The Annals of Statistics
2021, Vol. 49, No. 2, 918–943
https://doi.org/10.1214/20-AOS1984

DISTRIBUTED LINEAR REGRESSION BY AVERAGING

BY EDGAR DOBRIBAN[1] AND YUE SHENG[2]

[1]*Department of Statistics, University of Pennsylvania, dobriban@wharton.upenn.edu*
[2]*Graduate Group in Applied Mathematics and Computational Science, Department of Mathematics, University of Pennsylvania, yuesheng@sas.upenn.edu*

Distributed statistical learning problems arise commonly when dealing with large datasets. In this setup, datasets are partitioned over machines, which compute locally, and communicate short messages. Communication is often the bottleneck. In this paper, we study one-step and iterative *weighted parameter averaging* in statistical *linear models* under *data parallelism*. We do linear regression on each machine, send the results to a central server and take a weighted average of the parameters. Optionally, we iterate, sending back the weighted average and doing local ridge regressions centered at it. How does this work compared to doing linear regression on the full data? Here, we study the performance loss in *estimation* and *test error*, and *confidence interval length* in high dimensions, where the number of parameters is comparable to the training data size.

We find the performance loss in one-step weighted averaging, and also give results for iterative averaging. We also find that different problems are affected differently by the distributed framework. Estimation error and confidence interval length increases a lot, while prediction error increases much less. We rely on recent results from random matrix theory, where we develop a new calculus of deterministic equivalents as a tool of broader interest.

1. Introduction. Datasets are constantly increasing in size and complexity. This leads to important challenges for practitioners. Statistical inference and machine learning, which used to be computationally convenient on small datasets, now bring an enormous computational burden.

Distributed computation is a universal approach to deal with large datasets. Datasets are partitioned across several machines (or workers). The machines perform computations locally and communicate only small bits of information with each other. They coordinate to compute the desired quantity. This is the standard approach taken at large technology companies, which routinely deal with huge datasets spread over computing units. What are the best ways to divide up and coordinate the work?

The same problem arises when the data is distributed due to privacy, security or ethical concerns. For instance, medical and healthcare data is typically distributed across hospitals or medical units. The parties agree that they want to aggregate the results. At the same time, they do not want other parties access their data. How can they compute the desired aggregates, without sharing the data?

In both cases, the key question is how to do statistical estimation and machine learning in a distributed setting. And what performance can the best methods achieve? This is a question of broad interest, and it is expected that the area of distributed estimation and computation will grow even more in the future.

In this paper, we develop precise theoretical answers to fundamental questions in distributed estimation. We study *one-step and iterative parameter averaging* in statistical *linear*

Received October 2019; revised June 2020.
MSC2020 subject classifications. Primary 62J05; secondary 65Y05, 68W10, 68W15.
Key words and phrases. Linear regression, distributed learning, parallel computation, random matrix theory, high dimensional.

models under *data parallelism*. Specifically, suppose in the simplest case that we do linear regression (Ordinary Least Squares, OLS) on each subset of a dataset distributed over k machines, and take an optimal weighted average of the regression coefficients. How do the statistical and predictive properties of this estimator compare to doing OLS on the full data?

We study the behavior of several learning and inference problems, such as *estimation error*, *test error* (i.e., out-of-sample *prediction error*) and *confidence intervals*. We also consider a high-dimensional (or proportional-limit) setting where the number of parameters is of the same order as the number of total samples (i.e., the size of the training data). We also study an analogous iterative algorithm, where we do local ridge regressions, take averages of the parameters on a central machine, send back the update to the local machines, and then again do local ridge, but where the penalty is centered around the previous mean. Our iterative algorithm falls between several classical methods such as ADMM and DANE, and we discuss connections.

We discover the following key phenomena, some of which are surprising in the context of existing work:

1. *Suboptimality*. One-step averaging is not optimal (even with optimal weights), meaning that it leads to a performance decay. In contrast to some recent work (see the related work section), we find that there is a clear performance loss due to one-step averaging *even if we split the data only into two subsets*. This loss is because the number of parameters is of the same order as the sample size. However, we can quantify this loss precisely.
2. *Strong problem-dependence*. Different learning and inference problems are affected differently by the distributed framework. Specifically, *estimation error and the length of confidence intervals increases a lot, while prediction error increases less*. The intuition is that prediction is a noisy task, and hence the extra error incurred is relatively smaller.
3. *Simple form and universality*. The asymptotic efficiencies for one step distributed learning have simple forms that are often *universal*. Specifically, they do not depend on the covariance matrix of the data, or on the sample sizes on the local machines. For instance, the estimation efficiency *decreases linearly in the number of machines k* (see Figure 1 and Table 1).
4. *Iterative parameter averaging has benefits*. We show that simple iterative parameter averaging mechanisms can reduce the error efficiently. We also exhibit computation-statistics

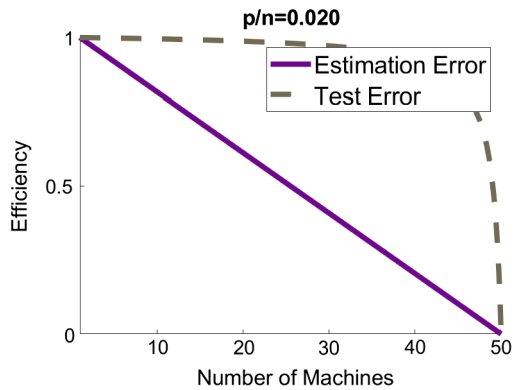

FIG. 1. *How much accuracy do we lose in distributed regression? The plots show the relative efficiency, that is, the ratio of errors, of the global least squares (OLS) estimator, compared to the distributed estimator averaging the local least squares estimators. This efficiency is at most unity, because the global estimator is more accurate. If the efficiency is close to unity, then averaging is accurate. We show the behavior of estimation and test error, as a function of number of machines. We see that estimation error is* much more affected *than test error. The specific formulas are given in Table 1.*

TABLE 1
Estimation, confidence interval and test efficiency as a function of number of machines k, the sample size n, and the dimension p. This is how much smaller the error of the global estimator is compared to the distributed estimator. These functions are plotted and described in Figure 1

Quantity	Relative efficiency (n, p, k)
Estimation and CIs	$\frac{n-kp}{n-p}$
Test error	$\frac{1}{1+\frac{p^2(k-1)}{n(n-kp)}}$

tradeoffs: depending on the hyperparameters, we can converge fast to statistically suboptimal solutions or vice versa.

While there is already a lot of work in this direction (see Section 2) our results are new and complementary. The key elements of novelty of our setting are: (1) The sample size and the dimension are comparable, and we do not assume sparsity. (2) We have a new mathematical approach, using recent results from asymptotic random matrix theory such as (Rubio and Mestre (2011)). Our approach also develops a novel theoretical tool, the *calculus of deterministic equivalents*, and we illustrate how it can be useful in other problems as well. (3) We consider several accuracy metrics (estimation, prediction) in a unified framework of so-called general linear functionals.

The code for our paper is available at http://www.github.com/dobriban/dist.

2. Some related work. In this section, we discuss some related work. There is a great deal of work in computer science and optimization on parallel and distributed computation (see, e.g., Bertsekas and Tsitsiklis (1989), Boyd et al. (2011), Bekkerman, Bilenko and Langford (2011)). In addition, there are several popular examples of distributed data processing frameworks: for instance, MapReduce (Dean and Ghemawat (2008)) and Spark (Zaharia et al. (2010)).

In contrast, there is less work on understanding the statistical properties, and the inherent computation-statistics tradeoffs, in distributed computation environments. This area has attracted increasing attention only in recent years; see, for instance, Mcdonald et al. (2009), Zhang, Wainwright and Duchi (2012), Zhang, Duchi and Wainwright (2013a, 2013b, 2015), Duchi et al. (2014), Braverman et al. (2016), Jordan, Lee and Yang (2019), Rosenblatt and Nadler (2016), Smith et al. (2017), Fan et al. (2019), Lin, Guo and Zhou (2017), Lee et al. (2017), Battey et al. (2018), Zhu and Lafferty (2018) and the references therein. See Huo and Cao (2019) for a review. We can only discuss the most closely related papers due to space limitations.

Zinkevich, Langford and Smola (2009) study the parallelization of SGD for learning, by reducing it to the study of delayed SGD; giving positive results for low latency "multicore" settings. They give an insightful discussion of the impact of various computational platforms, such as shared memory architectures, clusters and grid computing. Mcdonald et al. (2009) propose averaging methods for special conditional maximum entropy models, showing variance reduction properties. Zinkevich et al. (2010) expand on this, proposing "parallel SGD" to average the SGD iterates computed on random subsets of the data. Their proof is based on the contraction properties of SGD.

Zhang, Duchi and Wainwright (2013b) bound the leading order term for MSE of averaged estimation in empirical risk minimization. Their bounds do not explicitly take dimension into

account. However, their empirical data example clearly has large dimension p, considering a logistic regression with sample size $n = 2.4 \cdot 10^8$, and $p = 740{,}000$, so that $n/p \approx 340$. In their experiments, they distribute the data over up to 128 machines. So, our regime, where k is of the same order as n/p, matches well their simulation setup. In addition, their concern is on regularized estimators, where they propose to estimate and reduce bias by subsampling.

Liu and Ihler (2014) study distributed estimation in statistical exponential families, connecting the efficiency loss from the global setting to the deviation from full exponential families. They also propose nonlinear KL-divergence-based combination methods, which can be more efficient than linear averaging.

Zhang, Duchi and Wainwright (2015) study divide and conquer kernel ridge regression, showing that the partition-based estimator achieves the statistical minimax rate over all estimators. Due to their generality, their results are more involved, and also their dimension is fixed. Lin, Guo and Zhou (2017) improve those results. Duchi et al. (2014) derive minimax bounds on distributed estimation where the number of bits communicated is controlled.

Rosenblatt and Nadler (2016) consider the distributed learning problem in three different settings. The first two settings are fixed dimensional. The third setting is high-dimensional M-estimation, where they study the first-order behavior of estimators using prior results from Donoho and Montanari (2016), El Karoui et al. (2013). This is possibly the most closely related work to ours in the literature. They use the following representation, derived in the previous works mentioned above: a high-dimensional M-estimator can be written as $\hat{\beta} = \beta + r(\gamma)\Sigma^{-1/2}\zeta(1 + o_P(1))$, where $\zeta \sim \mathcal{N}(0, I_p/p)$, γ is the limit of p/n, and $r(\gamma)$ is a constant depending on the loss function, whose expression can be found in Donoho and Montanari (2016), El Karoui et al. (2013).

They derive a relative efficiency formula in this setting, which for OLS takes the form

$$\frac{\mathbb{E}\|\hat{\beta}_{\text{dist}} - \beta\|^2}{\mathbb{E}\|\hat{\beta} - \beta\|^2} = 1 + \gamma(1 - 1/k) + O(\gamma^2).$$

In contrast, our result for this case (Theorem 5.1) is equal to

$$\frac{1-\gamma}{1-k\gamma} = 1 + \gamma\frac{k-1}{1-k\gamma}.$$

Thus, our result is much more precise, and in fact exact, while of course being limited to the special case of linear regression.

In a heterogeneous data setting, Zhao, Cheng and Liu (2016) fit partially linear models, and estimate the common part by averaging. For model selection problems in GLM, Chen and Xie (2014) propose weighted majority voting methods. Lee et al. (2017) study sparse linear regression, showing that averaging debiased lasso estimators can achieve the optimal estimation rate if the number of machines is not too large. Battey et al. (2018) study a similar problem, also including hypothesis testing under more general sparse models. Shi, Lu and Song (2018), Banerjee, Durot and Sen (2019) show that in problems with nonstandard rates, averaging can lead to improved pointwise inference, while decreasing performance in a uniform sense. Volgushev, Chao and Cheng (2019) (among other contributions) provide conditions under which averaging quantile regression estimators have an optimal rate. Banerjee and Durot (2018) propose improvements based on communicating smoothed data, and fitting estimators after. Szabo and van Zanten (2018) study estimation methods under communication constraints in nonparametric random design regression model, deriving both minimax lower bounds and optimal methods.

See Section 7 for more discussion of multiround methods.

3. One-step weighted averaging: General linear functionals. We consider the standard linear model

$$Y = X\beta + \varepsilon.$$

Here, we have an outcome variable y along with some p covariates $x = (x^1, \ldots, x^p)^\top$, and want to understand their relationship. We observe n such data points, arranging their outcomes into the $n \times 1$ vector Y, and their covariates into the $n \times p$ matrix X. We assume that Y depends linearly on X, via some unknown $p \times 1$ parameter vector β.

We assume there are more samples than training data points, that is, $n > p$, while p can also be large. In that case, a gold standard is the usual least squares estimator (ordinary least squares or OLS)

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y.$$

We also assume that the coordinates of the noise ε are uncorrelated and have variance σ^2.

Suppose now that the samples are distributed across k machines (these can be real machines, but they can also be—say—sites or hospitals in medical applications, or mobile devices in federated learning). The ith machine has the $n_i \times p$ matrix X_i, containing n_i samples, and also the $n_i \times 1$ vector Y_i of the corresponding outcomes for those samples. Thus, the ith worker has access to only a subset of training n_i data points out of the total of n training data points. For instance, if the data points denote n users, then they may be partitioned into k sets based on country of residence, and we may have n_1 samples from the United States on one server, n_2 samples from Canada on another server, etc. The broad question is: How can we estimate the unknown regression parameter β if we need to do most of the computations locally?

Let us write the partitioned data as

$$X = \begin{bmatrix} X_1 \\ \cdots \\ X_k \end{bmatrix}, \qquad Y = \begin{bmatrix} Y_1 \\ \cdots \\ Y_k \end{bmatrix}.$$

We also assume that each *local* OLS estimator $\hat{\beta}_i = (X_i^\top X_i)^{-1} X_i^\top Y_i$ is well defined, which requires that the number of local training data points n_i must be at least p on each machine (so $n_i \geq p$). We first consider combining the local OLS estimators at a parameter server via one-step weighted averaging. Since they are uncorrelated and unbiased for β, we consider unbiased weighted estimators

$$\hat{\beta}_{\text{dist}}(w) = \sum_{i=1}^{k} w_i \hat{\beta}_i$$

with $\sum_{i=1}^{k} w_i = 1$.

Here, we want to mention a crucial difference between distributed linear regression and other more complicated distributed statistical learning problems. That is, the local OLS estimators are unbiased, but in more complex problems there is usually a local bias term of order $1/n_i$. If a local problem has too few samples: $n_i \leq \sqrt{n}$, the bias starts to dominate the convergence rate of the averaged estimator. This is the so-called "$\sqrt{n}$-barrier" in distributed statistical learning (Zhang, Duchi and Wainwright (2013b, 2015), Fan, Guo and Wang (2019)). Luckily, this does not occur in our setting.

We introduce a "general linear functional" framework to study learning tasks such as estimation and prediction in a unified way. In the general framework, we predict *linear functionals* of β of the form

$$L_A = A\beta + Z.$$

Here, A is a fixed $d \times p$ matrix, and Z is a zero-mean Gaussian noise vector of dimension d, with covariance matrix $\mathrm{Cov}[Z] = h\sigma^2 I_d$, for some scalar parameter $h \geq 0$. We denote the covariance matrix between ε and Z by N, so that $\mathrm{Cov}[\varepsilon, Z] = N$. If $h = 0$, we say that there is no noise. In that case, we necessarily have $N = 0$.

We predict the linear functional L_A via plug-in based on some estimator $\hat{\beta}_0$ (typically OLS or distributed OLS)

$$\hat{L}_A(\hat{\beta}_0) = A\hat{\beta}_0.$$

We measure the quality of estimation by the mean squared error

$$M(\hat{\beta}_0) = \mathbb{E}\|L_A - \hat{L}_A(\hat{\beta}_0)\|^2.$$

We compute the *relative efficiency* of OLS $\hat{\beta}$ compared to a weighted distributed estimator $\hat{\beta}_{\mathrm{dist}} = \hat{\beta}_{\mathrm{dist}}(w)$:

$$E(A, d; X_1, \ldots, X_k) := \frac{M(\hat{\beta})}{M(\hat{\beta}_{\mathrm{dist}})}.$$

The relative efficiency is a fundamental quantity, giving the loss of accuracy due to distributed estimation.

3.1. *Examples*. We now show how several learning and inference problems fall into the general framework. See Table 2 for a concise summary.

- *Parameter estimation*. In parameter estimation, we want to estimate the regression coefficient vector β using $\hat{\beta}$. This is an example of the general framework by taking $A = I_p$, and without noise (so that $h = 0$).
- *Regression function estimation*. We can use $X\hat{\beta}$ to estimate the regression function $\mathbb{E}(Y|X) = X\beta$. In this case, the transform matrix is $A = X$, the linear functional is $L_A = X\beta$, the predictor is $\hat{L}_A = X\hat{\beta}$, and there is no noise.
- *Out-of-sample prediction (Test error)*. For out-of-sample prediction, or test error, we consider a test data point (x_t, y_t), generated from the same model $y_t = x_t^\top \beta + \varepsilon_t$, where x_t, ε_t are independent of X, ε, and only x_t is observable. We want to use $x_t^\top \hat{\beta}$ to predict y_t.

  This corresponds to predicting the linear functional $L_{x_t} = x_t^\top \beta + \varepsilon_t$, so that $A = x_t^\top$, and the noise is $Z = \varepsilon_t$, which is uncorrelated with the noise ε in the original problem.
- *In-sample prediction (Training error)*. For in-sample prediction, or training error, we consider predicting the response vector Y, using the model fit $X\hat{\beta}$. Therefore, the functional

TABLE 2

A general framework for finite-sample efficiency calculations. The rows show the various statistical problems studied in our work, namely estimation, confidence interval formation, in-sample prediction, out-of-sample prediction and regression function estimation. The elements of the row show how these tasks fall in the framework of linear functional prediction described in the main body

Statistical learning problem	L_A	$\hat{L}_A$	A	h	N
Estimation	β	$\hat{\beta}$	I_p	0	0
Regression function estimation	$X\beta$	$X\hat{\beta}$	X	0	0
Confidence interval	β_j	$\hat{\beta}_j$	$E_j^\top$	0	0
Test error	$x_t^\top \beta + \varepsilon_t$	$x_t^\top \hat{\beta}$	$x_t^\top$	1	0
Training error	$X\beta + \varepsilon$	$X\hat{\beta}$	X	1	$\sigma^2 I_n$

L_A is $L_A = Y = X\beta + \varepsilon$. This agrees with regression function estimation, except for the noise $Z = \varepsilon$, which is identical to the original noise. Hence, the noise scale is $h = 1$, and $N = \mathrm{Cov}[\varepsilon, Z] = \sigma^2 I_n$.

- *Confidence intervals*. To construct confidence intervals for individual coordinates, we consider the normal model $Y \sim \mathcal{N}(X\beta, \sigma^2 I_n)$. Assuming σ^2 is known, a confidence interval with coverage $1 - \alpha$ for a given coordinate β_j is

$$\hat{\beta}_j \pm \sigma z_{\alpha/2} V_j^{1/2},$$

where $z_\alpha = \Phi^{-1}(\alpha)$ is the inverse normal CDF, and V_j is the jth diagonal entry of $(X^\top X)^{-1}$.

Therefore, we can measure the difficulty of the problem by V_j. The larger V_j is, the longer the confidence interval. This also measures the difficulty of estimating the coordinate $L_A = \beta_j$. This can be fit in our general framework by choosing $A = E_j^\top$, the $1 \times p$ vector of zeros, with only a one in the jth coordinate. This problem is noiseless. In this sense, the problem of confidence intervals is the same as the estimation accuracy for individual coordinates of β.

If σ is not known, then we we first need to estimate it in a distributed way. This is an interesting problem in itself, but beyond the scope of our current work.

3.2. *Finite sample results*. We now show how to calculate the efficiency explicitly in the general framework. We start with the simpler case where $h = 0$. We then have for the OLS estimator

$$M(\hat{\beta}) = \sigma^2 \cdot \mathrm{tr}\big[(X^\top X)^{-1} A^\top A\big].$$

For the distributed estimator with weights w_i summing to one, given by $\hat{\beta}_{\mathrm{dist}}(w) = \sum_i w_i \hat{\beta}_i$, we have

$$M(\hat{\beta}_{\mathrm{dist}}) = \sigma^2 \cdot \left(\sum_{i=1}^k w_i^2 \cdot \mathrm{tr}\big[(X_i^\top X_i)^{-1} A^\top A\big] \right).$$

Using a simple Cauchy–Schwarz inequality (see Section A for the argument for parameter estimation), we find that the optimal efficiency for the optimal weights is

$$E(A; X_1, \ldots, X_k) = \mathrm{tr}\big[(X^\top X)^{-1} A^\top A\big] \cdot \sum_{i=1}^k \frac{1}{\mathrm{tr}[(X_i^\top X_i)^{-1} A^\top A]}. \tag{1}$$

This shows that the key to understanding the efficiency are the traces $\mathrm{tr}[(X_i^\top X_i)^{-1} A^\top A]$. Proving that the efficiency is at most unity turns out to require the concavity of the matrix functional $1/\mathrm{tr}(X^{-1} A^\top A)$. This is a consequence of classical results in convex analysis; see, for instance, Davis (1957), Lewis (1996). For completeness, we give a short self-contained proof in Section B of the Supplementary Material (Dobriban and Sheng (2021)). In addition, from the formula, we observe that there is no loss of efficiency if the local datasets are i.i.d. and all the Gram matrices $X_i^\top X_i / n_i$ converge to the true population covariance matrix.

PROPOSITION 3.1 (Concavity for general efficiency, Davis (1957), Lewis (1996)). *The function $f(X) = 1/\mathrm{tr}(X^{-1} A^\top A)$ is a concave function defined on positive definite matrices. As a consequence, the general relative efficiency for distributed estimation is at most unity for any matrices X_i:*

$$E(A; X_1, \ldots, X_k) \le 1.$$

For the more general case when $h \neq 0$, we can also find the OLS MSE as

$$M(\hat{\beta}) = \sigma^2 \cdot \big[\operatorname{tr}\big((X^\top X)^{-1} A^\top A\big) - 2\operatorname{tr}\big(A(X^\top X)^{-1} X^\top N\big) + hd\big].$$

For the distributed estimator, we can find, denoting $N_i := \operatorname{Cov}[\varepsilon_i, Z]$,

$$M(\hat{\beta}_{\text{dist}}) = \sigma^2 \cdot \left(\sum_{i=1}^{k} w_i^2 \cdot \operatorname{tr}\big[(X_i^\top X_i)^{-1} A^\top A\big] - 2w_i \cdot \operatorname{tr}\big(A(X_i^\top X_i)^{-1} X_i^\top N_i\big)\right) + \sigma^2 hd.$$

Let $a_i = \operatorname{tr}[(X_i^\top X_i)^{-1} A^\top A]$, and $b_i = \operatorname{tr}(A(X_i^\top X_i)^{-1} X_i^\top N_i)$. The optimal weights can be found from a quadratic optimization problem:

$$w_i = \frac{\lambda^* + b_i}{a_i}, \quad \lambda^* := \frac{1 - \sum_{i=1}^{k} \frac{b_i}{a_i}}{\sum_{i=1}^{k} \frac{1}{a_i}}.$$

The resulting formula for the optimal weights, and for the global optimum, can be calculated explicitly. The details can be found in the Supplementary Material (Section C) (Dobriban and Sheng (2021)).

4. Calculus of deterministic equivalents.

4.1. *A calculus of deterministic equivalents in RMT.* We saw that the relative efficiency depends on the trace functionals $\operatorname{tr}[(X^\top X)^{-1} A^\top A]$, for specific matrices A. To find their limits, we will use the technique of *deterministic equivalents* from random matrix theory. This is a method to find the almost sure limits of random quantities; see, for example, Hachem, Loubaton and Najim (2007), Couillet, Debbah and Silverstein (2011) and the related work section below.

For instance, the well-known Marchenko–Pastur (MP) law for the eigenvalues of random matrices (Marchenko and Pastur (1967), Bai and Silverstein (2009)) states that the eigenvalue distribution of certain random matrices is asymptotically deterministic. More generally, one of the best ways to understand the MP law is that *resolvents are asymptotically deterministic*. Indeed, let $\widehat{\Sigma} = n^{-1} X^\top X$, where $X = Z\Sigma^{1/2}$ and Z is a random matrix with i.i.d. entries of zero mean and unit variance. Then the MP law means that for any z with positive imaginary part, we have the equivalence

$$(\widehat{\Sigma} - zI)^{-1} \asymp (x_p \Sigma - zI)^{-1},$$

for a certain scalar $x_p = x(\Sigma, n, p, z)$ (that will be specified later). At this stage, we can think of the equivalence entrywise, but we will make this precise next. The above formulation has appeared in some early works by VI Serdobolskii; see, for example, Serdobolskii (1983), and Theorem 1 on page 15 of Serdobolskii (2007) for a very clear statement.

To elaborate, the MP law is usually stated in terms of the convergence of the empirical spectral distribution of the sample covariance matrix $\widehat{\Sigma}$. This is derived directly from the convergence of the Stieltjes transform of $\widehat{\Sigma}$. The Stieltjes transform is simply the trace of the scaled resolvent $n^{-1}(\widehat{\Sigma} - zI)^{-1}$, which is a linear functional of the entries of the resolvent. Hence, its convergence can be derived from the calculus. However, it is less commonly discussed that the proof techniques used to derive the MP law also yield as a byproduct the convergence of all linear functionals, not just those involving the diagonal, which leads to our calculus.

The consequence is that simple linear functionals of the random matrix $(\widehat{\Sigma} - zI)^{-1}$ have a deterministic equivalent based on $(x_p \Sigma - zI)^{-1}$. In particular, we can approximate the needed trace functionals by simpler deterministic quantities. For this, we will take a principled approach and define some appropriate notions for a *calculus of deterministic equivalents*, which allows us to do calculations in a simple and effective way.

First, we make more precise the notion of equivalence. We say that the (deterministic or random) not necessarily symmetric matrix sequences A_n, B_n of growing dimensions are *equivalent*, and write

$$A_n \asymp B_n$$

if

$$\lim_{n\to\infty} \left|\operatorname{tr}\left[C_n(A_n - B_n)\right]\right| = 0$$

almost surely, for any sequence C_n of not necessarily symmetric matrices with bounded trace norm, that is, such that

$$\limsup \|C_n\|_{\mathrm{tr}} < \infty.$$

We call such a sequence C_n a *standard sequence*. Recall here that the trace norm (or nuclear norm) is defined by $\|M\|_{\mathrm{tr}} = \operatorname{tr}((M^\top M)^{1/2}) = \sum_i \sigma_i$, where σ_i are the singular values of M.

4.2. *General MP theorem.* To find the limits of the efficiencies, the most important deterministic equivalent will be the following result, essentially a consequence of the generalized Marchenko–Pastur theorem of Rubio and Mestre (2011) (see Section D for the argument). We study the more general setting of elliptical data. In this model, the data samples may have different scalings, having the form $x_i = g_i^{1/2}\Sigma^{1/2}z_i$, for some vector z_i with iid entries, and for datapoint-specific *scale parameters* g_i. Arranging the data as the rows of the matrix X, that takes the form

$$X = \Gamma^{1/2} Z \Sigma^{1/2},$$

where Z and Γ are as before: Z has i.i.d. standardized entries, while Σ is the covariance matrix of the features. Now Γ is the diagonal *scaling matrix* containing the scales g_i of the samples. This model has a long history in multivariate statistics (e.g., Mardia, Kent and Bibby (1979)).

THEOREM 4.1 (Deterministic equivalent in elliptical models, consequence of Rubio and Mestre (2011)). *Let the $n \times p$ data matrix X follow the elliptical model*

$$X = \Gamma^{1/2} Z \Sigma^{1/2},$$

where Γ is an $n \times n$ diagonal matrix with nonnegative entries representing the scales of the n observations, and Σ is a $p \times p$ positive definite matrix representing the covariance matrix of the p features. Assume the following:

1. *The entries of Z are i.i.d. random variables with mean zero, unit variance, and finite $8 + c$-th moment, for some $c > 0$.*
2. *The eigenvalues of Σ, and the entries of Γ, are uniformly bounded away from zero and infinity.*
3. *We have $n, p \to \infty$, with $\gamma_p = p/n$ bounded away from zero and infinity.*

Let $\widehat{\Sigma} = n^{-1}X^\top X$ be the sample covariance matrix. Then $\widehat{\Sigma}$ is equivalent to a scaled version of the population covariance

$$\widehat{\Sigma}^{-1} \asymp \Sigma^{-1} \cdot e_p.$$

Here, $e_p = e_p(n, p, \Gamma) > 0$ is the unique solution of the fixed-point equation

$$1 = \frac{1}{n}\operatorname{tr}\left[e_p\Gamma(I + \gamma_p e_p \Gamma)^{-1}\right].$$

Thus, the inverse sample covariance matrix has a deterministic equivalent in terms of a scaled version of the inverse population covariance matrix. This result does not require the convergence of the aspect ratio p/n, or of the e.s.d. of Σ, and Γ, as is sometimes the case in random matrix theory. However, if the empirical spectral distribution of the scales Γ tends to G, the above equation has the limit

$$\int \frac{se}{1+\gamma se}\, dG(s) = 1.$$

The usual MP theorem is a special case of the above result where $\Gamma = I_n$. As a result, we obtain the following corollary.

COROLLARY 4.2 (Deterministic equivalent in MP models). *Let the $n \times p$ data matrix X follow the model $X = Z\Sigma^{1/2}$, where Σ is a $p \times p$ positive definite matrix representing the covariance matrix of the p features. Assume the same conditions on Σ from Theorem 4.1. Then $\widehat{\Sigma}$ is equivalent to a scaled version of the population covariance*

$$\widehat{\Sigma}^{-1} \asymp \frac{1}{1-\gamma_p} \cdot \Sigma^{-1}.$$

The proof is immediate, by checking that $e_p = 1/(1-\gamma_p)$ in this case.

To motivate the need for this result, note that the relative efficiency (1) depends on linear functionals of $\widehat{\Sigma}^{-1}$ and $\widehat{\Sigma}_i^{-1}$. For instance, for estimation error, we will derive below that the relative efficiency is $\operatorname{tr}[(X^\top X)^{-1}] \cdot [\sum_{i=1}^k \frac{1}{\operatorname{tr}[(X_i^\top X_i)^{-1}]}]$. Now, since $\operatorname{tr}[(X^\top X)^{-1}] = \operatorname{tr}[\widehat{\Sigma}^{-1}]/n$, we can use the above result to calculate $\operatorname{tr}[\widehat{\Sigma}^{-1}] \asymp \frac{1}{1-\gamma_p} \cdot \operatorname{tr}[\Sigma^{-1}]$ and get the form of the efficiency.

4.2.1. *Related work on deterministic equivalents.* There are several works in random matrix theory on deterministic equivalents. One of the early works is Serdobolskii (1983); see Serdobolskii (2007) for a modern summary. The name "deterministic equivalents" and technique was more recently introduced and repopularized by Hachem, Loubaton and Najim (2007) for signal-plus-noise matrices. Later Couillet, Debbah and Silverstein (2011) developed deterministic equivalents for matrix models of the type $\sum_{k=1}^B R_k^{1/2} X_k T_k X_k^\top R_k^{1/2}$, motivated by wireless communications; see the book Couillet and Debbah (2011) for a summary of related work. See also Müller and Debbah (2016) for a tutorial. However, many of these results are stated only for some fixed functional of the resolvent, such as the Stieltjes transform. One of our points here is that there is a much more general picture.

Rubio and Mestre (2011) is one of the few works that explicitly states more general convergence of arbitrary trace functionals of the resolvent. Our results are essentially a consequence of theirs.

However, we think that it is valuable to define a set of rules, a "calculus" for working with deterministic equivalents, and we use those techniques in our paper. Similar ideas for operations on deterministic equivalents have appeared in Peacock, Collings and Honig (2008), for the specific case of a matrix product. Our approach is more general, and allows many more matrix operations, see below.

4.3. *Rules of calculus.* The calculus of deterministic equivalents has several properties that simplify calculations. We think these justify the name of *calculus*. Below, we will denote by A_n, B_n, C_n, etc., sequences of deterministic or random matrices. See Section E in the Supplementary Material (Dobriban and Sheng (2021)) for the proof.

THEOREM 4.3 (Rules of calculus). *The calculus of deterministic equivalents has the following properties*:

1. Equivalence. *The* $\asymp$ *relation is indeed an equivalence relation.*
2. Sum. *If* $A_n \asymp B_n$ *and* $C_n \asymp D_n$, *then* $A_n + C_n \asymp B_n + D_n$.
3. Product. *If* A_n *is a sequence of matrices with bounded operator norms, that is,* $\|A_n\|_{\mathrm{op}} < \infty$, *and* $B_n \asymp C_n$, *then* $A_n B_n \asymp A_n C_n$.
4. Trace. *If* $A_n \asymp B_n$, *then* $\mathrm{tr}\{n^{-1}A_n\} - \mathrm{tr}\{n^{-1}B_n\} \to 0$ *almost surely.*
5. Stieltjes transforms. *As a consequence, if* $(A_n - zI_n)^{-1} \asymp (B_n - zI_n)^{-1}$ *for symmetric matrices* A_n, B_n, *then* $m_{A_n}(z) - m_{B_n}(z) \to 0$ *almost surely. Here,* $m_{X_n}(z) = n^{-1}\mathrm{tr}(X_n - zI_n)^{-1}$ *is the Stieltjes transform of the empirical spectral distribution of* X_n.

In addition, the calculus of deterministic equivalents has additional properties, such as continuous mapping theorems, differentiability, etc. We have developed the differentiability in the follow-up work (Dobriban and Sheng (2019)).

We also briefly sketch several applications of the calculus of deterministic equivalents in Section F in the Supplementary Material (Dobriban and Sheng (2021)), to studying the risk of ridge regression in high dimensions, including in the distributed setting, gradient flow for least squares, interpolation in high dimensions, heteroskedastic PCA, as well as exponential family PCA. We emphasize that in each case, including for the formulas of asymptotic efficiencies in the current work, there are other proof techniques, but they tend to be more case-by-case. The calculus provides a unified set of methods, and separate results can be seen as applications of the same approach.

5. Examples. We now use the calculus of deterministic equivalents to find the limits of the trace functionals in our general framework. We study each problem in turn. For asymptotics, we consider as before elliptical models. The data on the ith machine takes the form

$$X_i = \Gamma_i^{1/2} Z_i \Sigma^{1/2},$$

where Γ_i contains the *scales* of the ith machine and Z_i is the appropriate submatrix of X.

In this model, it turns out that the efficiencies can be expressed in a simple way via the η-transform (Tulino and Verdú (2004)). The η-transform of a distribution G is

$$\eta(x) = \mathbb{E}_G \frac{1}{1 + xT},$$

for all x for which this expectation is well defined. We will see that the efficiencies can be expressed in terms of the functional inverse f of the η-transform evaluated at the specific value $1 - \gamma$:

$$f(\gamma, G) = \eta_G^{-1}(1 - \gamma). \tag{2}$$

We think of elliptical models where the limiting distribution of the scales $g_1, \ldots, g_n$ is G. For some insight on the behavior of η and f, consider first the case when G is a point mass at unity, $G = \delta_1$. In this case, all scales are equal, so this is just the usual Marchenko–Pastur model. Then we have $\eta(x) = 1/(1 + x)$, while $f(\gamma, G) = \gamma/(1 - \gamma)$; see Figure 2 for the plots. The key points to notice are that η is a decreasing function of x, with $\eta(0) = 1$, and $\lim_{x\to\infty} \eta(x) = 0$. Moreover, f is an increasing function on $[0, 1]$ with $f(0) = 0$, $\lim_{\eta\to 1} f(\eta) = +\infty$. The same qualitative properties hold in general for compactly supported distributions G bounded away from 0.

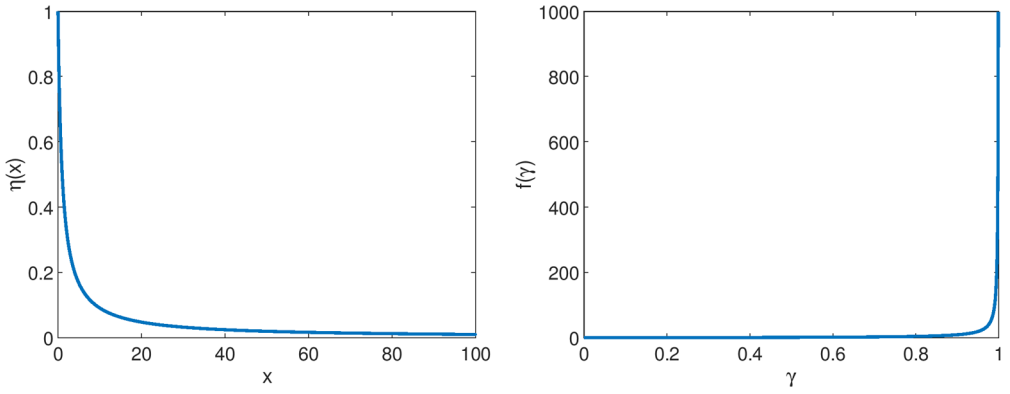

FIG. 2. *Plots of η and f for G being the point mass at unity.*

5.1. *Parameter estimation.* For estimating the parameter, we have $\mathbb{E}\|\beta - \hat{\beta}\|^2 = \sigma^2 \operatorname{tr}(X^\top X)^{-1}$. We find via (1) the estimation efficiency

$$\mathrm{RE}(X_1, \ldots, X_k) = \operatorname{tr}\big[\big(X^\top X\big)^{-1}\big] \cdot \left[\sum_{i=1}^{k} \frac{1}{\operatorname{tr}[(X_i^\top X_i)^{-1}]}\right].$$

Recall that $X^\top X = \sum_{i=1}^{k} X_i^\top X_i$. Recall that the empirical spectral distribution (e.s.d.) of a symmetric matrix M is simply the CDF of its eigenvalues (which are all real-valued). More formally, it is the discrete distribution F_p that places equal mass on all eigenvalues of M.

THEOREM 5.1 (RE for elliptical and MP models). *Under the conditions of Theorem 4.1, suppose that, as $n_i \to \infty$ with $p/n_i \to \gamma_i \in (0, 1)$, the e.s.d. of Γ converges weakly to some G, the e.s.d. of each Γ_i converges weakly to some G_i, and that the e.s.d. of Σ converges weakly to H. Suppose that H is compactly supported away from the origin, while G is also compactly supported and does not have a point mass at the origin. Then the RE has almost sure limit*

$$\mathrm{ARE} = f(\gamma, G) \cdot \sum_{i=1}^{k} \frac{1}{f(\gamma_i, G_i)}.$$

For Marchenko–Pastur models, the RE has the form $(1/\gamma - k)/(1/\gamma - 1)$.

See Section G in the Supplementary Material (Dobriban and Sheng (2021)) for the proof. For MP models, for any finite sample size n, dimension p, and number of machines k, we can approximate the ARE as

$$\mathrm{ARE} \approx \frac{n - kp}{n - p}.$$

This efficiency for MP models depends on a simple linear way on k. We find this to be a surprisingly simple formula, which can also be easily computed in practice. Moreover, the formula has several more intriguing properties:

1. The ARE *decreases linearly* with the number of machines k. This holds as long as $\mathrm{ARE} \geq 0$. At the threshold case $\mathrm{ARE} = 0$, there is a phase transition. The reason is that there is a singularity, and the OLS estimator is undefined for at least one machine.

   However, we should be cautious about interpreting the linear decrease. For the root mean squared error (RMSE), the efficiency is the square root of the ARE above, and thus does not have a linear decrease.

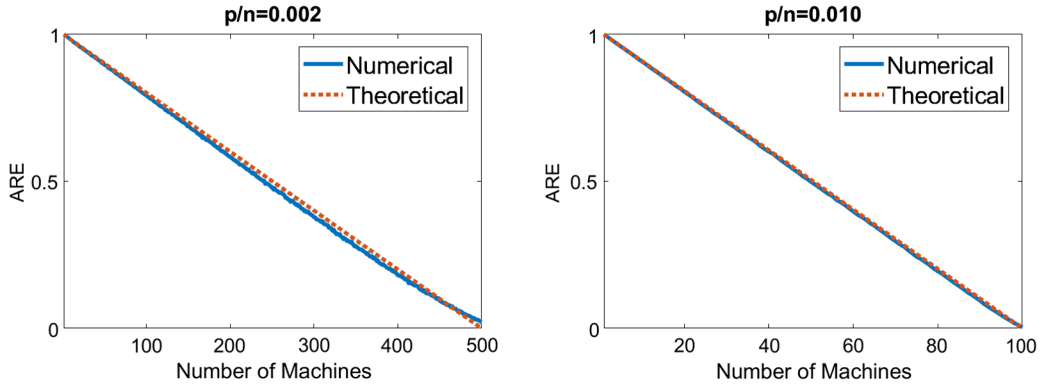

FIG. 3. *Comparison of empirical and theoretical ARE for standard sample covariance matrices. Left*: $n = 10{,}000$, $p = 20$. *Right*: $n = 10{,}000$, $p = 100$.

2. The ARE has two important *universality* properties.

   (a) First, it *does not depend* on how the samples are distributed across the different machines, that is, it is independent of the specific sample sizes n_i.

   (b) Second, it *does not depend* on the covariance matrix Σ of the samples. This is in contrast to the estimation error of OLS, which does in fact depend on the covariance structure. Therefore, we think that the cancellation of Σ in the ARE is noteworthy.

The ARE is also very accurate in simulations. See Figure 3 for an example. Here, we report the results of a simulation where we generate an $n \times p$ random matrix X such that the rows are distributed independently as $x_i \sim \mathcal{N}(0, \Sigma)$. We take Σ to be diagonal with entries chosen uniformly at random between 1 and 2. We choose $n > p$, and for each value of k such that $k < n/p$, we split the data into k groups of a random size n_i. To ensure that each group has a size $n_i \geq p$, we first let $n_i^0 = p$, and then distribute the remaining samples uniformly at random. We then show the theoretical results compared to the theoretical ARE. We observe that the two agree closely.

5.2. *Regression function estimation.* For estimating the regression function, we have $\mathbb{E}\|X(\beta - \hat{\beta})\|^2 = \sigma^2 p$. We then find via equation (1) the prediction efficiency

$$\mathrm{FE}(X_1, \ldots, X_k) = \sum_{i=1}^{k} \frac{p}{\mathrm{tr}((X_i^\top X_i)^{-1} X^\top X))}.$$

For asymptotics, we consider as before elliptical models.

THEOREM 5.2 (FE for elliptical and MP models). *Under the conditions of Theorems* 4.1 *and* 5.1, *the FE has the almost sure limit*

$$\mathrm{FE}(X_1, \ldots, X_k) \to_{\text{a.s.}} \sum_{i=1}^{k} \frac{1}{1 + (\frac{1}{\gamma}\mathbb{E}_G T - \frac{1}{\gamma_i}\mathbb{E}_{G_i} T) f(\gamma_i, G_i)}.$$

Under Marchenko–Pastur models, the conditions of Corollary 4.2, *the FE has the almost sure limit* $\frac{\gamma}{1-\gamma} \sum_{i=1}^{k} \frac{1-\gamma_i}{\gamma_i}$.

See Section G.6 for the proof. This efficiency is more complex than that for estimation error; specifically it generally depends on the individual γ_i and not just γ.

5.3. *In-sample prediction* (*Training error*). For in-sample prediction, we start with the well-known formula

$$\mathbb{E}\|X(\beta-\hat{\beta})+\varepsilon\|^2=\sigma^2[n-\operatorname{tr}((X^\top X)^{-1}X^\top X]=\sigma^2(n-p).$$

As we saw, to fit in-sample prediction in the general framework, we need to take the transform matrix $A=X$, the noise $Z=\varepsilon$, and the covariance matrices $N_i=\operatorname{Cov}[\varepsilon_i,Z]=\operatorname{Cov}[\varepsilon_i,\varepsilon]$. Then, in the formula for optimal weights we need to take $a_i=\operatorname{tr}[(X_i^\top X_i)^{-1}X^\top X]$ and $b_i=\operatorname{tr}(X(X_i^\top X_i)^{-1}X_i^\top N_i)=\operatorname{tr}[(X_i^\top X_i)^{-1}X_i^\top N_i X]=\operatorname{tr}[(X_i^\top X_i)^{-1}X_i^\top X_i]=p$. Therefore, the optimal error for distributed regression is achieved by the weights

$$w_i=\frac{\lambda-b_i}{a_i}=\frac{\lambda-p}{a_i},\quad \lambda=\frac{1-\sum_{i=1}^k\frac{b_i}{a_i}}{\sum_{i=1}^k\frac{1}{a_i}}=\frac{1}{\sum_{i=1}^k\frac{1}{a_i}}-p.$$

Plugging these into $M(\hat{\beta}_{\mathrm{dist}})$ given in the general framework, we find

$$M(\hat{\beta}_{\mathrm{dist}})=\sigma^2\biggl(n-2p+\frac{1}{\sum_{i=1}^k\frac{1}{a_i}}\biggr),\quad a_i=\operatorname{tr}\bigl((X_i^\top X_i)^{-1}X^\top X\bigr).$$

Thus, the optimal in-sample prediction efficiency is

$$\mathrm{IE}(X_1,\dots,X_k)=\frac{n-p}{n-2p+\frac{1}{\sum_{i=1}^k\frac{1}{\operatorname{tr}((X_i^\top X_i)^{-1}X^\top X)}}}.$$

For asymptotics in elliptical models, we find the following.

THEOREM 5.3 (IE for elliptical and MP models). *Under the conditions of Theorems* 4.1 *and* 5.1, *the IE has the almost sure limit*

$$\mathrm{IE}(X_1,\dots,X_k)\to_{\mathrm{a.s.}}\frac{1-\gamma}{1-2\gamma+\frac{1}{\sum_{i=1}^k\psi(\gamma_i,G_i)}},$$

where ψ *is the following functional of the distributions* G_i *and* G, *depending on the inverse of the* η*-transform* f *defined in equation* (2):

$$\psi(\gamma_i,G_i)=\frac{1}{\gamma+(\mathbb{E}_G T-\frac{\gamma}{\gamma_i}\mathbb{E}_{G_i}T)f(\gamma_i,G_i)}.$$

Under the conditions of Corollary 4.2, *the IE has the almost sure limit*

$$\mathrm{IE}(X_1,\dots,X_k)\to_{\mathrm{a.s.}}\frac{1-\gamma}{1-2\gamma+\frac{\gamma(1-\gamma)}{1-k\gamma}}=\frac{1}{1+\frac{(k-1)\gamma^2}{(1-k\gamma)(1-\gamma)}}.$$

See Section G.7 for the proof. This efficiency does not depend on a simple linear way on k, but rather via a ratio of two linear functions of k. However, it can be checked that many of the properties (e.g., monotonicity) for ARE still hold here.

5.4. *Out-of-sample prediction* (*Test error*). In out-of-sample prediction, we consider a test datapoint (x_t,y_t), generated from the same model $y_t=x_t^\top\beta+\varepsilon_t$, where x_t, ε_t are independent of X, ε, and only x_t is observable. We want to use $x_t^\top\hat{\beta}$ to predict y_t. We compare the prediction error of two estimators:

$$\mathrm{OE}(x_t;X_1,\dots,X_k):=\frac{\mathbb{E}[(y_t-x_t^\top\hat{\beta})^2]}{\mathbb{E}[(y_t-x_t^\top\hat{\beta}_{\mathrm{dist}})^2]}.$$

In our general framework, we saw that this corresponds to predicting the linear functional $x_t^\top \beta + \varepsilon_t$. Based on equation (1), the optimal out-of-sample prediction efficiency is

$$\mathrm{OE}(x_t; X_1, \ldots, X_k) = \frac{1 + x_t^\top (X^\top X)^{-1} x_t}{1 + \frac{1}{\sum_{i=1}^k \frac{1}{x_t^\top (X_i^\top X_i)^{-1} x_t}}}.$$

For asymptotics in elliptical models, we find the following result. Since the samples have the form $x_i = g_i^{1/2} \Sigma^{1/2} z_i$, the test sample depends on a scale parameter g_t.

THEOREM 5.4 (OE for elliptical and MP models). *Under the conditions of Theorems 4.1 and 5.1, the OE has the almost sure limit, conditional on* g_t,

$$\mathrm{OE}(x_t; X_1, \ldots, X_k) \to_{\text{a.s.}} \frac{1 + g_t \cdot f(\gamma, G)}{1 + \frac{g_t}{\sum_{i=1}^k \frac{1}{f(\gamma_i, G_i)}}}.$$

For Marchenko–Pastur models under the conditions of Corollary 4.2, the OE has the almost sure limit

$$\frac{\frac{1}{1-\gamma}}{1 + \frac{\gamma}{1-k\gamma}} = \frac{1}{1 + \frac{(k-1)\gamma^2}{1-k\gamma}}.$$

See Section G.8 for the proof. If the scale parameter g_t is random, then the OE typically does not have an almost sure limit, and converges in distribution to a random variable instead. We mention that Theorem 5.4 holds under even weaker conditions, if we are only given the $4 + c$-th moment of z_1 instead of $8 + c$-th one. The argument is slightly different, and is presented in the location referenced above.

One can check that that OE $\geq$ RE. Thus, out-of-sample prediction incurs a smaller efficiency loss than estimation. The intuition is that the out-of-sample prediction always involves a fixed loss due to the *irreducible noise* in the test sample, which "amortizes" the problem. Moreover,

$$\mathrm{OE} \geq \mathrm{IE} \geq \mathrm{RE}.$$

The intuition here is that IE incurs a smaller fixed loss than OE, because the noise in the training set is effectively reduced, as it is already partly fit by our estimation process. So the graph of IE will be in between the other two criteria. See Figure 4. We also see that the IE is typically very close to OE.

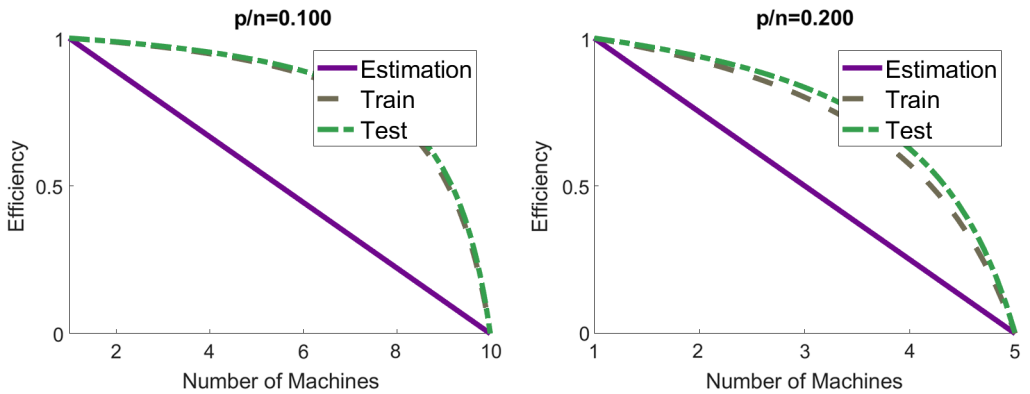

FIG. 4. *Relative efficiency for the Marchenko–Pastur model.*

In addition, the increase of the *reducible* part of the error is the same as for estimation error. The prediction error has two components: the irreducible noise, and the reducible error. The reducible error has the same behavior as for estimation, and thus on Figure 4 it would have the same plot as the curve for estimation.

5.5. *Confidence intervals*. To form confidence intervals, we consider the normal model $Y \sim \mathcal{N}(X\beta, \sigma^2 I_n)$. Recall that in this model the OLS estimator has distribution $\hat{\beta} \sim \mathcal{N}(\beta, \sigma^2(X^\top X)^{-1})$. Assuming σ^2 is known, an exact level $1-\alpha$ confidence interval for a given coordinate β_j can be formed as

$$\hat{\beta}_j \pm \sigma z_{\alpha/2} V_j^{1/2},$$

where $z_\alpha = \Phi^{-1}(\alpha)$ is the inverse normal CDF, and V_j is the jth diagonal entry of $(X^\top X)^{-1}$. We follow the same program as before, comparing the length of the confidence intervals formed based on our two estimators. However, for technical reasons it is more convenient to work with squared length.

Thus we consider the criterion

$$\mathrm{CE}(j; X_1, \ldots, X_k) := \frac{V_j}{V_{j,\mathrm{dist}}}.$$

Here, $V_{j,\mathrm{dist}}$ is the variance of the jth entry of an optimally weighted distributed estimator. As we saw in our framework, this is equivalent to estimating the jth coordinate of β. Hence the optimal confidence interval efficiency is

$$\mathrm{CE}(j; X_1, \ldots, X_k) = \left[(X^\top X)^{-1}\right]_{jj} \cdot \sum_{i=1}^{k} \frac{1}{\left[(X_i^\top X_i)^{-1}\right]_{jj}}. \tag{3}$$

For asymptotics, we find the following.

THEOREM 5.5 (CE for elliptical and MP models). *Under the conditions of Theorems 4.1 and 5.1, the CE has the same limit as the ARE from Theorem 5.1. Therefore, for Marchenko–Pastur models, the CE also has the form before,* $\mathrm{CE}(j) = (1/\gamma - k)/(1/\gamma - 1)$.

See Section G.9 for the proof.

5.6. *Understanding and comparing the efficiencies*. We give two perspectives for understanding and comparing the efficiencies. The key qualitative insight is that estimation and CIs are much more affected than prediction.

Criticality of k. We ask: What is the largest number of machines such that the asymptotic efficiency is at least 1/2? Let us call this the *critical* number of machines. It is easy to check that for estimation and CIs, $k_R = (\gamma+1)/(2\gamma)$. For training error, $k_{\mathrm{Tr}} = (\gamma^2 - \gamma + 1)/\gamma$, while for test error, $k_{\mathrm{Te}} = (\gamma^2+1)/(\gamma^2+\gamma)$.

We also have the following asymptotics as $\gamma \to 0$:

$$k_R \asymp 1/(2\gamma),$$

while

$$k_{\mathrm{Tr}} \asymp k_{\mathrm{Te}} \asymp 1/\gamma.$$

So the number of machines that can be used is nearly maximal (i.e., n/p) for training and test error, while it is about *half that* for estimation error and CIs. This shows quantitatively that estimation and CIs are much more affected by distributed averaging than prediction.

Edge efficiency. The maximum number of machines that we can use is approximately $k^* = 1/\gamma - 1$, for small γ. Let us define the *edge efficiency* e^* as the relative efficiency achieved at this edge case. For estimation and CIs, we have $e_R^* = \gamma/(1-\gamma)$. For training error, $e_{\mathrm{Tr}}^* = (1-\gamma)/(2-3\gamma)$, and for test error, $e_{\mathrm{Te}}^* = 1/[2(1-\gamma)]$.

We also have the following asymptotic values as $\gamma \to 0$:

$$e_R^* \asymp \gamma,$$

while

$$e_{\mathrm{Tr}}^* \asymp \frac{1}{2} + \frac{\gamma}{4} \quad \text{and} \quad e_{\mathrm{Te}}^* \asymp \frac{1}{2} + \frac{\gamma}{2}.$$

This shows that for $n \gg p$ the edge efficiency is vanishing for estimation and CIs, while it is approximately 1/2 for training and test error. Thus, even for the maximal number of machines, prediction error is not greatly increased.

6. Insights for parameter estimation. There are additional insights for the special case of parameter estimation. First, it is of interest to understand the performance of one-step weighted averaging with suboptimal weights w_i. How much do we lose compared to the optimal performance if we do not use the right weights? In practice, it may seem reasonable to take a simple average of all estimators. We have performed that analysis in the Supplementary Material (Section H.1) (Dobriban and Sheng (2021)), and we found that the loss can be viewed in terms of an inequality between the arithmetic and harmonic means.

There are several more remarkable properties. We have studied the monotonicity properties and interpretation of the relative efficiency; see the Supplementary Material (Section H.2) (Dobriban and Sheng (2021)). We have also given a multiresponse regression characterization that heuristically gives an upper bound on the "*degrees of freedom*" for distributed regression (Section H.3).

For elliptical data, the graph of ARE is a curve below the straight line from before. The interpretation is that for elliptical distributions, there is a larger efficiency loss in one-step weighted averaging. Intuitively, the problem becomes "more nonorthogonal" due to the additional variability from sample to sample.

It is natural to ask which elliptical distributions are difficult for distributed estimation. For what scale distributions G does the distributed setting have a strong effect on the learning accuracy? Intuitively, if some of the scales are much larger than others, then they "dominate" the problem, and may effectively reduce the sample size. We show that this intuition is correct, and we find a sequence of scale distributions G_τ such that distributed estimation is "arbitrarily bad," so that the ARE decreases very rapidly, and approaches zero even for two machines (see Section G.1 in the Supplementary Material (Dobriban and Sheng (2021))).

7. Multishot methods. While our focus has been on methods with one round of communication, in practice it is more common to use iterative methods with several rounds of communication. These usually improve statistical accuracy. A great deal of research has been done on multi-shot distributed algorithms. Due to limited space, here we will only list and analyze some of them. Our least squares objective can be written as a sum of least squares objectives for each machine as

$$f(\beta) = \frac{1}{k}\sum_{i=1}^{k} f_i(\beta) = \frac{1}{k}\sum_{i=1}^{k} \|X_i\beta - Y_i\|_2^2.$$

Here, each machine has access only to local data (X_i, Y_i). With this formulation, there are a large number of standard optimization methods to minimize this objective: distributed gradient descent, alternating directions method of multipliers, and several others we discuss

below. We will focus on parameter server architectures, where each machine communicates independently with a central server.

Distributed gradient descent. A simple multiround approach to distributed learning is synchronous distributed gradient descent (DGD), as discussed, for example, in Chu et al. (2007). This maintains iterates $\hat{\beta}^t$, started with some standard value, such as $\hat{\beta}^0 = 0$. At each iteration t, each local machine calculates the gradient $\nabla f_i(\hat{\beta}^t)$ at the current iterate $\hat{\beta}^t$, and then sends the local gradient to the server to obtain the overall gradient

$$\nabla f(\hat{\beta}^t) = \frac{1}{k}\sum_{i=1}^{k} \nabla f_i(\hat{\beta}^t).$$

Then the center server sends the updated parameter $\hat{\beta}^{t+1} = \hat{\beta}^t - \alpha \nabla f(\hat{\beta}^t)$ back to the local machines, where α is the learning rate (LR). This synchronous implementation is *identical* to centralized gradient descent. Thus for smooth and strongly convex objectives and suitably small α, $\mathcal{O}(L/\lambda \log(1/\varepsilon))$ communication rounds are sufficient to attain an ε-suboptimal solution in terms of objective value, where L, λ are the smoothness and strong convexity parameters (e.g., Boyd and Vandenberghe (2004)).

1. Many works study the optimization properties of GD/synchronous DGD, in terms of convergence rate to the optimal objective or parameter value. From a statistical point of view, the GD iterates start with large bias and small variance, and gradually reduce bias, while slightly increasing the variance. This has motivated work on the risk properties of GD, emphasizing early stopping (e.g., Yao, Rosasco and Caponnetto (2007), Ali, Kolter and Tibshirani (2019)). Recently, Ali, Kolter and Tibshirani (2019) gave a more refined analysis of the estimation risk of GD for OLS, showing that its risk at an optimal stopping time is at most 1.22 times the risk of optimally tuned ridge regression.

2. Compared to GD, one-shot weighted averaging has several advantages: it is simpler to implement, as it requires no iterations. It requires fewer tuning parameters, and those can be set optimally in an easy way, unlike the LR α. The weights are proportional to $1/\operatorname{tr}[(X_i^T X_i)^{-1}]$, which can be computed locally. We point out that GD is sensitive to the learning rate: this has to be bounded (by $2/\lambda_{\max}(X^\top X)$ for OLS) to converge, and the convergence can be faster for large LR, hence in practice sophisticated LR schedules are used. This can make DGD complicated to use. In addition, in practice DGD is susceptible to stragglers, that is, machines that take too long to compute their answers. To mitigate this problems, asynchronous DGD algorithms (e.g., Tsitsiklis, Bertsekas and Athans (1986), Nedić and Ozdaglar (2009)), and other sophisticated coding ideas (Tandon et al. (2017)) have been proposed. However, those lead to additional complexity and hyperparameters to tune (e.g., for async algorithms: how much to wait, how to aggregate nonstraggler gradients).

3. One may also use other gradient based methods, such as accelerated or quasi-Newton methods, for example, L-BFGS (Agarwal et al. (2014)).

Alternating Direction Method of Multipliers (ADMM). Another approach is the alternating direction method of multipliers (see Boyd et al. (2011) for an exposition) and its variants. In ADMM, we alternate between solving local problems, global averaging, and computing local dual variables. For us, at time step t of ADMM, each local machine calculates a local estimator

$$\hat{\beta}_i^{t+1} = \left(X_i^\top X_i + \rho I\right)^{-1}\left[X_i^\top Y_i + \rho\left(\hat{\beta}^t - u_i^t\right)\right]$$

(where ρ is a hyperparameter) and sends it to the parameter server to get an average

$$\hat{\beta}^{t+1} = \frac{1}{k}\sum_{i=1}^{k} \hat{\beta}_i^{t+1}.$$

Finally, the server sends $\hat{\beta}^{t+1}$ back to the local machines to update the dual variables

$$u_i^{t+1} = u_i^t + \hat{\beta}_i^{t+1} - \hat{\beta}^{t+1}.$$

These three steps can be written as a linear recursion $z^{t+1} = Az^t + b$ for a state variable z^t including $\hat{\beta}^t$, $\hat{\beta}_i^t$ and u_i^t. If all singular values of A are less than one, then the iteration converges to a fixed point solving $z = Az + b$, so that $z = (I - A)^{-1}b$. However, it seems hard to prove convergence in our asymptotic setting.

Distributed Approximate Newton-type Method (DANE). Shamir, Srebro and Zhang (2014) proposed an approximate Newton-like method (DANE), which uses that the subproblems are similar. For our problem, DANE aggregates the local gradients on the parameter server at each step t, and sends this quantity, that is, $X^\top(X\hat{\beta}^t - Y)/(2k)$ to all machines. Then each machine computes a local estimator by a gradient step in the direction of a regularized local Hessian $X_i^\top X_i + \rho I$,

$$\hat{\beta}_i^{t+1} = \hat{\beta}^t + \frac{\eta}{k} \cdot (X_i^\top X_i + \rho I)^{-1} X^\top (Y - X\hat{\beta}^t),$$

where ρ is the regularizer and η is the learning rate. The machines send it to the server to get the aggregated estimator

$$\hat{\beta}^{t+1} = \frac{1}{k} \sum_{i=1}^{k} \hat{\beta}_i^{t+1}.$$

For a noiseless model where $Y = X\beta$, we can summarize the update rule as

$$\hat{\beta}^{t+1} - \beta = \left(I - \frac{\eta}{k^2} \cdot \sum_{i=1}^{k} (X_i^\top X_i + \rho I)^{-1} X^\top X \right) (\hat{\beta}^t - \beta),$$

so we have the error bound

$$\|\hat{\beta}^t - \beta\|_2 \le \left\| I - \frac{\eta}{k^2} \cdot \sum_{i=1}^{k} (X_i^\top X_i + \rho I)^{-1} X^\top X \right\|_2^t \cdot \|\hat{\beta}^0 - \beta\|_2.$$

In Shamir, Srebro and Zhang (2014), the authors showed that given a suitable learning rate η and regularizer ρ, when $X_i^\top X_i$ is close to $X^\top X/k$, $\hat{\beta}^t \to \beta$ as $t \to \infty$.

For a noisy linear model $Y = X\beta + \varepsilon$, the limit of $\hat{\beta}^t$ is exactly the OLS estimator of the whole data set, and we have the following recursion formula:

$$\hat{\beta}^{t+1} - (X^\top X)^{-1} X^\top Y = \left(I - \frac{\eta}{k^2} \cdot \sum_{i=1}^{k} (X_i^\top X_i + \rho I)^{-1} X^\top X \right) (\hat{\beta}^t - (X^\top X)^{-1} X^\top Y),$$

and the convergence guarantee is the same as for the noiseless case.

Iterative averaging method. Here, we describe an iterative averaging method for distributed linear regression. This method turns out to be connected to DANE, and it has the advantage that it can be analyzed more conveniently. We define a sequence of *local estimates* $\hat{\beta}_i^t$ and *global estimates* $\hat{\beta}^t$ with initialization $\hat{\beta}^0 = 0$. At the tth step, we update the local estimate by the following weighted average of the local ridge regression estimator and the current global estimate $\hat{\beta}^t$:

$$\hat{\beta}_i^{t+1} = (X_i^\top X_i + n_i \rho_i I)^{-1} (X_i^\top Y_i + n_i \rho_i \hat{\beta}^t).$$

Then we average the local estimates

$$\hat{\beta}^{t+1} = \frac{1}{k} \sum_{i} \hat{\beta}_i^{t+1}.$$

To understand this, let us first consider a noiseless model where $Y_i = X_i\beta$. In that case, we can also write this update as a weighted average,

$$\hat{\beta}_i^{t+1} = (I - W_i)\beta + W_i\hat{\beta}^t,$$

where

$$W_i = n_i\rho_i \cdot \left(X_i^\top X_i + n_i\rho_i I\right)^{-1}$$

is the weight matrix of the global estimate. Propagating the iterative update to the global machine, we find a linear update rule:

$$\hat{\beta}^{t+1} = \frac{1}{k}\sum_i W_i\hat{\beta}^t + \left(I - \frac{1}{k}\sum_i W_i\right)\beta = W\hat{\beta}^t + (I - W)\beta,$$

where $W = \frac{1}{k}\sum_i W_i$. Hence, the error is updated as

$$\hat{\beta}^{t+1} - \beta = W \cdot [\hat{\beta}^t - \beta] = \left(I - \frac{1}{k}\sum_{i=1}^k \left(X_i^\top X_i + n_i\rho_i I\right)^{-1} X_i^\top X_i\right)(\hat{\beta}^t - \beta).$$

This recursion relation is very similar to the one for DANE; we just need to replace $X^\top X/k$ by $X_i^\top X_i$ (and in practice usually $\eta = 1$ is used). The only difference is that DANE has a step where we need to collect the local gradients to get the global gradient, and then broadcast it to all local machines. Our iterative averaging method has lower communication cost.

In terms of convergence, $\hat{\beta}^{t+1}$ will converge geometrically to β for all β, if and only if the largest eigenvalue of W is strictly less than 1. It is not hard to see that this holds if at least one $X_i^\top X_i$ has positive eigenvalues by using the fact $\lambda_{\max}(A + B) \le \lambda_{\max}(A) + \lambda_{\max}(B)$. When the samples are uniformly distributed, we should have $X^\top X/k \approx X_i^\top X_i$, which means the convergence rates of DANE and iterative averaging should be very close. Hence, in terms of the total cost (communication and computation), our iterative averaging should compare favorably to DANE.

To summarize the noiseless case, we can formulate the following result.

THEOREM 7.1 (Convergence of iterative averaging, noiseless case). *Consider the iterative averaging method described above. In the noiseless case when $Y_i = X_i\beta$, we have the following*: *If at least one $X_i^\top X_i$ has positive eigenvalues, then the iterates converge to the true coefficients geometrically, $\hat{\beta}^t \to \beta$, and*

$$\left\|\hat{\beta}^t - \beta\right\|_2 \le \lambda_{\max}\left(\frac{1}{k}\sum_{i=1}^k n_i\rho_i \cdot \left(X_i^\top X_i + n_i\rho_i I\right)^{-1}\right)^t \cdot \|\beta\|_2.$$

Consider now the noisy case when $Y_i = X_i\beta + \varepsilon_i$ with the same assumptions as in the rest of the paper. We have

$$\begin{aligned}\hat{\beta}_i^{t+1} &= W_i\hat{\beta}^t + (I - W_i)\beta + \left(X_i^\top X_i + n_i\rho_i I\right)^{-1} X_i^\top \varepsilon_i \\ &=_d W_i\hat{\beta}^t + (I - W_i)\beta + \left(X_i^\top X_i + n_i\rho_i I\right)^{-1} X_i^\top X_i \cdot Z_i \\ &= W_i\hat{\beta}^t + (I - W_i)(\beta + Z_i),\end{aligned}$$

where $Z_i \sim \mathcal{N}(0, \sigma^2[X_i^\top X_i]^{-1})$. As before, defining Z appropriately

$$\begin{aligned}\hat{\beta}^{t+1} &= W\hat{\beta}^t + (I - W)\beta + \frac{1}{k}\sum_{i=1}^k (I - W_i)Z_i \\ &= W\hat{\beta}^t + (I - W)\beta + Z,\end{aligned}$$

so $\hat{\beta}^{t+1} - \beta = W \cdot [\hat{\beta}^t - \beta] + Z$.

With noise, $\hat{\beta}^t$ does not converge to OLS, but to the following quantity:

$$\hat{\beta}_* = \left(\sum_{i=1}^k (X_i^\top X_i + n_i\rho_i I)^{-1} X_i^\top X_i\right)^{-1} \cdot \sum_{i=1}^k (X_i^\top X_i + n_i\rho_i I)^{-1} X_i^\top Y_i.$$

We can check that $\hat{\beta}_*$ is an unbiased estimator for β and $\hat{\beta}^{t+1} - \hat{\beta}_* = W \cdot [\hat{\beta}^t - \hat{\beta}_*]$.

Under the conditions of Theorem 7.1, we have $\hat{\beta}^t \to \hat{\beta}_*$, and the MSE for $\hat{\beta}_*$ is

$$\begin{aligned}\mathbb{E}\|\hat{\beta}_* - \beta\|^2 &= \mathbb{E}\|(I-W)^{-1}Z\|^2 = \mathbb{E}\left\|\frac{1}{k}\sum_{i=1}^k (I-W)^{-1}(I-W_i)Z_i\right\|^2 \\ &= \frac{\sigma^2}{k^2}\sum_{i=1}^k \operatorname{tr}\left[(I-W)^{-2}(X_i^\top X_i + n_i\rho_i I)^{-2} X_i^\top X_i\right] \\ &= \sigma^2 \sum_{i=1}^k \operatorname{tr}\left[\left(\sum_{i=1}^k (X_i^\top X_i + n_i\rho_i I)^{-1} X_i^\top X_i\right)^{-2} (X_i^\top X_i + n_i\rho_i I)^{-2} X_i^\top X_i\right].\end{aligned}$$

How large is this MSE, and how does it depend on ρ_i? We have the following results.

THEOREM 7.2 (Properties of Iterative averaging, noisy case). *Consider the iterative averaging method described above. In the noisy case when $Y_i = X_i\beta + \varepsilon$, we have the following*:

1. *If at least one $X_i^\top X_i$ has strictly positive eigenvalues, then the iterates converge to the following limiting unbiased estimator*

$$\hat{\beta}_* = \left(\sum_{i=1}^k (X_i^\top X_i + n_i\rho_i I)^{-1} X_i^\top X_i\right)^{-1} \cdot \sum_{i=1}^k (X_i^\top X_i + n_i\rho_i I)^{-1} X_i^\top Y_i,$$

and the convergence is geometric

$$\|\hat{\beta}^t - \hat{\beta}_*\|_2 \le \lambda_{\max}\left(\frac{1}{k}\sum_{i=1}^k n_i\rho_i \cdot (X_i^\top X_i + n_i\rho_i I)^{-1}\right)^t \cdot \|\hat{\beta}_*\|_2.$$

2. *The mean squared error of $\hat{\beta}_*$ has the following form*:

$$\mathbb{E}\|\hat{\beta}_* - \beta\|^2 = \sigma^2 \sum_{i=1}^k \operatorname{tr}\left[\left(\sum_{i=1}^k (X_i^\top X_i + n_i\rho_i I)^{-1} X_i^\top X_i\right)^{-2} (X_i^\top X_i + n_i\rho_i I)^{-2} X_i^\top X_i\right].$$

3. *Suppose the samples are evenly distributed, that is, $n_1 = n_2 = \cdots = n_k = n/k$ and the regularizers are all the same $\rho_1 = \rho_2 = \cdots = \rho_k = \rho$. The MSE is a differentiable function $\psi(\rho)$ of the regularizer $\rho \in [0, +\infty)$, with derivative*

$$\begin{aligned}\psi'(\rho) = \frac{2k}{n}\operatorname{tr}\Bigg[&\Delta^{-1}\sum_{i=1}^k (\widehat{\Sigma}_i + \rho I)^{-2}\widehat{\Sigma}_i \cdot \Delta^{-2}\sum_{i=1}^k (\widehat{\Sigma}_i + \rho I)^{-2}\widehat{\Sigma}_i \\ &- \Delta^{-2}\sum_{i=1}^k (\widehat{\Sigma}_i + \rho I)^{-3}\widehat{\Sigma}_i\Bigg],\end{aligned}$$

where $\widehat{\Sigma}_i = X_i^\top X_i/n_i$ and $\Delta := \sum_{i=1}^k (\widehat{\Sigma}_i + \rho I)^{-1}\widehat{\Sigma}_i$.

4. *$\psi(\rho)$ is a nonincreasing function on $[0, +\infty)$ and $\psi'(0) < 0$. So for any $\rho > 0$, $\psi(\rho) < \psi(0)$, that is, the MSE of the iterative averaging estimator with positive regularizer is smaller than the MSE of the one-step averaging estimator.*

5. *When $\rho = 0$, $\hat{\beta}_*$ reduces to the one-step averaging estimator $1/k \cdot \sum_{i=1}^{k} (X_i^\top X_i)^{-1} X_i^\top Y_i$ with MSE*

$$\psi(0) = \sigma^2/k^2 \cdot \sum_{i=1}^{k} \operatorname{tr}\bigl(X_i^\top X_i\bigr)^{-1}.$$

When $\rho \to +\infty$, $\hat{\beta}_*$ converges to the OLS estimator $(X^\top X)^{-1} X^\top Y$ with MSE

$$\lim_{\rho \to +\infty} \psi(\rho) = \sigma^2 \operatorname{tr}\bigl(X^\top X\bigr)^{-1}.$$

See Section I of the Supplementary Material (Dobriban and Sheng (2021)) for the proof. The argument for monotonicity relies on Schur complements, and is quite nontrivial. From Theorem 7.2, it appears we should choose the regularizer ρ as large as possible, since the limiting estimator $\hat{\beta}_*$ will converge to the OLS estimator as $\rho \to \infty$. This is true for statistical accuracy. However, there is a computational tradeoff, since the convergence rate of $\hat{\beta}^t$ to $\hat{\beta}_*$ is slower for large ρ.

Moreover, one may argue that $\hat{\beta}_*$ reduces to the naive averaging estimator but not the optimally weighted averaging estimator when $\rho = 0$. However, we have shown in the Supplementary Material (Section H.1) (Dobriban and Sheng (2021)) that for evenly distributed samples, the MSE of the naive averaging estimator and the optimally weighted averaging estimator is asymptotically the same. Thus, there exists a regularizer such that the iterative averaging estimator has smaller MSE than the one-step weighted averaging estimator.

Other approaches. There are many other approaches to distributed learning. *Dual averaging for decentralized optimization* over a network (Duchi, Agarwal and Wainwright (2012)) builds on Nesterov's dual averaging method (Nesterov (2009)). It chooses the iterates to minimize an averaged first-order approximation to the function, regularized with a proximal function. The *communication-efficient surrogate likelihood* approximates the objective by an expression of the form $\tilde{f}(\beta) = f_1(\beta) - \beta^\top(\nabla f_1(\bar{\beta}) - \nabla f(\bar{\beta}))$, where $\bar{\beta}$ is a preliminary estimator (Jordan, Lee and Yang (2019), Wang et al. (2017)). Chen, Liu and Zhang (2019) propose a related method for quantile regression. Both are related to DANE (Shamir, Srebro and Zhang (2014)).

Chen, Liu and Zhang (2018) study divide and conquer SGD (DC-SGD), running SGD on each machine and averaging the results. They also propose a distributed first-order Newton-type estimator starting with a preliminary estimator $\bar{\beta}$, of the form $\bar{\beta} - \Sigma^{-1}(k^{-1} \sum_i \nabla f_i(\bar{\beta}))$, where Σ is the population Hessian. They show how to numerically estimate this efficiently, and also develop a more accurate multiround version.

7.1. *Numerical comparisons*. We report simulations to compare the convergence rate and statistical accuracy of the one-shot weighted method with some popular multishot methods described above (Figure 5). Here, we work with a linear model $Y = X\beta + \varepsilon$, where X, β and ε all follow standard normal distributions. We take $n = 10{,}000$, $p = 100$ and $k = 20$. We plot the relative efficiencies of different methods against the number of iterations.

We can see that the one-shot weighted method is good in some cases. The multishot methods usually need several iterations to achieve better statistical accuracy. When the communication cost is large, one-shot methods are attractive. Also, we can clearly see the computation versus accuracy tradeoff for the iterative averaging method from the plots. When the regularizer is small, the convergence is fast, but in the end the accuracy is not as good as the other multishot methods. On the other hand, if the regularizer is large, we have a better accuracy with slower convergence. Moreover, the widely-used multishot methods can require a lot of work for parameter tuning, and sometimes it is very difficult to find the optimal parameters.

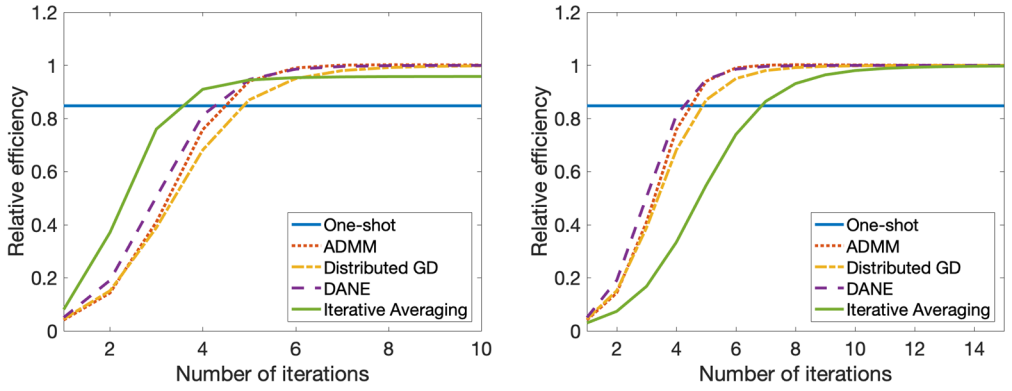

FIG. 5. *Comparison of the one-shot weighted method and several widely used multishot methods.*

See Figure 6 for an example. In contrast, weighted averaging requires less tuning, making it a more attractive method.

We have performed several more numerical simulations to verify our theory, in addition to the results shown in the paper. Due to space limitations, these are presented in the Supplementary Material (Dobriban and Sheng (2021)). In Section K, we present an empirical data example to assess the accuracy of our theoretical results for one-shot averaging. We find that they can be quite accurate.

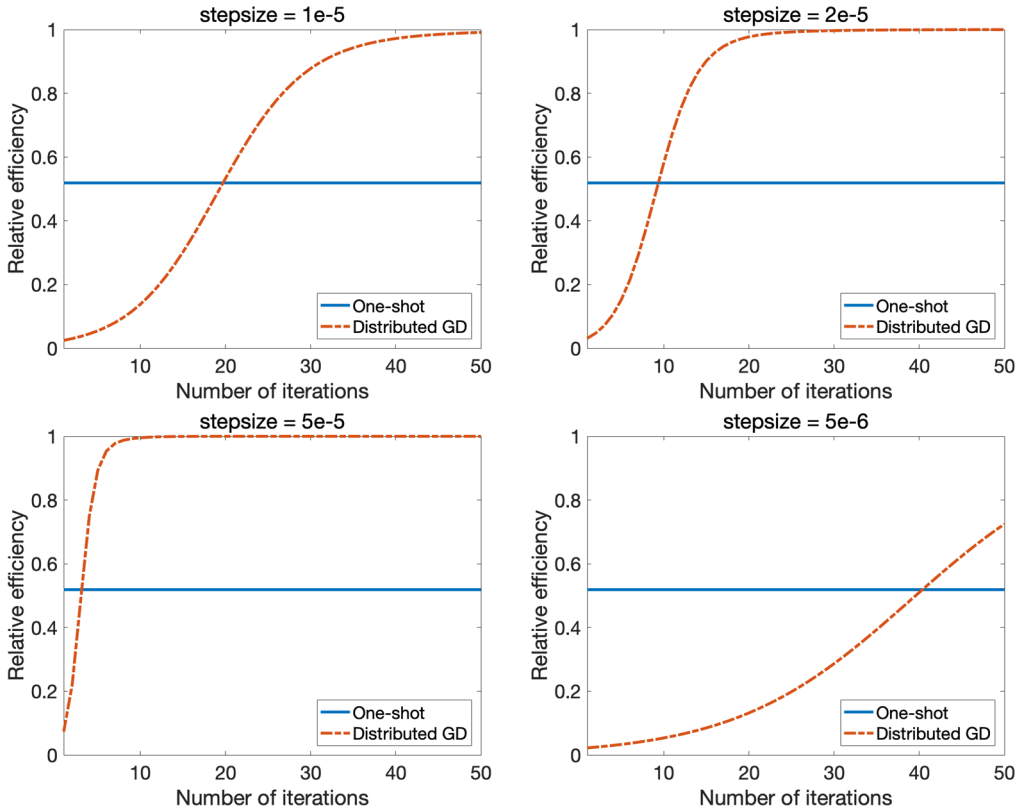

FIG. 6. *Comparison of the one-shot weighted method and distributed GD with different stepsizes. The point of this figure is that the behavior of GD depends strongly on the stepsize. In particular, the number of iterations needed to reach the performance of one-shot regression can vary a lot.*

Acknowledgments. We thank the Associate Editor and referees for helpful comments that have significantly improved the paper. We thank Jason D. Lee, Philip Gressman, Andreas Haeberleen, Boaz Nadler, Balasubramanian Narashiman and Ziwei Zhu for helpful discussions. We are grateful to Sifan Liu for providing an initial script for processing the empirical data. We thank John Duchi for pointing out references from convex optimization showing the concavity of the relative efficiencies (Proposition 3.1).

This work was partially supported by NSF BIGDATA grant IIS 1837992.

SUPPLEMENTARY MATERIAL

Supplement to "Distributed linear regression by averaging" (DOI: 10.1214/20-AOS1984SUPP; .pdf). The supplement contains mathematical proofs and results to complete the main text, additional numerical simulation results, and examples of empirical data analysis.

REFERENCES

AGARWAL, A., CHAPELLE, O., DUDÍK, M. and LANGFORD, J. (2014). A reliable effective terascale linear learning system. *J. Mach. Learn. Res.* **15** 1111–1133. MR3195340

ALI, A., KOLTER, J. Z. and TIBSHIRANI, R. J. (2019). A continuous-time view of early stopping for least squares regression. In *Proceedings of Machine Learning Research* **89** 1370–1378.

BAI, Z. and SILVERSTEIN, J. W. (2009). *Spectral Analysis of Large Dimensional Random Matrices*. *Springer Series in Statistics*. Springer, New York. MR2567175 https://doi.org/10.1007/978-1-4419-0661-8

BANERJEE, M. and DUROT, C. (2018). Removing the curse of superefficiency: An effective strategy for distributed computing in isotonic regression. Preprint. Available at arXiv:1806.08542.

BANERJEE, M., DUROT, C. and SEN, B. (2019). Divide and conquer in nonstandard problems and the superefficiency phenomenon. *Ann. Statist.* **47** 720–757. MR3909948 https://doi.org/10.1214/17-AOS1633

BATTEY, H., FAN, J., LIU, H., LU, J. and ZHU, Z. (2018). Distributed testing and estimation under sparse high dimensional models. *Ann. Statist.* **46** 1352–1382. MR3798006 https://doi.org/10.1214/17-AOS1587

BEKKERMAN, R., BILENKO, M. and LANGFORD, J. (2011). *Scaling up Machine Learning*: *Parallel and Distributed Approaches*. Cambridge Univ. Press, Cambridge.

BERTSEKAS, D. P. and TSITSIKLIS, J. N. (1989). *Parallel and Distributed Computation*: *Numerical Methods* **23**. Prentice Hall, Englewood Cliffs, NJ.

BOYD, S. and VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge Univ. Press, Cambridge. MR2061575 https://doi.org/10.1017/CBO9780511804441

BOYD, S., PARIKH, N., CHU, E., PELEATO, B. and ECKSTEIN, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3** 1–122.

BRAVERMAN, M., GARG, A., MA, T., NGUYEN, H. L. and WOODRUFF, D. P. (2016). Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *STOC'*16—*Proceedings of the* 48*th Annual ACM SIGACT Symposium on Theory of Computing* 1011–1020. ACM, New York. MR3536632 https://doi.org/10.1145/2897518.2897582

CHEN, X., LIU, W. and ZHANG, Y. First-order newton-type estimator for distributed estimation and inference. Preprint. Available at arXiv:1811.11368.

CHEN, X., LIU, W. and ZHANG, Y. (2019). Quantile regression under memory constraint. *Ann. Statist.* **47** 3244–3273. MR4025741 https://doi.org/10.1214/18-AOS1777

CHEN, X. and XIE, M. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statist. Sinica* **24** 1655–1684. MR3308656

CHU, C.-T., KIM, S. K., LIN, Y.-A., YU, Y., BRADSKI, G., OLUKOTUN, K. and NG, A. Y. (2007). Map-reduce for machine learning on multicore. In *Advances in Neural Information Processing Systems* 281–288.

COUILLET, R. and DEBBAH, M. (2011). *Random Matrix Methods for Wireless Communications*. Cambridge Univ. Press, Cambridge. MR2884783 https://doi.org/10.1017/CBO9780511994746

COUILLET, R., DEBBAH, M. and SILVERSTEIN, J. W. (2011). A deterministic equivalent for the analysis of correlated MIMO multiple access channels. *IEEE Trans. Inf. Theory* **57** 3493–3514. MR2817033 https://doi.org/10.1109/TIT.2011.2133151

DAVIS, C. (1957). All convex invariant functions of Hermitian matrices. *Arch. Math.* **8** 276–278. MR0090572 https://doi.org/10.1007/BF01898787

DEAN, J. and GHEMAWAT, S. (2008). Mapreduce: Simplified data processing on large clusters. *Commun. ACM* **51** 107–113.

DOBRIBAN, E. and SHENG, Y. (2019). One-shot distributed ridge regression in high dimensions. Preprint. Available at arXiv:1903.09321.

DOBRIBAN, E. and SHENG, Y. (2021). Supplement to "Distributed linear regression by averaging." https://doi.org/10.1214/20-AOS1984SUPP

DONOHO, D. and MONTANARI, A. (2016). High dimensional robust M-estimation: Asymptotic variance via approximate message passing. *Probab. Theory Related Fields* **166** 935–969. MR3568043 https://doi.org/10.1007/s00440-015-0675-z

DUCHI, J. C., AGARWAL, A. and WAINWRIGHT, M. J. (2012). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Automat. Control* **57** 592–606. MR2932818 https://doi.org/10.1109/TAC.2011.2161027

DUCHI, J. C., JORDAN, M. I., WAINWRIGHT, M. J. and ZHANG, Y. (2014). Optimality guarantees for distributed statistical estimation. Preprint. Available at arXiv:1405.0782.

EL KAROUI, N., BEAN, D., BICKEL, P. J., LIM, C. and YU, B. (2013). On robust regression with high-dimensional predictors. *Proc. Natl. Acad. Sci. USA* **110** 14557–14562.

FAN, J., GUO, Y. and WANG, K. (2019). Communication-efficient accurate statistical estimation. Preprint. Available at arXiv:1906.04870.

FAN, J., WANG, D., WANG, K. and ZHU, Z. (2019). Distributed estimation of principal eigenspaces. *Ann. Statist.* **47** 3009–3031. MR4025733 https://doi.org/10.1214/18-AOS1713

HACHEM, W., LOUBATON, P. and NAJIM, J. (2007). Deterministic equivalents for certain functionals of large random matrices. *Ann. Appl. Probab.* **17** 875–930. MR2326235 https://doi.org/10.1214/105051606000000925

HUO, X. and CAO, S. (2019). Aggregated inference. *Wiley Interdiscip. Rev.: Comput. Stat.* **11** e1451, 13. MR3897175 https://doi.org/10.1002/wics.1451

JORDAN, M. I., LEE, J. D. and YANG, Y. (2019). Communication-efficient distributed statistical inference. *J. Amer. Statist. Assoc.* **114** 668–681. MR3963171 https://doi.org/10.1080/01621459.2018.1429274

LEE, J. D., LIU, Q., SUN, Y. and TAYLOR, J. E. (2017). Communication-efficient sparse regression. *J. Mach. Learn. Res.* **18** Paper No. 5, 30. MR3625709

LEWIS, A. S. (1996). Convex analysis on the Hermitian matrices. *SIAM J. Optim.* **6** 164–177. MR1377729 https://doi.org/10.1137/0806009

LIN, S.-B., GUO, X. and ZHOU, D.-X. (2017). Distributed learning with regularized least squares. *J. Mach. Learn. Res.* **18** Paper No. 92, 31. MR3714255 https://doi.org/10.1016/j.physletb.2016.11.035

LIU, Q. and IHLER, A. T. (2014). Distributed estimation, information loss and exponential families. In *Advances in Neural Information Processing Systems* 1098–1106.

MARCHENKO, V. A. and PASTUR, L. A. (1967). Distribution of eigenvalues in certain sets of random matrices. *Mat. Sb. (N.S.)* **72 (114)** 507–536. MR0208649

MARDIA, K. V., KENT, J. T. and BIBBY, J. M. (1979). *Multivariate Analysis. Probability and Mathematical Statistics: A Series of Monographs and Textbooks*. Academic Press, London. MR0560319

MCDONALD, R., MOHRI, M., SILBERMAN, N., WALKER, D. and MANN, G. S. (2009). Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems* 1231–1239.

MÜLLER, A. and DEBBAH, M. (2016). Random matrix theory tutorial—Introduction to deterministic equivalents. *Traitement Signal* **33** 223–248.

NEDIĆ, A. and OZDAGLAR, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Automat. Control* **54** 48–61. MR2478070 https://doi.org/10.1109/TAC.2008.2009515

NESTEROV, Y. (2009). Primal-dual subgradient methods for convex problems. *Math. Program.* **120** 221–259. MR2496434 https://doi.org/10.1007/s10107-007-0149-x

PEACOCK, M. J. M., COLLINGS, I. B. and HONIG, M. L. (2008). Eigenvalue distributions of sums and products of large random matrices via incremental matrix expansions. *IEEE Trans. Inf. Theory* **54** 2123–2138. MR2450853 https://doi.org/10.1109/TIT.2008.920221

ROSENBLATT, J. D. and NADLER, B. (2016). On the optimality of averaging in distributed statistical learning. *Inf. Inference* **5** 379–404. MR3609865 https://doi.org/10.1093/imaiai/iaw013

RUBIO, F. and MESTRE, X. (2011). Spectral convergence for a general class of random matrices. *Statist. Probab. Lett.* **81** 592–602. MR2772917 https://doi.org/10.1016/j.spl.2011.01.004

SERDOBOLSKII, V. I. (1983). On minimum error probability in discriminant analysis. *Dokl. Akad. Nauk SSSR* **27** 720–725.

SERDOBOLSKII, V. I. (2007). *Multiparametric Statistics*. Elsevier, Amsterdam. MR2531357

SHAMIR, O., SREBRO, N. and ZHANG, T. (2014). Communication-efficient distributed optimization using an approximate Newton-type method. In *Proceedings of the* 31*st International Conference on Machine Learning* **32** 1000–1008.

SHI, C., LU, W. and SONG, R. (2018). A massive data framework for M-estimators with cubic-rate. *J. Amer. Statist. Assoc.* **113** 1698–1709. MR3902239 https://doi.org/10.1080/01621459.2017.1360779

SMITH, V., FORTE, S., MA, C., TAKÁČ, M., JORDAN, M. I. and JAGGI, M. (2017). CoCoA: A general framework for communication-efficient distributed optimization. *J. Mach. Learn. Res.* **18** Paper No. 230, 49. MR3845529

SZABO, B. and VAN ZANTEN, H. (2018). Adaptive distributed methods under communication constraints. Preprint. Available at arXiv:1804.00864.

TANDON, R., LEI, Q., DIMAKIS, A. G. and KARAMPATZIAKIS, N. (2017). Gradient coding: Avoiding stragglers in distributed learning. In *International Conference on Machine Learning* 3368–3376.

TSITSIKLIS, J. N., BERTSEKAS, D. P. and ATHANS, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Automat. Control* **31** 803–812. MR0853431 https://doi.org/10.1109/TAC.1986.1104412

TULINO, A. M. and VERDÚ, S. (2004). Random matrix theory and wireless communications. *Commun. Inf. Theory* **1** 1–182.

VOLGUSHEV, S., CHAO, S.-K. and CHENG, G. (2019). Distributed inference for quantile regression processes. *Ann. Statist.* **47** 1634–1662. MR3911125 https://doi.org/10.1214/18-AOS1730

WANG, J., KOLAR, M., SREBRO, N. and ZHANG, T. (2017). Efficient distributed learning with sparsity. In *Proceedings of the* 34*th International Conference on Machine Learning* **70** 3636–3645. JMLR.org.

YAO, Y., ROSASCO, L. and CAPONNETTO, A. (2007). On early stopping in gradient descent learning. *Constr. Approx.* **26** 289–315. MR2327601 https://doi.org/10.1007/s00365-006-0663-2

ZAHARIA, M., CHOWDHURY, M., FRANKLIN, M. J., SHENKER, S. and STOICA, I. (2010). Spark: Cluster computing with working sets. *HotCloud* **10** 95.

ZHANG, Y., DUCHI, J. and WAINWRIGHT, M. (2013a) Divide and conquer kernel ridge regression. In *Conference on Learning Theory* 592–617.

ZHANG, Y., DUCHI, J. C. and WAINWRIGHT, M. J. (2013b). Communication-efficient algorithms for statistical optimization. *J. Mach. Learn. Res.* **14** 3321–3363. MR3144464

ZHANG, Y., DUCHI, J. and WAINWRIGHT, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res.* **16** 3299–3340. MR3450540

ZHANG, Y., WAINWRIGHT, M. J. and DUCHI, J. C. (2012). Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems* 1502–1510.

ZHAO, T., CHENG, G. and LIU, H. (2016). A partially linear framework for massive heterogeneous data. *Ann. Statist.* **44** 1400–1437. MR3519928 https://doi.org/10.1214/15-AOS1410

ZHU, Y. and LAFFERTY, J. (2018). Distributed nonparametric regression under communication constraints. Preprint. Available at arXiv:1803.01302.

ZINKEVICH, M., LANGFORD, J. and SMOLA, A. J. (2009). Slow learners are fast. In *Advances in Neural Information Processing Systems* 2331–2339.

ZINKEVICH, M., WEIMER, M., LI, L. and SMOLA, A. J. (2010). Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems* 2595–2603.