Communication-Efficient Distributed SVD via Local Power Iterations

Xiang Li [1] **Shusen Wang** [2] **Kun Chen** [1] **Zhihua Zhang** [1]

Abstract

We study distributed computing of the truncated singular value decomposition problem. We develop an algorithm that we call `LocalPower` for improving communication efficiency. Specifically, we uniformly partition the dataset among m nodes and alternate between multiple (precisely p) local power iterations and one global aggregation. In the aggregation, we propose to weight each local eigenvector matrix with orthogonal Procrustes transformation (OPT). As a practical surrogate of OPT, sign-fixing, which uses a diagonal matrix with ± 1 entries as weights, has better computation complexity and stability in experiments. We theoretically show that under certain assumptions `LocalPower` lowers the required number of communications by a factor of p to reach a constant accuracy. We also show that the strategy of periodically decaying p helps obtain high-precision solutions. We conduct experiments to demonstrate the effectiveness of `LocalPower`.

1. Introduction

In this paper we consider the truncated singular value decomposition (SVD) which has broad applications in machine learning, such as dimension reduction (Wold et al., 1987), matrix completion (Candès & Recht, 2009), and information retrieval (Deerwester et al., 1990). Let $\mathbf{a}_1, \cdots, \mathbf{a}_n \in \mathbb{R}^d$ be sampled i.i.d. from some fixed but unknown distribution. The goal is to compute the k ($k < \min\{d, n\}$) singular vectors of $\mathbf{A} \triangleq [\mathbf{a}_1, \ldots, \mathbf{a}_n]^\top \in \mathbb{R}^{n \times d}$. Let $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ contain the top k singular vectors. The power iteration and its variants such as Krylov subspace iterations are common approaches to the truncated SVD. They have $\mathcal{O}(nd)$ space complexity and $\mathcal{O}(ndk)$ per-iteration time complexity. They take $\tilde{\mathcal{O}}(\log \frac{d}{\epsilon})$ iterations to converge to ϵ precision, where $\tilde{\mathcal{O}}$ hides the spectral gap and constants (Golub & Van Loan, 2012; Saad, 2011).

When either n or d is big, the data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ may not fit in the memory, making standard single-machine algorithms infeasible. A distributed power iteration is feasible and practical for large-scale truncated SVDs. In particular, we partition the rows of $\mathbf{A}$ among m worker nodes (see Figure 1(a)) and let the nodes perform power iterations in parallel (see Figure 1(b)). In every iteration, every node performs $\mathcal{O}(\frac{ndk}{m})$ FLOPs (suppose the load is balanced), while the server performs only $\mathcal{O}(dk^2)$ FLOPs.

When solving large-scale matrix computation problems, communication costs are not negligible; in fact, communication costs can outweigh computation costs. The large-scale SVD experiments in (Gittens et al., 2016; Wang et al., 2019) show that the runtime caused by communication and straggler's effect[1] can exceed the computation time. Due to the communication costs and other overheads, parallel computing can even demonstrate anti-scaling; that is, when m is big, the overall wall-clock runtime increases with m. Reducing the frequency of communications will reduce the communication and synchronization costs and thereby improving the scalability.

1.1. Our Contributions

Inspired by the `FedAvg` algorithm (McMahan et al., 2017), we propose an algorithm called `LocalPower` to improve communication-efficiency. `LocalPower` is based on the distributed power iteration (`DPI`) described in Figure 1. The difference is that `LocalPower` makes every node locally perform orthogonal iterations using its own data for p iterations. In the case for $p = 1$, `LocalPower` degenerates to `DPI`. When $p \geq 2$, local updates are employed to reduce communication frequency.

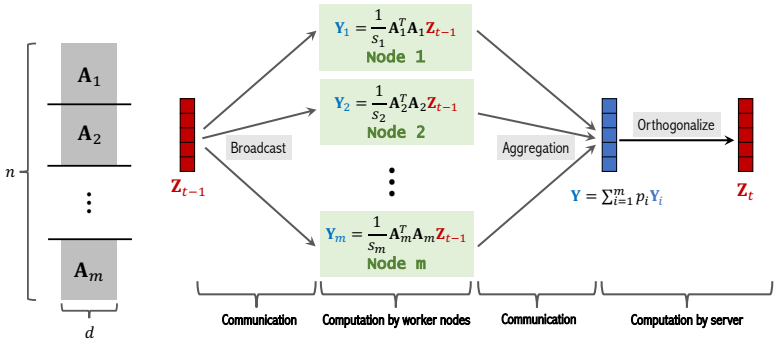
In practice, a naive implementation of the proposed `LocalPower` does not work very well. We propose three effective techniques for improving `LocalPower`:

- We propose to decay the communication interval, p, over time. In this way, the loss drops fast in the beginning and converge to the optimal solution in the end. Without the decay strategy, `LocalPower` is not

[1]School of Mathematical Sciences, Peking University, China [2]Department of Computer Science, Stevens Institute of Technology, USA. Correspondence to: Xiang Li <lx10077@pku.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Straggler's effect means that one outlier node is tremendously slower than the rest, and the system waits for the slowest to complete.



(a) Partition. (b) Standard parallel power iteration (`DPI`).

Notation	Definition
$\mathbf{A}$	data matrix
n	number or rows
d	number of columns
ρ	rank of data matrix
m	number of partitions
k	target rank of truncated SVD
r	running columns
p	number of local iterations
s_i	number of rows in i-th node
$p_i = \frac{s_i}{n}$	fraction of rows in i-th node

(c) Commonly used symbols

Figure 1. (a) The $n \times d$ data matrix $\mathbf{A}$ is partitioned among m worker nodes. (b) In every iteration of the distributed power iteration, there are two rounds of communications. Most of the computations are performed by the worker nodes. (c) Commonly used symbols.

guaranteed to converge to the optimum.

- Orthogonal Procrustes transformation (OPT) post-processes the output matrices of the m nodes after each iteration so that the m matrices are close to each other. OPT makes `LocalPower` stable at the cost of more computation.

- To reduce the computation of OPT, we replace its orthogonal space to the set of all diagonal matrices with ± 1 entries. In this way, OPT becomes the sign-fixing technique which is stable (slightly worse than OPT) and efficient. Sign fixing was originally proposed by Garber et al. (2017) for the special case of $k = 1$, while we generalize sign-fixing to $k > 1$.

In summary, this work's contributions include the new algorithm, `LocalPower`, its convergence analysis, and the effective techniques for improving `LocalPower`.

The remainder of this paper is organized as follows. In Section 2, we define notation and give preliminary backgrounds on the orthogonal Procrustes problem and the distributed power iteration. In Section 3, we propose `LocalPower` and its variants and then provide theoretical analysis in Section 4. In Section 5, we conduct experiments to illustrate the effectiveness of `LocalPower` and to validate our theoretical results. In Section 6, we give further discussions on some aspects of `LocalPower`. All proof details can be found at Appendix A. In Appendix D, we discuss related work on SVD and parallel algorithms.

2. Preliminary

Notation. For any $\mathbf{A} \in \mathbb{R}^{n\times d}$, we use $\|\mathbf{A}\|_2$ and $\|\mathbf{A}\|_F$ to denote its spectral norm and Frobenius norm. Let $\mathbf{A}^\dagger \in \mathbb{R}^{d\times n}$ denote the Moore-Penrose pseudo-inverse of $\mathbf{A}$. For any positive integer T, let $[T] = \{1, 2, \cdots, T\}$. $\mathcal{O}_{d\times k}$ is the set of all $d \times k$ column orthonormal matrices ($1 \le k \le d$). $\mathcal{O}_k$, short for $\mathcal{O}_{k\times k}$, denotes the set of $k \times k$ orthogonal matrices. $\mathcal{R}(\mathbf{U})$ denotes the subspace spanned by the columns of $\mathbf{U}$. The commonly used notation is summarized in Figure 1(c).

Power iteration. The top k right singular vectors of $\mathbf{A}$ can be obtained by the subspace iteration that repeats

$$\mathbf{Y} \longleftarrow \mathbf{MZ} \quad \text{and} \quad \mathbf{Z} \longleftarrow \mathsf{orth}\big(\mathbf{Y}\big), \tag{1}$$

where $\mathbf{M} = \frac{1}{n}\mathbf{A}^\top\mathbf{A}$. In every power iteration, computing $\mathbf{Y}$ has $\mathcal{O}(ndk)$ time complexity, and orthogonalizing $\mathbf{Y}$ has $\mathcal{O}(dk^2)$ time complexity. It is well known that the tangent of principle angles between $\mathcal{R}(\mathbf{Z})$ and $\mathcal{R}(\mathbf{U}_k)$ converges to zero geometrically (Arbenz et al., 2012; Saad, 2011) and thus so their projection distance.

Distributed power iteration (`DPI`) is a direct distributed variant of power iteration. Consider data parallelism and partition the data (rows of $\mathbf{A}$) among m worker nodes. See Figure 1(a) for the illustration. We partition $\mathbf{A}$ as $\mathbf{A}^\top = [\mathbf{A}_1^\top, \cdots, \mathbf{A}_m^\top]$ where $\mathbf{A}_i \in \mathbb{R}^{s_i\times d}$ contains s_i rows of $\mathbf{A}$. Using m worker nodes and data parallelism, one power iteration works in four steps. First, the server broadcasts $\mathbf{Z}$ to the workers, which has $\mathcal{O}(dk)$ or $\mathcal{O}(dkm)$ communication complexity (depending on the network structure). Second, every worker (say, the i-th) locally computes

$$\mathbf{Y}_i = \mathbf{M}_i\mathbf{Z} \in \mathbb{R}^{d\times k} \quad \text{with} \quad \mathbf{M}_i = \frac{1}{s_i}\mathbf{A}_i^\top\mathbf{A}_i, \tag{2}$$

which has $\mathcal{O}(d^2k)$ or $\mathcal{O}(s_idk)$ time complexity. Third, the server aggregates $\mathbf{Y}_i$, for all $i \in [m]$, to obtain $\mathbf{Y} = \sum_{i=1}^m p_i\mathbf{Y}_i$; this step is equivalent to $\mathbf{Y} = \mathbf{MZ}$, where $\mathbf{M} = \sum_{i=1}^m p_i\mathbf{M}_i$ with $p_i = \frac{s_i}{n}$. It has $\mathcal{O}(dk)$ or $\mathcal{O}(dkm)$ communication complexity. Last, the server locally orthogonalizes $\mathbf{Y}$ to obtain $\mathbf{Z} = \mathsf{orth}(\mathbf{Y})$, which has merely $\mathcal{O}(dk^2)$

Algorithm 1 `LocalPower`

1: **Input:** distributed dataset $\{\mathbf{A}_i\}_{i=1}^m$, target rank k, iteration rank $r \geq k$, number of iterations T.
2: **Initialization:** generate a standard Gaussian matrix, $\mathbf{Y}_0$;
3: **for** $t = 0$ **to** T **do**
4: **Broadcast:** If $t \in \mathcal{I}_T$, the server sends $\mathbf{Y}_t$ to workers; let $\mathbf{Y}_t^{(i)} \leftarrow \mathbf{Y}_t$;
5: **Local computation:** For all $i \in [m]$, the i-th worker locally computes
$$\mathbf{Z}_t^{(i)} = \mathsf{orth}(\mathbf{Y}_t^{(i)}) \quad \text{and} \quad \mathbf{Y}_{t+1}^{(i)} = \tfrac{1}{s_i}\mathbf{A}_i^\top \mathbf{A}_i \mathbf{Z}_t^{(i)};$$
6: **Aggregation:** If $(t+1) \in \mathcal{I}_T$, the server computes $\mathbf{Y}_{t+1} = \sum_{i=1}^m p_i \mathbf{Y}_{t+1}^{(i)}$;
7: **end for**
8: **Output:** $\mathsf{orth}(\mathbf{Y}_{t+1})$.

time complexity. The algorithm is described in Figure 1(b). The following lemma is a well-known result (Arbenz et al., 2012; Saad, 2011).

Lemma 1. *To obtain a column-orthonormal matrix $\mathbf{Z}$ such that the subspace distance $dist(\mathbf{Z}, \mathbf{U}_k) \leq \epsilon$ (see Definition 1 for detail), with high probability, the communication needed by `DPI` is*

$$\Omega\left(\frac{\sigma_k}{\sigma_k - \sigma_{k+1}} \log\left(\frac{d}{\epsilon}\right)\right). \tag{3}$$

Here, σ_j is the j-th largest singular value of the matrix $\mathbf{M}$.

3. Algorithms

`LocalPower` is a new algorithm that we propose for improving communication efficiency. It is described in Algorithm 1. Its basic idea is to trade more local power iterations for fewer communications via reducing the communication frequency. Between two communications, every worker node locally runs eqn. (2) for multiple times. We let the set $\mathcal{I}_T$ ($\subseteq [T]$) index the iterations that perform communications; for example,

$$\mathcal{I}_T \;=\; \{0, p, 2p, \cdots, T\}$$

means that the algorithm communicates once after p lower power iterations. The cardinality $|\mathcal{I}_T|$ is the total number of communications.

Suppose `LocalPower` performs one communication every p iterations. In T iterations, each worker performs $\mathcal{O}(s_i dkT)$ FLOPs, the server performs $\mathcal{O}(dk^2T/p)$ FLOPs, and the overal communication complexity is $\mathcal{O}(dkT/p)$. The standard distributed power iteration is a special case of `LocalPower` with $p = 1$, that is, $\mathcal{I}_T = \{0, 1, 2, \cdots, T\}$.[2]

[2]The reason why we average $\mathbf{Y}_t^{(i)}$ instead of $\mathbf{Z}_t^{(i)}$ is that we hope `LocalPower` is reduced to `DPI` when $p = 1$.

One-shot SVD, aka divide-and-conquer SVD, (Liang et al., 2014; Garber et al., 2017; Fan et al., 2019), is a special case of `LocalPower` with $p = T$, that is, $\mathcal{I}_T = \{0, T\}$.

Decaying p. In practice, it is helpful to use a big p in the beginning but let $p = 1$ in the end. For example, we can decrease p by half every few communications. The rationale is that the error of `LocalPower` does not converge to zero if p is big; see the theoretical analysis in the next section. Our empirical observation confirms the theories: if p is set big, then the error drops very fast in the beginning, but it does not vanish with the iterations.

Orthogonal Procrustes Transformation. In Algorithm 1, the i-th worker locally computes

$$\mathbf{Y}_{t+1}^{(i)} \;=\; \tfrac{1}{s_i}\mathbf{A}_i^\top \mathbf{A}_i \mathbf{Z}_t^{(i)}.$$

When it comes to the time of communication (i.e., $t+1 \in \mathcal{I}_T$), we replace the equation by the following steps. First, we choose the device which has the maximum number of samples as a base. Without loss of generality, we can assume the first device is selected (which indicates $1 = \mathrm{argmin}_{i\in[m]} p_i$). Second, we compute

$$\mathbf{O}_t^{(i)} \;=\; \operatorname*{argmin}_{\mathbf{O}\in\mathcal{O}_k} \left\|\mathbf{Z}_t^{(i)}\mathbf{O} - \mathbf{Z}_t^{(1)}\right\|_F^2. \tag{4}$$

Eqn. (4) is a classic matrix approximation problem in linear algebra, named as the Procrustes problem (Schönemann, 1966; Cape, 2020). The solution to eqn. (4) is referred to as orthogonal Procrustes transformation (OPT) and has a closed form:

$$\mathbf{O}_t^{(i)} \;=\; \mathbf{W}_1\mathbf{W}_2^\top,$$

where $\mathbf{W}_1\mathbf{\Sigma}\mathbf{W}_2^\top$ is the SVD of $(\mathbf{Z}_t^{(i)})^\top \mathbf{Z}_t^{(1)}$. Finally, we compute

$$\mathbf{Y}_{t+1}^{(i)} \;=\; \tfrac{1}{s_i}\mathbf{A}_i^\top \mathbf{A}_i \mathbf{Z}_t^{(i)} \mathbf{O}_t^{(i)}.$$

Remak 1. *Intuitively, such $\mathbf{O}_t^{(i)}$ adjusts $\mathbf{Z}_t^{(i)}$ such that it aligns with $\mathbf{Z}_t^{(1)}$ better. In an ideal case, all $\mathbf{Z}_t^{(i)}$'s would be identical with $\mathbf{Z}_t^{(1)}$ and thus the aggregation step (line 6 in Algorithm 1) would be the same as that in `DPI`. From our theory, it is important to use OPT. It weakens the assumption on the smallness of a residual error which is incurred by local computation. From our experiments, it stabilizes vanilla `LocalPower` and achieves much smaller errors.*

Remak 2. *To compute such $\mathbf{O}_t^{(i)}$, the i-th client should communicate $\mathbf{Z}_t^{(i)}$ to the server, which results in additional communication cost. However, the cost is the same in magnitude as that of sending $\mathbf{Y}_{t+1}^{(i)}$ in the aggregation step. Besides, the computation of $\mathbf{O}_t^{(i)}$ as well as the communication of $\mathbf{Y}_{t+1}^{(i)}$ only happens when $t+1 \in \mathcal{I}_T$. These make the additional communication cost affordable.*

Sign-Fixing. While OPT makes `LocalPower` more stable in practice, OPT incurs more local computation. Specifically, it has time complexity $\mathcal{O}(dk^2)$ via calling the SVD of $(\mathbf{Z}_t^{(i)})^\top \mathbf{Z}_t^{(1)}$. To attain both efficiency and stability, we propose to replace the $k \times k$ matrix $\mathbf{O}^{(i)}$ in eqn. (4) by

$$\mathbf{D}_t^{(i)} \;=\; \operatorname*{argmin}_{\mathbf{D}\in\mathcal{D}_k} \big\|\mathbf{Z}_t^{(i)}\mathbf{D} - \mathbf{Z}_t^{(1)}\big\|_F^2, \tag{5}$$

where $\mathcal{D}_k$ denotes all the $k \times k$ diagonal matrices with ± 1 diagonal entries. $\mathbf{D}_t^{(i)}$ can be computed in $\mathcal{O}(kd)$ time by

$$\mathbf{D}_t^{(i)}[j,j] \;=\; \operatorname{sgn}\Big(\Big\langle \mathbf{Z}_t^{(i)}[:,j]\,,\, \mathbf{Z}_t^{(1)}[:,j]\Big\rangle\Big), \quad \forall\, j\in[k].$$

We empirically observe that sign-fixing serves as a good practical surrogate of OPT; it maintains good stability and achieves comparably small errors.

Remak 3. *If we decay p, p will drop to one after a few communications. When $p = 1$, we stop using OPT (or sign-fixing); we simply set $\mathbf{O}_t^{(i)}$ (or $\mathbf{D}_t^{(i)}$) to the identity matrix.*

Remak 4. *The technique of sign-fixing has been proposed in the setting of $k = 1$ by Garber et al. (2017). In the $k = 1$ setting, OPT and sign-fixing coincide with each other. In eqn. (5), we provide a simple way to extend it to high-dimensional $k > 1$. We compute $\mathbf{D}_t^{(i)}$ that simultaneously adjusts the signs of columns of $\mathbf{Z}_t^{(i)}$ and $\mathbf{Z}_t^{(1)}$. There exists other way to handle the high-dimensional sign-fixing problem. For example, if first k eigenvalues are well-separated from others, we can reduce the top-k sign-fixing problem to the one-dimensional sign-fixing problem instanced k times.*

4. Convergence Analysis

In this section we analyze the convergence of `LocalPower` and show the benefit of OPT under an ideal setting. We use the projection distance of two subspaces as the metric for convergence evaluation.

Definition 1 (Projection Distance)**.** *Let $\mathbf{U}, \widetilde{\mathbf{U}} \in \mathcal{O}_{d\times k}$ be any matrices with orthonormal columns. The projection distance[3] between them is*

$$\operatorname{dist}(\mathbf{U}, \widetilde{\mathbf{U}}) \;\triangleq\; \big\|\mathbf{U}\mathbf{U}^\top - \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top\big\|_2.$$

Projection distance is equivalent to $\operatorname{dist}(\mathbf{U}, \widetilde{\mathbf{U}}) = \sin\theta_k(\mathbf{U}, \widetilde{\mathbf{U}})$ where $\theta_k(\mathbf{U}, \widetilde{\mathbf{U}})$ denotes the k-th largest principal angle between the subspaces spanned by $\mathbf{U}$ and $\widetilde{\mathbf{U}}$. Principal angles quantify how different two subspaces are. We can actually calculate

$$\theta_1\big(\mathbf{U}, \widetilde{\mathbf{U}}\big),\;\; \theta_2\big(\mathbf{U}, \widetilde{\mathbf{U}}\big),\;\; \cdots,\; \theta_k\big(\mathbf{U}, \widetilde{\mathbf{U}}\big)$$

via the SVD of $\mathbf{U}^\top\widetilde{\mathbf{U}}$. The l-th largest singular value of $\mathbf{U}^\top\widetilde{\mathbf{U}}$ is equal to $\cos\theta_l(\mathbf{U}, \widetilde{\mathbf{U}})$ for all $l = 1, \cdots, k$.

Definition 2 (Local Approximation)**.** *Let $\mathbf{M}_i = \frac{1}{s_i}\mathbf{A}_i^\top\mathbf{A}_i$ be hosted by the i-th worker. Let $\mathbf{M} = \frac{1}{n}\sum_{i=1}^m \mathbf{A}_i^\top\mathbf{A}_i = \sum_{i=1}^m p_i\mathbf{M}_i$. Define*

$$\eta \;\triangleq\; \max_{i\in[m]} \frac{\|\mathbf{M}_i - \mathbf{M}\|_2}{\|\mathbf{M}\|_2},$$

which measures how far the local matrices, $\mathbf{M}_1, \cdots, \mathbf{M}_m$, are from $\mathbf{M}$. If $s_i = p_i n$ is sufficiently larger than d, then η is sufficiently small.

Definition 3 (Residual Error)**.** *If OPT is not used, define*

$$\rho_t \;\triangleq\; \max_{i\in[m]} \big\|\mathbf{Z}_t^{(i)} - \mathbf{Z}_t^{(1)}\big\|_2.$$

If OPT is used, define

$$\rho_t \;\triangleq\; \max_{i\in[m]} \min_{\mathbf{O}\in\mathcal{O}_k} \big\|\mathbf{Z}_t^{(i)}\mathbf{O} - \mathbf{Z}_t^{(1)}\big\|_2.$$

The residual error ρ_t measures how the local top-k eigenspace estimator varies across the m worker. Based on the definition, using OPT makes ρ_t smaller than without using OPT. When $t \in \mathcal{I}_T$, $\mathbf{Z}_t^{(1)} = \cdots = \mathbf{Z}_t^{(m)}$ and thus $\rho_t = 0$. When $t \notin \mathcal{I}_T$, each local update would enlarge ρ_t. Hence, intuitively ρ_t depends on p, i.e., the local iterations between two communications. However, later we will show that with OPT ρ_t does not depend on p (when p is sufficiently large) while it depends on p without OPT. A residual error is inevitable in previous literature of empirical risk minimization that uses local updates to improve communication efficiency (Stich, 2018; Wang & Joshi, 2018b; Yu et al., 2019; Li et al., 2019a;b). In our case, it takes the form of ρ_t.

Assumption 1 (Uniformly small residual errors)**.** *Let r be the running column number, σ_j be the j-th largest singular value of $\mathbf{M}$, and $\epsilon \in (0, 0.5)$ be a constant. Assume $\eta \le \frac{1}{3\kappa}$ where $\kappa = \|\mathbf{M}\|\|\mathbf{M}^\dagger\|$ is the condition number of $\mathbf{M}$. Assume for all $t \in [T]$,*

$$\eta\cdot 1_{t\notin\mathcal{I}_T} + (1-p_{\max})(\rho_t + \rho_{t-1}1_{t\notin\mathcal{I}_T}) \;=\; \mathcal{O}(\epsilon_0), \tag{6}$$

where $p_{\max} = \max_{i\in[m]} p_i$, $1_{t\notin\mathcal{I}_T}$ is the indication function of the event $\{t \notin \mathcal{I}_T\}$, and

$$\epsilon_0 \;\triangleq\; \frac{\sigma_k - \sigma_{k+1}}{\sigma_1\kappa}\min\left\{\frac{\sqrt{r}-\sqrt{k-1}}{\tau\sqrt{d}}, \epsilon\right\}$$

for some small constant $\tau > 0$.

Theorem 1 (Convergence for `LocalPower`)**.** *Let τ be a positive constant, and Assumption 1 hold. Then, after $|\mathcal{I}_T|$ rounds of communication where*

$$T = \Omega\left(\frac{\sigma_k}{\sigma_k - \sigma_{k+1}}\log\left(\frac{\tau d}{\epsilon}\right)\right),$$

[3] Unlike the spectral norm or the Frobenius norm, the projection norm will not fall short of accounting for global orthonormal transformation. Check Ye & Lim (2014) to find more information about distance between two spaces.

with probability at least $1-\tau^{-\Omega(r+1-k)}-e^{-\Omega(d)}$*, we have*

$$\operatorname{dist}(\mathbf{Z}_T, \mathbf{U}_k) = \sin\theta_k(\mathbf{Z}_T, \mathbf{U}_k) \le \epsilon.$$

Theorem 1 shows `LocalPower` takes $T = \widetilde{\Theta}\left(\frac{\sigma_k}{\sigma_k-\sigma_{k+1}}\right)$ iterations to obtain an ϵ-optimal solution, the same quantity required by `DPI` . However, `LocalPower` uses less communications. For example, with $\mathcal{I}_T = \{0, p, 2p, \cdots, T\}$, `LocalPower` makes only $|\mathcal{I}_T| = \widetilde{\Theta}\left(\frac{1}{p}\frac{\sigma_k}{\sigma_k-\sigma_{k+1}}\right)$ communications, saving a factor of p than `DPI` .

Theorem 1 depends on Assumption 1 which requires eqn. (6) holds for all $t \in [T]$. What's more, the final error ϵ is positively related to η and ρ_t via eqn. (6). The first part of eqn. (6) (i.e., $\eta \cdot 1_{t\notin\mathcal{I}_T}$) is incurred by the variety of $\mathbf{M}_i$'s. So, if all devices have access to $\mathbf{M}$ (which implies $\mathbf{M}_1 = \cdots = \mathbf{M}_m$), then it would vanish. The second part eqn. (6) is brought by intermittent communication. Indeed, if communication happens at iteration t (i.e., $t \in \mathcal{I}_T$), we have $\rho_t = 0$ and $1_{t\notin\mathcal{I}_T} = 0$, implying eqn. (6) holds obviously. Without communication, ρ_t is likely to grow continually, which is harmful to obtaining an accurate solution. Therefore, the assumption actually requires the communication interval p is not too large. From another hand, when p is fixed, the assumption instead imposes restriction on η when $t \notin \mathcal{I}_T$, because we show in Theorem 2 that ρ_t is bounded by a function of η. If OPT is used, then $\rho_t = \mathcal{O}(\eta)$, without dependence on p. However, if OPT is not used, then $\rho_t = \mathcal{O}(\sqrt{k}p\kappa^p\eta)$ has an exponential dependence on p.

Theorem 2 (Benefits of OPT)**.** *Let* $\tau(t) \in \mathcal{I}_T$ *be the nearest communication time before* t *and* $p = t - \tau(t)$*. Let* e *be the natural constant. Assume* $\eta \le \min(\frac{1}{3\kappa}, \frac{1}{p})$*.*

- *With OPT,* ρ_t *is bounded by*

$$\min\left\{2\mathrm{e}^2\kappa^p p\eta, \frac{\eta\sigma_1}{\delta_k} + 2\gamma_k^{p/4}C_t\right\} = \mathcal{O}(\eta),$$

  where $\gamma_k \in (0,1)$*,* $\delta_k = \Theta(\sigma_k - \sigma_{k+1})$*, and* $\limsup_t C_t = \mathcal{O}(\eta+\epsilon)$*.*

- *Without OPT,* ρ_t *is bounded by*

$$4\mathrm{e}\sqrt{k}p\kappa^p\eta = \mathcal{O}(\sqrt{k}p\kappa^p\eta).$$

Why using OPT has such an exponential improvement on dependence on p in theory? This is mainly because of the property of OPT. Let $\mathbf{O}^* = \operatorname{argmin}_{\mathbf{O}\in\mathcal{O}_k}\|\mathbf{U} - \widetilde{\mathbf{U}}\mathbf{O}\|_F$ for $\mathbf{U}, \widetilde{\mathbf{U}} \in \mathcal{O}_{d\times k}$. Then, up to some universal constant, we have $\|\mathbf{U} - \widetilde{\mathbf{U}}\mathbf{O}^*\|_2 \approxeq \operatorname{dist}(\mathbf{U}, \widetilde{\mathbf{U}})$. See Lemma 3 in Appendix for a formal statement and detailed proof. It implies up to a tractable orthonormal transformation, the difference between the orthonormal bases of two subspaces is no larger than the projection distance between the subspaces. By the Davis-Kahan theorem (see Lemma 11), their projection distance is not larger than $\mathcal{O}(\eta)$ up to some problem-dependent constants. However, without OPT, we have to use perturbation theory to bound ρ_t, which inevitably results in exponential dependence on p.

5. Experiments

Settings. We use 15 datasets available on the LIBSVM website.[4] The n data samples are randomly shuffled and then partitioned among m nodes so that each node has $s = \frac{n}{m}$ samples. We set $m = \max(\lfloor\frac{n}{1000}\rfloor, 3)$ so that each node has $s = 1,000$ samples, unless n is too small. The features are normalized so that all the values are between -1 and 1. All the algorithms start from the same initialization $\mathbf{Y}_0$. We fix the target rank to $k = 5$. Our focus is on communication efficiency, so we use communication rounds for evaluating the compared algorithms. Due to the space limit, we defer more experiment details and additional experiment results to Appendix E.

Compared algorithms. We evaluate three variants of `LocalPower` : the vanilla version, with OPT, and with sign-fixing. We compare our algorithms with one-shot algorithms, UDA (Fan et al., 2019), WDA (Bhaskara & Wijewardena, 2019), and DR-SVD[5]; the algorithms are described in Appendix E.2.

Final precision. In this set of experiments, we study the precision when the algorithms converge. For three variants of `LocalPower` we fix $p = 4$ (without decaying p). We run each algorithm 10 times and report the mean and standard deviation (std) of the final errors. Due to limited space, Table 1 shows the results on 7 datasets. Table 6 and Figure 4 (in the appendix) present all the results on the 15 datasets. Out of the 15 datasets, `LocalPower` has the smallest error mean and std on 12 datasets. The results indicate that one-shot methods do not find high-precision solutions unless the local data size is sufficiently large.

The final error depends on p. With $p > 1$, the final error, $\lim_{t\to\infty}\sin\theta_k(\mathbf{Z}_t, \mathbf{U}_k)$, does not convergence to zero; instead, it remains to be a constant after a number of iterations. Figure 3(c) shows that the final error depends on p: the bigger p is, the bigger the final error is. The final error is not sensitive to p. The final error stops growing with p when p is sufficiently large. Note that `LocalPower` as $p \to \infty$ becomes a one-shot algorithm, that is, the algorithm performs only one aggregation.[6] One-shot algorithms typically have

[4] This page contains them all. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. See Table 4 in the Appendix for n, d information.

[5] It is a direct distributed variant of Randomized SVD, the latter proposed by Halko et al. (2011).

[6] The one-shot method is different from those we introduced in

reasonable empirical performance and theoretical bounds.

The final error depends on m. Big m means smaller local sample size, $s = \frac{n}{m}$, and thereby big matrix approximation error, η (in Definition 2). Our theory indicates that big m (and thereby big η) is bad for the final error. The empirical results in Figure 3(c) corroborate our theories.

Effect of local power iterations. In this set of experiments, we set p to 1, 2, 4, or 8 (without decaying p) and compare the convergence curves. Note that `LocalPower` with $p = 1$ is the standard distributed power iteration (`DPI`). We plot the error, $\sin\theta_k(\mathbf{Z}_t, \mathbf{U}_k)$, against communications. The convergence curves indicate how p affects the communication efficiency. Figure 2(a) shows the experimental results on one dataset. Due to page limit, the results on the other datasets are left to the appendix; see Figures 5, 6, and 7. In all the experiments, large p leads to fast convergence in the beginning but has a nonvanishing error in the end.

Some machine learning tasks, such as principal component analysis and latent semantic analysis (Deerwester et al., 1990), do not require high-precision solutions. In this case, `LocalPower` is advantageous over `DPI`, as `LocalPower` finds a satisfactory solution using very few communications. For two-stages methods like (Garber et al., 2017) It is also implied that `LocalPower` helps If a higher precision is required, we can decay p so that `LocalPower` will have the same precision as `DPI`. While one-shot algorithms are more communication-efficient, their precision is too low unless each node has a large sample size.

The decay strategy. We have observed that large p fastens initial convergence but enlarges the final error. By contrast, $p = 1$ has the lowest error (which actually can be zero) but also the lowest convergence rate. Similar phenomena have been previously observed in distributed empirical risk minimization (Wang & Joshi, 2018a; Li et al., 2019b). To allow for both fast initial convergence and vanishing final error, we are motivated to decay p gradually. We halve p every iteration until it reaches 1. We apply the decay strategy to the three variants of `LocalPower`. For each setting and each dataset, we repeat the experiment 10 times and report the mean and std. Table 2 and Figure 3(a) show the results on some datasets. The results on all the 15 datasets are left to the appendix; see Table 7, Figures 8, 9, and 10. The decay strategy not only makes convergence faster but also improves the final precision well.

Stability. In almost all the experiments, `LocalPower` with OPT has smaller std and more stable convergence curves than `LocalPower` without OPT. Why does OPT improve stability. Theorem 2 shows that with OPT, ρ_t (in Definition 3) has a linear function of p. Even if p is large, Assumption 1 can be satisfied, and thus Theorem 1 guarantees the convergence of `LocalPower` with OPT. However, Theorem 2 shows that without using OPT, ρ_t is an exponential function of p. If p is large, Assumption 1 is violated, and thus the convergence of `LocalPower` without OPT is not guaranteed.

Sign-fixing is practical alternative to OPT. Table 1 and Figures 5 and 6 show that sign-fixing has comparable stability as OPT. To explain why sign-fixing works, we first explain what causes instability. Note that if we flip the signs of some columns of $\mathbf{Z}_t^{(i)}$, the subspace $\mathcal{R}(\mathbf{Z}_t^{(i)})$ remains the same. During the local power iterations on the i-th node, the signs of the columns of $\mathbf{Z}_t^{(i)}$ can flip. While the sign flipping does not affect $\mathcal{R}(\mathbf{Z}_t^{(i)})$, it changes the outcome of the aggregation of $\mathbf{Z}_t^{(1)}, \cdots, \mathbf{Z}_t^{(m)}$. The sign-fixing method can counteract sign flippings and thereby stabilizes `LocalPower`.

Table 2 shows that `LocalPower` with decaying p has better stability. With the decaying strategy used, p will drop to 1 after several communications, and `LocalPower` becomes the standard `DPI` which does not suffer from the instability issue.

Effect of local sample size. Since the n data samples are partitioned among m nodes uniformly at random, every node holds $s = \frac{n}{m}$ samples. Figure 3(b) shows that small m, equivalently, big s, is good for `LocalPower`. We use $\eta = \max_{i\in[m]} \|\mathbf{M}_i - \mathbf{M}\|_2 / \|\mathbf{M}\|_2$ to measure the difference between a local covariance matrix and the full one. We give the values of η under different uniform partitions in Table 3. It shows that if s is large (so m is small), η is small, which implies $\mathbf{M}_1, \cdots, \mathbf{M}_m$ well approximate the global matrix $\mathbf{M}$, and the residuals accumulated by the local iterations are small. It in turn makes the curves with small m have small errors. This can be explained by our theories.

6. Discussion

Smallness on η. Theorem 1 requires $\eta = \mathcal{O}(\frac{1}{\kappa})$ which might be too stringent in practice. If we use a refined analysis just like Guo et al. (2021), it can be relaxed to $\eta = \mathcal{O}(1)$ as well as ϵ_0 whose dependence on κ can be removed.[7] Besides, the concurrent work (Charisopoulos et al., 2020) provides sharper analysis on one-shot average via OPT, which might be used to refine our analysis and relax the strictness on η further.

related work. It simply averages local top-k eigenvectors rather than distributed averaging methods (see Algorithm 2 and 3).

[7] In particular, Guo et al. (2021) analyzes the convergence of the virtual sequence in a form of $\overline{\mathbf{Z}}_t = \sum_{i=1}^n p_i \mathbf{Z}_t^{(i)} \mathbf{D}_t^{(i)}$, while we focus on the weighted $Y_t^{(i)}$, i.e., $\overline{\mathbf{Y}}_t = \sum_{i=1}^n p_i \mathbf{Y}_t^{(i)} \mathbf{D}_t^{(i)}$. Roughly speaking, $\|\overline{\mathbf{Y}}_t\|$ is about $\|\mathbf{M}\|_2$ larger than $\|\overline{\mathbf{Z}}_t\|$, while $\|\overline{\mathbf{Y}}_t^\dagger\|$ is about $\|\mathbf{M}^\dagger\|_2$ smaller than $\|\overline{\mathbf{Z}}_t^\dagger\|$. It leads to an additional factor $\kappa = \|\mathbf{M}\|_2 \|\mathbf{M}^\dagger\|_2$.

Table 1. We report the errors of three proposed algorithms and three baselines methods on seven datasets. We show the mean errors of ten repeated experiments with its standard deviation enclosed in parentheses. The result of full fifteen datasets is shown in Table 6.

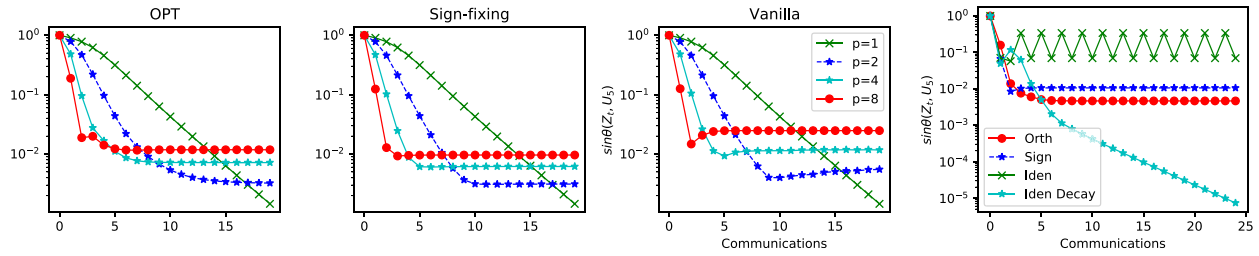
Datasets	LocalPower ($p = 4$)			DR-SVD	UDA	WDA
	OPT	Sign-fixing	Vanilla			
A9a	**4.09e-03** (**4.20e-4**)	5.82e-03 (1.41e-3)	8.13e-02 (3.44e-2)	4.63e-02 (9.24e-3)	2.64e-02 (1.58e-2)	2.40e-02 (1.50e-2)
Abalone	**3.16e-03** (2.89e-3)	3.85e-03 (**2.54e-3**)	3.03e-02 (5.70e-2)	3.20e-01 (2.30e-1)	1.03e-01 (9.38e-2)	1.03e-01 (9.18e-2)
Acoustic	**1.83e-03** (4.40e-4)	2.03e-03 (**3.90e-4**)	2.38e-03 (8.5e-4)	1.54e-02 (6.59e-3)	7.76e-03 (2.64e-3)	6.67e-03 (2.42e-3)
Combined	6.01e-03 (1.59e-3)	**5.57e-03** (**1.05e-3**)	2.47e-02 (3.40e-2)	5.19e-02 (6.23e-3)	4.63e-02 (2.97e-3)	4.16e-02 (2.76e-2)
Connect-4	**1.27e-02** (4.52e-3)	1.81e-02 (**3.79e-3**)	1.70e-02 (4.35e-3)	1.61e-02 (2.96e-3)	1.65e-01 (3.48e-2)	1.56e-0 1(3.26e-2)
Covtype	7.38e-03 (6.50e-4)	**6.23e-03** (**4.70e-4**)	1.28e-02 (1.88e-3)	1.82e-01 (8.73e-2)	6.09e-02 (9.70e-3)	5.60e-02 (9.41e-3)
MSD	9.90e-03 (1.21e-3)	**9.62e-03** (**5.20e-4**)	1.44e-02 (1.58e-3)	3.01e-02 (9.64e-3)	1.55e-02 (1.39e-3)	1.92e-02 (1.14e-3)

Table 2. Error comparison of LocalPower with decay strategy under the same setting of Table 1. See Table 7 for full results. In theory, LocalPower with decay strategy achieves zero error.

Datasets	LocalPower with $p = 4$ and the decay strategy		
	OPT	Sign-fixing	Vanilla
A9a	4.84e-03 (1.40e-02)	1.52e-03 (4.08e-03)	**3.11e-04** (**4.84e-04**)
Abalone	**3.50e-10** (4.10e-10)	4.14e-10 (**4.00e-10**)	6.12e-10 (6.77e-10)
Acoustic	**1.40e-05** (**2.16e-05**)	1.92e-05 (3.72e-05)	2.28e-05 (4.91e-05)
Combined	3.68e-03 (5.63e-03)	7.74e-03 (1.70e-02)	**2.99e-03** (**3.88e-03**)
Connect-4	4.90e-03 (8.47e-03)	3.58e-03 (4.35e-03)	**3.09e-03** (**3.16e-03**)
Covtype	5.57e-04 (1.55e-03)	**4.95e-05** (**5.40e-05**)	8.01e-05 (8.62e-05)
MSD	2.75e-05 (3.34e-05)	**2.47e-05** (3.27e-05)	3.02e-05 (**2.10e-05**)

Table 3. The value of η under uniform partitions on some datasets. It can be seen that for a fixed n, the larger m, the larger η. Full results see Table 5.

Dataset	$m = 20$	$m = 40$	$m = 60$
A9a	0.034	0.0563	0.0701
Abalone	0.1089	0.23	0.2458
Acoustic	0.0063	0.0107	0.0134
Combined	0.006	0.0089	0.0113
Connect-4	0.0376	0.054	0.0771
Covtype	0.0078	0.011	0.0159
MSD	0.0007	0.0009	0.0012



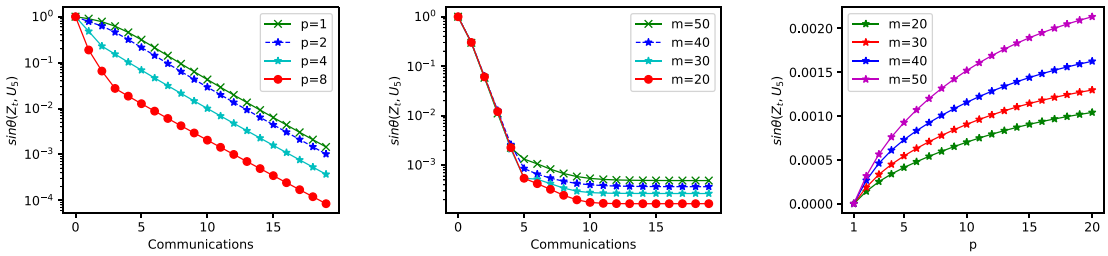
(a) The performance of LocalPower with different p on Covtype dataset (b) Stability on A9a dataset

Figure 2. (a) We illustrate the convergence of LocalPower with different $\mathcal{F}$'s and various p on Covtype dataset where $\mathbf{A} \in \mathbb{R}^{581,012\times 54}$. See Figure 5, 6 and 7 for full results. (b) The vanilla LocalPower sometimes fluctuates and even diverges (see Figure 7 for full results). We can stabilize it in two ways: (i) use $\mathcal{O}_k$ or $\mathcal{D}$ instead or (ii) use the decay strategy.

Increase local sample size. In addition to OPT or the decay strategy, we find that increasing local data size also reduces the final error. Intuitively, if s_i is sufficiently large, then $\mathbf{M}_i = \frac{1}{s_i}\mathbf{A}_i^\top \mathbf{A}_i$ will be very close to $\mathbf{M} = \frac{1}{n}\mathbf{A}^\top \mathbf{A}$. Actually, this is true if we construct each $\mathbf{A}_i$ by sampling uniformly from the overall data $\mathbf{A}$ (see Lemma 2). Therefore, to make η sufficiently small, we can increase local data size. If the total number of rows n is fixed in advance, increasing each s_i is equivalent to decreasing the number of worker nodes m.

The term $\eta = \max_{i\in[m]} \|\mathbf{M}_i - \mathbf{M}\|_2 / \|\mathbf{M}\|_2$ is commonly used to analyze matrix approximation problems. It aims to ensure each $\mathbf{A}_i$ is a typical representative of the whole dataset $\mathbf{A}$. Prior work (Gittens & Mahoney, 2016; Woodruff, 2014; Wang et al., 2016) showed that uniform sampling and the partition size in Lemma 2 suffice for that $\mathbf{M}_i$ well approximates $\mathbf{M}$. The proof is based on matrix Bernstein (Tropp, 2015). Therefore, under uniform sampling, the smallness of η means sufficiently large local dataset size (or equivalently a small number of worker nodes). This can be also seen in Table 3.

One may doubt the motivation of each device anticipating the cooperated eigenspace estimation due to the large local dataset assumption. Here we focus on the empirical PCA



(a) Decay strategy (b) Vary device number m (c) Error dependence on p and m

Figure 3. Some results on Covtype dataset. (a) A typical convergence curve of the decay strategy. See Figure 8, 9 and 10 for full results. (b) The smaller m, the faster convergence as well as the smaller error. See Figure 11 and 12 for full results. (c) The error depends positively on p and m. See Figure 13 for full results.

rather than the population PCA. This implies we inevitably suffer a statistic error that will diminish if we have an infinite number of total samples. As a result, if m devices participate in the training with comparable local data size, the statistical error can be reduced by a factor of $\sqrt{m}$. See Appendix B for more details.

Lemma 2 (Uniform sampling.)**.** *Let $\epsilon, \delta \in (0,1)$. Assume the rows of $\mathbf{A}_i$ are sampled from the rows of $\mathbf{A}$ uniformly at random. Assume each node has sufficiently many samples, that is, for all $i \in m$,*

$$s_i \geq \frac{3\mu\rho}{\epsilon^2} \log\left(\frac{\rho m}{\delta}\right),$$

where $\rho = \operatorname{rank}(\mathbf{A})$ and μ is the row coherence of $\mathbf{A}$.[8] *With probability greater than $1-\delta$, we have*

$$\eta = \max_{i\in[m]} \|\mathbf{M}_i - \mathbf{M}\|_2 / \|\mathbf{M}\|_2 \leq \epsilon.$$

Error dependence. The choice of $\mathcal{I}_T$ determines the frequency `LocalPower` communicates. We explore the use of $\mathcal{I}_T = \{0, p, 2p, \cdots, p\}$ and the decay strategy in experiments. When $p = 1$, `LocalPower` reduces to `DPI`. As a result, both the residual errors Ψ_t and Ω_t vanish. As shown in Lemma 1, `DPI` converges to zero error. When $p \geq 2$, the error $\sin\theta_k$ typically increases with p and is non-zero. Corollary 1 depicts the relationship between the error and problem-dependent parameters including n, m, p. The proof is provided in Appendix A.5. It can be proved by Theorem 2 and Lemma 2.

Corollary 1. *Under uniform sampling and assuming $s_i = \Theta(\frac{n}{m})$ and n is sufficiently large, , with probability 1- δ, `LocalPower` with OPT has an asymptotic error satisfying*

$$\limsup_{t\to\infty} \sin\theta_k(\mathbf{Z}_T, \mathbf{U}_k) = \mathcal{O}\left(h_p\left(\sqrt{\frac{m}{n}}\right)\right),$$

where $h_p(x)$ is non-negative and increasing in (typically both p and) x, and it satisfies $h_1(x) = 0$ as well as $0 \leq h_p(x) \leq Cx$ for some C. We hide constants $\sigma_k, k, d, \rho, \kappa, \delta$ in the big-$\mathcal{O}$ notation and $h_p(\cdot)$. However, with any decay strategy in which p converges to 1 finally, `LocalPower` achieves zero error asymptotically.

Corollary 1 says that when p goes to infinity, the error is saturated and has a finite limit, because $h_p(\cdot)$ is bounded. The curve of error v.s. p and m in Figure 3(c) validates the conclusion. Indeed, the extreme case of super large p means `LocalPower` reduces to the one-shot method, which has a non-zero optimization error typically. Corollary 1 also reveals methods to reduce error. To that end, we can (i) use the decay strategy ($p \downarrow$) to achieve arbitrary error or (ii) reduce the number of devices ($m \downarrow$) or collect more data points $n \uparrow$. Both methods work in experiments empirically.

Dependence on $\sigma_k - \sigma_{k+1}$. Our result depends on $\sigma_k - \sigma_{k+1}$ even when $r > k$ where r is the number of columns used in subspace iteration. If we borrow the tool of Balcan et al. (2016a) rather than that of Hardt & Price (2014), we can improve the result to a slightly milder dependency on $\sigma_k - \sigma_{q+1}$, where q is any intermediate integer between k and r. In particular, the required iteration T will decrease from $\widetilde{\mathcal{O}}\left(\frac{\sigma_k}{\sigma_k - \sigma_{k+1}}\right)$ to $\widetilde{\mathcal{O}}\left(\frac{\sigma_k}{\sigma_k - \sigma_{q+1}}\right)$. It means using additional columns fastens convergence. For a formal statement, please refer to Appendix C.

Further extensions. Our proposed `LocalPower` is simple, effective and well-grounded. While we analyze it only on the centralized setting, `LocalPower` can be extended to broader settings, such as decentralized setting (Gang et al., 2019) and streaming setting (Raja & Bajwa, 2020). To further reduce the communication complexity, we can combine `LocalPower` with sketching techniques (Boutsidis et al., 2016; Balcan et al., 2016b). For example, we could sketch each $\mathbf{Y}_t^{(i)}$ and communicate the compressed iterates to a

[8] The row coherence of $\mathbf{A}$ is defined by $\mu(\mathbf{A}) = \frac{n}{d}\max_j \|\mathbf{u}_j\|_2^2 \in [1, \frac{n}{d}]$ where $\mathbf{u}_j$ comes from the column orthonormal bases of $\mathbf{A}$.

central server in each iteration. We leave the extensions to our future work. Besides, in typical federated learning structures, real systems clients might not correspond to the central server due to connection failure. It is also possible to consider partial participation of clients and the optimal way of client selection (Reisizadeh et al., 2020; Chen et al., 2020). Guo et al. (2021) makes an attempt towards the direction.

7. Conclusion

We have developed a communication-efficient distributed algorithm named `LocalPower` to solve the truncated SVD. Every worker machine performs multiple (say p) local power iterations between two consecutive communications. We have theoretically shown that `LocalPower` converges p times faster (in terms of communication) than the baseline distributed power iteration, if the residual error is uniformly small. To reduce the residual error, we can (i) use OPT or sign-fixing, (ii) make use of a decay strategy that halves p gradually, and (iii) increase local data size. Both OPT and sign-fixing are more stable, while sign-fixing additionally is computationally efficient. The strategy is motivated by an experimental phenomenon that large p often leads to a quick initial drop of loss but a higher final error. The decay strategy obtains zero error asymptotically in theory and has better convergence performance in experiments. We have conducted the thorough experiments to show the effectiveness of `LocalPower` and all the theories are agree with our empirical experiments.

Acknowledgement

Li, Chen and Zhang have been supported by the National Key Research and Development Project of China (No. 2018AAA0101004 & 2020AAA0104400), and Beijing Academy of Artificial Intelligence (BAAI).

References

Allen-Zhu, Z. and Li, Y. Lazysvd: even faster svd decomposition yet without agonizing pain. In *Advances in Neural Information Processing Systems*, pp. 974–982, 2016. 24

Arbenz, P., Kressner, D., and Zürich, D.-M. E. Lecture notes on solving large scale eigenvalue problems. *D-MATH, EHT Zurich*, 2012. 2, 3

Arora, R., Cotter, A., and Srebro, N. Stochastic optimization of pca with capped msg. In *Advances in Neural Information Processing Systems*, pp. 1815–1823, 2013. 24

Balcan, M.-F., Du, S. S., Wang, Y., and Yu, A. W. An improved gap-dependency analysis of the noisy power method. In *Conference on Learning Theory*, pp. 284–309, 2016a. 8, 23, 24, 25

Balcan, M. F., Liang, Y., Song, L., Woodruff, D., and Xie, B. Communication efficient distributed kernel principal component analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 725–734, 2016b. 8

Bhaskara, A. and Wijewardena, P. M. On distributed averaging for stochastic k-pca. In *Advances in Neural Information Processing Systems*, pp. 11024–11033, 2019. 5, 24, 25, 26

Boutsidis, C., Woodruff, D. P., and Zhong, P. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 236–249. ACM, 2016. 8

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009. 1

Cape, J. Orthogonal procrustes and norm-dependent optimality. *The Electronic Journal of Linear Algebra*, 36(36): 158–168, 2020. 3, 12, 13

Charisopoulos, V., Benson, A. R., and Damle, A. Communication-efficient distributed eigenspace estimation. *arXiv preprint arXiv:2009.02436*, 2020. 6, 24

Chen, W., Horvath, S., and Richtarik, P. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020. 9

Chen, X., Lee, J. D., Li, H., and Yang, Y. Distributed estimation for principal component analysis: An enlarged eigenspace analysis. *Journal of the American Statistical Association*, pp. 1–12, 2021. 24

De Sa, C., He, B., Mitliagkas, I., Ré, C., and Xu, P. Accelerated stochastic power iteration. *Proceedings of machine learning research*, 84:58, 2018. 24

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990. 1, 6

Fan, J., Wang, D., Wang, K., Zhu, Z., et al. Distributed estimation of principal eigenspaces. *The Annals of Statistics*, 47(6):3009–3031, 2019. 3, 5, 24, 25, 26

Gang, A., Raja, H., and Bajwa, W. U. Fast and communication-efficient distributed pca. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7450–7454. IEEE, 2019. 8, 24, 25

Garber, D. and Hazan, E. Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015. 24

Garber, D., Hazan, E., Jin, C., Kakade, S. M., Musco, C., Netrapalli, P., and Sidford, A. Faster eigenvector computation via shift-and-invert preconditioning. In *ICML*, pp. 2626–2634, 2016. 24

Garber, D., Shamir, O., and Srebro, N. Communication-efficient algorithms for distributed stochastic principal component analysis. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1203–1212. JMLR. org, 2017. 2, 3, 4, 6, 24

Gittens, A. and Mahoney, M. W. Revisiting the Nyström method for improved large-scale machine learning. *Journal of Machine Learning Research*, 17(1):3977–4041, 2016. 7

Gittens, A., Devarakonda, A., Racah, E., Ringenburg, M., Gerhardt, L., Kottalam, J., Liu, J., Maschhoff, K., Canon, S., and Chhugani, J. Matrix factorizations at scale: a comparison of scientific data analytics in spark and C+ MPI using three case studies. In *IEEE International Conference on Big Data*, 2016. 1

Golub, G. H. and Van Loan, C. F. *Matrix computations*, volume 3. JHU Press, 2012. 1, 24

Grammenos, A., Mendoza-Smith, R., Mascolo, C., and Crowcroft, J. Federated principal component analysis. *arXiv preprint arXiv:1907.08059*, 2019. 24

Guo, X., Li, X., Chang, X., Wang, S., and Zhang, Z. Privacy-preserving distributed svd via federated power. *arXiv preprint arXiv:2103.00704*, 2021. 6, 9

Halko, N., Martinsson, P.-G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011. 5, 26

Hardt, M. and Price, E. The noisy power method: A meta algorithm with applications. In *Advances in Neural Information Processing Systems*, pp. 2861–2869, 2014. 8, 14, 15, 22, 23, 25

Ji-guang, S. Perturbation of angles between linear subspaces. *Journal of Computational Mathematics*, pp. 58–61, 1987. 18

Khaled, A., Mishchenko, K., and Richtárik, P. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019. 24

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019a. 4, 24

Li, X., Yang, W., Wang, S., and Zhang, Z. Communication efficient decentralized training with multiple local updates. *arXiv preprint arXiv:1910.09126*, 2019b. 4, 6, 24

Liang, Y., Balcan, M.-F. F., Kanchanapally, V., and Woodruff, D. Improved distributed principal component analysis. In *Advances in Neural Information Processing Systems*, pp. 3113–3121, 2014. 3

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics (AISTATS)*, 2017. 1, 24

Musco, C. and Musco, C. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 24

Oja, E. and Karhunen, J. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985. 24

Raja, H. and Bajwa, W. U. Distributed stochastic algorithms for high-rate streaming principal component analysis. *arXiv preprint arXiv:2001.01017*, 2020. 8, 25

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031. PMLR, 2020. 9

Saad, Y. Numerical methods for large eigenvalue problems. *preparation. Available from: http://www-users. cs. umn. edu/saad/books. html*, 2011. 1, 2, 3, 24

Schönemann, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 3, 12, 13

Shamir, O. A stochastic pca and svd algorithm with an exponential convergence rate. In *International Conference on Machine Learning*, pp. 144–152, 2015. 24

Shamir, O. Convergence of stochastic gradient descent for pca. In *International Conference on Machine Learning*, pp. 257–265, 2016. 24

Simchowitz, M., Alaoui, A. E., and Recht, B. On the gap between strict-saddles and true convexity: An omega (log d) lower bound for eigenvector approximation. *arXiv preprint arXiv:1704.04548*, 2017. 25

Stich, S. U. Local SGD converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018. 4, 24

Sun, J.-g. On perturbation bounds for the qr factorization. *Linear algebra and its applications*, 215:95–111, 1995. 20

Tropp, J. A. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12 (4):389–434, 2012. 23

Tropp, J. A. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015. 7

Vu, V. Q., Lei, J., et al. Minimax sparse principal subspace estimation in high dimensions. *The Annals of Statistics*, 41(6):2905–2947, 2013. 13

Wang, J. and Joshi, G. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update sgd. *arXiv preprint arXiv:1810.08313*, 2018a. 6

Wang, J. and Joshi, G. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576*, 2018b. 4, 24

Wang, S., Luo, L., and Zhang, Z. SPSD matrix approximation vis column selection: Theories, algorithms, and extensions. *Journal of Machine Learning Research*, 17 (49):1–49, 2016. 7

Wang, S., Gittens, A., and Mahoney, M. W. Scalable kernel k-means clustering with Nystrom approximation: Relative-error bounds. *Journal of Machine Learning Research*, 20(12):1–49, 2019. 1

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. 1

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014. 7

Wu, S. X., Wai, H.-T., Li, L., and Scaglione, A. A review of distributed algorithms for principal component analysis. *Proceedings of the IEEE*, 106(8):1321–1340, 2018. 24

Xu, Z. Gradient descent meets shift-and-invert preconditioning for eigenvector computation. *Advances in Neural Information Processing Systems*, 31:2825–2834, 2018. 24

Ye, K. and Lim, L.-H. Distance between subspaces of different dimensions. *arXiv preprint arXiv:1407.0900*, 4, 2014. 4, 12

Yu, H., Yang, S., and Zhu, S. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *AAAI Conference on Artificial Intelligence*, 2019. 4, 24

Zhou, F. and Cong, G. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017. 24