

Lecture 1/2 What is Machine Learning?

IEMS 402 Statistical Learning

Northwestern

Logistics

Logistics

- Course Website: <https://2prime.github.io/teaching/2025-Statistical-Learning>
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)
$$\max(HW1, HW8) + \max(HW2, HW3) + \max(HW4, HW5) + \max(HW6, HW7).$$

- [\[Homework 1\]](#) Review of Probability and Optimization Review of technical basic
- [\[Homework 2\]](#) Bias and Variance Trade-off 1
- [\[Homework 3\]](#) Bias and Variance Trade-off 2
- [\[Homework 4\]](#) Asymptotic Theory 1
- [\[Homework 5\]](#) Asymptotic Theory 2
- [\[Homework 6\]](#) Non-Asymptotic Theory 1
- [\[Homework 7\]](#) Non-Asymptotic Theory 2
- [\[Homework 8\]](#) Advanced Topics Advanced research in OR

- Latex and overleaf (not required)

Logistics

- Course Website: <https://2prime.github.io/teaching/2025-Statistical-Learning>
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)
$$\max(HW1, HW8) + \max(HW2, HW3) + \max(HW4, HW5) + \max(HW6, HW7).$$

- [\[Homework 1\]](#) Review of Probability and Optimization
- [\[Homework 2\]](#) Bias and Variance Trade-off 1
- [\[Homework 3\]](#) Bias and Variance Trade-off 2
- [\[Homework 4\]](#) Asymptotic Theory 1
- [\[Homework 5\]](#) Asymptotic Theory 2
- [\[Homework 6\]](#) Non-Asymptotic Theory 1
- [\[Homework 7\]](#) Non-Asymptotic Theory 2
- [\[Homework 8\]](#) Advanced Topics

Easy

Easy

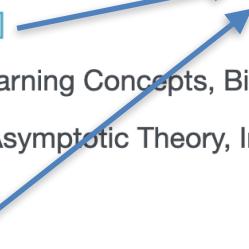
Easy

- Latex and overleaf (not required)

Logistics

- Course Website: <https://2prime.github.io/teaching/2025-Statistical-Learning>
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)

Exams

- [Practice Mid-Term Exam]   The same technique as the exam
 - Modern Machine Learning Concepts, Bias and Variance Trade-off
 - Kernel Smoothing, Asymptotic Theory, Influence Function Concentration Inequality, Uniform Bound
- [Practice Final Exam] 
 - Rademacher complexity, Covering Number, Dudley's theorem
 - RKHS, Optimal Transport, Robust Learning

Logistics

- Course Website: <https://2prime.github.io/teaching/2025-Statistical-Learning>
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)

The screenshot shows a LaTeX editor interface with a 'Code Editor' tab active. The code editor contains the following LaTeX code:

```
\documentclass[twoside]{article}
\setlength{\oddsidemargin}{0.25 in}
\setlength{\evensidemargin}{-0.25 in}
\setlength{\topmargin}{-0.6 in}
\setlength{\textwidth}{6.5 in}
\setlength{\textheight}{8.5 in}
\setlength{\headsep}{0.75 in}
\setlength{\parindent}{0 in}
\setlength{\parskip}{0.1 in}

% ADD PACKAGES here:
%
\usepackage{amsmath,amssymb,amsthm}
\usepackage{geometry}
\usepackage{hyperref}
\usepackage{bm}

%
\usepackage{amsmath,amsfonts,amssymb,graphicx,mathtools,flexisym}
\newtheorem{problem}{Problem}
%
% The following commands set up the lecnum (lecture number)
23 \% counter and make various numbering schemes work relative
24 \% to the lecture number.
25 %
26 \newcounter{lecnum}
27 \renewcommand{\thepage}{\thelcnum-\arabic{page}}
```

The right side of the interface shows a rendered PDF document titled 'IEMS 402: Statistical Learning' with the subtitle 'Lecture 15: Optimal Transport'. The PDF includes a disclaimer about the notes not being subject to formal scrutiny and being distributed with permission. It then lists two sections: '15.1 Introduction to Optimal Transport' and '15.2 Discrete Optimal Transport'. The '15.2' section includes a sub-section '15.2.1 Discrete Measures' with mathematical notation for probability measures μ and ν supported on finite sets.

Refine my note

Logistics

- Course Website: <https://2prime.github.io/teaching/2025-Statistical-Learning>
- Grading: Problem Sets (15%) + Exams (80%) + Scribe Note (5%)
- Textbook: Bach, Francis. Learning theory from first principles. MIT press, 2024.
 - https://www.di.ens.fr/~fbach/ltpf_book.pdf

Gradescope
Campuswire
ChatGPT Tutor!

Late Work Policy

- For your first late assignment within 12 hours after the deadline (as indicated on Gradescope), no point deductions.
- All subsequent assignments submitted within 12 hours after the deadline will convert to a zero at the end of semester.
- In all cases, work submitted 12 hours or more after the deadline will not be accepted.

Preliminary

Review Document:

<https://2prime.github.io/files/IEMS402/IEMS402ProbOptReview.pdf>

Calculus, Linear Algebra

IEMS 302 Probability Probability and Statistics: Strong Law of Large Numbers, Central Limit Theorem, Big-O, little-o notation,

Optimization Theory: **Lagrangian Duality Theory IEMS 450-2: Mathematical Optimization II**
(Interestingly, IEMS 450-1 is not required)

You **need** to know

Law of strong numbers, Central Limit Theorem, Continuous Map Theorem, Slutsky Theorem, Markov's Inequality

You **don't need** to distinguish Convergence in Probability/Covergence in distribution, you just need to write →

Online Calibration with Human Feedback

问题 回复 设置

Feedback for IEMS402 Lecture 2

This feedback will help calibrate future lectures. Feel free to answer any subset of the questions (it is encouraged to at least answer the first question on pace).

The pace of material was

1 2 3 4 5
Much too slow Much too fast

What parts were confusing?

详答文本

What was most surprising/interesting?

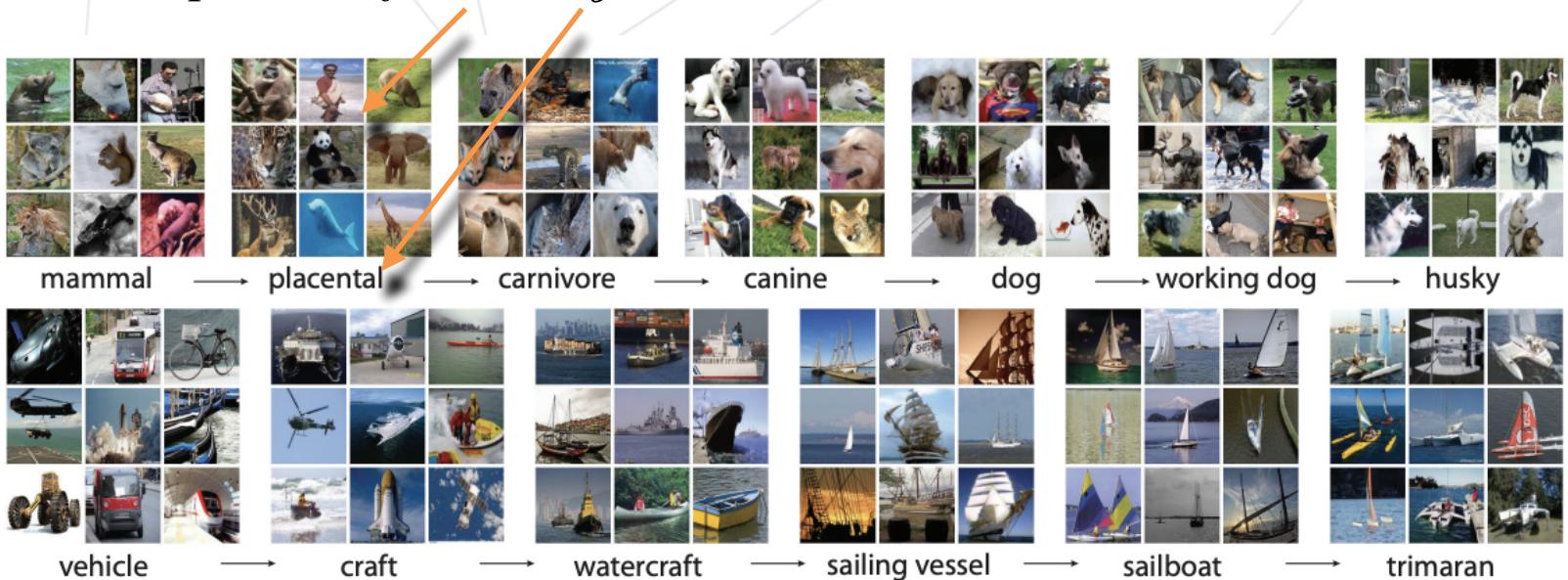
详答文本

Feedback for each lecture

Supervised Learning

Supervised Learning

- Aim: learn a predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$



Supervised Learning

- Aim: learn a predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$
- What is a good predictor? -> evaluation criteria

$$\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))] = \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, f(x)) dp(x, y).$$

Evaluate the error of label and prediction

Supervised Learning

- Aim: learn a predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$
- What is a good predictor? -> evaluation criteria

$$\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))] = \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, f(x)) dp(x, y).$$

Evaluate the error of label and prediction



If I want to know the risk, I need to have all the data in the univers?

Empirical Risk: $\hat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i))$, where $\{(x_i, y_i)\}_{i=1}^n$ is a collected dataset

Conditional Risk

$$\mathcal{R}(f) = \mathbb{E}_{x' \sim p} \left[\mathbb{E} \left[\ell(y, f(x')) \mid x = x' \right] \right] = \int_{\mathcal{X}} \underbrace{\mathbb{E} \left[\ell(y, f(x')) \mid x = x' \right]}_{\text{Conditional Risk: } r(z \mid x')} dp(x') .$$

$$\text{Conditional Risk: } r(z \mid x') = \mathbb{E} \left[\ell(y, z) \mid x = x' \right]$$

- Bayes Predictor: $f^*(x') \in \arg \min_{z \in \mathcal{Y}} \mathbb{E} \left[\ell(y, z) \mid x = x' \right] = \arg \min_{z \in \mathcal{Y}} r(z \mid x') .$

* means the best

Conditional Risk

$$\mathcal{R}(f) = \mathbb{E}_{x' \sim p} \left[\mathbb{E} \left[\ell(y, f(x')) \mid x = x' \right] \right] = \int_{\mathcal{X}} \underbrace{\mathbb{E} \left[\ell(y, f(x')) \mid x = x' \right]}_{\text{Conditional Risk: } r(z \mid x')} dp(x').$$

- Bayes Predictor: $f^*(x') \in \arg \min_{z \in \mathcal{Y}} \mathbb{E} \left[\ell(y, z) \mid x = x' \right] = \arg \min_{z \in \mathcal{Y}} r(z \mid x')$.



What is the Bayes Predictor of ℓ_2 loss or ℓ_1 loss?

How to design a loss function

- Method 1: Know what is your Bayes Predictor! [Homework 1 Question 1.](#)

How to design a loss function

- Method 1: Know what is your Bayes Predictor! [Homework 1 Question 1.](#)
- Method 2: Use Max likelihood
 - Step 1: understand what is your $p(y|x)$, e.g. Gaussian, heavy tail distribution
 - Step 2: What is the log-likelihood of dataset $\{(x_i, y_i)\}_{i=1}^n$?

How to design a loss function

- Method 1: Know what is your Bayes Predictor! [Homework 1 Question 1.](#)
- Method 2: Use Max likelihood
 - Step 1: understand what is your $p(y|x)$, e.g. Gaussian, heavy tail distribution
 - Step 2: What is the log-likelihood of dataset $\{(x_i, y_i)\}_{i=1}^n$?
 - $\log \prod_{i=1}^n p(y_i|x_i) = \sum_{i=1}^n \log p(y_i|x_i)$
 - Step 3: use $\log p(\cdot|x_i)$ as your loss function!



How can I get the ℓ_2 loss using this methods?

Example: Logistic Regression

Consider a binary classification with $p(y_i = 1 \mid \mathbf{x}_i, \theta) = \sigma(\mathbf{x}_i^\top \theta) = \frac{1}{1 + e^{-\mathbf{x}_i^\top \theta}}$

Example: Gaussian with Learned Variance

Example (*Gaussian with Learned Variance Leads to Sparsity*)

Not Required

$$\begin{aligned}\ell(\mu, \sigma^2) &= \sum_{i=1}^n \log P(y_i | \mu(x_i), \sigma(x_i)^2) \\ &= \sum_{i=1}^n \left(-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\sigma(x_i)^2) - \frac{(y_i - \mu(x_i))^2}{2\sigma(x_i)^2} \right) \\ &= -\frac{n}{2} \ln(2\pi(x_i)) - \underbrace{\frac{n}{2} \ln(\sigma(x_i)^2)}_{\text{sparse regularization}} - \underbrace{\sum_{i=1}^n \frac{(y_i - \mu(x_i))^2}{2\sigma(x_i)^2}}_{\text{weighted } \ell_2 \text{ loss}}\end{aligned}$$

Empirical Risk Minimization



I want an estimator to minimize the risk, but I can only get the empirical risk? What's the best thing I can do?

- Consider a parameterized family of prediction functions (often referred to as models) $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, e.g.
 - Linear prediction
 - Neural Network
- Empirical Risk Minimization: $\hat{\theta} \in \hat{\mathcal{R}}(f_\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i))$.

$$\hat{\mathcal{R}}(f_\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i))$$

^ means empirical

The only theorem: Risk Decomposition

$$\mathcal{R}(f_{\hat{\theta}}) - \mathcal{R}^* = \left\{ \mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) \right\} + \left\{ \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - \mathcal{R}^* \right\}$$


Estimation error


Approximation error

The only theorem: Risk Decomposition

$$\mathcal{R}(f_{\hat{\theta}}) - \mathcal{R}^* = \left\{ \mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) \right\} + \left\{ \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - \mathcal{R}^* \right\}$$

Estimation error Approximation error

For an ERM Estimator: ||

$$\mathcal{R}(f_{\hat{\theta}}) - \hat{\mathcal{R}}(f_{\hat{\theta}}) + \hat{\mathcal{R}}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \hat{\mathcal{R}}(f_{\theta'}) + \inf_{\theta' \in \Theta} \hat{\mathcal{R}}(f_{\theta'}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'})$$

Generalization error Optimization error Generalization error

The only theorem: Risk Decomposition

$$\mathcal{R}(f_{\hat{\theta}}) - \mathcal{R}^* = \left\{ \mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) \right\} + \left\{ \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - \mathcal{R}^* \right\}$$

Estimation error Approximation error

For an ERM Estimator: $\|$

$$\mathcal{R}(f_{\hat{\theta}}) - \hat{R}(f_{\hat{\theta}}) + \hat{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \hat{\mathcal{R}}(f_{\theta'}) + \inf_{\theta' \in \Theta} \hat{\mathcal{R}}(f_{\theta'}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'})$$

Generalization error Optimization error Generalization error

$\leq 2 \sup_{\theta \in \Theta} |R(f_{\theta}) - \hat{R}(f_{\hat{\theta}})|$ **Uniform Bound!**

Pro and Con of ERM

- Pro:
 - Flexible
 - Algorithms are available (e.g. SGD)
- Con:
 - can be relatively hard to optimize when the optimization formulation is not convex (e.g., neural networks);
 - the dependence on parameters can be complex (e.g., neural networks);
 - need some capacity control to avoid overfitting

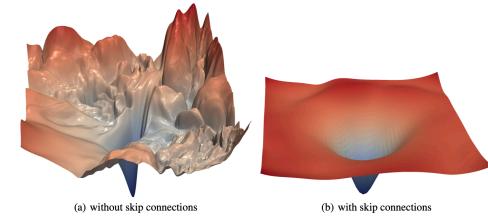


Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Our course is about overfitting!

Difference between 401 and 402

Statistics

Learning

- Difference 1: Parameter Convergence and Risk Convergence
- Difference 2: Parametric and Non-parametric



You use a parameterized family in Empirical risk minimization, why you call “non-parametric”?

Hardness of ERM

Error of ERM

IEMS 402 Focus

Assume to be 0

Approximation Error + Generalization Error + Optimization Error

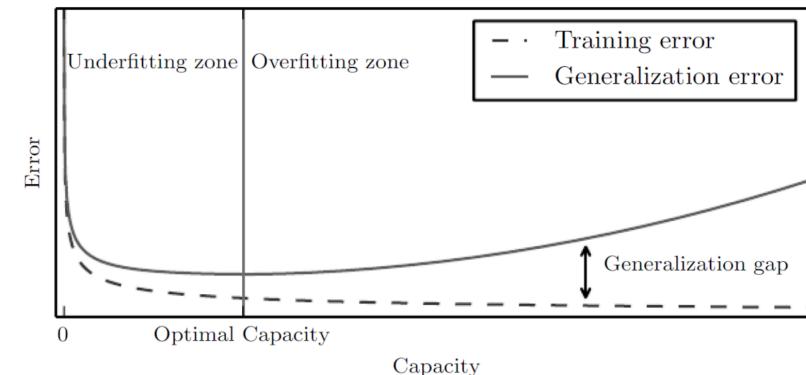
$$\inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - R^*$$

$$\sup_{\theta \in \Theta} |R(f_{\theta}) - \hat{R}(f_{\hat{\theta}})|$$

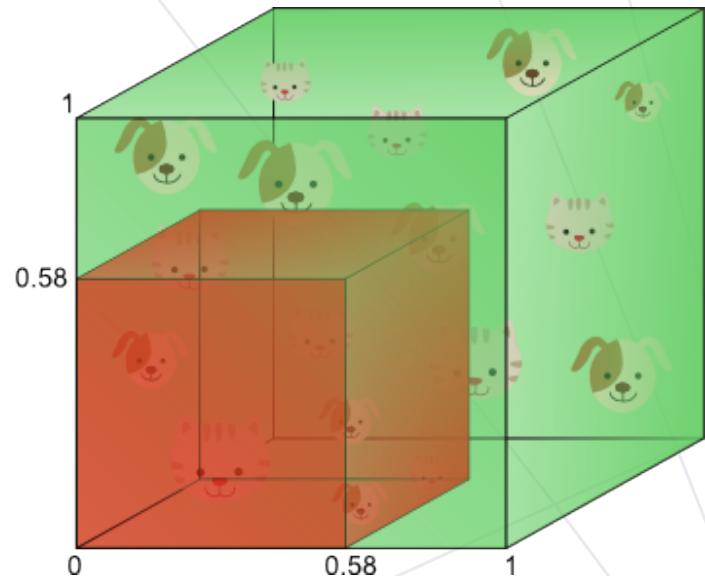
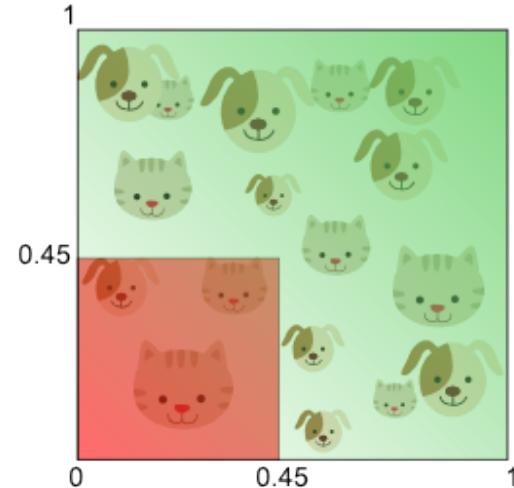
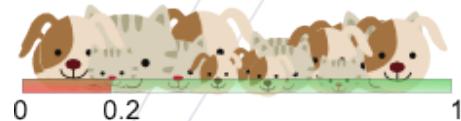
When we use more powerful parameterized family, e.g. Θ is larger:

- Approximation error is smaller!
- Generalization error is larger!

Bias-Variance Trade-off



Approximation: Curse of Dimensionality



Formulation: Approximate a smooth function

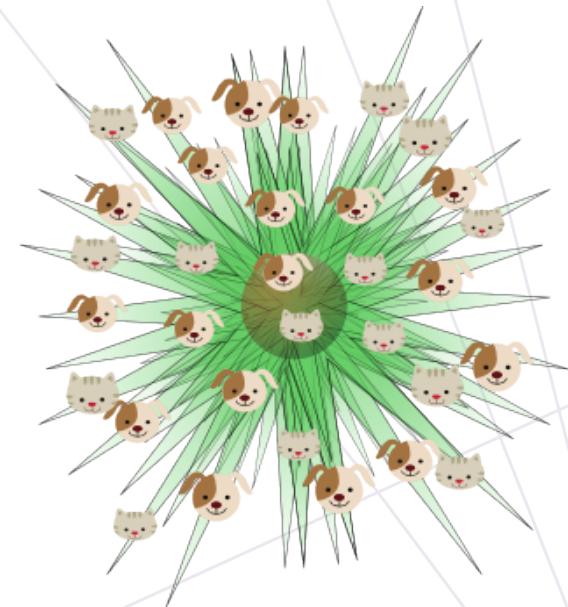
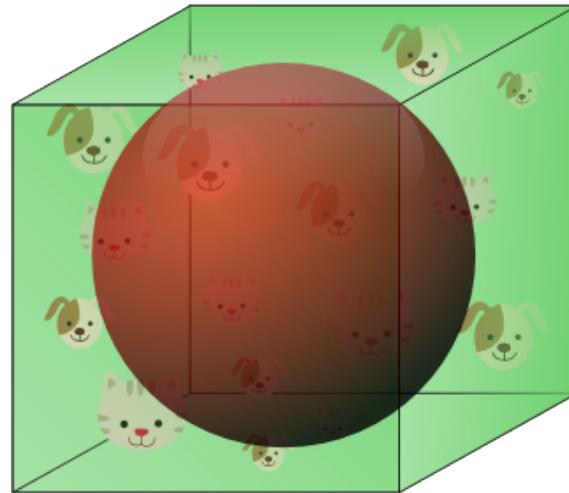
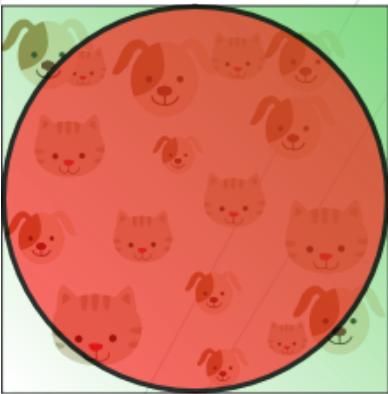
Fact. The number of parameters N required to achieve an approximation error of at most ϵ can be estimated by:

$$N \approx \left(\frac{1}{\epsilon} \right)^{\frac{d}{s}}$$

Dimension
smoothness

- Another Formulation see [Homework 1 Question 3.](#)

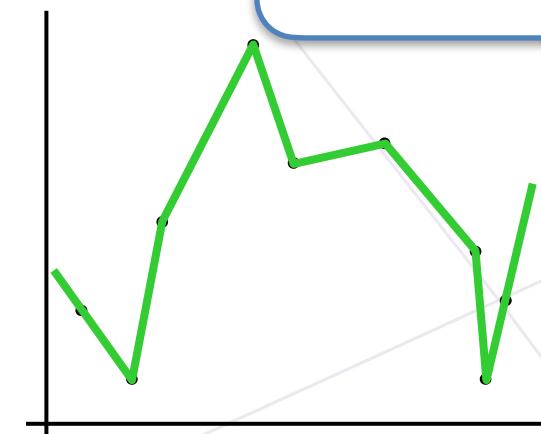
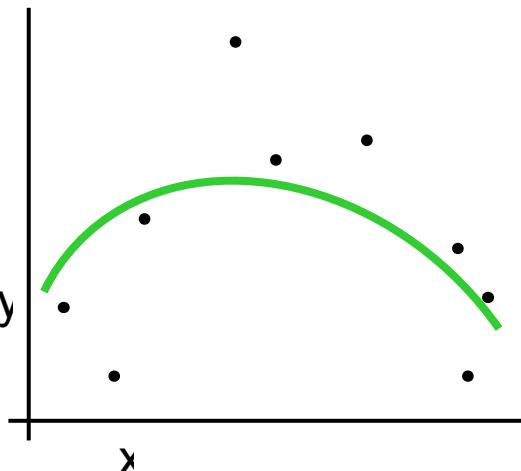
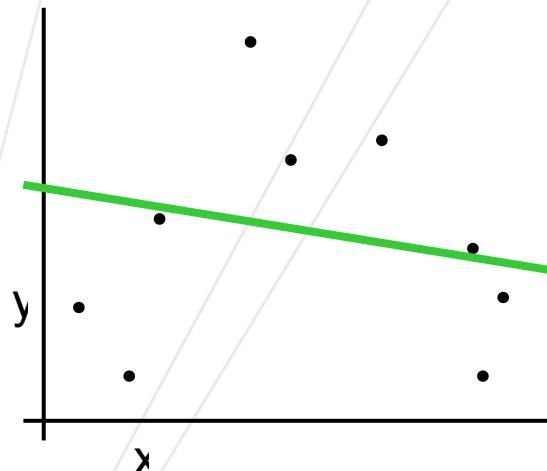
How to think about High Dimension



Generalization: Overfitting?

$$y = f(x) + \text{noise}$$

Can we learn f from this data?



Repeated Parrot
vs
understanding

Degree of Freedom

Suppose that we observe $y_i = r(x_i) + \epsilon_i (i = 1, \dots, n)$, where the errors ϵ_i are uncorrelated with common variance $\sigma^2 > 0$

Now consider the fitted values $\hat{y}_i = \hat{r}(x_i)$ from a regression estimator \hat{r} .

Degree of freedom is defined as
$$df(\hat{y}) = \frac{1}{\sigma^2} \sum_{i=1}^n \text{Cov}(\hat{y}_i, y_i).$$

“How much I remember the label”

Degree of freedom

Fact. $\mathbb{E} \left[\frac{1}{n} \sum_{i=1}^n (y'_i - \hat{y}_i)^2 \right] - \mathbb{E} \left[\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right] = \frac{2\sigma^2}{n} \text{df}(\hat{y}).$


Generalization error

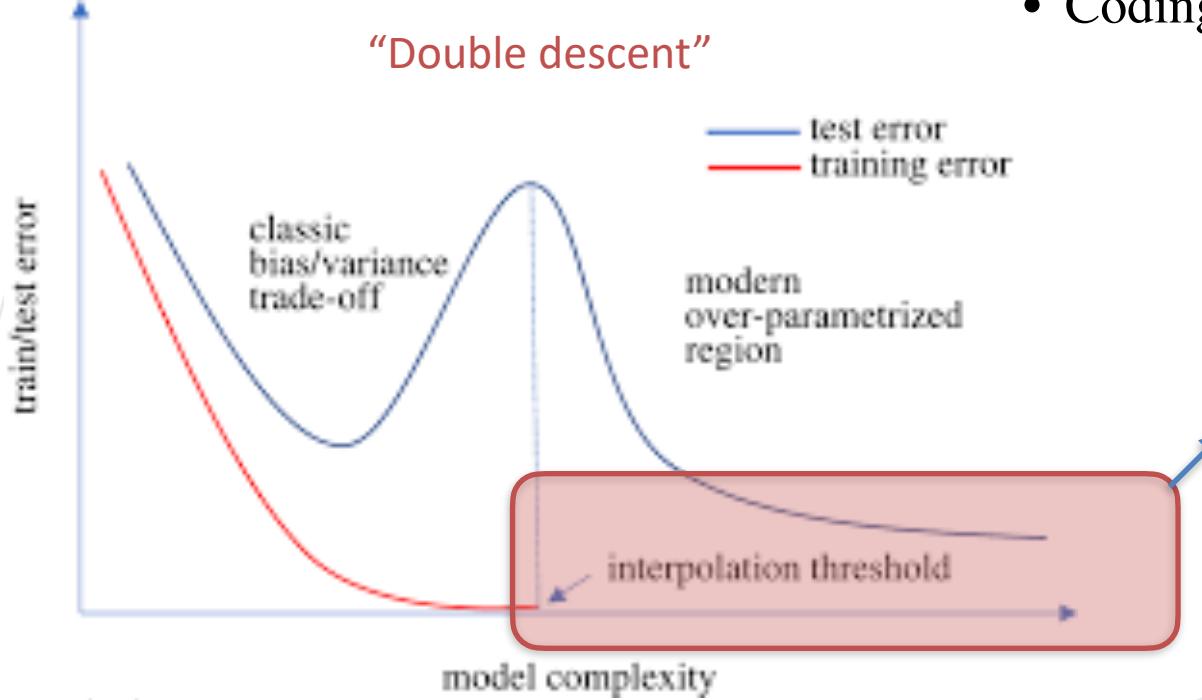
Example of DOF 1

Example of DOF 2

Not Required

However...

- Coding: [Homework 2 Question 3.](#)

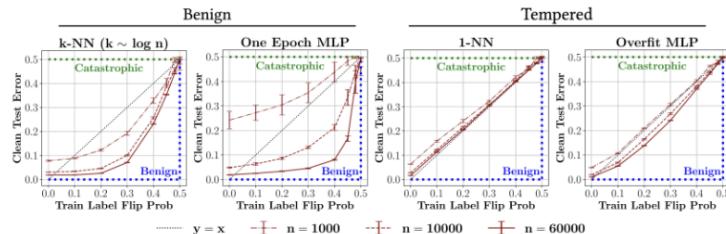
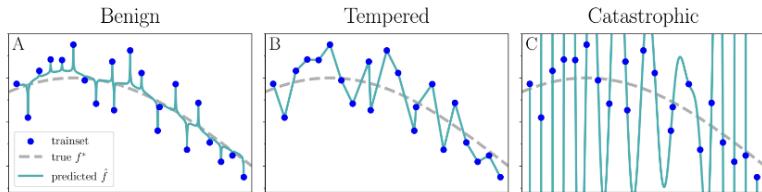


All the data can be remembered
 $\#parameter > \#data$

Taxonomy of (over)fitting

	Regression	Classification
Benign	$\lim_{n \rightarrow \infty} \mathcal{R}_n = R^*$	$\lim_{n \rightarrow \infty} \mathcal{R}_n = R^*$
Tempered	$\lim_{n \rightarrow \infty} \mathcal{R}_n \in (R^*, \infty)$	$\lim_{n \rightarrow \infty} \mathcal{R}_n \in (R^*, 1 - \frac{1}{K})$
Catastrophic	$\lim_{n \rightarrow \infty} \mathcal{R}_n = \infty$	$\lim_{n \rightarrow \infty} \mathcal{R}_n = 1 - \frac{1}{K}$

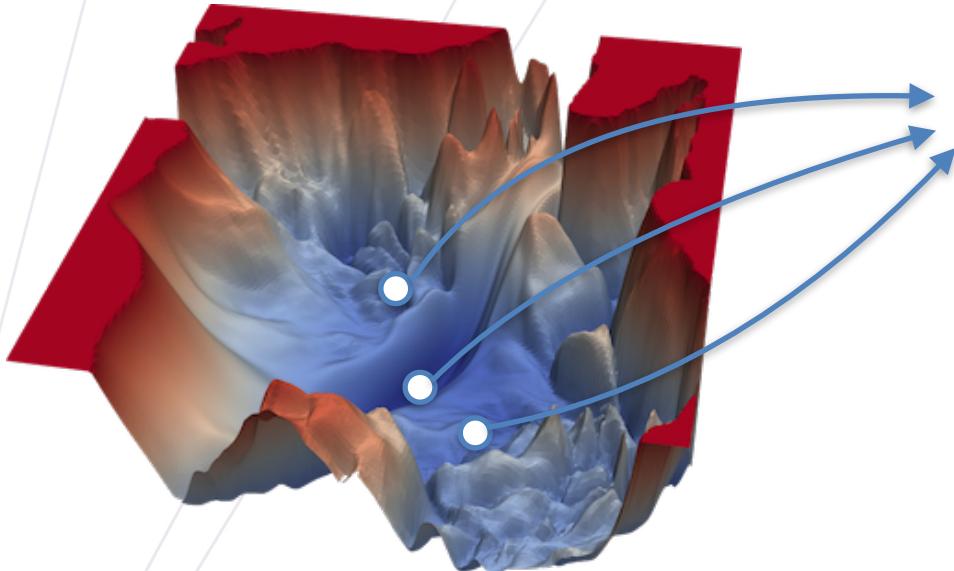
Table -1.1: Taxonomy of (over)fitting.



Mallinar, Neil, et al. "Benign, tempered, or catastrophic: A taxonomy of overfitting (2022)." arXiv preprint arXiv:2207.06569.

Implicit bias

“Multiple Minima”

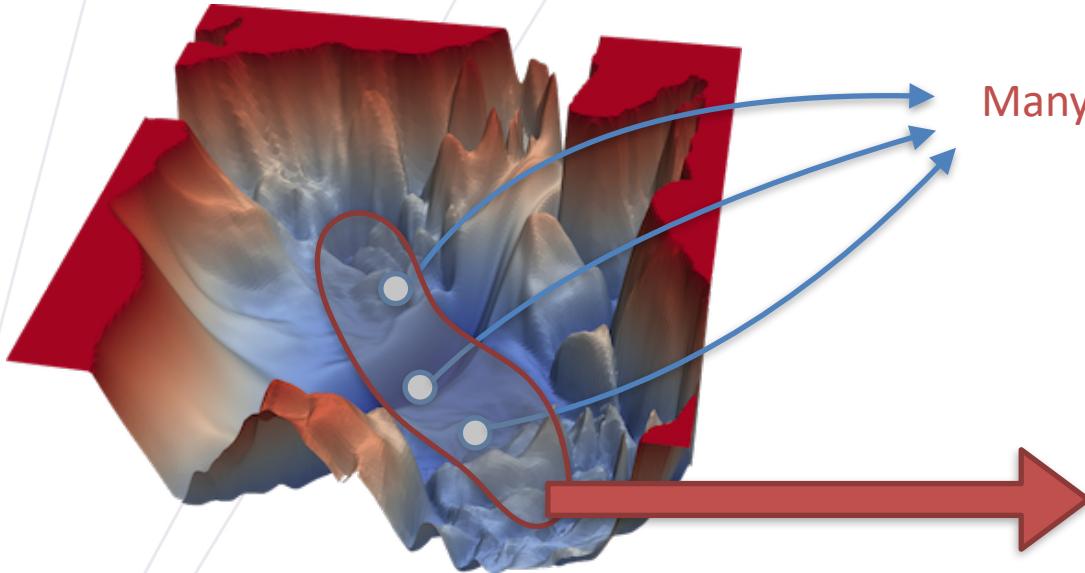


Many models can achieve low training loss

Loss landscape of VGG on CIFAR

Implicit bias

“Multiple Minima”



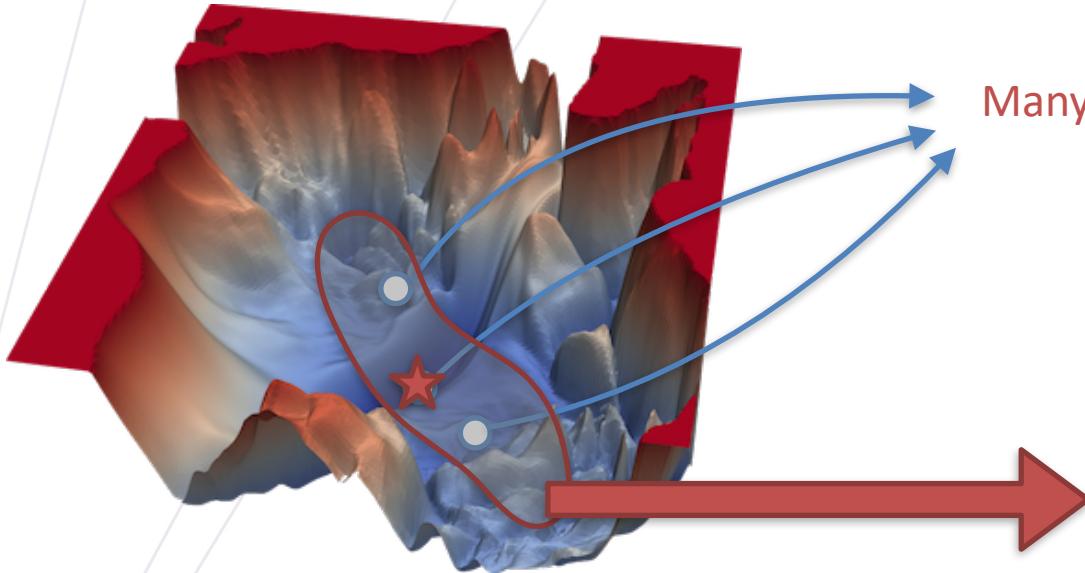
Loss landscape of VGG on CIFAR

Traditional bounds:

$$\sup_{\theta \in \Theta} |R(f_\theta) - \hat{R}(f_{\hat{\theta}})|$$

Implicit bias

“Multiple Minima”



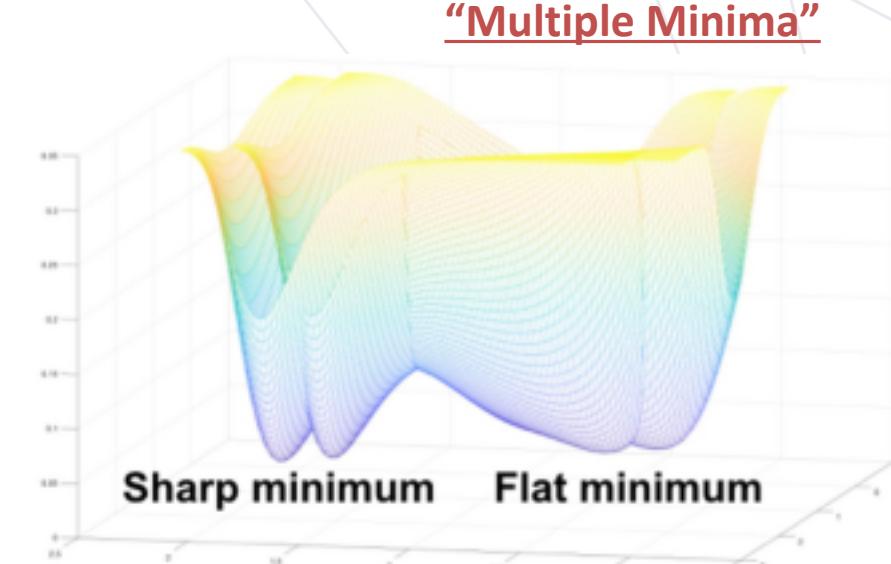
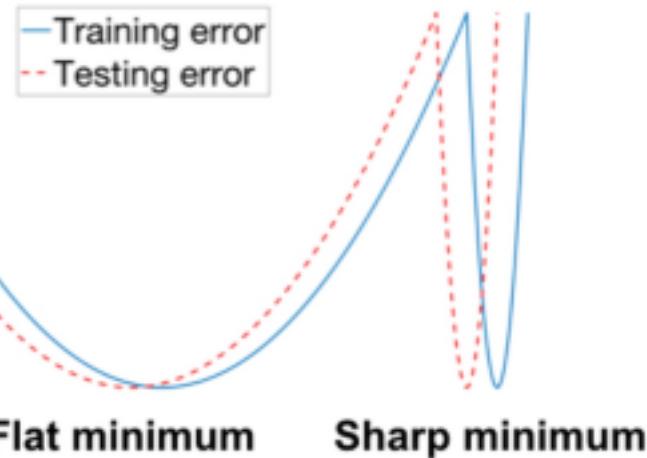
Loss landscape of VGG on CIFAR

Many models can achieve low training loss

Traditional bounds:

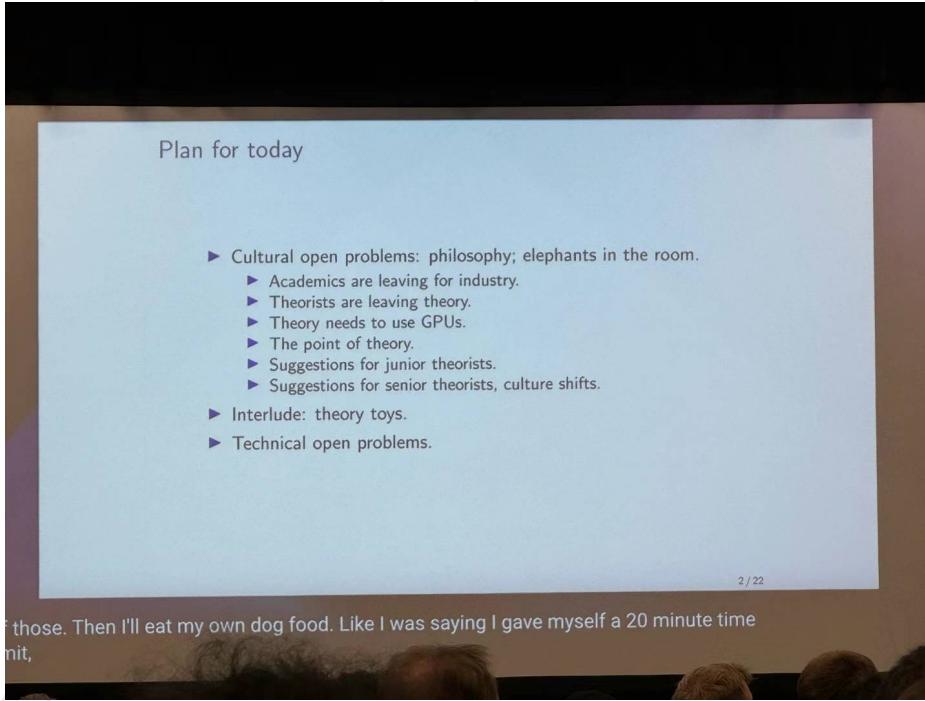
$$\sup_{\theta \in \Theta} |R(f_\theta) - \hat{R}(f_{\hat{\theta}})|$$

What's special about over-para



Last Note on Learning Theory

ML Theory workshop @Neurips24



[https://cims.nyu.edu/~matus/
neurips.2024.workshop/talk.pdf](https://cims.nyu.edu/~matus/neurips.2024.workshop/talk.pdf)