Communication-efficient Algorithms for Distributed Stochastic Principal Component Analysis

Dan Garber [1] **Ohad Shamir** [2] **Nathan Srebro** [3]

Abstract

We study the fundamental problem of *Principal Component Analysis* in a statistical distributed setting in which each machine out of m stores a sample of n points sampled i.i.d. from a single unknown distribution. We study algorithms for estimating the leading principal component of the population covariance matrix that are both communication-efficient and achieve estimation error of the order of the centralized ERM solution that uses all mn samples. On the negative side, we show that in contrast to results obtained for distributed estimation under convexity assumptions, for the PCA objective, simply averaging the local ERM solutions cannot guarantee error that is consistent with the centralized ERM. We show that this unfortunate phenomena can be remedied by performing a simple correction step which correlates between the individual solutions, and provides an estimator that is consistent with the centralized ERM for sufficiently-large n. We also introduce an iterative distributed algorithm that is applicable in any regime of n, which is based on distributed matrix-vector products. The algorithm gives significant acceleration in terms of communication rounds over previous distributed algorithms, in a wide regime of parameters.

1. Introduction

Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1933; Jolliffe, 2002) is one of the most celebrated and popular techniques in data analysis and machine learning. For data that consists of N vectors in $\mathbb{R}^d$, $\mathbf{x}_1, ..., \mathbf{x}_N$, with normalized covariance matrix $\hat{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^\top$, The PCA method finds the k-dimensional subspace (which corresponds to the span of the top k principal components) such that the projection of the data onto the subspace has largest variance, i.e., it is the solution to the optimization problem:

$$\max_{\mathbf{W}\in\mathbb{R}^{d\times k}, \mathbf{W}^T\mathbf{W}=\mathbf{I}} \|\hat{\mathbf{X}}\mathbf{W}\|_F^2. \tag{1}$$

PCA is often considered in a statistical setting in which the assumption is that the input vectors are not arbitrary but sampled i.i.d. from some fixed but unknown distribution with certain general characteristics $\mathcal{D}$. Then, it is often of interest to use the observed sample to estimate the top k principal components of the population covariance matrix, rather then that of the sample, which leads to the modified optimization problem:

$$\max_{\mathbf{W}\in\mathbb{R}^{d\times k}, \mathbf{W}^T\mathbf{W}=\mathbf{I}} \|\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathbf{x}\mathbf{x}^\top\right]\mathbf{W}\|_F^2. \tag{2}$$

Of course the empirical estimation problem (1) and the population estimation problem (2) are well connected, and it is well-known that under mild assumptions on the distribution $\mathcal{D}$ and given a sufficiently large sample, we can guarantee small estimation error in (2) by solving optimization problem (1).

In this work we consider the problem of estimating the first principal component (i.e., $k = 1$) in a statistical and distributed setting. We assume the availability of m machines, each of which stores a sample of n vectors sampled i.i.d from a fixed distribution $\mathcal{D}$ over $\mathbb{R}^d$, and we are interested in algorithms that can be applied efficiently to solve Problem (2) for $k = 1$, with estimation error that approaches that of a *centralized* algorithm, which has access to all mn samples and does not pay for communication between machines. Indeed, when considering the efficiency of algorithms, we will mainly focus on the amount of communication between machines they require, since this is often the most expensive resource in distributed computing. We note that the i.i.d. assumption is standard in many applications of PCA, and can be leveraged to get more efficient algorithms than when the data partition is arbitrary. Also, we will make a standard assumption that the population covariance matrix has a non-zero additive gap between the

[1] Technion - Israel Institute of Technology, Haifa, Israel [2] Weizmann Institute of Science, Rehovot, Israel [3] Toyota Technological Institute, Illinois, USA. Correspondence to: Dan Garber <dangar@technion.ac.il>, Ohad Shamir <ohad.Shamir@weizmann.ac.il>, Nathan Srebro <nati@ttic.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

first and second eigenvalues, which makes the problem of estimating the leading principal component meaningful.

A main challenge that often arises in many computational settings of principal components is that it leads to inherently non-convex optimization problems. While many times these problems turn out to admit efficient algorithms, the rich toolbox of optimization and statistical estimation procedures developed for *convex* problems often cannot be directly applied to problems such as (1) and (2). Instead, one often needs to consider a specialized and more involved analysis, to get analogous convergence results for the PCA problem. This for instance was the case in a recent wave of results that applied concepts such as stochastic gradient updates (Balsubramani et al., 2013; Shamir, 2016a; Jain et al., 2016b; Allen Zhu & Li, 2016b) and variance reduction (Shamir, 2015; 2016c; Garber & Hazan, 2015; Garber et al., 2016; Allen Zhu & Li, 2016a) to the PCA problem. This is also the case in our distributed setting. For instance, (Zhang et al., 2013) proposed communication-efficient algorithms for a distributed statistical estimation settings, similar to ours, but under convexity assumptions. The authors show that under their assumptions, in a wide regime of parameters (namely when the per-machine sample size n is large enough), then a simple averaging of the *empirical risk minimizers* (ERM), computed locally on each machine, leads to estimation error of the population parameters of the order the centralized ERM solution. While averaging makes perfect sense in a convex setting, it is clear that it can completely fail in a non-convex setting. Indeed, we show that already for the PCA problem with $k = 1$, simply averaging the local ERM solutions (and normalizing to obtain a unit vector as required), cannot improve significantly over the estimation error of any single machine. We then show that a simple fix to the above scheme, namely correlating the directions of individual ERM solutions, remedies this phenomena and results in estimation error similar to that of the centralized ERM solution. Much like the results of (Zhang et al., 2013), this result only holds in the regime when the per-machine sample size n is sufficiently large. As discussed, due to the inherent non-convexity of the PCA objective, this approach requires a novel analysis tailored to the PCA problem. In this context, we view this work as an initiation of a research effort to understand how to efficiently aggregate statistical estimators in a distributed non-convex setting.

A second line of results for distributed estimation under convexity assumptions consider iterative algorithms that perform multiple communication rounds and are based on distributed gradient computations (some examples include (Shamir et al., 2014; Zhang & Lin, 2015; Lee et al., 2015; Shamir, 2016b; Jaggi et al., 2014; Reddi et al., 2016)). The benefit of these methods is that (a) they provide meaningful estimation error guarantees in a much wider regime of parameters than the "one-shot" aggregation methods (namely in terms of the number of samples per machine), and (b), due to their iterative nature, they allow to approximate the centralized ERM solution arbitrary well. Unfortunately, these methods, all of which rely heavily on convexity assumption, cannot be directly applied to the PCA problem. Towards designing efficient distributed iterative methods for our PCA setting, we consider the application of the recently proposed method of *Shift-and-Invert power iterations* (S&I) for PCA (Garber & Hazan, 2015; Garber et al., 2016). The S&I method reduces the problem of computing the leading eigenvector of a real positive semidefinite matrix to that of approximately solving a small number (i.e. poly-logarithmic in the problem parameters) of systems of linear equations. These in turn, could be efficiently solved by arbitrary distributed convex solvers. We show that coupling the S&I method with the stochastic pre-conditioning technique for linear systems proposed in (Zhang & Lin, 2015) and well known fast gradient methods such as the conjugate gradient method, gives state-of-the-art guarantees in terms of communication costs, and provides a significant improvement over distributed variants of classical fast eigenvector algorithms such as power iterations and the faster Lanczos algorithm. Much like its convex counterparts, which only rely on distributed gradient computations and simple vector aggregations, our iterative method only relies on distributed matrix-vector products, i.e., it requires each machine to only send products of its local empirical covariance matrix with some input vector.

Beyond the results described so far, (Liang et al., 2014; Boutsidis et al., 2016) studied distributed algorithms for PCA in a *deterministic* setting in which the partition of the data across machines is arbitrary and communication is measured in terms of number of transmitted bits. The approximation guarantees provided in these works are in terms of the projection of the data onto the leading principal components (instead of alignment between the estimate and the optimal solution, studied in this paper). Applying these results to our setting will give a number of communication rounds that scales like $\mathrm{poly}(\epsilon^{-1}\delta^{-1})$, where ϵ is the desired error and δ is the population eigengap. In our setting, ϵ will scale with the inverse of the size of the sample, i.e., $\epsilon \approx (mn)^{-1}$, which for these algorithms will result in amount of communication that is polynomial in the size of the data. In contrast, we will be interested in algorithms whose communication costs does not scale with n at all. In this context we note that, by focusing on algorithms that either perform simple aggregation of local ERM solutions, or perform only distributed matrix-vector products with the empirical covariance matrix, we can circumvent the need to measure communication explicitly in terms of the number of bits transmitted, which often burdens the analysis of natural algorithms, such as those proposed here.

2. Preliminaries

2.1. Notation and problem setting

We write vectors in $\mathbb{R}^d$ in boldface lower-case letters (e.g., $\mathbf{v}$), matrices in boldface upper-case letters (e.g., $\mathbf{X}$), and scalars are written as lightface letters (e.g., c). We let $\|\cdot\|$ denote the standard Euclidean norm for vectors and the spectral norm for matrices.

We consider the following statistical distributed setting. Let $\mathcal{D}$ be a distribution over vectors in $\mathbb{R}^d$ with squared ℓ_2 norm at most b, for some $b > 0$. We consider a setting in which m machines, numbered $1...m$, are each given a dataset of n samples drawn i.i.d. from $\mathcal{D}$. We let $\mathbf{v}_1$ denote a leading eigenvector of the population covariance matrix $\mathbf{X} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\mathbf{x}^\top]$. Our goal is to efficiently (mainly in terms of communication) find an estimate $\mathbf{w}$ for $\mathbf{v}_1$, i.e., a unit vector that maximizes the product $(\mathbf{v}_1^\top\mathbf{w})^2$ with high probability. Towards this end, we assume that the population covariance matrix $\mathbf{X}$ has a non-zero eigengap δ, i.e., $\delta := \lambda_1(\mathbf{X}) - \lambda_2(\mathbf{X}) > 0$, where $\lambda_i(\cdot)$ denotes the ith largest eigenvalue of a symmetric real matrix. Note that $\delta > 0$ is necessary for $\mathbf{v}_1$ to be uniquely defined (up to sign).

In addition, we let $\hat{\mathbf{X}}_i$ denote the empirical covariance matrix of the sample stored on machine i for every $i \in [m]$, i.e., $\hat{\mathbf{X}}_i = \frac{1}{n}\sum_{j=1}^n \mathbf{x}_j^{(i)}\mathbf{x}_j^{(i)\top}$, where $\mathbf{x}_1^{(i)}...\mathbf{x}_n^{(i)}$ are the samples stored on machine i. We let $\hat{\mathbf{X}}$ denote the empirical covariance matrix of the union of points across all machines i.e., $\hat{\mathbf{X}} = \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{X}}_i$.

Our model of communication assumes that the m machines work in rounds during which a central machine (w.l.o.g. machine 1) can send a single vector in $\mathbb{R}^d$ to all other machines, or every machine can send either the leading eigenvector of its local empirical covariance matrix, or the product of a single input vector with its local covariance, to machine 1. We will measure communication complexity in terms of number of such rounds required to achieve a certain estimation error.

2.1.1. The centralized solution

Our primary benchmark for measuring performance will be the *centralized empirical risk minimizer* which is the leading eigenvector of the aggregated empirical covariance matrix $\hat{\mathbf{X}}$.

The following standard result bounds the error of the centralized ERM.

Lemma 1 (Risk of centralized ERM). *Fix $p \in (0,1)$. Suppose that $\delta > 0$ and let $\hat{\mathbf{v}}_1$ denote the leading eigenvector of $\hat{\mathbf{X}}$, i.e., $\hat{\mathbf{v}}_1 \in \arg\max_{\mathbf{v}:\|\mathbf{v}\|=1} \mathbf{v}^\top\hat{\mathbf{X}}\mathbf{v}$. Then it holds w.p. at least $1-p$ that*

$$1 - (\mathbf{v}_1^\top\hat{\mathbf{v}}_1)^2 \ \leq\ \epsilon_{ERM}(p) := \frac{32b^2\ln(d/p)}{mn\delta^2}. \qquad (3)$$

Lemma 1 is a direct consequence of the following standard concentration argument for random matrices, and the Davis-Kahan sin(θ) theorem (whose proof is given in the appendix for completeness):

Theorem 1 (Matrix Hoeffding, see (Tropp, 2012))**.** *Let $\mathcal{D}$ be a distribution over vectors with squared ℓ_2 norm at most b, and let $\mathbf{X} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\mathbf{x}^\top]$. Let $\hat{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top$, where $\mathbf{x}_1, ..., \mathbf{x}_n$ are sampled i.i.d. from $\mathcal{D}$. Then, it holds that $\forall\epsilon > 0:\ \Pr\left(\|\hat{\mathbf{X}} - \mathbf{X}\| \geq \epsilon\right) \leq d\cdot\exp\left(-\frac{\epsilon^2 n}{16b^2}\right)$.*

Theorem 2 (Davis-Kahan sin(θ) theorem)**.** *Let $\mathbf{X}, \mathbf{Y}$ be symmetric real $d\times d$ matrices with leading eigenvectors $\mathbf{v}_\mathbf{X}$ and $\mathbf{v}_\mathbf{Y}$ respetively. Also, suppose that $\delta(\mathbf{X}) := \lambda_1(\mathbf{X}) - \lambda_2(\mathbf{X}) > 0$. Then it holds that $1 - \left(\mathbf{v}_\mathbf{X}^\top\mathbf{v}_\mathbf{Y}\right)^2 \leq 2\frac{\|\mathbf{X}-\mathbf{Y}\|^2}{\delta(\mathbf{X})^2}$.*

2.2. Informal statement of main results and previous algorithms

We now informally describe our main results, followed by a detailed description of previous approaches that are directly applicable to our setting. The algorithmic results (both new and old) are summarized in Table 1.

2.2.1. Main results

Failure of simple averaging of local ERM solutions We show that a natural approach of simply averaging the individual leading eigenvectors of the empirical covariance matrices $\hat{\mathbf{X}}_i$ (and normalizing the obtain a unit vector) cannot significantly improve (beyond logarithmic factors) over the performance of any of the individual eigenvectors. More concretely, if we let $\hat{\mathbf{v}}_1^{(i)}$ denote the leading eigenvector of $\hat{\mathbf{X}}_i$ for any $i \in [m]$, and we denote their average by $\bar{\mathbf{v}}_1 = \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{v}}_1^{(i)}$, then there exists a distribution $\mathcal{D}$ over vectors with magnitude $O(1)$ and covariance eigengap $\delta = 1$, such that

$$\forall m, n: \quad \mathbb{E}_\mathcal{D}\left[1 - \left(\frac{\bar{\mathbf{v}}_1^\top\mathbf{v}_1}{\|\bar{\mathbf{v}}_1\|}\right)^2\right] = \Omega\left(\frac{1}{n}\right),$$

See Theorem 3 in Section 3 for the complete and formal argument.

A successful single communication round algorithm via correlation of individual ERM solutions We show that if prior to averaging the local ERM solutions, as suggested above, we correlate their directions by aligning them according to any single machine (say machine number 1), i.e., we let $\bar{\mathbf{v}}_1 = \frac{1}{m}\sum_{i=1}^m \mathrm{sign}(\hat{\mathbf{v}}_1^{(i)\top}\hat{\mathbf{v}}_1^{(1)})\hat{\mathbf{v}}_1^{(i)}$, then this guarantees that for any $p \in (0,1)$, w.p. at least $1-p$,

$$1 - \left(\frac{\bar{\mathbf{v}}_1^\top\mathbf{v}_1}{\|\bar{\mathbf{v}}_1\|}\right)^2 = O\left(\frac{b^2\ln\left(\frac{dm}{p}\right)}{\delta^2 mn} + \frac{b^4\ln^2\left(\frac{dm}{p}\right)}{\delta^4 n^2}\right). \quad (4)$$

Method	$1-(\mathbf{w}^\top\mathbf{v}_1)^2$ w.p. $3/4$	# communcation rounds
Centralized ERM	$\epsilon_{\text{ERM}} = \Theta(\frac{b^2 \ln d}{\delta^2 mn})$	-
Distributed Power Method	$\epsilon_{\text{ERM}} \cdot (1+o(1))$	$\tilde{O}(\lambda_1/\delta)$
Distributed Lanczos	$\epsilon_{\text{ERM}} \cdot (1+o(1))$	$\tilde{O}(\sqrt{\lambda_1/\delta})$
"Hot-potato" SGD	$O(\epsilon_{\text{ERM}})$	m
Average of ERMs with sign-fixing (Theorem 4)	$O(\epsilon_{\text{ERM}}) + O\left(\frac{b^4 \ln^2 d}{\delta^4 n^2}\right)$	1
Distributed Shift&Invert + precond. linear systems (Theorem 6)	$\epsilon_{\text{ERM}} \cdot (1+o(1))$	$\tilde{O}(\min\{(b/\delta)^{1/2}n^{-1/4},\ m^{1/4}\})$

Table 1. Comparison of estimation error and number of communication rounds. For simplicity we fix the failure probability to $p = 1/4$ and assume mn is in the regime in which Lemma 1 is meaningful, i.e, $mn = \Omega(b^2\delta^{-2}\ln d)$. The $\tilde{O}(\cdot)$ suppresses logarithmic factors in $b, d, 1/p, 1/\epsilon_{\text{ERM}}$. For the result of Theorem 4 we assume the regime $m = O(d)$. The sub-constant $o(1)$ factors could be made, in principle, arbitrary small in all relevant results by trading approximation with communication.

See Theorem 4 in Section 3 for the complete and formal result.

In particular, in the likely scenario when $m = O(d/p)$ we have that w.p. at least $1-p$, $1-\left(\bar{\mathbf{v}}_1^\top \mathbf{v}_1/\|\bar{\mathbf{v}}_1\|\right)^2 = \epsilon_{\text{ERM}}(p)) \cdot O\left(1 + m^2 \cdot \epsilon_{\text{ERM}}(p)\right)$, where $\epsilon_{\text{ERM}}(p))$ is defined in Eq. (3). Another related interpretation of the results is that the bound in Eq. (4) is comparable with ϵ_{ERM} (up to poly-log factors) when $n = \Omega\left(\delta^{-2}b^2 m\ln(dm/p)\right)$.

We also show a matching lower bound that the bound in Eq. (4) is tight (up to poly-log factors) for this aggregation method.

A multi communication round algorithm We present a distributed algorithm based on the *Shift-and-Invert* framework for leading eigenvector computation (Garber & Hazan, 2015; Garber et al., 2016) which is applied to explicitly solving the centralized ERM problem. We show that for any $p \in (0,1)$, when $mn = \Omega(b^2\ln(d/p)/\delta^2)$ (i.e., when Lemma 3 is meaningful), the algorithm produces a solution $\mathbf{w}$ such that w.p. at least $1-p$,

$$1-(\mathbf{v}_1^\top\mathbf{w})^2 \;\leq\; \epsilon_{\text{ERM}}(p)) \cdot (1+o(1))\,, \tag{5}$$

where $\epsilon_{\text{ERM}}(p))$ is defined in Eq. (3). The algorithm performs overall $\tilde{O}(\sqrt{b}\delta^{-1/2}n^{-1/4})$ distributed matrix-vector products with the centralized empirical covariance matrix $\hat{\mathbf{X}}$ [1]. The $\tilde{O}(\cdot)$ notation hides poly-logarithmic factors in $1/p, 1/\delta, d, 1/\epsilon_{\text{ERM}}(p)$. See Theorem 6 in Section 4 for the complete and formal result.

We note that in particular, under our assumption that $mn = \tilde{\Omega}(b^2/\delta^2)$, it holds that the number of distributed matrix-vector products is upper bounded by $\tilde{O}(m^{1/4})$. Moreover, in the regime $n = \Omega(b^2\delta^{-2})$, we can see that the number of distributed matrix-vector products depends only poly-logarithmically on the problem parameters.

In general, the sub-constant $o(1)$ factor in (5) could be made arbitrarily small by trading the approximation error with the number of distributed matrix-vector products.

2.2.2. Previous algorithms

Distributed versions of classical iterative algorithms: Classical fast iterative algorithms for computing the leading eigenvector of a positive semidefinite matrix, such as the well-known Power Method and the Lanczos Algorithm, require iterative multiplications of the input matrix ($\hat{\mathbf{X}}$ in our case) with the current estimate. It is thus straightforward to implement these algorithms in our distributed setting, by multiplying the same vector with the covariance matrices at each machine, and averaging the result. Thus, by well-known convergence guarantees of these two methods, we will have that for a fixed $\epsilon > 0$, these methods produce a unit vector $\mathbf{w}$ such that, for any $p \in (0,1)$, $1-(\mathbf{w}^\top\hat{\mathbf{v}}_1)^2 \leq \epsilon$ w.p. at least $1-p$, after $O(\hat{\lambda}_1\hat{\delta}^{-1}\ln(d/p\epsilon))$ rounds for the Power Method and $O(\sqrt{\hat{\lambda}_1\hat{\delta}^{-1}}\ln(d/p\epsilon))$ for the Lanczos Algorithm, where $\hat{\lambda}_1, \hat{\delta}$ denote the leading eigenvalue and eigengap of $\hat{\mathbf{X}}$, respectively. Moreover, in the regime of mn in which Lemma 1 is meaningful, we can replace $\hat{\lambda}_1, \hat{\delta}$ with λ_1, δ in the above bounds, and the result will still hold with high probability.

Simple calculations show that in the regime of mn in which Lemma 1 is meaningful, it holds that our Shift-and-Invert-based algorithm outperforms distributed Lanczos (in terms of worst-case guarantees) whenever $n = \tilde{\Omega}(b^2/\lambda_1^2)$.

"Hot potato" SGD: Another straightforward approach is to apply a sequential algorithm for direct risk minimization that can process the data-points one by one, such as stochastic gradient descent (SGD), by passing its state from one machine to the next, after completing a full pass over the machine's data. Clearly, this process of making a full pass over the data of a certain machine before sending the final estimate to the next one, requires overall m communication rounds in order to make a full pass over all mn points. SGD for PCA was studied in several results in recent years (Balsubramani et al., 2013; Shamir, 2016a;c;

[1] i.e., on each round, each machine i sends the product of an input vector in $\mathbb{R}^d$ with its local covariance matrix $\hat{\mathbf{X}}_i$.

Jain et al., 2016a; Allen Zhu & Li, 2016b). For instance applying the result of (Jain et al., 2016a) in this way will result in a final estimate $\mathbf{w}$ satisfying

$$1 - (\mathbf{w}^\top \mathbf{v}_1)^2 \;=\; O\left(\frac{b^2 \ln d}{\delta^2 mn}\right) \quad \text{w.p. at least } 3/4. \qquad (6)$$

We note that in the regime in which the bound in (6) is meaningful it holds that the number of communication rounds of our Shift-and-Invert-based algorithm is upper-bounded by $\tilde{O}(m^{1/4})$ which for sufficiently large m dominates the communication complexity of SGD.

3. Single Communication Round Algorithms via ERM on Each Machine

In this section we consider distributed algorithms that require only a single round of communication. Naturally for this regime, all algorithms will be based on aggregating the ERM solutions of the individual machines, i.e., each machine i only sends the leading eigenvector of its empirical covariance matrix $\hat{\mathbf{X}}_i$ to a centralized machine (without loss of generality, machine 1) which it turn combines them to a single unit vector in some manner.

3.1. Simple averaging of eigenvectors fail

Perhaps the simplest method to aggregate the individual eigenvectors of each machine is to average them, and then normalize to obtain a unit vector. For instance, in the distributed statistical setting considered in (Zhang et al., 2013), in which the objective is *strongly convex*, it was shown that simply averaging the individual ERM solutions leads, in a meaningful regime of parameters, to estimation error of the order of the centralized ERM solution. However, here we show that for PCA, in which the objective is certainly not convex, this approach fails practically in any regime, in the sense that the error of the returned aggregated solution can be no better than that returned by any single machine.

Theorem 3. *There exists a distribution over vectors in $\mathbb{R}^2$ with ℓ_2 norm bounded by a universal constant for which the eigengap in the covariance matrix is 1 (i.e., $\delta = 1$), such that if each machine i returns an estimate $\hat{\mathbf{v}}_1^{(i)}$ which is an unbiased leading eigenvector of $\hat{\mathbf{X}}_i$ (i.e., both outcomes $-\hat{\mathbf{v}}_1^{(i)}, +\hat{\mathbf{v}}_1^{(i)}$ are equally likely), then the aggregated vector $\bar{\mathbf{v}}_1 = \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{v}}_1^{(i)}$ satisfies*

$$\forall m, n: \quad \mathbb{E}\left[1 - \left\langle \frac{\bar{\mathbf{v}}_1}{\|\bar{\mathbf{v}}_1\|}, \mathbf{v}_1 \right\rangle^2\right] \;=\; \Omega(1/n).$$

The proof is given in the appendix.

3.2. Averaging with Sign Fixing

As evident from the statement of Theorem 3, an important assumption is that each machine produces an unbiased estimate, in the sense that the sign of the outcome is uniform and independent of the other machines. This hints that correlating the signs of the different estimates can circumvent the lower bound result in Theorem 3. It turns out that this is indeed the case, as captured by the following theorem:

Theorem 4. *Let $\tilde{\mathbf{w}}_i$ be the leading eigenvector of $\hat{\mathbf{X}}_i$ for any $i \in [m]$, and consider the unit vector*

$$\mathbf{w} = \frac{\sum_{i=1}^m \operatorname{sign}(\tilde{\mathbf{w}}_i^\top \tilde{\mathbf{w}}_1)\tilde{\mathbf{w}}_i}{\|\sum_{i=1}^m \operatorname{sign}(\tilde{\mathbf{w}}_i^\top \tilde{\mathbf{w}}_1)\tilde{\mathbf{w}}_i\|}. \qquad (7)$$

Then, for any $p \in (0,1)$, it holds w.p. at least $1-p$ that

$$1 - (\mathbf{v}_1^\top \mathbf{w})^2 \;=\; O\left(\frac{b^2 \log\left(\frac{dm}{p}\right)}{\delta^2 mn} + \frac{b^4 \log^2\left(\frac{dm}{p}\right)}{\delta^4 n^2}\right).$$

For ease of presentation, throughout the rest of this section we denote the correlated vector $\hat{\mathbf{w}}_i = \operatorname{sign}(\tilde{\mathbf{w}}_i^\top \tilde{\mathbf{w}}_1)\tilde{\mathbf{w}}_i$ for any $i \in [m]$.

The main step towards proving Theorem 4 is to consider each $\hat{\mathbf{w}}_i$ as an approximately unbiased perturbation of the true leading eigenvector $\mathbf{v}_1$ and to upper bound the magnitude of this perturbation. This is carried out in the following much more general and self-contained lemma, which might be of independent interest. The proof is given in the appendix.

Lemma 2. *Let $\mathbf{A}$ be a positive semidefinite matrix with some fixed leading eigenvector $\mathbf{v}_1$, a leading eigenvalue λ_1 and an eigengap $\delta := \lambda_1(\mathbf{A}) - \lambda_2(\mathbf{A}) > 0$. Let $\hat{\mathbf{A}}$ be some positive semidefinite matrix such that $\|\hat{\mathbf{A}} - \mathbf{A}\| \leq \delta/4$. Then there is a unique leading eigenvector $\hat{\mathbf{v}}_1$ of $\hat{\mathbf{A}}$ such that $\langle \hat{\mathbf{v}}_1, \mathbf{v}\rangle \geq 0$, and*

$$\left\| \hat{\mathbf{v}}_1 - \mathbf{v}_1 - (\lambda_1 \mathbf{I} - \mathbf{A})^\dagger (\hat{\mathbf{A}} - \mathbf{A})\mathbf{v}_1 \right\| \;\leq\; \frac{c\|\hat{\mathbf{A}} - \mathbf{A}\|^2}{\delta^2},$$

where $\dagger$ denotes the pseudo-inverse, and c is a positive numerical constant.

Lemma 2 is central to the proof of the following Lemma, of which the proof of Theorem 4 is an easy consequence. We defer the proof of both the Lemma and that of Theorem 4 to the appendix.

Lemma 3. *The following two conditions hold with probability at least $1-p-d\exp(-\delta^2 n/cb^2)$, for some numerical constants $c, c' > 0$:*

- *The leading eigenvalue of every $\hat{\mathbf{X}}_i$ is simple, i.e., $\lambda_1(\hat{\mathbf{X}}_i) - \lambda_2(\hat{\mathbf{X}}_i) > 0$.*
- *Fixing $\mathbf{v}_1$, there exist unique leading eigenvectors $\hat{\mathbf{v}}_1^i, \ldots, \hat{\mathbf{v}}_m^i$ of $\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_m$, such that $\max_i \|\hat{\mathbf{v}}_1^i - \mathbf{v}_1\| \leq \frac{1}{4}$, and $\left\|\frac{1}{m}\sum_{i=1}^m \hat{\mathbf{v}}_1^i - \mathbf{v}_1\right\| \leq c'\left(\frac{b^2\log(2dm/p)}{\delta^2 n} + \sqrt{\frac{b^2\log(2dm/p)}{\delta^2 mn}}\right)$.*

3.3. Lower Bound for Sign Fixing

We now show that the result of Theorem 4 is tight up to poly-logarithmic factors and cannot be improved in general:

Theorem 5. *For any $\delta \in (0,1)$ and $d > 1$, there exist a distribution over vectors in $\mathbb{R}^d$ (of norm at most a universal constant) with eigengap δ in the covariance matrix, such that for any number of machines m and for per-machine sample size any n sufficiently larger than $1/\delta^2$, the aggregated vector $\bar{\mathbf{v}}_1 = \frac{1}{m}\sum_{i=1}^{m}\hat{\mathbf{v}}_1^{(i)}$ (even after sign fixing with the population eigenvector $\mathbf{v}_1$) satisfies*

$$\mathbb{E}\left[1 - \left\langle \frac{\bar{\mathbf{v}}_1}{\|\bar{\mathbf{v}}_1\|}, \mathbf{e}_1 \right\rangle^2\right] = \Omega\left(\frac{1}{\delta^2 mn} + \frac{1}{\delta^4 n^2}\right)$$

The proof is given in the appendix.

4. A Multi-round Algorithm based on Shift-and-Invert Iterations

In this section we move on to consider distributed algorithms that perform multiple communication rounds. The main motivation, beyond improving some poly-logarithmic factors in the estimation error, is to obtain a result that does not require the per-machine sample size n to grow with the number of machines m, as in the result of Theorem 4.

Towards this end we consider the use of the Shift-and-Invert meta-algorithm, originally described in (Garber & Hazan, 2015; Garber et al., 2016), to explicitly solve the centralized ERM objective, i.e., find a unit vector that is an approximate solution to $\max_{\mathbf{v}:\|\mathbf{v}\|=1}\mathbf{v}^\top\hat{\mathbf{X}}\mathbf{v}$.

Throughout this section we let $\hat{\lambda}_1, \hat{\delta}$ denote the leading eigenvalue and eigengap of $\hat{\mathbf{X}}$, respectively. Also, we assume without loss of generality that $b = 1$ (i.e., all data points lie in the unit Euclidean ball).

Since our approach is to approximate the population risk by approximating the empirical risk, we state the following simple lemma for completeness (a proof is given in the appendix).

Lemma 4 (Risk of approximated-ERM for PCA)**.** *Let $\mathbf{w}$ be a unit vector such that $(\mathbf{w}^\top\hat{\mathbf{v}}_1)^2 \geq 1 - \epsilon$, for some fixed $\epsilon > 0$, where $\hat{\mathbf{v}}_1$ is the leading eigenvector of $\hat{\mathbf{X}}$. Then it holds that $1 - (\mathbf{w}^\top\mathbf{v}_1)^2 \leq 1 - (\mathbf{w}^\top\hat{\mathbf{v}}_1)^2 + \sqrt{2\epsilon}$.*

4.1. The Shift-and-Invert meta-algorithm

The Shift-and-Invert algorithm (Garber & Hazan, 2015; Garber et al., 2016) efficiently reduces the problem of computing the leading eigenvector of a positive semidefinite matrix $\hat{\mathbf{X}}$ to that of approximately-solving a poly-logarithmic number of linear systems, i.e., finding approximate minimizers of convex quadratic optimization problems of the form

$$\min_{\mathbf{z}\in\mathbb{R}^d}\{F_{\lambda,\mathbf{w}}(\mathbf{z}) := \frac{1}{2}\mathbf{z}^\top(\lambda\mathbf{I} - \hat{\mathbf{X}})\mathbf{z} - \mathbf{z}^\top\mathbf{w}\}, \qquad (8)$$

where $\lambda > \lambda_1(\hat{\mathbf{X}})$ is a shifting parameter. The algorithm is essentially based on applying *power iterations* to a shifted and inverted matrix $(\lambda\mathbf{I} - \hat{\mathbf{X}})^{-1}$, where the shifting parameter λ is carefully chosen. The algorithm that implements this reduction, originally described in (Garber & Hazan, 2015), is given below (see Algorithm 1).

Algorithm 1 Shift-and-Invert Power Method

```
Input: estimate $\tilde{\delta}$ for the gap $\hat{\delta}$, accuracy $\epsilon \in (0,1)$, failure probability $p$
Set: $m_1 \leftarrow \lceil 8\ln\left(144d/p^2\right)\rceil, m_2 \leftarrow \lceil\frac{3}{2}\ln\left(\frac{18d}{p^2\epsilon}\right)\rceil$
Set: $\tilde{\epsilon} \leftarrow \min\left\{\frac{1}{16}\left(\tilde{\delta}/8\right)^{m_1+1}, \frac{\epsilon}{4}\left(\tilde{\delta}/8\right)^{m_2+1}\right\}$
Set: $\lambda_{(0)} \leftarrow 1 + \tilde{\delta}$ ,  $\hat{\mathbf{w}}_0 \leftarrow$ random unit vector,  $s \leftarrow 0$
repeat
    $s \leftarrow s + 1$ ,  $\mathbf{M}_s \leftarrow (\lambda_{(s-1)}\mathbf{I} - \hat{\mathbf{X}})$
    for $t = 1...m_1$ do
        Find approx. minimizer - $\hat{\mathbf{w}}_t$ of $F_{\lambda_{(s-1)},\hat{\mathbf{w}}_{t-1}}(\mathbf{z})$
        such that $\|\hat{\mathbf{w}}_t - \mathbf{M}_s^{-1}\hat{\mathbf{w}}_{t-1}\| \leq \tilde{\epsilon}$
    end for
    $\mathbf{w}_s \leftarrow \hat{\mathbf{w}}_{m_1}/\|\hat{\mathbf{w}}_{m_1}\|$
    Find approx. minimizer - $\mathbf{v}_s$ of $F_{\lambda_{(s-1)},\mathbf{w}_s}(\mathbf{z})$ such
    that $\|\mathbf{v}_s - \mathbf{M}_s^{-1}\mathbf{w}_s\| \leq \tilde{\epsilon}$
    $\Delta_s \leftarrow \frac{1}{2}\cdot\frac{1}{\mathbf{w}_s^\top\mathbf{v}_s - \tilde{\epsilon}}$ ,  $\lambda_{(s)} \leftarrow \lambda_{(s-1)} - \frac{\Delta_s}{2}$
until $\Delta_s \leq \tilde{\delta}$
$\lambda_{(f)} \leftarrow \lambda_{(s)}$ ,  $\mathbf{M}_f \leftarrow (\lambda_{(f)}\mathbf{I} - \hat{\mathbf{X}})$
for $t = 1...m_2$ do
    Find approx. minimizer - $\hat{\mathbf{w}}_t$ of $F_{\lambda_{(f)},\hat{\mathbf{w}}_{t-1}}(\mathbf{z})$ such
    that $\|\hat{\mathbf{w}}_t - \mathbf{M}_f^{-1}\hat{\mathbf{w}}_{t-1}\| \leq \tilde{\epsilon}$
end for
Return: $\mathbf{w}_f \leftarrow \hat{\mathbf{w}}_{m_2}/\|\hat{\mathbf{w}}_{m_2}\|$
```

Lemma 5 (Efficient reduction of top eigenvector to convex optimization; originally Theorem 4.2 in (Garber & Hazan, 2015))**.** *Suppose that $\hat{\delta} := \lambda_1(\hat{\mathbf{X}}) - \lambda_2(\hat{\mathbf{X}}) > 0$ and suppose that the estimate $\tilde{\delta}$ in Algorithm 1 satisfies $\tilde{\delta} \in [\hat{\delta}/2, 3\hat{\delta}/4]$. Then, with probability at least $1 - p$, Algorithm 1 finds a unit vector $\mathbf{w}_f$ such that $(\mathbf{w}_f^\top\hat{\mathbf{v}}_1)^2 \geq 1 - \epsilon$, and the total number of optimization problems of the form (8) solved during the run of the algorithm, is upper bounded by $O\left(\ln(d/p)\ln(\hat{\delta}^{-1}) + \ln\left(\frac{d}{p\epsilon}\right)\right)$. Moreover, throughout the run of the algorithm it holds that $1 + \hat{\delta} \geq \lambda_{(s)} - \hat{\lambda}_1 = \Omega(\hat{\delta})$.*

Remark: the purpose of the *repeat-until* loop in Algorithm 1 is to efficiently find a shifting parameter $\lambda_{(f)}$ such that $c_1\hat{\delta} \leq \lambda_{(f)} - \hat{\lambda}_1 \leq c_2\hat{\delta}$ for some universal constants $c_2 > c_1 > 0$. When n satisfies $n = \Omega(\delta^{-2}\ln(d/p))$, we can directly find (w.h.p) such a shifting parameter, by simply estimating $\hat{\lambda}_1, \hat{\delta}$ from the data of a *single machine*. Also, we can take $\hat{\mathbf{w}}_0$ to be the leading eigenvector of any single machine, since this will already have a constant correlation with $\hat{\mathbf{v}}_1$. Thus, for such n, the total number of optimization problems can be reduced to $O(\ln(p^{-1}\epsilon^{-1}))$.

Algorithm 1 is a meta-algorithm in the sense that the choice of solver for the optimization problems $\min F_{\lambda,\mathbf{w}}$ is unspecified, and any solver will do. A simple calculation shows that a naive application of either the conjugate gradient method or Nesterov's accelerated gradient method to solve these optimization problems in a distributed manner, i.e., the computation of the gradient vector is distributed across machines, will require overall $\tilde{O}\big(\sqrt{\hat{\lambda}_1/\hat{\delta}}\big)$ communication rounds, which does not give any improvement over the distributed Lanczos approach, described in Subsection 2.2.2. However, this can be substantially improved by taking advantage of the fact that the data on all machines is sampled i.i.d. from the same distribution. In particular, we present below an approach based on applying a preconditioner to the optimization Problem (8), in the spirit of the one described in (Zhang & Lin, 2015).

4.2. Faster Distributed Approximation of Linear Systems via Local Preconditioning

Let $\mathbf{M} = \lambda\mathbf{I} - \hat{\mathbf{X}}$, for some shift parameter $\lambda > \hat{\lambda}_1$, and define the pre-conditioning matrix $\mathbf{C} = (\lambda+\mu)\mathbf{I} - \hat{\mathbf{X}}_1$, where μ is required so $\mathbf{C}$ is invertible. Consider now solving the following modified quadratic problem:

$$\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y}) := \frac{1}{2}\mathbf{y}^\top\mathbf{C}^{-1/2}\mathbf{M}\mathbf{C}^{-1/2}\mathbf{y} - \mathbf{y}^\top\mathbf{C}^{-1/2}\mathbf{w}. \quad (9)$$

Note that if $\mathbf{y}^*$ is the optimal solution to Problem (9), i.e.,

$$\mathbf{y}^* = \mathbf{C}^{1/2}\mathbf{M}^{-1}\mathbf{C}^{1/2}\mathbf{C}^{-1/2}\mathbf{w} = \mathbf{C}^{1/2}\mathbf{M}^{-1}\mathbf{w},$$

then $\mathbf{z}^* := \mathbf{C}^{-1/2}\mathbf{y}^*$ is the optimal solution to Problem (8).

The idea behind choosing $\mathbf{C}$ this way is very intuitive. Ideally we could have chosen $\mathbf{C} = \mathbf{M}$, making the condition number of $\tilde{F}_{\lambda,\mathbf{w}}$ equal to $\kappa(\tilde{F}_{\lambda,\mathbf{w}}) = 1$, which is the best we can hope for. The problem of course is that this requires us to explicitly compute $\mathbf{M}^{-1/2}$, which is more challenging then just computing the leading eigenvector of $\hat{\mathbf{X}}$. The next best thing is thus to choose $\mathbf{C}$ based only on the data available on any single machine, which allows computing $\mathbf{C}^{-1/2}$ without additional communication overhead, and leads to the choice described above. The following lemma, rephrased from (Zhang & Lin, 2015), quantifies exactly how such a choice of $\mathbf{C}$ helps in improving the condition number of the new optimization problem, Problem (9). The proof is given in the appendix.

Lemma 6. *Suppose that $\mu \geq \|\hat{\mathbf{X}} - \hat{\mathbf{X}}_1\|$. Then, $\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y})$ is 1-smooth and $\left(\frac{\lambda-\hat{\lambda}_1}{(\lambda-\hat{\lambda}_1)+2\mu}\right)$-strongly convex. In particular, $\kappa(\tilde{F}_{\lambda,\mathbf{w}}) \leq 1 + 2\mu/(\lambda-\hat{\lambda}_1)$. Moreover, fixing $\tilde{\mathbf{y}} \in \mathbb{R}^d$, if we let $\tilde{\mathbf{z}} := \mathbf{C}^{-1/2}\tilde{\mathbf{y}}$, then it holds that $\|\tilde{\mathbf{z}} - \mathbf{M}^{-1}\mathbf{w}\| \leq (\lambda - \hat{\lambda}_1)^{-1/2}\|\tilde{\mathbf{y}} - \mathbf{C}^{1/2}\mathbf{M}^{-1}\mathbf{w}\|$. In particular, for any $p \in (0,1)$, if we set $\mu = 4\sqrt{\ln(d/p)/n}$, then the above holds with probability at least $1-p$, where this probability depends only on the randomness in $\hat{\mathbf{X}}_1$.*

4.2.1. Solving the pre-conditioned linear systems

We now discuss the application of gradient-based algorithms for finding an approximate minimizer of the preconditioned problem, Problem (9), in our distributed setting. Towards this end we require a *distributed* implementation for the *first-order oracle* of $\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y})$ (i.e., computation of the value and gradient vector at a queried point).

A straight-forward implementation of the first-order oracle in our distributed setting is given in Algorithm 2.

Algorithm 2 Distributed First-Order Oracle for $\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y})$

```
Input: shift parameter $\lambda > 0$, regularization parameter $\mu > 0$, vector $\mathbf{w} \in \mathbb{R}^d$, query vector $\mathbf{y} \in \mathbb{R}^d$
send $\tilde{\mathbf{y}} := \mathbf{C}^{-1/2}\mathbf{y}$ to machines $\{2,\dots,m\}$ for $\mathbf{C} := (\lambda+\mu)\mathbf{I} - \hat{\mathbf{X}}_1$ {executed on machine 1}
for $i = 1...m$ do
    send $\tilde{\nabla}_i := \hat{\mathbf{X}}_i\tilde{\mathbf{y}}$ to machine 1 {executed on each machine $i$}
end for
aggregate $\tilde{\nabla} := \frac{1}{m}\sum_{i=1}^m \tilde{\nabla}_i$ {executed on machine 1}
compute $\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y}) = \frac{1}{2}(\lambda\mathbf{y}^\top\mathbf{C}^{-1}\mathbf{y} - \mathbf{y}^\top\mathbf{C}^{-1/2}\tilde{\nabla}) - \mathbf{y}^\top\mathbf{C}^{-1/2}\mathbf{w}$ {executed on machine 1}
compute $\nabla\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y}) = \lambda\mathbf{C}^{-1}\mathbf{y} - \mathbf{C}^{-1/2}\tilde{\nabla} - \mathbf{C}^{-1/2}\mathbf{w}$ {executed on machine 1}
return: $(\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y}), \nabla\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y}))$
```

We have the following lemma, the proof of which is deferred to the appendix.

Lemma 7. *Fix some $\lambda > \lambda_1(\hat{\mathbf{X}})$ and $\mathbf{w} \in \mathbb{R}^d$, and let $1 \geq \mu > 0$ be as in Lemma 6. Fix $\epsilon > 0$. Consider the following two-step algorithm:*

1. *Apply either the conjugate gradient method or Nesterov's accelerated method with the distributed first-order oracle described in Algorithm 2 to find $\tilde{\mathbf{y}} \in \mathbb{R}^d$ such that $\tilde{F}_{\lambda,\mathbf{w}}(\tilde{\mathbf{y}}) - \min_{\mathbf{y}\in\mathbb{R}^d}\tilde{F}_{\lambda,\mathbf{w}}(\mathbf{y}) \leq \epsilon'$*
2. *Return $\tilde{\mathbf{z}} = \mathbf{C}^{-1/2}\tilde{\mathbf{y}}$.*

Then, for $\epsilon' = \frac{\epsilon}{2}\left(1 + \frac{2\mu}{\lambda-\hat{\lambda}_1}\right)^{-1}(\lambda - \hat{\lambda}_1)$ it holds that $\|\tilde{\mathbf{z}} - (\lambda\mathbf{I} - \hat{\mathbf{X}}_1)^{-1}\mathbf{w}\| \leq \epsilon$, and the total number distributed matrix-vector products with the empirical covariance matrix $\hat{\mathbf{X}}$ required to compute $\tilde{\mathbf{z}}$ is upper-bounded by

$$O\left(\sqrt{1 + 2\mu(\lambda-\hat{\lambda}_1)^{-1}}\ln\left(\left(1 + \frac{2\mu}{\lambda-\hat{\lambda}_1}\right)\|\mathbf{w}\|/[(\lambda-\hat{\lambda}_1)\epsilon]\right)\right).$$

4.3. Putting it all together

We now state our main result for this section, which is a simple consequence of the previous lemmas. The full proof is given in the appendix.

Theorem 6. *Fix $\epsilon \in (0,1)$ and $p \in (0,1)$. Suppose that $mn = \Omega(\delta^{-2}\ln(d/p))$. Set $\mu = 4\sqrt{\ln(3d/p)/n}$. Applying the Shift-and-Invert algorithm, Algorithm 1, with the parameters $\epsilon, p/3$, and applying the algorithm in Lemma 7 with the parameter μ, to approximately solve the linear systems, yields with probability at least $1-p$ a unit vector $\mathbf{w}_f$ such that $(\mathbf{w}_f^\top \hat{\mathbf{v}}_1)^2 \geq 1-\epsilon$, after executing at most*

$$O\left(\sqrt{\frac{\sqrt{\ln(d/p)}}{\delta\sqrt{n}}}\left[\ln\left(\frac{d}{p\epsilon^2}\right)\ln\left(\frac{\sqrt{\ln(d/p)}}{\delta^2\sqrt{n}}\right) + \ln^2\left(\frac{d}{p\epsilon^2}\right)\ln\left(\frac{1}{\delta}\right)\right]\right) = \tilde{O}\left(\sqrt{\frac{1}{\delta\sqrt{n}}}\right)$$

distributed matrix-vector products with the empirical covariance matrix $\hat{\mathbf{X}}$.

5. Experiments

To validate some of our theoretical findings we conducted experiments with single-round algorithms on synthetic data. We generated synthetic datasets using two distributions. For both distributions we used the covariance matrix $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top$ with $\mathbf{U}$ being a random $d \times d$ orthonormal matrix and $\boldsymbol{\Sigma}$ is diagonal satisfying: $\boldsymbol{\Sigma}(1,1) = 1$, $\boldsymbol{\Sigma}(2,2) = 0.8$, $\forall j \geq 3: \boldsymbol{\Sigma}(j,j) = 0.9 \cdot \boldsymbol{\Sigma}(j-1,j-1)$, i.e., $\delta = 0.2$. One dataset was generated according to the normal distributions $\mathcal{N}(0,\mathbf{X})$, and for the second datasets we generated samples by taking $\mathbf{x} = \sqrt{3/2}\mathbf{X}^{1/2}\mathbf{y}$ where $\mathbf{y} \sim U[-1,1]$. In both cases we set $d = 300$.

Beyond the single-round algorithms that are based on aggregating the individual ERM solutions described so far, we propose an additional natural aggregation approach, based on aggregating the individual projection matrices. More concretely, letting $\{\hat{\mathbf{v}}_1^{(i)}\}_{i=1}^m$ denote the leading eigenvectors of the individual machines, let $\bar{\mathbf{P}}_1 := \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{v}}_1^{(i)}\hat{\mathbf{v}}_1^{(i)\top}$. We then take the final estimate $\mathbf{w}$ to be the leading eigenvector of the aggregated matrix $\bar{\mathbf{P}}_1$. Note that as with the sign-fixing based aggregation, this approach also resolves the sign-ambiguity in the estimates produced by the different machines, which circumvents the lower bound result of Theorem 3.

For both datasets we fixed the number of machines to $m = 25$. We tested the estimation error (i.e., the value $1-(\mathbf{w}^\top\mathbf{v}_1)^2$ where $\mathbf{v}_1$ is the leading eigenvector of $\mathbf{X}$ and $\mathbf{w}$ is the estimator) of five benchmarks vs. the per-machine sample size n: the centralized solution $\hat{\mathbf{v}}_1$, the average of the individual (unbiased) ERM solutions (normalized to unit norm),the average of ERM solutions with sign-fixing, and the leading eigenvector of the averaged projection matrix. We also plotted the average loss of the individual ERM solutions. Results are averaged over 400 independent runs.

The results for the normal distribution appear in Figure 1. The results for the uniform-based distribution are very similar and are deferred to the appendix. We can see that, as our lower bound in Theorem 3 suggests, simply averaging and normalizing the individual ERM solutions has significantly worse performance than the centralized ERM solution. Perhaps surprisingly, the performance of this estimator is even worse than the average error of an estimate computed using only a single machine. We see that both aggregation methods that are based on correlating the individual ERM solutions, namely the sign-fixing-based estimator, and the proposed averaging-of-projections heuristic, are asymptotically consistent with the centralized ERM. In particular, the averaging-of-projections scheme, at least empirically, significantly outperforms the sign-fixing approach, which justifies further theoretical investigation of this heuristic. For the sign fixing approach, we can see that as suggested by our bounds, the estimator is not consistent with the centralized ERM solution for small values of n.

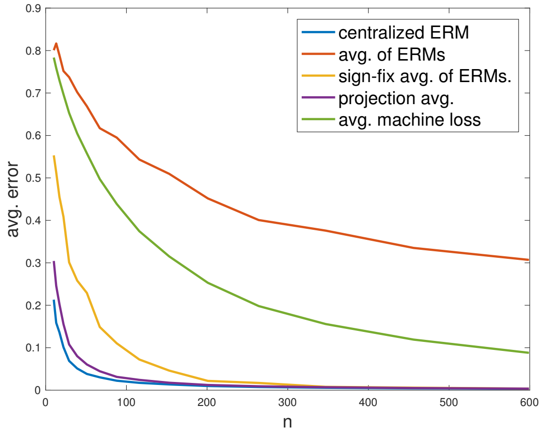

Figure 1. Estimation error vs. the per-machine sample size n for a normal distribution.

6. Discussion

We presented communication-efficient algorithms for distributed statistical estimation of principal components. Focusing on our results for methods based on a single communication round, we initiated a study of how to correctly aggregate distributed ERM solutions in a non-convex setting. An important take-home message of our work is that in a non-convex setting, simply averaging the local solutions is not a good idea. On the positive side, we show that a very simple correction (i.e., sign-fixing) is possible by leveraging the specific structure of the problem at hand. It is thus interesting to develop a richer theory of how to perform such aggregations in more involved non-convex problems.

References

Eigenvalues and eigenvectors of 2x2 matrices. http://www.math.harvard.edu/archive/21b_fall_04/exhibits/2dmatrices/.

Allen Zhu, Zeyuan and Li, Yuanzhi. Even faster SVD decomposition yet without agonizing pain. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 974–982, 2016a.

Allen Zhu, Zeyuan and Li, Yuanzhi. Fast global convergence of online PCA. *CoRR*, abs/1607.07837, 2016b.

Balsubramani, Akshay, Dasgupta, Sanjoy, and Freund, Yoav. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pp. 3174–3182, 2013.

Boutsidis, Christos, Woodruff, David P, and Zhong, Peilin. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 236–249. ACM, 2016.

Garber, Dan and Hazan, Elad. Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

Garber, Dan, Hazan, Elad, Jin, Chi, Kakade, Sham M., Musco, Cameron, Netrapalli, Praneeth, and Sidford, Aaron. Faster eigenvector computation via shift-and-invert preconditioning. *CoRR*, abs/1605.08754, 2016.

Golub, Gene H and Pereyra, Victor. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.*, 24, 1933.

Jaggi, Martin, Smith, Virginia, Takác, Martin, Terhorst, Jonathan, Krishnan, Sanjay, Hofmann, Thomas, and Jordan, Michael I. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pp. 3068–3076, 2014.

Jain, Prateek, Jin, Chi, Kakade, Sham M, Netrapalli, Praneeth, and Sidford, Aaron. Matching matrix bernstein with little memory: Near-optimal finite sample guarantees for oja's algorithm. *arXiv preprint arXiv:1602.06929*, 2016a.

Jain, Prateek, Jin, Chi, Kakade, Sham M, Netrapalli, Praneeth, and Sidford, Aaron. Matching matrix bernstein with little memory: Near-optimal finite sample guarantees for oja's algorithm. *arXiv preprint arXiv:1602.06929*, 2016b.

Jolliffe, IT. Principal component analysis. 2002. *Springverlag, New York*, 2002.

Lee, Jason D., Ma, Tengyu, and Lin, Qihang. Distributed stochastic variance reduced gradient methods. *CoRR*, abs/1507.07595, 2015.

Liang, Yingyu, Balcan, Maria-Florina F, Kanchanapally, Vandana, and Woodruff, David. Improved distributed principal component analysis. In *NIPS*, 2014.

Magnus, Jan R. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1(02):179–191, 1985.

Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

Reddi, Sashank J., Konecný, Jakub, Richtárik, Peter, Póczos, Barnabás, and Smola, Alexander J. AIDE: fast and communication efficient distributed optimization. *CoRR*, abs/1608.06879, 2016.

Shamir, Ohad. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 144–152, 2015.

Shamir, Ohad. Convergence of stochastic gradient descent for PCA:. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 257–265, 2016a.

Shamir, Ohad. Without-replacement sampling for stochastic gradient methods. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 46–54, 2016b.

Shamir, Ohad. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 248–256, 2016c.

Shamir, Ohad, Srebro, Nathan, and Zhang, Tong. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1000–1008, 2014.

Tropp, Joel A. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Yu, Yi, Wang, Tengyao, and Samworth, Richard J. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Zhang, Yuchen and Lin, Xiao. Disco: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 362–370, 2015.

Zhang, Yuchen, Duchi, John C, and Wainwright, Martin J. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14: 3321–3363, 2013.