Predictive models from interpolation

Daniel Hsu

Computer Science Department & Data Science Institute

Columbia University

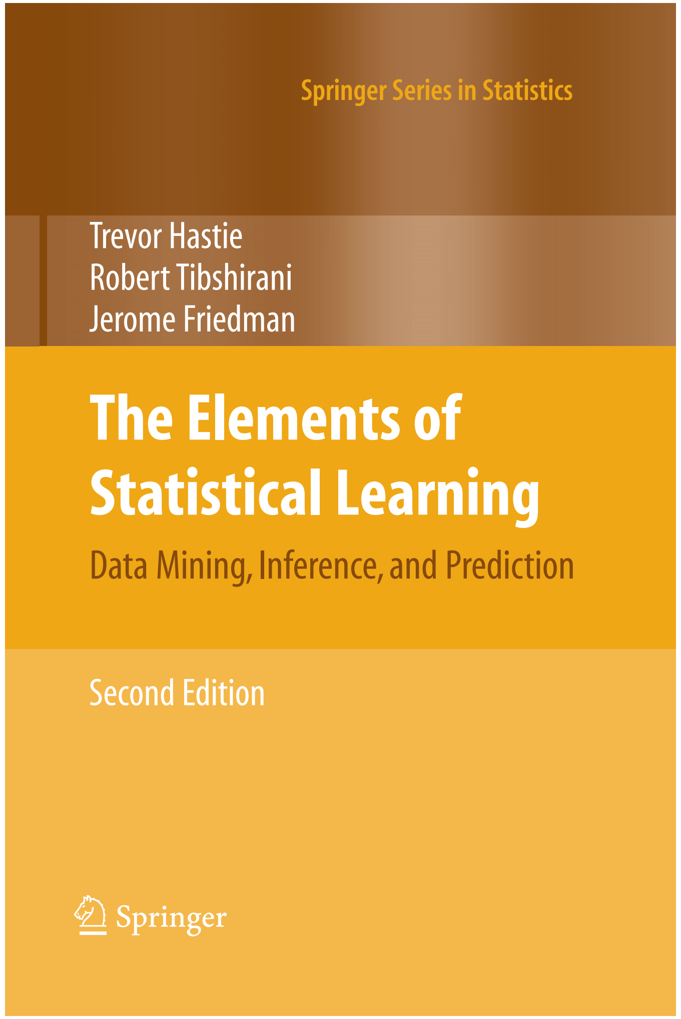
Center for Machine Learning @ University of Maryland
October 7, 2019

Spoilers

"A model with zero training error is overfit to the training data and will typically generalize poorly."

– Hastie, Tibshirani, & Friedman, *The Elements of Statistical Learning*

We'll give empirical and theoretical evidence against this conventional wisdom, at least in some "modern" settings of machine learning.



Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

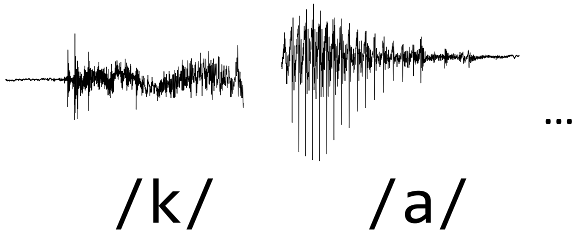
Second Edition

Springer

Outline

1. Empirical evidence that counter the conventional wisdom
2. Interpolation via local prediction
3. Interpolation via neural nets and linear models
4. Brief remark about adversarial examples

Supervised learning



Training data (labeled examples)
$(x_1, y_1), \dots, (x_n, y_n)$ from $\mathcal{X} \times \mathcal{Y}$

(IID from P)

$w \leftarrow w - \eta \nabla \hat{\mathcal{R}}(w)$

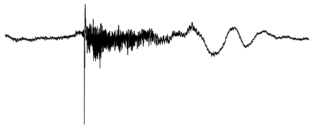
Learning algorithm

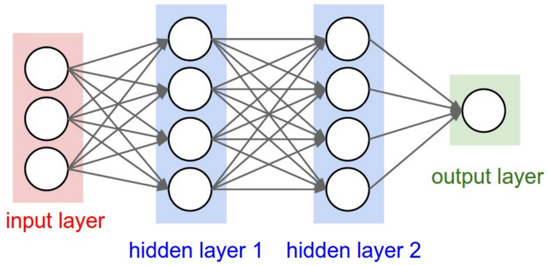
Risk: $\mathcal{R}(f) := \mathbb{E}[\ell(f(x'), y')]$
where $(x', y') \sim P$

Test point
$x' \in \mathcal{X}$

Prediction function
$\hat{f}: \mathcal{X} \to \mathcal{Y}$

Predicted label
$\hat{f}(x') \in \mathcal{Y}$





/t/

Standard approach to supervised learning

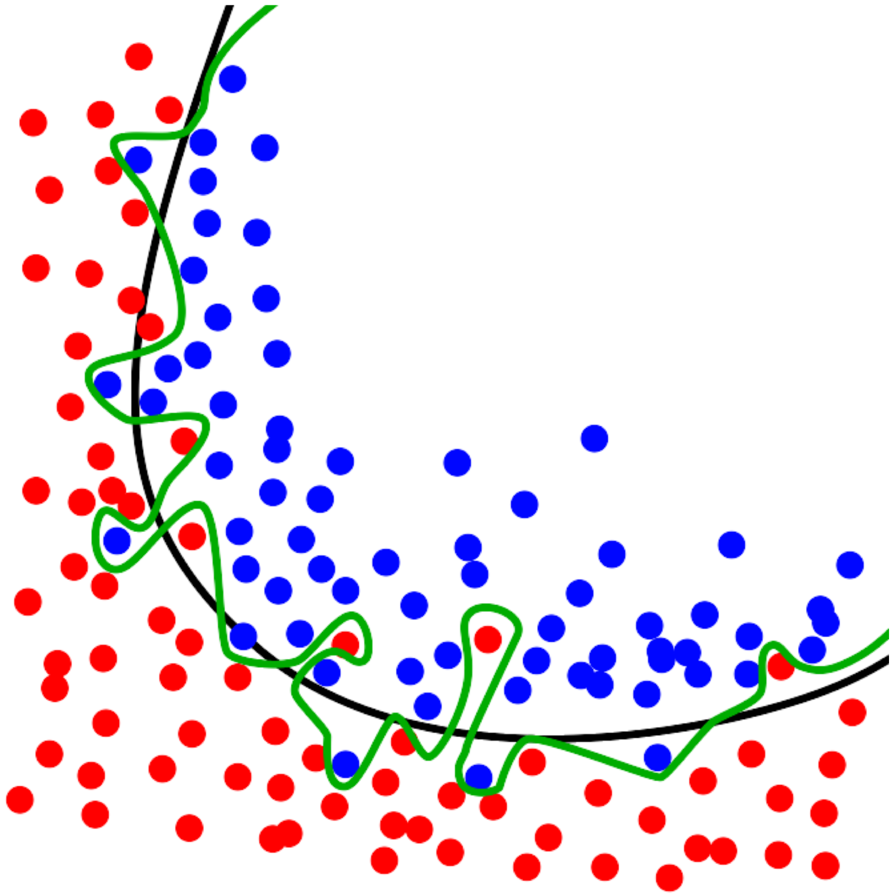
- Choose (parameterized) **function class** $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$
  - E.g., linear functions, polynomials, neural networks with certain architecture
- Use optimization algorithm to (attempt to) minimize **empirical risk**

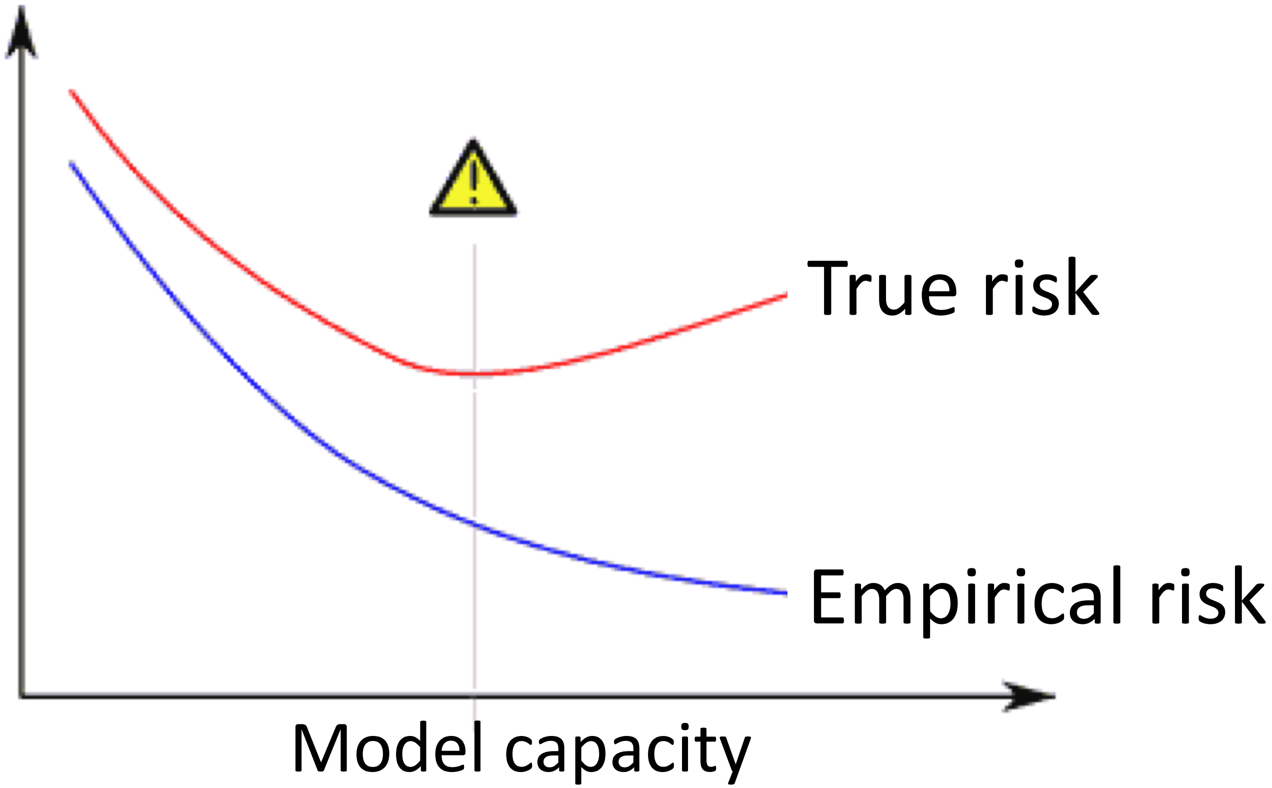
$$\hat{\mathcal{R}}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

(a.k.a. **training error**).

- **But how "big" or "complex" should this function class be?**
  (Degree of polynomial, size of neural network architecture, …)

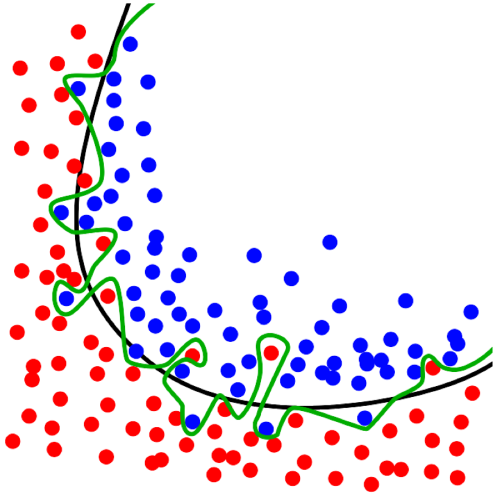
Overfitting

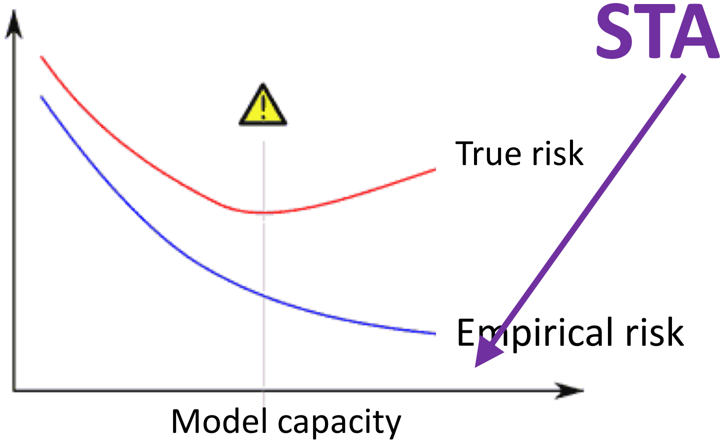




Deep learning practice

- Ruslan Salakhutdinov (Foundations of Machine Learning Boot Camp @ Simons Institute, January 2017)
  - (Paraphrased) "First, choose a network architecture large enough such that it is easy to overfit your training data. [...] Then, add regularization."





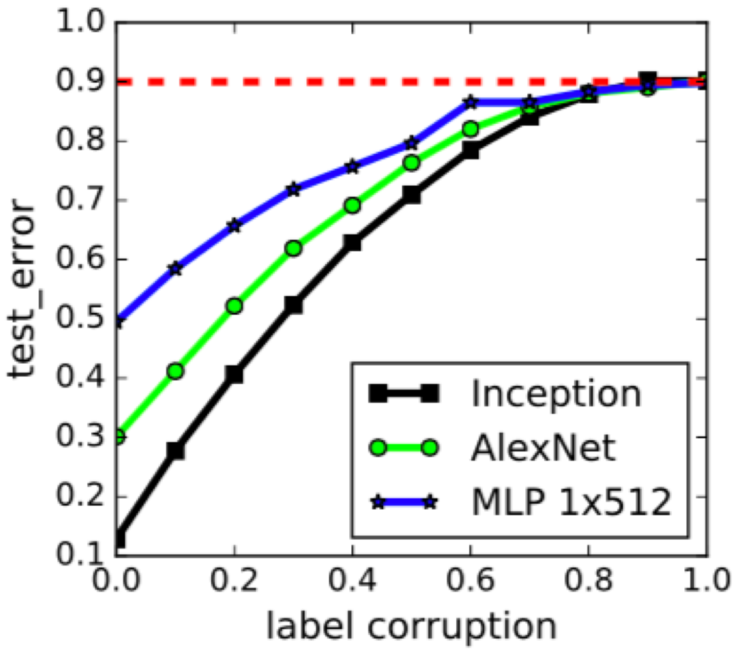
START HERE???

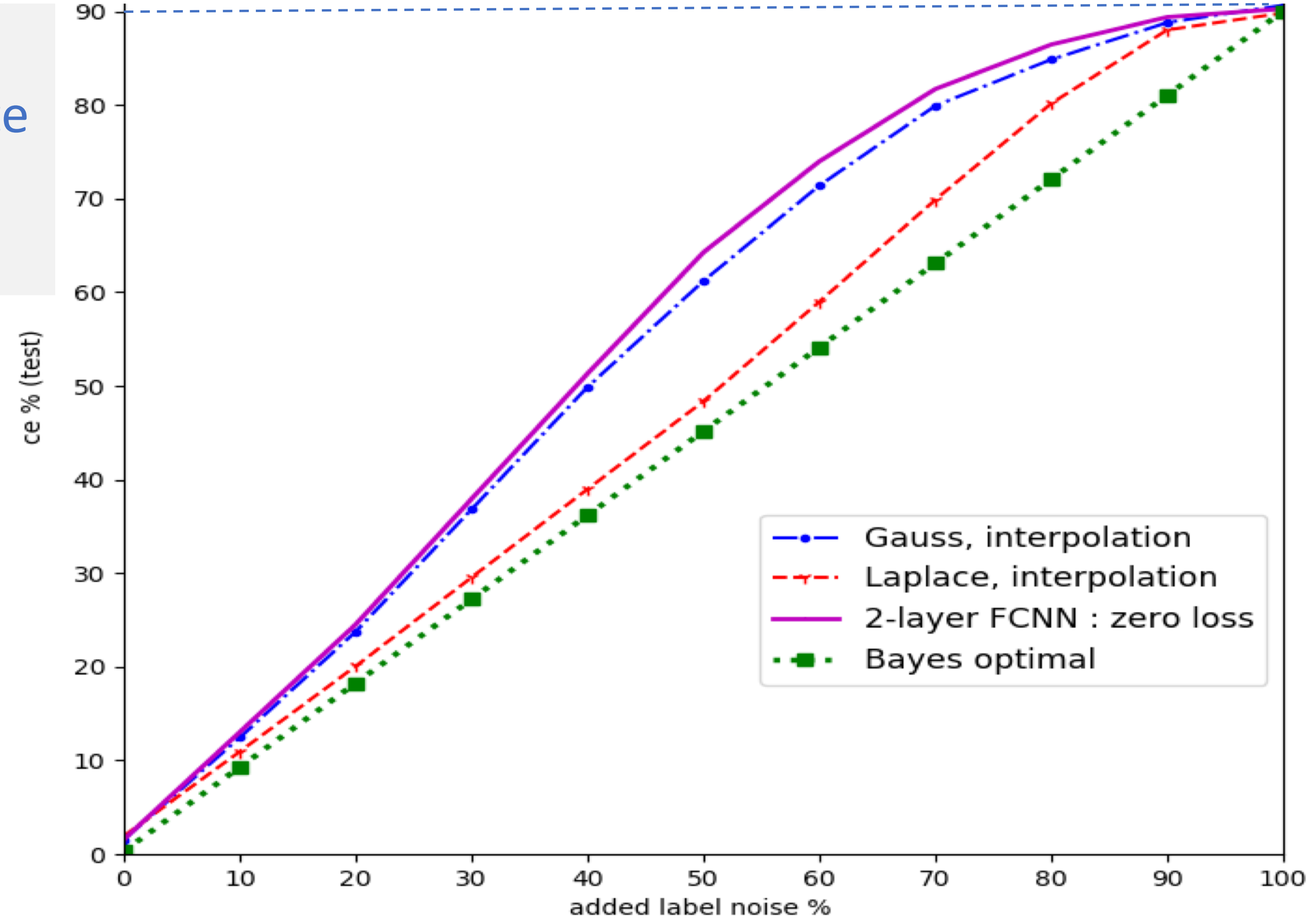
Empirical observations

(Zhang, Bengio, Hardt, Recht, & Vinyals, 2017; Belkin, Ma, & Mandal, 2018)

Neural nets & kernel machines:

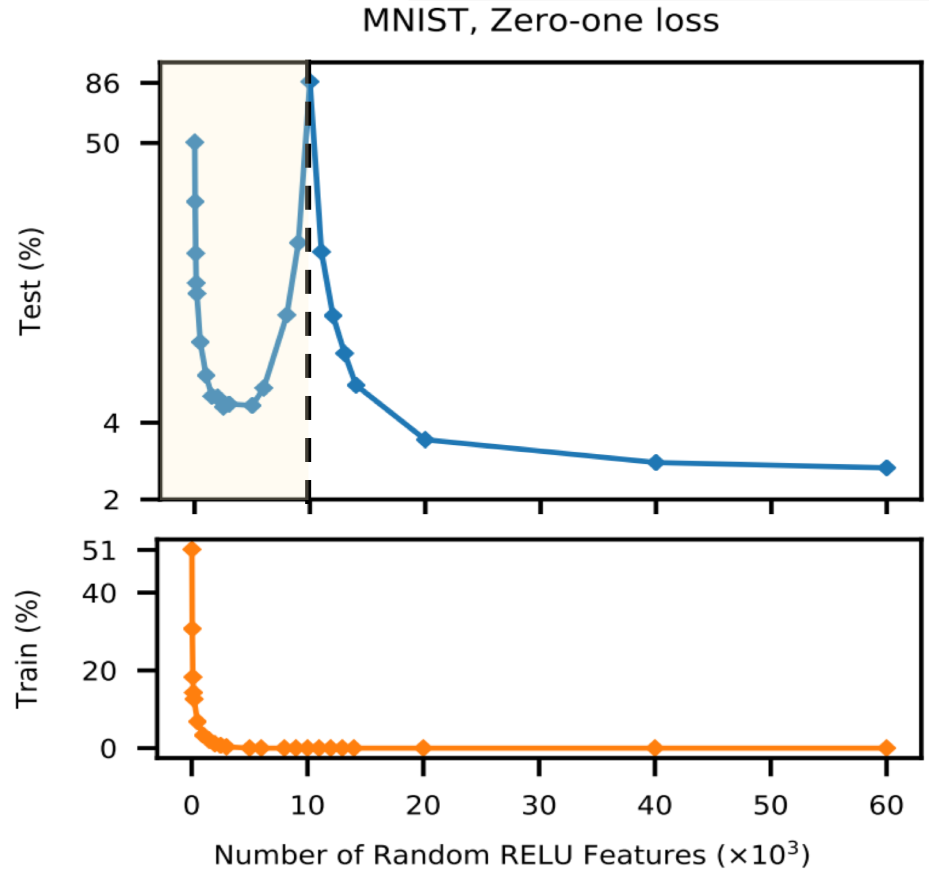
- Large-enough models interpolate noisy training data but are still accurate out-of-sample!



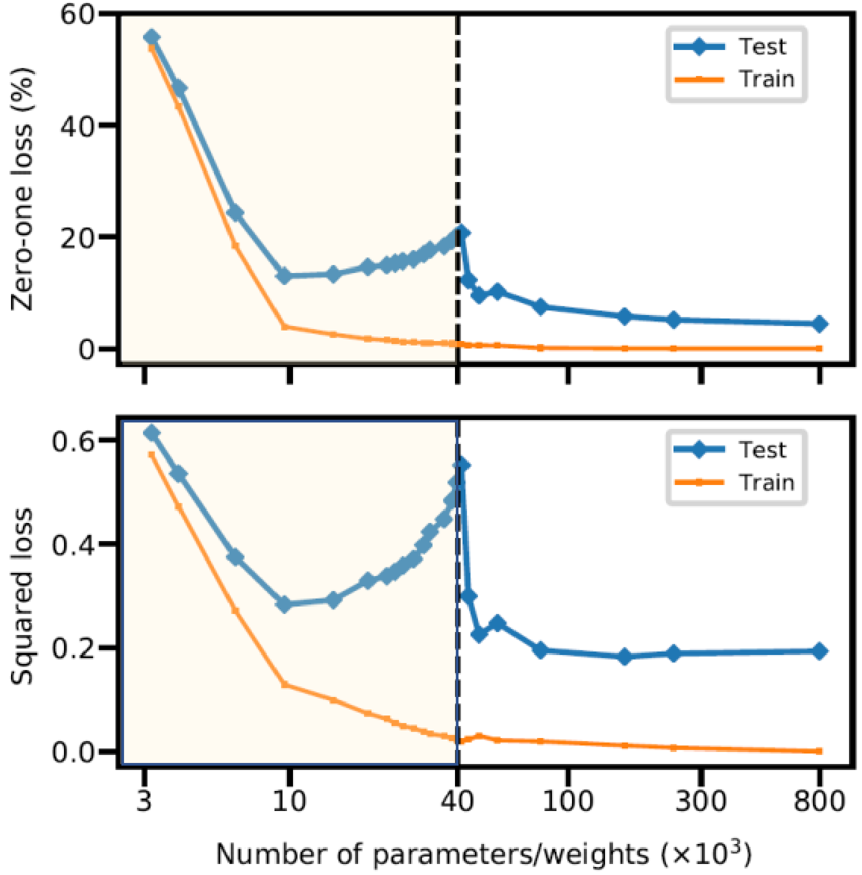


More empirical observations [Belkin, H., Ma, Mandal, PNAS'19]

But: Not every interpolating model is equally good!



Random first layer



Trained first layer

Interpolation in machine learning

- **Supervised learning**: use training examples to find function that predicts accurately on new example
- **Interpolation**: find function that perfectly fits training examples

---

- **PAC learning** (Valiant, 1984; Blumer, Ehrenfeucht, Haussler, & Warmuth, 1987; …)
  - realizable, noise-free setting with bounded-capacity hypothesis class
- **Regression models** (Whittaker, 1915; Shannon, 1949; …)
  - noise-free data with "simple" models (e.g., linear models with $n \geq p$)

Far from what we are observing in practice…

Our goals

- **Counter the "conventional wisdom" re: interpolation**
  Show interpolation methods can be consistent (or almost consistent) for classification & regression
  - Simplicial interpolation
  - Weighted & interpolated nearest neighbor
  - Neural nets / linear models
- Identify **useful properties of good interpolation** methods

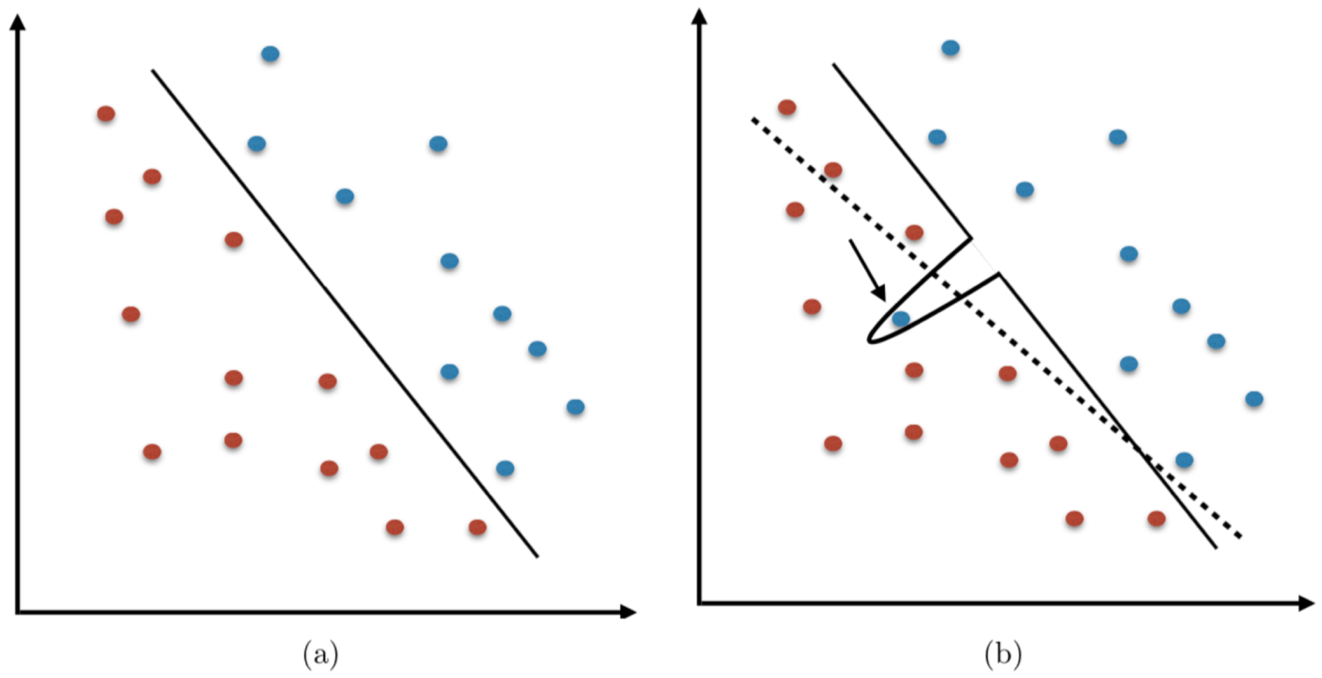
Interpolation via local prediction

Even more empirical observations

(Wyner, Olson, Bleich, & Mease, 2017)

AdaBoost + large decision trees / Random forests:

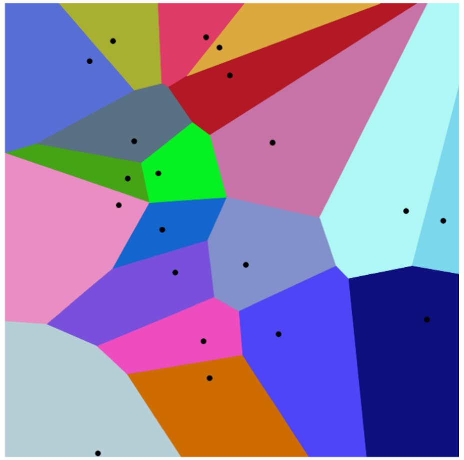
- Interpret as local interpolation methods
- Flexibility -> robustness to label noise



Existing theory about local interpolation

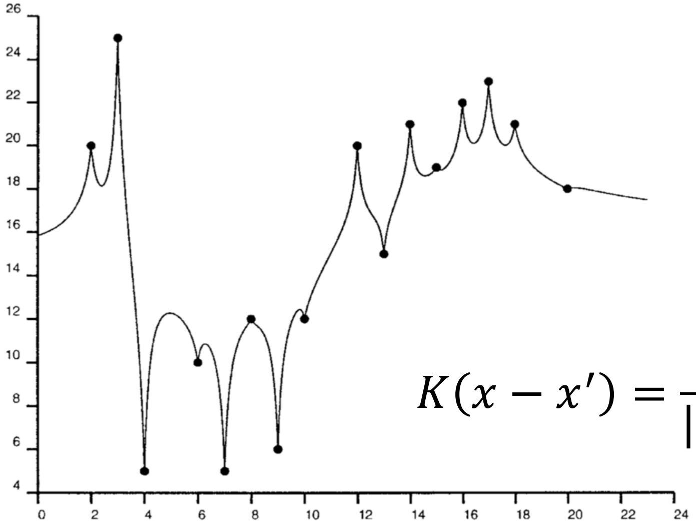
Nearest neighbor (Cover & Hart, 1967)

- Predict with label of nearest training example
- Interpolates training data
- Risk → $2 \cdot \mathrm{OPT}$ (sort of)



Hilbert kernel (Devroye, Györfi, & Krzyżak, 1998)

- Special kind of smoothing kernel regression (like Shepard's method)
- Interpolates training data
- Consistent*, but no convergence rates



$$K(x - x') = \frac{1}{\|x - x'\|^d}$$

FIG. 2. The Hilbert kernel regression estimate with $a = 1$.

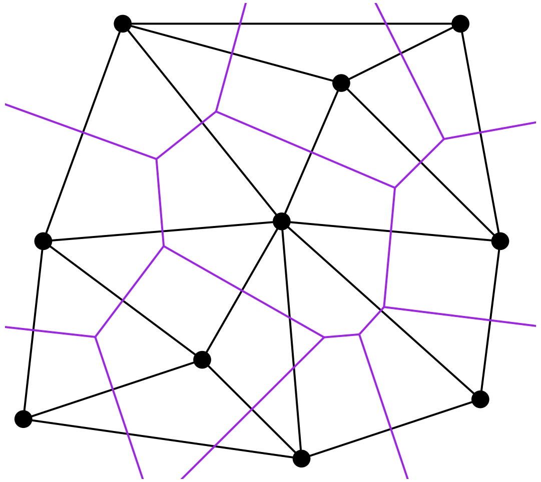
Preliminaries

- Construct estimate $\hat{\eta}_n$ of the **regression function**
  $$\eta(x) = \mathbb{E}[\, y' \mid x' = x \,]$$
- For **binary classification** $\mathcal{Y} = \{0,1\}$:
  - $\eta(x) = \Pr(y' = 1 \mid x' = x)$
  - **Optimal classifier**: $f^*(x) = \mathbb{I}_{\eta(x) > \frac{1}{2}}$
  - **Plug-in classifier**: $\hat{f}_n(x) = \mathbb{I}_{\hat{\eta}_n(x) > \frac{1}{2}}$ based on estimate $\hat{\eta}_n$
- **Questions**:
  What is the risk as $n \to \infty$? At what rate does it converge?

I. Simplicial interpolation

AKA "Triangulated irregular network" (Franklin, 1973)

- IID training examples $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^d \times [0,1]$
  - Partition $C \coloneqq \operatorname{conv}(x_1, \dots, x_n)$ into simplices with x_i as vertices via Delaunay.
  - Define $\hat{\eta}_n(x)$ on each simplex by affine interpolation of vertices' labels.
  - Result is piecewise linear on C. (Punt on what happens outside of C.)
- For classification ($y \in \{0,1\}$), $\hat{f}_n$ is plug-in classifier based on $\hat{\eta}_n$.



Asymptotic risk for simplicial interpolation

[Belkin, H., Mitra, NeurIPS'18]

Theorem (classification): Assume distribution of x' is uniform on some convex set, and η is bounded away from $1/2$. Then simplicial interpolation's plug-in classifier $\hat{f}_n$ satisfies

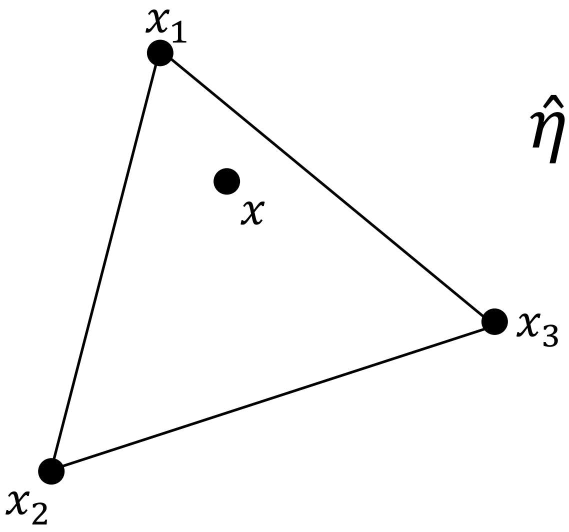
$$\limsup_n \mathbb{E}[\text{zero/one loss}] \ \le \left(1 + e^{-\Omega(d)}\right) \cdot \mathrm{OPT}$$

- **C.f. nearest neighbor classifier**: $\limsup_n \mathbb{E}\left[\mathcal{R}(\hat{f})\right] \ \approx 2 \cdot \mathcal{R}(f^*)$
- **For regression** (squared error):

$$\limsup_n \mathbb{E}[\text{squared error}] \ \le \left(1 + O\left(\frac{1}{d}\right)\right) \cdot \mathrm{OPT}$$

What happens on a single simplex

- Simplex on $x_1, \dots, x_{d+1}$ with corresponding labels $y_1, \dots, y_{d+1}$
- Test point x in simplex, with barycentric coordinates $(w_1, \dots, w_{d+1})$.
- Linear interpolation at x (i.e., least squares fit, evaluated at x):

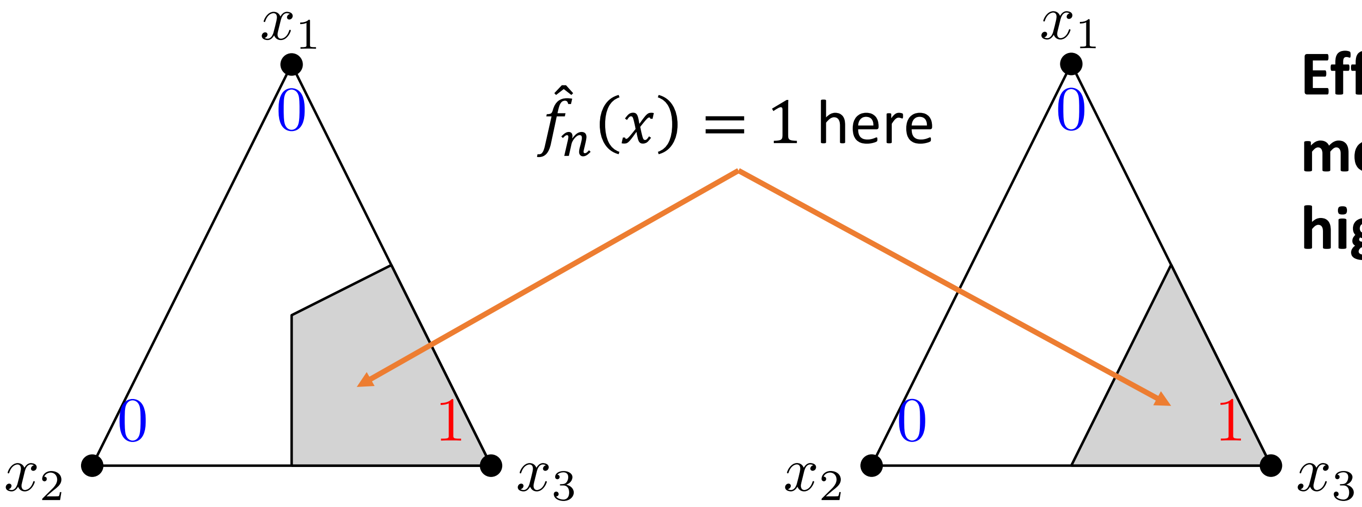


$$\hat{\eta}_n(x) = \sum_{i=1}^{d+1} w_i y_i$$

Key idea: aggregates information from all vertices to make prediction. (C.f. nearest neighbor rule.)

Comparison to nearest neighbor rule

- Suppose $\eta(x) = \Pr(y = 1 \mid x) < 1/2$ for all points in a simplex
  - Optimal prediction of f^* is 0 for all points in simplex.
- Suppose $y_1 = \cdots = y_d = 0$, but $y_{d+1} = 1$ (due to "label noise")



$\hat{f}_n(x) = 1$ here

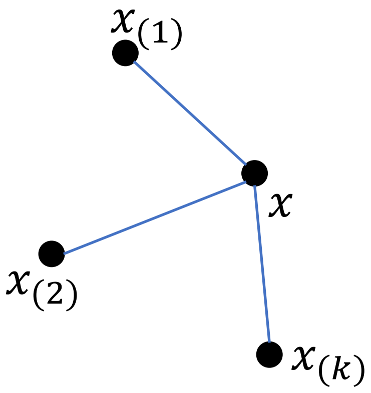
Nearest neighbor rule

Simplicial interpolation

Effect is exponentially more pronounced in high dimensions!

II. Weighted & interpolated NN scheme

- For given test point x, let $x_{(1)}, \dots, x_{(k)}$ be k nearest neighbors in training data, and let $y_{(1)}, \dots, y_{(k)}$ be corresponding labels.



Define

$$\hat{\eta}_n(x) = \frac{\sum_{i=1}^{k} w(x, x_{(i)})\, y_{(i)}}{\sum_{i=1}^{k} w(x, x_{(i)})}$$

where

$$w(x, x_{(i)}) = \left\| x - x_{(i)} \right\|^{-\delta}, \qquad \delta > 0$$

Interpolation: $\hat{\eta}_n(x) \to y_i$ as $x \to x_i$

Convergence rates for WiNN

[Belkin, H., Mitra, NeurIPS'18]

Theorem: Assume distribution of x' is uniform on some compact set satisfying regularity condition, and η is α-Holder smooth.

For appropriate setting of k, weighted & interpolated NN estimate $\hat{\eta}_n$ satisfies

$$\mathbb{E}\left[\left(\hat{\eta}_n(X) - \eta(X)\right)^2\right] \leq O\left(n^{-2\alpha/(2\alpha+d)}\right)$$

- Consistency + optimal rates of convergence for interpolating method.
- Follow-up work by Belkin, Rakhlin, Tsybakov '19: also for Nadaraya-Watson with compact & singular kernel.

Comparison to Hilbert kernel estimate

Weighted & interpolated NN

$$\hat{\eta}_n(x) = \frac{\sum_{i=1}^{k} w(x, x_{(i)})\, y_{(i)}}{\sum_{i=1}^{k} w(x, x_{(i)})}$$

$$w(x, x_{(i)}) = \|x - x_{(i)}\|^{-\delta}$$

Optimal non-parametric rates

Hilbert kernel (Devroye, Györfi, & Krzyżak, 1998)

$$\hat{\eta}_n(x) = \frac{\sum_{i=1}^{n} w(x, x_i)\, y_i}{\sum_{i=1}^{n} w(x, x_i)}$$

$$w(x, x_i) = \|x - x_i\|^{-\delta}$$

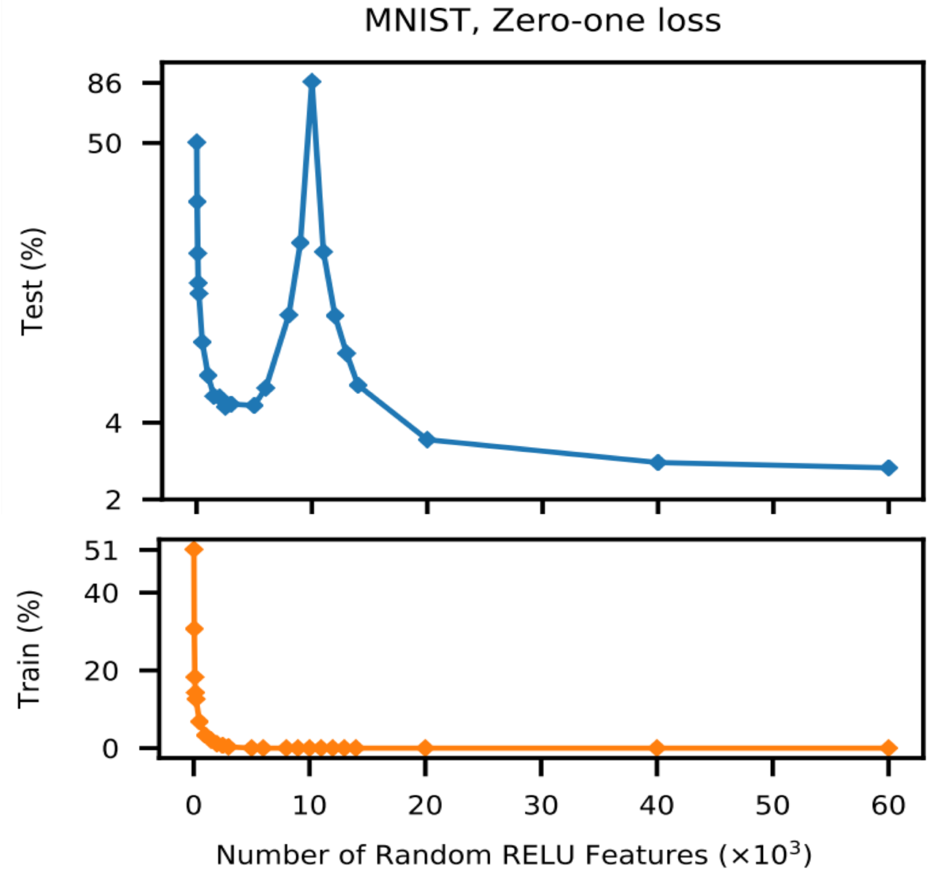
Consistent ($\delta = d$), but no rates

Localization is essential to get non-asymptotic rate.

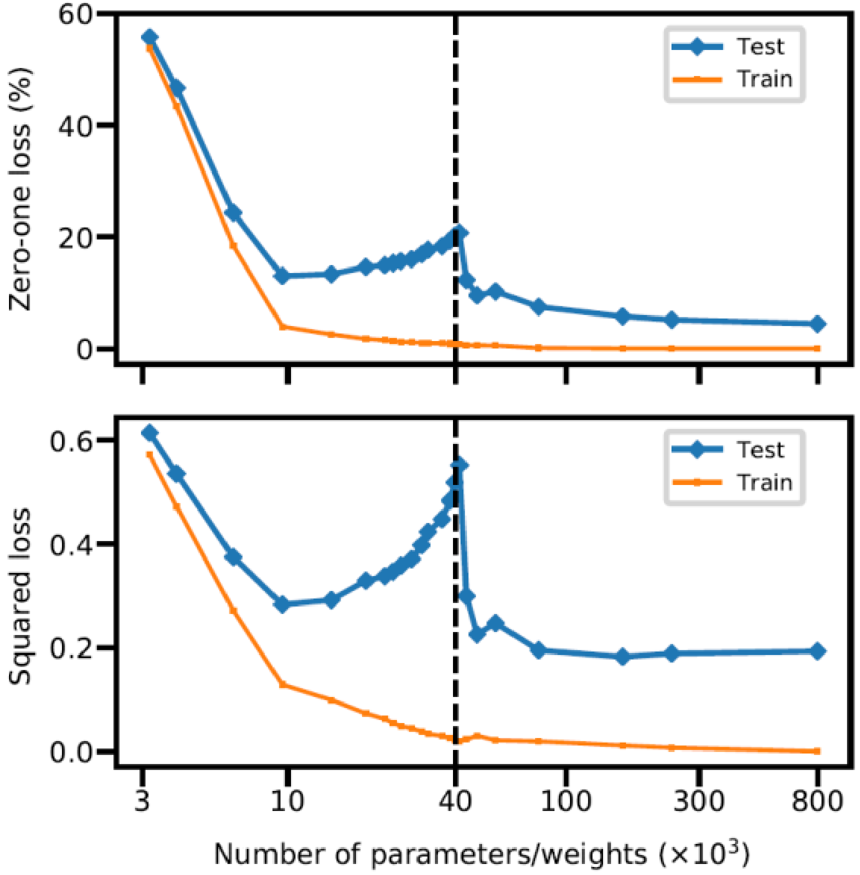
Interpolation via neural nets and linear models

Two layer fully-connected neural networks

[Belkin, H., Ma, Mandal, PNAS'19]



Random first layer

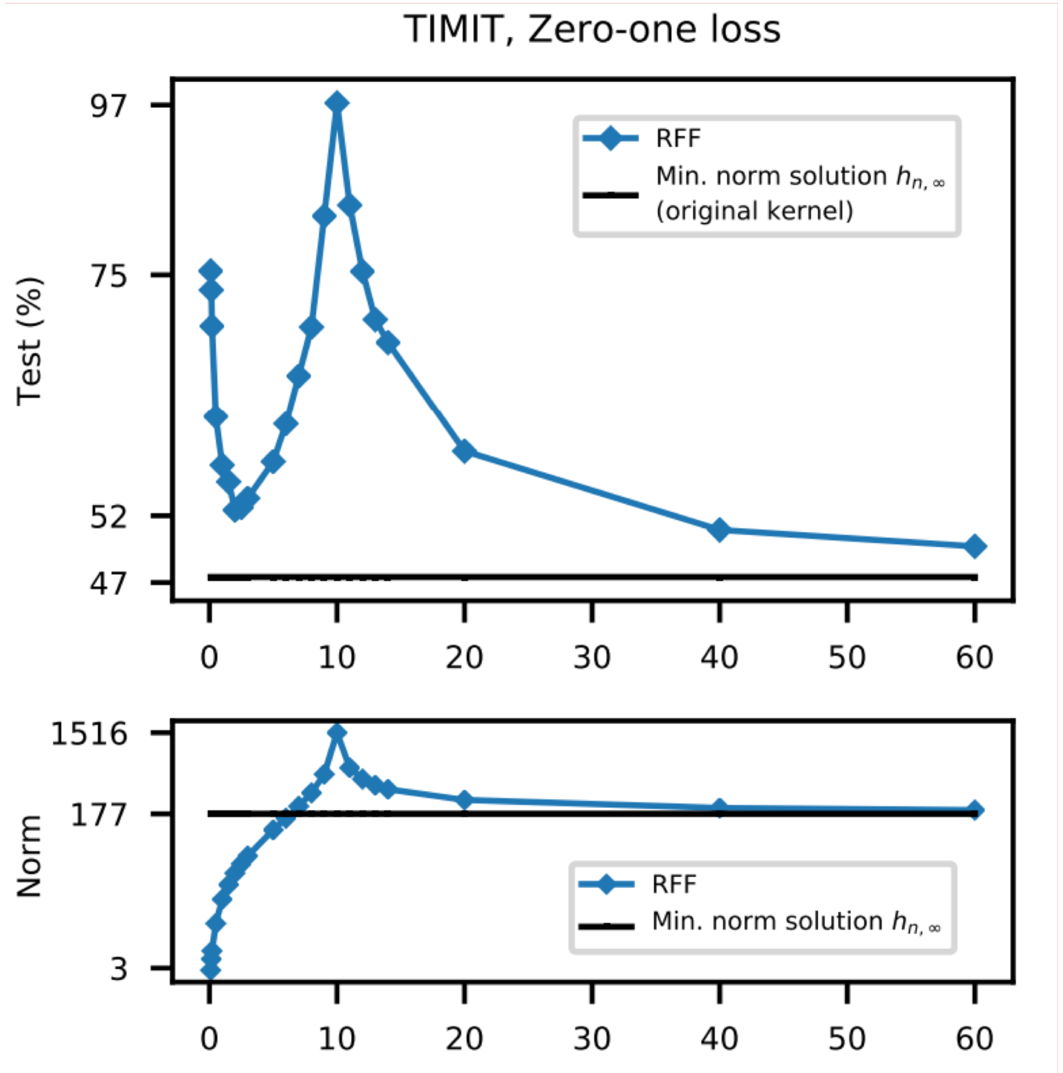


Trained first layer

Approximating a kernel machine?

- Effectiveness of interpolation depends on ability to align with the "right" inductive bias
- E.g., low RKHS norm
- "Occam's razor" approach:
  - Among all functions that fit the data, pick the one with smallest RKHS norm.

[Belkin, H., Ma, Mandal, PNAS'19]



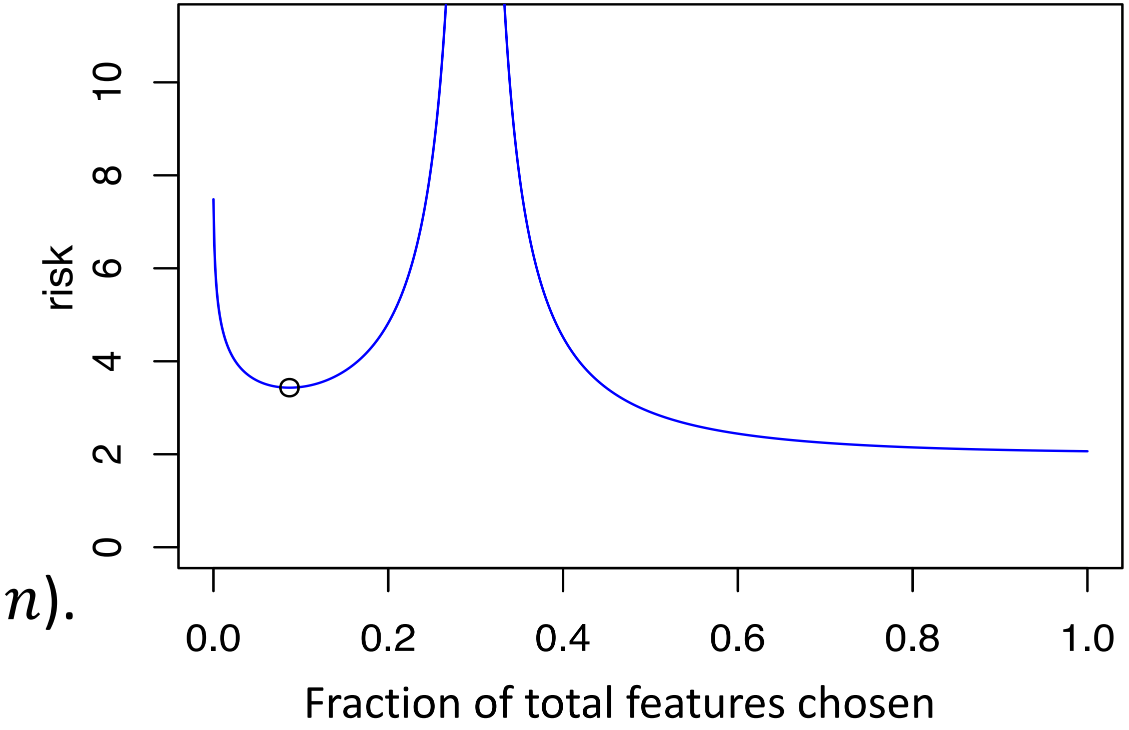
Linear regression with weak features

[Belkin, H., and Xu, '19+; Xu and H., NeurIPS'19]

Gaussian design linear model with D features
All features are "relevant" but equally weak

Only use p of the features ($1 \leq p \leq D$)
Least squares ($p \leq n$) or least norm ($p \geq n$) fit

Theorem: If eigenvalues decay slowly,
minimum is beyond point of interpolation ($p > n$).



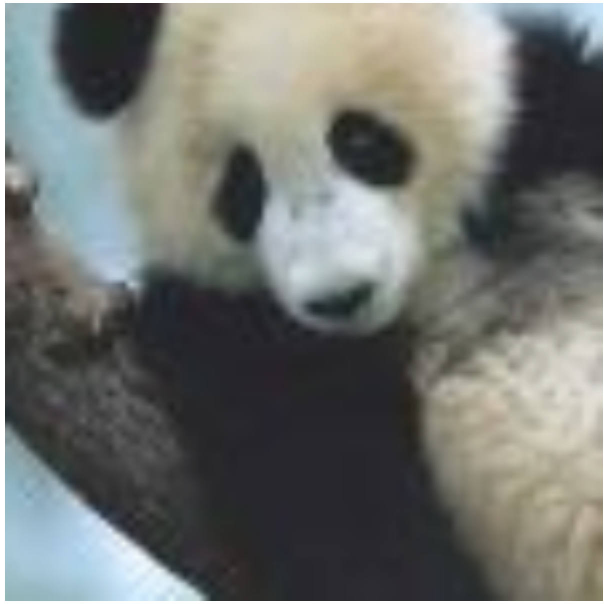
Concurrent work by Hastie, Montanari, Rosset, Tibshirani '19.

Other recent analyses of linear models: Muthukumar, Vodrahalli, Sahai, '19; Bartlett, Long, Lugosi, Tsigler, '19.

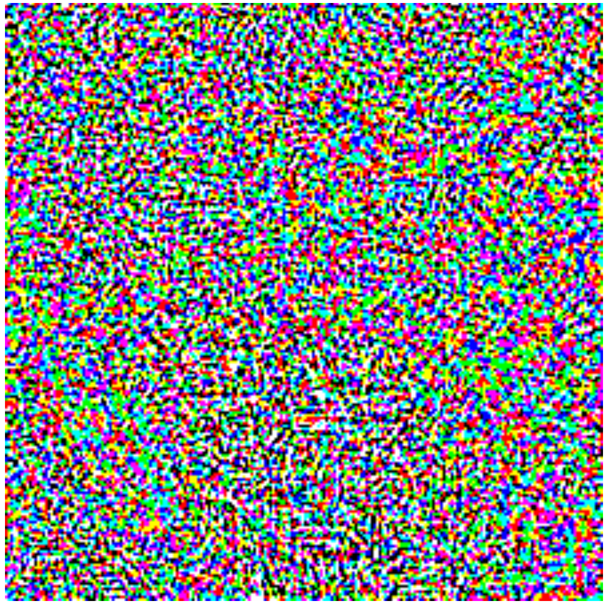
Follow-up work by Mei and Montanari '19 establishes similar results for non-linear random features models

Adversarial examples

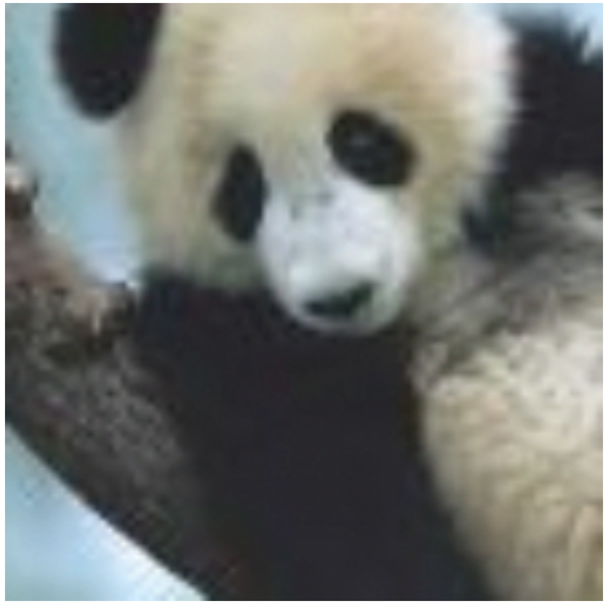
Adversarial examples (Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, '14; Goodfellow, Shlens, Szegedy, '15)



$+ \quad \epsilon$



$=$



$\hat{f}(x) = \text{"panda"}$

$\hat{f}(\tilde{x}) = \text{"gibbon"}$

Inevitability of adversarial examples

[Belkin, H., Mitra, NeurIPS'18]

- Are adversarial examples inevitable when interpolating noisy data?
  - Assume compact domain Ω for x's.
  - "Adversarial examples" for interpolating classifier $\hat{f}_n$:
    $$A_n \coloneqq \{\, x \in \Omega : \hat{f}_n(x) \neq f^*(x) \,\}$$
  - **Proposition**: If as η is always bounded away from 0 and 1 (i.e., labels are not deterministic), then A_n is asymptotically dense in Ω.
  - [For any $\epsilon > 0$ and $\delta \in (0,1)$, for n sufficiently large, every $x \in \Omega$ is within distance ϵ of A_n with probability at least $1 - \delta$.]

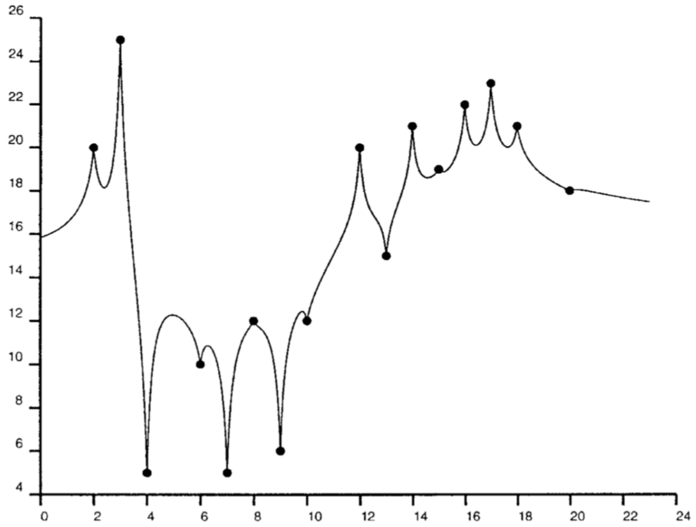


FIG. 2. The Hilbert kernel regression estimate with $a = 1$.

Conclusions/open problems

1. Interpolation **is** compatible with good statistical properties.
2. They work by relying (exclusively!) on **inductive bias**: e.g.,
    1. Smoothness from local averaging in high-dimensions.
    2. Low function space norm.
3. But "adversarial examples" may be inevitable.

Open problems:

- Characterize inductive biases of other common learning algorithms.
- Behavior for deep neural networks?
- Benefits of interpolation?

Acknowledgements

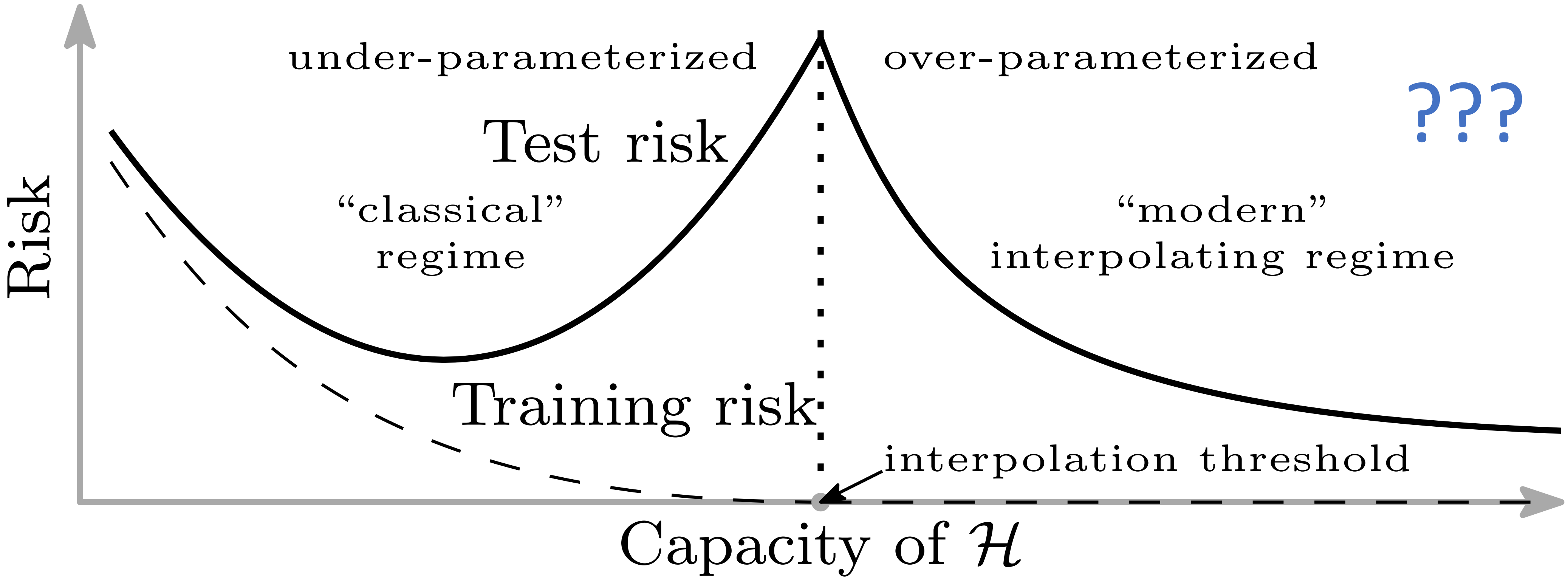
- **Collaborators**:
  Misha Belkin, Siyuan Ma, Soumik Mandal, Partha Mitra, Ji Xu
- National Science Foundation
- Sloan Foundation
- Simons Institute for the Theory of Computing

arXiv references:
1806.05161
1812.11118
1903.07571
1906.01139

Thank you!

"Double descent" risk curve

[Belkin, H., Ma, Mandal, PNAS'19]



???