STAT 391

5/23/23

Lecture 17

it's not too late!

Participation

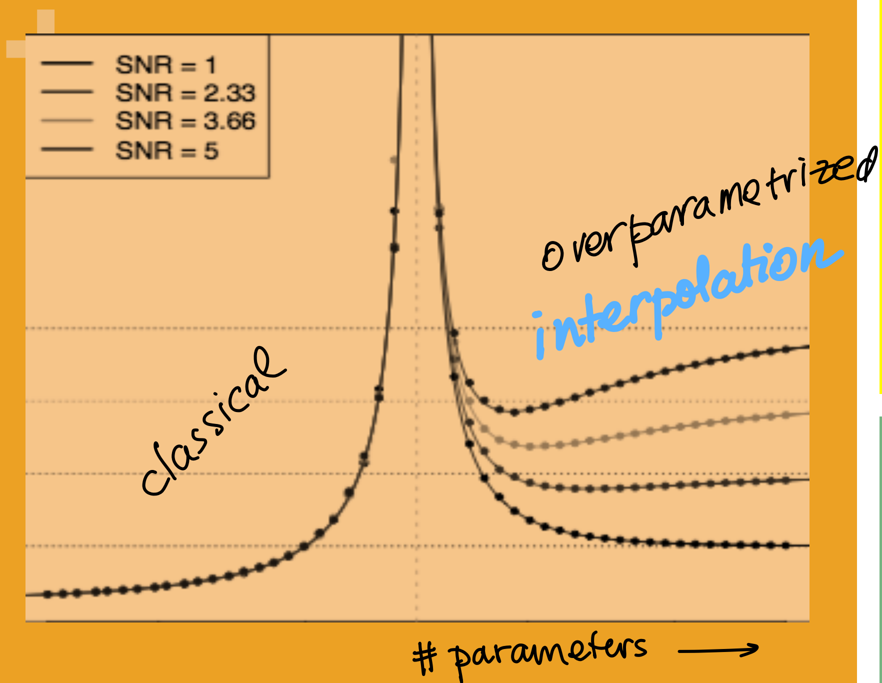
Regression - double descent

Logistic regression

HW 6 t.b. posted
due Friday 6/1
optional

All HW t.b. graded on Friday 6/1

Exam week
Exam reviews S.P. / MMP
t.b scheduled



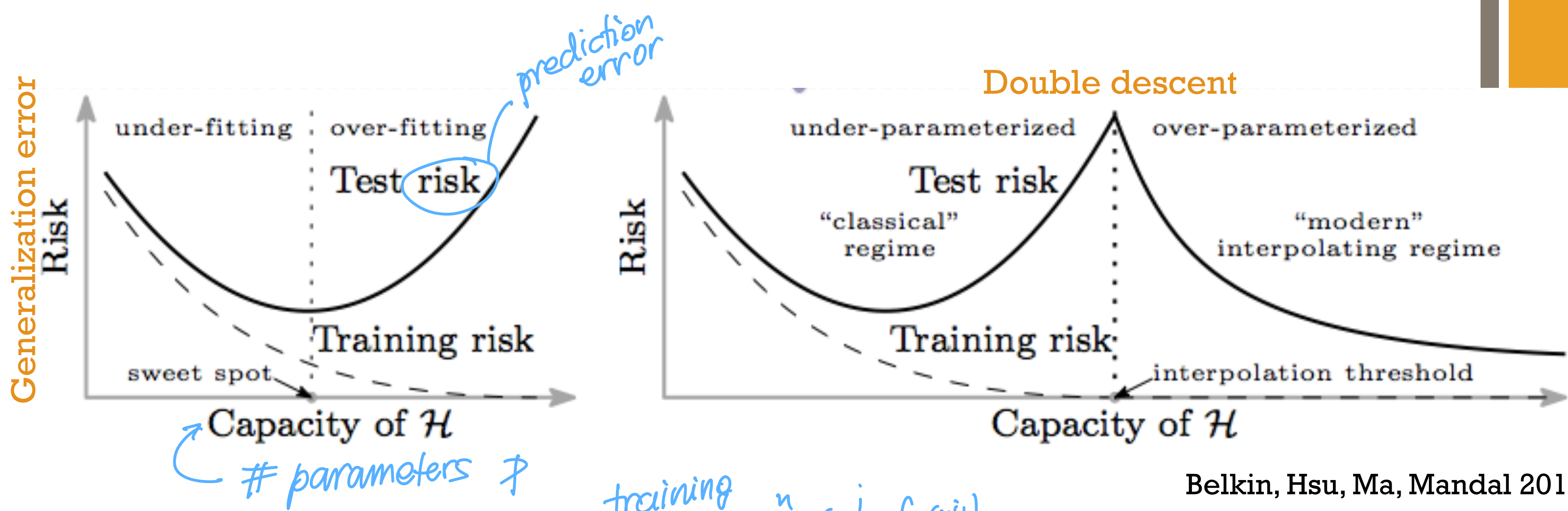
Double Descent

STAT 535+LPL2019

Beyond the Bias-Variance trade-off

Marina Meila
University of Washington

What is observed



Belkin, Hsu, Ma, Mandal 2018

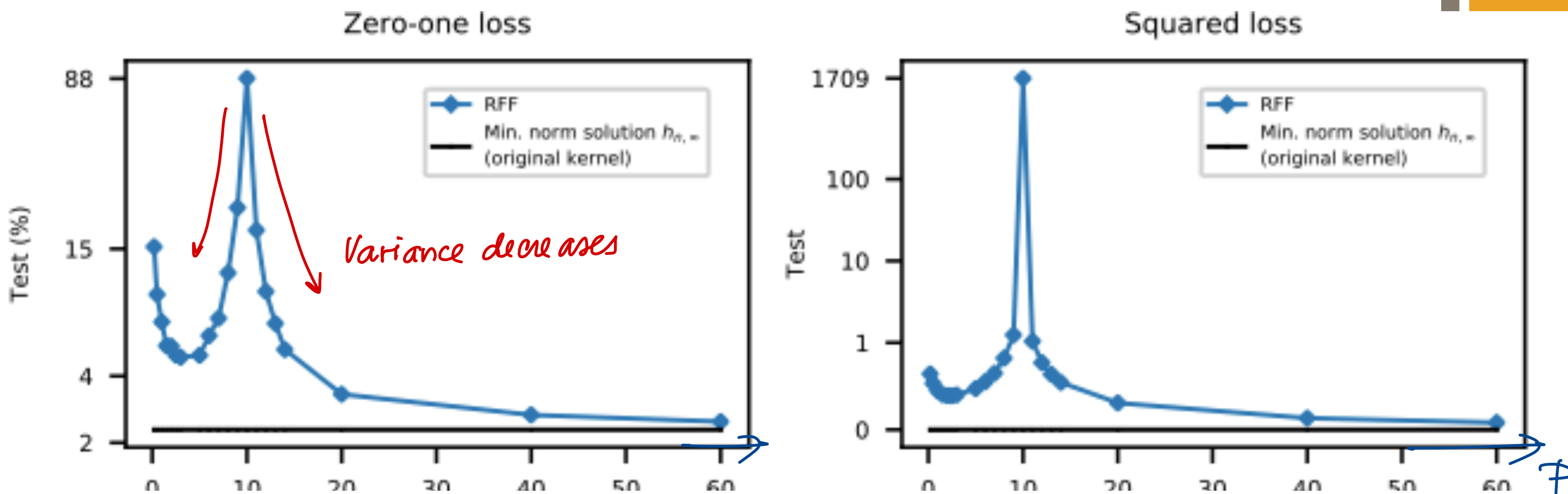
training err $= \frac{1}{n}\sum_{i=1}^{n}(y^i - f_\theta(x^i))$ $\uparrow \mathcal{F}_p$ $\#\theta = p$

test err $= \frac{1}{n_V}\sum_{j=1}^{n_V}(- '' -)$ $\{x^j, y^j \in \mathcal{D}^{validation}\}$

"Squared Loss"

- Classical regime p < N
- Modern/Deep Learning/High dimensional regime N > n
  - Think N fixed, p increases, gamma=p/N
  - Training error = 0 (interpolation)
  - Test error decreases with p (or gamma)

What is observed



p

Belkin, Hsu, Ma, Mandal 2018

- Double descent curves for the generalization error
  - Random Fourier Features (RFF) SVM
  - ReLU 2 layer networks (with random first layer weights)
  - Random Forests, l2-Adaboost
  - Linear regression

observed for many classes of predictors

- With and without noise

Ex: 1) x = image $y = 1$ iff contains car $\{(x^{1:n}, y^{1:n})\}$ training set

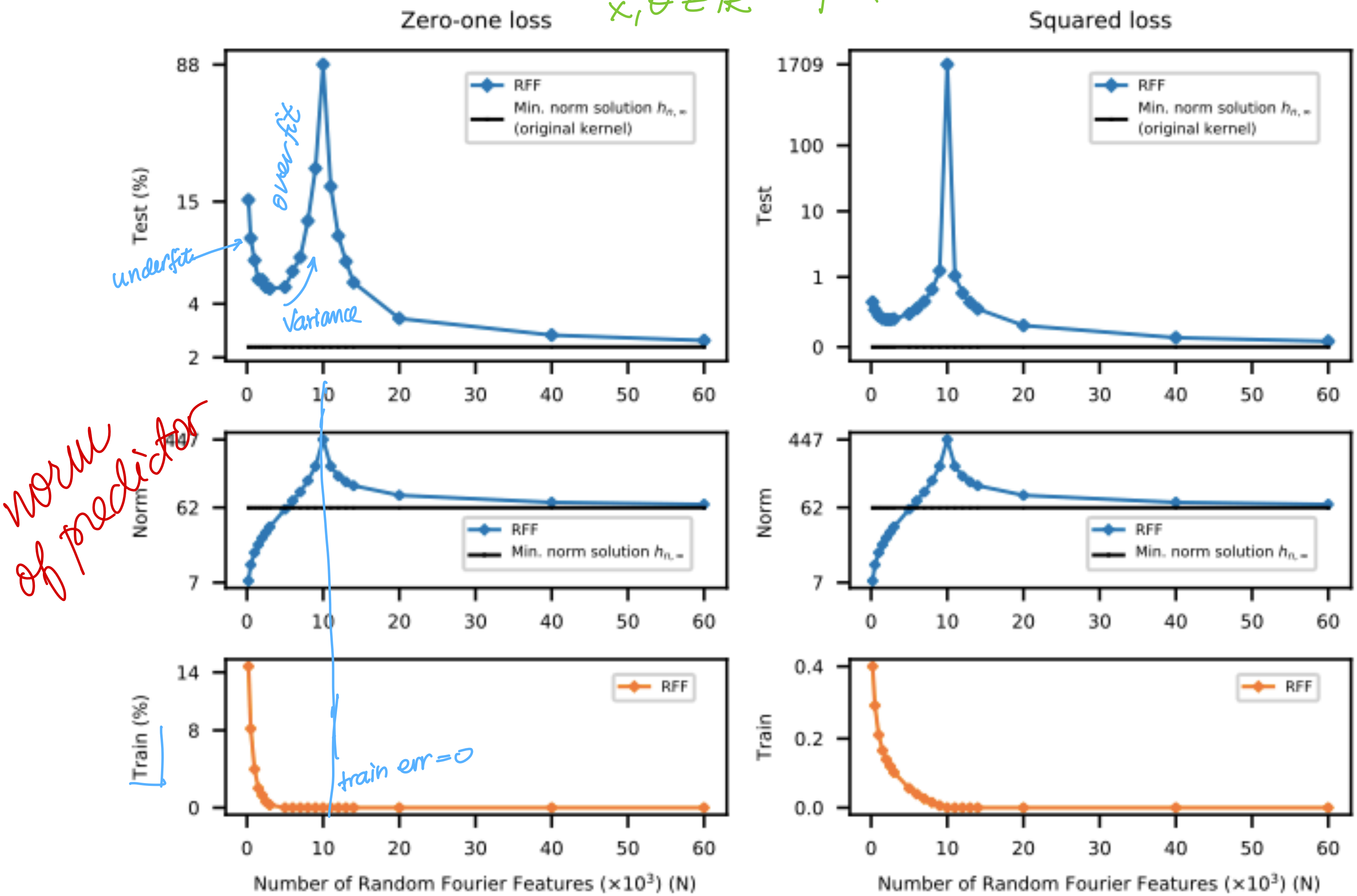
$f_{\theta_p} \in \mathcal{F}_p$ neural net p = # weights or # layers in neural net

Two cases of increasing p

+ "NN"

RFF $= \mathcal{F} = \{f_\theta\}$ 2) $f_\theta = \theta^T x$

$x, \theta \in \mathbb{R}^p$ $p \uparrow \Leftrightarrow$ resolution of x increases



$\|v\|^2 = \sum_j v_j^2$

$\|$ neural network $\|^2 \equiv \sum_{j=1}^{p} w_j^2$

$w_{1:p}$ = weight of neural network

1. Double descent $\exists$ $\Rightarrow$ Interpolation $\Longleftrightarrow$ $f_\theta(x^i) = y^i$ for $i = 1:n$

2. in Interpolation regime

- $\|\hat{\theta}\| \to$ to min > 0

hypothesis $\|\hat{\theta}\|$ small $\Rightarrow$ Var $\hat{\theta}$ small

- $\exists$ infinite number θ so that $f_\theta(x^{1:n}) = y^{1:n}$

$\|\theta\|$ small $\Longleftrightarrow$ f_θ smooth

Intuition:

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix} \qquad w^0 = \left[\frac{1}{p} \ \cdots \ \frac{1}{p}\right]^T \text{ constant vector} = \text{smoothest!!}$$

$$\sum_{j=1}^{p} |w_j| = 1 \qquad \|w^0\|^2 = p \cdot \frac{1}{p^2} = \frac{1}{p} \leq \|w\|^2 \qquad w \in \mathbb{R}^p, \ \sum |w_j| = 1$$

$\min \|w\|^2$ $\Rightarrow$

$\sum |w_j| = 1$

$\frac{1}{p} \searrow$ for $p \nearrow$ $\Rightarrow$ smoother for $p \nearrow$

3. $\|\theta\|$ small $\Rightarrow$ f_θ smooth $\Rightarrow$ low variance !!

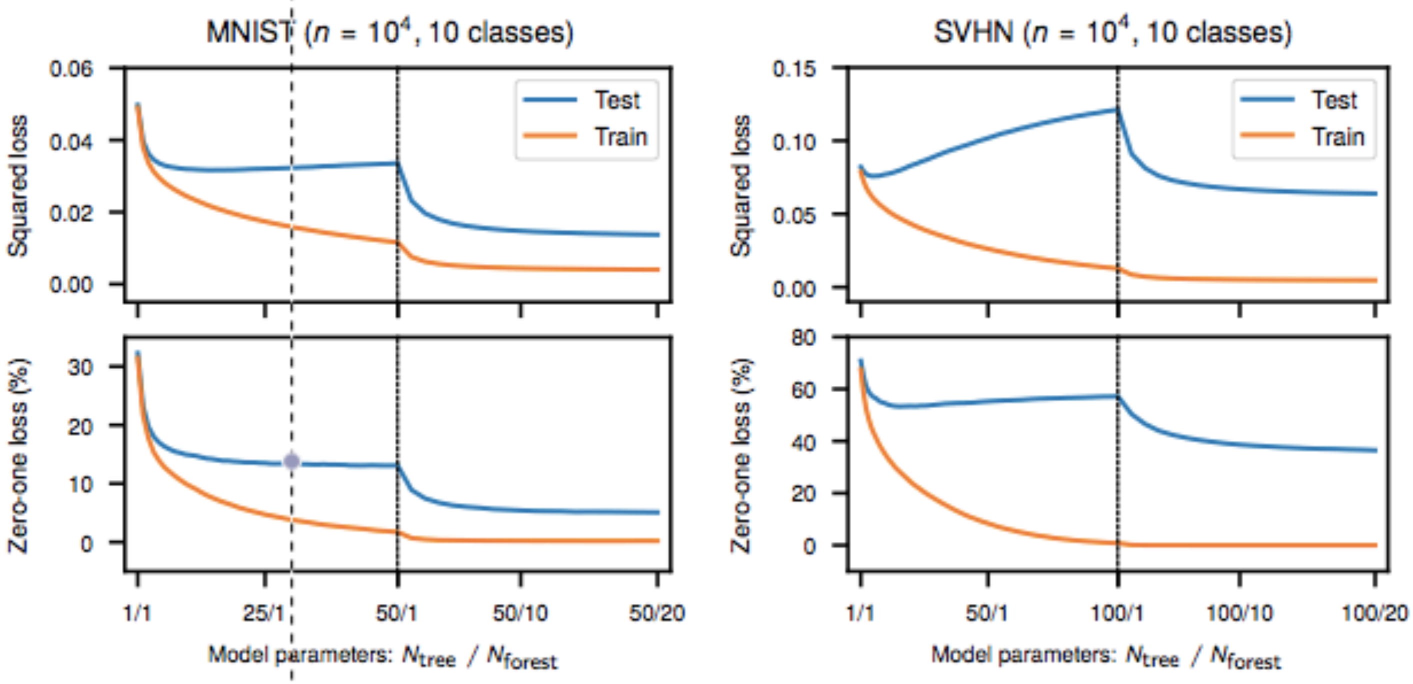
$p_1 < p_2 < p_3$

$\mathcal{F}_{p_1} \subset \mathcal{F}_{p_2} \subset \mathcal{F}_{p_3} \ldots$

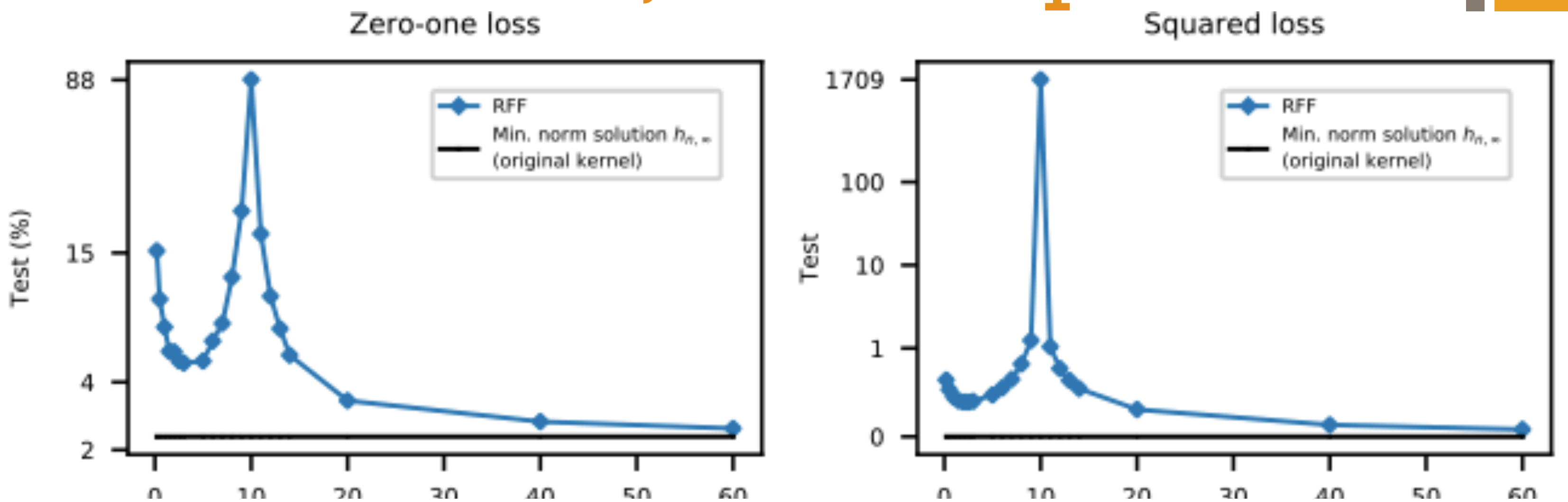
more models to choose from !!

4. If training algorithm can choose smoothest $f_\theta \in \mathcal{F}_p$ $\Rightarrow$ $f_{\theta_{p_2}}$ smoother than $f_{\theta_{p_1}}$

Boosted decision trees



Double descent, the case p > N



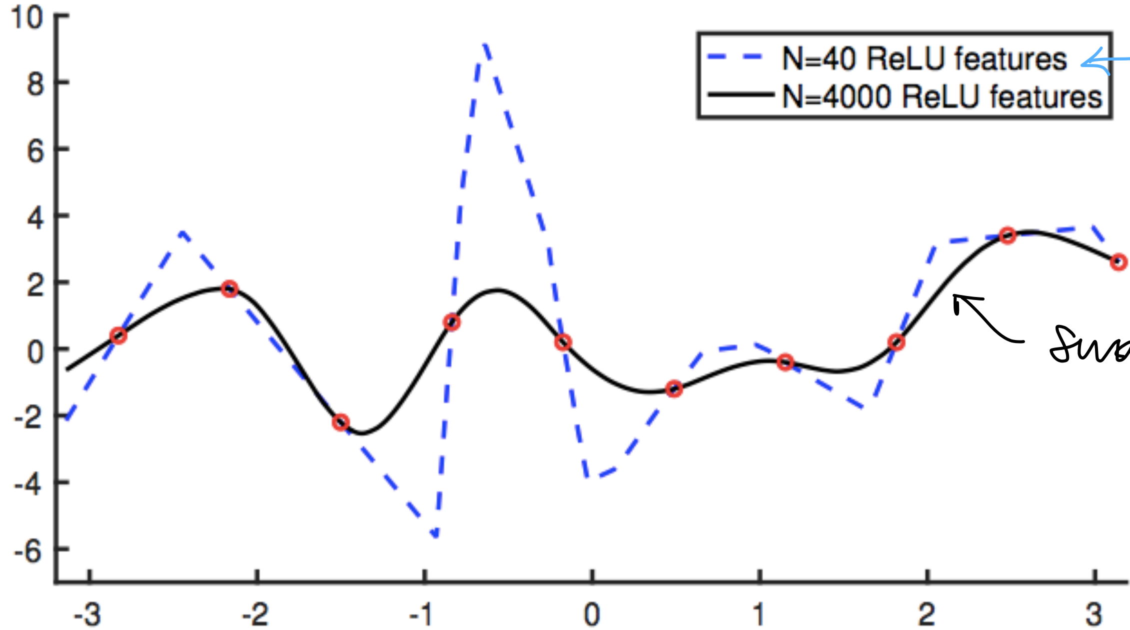
Belkin, Hsu, Ma, Mandal 2018

p →

parameters

- Model y = <phi(x), beta >
- Large N (cover a compact data domain)
- Features random
- Min-norm solution beta*

Main intuition [Belkin et al.]



Neural Net

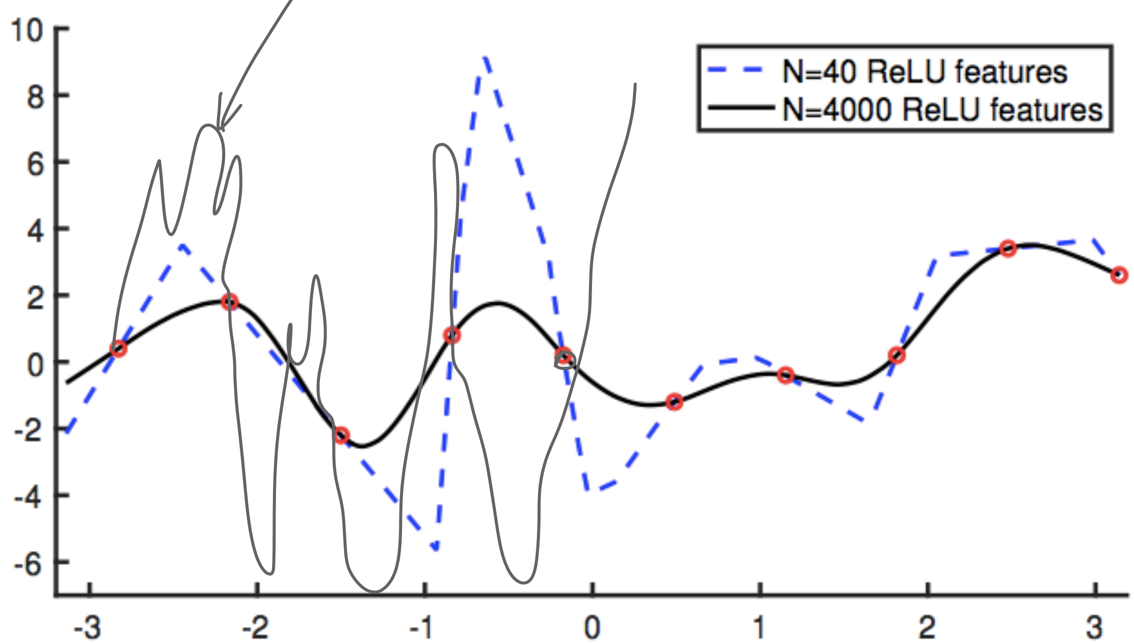
n = 10

smoother for more parameters

- The target function h^* is (mostly) smooth
  - i.e. $||h^*||_{RKHS}$ is small
- $p > N$, no noise, hence h_p interpolates data
- Train to minimize $||h_p||$ subject to 0 training error
- Then $||h_p||$ will decrease with p!

Main intuition [Belkin et al.]

∃ many f's that interpolate!



- The target function h^* is (mostly) smooth
  - i.e. $||h^*||_{RKHS}$ is small
- $p > N$, no noise, hence h_p interpolates data
- Train to minimize $||h_p||$ subject to 0 training error
- Then $||h_p||$ will decrease with p!

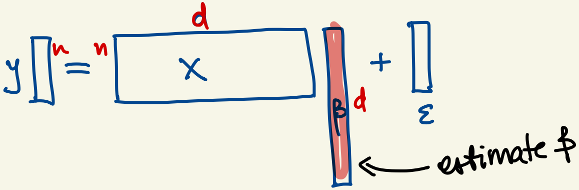
Surprising fact:
gradient descent finds smooth(est) f_θ !!

Linear regression

- True model $\quad y = (\beta^*)^T x + \varepsilon \iff y^i = (\beta^*)^T x^i + \varepsilon^i \quad i = 1:n$

$$\mathcal{D} = \{(x^i, y^i)\}_{i=1:n}$$

- $n < d \qquad x, \beta, \beta^* \in \mathbb{R}^d$



- $y = X\beta$ has ∞ solutions
- $\cancel{X^T X} \in \mathbb{R}^{d \times d}$ with rank $n < d$

can't solve this way!

if $n > d \Rightarrow \beta^{ML} = (X^T X)^{-1} X^T y$

- Do gradient descent instead

$$\min -\ell(\beta) = \min_{\beta} \sum_{i=1}^{n} (y^i - \beta^T x^i)^2$$

Exercise

$$-\frac{\partial \ell}{\partial \beta} = 2\left(X^T X \beta - X^T y\right)$$

$$\frac{\partial \ell}{\partial \beta} = \left[\frac{\partial \ell}{\partial \beta_j}\right]_{j=1:d}$$

Gradient descent Alg

Init $\quad \beta \leftarrow 0$

For $t = 1, 2, \ldots$ until convergence

$\quad \beta \leftarrow \beta + \eta \dfrac{\partial \ell}{\partial \beta}$

step size

Out $\quad \hat{\beta} = \beta$ at convergence

Properties of $\hat{\beta}$

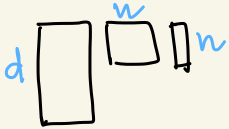
Ex (Linear algebra, not easy)
Prove Prop 1, 2, 3

XX^T, X^TX same singular value

Prop 1 $y = X\hat{\beta}$ interpolation

Prop 2 $\hat{\beta} = X^T K^{-1} y$ $K = XX^T$ Gram matrix

$n \times n$, full rank



Prop 3 $\|\hat{\beta}\| = \underset{y = X\beta}{\operatorname{argmin}} \|\beta\|$ min norm interpolating solution

Logistic Regression

Used for: Classification

$x \in \mathbb{R}^d$

$y \in \{0,1\}$ "Label"

$\mathcal{D} = \{(x^i, y^i)_{i=1:n}\}$

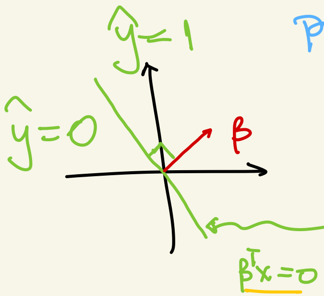
meaning   parametrization

Model $$f(x) = \ln \underbrace{\underbrace{\frac{P_{Y|X}(y=1|x)}{P_{Y|X}(y=0|x)}}_{\text{odds } \in (0,\infty)}}_{\text{log odds } \in (-\infty,\infty)} = \underline{\beta^T x} \qquad \beta \in \mathbb{R}^d$$

Classification with $f = \beta^T x$ (β known)

$$\hat{y} = \frac{\text{sign } f(x) + 1}{2} \iff f(x) > 0 \iff P_{Y|X}(y=1|x) > P_{Y|X}(y=0|x)$$

Probabilistic classifier



linear classifier $\iff$ decision boundary linear

- $P_{y|x}[y=1|x] = p$

$$\boxed{\begin{matrix} f \\ \shortparallel \\ \beta^T x \end{matrix}} = \ln \frac{p}{1-p} = \ln \frac{P_{y|x}(y=1|x)}{\underbrace{P_{y|x}(y=0|x)}} \Rightarrow e^f = \frac{p}{1-p} \Rightarrow \boxed{\begin{matrix} p = \dfrac{e^f}{1+e^f} \in (0,1) \\ 1-p = \dfrac{1}{1+e^f} \end{matrix}}$$

$p + (1-p) = 1$

- Estimating β by Max likelihood

confidence of classification $\quad P_{y|x}[y|x] = \dfrac{e^{yf}}{1+e^f}$

likelihood

$$L(\beta) = P_r[y^{1:n} | x^{1:n}, \beta]$$

$$= \prod_{i=1}^{n} P_{y|x}[y^i | x^i, \beta] = \prod_{i=1}^{n} \frac{e^{y^i \beta^T x^i}}{1+e^{\beta^T x^i}}$$

p for $y^i=1$

$1-p$ for $y^i=0$

log likelihood

$$\ell(\beta) = \sum_{i=1}^{n} \left[y^i \beta^T x^i - \ln\left(1+e^{\beta^T x^i}\right) \right]$$

maximize by Gradient Ascent

$$\ell(\beta) = \sum_{i=1}^{n} \left[y^i \underbrace{\beta^T x^i}_{f} - \ln\left(1 + e^{\beta^T x^i}\right) \right]$$

Gradient $\frac{\partial \ell}{\partial \beta} \in \mathbb{R}^d$

$$\frac{\partial \ell}{\partial \beta_j} = \frac{\partial \ell}{\partial f} \cdot \frac{\partial f}{\partial \beta_j}$$

$$\frac{\partial \ell}{\partial \beta} = \left[\frac{\partial \ell}{\partial \beta_j} \right]_{j=1:d} = \frac{\partial \ell}{\partial f} \left[\frac{\partial f}{\partial \beta_j} \right]_{j=1:p}$$

$$\frac{\partial f}{\partial \beta} = \frac{\partial}{\partial \beta}(\beta^T x) = x$$ ← Exercise prove this