

CSE 446

# Regularization & Sparsity

---

Natasha Jaques



# Bias-Variance Tradeoff

$$\mathbb{E}_{Y|X}[\mathbb{E}_{\mathcal{D}}[(Y - \hat{f}_{\mathcal{D}}(x))^2] | X = x] = \mathbb{E}_{Y|X}[(Y - \eta(x))^2 | X = x]$$

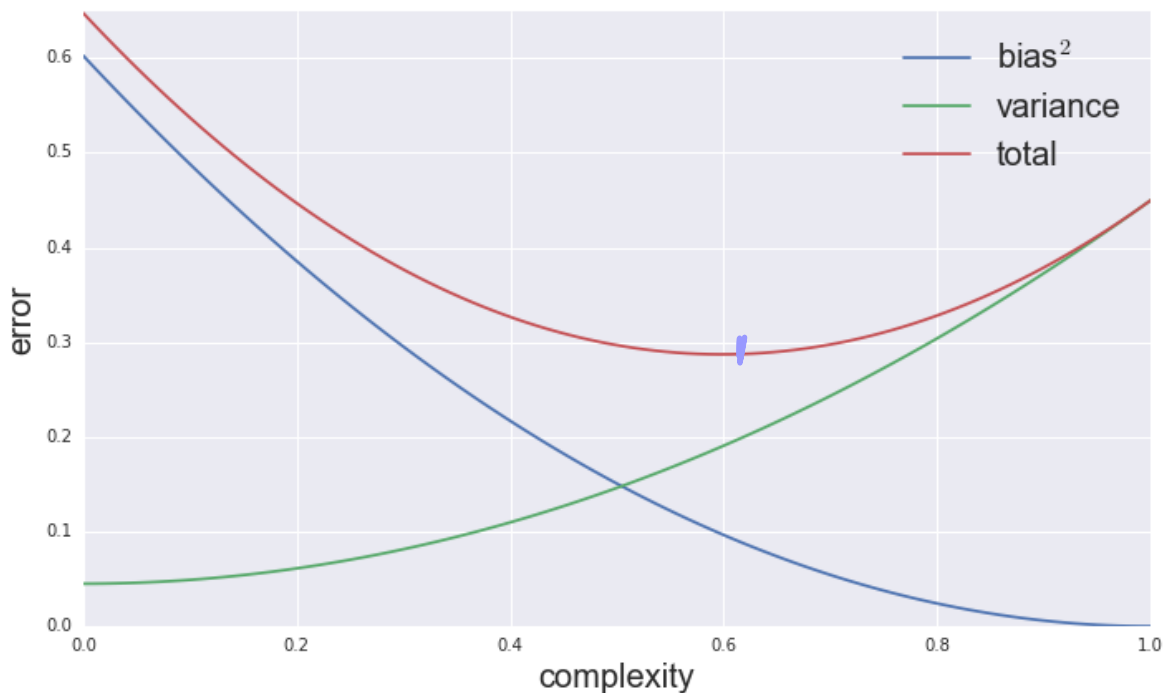
True generalization error

irreducible error

$$+(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2 + \mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]$$

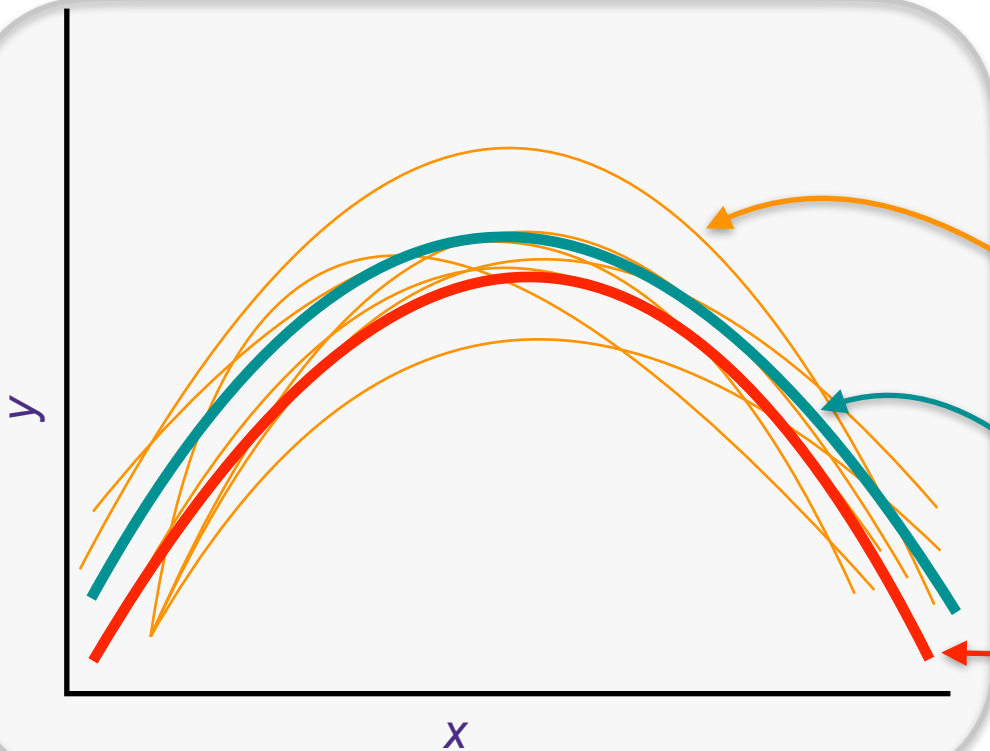
biased squared

variance



# Bias-Variance Tradeoff

$$\underbrace{\mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{learning error}} = \underbrace{(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2}_{\text{biased squared}} + \underbrace{\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{variance}}$$



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

$$\mathbb{E}_{\mathcal{D}}[\hat{f}(x)]$$

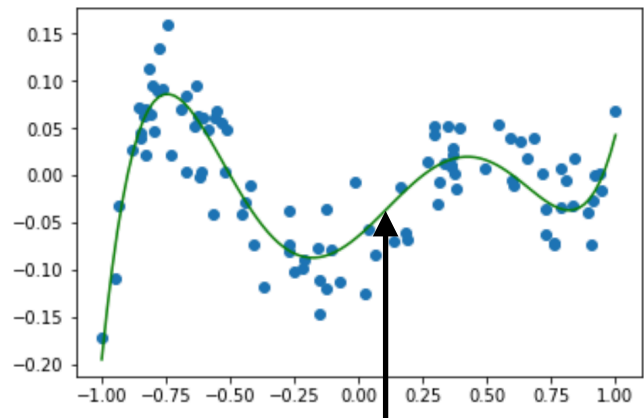
$$\eta(x) = \mathbb{E}_{Y|X}[Y|X = x]$$

$p_{X,Y}$

# Test error vs. model complexity

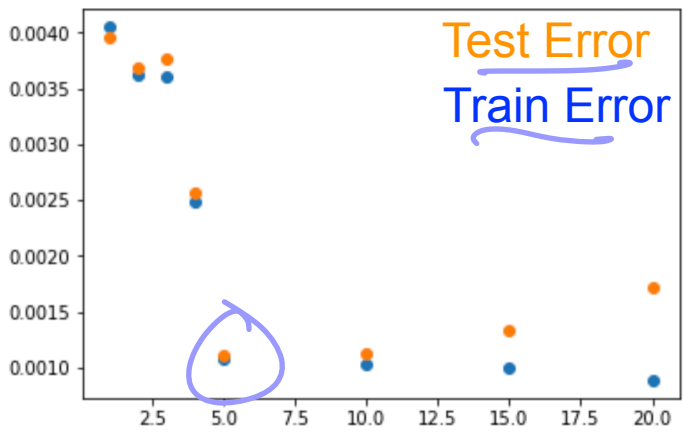
Simple model: *high bias*  
 Model complexity is below the complexity of  $\eta(x)$

Complex model: Fits noise in train data, diverging from  $\eta(x)$   
*green*

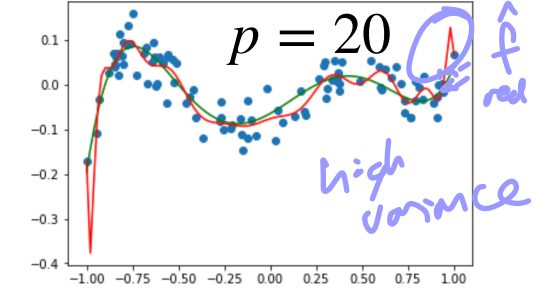
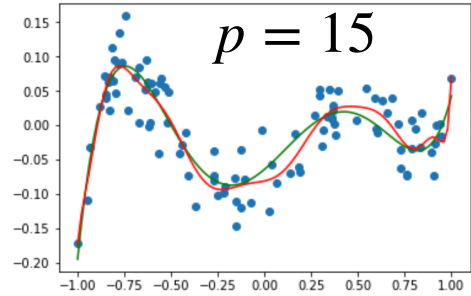
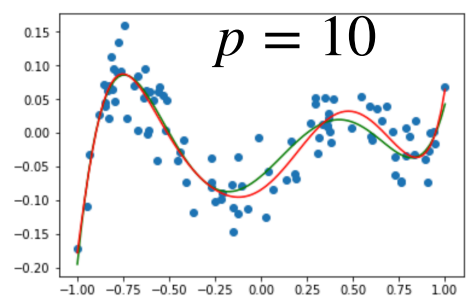
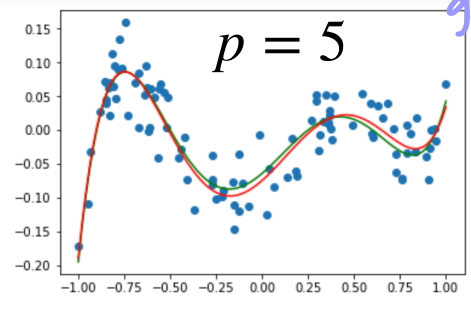
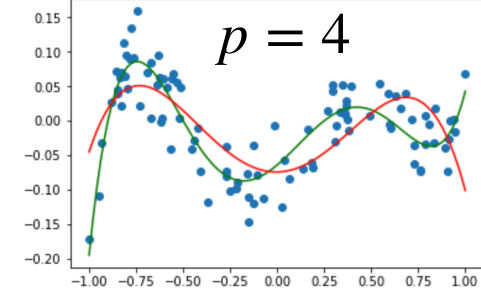
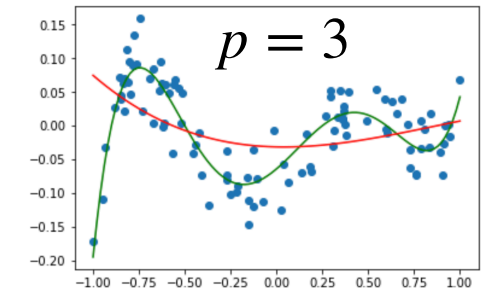
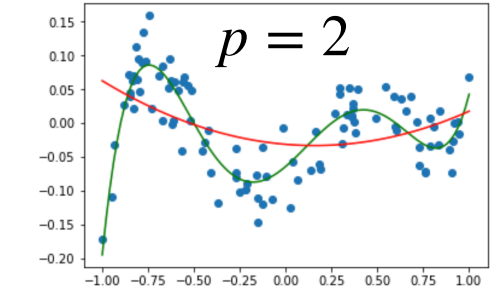
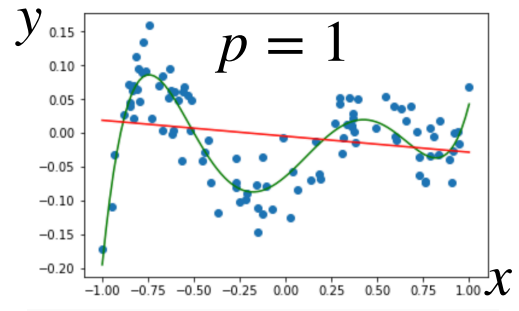


Optimal predictor  $\eta(x)$  is degree-5 polynomial

Error

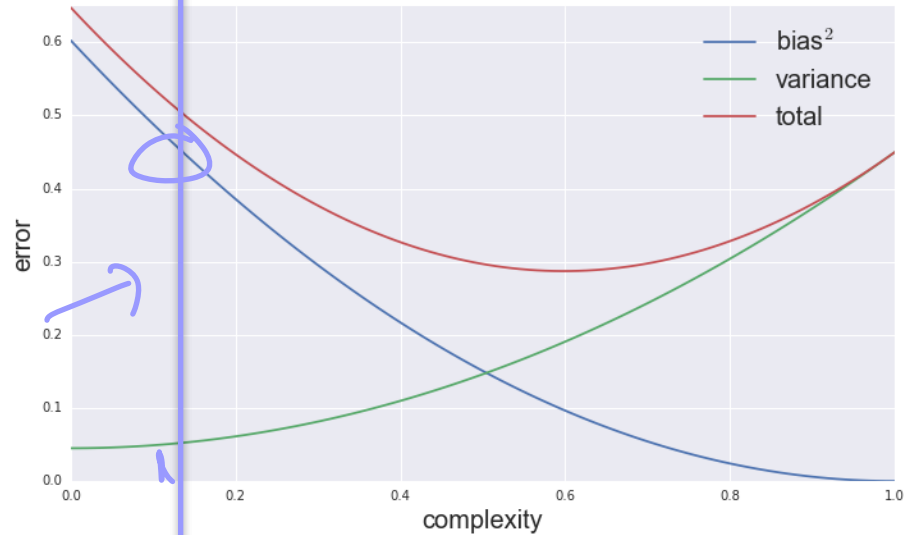


degree  $p$  of the polynomial regression



# Recap: Bias-variance tradeoff with simple model

(Conceptual) bias variance tradeoff



- When model **complexity is low** (lower than the optimal predictor  $\eta(x)$ )

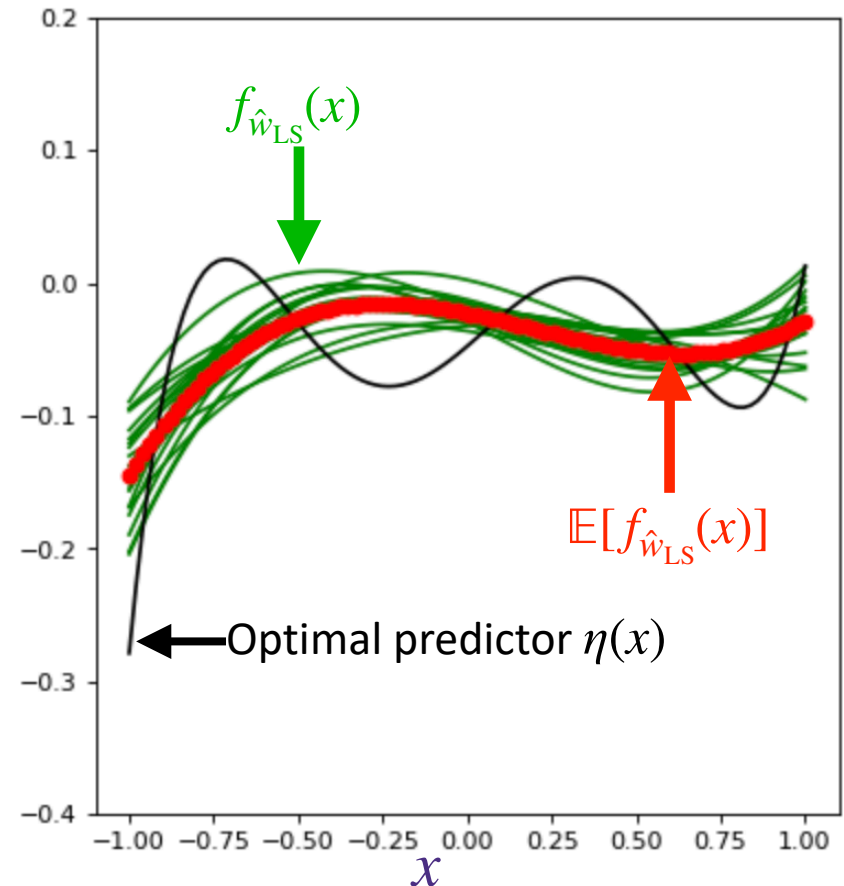
- Bias<sup>2</sup> of our predictor,  $(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2$ , is large

- Variance of our predictor,  $\mathbb{E}_{\mathcal{D}} [ (\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2 ]$ , is small

- If we have more samples (larger  $n$ ), then

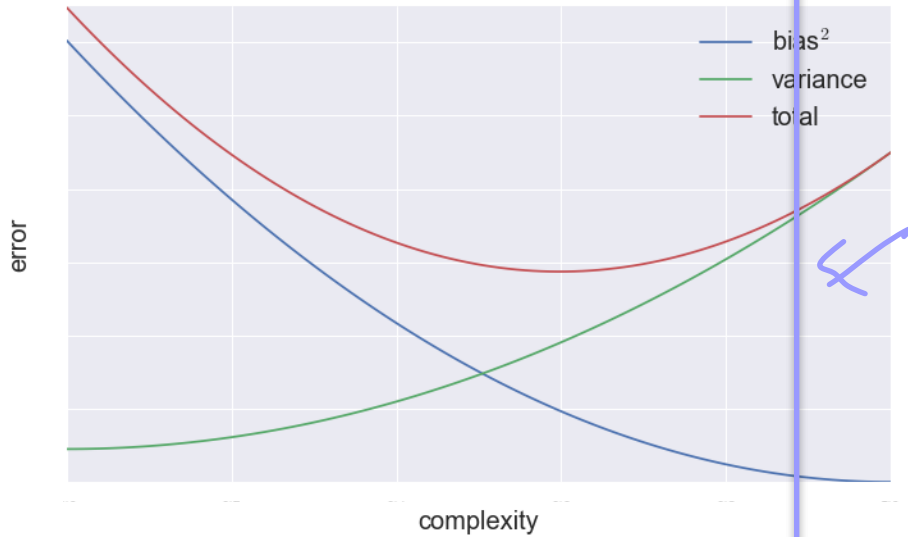
- What happens to bias?  $\rightarrow$  remains the same
- What happens to variance?  $\rightarrow$  goes down
- What happens to overall test error?  $\rightarrow$  same-ish

With degree-3 polynomials, we underfit



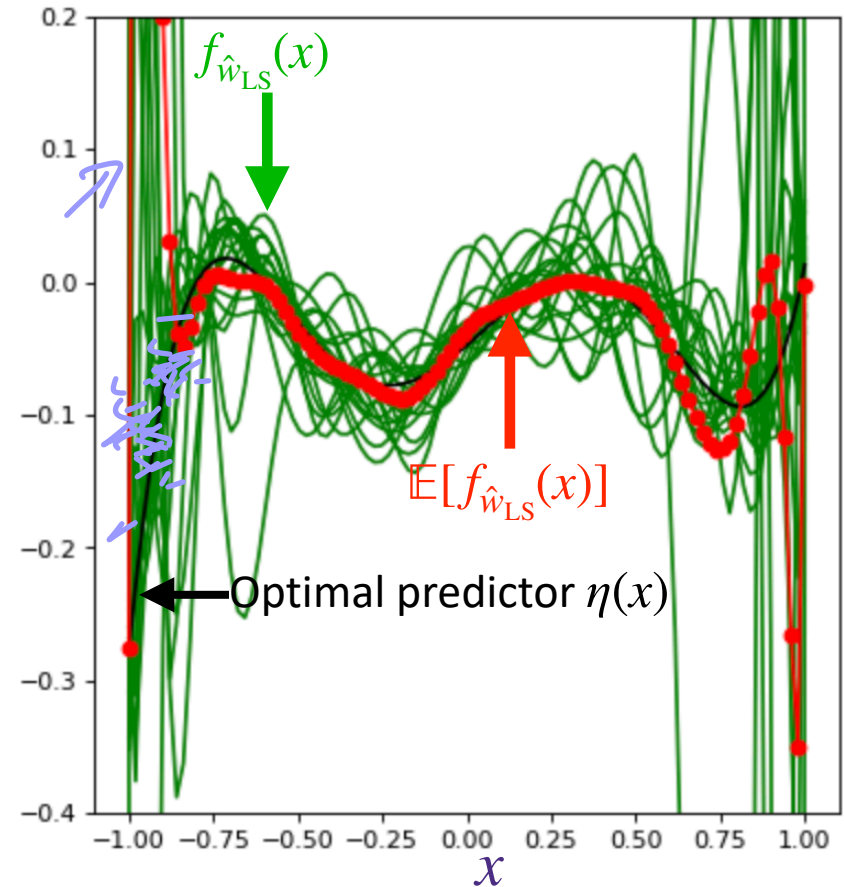
# Recap: Bias-variance tradeoff with complex model

(Conceptual) bias variance tradeoff



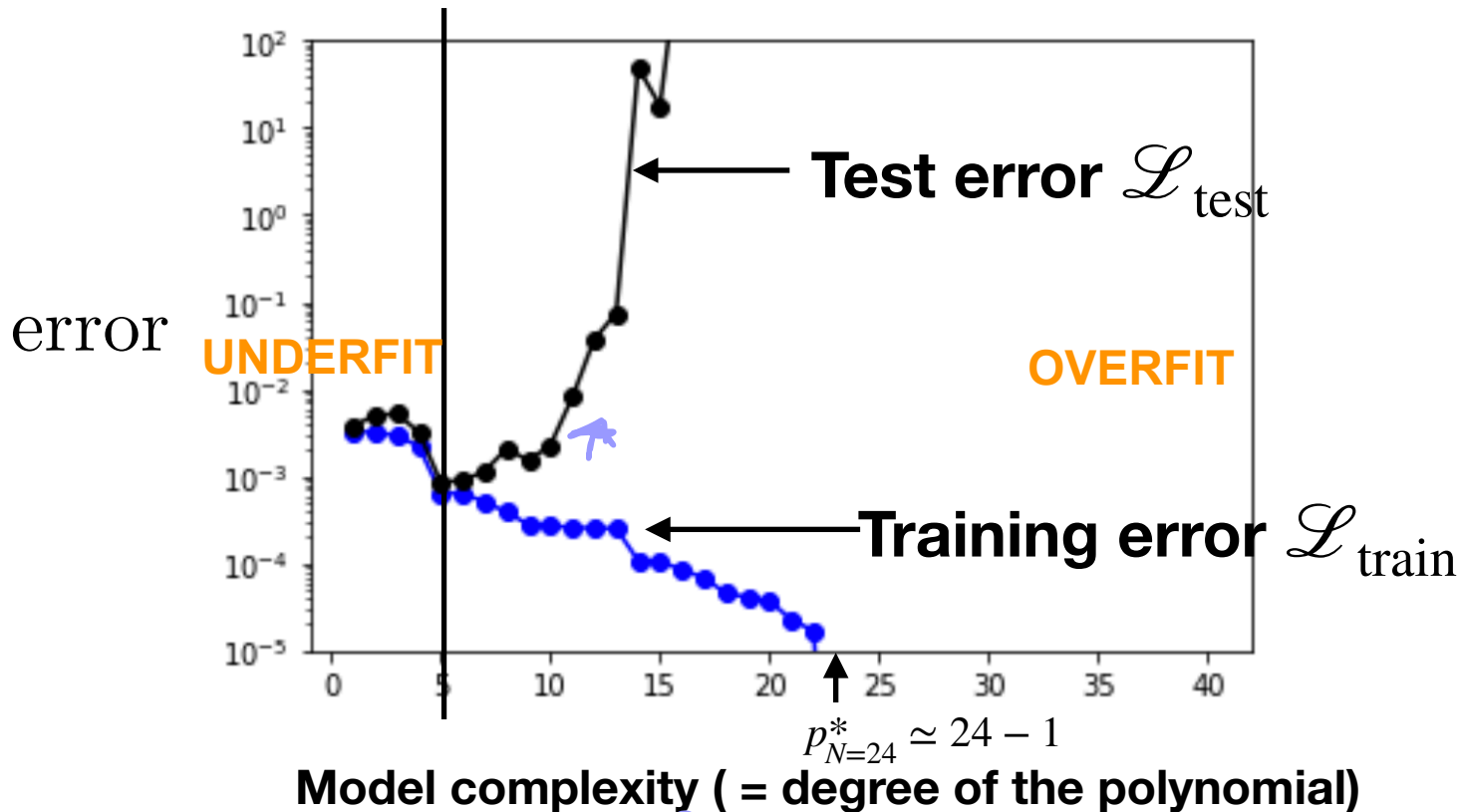
- When model complexity is high (higher than the optimal predictor  $\eta(x)$ )
  - Bias of our predictor,  $(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2$ , is small
  - Variance of our predictor,  $\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]$ , is large
- **If we have more samples (larger  $n$ ), then**
  - What happens to bias? *stays the same*
  - What happens to variance? *goes down*
  - What happens to overall test error? *goes down*

With degree-20 polynomials, we overfit



# Optimal model complexity depends on dataset size

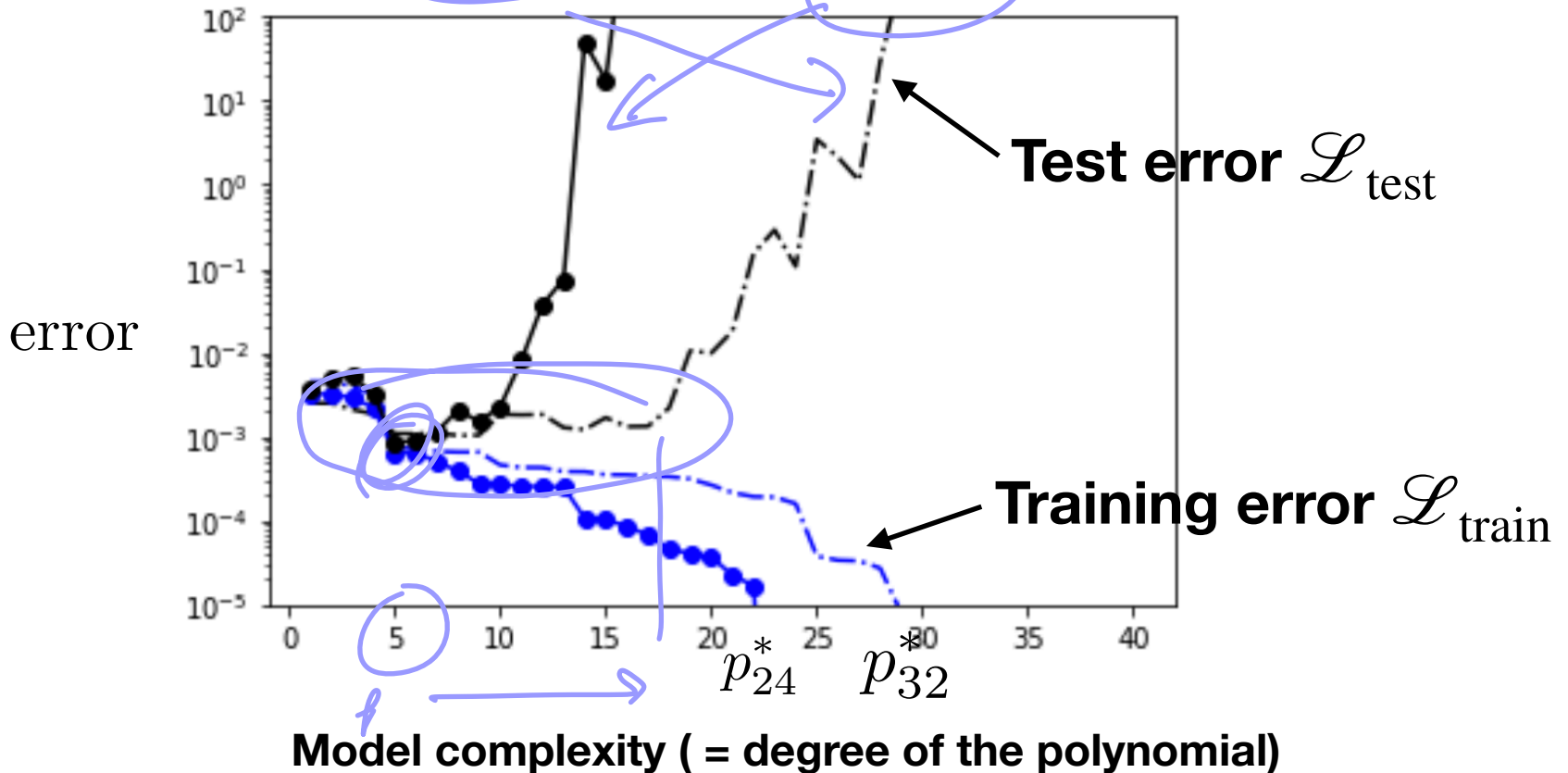
- Assume  $N=30$



- Given sample size  $N$  there is a threshold,  $p_N^*$ , where training error is zero
- Training error is **always** monotonically non-increasing
- Test error has a trend of going down and then up, but fluctuates

# Variance decreases with more data, letting you fit more complex models

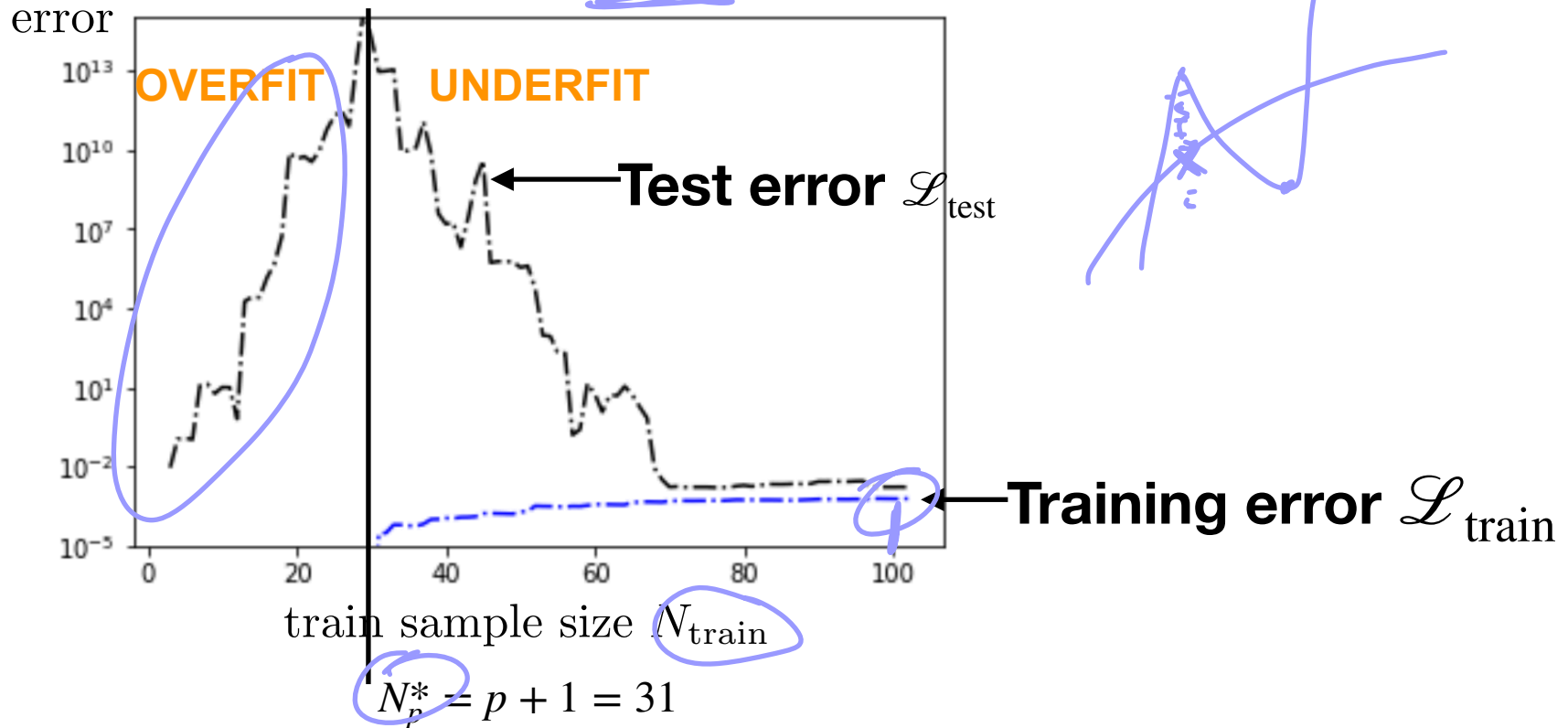
- Now compare  $N=40$  to previous  $N=30$  case



- The threshold,  $p_N^*$ , moves right as dataset size increases
- Training error tends to increase, because more points need to fit
- Test error tends to decrease, because Variance decreases

# Variance decreases with more data, letting you fit more complex models

- Choose model complexity  $p=30$ , vary dataset size  $n$



- There is a threshold,  $N_p^*$ , below which training error is zero (extreme overfit)
- Above the threshold, test error tends to decrease
- Training error tends to increase (harder to fit so much data with simple model)

Regularization helps avoid overfitting

# Ridge Regression

---

# Regularization in Linear Regression

---

Recall Least Squares:  $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

$$= \arg \min_w (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w)$$

when  $(\mathbf{X}^T \mathbf{X})^{-1}$  exists....  $= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

# Regularization in Linear Regression

---

Recall Least Squares:  $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

$$= \arg \min_w (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w)$$

when  $(\mathbf{X}^T \mathbf{X})^{-1}$  exists....  $= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

What if  $x_i \in \mathbb{R}^d$  and  $d > n$ ?

# Regularization in Linear Regression

---

Recall Least Squares:  $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

When  $x_i \in \mathbb{R}^d$  and  $d > n$  the objective function is flat in some directions:



# Regularization in Linear Regression

---

Recall Least Squares:  $\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$

When  $x_i \in \mathbb{R}^d$  and  $d > n$  the objective function is flat in some directions:

Implies optimal solution is *not unique* and unstable due to lack of curvature:

- small changes in training data result in large changes in solution
- often the magnitudes of  $w$  are “very large”



**Regularization imposes “simpler” solutions by a “complexity” penalty**

# Sensitivity increases overfitting

$y = wx + b$

- For a linear model,  
$$y \simeq b + w_1x_1 + w_2x_2 + \dots + w_dx_d$$
if  $|w_j|$  is large then the prediction is **sensitive** to small changes in  $x_i$
- Large **sensitivity** leads to overfitting and poor generalization, and equivalently models that overfit tend to have large weights
- Note that  $b$  is a constant and hence there is no sensitivity for the offset  $b$   
*never regularize  $b$*
- In **Ridge Regression**, we use a regularizer  $\|w\|_2^2$  to measure and control the sensitivity of the predictor
- And optimize for small loss and small sensitivity, by adding a **regularizer** in the objective (assume no offset for now)

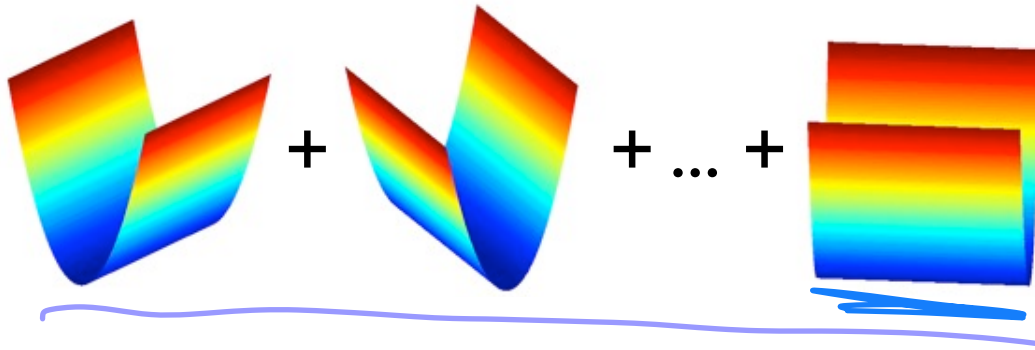
$$\hat{w}_{\text{ridge}} = \arg \min_w \left\{ \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2 \right\}$$

*regularization coefficient*

# Ridge Regression

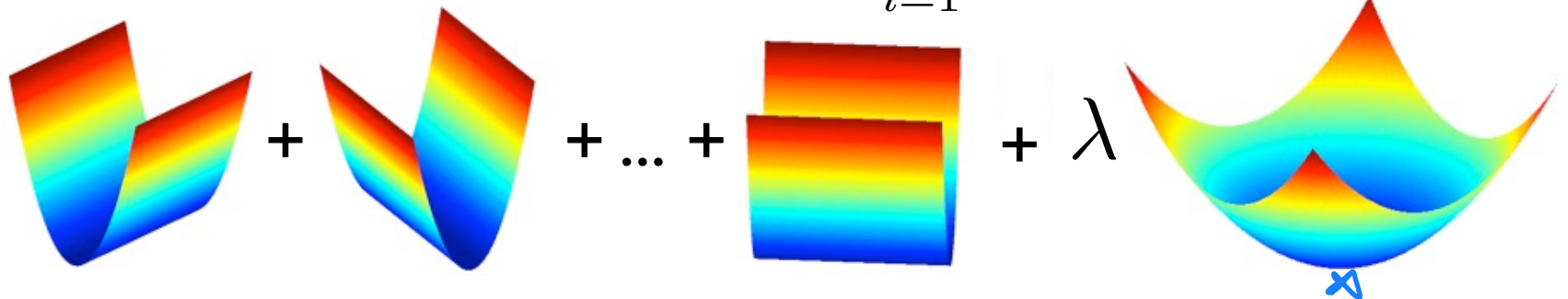
- Old Least squares objective:

$$\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$$



- Ridge Regression objective:

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda ||w||_2^2$$



# Minimizing the Ridge Regression Objective

$X \in n \times d$   
 $w \in d \times 1$

$$\hat{w}_{\text{ridge}} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

*square*  
L2 norm

$$\|w\|_p \triangleq (|w_1|^p + \dots + |w_d|^p)^{1/p}$$

$$I^d = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \in d \times d$$

$\xrightarrow{d}$

$$= \|y - Xw\|^2 + \lambda \|w\|^2$$

$$= (Xw - y)^T (Xw - y) + \lambda w^T w$$

$$\nabla_w f = 2X^T(Xw - y) + 2\lambda w = 0$$

$$(X^T X + \lambda I)w = X^T y$$

$$\hat{w}_{\text{RIDGE}} = (X^T X + \lambda I)^{-1} X^T y$$

Scalar derivative	Vector derivative
$f(x) \rightarrow \frac{df}{dx}$	$f(\mathbf{x}) \rightarrow \frac{df}{d\mathbf{x}}$
$bx \rightarrow b$	$\mathbf{x}^T \mathbf{B} \rightarrow \mathbf{B}$
$bx \rightarrow b$	$\mathbf{x}^T \mathbf{b} \rightarrow \mathbf{b}$
$x^2 \rightarrow 2x$	$\mathbf{x}^T \mathbf{x} \rightarrow 2\mathbf{x}$
$bx^2 \rightarrow 2bx$	$\mathbf{x}^T \mathbf{B} \mathbf{x} \rightarrow 2\mathbf{B} \mathbf{x}$

From our previous lin. regression lecture we had:

$$\underset{\omega}{\text{argmin}} (y - X\omega)^T (y - X\omega) + \lambda \omega^T \omega$$

$$\nabla_{\omega} [-2y^T X \omega + \omega^T X^T X \omega + \lambda \omega^T \omega]$$

$$-2X^T y + 2X^T X \omega + 2\lambda \omega = 0$$

$$X^T X \omega + \lambda \omega = X^T y$$

$$(X^T X + \lambda I) \omega = X^T y$$

$$\hat{\omega}_{\text{RIDGE}} = (X^T X + \lambda I)^{-1} X^T y$$

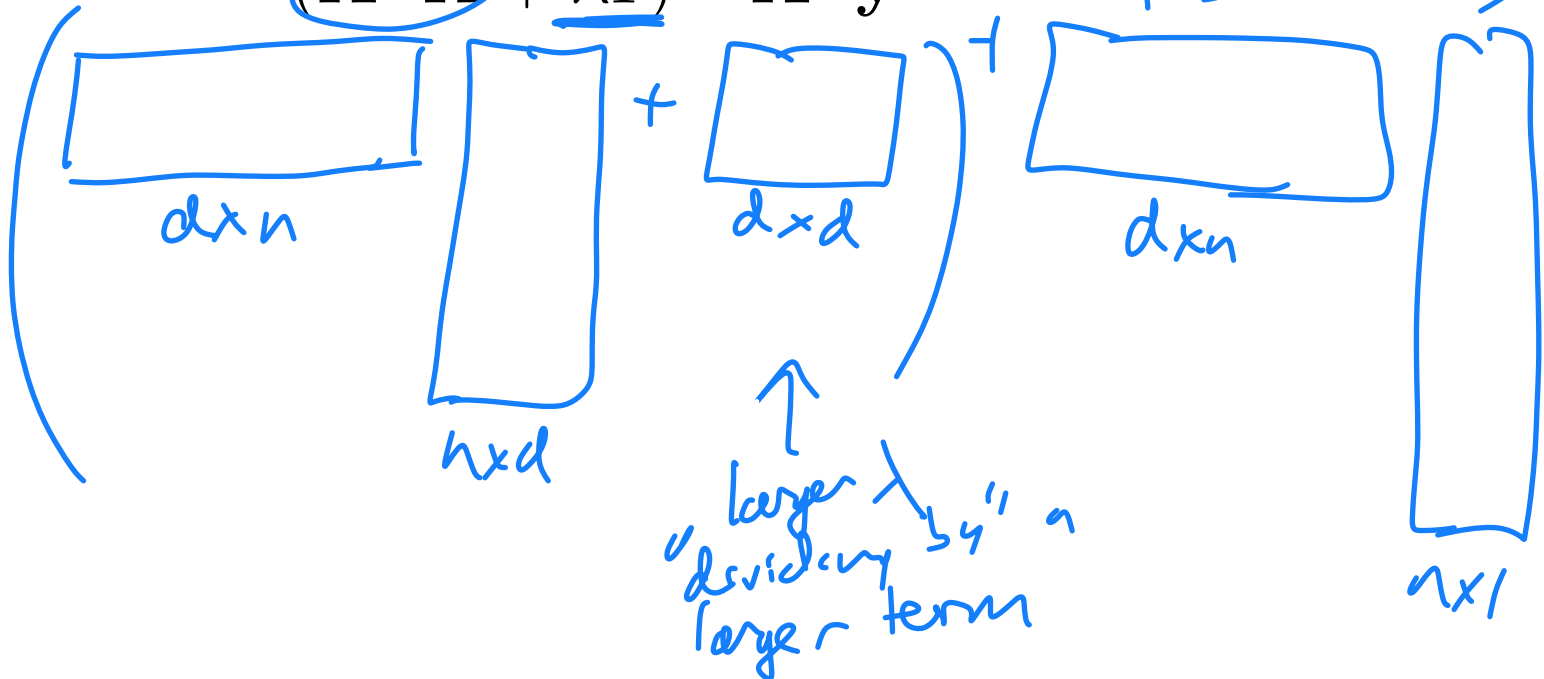
# Shrinkage Properties

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

$X \in n \times d$

$$= (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

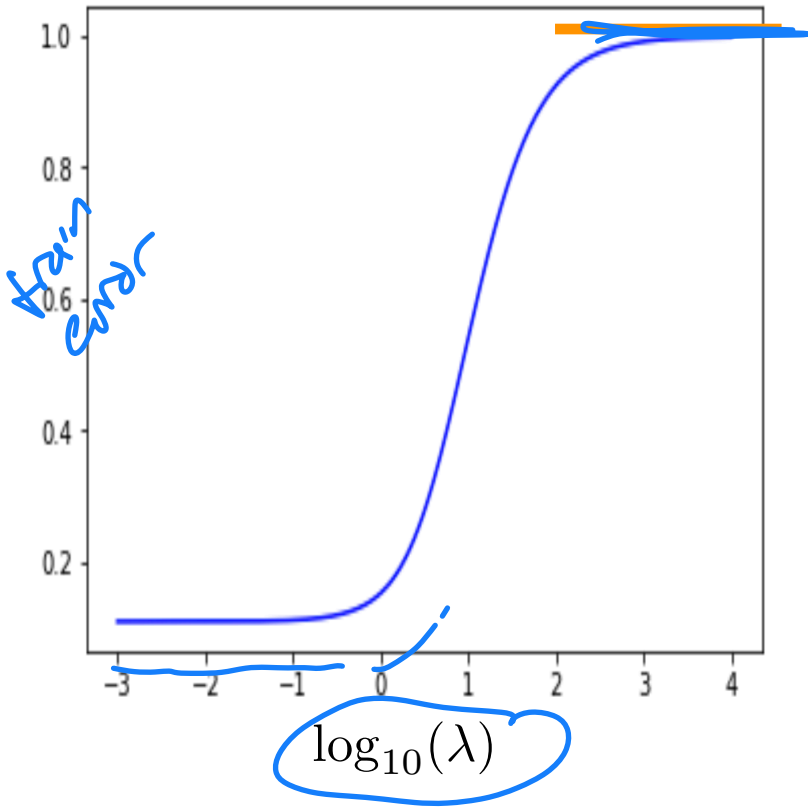
PSD  
GPD



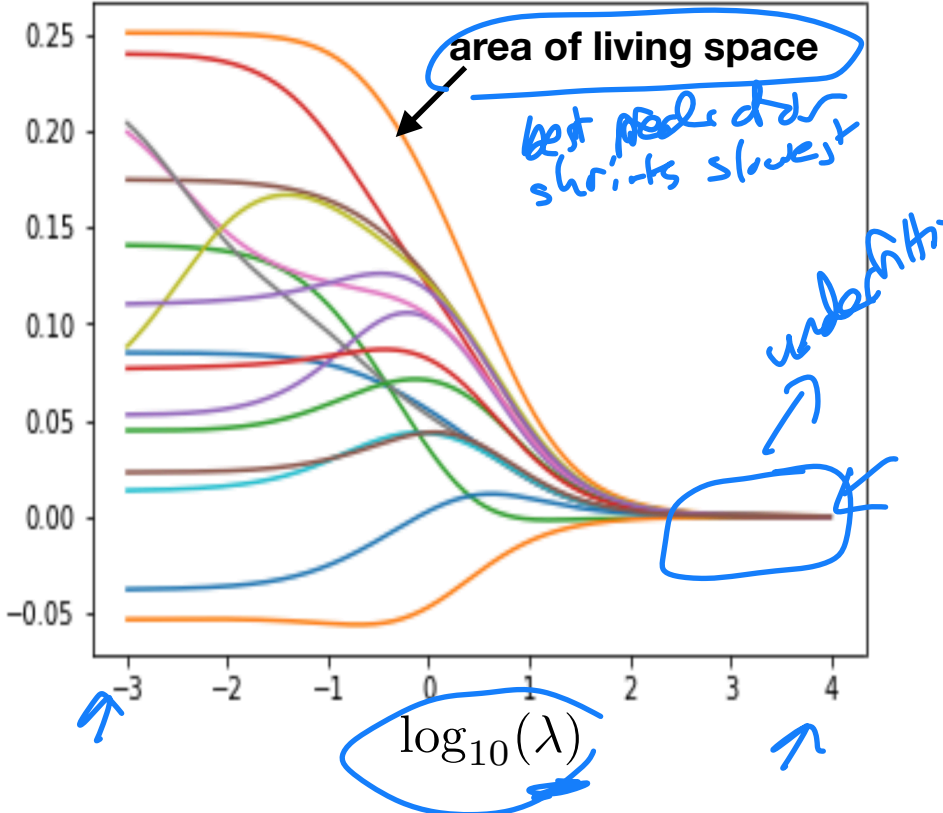
- When  $\lambda = 0$ , this gives the least squares model
- This defines a family of models hyper-parametrized by  $\lambda$
- Large  $\lambda$  means more regularization and simpler model
- Small  $\lambda$  means less regularization and more complex model

**Ridge regression:** minimize  $\sum_{i=1}^n (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$

training MSE  $\frac{1}{n} \sum_{i=1}^n (y_i - x_i^T \hat{w}_{\text{ridge}}^{(\lambda)})^2$



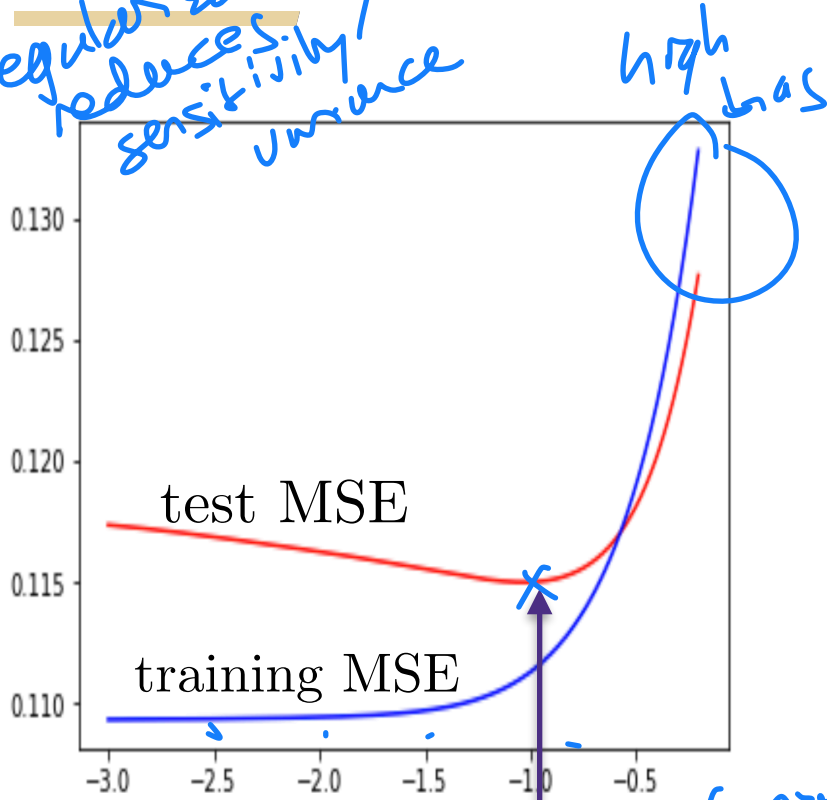
Housing price predictor  $w_i$ 's



- Left plot: leftmost training error is with no regularization: 0.1093
- Left plot: rightmost training error is variance of the training data: 0.9991
- Right plot: called **regularization path**

Ridge regression: minimize  $\sum_{i=1}^n (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$

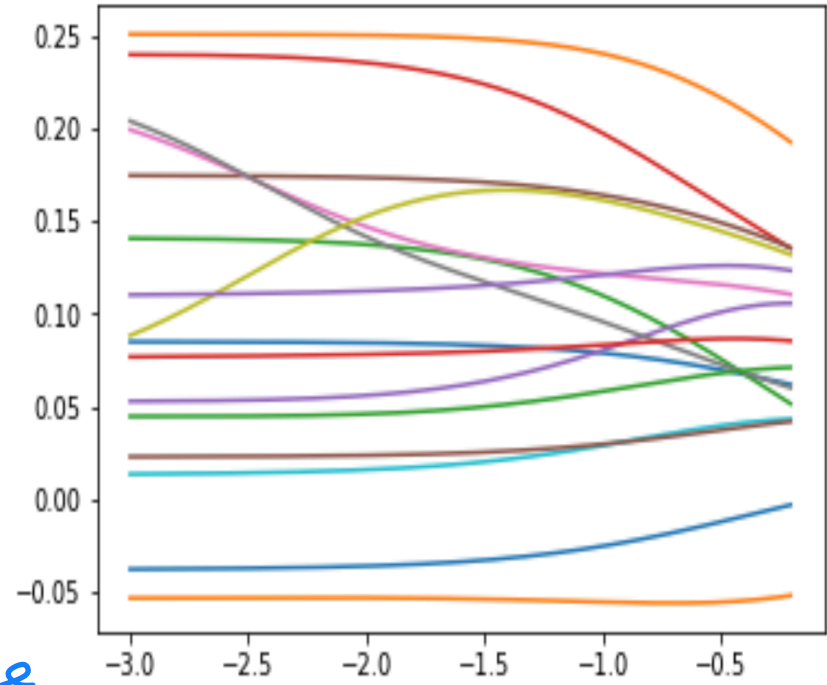
regularization  
reduces  
sensitivity/  
variance



$\log_{10}(\lambda)$

bias/variance trade-off  
we're not changing the model class, we're just changing  $\lambda$

Housing price predictor  $w'_i$ 's



$\log_{10}(\lambda)$

- this gain in test MSE comes from shrinking  $w$ 's to get a less sensitive predictor (which in turn reduces the variance)

- this is the role of regularizer

# Bias-Variance Properties

- Recall:  $\hat{w}_{\text{ridge}} = \underbrace{(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model:  $x_i \sim P_X$ ,  $\mathbf{y} = \mathbf{X}w + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$   
for some ground truth model parameter  $w$  *ground truth*
- The true error at a sample with feature  $x$  is

$$\mathbb{E}_{y, \mathcal{D}_{\text{train}} | x} [(y - x^T \hat{w}_{\text{ridge}})^2 | x]$$

*test sample*  $\uparrow$   $\rightarrow$  fit to a mean set  $X, y$

# Bias-Variance Properties

- Recall:  $\hat{\mathbf{w}}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model:  $x_i \sim P_X$ ,  $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- The true error at a sample with feature  $x$  is

$$\begin{aligned} & \mathbb{E}_{\mathbf{y}, \mathcal{D}_{\text{train}} | x} [(y - x^T \hat{\mathbf{w}}_{\text{ridge}})^2 | x] \\ &= \underbrace{\mathbb{E}_{y|x} [(y - \mathbb{E}[y|x])^2 | x]}_{\text{Irreducible Error}} + \underbrace{\mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}[y|x] - x^T \hat{\mathbf{w}}_{\text{ridge}})^2 | x]}_{\text{Learning Error}} \end{aligned}$$

*same as before*  
 $\rightarrow \pi(x)$

# Bias-Variance Properties

- Recall:  $\hat{w}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model:  $x_i \sim P_X$ ,  $\mathbf{y} = \mathbf{X}w + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- The true error at a sample with feature  $x$  is

$$\begin{aligned} & \mathbb{E}_{y, \mathcal{D}_{\text{train}} | x} [(y - x^T \hat{w}_{\text{ridge}})^2 | x] \\ &= \mathbb{E}_{y|x} [(y - \mathbb{E}[y | x])^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}[y | x] - x^T \hat{w}_{\text{ridge}})^2 | x] \\ &= \mathbb{E}_{y|x} [(y - x^T w)^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(x^T \underbrace{w}_{\text{ground truth}} - x^T \hat{w}_{\text{ridge}})^2 | x] \end{aligned}$$

# Bias-Variance Properties

- Recall:  $\hat{w}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model:  $x_i \sim P_X$ ,  $\mathbf{y} = \mathbf{X}w + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- The true error at a sample with feature  $x$  is

$$\mathbb{E}_{y, \mathcal{D}_{\text{train}} | x} [(y - x^T \hat{w}_{\text{ridge}})^2 | x]$$

$$= \mathbb{E}_{y|x} [(y - \mathbb{E}[y | x])^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}[y | x] - x^T \hat{w}_{\text{ridge}})^2 | x]$$

$$= \mathbb{E}_{y|x} [(y - x^T w)^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(x^T w - x^T \hat{w}_{\text{ridge}})^2 | x]$$

$$= \underbrace{\sigma^2}_{\text{Irreduc. Error}} + \underbrace{(x^T w - \mathbb{E}_{\tilde{\mathcal{D}}_{\text{train}}} [x^T \hat{w}_{\text{ridge}} | x])^2}_{\text{Bias-squared}} + \underbrace{\mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}_{\tilde{\mathcal{D}}_{\text{train}}} [x^T \hat{w}_{\text{ridge}} | x] - x^T \hat{w}_{\text{ridge}})^2 | x]}_{\text{Variance}}$$

Irreduc. Error

Bias-squared

Variance

# Bias-Variance Properties

$$\propto \lambda \|w\|_2^2$$

- Recall:  $\hat{w}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model:  $x_i \sim P_X$ ,  $\mathbf{y} = \mathbf{X}w + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- The true error at a sample with feature  $x$  is

$$\begin{aligned} & \mathbb{E}_{y, \mathcal{D}_{\text{train}} | x} [(y - x^T \hat{w}_{\text{ridge}})^2 | x] \\ &= \mathbb{E}_{y|x} [(y - \mathbb{E}[y | x])^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}[y | x] - x^T \hat{w}_{\text{ridge}})^2 | x] \\ &= \mathbb{E}_{y|x} [(y - x^T w)^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(x^T w - x^T \hat{w}_{\text{ridge}})^2 | x] \\ &= \underbrace{\sigma^2}_{\text{Irreduc. Error}} + \underbrace{(x^T w - \mathbb{E}_{\mathcal{D}_{\text{train}}} [x^T \hat{w}_{\text{ridge}} | x])^2}_{\text{Bias-squared}} + \underbrace{\mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}_{\mathcal{D}_{\text{train}}} [x^T \hat{w}_{\text{ridge}} | x] - x^T \hat{w}_{\text{ridge}})^2 | x]}_{\text{Variance}} \end{aligned}$$

Irreduc. Error

Bias-squared

Variance

Suppose  $\mathbf{X}^T \mathbf{X} = n\mathbf{I}$ , then  $\hat{w}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T (\mathbf{X}w + \epsilon)$

$$= \frac{n}{n + \lambda} w + \frac{1}{n + \lambda} \mathbf{X}^T \epsilon$$

sampling data from independent Gaussians

$\lambda$  trades off weight on data vs. noise

# Bias-Variance Properties

Suppose  $\mathbf{X}^T \mathbf{X} = n\mathbf{I}$ , then

$$\hat{w}_{\text{ridge}} = \frac{n}{n + \lambda} w + \frac{1}{n + \lambda} \mathbf{X}^T \epsilon$$

- Recall:  $\hat{w}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$
- To analyze bias-variance tradeoff, we need to assume probabilistic generative model:  $x_i \sim P_X$ ,  $\mathbf{y} = \mathbf{X}w + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- The true error at a sample with feature  $x$  is

$$\mathbb{E}_{y, \mathcal{D}_{\text{train}} | x} [(y - x^T \hat{w}_{\text{ridge}})^2 | x]$$

$$= \mathbb{E}_{y|x} [(y - \mathbb{E}[y | x])^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}[y | x] - x^T \hat{w}_{\text{ridge}})^2 | x]$$

$$= \mathbb{E}_{y|x} [(y - x^T w)^2 | x] + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(x^T w - x^T \hat{w}_{\text{ridge}})^2 | x]$$

$$= \sigma^2 + (x^T w - \mathbb{E}_{\mathcal{D}_{\text{train}}} [x^T \hat{w}_{\text{ridge}} | x])^2 + \mathbb{E}_{\mathcal{D}_{\text{train}}} [(\mathbb{E}_{\mathcal{D}_{\text{train}}} [x^T \hat{w}_{\text{ridge}} | x] - x^T \hat{w}_{\text{ridge}})^2 | x]$$

(verify at home)

$$= \sigma^2 + \frac{\lambda^2}{(n + \lambda)^2} (w^T x)^2 + \frac{\sigma^2 n}{(n + \lambda)^2} \|x\|_2^2$$

Irreduc. Error

Bias-squared

Variance

$\lambda$  trades off bias vs. variance  
larger  $\lambda \rightarrow$  smaller variance, larger bias

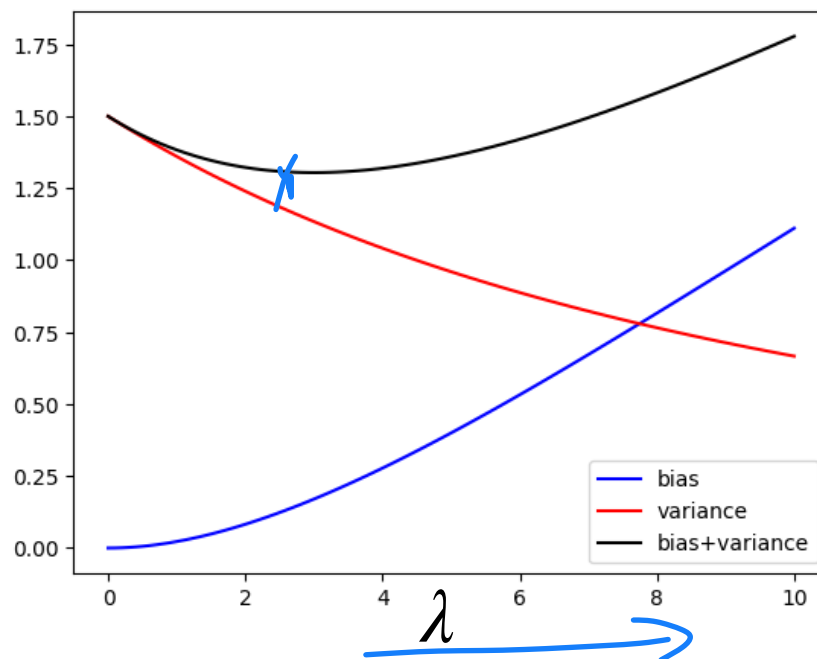
# Bias-Variance Properties

Suppose  $\mathbf{X}^T \mathbf{X} = n\mathbf{I}$ ,

- Ridge regressor:  $\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$
- True error

$$\mathbb{E}_{y, \mathcal{D}_{train}|x} [(y - x^T \hat{w}_{ridge})^2 | x] = \underbrace{\sigma^2 + \frac{\lambda^2}{(n + \lambda)^2} (w^T x)^2}_{\text{Bias-squared}} + \underbrace{\frac{\sigma^2 n}{(n + \lambda)^2} \|x\|_2^2}_{\text{Variance}}$$

$$d=10, n=20, \sigma^2 = 3.0, \|w\|_2^2 = 10$$



as  $\lambda \rightarrow 0$ ,  
 $\hat{w}_{ridge} \rightarrow \hat{w}_{LS}$

as  $\lambda \rightarrow \infty$ ,  
 $\hat{w}_{ridge} \rightarrow 0$

# What you need to know...

---

## > Regularization

- Penalizes complex models towards preferred, simpler models

## > Ridge regression

- L<sub>2</sub> penalized least-squares regression
  - Regularization parameter trades off model complexity with training error
  - Never regularize the offset!
- 

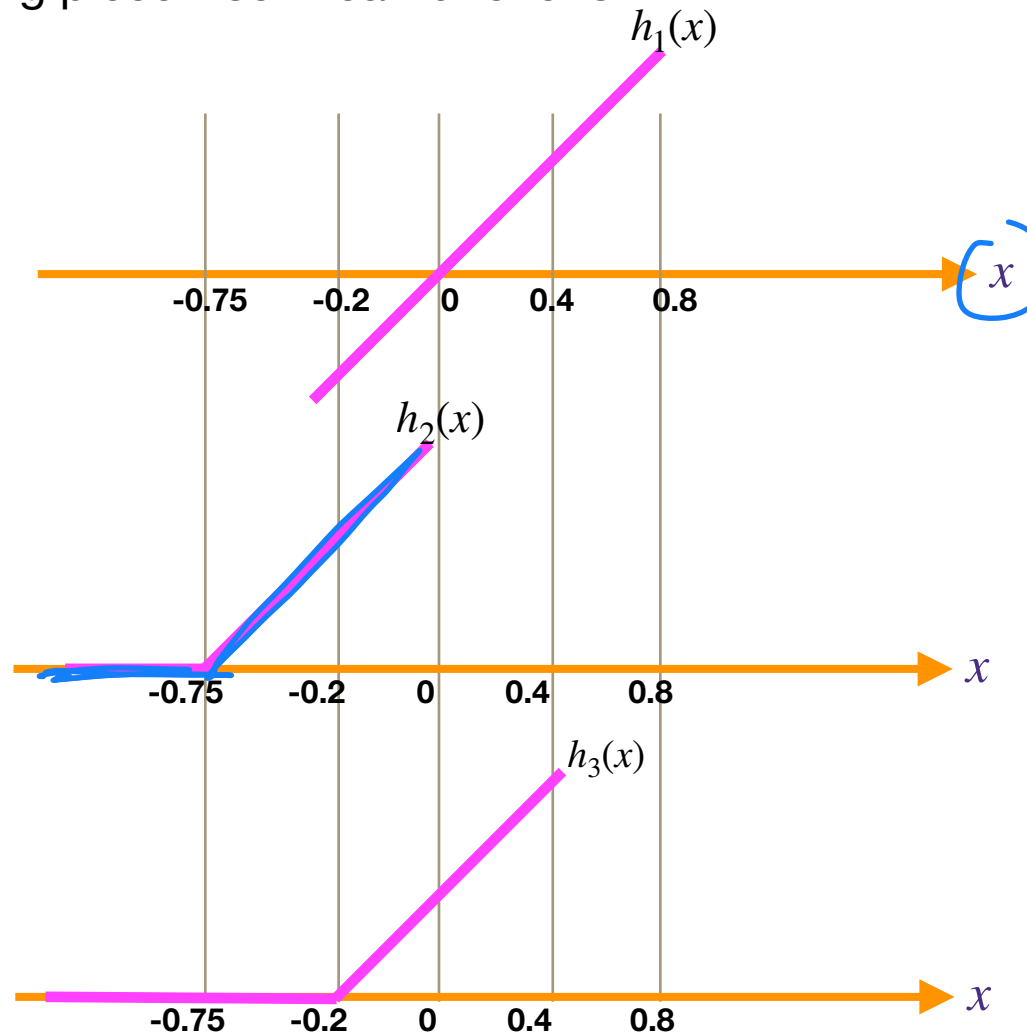
$$\lambda \|w\|_2^2$$

# Example: piecewise linear fit

- we fit a linear model:
$$f(x) = b + w_1 h_1(x) + w_2 h_2(x) + w_3 h_3(x) + w_4 h_4(x) + w_5 h_5(x)$$
- with a specific choice of features using piecewise linear functions

$$\underline{h(x)} = \begin{bmatrix} h_1(x) \\ h_2(x) \\ h_3(x) \\ h_4(x) \\ h_5(x) \end{bmatrix} = \begin{bmatrix} x \\ [x + 0.75]^+ \\ [x + 0.2]^+ \\ [x - 0.4]^+ \\ [x - 0.8]^+ \end{bmatrix}$$

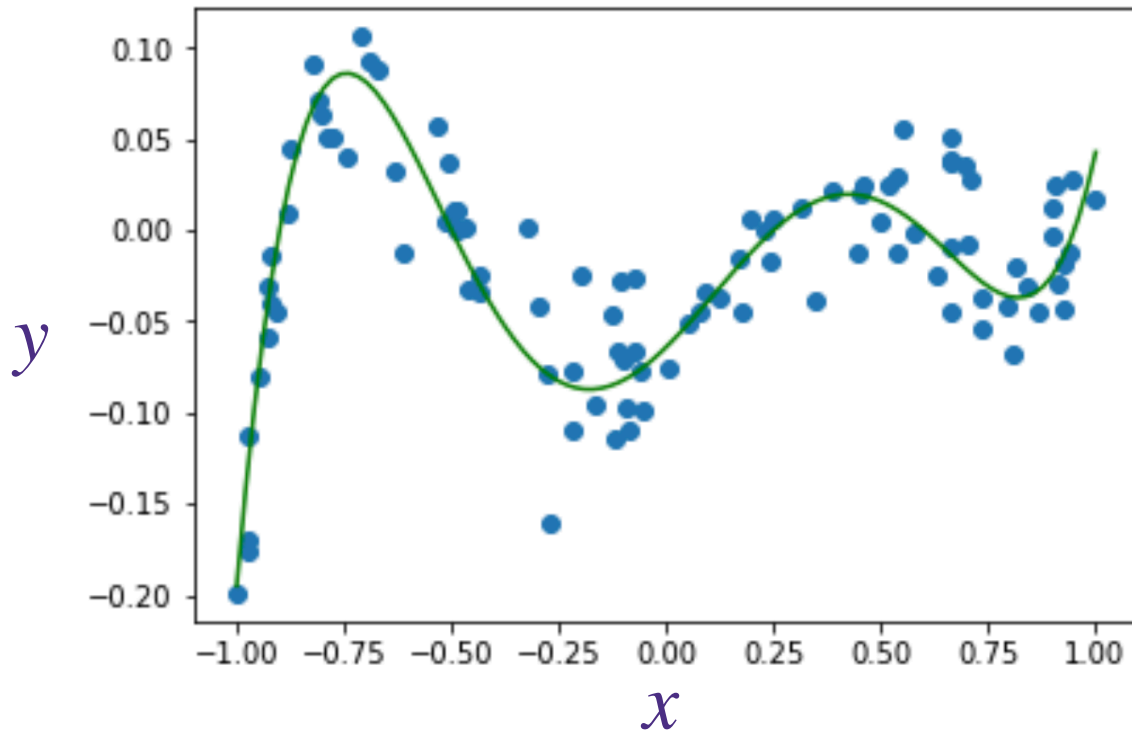
$[a]^+ \triangleq \max\{a, 0\}$   
→ relu  
rectified  
linear  
unit



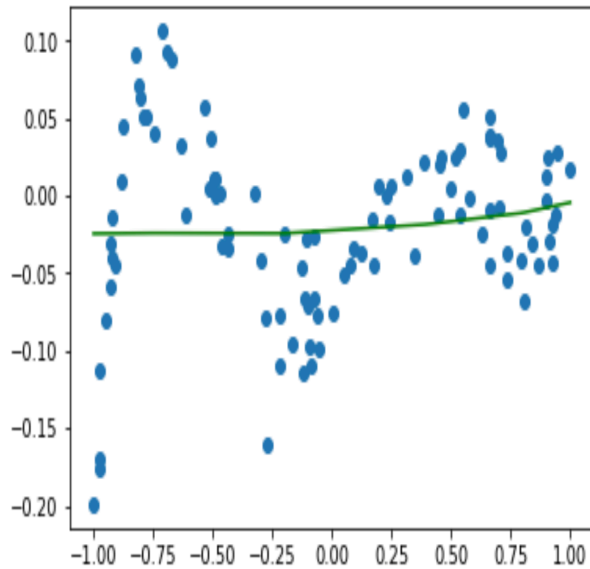


# Example: piecewise linear fit

- we fit a linear model:  
$$f(x) = b + w_1h_1(x) + w_2h_2(x) + w_3h_3(x) + w_4h_4(x) + w_5h_5(x)$$
- with a specific choice of features using piecewise linear functions



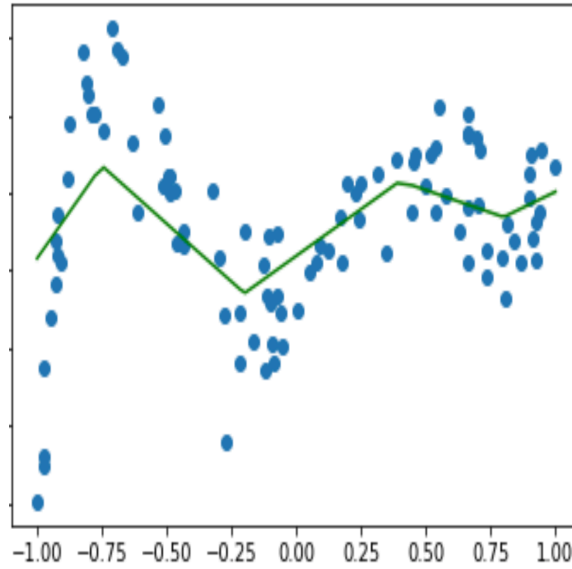
# Example: piecewise linear fit (ridge regression)



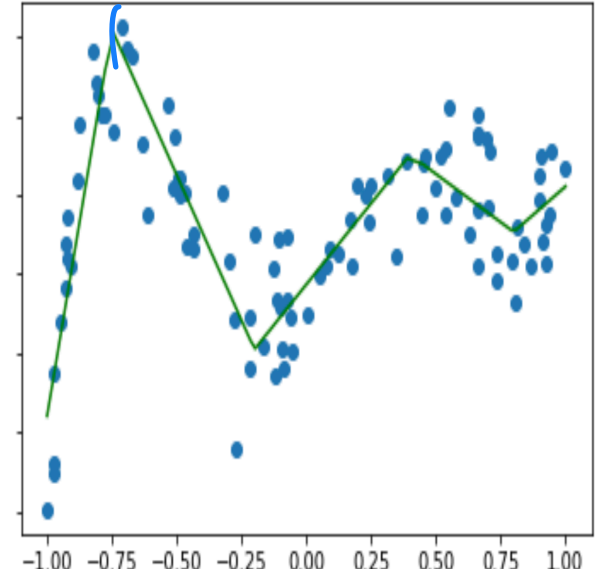
$\lambda = 1$

high regularization  
less complex

smaller weights  $\rightarrow$   
less change in  
slope in each region



$\lambda = 0.005$

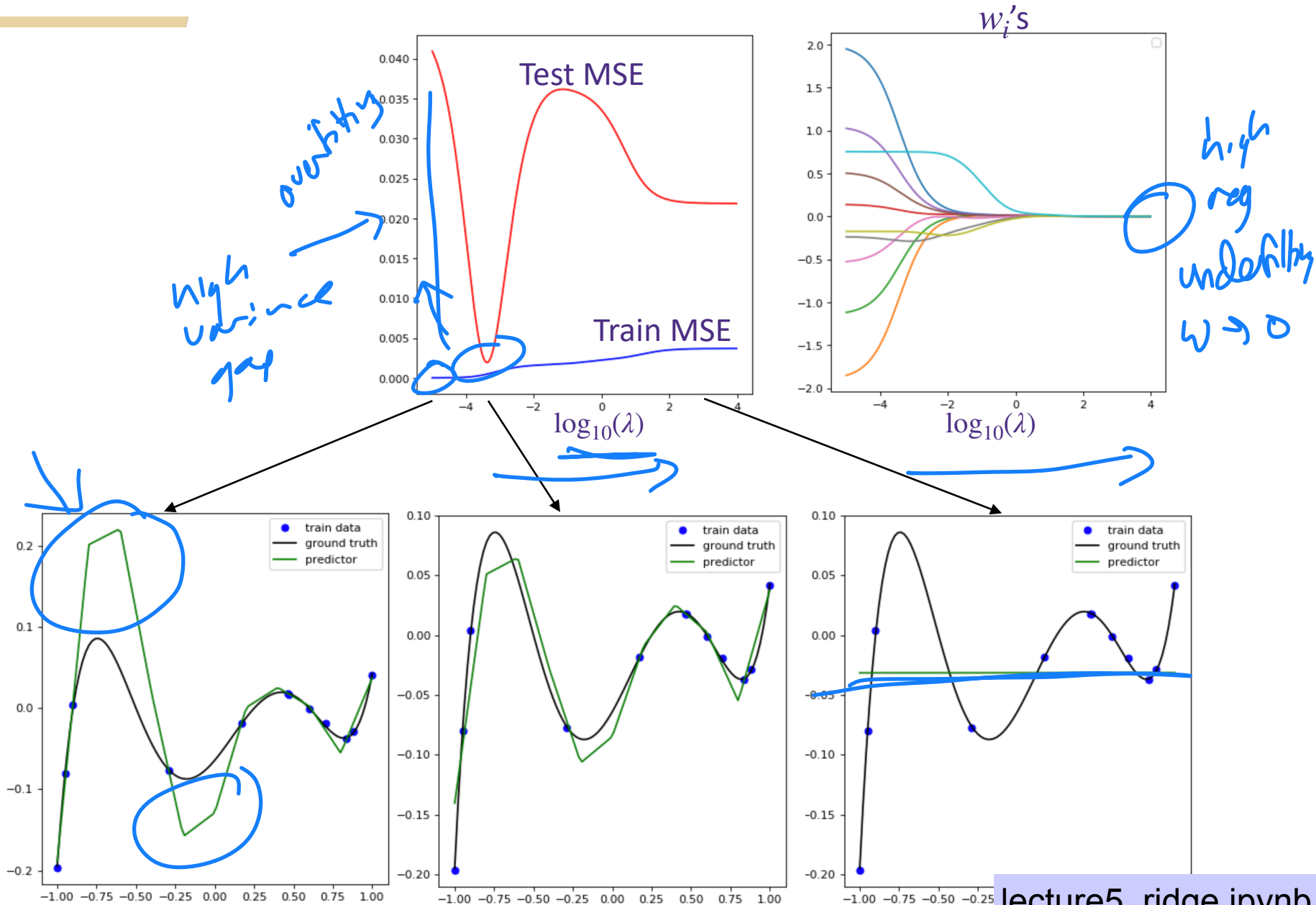


$\lambda = 0.000001$

big change in  
each region

We do not observe overfitting, as  $d=5$   $\ll$   $n=100$

# Piecewise linear with $w \in \mathbb{R}^{10}$ and $n=11$ samples



# Sparsity & the LASSO

---

How to make the model compact and interpretable

# Sparsity

$$\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$$

- We learned to measure sensitivity by the size of weights:  $\|w\|_2^2$
- Vector  $w$  is sparse, if many entries are zero
  - A vector  $w$  is said to be  $k$ -sparse if at most  $k$  entries are non-zero
  - We are interested in  $k$ -sparse  $w$  with  $k \ll d$
  - Why do we prefer sparse vector  $w$  in practice?

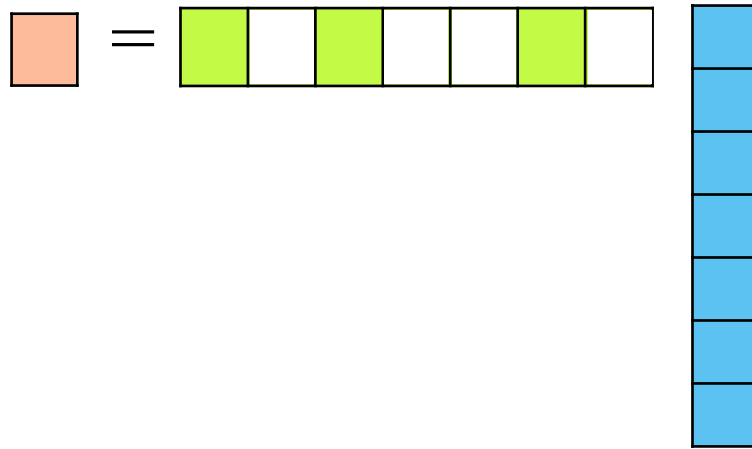
- computationally more efficient ✓
- get rid of redundant/spurious features ✓  
y remove more PII
- explainability / interpretability ✓

# Sparsity

$$\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$$

- Vector  $w$  is **sparse**, if many entries are zero
  - Efficiency**: If  $\text{size}(w) = 100 \text{ Billion}$ , each prediction  $w^T x$  is expensive:
    - If  $w$  is sparse, prediction computation only depends on number of non-zeros in  $w$

$$\hat{y}_i = \hat{w}_{LS}^T x_i$$



$$= \sum_{j=1}^d \hat{w}_{LS}[j] \times x_i[j] = \sum_{j: \hat{w}_{LS}[j] \neq 0} \hat{w}_{LS}[j] \times x_i[j]$$

Computational complexity decreases from  $2d$  to  $2k$  for  $k$ -sparse  $\hat{w}_{LS}$

# Sparsity

$$\hat{w}_{LS} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$$

- Vector  $w$  is **sparse**, if many entries are zero
  - Interpretability:** What are the relevant features to make a prediction?

Basmt sqft finished  
Basmt sqft in area



Lot size  
Single Family  
Year built  
Last sold price

Last sale price/sqft

Finished sqft  
Unfinished sqft

Finished basement sqft

# floors  
Flooring types

Parking type  
Parking amount  
Cooling

Heating  
Exterior materials  
Roof type  
Structure style

Dishwasher  
Garbage disposal  
Microwave  
Range / Oven  
Refrigerator  
Washer  
Dryer  
Laundry location  
Heating type  
Jetted Tub  
Deck  
Fenced Yard  
Lawn  
Garden  
Sprinkler System

How do we find “best” subset of features useful in predicting the price among all possible combinations?

# Finding best subset of features that explain the outcome/label: Exhaustive

- Try all subsets of size 1, 2, 3, ... and one that minimizes validation error
  - Problem?
  - Any Ideas?

exponential in  $d$

Feature reduction SVD

# Finding best subset: Greedy

## Forward stepwise:

Starting from simple model and iteratively add features most useful to fit

### Forward Greedy

1:  $T \leftarrow \emptyset$

2: For  $j = 1, \dots, k$  do

3:  $j^* \leftarrow \arg \min_{\ell} \min_w \sum_{i=1}^n \left( y_i - \sum_{j \in T \cup \{\ell\}} w[j] \times x_i[j] \right)^2$

4:  $T \leftarrow T \cup \{j^*\}$

$O(d^2)$

## Backward stepwise:

Start with full model and iteratively remove features least useful to fit

## Combining forward and backward steps:

In forward algorithm, insert steps to remove features no longer as important

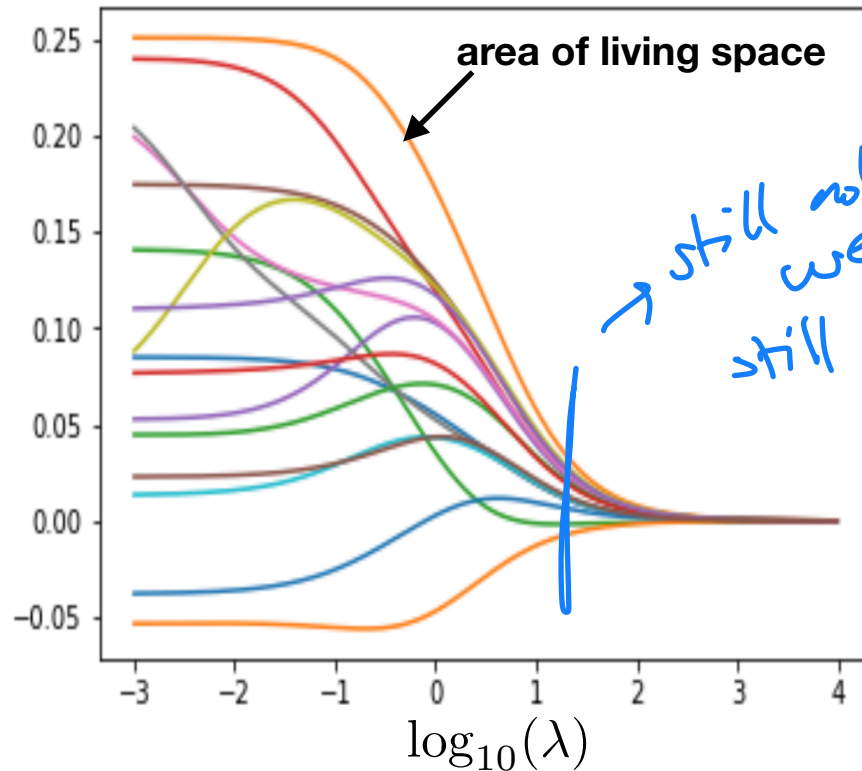
*Lots of other variants, too.*

# Finding best subset: Regularize

Recall that Ridge regression makes coefficients small

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

$w_i$ 's



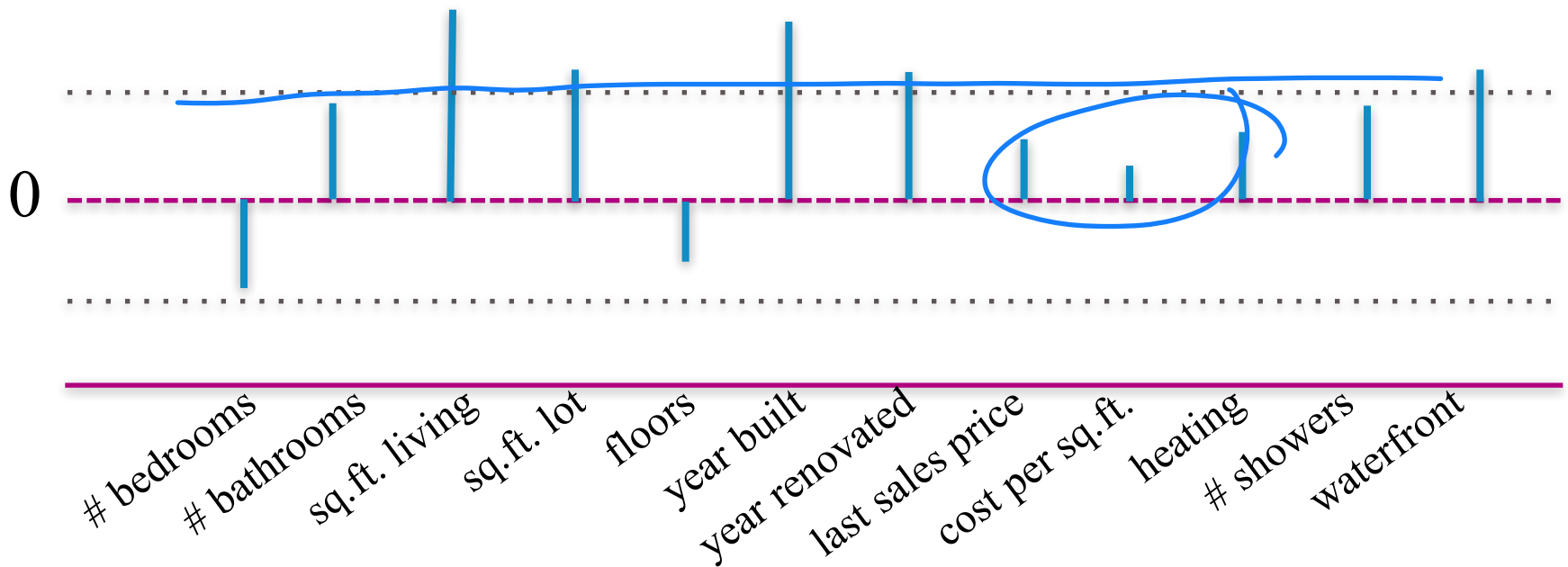
still not sending weights to 0 exactly  
still not sparse

# Thresholded Ridge Regression

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

→ how to pick threshold?

- Why don't we just set **small** ridge coefficients to 0?
  - **Any issues?**

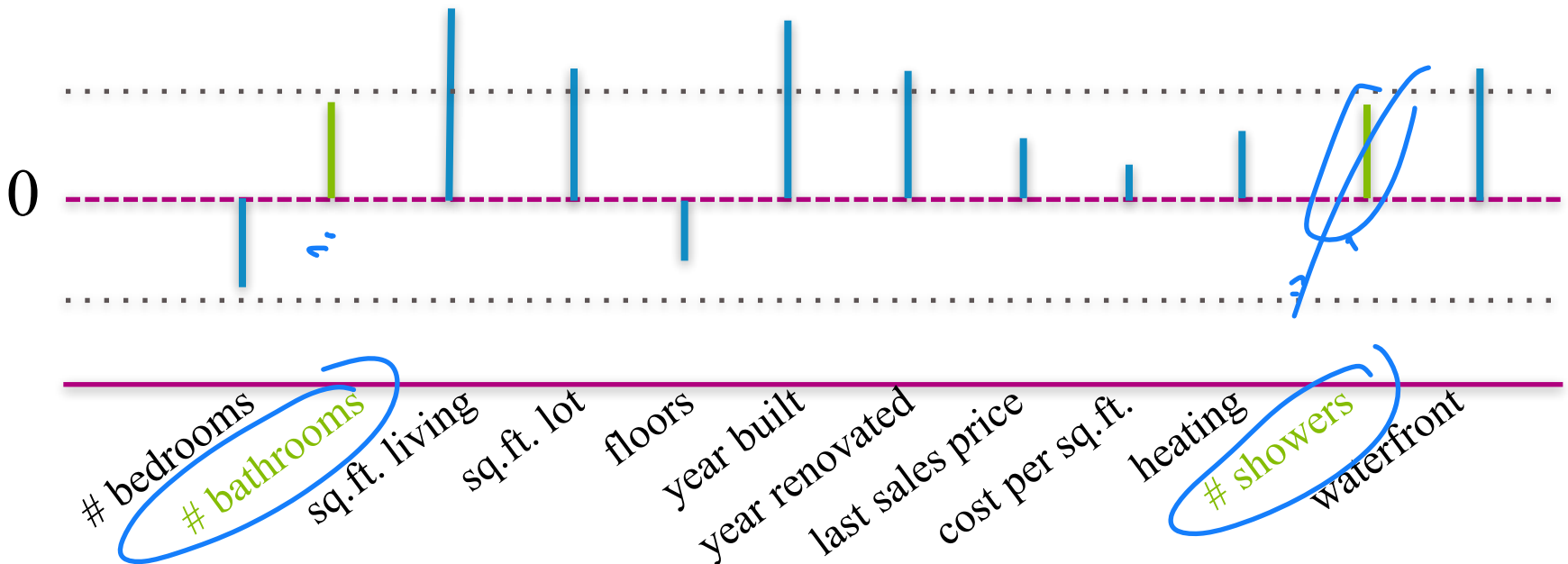


# Thresholded Ridge Regression

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

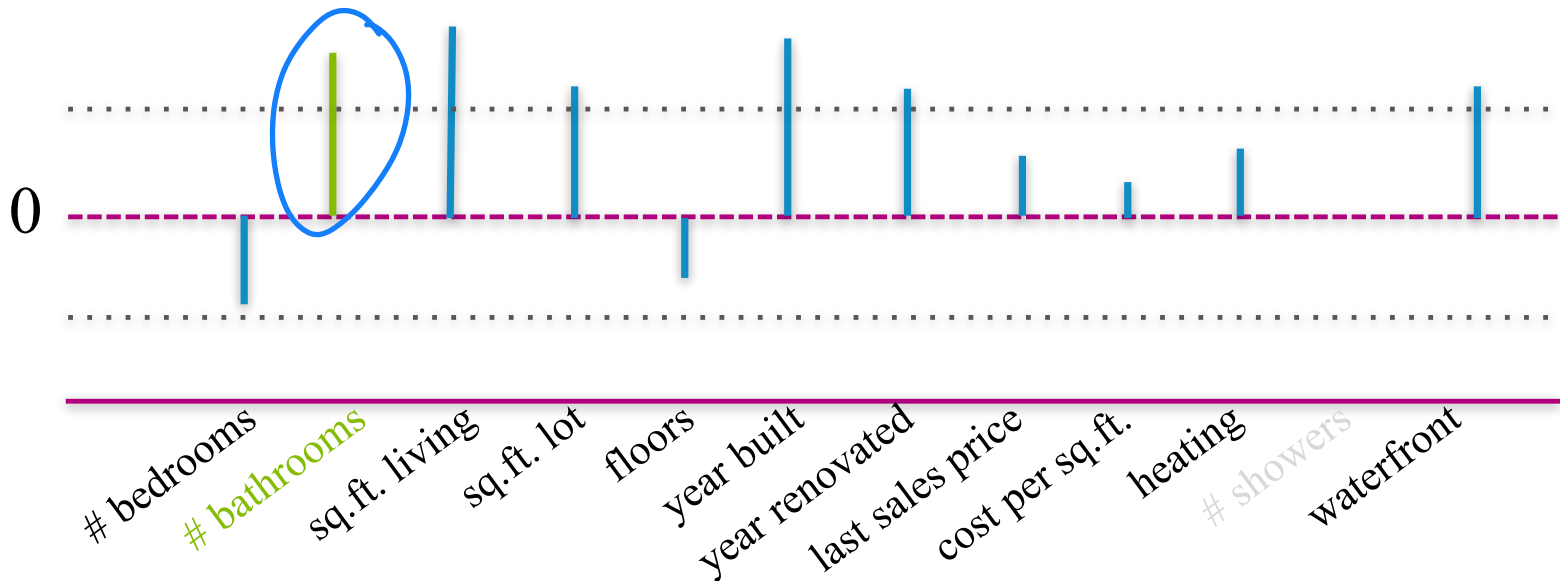
- Consider two **related** features (bathrooms, showers)
- Consider  $w[\text{bath}] = 1$  and  $w[\text{shower}] = 1$ , and  $w[\text{bath}] = 2$  and  $w[\text{shower}] = 0$ , which one does ridge regression choose? (assuming #bathroom=#showers in every house)

$$1^2 + 1^2 = 2$$
$$2^2 + 0^2 = 4$$



# Thresholded Ridge Regression

- What if we didn't include showers? Weight on bathrooms increases!
- We want a feature selection scheme that selects one of (#bathroom) or (#showers) automatically, using the fact that if you delete #showers #bathroom is an important feature

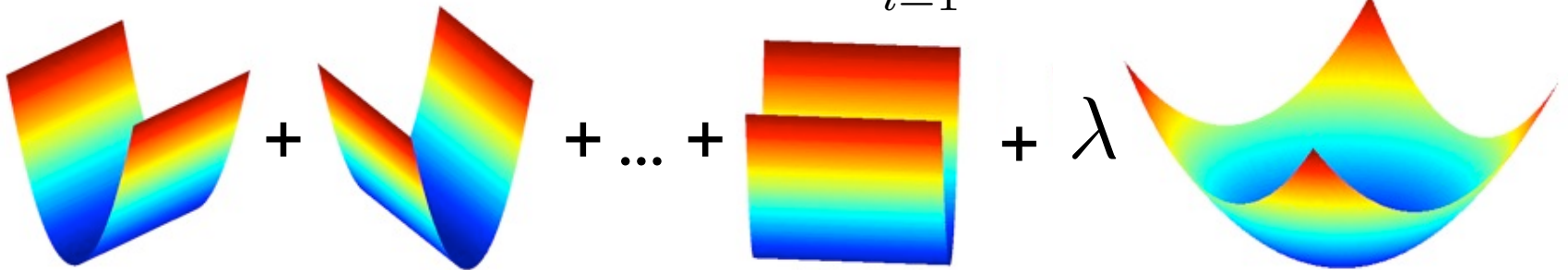


**Can another regularizer perform selection automatically?**

# Ridge vs. Lasso Regression

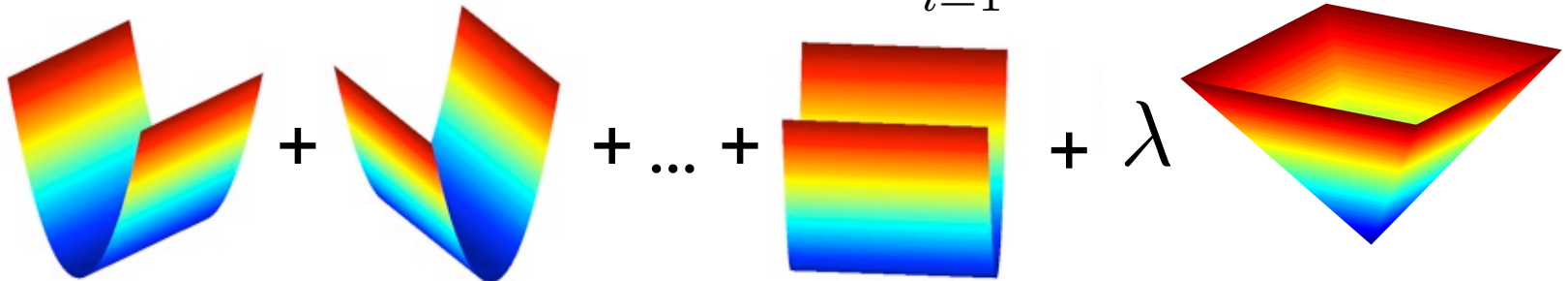
- Ridge Regression objective:

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$



- Lasso objective:

$$\hat{w}_{lasso} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_1$$



# Ridge vs. LASSO Regression

LASSO = Least Absolute Shrinkage and Selection Operator

- Recall Ridge Regression objective:

$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

- sensitivity of a model  $w$  is measured in squared  $\ell_2$  norm  $\|w\|_2^2$
- A principled method to get sparse model is **Lasso** with regularized objective:

$$\hat{w}_{lasso} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_1$$

- sensitivity of a model  $w$  is measured in  $\ell_1$  norm:

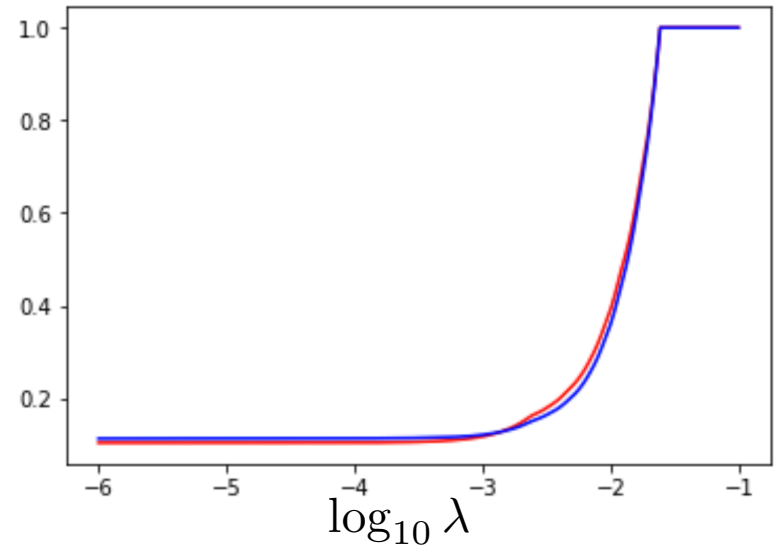
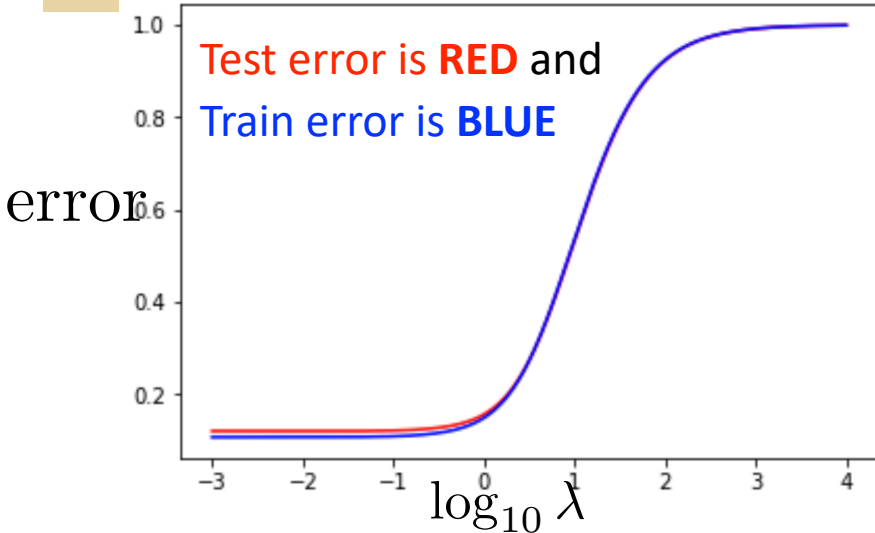
$$\|w\|_1 = \sum_{j=1}^d |w[j]|$$

absolute value

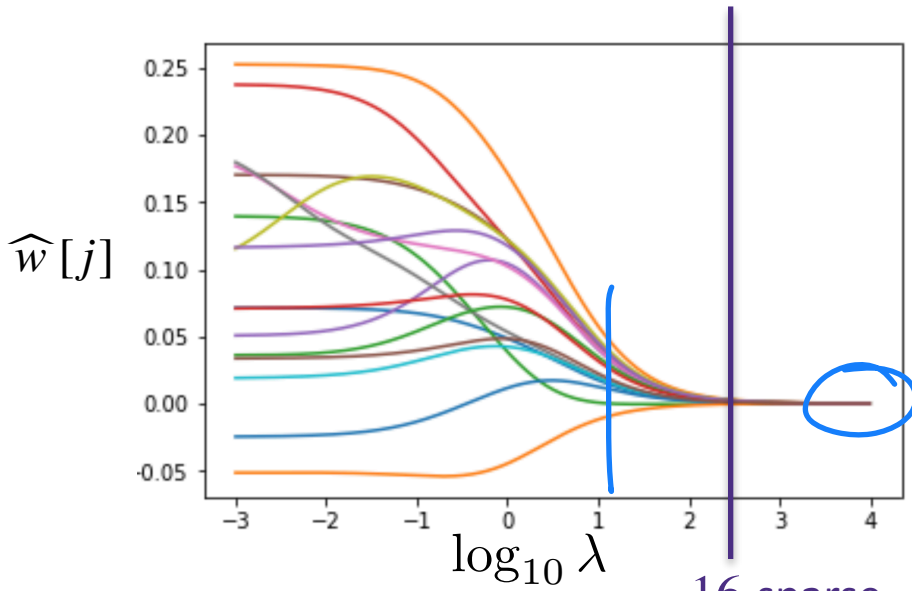
$\ell_p$ -norm of a vector  $w \in \mathbb{R}^d$  is

$$\|w\|_p \triangleq \left( \sum_{j=1}^d |w[j]|^p \right)^{1/p}$$

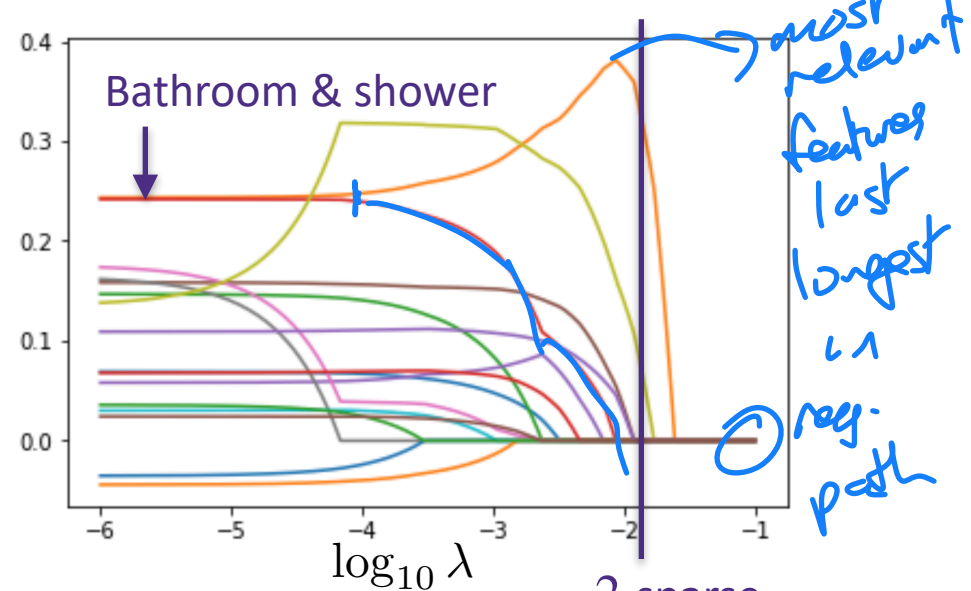
# Example: house price with 16 features



- Regularization path for LASSO shows that weights drop to exactly zero as  $\lambda$  increases



Ridge regression



Lasso regression

# LASSO regression naturally gives sparse features

- **Feature selection** with LASSO regression
  1. **Model selection**: choose  $\lambda$  based on cross validation error
  2. **Feature selection**: keep only those features with non-zero (or not-too-small) parameters in  $w$  at optimal  $\lambda$
  3. **retrain** with the sparse model and  $\lambda = 0$

why do we need to retrain?

# Example: piecewise-linear fit

$$h_0(x) = 1$$

- We use LASSO on the piece-wise linear example  $h_i(x) = [x + 1.1 - 0.1i]^+$

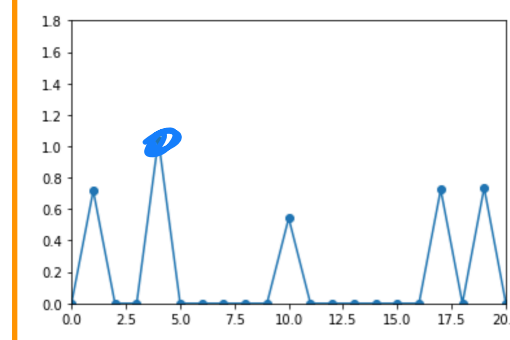
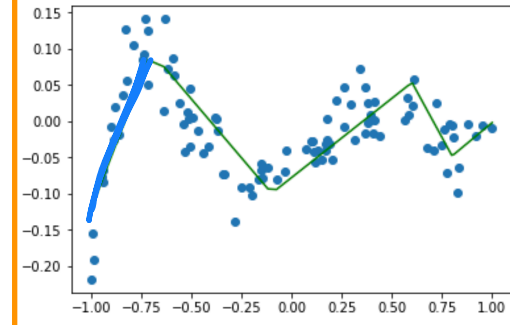
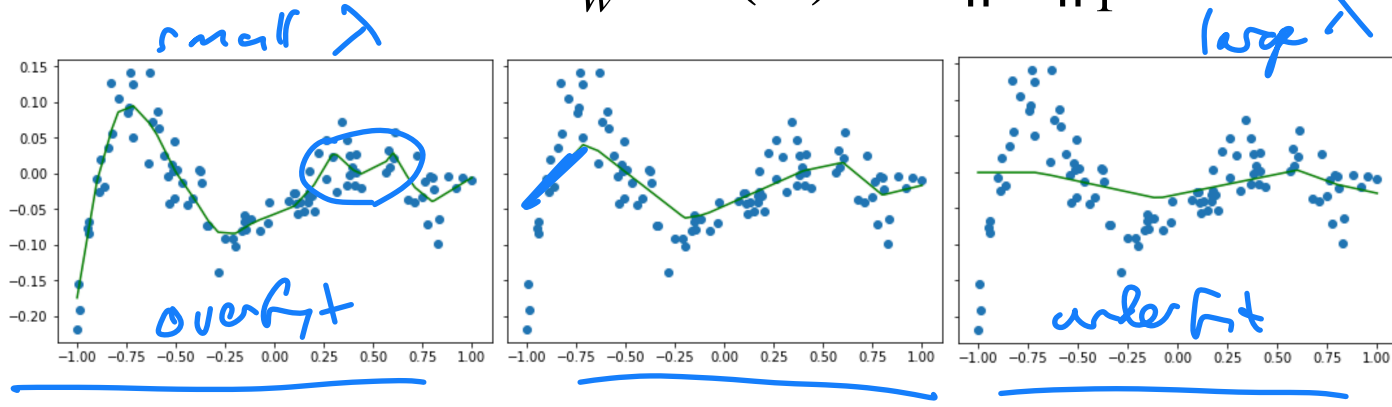
Step 1: find optimal  $\lambda^*$

pretty good  $\lambda$  by hand

$$\text{minimize}_w \mathcal{L}(w) + \lambda \|w\|_1$$

Step 3: retrain

$$\text{minimize}_w \mathcal{L}(w)$$



$$\lambda = 10^{-8}$$

$$\lambda = 10^{-4}$$

$$\lambda = 2 \times 10^{-4}$$

$$\lambda = 0$$

- de-biasing (via re-training) is critical!

but only use selected features

# Regularized Least Squares

Ridge :  $r(w) = \|w\|_2^2$

Lasso :  $r(w) = \|w\|_1$

$$\hat{w}_r = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda r(w)$$

Theorem

$\lambda$  regularized LS  $\longleftrightarrow$

$$\min \|Y - Xw\|^2 + \lambda \|w\|^2$$

$\mu$ -constrained LS

$$\begin{aligned} \min & \|Y - Xw\|^2 \\ \text{s.t.} & \|w\|^2 \leq \mu \end{aligned}$$

$\uparrow \lambda \equiv$  more constrained

# Regularized Least Squares

- Regularized optimization:

$$\hat{w}_r = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda r(w)$$

$$\text{Ridge : } r(w) = \|w\|_2^2$$

$$\text{Lasso : } r(w) = \|w\|_1$$

- For any  $\lambda^* \geq 0$  for which  $\hat{w}_r$  achieves the minimum, there exists a  $\mu^* \geq 0$  such that the solution of the constrained optimization,  $\hat{w}_c$ , is the same as the solution of the regularized optimization,  $\hat{w}_r$ , where

$$\hat{w}_c = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \quad \text{subject to } r(w) \leq \mu^*$$

- so there are pairs of  $(\lambda, \mu)$  whose optimal solution  $\hat{w}_r$  are the same for the regularized optimization and constrained optimization

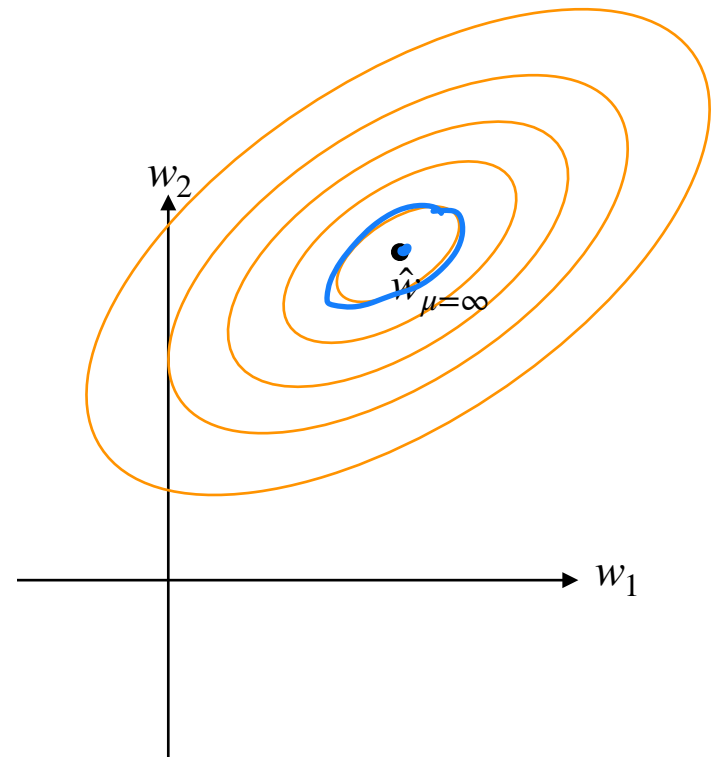
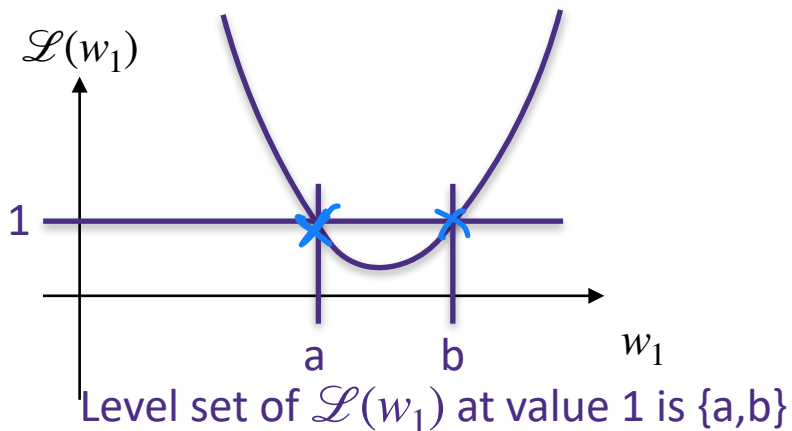
# Why does LASSO give sparse solutions?

$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

- the **level set** of a function  $\mathcal{L}(w_1, w_2)$  is defined as the set of points  $(w_1, w_2)$  that have the same function value
- the level set of a quadratic function is an oval
- the center of the oval is the least squares solution  $\hat{w}_{\mu=\infty} = \hat{w}_{\text{LS}}$

1-D example with quadratic loss



# Why does Lasso give sparse solutions?

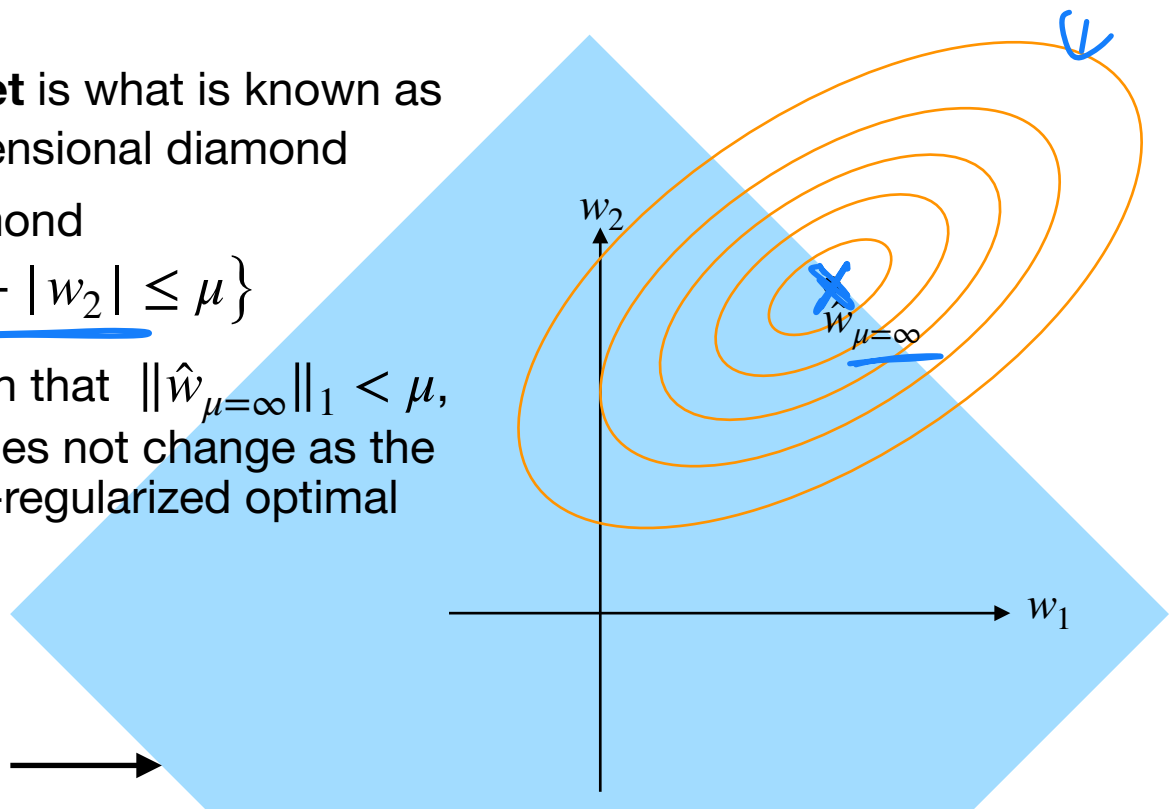
$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

- as we decrease  $\mu$  from infinity, the feasible set becomes smaller
- the shape of the **feasible set** is what is known as  $L_1$  ball, which is a high dimensional diamond
- In 2-dimensions, it is a diamond

$$\{(w_1, w_2) \mid |w_1| + |w_2| \leq \mu\}$$

- when  $\mu$  is large enough such that  $\|\hat{w}_{\mu=\infty}\|_1 < \mu$ , then the optimal solution does not change as the feasible set includes the un-regularized optimal solution



feasible set:  $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$  →

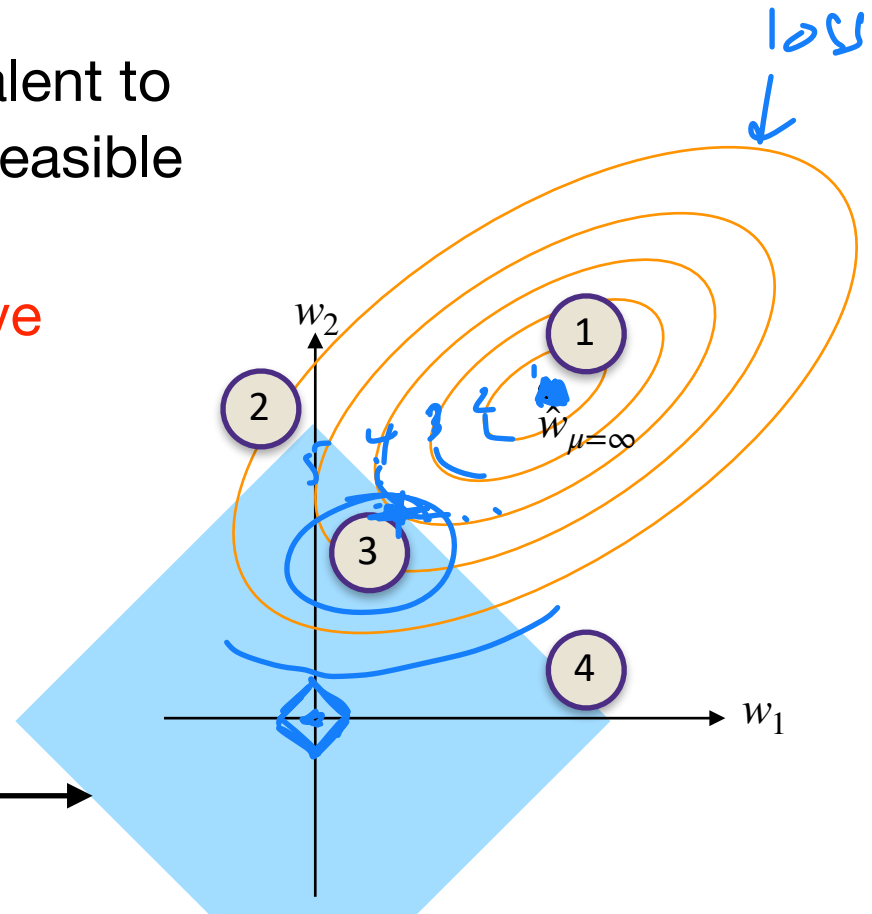
# Why does Lasso give sparse solutions?

$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

- As  $\mu$  decreases (which is equivalent to increasing regularization  $\lambda$ ) the feasible set (blue diamond) shrinks
- **The optimal solution of the above optimization is ?**

feasible set:  $\{w \in \mathbb{R}^2 \mid \|w\|_1 \leq \mu\}$  →

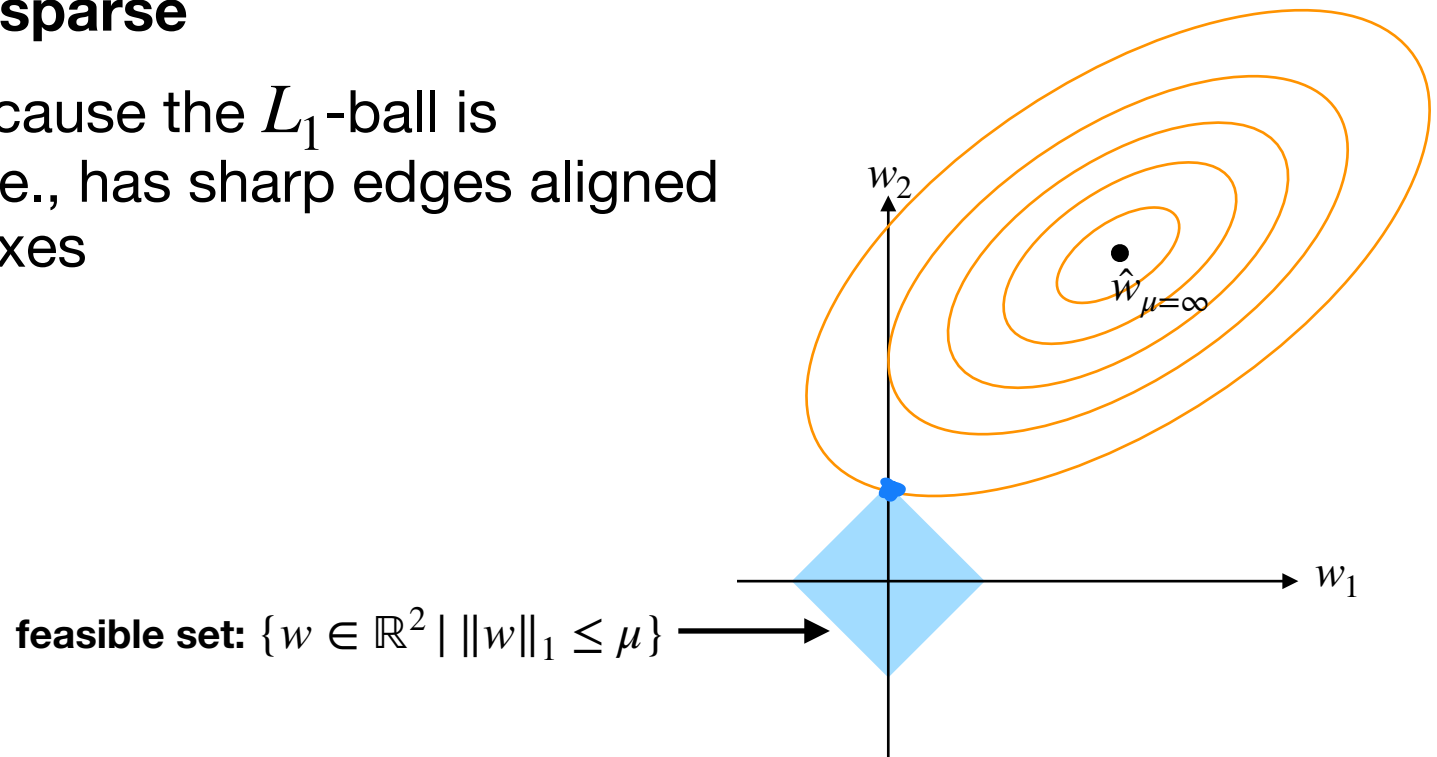


# Why does Lasso give sparse solutions?

$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

$$\text{subject to } \|w\|_1 \leq \mu$$

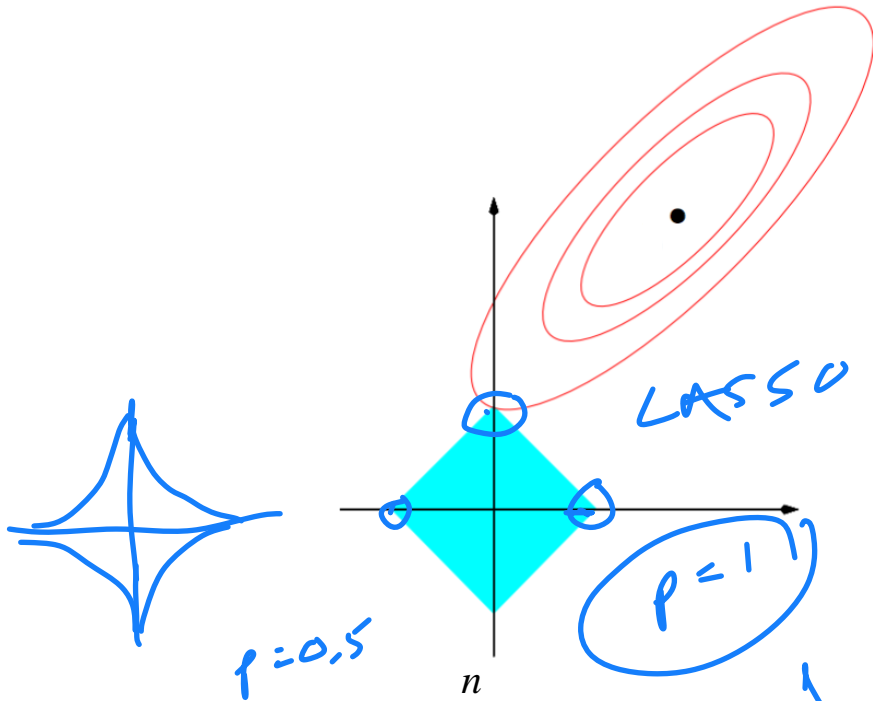
- For small enough  $\mu$ , the optimal solution becomes **sparse**
- This is because the  $L_1$ -ball is “pointy”, i.e., has sharp edges aligned with the axes



# Constrained Least Squares

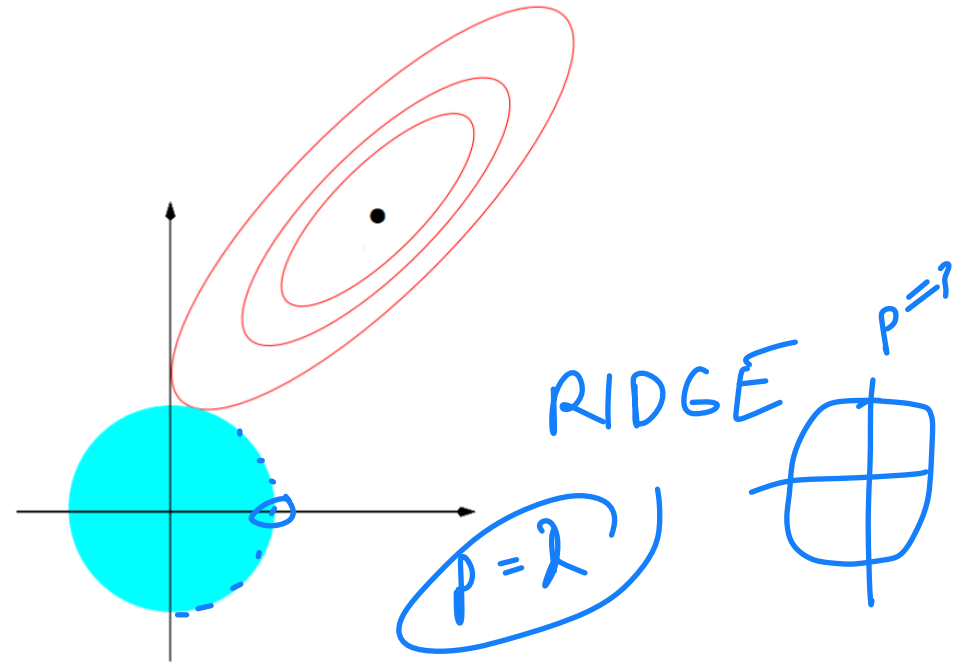
$L_p$  norm of  $\omega$   
 $\|w\|_p = \left( \sum_{j=1}^n |w_j|^p \right)^{1/p}$

- LASSO regression finds sparse solutions, as  $L_1$ -ball is “pointy”
- Ridge regression finds dense solutions, as  $L_2$ -ball is “smooth”



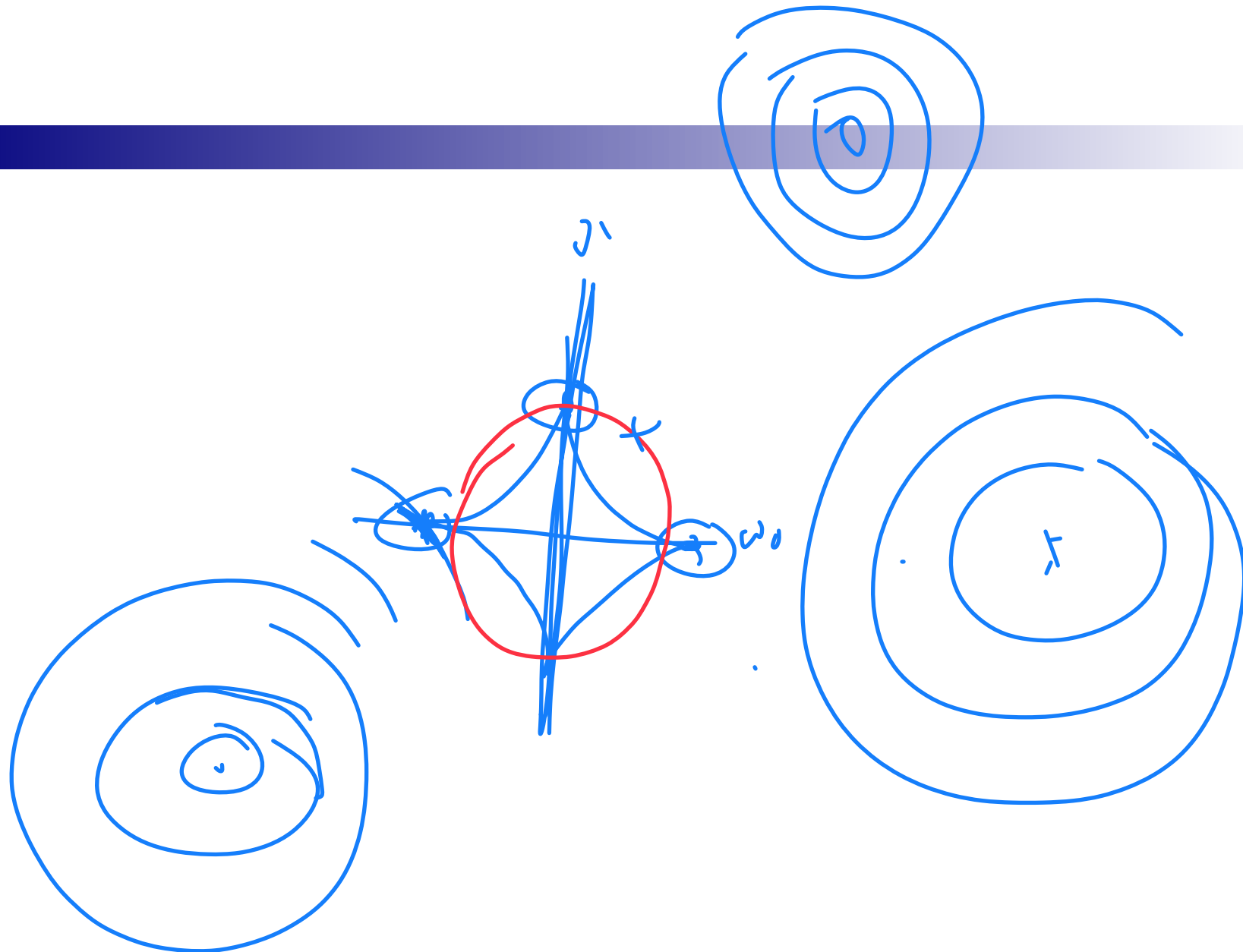
minimize<sub>w</sub>  $\sum_{i=1}^n (w^T x_i - y_i)^2$

subject to  $\|w\|_1 \leq \mu$



minimize<sub>w</sub>  $\sum_{i=1}^n (w^T x_i - y_i)^2$

subject to  $\|w\|_2 \leq \mu$



# L1 Ball in Higher Dimensions

## > L1 ball 3 dimensions

