Statistical Learning Theory

Machine Learning Summer School, Kyoto, Japan

Alexander (Sasha) Rakhlin

University of Pennsylvania, The Wharton School
Penn Research in Machine Learning (PRiML)

August 27-28, 2012

References

Parts of these lectures are based on

- O. Bousquet, S. Boucheron, G. Lugosi:
  "Introduction to Statistical Learning Theory", 2004.
- MLSS notes by O. Bousquet
- S. Mendelson: "A Few Notes on Statistical Learning Theory"
- Lecture notes by S. Shalev-Shwartz
- Lecture notes (S. R. and K. Sridharan)
  http://stat.wharton.upenn.edu/~rakhlin/courses/stat928/stat928_notes.pdf

Prerequisites: a basic familiarity with Probability is assumed.

Outline

Introduction

Statistical Learning Theory

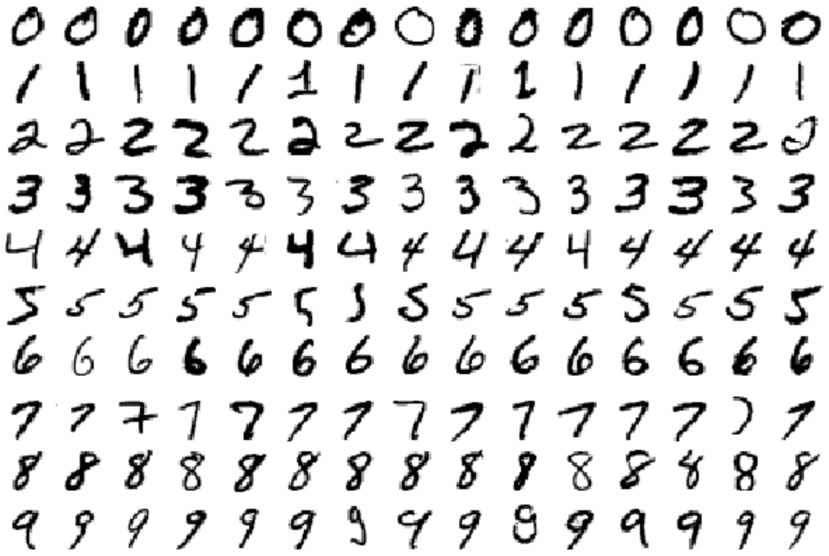
- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Example #1: Handwritten Digit Recognition

Imagine you are asked to write a computer program that recognizes postal codes on envelopes. You observe the huge amount of variation and ambiguity in the data:

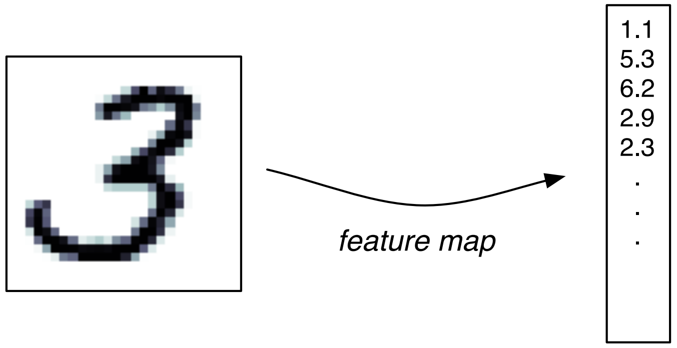


One can try to hard-code all the possibilities, but likely to fail. It would be nice if a program looked at a large corpus of data and learned the distinctions!

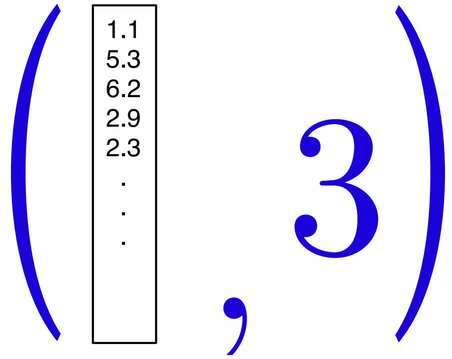
This picture of MNIST dataset was yanked from http://www.heikohoffmann.de/htmlthesis/node144.html

Example #1: Handwritten Digit Recognition

Need to represent data in the computer. Pixel intensities is one possibility, but not necessarily the best one. Feature representation:



We also need to specify the “label” of this example: “3”. The *labeled example* is then

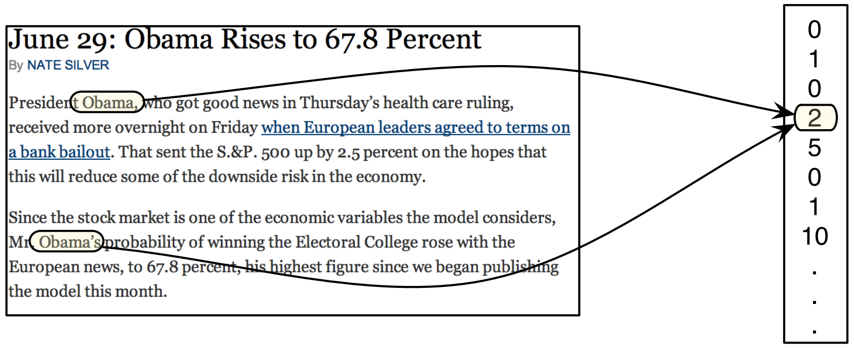


After looking at many of these examples, we want the program to *predict* the label of the next hand-written digit.

Example #2: Predict Topic of a News Article

You would like to automatically collect news stories from the web and display them to the reader in the best possible way. You would like to group or filter these articles by *topic*. Hard-coding possible topics for articles is a daunting task!

Representation in the computer:



June 29: Obama Rises to 67.8 Percent

By NATE SILVER

President Obama, who got good news in Thursday's health care ruling, received more overnight on Friday when European leaders agreed to terms on a bank bailout. That sent the S.&P. 500 up by 2.5 percent on the hopes that this will reduce some of the downside risk in the economy.

Since the stock market is one of the economic variables the model considers, Mr. Obama's probability of winning the Electoral College rose with the European news, to 67.8 percent, his highest figure since we began publishing the model this month.

This is a bag-of-words representation. If "1" stands for the category "politics", then this example can be represented as

$$\left(\boxed{\begin{array}{c}0\\1\\0\\2\\5\\0\\1\\10\\.\\.\\.\end{array}}\,,1\right)$$

After looking at many of such examples, we would like the program to predict the topic of a new article.

Why Machine Learning?

- Impossible to hard-code all the knowledge into a computer program.
- The systems need to be adaptive to the changes in the environment.

Examples:

- Computer vision: face detection, face recognition
- Audio: voice recognition, parsing
- Text: document topics, translation
- Ad placement on web pages
- Movie recommendations
- Email spam detection

Machine Learning

(Human) learning is the process of acquiring knowledge or skill.

Quite vague. How can we build a mathematical theory for something so imprecise?

Machine Learning is concerned with the design and analysis of algorithms that improve performance after observing data.

That is, the acquired knowledge comes from data.

We need to make mathematically precise the following terms: *performance*, *improve*, *data*.

Learning from Examples

How is it possible to conclude something general from specific examples?

Learning is inherently an ill-posed problem, as there are many alternatives that could be consistent with the observed examples.

Learning can be seen as the process of *induction* (as opposed to *deduction*): “extrapolating” from examples.

Prior knowledge is how we make the problem well-posed.

Memorization is not learning, not induction. Our theory should make this apparent.

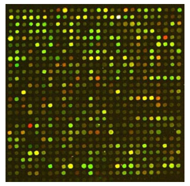
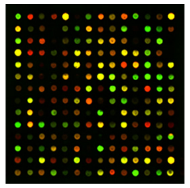
Very important to delineate assumptions. Then we will be able to prove mathematically that certain learning algorithms perform well.

Data

Space of inputs (or, predictors): $\mathcal{X}$

- ▷ e.g. $x \in \mathcal{X} \subset \{0, 1, \dots, 2^{16}\}^{64}$ is a string of pixel intensities in an 8×8 image.
- ▷ e.g. $x \in \mathcal{X} \subset \mathbb{R}^{33,000}$ is a set of gene expression levels.

$x_1 =$ $x_2 =$ $\dots$

$x_1 =$  $x_2 =$  $\dots$

$$x_1 = \begin{bmatrix} 5 \\ 1 \\ 22 \\ \vdots \end{bmatrix} \qquad x_2 = \begin{bmatrix} 1 \\ 0 \\ 17 \\ \vdots \end{bmatrix}$$

# cigarettes/day

# drinks/day

BMI

Data

Sometimes the space $\mathcal{X}$ is uniquely defined for the problem. In other cases, such as in vision/text/audio applications, many possibilities exist, and a good feature representation is key to obtaining good performance.

This important part of machine learning applications will not be discussed in this lecture, and we will assume that $\mathcal{X}$ has been chosen by the practitioner.

Data

Space of outputs (or, responses): $\mathcal{Y}$

▷ e.g. $y \in \mathcal{Y} = \{0, 1\}$ is a binary label ($1 =$ "cat")

▷ e.g. $y \in \mathcal{Y} = [0, 200]$ is life expectancy

A pair (x, y) is a *labeled* example.

▷ e.g. (x, y) is an example of an image with a label $y = 1$, which stands for the presence of a face in the image x

Dataset (or *training data*): examples $\{(x_1, y_1), \dots, (x_n, y_n)\}$

▷ e.g. a collection of images labeled according to the presence or absence of a face

The Multitude of Learning Frameworks

Presence/absence of labeled data:

- Supervised Learning: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Unsupervised Learning: $\{x_1, \ldots, x_n\}$
- Semi-supervised Learning: a mix of the above

This distinction is important, as labels are often difficult or expensive to obtain (e.g. can collect a large corpus of emails, but which ones are spam?)

Types of labels:

- Binary Classification / Pattern Recognition: $\mathcal{Y} = \{0, 1\}$
- Multiclass: $\mathcal{Y} = \{0, \ldots, K\}$
- Regression: $\mathcal{Y} \subseteq \mathbb{R}$
- Structure prediction: $\mathcal{Y}$ is a set of complex objects (graphs, translations)

The Multitude of Learning Frameworks

Problems also differ in the protocol for obtaining data:

- Passive
- Active

and in assumptions on data:

- Batch (typically i.i.d.)
- Online (i.i.d. or worst-case or some stochastic process)

Even more involved: Reinforcement Learning and other frameworks.

Why Theory?

> *"... theory is the first term in the Taylor series of practice"*
> *– Thomas M. Cover, "1990 Shannon Lecture"*

Theory and Practice should go hand-in-hand.

Boosting, Support Vector Machines – came from theoretical considerations.

Sometimes, theory is suggesting practical methods, sometimes practice comes ahead and theory tries to catch up and explain the performance.

This tutorial

First 2/3 of the tutorial: we will study the problem of *supervised learning* (with a focus on binary classification) with an i.i.d. assumption on the data.

The last 1/3 of the tutorial: we will turn to online learning without the i.i.d. assumption.

Outline

- Introduction

- Statistical Learning Theory
  - The Setting of SLT
  - Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
  - Tools from Probability, Empirical Processes
  - From Finite to Infinite Classes
  - Uniform Convergence, Symmetrization, and Rademacher Complexity
  - Large Margin Theory for Classification
  - Properties of Rademacher Complexity
  - Covering Numbers and Scale-Sensitive Dimensions
  - Faster Rates
  - Model Selection

- Sequential Prediction / Online Learning
  - Motivation
  - Supervised Learning
  - Online Convex and Linear Optimization
  - Online-to-Batch Conversion, SVM optimization

Outline

- Introduction

- Statistical Learning Theory
  - The Setting of SLT
  - Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
  - Tools from Probability, Empirical Processes
  - From Finite to Infinite Classes
  - Uniform Convergence, Symmetrization, and Rademacher Complexity
  - Large Margin Theory for Classification
  - Properties of Rademacher Complexity
  - Covering Numbers and Scale-Sensitive Dimensions
  - Faster Rates
  - Model Selection

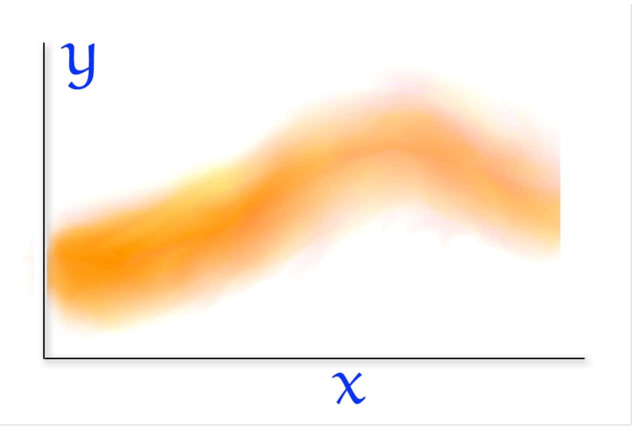
- Sequential Prediction / Online Learning
  - Motivation
  - Supervised Learning
  - Online Convex and Linear Optimization
  - Online-to-Batch Conversion, SVM optimization

Statistical Learning Theory

The variable x is related to y, and we would like to learn this relationship from data.

The relationship is encapsulated by a distribution P on $\mathcal{X} \times \mathcal{Y}$.

Example: $x = [\text{weight}, \text{blood glucose}, \ldots]$ and y is the risk of diabetes. We assume there is a relationship between x and y: it is less likely to see certain x co-occur with "low risk" and unlikely to see some other x co-occur with "high risk". This relationship is encapsulated by $P(x, y)$.



This is an assumption about the *population* of all (x, y). However, what we see is a *sample*.

Statistical Learning Theory

Data denoted by $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where n is the sample size.

The distribution P is unknown to us (otherwise, there is no learning to be done).

The observed data are sampled independently from P (the *i.i.d. assumption*)

It is often helpful to write $P = P_x \times P_{y|x}$. The distribution P_x on the inputs is called the *marginal distribution*, while $P_{y|x}$ is the *conditional distribution*.

Statistical Learning Theory

Upon observing the training data $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the learner is asked to summarize what she had learned about the relationship between x and y.

The learner's summary takes the form of a function $\hat{f}_n : \mathcal{X} \mapsto \mathcal{Y}$. The *hat* indicates that this function depends on the training data.

Learning algorithm: a mapping $\{(x_1, y_1), \dots, (x_n, y_n)\} \longmapsto \hat{f}_n$.

The quality of the learned relationship is given by comparing the response $\hat{f}_n(x)$ to y for a pair (x, y) independently drawn from the same distribution P:

$$\mathbb{E}_{(x,y)} \ell(\hat{f}_n(x), y)$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ is a *loss function*. This is our measure of performance.

Loss Functions

- Indicator loss (classification): $\ell(y, y') = \mathbf{I}_{\{y \neq y'\}}$
- Square loss: $\ell(y, y') = (y - y')^2$
- Absolute loss: $\ell(y, y') = |y - y'|$

Examples

Probably the simplest learning algorithm that you are probably familiar with is *linear least squares*:

Given $(x_1, y_1), \ldots, (x_n, y_n)$, let

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2$$

and define

$$\hat{f}_n(x) = \langle \hat{\beta}, x \rangle$$

Another basic method is *regularized least squares*:

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2 + \lambda \|\beta\|^2$$

Expected Loss and Empirical Loss

The *expected loss* of any function $f : \mathcal{X} \mapsto \mathcal{Y}$ is

$$\mathbf{L}(f) = \mathbb{E}\ell(f(x), y)$$

Since P is unknown, we cannot calculate $\mathbf{L}(f)$.

However, we can calculate the *empirical loss* of $f : \mathcal{X} \mapsto \mathcal{Y}$

$$\hat{\mathbf{L}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

... again, what is random here?

Since data $(x_1, y_1), \ldots, (x_n, y_n)$ are a random i.i.d. draw from P,

- $\hat{\mathbf{L}}(f)$ is a random quantity
- $\hat{f}_n$ is a random quantity (a random function, output of our learning procedure after seeing data)
- hence, $\mathbf{L}(\hat{f}_n)$ is also a random quantity
- for a given $f : \mathcal{X} \to \mathcal{Y}$, the quantity $\mathbf{L}(f)$ is *not* random!

It is important that these are understood before we proceed further.

Data Generated By A Probability Distribution

We assume that X and Y are two sets of random variables. We are given a **training set** S consisting n samples drawn i.i.d. from the probability distribution $\mu(z)$ on $Z = X \times Y$:

$$(x_1, y_1), \ldots, (x_n, y_n)$$

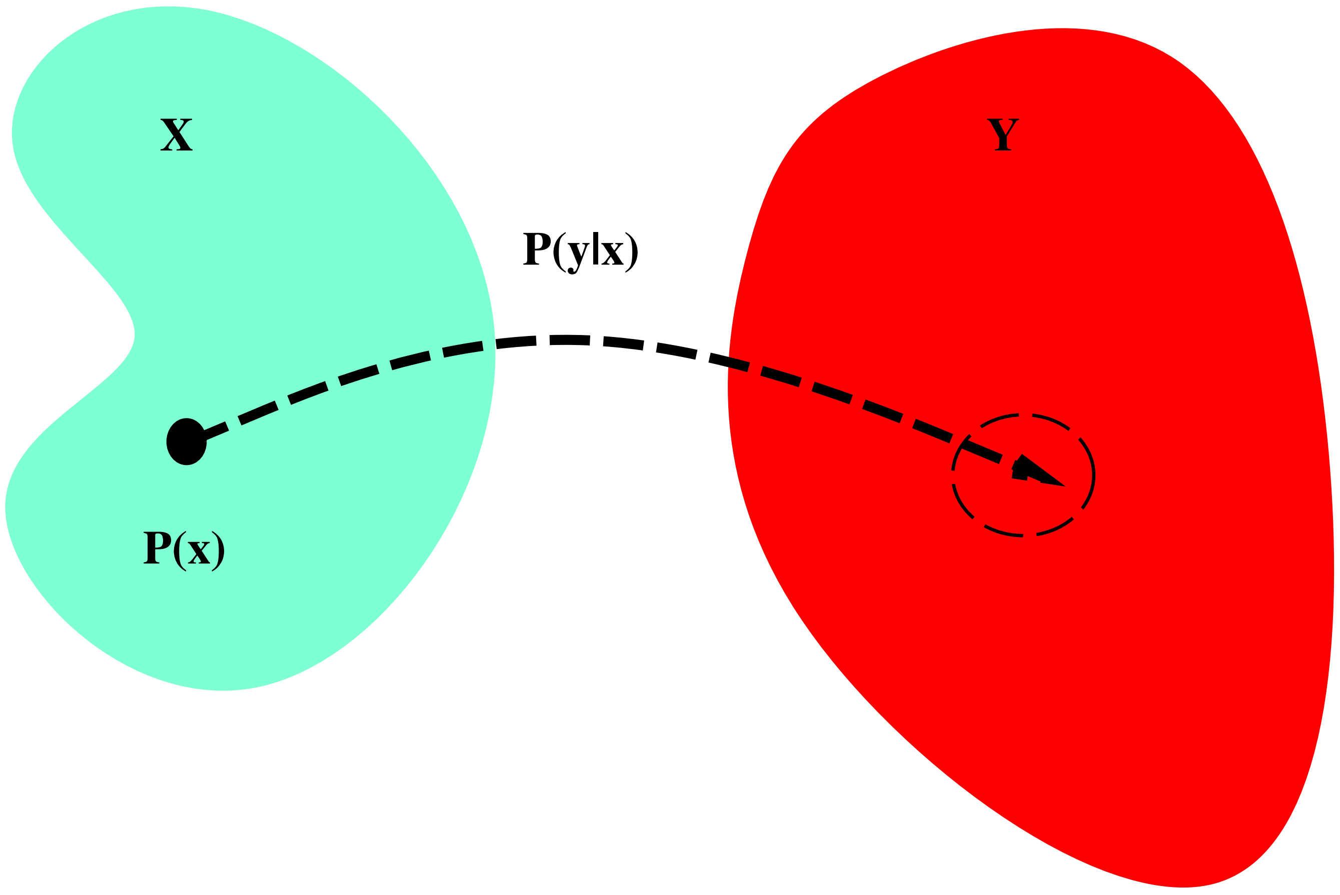
that is $z_1, \ldots, z_n$.

We will make frequent use of the **conditional probability of y given x**, written $p(y|x)$:

$$\mu(z) = p(x, y) = p(y|x) \cdot p(x)$$

It is crucial to note that we view $p(x, y)$ as **fixed** but **unknown**.

Probabilistic setting



Hypothesis Space

The **hypothesis space** $\mathcal{H}$ is the space of functions that we allow our algorithm to search. It is often chosen with respect to the amount of data available.

Learning As Function Approximation From Samples: Regression and Classification

The basic goal of **supervised learning** is to use the training set S to “learn” a function f_S that looks at a new x value x_{new} and predicts the associated value of y:

$$y_{pred} = f_S(x_{new}).$$

If y is a real-valued random variable, we have **regression**.

If y takes values from an unordered finite set, we have **pattern classification**. In two-class pattern classification problems, we assign one class a y value of 1, and the other class a y value of -1.

Loss Functions

In order to measure goodness of our function, we need a **loss function** V. In general, we let $V(f, z) = V(f(x), y^*)$ denote the price we pay when we see x and guess that the associated y value is $f(x)$ when it is actually y^*.

Common Loss Functions For Regression

For regression, the most common loss function is square loss or L2 loss:

$$V(f(x), y) = (f(x) - y)^2.$$

We could also use the absolute value, or L1 loss:

$$V(f(x), y) = |f(x) - y|.$$

Vapnik's more general ϵ-insensitive loss function is:

$$V(f(x), y) = (|f(x) - y| - \epsilon)_+.$$

Common Loss Functions For Classification

For binary classification, the most intuitive loss is the 0-1 loss:

$$V(f(x), y) = \Theta(-yf(x)).$$

For tractability and other reasons, we often use the hinge loss (implicitely introduced by Vapnik) in binary classification:

$$V(f(x), y) = (1 - y \cdot f(x))_+.$$

The learning problem: summary so far

There is an unknown **probability distribution** on the product space $Z = X \times Y$, written $\mu(z) = \mu(x, y)$. We assume that X is a compact domain in Euclidean space and Y a closed subset of $\mathbb{R}^k$.

The **training set** $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} = z_1, ...z_n$ consists of n samples drawn i.i.d. from μ.

$\mathcal{H}$ is the **hypothesis space**, a space of functions $f : X \to Y$.

A **learning algorithm** is a map $L : Z^n \to \mathcal{H}$ that looks at S and selects from $\mathcal{H}$ a function $f_S : \mathbf{x} \to y$ such that $f_S(\mathbf{x}) \approx y$ *in a predictive way*.

Empirical error, generalization error, generalization

Given a function f, a loss function V, and a probability distribution μ over Z, the **expected or true error** of f is:

$$I[f] = \mathbb{E}_z V[f, z] = \int_Z V(f, z) d\mu(z)$$

which is the **expected loss** on a new example drawn at random from μ.

We would like to make $I[f]$ small, but in general we do not know μ.

Given a function f, a loss function V, and a training set S consisting of n data points, the **empirical error** of f is:

$$I_S[f] = \frac{1}{n} \sum V(f, z_i).$$

Empirical error, generalization error, generalization

A very natural requirement for f_S is distribution independent **generalization**

$$\forall \mu, \lim_{n \to \infty} |I_S[f_S] - I[f_S]| = 0 \textit{ in probability.}$$

In other words, the training error for the ERM solution must converge to the expected error and thus be a "proxy" for it. Otherwise the solution would not be "predictive".

A desirable additional requirement is **universal consistency**

$$\forall \varepsilon > 0 \lim_{n \to \infty} \sup_{\mu} \mathbb{P}_S \left\{ I[f_S] > \inf_{f \in \mathcal{H}} I[f] + \varepsilon \right\} = 0.$$

A reminder: convergence in probability

Let $\{X_n\}$ be a sequence of bounded random variables. We say that

$$\lim_{n\to\infty} X_n = X \textit{ in probability}$$

if

$$\forall \varepsilon > 0 \lim_{n\to\infty} \mathbb{P}\{\|X_n - X\| \geq \varepsilon\} = 0$$

or if for each n there exists a ε_n and a δ_n such that

$$\mathbb{P}\left\{\|X_n - X\| \geq \varepsilon_n\right\} \leq \delta_n,$$

with ε_n and δ_n going to zero for $n \to \infty$.

3. ERM and conditions for generalization (and consistency)

Given a training set S and a function space $\mathcal{H}$, empirical risk minimization (Vapnik) is the algorithm that looks at S and selects f_S as

$$f_S = \arg\min_{f \in \mathcal{H}} I_S[f].$$

This problem does not in general show generalization and is also **ill-posed**, depending on the choice of $\mathcal{H}$.

If the minimum does not exist we can work with the infimum.

Notice: *For ERM generalization and consistency are equivalent*.

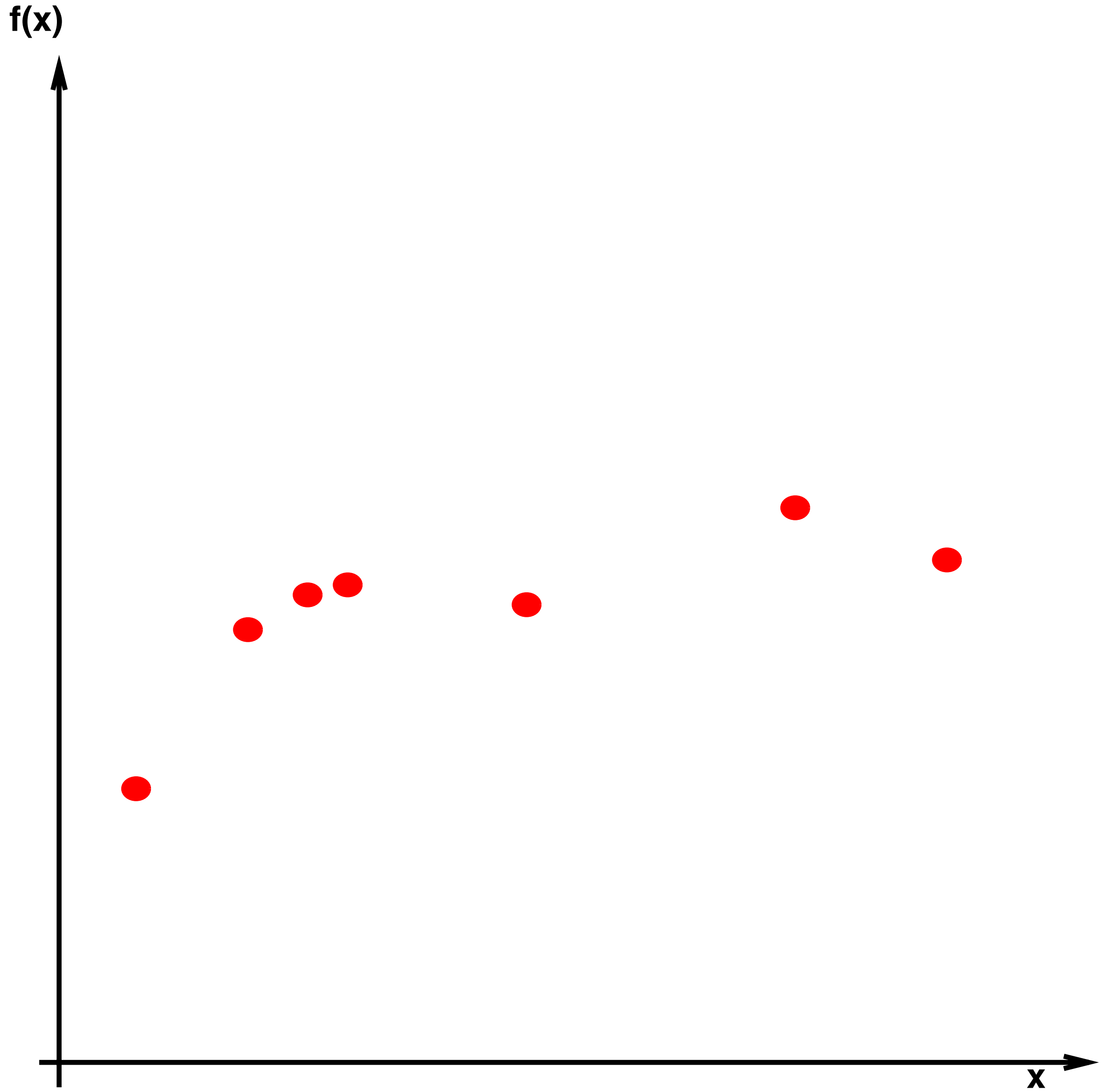
Generalization and Well-posedness of Empirical Risk Minimization

For the solution of ERM to be useful in the context of learning, the solution must

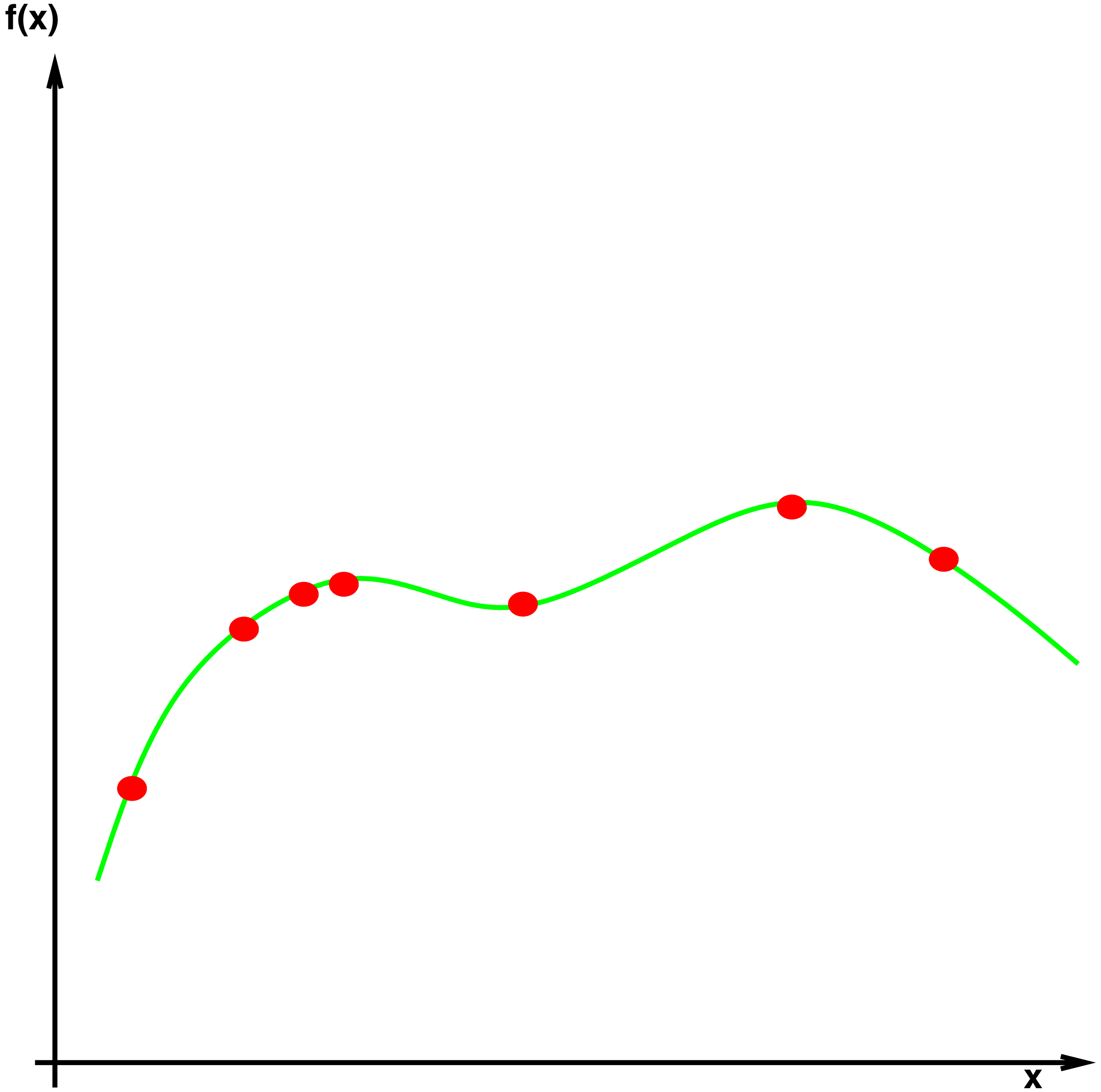
- “generalize”

- exist, be unique and be “stable” (well-posedness).

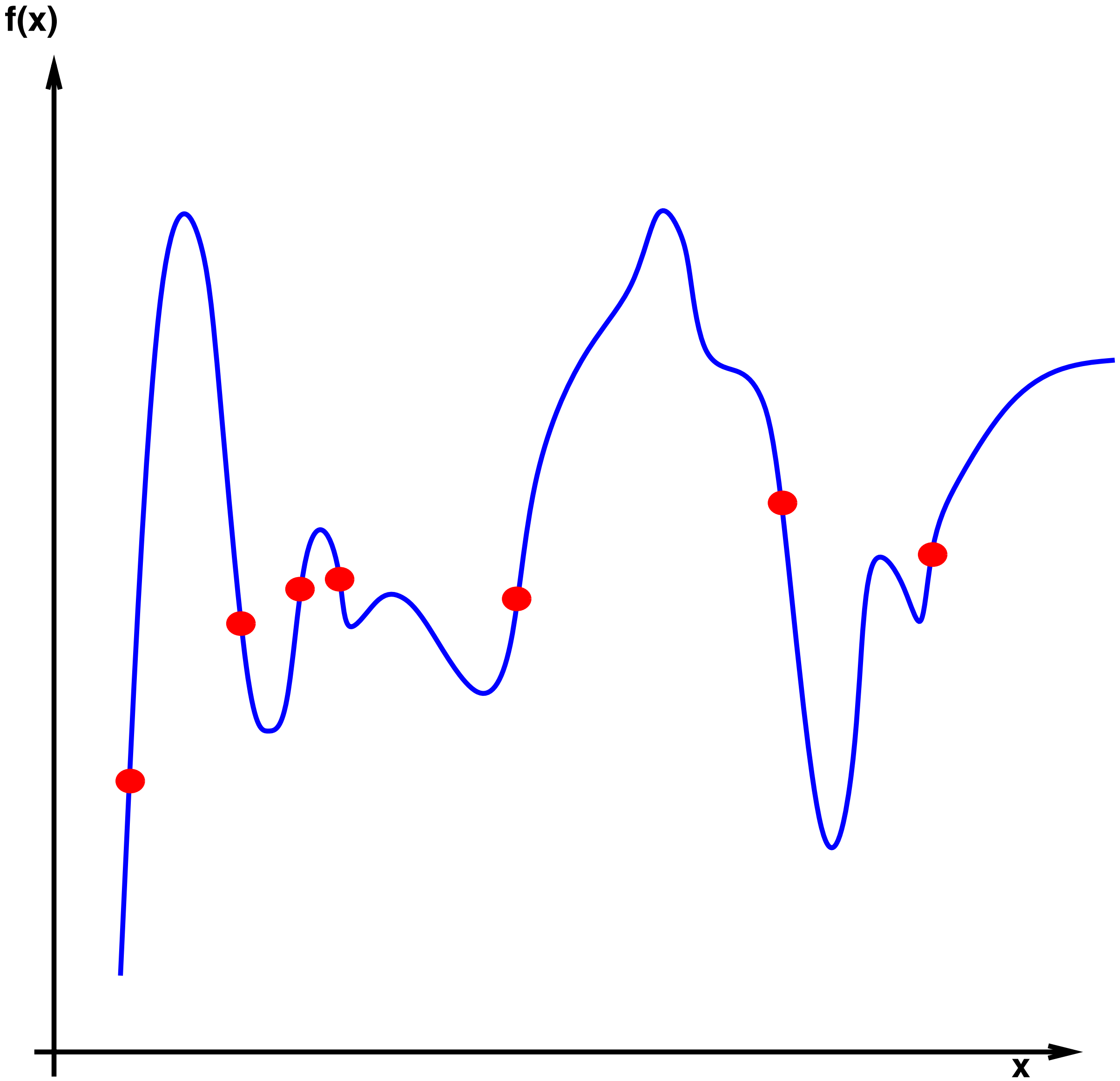
Here is a graphical example for generalization: given a certain number of samples...



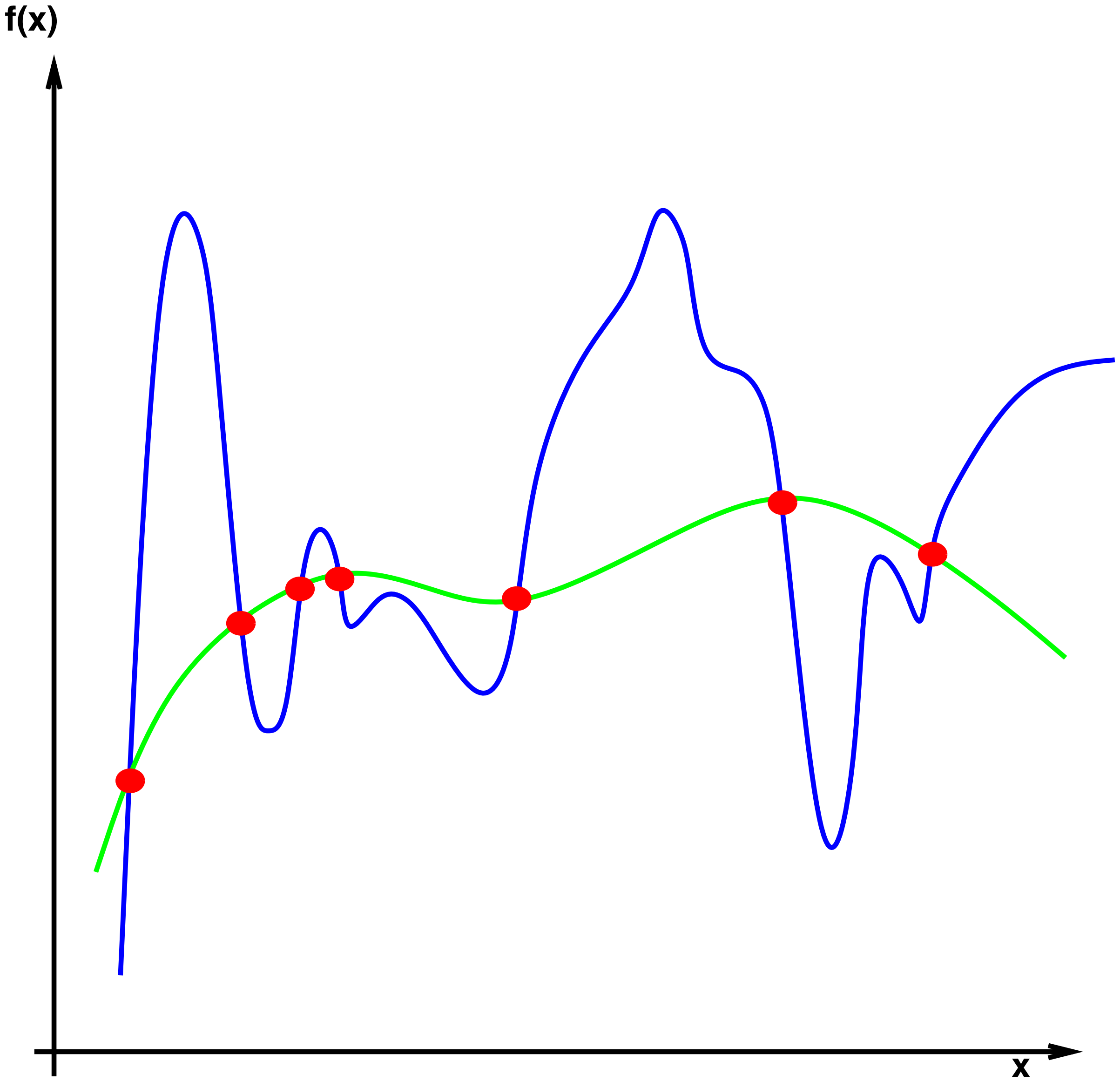
suppose this is the “true” solution...



... but supose ERM gives this solution!



How can I guarantee that for a sufficient number of examples the ERM solution will converge to the true one?



Classical conditions for consistency of ERM

Uniform Glivenko-Cantelli Classes

$\mathcal{L} = \{\mathcal{H}, V\}$ is a (weak) uniform Glivenko-Cantelli (uGC) class

if

$$\forall \varepsilon > 0 \lim_{n \to \infty} \sup_{\mu} \mathbb{P}_S \left\{ \sup_{\ell \in \mathcal{L}} |I[\ell] - I_S[\ell]| > \varepsilon \right\} = 0.$$

Theorem [Vapnik and Červonenkis (71), Alon et al (97), Dudley, Giné, and Zinn (91)]

A necessary and sufficient condition for consistency of ERM is that $\mathcal{L}$ is uGC.

Thus, as we will see later, a proper choice of the hypothesis space $\mathcal{H}$ ensures generalization of ERM (and consistency since for ERM generalization is necessary and sufficient for consistency and viceversa). We will be exploring the uGC definition (and equivalent definitions) in detail in 9.520.

Well-posedness of ERM

ERM is in general an ill-posed problem. It can be made well-posed by an appropriate choice of $\mathcal{H}$.

As we will see later, well-posedness is mainly used to mean *stability* of the solution: f_S depends continuously on the training set S. In particular, changing one of the training points should affect less and less the solution as n goes to infinity.

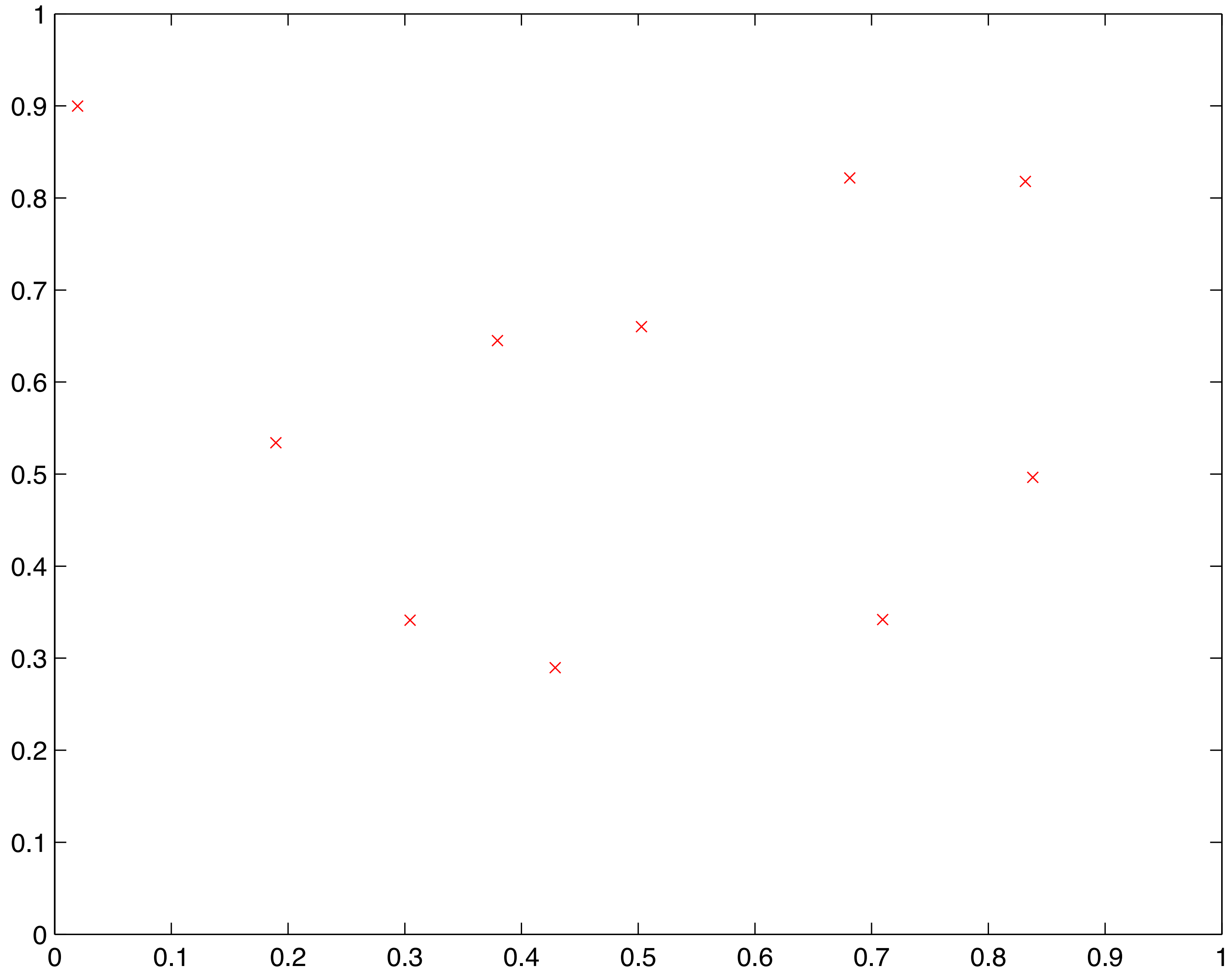
General definition of Well-Posed and Ill-Posed problems

A problem is **well-posed** if its solution:

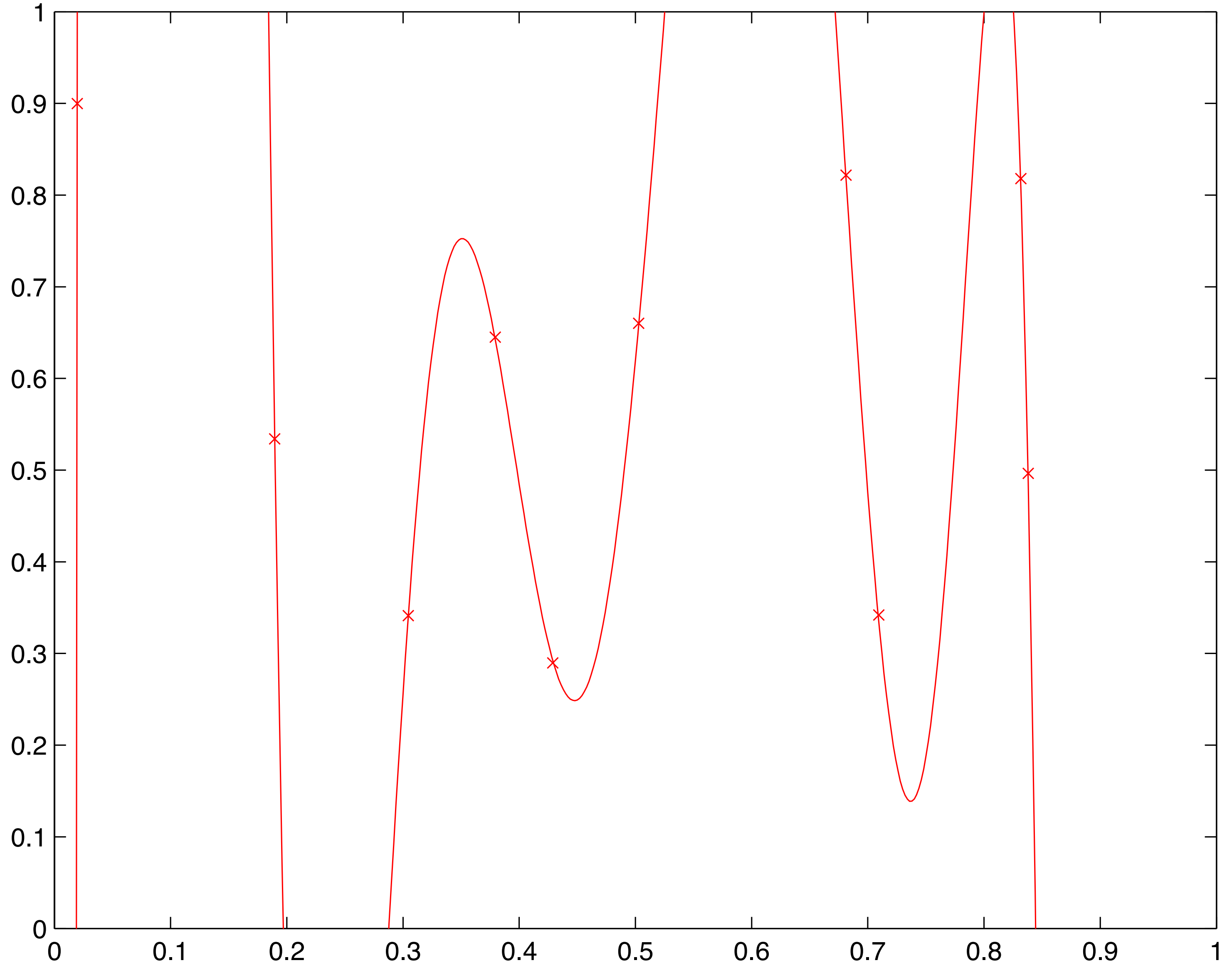
- exists
- is unique
- depends continuously on the data (e.g. it is *stable*)

A problem is **ill-posed** if it is not well-posed.

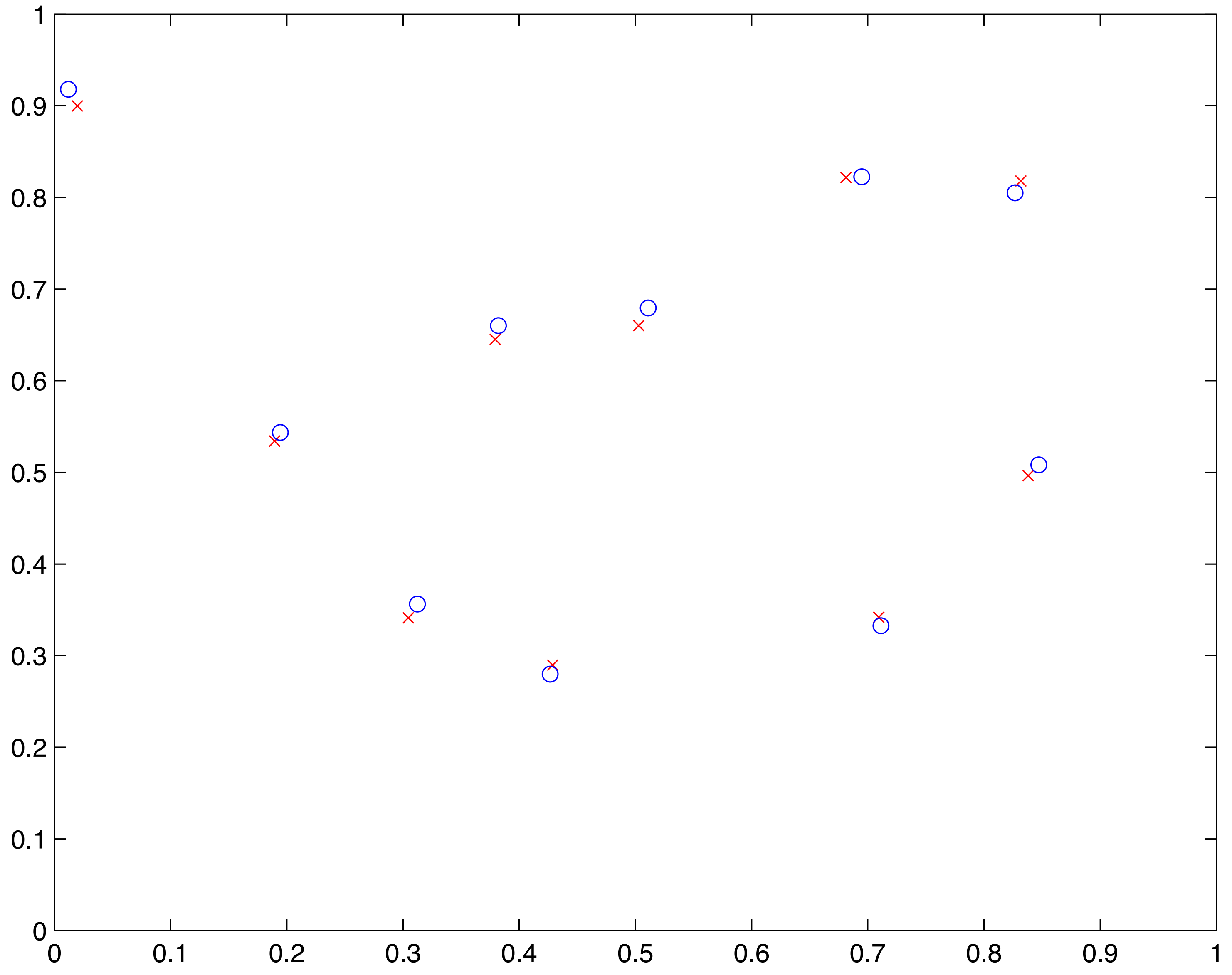
Here is a graphical example for stability: given 10 samples...



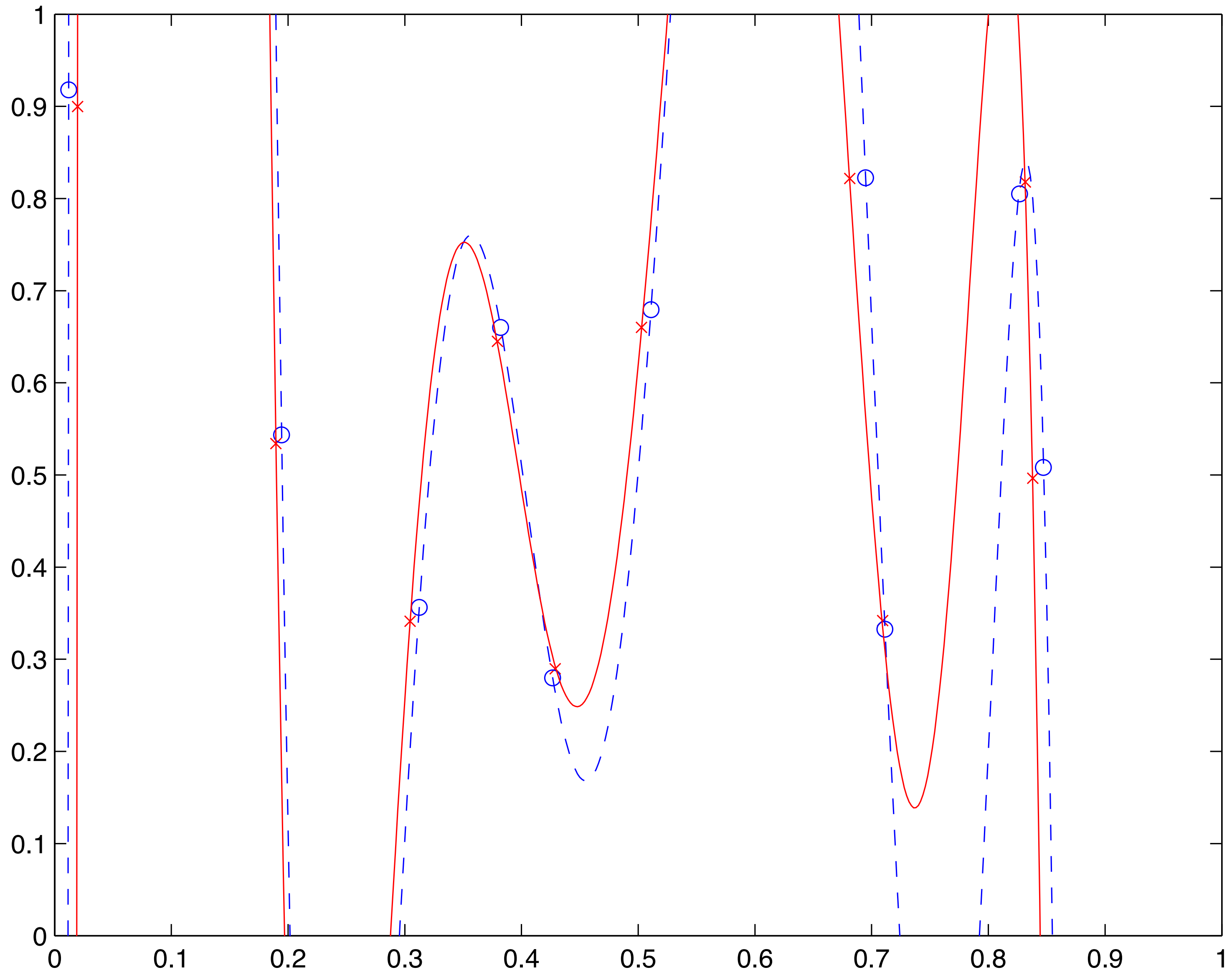
...we can find the smoothest interpolating polynomial.



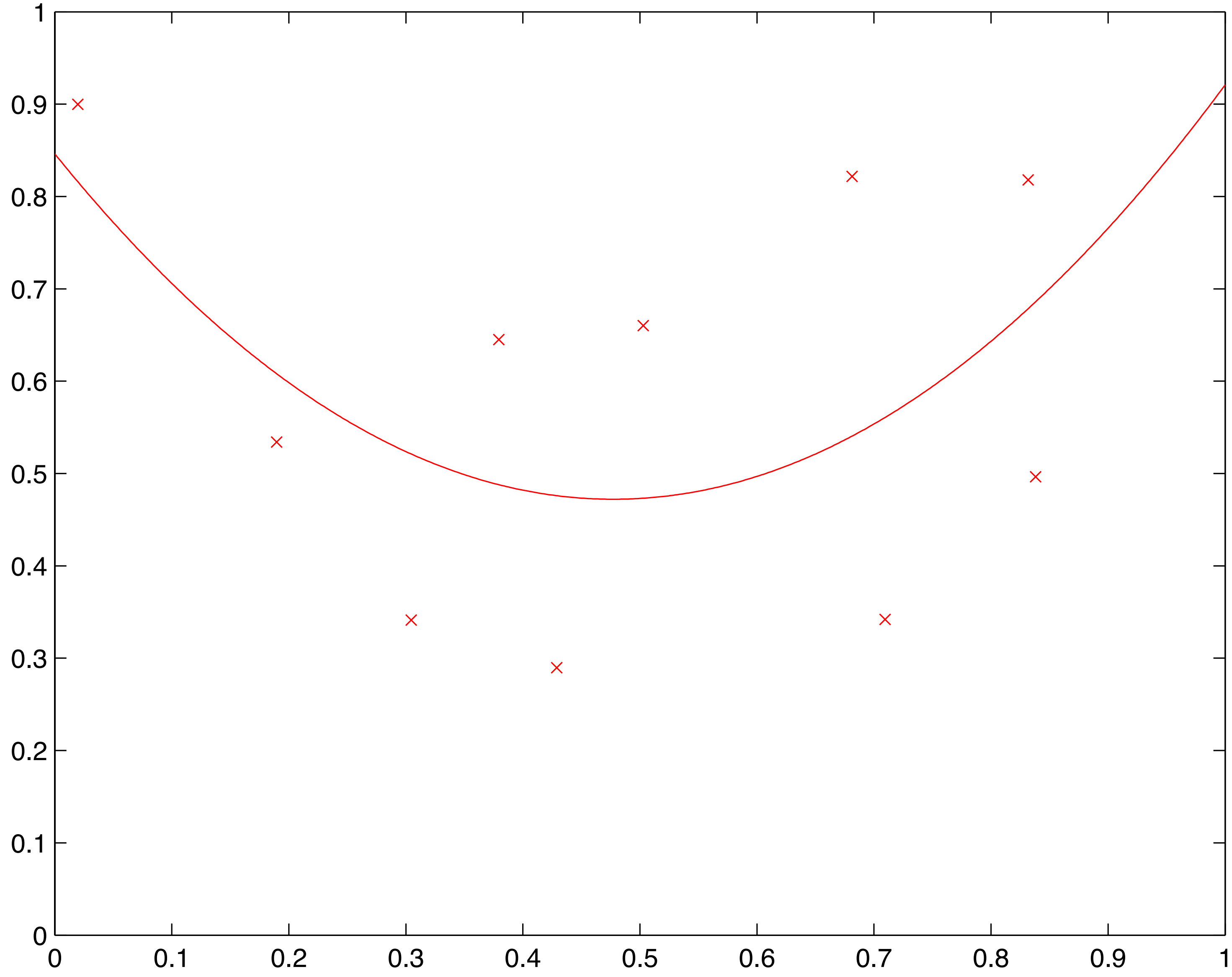
But if we perturb the points slightly...



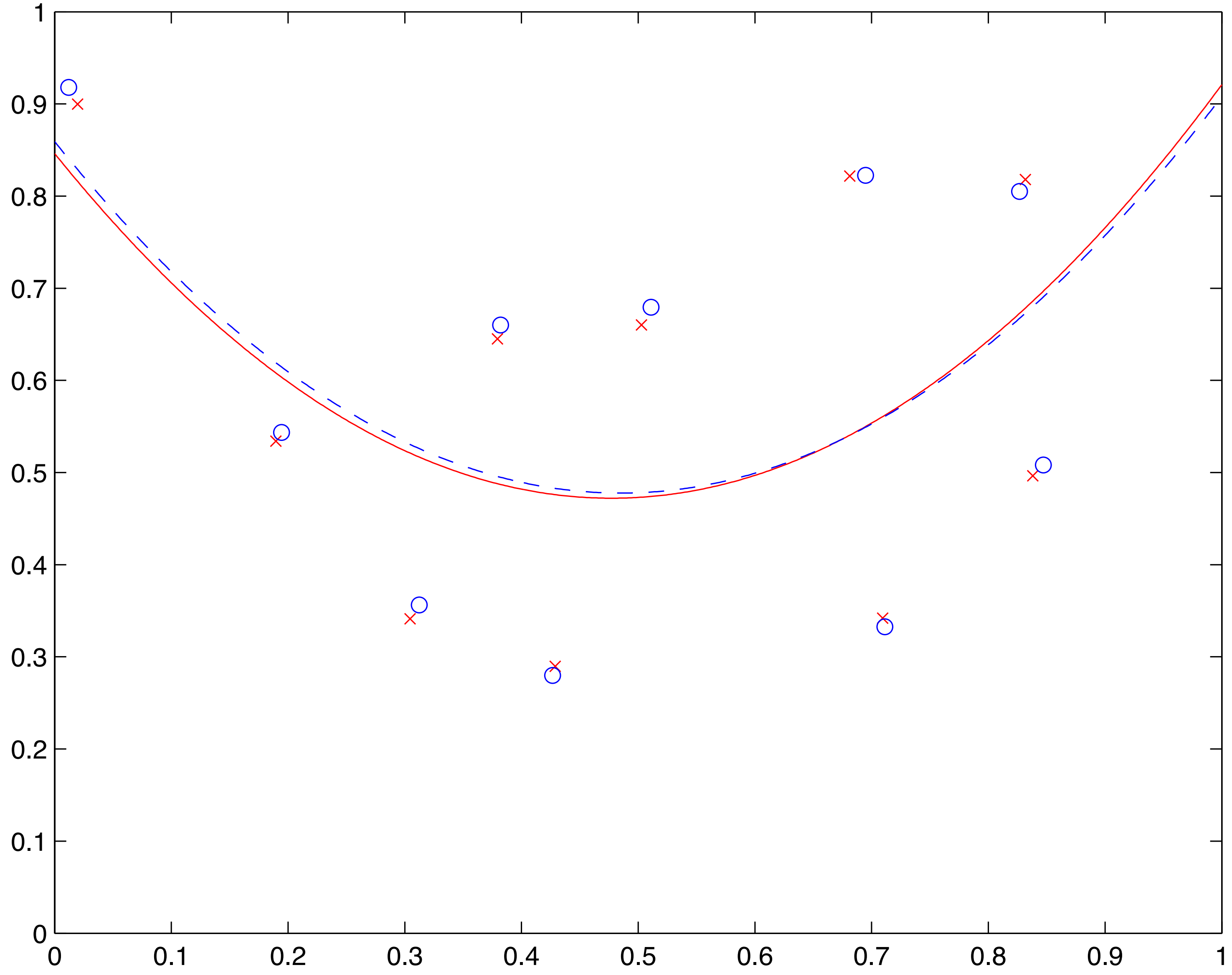
...the solution changes a lot.



If we restrict ourselves to degree two polynomials...



...the solution varies only a small amount under a small perturbation.



Regularization

The basic idea of regularization (originally introduced independently of the learning problem) is to restore well-posedness of ERM by constraining the hypothesis space $\mathcal{H}$. The direct way – minimize the empirical error subject to f in a ball in an appropriate $\mathcal{H}$ – is called Ivanov regularization. The indirect way is Tikhonov regularization (which is not ERM).

Ivanov and Tikhonov Regularization

ERM finds the function in $\mathcal{H}$ which minimizes

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

which in general – for arbitrary hypothesis space $\mathcal{H}$ – is *ill-posed*. Ivanov regularizes by finding the function that minimizes

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

while satisfying

$$\|f\|_K^2 \leq A$$

Alternatively, Tikhonov regularization minimizes over the hypothesis space $\mathcal{H}$, for a fixed positive parameter λ, the regularized functional

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i) + \gamma\|f\|_K^2, \qquad (1)$$

where $\|f\|_K^2$ is the norm in $\mathcal{H}_K$ – the Reproducing Kernel Hilbert Space (RKHS), defined by the kernel K.

Tikhonov Regularization

As we will see in future classes

- Tikhonov regularization ensures well-posedness eg existence, uniqueness and especially *stability* (in a very strong form) of the solution

- Tikhonov regularization ensures generalization

- Tikhonov regularization is closely related to – but different from – Ivanov regularization, eg ERM on a hypothesis space $\mathcal{H}$ which is a ball in a RKHS.

Well-posed and Ill-posed problems

Hadamard introduced the definition of ill-posedness. Ill-posed problems are typically inverse problems.

As an example, assume g is a function in Y and u is a function in X, with Y and X Hilbert spaces. Then given the linear, continuous operator L, consider the equation

$$g = Lu.$$

The direct problem is is to compute g given u; the inverse problem is to compute u given the data g. In the learning case L is somewhat similar to a "sampling" operation.

The inverse problem of finding u is well-posed when

- the solution exists,
- is unique and
- is *stable*, that is depends continuously on the initial data g.

Ill-posed problems fail to satisfy one or more of these criteria. Often the term ill-posed applies to problems that are **not stable**, which in a sense is the key condition.

Sample Error (also called Estimation Error)

Let $f_{\mathcal{H}}$ be the function in $\mathcal{H}$ with the smallest true risk.

We have defined the **generalization error** to be $I_S[f_S] - I[f_S]$.

We define the **sample error** to be $I[f_S] - I[f_{\mathcal{H}}]$, the difference in true risk between the best function in $\mathcal{H}$ and the function in $\mathcal{H}$ we actually find. This is what we pay because our finite sample does not give us enough information to choose to the "best" function in $\mathcal{H}$. We'd like this to be small. *Consistency* – defined earlier – is equivalent to the sample error going to zero for $n \to \infty$.

A main topic of this course is "bounding" the generalization error. Another topic – the main one in classical learning theory and statistics – is bounding the sample error, that is determining conditions under which we can state that $I[f_S] - I[f_{\mathcal{H}}]$ will be small (with high probability).

As a simple rule, we expect that if $\mathcal{H}$ is "well-behaved", then, as n gets large the sample error will become small.

Approximation Errror

Let f_0 be the function in $\mathcal{T}$ with the smallest true risk.

We define the **approximation error** to be $I[f_{\mathcal{H}}] - I[f_0]$, the difference in true risk between the best function in $\mathcal{H}$ and the best function in $\mathcal{T}$. This is what we pay because $\mathcal{H}$ is smaller than $\mathcal{T}$. We'd like this error to be small too. In much of the following we can assume that $I[f_0] = 0$.

We will focus less on the approximation error in 9.520, but we will explore it.

As a simple rule, we expect that as $\mathcal{H}$ grows bigger, the approximation error gets smaller. If $\mathcal{T} \subseteq \mathcal{H}$ – which is a situation called *the realizable setting* –the approximation error is zero.

Error

We define the **error** to be $I[f_S] - I[f_0]$, the difference in true risk between the function we actually find and the best function in $\mathcal{T}$. We'd really like this to be small. As we mentioned, often we can assume that the **error** is simply $I[f_S]$.

The error is the sum of the sample error and the approximation error:

$$I[f_S] - I[f_0] = (I[f_S] - I[f_{\mathcal{H}}]) + (I[f_{\mathcal{H}}] - I[f_0])$$

If we can make both the approximation and the sample error small, the error will be small. There is a tradeoff between the approximation error and the sample error...

The Approximation/Sample Tradeoff

It should already be intuitively clear that making $\mathcal{H}$ big makes the approximation error small. This implies that we can (help) make the error small by making $\mathcal{H}$ big.

On the other hand, we will show that making $\mathcal{H}$ small will make the sample error small. In particular for ERM, if $\mathcal{H}$ is a uGC class, the generalization error and the sample error will go to zero as $n \to \infty$, but how quickly depends directly on the "size" of $\mathcal{H}$. This implies that we want to keep $\mathcal{H}$ as small as possible. (Furthermore, $\mathcal{T}$ itself may or may not be a uGC class.)

Ideally, we would like to find the optimal tradeoff between these conflicting requirements.

The Gold Standard

Within the framework we set up, the smallest expected loss is achieved by the *Bayes optimal* function

$$f^* = \arg\min_f \mathbf{L}(f)$$

where the minimization is over all (measurable) prediction rules $f : \mathcal{X} \mapsto \mathcal{Y}$.

The value of the lowest expected loss is called the *Bayes error*:

$$\mathbf{L}(f^*) = \inf_f \mathbf{L}(f)$$

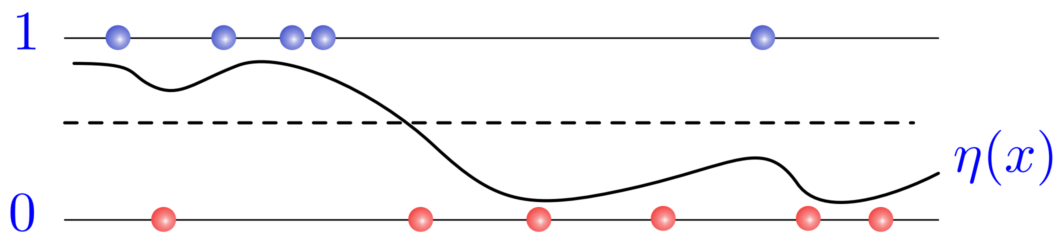
Of course, we cannot calculate any of these quantities since P is unknown.

Bayes Optimal Function

Bayes optimal function f^* takes on the following forms in these two particular cases:

- Binary classification ($\mathcal{Y} = \{0, 1\}$) with the indicator loss:

$$f^*(x) = \mathbf{I}_{\{\eta(x) \geq 1/2\}}, \quad \text{where} \quad \eta(x) = \mathbb{E}[Y|X = x]$$

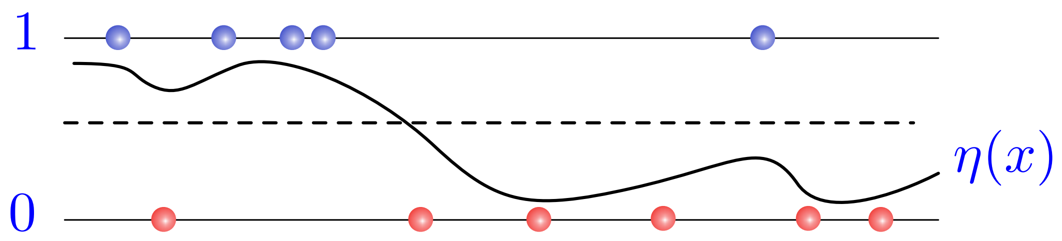


Bayes Optimal Function

Bayes optimal function f^* takes on the following forms in these two particular cases:

- Binary classification ($\mathcal{Y} = \{0, 1\}$) with the indicator loss:

$$f^*(x) = \mathbf{I}_{\{\eta(x) \geq 1/2\}}, \quad \text{where} \quad \eta(x) = \mathbb{E}[Y|X = x]$$



- Regression ($\mathcal{Y} = \mathbb{R}$) with squared loss:

$$f^*(x) = \eta(x), \quad \text{where} \quad \eta(x) = \mathbb{E}[Y|X = x]$$

The big question: is there a way to construct a learning algorithm with a guarantee that

$$\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*)$$

is small for large enough sample size n?

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Consistency

An algorithm that ensures

$$\lim_{n \to \infty} \mathbf{L}(\hat{f}_n) = \mathbf{L}(f^*) \qquad \text{almost surely}$$

is called *consistent*. Consistency ensures that our algorithm is approaching the best possible prediction performance as the sample size increases.

The good news: consistency is possible to achieve.

- easy if $\mathcal{X}$ is a finite or countable set
- not too hard if $\mathcal{X}$ is infinite, and the underlying relationship between x and y is "continuous"

The bad news...

In general, we cannot prove anything "interesting" about $\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*)$, unless we make further assumptions (incorporate *prior knowledge*).

What do we mean by "nothing interesting"? This is the subject of the so-called "No Free Lunch" Theorems. Unless we posit further assumptions,

The bad news...

In general, we cannot prove anything "interesting" about $\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*)$, unless we make further assumptions (incorporate *prior knowledge*).

What do we mean by "nothing interesting"? This is the subject of the so-called "No Free Lunch" Theorems. Unless we posit further assumptions,

- For any algorithm $\hat{f}_n$, any n and any $\epsilon > 0$, there exists a distribution P such that $\mathbf{L}(f^*) = 0$ and

$$\mathbb{E}\mathbf{L}(\hat{f}_n) \geq \frac{1}{2} - \epsilon$$

The bad news...

In general, we cannot prove anything "interesting" about $\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*)$, unless we make further assumptions (incorporate *prior knowledge*).

What do we mean by "nothing interesting"? This is the subject of the so-called "No Free Lunch" Theorems. Unless we posit further assumptions,

- For any algorithm $\hat{f}_n$, any n and any $\epsilon > 0$, there exists a distribution P such that $\mathbf{L}(f^*) = 0$ and

$$\mathbb{E}\mathbf{L}(\hat{f}_n) \geq \frac{1}{2} - \epsilon$$

- For any algorithm $\hat{f}_n$, and any sequence a_n that converges to 0, there exists a probability distribution P such that $\mathbf{L}(f^*) = 0$ and for all n

$$\mathbb{E}\mathbf{L}(\hat{f}_n) \geq a_n$$

Reference: (Devroye, Györfi, Lugosi: *A Probabilistic Theory of Pattern Recognition*), (Bousquet, Boucheron, Lugosi, 2004)

is this really “bad news”?

Not really. We always have some domain knowledge.

Two ways of incorporating prior knowledge:

- Direct way: assume that the distribution P is not arbitrary (also known as a modeling approach, generative approach, statistical modeling)
- Indirect way: redefine the goal to perform as well as a reference set $\mathcal{F}$ of predictors:

  $$\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)$$

  This is known as a discriminative approach. $\mathcal{F}$ encapsulates our *inductive bias*.

Pros/Cons of the two approaches

Pros of the discriminative approach: we never assume that P takes some particular form, but we rather put our prior knowledge into “what are the types of predictors that will do well”. Cons: cannot really interpret $\hat{f}_n$.

Pros of the generative approach: can estimate the model / parameters of the distribution (*inference*). Cons: it is not clear what the analysis says if the assumption is actually violated.

Both approaches have their advantages. A machine learning researcher or practitioner should ideally know both and should understand their strengths and weaknesses.

In this tutorial we only focus on the discriminative approach.

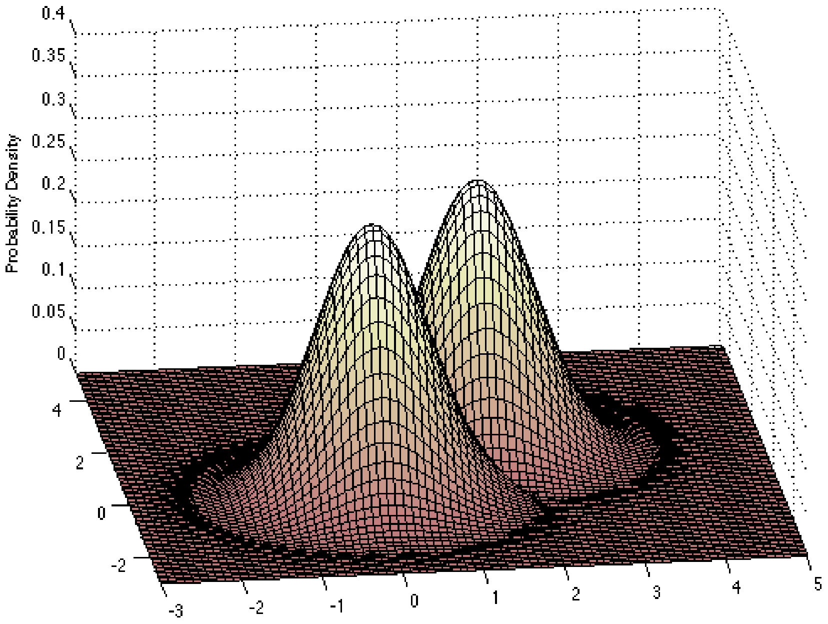
Example: Linear Discriminant Analysis

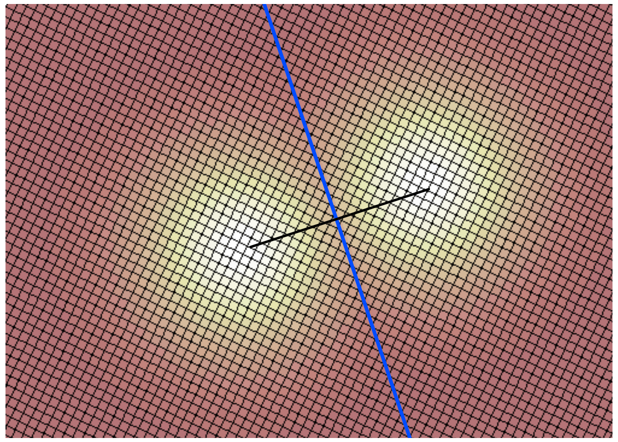
Consider the classification problem with $\mathcal{Y} = \{0, 1\}$. Suppose the *class-conditional densities* are multivariate Gaussian with the same covariance $\Sigma = I$:

$$p(x|y = 0) = (2\pi)^{-k/2} \exp\left\{-\frac{1}{2}\|x - \mu_0\|^2\right\}$$

and

$$p(x|y = 1) = (2\pi)^{-k/2} \exp\left\{-\frac{1}{2}\|x - \mu_1\|^2\right\}$$





The "best" (Bayes) classifier is $f^* = \mathbf{I}_{\{P(y=1|x) \geq 1/2\}}$ which corresponds to the half-space defined by the decision boundary $p(x|y = 1) \geq p(x|y = 0)$. This boundary is *linear*.

Example: Linear Discriminant Analysis

The (linear) optimal decision boundary comes from our generative assumption on the form of the underlying distribution.

Alternatively, we could have indirectly postulated that we will be looking for a linear discriminant between the two classes, without making distributional assumptions. Such linear discriminant (classification) functions are

$$\mathbf{I}_{\{\langle w,x\rangle \geq b\}}$$

for a unit-norm w and some bias $b \in \mathbb{R}$.

Quadratic Discriminant Analysis: If unequal correlation matrices Σ_1 and Σ_2 are assumed, the resulting boundary is quadratic. We can then define classification function by

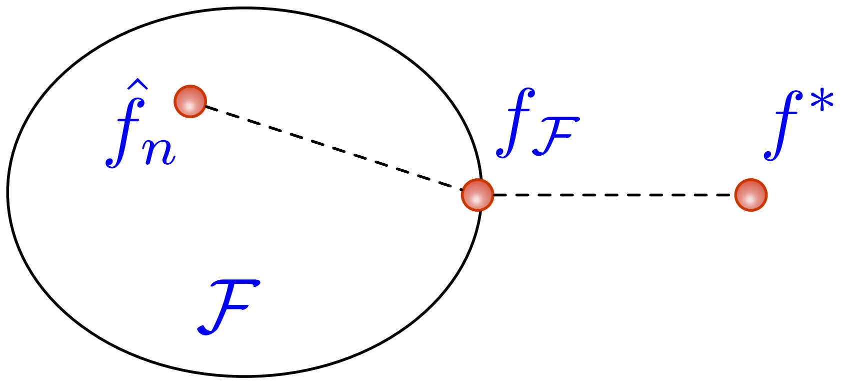
$$\mathbf{I}_{\{q(x) \geq 0\}}$$

where $q(x)$ is a quadratic function.

Bias-Variance Tradeoff

How do we choose the inductive bias $\mathcal{F}$?

$$\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*) = \underbrace{\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)}_{\text{Estimation Error}} \quad + \quad \underbrace{\inf_{f \in \mathcal{F}} \mathbf{L}(f) - \mathbf{L}(f^*)}_{\text{Approximation Error}}$$



Clearly, the two terms are at odds with each other:

- Making $\mathcal{F}$ larger means smaller approximation error but (as we will see) larger estimation error
- Taking a larger sample n means smaller estimation error and has no effect on the approximation error.
- Thus, it makes sense to trade off size of $\mathcal{F}$ and n. This is called *Structural Risk Minimization*, or *Method of Sieves*, or *Model Selection*.

Bias-Variance Tradeoff

We will only focus on the estimation error, yet the ideas we develop will make it possible to read about model selection on your own.

Note: if we guessed correctly and $f^* \in \mathcal{F}$, then

$$\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*) = \mathbf{L}(\hat{f}_n) - \inf_{f\in\mathcal{F}} \mathbf{L}(f)$$

For a particular problem, one hopes that prior knowledge about the problem can ensure that the approximation error $\inf_{f\in\mathcal{F}} \mathbf{L}(f) - \mathbf{L}(f^*)$ is small.

Occam's Razor

Occam's Razor is often quoted as a principle for choosing the simplest theory or explanation out of the possible ones.

However, this is a rather philosophical argument since simplicity is not uniquely defined. We will discuss this issue later.

What we will do is to try to understand "complexity" when it comes to behavior of certain stochastic processes. Such a question will be well-defined mathematically.

Looking Ahead

So far: represented prior knowledge by means of the class $\mathcal{F}$.

Looking forward, we can find an algorithm that, after looking at a dataset of size n, produces $\hat{f}_n$ such that

$$\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)$$

decreases (in a certain sense which we will make precise) at a non-trivial rate which depends on “richness” of $\mathcal{F}$.

This will give a *sample complexity* guarantee: how many samples are needed to make the error smaller than a desired accuracy.

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Types of Bounds

In expectation vs in probability (control the mean vs control the tails):

$$\mathbb{E}\left\{\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)\right\} < \psi(n) \qquad \text{vs} \qquad \mathbb{P}\left(\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \geq \epsilon\right) < \psi(n, \epsilon)$$

Types of Bounds

In expectation vs in probability (control the mean vs control the tails):

$$\mathbb{E}\left\{\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)\right\} < \psi(n) \qquad \text{vs} \qquad \mathbb{P}\left(\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \geq \epsilon\right) < \psi(n, \epsilon)$$

The in-probability bound can be inverted as

$$\mathbb{P}\left(\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \geq \phi(\delta, n)\right) < \delta$$

by setting $\delta := \psi(\epsilon, n)$ and solving for ϵ.

In this lecture, we are after the function $\phi(\delta, n)$. We will call it "the rate".

"With high probability" typically means logarithmic dependence of $\phi(\delta, n)$ on $1/\delta$. Very desirable: the bound grows only modestly even for high confidence bounds.

Sample Complexity

Sample complexity is the sample size required by the algorithm $\hat{f}_n$ to guarantee $\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \le \epsilon$ with probability at least $1 - \delta$. Of course, we just need to invert a bound

$$\mathbb{P}\left(\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \ge \phi(\delta, n)\right) < \delta$$

by setting $\epsilon := \phi(\delta, n)$ and solving for n. In other words, $n(\epsilon, \delta)$ is sample complexity of the algorithm $\hat{f}_n$ if

$$\mathbb{P}\left(\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \ge \epsilon\right) \le \delta$$

as soon as $n \ge n(\epsilon, \delta)$.

Hence, “rate” can be translated into “sample complexity” and vice versa.

Easy to remember: rate $O(1/\sqrt{n})$ means $O(1/\epsilon^2)$ sample complexity, whereas rate $O(1/n)$ is a smaller $O(1/\epsilon)$ sample complexity.

Types of Bounds

Other distinctions to keep in mind: We can ask for bounds (either in expectation or in probability) on the following random variables:

$$\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*) \qquad \text{(A)}$$

$$\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \qquad \text{(B)}$$

$$\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n) \qquad \text{(C)}$$

$$\sup_{f \in \mathcal{F}} \left\{ \mathbf{L}(f) - \hat{\mathbf{L}}(f) \right\} \qquad \text{(D)}$$

$$\sup_{f \in \mathcal{F}} \left\{ \mathbf{L}(f) - \hat{\mathbf{L}}(f) - \mathrm{pen}_n(f) \right\} \qquad \text{(E)}$$

Let’s make sure we understand the differences between these random quantities!

Types of Bounds

Upper bounds on (D) and (E) are used as *tools* for achieving the other bounds. Let's see why.

Obviously, for any algorithm that outputs $\hat{f}_n \in \mathcal{F}$,

$$\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n) \leq \sup_{f\in\mathcal{F}} \left\{ \mathbf{L}(f) - \hat{\mathbf{L}}(f) \right\}$$

and so a bound on (D) implies a bound on (C).

How about a bound on (B)? Is it implied by (C) or (D)? It depends on what the algorithm does!

Denote $f_{\mathcal{F}} = \arg\min_{f\in\mathcal{F}} \mathbf{L}(f)$. Suppose (D) is small. It then makes sense to ask the learning algorithm to minimize or (approximately minimize) the empirical error (why?)

Canonical Algorithms

Empirical Risk Minimization (ERM) algorithm:

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{\mathbf{L}}(f)$$

Regularized Empirical Risk Minimization algorithm:

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{\mathbf{L}}(f) + \mathrm{pen}_n(f)$$

We will deal with the regularized ERM a bit later. For now, let's focus on ERM.

Remark: to actually *compute* $f \in \mathcal{F}$ minimizing the above objectives, one needs to employ some optimization methods. In practice, the objective might be optimized only approximately.

Performance of ERM

If $\hat{f}_n$ is an ERM,

$$\begin{aligned}
\mathbf{L}(\hat{f}_n) - \mathbf{L}(f_{\mathcal{F}}) &\le \left\{\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n)\right\} + \left\{\hat{\mathbf{L}}(\hat{f}_n) - \hat{\mathbf{L}}(f_{\mathcal{F}})\right\} + \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\} \\
&\le \underbrace{\left\{\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n)\right\}}_{(C)} + \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\} \\
&\le \underbrace{\sup_{f\in\mathcal{F}}\left\{\mathbf{L}(f) - \hat{\mathbf{L}}(f)\right\}}_{(D)} + \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\}
\end{aligned}$$

because the second term is negative. So, (C) also implies a bound on (B) when $\hat{f}_n$ is ERM (or "close" to ERM). Also, (D) also implies a bound on (B).

What about this extra term $\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})$? Central Limit Theorem says that for i.i.d. random variables with bounded second moment, the average converges to the expectation. Let's quantify this.

Hoeffding Inequality

Let $W, W_1, \ldots, W_n$ be i.i.d. such that $\mathbb{P}\left(a \le W \le b\right) = 1$. Then

$$\mathbb{P}\left(\mathbb{E}W - \frac{1}{n}\sum_{i=1}^{n} W_i > \epsilon\right) \le \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

and

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} W_i - \mathbb{E}W > \epsilon\right) \le \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

Let $W_i = \ell(f_{\mathcal{F}}(x_i), y_i)$. Clearly, $W_1, \ldots, W_i$ are i.i.d. Then,

$$\mathbb{P}\left(\left|\mathbf{L}(f_{\mathcal{F}}) - \hat{\mathbf{L}}(f_{\mathcal{F}})\right| > \epsilon\right) \le 2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

assuming $a \le \ell(f_{\mathcal{F}}(x), y) \le b$ for all $x \in \mathcal{X}, y \in \mathcal{Y}$.

Wait, Are We Done?

Can't we conclude directly that (C) is small? That is,

$$\mathbb{P}\left(\mathbb{E}\ell(\hat{f}_n(x), y) - \frac{1}{n}\sum_{i=1}^{n}\ell(\hat{f}_n(x_i), y_i) > \epsilon\right) \leq 2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right) \quad ?$$

Wait, Are We Done?

Can't we conclude directly that (C) is small? That is,

$$\mathbb{P}\left(\mathbb{E}\ell(\hat{f}_n(x), y) - \frac{1}{n}\sum_{i=1}^{n}\ell(\hat{f}_n(x_i), y_i) > \epsilon\right) \leq 2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right) \quad ?$$

No! The random variables $\ell(\hat{f}_n(x_i), y_i)$ are not necessarily independent and it is possible that

$$\mathbb{E}\ell(\hat{f}_n(x), y) = \mathbb{E}W \neq \mathbb{E}\ell(\hat{f}_n(x_i), y_i) = \mathbb{E}W_i$$

The expected loss is "out of sample performance" while the second term is "in sample".

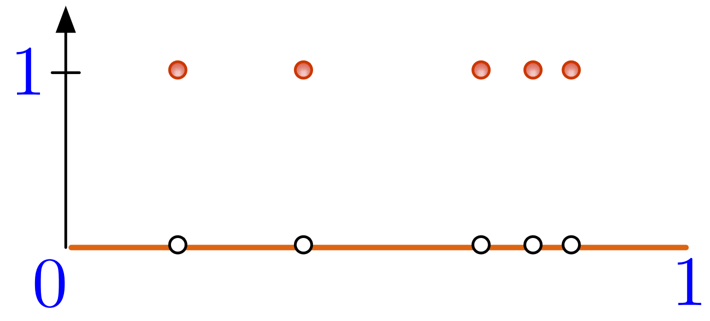
We say that $\ell(\hat{f}_n(x_i), y_i)$ is a *biased estimate* of $\mathbb{E}\ell(\hat{f}_n(x), y)$.

How bad can this bias be?

Example

- $\mathcal{X} = [0,1]$, $\mathcal{Y} = \{0,1\}$
- $\ell(f(X_i), Y_i) = \mathbf{I}_{\{f(X_i) \neq Y_i\}}$
- distribution $P = P_x \times P_{y|x}$ with $P_x = \text{Unif}[0,1]$ and $P_{y|x} = \delta_{y=1}$
- function class

$$\mathcal{F} = \cup_{n \in \mathbb{N}} \left\{ f = f_S : S \subset \mathcal{X}, |S| = n, f_S(x) = \mathbf{I}_{\{x \in S\}} \right\}$$



ERM $\hat{f}_n$ memorizes (perfectly fits) the data, but has no ability to generalize. Observe that

$$0 = \mathbb{E}\ell(\hat{f}_n(x_i), y_i) \neq \mathbb{E}\ell(\hat{f}_n(x), y) = 1$$

This phenomenon is called *overfitting.*

Example

Not only is (C) large in this example. Also, uniform deviations (D) do not converge to zero.

For any $n \in \mathbb{N}$ and any $(x_1, y_1), \ldots, (x_n, y_n) \sim P$

$$\sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{x,y} \ell(f(x), y) - \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \right\} = 1$$

Where do we go from here? Two approaches:

1. understand how to upper bound uniform deviations (D)
2. find properties of algorithms that limit in some way the bias of $\ell(\hat{f}_n(x_i), y_i)$. *Stability* and *compression* are two such approaches.

Uniform Deviations

We first focus on understanding

$$\sup_{f\in\mathcal{F}}\left\{\mathbb{E}_{x,y}\ell(f(x),y)-\frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i),y_i)\right\}$$

If $\mathcal{F}=\{f_0\}$ consists of a single function, then clearly

$$\sup_{f\in\mathcal{F}}\left\{\mathbb{E}\ell(f(x),y)-\frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i),y_i)\right\}=\left\{\mathbb{E}\ell(f_0(x),y)-\frac{1}{n}\sum_{i=1}^{n}\ell(f_0(x_i),y_i)\right\}$$

This quantity is $O_P(1/\sqrt{n})$ by Hoeffding's inequality, assuming $a\le\ell(f_0(x),y)\le b$.

Moral: for "simple" classes $\mathcal{F}$ the uniform deviations (D) can be bounded while for "rich" classes not. We will see how far we can push the size of $\mathcal{F}$.

A bit of notation to simplify things...

To ease the notation,

- Let $z_i = (x_i, y_i)$ so that the training data is $\{z_1, \dots, z_n\}$
- $g(z) = \ell(f(x), y)$ for $z = (x, y)$
- Loss class $\mathcal{G} = \{g : g(z) = \ell(f(x), y)\} = \ell \circ \mathcal{F}$
- $\hat{g}_n = \ell(\hat{f}_n(\cdot), \cdot)$, $g_{\mathcal{G}} = \ell(f_{\mathcal{F}}(\cdot), \cdot)$
- $g^* = \arg\min_g \mathbb{E}g(z) = \ell(f^*(\cdot), \cdot)$ is Bayes optimal (loss) function

We can now work with the set $\mathcal{G}$, but keep in mind that each $g \in \mathcal{G}$ corresponds to an $f \in \mathcal{F}$:

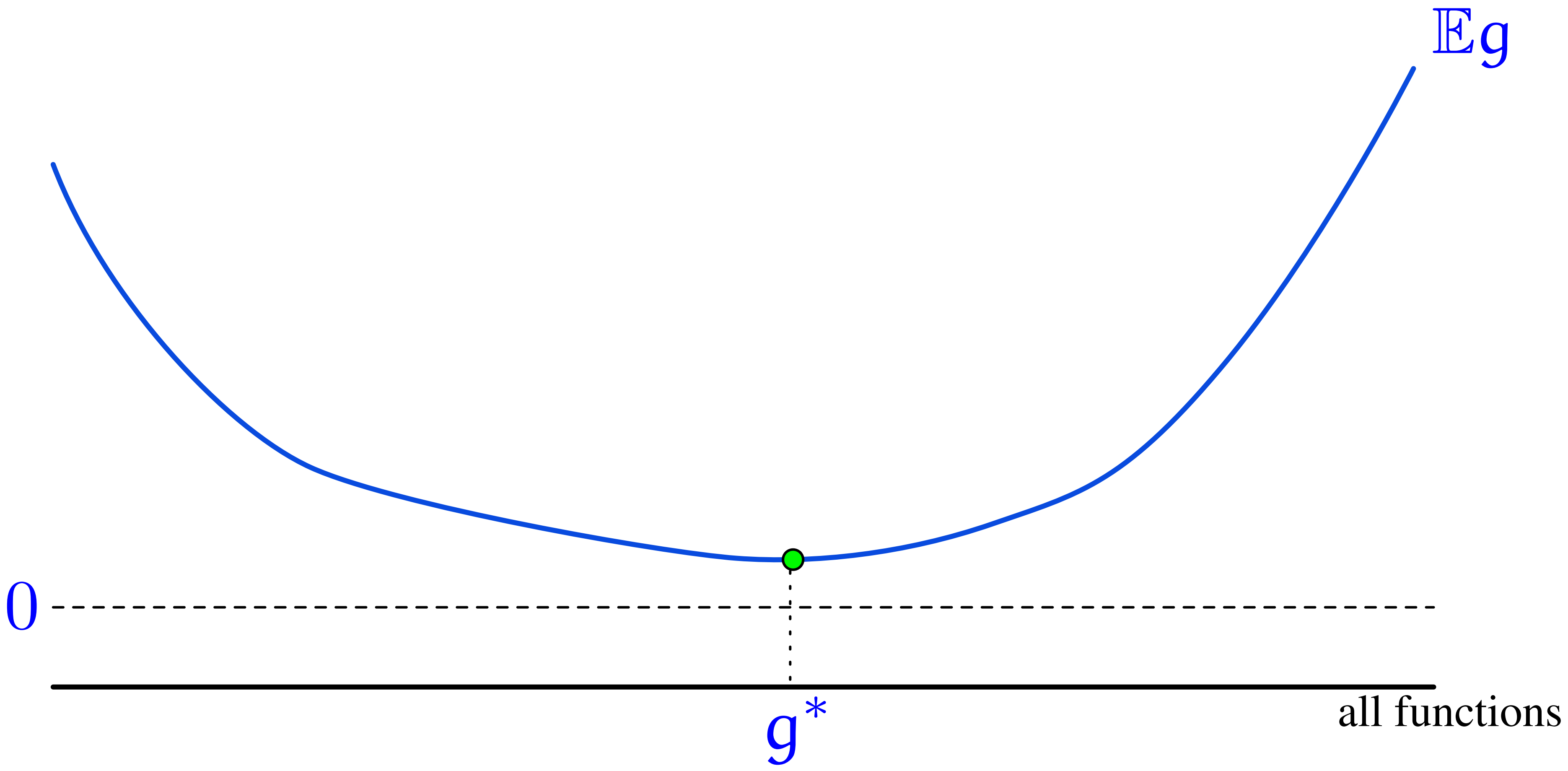
$$g \in \mathcal{G} \quad \longleftrightarrow \quad f \in \mathcal{F}$$

Once again, the quantity of interest is

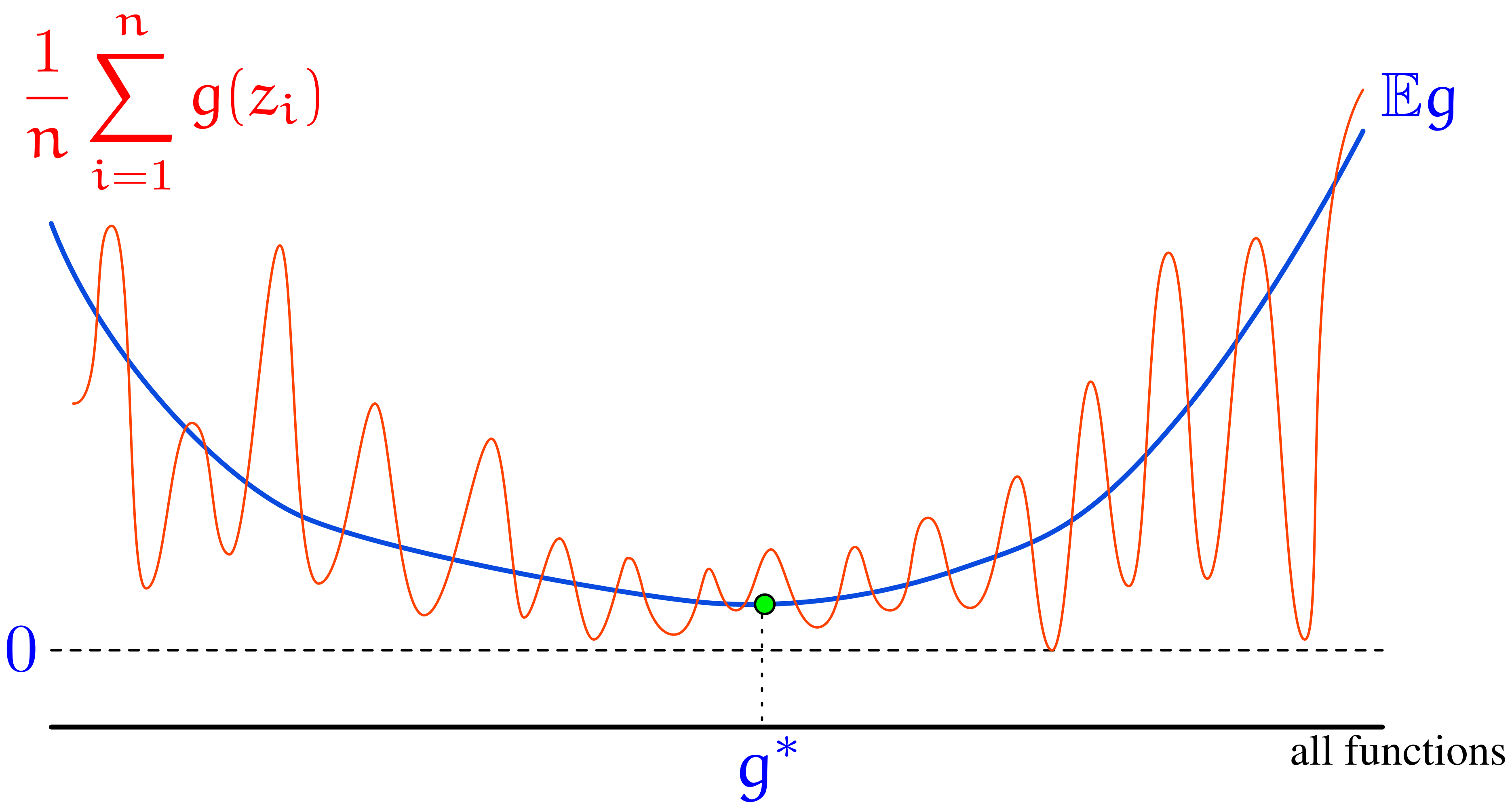
$$\sup_{g \in \mathcal{G}} \left\{ \mathbb{E}g(z) - \frac{1}{n} \sum_{i=1}^{n} g(z_i) \right\}$$

On the next slide, we visualize deviations $\mathbb{E}g(z) - \frac{1}{n}\sum_{i=1}^{n} g(z_i)$ for all possible functions g and discuss all the concepts introduces so far.

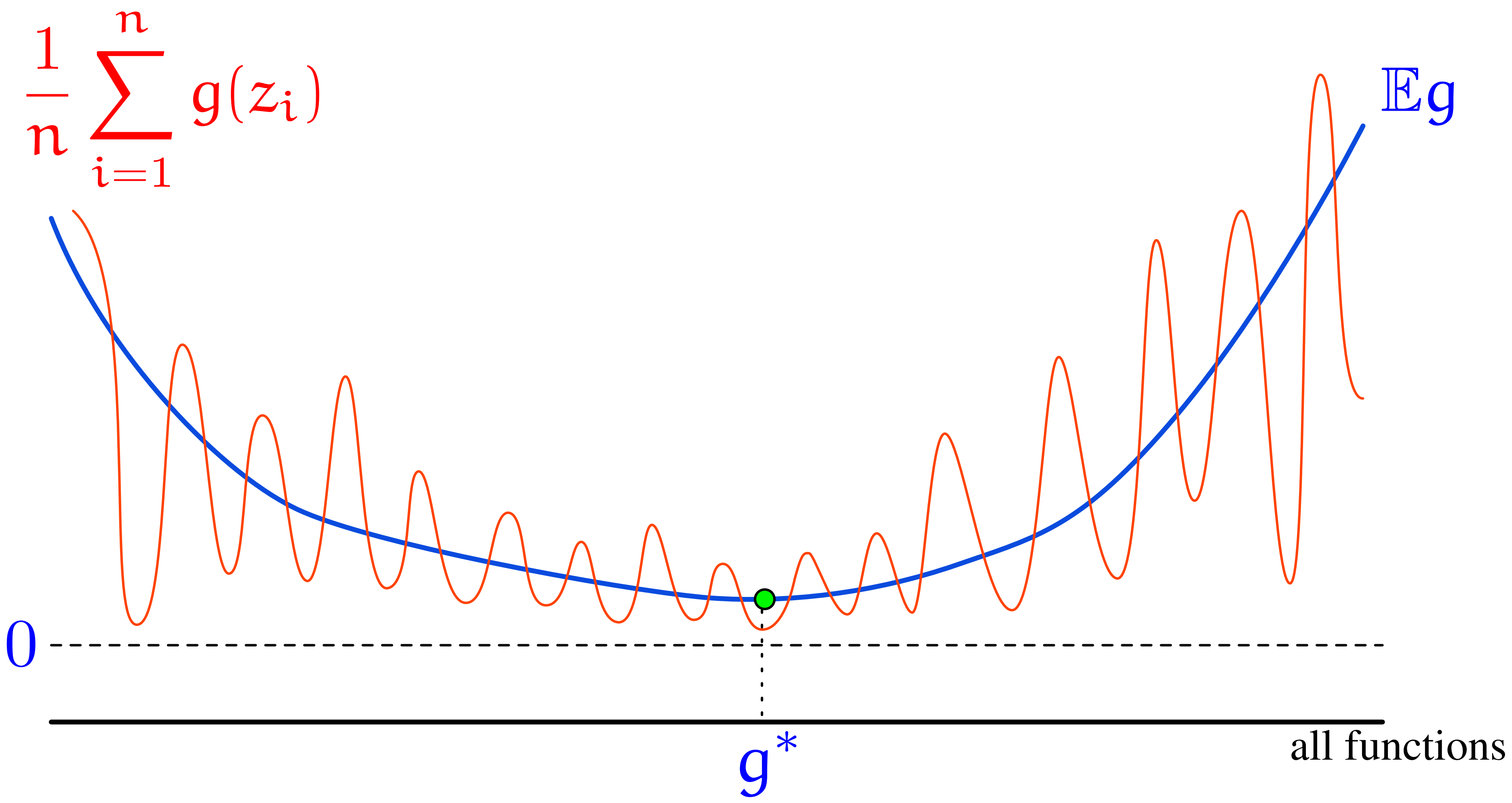
Empirical Process Viewpoint



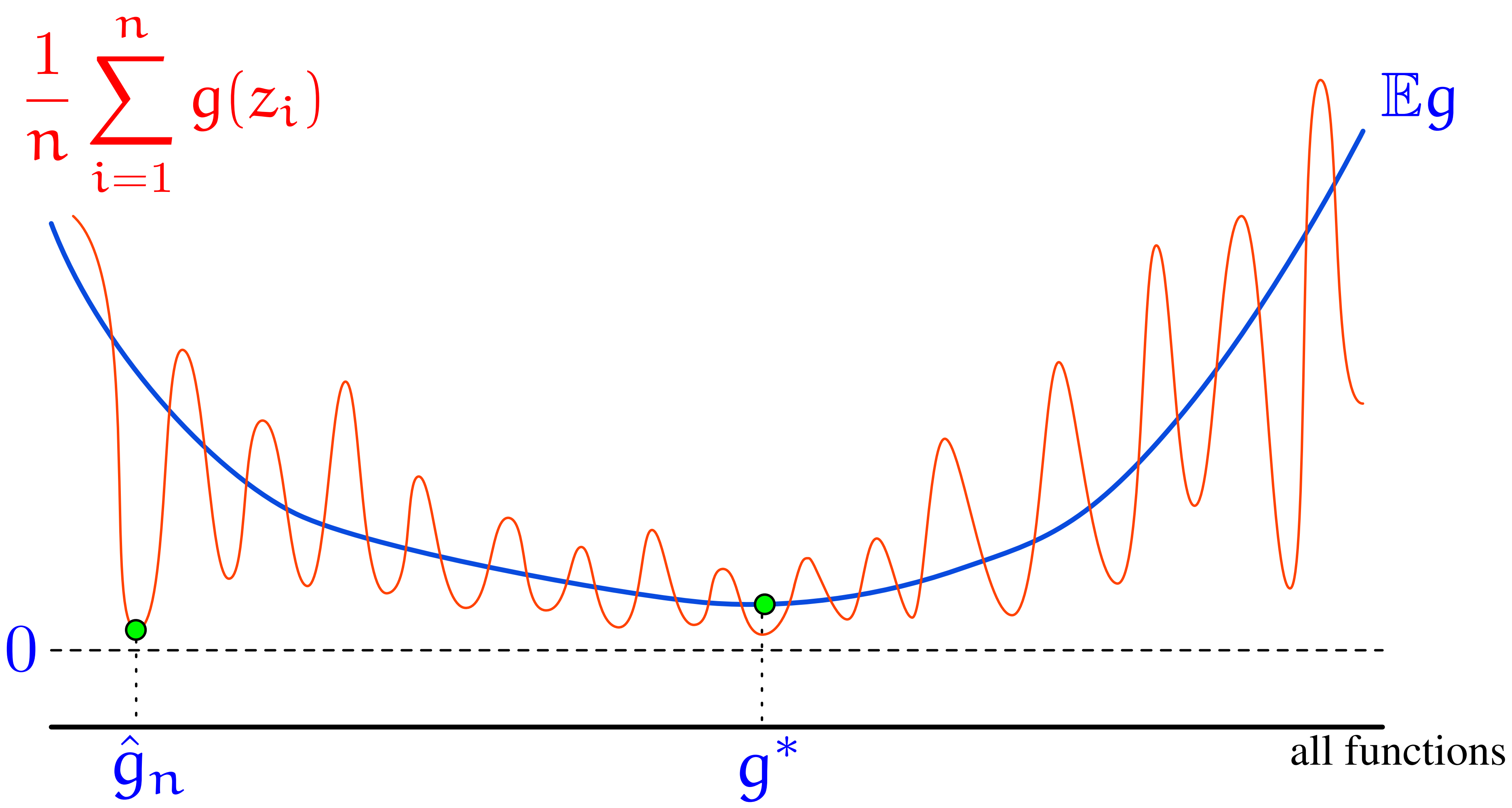
Empirical Process Viewpoint



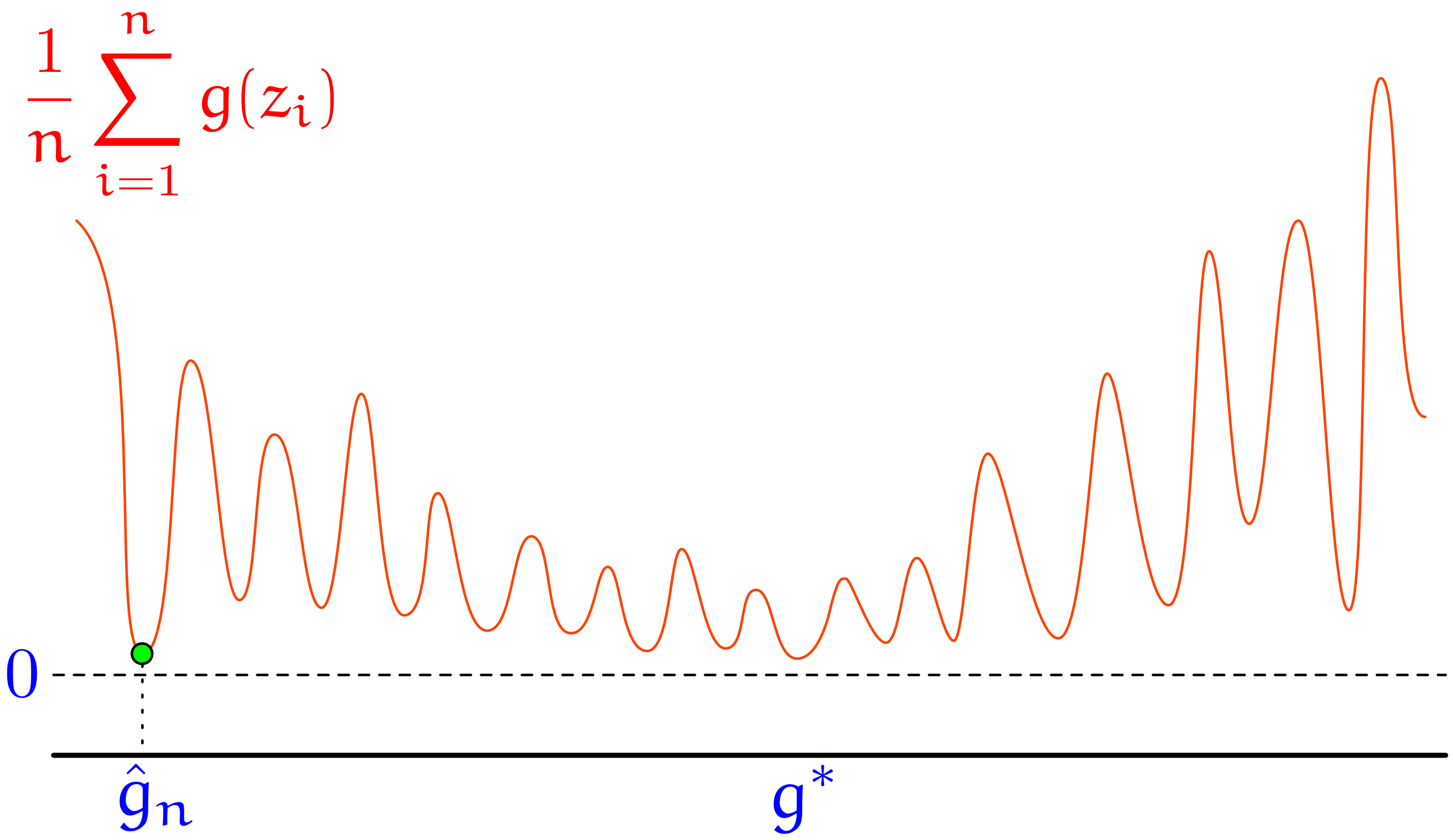
Empirical Process Viewpoint



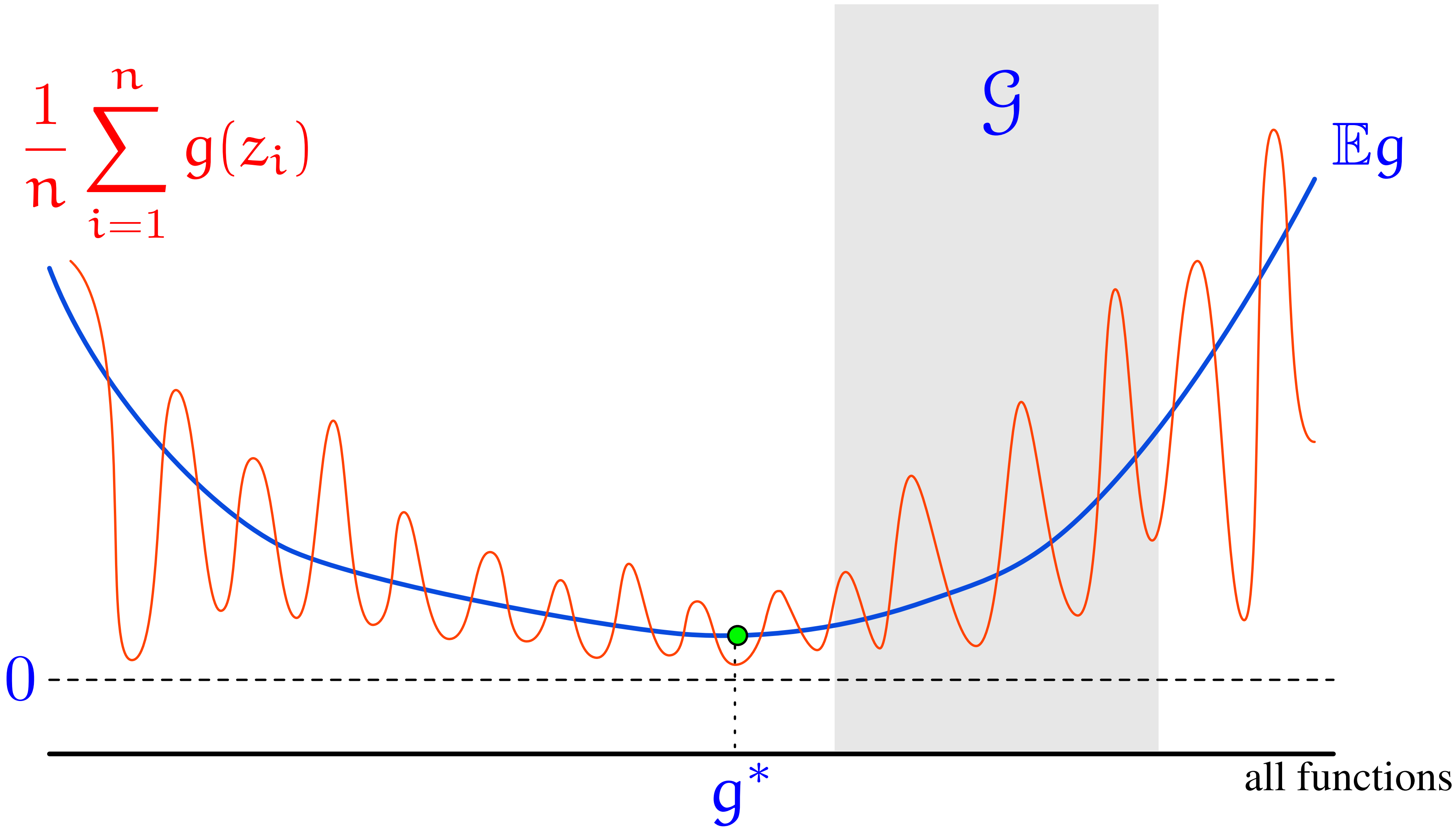
Empirical Process Viewpoint



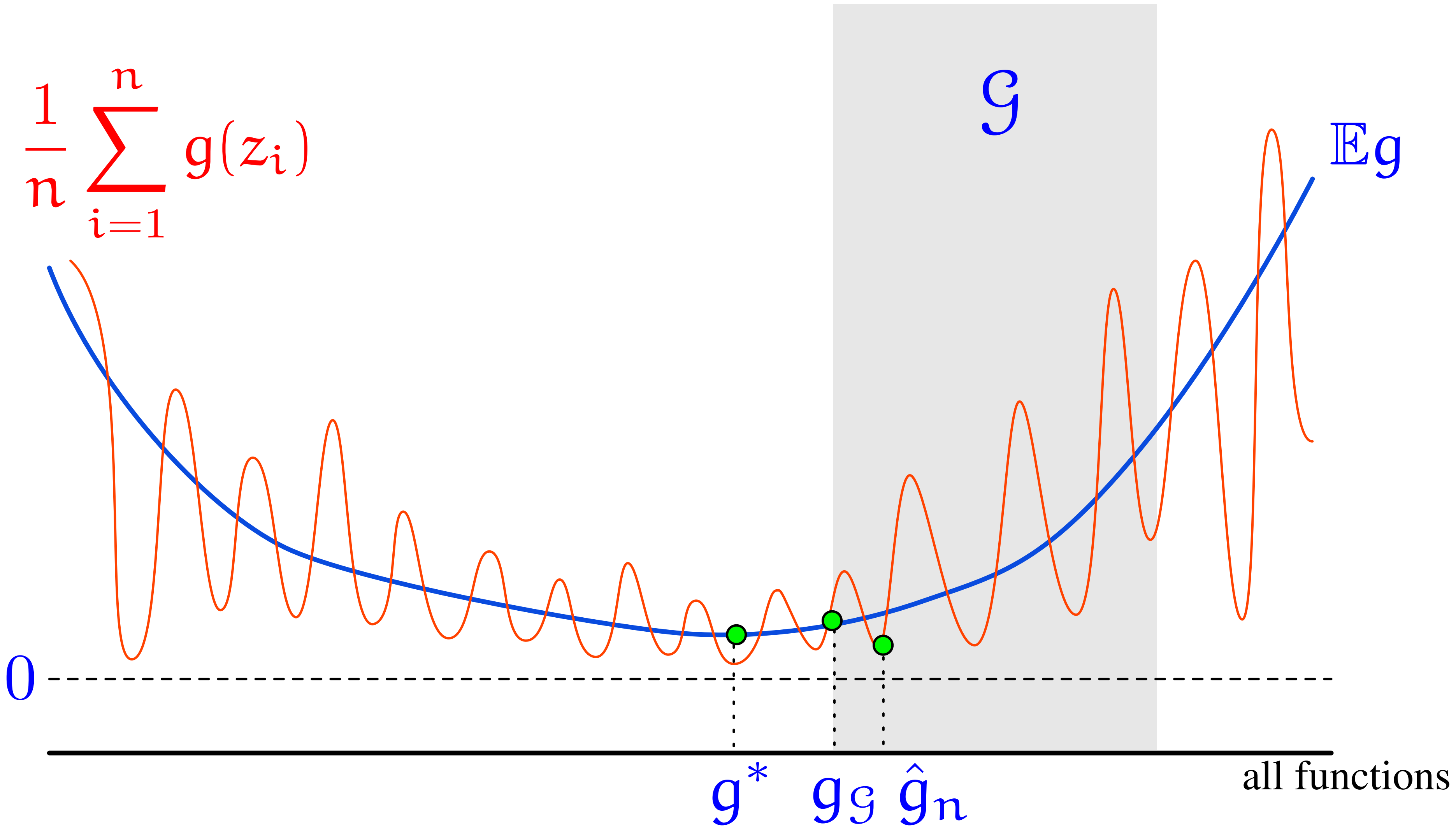
Empirical Process Viewpoint



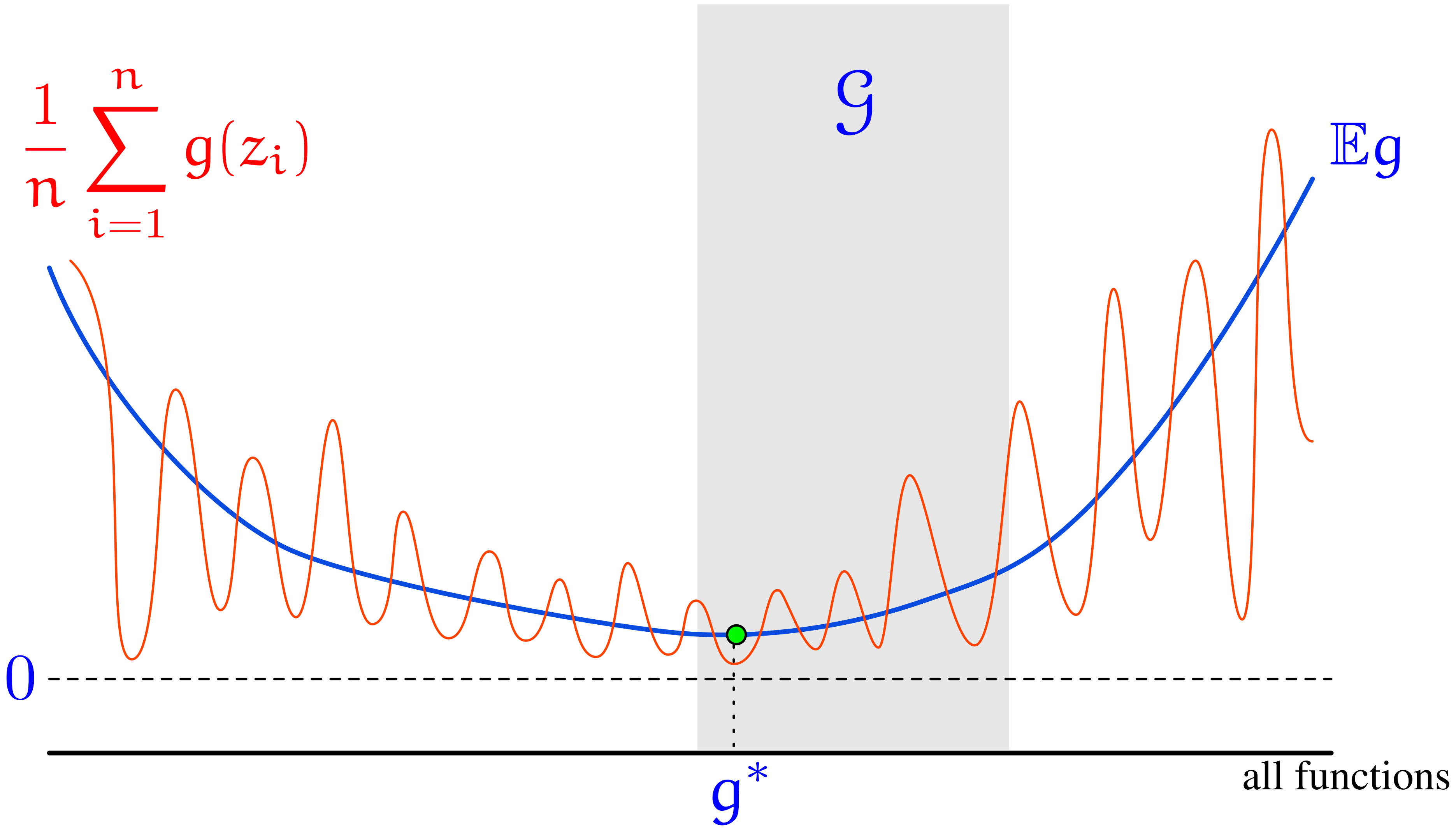
Empirical Process Viewpoint



Empirical Process Viewpoint



Empirical Process Viewpoint



Empirical Process Viewpoint

A *stochastic process* is a collection of random variables indexed by some set.

An *empirical process* is a stochastic process

$$\left\{ \mathbb{E}g(z) - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \right\}_{g \in \mathcal{G}}$$

indexed by a function class $\mathcal{G}$.

Uniform Law of Large Numbers:

$$\sup_{g \in \mathcal{G}} \left| \mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \right| \to 0$$

in probability.

Empirical Process Viewpoint

A *stochastic process* is a collection of random variables indexed by some set.

An *empirical process* is a stochastic process

$$\left\{ \mathbb{E} g(z) - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \right\}_{g\in\mathcal{G}}$$

indexed by a function class $\mathcal{G}$.

Uniform Law of Large Numbers:

$$\sup_{g\in\mathcal{G}} \left| \mathbb{E} g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \right| \to 0$$

in probability.

Key question: How "big" can $\mathcal{G}$ be for the supremum of the empirical process to still be manageable?

Union Bound (Boole's inequality)

Boole's inequality: for a finite or countable set of events,

$$\mathbb{P}\left(\cup_j A_j\right) \leq \sum_j \mathbb{P}\left(A_j\right)$$

Let $\mathcal{G} = \{g_1, \ldots, g_N\}$. Then

$$\mathbb{P}\left(\exists g \in \mathcal{G} \; : \; \mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) > \epsilon\right) \leq \sum_{j=1}^{N} \mathbb{P}\left(\mathbb{E}g_j - \frac{1}{n}\sum_{i=1}^{n} g_j(z_i) > \epsilon\right)$$

Assuming $\mathbb{P}\left(a \leq g(z_i) \leq b\right) = 1$ for every $g \in \mathcal{G}$,

$$\mathbb{P}\left(\sup_{g \in \mathcal{G}}\left\{\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i)\right\} > \epsilon\right) \leq N \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

Finite Class

Alternatively, we set $\delta = N \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$ and write

$$\mathbb{P}\left(\sup_{g\in\mathcal{G}}\left\{\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i)\right\} > (b-a)\sqrt{\frac{\log(N)+\log(1/\delta)}{2n}}\right) \leq \delta$$

Another way to write it: with probability at least $1-\delta$,

$$\sup_{g\in\mathcal{G}}\left\{\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i)\right\} \leq (b-a)\sqrt{\frac{\log(N)+\log(1/\delta)}{2n}}$$

Hence, with probability at least $1-\delta$, the ERM algorithm $\hat{f}_n$ for a class $\mathcal{F}$ of cardinality N satisfies

$$\mathbf{L}(\hat{f}_n) - \inf_{f\in\mathcal{F}} \mathbf{L}(f) \leq 2(b-a)\sqrt{\frac{\log(N)+\log(1/\delta)}{2n}}$$

assuming $a \leq \ell(f(x), y) \leq b$ for all $f \in \mathcal{F}$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$.

The constant 2 is due to the $\mathbf{L}(f_{\mathcal{F}}) - \hat{\mathbf{L}}(f_{\mathcal{F}})$ term. This is a loose upper bound.

Once again...

A take-away message is that the following two statements are worlds apart:

$$\boxed{\text{with probability at least } 1-\delta, \text{ for any } g \in \mathcal{G}, \quad \mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \leq \epsilon}$$

vs

$$\boxed{\text{for any } g \in \mathcal{G}, \text{ with probability at least } 1-\delta, \quad \mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \leq \epsilon}$$

The second statement follows from CLT, while the first statement is often difficult to obtain and only holds for some $\mathcal{G}$.

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Plan

- Measuring the complexity of function spaces.
- Definitions of VC dimension and scale sensitive versions.
- Necessary and sufficient conditions for uniform convergence.

Uniform convergence for classification

Our loss function is now $V(f(x), y) = \Theta(-yf(x))$ and our RKHS is $||f||_K^2 \leq M$.

Our goal is to bound the following

$$P\left\{\sup_{f\in\mathcal{H}:\|f\|_K^2\leq M} |I[f] - I_S[f]| > \epsilon\right\}.$$

For one function we could use the Chernoff bound

$$P\left\{|I[f] - I_S[f]| > \epsilon\right\} < 2\exp(-2\epsilon^2\ell).$$

Uniform convergence for classification (cont)

We then would want to use the union bound over the number of ”essential” functions in the class which we already determined. We have seen how to relate the ϵ in the bound with the r covering radius for square loss.

What about if $V(f(x), y) = \Theta(-yf(x))$?

Classification is scale insensitive

The key result in computing $r(\epsilon)$ was showing that if

$$||f_1(x) - f_2(x)||_\infty < r(\epsilon)$$

then

$$|V(f_1(x), y) - V(f_2(x), y)| \leq \epsilon \qquad \forall x, y.$$

For the classification loss function $\epsilon = 1$ and varying $r(\epsilon)$ has no effect.

Counting classification functions

Given ℓ points $\{(x_1, y_1), ..., (x_\ell, y_\ell)\}$, for every $f \in \mathcal{H}(M)$ we get different "labelings" $\{\Theta(-y_1 f(x_1)), ..., \Theta(-y_\ell f(x_\ell))\}$ (or, alternatively, different vertices of the $[0, 1]^\ell$ cube are spanned).

We define the random VC entropy as the number of labelings that can be implemented over $f \in \mathcal{H}(M)$ written as

$$\mathcal{N}^{\mathcal{H}(M)}((x_1, y_1), ..., (x_\ell, y_\ell)).$$

An obvious property of $\mathcal{N}^{\mathcal{H}(M)}((x_1, y_1), ..., (x_\ell, y_\ell))$ is:

$$\mathcal{N}^{\mathcal{H}(M)}((x_1, y_1), ..., (x_\ell, y_\ell)) \leq 2^\ell.$$

Counting classification functions

Notice that

$$\mathcal{N}^{\mathcal{H}(M)}((x_1, y_1), ..., (x_\ell, y_\ell)).$$

depends on data so we need to take the expectation to use it

$$\overline{\mathcal{N}} = \mathbb{E}_{x_1,y_1,...,x_\ell,y_\ell} \mathcal{N}^{\mathcal{H}(M)}((x_1, y_1), ..., (x_\ell, y_\ell)).$$

We can use the following bound

$$\mathbb{P}\left\{ \sup_{f \in \mathcal{H}: \|f\|_K^2 \leq M} |I[f] - I_S[f]| > \epsilon \right\} < 2\overline{\mathcal{N}} \, \text{exp}(-2\epsilon^2 \ell).$$

A necessary and sufficient condition

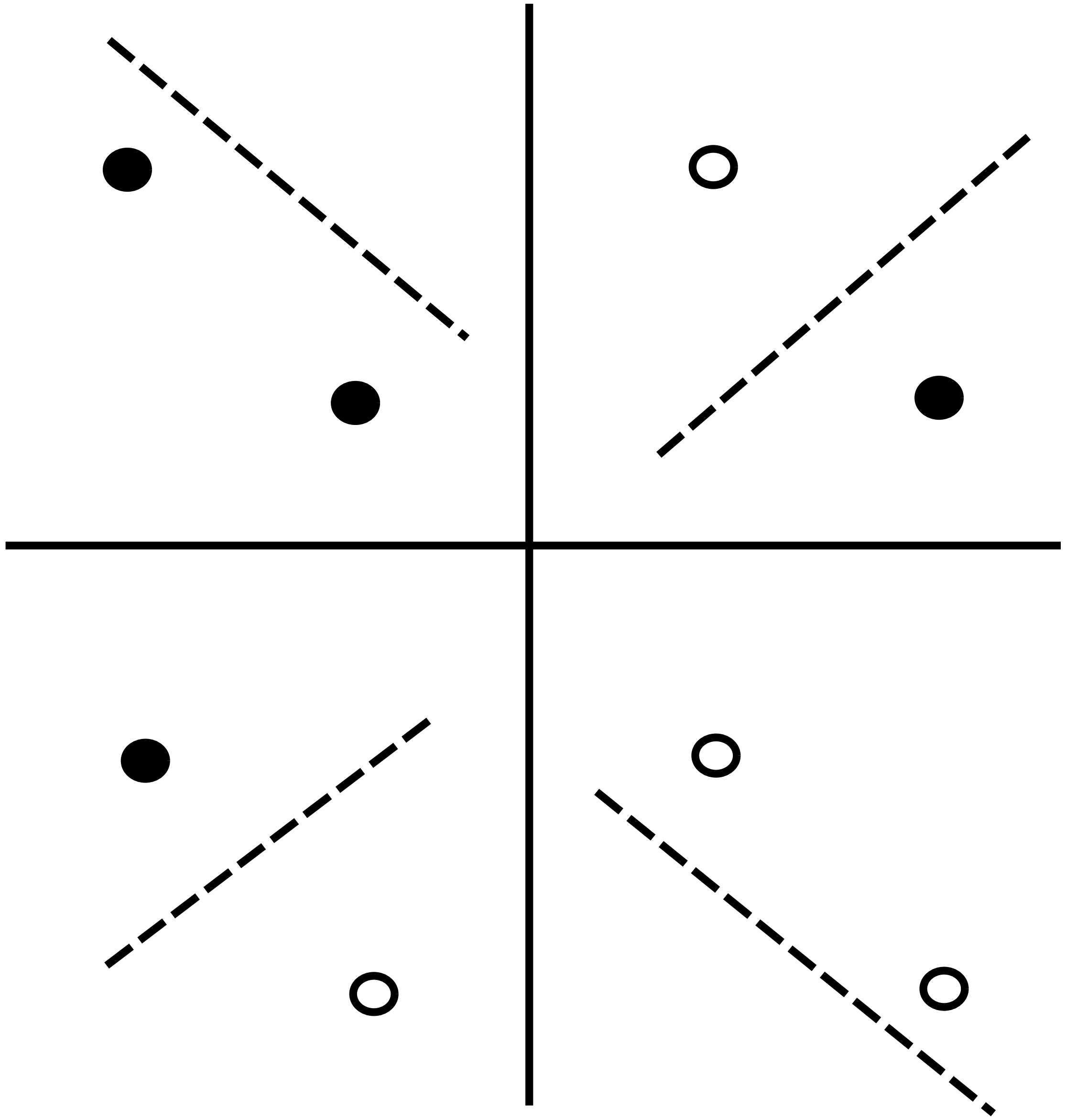
Iff

$$\lim_{\ell \to \infty} \frac{\log \overline{\mathcal{N}}}{\ell} \to 0,$$

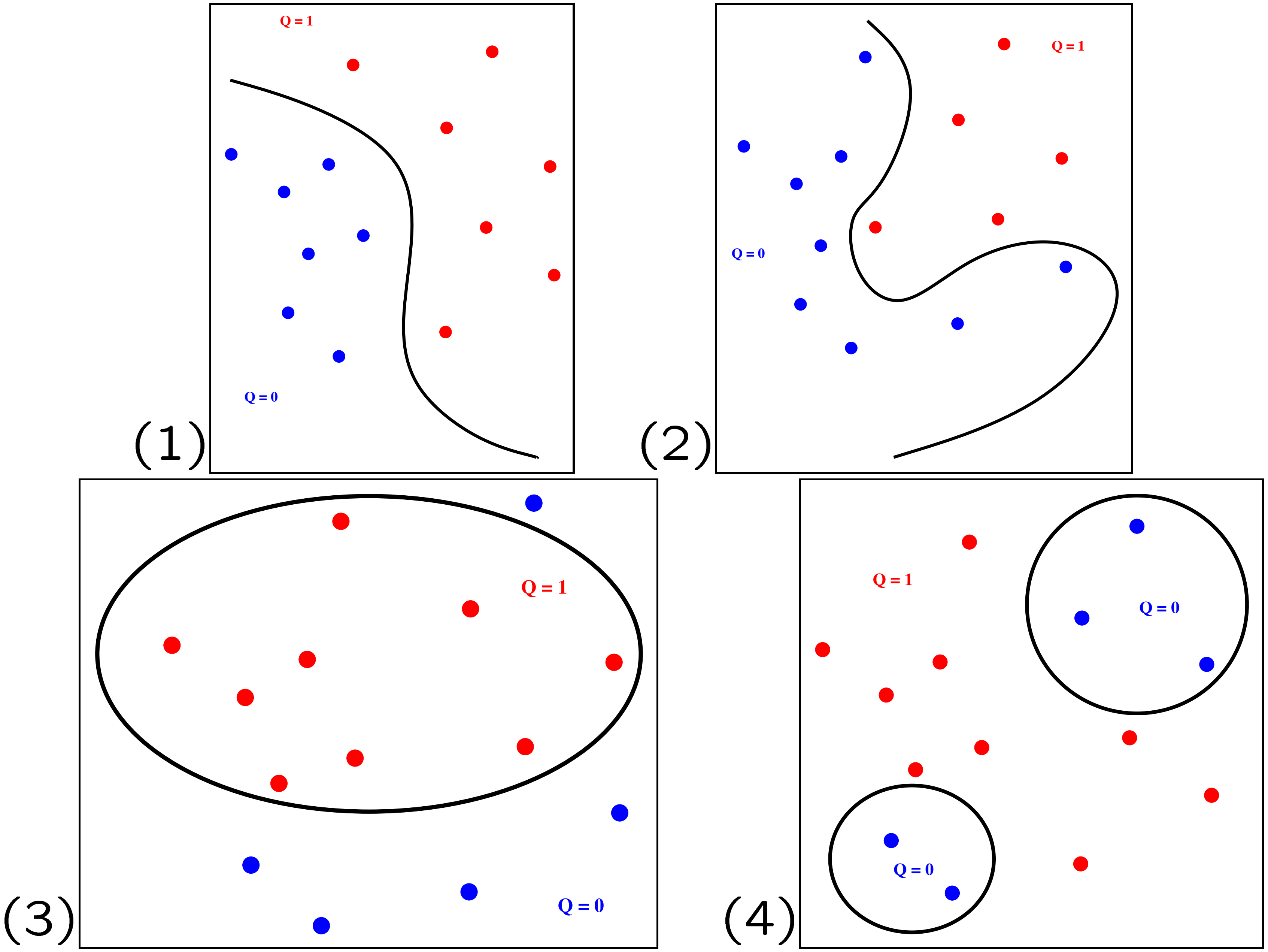
do we get uniform convergence in probability.

So the capacity can increase polynomially in ℓ but not exponentially.

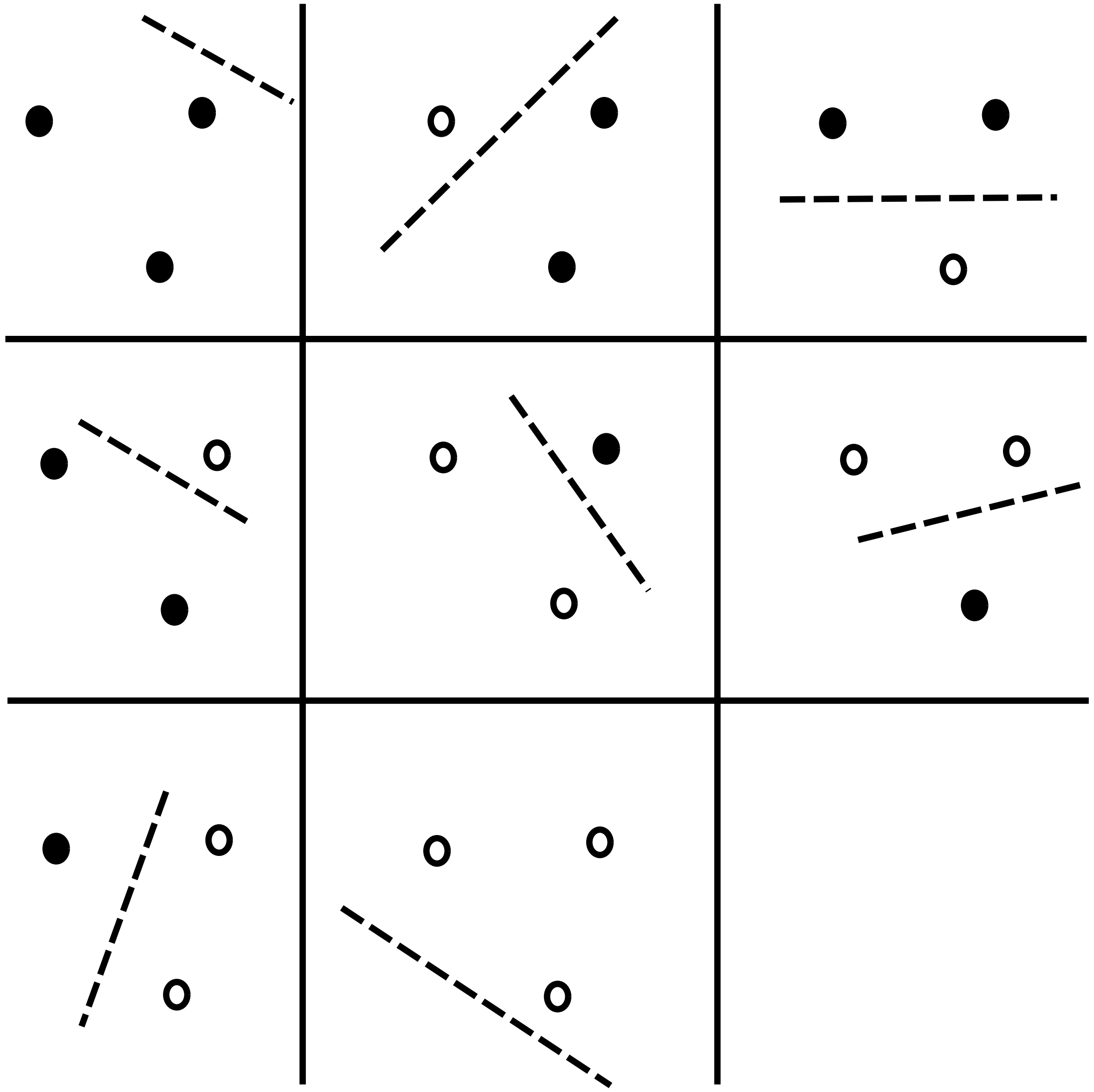
Implementation of different labelings



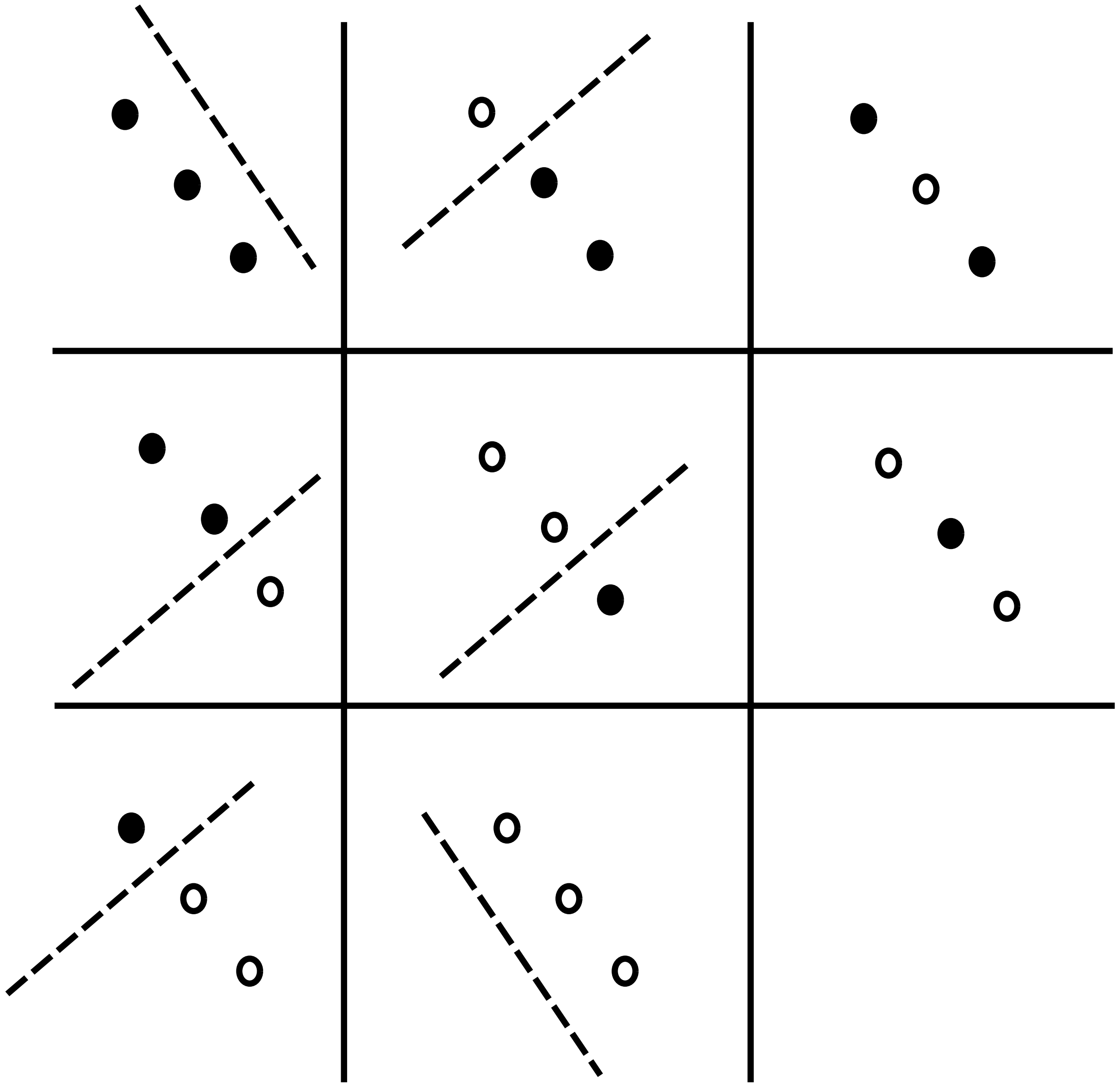
Implementation of different labelings



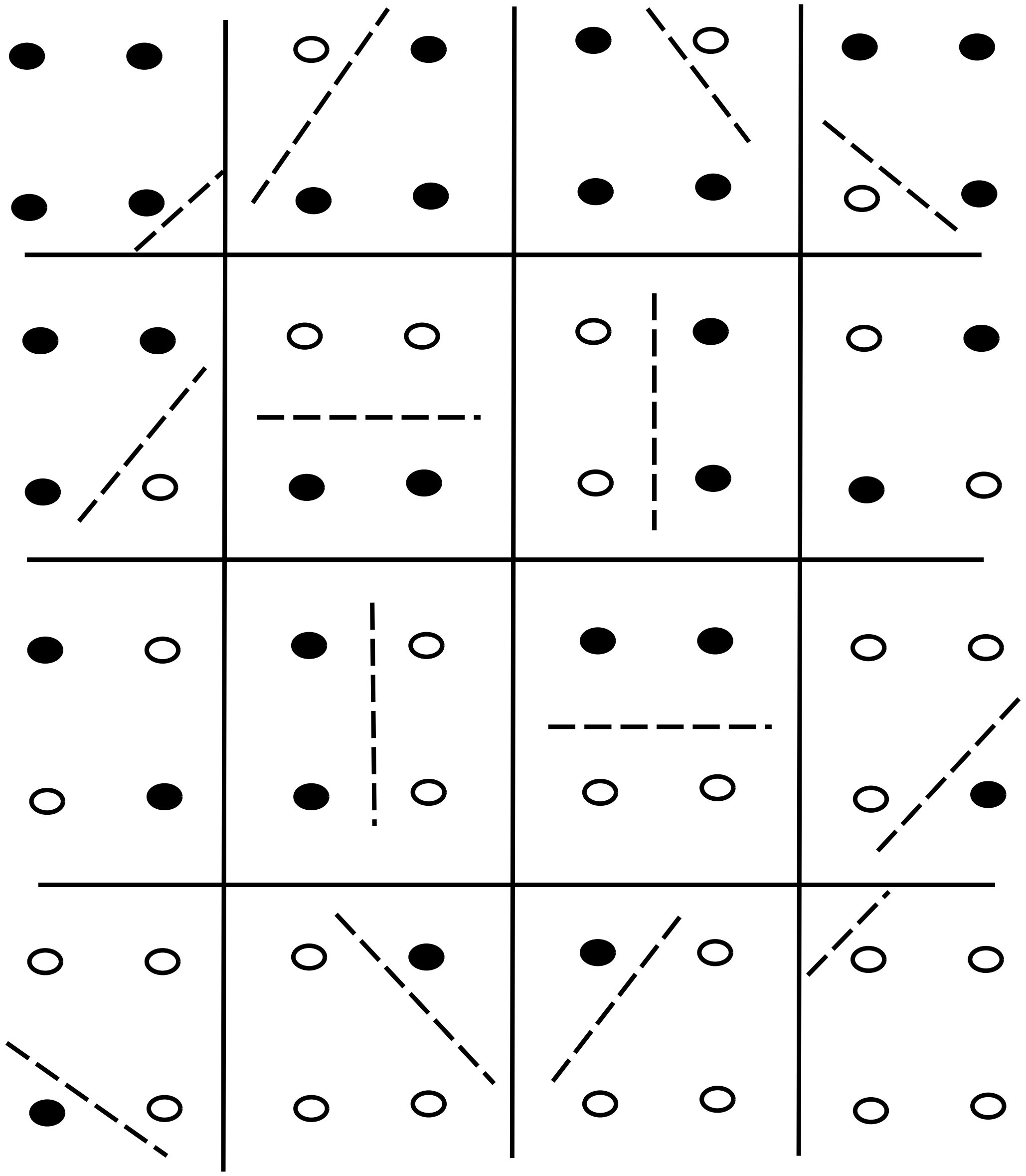
The 8 possible labelings of 3 points in 2D



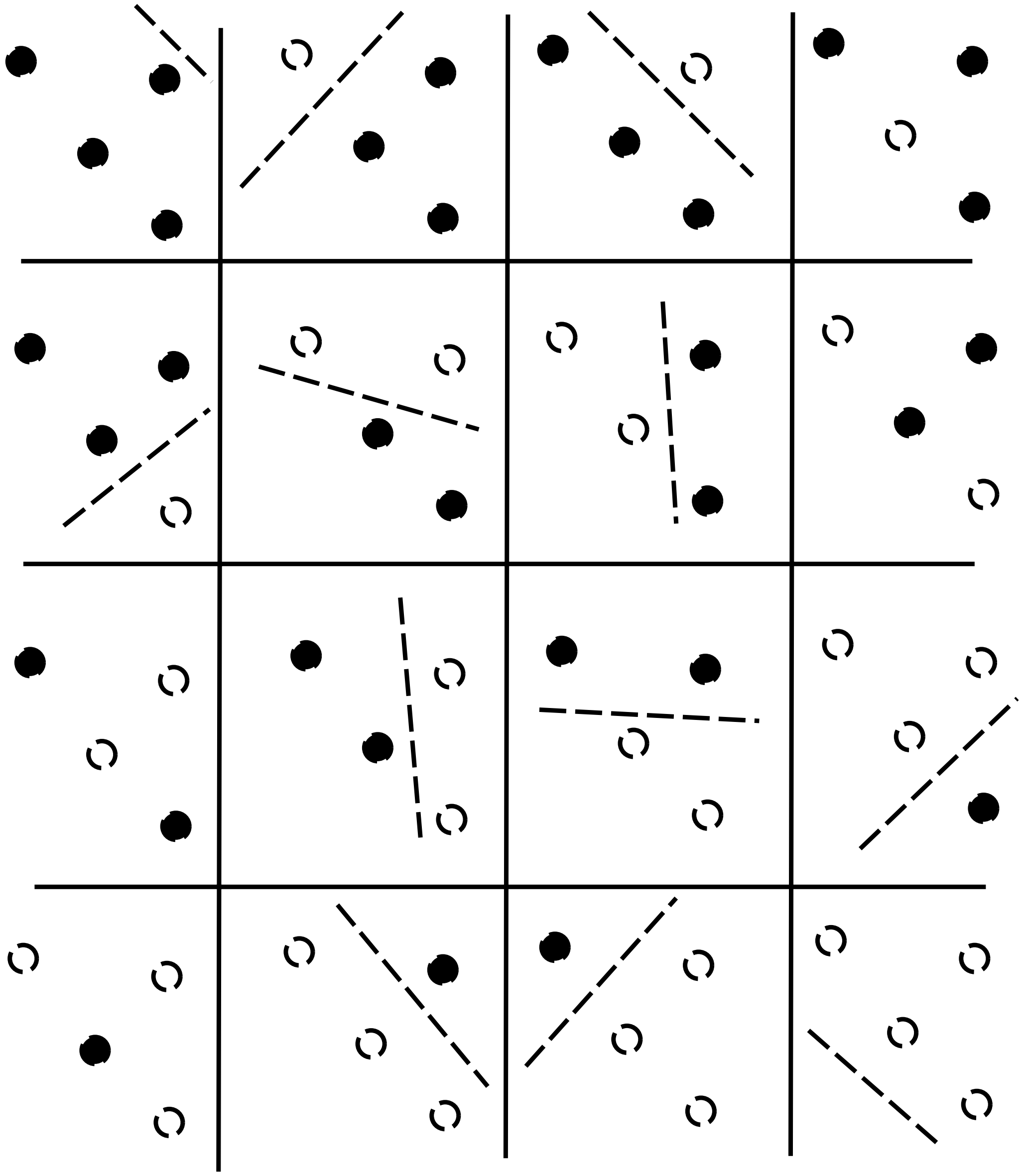
Example



Example



Example



How Many Labelings?
Sauer's Lemma

If the hypothesis space can separate h points in all possible (2^h ways), then $\ell > h$ points can be labeled in

$$\sum_{i=1}^{h} \binom{\ell}{i} < \left(\frac{e\ell}{h}\right)^h$$

possible ways and

$$\sum_{i=1}^{h} \binom{\ell}{i} < 2^\ell.$$

VC-dimension

The VC-dimension of a set of binary functions is h if and only if

- There is **at least one set of h points** that can be labeled in all possible ways;

- there is **no set of $h+1$ points** that can be labeled in all possible ways;

Classification

The finiteness of the VC-dimension of the set of functions $f \in \mathcal{H}(M)$ for the classification loss is a **necessary and sufficient** for uniform convergence of Ivanov regularization (empirical risk minimization in a bounded function class) for arbitrary probability distributions with a fast rate of convergence.

$$\overline{\mathcal{N}} \leq \left(\frac{e\ell}{h}\right)^{h}.$$

VC-bound

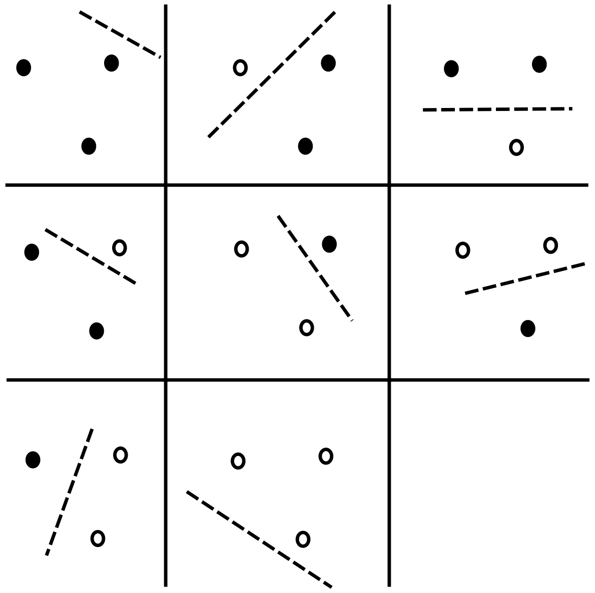
We can now bound the defect in the case of classification

$$P\left\{\sup_{f\in\mathcal{H}:\|f\|_K^2\le M}|I[f]-I_S[f]|>\epsilon\right\}<2\left(\frac{e\ell}{h}\right)^h\exp(-2\epsilon^2\ell).$$

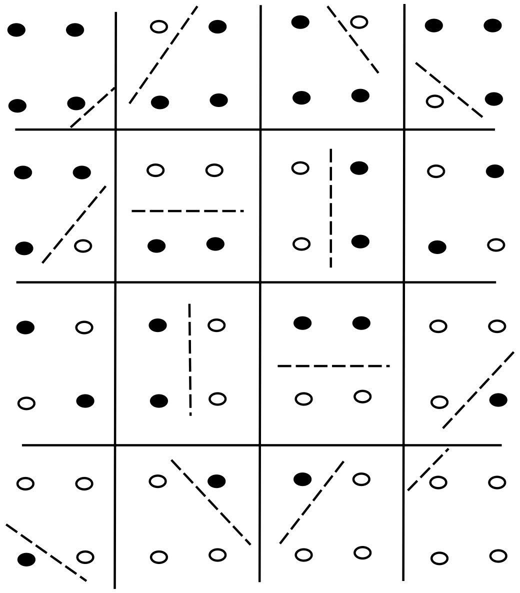
Which allows us to state that with probability $1-\delta$

$$I[f]\le I_S[f]+\sqrt{\frac{h\ln(e\ell/h)-\ln(\delta/2)}{\ell}}.$$

VC dimension of hyperplanes



all the possible labelings



not all the possible labelings

VC-dimension = 3

VC-dimension and free parameters

The VC-dimension is proportional, but not necessarily equal, to the number of parameters.

- For Multilayer Perceptrons with hard thresholds $h \propto n \ln n$ (Maass, 1994);

- For Multilayer perceptrons with standard sigmoid thresholds $h \propto n^2$ (Koiran and Sontag, 1995);

- For classification functions of the form $\theta(-y \sin(\alpha x))$ the VC-dimension is infinite;

Empirical covering numbers

Instead of using the sup norm as the metric of our cover we can use

$$d_{x_\ell}(f_1, f_2) = \max_{x_i} |f_1(x_i) - f_2(x_i)|.$$

The **empirical covering number** $\mathcal{N}(\mathcal{H}, r, d_{x_\ell})$ is the minimal $m \in \mathbb{N}$ such that there exists m disks in $\mathcal{H}$ with radius r covering function values at ℓ points.

Empirical covering numbers

Notice that

$$\mathcal{N}(\mathcal{H}, r, d_{x_\ell}).$$

depends on data so we need to take the expectation to use it

$$\overline{\mathcal{N}} = \mathbb{E}_S \mathcal{N}(\mathcal{H}, r, d_{x_\ell}).$$

A necessary and sufficient condition

Iff for any given $r > 0$

$$\lim_{\ell \to \infty} \frac{\log \overline{\mathcal{N}}}{\ell} \to 0,$$

do we get uniform convergence in probability.

So the capacity can increase polynomially in ℓ but not exponentially at any scale.

Is there a number like VC dimension for classification that can be used to bound the empirical cover ?

V_γ dimension and shattering

The V_γ-dimension of $\mathcal{F}_{\mathcal{H},V}$ is defined as the the maximum number h of vectors $\{(x_1, y_1), \ldots, (x_h, y_h)\}$ that can be separated into two classes in all 2^h possible ways using rules:

$$\begin{array}{l} \text{class 1 if: } V(y_i, f(x_i)) \geq s + \gamma \\ \text{class 0 if: } V(y_i, f(x_i)) \leq s - \gamma \end{array}$$

for some $s \geq 0$. If, for any number N, it is possible to find N points that can be separated in all possible ways, the V_γ-dimension is infinite.

Key result

(Alon et al. 93)
Finiteness of the V_γ dimension for every $\gamma > 0$ is a **necessary and sufficient** condition for distribution independent uniform convergence of the ERM method for real-valued functions.

(Mendelson and Vershynin 03)
Compactness of the $L2$ covering number for every scale $\epsilon > 0$ is a **necessary and sufficient** condition for distribution independent uniform convergence of the ERM method for real-valued functions.

V_γ dimension

The expectation of the cover is bounded by the V_γ dimension

$$\mathbb{E}_S \mathcal{N}(\mathcal{H}, r, d_{x_\ell}) \leq 2 \left(\frac{4\ell}{r^2}\right)^{h \log(2e\ell/(hr))}.$$

For the square loss bounded with the same constants as we saw in last class we get

$$\mathbb{P}\left\{\sup_{f \in \mathcal{H}} |I[f] - I_S[f]| \leq \epsilon\right\} \leq 1 - 4 \left(\frac{4\ell}{(\epsilon/8B')^2}\right)^{h \log(2e\ell/(h(\epsilon/8B')))} \exp(-\epsilon^2 \ell/B^2).$$

Countable Class: Weighted Union Bound

Let $\mathcal{G}$ be countable and fix a distribution w on $\mathcal{G}$ such that $\sum_{g \in \mathcal{G}} w(g) \leq 1$. For any $\delta > 0$, for any $g \in \mathcal{G}$

$$\mathbb{P}\left(\mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \geq (b-a)\sqrt{\frac{\log 1/w(g) + \log(1/\delta)}{2n}} \right) \leq \delta \cdot w(g)$$

by Hoeffding's inequality (easy to verify!). By the Union Bound,

$$\mathbb{P}\left(\exists g \in \mathcal{G} : \mathbb{E}g - \frac{1}{n}\sum_{i=1}^{n} g(z_i) \geq (b-a)\sqrt{\frac{\log 1/w(g) + \log(1/\delta)}{2n}} \right) \leq \delta \sum_{g \in \mathcal{G}} w(g) \leq \delta$$

Therefore, with probability at least $1-\delta$, for all $f \in \mathcal{F}$

$$\mathbf{L}(f) - \hat{\mathbf{L}}(f) \leq \underbrace{(b-a)\sqrt{\frac{\log 1/w(f) + \log(1/\delta)}{2n}}}_{\text{pen}_n(f)}$$

Countable Class: Weighted Union Bound

If $\hat{f}_n$ is a regularized ERM,

$$\begin{aligned}\mathbf{L}(\hat{f}_n) - \mathbf{L}(f_{\mathcal{F}}) &\le \left\{\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n) - \mathrm{pen}_n(\hat{f}_n)\right\} \\ &+ \left\{\hat{\mathbf{L}}(\hat{f}_n) + \mathrm{pen}_n(\hat{f}_n) - \hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathrm{pen}_n(f_{\mathcal{F}})\right\} \\ &+ \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\} + \mathrm{pen}_n(f_{\mathcal{F}}) \\ &\le \sup_{f \in \mathcal{F}} \left\{\mathbf{L}(f) - \hat{\mathbf{L}}(f) - \mathrm{pen}_n(f)\right\} + \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\} + \mathrm{pen}_n(f_{\mathcal{F}})\end{aligned}$$

So, (E) implies a bound on (B) when $\hat{f}_n$ is regularized ERM.
From the weighted union bound for a countable class:

$$\begin{aligned}\mathbf{L}(\hat{f}_n) - \mathbf{L}(f_{\mathcal{F}}) &\le \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\} + \mathrm{pen}_n(f_{\mathcal{F}}) \\ &\le 2(b-a)\sqrt{\frac{\log 1/w(f_{\mathcal{F}}) + \log(1/\delta)}{2n}}\end{aligned}$$

Uncountable Class: Compression Bounds

Let us make the dependence of the algorithm $\hat{f}_n$ on the training set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ explicit: $\hat{f}_n = \hat{f}_n[S]$.

Suppose $\mathcal{F}$ has the property that there exists a "compression function" C_k which selects from any dataset S of any size n a subset of k labeled examples $C_k(S) \subseteq S$ such that the algorithm can be written as

$$\hat{f}_n[S] = \hat{f}_k[C_k(S)]$$

Then,

$$\begin{aligned}
\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n) &= \mathbb{E}\ell(\hat{f}_k[C_k(S)](x), y) - \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{f}_k[C_k(S)](x_i), y_i) \\
&\leq \max_{I \subset \{1,\dots,n\}, |I| \leq k} \left\{ \mathbb{E}\ell(\hat{f}_k[S_I](x), y) - \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{f}_k[S_I](x_i), y_i) \right\}
\end{aligned}$$

Uncountable Class: Compression Bounds

Since $\hat{f}_k[S_I]$ only depends on k out of n points, the empirical average is "mostly out of sample". Adding and subtracting

$$\frac{1}{n}\sum_{(x',y')\in W} \ell(\hat{f}_k[S_I](x'),y')$$

for an additional set of i.i.d. random variables $W = \{(x'_1,y'_1),\ldots,(x'_k,y'_k)\}$ results in an upper bound

$$\max_{I\subset\{1,\ldots,n\},|I|\le k}\left\{\mathbb{E}\ell(\hat{f}_k[S_I](x),y) - \frac{1}{n}\sum_{(x,y)\in S\setminus S_I\cup W_{|I|}} \ell(\hat{f}_k[S_I](x),y)\right\} + \frac{(b-a)k}{n}$$

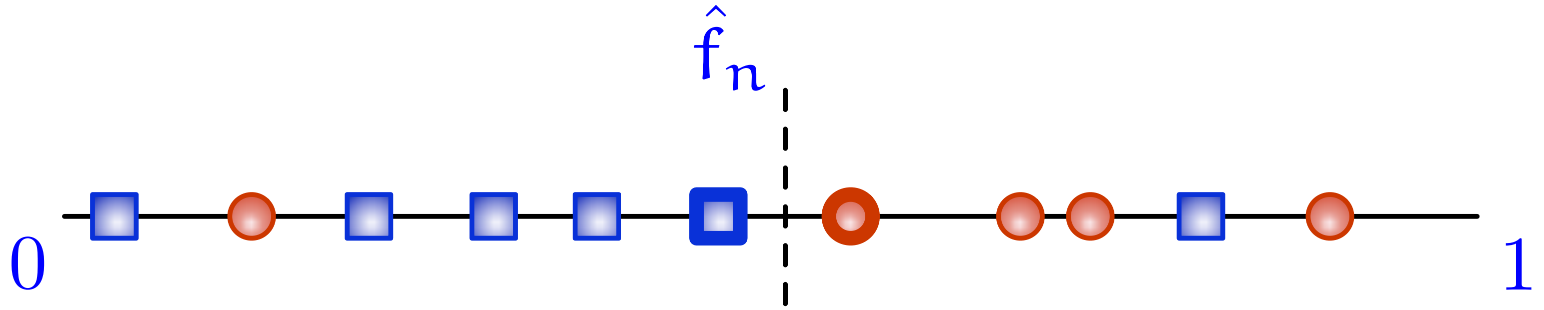
We appeal to the union bound over the $\binom{n}{k}$ possibilities, with a Hoeffding's bound for each. Then with probability at least $1-\delta$,

$$\mathbf{L}(\hat{f}_n) - \inf_{f\in\mathcal{F}} \mathbf{L}(f) \le 2(b-a)\sqrt{\frac{k\log(en/k)+\log(1/\delta)}{2n}} + \frac{(b-a)k}{n}$$

assuming $a \le \ell(f(x),y) \le b$ for all $f\in\mathcal{F}$, $x\in\mathcal{X}$, $y\in\mathcal{Y}$.

Example: Classification with Thresholds in 1D

- $\mathcal{X} = [0,1]$, $\mathcal{Y} = \{0,1\}$
- $\mathcal{F} = \left\{f_\theta : f_\theta(x) = \mathbf{I}_{\{x \geq \theta\}}, \theta \in [0,1]\right\}$
- $\ell(f_\theta(x), y) = \mathbf{I}_{\{f_\theta(x) \neq y\}}$



For any set of data $(x_1, y_1), \dots, (x_n, y_n)$, the ERM solution $\hat{f}_n$ has the property that the first occurrence x_l on the left of the threshold has label $y_l = 0$, while first occurrence x_r on the right – label $y_r = 1$.

Enough to take $k = 2$ and define $\hat{f}_n[S] = \hat{f}_2[(x_l, 0), (x_r, 1)]$.

Stability

Yet another way to limit the bias of $\ell(\hat{f}_n(x_i), y_i)$ as an estimate of $\mathbf{L}(\hat{f}_n)$ is through a notion of stability.

An algorithm $\hat{f}_n$ is *stable* if a change (or removal) of a single data point does not change (in a certain mathematical sense) the function $\hat{f}_n$ by much.

Of course, a dumb algorithm which outputs $\hat{f}_n = f_0$ without even looking at data is very stable and $\ell(\hat{f}_n(x_i), y_i)$ are independent random variables... But it is not a good algorithm! We would like to have an algorithm that both approximately minimizes the empirical error and is stable.

Turns out, certain types of regularization methods are stable. Example:

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_K^2$$

where $\|\cdot\|$ is the norm induced by the kernel of a reproducing kernel Hilbert space (RKHS) $\mathcal{F}$.

Summary so far

We proved upper bounds on $\mathbf{L}(\hat{f}_n) - \mathbf{L}(f_{\mathcal{F}})$ for

- ERM over a finite class
- Regularized ERM over a countable class (weighted union bound)
- ERM over classes $\mathcal{F}$ with the compression property
- ERM or Regularized ERM that are stable (only sketched it)

What about a more general situation? Is there a way to measure *complexity* of $\mathcal{F}$ that tells us whether ERM will succeed?

Outline

- Introduction
- Statistical Learning Theory
    - The Setting of SLT
    - Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
    - Tools from Probability, Empirical Processes
    - From Finite to Infinite Classes
    - Uniform Convergence, Symmetrization, and Rademacher Complexity
    - Large Margin Theory for Classification
    - Properties of Rademacher Complexity
    - Covering Numbers and Scale-Sensitive Dimensions
    - Faster Rates
    - Model Selection
- Sequential Prediction / Online Learning
    - Motivation
    - Supervised Learning
    - Online Convex and Linear Optimization
    - Online-to-Batch Conversion, SVM optimization

Uniform Convergence and Symmetrization

Let $z'_1, \dots, z'_n$ be another set of n i.i.d. random variables from P.
Let $\epsilon_1, \dots, \epsilon_n$ be i.i.d. Rademacher random variables:

$$\mathbb{P}(\epsilon_i = -1) = \mathbb{P}(\epsilon_i = +1) = 1/2$$

Let's get through a few manipulations:

$$\mathbb{E} \sup_{g \in \mathcal{G}} \left\{ \mathbb{E} g(z) - \frac{1}{n} \sum_{i=1}^{n} g(z_i) \right\} = \mathbb{E}_{z_{1:n}} \sup_{g \in \mathcal{G}} \left\{ \mathbb{E}_{z'_{1:n}} \left\{ \frac{1}{n} \sum_{i=1}^{n} g(z'_i) \right\} - \frac{1}{n} \sum_{i=1}^{n} g(z_i) \right\}$$

By Jensen's inequality, this is upper bounded by

$$\mathbb{E}_{z_{1:n}, z'_{1:n}} \sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} g(z'_i) - \frac{1}{n} \sum_{i=1}^{n} g(z_i) \right\}$$

which is equal to

$$\mathbb{E}_{\epsilon_{1:n}} \mathbb{E}_{z_{1:n}, z'_{1:n}} \sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \left(g(z'_i) - g(z_i) \right) \right\}$$

Uniform Convergence and Symmetrization

$$
\begin{aligned}
\mathbb{E}_{\epsilon_{1:n}} \mathbb{E}_{z_{1:n}, z'_{1:n}} \sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \epsilon_i (g(z'_i) - g(z_i)) \right\} \\
&\leq \mathbb{E} \sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g(z'_i) \right\} + \mathbb{E} \sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} -\epsilon_i g(z_i) \right\} \\
&= 2\mathbb{E} \sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g(z_i) \right\}
\end{aligned}
$$

The *empirical Rademacher averages* of $\mathcal{G}$ are defined as

$$\widehat{\mathscr{R}}_n(\mathcal{G}) = \mathbb{E} \left[\sup_{g \in \mathcal{G}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \epsilon_i g(z_i) \right\} \middle| z_1, \dots, z_n \right]$$

The *Rademacher average* (or *Rademacher complexity*) of $\mathcal{G}$ is

$$\mathscr{R}_n(\mathcal{G}) = \mathbb{E}_{z_{1:n}} \widehat{\mathscr{R}}_n(\mathcal{G})$$

Classification: Loss Function Disappears

Let us focus on binary classification with indicator loss and let $\mathcal{F}$ be a class of $\{0,1\}$-valued functions. We have

$$\ell(f(x),y) = \mathbf{I}_{\{f(x)\neq y\}} = (1-2y)f(x)+y$$

and thus

$$\begin{aligned}\widehat{\mathscr{R}}_n(\mathcal{G}) &= \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left\{\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(f(x_i)(1-2y_i)+y_i)\right\}\,\middle|\,(x_1,y_1)\ldots,(x_n,y_n)\right] \\ &= \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left\{\frac{1}{n}\sum_{i=1}^{n}\epsilon_i f(x_i)\right\}\,\middle|\,x_1,\ldots,x_n\right] = \widehat{\mathscr{R}}_n(\mathcal{F})\end{aligned}$$

because, given $y_1,\ldots,y_n$, the distribution of $\epsilon_i(1-2y_i)$ is the same as ϵ_i.

Vapnik-Chervonenkis Theory for Classification

We are now left examining

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left\{\frac{1}{n}\sum_{i=1}^{n}\epsilon_i f(x_i)\right\}\,\middle|\, x_1,\ldots,x_n\right]$$

Given $x_1,\ldots,x_n$, define the projection of $\mathcal{F}$ onto sample:

$$\mathcal{F}|_{x_{1:n}} = \left\{(f(x_1),\ldots,f(x_n)) \in \{0,1\}^n : f\in\mathcal{F}\right\} \subseteq \{0,1\}^n$$

Clearly, this is a finite set and

$$\widehat{\mathscr{R}}_n(\mathcal{F}) = \mathbb{E}_{\epsilon_{1:n}} \max_{v\in\mathcal{F}|_{x_{1:n}}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i v_i \le \sqrt{\frac{2\log\mathrm{card}(\mathcal{F}|_{x_{1:n}})}{n}}$$

This is because a maximum of N (sub)Gaussian random variables $\sim \sqrt{\log N}$.

The bound is nontrivial as long as $\log\mathrm{card}(\mathcal{F}|_{x_{1:n}}) = o(n)$.

Vapnik-Chervonenkis Theory for Classification

The *growth function* is defined as

$$\Pi_{\mathcal{F}}(n) = \max\left\{\text{card}(\mathcal{F}|_{x_1,\ldots,x_n}) : x_1,\ldots,x_n \in \mathcal{X}\right\}$$

The growth function measures *expressiveness* of $\mathcal{F}$. In particular, if $\mathcal{F}$ can produce all possible signs (that is, $\Pi_{\mathcal{F}}(n) = 2^n$), the bound becomes useless.

We say that $\mathcal{F}$ *shatters* some set $x_1,\ldots,x_n$ if $\mathcal{F}|_{x^n} = \{0,1\}^n$.

The *Vapnik-Chervonenkis (VC) dimension* of the class $\mathcal{F}$ is defined as

$$vc(\mathcal{F}) = \max\left\{d : \Pi_{\mathcal{F}}(t) = 2^t\right\}$$

Vapnik-Chervonenkis-Sauer-Shelah Lemma: If $d = vc(\mathcal{F}) < \infty$, then

$$\Pi_{\mathcal{F}}(n) \le \sum_{i=0}^{d} \binom{n}{d} \le \left(\frac{en}{d}\right)^d$$

Vapnik-Chervonenkis Theory for Classification

Conclusion: for any $\mathcal{F}$ with $\mathrm{vc}(\mathcal{F}) < \infty$, the ERM algorithm satisfies

$$\mathbb{E}\left\{\mathbf{L}(\hat{f}_n) - \inf_{f\in\mathcal{F}} \mathbf{L}(f)\right\} \le 2\sqrt{\frac{2d\log(en/d)}{n}}$$

While we proved the result in expectation, the same type of bound holds with high probability.

VC dimension is a combinatorial dimension of a binary-valued function class. Its finiteness is necessary and sufficient for learnability if we place no assumptions on the distribution P.

Remark: the bound is similar to that obtained through compression. In fact, the exact relationship between compression and VC dimension is still an open question.

Vapnik-Chervonenkis Theory for Classification

Examples of VC classes:

- Half-spaces $\mathcal{F} = \left\{\mathbf{I}_{\{\langle w,x\rangle+b\geq 0\}} : w \in \mathbb{R}^d, \|w\| = 1, b \in \mathbb{R}\right\}$ has $\mathrm{vc}(\mathcal{F}) = d + 1$
- For a vector space $\mathcal{H}$ of dimension d, VC dimension of $\mathcal{F} = \{\mathbf{I}_{\{h(x)\geq 0\}} : h \in \mathcal{H}\}$ is at most d
- The set of Euclidean balls $\mathcal{F} = \left\{\mathbf{I}_{\left\{\sum_{i=1}^d \|x_i - a_i\|^2 \leq b\right\}} : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$ has VC dimension at most $d + 2$.
- Functions that can be computed using a finite number of arithmetic operations (see *(Goldberg and Jerrum, 1995)*)

However: $\mathcal{F} = \left\{f_\alpha(x) = \mathbf{I}_{\{\sin(\alpha x)\geq 0\}} : \alpha \in \mathbb{R}\right\}$ has infinite VC dimension, so it is not correct to think of VC dimension as the number of parameters!

Vapnik-Chervonenkis Theory for Classification

Examples of VC classes:

- Half-spaces $\mathcal{F} = \left\{\mathbf{I}_{\{\langle w,x\rangle + b \geq 0\}} : w \in \mathbb{R}^d, \|w\| = 1, b \in \mathbb{R}\right\}$ has $\mathrm{vc}(\mathcal{F}) = d + 1$
- For a vector space $\mathcal{H}$ of dimension d, VC dimension of $\mathcal{F} = \{\mathbf{I}_{\{h(x)\geq 0\}} : h \in \mathcal{H}\}$ is at most d
- The set of Euclidean balls $\mathcal{F} = \left\{\mathbf{I}_{\left\{\sum_{i=1}^d \|x_i - a_i\|^2 \leq b\right\}} : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$ has VC dimension at most $d + 2$.
- Functions that can be computed using a finite number of arithmetic operations (see *(Goldberg and Jerrum, 1995)*)

However: $\mathcal{F} = \left\{f_\alpha(x) = \mathbf{I}_{\{\sin(\alpha x) \geq 0\}} : \alpha \in \mathbb{R}\right\}$ has infinite VC dimension, so it is not correct to think of VC dimension as the number of parameters!

Unfortunately, the VC theory is unable to explain the good performance of neural networks and Support Vector Machines! This prompted the development of a margin-based theory.

Outline

- Introduction
- Statistical Learning Theory
  - The Setting of SLT
  - Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
  - Tools from Probability, Empirical Processes
  - From Finite to Infinite Classes
  - Uniform Convergence, Symmetrization, and Rademacher Complexity
  - Large Margin Theory for Classification
  - Properties of Rademacher Complexity
  - Covering Numbers and Scale-Sensitive Dimensions
  - Faster Rates
  - Model Selection
- Sequential Prediction / Online Learning
  - Motivation
  - Supervised Learning
  - Online Convex and Linear Optimization
  - Online-to-Batch Conversion, SVM optimization

Classification with Real-Valued Functions

Many methods use

$$\mathbb{I}(\mathcal{F}) = \{\mathbf{I}_{\{f \geq 0\}} : f \in \mathcal{F}\}$$

for classification. The VC dimension can be very large, yet in practice the methods work well.

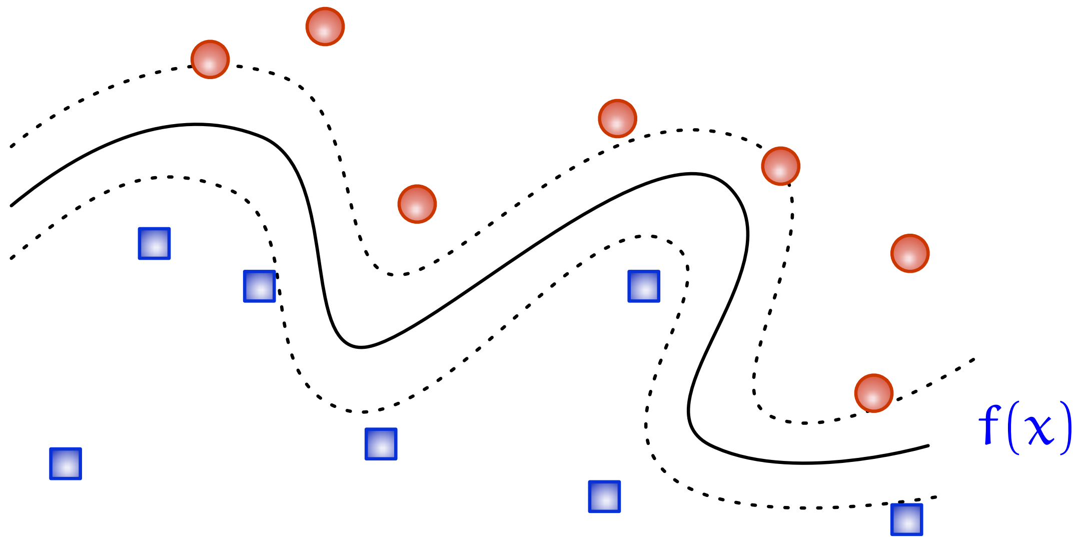
Example: $f(x) = f_w(x) = \langle w, \psi(x) \rangle$ where ψ is a mapping to a high-dimensional feature space (see Kernel Methods). The VC dimension of the set is typically huge (equal to the dimensionality of $\psi(x)$) or infinite, yet the methods perform well!

Is there an explanation beyond VC theory?

Margins

Hard margin:

$$\exists f \in \mathcal{F} : \quad \forall i, \qquad y_i f(x_i) \geq \gamma$$



More generally, we hope to have

$$\exists f \in \mathcal{F} : \quad \frac{\mathrm{card}(\{i : y_i f(x_i) < \gamma\})}{n} \quad \text{is small}$$

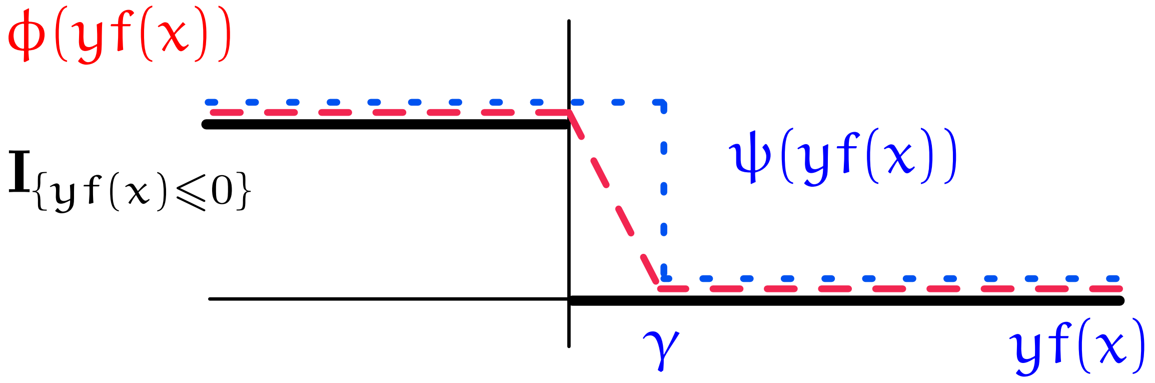
Surrogate Loss

Define

$$\phi(s) = \begin{cases} 1 & \text{if } s \le 0 \\ 1 - s/\gamma & \text{if } 0 < s < \gamma \\ 0 & \text{if } s \ge \gamma \end{cases}$$

Then: $\mathbf{I}_{\{y \neq \text{sign}(f(x))\}} = \mathbf{I}_{\{yf(x) \le 0\}} \le \phi(yf(x)) \le \psi(yf(x)) = \mathbf{I}_{\{yf(x) \le \gamma\}}$

The function ϕ is an example of a *surrogate loss function.*



Let

$$\mathbf{L}_{\phi}(f) = \mathbb{E}\phi(yf(x)) \qquad \text{and} \qquad \hat{\mathbf{L}}_{\phi}(f) = \frac{1}{n}\sum_{i=1}^{n}\phi(y_i f(x_i))$$

Then

$$\mathbf{L}(f) \le \mathbf{L}_{\phi}(f), \quad \hat{\mathbf{L}}_{\phi}(f) \le \hat{\mathbf{L}}_{\psi}(f)$$

Surrogate Loss

Now consider uniform deviations for the surrogate loss:

$$\mathbb{E} \sup_{f \in \mathcal{F}} \left\{ \mathbf{L}_\phi(f) - \hat{\mathbf{L}}_\phi(f) \right\}$$

We had shown that this quantity is at most $2\mathscr{R}_n(\phi(\mathcal{F}))$ for

$$\phi(\mathcal{F}) = \left\{ g(z) = \phi(yf(x)) : f \in \mathcal{F} \right\}$$

A useful property of Rademacher averages:

$$\mathscr{R}_n(\phi(\mathcal{F})) \le L \mathscr{R}_n(\mathcal{F}) \qquad \text{if } \phi \text{ is } L\text{-Lipschitz.}$$

Observe that in our example ϕ is $1/\gamma$-Lipschitz. Hence,

$$\mathbb{E} \sup_{f \in \mathcal{F}} \left\{ \mathbf{L}_\phi(f) - \hat{\mathbf{L}}_\phi(f) \right\} \le \frac{2}{\gamma} \mathscr{R}_n(\mathcal{F})$$

Margin Bound

Same result in high probability: with probability at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} \{\mathbf{L}_\phi(f) - \hat{\mathbf{L}}_\phi(f)\} \le \frac{2}{\gamma}\mathscr{R}_n(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

With probability at least $1 - \delta$, for all $f \in \mathcal{F}$

$$\mathbf{L}(f) \le \hat{\mathbf{L}}_\psi(f) + \frac{2}{\gamma}\mathscr{R}_n(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

If $\hat{f}_n$ is minimizing margin loss

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \phi(y_i f(x_i))$$

then with probability at least $1 - \delta$

$$\mathbf{L}(\hat{f}_n) \le \inf_{f \in \mathcal{F}} \mathbf{L}_\psi(f) + \frac{4}{\gamma}\mathscr{R}_n(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}}$$

Note: ϕ assumes knowledge of γ, but this assumption can be removed.

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Useful Properties

1. If $\mathcal{F} \subseteq \mathcal{G}$, then $\widehat{\mathscr{R}}_n(\mathcal{F}) \leq \widehat{\mathscr{R}}_n(\mathcal{G})$
2. $\widehat{\mathscr{R}}_n(\mathcal{F}) = \widehat{\mathscr{R}}_n(\mathrm{conv}(\mathcal{F}))$
3. For any $c \in \mathbb{R}$, $\widehat{\mathscr{R}}_n(c\mathcal{F}) = |c|\widehat{\mathscr{R}}_n(\mathcal{F})$
4. If $\phi : \mathbb{R} \mapsto \mathbb{R}$ is L-Lipschitz (that is, $\phi(a) - \phi(b) \leq L|a - b|$ for all $a, b \in \mathbb{R}$), then
$$\widehat{\mathscr{R}}_n(\phi \circ \mathcal{F}) \leq L\widehat{\mathscr{R}}_n(\mathcal{F})$$

Rademacher Complexity of Kernel Classes

- Feature map $\phi : \mathcal{X} \mapsto \ell_2$ and p.d. kernel $K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$
- The set $\mathcal{F}_B = \left\{ f(x) = \langle w, \phi(x) \rangle : \|w\| \le B \right\}$ is a ball in $\mathcal{H}$
- Reproducing property $f(x) = \langle f, K(x, \cdot) \rangle$

An easy calculation shows that empirical Rademacher averages are upper bounded as

$$\begin{aligned}
\widehat{\mathscr{R}}_n(\mathcal{F}_B) &= \mathbb{E} \sup_{f \in \mathcal{F}_1} \frac{1}{n} \sum_{i=1}^n \epsilon_i f(x_i) = \mathbb{E} \sup_{f \in \mathcal{F}_B} \frac{1}{n} \sum_{i=1}^n \epsilon_i \langle f, K(x_i, \cdot) \rangle \\
&= \mathbb{E} \sup_{f \in \mathcal{F}_B} \left\langle f, \frac{1}{n} \sum_{i=1}^n \epsilon_i K(x_i, \cdot) \right\rangle = B \cdot \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \epsilon_i K(x_i, \cdot) \right\| \\
&= \frac{B}{n} \mathbb{E} \left(\sum_{i,j=1}^n \epsilon_i \epsilon_j \langle K(x_i, \cdot), K(x_j, \cdot) \rangle \right)^{-1/2} \\
&\le \frac{B}{n} \left(\sum_{i=1}^n K(x_i, x_i) \right)^{-1/2}
\end{aligned}$$

A data-independent bound of $O(B\kappa/\sqrt{n})$ can be obtained if $\sup_{x \in \mathcal{X}} K(x, x) \le \kappa^2$. Then κ and B are the effective “dimensions”.

Other Examples

Using properties of Rademacher averages, we may establish guarantees for learning with neural networks, decision trees, and so on.

Powerful technique, typically requires only a few lines of algebra.

Occasionally, covering numbers and scale-sensitive dimensions can be easier to deal with.

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Real-Valued Functions: Covering Numbers

Consider

- a class $\mathcal{F}$ of $[-1,1]$-valued functions
- let $\mathcal{Y} = [-1,1]$, $\ell(f(x),y) = |f(x) - y|$

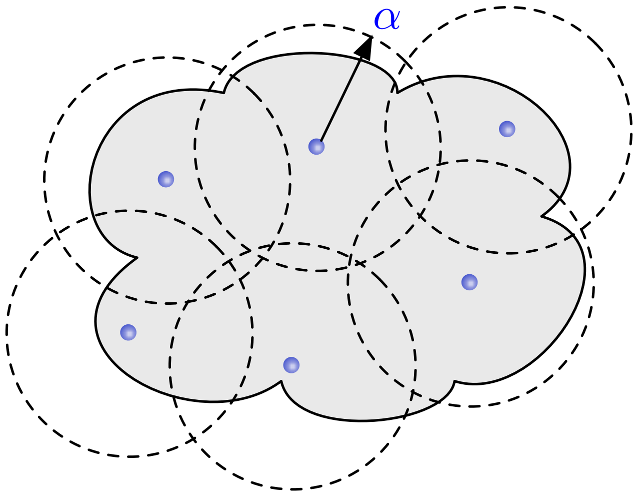
We have

$$\mathbb{E} \sup_{f \in \mathcal{F}} \mathbf{L}(f) - \hat{\mathbf{L}}(f) \leq 2\mathbb{E}_{x_{1:n}} \widehat{\mathscr{R}}_n(\mathcal{F})$$

For real-valued functions the cardinality of $\mathcal{F}|_{x_{1:n}}$ is infinite. However, similar functions f and f' with

$$(f(x_1), \ldots, f(x_n)) \approx (f'(x_1), \ldots, f'(x_n))$$

should be treated as the same.



Real-Valued Functions: Covering Numbers

Given $\alpha > 0$, suppose we can find $V \subset [-1,1]^n$ of finite cardinality such that

$$\forall f, \exists v^f \in V, \quad \text{s.t.} \quad \frac{1}{n}\sum_{i=1}^{n} |f(x_i) - v_i^f| \le \alpha$$

Then

$$\begin{aligned}
\widehat{\mathscr{R}}_n(\mathcal{F}) &= \mathbb{E}_{\epsilon_{1:n}} \sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i f(x_i) \\
&= \mathbb{E}_{\epsilon_{1:n}} \sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i (f(x_i) - v_i^f) + \mathbb{E}_{\epsilon_{1:n}} \sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i v_i^f \\
&\le \alpha + \mathbb{E}_{\epsilon_{1:n}} \max_{v \in V} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i v_i
\end{aligned}$$

Now we are back to the set of finite cardinality:

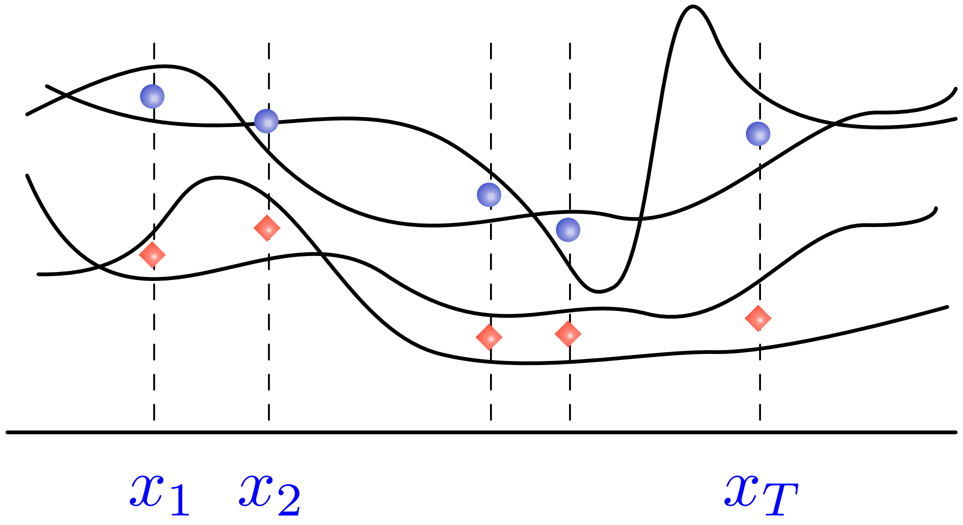
$$\widehat{\mathscr{R}}_n(\mathcal{F}) \le \alpha + \sqrt{\frac{2\log \operatorname{card}(V)}{n}}$$

Real-Valued Functions: Covering Numbers

Such a set V is called an α-cover (or α-net). More precisely, a set V is an α-cover with respect to ℓ_p norm if

$$\forall f, \exists v^f \in V, \quad \text{s.t.} \quad \frac{1}{n}\sum_{i=1}^{n} |f(x_i) - v_i^f|^p \le \alpha^p$$

The size of the smallest α-cover is denoted by $\mathcal{N}_p(\mathcal{F}|_{x_{1:n}}, \alpha)$.



Above : Two sets of levels provide an α-cover for the four functions. Only the values of functions on $x_1, \ldots, x_T$ are relevant.

Real-Valued Functions: Covering Numbers

We have proved that for any $x_1, \ldots, x_n$,

$$\widehat{\mathscr{R}}_n(\mathcal{F}) \leq \inf_{\alpha \geq 0} \left\{ \alpha + \frac{1}{\sqrt{n}} \sqrt{2 \log \operatorname{card}(\mathcal{N}_1(\mathcal{F}|_{x_{1:n}}, \alpha))} \right\}$$

A better bound (called Dudley entropy integral):

$$\widehat{\mathscr{R}}_n(\mathcal{F}) \leq \inf_{\alpha \geq 0} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_{\alpha}^{1} \sqrt{2 \log \operatorname{card}(\mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \delta))} \, d\delta \right\}$$

Example: Nondecreasing functions.

Consider the set $\mathcal{F}$ of nondecreasing functions $\mathbb{R} \mapsto [-1,1]$.

While $\mathcal{F}$ is a very large set, $\mathcal{F}|_{x_{1:n}}$ is not that large:

$$\mathcal{N}_1(\mathcal{F}|_{x_{1:n}}, \alpha) \le \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) \le n^{2/\alpha}.$$

The first bound on the previous slide yields

$$\inf_{\alpha \ge 0} \left\{ \alpha + \frac{1}{\sqrt{\alpha n}} \sqrt{4 \log(n)} \right\} = \tilde{O}(n^{-1/3})$$

while the second bound (the Dudley entropy integral)

$$\inf_{\alpha \ge 0} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_{\alpha}^{1} \sqrt{4/\delta \log(n)} d\delta \right\} = \tilde{O}(n^{-1/2})$$

where the $\tilde{O}$ notation hides logarithmic factors.

Scale-Sensitive Dimensions

We say that $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ α-shatters a set $(x_1, \dots, x_T)$ if there exist $(y_1, \dots, y_T) \in \mathbb{R}^T$ (called a *witness to shattering*) with the following property:

$$\forall (b_1, \dots, b_T) \in \{0, 1\}^T, \ \exists f \in \mathcal{F} \quad \text{s.t.}$$

$$f(x_t) > y_t + \frac{\alpha}{2} \ \text{ if } \ b_t = 1 \quad \text{and} \quad f(x_t) < y_t - \frac{\alpha}{2} \ \text{ if } \ b_t = 0$$

The *fat-shattering dimension* of $\mathcal{F}$ at scale α, denoted by $\text{fat}(\mathcal{F}, \alpha)$, is the size of the largest α-shattered set.

Scale-Sensitive Dimensions

We say that $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ α-shatters a set $(x_1, \ldots, x_T)$ if there exist $(y_1, \ldots, y_T) \in \mathbb{R}^T$ (called a *witness to shattering*) with the following property:

$$\forall (b_1, \ldots, b_T) \in \{0,1\}^T, \ \exists f \in \mathcal{F} \quad \text{s.t.}$$

$$f(x_t) > y_t + \frac{\alpha}{2} \ \text{ if } \ b_t = 1 \quad \text{and} \quad f(x_t) < y_t - \frac{\alpha}{2} \ \text{ if } \ b_t = 0$$

The *fat-shattering dimension* of $\mathcal{F}$ at scale α, denoted by $\mathrm{fat}(\mathcal{F}, \alpha)$, is the size of the largest α-shattered set.

Wait, another measure of complexity of $\mathcal{F}$? How is it related to covering numbers?

Scale-Sensitive Dimensions

We say that $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ α-shatters a set $(x_1, \ldots, x_T)$ if there exist $(y_1, \ldots, y_T) \in \mathbb{R}^T$ (called a *witness to shattering*) with the following property:

$$\forall (b_1, \ldots, b_T) \in \{0,1\}^T, \ \exists f \in \mathcal{F} \quad \text{s.t.}$$

$$f(x_t) > y_t + \frac{\alpha}{2} \ \text{ if } \ b_t = 1 \quad \text{and} \quad f(x_t) < y_t - \frac{\alpha}{2} \ \text{ if } \ b_t = 0$$

The *fat-shattering dimension* of $\mathcal{F}$ at scale α, denoted by $\text{fat}(\mathcal{F}, \alpha)$, is the size of the largest α-shattered set.

Wait, another measure of complexity of $\mathcal{F}$? How is it related to covering numbers?

Theorem (Mendelson & Vershynin): For $\mathcal{F} \subseteq [-1,1]^{\mathcal{X}}$ and any $0 < \alpha < 1$,

$$\mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) \leq \left(\frac{2}{\alpha}\right)^{K \cdot \text{fat}(\mathcal{F}, c\alpha)}$$

where K, c are positive absolute constants.

Quick Summary

We are after uniform deviations in order to understand performance of ERM. Rademacher averages is a nice measure with useful properties. They can be further upper bounded by covering numbers through the Dudley entropy integral. In turn, covering numbers can be controlled via the fat-shattering combinatorial dimension. Whew!

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Faster Rates

Are there situations when

$$\mathbb{E}\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)$$

approaches 0 faster than $O(1/\sqrt{n})$?

Yes! We can beat the Central Limit Theorem!

How is this possible??

Recall that the CLT tells us about convergence of average to the expectation for random variables with bounded second moment. What if this variance is small?

Faster Rates: Classification

Consider the problem of binary classification with the indicator loss and a class $\mathcal{F}$ of $\{0,1\}$-valued functions. For any $f \in \mathcal{F}$,

$$\frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

is an average of n Bernoulli random variables with bias $p = \mathbb{E}\ell(f(x), y)$. Exact expression for the binomial tails:

$$\mathbb{P}\left(\mathbf{L}(f) - \hat{\mathbf{L}}(f) > \epsilon\right) = \sum_{i=0}^{\lfloor n(p-\epsilon) \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$

Further upper bounds:

$\exp\left\{-\frac{n\epsilon^2}{2p(1-p)+2\epsilon/3}\right\}$	Bernstein
$\exp\left\{-2n\epsilon^2\right\}$	Hoeffding

Faster Rates: Classification

Inverting

$$\exp\left\{-\frac{n\epsilon^2}{2p(1-p)+2\epsilon/3}\right\} \le \exp\left\{-\frac{n\epsilon^2}{2p+2\epsilon/3}\right\} =: \delta$$

yields that for any $f \in \mathcal{F}$, with probability at least $1-\delta$

$$\mathbf{L}(f) \le \hat{\mathbf{L}}(f) + \sqrt{\frac{2\mathbf{L}(f)\log(1/\delta)}{n}} + \frac{2\log(1/\delta)}{3n}$$

For non-negative numbers A, B, C

$$A \le B + C\sqrt{A} \qquad \text{implies} \qquad A \le B + C^2 + \sqrt{B}C$$

Therefore for any $f \in \mathcal{F}$, with probability at least $1-\delta$,

$$\mathbf{L}(f) \le \hat{\mathbf{L}}(f) + \sqrt{\frac{2\hat{\mathbf{L}}(f)\log(1/\delta)}{n}} + \frac{4\log(1/\delta)}{n}$$

Faster Rates: Classification

By the Union Bound, for $\mathcal{F}$ with finite $N = \text{card}(F)$, with probability at least $1 - \delta$,

$$\forall f \in \mathcal{F} : \qquad \mathbf{L}(f) \leq \hat{\mathbf{L}}(f) + \sqrt{\frac{2\hat{\mathbf{L}}(f)\log(N/\delta)}{n}} + \frac{4\log(N/\delta)}{n}$$

For an empirical minimizer $\hat{f}_n$, with probability at least $1 - \delta$, a zero empirical loss $\hat{\mathbf{L}}(\hat{f}_n) = 0$ implies

$$\mathbf{L}(\hat{f}_n) \leq \frac{4\log(N/\delta)}{n}$$

This happens, for instance, in the so-called *noiseless case*: $\mathbf{L}(f_{\mathcal{F}}) = 0$. Indeed, then $\hat{\mathbf{L}}(f_{\mathcal{F}}) = 0$ and thus $\hat{\mathbf{L}}(\hat{f}_n) = 0$.

Summary: Minimax Viewpoint

Value of a game where we choose an algorithm, Nature chooses a distribution $P \in \mathcal{P}$, and our payoff is the expected loss of our algorithm relative to the best in $\mathcal{F}$:

$$\mathcal{V}^{\text{iid}}(\mathcal{F}, \mathcal{P}, n) = \inf_{\hat{f}_n} \sup_{P \in \mathcal{P}} \left\{ \mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f) \right\}$$

If we make no assumption on the distribution P, then $\mathcal{P}$ is the set of all distributions. Many of the results we obtained in this lecture are for this distribution-free case. However, one may view margin-based results and the above fast rates for the noiseless case as studying $\mathcal{V}^{\text{iid}}(\mathcal{F}, \mathcal{P}, n)$ when $\mathcal{P}$ is "nicer".

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Model Selection

For a given class $\mathcal{F}$, we have proved statements of the type

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}}\{\mathbf{L}(f) - \hat{\mathbf{L}}(f)\} \geq \phi(\delta, n, \mathcal{F})\right) < \delta$$

Now, take a countable nested sieve of models

$$\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \dots$$

such that $\mathcal{H} = \cup_{i=1}^{\infty}\mathcal{F}_i$ is a very large set that will surely capture the Bayes function.

For a function $f \in \mathcal{H}$, let $k(f)$ be the smallest index of $\mathcal{F}_k$ that contains f. Let us write $\phi_n(\delta, i)$ for $\phi(\delta, n, \mathcal{F}_i)$.

Let us put a distribution $w(i)$ on the models, with $\sum_{i=1}^{\infty} w(i) = 1$. Then for every i,

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}_i}\{\mathbf{L}(f) - \hat{\mathbf{L}}(f)\} \geq \phi_n(\delta w(i), i)\right) < \delta \cdot w(i)$$

simply by replacing δ with $\delta w(i)$.

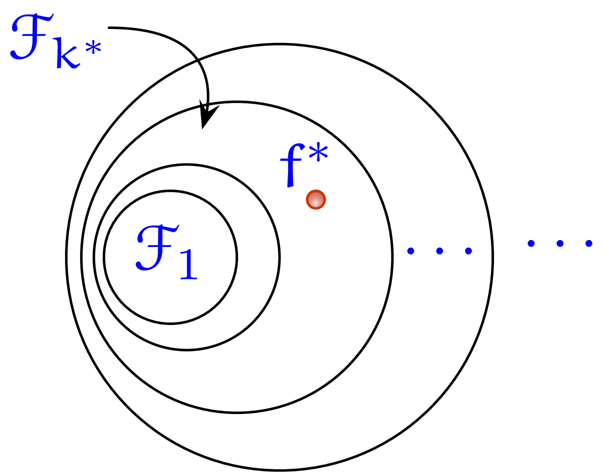
Now, taking a union bound:

$$\mathbb{P}\left(\sup_{f\in\mathcal{H}}\left\{\mathbf{L}(f)-\hat{\mathbf{L}}(f)\right\}\geq\phi_n(\delta w(k(f)),k(f))\right)<\sum_i\delta w(i)\leq\delta$$

Consider the penalized method

$$\begin{aligned}\hat{f}_n &= \arg\min_{f\in\mathcal{H}}\left\{\hat{\mathbf{L}}(f)+\phi_n(\delta w(k(f)),k(f))\right\}\\ &= \arg\min_{i,f\in\mathcal{F}_i}\left\{\hat{\mathbf{L}}(f)+\phi_n(\delta w(i),i)\right\}\end{aligned}$$

This balances fit to data and the complexity of the model. Of course, this is exactly a regularized ERM form analyzed earlier.



Let $k^*=k(f^*)$ be the (smallest) model $\mathcal{F}_i$ that contains the optimal function.

Exactly as on the slide “Countable Class: Weighted Union Bound”,

$$\begin{aligned}\mathbf{L}(\hat{f}_n) - \mathbf{L}(f^*) &\leq \left\{\mathbf{L}(\hat{f}_n) - \hat{\mathbf{L}}(\hat{f}_n) - \mathrm{pen}_n(\hat{f}_n)\right\} \\ &+ \left\{\hat{\mathbf{L}}(\hat{f}_n) + \mathrm{pen}_n(\hat{f}_n) - \hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathrm{pen}_n(f_{\mathcal{F}})\right\} \\ &+ \left\{\hat{\mathbf{L}}(f_{\mathcal{F}}) - \mathbf{L}(f_{\mathcal{F}})\right\} + \mathrm{pen}_n(f_{\mathcal{F}}) \\ &\leq \hat{\mathbf{L}}(f^*) - \mathbf{L}(f^*) + \mathrm{pen}_n(f^*) \\ &= \hat{\mathbf{L}}(f^*) - \mathbf{L}(f^*) + \phi_n(\delta w(k^*), k^*)\end{aligned}$$

The first part of this bound is $O_P(1/\sqrt{n})$ by the CLT, just as before.

If the dependence of ϕ on $1/\delta$ is logarithmic, then taking $w(i) = 2^{-i}$ simply implies an additional additive i^*, a penalty for not knowing the model in advance.

Conclusion: given uniform deviation bounds for a single class $\mathcal{F}$, as developed earlier, we can perform model selection by penalizing model complexity!

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Outline

Introduction

Statistical Learning Theory
- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning
- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Looking back: Statistical Learning

- future looks like the past
- modeled as i.i.d. data
- evaluated on a random sample from the same distribution
- developed various measures of complexity of $\mathcal{F}$

Example #1: Bit Prediction

Predict a binary sequence $y_1, y_2, \ldots \in \{0, 1\}$, which is revealed one by one. At step t, make a prediction z_t of the t-th bit, then y_t is revealed.

Let $c_t = \mathbf{I}_{\{z_t = y_t\}}$. Goal: make $\bar{c}_n = \frac{1}{n} \sum_{t=1}^{n} c_t$ large.

Suppose we are told that the sequence presented is Bernoulli with an unknown bias p. How should we choose predictions?

Example #1: Bit Prediction

Of course, we should do majority vote over the past outcomes

$$z_t = \mathbf{I}_{\{\bar{y}_{t-1} \geq 1/2\}}$$

where $\bar{y}_{t-1} = \frac{1}{t-1}\sum_{s=1}^{t-1} y_s$. This algorithm guarantees $\bar{c}_t \to \max\{p, 1-p\}$ and

$$\liminf_{t\to\infty} \left(\bar{c}_t - \max\{\bar{z}_t, 1-\bar{z}_t\}\right) \geq 0 \qquad \text{almost surely} \qquad (\star)$$

Example #1: Bit Prediction

Of course, we should do majority vote over the past outcomes

$$z_t = \mathbf{I}_{\{\bar{y}_{t-1} \geq 1/2\}}$$

where $\bar{y}_{t-1} = \frac{1}{t-1}\sum_{s=1}^{t-1} y_s$. This algorithm guarantees $\bar{c}_t \to \max\{p, 1-p\}$ and

$$\liminf_{t\to\infty} \left(\bar{c}_t - \max\{\bar{z}_t, 1-\bar{z}_t\}\right) \geq 0 \qquad \text{almost surely} \qquad (*)$$

Claim: there is an algorithm that ensures $(*)$ for an arbitrary sequence.
Any idea how to do it?

Example #1: Bit Prediction

Of course, we should do majority vote over the past outcomes

$$z_t = \mathbf{I}_{\{\bar{y}_{t-1} \geq 1/2\}}$$

where $\bar{y}_{t-1} = \frac{1}{t-1} \sum_{s=1}^{t-1} y_s$. This algorithm guarantees $\bar{c}_t \to \max\{p, 1-p\}$ and

$$\liminf_{t \to \infty} \left(\bar{c}_t - \max\{\bar{z}_t, 1 - \bar{z}_t\}\right) \geq 0 \qquad \text{almost surely} \qquad (*)$$

Claim: there is an algorithm that ensures $(*)$ for an arbitrary sequence. Any idea how to do it?

Another way to formulate $(*)$: number of mistakes should be not much more than made by the best of the two "experts", one predicting "1" all the time, the other constantly predicting "0".

Example #1: Bit Prediction

Of course, we should do majority vote over the past outcomes

$$z_t = \mathbf{I}_{\{\bar{y}_{t-1} \geq 1/2\}}$$

where $\bar{y}_{t-1} = \frac{1}{t-1}\sum_{s=1}^{t-1} y_s$. This algorithm guarantees $\bar{c}_t \to \max\{p, 1-p\}$ and

$$\liminf_{t\to\infty} \left(\bar{c}_t - \max\{\bar{z}_t, 1-\bar{z}_t\}\right) \geq 0 \qquad \text{almost surely} \quad (*)$$

Claim: there is an algorithm that ensures $(*)$ for an arbitrary sequence. Any idea how to do it?

Another way to formulate $(*)$: number of mistakes should be not much more than made by the best of the two "experts", one predicting "1" all the time, the other constantly predicting "0".

Note the difference: estimating a hypothesized model vs competing against a reference set. We had seen this distinction in the previous lecture.

Example #2: Email Spam Detection

We are tasked with developing a spam detection program that needs to be adaptive to malicious attacks.

- $x_1, \dots, x_n$ are email messages, revealed one-by-one
- upon observing the message x_t, the learner (spam detector) needs to decide whether it is spam or not spam ($\hat{y}_t \in \{0,1\}$)
- the actual label $y_t \in \{0,1\}$ is revealed (e.g. by the user)

Do it seem plausible that $(x_1, y_1), \dots, (x_n, y_n)$ are i.i.d. from some distribution P?

Probably not... In fact, the sequence might even be *adversarially* chosen. In fact, spammers *adapt* and try to improve their strategies.

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Online Learning (Supervised)

- No assumption that there is a single distribution P
- Data not given all at once, but rather in the online fashion
- As before, $\mathcal{X}$ is the space of inputs, $\mathcal{Y}$ the space of outputs
- Loss function $\ell(y_1, y_2)$

Online protocol (*supervised learning*):

```
For $t = 1, \ldots, n$
    Observe $x_t$, predict $\hat{y}_t$, observe $y_t$
```

Goal: keep **regret** small:

$$\mathrm{Reg}_n = \frac{1}{n}\sum_{t=1}^{n} \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \ell(f(x_t), y_t)$$

A bound on Reg_n should hold for any sequence $(x_1, y_1), \ldots, (x_n, y_n)$!

Pros/Cons of Online Learning

The good:

- An upper bound on regret implies good performance relative to the set $\mathcal{F}$ *no matter how adversarial the sequence is.*
- Online methods are typically computationally attractive as they process one data point at a time. Used when data sets are huge.
- Interesting research connections to Game Theory, Information Theory, Statistics, Computer Science.

The bad:

- A regret bound implies good performance only if one of the elements of $\mathcal{F}$ has good performance (just as in Statistical Learning). However, for non-iid sequences a single $\mathsf{f} \in \mathcal{F}$ might not be good at all! To alleviate this problem, the comparator set $\mathcal{F}$ can be made into a set of more complex strategies.
- There might be some (non-i.i.d.) structure of sequences that we are not exploiting (this is an interesting area of research!)

Setting Up the Minimax Value

First, it turns out that $\hat{y}_t$ has to be a *randomized* prediction: we need to decide on a distribution $q_t \in \Delta(\mathcal{Y})$ and then draw $\hat{y}_t$ from q_t.

The minimax best that both the learner and the adversary (or, Nature) can do is

$$\mathcal{V}(\mathcal{F}, n) = \left\langle\!\!\left\langle \sup_{x_t \in \mathcal{X}} \inf_{q_t \in \Delta} \sup_{y_t \in \mathcal{Y}} \mathop{\mathbb{E}}_{y_t \sim q_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \left\{ \frac{1}{n} \sum_{t=1}^{n} \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} \ell(f(x_t), y_t) \right\}$$

This is an awkward and long expression, so no need to be worried. All you need to know right now is:

- An upper bound on $\mathcal{V}(\mathcal{F}, n)$ guarantees existence of a strategy (learning algorithm) that will suffer at most that much regret.
- A lower bound on $\mathcal{V}(\mathcal{F}, n)$ means the adversary can inflict at least that much damage, no matter what the learning algorithm does.

It is interesting to study $\mathcal{V}(\mathcal{F}, n)$! It turns out, many of the tools we used in Statistical Learning can be extended to study Online Learning!

Sequential Rademacher Complexity

A (complete binary) $\mathcal{X}$-valued tree $\mathbf{x}$ of depth n is a collection of functions $\mathbf{x}_1, \ldots, \mathbf{x}_n$ such that $\mathbf{x}_i : \{\pm 1\}^{i-1} \mapsto \mathcal{X}$ and $\mathbf{x}_1$ is a constant function.

A sequence $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ defines a path in $\mathbf{x}$:

$$\mathbf{x}_1, \ \mathbf{x}_2(\epsilon_1), \ \mathbf{x}_3(\epsilon_1, \epsilon_2), \ldots, \ \mathbf{x}_n(\epsilon_1, \ldots, \epsilon_{n-1})$$

Define *sequential Rademacher complexity* as

$$\mathscr{R}_n^{\mathrm{seq}}(\mathcal{F}, n) = \sup_{\mathbf{x}} \mathbb{E}_{\epsilon_{1:n}} \sup_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1})) \right\}$$

where the supremum is over all $\mathcal{X}$-valued trees of depth n.

Theorem

Let $\mathcal{Y} = \{0, 1\}$ and $\mathcal{F}$ is a class of binary-valued functions. Let ℓ be the indicator loss. Then

$$\mathcal{V}(\mathcal{F}, n) \leq 2\mathscr{R}_n^{seq}(\mathcal{F}, n)$$

Finite Class

Suppose $\mathcal{F}$ is finite, $N = \text{card}(\mathcal{F})$. Then for any tree $\mathbf{x}$,

$$\mathbb{E}_{\epsilon_{1:n}} \sup_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{t=1}^{n} \epsilon_t f(\mathbf{x}_t(\epsilon_{1:t-1})) \right\} \leq \sqrt{\frac{2 \log N}{n}}$$

because, again, this is a maximum of N (sub)Gaussian Random variables!

Hence,

$$\mathcal{V}(\mathcal{F}, n) \leq 2\sqrt{\frac{2 \log N}{n}}$$

This bound is basically the same as that for Statistical Learning with a finite number of functions!

Therefore, there must exist an algorithm for predicting $\hat{y}_t$ given x_t such that regret scales as $O\left(\sqrt{\frac{\log N}{n}}\right)$. What is it?

Exponential Weights, or the Experts Algorithm

We think of each element $\{f_1, \dots, f_N\} = \mathcal{F}$ as an expert who gives a prediction $f_i(x_t)$ given side information x_t. We keep distribution w_t over experts, according to their performance.

Let $w_1 = (1/N, \dots, 1/N)$, $\eta = \sqrt{(8 \log N)/T}$.

To predict at round t, observe x_t, pick $i_t \sim w_t$ and set $\hat{y}_t = f_{i_t}(x_t)$.

Update

$$w_{t+1}(i) \propto w_t(i) \exp\left\{-\eta \mathbf{I}_{\{f_i(x_t) \neq y_t\}}\right\}$$

Claim: for any sequence $(x_1, y_1), \dots, (x_n, y_n)$, with probability at least $1 - \delta$

$$\frac{1}{n} \sum_{t=1}^{n} \mathbf{I}_{\{\hat{y}_t \neq y_t\}} - \inf_{f \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} \mathbf{I}_{\{f(x_t) \neq y_t\}} \leq \sqrt{\frac{\log N}{2n}} + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Useful Properties of Sequential Rademacher Complexity

Sequential Rademacher complexity enjoys the same nice properties as its iid cousin, except for the Lipschitz contraction (4). At the moment we can only prove

$$\mathscr{R}_n^{\text{seq}}(\phi \circ \mathcal{F}) \leq L\mathscr{R}_n^{\text{seq}}(\mathcal{F}) \times O(\log^{3/2} n)$$

It is an open question whether this logarithmic factor can be removed...

Theory for Online Learning

There is now a theory with combinatorial parameters, covering numbers, and even a recipe for developing online algorithms!

Many of the relevant concepts (e.g. sequential Rademacher complexity) are generalizations of the i.i.d. analogues to the case of dependent data.

Coupled with the online-to-batch conversion we introduce in a few slides, there is now an interesting possibility of developing new computationally attractive algorithms for statistical learning. One such example will be presented.

Theory for Online Learning

Statistical Learning	Online Learning
i.i.d. data	arbitrary sequences
tuples of data	binary trees
Rademacher averages	sequential Rademacher complexity
covering / packing numbers	tree cover
Dudley entropy integral	analogous result with tree cover
VC dimension	Littlestone's dimension
Scale-sensitive dimension	analogue for trees
Vapnik-Chervonenkis-Sauer-Shelah Lemma	analogous combinatorial result for trees
ERM and regularized ERM	many interesting algorithms

Outline

Introduction

Statistical Learning Theory

- The Setting of SLT
- Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
- Tools from Probability, Empirical Processes
- From Finite to Infinite Classes
- Uniform Convergence, Symmetrization, and Rademacher Complexity
- Large Margin Theory for Classification
- Properties of Rademacher Complexity
- Covering Numbers and Scale-Sensitive Dimensions
- Faster Rates
- Model Selection

Sequential Prediction / Online Learning

- Motivation
- Supervised Learning
- Online Convex and Linear Optimization
- Online-to-Batch Conversion, SVM optimization

Online Convex and Linear Optimization

For many problems, $\ell(f,(x,y))$ is convex in f and $\mathcal{F}$ is a convex set. Let us simply write $\ell(f,z)$, where the move z need not be of the form (x,y).

- ▷ e.g. square loss $\ell(f,(x,y)) = (\langle f,x\rangle - y)^2$ for linear regression.
- ▷ e.g. hinge loss $\ell(f,(x,y)) = \max\{0, 1 - y\langle f,x\rangle\}$, a surrogate loss for classification.

We may then use optimization algorithms for updating our hypothesis after seeing each additional data point.

Online Convex and Linear Optimization

Online protocol (*Online Convex Optimization*):

```
For $t = 1, \ldots, n$
    Predict $f_t \in \mathcal{F}$, observe $z_t$
```

Goal: keep **regret** small:

$$\mathrm{Reg}_n = \frac{1}{n}\sum_{t=1}^{n} \ell(f_t, z_t) - \inf_{f \in \mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \ell(f, z_t)$$

Online Linear Optimization is a particular case when $\ell(f, z) = \langle f, z \rangle$.

Gradient Descent

At time $t = 1, \ldots, n$, predict $f_t \in \mathcal{F}$, observe z_t, update

$$f'_{t+1} = f_t - \eta \nabla \ell(f_t, z_t)$$

and project f'_{t+1} to the set $\mathcal{F}$, yielding f_{t+1}.

- η is a learning rate (step size)
- gradient is with respect to the first coordinate

This simple algorithm guarantees that for any $f \in \mathcal{F}$

$$\frac{1}{n}\sum_{t=1}^{n} \ell(f_t, z_t) - \frac{1}{n}\sum_{t=1}^{n} \ell(f, z_t) \le \frac{1}{n}\sum_{t=1}^{n} \langle f_t, \nabla \ell(f_t, z_t)\rangle - \frac{1}{n}\sum_{t=1}^{n} \langle f, \nabla \ell(f_t, z_t)\rangle$$
$$\le O(n^{-1/2})$$

as long as $\|\nabla \ell(f_t, z_t)\| \le c$ for some constant c, for all t, and $\mathcal{F}$ has a bounded diameter.

Gradient Descent for Strongly Convex Functions

Assume that for any z, $\ell(\cdot, z)$ is strongly convex in the first argument. That is, $\ell(f, z) - \frac{1}{2}\|f\|^2$ is a convex function.

The same gradient descent algorithm with a different step size η guarantees that for any $f^* \in \mathcal{F}$

$$\frac{1}{n}\sum_{t=1}^{n}\ell(f_t, z_t) - \frac{1}{n}\sum_{t=1}^{n}\ell(f, z_t) \leq O\left(\frac{\log(n)}{n}\right),$$

a faster rate.

Outline

- Introduction
- Statistical Learning Theory
  - The Setting of SLT
  - Consistency, No Free Lunch Theorems, Bias-Variance Tradeoff
  - Tools from Probability, Empirical Processes
  - From Finite to Infinite Classes
  - Uniform Convergence, Symmetrization, and Rademacher Complexity
  - Large Margin Theory for Classification
  - Properties of Rademacher Complexity
  - Covering Numbers and Scale-Sensitive Dimensions
  - Faster Rates
  - Model Selection
- Sequential Prediction / Online Learning
  - Motivation
  - Supervised Learning
  - Online Convex and Linear Optimization
  - Online-to-Batch Conversion, SVM optimization

How to use regret bounds for i.i.d. data

Suppose we have a regret bound

$$\frac{1}{n}\sum_{t=1}^{n}\ell(f_t, z_t) - \inf_{f\in\mathcal{F}}\frac{1}{n}\sum_{t=1}^{n}\ell(f, z_t) \le R_n$$

that holds for all sequences $z_1, \ldots, z_n$, for some $R_n \to 0$.

Assume $z_1, \ldots, z_n$ are i.i.d. with distribution P. Run the regret minimization algorithm on these data and let $\bar{f} = \frac{1}{n}\sum_{t=1}^{n} f_t$. Then

$$\mathbb{E}_{z,z_1,\ldots,z_n}\ell(\bar{f}, z) \le \mathbb{E}\left\{\frac{1}{n}\sum_{t=1}^{n}\ell(f_t, z)\right\} = \mathbb{E}\left\{\frac{1}{n}\sum_{t=1}^{n}\ell(f_t, z_t)\right\}$$

where the last step holds because f_t only depends on $z_1, \ldots, z_{t-1}$. Also,

$$\mathbb{E}\left\{\inf_{f\in\mathcal{F}}\frac{1}{n}\sum_{t=1}^{n}\ell(f, z_t)\right\} \le \inf_{f\in\mathcal{F}}\mathbb{E}\left\{\frac{1}{n}\sum_{t=1}^{n}\ell(f, z_t)\right\} = \mathbb{E}_z\ell(f_{\mathcal{F}}, z)$$

Combining,

$$\mathbb{E}\mathbf{L}(\bar{f}) - \inf_{f\in\mathcal{F}}\mathbf{L}(f) \le R_n$$

How to use regret bounds for i.i.d. data

This gives an alternative way of proving bounds on

$$\mathbb{E}\mathbf{L}(\hat{f}_n) - \inf_{f \in \mathcal{F}} \mathbf{L}(f)$$

by using $\hat{f}_n = \bar{f}$, the average of the trajectory of an online learning algorithm.

Next, we present an interesting application of this idea.

Pegasos

Support Vector Machine is a fancy name for the algorithm

$$\hat{f}_n = \arg\min_{f \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y_i \langle f, x_i \rangle\} + \frac{\lambda}{2} \|f\|^2$$

in the linear case.

The objective can be "kernelized" for representing linear separators in higher-dimensional feature space. The hinge loss is convex in f.

Write

$$\ell(f, z) = \max\{0, 1 - y \langle f, x \rangle\} + \frac{\lambda}{2} \|f\|^2$$

for $z = (x, y)$. Then the objective of SVM can be written as

$$\min_f \mathbb{E}\ell(f, z)$$

The expectation is with respect to the *empirical distribution* $\frac{1}{m} \sum_{i=1}^{m} \delta_{(x_i, y_i)}$.

Then an i.i.d. sample $z_1, \ldots, z_n$ from the empirical distribution is simply a draw with replacement from the dataset $\{(x_1, y_1), \ldots, (x_m, y_m)\}$.

Pegasos

A gradient descent $f_{t+1} = f_t - \eta \nabla \ell(f_t, z_t)$ with

$$\nabla \ell(f_t, z_t) = -y_t x_t \mathbf{I}_{\{y_t \langle f_t, x_t \rangle < 1\}} + \lambda f_t$$

then gives a guarantee

$$\mathbb{E}\ell(\bar{f}, z) - \inf_{f \in \mathcal{F}} \mathbb{E}\ell(f, z) \leq R_n$$

Since $\ell(f, z)$ is λ-strongly convex, the rate $R_n = O(\log(n)/n)$.

```
Pegasos (Shalev-Shwartz et al, 2010)
For $t = 1, \ldots, n$
  Choose a random example $(x_{i_t}, y_{i_t})$ from the dataset. Set $\eta = 1/(\lambda t)$
  If $y_{i_t} \langle f_t, x_{i_t} \rangle < 1$, update $f_{t+1} = (1 - \eta_t \lambda) f_t + \eta_t x_{i_t} y_{i_t}$
  else, update $f_{t+1} = (1 - \eta_t \lambda) f_t$
```

The algorithm and analysis are due to (S. Shalev-Shwartz, Singer, Srebro, Cotter, 2010)

Pegasos

We conclude that $\bar{f} = \frac{1}{n} \sum_{t=1}^{n} f_t$ computed using the gradient descent algorithm is an $\tilde{O}(n^{-1})$-approximate minimizer of the SVM objective after n steps.

This gives an $O(d/(\lambda\epsilon))$ time to converge to an ϵ-minimizer. Very fast SVM solver, attractive for large datasets!

Summary

Key points for both statistical and online learning:

- obtained performance guarantees with minimal assumptions
- prior knowledge is captured by the comparator term
- understanding the inherent complexity of the comparator set
- key techniques: empirical processes for iid and non-iid data
- interesting relationships between statistical and online learning
- computation and statistics – a basis of machine learning

From Classical Statistics to Modern ML: the Lessons of Deep Learning

Mikhail Belkin
Ohio State University,
Department of Computer Science and Engineering,
Department of Statistics

IAS Workshop on Theory of Deep Learning:
Where next?

Empirical Risk Minimization

Most theoretical analyses for ML are based on ERM

Empirical risk

$$f_{ERM}^{*} = \underset{f\in\mathcal{H}}{argmin}\ \frac{1}{n}\sum_{training\ data} L(f(x_i), y_i)$$

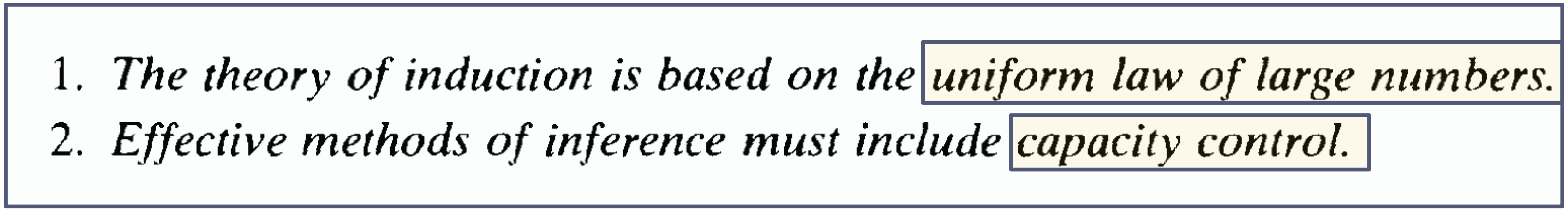
Minimize empirical risk over a class of functions $\mathcal{H}$.

- …

The ERM/SRM theory of learning

Goal of ML: $f^* = \underset{f}{\arg\min}\, E_{unseen\ data}\, L(f(x), y)$

Goal of ERM $f^*_{ERM} = \arg \min_{f_w \in \mathcal{H}} \frac{1}{n} \sum_{training\ data} L(f_w(x_i), y_i)$



1. *The theory of induction is based on the uniform law of large numbers.*
2. *Effective methods of inference must include capacity control.*

V. Vapnik, Statistical Learning Theory, 1998

1. Empirical loss of any $f \in \mathcal{H}$ approximates expected loss of f.
2. $\mathcal{H}$ contains functions that approximate f^*.

$$(1)+(2) \implies E_{unseen\ data}\, L\big(f^*_{ERM}(x), y\big) \approx E_{unseen\ data}\, L(f^*(x), y)$$

…

Uniform laws of large numbers

WYSIWYG bounds VC-dim, fat shattering, Rademacher, covering numbers, margin…

Model or function complexity, e.g., VC, margin or $\|f\|_{\mathcal{H}}$

Expected risk: what you get

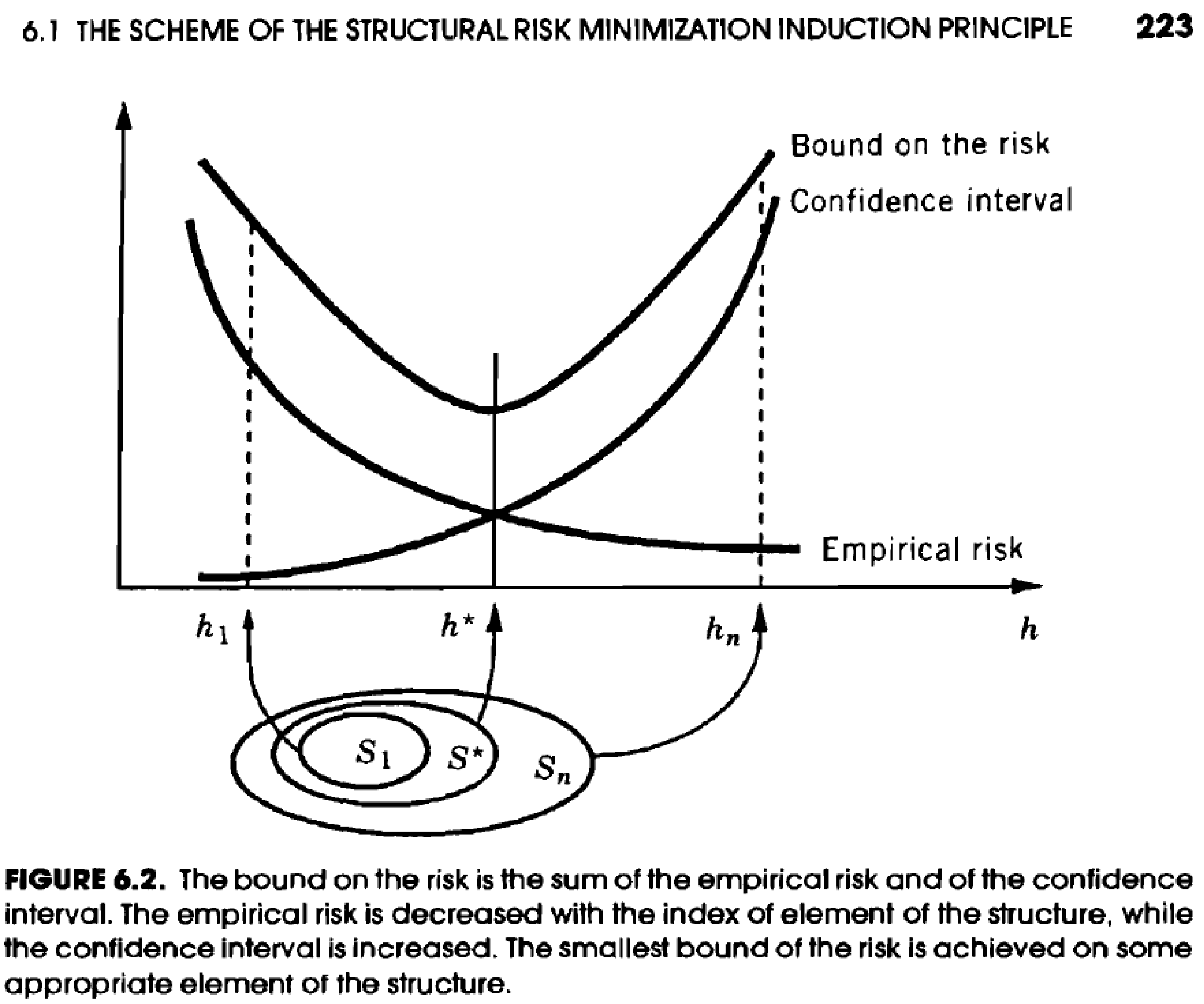
Empirical risk: what you see

$$E(L(f^*_{ERM}, y)) \leq \frac{1}{n}\sum L\,(f^*_{ERM}(x_i), y_i)\ + O^*\left(\sqrt{\frac{c}{n}}\right)$$

Margin and other "a posteriori" bounds allow $\mathcal{H}$ to be data-dependent.

Capacity control

…

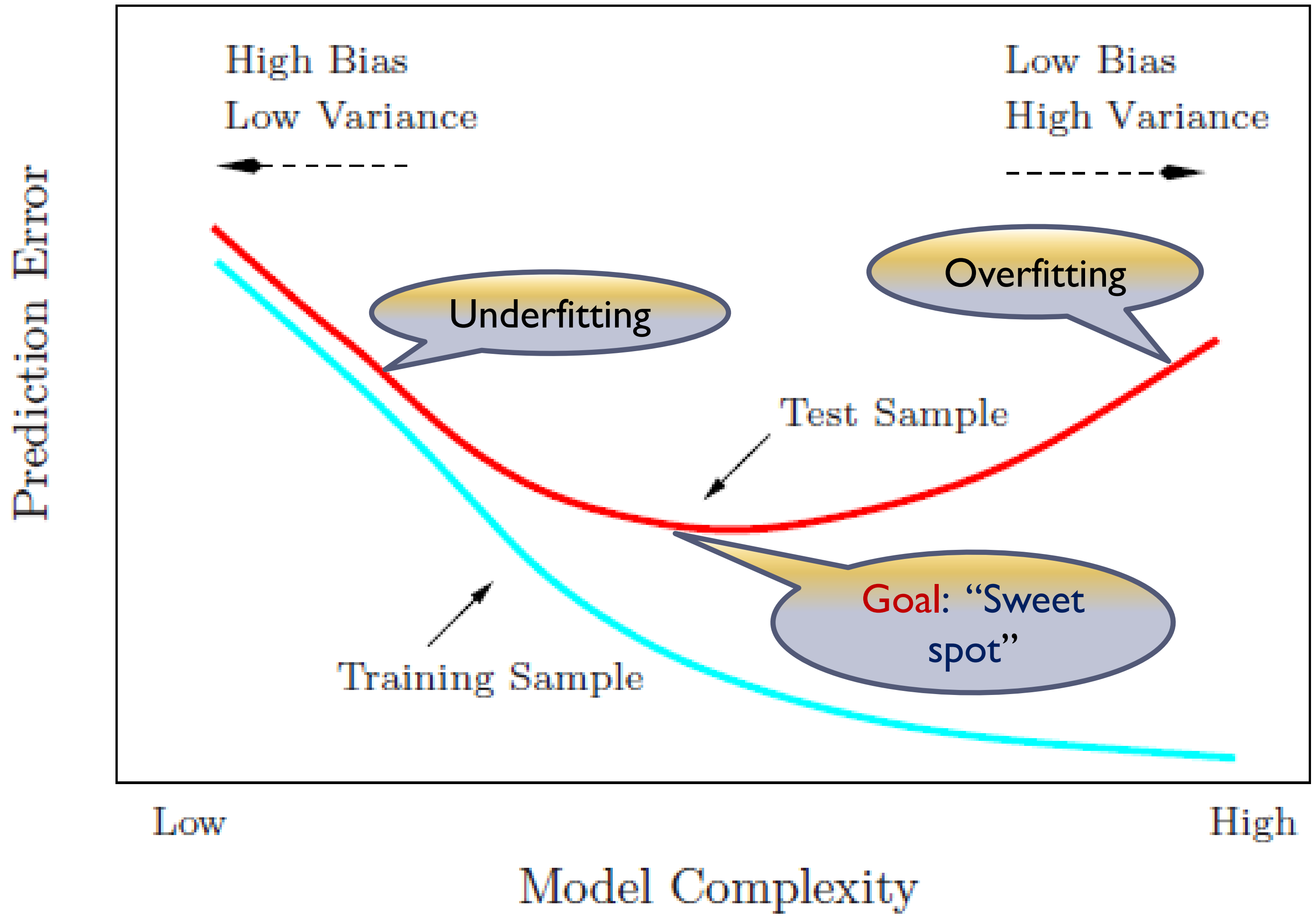


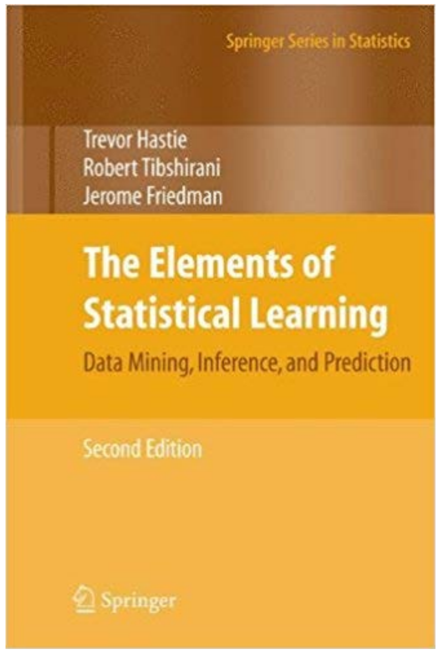
6.1 THE SCHEME OF THE STRUCTURAL RISK MINIMIZATION INDUCTION PRINCIPLE 223

FIGURE 6.2. The bound on the risk is the sum of the empirical risk and of the confidence interval. The empirical risk is decreased with the index of element of the structure, while the confidence interval is increased. The smallest bound of the risk is achieved on some appropriate element of the structure.

V.Vapnik, **Statistical Learning Theory**, 1998

U-shaped generalization curve





Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

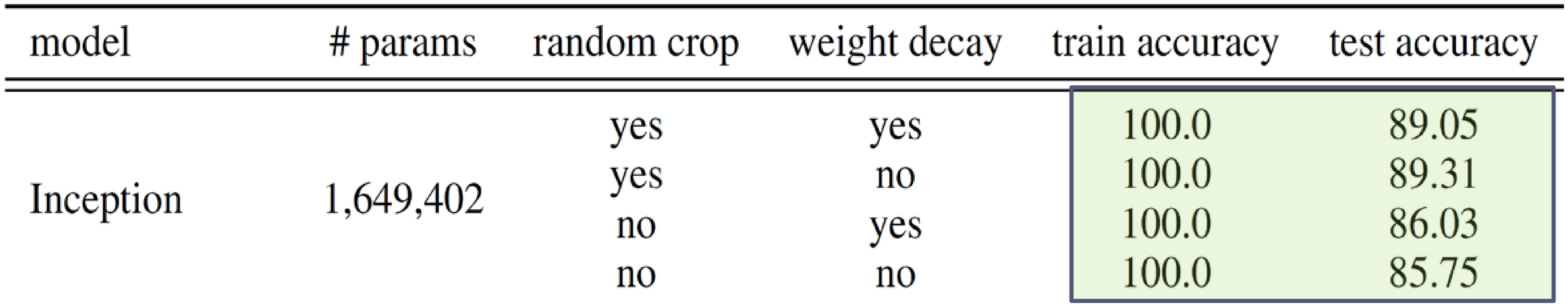
Springer

However, a model with zero training error is overfit to the training data and will typically generalize poorly.

Interpolation

Page 194

Does interpolation overfit?



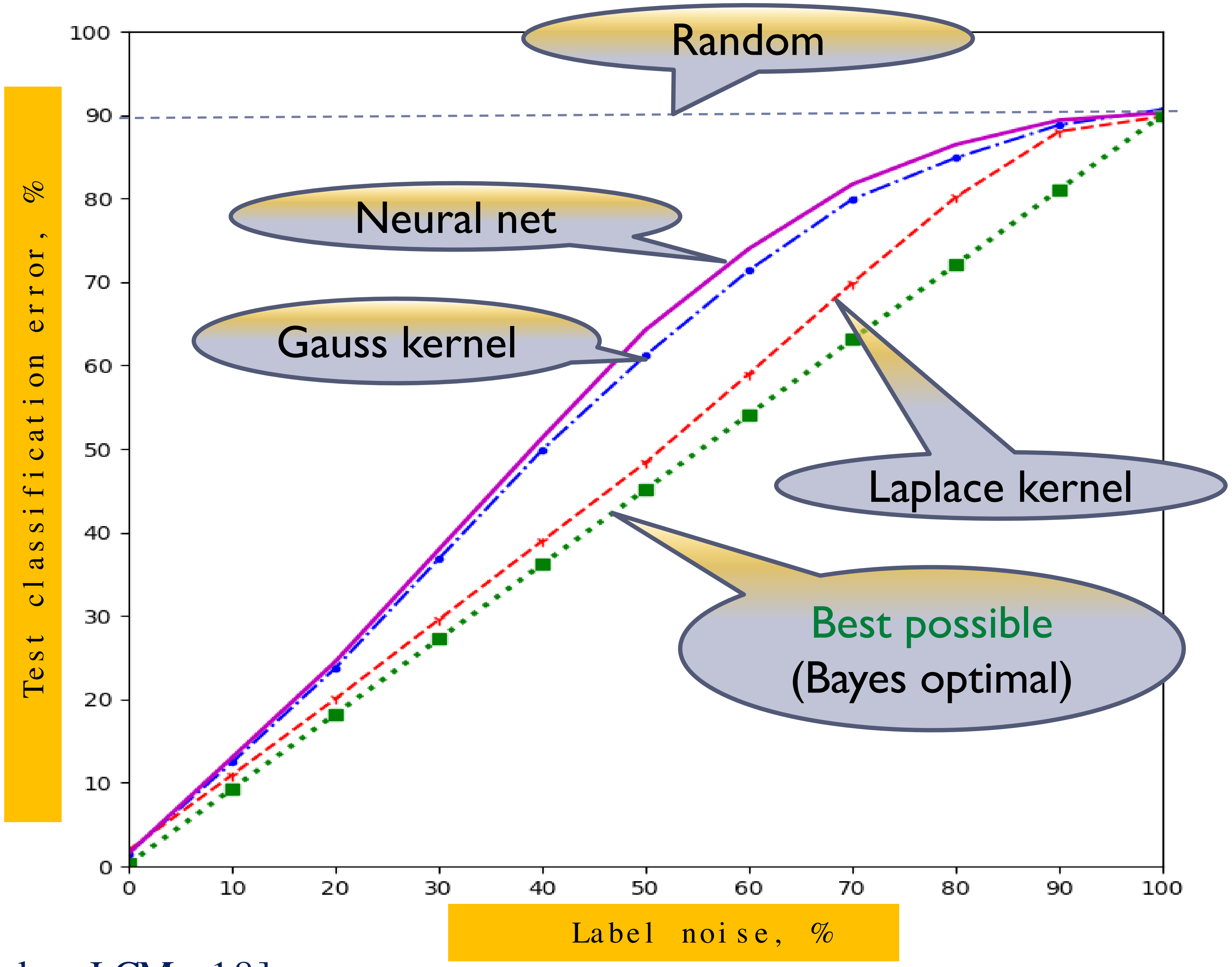
model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75

[CIFAR 10, from *Understanding deep learning requires rethinking generalization*, Zhang, et al, 2017]

Suggestive, but does not on its own invalidate the ERM theory/uniform bounds.

Interpolation does not overfit even for very noisy data

All methods (except Bayes optimal) have zero training square loss.



[B., Ma, Mandal, ICML 18]

Uniform bounds:

VC-dim, fat shattering, Rademacher, covering numbers, PAC-Bayes, margin…

Model or function complexity, e.g., VC or $\|f\|_{\mathcal{H}}$

Test loss   Training loss

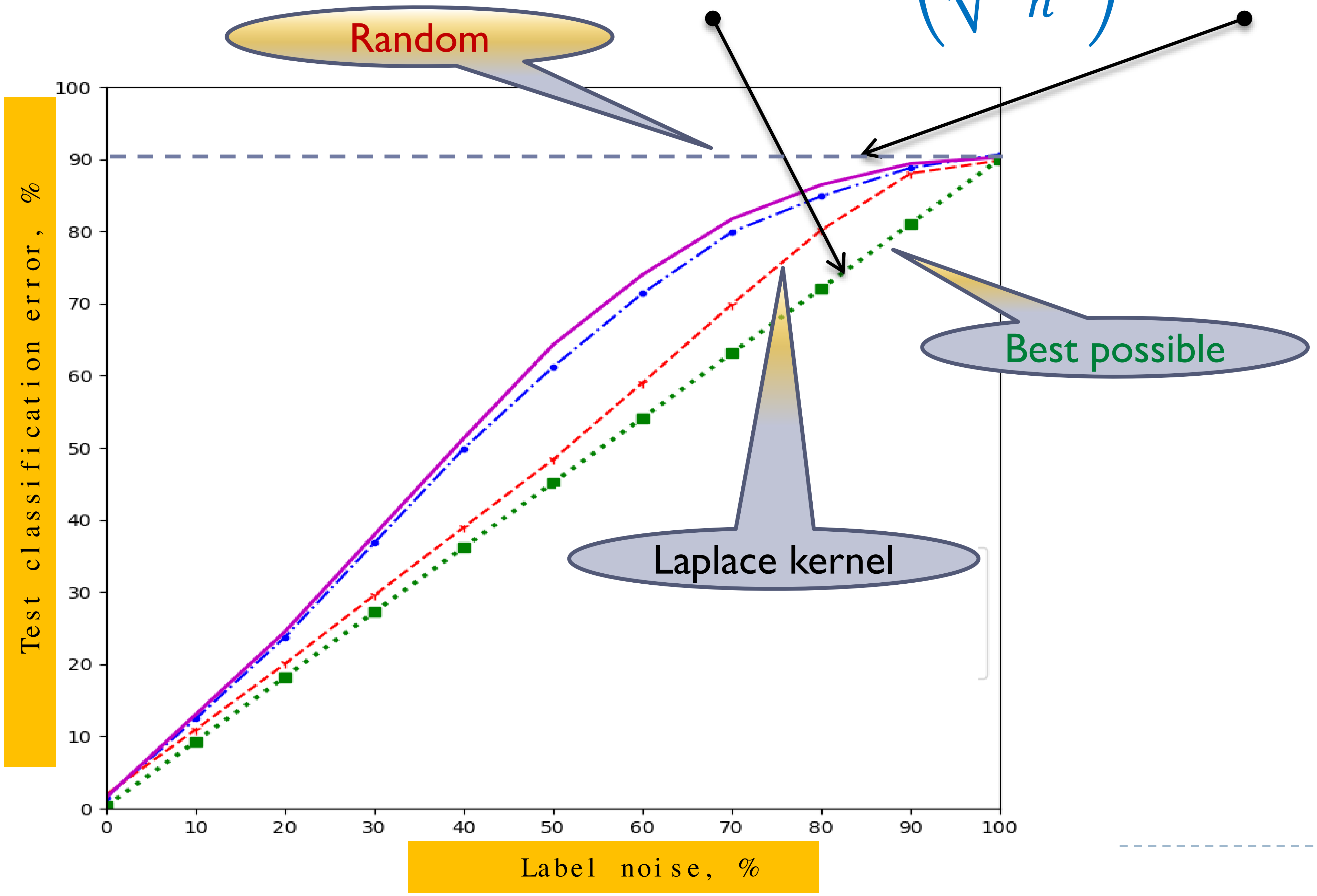
$$E(L(f^*,y)) \le \underbrace{\cancel{\frac{1}{n}\sum L(f^*(x_i),y_i)}}_{=0} + O^*\left(\sqrt{\frac{c}{n}}\right)$$

Can uniform bounds account for generalization under interpolation?

Bounds?

What kind of generalization bound could work here?

$$0.7 < O^*\left(\sqrt{\frac{c(n)}{n}}\right) < 0.9$$



Why bounds fail

$$\overset{\text{correct}}{0.7 <} O^*\left(\sqrt{\frac{c(n)}{n}}\right) \overset{\text{nontrivial}}{< 0.9} \qquad n \to \infty$$

1. The constant in O^* needs to be exact. There are no known bounds like that.

2. Conceptually, how would the quantity $c(n)$ “know” about the Bayes risk?

Interpolation is best practice for deep learning

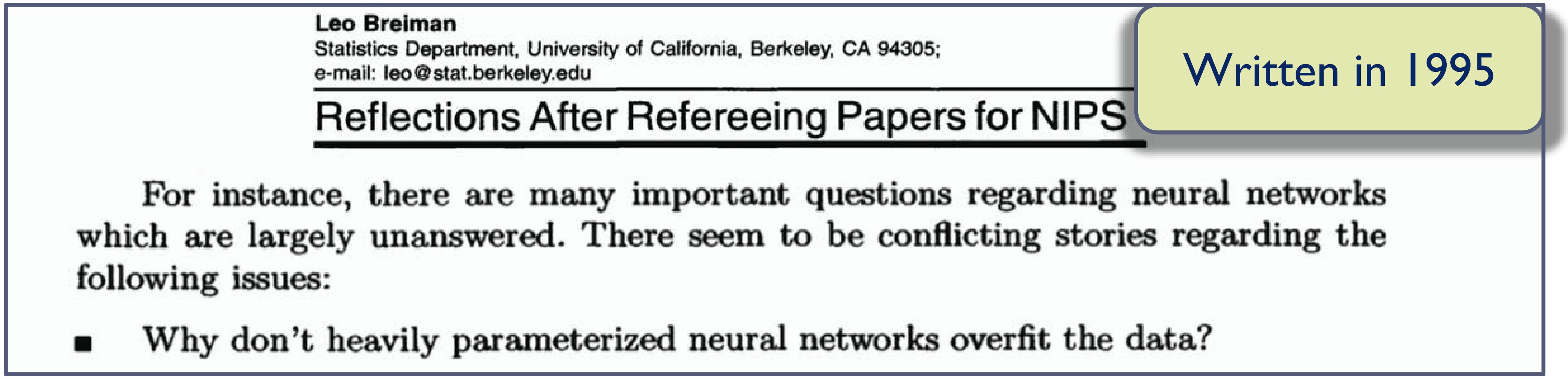
From Ruslan Salakhutdinov's tutorial (Simons Institute, 2017):

The best way to solve the problem from practical standpoint is you build a very big system …basically you want to make sure you hit the zero training error.

Historical recognition

Yann Lecun (IPAM talk, 2018):

Deep learning breaks some basic rules of statistics.



Leo Breiman
Statistics Department, University of California, Berkeley, CA 94305;
e-mail: leo@stat.berkeley.edu

Reflections After Refereeing Papers for NIPS

For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

- Why don't heavily parameterized neural networks overfit the data?

Written in 1995

Where we are now: the key lesson

The new theory of induction cannot be based on uniform laws of large numbers with capacity control.

Where next?

Generalization theory for interpolation?

What theoretical analyses do we have?

- VC-dimension/Rademacher complexity/covering/Pac-Bayes/margin bounds.
  - Cannot deal with interpolated classifiers when Bayes risk is non-zero.
  - Generalization gap cannot be bound when empirical risk is zero.
- Algorithmic stability.
  - Does not apply when empirical risk is zero, expected risk nonzero.

Uniform bounds:

training loss $= 0$

$\not\approx$

expected loss

- Regularization-type analyses (Tikhonov, early stopping/SGD, etc.)
  - Diverge as $\lambda \to 0$ for fixed n.

Typically Diverge

- Classical smoothing methods (nearest neighbors, Nadaraya–Watson).
  - Most classical analyses do not support interpolation.
  - But 1-NN! (Also Hilbert regression Scheme, [Devroye, et al. 98])

Oracle bounds

expected loss

$\approx$

optimal loss

A way forward?

1-nearest neighbor classifier is very suggestive.

Interpolating classifier with a non-trivial (sharp!) performance guarantee.

Twice the Bayes risk [Cover, Hart, 67].

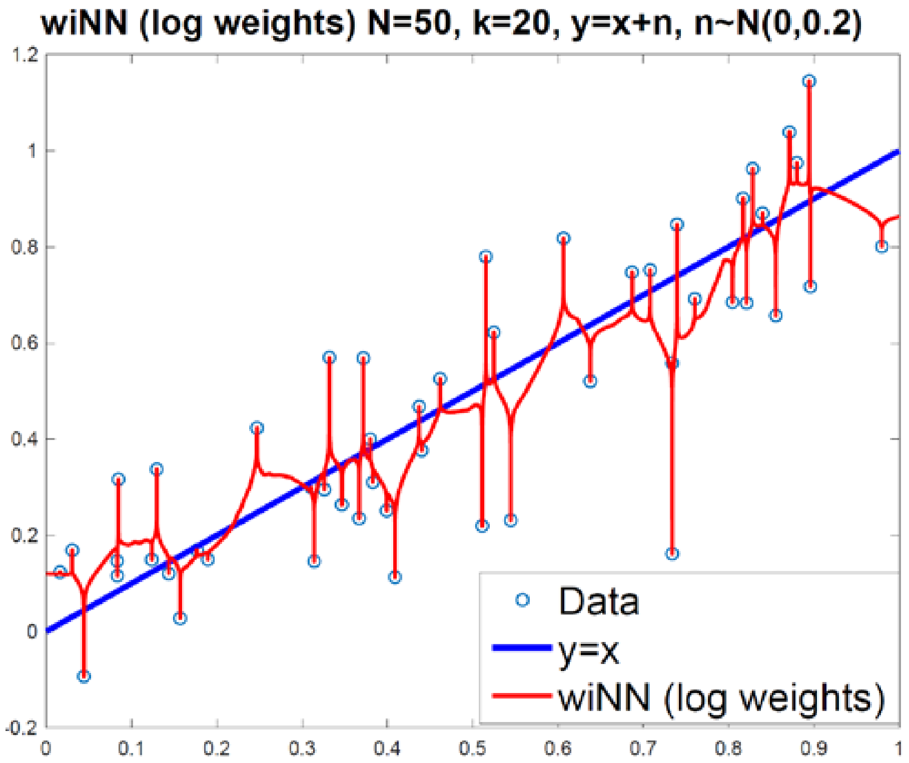
- Analysis not based on complexity bounds.
- Estimating expected loss, not the generalization gap.

Interpolated k-NN schemes

$$f(x) = \frac{\sum y_i k(x_i,x)}{\sum k(x_i,x)}$$

$$k(x_i,x) = \frac{1}{||x-x_i||^\alpha},\ k(x_i,x) = -\log||x-x_i||$$

(cf. Shepard's interpolation)



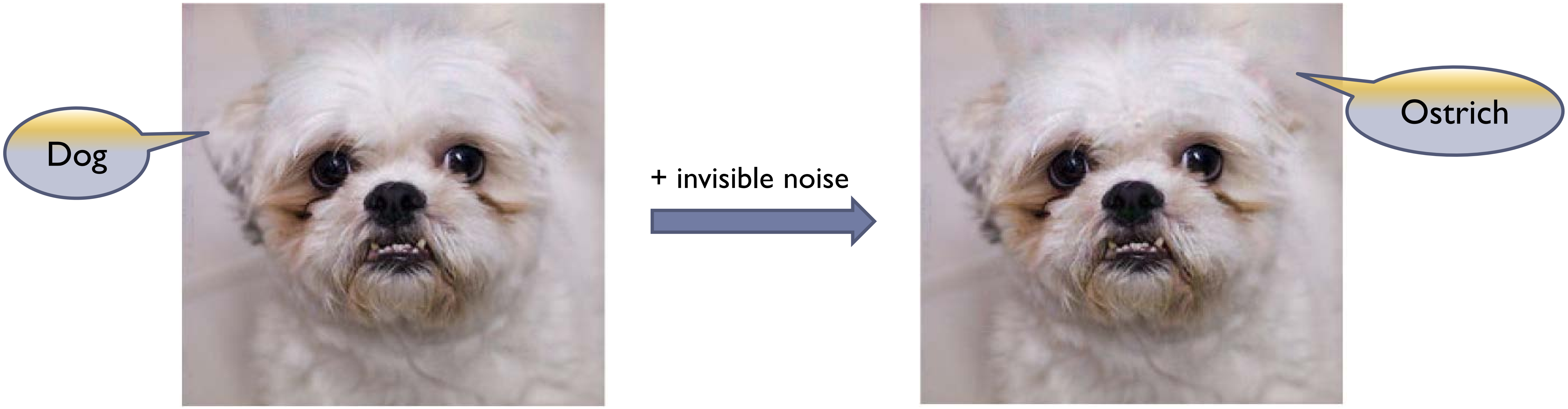
Theorem:

Weighted (interpolated) k-nn schemes with certain singular kernels are consistent (converge to Bayes optimal) for classification in any dimension.

Moreover, statistically (minimax) optimal for regression in any dimension.

[B., Hsu, Mitra, NeuriPS 18] [B., Rakhlin, Tsybakov, AIStats 19]

Interpolation and adversarial examples



From Szegedy, at al, ICLR 2014

Theorem: adversarial examples for interpolated classifiers are asymptotically dense (assuming the labels are not deterministic).

[B., Hsu, Mitra, NeurIPS 18]

This talk so far:

A. Effectiveness of interpolation.

B. Theory of interpolation cannot be based on uniform bounds.

C. Statistical validity of interpolating nearest neighbor methods.

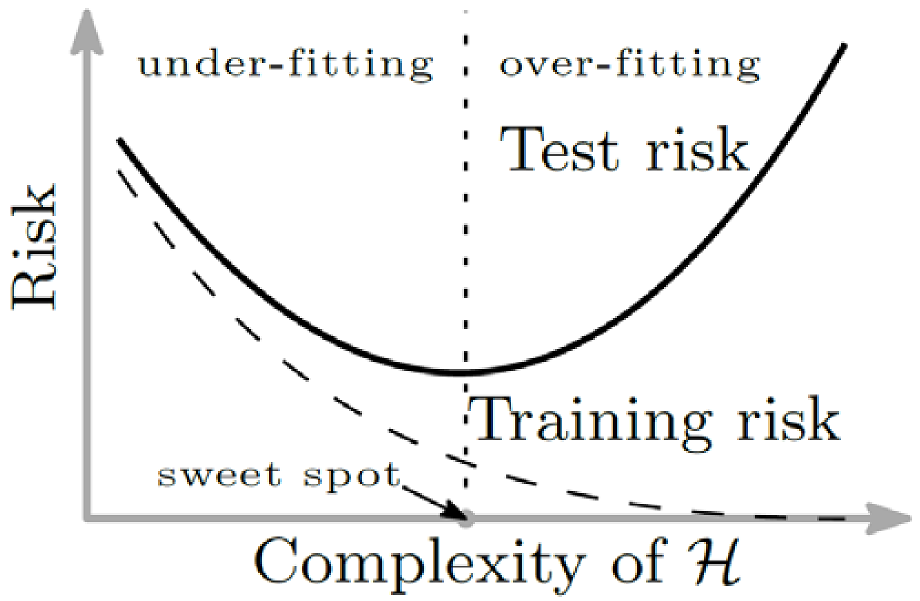
Yet, there is a mismatch between A and C. Methods we considered theoretically seem quite different from those used in practice.

Key questions:

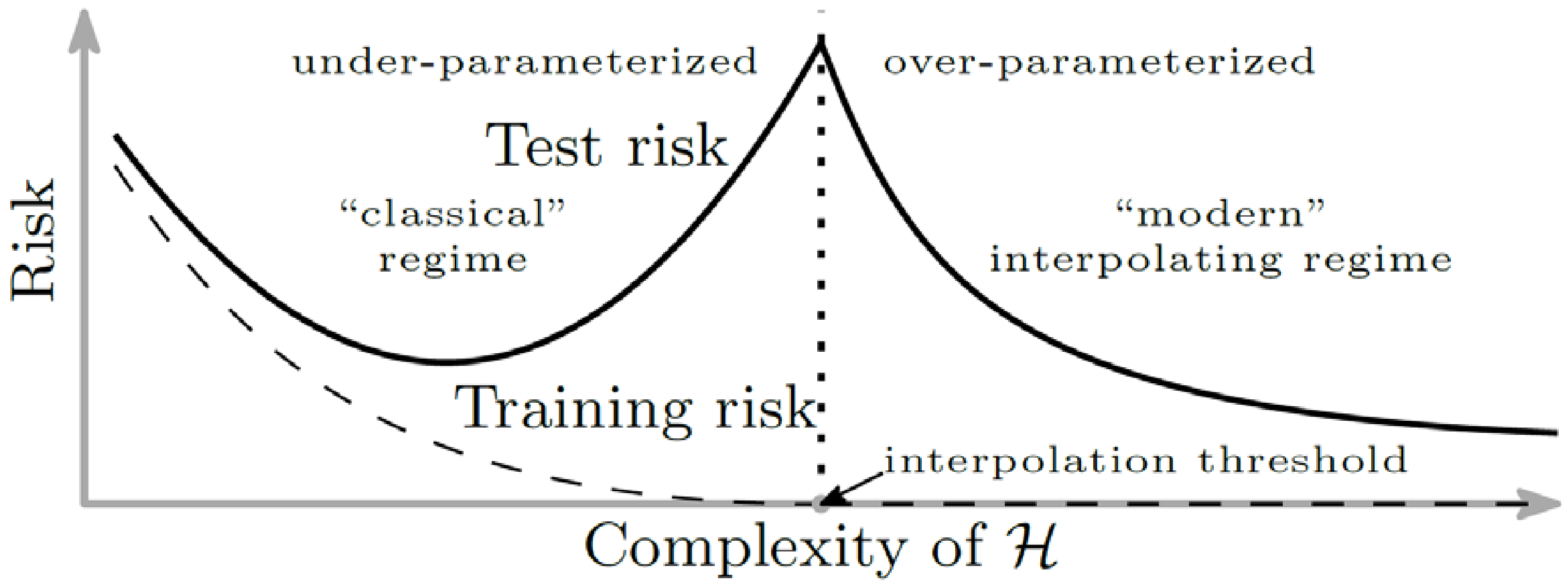
- How do classical analyses relate to interpolation?
- Dependence of generalization on model complexity?
- What is the role of optimization?

“Double descent” risk curve

Classical risk curve

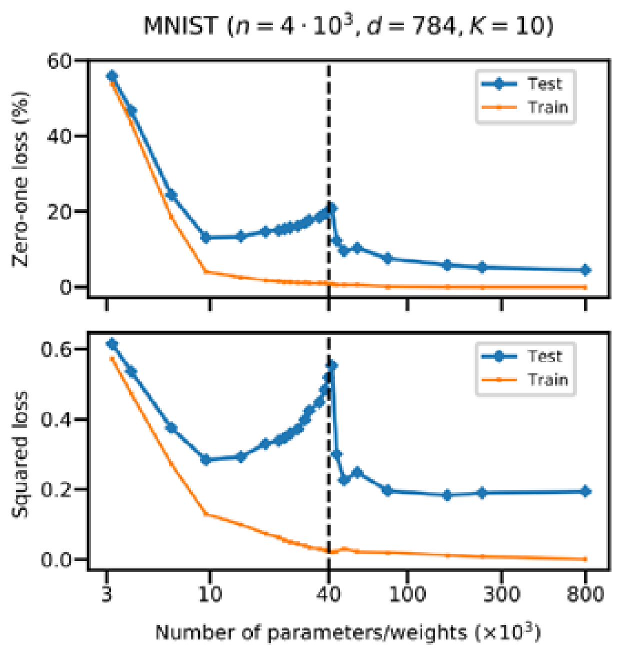


New “double descent” risk curve

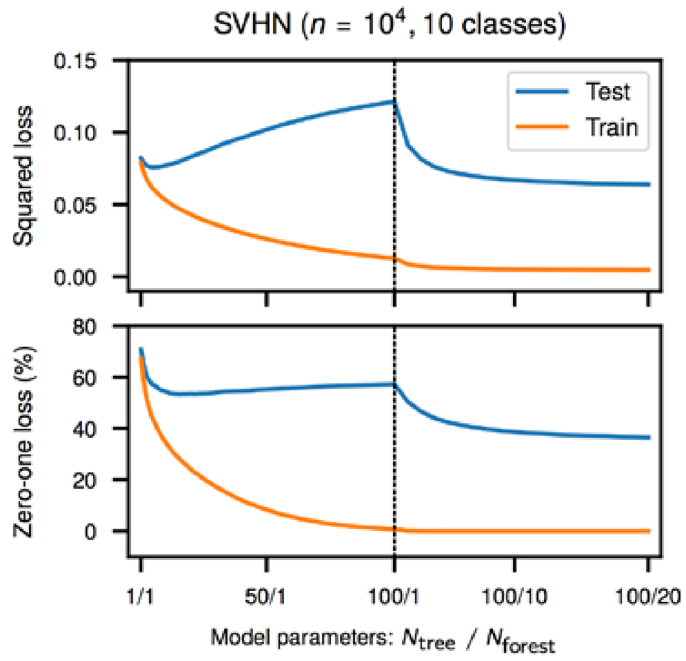


[B., Hsu, Ma, Mandal, PNAS 2019]

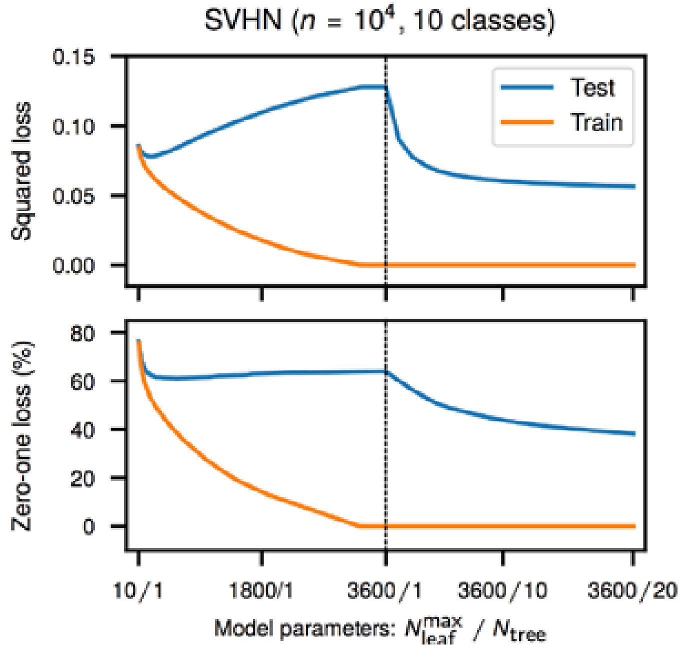
Fully connected network

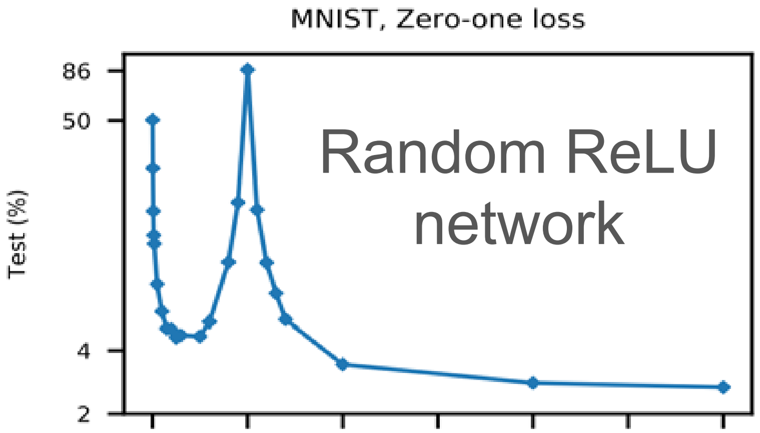


Random Forest

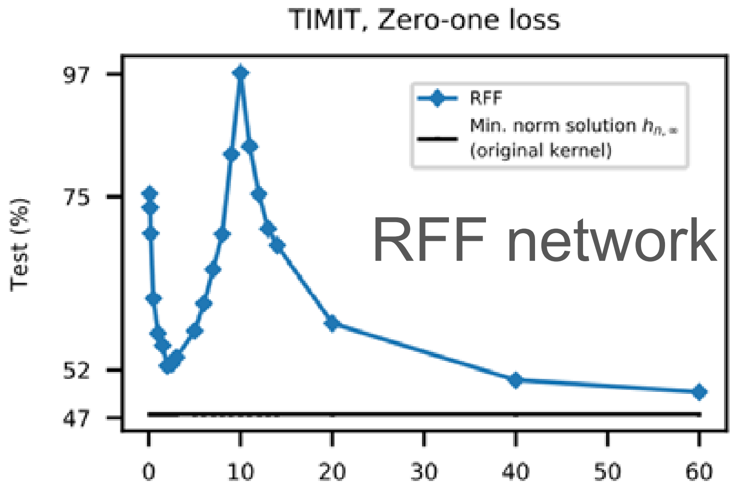


L2-boost

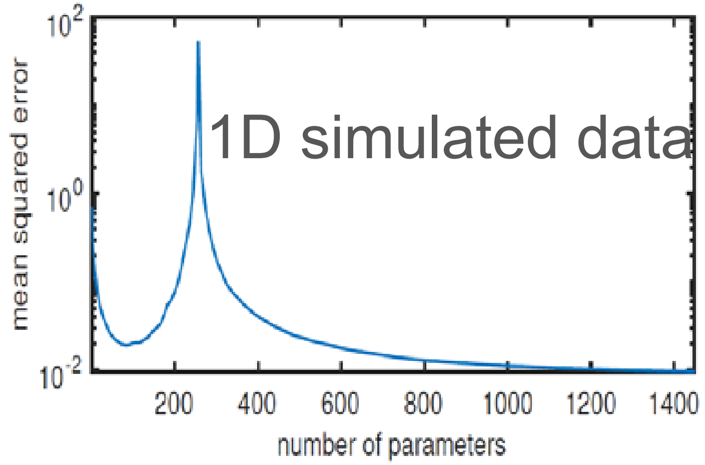




Random ReLU network

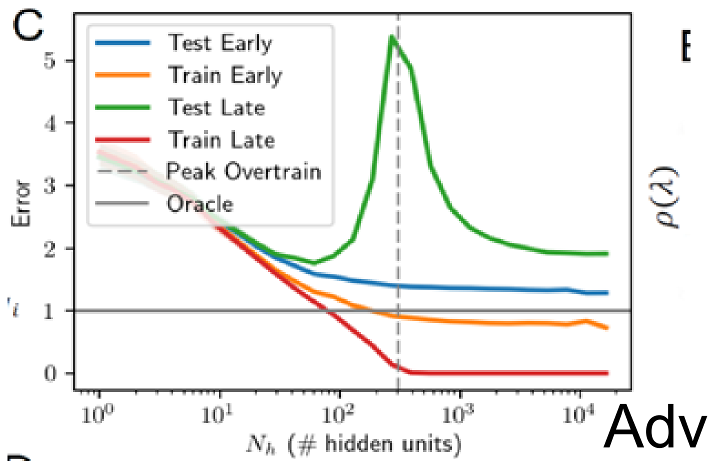


RFF network

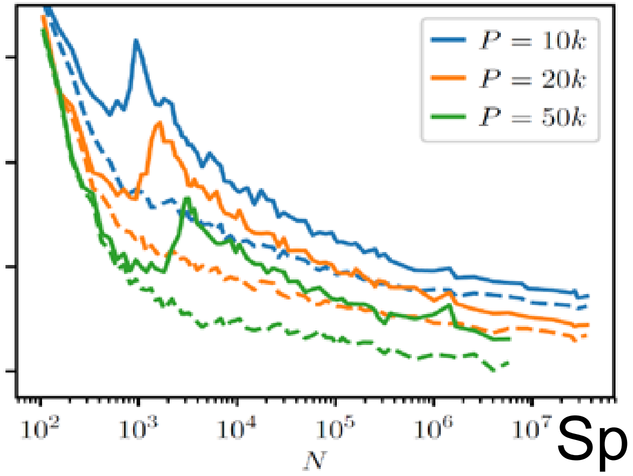


1D simulated data

[B., Hsu, Ma, Mandal, 18]

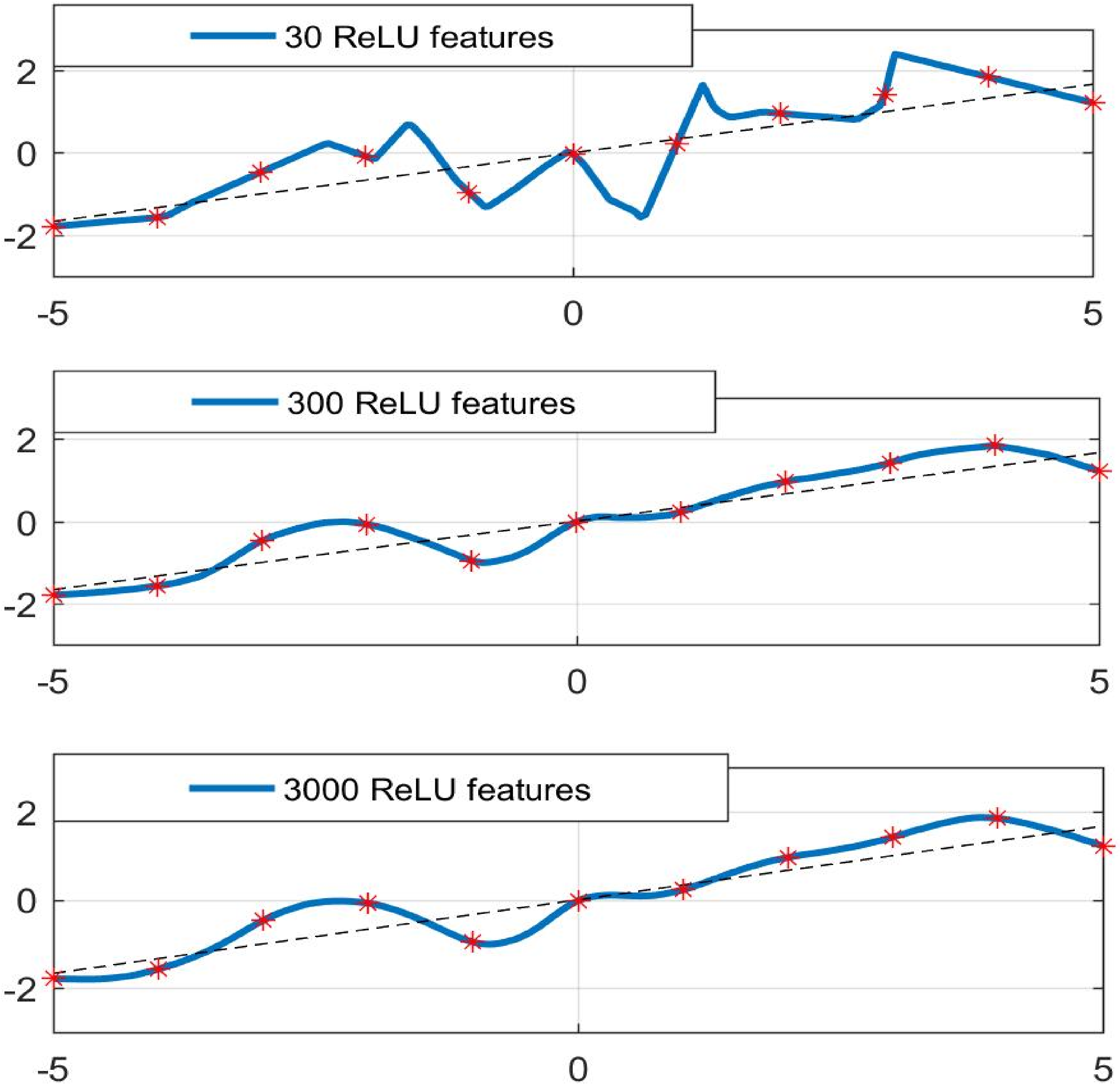


Advani, Saxe, 2017



Spigler, et al, 2018

More parameters are better: an example



Random Fourier networks

Random Fourier Features networks [Rahimi, Recht, NIPS 2007]

$$h_{n,N}(x) = \sum_{j=1}^{N} \alpha_j \, e^{i\pi\langle w_j, x\rangle}$$

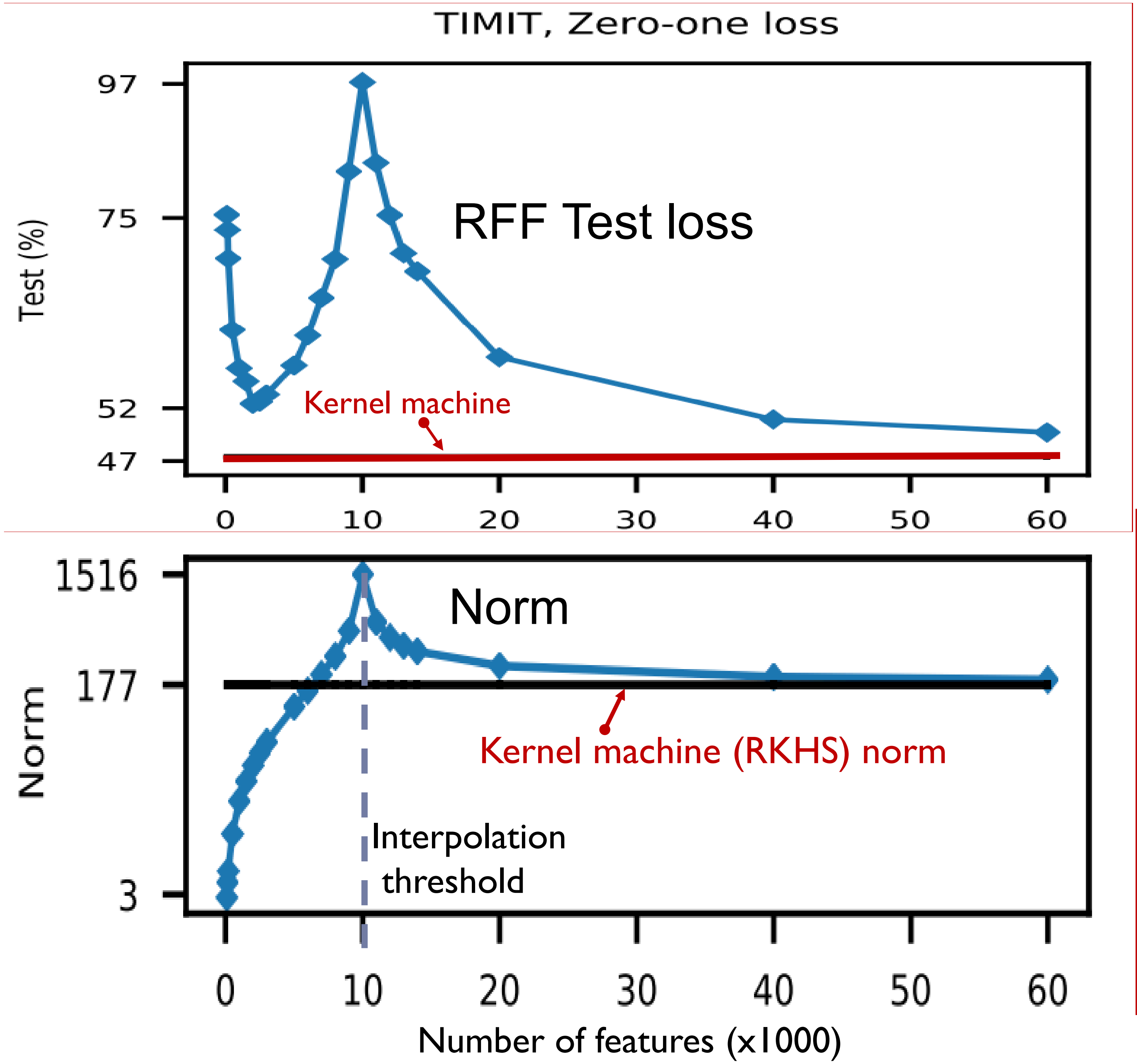
Neural network with one hidden layer, *cos* non-linearity, fixed first layer weights. Hidden layer of size N. Data size n.

Key property:

$$\lim_{N\to\infty} h_{n,N}(x) = \text{kernel machine}$$

.

What is the mechanism?



$N \to \infty$ -- infinite neural net

=

kernel machine

=

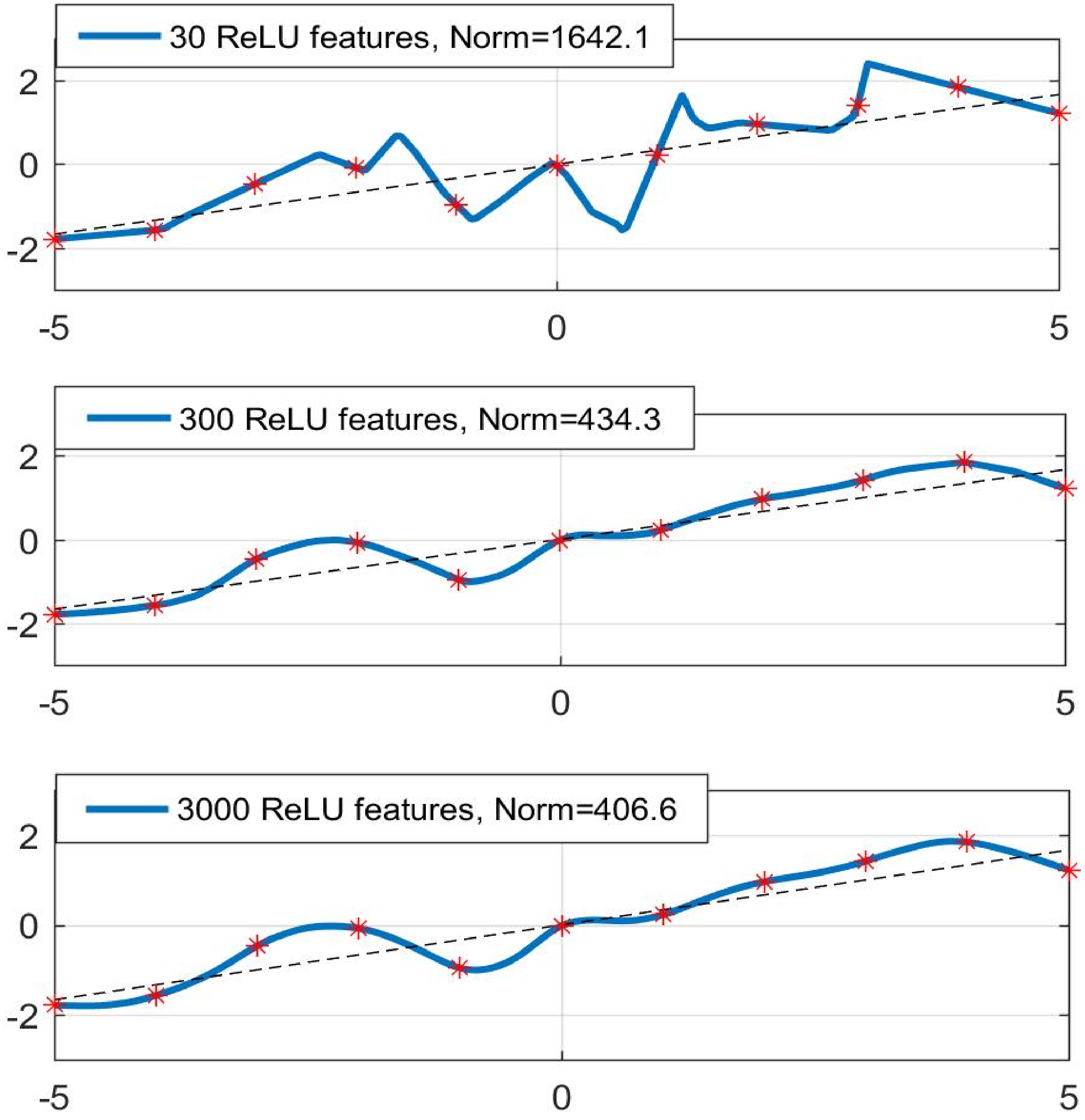
minimum norm solution

$$\operatorname{argmin}_{h\in\mathcal{H},\, h(x_i)=y_i} \; ||h||_{\mathcal{H}}$$

More features ⇒

better approximation
to minimum norm solution

Smaller norm ensures smoothness



Is infinite width optimal?

Infinite net (kernel machine) $h_{n,\infty}$ is near-optimal empirically.

Suppose $\forall_i \;\; y_i = h^*(x_i)$ for some $h^* \in \mathcal{H}$ (Gaussian RKHS).

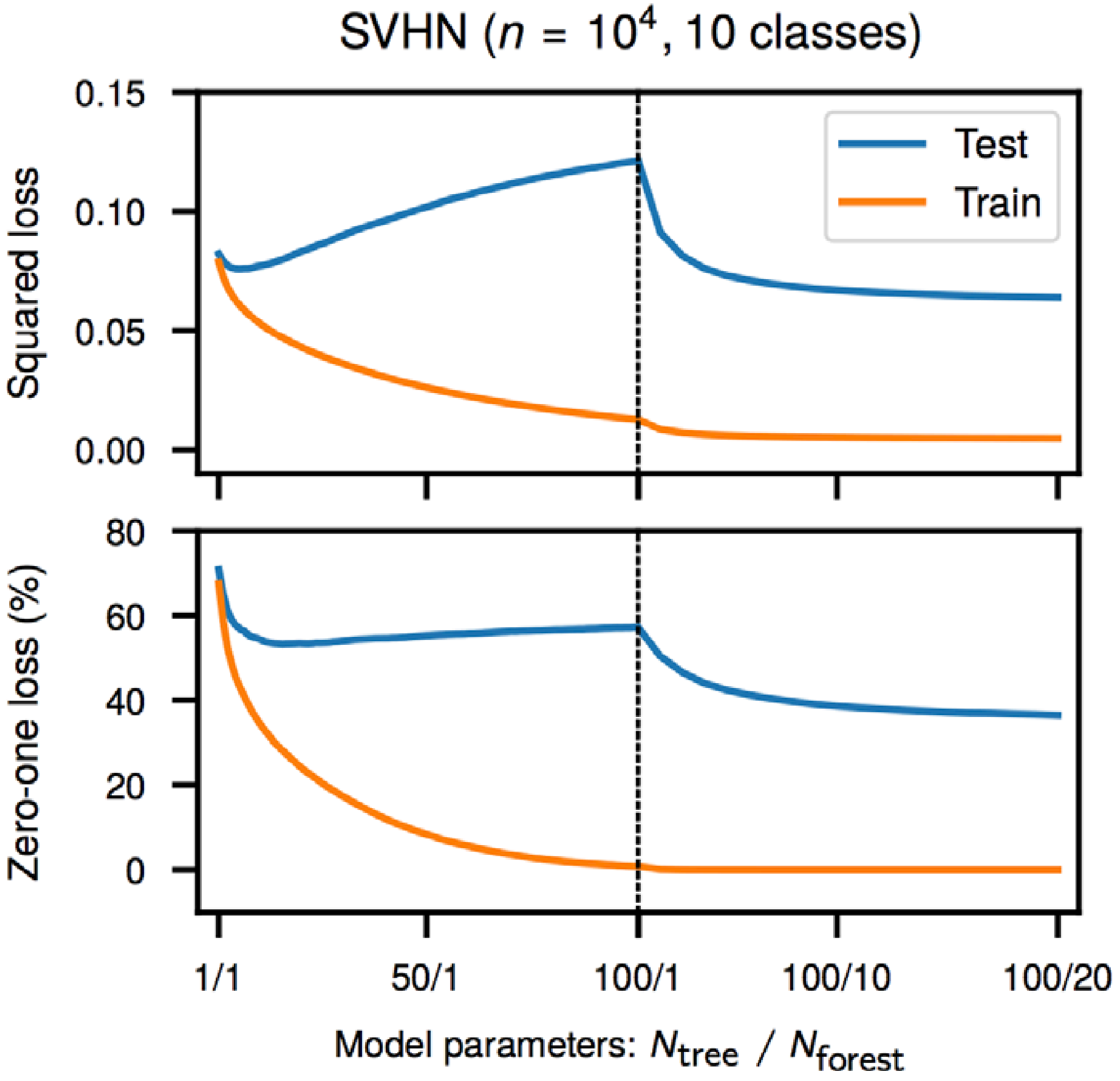
Theorem (noiseless case):

$$\left|h^*(x) - \; h_{n,\infty}(x)\right| < \; Ae^{-B(n/\log n)^{1/d}} \; ||h^*||_{\mathcal{H}}$$

Compare to $O\left(\frac{1}{\sqrt{n}}\right)$ for classical bias-variance analyses.

[B., Hsu, Ma, Mandal, PNAS 19]

Smoothness by averaging

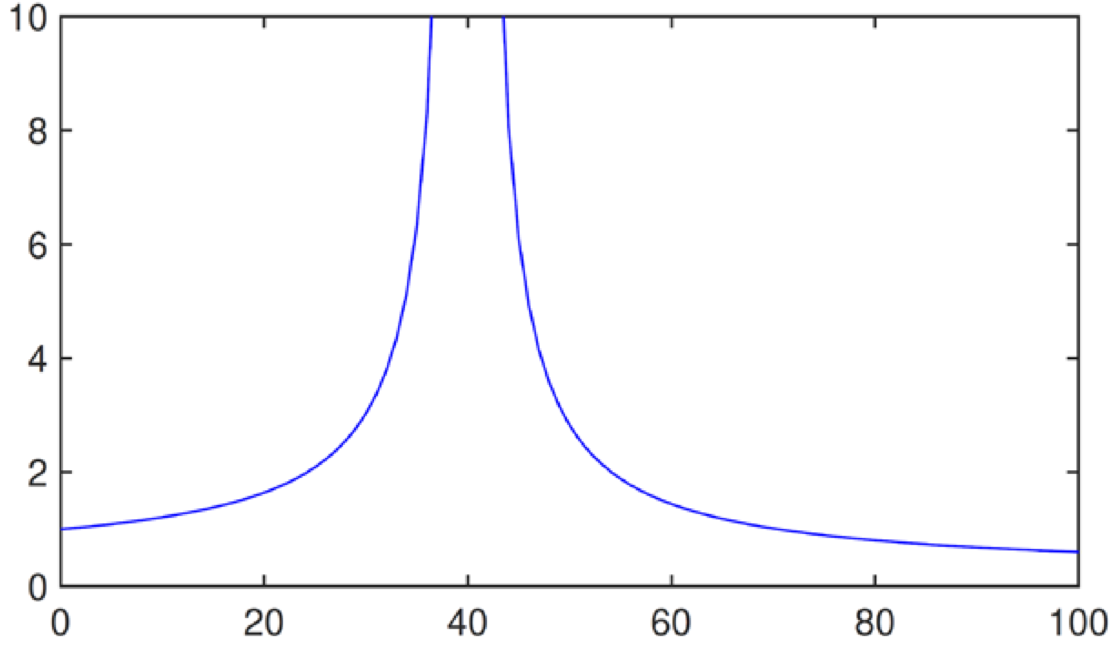


An average of interpolating trees is interpolating and better than any individual tree.

Cf. PERT [Cutler, Zhao 01]

Double Descent in Random Feature settings

Choosing maximum number of features is provably optimal under the "weak random feature" model.



[B., Hsu, Xu, 19].

Related work: [Bartlett, Long, Lugosi, Tsigler 19], [Hastie, Montanari, Rosset, Tibshirani 19] [Mitra, 19], [Muthukumar, Vodrahalli, Sahai, 19] [Mei, Montanari, 19] [Liang, Rakhlin, 19], [Liang, Rakhlin, Zhai, 19]

Significant evidence that deep neural networks exhibit similar properties.

Framework for modern ML

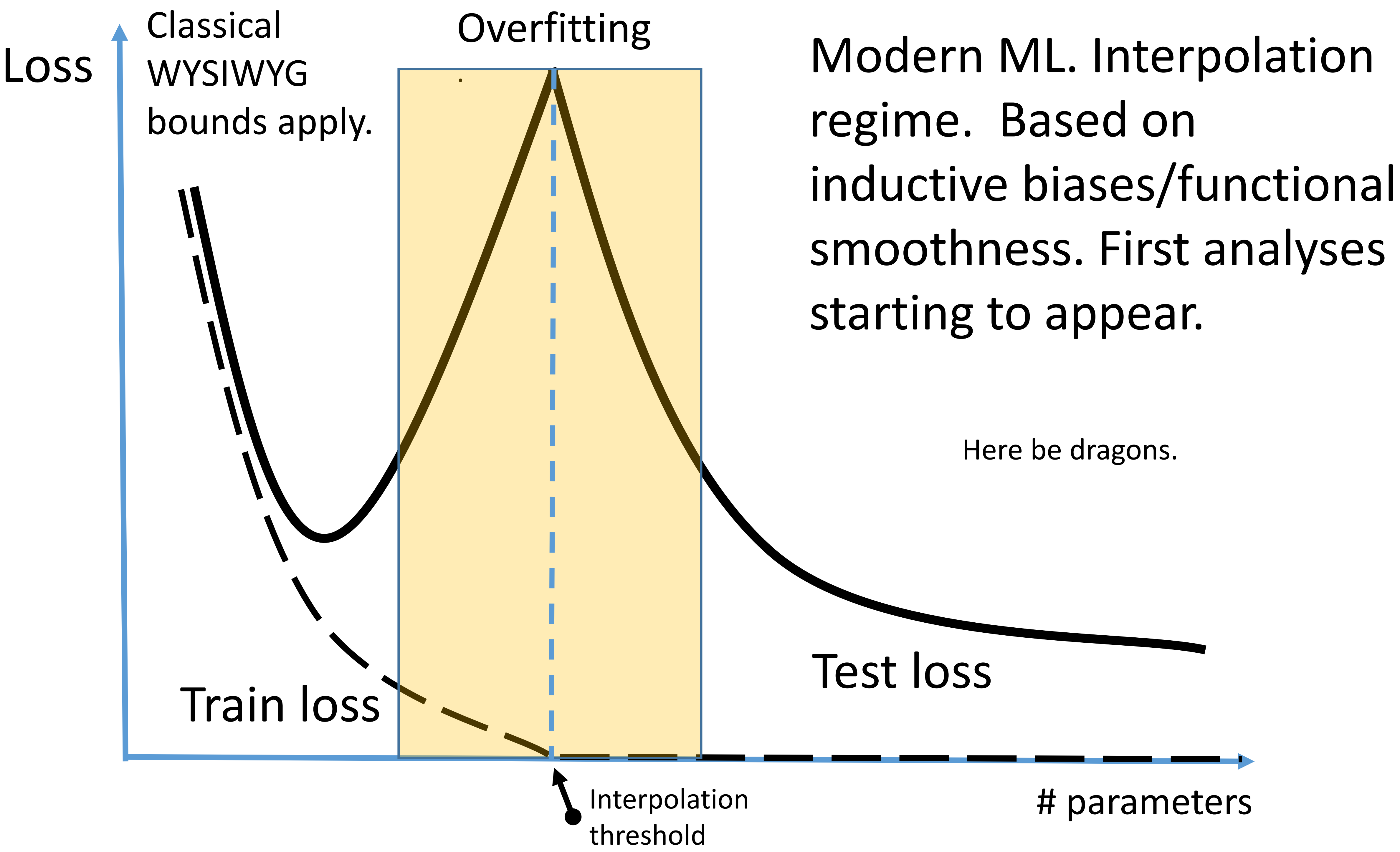
Occam's razor based on inductive bias: Maximize smoothness subject to interpolating the data.

Three ways to increase smoothness:

- Explicit: minimum functional norm solutions
  - Exact: kernel machines.
  - Approximate: RFF, ReLU features.
- Implicit: SGD/optimization (Neural networks)
- Averaging (Bagging, L2-boost).

All coincide for kernel machines.

The landscape of generalization



Classical WYSIWYG bounds apply.

Overfitting

Modern ML. Interpolation regime. Based on inductive biases/functional smoothness. First analyses starting to appear.

Here be dragons.

This talk

- ➢Statistical theory of interpolation.
  - Why classical bounds do not apply.
  - Statistical validity of interpolation.

- ➢The generalization landscape of Machine Learning.
  - Double Descent: reconciling interpolation and the classical U curve.
  - Occams razor: more features is better.

- ➢Interpolation and optimization
  - Easy optimization + fast SGD (+ good generalization).
  - Learning from deep learning for efficient kernel machines.

Optimization: classical

Classical (under-parametrized):

- Many local minima.
- SGD (fixed step size) does not converge.

Modern Optimization

Modern (interpolation/over-parametrized).

1. Every local minimum is global (for networks wide enough)
[Li, Ding, Sun, 18], [Yu, Chen, 95]

2. Local methods converge to global optima
[Kawaguchi, 16] [Soheil, et al, 16] [Bartlett, et al, 17]
[Soltanolkotabi, et al, 17, 18] [Du, et al, 19] …

3. Small batch SGD (fixed step size) converges as fast as GD per iteration.
[Ma, Bassily, **B.**, ICML 18] [Bassily, Ma, **B.**, 18]

Why SGD?

$$w^* = \underset{w}{\operatorname{argmin}} L(w) = \underset{w}{\operatorname{argmin}} \frac{1}{n} \sum L_i(w)$$

SGD Idea: optimize $\sum L_i(w)$, m at a time.

Error after t steps GD: e^{-t}

SGD: $1/t$

What is the reason for practical success?

Key point: SGD is not simply GD with noisy gradient estimates.

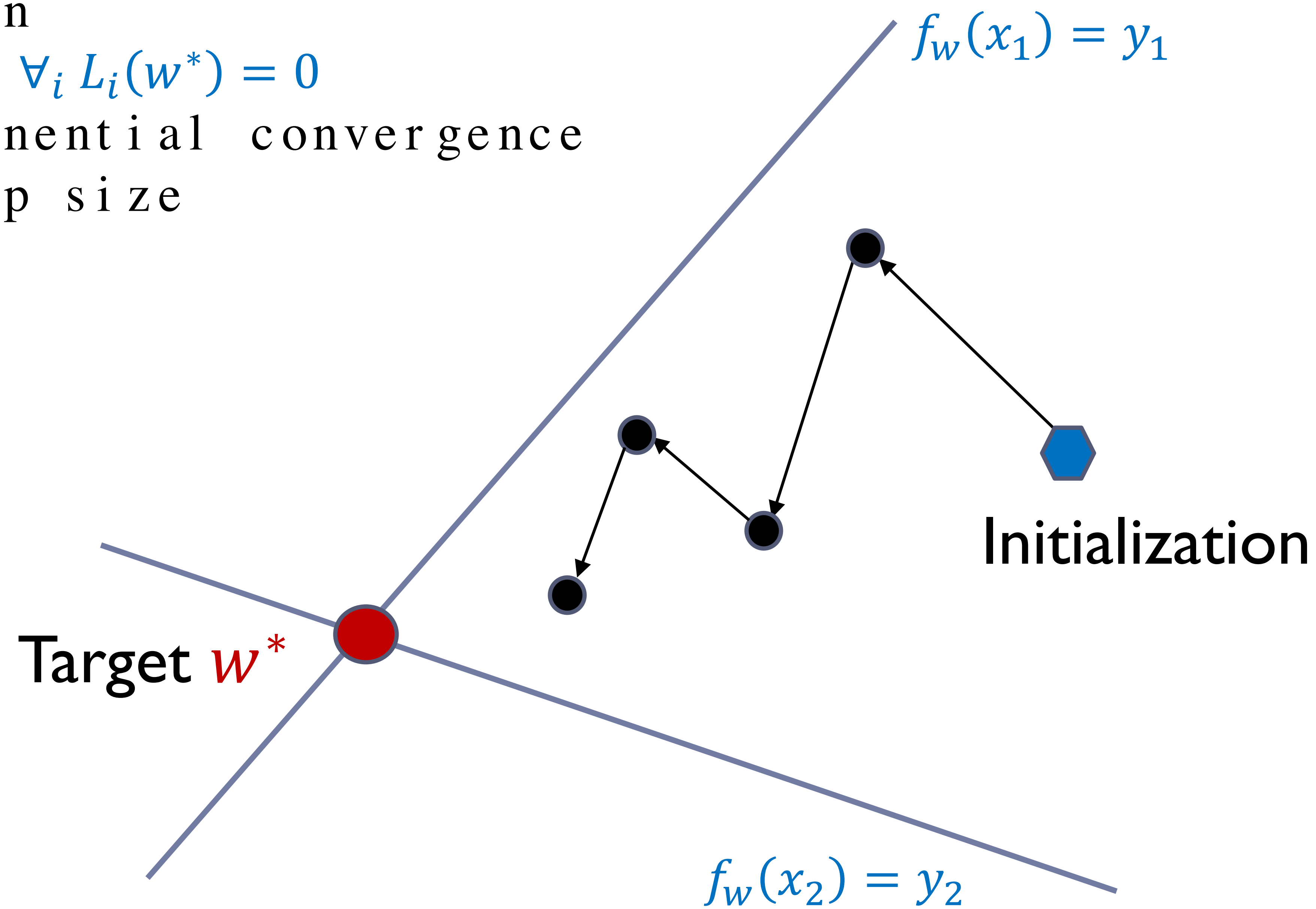
SGD under interpolation

Key observation:

Interpolation

$f_{w^*}(x_i) = y_i \;\; \Rightarrow \;\; \forall_i \, L_i(w^*) = 0$

implies exponential convergence w. fixed step size



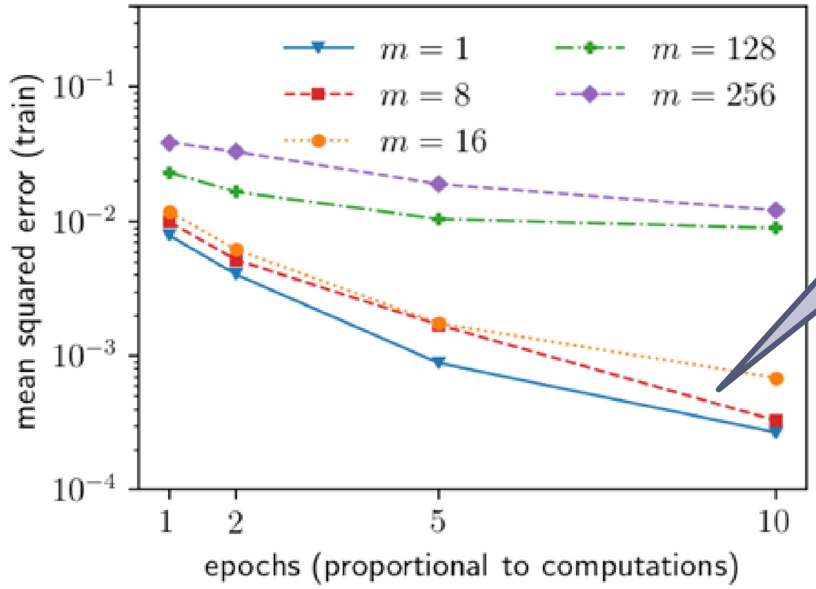
SGD is (much) faster than GD

"Theorem": one SGD iteration with mini-batch size $m^* = \frac{tr\ H}{\lambda_1(H)}$ is equivalent to an iteration (=epoch) of full GD.

Savings per epoch: n/m^*.

Real data example.

One step of SGD with mini-batch
$m^* \approx 8$
=
One step of GD.

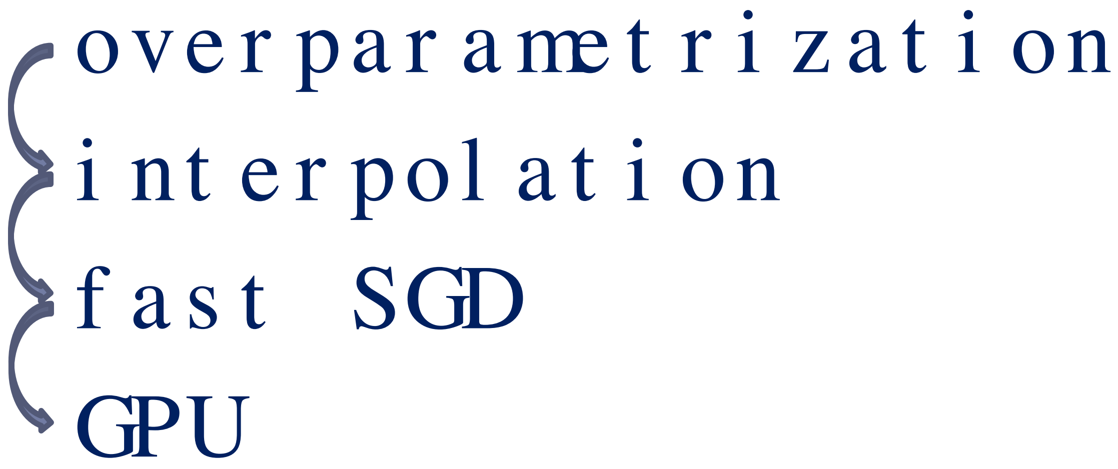


$m^* = 8$

[Ma, Bassily, **B.**, ICML 18]

The power of interpolation

Optimization in modern deep learning:



overparametrization

interpolation

fast SGD

GPU

SGD computational gain over GD $O\left(\frac{n}{m^*}\right)$

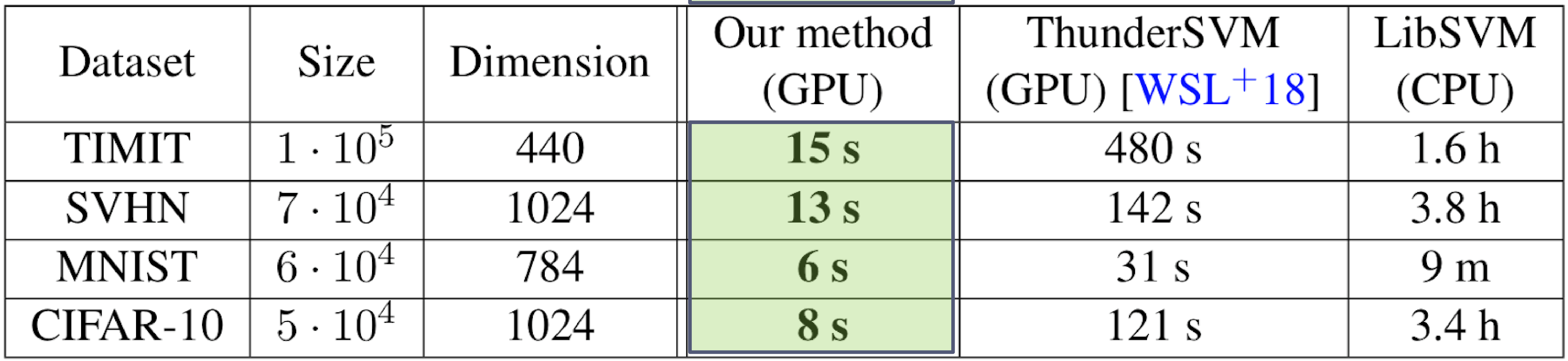
* GPU ~100x over CPU.

$n = 10^6, m^* = 8$:

SGD on GPU ~10^7x faster than GD on CPU!

Learning from deep learning: fast and effective kernel machines

EigenPro 2.0



Dataset	Size	Dimension	Our method (GPU)	ThunderSVM (GPU) [WSL+18]	LibSVM (CPU)
TIMIT	$1 \cdot 10^5$	440	**15 s**	480 s	1.6 h
SVHN	$7 \cdot 10^4$	1024	**13 s**	142 s	3.8 h
MNIST	$6 \cdot 10^4$	784	**6 s**	31 s	9 m
CIFAR-10	$5 \cdot 10^4$	1024	**8 s**	121 s	3.4 h

EigenPro: preconditioned SGD for kernel machines. Batch size/preconditioner optimized to take full advantage of GPU.

Code: https://github.com/EigenPro

[Ma, B., NIPS 17, SysML 19]

Points and lessons

- ERM cannot a sole foundation for modern ML.
  - Instead of uniform laws of large numbers, need to study inductive biases. Still early but analyses are starting to appear.
  - Key concept is interpolation, not over-parametrization. Over-parametrization enables interpolation but is not sufficient. Classical methods, kernels machines/splines are infinitely over-parametrized.

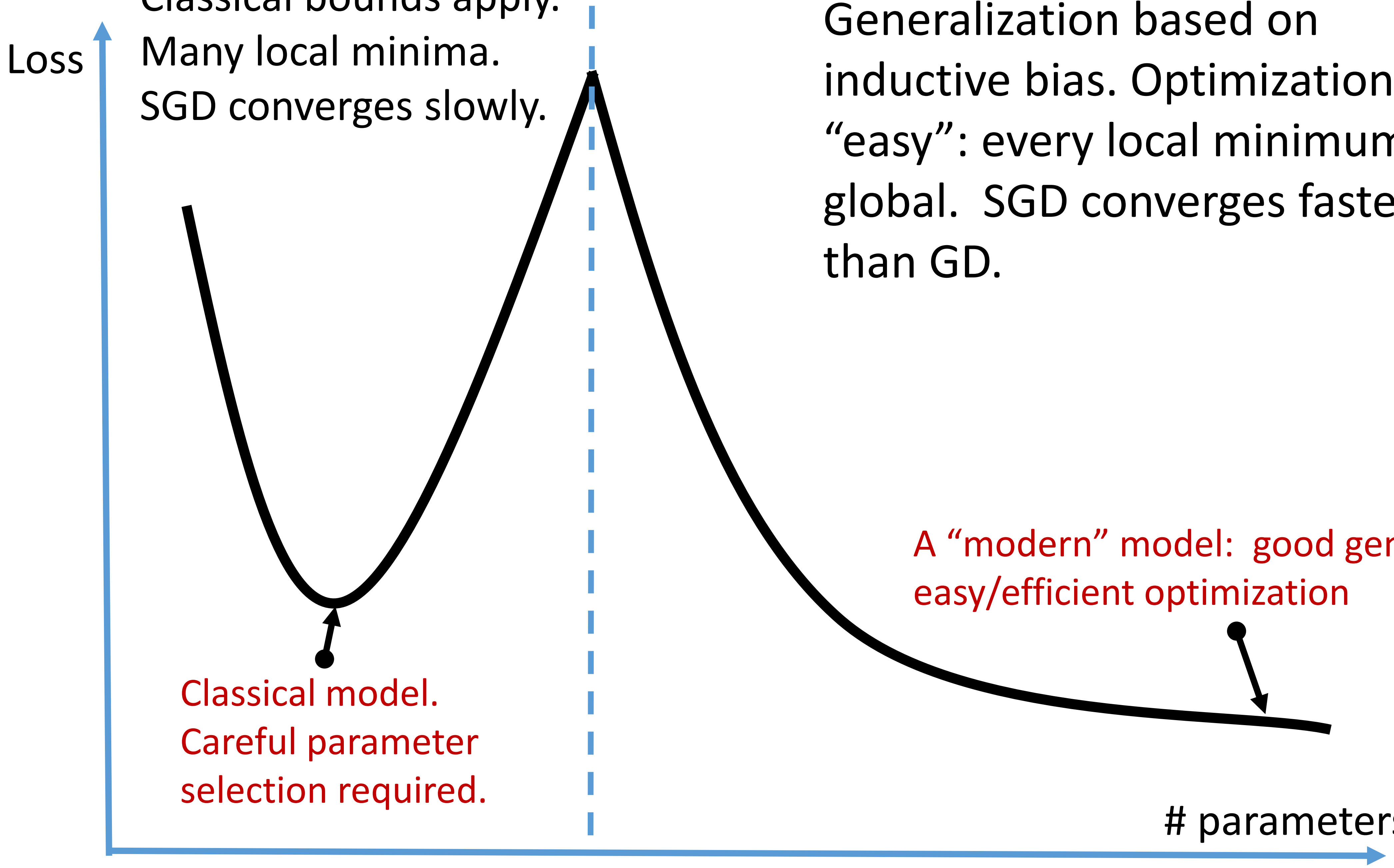
- Empirical loss is a useful optimization target, not a meaningful statistic for the expected loss.

- Optimization is qualitatively different under interpolation.
  - Every local minimum is global.
  - SGD is overwhelmingly faster than GD.

From classical statistics to modern ML

Classical.

Modern ML (interpolation regime).



Classical bounds apply.
Many local minima.
SGD converges slowly.

Generalization based on inductive bias. Optimization is “easy”: every local minimum is global. SGD converges faster than GD.

Classical model.
Careful parameter selection required.

A “modern” model: good generalization + easy/efficient optimization

Collaborators:

Siyuan Ma, Ohio State University
Soumik Mandal, Ohio State University

Daniel Hsu, Columbia University
Raef Bassily, Ohio State University
Partha Mitra, Spring Harbor Labs.
Sasha Rakhlin, MIT
Sasha Tsybakov, ENSAE

Thank you