# One-shot Distributed Ridge Regression in High Dimensions

**Edgar Dobriban** [*1] **Yue Sheng** [*1 2]

## Abstract

To scale up data analysis, distributed and parallel computing approaches are increasingly needed. Here we study a fundamental problem in this area: How to do ridge regression in a distributed computing environment? We study one-shot methods constructing weighted combinations of ridge regression estimators computed on each machine. By analyzing the mean squared error in a high dimensional model where each predictor has a small effect, we discover several new phenomena including that the *efficiency depends strongly on the signal strength*, but does not degrade with many workers, the risk *decouples* over machines, and the unexpected consequence that the *optimal weights do not sum to unity*. We also propose a new optimally weighted one-shot ridge regression algorithm. Our results are supported by simulations and real data analysis.

## 1. Introduction

Computers have changed all aspects of our world. Importantly, computing has made data analysis more convenient than ever before. However, computers also pose limitations and challenges for data science. For instance, hardware architecture is based on a model of a universal computer—a Turing machine—but in fact has physical limitations of storage, memory, processing speed, and communication bandwidth over a network. As large datasets become more and more common in all areas of human activity, we need to think carefully about working with these limitations.

How can we design methods for data analysis (statistics and machine learning) that scale to large datasets? A general approach is *distributed and parallel computing*. Roughly speaking, the data is divided up among computing units, which perform most of the computation locally, and synchronize by passing relatively short messages. While the idea is simple, a good implementation can be hard. Moreover, different problems have different inherent needs in terms of local computation and global communication resources. For instance, in statistical problems with high levels of noise, simple one-shot schemes like averaging estimators computed on local datasets can sometimes work well.

In this paper, we study a fundamental problem in this area. We are interested in linear regression, which is arguably one of the most important problems in statistics and machine learning. A popular method for this model is *ridge regression* (also known as $\ell_2$ or Tikhonov regularization), which regularizes the estimates using a quadratic penalty to improve estimation and prediction accuracy. We aim to understand how to do ridge regression in a distributed computing environment. We are also interested in the *high-dimensional* setting, where the number of features can be large. We also work in a random-effects model where each predictor has a small effect on the outcome, which is the model for which ridge regression is best suited.

### 1.1. Main Results and Contributions

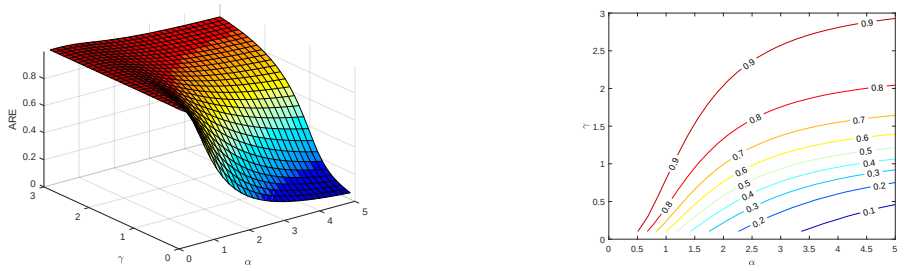
In contrast to existing work, we introduce a new mathematical approach to the problem. We leverage and further develop sophisticated recent techniques from random matrix theory and free probability theory in our analysis. This enables us to make contributions that were unattainable using more "traditional" mathematical approaches.

We find the limiting mean squared error of the one-shot distributed ridge estimator, which takes a weighted sum of the local ridge estimators computed on individual machines. This is a highly communication efficient method. We characterize the optimal weights and tuning parameters, as well as the *relative efficiency* compared to centralized ridge regression, meaning the ratio of the risk of usual ridge to the distributed estimator. This can precisely pinpoint the computation-accuracy tradeoff achieved via one-shot distributed estimation. See Figure 1 for an illustration.

As a consequence of our detailed risk analysis, we make several discoveries:

[*]Equal contribution [1]Wharton Statistics Department, University of Pennsylvania, Philadelphia, PA, USA [2]Graduate Group in Applied Mathematics and Computational Science, University of Pennsylvania, Philadelphia, PA, USA. Correspondence to: Edgar Dobriban <dobriban@wharton.upenn.edu>, Yue Sheng <yuesheng@sas.upenn.edu>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Online, PMLR 119, 2020. 




(a) Surface (b) Contour

*Figure 1.* Efficiency loss due to one-shot distributed learning. This plot shows the relative mean squared error of centralized ridge regression compared to optimally weighted one-shot distributed ridge regression. This quantity is at most unity, and the larger, the "better" distributed ridge works. See the text for details.

**Efficiency Depends Strongly on Signal Strength.** The statistical efficiency of the one-shot distributed ridge estimator depends strongly on signal strength. The efficiency is generally high (meaning distributed ridge regression works well) when the signal strength is low.

**Infinite-worker Limit.** The one-shot distributed estimator does not lose all efficiency compared to the ridge estimator even in the limit of *infinitely many machines*. Somewhat surprisingly, simple one-shot weighted combination methods work well even for very large numbers of machines. It is nontrivial that this can be achieved by communication-efficient methods. This is also important from a practical perspective.

**Decoupling.** When the features are uncorrelated, the problem of choosing the optimal regularization parameters *decouples* over the different machines. We can choose them in a locally optimal way, and they are also globally optimal. This is a delicate result, and is not true in general for correlated features. Moreover, this discovery is also important in practice, because it gives conditions under which we can choose the regularization parameters separately for each machine, thus saving valuable computational resources.

**Optimal Weights Do not Sum to Unity.** We uncover unexpected properties of the optimal weights. Naively, one may think that the weights need to sum to unity, meaning that we need a weighted average. However, it turns out the optimal weights sum to more than unity, because of the negative bias of the ridge estimator. This means that *any type of averaging method is suboptimal*.

Based on these results, we propose a new optimally weighted one-shot ridge regression algorithm. We also confirm these results in detailed simulation studies and on an empirical data example, using the Million Song Dataset.

Some aspects of our work can help practitioners directly, while others are developed for deepening our understanding of the problem.

### 1.2. Prior Work

The area of distributed statistics and machine learning has attracted increasing attention only relatively recently, see for instance (Mcdonald et al., 2009; Zhang et al., 2012; 2013b), (Li et al., 2013; Zhang et al., 2013a; Duchi et al., 2014; Chen & Xie, 2014; Mackey et al., 2011; Zhang et al., 2015; Braverman et al., 2016; Jordan et al., 2016; Rosenblatt & Nadler, 2016; Smith et al., 2016; Banerjee et al., 2016; Zhao et al., 2016; Xu et al., 2016; Fan et al., 2017; Lin et al., 2017; Lee et al., 2017; Volgushev et al., 2017; Shang & Cheng, 2017; Battey et al., 2018; Zhu & Lafferty, 2018; Chen et al., 2018a;b; Wang et al., 2018; Shi et al., 2018; Duan et al., 2018; Liu et al., 2018; Richards et al., 2020), and the references therein. See (Huo & Cao, 2018) for a review. We can only discuss the most closely related papers due to space limitations.

(Zhang et al., 2013b) study the MSE of averaged estimation in empirical risk minimization. Later (Zhang et al., 2015) study divide and conquer *kernel ridge regression*, showing that the partition-based estimator achieves the statistical minimax rate over all estimators, when the number of machines is not too large. These results are very general, however they are not as explicit or precise as our results. (Lin et al., 2017) improve the above results, removing certain eigenvalue assumptions on the kernel, and sharpening the rate. In the same framework, (Guo et al., 2017) study regularization kernel networks, and propose a debiasing scheme that can improve the behavior of distributed estimators. (Xu et al., 2016) propose a distributed General Cross-Validation method to choose the regularization parameter.

(Rosenblatt & Nadler, 2016) consider averaging in distributed learning in fixed and high-dimensional M-estimation, without studying regularization. (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) in addition studies hypothesis testing under more general sparse models. These last two works

are on a different problem (sparse regression), whereas we study ridge regression in random-effects models.

### 1.3. Paper Organization

Section 2 introduces our model. Section 3 presents our main theoretical results. We start with finite sample results, then provide asymptotic results for generally correlated features. We consider the special case of an identity covariance, where we study in detail the properties of the estimation error, tuning parameters and decoupling. We also provide a practical algorithm. Section 4 contains experiments on real data. The supplementary material provides proofs for the theorems, additional discussion and experiments.

## 2. Preliminaries

We consider the standard linear model

$$Y = X\beta + \varepsilon. \tag{1}$$

Here $Y \in \mathbb{R}^n$ is the $n$-dimensional continuous outcome vector of $n$ independent samples, $X$ is the $n \times p$ design matrix containing the values of $p$ features for each sample. Moreover, $\beta = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$ is the $p$-dimensional vector of unknown regression coefficients. We assume that the coordinates of the random noise $\varepsilon$ are independent random variables with mean zero and variance $\sigma^2$.

The *ridge regression* (or Tikhonov regularization) estimator is a popular method for estimation and prediction in linear models. Recall that the ridge estimator of $\beta$ is

$$\widehat{\beta}(\lambda) = (X^\top X + n\lambda I_p)^{-1} X^\top Y, \tag{2}$$

where $\lambda$ is a tuning parameter.

Suppose we are in a distributed computation setting. The samples are distributed across $k$ sites or machines. For simplicity we call the sites "machines". We can write the partitioned data as

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

Thus the $i$-th machine contains $n_i$ samples whose features are stored in the $n_i \times p$ matrix $X_i$ and also the corresponding $n_i \times 1$ outcome vector $Y_i$.

Since the ridge regression estimator is a widely used method with certain optimality properties, we aim to understand how we can approximate it in a distributed setting. Specifically, we will focus on one-shot *weighting* methods, where we perform ridge regression locally on each subset of the data, and then aggregate the regression coefficients by a weighted sum. There are several reasons to consider weighting methods:

1. This is a practical method with *minimal communication cost*. When communication is expensive, it is imperative to develop methods that minimize communication cost. In this case, one-shot weighting methods are attractive, and so it is important to understand how they work. In a well-known course on scalable machine learning, Alex Smola calls such methods "idiot-proof" (Smola, 2012), meaning that they are straightforward to implement (unlike some of the more sophisticated methods).
2. Averaging (which is a special case of one-shot weighting) has already been studied in several works on distributed ridge regression (e.g., (Zhang et al., 2015; Lin et al., 2017)), and much more broadly in distributed learning, see the related work section for details. Such methods are *known to be rate-optimal* under certain conditions.
3. Weighting may serve as a useful *initialization to iterative methods*. In practical distributed learning problems, iterative optimization algorithms such as distributed gradient descent may be used.

Therefore, we define *local* ridge estimators for each dataset $X_i, Y_i$, with regularization parameter $\lambda_i$ as

$$\widehat{\beta}_i(\lambda_i) = (X_i^\top X_i + n_i\lambda_i I_p)^{-1} X_i^\top Y_i. \tag{3}$$

We consider combining the local ridge estimators at a central server via a one-step weighted summation. We will find the optimally weighted one-shot distributed estimator

$$\widehat{\beta}_{\text{dist}}(w) = \sum_{i=1}^{k} w_i \widehat{\beta}_i. \tag{4}$$

We will write $\widehat{\Sigma} = n^{-1} X^\top X$ for the sample (uncentered) covariance and $\widehat{\Sigma}_i = n_i^{-1} X_i^\top X_i$ the sample covariance of the local datasets.

## 3. Main Results

### 3.1. Finite Sample Results

A stepping stone to our analysis is the following key result.

**Theorem 1** (Finite sample risk and efficiency of optimally weighted distributed ridge for fixed regularization parameters)**.** *Consider the distributed ridge regression problem described above. The optimal weights that minimize the mean squared error $\mathbb{E}\|\widehat{\beta}_{\text{dist}}(w) - \beta\|^2$ of the distributed estimator $\widehat{\beta}_{\text{dist}}(w)$ are*

$$w^* = (A + R)^{-1} v, \tag{5}$$

*where the quantities $v, A, R$ are defined below.*

1. *$v$ is a $k$-dimensional vector with $i$-th coordinate $\beta^\top Q_i \beta$, and $Q_i$ are the $p \times p$ matrices $Q_i = (\widehat{\Sigma}_i + \lambda_i I_p)^{-1} \widehat{\Sigma}_i$,*

2. *$A$ is a $k \times k$ matrix with $(i,j)$-th entry $\beta^\top Q_i Q_j \beta$, and*
3. *$R$ is a $k \times k$ diagonal matrix with $i$-th diagonal entry $n_i^{-1}\sigma^2 \operatorname{tr}[(\widehat{\Sigma}_i + \lambda_i I_p)^{-2}\widehat{\Sigma}_i]$.*

*The mean squared error of the* optimally weighted *distributed ridge regression estimator $\widehat{\beta}_{\text{dist}}$ with $k$ sites equals*

$$\text{MSE}^*_{\text{dist}} = \mathbb{E}\|\widehat{\beta}_{\text{dist}}(w^*) - \beta\|^2 = \|\beta\|^2 - v^\top (A+R)^{-1} v. \tag{6}$$

See the supplementary material for the proof. This result quantifies the mean squared error of the optimally weighted distributed ridge estimator for fixed regularization parameters $\lambda_i$. Later we will choose the regularization parameters optimally. The result gives an exact formula for the optimal weights. However, they depend on the unknown regression coefficients $\beta$, and are thus not directly usable in practice. Instead, our approach is to make stronger assumptions on $\beta$ under which we can develop estimators for the weights.

### 3.2. Asymptotic Results

We will consider an asymptotic setting that leads to more explicit results. We first introduce the model and some fundamentals of random matrix theory.

**Random-effects Model.** Recall our basic linear model (1). Next, we also assume that a *random-effects model* holds. We assume $\beta$ is random—independently of $\varepsilon$—with coordinates that are themselves independent random variables with mean zero and variance $p^{-1}\sigma^2\alpha^2$. Thus, each feature contributes a small random amount to the outcome. Ridge regression is designed to work well in such a setting, and has several optimality properties in variants of this model. The parameters are now $\theta = (\sigma^2, \alpha^2)$: the *noise level* $\sigma^2$ and the *signal-to-noise ratio* $\alpha^2$ respectively. This parametrization is standard (e.g. (Searle et al., 2009; Dicker & Erdogdu, 2017; Dobriban & Wager, 2018)).

**Random Matrix Theory (RMT).** We will focus on "Marchenko-Pastur" (MP) type sample covariance matrices, which are popular in statistics (see e.g., (Bai & Silverstein, 2009; Anderson, 2003; Paul & Aue, 2014; Yao et al., 2015)). A key concept is the spectral distribution, which for a $p \times p$ symmetric matrix $A$ is the distribution $F_A$ that places equal mass on all eigenvalues $\lambda_i(A)$ of $\Sigma$. This has cumulative distribution function (CDF) $F_A(x) = p^{-1}\sum_{i=1}^p \mathbf{1}(\lambda_i(A) \le x)$. A central result in the area is the Marchenko-Pastur theorem, which states that eigenvalue distributions of sample covariance matrices converge (Marchenko & Pastur, 1967; Bai & Silverstein, 2009). We state the required assumptions below:

**Assumption 1.** *Consider the following conditions:*

1. *The $n \times p$ design matrix $X$ is generated as $X = Z\Sigma^{1/2}$ for an $n \times p$ matrix $Z$ with i.i.d. entries (viewed as coming from an infinite array), satisfying $\mathbb{E}[Z_{ij}] = 0$ and $\mathbb{E}[Z_{ij}^2] = 1$, and a deterministic $p \times p$ positive semidefinite population covariance matrix $\Sigma$.*
2. *The sample size $n$ grows to infinity proportionally with the dimension $p$, i.e. $n, p \to \infty$ and $p/n \to \gamma \in (0, \infty)$.*
3. *The sequence of spectral distributions $F_\Sigma := F_{\Sigma,n,p}$ of $\Sigma := \Sigma_{n,p}$ converges weakly to a limiting distribution $H$ supported on $[0,\infty)$, called the population spectral distribution.*

Then, the Marchenko-Pastur theorem states that with probability 1, the spectral distribution $F_{\widehat{\Sigma}}$ of the sample covariance matrix $\widehat{\Sigma}$ also converges weakly (in distribution) to a limiting distribution $F_\gamma := F_\gamma(H)$ supported on $[0,\infty)$ (Marchenko & Pastur, 1967; Bai & Silverstein, 2009). The limiting distribution is determined uniquely by a fixed-point equation for its *Stieltjes transform*, which is defined for any distribution $G$ supported on $[0,\infty)$ as

$$m_G(z) := \int_0^\infty \frac{1}{t-z} dG(t), \quad z \in \mathbb{C} \setminus \mathbb{R}^+. \tag{7}$$

With this notation, the Stieltjes transform of the spectral measure of $\widehat{\Sigma}$ satisfies

$$m_{\widehat{\Sigma}}(z) = p^{-1}\operatorname{tr}[(\widehat{\Sigma} - zI_p)^{-1}] \to_{a.s.} m_{F_\gamma}(z), \quad z \in \mathbb{C}\backslash\mathbb{R}^+, \tag{8}$$

where $m_{F_\gamma}(z)$ is the Stieltjes transform of $F$.

We are now ready to study the asymptotics of the risk. We will denote by $T$ a random variable distributed according to $H$, so that $\mathbb{E}_H g(T)$ denotes the mean of $g(T)$ when $T$ is a random variable distributed according to the limit spectral distribution $H$.

Now, we can state one of our main results.

**Theorem 2** (Asymptotics for distributed ridge, arbitrary regularization)**.** *In the linear random-effects model under Assumption 1, suppose that the eigenvalues of $\Sigma$ are uniformly bounded away from zero and infinity, and that the entries of $Z$ have a finite $8+c$-th moment for some $c > 0$. Suppose moreover that the local sample sizes $n_i$ grow proportionally to $p$, so that $p/n_i \to \gamma_i > 0$.*

*Then the optimal weights for distributed ridge regression, and its MSE, converge to definite limits. Recall from Theorem 1 that we have the formulas $w^* = (A+R)^{-1}v$ and $\text{MSE}^*_{\text{dist}} = \|\beta\|^2 - v^\top(A+R)^{-1}v$ for the optimal finite sample weights and risk, and thus it is enough to find the limit of $v$, $A$ and $R$. These have the following limits:*

1. *With probability one, we have the convergence $v \to V \in \mathbb{R}^k$. The $i$-th coordinate of the limit $V$ is*

$$V_i = \sigma^2\alpha^2 \mathbb{E}_H \frac{x_i T}{x_i T + \lambda_i} = \sigma^2\alpha^2(1 - \lambda_i m_{F_{\gamma_i}}(-\lambda_i)). \tag{9}$$

*Recall that $H$ is the limiting population spectral distribution of $\Sigma$, and $T$ is a random variable distributed according to $H$. Among the empirical quantities, $F_{\gamma_i}$ is the limiting empirical spectral distribution of $\widehat{\Sigma}_i$ and $x_i := x_i(H, \lambda_i, \gamma_i) > 0$ exists as the unique solution of the fixed point equation*

$$1 - x_i = \gamma_i \left[1 - \mathbb{E}_H \frac{\lambda_i}{x_i T + \lambda_i}\right]. \tag{10}$$

2. *With probability one, $A \to \mathcal{A} \in \mathbb{R}^{k\times k}$. For $i \neq j$, the $(i, j)$-th entry of $\mathcal{A}$ is, in terms of the population spectral distribution $H$,*

$$\mathcal{A}_{ij} = \sigma^2\alpha^2 \mathbb{E}_H \frac{x_i x_j T^2}{(x_i T + \lambda_i)(x_j T + \lambda_j)}. \tag{11}$$

*The $i$-th diagonal entry of $\mathcal{A}$ is*

$$\mathcal{A}_{ii} = \sigma^2\alpha^2 \left[1 - 2\lambda_i m_{F_{\gamma_i}}(-\lambda_i) + \lambda_i^2 m'_{F_{\gamma_i}}(-\lambda_i)\right]. \tag{12}$$

3. *With probability one, the diagonal matrix $R$ converges, $R \to \mathcal{R} \in \mathbb{R}^{k\times k}$, where of course $\mathcal{R}$ is also diagonal. The $i$-th diagonal entry of $\mathcal{R}$ is*

$$\mathcal{R}_{ii} = \sigma^2 \left[\gamma_i m_{F_{\gamma_i}}(-\lambda_i) - \gamma_i\lambda_i m'_{F_{\gamma_i}}(-\lambda_i)\right]. \tag{13}$$

*The limiting weights and MSE are then*

$$\mathcal{W}_k^* = (\mathcal{A} + \mathcal{R})^{-1} V$$

*and*

$$\mathcal{M}_k = \sigma^2\alpha^2 - V^\top (\mathcal{A} + \mathcal{R})^{-1} V.$$

See the supplementary material for the proof. The statement may look complicated, but the formulas simplify considerably in the uncorrelated case $\Sigma = I_p$.

### 3.3. Special Case: Identity Covariance

When the population covariance matrix is the identity, that is $\Sigma = I$, the results simplify considerably. In this case the features are uncorrelated. It is known that the limiting Stieltjes transform $m_{F_\gamma} := m_\gamma$ of $\widehat{\Sigma}$ has the explicit form (Marchenko & Pastur, 1967):

$$m_\gamma(z) = \frac{(z + \gamma - 1) + \sqrt{(z + \gamma - 1)^2 - 4z\gamma}}{-2z\gamma}. \tag{14}$$

We can get explicit formulas for the optimal weights using this expression. From Theorem 2, we conclude the following simplified result:

**Theorem 3** (Asymptotics for isotropic population covariance, arbitrary regularization parameters)**.** *In addition to the assumptions of Theorem 2, suppose that the population covariance matrix $\Sigma = I$. Then the limits of $v, A$ and $R$ have simple explicit forms:*

1. *The $i$-th coordinate of $V$ is:*

$$V_i = \sigma^2\alpha^2[1 - \lambda_i m_{\gamma_i}(-\lambda_i)], \tag{15}$$

*where $m_{\gamma_i}(-\lambda_i)$ is the Stieltjes transform from equation (14).*

2. *The off-diagonal entries of $\mathcal{A}$ are*

$$\mathcal{A}_{ij} = \sigma^2\alpha^2[1 - \lambda_i m_{\gamma_i}(-\lambda_i)] \cdot [1 - \lambda_j m_{\gamma_j}(-\lambda_j)]. \tag{16}$$

*The diagonal entries $\mathcal{A}$ are*

$$\mathcal{A}_{ii} = \sigma^2\alpha^2 \left[1 - 2\lambda_i m_{\gamma_i}(-\lambda_i) + \lambda_i^2 m'_{\gamma_i}(-\lambda_i)\right]. \tag{17}$$

3. *The $i$-th diagonal entry of $\mathcal{R}$ is*

$$\mathcal{R}_{ii} = \sigma^2\gamma_i \left[m_{\gamma_i}(-\lambda_i) - \lambda_i m'_{\gamma_i}(-\lambda_i)\right]. \tag{18}$$

*The limiting optimal weights for combining the local ridge estimators are $\mathcal{W}_k^* = (\mathcal{A} + \mathcal{R})^{-1}V$, and MSE of the optimally weighted distributed estimator is*

$$\mathcal{M}_k = \frac{\sigma^2\alpha^2}{1 + \sum_{i=1}^k \frac{V_i^2}{\sigma^2\alpha^2(\mathcal{R}_{ii} + \mathcal{A}_{ii}) - V_i^2}}. \tag{19}$$

This theorem shows the surprising fact that the limiting risk *decouples* over the different machines. By this we mean that the liming risk can be written in a simple form, involving a sum of terms depending on each machine, without any interaction. This seems like a major surprise. See the supplementary material for the proof and more detailed explanation on the decoupling phenomenon.

An important consequence of the decoupling is that *we can optimize the individual risks over the tuning parameters $\lambda_i$ separately*.

**Theorem 4** (Optimal regularization (tuning) parameters, and risk)**.** *Under the assumptions of Theorem 3, the optimal regularization (tuning) parameters $\lambda_i$ that minimize the local MSEs also minimize the distributed risk $\mathcal{M}_k$. They have the form*

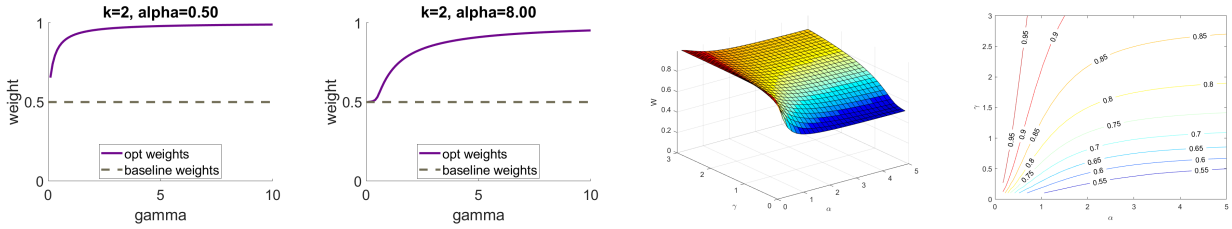
$$\lambda_i = \frac{\gamma_i}{\alpha^2}, \quad i = 1, 2, \ldots, k. \tag{20}$$

*Moreover, the risk $\mathcal{M}_k$ of distributed ridge regression with optimally tuned regularization parameters is*

$$\mathcal{M}_k = \frac{\sigma^2\alpha^2}{1 + \sum_{i=1}^k \left[\frac{\alpha^2}{\gamma_i m_{\gamma_i}(-\gamma_i/\alpha^2)} - 1\right]}, \tag{21}$$

See the supplementary material for the proof.

To compare the distributed and centralized estimators, we will study their (asymptotic) relative efficiency (ARE), which is the (limit of the) ratio of their mean squared errors.




(a) $\alpha$=0.5 (b) $\alpha$=8 (c) Surface (d) Contour

*Figure 2.* Plots of optimal weights for different $\alpha$, and surface and contour plots of the optimal weights.

Here we assume each estimator is optimally tuned. This quantity, which is at most unity, captures the loss of efficiency due to the distributed setting. An ARE close to 1 is "good", while an ARE close to 0 is "bad". From the results above, it follows that the ARE has the form

$$\text{ARE} = \frac{\mathcal{M}_1}{\mathcal{M}_k} = \frac{\phi(\gamma)}{\alpha^2}\left[1 + \sum_{i=1}^{k}\left(\frac{\alpha^2}{\phi(\gamma_i)} - 1\right)\right], \quad (22)$$

where $\phi(\gamma) := \gamma m_\gamma(-\gamma/\alpha^2)$ equals the optimally tuned global risk $\mathcal{M}_1$ up to a factor $\sigma^2$.

We have the following properties of the ARE.

**Theorem 5** (Properties of the asymptotic relative efficiency (ARE))**.** *The asymptotic relative efficiency (ARE) has the following properties:*

1. **Worst case is equally distributed data***: For fixed $k, \alpha^2$ and $\gamma$, the* ARE *attains its minimum when the samples are* equally distributed *across $k$ machines, i.e. $\gamma_1 = \gamma_2 = \cdots = \gamma_k = k\gamma$. We denote the minimal value by $\psi(k, \gamma, \alpha^2)$. That is*

$$\min_{\gamma_i} \text{ARE} = \psi(k, \gamma, \alpha^2) = \frac{\phi(\gamma)}{\alpha^2}\left(1 - k + \frac{k\alpha^2}{\phi(k\gamma)}\right). \quad (23)$$

2. **Adding more machines leads to efficiency loss***: For fixed $\alpha^2$ and $\gamma$, $\psi(k, \gamma, \alpha^2)$ is a decreasing function on $k \in [1, \infty)$ with $\lim_{k\to 1_+} \psi(k, \gamma, \alpha^2) = 1$ and* infinite-worker limit

$$\lim_{k\to\infty} \psi(k, \gamma, \alpha^2) = h(\alpha^2, \gamma) < 1.$$

*Here we evaluate $\psi$ as a continuous function of $k$ for convenience, although originally it is only well-defined for $k \in \mathbb{N}$.*

3. **Form of the infinite-worker limit***: As a function of $\alpha^2$ and $\gamma$, $h(\alpha^2, \gamma)$ has the explicit form*

$$h(\alpha^2, \gamma) = \frac{\Delta + \sqrt{\Delta^2 + 4\gamma^2/\alpha^2}}{2\gamma}\left(1 + \frac{\alpha^2}{\gamma(1+\alpha^2)}\right), \quad (24)$$

*where $\Delta := -\gamma/\alpha^2 + \gamma - 1$.*

See the supplementary material for the proof. See Figure 1 for the surface and contour plots of $h(\alpha^2, \gamma)$. The efficiency loss tends to be larger (ARE is smaller) when the signal-to-noise ratio $\alpha^2$ is larger. The plots confirm the theoretical result that the efficiency always decreases with the number of machines. Relatively speaking, the distributed problem becomes "easier" as the dimension increases, compared to the aggregated problem (i.e., the ARE increases in $\gamma$ for fixed parameters). This can be viewed as a blessing of dimensionality.

We also observe a nontrivial *infinite-worker limit*. Even in the limit of many machines, distributed ridge *does not lose all efficiency*. This is one of the few results in the distributed learning literature where one-step weighting gives nontrivial results for *arbitrary large $k$*.

Overall, the ARE is generally large, *except* when $\gamma$ is small and $\alpha$ is large. This is a setting with strong signal and relatively low dimension, which is also the "easiest" setting from a statistical point of view. In this case, perhaps we should use other techniques for distributed estimation, such as iterative methods.

Next, we study properties of the optimal weights. This is important, because choosing them is crucial in practice. The literature on distributed regression typically considers simple averages of local estimators, for which $\widehat{\beta}_{\text{dist}} = k^{-1}\sum_{i=1}^{k}\widehat{\beta}_i$ (see, e.g. (Zhang et al., 2015; Lee et al., 2017; Battey et al., 2018)). In contrast, we will find that the optimal weights *do not sum up to unity*.

Formally, we have the following properties of the optimal weights.

**Theorem 6** (Properties of the optimal weights)**.** *The asymptotically optimal weights $\mathcal{W}_k^* = (\mathcal{A} + \mathcal{R})^{-1}V$ have the following properties:*

1. **Form of the optimal weights***: The $i$-th coordinate of $\mathcal{W}_k$ is:*

$$\mathcal{W}_{k,i} = \left(\frac{\frac{\alpha^2}{\gamma_i m_{\gamma_i}(-\gamma_i/\alpha^2)}}{1 + \sum_{i=1}^{k}\left[\frac{\alpha^2}{\gamma_i m_{\gamma_i}(-\gamma_i/\alpha^2)} - 1\right]}\right),$$

*and the sum of the limiting weights is always greater*

*than or equal to one:* $\sum_{i=1}^{k} \mathcal{W}_{k,i} \geq 1$. *When* $k \geq 2$, *the sum is strictly greater than one:* $\sum_{i=1}^{k} \mathcal{W}_{k,i} > 1$.

2. **Evenly distributed optimal weights***: When the samples are evenly distributed, so that all limiting aspect ratios* $\gamma_i$ *are equal,* $\gamma_i = k\gamma$, *then all* $\mathcal{W}_{k,i}$ *equal the* optimal weight function $\mathcal{W}(k, \gamma, \alpha^2)$, *which has the form*

$$\mathcal{W}(k, \gamma, \alpha^2) = \frac{\alpha^2}{\alpha^2 k + (1-k)k\gamma \cdot m_{k\gamma}(-k\gamma/\alpha^2)}.$$

3. **Limiting cases***: For fixed* $k$ *and* $\alpha^2$, *the optimal weight function* $\mathcal{W}(k, \gamma, \alpha^2)$ *is an increasing function of* $\gamma \in [0, \infty)$ *with* $\lim_{\gamma \to 0_+} \mathcal{W}(\gamma) = 1/k$ *and* $\lim_{\gamma \to \infty} \mathcal{W}(\gamma) = 1$.

See the supplementary material for the proof. See Figure 2 for some plots of the optimal weight function with $k = 2$. The optimal weights are usually large, and always greater than $1/k$. In the low dimensional setting where $\gamma \to 0$, the weights tend to the uniform average $1/k$. Hence in this setting we recover the classical uniform averaging methods, which makes sense, because ridge regression with optimal regularization parameter tends to linear regression in this regime.

Why are the weights large, and why do they sum to a quantity greater than one? The short answer is that ridge regression is *negatively* (or *downward*) biased, and so we must *counter the effect of bias by upweighting*. We provide a slightly more detailed intuitive explanation in the supplement.

### 3.4. Implications and Practical Relevance

We discuss some of the implications of our results. Our finite-sample results show that the optimal way to weight the estimators depends on functionals of the unknown parameter $\beta$, while the asymptotic results in general depend on the eigenvalues of $\widehat{\Sigma}$ (or $\Sigma$). These are unavailable in practice, and hence these results can typically not be used on real datasets. However, since our results are precise (they capture the *truth* about the problem), we view this as saying that *the problem is hard*. Thus, optimal weighting for ridge regression is a challenging statistical problem. In practice that means that we may be content with uniform weighting. It remains to be investigated in future work how much we should up-adjust those equal weights.

The optimal weights become usable only when $\Sigma = I$. In practice, we can get closer to this assumption by using some form of *whitening* on the data, for instance by scaling all variables to the same scale, by estimating $\Sigma$ over restricted classes, such as assuming block-covariance structures. Alternatively, we can use correlation screening, where we remove features with high correlation. At this stage, all these approaches are heuristic, but we include them to explain how our results can be relevant in practice. It is a topic of future research to make these ideas more concrete.

**Algorithm 1:** Optimally weighted distributed ridge regression

**Input :** Data matrices $X_i$ ($n_i \times p$) and outcomes $Y_i$ ($n_i \times 1$), distributed across $k$ sites

**Output :** Distributed ridge estimator $\widehat{\beta}_{dist}$ of regression coefficients $\beta$

**for** $i \leftarrow 1$ **to** $k$ *(in parallel)* **do**
- Compute the MLE $\widehat{\theta}_i = (\widehat{\sigma}_i^2, \hat{\alpha}_i^2)$ locally on $i$-th machine;
- Set local aspect ratio $\gamma_i = p/n_i$;
- Set regularization parameter $\lambda_i = \gamma_i / \hat{\alpha}_i^2$;
- Compute the local ridge estimator $\widehat{\beta}_i(\lambda_i) = (X_i^\top X_i + n_i \lambda_i I_p)^{-1} X_i^\top Y_i$;
- Send $\widehat{\theta}_i, \gamma_i$ and $\widehat{\beta}_i$ to the global data center.

**end**

At the data center, combine $\widehat{\theta}_i$ to get a global estimator $\widehat{\theta} = (\widehat{\sigma}^2, \hat{\alpha}^2)$, by $\widehat{\theta} = k^{-1} \sum_{i=1}^{k} \widehat{\theta}_i$;

Use $\phi(\gamma_i) = \gamma_i m_{\gamma_i}(-\gamma_i/\hat{\alpha}^2)$ to compute the optimal weights $\omega$ with $i$-th coordinate

$$\omega_i = \frac{\hat{\alpha}^2}{\phi(\gamma_i) \cdot \left(1 + \sum_{i=1}^{k} \left[\frac{\hat{\alpha}^2}{\phi(\gamma_i)} - 1\right]\right)};$$

Output distributed ridge estimator $\widehat{\beta}_{dist} = \sum_{i=1}^{k} \omega_i \widehat{\beta}_i$.

### 3.5. An Algorithm for Distributed Ridge Regression

For identity covariance, our results lead to a very concrete algorithm for optimally weighted distributed ridge regression. In order to develop practical methods, it is crucial to estimate the optimal weights, for which we only need to estimate the signal-to-noise ratio $\alpha^2$ and the noise level $\sigma^2$. Estimating these two parameters is a well-known problem, and several approaches have been proposed, for instance restricted maximum likelihood (REML) estimators (Jiang, 1996; Searle et al., 2009; Dicker, 2014; Dicker & Erdogdu, 2016; Jiang et al., 2016), etc. We can use—for instance—results from (Dicker & Erdogdu, 2017), who showed that the Gaussian MLE is consistent and asymptotically efficient for $\theta = (\sigma^2, \alpha^2)$ even in the non-Gaussian setting of this paper.

Recall that we have $n$ samples distributed across $k$ machines. On the $i$-th machine, we compute a local ridge estimator $\widehat{\beta}_i$, local estimators $\widehat{\sigma}_i^2$, $\hat{\alpha}_i^2$ of the signal-to-noise ratio and the noise level, and quantities needed to find the optimal weights. Then, we send them to a global machine or data

center, and aggregate them to compute a weighted ridge estimator. See Algorithm 1. This algorithm is communication efficient as the local machines only need to send the local ridge estimator $\widehat{\beta}_i$ and some scalars to the global datacenter.

We assume the data are already mean-centered, which can be performed exactly in one additional round of communication, or approximately by centering the individual datasets. Our algorithm is designed for the scenario when the population covariance matrix of the design $X$ is close to identity, meaning the features are nearly uncorrelated. In some cases, we can broaden the applicability of the algorithm by using techniques like whitening or variable pruning. For example, if we have a good estimator $\widehat{\Sigma}$ of the population covariance matrix $\Sigma$, then we can transform the local design matrix $X_i$ to $\tilde{X}_i$ by whitening: $\tilde{X}_i = X_i \widehat{\Sigma}^{-1/2}$.

## 4. Experiments on Empirical Data

In this section, we present an empirical data example to examine the accuracy of our theoretical results. It is reasonable to compare the performance of different estimators in terms of the prediction (test) error. Figure 3 shows a comparison of three estimators including our optimal weighted estimator on the Million Song Year Prediction Dataset (MSD) (Bertin-Mahieux et al., 2011).

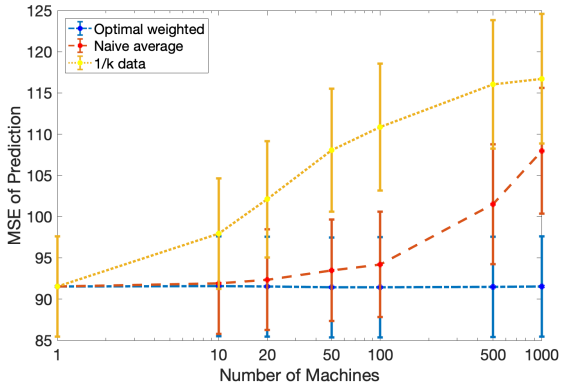



*Figure 3.* Million Song Year Prediction Dataset (MSD).

Specifically, we perform the following steps in our data analysis. We download the dataset from the UC Irvine Machine Learning Repository. The original dataset has $N = 515,345$ (the first $463,715$ for training, the rest for testing), and $p = 91$ features. We attempt to predict the release year of a song. Before doing distributed regression, we first center both the design matrix $X$ and the outcome $Y$. Then we whiten the design matrix by transforming $X$ to $\tilde{X} = X(X^\top X/n)^{-1/2}$. One may also consider whitening the design matrices locally, but that would not correspond to fitting the same regression model on each machine.

For each experiment, we randomly choose $n_{train} = 10,000$ samples from the training set and $n_{test} = 1,000$ samples from the test set. We construct the estimators on the training samples. Then we perform ridge regression in a distributed way to obtain our optimal weighted estimator as described in Algorithm 1. We measure its performance on the test data by computing its MSE for prediction. We choose the number of machines to be $k = 1, 10, 20, 50, 100, 500, 1,000$, and we distribute the data evenly across the $k$ machines.

For comparison, we also consider two other estimators:

1. The distributed estimator where we take the naive average (weight for each local estimator is simply $1/k$) and choose the local tuning parameter $\lambda_i = p/(n_{train} \cdot \hat{\alpha}^2)$. This formally agrees with the divide-and-conquer type estimator proposed in (Zhang et al., 2015).
2. The estimator using only a fraction $1/k$ of the data, which is just one of the local estimators. For this estimator, we choose the tuning parameter $\lambda = kp/(n_{train} \cdot \hat{\alpha}^2)$.

We repeat the experiment $T = 100$ times, and report the average and one standard deviation over all experiments on Figure 3. Each time we randomly collect new training and test sets.

From Figure 3, we observe the following:

1. The optimal weighted estimator has smaller MSE than the local estimator, which means weighting can indeed help. The variance is also reduced by weighting.
2. The performance of the two distributed estimators is very close. However, our optimal weighted estimator is more accurate when the number of machines is large.
3. Data splitting has little impact on the performance of the optimal weighted estimator. This phenomenon is compatible with our theory. Since the signal-to-noise ratio $\alpha^2$ is about $0.3$ for this data set, we are in a low SNR scenario. In the supplementary material, we provide formulas and plots of the relative efficiency for prediction. These show that the performance of the distributed estimator should be very close to the global estimator in this case.

To conclude, in terms of computation-statistics tradeoff, this example suggests a very positive outlook on using distributed ridge regression: The accuracy is affected very little even though the data is split up into 100 parts. Thus we save at least 100x in computation time, while we have nearly no loss in performance.

## Acknowledgements

The authors would like to thank the reviewers for their helpful comments. The authors thank Yuekai Sun for discussions motivating our study, as well as John Duchi, Jason D. Lee,

Xinran Li, Jonathan Rosenblatt, Feng Ruan, and Linjun Zhang for helpful discussions. They are grateful to Sifan Liu for thorough comments on an earlier version of the manuscript. They are also grateful to the associate editor and referees for valuable suggestions. ED was partially supported by NSF BIGDATA grant IIS 1837992.

## References

Anderson, T. W. *An Introduction to Multivariate Statistical Analysis*. Wiley New York, 2003.

Bai, Z. and Silverstein, J. W. *Spectral analysis of large dimensional random matrices*. Springer Series in Statistics. Springer, 2009.

Banerjee, M., Durot, C., and Sen, B. Divide and conquer in non-standard problems and the super-efficiency phenomenon. *arXiv preprint arXiv:1605.04446*, 2016.

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. Distributed testing and estimation under sparse high dimensional models. *The Annals of Statistics*, 46(3):1352–1382, 2018.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

Braverman, M., Garg, A., Ma, T., Nguyen, H. L., and Woodruff, D. P. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 1011–1020. ACM, 2016.

Chen, X. and Xie, M.-g. A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, pp. 1655–1684, 2014.

Chen, X., Liu, W., and Zhang, Y. Quantile regression under memory constraint. *arXiv preprint arxiv:1810.08264*, 2018a.

Chen, X., Liu, W., and Zhang, Y. First-order newton-type estimator for distributed estimation and inference. *arXiv preprint arxiv:1811.11368*, 2018b.

Dicker, L. and Erdogdu, M. Flexible results for quadratic forms with applications to variance components estimation. *The Annals of Statistics*, 45(1):386–414, 2017.

Dicker, L. H. Variance estimation in high-dimensional linear models. *Biometrika*, 101(2):269–284, 2014.

Dicker, L. H. and Erdogdu, M. A. Maximum likelihood for variance estimation in high-dimensional linear models. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 159–167. PMLR, 2016.

Dobriban, E. and Wager, S. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279, 2018.

Duan, J., Qiao, X., and Cheng, G. Distributed nearest neighbor classification. *arXiv preprint arXiv:1812.05005*, 2018.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Zhang, Y. Optimality guarantees for distributed statistical estimation. *arXiv preprint arXiv:1405.0782*, 2014.

Fan, J., Wang, D., Wang, K., and Zhu, Z. Distributed estimation of principal eigenspaces. *arXiv preprint arXiv:1702.06488*, 2017.

Guo, Z.-C., Shi, L., and Wu, Q. Learning theory of distributed regression with bias corrected regularization kernel network. *The Journal of Machine Learning Research*, 18(1):4237–4261, 2017.

Huo, X. and Cao, S. Aggregated inference. *Wiley Interdisciplinary Reviews: Computational Statistics*, pp. e1451, 2018.

Jiang, J. Reml estimation: asymptotic behavior and related topics. *The Annals of Statistics*, 24(1):255–286, 1996.

Jiang, J., Li, C., Paul, D., Yang, C., and Zhao, H. On high-dimensional misspecified mixed model analysis in genome-wide association study. *The Annals of Statistics*, 44(5):2127–2160, 2016.

Jordan, M. I., Lee, J. D., and Yang, Y. Communication-efficient distributed statistical inference. *arXiv preprint arXiv:1605.07689*, 2016.

Lee, J. D., Liu, Q., Sun, Y., and Taylor, J. E. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30, 2017.

Li, R., Lin, D. K., and Li, B. Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*, 29(5):399–409, 2013.

Lin, S.-B., Guo, X., and Zhou, D.-X. Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Liu, M., Shang, Z., and Cheng, G. How many machines can we use in parallel computing for kernel ridge regression? *arXiv preprint arXiv:1805.09948*, 2018.

Mackey, L. W., Jordan, M. I., and Talwalkar, A. Divide-and-conquer matrix factorization. In *Advances in neural information processing systems*, pp. 1134–1142, 2011.

Marchenko, V. A. and Pastur, L. A. Distribution of eigenvalues for some sets of random matrices. *Mat. Sb.*, 114 (4):507–536, 1967.

Mcdonald, R., Mohri, M., Silberman, N., Walker, D., and Mann, G. S. Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems*, pp. 1231–1239, 2009.

Paul, D. and Aue, A. Random matrix theory in statistics: A review. *Journal of Statistical Planning and Inference*, 150:1–29, 2014.

Richards, D., Rebeschini, P., and Rosasco, L. Decentralised learning with random features and distributed gradient descent. *arXiv preprint arXiv:2007.00360*, 2020.

Rosenblatt, J. D. and Nadler, B. On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4):379–404, 2016.

Searle, S. R., Casella, G., and McCulloch, C. E. *Variance components*, volume 391. John Wiley & Sons, 2009.

Shang, Z. and Cheng, G. Computational limits of a distributed algorithm for smoothing spline. *The Journal of Machine Learning Research*, 18(1):3809–3845, 2017.

Shi, C., Lu, W., and Song, R. A massive data framework for m-estimators with cubic-rate. *Journal of the American Statistical Association*, pp. 1–12, 2018.

Smith, V., Forte, S., Ma, C., Takác, M., Jordan, M. I., and Jaggi, M. Cocoa: A general framework for communication-efficient distributed optimization. *arXiv preprint arXiv:1611.02189*, 2016.

Smola, A. Course notes on scalable machine learning, 2012.

Volgushev, S., Chao, S.-K., and Cheng, G. Distributed inference for quantile regression processes. *arXiv preprint arXiv:1701.06088*, 2017.

Wang, X., Yang, Z., Chen, X., and Liu, W. Distributed inference for linear support vector machine. *arXiv preprint arxiv:1811.11922*, 2018.

Xu, G., Shang, Z., and Cheng, G. Optimal tuning for divide-and-conquer kernel ridge regression with massive data. *arXiv preprint arXiv:1612.05907*, 2016.

Yao, J., Bai, Z., and Zheng, S. *Large Sample Covariance Matrices and High-Dimensional Data Analysis*. Cambridge University Press, 2015.

Zhang, Y., Wainwright, M. J., and Duchi, J. C. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pp. 1502–1510, 2012.

Zhang, Y., Duchi, J., and Wainwright, M. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pp. 592–617, 2013a.

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

Zhang, Y., Duchi, J., and Wainwright, M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Zhao, T., Cheng, G., and Liu, H. A partially linear framework for massive heterogeneous data. *Annals of statistics*, 44(4):1400, 2016.

Zhu, Y. and Lafferty, J. Distributed nonparametric regression under communication constraints. *arXiv preprint arXiv:1803.01302*, 2018.