Distributed Learning

Nicole Mücke (nicole.muecke@tu-braunschweig.de)

Leibniz Summer School 2021

Table of contents

Introduction: DL

What is DL ?

- large size of training datasets generally offers improvement in model performance, however the training process becomes computationally expensive and time consuming

  Example: training a state-of-the-art ResNet-50 model (in 90 epochs) on the ImageNet dataset with a Nvidia Tesla V100 GPU requires about two days [WWS+20]

- distributed learning (DL) is a very common strategy to reduce the overall training time by exploiting multiple computing devices
  $\longrightarrow$ datasets are partitioned over machines, which compute locally, and communicate short messages
- communication often the bottleneck
- here: focus on communication efficient methods

Distributed OLS in Linear Models - Underparameterized Regime

Random-design linear regression

- random pair $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ with unknown joint distribution P
- noise $\epsilon \in \mathbb{R}^d$

Random-design linear regression

- random pair $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ with unknown joint distribution P
- noise $\epsilon \in \mathbb{R}^d$
- **aim:** predict label y given a new input $x \in \mathbb{R}^d$
- minimize prediction risk

$$\min_f \mathcal{R}(f) \ , \quad \mathcal{R}(f) := \mathbb{E}[(f(X) - Y)^2]$$

Random-design linear regression

- random pair $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ with unknown joint distribution P
- noise $\epsilon \in \mathbb{R}^d$
- **aim:** predict label y given a new input $x \in \mathbb{R}^d$
- minimize prediction risk

$$\min_f \mathcal{R}(f) \ , \quad \mathcal{R}(f) := \mathbb{E}[(f(X) - Y)^2]$$

note: the optimal predictor f^* among all (measurable) functions is the regression function

$$f^*(x) = \mathbb{E}[Y|X = x]$$

Random-design linear regression

- random pair $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ with unknown joint distribution P
- noise $\epsilon \in \mathbb{R}^d$
- **aim:** predict label y given a new input $x \in \mathbb{R}^d$
- minimize prediction risk

$$\min_f \mathcal{R}(f) \ , \quad \mathcal{R}(f) := \mathbb{E}[(f(X) - Y)^2]$$

note: the optimal predictor f^* among all (measurable) functions is the regression function

$$f^*(x) = \mathbb{E}[Y|X = x]$$

Assumption:

We assume $\mathbb{E}[||X||^2] < \infty$, $\mathbb{E}[Y^2] < \infty$. Moreover, our model is well-specified, i.e. $f^*(x) = \langle \beta^*, x \rangle$ for some $\beta^* \in \mathbb{R}^d$.

Model: $Y = \langle \beta^*, X \rangle + \epsilon$

Random-design linear regression

- Optimal β^* cannot be found directly! How can β^* be approximated given the data

$$D := \{(x_1, y_1), ..., (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R} \text{ ?}$$

Random-design linear regression

- Optimal β^* cannot be found directly! How can β^* be approximated given the data

$$D := \{(x_1, y_1), ..., (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R} \text{ ?}$$

- empirical risk minimization:

$$\min_{\beta \in \mathbb{R}^d} \widehat{\mathcal{R}}(\beta) \,, \quad \widehat{\mathcal{R}}(\beta) := \frac{1}{n} \sum_{j=1}^{n} (\langle \beta, x_j \rangle - y_j)^2$$
$$= \frac{1}{n} ||\mathrm{X}\beta - \mathrm{Y}||^2$$

with data matrix $\mathrm{X} \in \mathbb{R}^{n \times d}$, response vector $\mathrm{Y} \in \mathbb{R}^n$

for now: $n > d$

Random-design linear regression

- Optimal β^* cannot be found directly! How can β^* be approximated given the data

$$D := \{(x_1, y_1), ..., (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R} \text{ ?}$$

- empirical risk minimization:

$$\min_{\beta \in \mathbb{R}^d} \widehat{\mathcal{R}}(\beta) \,, \quad \widehat{\mathcal{R}}(\beta) := \frac{1}{n} \sum_{j=1}^{n} (\langle \beta, x_j \rangle - y_j)^2$$
$$= \frac{1}{n} ||\mathrm{X}\beta - \mathrm{Y}||^2$$

with data matrix $\mathrm{X} \in \mathbb{R}^{n \times d}$, response vector $\mathrm{Y} \in \mathbb{R}^n$

for now: $n > d$

Definition:

An ordinary least-squares estimator (OLSE) of β^* is defined to be any $\hat{\beta} \in \mathbb{R}^d$ such that

$$||X\hat{\beta} - Y||^2 = \min_{\beta \in \mathbb{R}^d} ||X\hat{\beta} - Y||^2 .$$

Definition:

An ordinary least-squares estimator (OLSE) of β^* is defined to be any $\hat{\beta} \in \mathbb{R}^d$ such that

$$||X\hat{\beta} - Y||^2 = \min_{\beta \in \mathbb{R}^d} ||X\hat{\beta} - Y||^2 .$$

- differentiating $||X\hat{\beta} - Y||$ w.r.t. $\hat{\beta}$ shows that any solution of the normal equation

$$X^T X\hat{\beta} = X^T Y$$

  is an OLSE of β^* (see [EHN96])

Definition:

An ordinary least-squares estimator (OLSE) of β^* is defined to be any $\hat{\beta} \in \mathbb{R}^d$ such that

$$||X\hat{\beta} - Y||^2 = \min_{\beta \in \mathbb{R}^d} ||X\hat{\beta} - Y||^2 .$$

- differentiating $||X\hat{\beta} - Y||$ w.r.t. $\hat{\beta}$ shows that any solution of the normal equation

$$X^T X\hat{\beta} = X^T Y$$

  is an OLSE of β^* (see [EHN96])
- if $rank(X) = d$ (i.e. full rank) then $(X^T X)^{-1}$ exists and the unique OLSE is

$$\hat{\beta}_{\text{OLS}} = (X^T X)^{-1} XY$$

- if X is not of full rank, then there are infinitely many OLSE's, the one with minimal norm is

$$\hat{\beta}_{\text{OLS}} = (X^T X)^{\dagger} XY$$

Definition:

An ordinary least-squares estimator (OLSE) of β^* is defined to be any $\hat{\beta} \in \mathbb{R}^d$ such that

$$||X\hat{\beta} - Y||^2 = \min_{\beta \in \mathbb{R}^d} ||X\hat{\beta} - Y||^2 \ .$$

- differentiating $||X\hat{\beta} - Y||$ w.r.t. $\hat{\beta}$ shows that any solution of the normal equation

$$X^T X\hat{\beta} = X^T Y$$

  is an OLSE of β^* (see [EHN96])
- if $rank(X) = d$ (i.e. full rank) then $(X^T X)^{-1}$ exists and the unique OLSE is

$$\hat{\beta}_{\text{OLS}} = (X^T X)^{-1} XY$$

- calculating an OLSE involves matrix inversion that scales as $\mathcal{O}(n^3)$ in time and memory!

Statistical Properties of OLSE

Assumptions:

1. The covariance operator $\Sigma = \mathbb{E}[XX^T] \in \mathbb{R}^{d \times d}$ is invertible.

Statistical Properties of OLSE

Assumptions:

1. The covariance operator $\Sigma = \mathbb{E}[XX^T] \in \mathbb{R}^{d \times d}$ is invertible.
2. Noise model: $\mathbb{E}[\varepsilon | X] = 0$ and $\mathbb{E}[\varepsilon^2 | X] \leq \sigma^2$.

Statistical Properties of OLSE

Assumptions:

1. The covariance operator $\Sigma = \mathbb{E}[XX^T] \in \mathbb{R}^{d \times d}$ is invertible.
2. Noise model: $\mathbb{E}[\varepsilon | X] = 0$ and $\mathbb{E}[\varepsilon^2 | X] \leq \sigma^2$.

Measure of Performance:

Definition and Lemma:

Let $\hat{\beta} \in \mathbb{R}^d$. The excess risk is given by

$$\begin{aligned} \mathcal{R}(\hat{\beta}) - \mathcal{R}(\beta^*) &= \mathbb{E}[||\Sigma^{\frac{1}{2}}(\hat{\beta} - \beta^*)||^2] \\ &= \text{Bias}(\hat{\beta}) + \text{Var}(\hat{\beta}) \ , \end{aligned}$$

Statistical Properties of OLSE

Assumptions:

1. The covariance operator $\Sigma = \mathbb{E}[XX^T] \in \mathbb{R}^{d\times d}$ is invertible.
2. Noise model: $\mathbb{E}[\varepsilon|X] = 0$ and $\mathbb{E}[\varepsilon^2|X] \leq \sigma^2$.

Measure of Performance:

Definition and Lemma:

Let $\hat{\beta} \in \mathbb{R}^d$. The excess risk is given by

$$\begin{aligned}\mathcal{R}(\hat{\beta}) - \mathcal{R}(\beta^*) &= \mathbb{E}[||\Sigma^{\frac{1}{2}}(\hat{\beta} - \beta^*)||^2] \\ &= \text{Bias}(\hat{\beta}) + \text{Var}(\hat{\beta}) \ ,\end{aligned}$$

where

$$\text{Bias}(\hat{\beta}) := \mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\hat{\beta}] - \beta^*)||^2\right] \ , \quad \text{Var}(\hat{\beta}) := \mathbb{E}\left[||\Sigma^{1/2}(\hat{\beta} - \mathbb{E}[\hat{\beta}])||^2\right] \ .$$

Statistical Properties of OLSE

Theorem:

1. $\text{Bias}(\hat{\beta}_{\text{OLS}}) = 0$, $\text{Var}(\hat{\beta}_{\text{OLS}}) = \sigma^2 \mathbb{E}[Tr[\Sigma(\mathrm{X}^T\mathrm{X})^{-1}]]$

Statistical Properties of OLSE

Theorem:

1. $\text{Bias}(\hat{\beta}_{\text{OLS}}) = 0$, $\text{Var}(\hat{\beta}_{\text{OLS}}) = \sigma^2 \mathbb{E}[Tr[\Sigma(\mathrm{X}^T\mathrm{X})^{-1}]]$
2. If $X \sim \mathcal{N}(0, \Sigma)$, then

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\hat{\beta}_{\text{OLS}} - \beta^*)||^2] = \frac{\sigma^2 d}{n - d - 1} \ .$$

Statistical Properties of OLSE

Theorem:

1. $\text{Bias}(\hat{\beta}_{\text{OLS}}) = 0$, $\text{Var}(\hat{\beta}_{\text{OLS}}) = \sigma^2 \mathbb{E}[Tr[\Sigma(X^T X)^{-1}]]$
2. If $X \sim \mathcal{N}(0, \Sigma)$, then
$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\hat{\beta}_{\text{OLS}} - \beta^*)||^2] = \frac{\sigma^2 d}{n - d - 1} \, .$$
3. If $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, then for some $C < \infty$
$$\frac{\sigma^2 d}{n - d + 1} \leq \mathbb{E}[||\Sigma^{\frac{1}{2}}(\hat{\beta}_{\text{OLS}} - \beta^*)||^2] \leq C \, \frac{\sigma^2 d}{n} \, ,$$
under a classical small ball assumption.

see e.g. [Mou19], [Sha06], [BF83]

Distributed OLS

split the data D evenly across local nodes $m = 1, ..., M$

- local data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$, local output vector $\mathrm{Y}_m \in \mathbb{R}^{\frac{n}{M}}$

Distributed OLS

split the data D evenly across local nodes $m = 1, ..., M$

- local data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$, local output vector $\mathrm{Y}_m \in \mathbb{R}^{\frac{n}{M}}$
- local OLSE: $\hat{\beta}_m = (\mathrm{X}_m^T \mathrm{X}_m)^{-1} \mathrm{X}_m \mathrm{Y}_m$

Distributed OLS

split the data D evenly across local nodes $m = 1, ..., M$

- local data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$, local output vector $\mathrm{Y}_m \in \mathbb{R}^{\frac{n}{M}}$
- local OLSE: $\hat{\beta}_m = (\mathrm{X}_m^T \mathrm{X}_m)^{-1} \mathrm{X}_m \mathrm{Y}_m$
- global average:

$$\bar{\beta}_M := \frac{1}{M} \sum_{m=1}^{M} \hat{\beta}_m$$

Distributed OLS

split the data D evenly across local nodes $m = 1, ..., M$

- local data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$, local output vector $\mathrm{Y}_m \in \mathbb{R}^{\frac{n}{M}}$
- local OLSE: $\hat{\beta}_m = (\mathrm{X}_m^T \mathrm{X}_m)^{-1} \mathrm{X}_m \mathrm{Y}_m$
- global average:

$$\bar{\beta}_M := \frac{1}{M} \sum_{m=1}^{M} \hat{\beta}_m$$

note: $\hat{\beta}_m$ is calculated in parallel; time complexity is now reduced to $\mathcal{O}\left(\left(\frac{n}{M}\right)^3\right)$ and memory $\mathcal{O}\left(\left(\frac{n}{M}\right)^2\right)$

Distributed OLS

split the data D evenly across local nodes $m = 1, ..., M$

- local data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$, local output vector $\mathrm{Y}_m \in \mathbb{R}^{\frac{n}{M}}$
- local OLSE: $\hat{\beta}_m = (\mathrm{X}_m^T \mathrm{X}_m)^{-1} \mathrm{X}_m \mathrm{Y}_m$
- global average:

$$\bar{\beta}_M := \frac{1}{M} \sum_{m=1}^{M} \hat{\beta}_m$$

note: $\hat{\beta}_m$ is calculated in parallel; time complexity is now reduced to $\mathcal{O}\left(\left(\frac{n}{M}\right)^3\right)$ and memory $\mathcal{O}\left(\left(\frac{n}{M}\right)^2\right)$

clear: the larger M, the more is complexity reduced

Distributed OLS

split the data D evenly across local nodes $m = 1, ..., M$

- local data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$, local output vector $\mathrm{Y}_m \in \mathbb{R}^{\frac{n}{M}}$
- local OLSE: $\hat{\beta}_m = (\mathrm{X}_m^T \mathrm{X}_m)^{-1} \mathrm{X}_m \mathrm{Y}_m$
- global average:

$$\bar{\beta}_M := \frac{1}{M} \sum_{m=1}^{M} \hat{\beta}_m$$

note: $\hat{\beta}_m$ is calculated in parallel; time complexity is now reduced to $\mathcal{O}\left(\left(\frac{n}{M}\right)^3\right)$ and memory $\mathcal{O}\left(\left(\frac{n}{M}\right)^2\right)$

clear: the larger M, the more is complexity reduced

Question: What is the (statistical) performance of $\bar{\beta}_M$ compared to the single machine approach?

Definition:

The relative prediction efficiency $\mathrm{Eff}(M)$ of $\bar{\beta}_M$ is defined as

$$\mathrm{Eff}(M) := \frac{\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_1 - \beta^*)||^2]}{\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2]} \ .$$

Definition:

The relative prediction efficiency $\mathrm{Eff}(M)$ of $\bar{\beta}_M$ is defined as

$$\mathrm{Eff}(M) := \frac{\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_1 - \beta^*)||^2]}{\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2]} \ .$$

note: $\mathrm{Eff}(M) \geq 1$ is good!

Bis-Variance Decomposition:

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2] = \text{Bias}(\bar{\beta}_M) + \text{Var}(\bar{\beta}_M)$$

Bis-Variance Decomposition:

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2] = \text{Bias}(\bar{\beta}_M) + \text{Var}(\bar{\beta}_M)$$

Bias:

$$\begin{aligned}\text{Bias}(\bar{\beta}_M) &= \mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\bar{\beta}_M] - \beta^*)||^2\right] \\ &= \mathbb{E}\left[\left|\left|\frac{1}{M}\sum_{m=1}^{M}\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)\right|\right|^2\right]\end{aligned}$$

Bis-Variance Decomposition:

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2] = \text{Bias}(\bar{\beta}_M) + \text{Var}(\bar{\beta}_M)$$

Bias:

$$\begin{aligned}
\text{Bias}(\bar{\beta}_M) &= \mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\bar{\beta}_M] - \beta^*)||^2\right] \\
&= \mathbb{E}\left[\left|\left|\frac{1}{M}\sum_{m=1}^{M}\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)\right|\right|^2\right] \\
&\leq \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)||^2\right]
\end{aligned}$$

Bis-Variance Decomposition:

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2] = \text{Bias}(\bar{\beta}_M) + \text{Var}(\bar{\beta}_M)$$

Bias:

$$\begin{aligned}
\text{Bias}(\bar{\beta}_M) &= \mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\bar{\beta}_M] - \beta^*)||^2\right] \\
&= \mathbb{E}\left[\left|\left|\frac{1}{M}\sum_{m=1}^{M}\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)\right|\right|^2\right] \\
&\leq \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)||^2\right] \\
&= \frac{1}{M}\sum_{m=1}^{M}\underbrace{\text{Bias}(\hat{\beta}_m)}_{=0}
\end{aligned}$$

Bis-Variance Decomposition:

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2] = \text{Bias}(\bar{\beta}_M) + \text{Var}(\bar{\beta}_M)$$

Bias:

$$\begin{aligned}
\text{Bias}(\bar{\beta}_M) &= \mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\bar{\beta}_M] - \beta^*)||^2\right] \\
&= \mathbb{E}\left[\left|\left|\frac{1}{M}\sum_{m=1}^{M}\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)\right|\right|^2\right] \\
&\leq \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[||\Sigma^{1/2}(\mathbb{E}[\hat{\beta}_m] - \beta^*)||^2\right] \\
&= \frac{1}{M}\sum_{m=1}^{M}\underbrace{\text{Bias}(\hat{\beta}_m)}_{=0} \\
&= 0
\end{aligned}$$

Variance:

$$\mathrm{Var}(\bar{\beta}_M) = \mathbb{E}\Big[||\Sigma^{1/2}(\bar{\beta}_M - \mathbb{E}[\bar{\beta}_M])||^2\Big]$$

Variance:

$$\begin{aligned}\mathrm{Var}(\bar{\beta}_M) &= \mathbb{E}\Big[||\Sigma^{1/2}(\bar{\beta}_M - \mathbb{E}[\bar{\beta}_M])||^2\Big] \\ &= \ldots\end{aligned}$$

Variance:

$$
\begin{aligned}
\mathrm{Var}(\bar{\beta}_M) &= \mathbb{E}\Big[||\Sigma^{1/2}(\bar{\beta}_M - \mathbb{E}[\bar{\beta}_M])||^2\Big] \\
&= \ldots \\
&= \frac{\sigma^2}{M^2}\sum_{m=1}^{M}\mathbb{E}\Big[Tr\Big[\Sigma^{1/2}(X_m^T X_m)^{\dagger}\Sigma^{1/2}\Big]\Big]
\end{aligned}
$$

Variance:

$$
\begin{aligned}
\mathrm{Var}(\bar{\beta}_M) &= \mathbb{E}\Big[||\Sigma^{1/2}(\bar{\beta}_M - \mathbb{E}[\bar{\beta}_M])||^2\Big] \\
&= \ldots \\
&= \frac{\sigma^2}{M^2}\sum_{m=1}^{M}\mathbb{E}\Big[Tr\Big[\Sigma^{1/2}(X_m^T X_m)^{\dagger}\Sigma^{1/2}\Big]\Big] \\
&\geq \frac{\sigma^2}{M}\,\frac{d}{\frac{n}{M}+1-d}
\end{aligned}
$$

Linear Loss in Efficiency

Theorem:

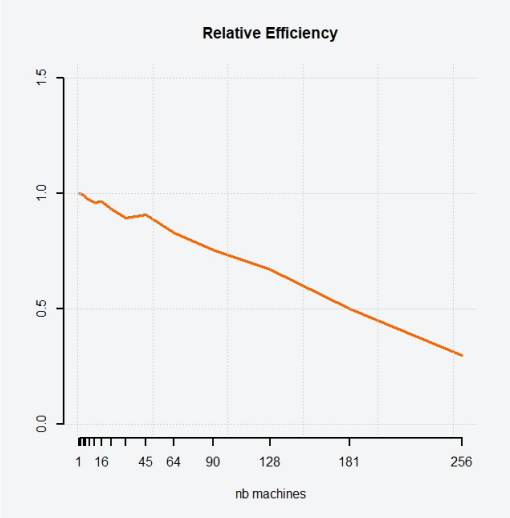
Let $\frac{n}{M} > d$ and $X \sim \mathcal{N}(0, \Sigma)$. The expected excess risk satisfies

$$\mathbb{E}[||\Sigma^{\frac{1}{2}}(\bar{\beta}_M - \beta^*)||^2] = \frac{\sigma^2}{M} \frac{d}{\frac{n}{M} - d - 1} .$$

Hence, $\mathrm{Eff}(M)$ decreases linearly with the number of machines M:

$$\mathrm{Eff}(M) = \frac{n}{n-d-1} - M \frac{d+1}{n-d-1} .$$

see e.g. [DS21], [RN16]



Linear Loss in Efficiency

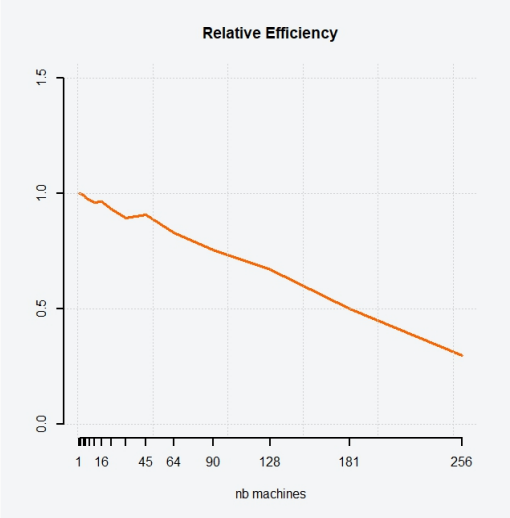
more generally:

Theorem:

Let $\frac{n}{M} > d$ and $\mathbb{V}[Y|X] \geq \sigma^2$. The relative efficiency satisfies

$$\mathrm{Eff}(M) \leq 1 - M\,\frac{d-1}{n}\,.$$

(follows from the lower bound for the distributed variance and the upper bound for the single OLSE)



Summary: OLS in the underparameterized Regime

- distributed OLS is unbiased
- the efficiency is determined by the local variances
- distributed learning reduces variance

Summary: OLS in the underparameterized Regime

- distributed OLS is unbiased
- the efficiency is determined by the local variances
- distributed learning reduces variance
- distributed OLS reduces complexity by $\mathcal{O}(M^3)$ but suffers (at least) a linear loss in efficiency

Distributed OLS in Linear Models - Overparameterized Regime

Local Overparameterized Setting: $\frac{n}{M} < d$

recall: local minimum norm estimator $\hat{\beta}_m$ is given by

$$\hat{\beta}_m = \mathrm{X}_m^T(\mathrm{X}_m\mathrm{X}_m^T)^{-1}\mathrm{Y}_m \ ,$$

Local Overparameterized Setting: $\frac{n}{M} < d$

recall: local minimum norm estimator $\hat{\beta}_m$ is given by

$$\hat{\beta}_m = \mathrm{X}_m^T(\mathrm{X}_m\mathrm{X}_m^T)^{-1}\mathrm{Y}_m \ ,$$

solves

$$\min_{\hat{\beta}\in\mathbb{R}^d} ||\hat{\beta}|| \ , \quad \text{s.th.} \quad ||\mathrm{X}_m\hat{\beta} - \mathrm{Y}_m||^2 = \min_{\beta} ||\mathrm{X}_m\beta - \mathrm{Y}_m||^2$$

Local Overparameterized Setting: $\frac{n}{M} < d$

recall: local minimum norm estimator $\hat{\beta}_m$ is given by

$$\hat{\beta}_m = \mathrm{X}_m^T(\mathrm{X}_m\mathrm{X}_m^T)^{-1}\mathrm{Y}_m \ ,$$

solves

$$\min_{\hat{\beta}\in\mathbb{R}^d} ||\hat{\beta}|| \ , \quad \text{s.th.} \quad ||\mathrm{X}_m\hat{\beta} - \mathrm{Y}_m||^2 = \min_{\beta} ||\mathrm{X}_m\beta - \mathrm{Y}_m||^2$$

and interpolates the data: $\left\langle \hat{\beta}_m, x_m^{(j)} \right\rangle = y_m^{(j)}, \quad (m = 1, ..., M, \ j = 1, ..., n/M)$

Local Overparameterized Setting: $\frac{n}{M} < d$

recall: local minimum norm estimator $\hat{\beta}_m$ is given by

$$\hat{\beta}_m = \mathrm{X}_m^T(\mathrm{X}_m\mathrm{X}_m^T)^{-1}\mathrm{Y}_m \ ,$$

solves

$$\min_{\hat{\beta}\in\mathbb{R}^d} ||\hat{\beta}|| \ , \quad \text{s.th.} \quad ||\mathrm{X}_m\hat{\beta} - \mathrm{Y}_m||^2 = \min_{\beta} ||\mathrm{X}_m\beta - \mathrm{Y}_m||^2$$

and interpolates the data: $\left\langle \hat{\beta}_m, x_m^{(j)} \right\rangle = y_m^{(j)}, \quad (m = 1, ..., M, \ j = 1, ..., n/M)$

note: interpolation means we have local overfitting (traditionally a bad thing)!

Local Overparameterized Setting: $\frac{n}{M} < d$

recall: local minimum norm estimator $\hat{\beta}_m$ is given by

$$\hat{\beta}_m = \mathrm{X}_m^T(\mathrm{X}_m\mathrm{X}_m^T)^{-1}\mathrm{Y}_m \ ,$$

solves

$$\min_{\hat{\beta}\in\mathbb{R}^d} ||\hat{\beta}|| \ , \quad \text{s.th.} \quad ||\mathrm{X}_m\hat{\beta} - \mathrm{Y}_m||^2 = \min_{\beta} ||\mathrm{X}_m\beta - \mathrm{Y}_m||^2$$

and interpolates the data: $\left\langle \hat{\beta}_m, x_m^{(j)} \right\rangle = y_m^{(j)}, \quad (m = 1, ..., M, \ j = 1, ..., n/M)$

note: interpolation means we have local overfitting (traditionally a bad thing)!

Question: What can we say about the efficiency of $\bar{\beta}_M$ in this regime ? Can overfitting be benign or harmless ?

Local Overparameterized Setting: $\frac{n}{M} < d$

recall: local minimum norm estimator $\hat{\beta}_m$ is given by

$$\hat{\beta}_m = \mathrm{X}_m^T(\mathrm{X}_m\mathrm{X}_m^T)^{-1}\mathrm{Y}_m \ ,$$

solves

$$\min_{\hat{\beta}\in\mathbb{R}^d} ||\hat{\beta}|| \ , \quad \text{s.th.} \quad ||\mathrm{X}_m\hat{\beta} - \mathrm{Y}_m||^2 = \min_{\beta} ||\mathrm{X}_m\beta - \mathrm{Y}_m||^2$$

and interpolates the data: $\left\langle \hat{\beta}_m, x_m^{(j)} \right\rangle = y_m^{(j)}, \quad (m = 1, ..., M, \ j = 1, ..., n/M)$

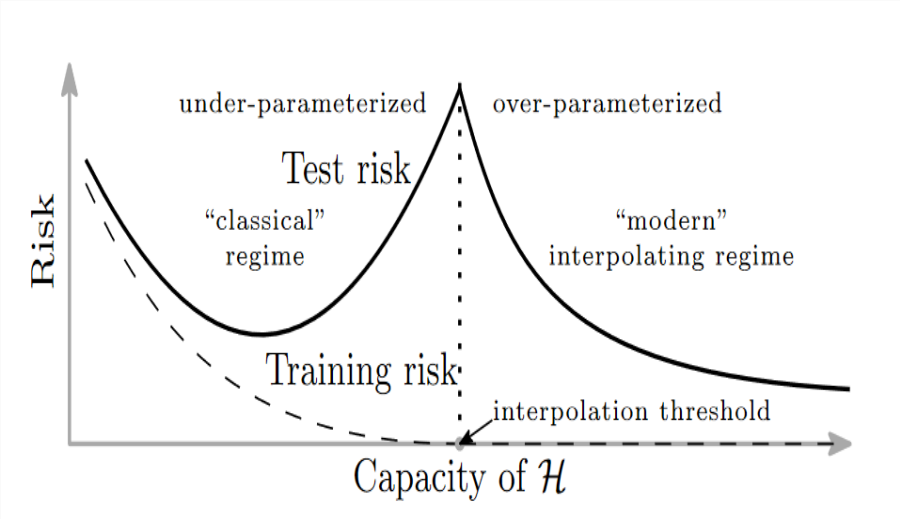
note: interpolation means we have local overfitting (traditionally a bad thing)!

Question: What can we say about the efficiency of $\bar{\beta}_M$ in this regime ? Can overfitting be benign or harmless ?

References: in the single machine setting, this topic has recently attracted lots of attention

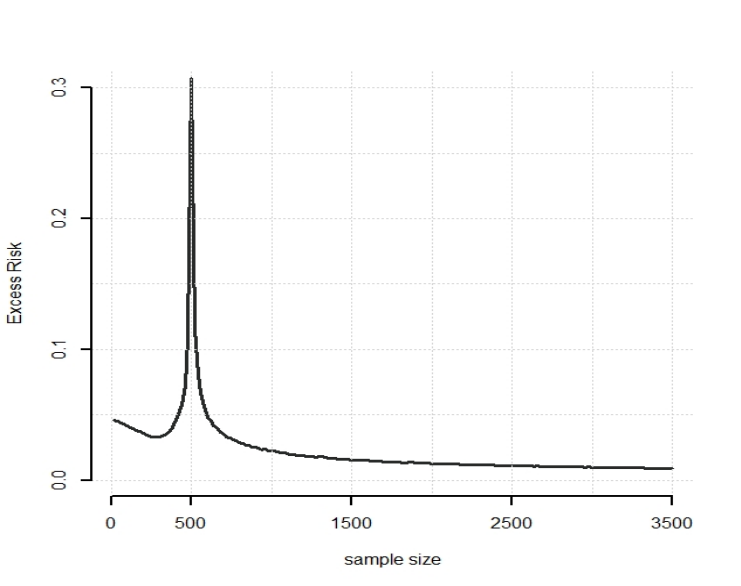
[CL20], [TB20], [BLLT20], [MVSS20], [KLS20], [RMR20], [LR+20], ...

Short Detour: Double Descent in ML



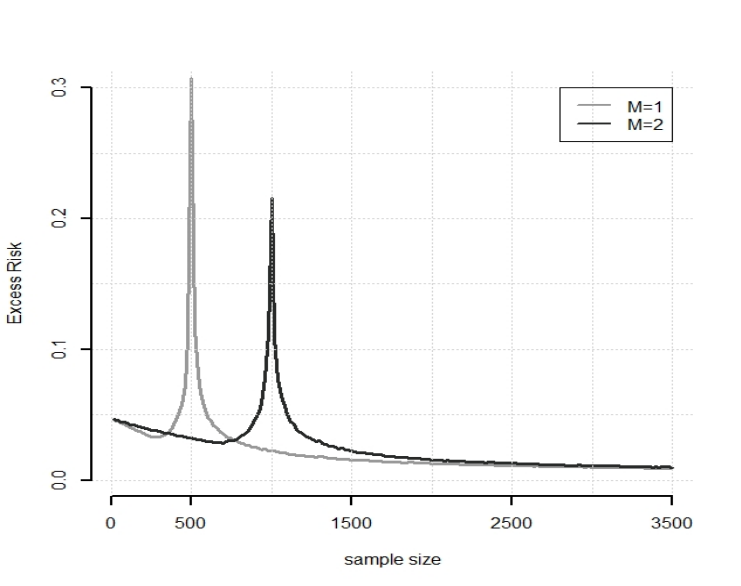
taken from [BHMM19]

Distributed OLS in (local) Overparameterized Setting



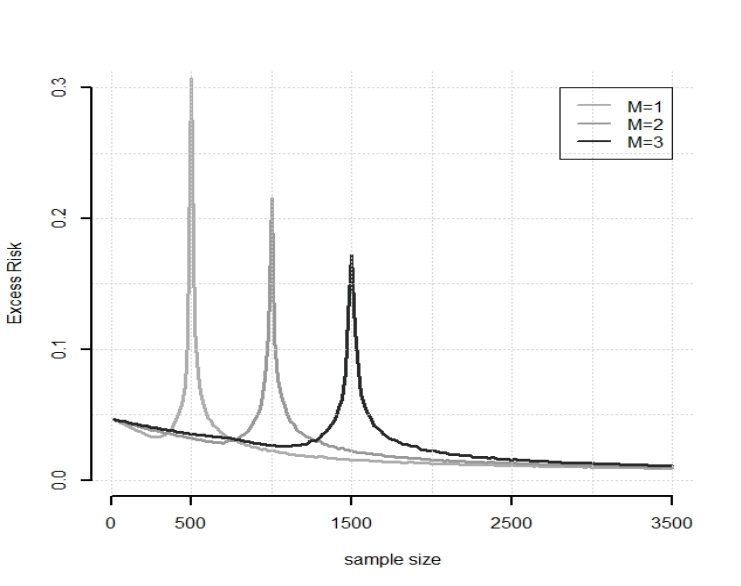
$d = 500$, peak at $n = d$

Distributed OLS in (local) Overparameterized Setting



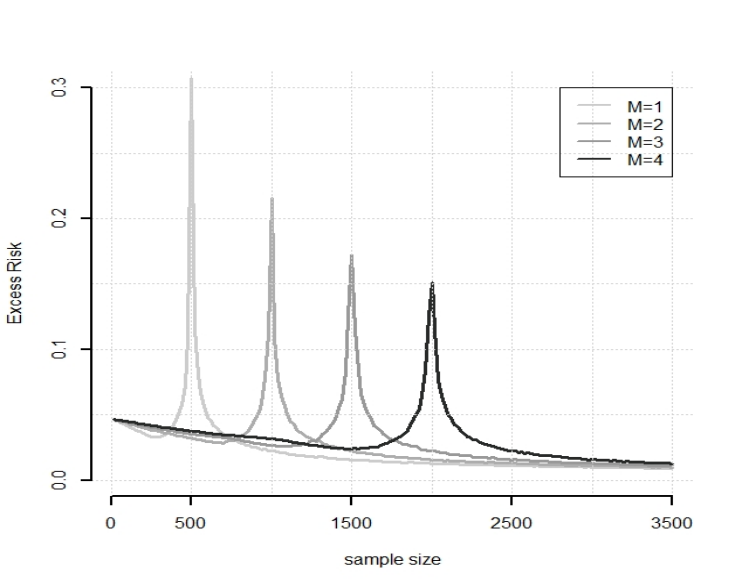
$d = 500$, peak at $n = d \cdot 2$

Distributed OLS in (local) Overparameterized Setting



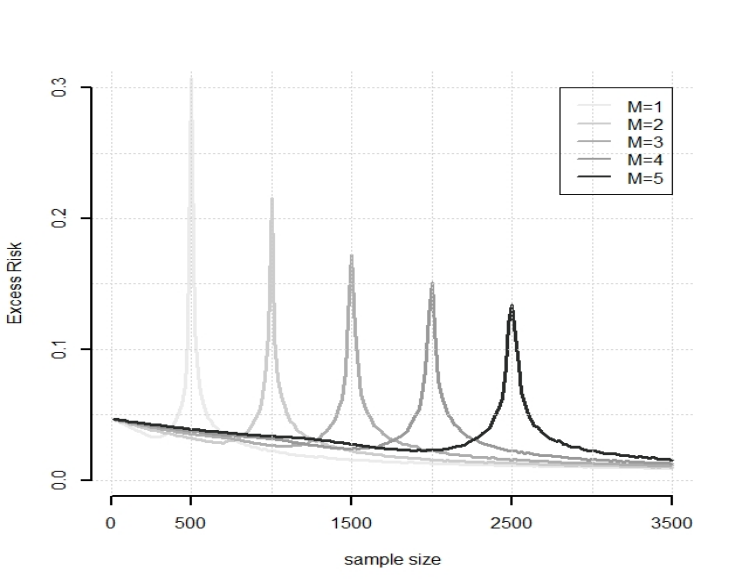
$d = 500$, peak at $n = d \cdot 3$

Distributed OLS in (local) Overparameterized Setting



$d = 500$, peak at $n = d \cdot 4$

Distributed OLS in (local) Overparameterized Setting



$d = 500$, peak at $n = d \cdot M$

A Lower Bound for Distributed Ridgeless Regression in finite Dimension

Assumptions:

1. Σ is invertible.
2. $\mathbb{E}[Y^2] < \infty$ and $\mathbb{E}[||X||^2] < \infty$.
3. For some $\sigma \geq 0$ we assume $\mathbb{V}[Y|X] \geq \sigma^2$ almost surely.
4. For any $m = 1, ..., M$, the data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$ has almost surely full rank.

A Lower Bound for Distributed Ridgeless Regression in finite Dimension

Assumptions:

1. Σ is invertible.
2. $\mathbb{E}[Y^2] < \infty$ and $\mathbb{E}[||X||^2] < \infty$.
3. For some $\sigma \geq 0$ we assume $\mathbb{V}[Y|X] \geq \sigma^2$ almost surely.
4. For any $m = 1, ..., M$, the data matrix $\mathrm{X}_m \in \mathbb{R}^{\frac{n}{M} \times d}$ has almost surely full rank.

Theorem [Reiss, M., Klein 21']:

$$\mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_M - \beta^*)||^2] \geq \frac{\sigma^2}{M} \frac{\min\{d, \frac{n}{M}\}}{\max\{d, \frac{n}{M}\} + 1 - \min\{d, \frac{n}{M}\}} .$$

Thus, we observe peaks at $d = \frac{n}{M}$ with height at least $\sigma^2 \frac{d}{M}$.

Excess Risk Upper Bound

recall: to evaluate the excess risk, we need a bias-variance decomposition

$$\mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_M - \beta^*)||^2] = \underbrace{\text{Bias}(\bar{\beta}_M)}_{\neq 0} + \text{Var}(\bar{\beta}_M)$$

Excess Risk Upper Bound

recall: to evaluate the excess risk, we need a bias-variance decomposition

$$\mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_M - \beta^*)||^2] = \underbrace{\text{Bias}(\bar{\beta}_M)}_{\neq 0} + \text{Var}(\bar{\beta}_M)$$

- calculating Var as before:

$$\text{Var}(\bar{\beta}_M) = \frac{\sigma^2}{M^2} \sum_{m=1}^{M} \mathbb{E}\Big[Tr[\Sigma^{1/2}(\mathrm{X}_m^T \mathrm{X}_m)^\dagger \Sigma^{1/2}] \Big]$$

- Bias no longer vanishing!

Bias Upper bound

- onvexity allows to deduce

$$\widehat{\text{Bias}}(\bar{\beta}_M) := ||\Sigma^{1/2}(\mathbb{E}_\epsilon[\bar{\beta}_M] - \beta^*)||^2 \leq \frac{1}{M}\sum_{m=1}^{M} ||\Sigma^{1/2}\tilde{\Pi}_m\beta^*||^2 \ ,$$

  with $\tilde{\Pi}_m := Id - X_m^T(X_m X_m^T)^\dagger X_m$ the orthogonal projection onto the nullspace of $X_m : \mathbb{R}^d \to \mathbb{R}^{\frac{n}{M}}$

- a concentration argument gives

$$\begin{aligned}
\text{Bias}(\bar{\beta}_M) &:= \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}[||\Sigma^{1/2}\tilde{\Pi}_m\beta^*||^2] \\
&\leq c\ \frac{1}{M}\sum_{m=1}^{M} \sqrt{\frac{M}{n}}\ ||\Sigma^{1/2}\beta^*||^2 \\
&= c\ \sqrt{\frac{M}{n}}\ ||\Sigma^{1/2}\beta^*||^2
\end{aligned}$$

Efficiency: Normal Distribution in the high dim Limit

Theorem [Reiss, M., Klein 21']:

Suppose $X \sim \mathcal{N}(0, Id_d)$ and $\Sigma = Id_d$. For $d \geq \frac{n}{M} + 2$, the bias and variance satisfy for some $c < \infty$

$$\mathrm{Bias}(\bar{\beta}_M) \leq c\, ||\beta^*||^2 \sqrt{\frac{M}{n}}\,, \quad \mathrm{Var}(\bar{\beta}_M) = \frac{\sigma^2}{M}\,\frac{n}{M(d-1)-n}\,.$$

Moreover, if $d \in \{\frac{n}{M} - 1, \frac{n}{M}, \frac{n}{M} + 1\}$, then $\mathbb{E}\big[\mathrm{Var}(\bar{\beta}_M)\big] = \infty$.

idea: to estimate the efficiency we upper bound the bias by the variance and use the previous lower bound for the excess risk

Corollary:

Define the signal-to-noise-ratio as $SNR := \frac{||\beta^*||}{\sigma}$. Assume that $\left(\frac{M_n d_n}{n}\right)_n$ is increasing. If for any $n \in \mathbb{N}$ sufficiently large the number of local nodes satisfies

$$M_n \lesssim \frac{1}{SNR^{4/5}} \left(\frac{1}{d_n}\right)^{2/5} n^{3/5} ,$$

then

$$\text{Bias}(\bar{\beta}_{M_n}) \leq \text{Var}(\bar{\beta}_{M_n})$$

and the expected excess risk satisfies

$$\mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_{M_n} - \beta^*)||^2] \leq 4\sigma^2 \frac{n}{M_n^2 d_n} .$$

Example: $d_n = \frac{n}{M}\gamma - 1$, where $\gamma > 1$ is called the aspect ratio

Corollary:

Define the signal-to-noise-ratio as $SNR := \frac{||\beta^*||}{\sigma}$. Assume that $\left(\frac{M_n d_n}{n}\right)_n$ is increasing. If for any $n \in \mathbb{N}$ sufficiently large the number of local nodes satisfies

$$M_n \lesssim \frac{1}{SNR^{4/5}} \left(\frac{1}{d_n}\right)^{2/5} n^{3/5} ,$$

then

$$\text{Bias}(\bar{\beta}_{M_n}) \leq \text{Var}(\bar{\beta}_{M_n})$$

and the expected excess risk satisfies

$$\mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_{M_n} - \beta^*)||^2] \leq 4\sigma^2 \frac{n}{M_n^2 d_n} .$$

Example: $d_n = \frac{n}{M}\gamma - 1$, where $\gamma > 1$ is called the aspect ratio

in this case, overfitting is benign, i.e. the (generalization) error converges to zero as $\frac{1}{M_n}$ for any choice $M_n \simeq n^\alpha$, $\alpha \in (0, \frac{1}{2}]$

Corollary (Efficiency):

Assumptions as above. If for any $n \in \mathbb{N}$ sufficiently large the number of local nodes satisfies

$$M_n \lesssim \frac{1}{SNR^{4/5}} \left(\frac{1}{d_n}\right)^{2/5} n^{3/5} ,$$

then

$$\mathrm{Eff}(M) \asymp M^2 .$$

Corollary (Efficiency):

Assumptions as above. If for any $n \in \mathbb{N}$ sufficiently large the number of local nodes satisfies

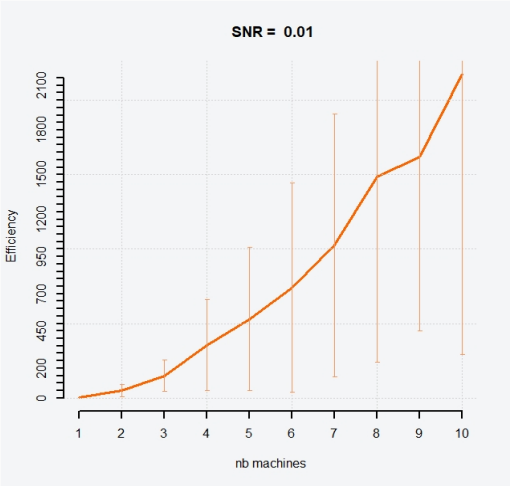
$$M_n \lesssim \frac{1}{SNR^{4/5}} \left(\frac{1}{d_n}\right)^{2/5} n^{3/5} ,$$

then

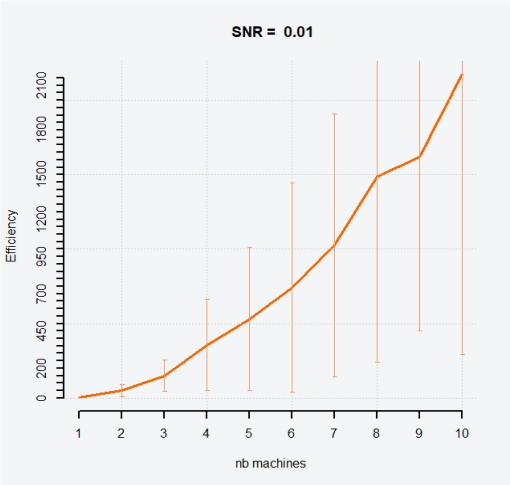
$$\mathrm{Eff}(M) \asymp M^2 .$$

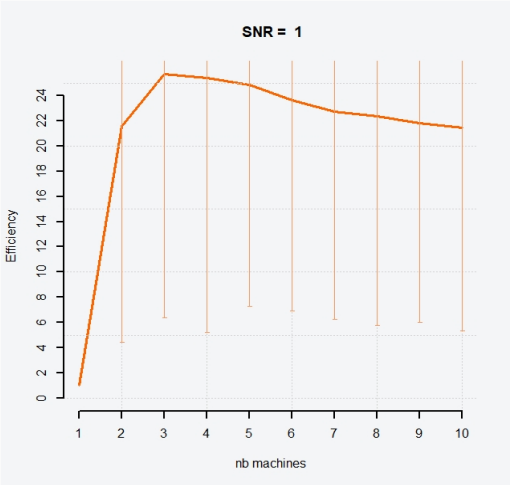
Proof: follows from the last Corollary and the previous lower bound

Distributed OLS in Overparameterized Setting



Distributed OLS in Overparameterized Setting





Question: What can we say about the efficiency in the other regime where M is (too) large ?

Question: What can we say about the efficiency in the other regime where M is (too) large ?

Corollary:

Assumptions as above. If for any $n \in \mathbb{N}$ sufficiently large the number of local nodes satisfies

$$M_n \gtrsim \frac{1}{SNR^{4/5}} \left(\frac{1}{d_n}\right)^{2/5} n^{3/5} ,$$

then

$$\text{Bias}(\bar{\beta}_{M_n}) \geq \text{Var}(\bar{\beta}_{M_n})$$

and the expected excess risk satisfies

$$\frac{\sigma^2}{M_n} \frac{n/M_n}{d+1-n/M_n} \lesssim \mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_{M_n} - \beta^*)||^2] \lesssim ||\beta^*||^2 \sqrt{\frac{M_n}{n}} .$$

Question: What can we say about the efficiency in the other regime where M is (too) large ?

Corollary:

Assumptions as above. If for any $n \in \mathbb{N}$ sufficiently large the number of local nodes satisfies

$$M_n \gtrsim \frac{1}{SNR^{4/5}} \left(\frac{1}{d_n}\right)^{2/5} n^{3/5} ,$$

then

$$\text{Bias}(\bar{\beta}_{M_n}) \geq \text{Var}(\bar{\beta}_{M_n})$$

and the expected excess risk satisfies

$$\frac{\sigma^2}{M_n} \frac{n/M_n}{d+1-n/M_n} \lesssim \mathbb{E}[||\Sigma^{1/2}(\bar{\beta}_{M_n} - \beta^*)||^2] \lesssim ||\beta^*||^2 \sqrt{\frac{M_n}{n}} .$$

overfitting is still benign for $M_n \simeq n^\alpha$, $\alpha \in [\frac{1}{2}, 1)$

Efficiency in the high dim limit

more concretely: $d = n\gamma - 1$, $\gamma > 1$ such that $d/n \to \gamma$ as $n \to \infty$, then

[1] In fact, improved bounds are ongoing research, also the extension to more general distributions!

Efficiency in the high dim limit

more concretely: $d = n\gamma - 1$, $\gamma > 1$ such that $d/n \to \gamma$ as $n \to \infty$, then

$$\frac{\sqrt{n}}{SNR^2\gamma}\frac{1}{\sqrt{M}} \lesssim \mathrm{Eff}(M) \lesssim SNR^2\gamma\frac{M}{\sqrt{n}}$$

[1] In fact, improved bounds are ongoing research, also the extension to more general distributions!

Efficiency in the high dim limit

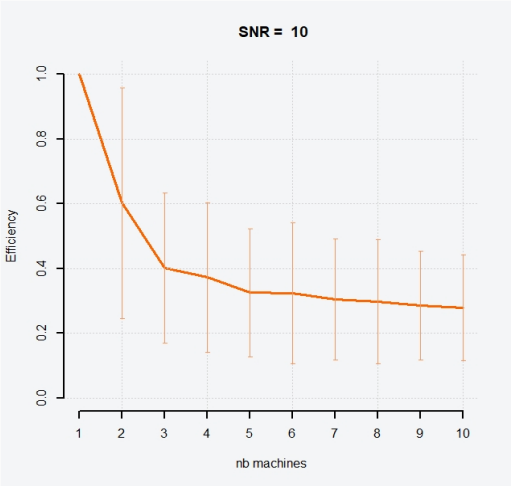
more concretely: $d = n\gamma - 1$, $\gamma > 1$ such that $d/n \to \gamma$ as $n \to \infty$, then

$$\frac{\sqrt{n}}{SNR^2\gamma}\frac{1}{\sqrt{M}} \lesssim \mathrm{Eff}(M) \lesssim SNR^2\gamma\frac{M}{\sqrt{n}}$$

- the upper bound seems very loose: this is due to a loose bias bound![1]
- loss in efficiency of not less than $1/\sqrt{M}$

[1] In fact, improved bounds are ongoing research, also the extension to more general distributions!

Distributed OLS in Overparameterized Setting



Summary

- the risk curve is not monotonically decreasing in the sample size (double descent); we observe peaks at the interpolation threshold $d = n \cdot M$

Summary

- the risk curve is not monotonically decreasing in the sample size (double descent); we observe peaks at the interpolation threshold $d = n \cdot M$
- the bias no longer vanishes and DL reduces variance

Summary

- the risk curve is not monotonically decreasing in the sample size (double descent); we observe peaks at the interpolation threshold $d = n \cdot M$
- the bias no longer vanishes and DL reduces variance
- if the number of machines is not too large, then the efficiency grow quadratically

Summary

- the risk curve is not monotonically decreasing in the sample size (double descent); we observe peaks at the interpolation threshold $d = n \cdot M$
- the bias no longer vanishes and DL reduces variance
- if the number of machines is not too large, then the efficiency grow quadratically
- DL in the overparameterized regime is more efficient when the SNR is low

Distributed Kernel Methods

Supervised Learning Problem: Nonparametric Regression

$$\inf_{f \in \mathcal{H}} \mathcal{R}(f) \,, \quad \mathcal{R}(f) := \int_{\mathcal{X} \times \mathcal{Y}} (f(x) - y)^2 dP(x, y)$$

- input space $\mathcal{X}$, output space $\mathcal{Y} \subset \mathbb{R}$
- $\mathcal{H}$ space of candidate solutions: reproducing kernel Hilbert space (RKHS) with kernel K

Supervised Learning Problem: Nonparametric Regression

$$\inf_{f \in \mathcal{H}} \mathcal{R}(f) \,, \quad \mathcal{R}(f) := \int_{\mathcal{X} \times \mathcal{Y}} (f(x) - y)^2 dP(x, y)$$

- input space $\mathcal{X}$, output space $\mathcal{Y} \subset \mathbb{R}$
- $\mathcal{H}$ space of candidate solutions: reproducing kernel Hilbert space (RKHS) with kernel K
- good empirical solution $\hat{f}$ should have small excess risk

$$\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{H}} \mathcal{R}(\hat{f})$$

Intro: Kernel Methods

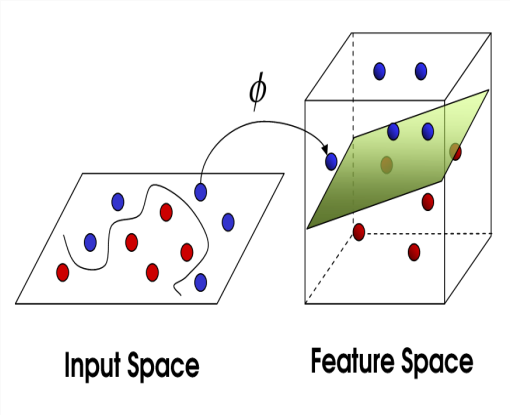
material from textbook [SC08, Chapter 4]

Definition:

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel on $\mathcal{X}$, if there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \ , \quad x, x' \in \mathcal{X} \ .$$

- kernel trick: kernel methods only require computing the inner products rather than ϕ itself (an explicit representation for ϕ is not necessary!)



Intro: Kernel Methods

How to decide if a given function k is a kernel when we do not know the feature map?

Theorem:

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if and only if it is symmetric and positive definite, i.e.

- symmetry: $k(x, x') = k(x', x)$ for all $x, x' \in \mathcal{X}$
- pd: For all $\alpha_1, ..., \alpha_n \in \mathbb{R}$, $x_1, ..., x_n \in \mathcal{X}$ we have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \ .$$

Intro: Intro: Kernel Methods

Definition:

Let $\mathcal{H}$ be a Hilbert space that consists of functions mapping from $\mathcal{X}$ into $\mathbb{R}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of $\mathcal{H}$ if $k_x(\cdot) := k(x, \cdot) \in \mathcal{H}$ for all $x \in \mathcal{X}$ and if the reproducing property holds:

$$f(x) = \langle f, k_x \rangle \ , \quad \forall f \in \mathcal{H} \ , \ \ x \in \mathcal{X} \ .$$

Intro: Intro: Kernel Methods

Definition:

Let $\mathcal{H}$ be a Hilbert space that consists of functions mapping from $\mathcal{X}$ into $\mathbb{R}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of $\mathcal{H}$ if $k_x(\cdot) := k(x, \cdot) \in \mathcal{H}$ for all $x \in \mathcal{X}$ and if the reproducing property holds:

$$f(x) = \langle f, k_x \rangle \ , \quad \forall f \in \mathcal{H} \ , \ \ x \in \mathcal{X} \ .$$

$\mathcal{H}$ is a reproducing kernel Hilbert space if for all $f \in \mathcal{H}$

$$|f(x)| \leq C_x \, ||f|| \ .$$

Theorem:

Every kernel k can be associated to a unique RKHS. Conversely, every RKHS has a unique kernel.

Example: Classics

1. Gaussian Kernel: $\sigma > 0$

$$k(x, x') = e^{-\sigma^2 ||x - x'||_2^2} , \quad x, x' \in \mathbb{R}^d .$$

2. Polynomial Kernel: $p \geq 0$, $c \in \mathbb{R}$

$$k(x, x') = (\langle x, x' \rangle + c)^p , \quad x, x' \in \mathbb{R}^d .$$

Example: Classics

1. Gaussian Kernel: $\sigma > 0$

$$k(x, x') = e^{-\sigma^2 ||x - x'||_2^2} , \quad x, x' \in \mathbb{R}^d .$$

2. Polynomial Kernel: $p \geq 0$, $c \in \mathbb{R}$

$$k(x, x') = (\langle x, x' \rangle + c)^p , \quad x, x' \in \mathbb{R}^d .$$

Example: Neural Tangent Kernel (NTK) [JGH18]

Let $f_\theta^{NN}(x)$ denote the output of a fully connected neural network, with $x \in \mathbb{R}^d$ and parameters $\theta \in \mathbb{R}^p$ to be learned. Assume this is done by some gradient based algorithm, initialized at θ_0. The feature map $\phi_{\theta_0} : \mathbb{R}^d \to \mathbb{R}^p$ at initialization θ_0

$$\phi_{\theta_0}(x) := \nabla f_\theta^{NN}(x) \mid_{\theta = \theta_0}$$

defines the NTK as

$$k(x, x') = \langle \phi_{\theta_0}(x), \phi_{\theta_0}(x') \rangle .$$

Kernel Ridge Regression (KRR)

- in kernel methods we consider functions of the form

$$f(x) = \sum_{j=1}^{n} \alpha_j K(x, x_j)$$

- coefficients $\alpha_1, ..., \alpha_n$ are derived from a convex optimization problem

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j)^2 + \lambda ||f||_{\mathcal{H}}^2$$

(Ridge Regression Estimator)

Kernel Ridge Regression (KRR)

- in kernel methods we consider functions of the form

$$f(x) = \sum_{j=1}^{n} \alpha_j K(x, x_j)$$

- coefficients $\alpha_1, ..., \alpha_n$ are derived from a convex optimization problem

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j)^2 + \lambda ||f||^2_{\mathcal{H}}$$

(Ridge Regression Estimator)

- computations are reduced to solving a linear system

$$(\mathbb{K} + \lambda n I)\alpha = \mathrm{Y}\ , \quad \hat{\alpha} = (\mathbb{K} + \lambda n I)^{-1}\mathrm{y}$$

where $\mathbb{K} = (K(x_i, x_j))_{i,j}$ is the $n \times n$ kernel matrix

Properties of KRR

Computations:

- solving the matrix inversion for large datasets is challenging
- direkt approach requires $\mathcal{O}(n^2)$ in space to allocate $\mathbb{K}$, $\mathcal{O}(n^2)$ kernel evaluations and $\mathcal{O}(n^3)$ in time to compute and invert $\mathbb{K}$

Properties of KRR

Computations:

- solving the matrix inversion for large datasets is challenging
- direkt approach requires $\mathcal{O}(n^2)$ in space to allocate $\mathbb{K}$, $\mathcal{O}(n^2)$ kernel evaluations and $\mathcal{O}(n^3)$ in time to compute and invert $\mathbb{K}$

Statistics:

- under basic assumptions, KRR achieves an error of $\mathcal{O}(1/\sqrt{n})$ for $\lambda_n = 1/\sqrt{n}$
- optimal in a minimax sense, can be improved under more stringent assumptions

Gradient Methods and Early Stopping

- direct empirical risk minimization (without additional regularization)

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j)^2$$

Gradient Methods and Early Stopping

- direct empirical risk minimization (without additional regularization)

$$\hat{f}_\lambda = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j)^2$$

- natural idea is to consider iterative solvers, in particular gradient methods

$$\alpha_t = \alpha_{t-1} + \gamma(\mathbb{K}\alpha_{t-1} - \mathrm{y})$$

  for a suitable step-size choice $\gamma > 0$

Gradient Methods and Early Stopping

- direct empirical risk minimization (without additional regularization)

$$\hat{f}_\lambda = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (f(x_j) - y_j)^2$$

- natural idea is to consider iterative solvers, in particular gradient methods

$$\alpha_t = \alpha_{t-1} + \gamma(\mathbb{K}\alpha_{t-1} - \mathrm{y})$$

  for a suitable step-size choice $\gamma > 0$

Computations:

- if t is the number of iterations, gradient methods require $\mathcal{O}(tn^2)$ in time, $\mathcal{O}(n^2)$ in memory and $\mathcal{O}(n^2)$ in kernel evaluations if the kernel matrix is stored
- note: kernel matrix can be computed on the fly with only $\mathcal{O}(n)$ memory but $\mathcal{O}(tn^2)$ kernel evaluations are required

Properties

Statistics:

- regularization is performed by early stopping a.k.a. implicit regularization (choosing a suitable stopping time) then $\mathcal{O}(\sqrt{n})$ iterations are needed to achieve the optimal rate of $\mathcal{O}(1/\sqrt{n})$[2]
- optimal in a minimax sense, can be improved under more stringent assumptions

[2] accelerated methods and stochastic methods are even faster

Properties

Statistics:

- regularization is performed by early stopping a.k.a. implicit regularization (choosing a suitable stopping time) then $\mathcal{O}(\sqrt{n})$ iterations are needed to achieve the optimal rate of $\mathcal{O}(1/\sqrt{n})$[2]
- optimal in a minimax sense, can be improved under more stringent assumptions
- time complexity improves, but number of kernel evaluations and memory requirements make application to large scale setting hard

[2] accelerated methods and stochastic methods are even faster

Distributed Kernel Methods

- again: assume the data is comprised of chunks $D = D_1 \cup ... \cup D_M$
- apply one of the above algorithms locally to compute a local estimator

$$\hat{f}_{m,\lambda} = \frac{M}{n} \sum_{j=1}^{n/M} \hat{\alpha}_j(\lambda) K(x_m^{(j)}, \cdot)$$

note: for GD, $\lambda = \frac{1}{\gamma T}$

Distributed Kernel Methods

- again: assume the data is comprised of chunks $D = D_1 \cup ... \cup D_M$
- apply one of the above algorithms locally to compute a local estimator

$$\hat{f}_{m,\lambda} = \frac{M}{n} \sum_{j=1}^{n/M} \hat{\alpha}_j(\lambda) K(x_m^{(j)}, \cdot)$$

  note: for GD, $\lambda = \frac{1}{\gamma T}$
- then build a global average $\bar{f}_\lambda = \frac{1}{M} \sum_{m=1}^{M} \hat{f}_{m,\lambda}$

Distributed Kernel Methods

- again: assume the data is comprised of chunks $D = D_1 \cup ... \cup D_M$
- apply one of the above algorithms locally to compute a local estimator

$$\hat{f}_{m,\lambda} = \frac{M}{n} \sum_{j=1}^{n/M} \hat{\alpha}_j(\lambda) K(x_m^{(j)}, \cdot)$$

  note: for GD, $\lambda = \frac{1}{\gamma T}$
- then build a global average $\bar{f}_\lambda = \frac{1}{M} \sum_{m=1}^{M} \hat{f}_{m,\lambda}$

Question: What can we say about the performance for these class of methods ?

Distributed Kernel Methods

- again: assume the data is comprised of chunks $D = D_1 \cup ... \cup D_M$
- apply one of the above algorithms locally to compute a local estimator

$$\hat{f}_{m,\lambda} = \frac{M}{n} \sum_{j=1}^{n/M} \hat{\alpha}_j(\lambda) K(x_m^{(j)}, \cdot)$$

  note: for GD, $\lambda = \frac{1}{\gamma T}$
- then build a global average $\bar{f}_\lambda = \frac{1}{M} \sum_{m=1}^{M} \hat{f}_{m,\lambda}$

Question: What can we say about the performance for these class of methods ?

crucial: choice of regularization parameter $\lambda > 0$

Assumptions:

1. Well-specified model: The regression function $f^*(x) = \mathbb{E}[Y|X = x]$ belongs to $\mathcal{H}$.

Assumptions:

1. Well-specified model: The regression function $f^*(x) = \mathbb{E}[Y|X = x]$ belongs to $\mathcal{H}$.
2. The kernel K is bounded a.s.: $\sup_{x,x'} K(x, x') =: \kappa^2 < \infty$

Assumptions:

1. Well-specified model: The regression function $f^*(x) = \mathbb{E}[Y|X = x]$ belongs to $\mathcal{H}$.
2. The kernel K is bounded a.s.: $\sup_{x,x'} K(x, x') =: \kappa^2 < \infty$
3. Bernstein-Noise: $\mathbb{E}[|Y - f^*(X)|^\ell \mid X] \leq \frac{1}{2}\ell!\sigma^2 B^{\ell-2}$ for all $\ell \geq 2$ and some $\sigma >$, $B > 0$.

Assumptions:

1. Well-specified model: The regression function $f^*(x) = \mathbb{E}[Y|X = x]$ belongs to $\mathcal{H}$.
2. The kernel K is bounded a.s.: $\sup_{x,x'} K(x, x') =: \kappa^2 < \infty$
3. Bernstein-Noise: $\mathbb{E}[|Y - f^*(X)|^\ell \mid X] \leq \frac{1}{2}\ell!\sigma^2 B^{\ell-2}$ for all $\ell \geq 2$ and some $\sigma >$, $B > 0$.

Definition:

1. The covariance operator $\Sigma : \mathcal{H} \to \mathcal{H}$ is defined as $\Sigma = \mathbb{E}[K_X \times K_X]$.

Assumptions:

1. Well-specified model: The regression function $f^*(x) = \mathbb{E}[Y|X = x]$ belongs to $\mathcal{H}$.
2. The kernel K is bounded a.s.: $\sup_{x,x'} K(x, x') =: \kappa^2 < \infty$
3. Bernstein-Noise: $\mathbb{E}[|Y - f^*(X)|^\ell \mid X] \leq \frac{1}{2}\ell!\sigma^2 B^{\ell-2}$ for all $\ell \geq 2$ and some $\sigma >$, $B > 0$.

Definition:

1. The covariance operator $\Sigma : \mathcal{H} \to \mathcal{H}$ is defined as $\Sigma = \mathbb{E}[K_X \times K_X]$.
2. The empirical covariance operator $\hat{\Sigma} : \mathcal{H} \to \mathcal{H}$ is defined as $\hat{\Sigma} = \frac{1}{n}\sum_{j=1}^{n} K_{x_j} \otimes K_{x_j}$.

Hardness of the Learning Problem

Assumption: (Regularity)

We assume that for some $r > 0$, the regression function satisfies for some $h \in \mathcal{H}$

$$f^* = \Sigma^r h \,, \quad ||h|| < \infty \,.$$

Hardness of the Learning Problem

Assumption: (Regularity)

We assume that for some $r > 0$, the regression function satisfies for some $h \in \mathcal{H}$

$$f^* = \Sigma^r h \,, \quad ||h|| < \infty \,.$$

- such an assumption is called a source condition
- it quantifies the hardness of the learning problem (the larger r, the easier the problem)

as before: good error bounds depend on the bias-variance decomposition

$$f^* - \bar{f}_\lambda \;=\; \widehat{\mathrm{Bias}}(\bar{f}_\lambda)) \;+\; \widehat{\mathrm{Var}}(\bar{f}_\lambda)$$

as before: good error bounds depend on the bias-variance decomposition

$$f^* - \bar{f}_\lambda \;=\; \widehat{\mathrm{Bias}}(\bar{f}_\lambda)) \;+\; \widehat{\mathrm{Var}}(\bar{f}_\lambda)$$

Bias:

$$\widehat{\mathrm{Bias}}(\bar{f}_\lambda) := \frac{1}{M}\sum_{m=1}^{M} \underbrace{(\hat{\Sigma}_m(\hat{\Sigma}_m + \lambda I)^{-1} - I)}_{\text{almost a projection for small } \lambda} \quad f^*$$

as before: good error bounds depend on the bias-variance decomposition

$$f^* - \bar{f}_\lambda \;=\; \widehat{\mathrm{Bias}}(\bar{f}_\lambda)) \;+\; \widehat{\mathrm{Var}}(\bar{f}_\lambda)$$

as before: good error bounds depend on the bias-variance decomposition

$$f^* - \bar{f}_\lambda \;=\; \widehat{\mathrm{Bias}}(\bar{f}_\lambda)) \;+\; \widehat{\mathrm{Var}}(\bar{f}_\lambda)$$

Variance:

$$\widehat{\mathrm{Var}}(\bar{f}_\lambda) := \frac{1}{M}\sum_{m=1}^{M}(\hat{\Sigma}_m + \lambda I)^{-1}(\hat{\Sigma}_m f^* - \widehat{\mathbb{E}}[YK_X])$$

with

$$\widehat{\mathbb{E}}[YK_X] := \frac{M}{n}\sum_{j=1}^{n/M} y_m^{(j)} K(x_m^{(j)}, \cdot)$$

is a sum i.i.d. variables with zero mean $\rightarrow$ apply concentration arguments

Theorem: [MB18]

Assume f^* is bounded and

$$M_n \leq n^\alpha \ , \quad \alpha < \frac{\min\{1, r\}}{r+1} \ .$$

The excess risk satisfies

$$\mathbb{E}[||\Sigma^{1/2}(f^* - \bar{f}_\lambda)||^2] \ \lesssim \ \lambda^{2r+1} + \frac{1}{\lambda n} \ .$$

Theorem: [MB18]

Assume f^* is bounded and

$$M_n \leq n^\alpha \ , \quad \alpha < \frac{\min\{1, r\}}{r+1} \ .$$

The excess risk satisfies

$$\mathbb{E}[||\Sigma^{1/2}(f^* - \bar{f}_\lambda)||^2] \lesssim \lambda^{2r+1} + \frac{1}{\lambda n} \ .$$

In particular, choosing

$$\lambda_n = \left(\frac{1}{\sqrt{n}}\right)^{\frac{1}{r+1}}$$

gives

$$\mathbb{E}[||\Sigma^{1/2}(f^* - \bar{f}_\lambda)||^2] \lesssim \left(\frac{1}{\sqrt{n}}\right)^{\frac{2r+1}{r+1}} \ .$$

Theorem: [MB18]

Assume f^* is bounded and

$$M_n \leq n^\alpha \ , \quad \alpha < \frac{\min\{1, r\}}{r+1} \ .$$

The excess risk satisfies

$$\mathbb{E}[||\Sigma^{1/2}(f^* - \bar{f}_\lambda)||^2] \lesssim \lambda^{2r+1} + \frac{1}{\lambda n} \ .$$

In particular, choosing

$$\lambda_n = \left(\frac{1}{\sqrt{n}}\right)^{\frac{1}{r+1}}$$

gives

$$\mathbb{E}[||\Sigma^{1/2}(f^* - \bar{f}_\lambda)||^2] \lesssim \left(\frac{1}{\sqrt{n}}\right)^{\frac{2r+1}{r+1}} \ .$$

This rate is optimal.

Remarks:

- even though each partitioned sub-problem is based only on the fraction n/M of samples, it is essential to regularize the partitioned sub-problems as though they had all n samples

Remarks:

- even though each partitioned sub-problem is based only on the fraction n/M of samples, it is essential to regularize the partitioned sub-problems as though they had all n samples
- from a local point of view, each sub-problem is under-regularized

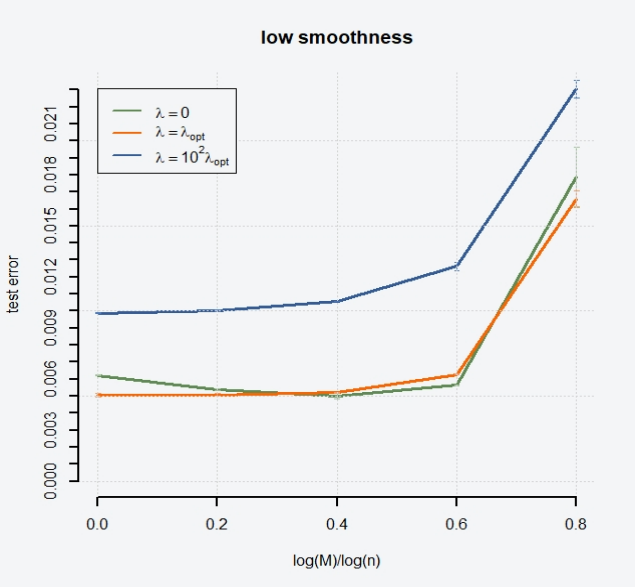
Remarks:

- even though each partitioned sub-problem is based only on the fraction n/M of samples, it is essential to regularize the partitioned sub-problems as though they had all n samples
- from a local point of view, each sub-problem is under-regularized
- this allows the bias of each local estimate to be very small, but it causes a detrimental blow-up in the variance

Remarks:

- even though each partitioned sub-problem is based only on the fraction n/M of samples, it is essential to regularize the partitioned sub-problems as though they had all n samples
- from a local point of view, each sub-problem is under-regularized
- this allows the bias of each local estimate to be very small, but it causes a detrimental blow-up in the variance
- averaging reduces variance enough that the resulting estimator $\bar{f}_\lambda$ still attains optimal convergence rate

Distributed regularized KRR



Summary

Method	**OLS under-param**	**OLS over-param**	**regularized kernel**
Efficiency as a fct. of M	decreases linearly	increases as M^2 (if M is not too large)	constant (if M is not too large and with optimal regularization)

References i

Leo Breiman and David Freedman.
How many variables should be entered in a regression equation?
Journal of the American Statistical Association, 78(381):131–136, 1983.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal.
Reconciling modern machine learning practice and the bias-variance trade-off, 2019.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler.
Benign overfitting in linear regression.
Proceedings of the National Academy of Sciences, 117(48):30063–30070, 2020.

Geoffrey Chinot and Matthieu Lerasle.
Benign overfitting in the large deviation regime.
arXiv preprint arXiv:2003.05838, 2020.

References ii

Edgar Dobriban and Yue Sheng.
Distributed linear regression by averaging.
The Annals of Statistics, 49(2):918–943, 2021.

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer.
Regularization of inverse problems, volume 375.
Springer Science & Business Media, 1996.

Arthur Jacot, Franck Gabriel, and Clément Hongler.
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Dmitry Kobak, Jonathan Lomond, and Benoit Sanchez.
The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization.
Journal of Machine Learning Research, 21(169):1–16, 2020.

References iii

Tengyuan Liang, Alexander Rakhlin, et al.
Just interpolate: Kernel ridgeless regression can generalize.
Annals of Statistics, 48(3):1329–1347, 2020.

Nicole Mücke and Gilles Blanchard.
Parallelizing spectrally regularized kernel algorithms.
The Journal of Machine Learning Research, 19(1):1069–1097, 2018.

Jaouad Mourtada.
Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices.
arXiv preprint arXiv:1912.10754, 2019.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai.
Harmless interpolation of noisy data in regression.
IEEE Journal on Selected Areas in Information Theory, 1(1):67–83, 2020.

References iv

Dominic Richards, Jaouad Mourtada, and Lorenzo Rosasco.
Asymptotics of ridge (less) regression under general source condition.
arXiv preprint arXiv:2006.06386, 2020.

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

I. Steinwart and A. Christmann.
Support Vector Machines.
Springer, 2008.

Jun Shao.
Mathematical statistics: exercises and solutions.
Springer Science & Business Media, 2006.

References v

Alexander Tsigler and Peter L Bartlett.
Benign overfitting in ridge regression.
arXiv preprint arXiv:2009.14286, 2020.

Yuxin Wang, Qiang Wang, Shaohuai Shi, Xin He, Zhenheng Tang, Kaiyong Zhao, and Xiaowen Chu.
Benchmarking the performance and energy efficiency of ai accelerators for ai training.
In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pages 744–751. IEEE, 2020.