma 2 and Chebyshev's inequality, we construct a lower bound on W_t. Let $\sigma_t^2 = \mathbb{E}\left[W_t^2\right] - \mathbb{E}\left[W_t\right]^2$. Using (18) and independence between each update matrix A_t:

$$\mathbb{E}\left[W_t\right] = v_1^\top \mathbb{E}\left[A_t A_{t-1} \cdots A_2 A_1\right] v_0 = \alpha_1 \lambda_1^t$$

Using the second moment of W_t:

$$\begin{aligned}\sigma_t^2 &= \mathbb{E}\left[W_t^2\right] - \mathbb{E}\left[W_t\right]^2 \\ &\leq \lambda_1^{2t}\left[\alpha_1^2 + \frac{mv_*^2}{(n-1)(1-\Delta^2)}(1-\alpha_1^2)\right] - \alpha_1^2\lambda_1^{2t} \\ &\leq \lambda_1^{2t}\frac{mv_*^2}{(n-1)(1-\Delta^2)}(1-\alpha_1^2) \end{aligned} \tag{39}$$

Applying Chebyshev's Inequality:

$$\mathbb{P}\Big[W_t \notin (\alpha_1\lambda_1^t - k\sigma_t,\ \alpha_1\lambda_1^t + k\sigma_t)\Big] \leq \frac{1}{k^2}$$

Taking $k = \sqrt{\delta^{-1}}$ yields $W_t \geq \alpha_1\lambda_1^t - k\sigma_t$ with probability at least $1-\delta$. To ensure the lower bound is meaningful, we require:

$$\alpha_1\lambda_1^t - k\sigma_t \geq c\alpha_1\lambda_1^t \iff \sigma_t^2 \leq \delta(1-c)^2\alpha_1^2\lambda_1^{2t}$$

Where $c > 0$. Using (39), the bound on σ_t^2 :

$$\frac{mv_*^2}{(n-1)(1-\Delta^2)}(1-\alpha_1^2) \leq \delta(1-c)^2\alpha_1^2$$

$$\frac{1-\alpha_1^2}{\alpha_1^2} \leq \left(\frac{\delta(1-c)^2(n-1)}{mv_*^2}\right)(1-\Delta^2)$$

We require the starting vector v_0 to satisfy the following tangent squared error requirement:

$$\tan^2\theta_0 \leq \left(1-\Delta^2\right)\left(\frac{\delta(1-c)^2(n-1)}{mv_*^2}\right) \tag{40}$$

Under the "warm start" assumption (11), v_0 satisfies this bound. Thus, with probability at least $1-\delta$, $W_t > c\alpha_1\lambda_1^t$. To prove the full error bound (1), define two events. Let E_Y be the event where $\left\{Y_t > \tilde{\lambda}_2^t\sqrt{\delta^{-1}(1-\alpha_1^2)}\right\}$ and let E_W be the event where $\{W_t < c\alpha_1\lambda_1^t\}$. By construction, $\mathbb{P}[E_Y] \leq \delta$ and $\mathbb{P}[E_X] \leq \delta$. By the union bound, the overall probability of failure is at most 2δ; taking the ratio $\frac{Y_t^2}{W_t^2}$ yields the sine squared-error bound.

VI. Simulations

In this section, we evaluate the empirical performance of the reuse-norm (4) and oracle-update (6) protocols. We use a PSD matrix $A \in \mathbb{R}^{n\times n}$ with $n = 1000$, eigengap $\Delta = \frac{\lambda_2}{\lambda_1} = 0.90$, and fixed error ratio of $\frac{m}{n} = 0.75$. Each algorithm is initialized with the same $v_0 \in \mathbb{R}^n$ drawn uniformly on the sphere. The update matrices are drawn i.i.d. $\mathrm{Unif}(\mathcal{D}_m)$ (i.e., uniformly across all configurations of m erasures) across iterations. This erasure pattern is generated once per trial and then applied across all competing algorithms. The resulting performance metrics are averaged over 50 trials. We evaluate the proposed power method variants against the following competitors:

1) Erasure-free Power Method: The standard power method with convergence rate Δ^2.
2) Oja's Algorithm [16]: We set $\eta_t = \frac{\log n}{(\lambda_1-\lambda_2)(\beta+t)}$. For simplicity, we take $\beta = 1$, which is outside of the prescribed range in [16], but is sufficient for empirical comparison. Using Theorem 3 in [16] to construct a coarse bound, we expect Oja's algorithm to produce iterates with error
$$\sin^2\theta_t \leq \frac{\mathcal{V}\log n}{(\lambda_1-\lambda_2)^2 t} + \left(\frac{2}{t}\right)^{2\log n}$$
where $\mathcal{V} = \frac{m}{n}\left(1-\frac{m}{n}\right)\lambda_1$.
3) OpAMP [19]: Although our choice of A does not meet the distributional requirements of the AMP framework, in practice we still might expect the iterates to converge to v_1 in a reasonable fashion.
4) Short-Dot Code [2]. We include the short-dot code as a representative of coded computation strategies. The n dot products required to compute iterate $x^{(t)} = A\hat{v}^{(t-1)}$ are encoded into $P = 2n$ dot products. Any $K = \frac{3}{2}n$ coded products are sufficient to exactly recover $x^{(t)}$.

Following the evaluation strategies in [19], We consider two main performance metrics: Convergence speed and computational efficiency.

A. Convergence Speed

We compute the empirical sine-squared error at each iteration. The error bound of Theorem 1 is computed using $\delta = 0.05$ and $c = 1/2$ for simplicity. In Figure 1, we observe the iterates generated by the oracle update rule in (6) converge to v_1 at the modified rate $\tilde{\Delta}^2 > \Delta^2$ predicted by Theorem 1. In addition, the performance of the data-dependent recursion (4) is almost exactly approximated by the oracle recursion. In Figure 2, we also find the empirical error of the OpAMP protocol decays at rate $\tilde{\Delta}^2$, even though A does not quite meet the requirements of the AMP framework. In Figure 3, we observe the proposed oracle update (6) converges to v_1 at a faster rate than Oja's algorithm. In addition, we also find that empirically, Oja's algorithm converges much faster than the $O(1/t)$ bound in [16]. This suggests that error guarantees may be improved by specifically exploiting the random row-erasure structure explored in this work.

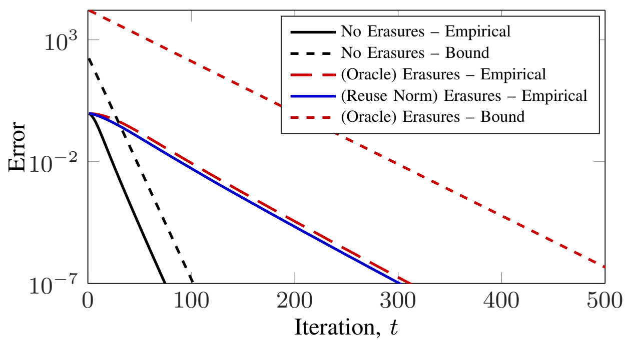

Fig. 1. (Oracle update (6), Reuse-Norm Update (4)) Sine-Squared Error vs. Iterations.

B. Computational Efficiency

We compute the correlation, defined as $\langle v_1, \hat{v}^{(t)} \rangle^2$ at each iteration. As a surrogate for computational efficiency, we track the number of erasure-free matrix-vector products of the form $A\hat{v}^{(t-1)}$ carried out at each iteration. In the erasure-free power method, exactly one full product is computed per iteration. For our proposed update rules in (4), (6), OpAMP, and Oja's algorithm, $(1 - \frac{m}{n})$ effective matrix-vector products are computed per iteration. In the short-dot code protocol, we assume that the full update $A\hat{v}^{(t-1)}$ is recovered if exactly $\frac{K}{n} = \frac{3}{2}$ effective matrix-vector products are computed per iteration. We compare the correlation against the number of erasure-free matrix-vector products in Figure 4. On a per-computation basis, we find the proposed data-dependent update (4) and the OpAMP protocol are slightly more efficient than the erasure-free method. The oracle update (6) is about as efficient as the standard power method. Oja's algorithm is slightly worse, largely due to its slower rate of convergence. However, all schemes outperformed the short-dot code, which demonstrated the lowest efficiency due to the redundant computations required to exactly recover each iterate. These results suggest it may be possible to deliberately inject row erasures into an algorithm, simultaneously reducing the computational load while still converging towards the target result.

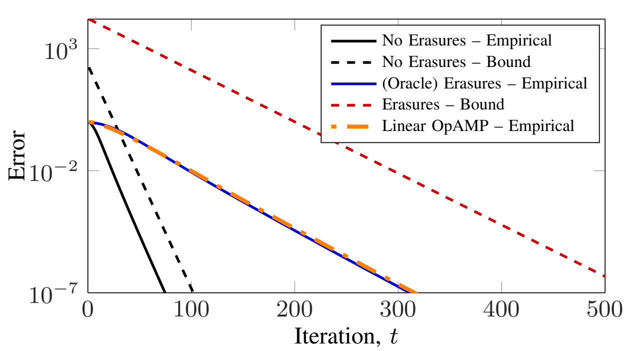

Fig. 2. (Oracle update (6), OpAMP) Sine-Squared Error vs. Iterations.

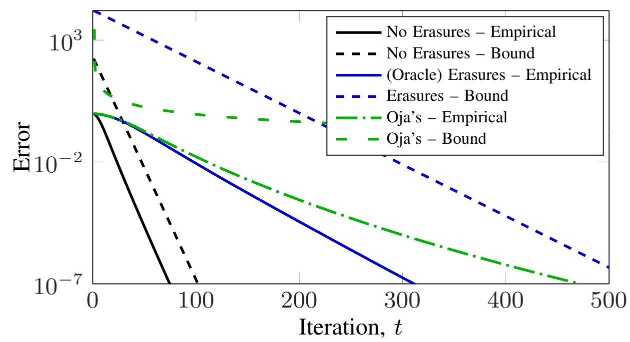

Fig. 3. (Oracle update (6), Oja's Algorithm) Sine-Squared Error vs. Iterations.

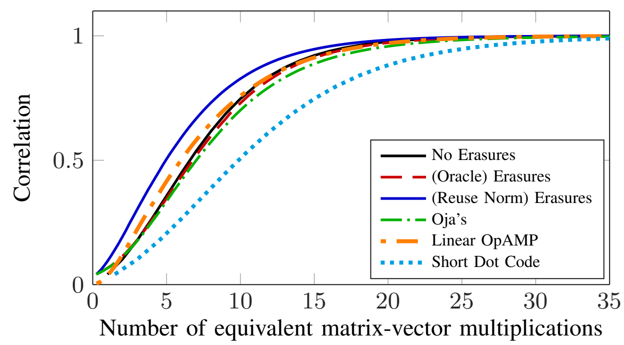

Fig. 4. Correlation vs. Complexity, $n = 1000, m = 750$

VII. Conclusions and Future Work

This paper analyzed a variation on the power method where a random subset of rows of the data matrix are erased at each iteration. Our bounds show that, given a good initialization, convergence to the principal eigenvector is exponentially fast (with high probability) at a rate governed by a modified spectral gap, which is a function of the original spectral gap and the erasure ratio. This convergence rate was validated numerically and it was shown that our proposed erasure-tolerant power method is competitive with respect to other approaches such as coded computing and Oja's algorithm.

Future lines of work should relax or eliminate the "warm start" condition in (11). Our simulations demonstrate that randomly sampling v_0 on the sphere is sufficient for convergence at rate $\tilde{\Delta}^2$. Furthermore, theoretical guarantees for the data-dependent update in (4) should be developed. While the oracle update is simpler to analyze, ultimately the data-dependent version is more practical to run, since it does not require knowledge of λ_1. Finally, our comparison with Oja's algorithm suggests possible improvements can be made to similar stochastic iterative algorithms operating in the random row-erasure setting discussed in this work.

APPENDIX

This appendix includes supporting lemmas used to prove the second moment bounds on Y_t and W_t. Owing to space constraints, the proofs are deferred to an extended version.

Lemma 3. Let $Q \in \mathbb{R}^{n\times n}$ be a PSD matrix that commutes with A. Define the following:

$$\omega_0 = 1 - 2\frac{m}{n} \qquad \omega_\ell = \frac{m}{n} \tag{41}$$
$$\omega_1 = \left(\frac{n-m}{n-1}\right)\left(\frac{m}{n}\right) \qquad \omega_2 = \left(\frac{m-1}{n-1}\right)\left(\frac{m}{n}\right)$$

If $A_t \sim \text{Unif}(\mathcal{D}_m)$ and taking $P_\ell = e_\ell e_\ell^\top$:

$$\begin{aligned}\mathbb{E}\left[A_t Q A_t\right] &= \omega_0 AQA + 2\omega_\ell \lambda_1 AQ \\ &+ \omega_2(\lambda_1 I - A)Q(\lambda_1 I - A) \\ &+ \omega_1(\lambda_1 I - A)\left(\sum_{\ell=1}^{n} P_\ell Q P_\ell\right)(\lambda_1 I - A)\end{aligned} \tag{42}$$

Corollary 1. Suppose $\bar{A}_t \sim \frac{1}{\lambda_1}\text{Unif}(\mathcal{D}_m)$. Then:

$$\mathbb{E}\left[\bar{A}_t Q \bar{A}_t\right] = \frac{1}{\lambda_1^2}\mathbb{E}\left[A_t Q A_t\right]$$

Lemma 4. Consider the matrix:

$$B^{(0)} = (I-A)^2 \tag{43}$$

with eigenvalues $\eta_i^{(0)} = (1-\Delta_i)^2$ and eigenvectors $v_1 \dots v_n$, where $\Delta_i = \frac{\lambda_i}{\lambda_1}$. Define the recursive update:

$$G^{(t)} = \mathbb{E}\left[\bar{A}B^{(t-1)}\bar{A}\right] \tag{44}$$

where $\bar{A} \sim \frac{1}{\lambda_1}\text{Unif}(\mathcal{D}_m)$. There exists a PSD matrix $B^{(t)} \in \mathbb{R}^{n\times n}$ that commutes with A, and $G^{(t)} \preceq B^{(t)}$. The eigenvalues of $B^{(t)}$ are defined recursively in (36).

Lemma 5. Let $\mathbb{E}\left[R_t\right]$ be the residual term defined by (33). Then:

$$\mathbb{E}\left[R_t\right] \leq v_*^2 \cdot v_0^\top B^{(t-1)} v_0 \tag{45}$$

where $B^{(t-1)}$ is the PSD matrix with eigendecomposition $\{\eta_i^{(t-1)}, v_i\}$, where the eigenvalues are defined by the recursion (36). *Proof*: Recall the definition of $\mathbb{E}\left[R_t\right]$:

$$\mathbb{E}\left[R_t\right] = \mathbb{E}\left[v_0^\top \bar{Z}_{t-1}^T B \bar{Z}_{t-1} v_0\right]$$

Where B is defined in (29) and $\bar{Z}_{t-1} = \bar{A}_t\bar{A}_{t-1}\cdots\bar{A}_2\bar{A}_1$. Each $\bar{A}_\tau$ is drawn i.i.d from $\mathcal{D}_m$, so $\bar{Z}_{t-1}$ takes on any of the N^{t-1} possible matrix products with equal probability. We can directly compute the total expectation:

$$\mathbb{E}\left[R_t\right] = \frac{1}{N^{t-1}} v_0^\top \left\{\sum_k z_{t-1,k}^T B z_{t-1,k}\right\} v_0 \tag{46}$$

where $z_{t-1,}$ is a specific product of $(t-1)$ effective matrices, indexed by $k, 1 \leq k \leq N^{t-1}$. Suppose all instances of $z_{t-1,k}$ that agree in the first $(t-2)$ iterations are grouped together; that is, all products of the form $\{z_{t-1,k} = \tilde{A}_\ell z_{t-2,k'}\}$, where $1 \leq k' \leq N^{t-2}$ and $\tilde{A}_\ell \in \frac{1}{\lambda_1}\mathcal{D}_m$. Reindexing from $k' \to k$ and summing:

$$\mathbb{E}\left[R_t\right] = \frac{1}{N^{t-2}} v_0^\top \left[\sum_k z_{t-2,k}^T \left\{\frac{1}{N}\sum_{\ell=1}^{N}\tilde{A}_\ell^T B \tilde{A}_\ell\right\} z_{t-2,k}\right] v_0$$

Using Lemma 10, $B \preceq v_*^2 B^{(0)}$, where $B^{(0)} = (I-A)^2$. Then the inner sum can be upper bounded using $G^{(1)} = \mathbb{E}\left[\bar{A}B^{(0)}\bar{A}\right]$, for $\bar{A} \sim \frac{1}{\lambda_1}\text{Unif}(\mathcal{D}_m)$:

$$\mathbb{E}\left[R_t\right] \leq \frac{v_*^2}{N^{t-2}} v_0^\top \left[\sum_{k=1}^{N^{t-2}} z_{t-2,k}^T G^{(1)} z_{t-2,k}\right] v_0$$

Using Lemma 4 to upper bound $G^{(1)} \preceq B^{(1)}$:

$$\mathbb{E}\left[R_t\right] \leq v_*^2 \cdot \frac{1}{N^{t-2}} v_0^\top \left[\sum_k z_{t-2,k}^T B^{(1)} z_{t-2,k}\right] v_0$$

Where $B^{(1)}$ commutes with A and is defined by the spectrum $\left\{\eta_i^{(1)}\right\}$ in (36). This process is repeated a total of $t-1$ times, yielding the desired matrix $B^{(t-1)}$. At each unfolding step, $B^{(\tau-1)}$ is updated to $B^{(\tau)}$ with the eigenvalue dynamics $\eta_i^{(\tau-1)} \to \eta_i^{(\tau)}$ described by (36).

Lemma 6. Let $\eta_i^{(t)}$ be defined according to the recursion in (36). Then:

$$\sum_{\tau=0}^{t-1} \eta_i^{(\tau)} \leq \left(\frac{n}{n-m}\right)\left(\frac{1}{1-\Delta^2}\right) \tag{47}$$

Proof: Using (36), we recall the definition of $\eta_i^{(t)}$:

$$\begin{aligned}\eta_i^{(t)} &= c_i \eta_i^{(t-1)} + \omega_1 \eta_*^{(t-1)}(1-\Delta_i)^2 \\ &= c_i \eta_i^{(t-1)} + \omega_1 (1-\Delta_i)^2 \max_i \eta_i^{(t-1)}\end{aligned}$$

Using Lemma 7, we can bound the sum:

$$\begin{aligned}\sum_{\tau=0}^{t-1} \eta_i^{(\tau)} &\leq \sum_{\tau=0}^{t-1}\left(\max_i c_i + \omega_1(1-\Delta_i^2)\right)^\tau \\ &\leq \sum_{\tau=0}^{t-1} c_*^\tau \leq \left(\frac{1}{1-c_*}\right)\end{aligned}$$

where $c_* = \max_i c_i + \omega_1(1-\Delta_i^2) \leq \tilde{\Delta}_2^2$. Using $\tilde{\Delta}_2^2 = (1-\frac{m}{n})\Delta^2 + \frac{m}{n}$, we arrive at the target bound:

$$\begin{aligned}\sum_{\tau=0}^{t-1} \eta_i^{(\tau)} &\leq \frac{1}{1-c_*} \leq \frac{1}{1-\left(1-\frac{m}{n}\right)\Delta^2 - \frac{m}{n}} \\ &\leq \frac{1}{\left(1-\frac{m}{n}\right)\left(1-\Delta^2\right)} \\ &\leq \left(\frac{n}{n-m}\right)\left(\frac{1}{1-\Delta^2}\right)\end{aligned}$$

Lemma 7. Define $\eta_i^{(t)}$ by the recursion (36). Then:

$$\eta_i^{(t)} \leq \left(\max_i c_i + \omega_1(1-\Delta_i^2)\right)^t, \ 2 \leq i \leq n \tag{48}$$

Proof: Use induction. For the base case $(t-1)$:

$$\begin{aligned}\eta_i^{(1)} &= c_i\eta_i^{(0)} + \omega_1(1-\Delta_i)^2\eta_*^{(0)} \\ &\leq \left(\max_i c_i\eta_i^{(0)} + \omega_1(1-\Delta_i)^2\eta_*^{(0)}\right) \\ &\leq \left(\max_i c_i + \omega_1(1-\Delta_i)^2\right)\eta_*^{(0)}\end{aligned}$$

Using the initial condition $\eta_*^{(0)} = 1$ proves the base case. For the induction step, we assume $\eta_i^{(t)} \leq \left(\max_i c_i + \omega_1(1-\Delta_i)^2\right)^t, \ 2 \leq i \leq n$. Then:

$$\begin{aligned}\eta_i^{(t+1)} &= c_i\eta_i^{(t)} + \omega_1(1-\Delta_i)^2\eta_{i*}^{(t)} \\ &\leq \left(\max_i c_i\eta_i^{(t)} + \omega_1(1-\Delta_i)^2\eta_{i*}^{(t)}\right) \\ &\leq \left(\max_i c_i + \omega_1(1-\Delta_i)^2\right)\eta_{i*}^{(t)} \\ &\leq \left(\max_i c_i + \omega_1(1-\Delta_i)^2\right)^{t+1}\end{aligned}$$

Lemma 8. Let $\mathcal{K} \subset [n], \ |\mathcal{K}| = m$ and let $\mathcal{P}_\mathcal{K} = \sum_{k\in\mathcal{K}} e_k e_k^\top$ be the projector onto the coordinates in $\mathcal{K}$. Then, with $N = \binom{n}{m}$:

$$\frac{1}{N}\sum_{\substack{\mathcal{K}\subset[n] \\ |\mathcal{K}|=m}} \mathcal{P}_\mathcal{K} = \left(\frac{m}{n}\right) I \tag{49}$$

Lemma 9. Let $\mathcal{K} \subset [n], \ |\mathcal{K}| = m$ and let $\mathcal{P}_\mathcal{K} = \sum_{k\in\mathcal{K}} e_k e_k^\top$ be the projector onto the coordinates in $\mathcal{K}$. Let $Q \in \mathbb{R}^{n\times n}$ be a PSD matrix. Then with $N = \binom{n}{m}$:

$$\frac{1}{N}\sum_{\substack{\mathcal{K}\subset[n] \\ |\mathcal{K}|=m}} \mathcal{P}_\mathcal{K} Q \mathcal{P}_\mathcal{K} = \omega_1\left[\sum_{\ell=1}^n P_\ell Q P_\ell\right] + \omega_2 Q \tag{50}$$

where $\omega_1 = \left(\frac{n-m}{n-1}\right)\left(\frac{m}{n}\right)$ and $\omega_2 = \left(\frac{m-1}{n-1}\right)\left(\frac{m}{n}\right)$.

Lemma 10. Let $B \in \mathbb{R}^{n\times n}$ be a PSD matrix. Let $b_* = \max_\ell(e_\ell^\top B e_\ell)$. Then for any $x \in \mathbb{R}^n$:

$$x^\top\left(\sum_{\ell=1}^n P_\ell B P_\ell\right)x \leq b_*\langle x, x\rangle \tag{51}$$

References

[1] J. Dean and L. A. Barroso, "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, 2013.

[2] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," *Advances In Neural Information Processing Systems*, vol. 29, 2016.

[3] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[4] Q. Yu, M. Maddah-Ali, and S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] S. Dutta, H. Jeong, Y. Yang, V. Cadambe, T. M. Low, and P. Grover, "Addressing unreliability in emerging devices and non-von Neumann architectures using coded computing," *Proceedings of the IEEE*, vol. 108, no. 8, pp. 1219–1234, 2020.

[6] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numerica*, vol. 29, pp. 403–572, 2020.

[7] S. Li, S. Avestimehr *et al.*, "Coded computing: Mitigating fundamental bottlenecks in large-scale distributed computing and machine learning," *Foundations and Trends® in Communications and Information Theory*, vol. 17, no. 1, pp. 1–148, 2020.

[8] H. Jeong, A. Devulapalli, V. R. Cadambe, and F. P. Calmon, "ϵ-approximate coded matrix multiplication is nearly twice as efficient as exact multiplication," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 845–854, 2021.

[9] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," *Advances in neural information processing systems*, vol. 27, 2014.

[10] P. Xu, B. He, C. De Sa, I. Mitliagkas, and C. Re, "Accelerated stochastic power iteration," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 58–67.

[11] Z. Xu and P. Li, "Faster noisy power method," in *International Conference on Algorithmic Learning Theory*. PMLR, 2022, pp. 1138–1164.

[12] Q. Lei, K. Zhong, and I. S. Dhillon, "Coordinate-wise power method," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[13] S. Shin, H. Zhao, and I. Shomorony, "Adaptive power method: Eigenvector estimation from sampled data," in *International Conference on Algorithmic Learning Theory*. PMLR, 2023, pp. 1387–1410.

[14] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of mathematical biology*, vol. 15, pp. 267–273, 1982.

[15] O. Shamir, "A stochastic PCA and SVD algorithm with an exponential convergence rate," in *International conference on machine learning*. PMLR, 2015, pp. 144–152.

[16] P. Jain, C. Jin, S. M. Kakade, P. Netrapalli, and A. Sidford, "Streaming PCA: Matching matrix Bernstein and near-optimal finite sample guarantees for Oja's algorithm," in *Conference on learning theory*. PMLR, 2016, pp. 1147–1164.

[17] K. Bhatia, A. Pacchiano, N. Flammarion, P. L. Bartlett, and M. I. Jordan, "Gen-Oja: Simple & efficient algorithm for streaming generalized eigenvector computation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[18] O. Y. Feng, R. Venkataramanan, C. Rush, R. J. Samworth *et al.*, "A unifying tutorial on approximate message passing," *Foundations and Trends® in Machine Learning*, vol. 15, no. 4, pp. 335–536, 2022.

[19] R. Rossetti, B. Nazer, and G. Reeves, "Linear operator approximate message passing (OpAMP)," *arXiv preprint arXiv:2405.08225*, 2024.