# Communication-Efficient Distributed SVD via Local Power Iterations

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

[1] School of Mathematical Sciences, Peking University, China

[2] Department of Computer Science, Stevens Institute of Technology, USA

## Abstract

We study distributed computing of the truncated singular value decomposition problem.

- We develop an algorithm that we call `LocalPower` for improving communication efficiency.
- 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`.

## Introduction

- The truncated singular value decomposition (SVD) which has broad applications in machine learning.
- 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.
- 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.
- Communication costs can outweigh computation costs in large-scale matrix computation problems. Thus it is crucial to save communication as possible.

## Distributed Power Iteration

To compute the top $k$ right singular vectors of $\mathbf{A}$, denoted by $\mathbf{V}_k$,

- Power iteration repeats

$$\mathbf{Y} \longleftarrow \mathbf{M}\mathbf{Z} \quad \text{and} \quad \mathbf{Z} \longleftarrow \mathsf{orth}(\mathbf{Y})$$

where $\mathbf{M} = \frac{1}{n}\mathbf{A}^\top\mathbf{A}$ and $\mathbf{Z} \in \mathbb{R}^{d\times k}$. The column space of $\mathbf{Z}$ will converges to $\mathbf{V}_k$ geometrically.
- In a distributed setting, we partition $\mathbf{A}$ as $\mathbf{A}^\top = [\mathbf{A}_1^\top, \cdots, \mathbf{A}_m^\top]$ with $\mathbf{A}_i \in \mathbb{R}^{s_i\times d}$.
- Distributed power iteration (DPI) performs an aggregation after each device runs one power iteration. See Figure 1.
- DPI needs $\Omega\left(\frac{\sigma_k}{\sigma_k-\sigma_{k+1}}\log\left(\frac{d}{\epsilon}\right)\right)$ communications to obtain an $\epsilon$-accurate solution. Here, $\sigma_j$ is the $j$-th largest singular value of the matrix $\mathbf{M}$.

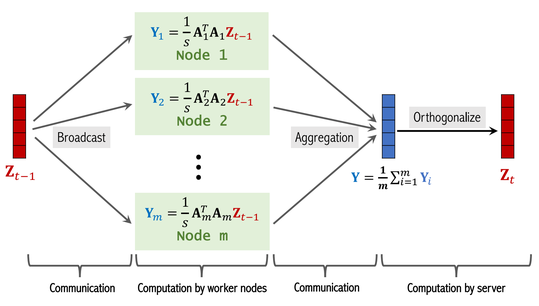


Figure 1:Distributed Power iteration

## LocalPower

Our main idea is to reduce the frequency of communications (see Figure 2). In this way, we hope to reduce the communication and synchronization costs and thereby improving the scalability.

- Between two communications, every worker node locally runs power iteration $p$ times.
- Distributed power iteration is a special case of `LocalPower` with $p = 1$.
- We propose three variants to improve the performance of `LocalPower` (see next block).

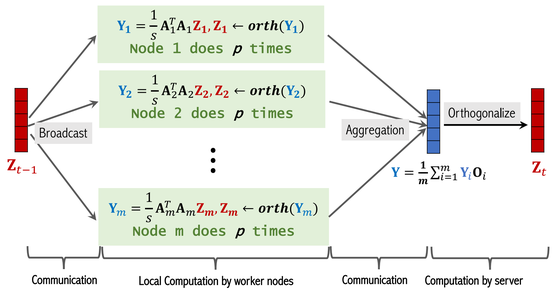


Figure 2:`LocalPower`

## Three Variants

- Decaying $p$ to obtain a high-accurate solution.
- For stability, using orthogonal procrustes transformation (OPT) to post-processes $\mathbf{Y}_{t+1}^{(i)}$ as $\mathbf{Y}_{t+1}^{(i)}\mathbf{O}_t^{(i)}$ where

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

$\mathbf{O}_t^{(i)}$ 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 $(\mathbf{Z}_t^{(i)})^\top\mathbf{Z}_t^{(1)}$.
- Sign-fixing replaces $\mathbf{O}^{(i)}$ in eqn. (1) by

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

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

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

## Theoretical Analysis

- If local data matrices $\mathbf{A}_i$'s are similar enough, i.e.,

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

is small enough or $p$ is not too large, `LocalPower` needs $O\left(\frac{1}{p}\frac{\sigma_k}{\sigma_k-\sigma_{k+1}}\log\left(\frac{d}{\epsilon}\right)\right)$ communications.
- For pure aggregation, we need $\eta = O(\frac{\epsilon}{\sqrt{k}p\kappa^p})$ where $\kappa = \|\mathbf{M}\|\|\mathbf{M}^\dagger\|$. When OPT is used, we only need $\eta = O(\epsilon)$.
- OPT relaxes the restriction on matrix similarities.

## Experiments

We use 15 datasets available on the LIBSVM. The $n$ data samples are randomly shuffled and then partitioned among $m$ nodes so that each node has $s = \frac{n}{m}$ samples. All the algorithms start from the same initialization $\mathbf{Y}_0$. We fix the target rank to $k = 5$. For limited space, we select some typical result from our paper.

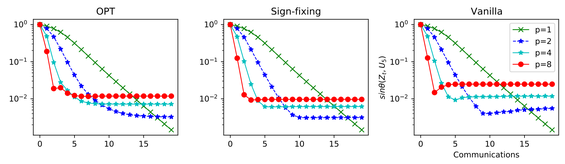


Figure 3:Different $p$ and $\mathbf{O}$ on Covtype ($581,012 \times 54$).

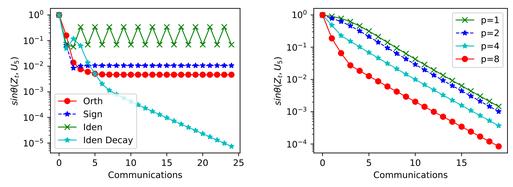


Figure 4:Left: OPT and sign-fixing are more stable on A9a. Right: typical convergence of the decay strategy on Covtype.

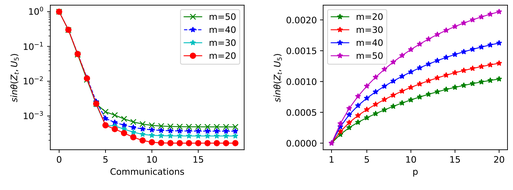


Figure 5:Left: The smaller $m$, the faster convergence as well as the smaller error. Right: The error depends positively on $p$ and $m$. Both are on Covtype.

## Conclusion

- `LocalPower` are more efficient than DPI.
- OPT and sign-fixing are more stable than pure aggregation.
- Decaying $p$ helps us better trade-off the communication efficiency and final error.