University of Pennsylvania
ESE PhD Colloquium - Feb 13th, 2023.

Selected Topics in

High-Dimensional Regression

From Double Descent to Learning under Distribution Shift

Behrad Moniri
bemoniri@seas.upenn.edu

Outline

1. Introduction
2. Linear Regression
3. Random Features Regression
4. Accuracy-on-the-line and Agreement-on-the-line

Last part is based on a recent joint work with Donghwan Lee, Xinmeng Huang, Edgar Dobriban, and Hamed Hassani.

Introduction

Introduction

- State-of-the-art deep learning models have millions or billions of parameters.

Introduction

- State-of-the-art deep learning models have millions or billions of parameters.
  - **Resnet18**: 11 million

Introduction

- State-of-the-art deep learning models have millions or billions of parameters.

  - **Resnet18**: 11 million
  - **DALL-E 2**: 3.5 billion

Introduction

- State-of-the-art deep learning models have millions or billions of parameters.

  - **Resnet18**: 11 million
  - **DALL-E 2**: 3.5 billion
  - **Chat GPT**: 175 billion

Introduction

- State-of-the-art deep learning models have millions or billions of parameters.
  - **Resnet18**: 11 million
  - **DALL-E 2**: 3.5 billion
  - **Chat GPT**: 175 billion
- Common wisdom suggests they should overfit.

Introduction

- State-of-the-art deep learning models have millions or billions of parameters.

  - **Resnet18**: 11 million
  - **DALL-E 2**: 3.5 billion
  - **Chat GPT**: 175 billion

- Common wisdom suggests they should overfit. But this can't be true :)

Experiments on MNIST

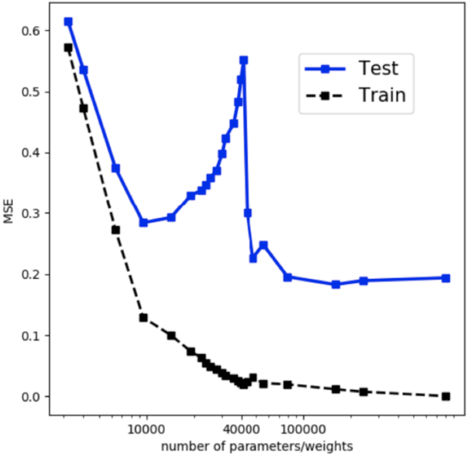


Figure: [Belkin et al., 2018].

Double Descent

Question:

Question:

Is double descent unique to deep neural networks?

Double Descent

Question:

Is double descent unique to deep neural networks?

No! It can even be seen in *very* simple models.

Linear Regression

Lets first define the problem.

High-Dimensional Linear Regression

Lets first define the problem.

- **Data Generation**:

Lets first define the problem.

- **Data Generation**:

$$\begin{cases} \beta \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d \times d} \\ \\ \end{cases}$$

High-Dimensional Linear Regression

Lets first define the problem.

- **Data Generation**:

$$\begin{cases} \beta \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d \times d} \\ \{x_i\}_{i=1}^n \sim \mathcal{N}(0, \Sigma) \\ \end{cases}$$

Lets first define the problem.

- **Data Generation**:

$$\begin{cases} \beta \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d} \\ \{x_i\}_{i=1}^n \sim \mathcal{N}(0, \Sigma) \\ y_i = \beta^\top x_i + \varepsilon_i \text{ where } \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \end{cases} \tag{1}$$

High-Dimensional Linear Regression

Lets first define the problem.

- **Data Generation**:

$$\begin{cases} \beta \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d} \\ \{x_i\}_{i=1}^n \sim \mathcal{N}(0, \Sigma) \\ y_i = \beta^\top x_i + \varepsilon_i \ \text{ where } \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \end{cases} \tag{1}$$

- **Fit with ridge regression**:

High-Dimensional Linear Regression

Lets first define the problem.

- **Data Generation**:

$$\begin{cases} \beta \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d} \\ \{x_i\}_{i=1}^n \sim \mathcal{N}(0, \Sigma) \\ y_i = \beta^\top x_i + \varepsilon_i \ \text{ where } \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \end{cases} \tag{1}$$

- **Fit with ridge regression**:

$$\hat{\beta}_\lambda = \arg\min_{b\in\mathbb{R}^d} \left[\frac{1}{n} \sum_{i=1}^n \left(y_i - b^\top x_i \right)^2 + \lambda ||b||_2^2 \right] \tag{2}$$

Ridgeless Limit

- *Ridgeless* Limit $\lambda \to 0$:

$$\hat{\beta}_0 = (X^\top X)^+ X^\top Y$$

- *Ridgeless* Limit $\lambda \to 0$:

$$\hat{\beta}_0 = (X^\top X)^+ X^\top Y$$

In the overparameterized case, this is the minimum-norm interpolator.

Ridgeless Limit

- *Ridgeless* Limit $\lambda \to 0$:

$$\hat{\beta}_0 = (X^\top X)^+ X^\top Y$$

  In the overparameterized case, this is the minimum-norm interpolator.

- When X has full column rank: $\hat{\beta}_0 = (X^\top X)^{-1} X^\top Y$.

Risk of the estimator

- **Question:** What is the risk of $\hat{\beta}_0$?

Risk of the estimator

- **Question:** What is the risk of $\hat{\beta}_0$?

$$
\begin{aligned}
R_X(\hat{\beta};\beta) &= \mathbb{E}[(x_o^\top \hat{\beta} - x_o^\top \beta)^2 \,|\, X] \\
&= \underbrace{\beta^T \Pi \Sigma \Pi \beta}_{B_X(\hat{\beta};\beta)} + \underbrace{\frac{\sigma^2}{n} \operatorname{Tr}\left(\hat{\Sigma}^+ \Sigma\right)}_{V_X(\hat{\beta};\beta)}
\end{aligned}
$$

where $\Pi = I - \hat{\Sigma}^+ \hat{\Sigma}$, and $\hat{\Sigma} = \frac{1}{n} X^\top X$.

Proportional Regime

$$R_X(\hat{\beta}; \beta_0) = \beta^T \Pi \Sigma \Pi \beta + \frac{\sigma^2}{n} \operatorname{Tr}\left(\hat{\Sigma}^+ \Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+ \hat{\Sigma}$$

Proportional Regime

$$R_X(\hat{\beta};\beta_0) = \beta^T \Pi\Sigma\Pi\beta + \frac{\sigma^2}{n}\operatorname{Tr}\left(\hat{\Sigma}^+\Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+\hat{\Sigma}$$

- This is not that insightful!

Proportional Regime

$$R_X(\hat{\beta}; \beta_0) = \beta^T \Pi \Sigma \Pi \beta + \frac{\sigma^2}{n} \operatorname{Tr}\left(\hat{\Sigma}^+ \Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+ \hat{\Sigma}$$

- This is not that insightful!
- How does $R_X(\beta_0, \hat{\beta})$ depend on sample size and dimension?

Proportional Regime

$$R_X(\hat{\beta}; \beta_0) = \beta^T \Pi \Sigma \Pi \beta + \frac{\sigma^2}{n} \operatorname{Tr}\left(\hat{\Sigma}^+ \Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+ \hat{\Sigma}$$

- This is not that insightful!
- How does $R_X(\beta_0, \hat{\beta})$ depend on sample size and dimension?
- This is well known in the regime where $n >> d$.
  (classical asymptotic statistics)

Proportional Regime

$$R_X(\hat{\beta};\beta_0) = \beta^T \Pi\Sigma\Pi\beta + \frac{\sigma^2}{n}\,\mathrm{Tr}\left(\hat{\Sigma}^+\Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+\hat{\Sigma}$$

- This is not that insightful!
- How does $R_X(\beta_0, \hat{\beta})$ depend on sample size and dimension?
- This is well known in the regime where $n >> d$.
  (classical asymptotic statistics)
- What about the regime where d and n are of the same order?

Proportional Regime

$$R_X(\hat{\beta}; \beta_0) = \beta^T \Pi \Sigma \Pi \beta + \frac{\sigma^2}{n} \operatorname{Tr}\left(\hat{\Sigma}^+ \Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+ \hat{\Sigma}$$

- This is not that insightful!
- How does $R_X(\beta_0, \hat{\beta})$ depend on sample size and dimension?
- This is well known in the regime where $n >> d$. (classical asymptotic statistics)
- What about the regime where d and n are of the same order? Let

$$n \to \infty, \quad d \to \infty, \quad \frac{d}{n} \to \gamma.$$

[Tulino and Verdu, 2004], [Dobriban and Wager, 2015], [Hastie et al., 2020].

Computing the risk

$$R_X(\hat{\beta}; \beta_0) = \beta^T \Pi \Sigma \Pi \beta + \frac{\sigma^2}{n} \operatorname{Tr}\left(\hat{\Sigma}^+ \Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+ \hat{\Sigma}$$

Computing the risk

$$R_X(\hat{\beta};\beta_0) = \beta^T \Pi\Sigma\Pi\beta + \frac{\sigma^2}{n}\,\mathrm{Tr}\left(\hat{\Sigma}^+\Sigma\right) \qquad \Pi = I - \hat{\Sigma}^+\hat{\Sigma}$$

One can use the Marchenko–Pastur Theorem to compute this limit.

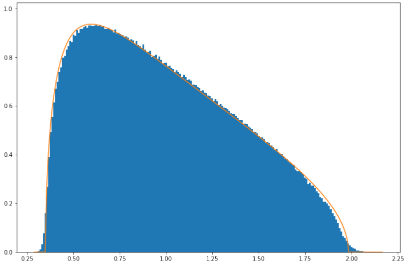


Figure: Histogram of the eigenvalues of $\hat{\Sigma}$ with $d/n \to \gamma$

Computing the risk

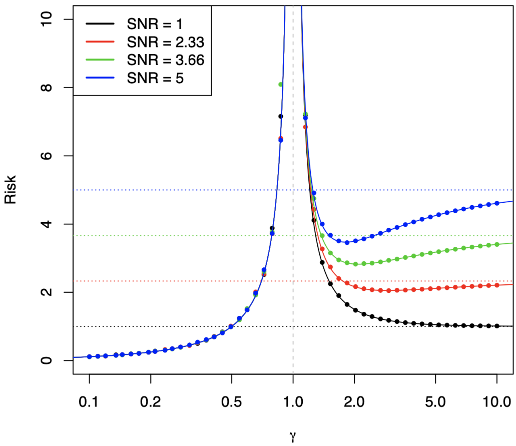
$$R_X(\hat{\beta};\beta_0) = \beta^T \Pi\Sigma\Pi\beta + \frac{\sigma^2}{n}\operatorname{Tr}\left(\hat{\Sigma}^{+}\Sigma\right) \rightarrow^{a.s.} \begin{cases} \sigma^2 \frac{\gamma}{1-\gamma} & \gamma < 1 \\ r^2(1-\frac{1}{\gamma}) + \frac{\sigma^2}{\gamma-1} & \gamma \geq 1 \end{cases}$$

where $d/n \rightarrow \gamma$ and $||\beta||^2 \rightarrow r^2$, where $\Sigma = I$ for simplicity.

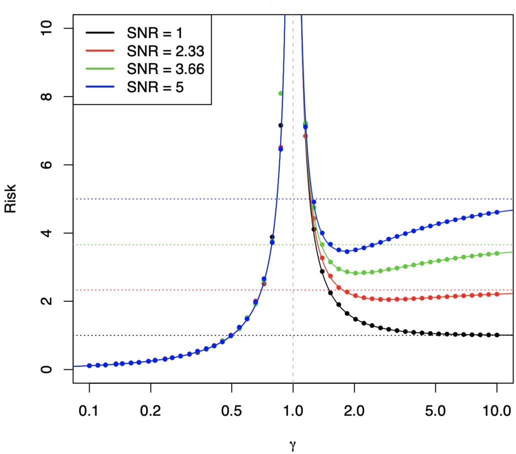
Computing the risk

$$R_X(\hat{\beta};\beta_0) = \beta^T \Pi\Sigma\Pi\beta + \frac{\sigma^2}{n}\operatorname{Tr}\left(\hat{\Sigma}^+\Sigma\right) \to^{a.s.} \begin{cases} \sigma^2 \frac{\gamma}{1-\gamma} & \gamma < 1 \\ r^2(1-\frac{1}{\gamma}) + \frac{\sigma^2}{\gamma-1} & \gamma \geq 1 \end{cases}$$

where $d/n \to \gamma$ and $||\beta||^2 \to r^2$, where $\Sigma = I$ for simplicity.

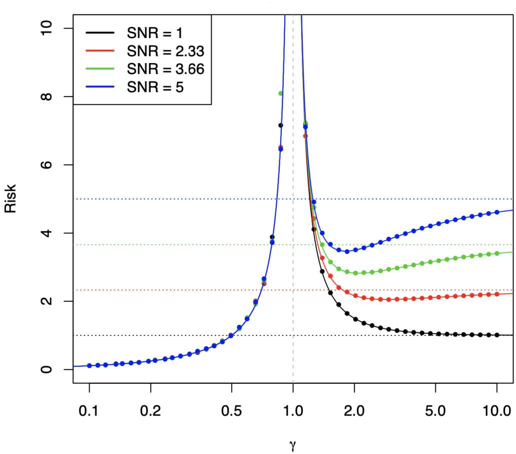


Computing the risk



This is good, but not double descent :)

Computing the risk



This is good, but not double descent :)

We need a mechanism to vary the overparameterization.

Random Features Regression

Random Features Regression

- Weight matrix $W \in \mathbb{R}^{N \times d}$ with i.i.d. random $N(0,1)$ entries, and activation function $\sigma : \mathbb{R} \to \mathbb{R}$:

$$F_W(x) = \sigma\left(\frac{1}{\sqrt{d}} W x\right) \in \mathbb{R}^N.$$

Random Features Regression

- Weight matrix $W \in \mathbb{R}^{N \times d}$ with i.i.d. random $N(0,1)$ entries, and activation function $\sigma : \mathbb{R} \to \mathbb{R}$:

$$F_W(x) = \sigma\left(\frac{1}{\sqrt{d}} W x\right) \in \mathbb{R}^N.$$

- The random features model is defined by

$$f_{W,a}(x) = \frac{1}{\sqrt{N}} a^\top F_W(x), \quad a \in \mathbb{R}^N.$$

Random Features Regression

- Weight matrix $W \in \mathbb{R}^{N \times d}$ with i.i.d. random $N(0,1)$ entries, and activation function $\sigma : \mathbb{R} \to \mathbb{R}$:

$$F_W(x) = \sigma\left(\frac{1}{\sqrt{d}} Wx\right) \in \mathbb{R}^N.$$

- The random features model is defined by

$$f_{W,a}(x) = \frac{1}{\sqrt{N}} a^\top F_W(x), \quad a \in \mathbb{R}^N.$$

- **Benefit**: variable capacity (N vs d parameters)

- Weight matrix $W \in \mathbb{R}^{N\times d}$ with i.i.d. random $N(0,1)$ entries, and activation function $\sigma : \mathbb{R} \to \mathbb{R}$:

$$F_W(x) = \sigma\left(\frac{1}{\sqrt{d}} Wx\right) \in \mathbb{R}^N.$$

- The random features model is defined by

$$f_{W,a}(x) = \frac{1}{\sqrt{N}} a^\top F_W(x), \quad a \in \mathbb{R}^N.$$

- **Benefit**: variable capacity (N vs d parameters)
- Neural network at early phase of training.

[Rahimi and Recht, 2007]

Training a RF Regression

- The random features model is defined by

$$f_{W,a}(x) = \frac{1}{\sqrt{N}} a^\top F_W(x), \quad a \in \mathbb{R}^N.$$

Training a RF Regression

- The random features model is defined by

$$f_{W,a}(x) = \frac{1}{\sqrt{N}} a^\top F_W(x), \quad a \in \mathbb{R}^N.$$

- We train it using ridge regularization:

$$\hat{a}_\lambda = \arg\min_{a \in \mathbb{R}^N} \left[\sum_{i=1}^{n} (y_i - f_{W,a}(x_i))^2 + \lambda \|a\|_2^2 \right]$$

Training a RF Regression

- The random features model is defined by

$$f_{W,a}(x) = \frac{1}{\sqrt{N}} a^\top F_W(x), \quad a \in \mathbb{R}^N.$$

- We train it using ridge regularization:

$$\hat{a}_\lambda = \arg\min_{a \in \mathbb{R}^N} \left[\sum_{i=1}^{n} (y_i - f_{W,a}(x_i))^2 + \lambda \|a\|_2^2 \right]$$

- **Proportional limit**:

$$n, N, d \to \infty, \quad \text{with } N/d \to \psi, \ n/d \to \phi.$$

[Mei and Montanari, 2019] and [Adlam and Pennington, 2020].

Universality Results

- **Input**: For linear regression and random features regression, the distribution of X can typically be replaced with a Gaussian with the same mean and covariance with no change.

Universality Results

- **Input**: For linear regression and random features regression, the distribution of X can typically be replaced with a Gaussian with the same mean and covariance with no change.
- **Nonlinearity**: In RF regression, we can replace

$$F_W(x) \approx \mu_1 Wx + \mu_2 \Theta.$$

where Θ is an independent Gaussian vector. Constants μ_1 and μ_2 are chosen to match the first and second moments.

Universality Results

- **Input**: For linear regression and random features regression, the distribution of X can typically be replaced with a Gaussian with the same mean and covariance with no change.
- **Nonlinearity**: In RF regression, we can replace

$$F_W(x) \approx \mu_1 W x + \mu_2 \Theta.$$

  where Θ is an independent Gaussian vector. Constants μ_1 and μ_2 are chosen to match the first and second moments.
- **Good or bad**?

Universality Results

- **Input**: For linear regression and random features regression, the distribution of X can typically be replaced with a Gaussian with the same mean and covariance with no change.
- **Nonlinearity**: In RF regression, we can replace

$$F_W(x) \approx \mu_1 W x + \mu_2 \Theta.$$

  where Θ is an independent Gaussian vector. Constants μ_1 and μ_2 are chosen to match the first and second moments.
- **Good or bad**? Can only learn a linear function; hence, set $y_i = \beta^\top x_i + \varepsilon_i$ as before.

Universality Results

- **Input**: For linear regression and random features regression, the distribution of X can typically be replaced with a Gaussian with the same mean and covariance with no change.
- **Nonlinearity**: In RF regression, we can replace

$$F_W(x) \approx \mu_1 W x + \mu_2 \Theta.$$

  where Θ is an independent Gaussian vector. Constants μ_1 and μ_2 are chosen to match the first and second moments.
- **Good or bad**? Can only learn a linear function; hence, set $y_i = \beta^\top x_i + \varepsilon_i$ as before.
- One gradient step on W? [Ba et al, 2022]

Universality Results

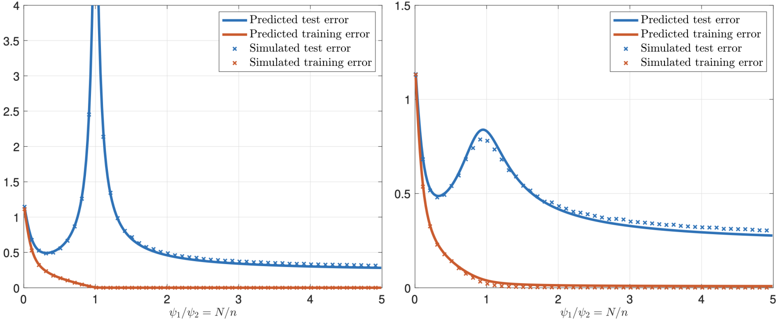
- **Input**: For linear regression and random features regression, the distribution of X can typically be replaced with a Gaussian with the same mean and covariance with no change.
- **Nonlinearity**: In RF regression, we can replace

$$F_W(x) \approx \mu_1 W x + \mu_2 \Theta.$$

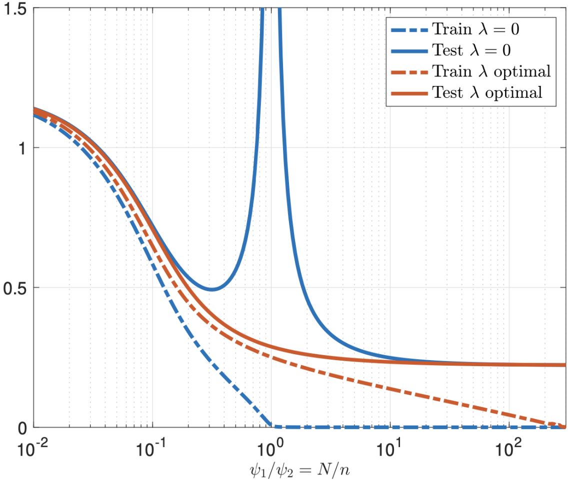
  where Θ is an independent Gaussian vector. Constants μ_1 and μ_2 are chosen to match the first and second moments.
- **Good or bad**? Can only learn a linear function; hence, set $y_i = \beta^\top x_i + \varepsilon_i$ as before.
- One gradient step on W? [Ba et al, 2022]

[Hu and Lu, 2020], [Mei and Montanari, 2020], [Hassani and Javanmard 2022], [Montanari and Saeed, 2022] and many others.

Double Descent!



$\lambda = 0+$ $\lambda = 3 \times 10^{-4}$


1.5
1
0.5
0
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
$\psi_1/\psi_2 = N/n$
Train $\lambda = 0$
Test $\lambda = 0$
Train λ optimal
Test λ optimal

Accuracy-on-the-line
and
Agreement-on-the-line

Distribution Shift

- Classic i.i.d. assumption between train/test:

$$P_{\text{train}}(x) = P_{\text{test}}(x), \quad P_{\text{train}}(y|x) = P_{\text{test}}(y|x)$$

Distribution Shift

- Classic i.i.d. assumption between train/test:

$$P_{\text{train}}(x) = P_{\text{test}}(x), \quad P_{\text{train}}(y|x) = P_{\text{test}}(y|x)$$

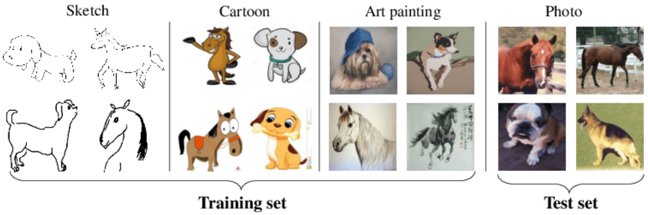
- Let's assume that this does not hold.

Distribution Shift

- Classic i.i.d. assumption between train/test:

$$P_{\text{train}}(x) = P_{\text{test}}(x), \quad P_{\text{train}}(y|x) = P_{\text{test}}(y|x)$$

- Let's assume that this does not hold.

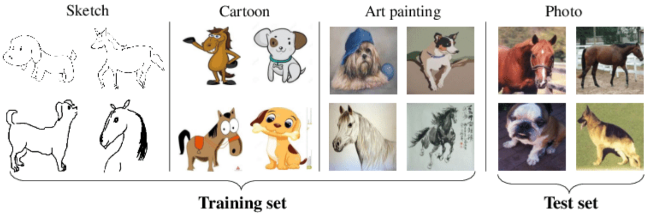


Distribution Shift

- Classic i.i.d. assumption between train/test:

$$P_{\text{train}}(x) = P_{\text{test}}(x), \quad P_{\text{train}}(y|x) = P_{\text{test}}(y|x)$$

- Let's assume that this does not hold.



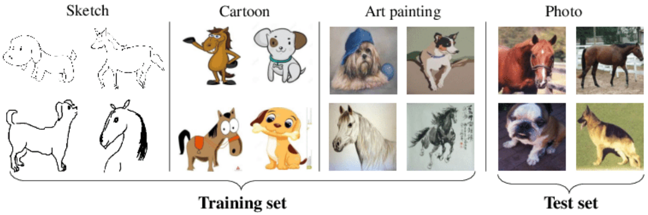
- How does our model perform in the test domain?

Distribution Shift

- Classic i.i.d. assumption between train/test:

$$P_{\text{train}}(x) = P_{\text{test}}(x), \quad P_{\text{train}}(y|x) = P_{\text{test}}(y|x)$$

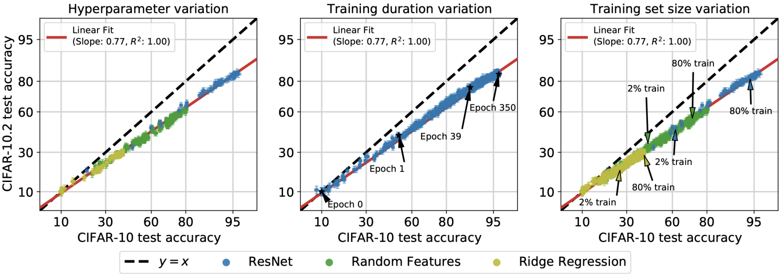
- Let's assume that this does not hold.



- How does our model perform in the test domain?
- Labeled data from test?

Observation 1: Accuracy-on-the-line

- Dating at least back to [Recht et al., 2019], we know that:



Accuracy-on-the-line in RF Regression

[Tripuraneni, Adlam, and Pennington, 2021] considered *covariate shift* in random features model.

Accuracy-on-the-line in RF Regression

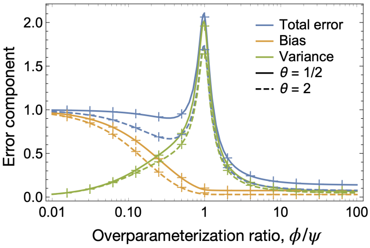
[Tripuraneni, Adlam, and Pennington, 2021] considered *covariate shift* in random features model.

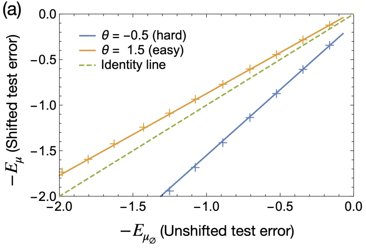
$$\textbf{Train}: x \sim \mathcal{N}(0, \Sigma_s) \to \textbf{Test}: x \sim \mathcal{N}(0, \Sigma_t)$$

Accuracy-on-the-line in RF Regression

[Tripuraneni, Adlam, and Pennington, 2021] considered *covariate shift* in random features model.

$$\textbf{Train}: x \sim \mathcal{N}(0, \Sigma_s) \to \textbf{Test}: x \sim \mathcal{N}(0, \Sigma_t)$$

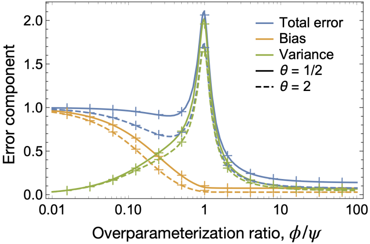


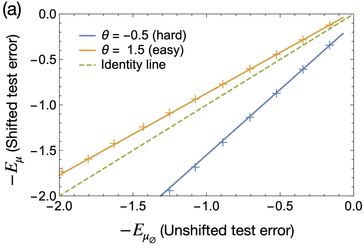


Accuracy-on-the-line in RF Regression

[Tripuraneni, Adlam, and Pennington, 2021] considered *covariate shift* in random features model.

$$\textbf{Train}: x \sim \mathcal{N}(0, \Sigma_s) \to \textbf{Test}: x \sim \mathcal{N}(0, \Sigma_t)$$





Does not necessarily hold for other shifts!

Observation 2: Agreement-on-the-line

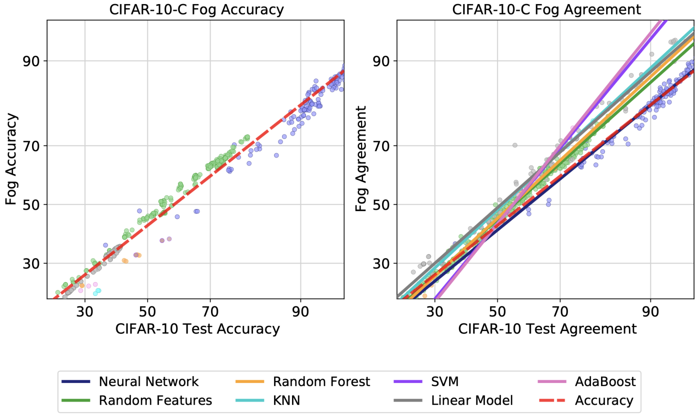
- Now back to the main question. Estimating test with only unlabeled data from test domain.

Observation 2: Agreement-on-the-line

- Now back to the main question. Estimating test with only unlabeled data from test domain.
- Recently, [Baek et al., 2022] suggested using (dis)agreement-on-the-line:

Observation 2: Agreement-on-the-line

- Now back to the main question. Estimating test with only unlabeled data from test domain.
- Recently, [Baek et al., 2022] suggested using (dis)agreement-on-the-line:



Types of (Dis)agreement?

Based on the type of randomness shared, we can define three non-trivial notions of disagreement:

- **I**ndependent:

$$\mathrm{Dis}_I = \mathbb{E}_{W_1,W_2,X_1,Y_1,X_2,Y_2,x}\left[\left(\hat{y}_{W_1,X_1,Y_1}(x) - \hat{y}_{W_2,X_2,Y_2}(x)\right)^2\right]$$

Types of (Dis)agreement?

Based on the type of randomness shared, we can define three non-trivial notions of disagreement:

- **I**ndependent:

$$\mathrm{Dis}_I = \mathbb{E}_{W_1,W_2,X_1,Y_1,X_2,Y_2,x}\left[\left(\hat{y}_{W_1,X_1,Y_1}(x) - \hat{y}_{W_2,X_2,Y_2}(x)\right)^2\right]$$

- **S**hared **S**ample:

$$\mathrm{Dis}_{SS} = \mathbb{E}_{W_1,W_2,X,Y,x}\left[\left(\hat{y}_{W_1,X,Y}(x) - \hat{y}_{W_2,X,Y}(x)\right)^2\right]$$

Types of (Dis)agreement?

Based on the type of randomness shared, we can define three non-trivial notions of disagreement:

- **I**ndependent:

$$\mathrm{Dis}_I = \mathbb{E}_{W_1,W_2,X_1,Y_1,X_2,Y_2,x}\left[\left(\hat{y}_{W_1,X_1,Y_1}(x) - \hat{y}_{W_2,X_2,Y_2}(x)\right)^2\right]$$

- **S**hared **S**ample:

$$\mathrm{Dis}_{SS} = \mathbb{E}_{W_1,W_2,X,Y,x}\left[\left(\hat{y}_{W_1,X,Y}(x) - \hat{y}_{W_2,X,Y}(x)\right)^2\right]$$

- **S**hared **W**eights:

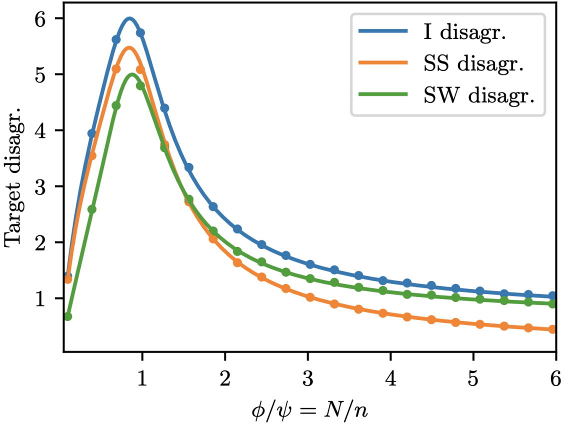
$$\mathrm{Dis}_{SW} = \mathbb{E}_{W,X_1,Y_1,X_2,Y_2,x}\left[\left(\hat{y}_{W,X_1,Y_1}(x) - \hat{y}_{W,X_2,Y_2}(x)\right)^2\right]$$

Asymptotics of Disagreement

We derive the asymptotics of disagreement in the proportional limit:

Asymptotics of Disagreement

We derive the asymptotics of disagreement in the proportional limit:



Agreement-on-the-line in RF Regression

Agreement-on-the-line is a nuanced phenomenon:

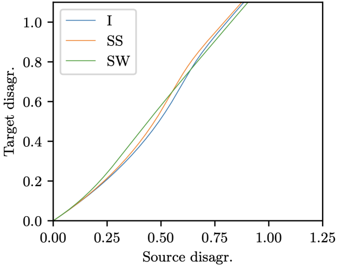
Agreement-on-the-line in RF Regression

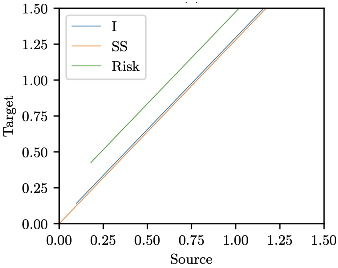
Agreement-on-the-line is a nuanced phenomenon:

- Overparameterized vs. Underparameterized (ridgeless):

Agreement-on-the-line is a nuanced phenomenon:

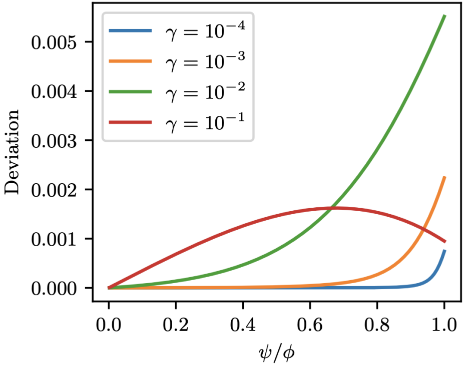
- Overparameterized vs. Underparameterized (ridgeless):





Agreement-on-the-line is a nuanced phenomenon.

- Non-zero Ridge:



Universality!

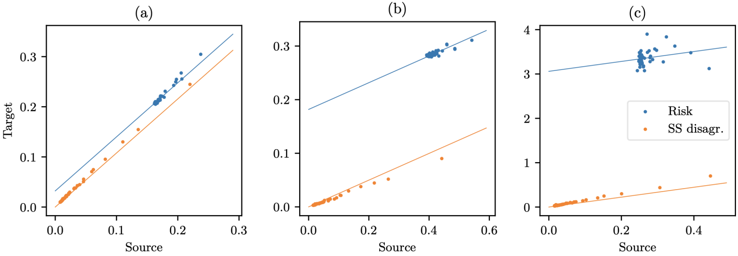


Figure 4: **(a)** CIFAR-10-C-Snow (severity 3) **(b)** Tiny ImageNet-C-Fog (severity 3) **(c)** Camelyon17;

References

1. Tulino and Verdu (2008).
   Random Matrix Theory and Wireless Communications,
   Foundations and Trends ® in Communications and Information Theory
2. Dobriban and Wager (2015).
   High-Dimensional Asymptotics of Prediction: Ridge Regression and Classification,
   The Annals of Statistics.
3. Belkin, Hsu, Ma, and Mandala (2018).
   Reconciling modern machine learning practice and the bias-variance trade-off,
   Proceedings of the National Academy of Sciences.
4. Hastie, Montanari, Rosset, and Tibshirani (2019).
   High-Dimensional Asymptotics of Prediction: Ridge Regression and Classification,
   The Annals of Statistics.
5. Mei and Montanari (2019).
   The Generalization Error of Random Features Regression: Precise asymptotics and the double descent curve,
   Communications on Pure and Applied Mathematics.

References

6. Recht, Roelofs, Schmidt, and Shankar (2019).
   Do ImageNet Classifiers Generalize to ImageNet?,
   International Conference on Machine Learning.
7. Hu and Lu (2020).
   Universality Laws for High-Dimensional Learning with Random Features,
   IEEE Transactions on Information Theory.
8. Tripuraneni, Adlam and Pennington (2021).
   Covariate Shift in High-Dimensional Random Feature Regression,
   Advances in Neural Information Processing Systems.
9. Montanari and Saeed (2022).
   Universality of Empirical Risk Minimization,
   Preprint.
10. Ba, Erdogdu, Suzuki, Wang, Wu, and Yang (2022).
   High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation,
   Preprint.

Frame Title

11. Hassani and Javanmard (2022).
The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression,
Preprint.

12. Baek, Jiang, Raghunathan, and Kolter (2022)
Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift,
Advances in Neural Information Processing Systems.

13. Lee, Moniri, Huang, Dobriban, and Hassani (2023)
Dimestifying Disagreement-on-the-Line in High Dimensions,
Preprint.

Thank You!