


# Bias-Variance

---





Features	Train MSE	Test MSE
All	2640	3224
S5 and BMI	3004	3453
S5	3869	4227
BMI	3540	4277
S4 and S3	4251	5302
S4	4278	5409
S3	4607	5419
None	5524	6352

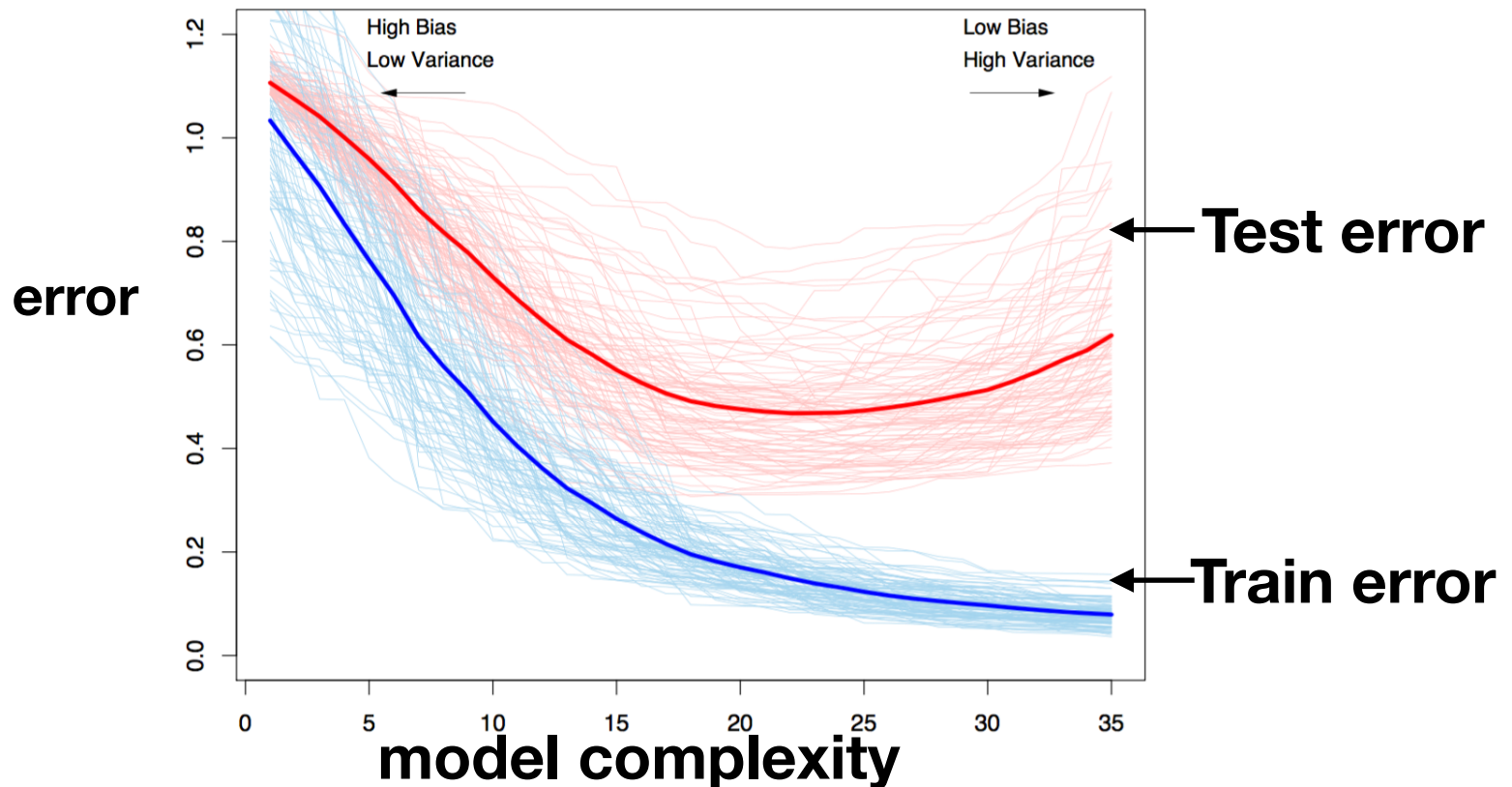
- **test MSE is the primary criteria for model selection**
- Using only 2 features (S5 and BMI), one can get very close to the prediction performance of using all features
- Combining S3 and S4 does not give any performance gain

# What does the bias-variance theory tell us?

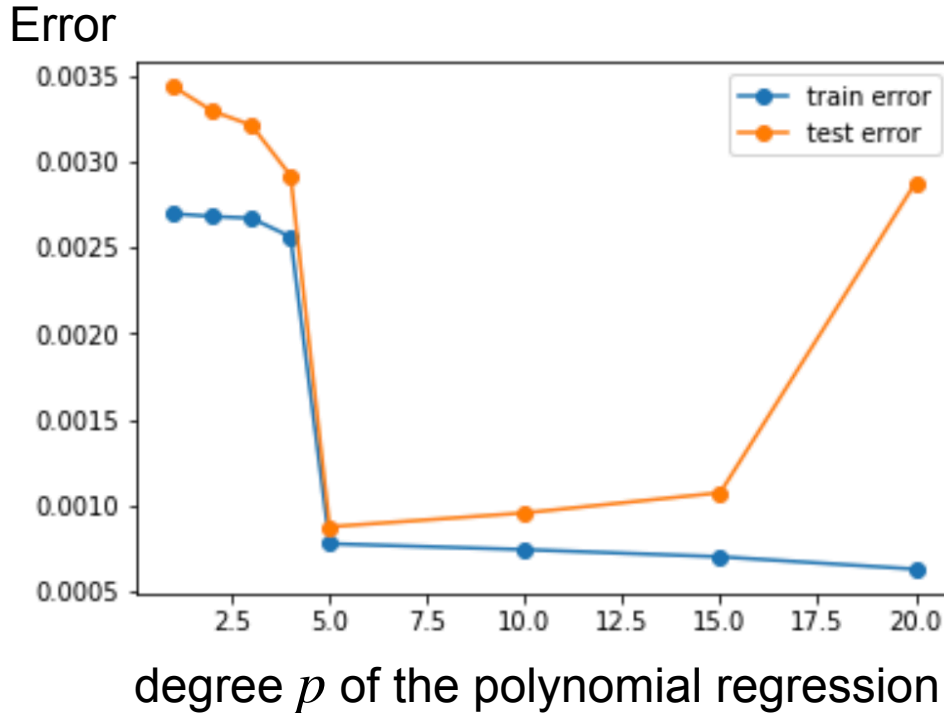
- **Train error** (random variable, randomness from  $\mathcal{D}$ )
  - Use  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \sim P_{X,Y}$  to find  $\widehat{w}$
  - Train error:  $\mathcal{L}_{\text{train}}(\widehat{w}_{\text{LS}}) = \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - \widehat{w}^T x_i)^2$
- recall the **test error** is an unbiased estimator of the **true error**
- **True error** (random variable, randomness from  $\mathcal{D}$ )
  - True error:  $\mathcal{L}_{\text{true}}(\widehat{w}) = \mathbb{E}_{(x,y) \sim P_{X,Y}} [(y - \widehat{w}^T x)^2]$
- **Test error** (random variable, randomness from  $\mathcal{D}$  and  $\mathcal{T}$ )
  - Use  $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^m \sim P_{X,Y}$
  - Test error:  $\mathcal{L}_{\text{test}}(\widehat{w}) = \frac{1}{|\mathcal{T}|} \sum_{(x_i, y_i) \in \mathcal{T}} (y_i - \widehat{w}^T x_i)^2$
- theory explains **true error**, and hence expected behavior of the (random) **test error**

# What does bias-variance theory tell us?

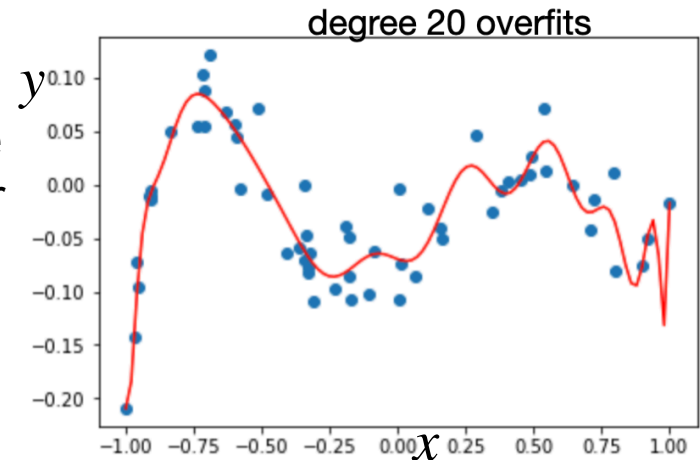
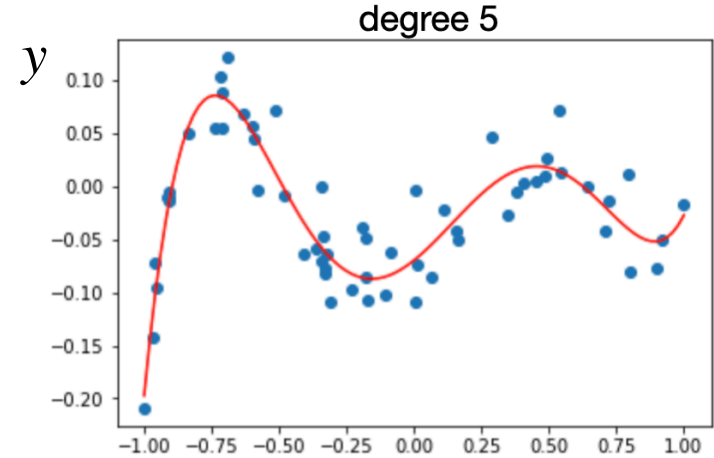
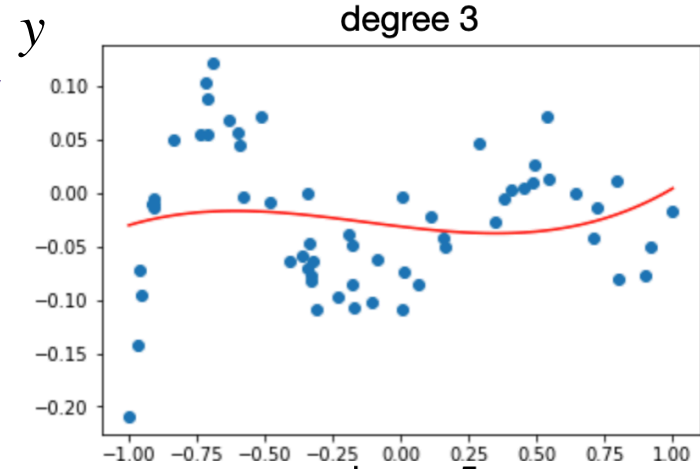
- Train error is optimistically biased (i.e. smaller) because the trained model is minimizing the train error
- Test error is unbiased estimate of the true error, if test data is never used in training a model or selecting the model complexity
- Each line is an i.i.d. instance of  $\mathcal{D}$  and  $\mathcal{T}$



# Train/test error vs. complexity



- **Model complexity** e.g., degree  $p$  of the polynomial model, number of features used in diabetes example
  - Related to the dimension of the model parameter
- **Train error** monotonically decreases with model complexity
- **Test error** has a U shape



# Statistical learning

Typical notation:

$X$  denotes a random variable

$x$  denotes a deterministic instance

- Suppose data is generated from a statistical model  $(X, Y) \sim P_{X,Y}$ 
  - and assume we know  $P_{X,Y}$  (just for now to explain statistical learning)
- **learning** aims to find a predictor  $\eta : \mathbb{R}^d \rightarrow \mathbb{R}$  that minimizes
  - expected error  $\mathbb{E}_{(X,Y) \sim P_{X,Y}}[(Y - \eta(X))^2]$
  - think of random  $(X, Y)$  as a new sample you will encounter when you deployed your learned model, and we care about its average performance
- We assume the function  $\eta(x)$  could be anything
  - it can take any value for each  $X = x$
- So the optimization can be done separately for each  $X = x$

$$\begin{aligned} \mathbb{E}_{(X,Y) \sim P_{X,Y}}[(Y - \eta(X))^2] &= \mathbb{E}_{X \sim P_X}[\mathbb{E}_{Y \sim P_{Y|X}}[(Y - \eta(x))^2 | X = x]] \\ &= \int \mathbb{E}_{Y \sim P_{Y|X}}[(Y - \eta(x))^2 | X = x] P_X(x) dx \end{aligned}$$

Or for discrete  $X$ ,

$$= \sum_x P_X(x) \mathbb{E}_{Y \sim P_{Y|X}}[(Y - \eta(x))^2 | X = x]$$

Where we used the chain rule:  $\mathbb{E}_{X,Y}[f(X, Y)] = \mathbb{E}_X[\mathbb{E}_{Y|X}[f(x, Y) | X = x]]$

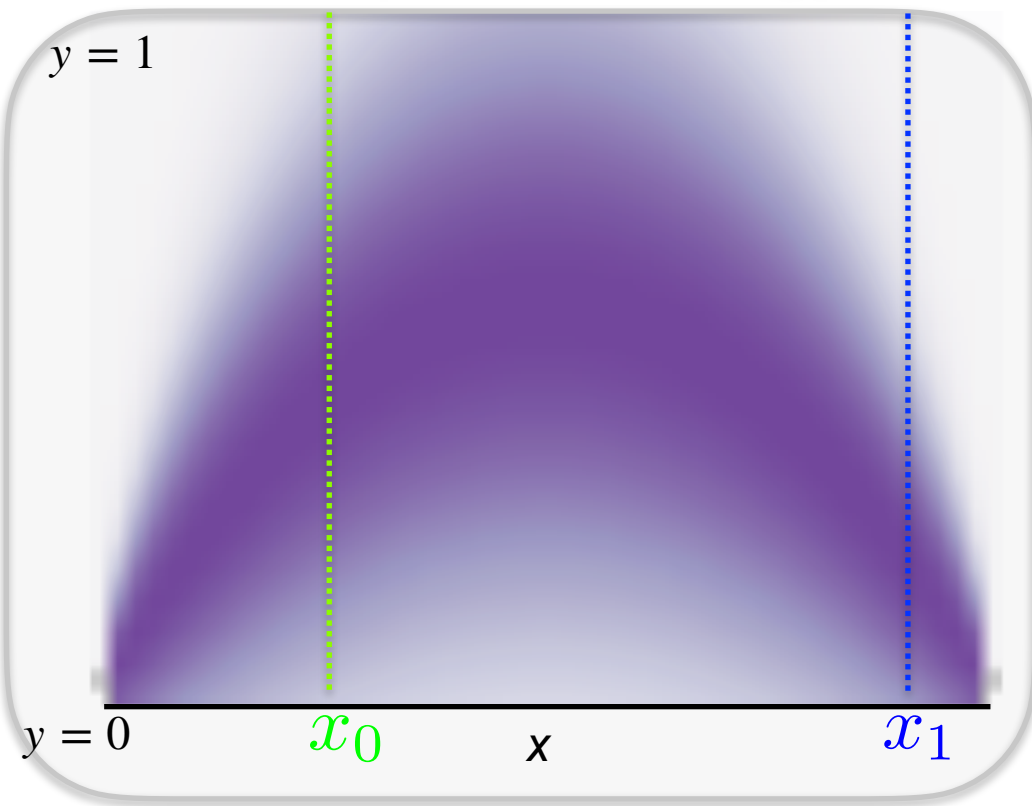
# Statistical learning

---

- The optimal predictor sets its value for each  $X = x$  separately
  - $\eta(x) = \arg \min_{a \in \mathbb{R}} \mathbb{E}_{Y \sim P_{Y|X}}[(Y - a)^2 | X = x]$
- The optimal solution is  $\eta(x) = \mathbb{E}_{Y \sim P_{Y|X}}[Y | X = x]$ ,  
which is the best prediction in  $\ell_2$ -loss/Mean Squared Error
- Claim:  $\mathbb{E}_{Y \sim P_{Y|X}}[Y | X = x] = \arg \min_{a \in \mathbb{R}} \mathbb{E}_{Y \sim P_{Y|X}}[(Y - a)^2 | X = x]$
- Proof:
  
  
  
  
  
  
  
  
  
  
- Can't implement optimal statistical estimator  $\eta(x) = \mathbb{E}[Y | X = x]$ 
  - as we do not know  $P_{X,Y}$  in practice
- This is only for the purpose of conceptual understanding

# Statistical Learning

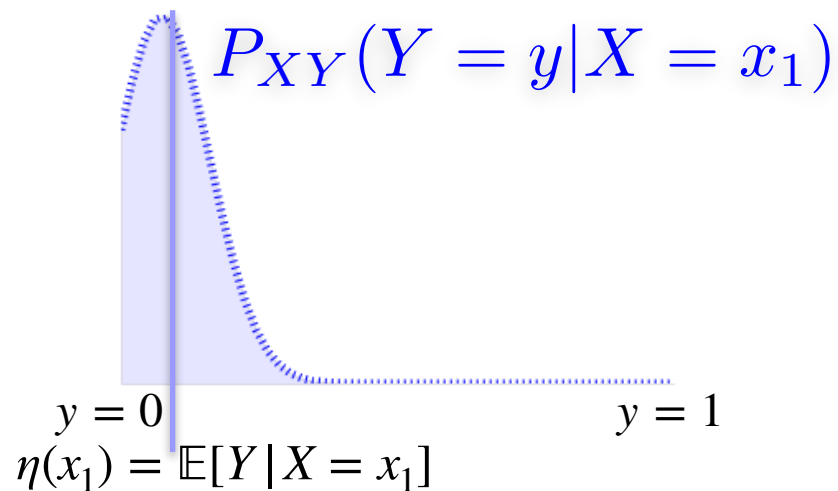
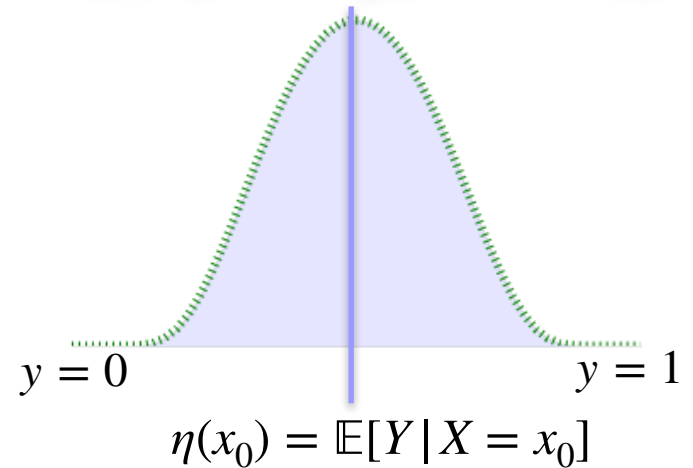
$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

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

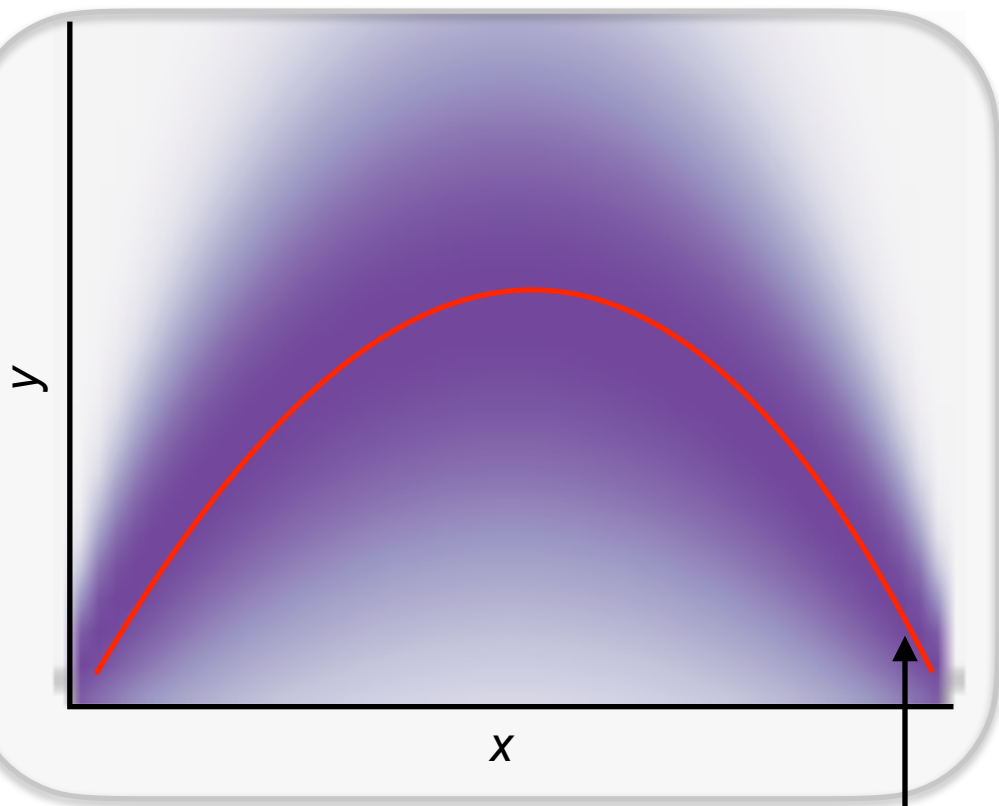
$$P_{XY}(Y = y|X = x_0)$$





# Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

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

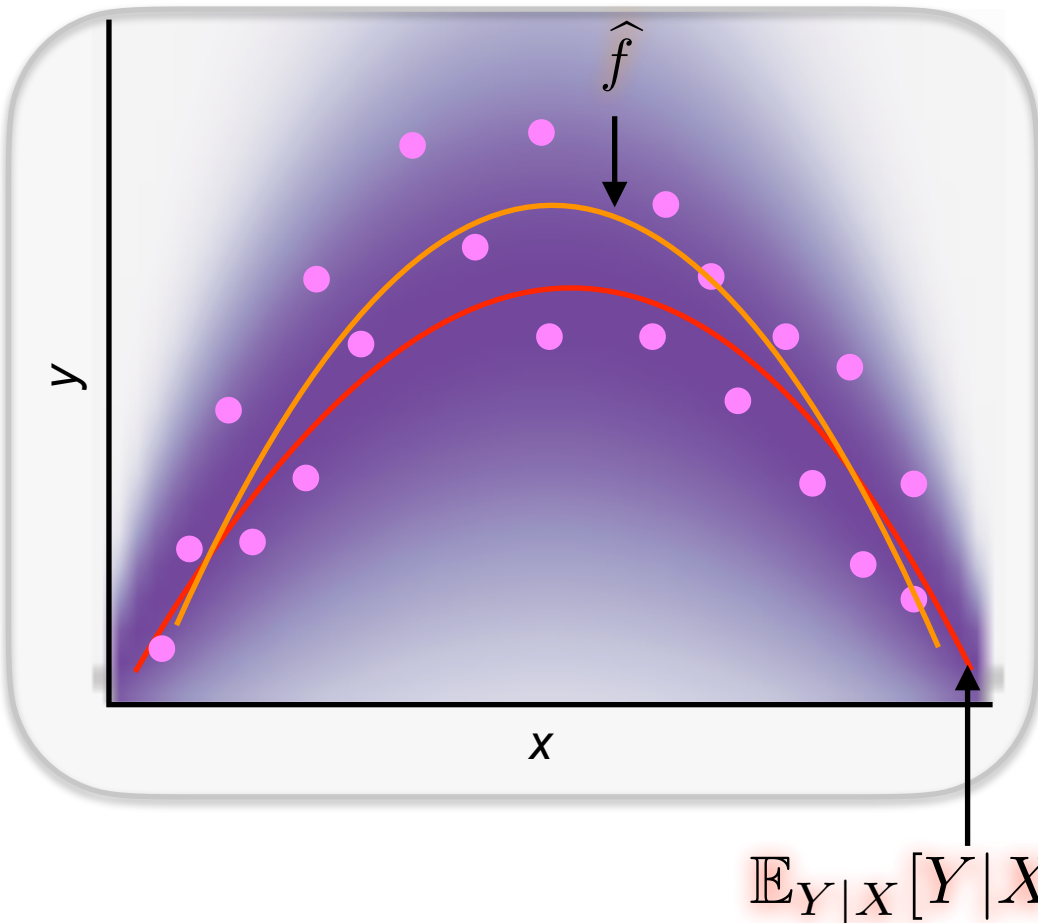
But we do not know  $P_{X,Y}$

We only have samples.

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

# Statistical Learning

$$P_{XY}(X = x, Y = y)$$



Ideally, we want to find:

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

But we only have samples:

$$(x_i, y_i) \stackrel{i.i.d.}{\sim} P_{XY} \quad \text{for } i = 1, \dots, n$$

So we need to restrict our predictor to a function class (e.g., linear, degree- $p$  polynomial) to avoid overfitting:

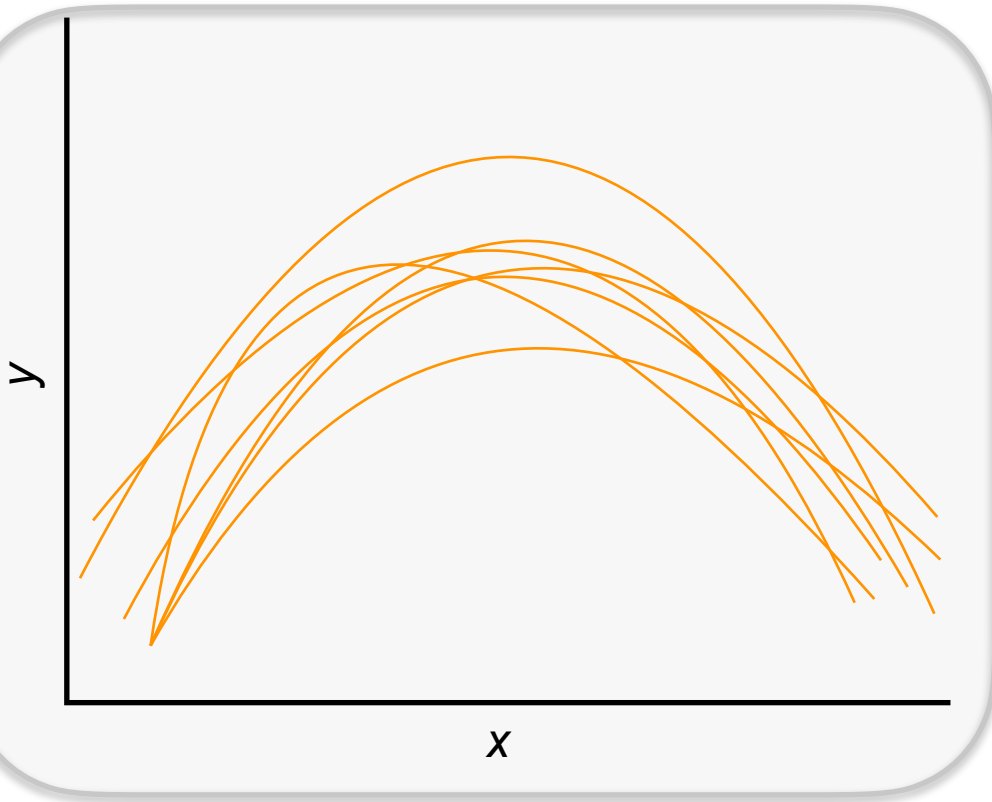
$$\hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

We care about how our predictor performs on future unseen data

$$\text{True Error of } \hat{f}: \mathbb{E}_{X,Y}[(Y - \hat{f}(X))^2]$$

Future prediction error  $\mathbb{E}_{X,Y}[(Y - \hat{f}(X))^2]$  is random  
because  $\hat{f}$  is random (whose randomness comes from training data  $\mathcal{D}$ )

$$P_{XY}(X = x, Y = y)$$



Each draw  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$  results in different  $\hat{f}$

# Bias-variance tradeoff

Notation:

I use predictor/model/estimate,  
interchangeably

## Ideal predictor

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

## Learned predictor

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

- We are interested in the **True Error** of a (random) learned predictor:

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

- But the analysis can be done for each  $X = x$  separately, so we analyze the **conditional true error**:

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

- And we care about the **average conditional true error**, averaged over training data:

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

written compactly as  $= \mathbb{E}[(Y - \hat{f}_{\mathcal{D}}(x))^2]$

# Bias-variance tradeoff

## Ideal predictor

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

## Learned predictor

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

- **Average conditional true error:**

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

# Bias-variance tradeoff

## Ideal predictor

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

## Learned predictor

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

- **Average conditional true error:**

$$\begin{aligned} \mathbb{E}_{\mathcal{D}, Y|x} [(Y - \hat{f}_{\mathcal{D}}(x))^2] &= \mathbb{E}_{\mathcal{D}, Y|x} [(Y - \eta(x) + \eta(x) - \hat{f}_{\mathcal{D}}(x))^2] \\ &= \mathbb{E}_{\mathcal{D}, Y|x} \left[ (Y - \eta(x))^2 + 2(Y - \eta(x))(\eta(x) - \hat{f}_{\mathcal{D}}(x)) + (\eta(x) - \hat{f}_{\mathcal{D}}(x))^2 \right] \\ &= \mathbb{E}_{Y|x} [(Y - \eta(x))^2] + \underbrace{2\mathbb{E}_{\mathcal{D}, Y|x} [(Y - \eta(x))(\eta(x) - \hat{f}_{\mathcal{D}}(x))]}_{=0} + \mathbb{E}_{\mathcal{D}} [(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2] \end{aligned}$$

(this follows from independence of  $\mathcal{D}$  and  $(X, Y)$  and

$$\mathbb{E}_{Y|x} [Y - \eta(x)] = \mathbb{E}[Y | X = x] - \eta(x) = 0$$

$$= \underbrace{\mathbb{E}_{Y|x} [(Y - \eta(x))^2]}_{\text{Irreducible error}} + \underbrace{\mathbb{E}_{\mathcal{D}} [(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2]}_{\text{Average learning error}}$$

### Irreducible error

- (a) Caused by stochastic label noise in  $P_{Y|X=x}$
- (b) cannot be reduced

### Average learning error

- Caused by
- (a) either using too “simple” of a model or
- (b) not enough data to learn the model accurately

# Bias-variance tradeoff

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

• **Average learning error:**

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

# Bias-variance tradeoff

---

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$



# Bias-variance tradeoff

---

## Ideal predictor

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

## Learned predictor

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

- Average learning error:

# Bias-variance tradeoff

---

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

• **Average learning error:**

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

# Bias-variance tradeoff

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

• **Average learning error:**

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2] &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x) \right)^2 \right] \\ &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 + 2(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)) \right] \end{aligned}$$

# Bias-variance tradeoff

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

• **Average learning error:**

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2] &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x) \right)^2 \right] \\ &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 + 2(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)) \right. \\ &\quad \left. + (\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2 \right] \end{aligned}$$

# Bias-variance tradeoff

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

• **Average learning error:**

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2] &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x) \right)^2 \right] \\ &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 + 2(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)) \right. \\ &\quad \left. + (\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2 \right] \\ &= \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 + \mathbb{E}_{\mathcal{D}} \left[ \left( \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x) \right)^2 \right] \end{aligned}$$

# Bias-variance tradeoff

**Ideal predictor**

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

**Learned predictor**

$$\hat{f}_{\mathcal{D}} = \arg \min_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} (y_i - f(x_i))^2$$

• **Average learning error:**

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[(\eta(x) - \hat{f}_{\mathcal{D}}(x))^2] &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] + \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x) \right)^2 \right] \\ &= \mathbb{E}_{\mathcal{D}} \left[ \left( \eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 + 2(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x)) \right. \\ &\quad \left. + (\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2 \right] \end{aligned}$$

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

**biased squared**

**variance**

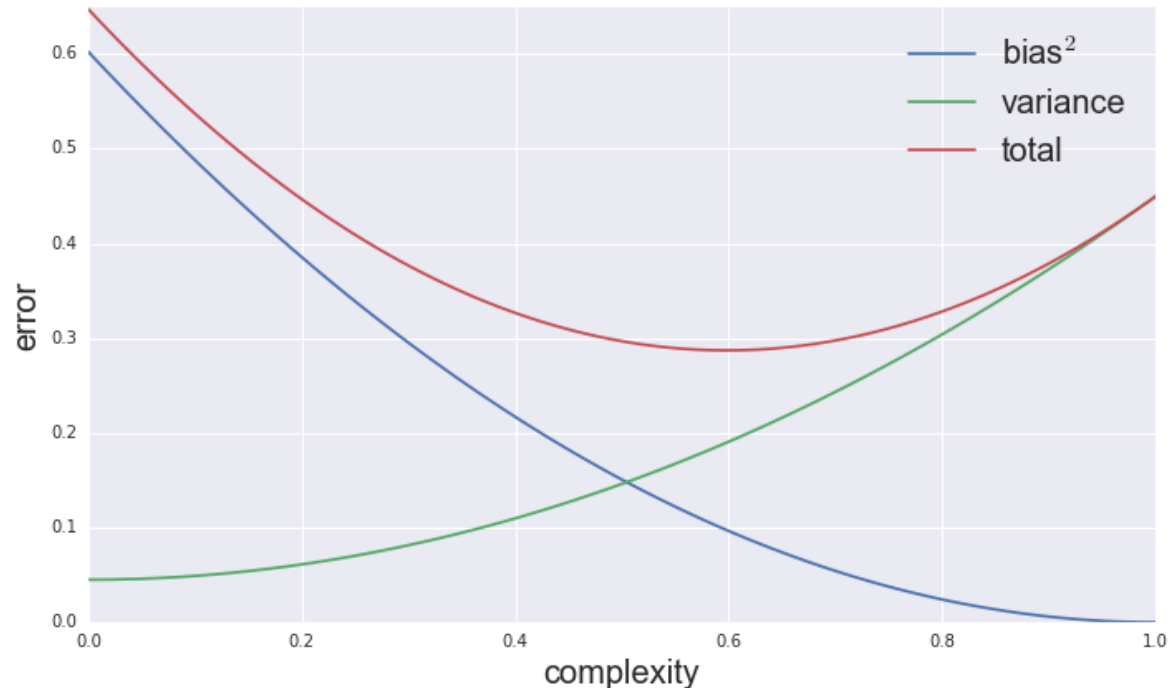
# Bias-variance tradeoff

- Average conditional true error:

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

**Bias squared:**  
measures how the predictor is mismatched with the best predictor in expectation

**variance:**  
measures how the predictor varies each time with a new training datasets



# Questions?

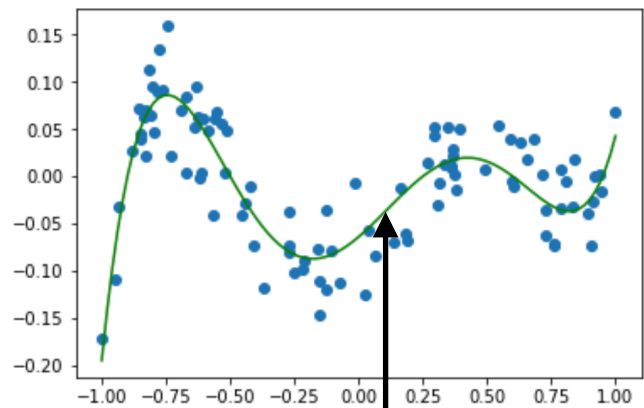
---



# Test error vs. model complexity

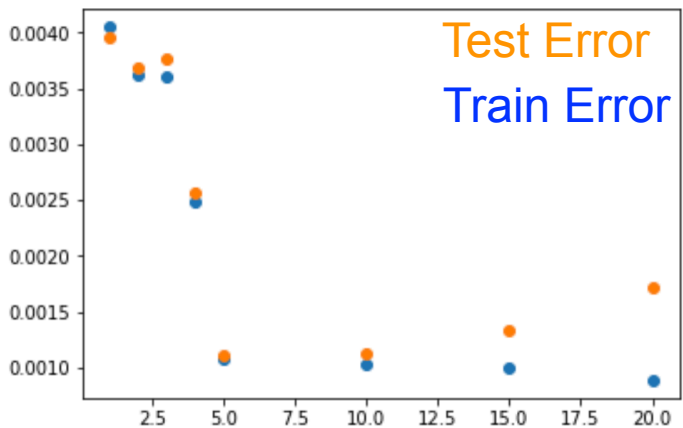
Simple model:  
Model complexity is below  
the complexity of  $\eta(x)$

Complex model:

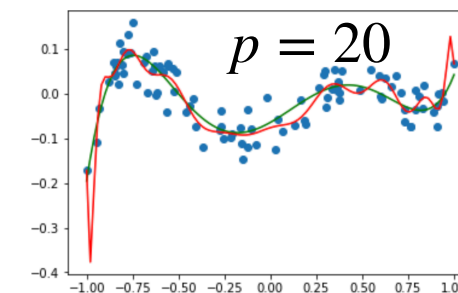
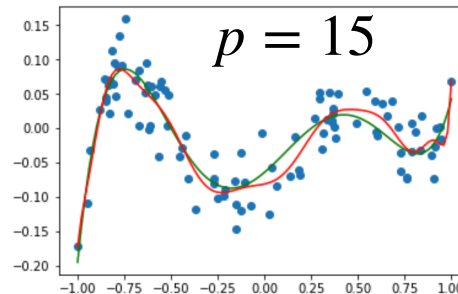
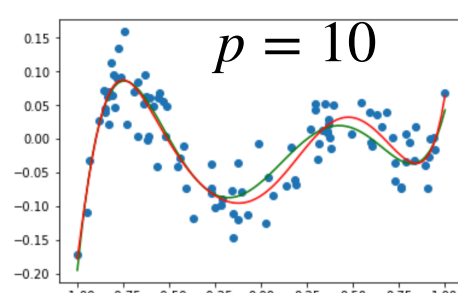
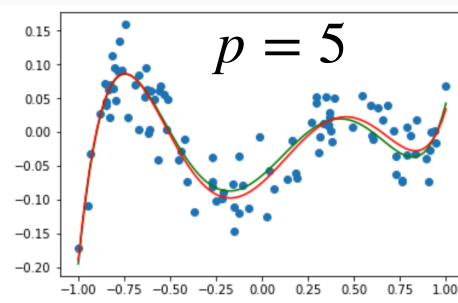
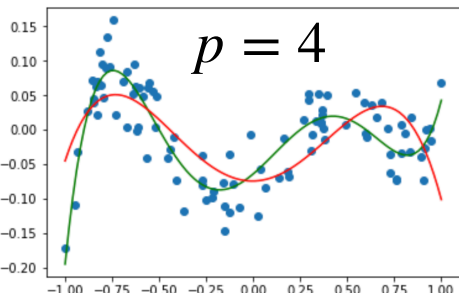
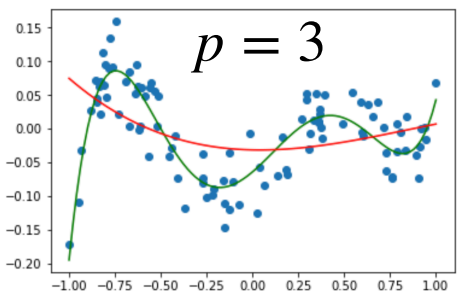
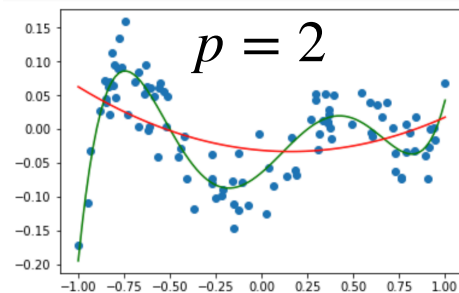
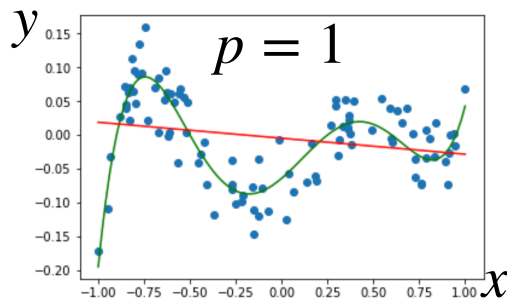


Optimal predictor  $\eta(x)$   
is degree-5 polynomial

Error

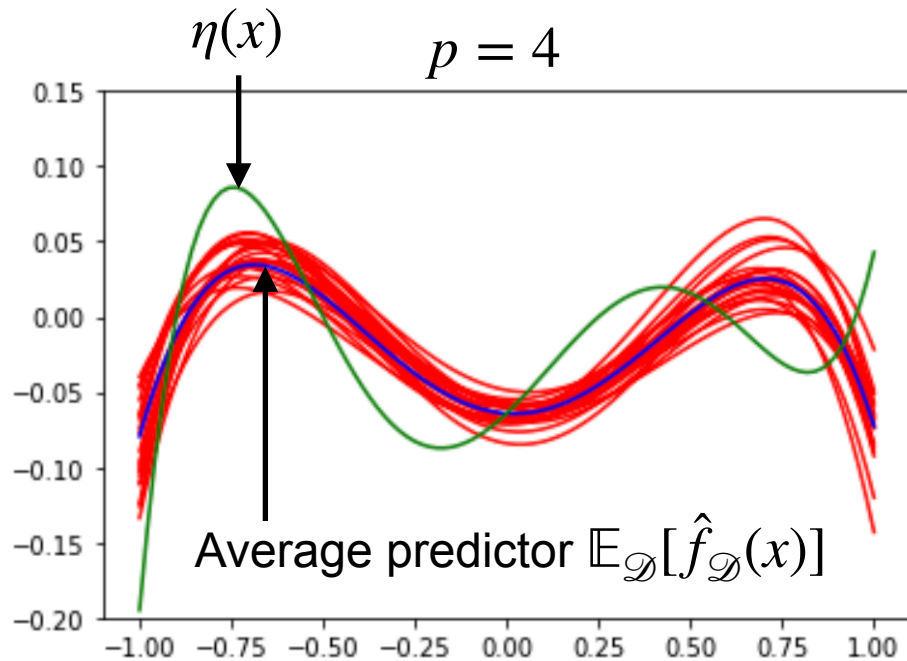
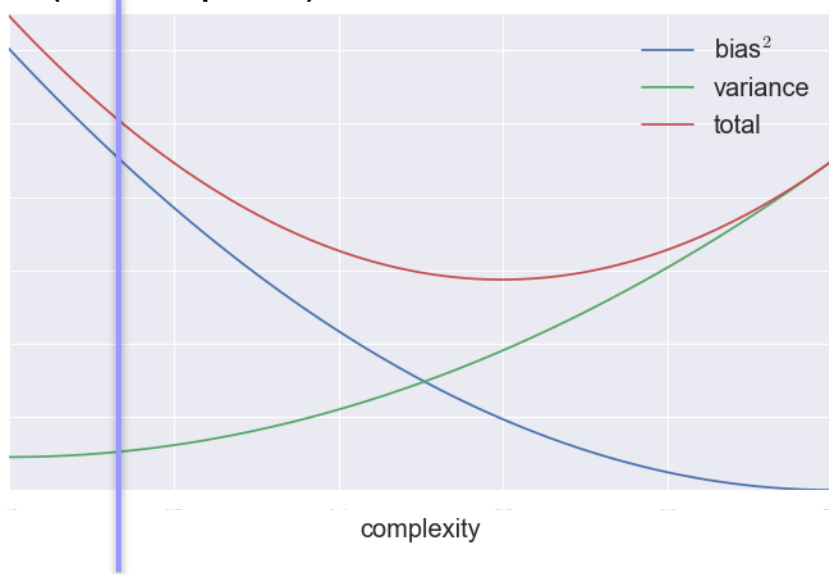


degree  $p$  of the polynomial regression



# Recap: Bias-variance tradeoff with simple model

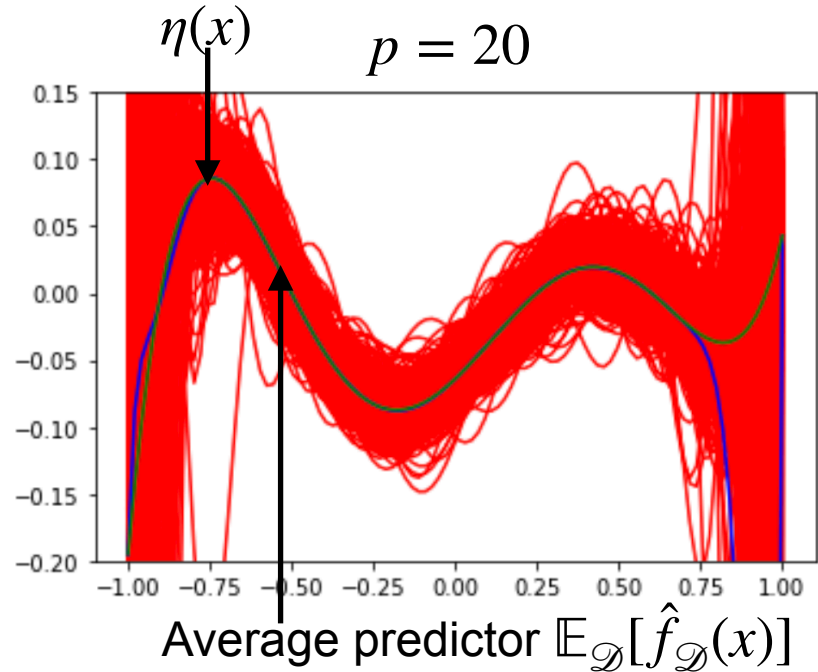
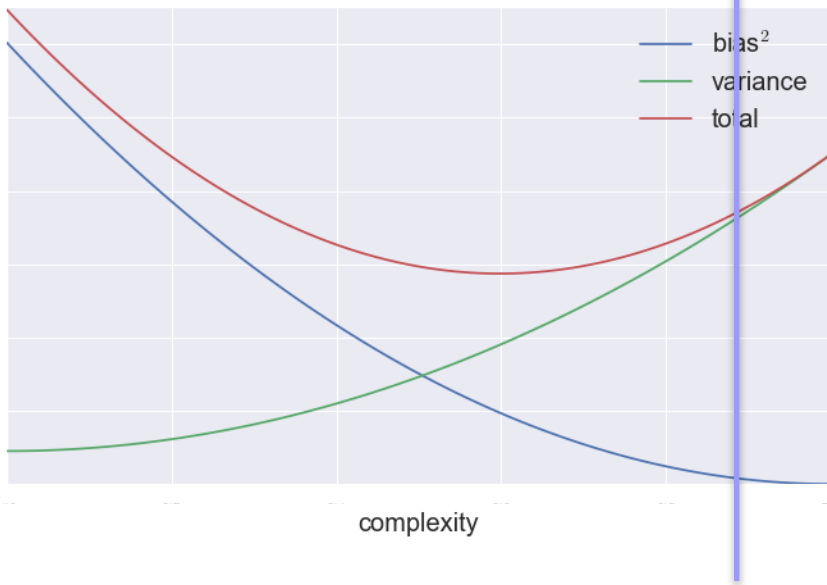
(Conceptual) bias variance tradeoff



- 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, then
    - Bias
    - Variance
    - Because Variance is already small, overall test error

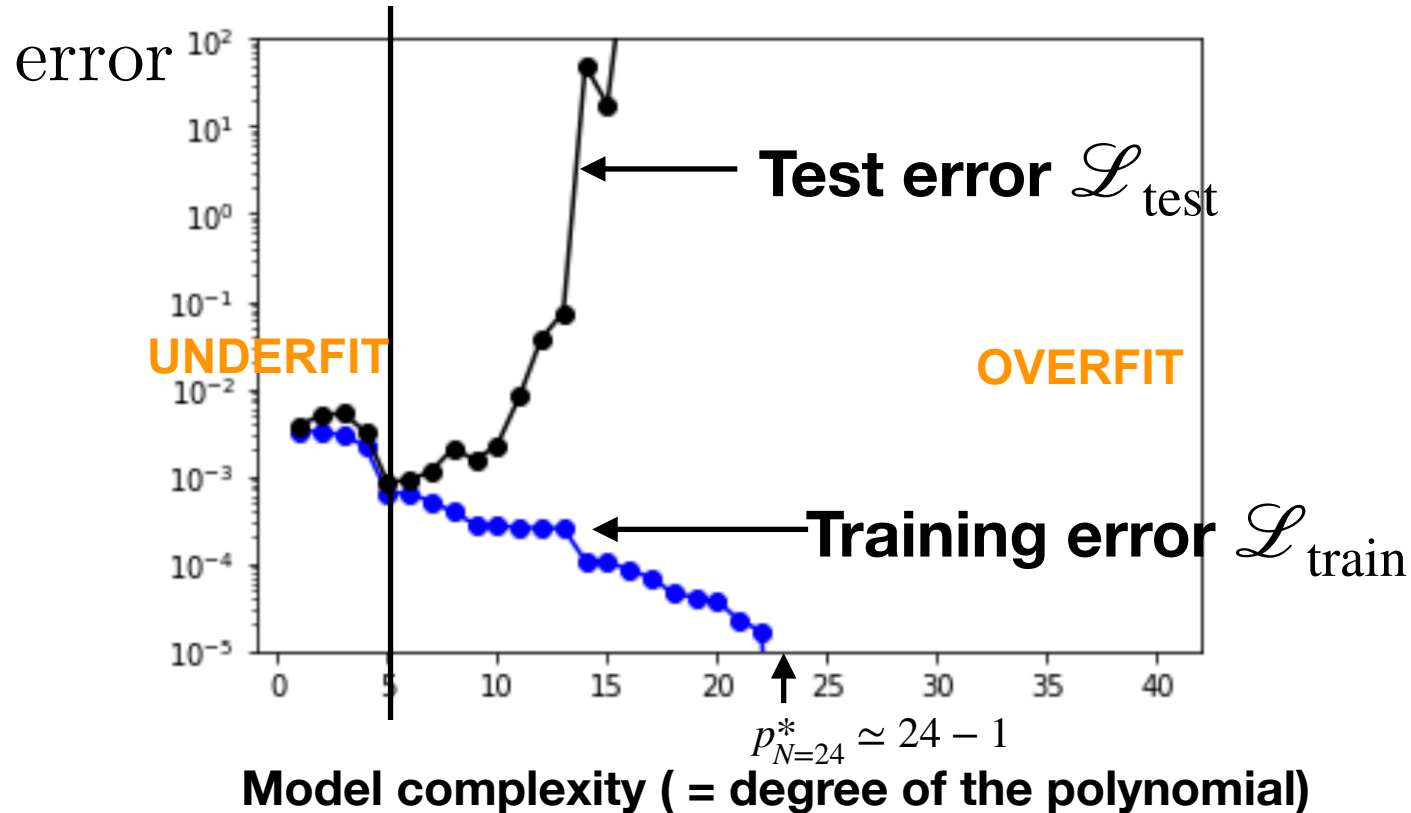
# Recap: Bias-variance tradeoff with simple model

(Conceptual) bias variance tradeoff



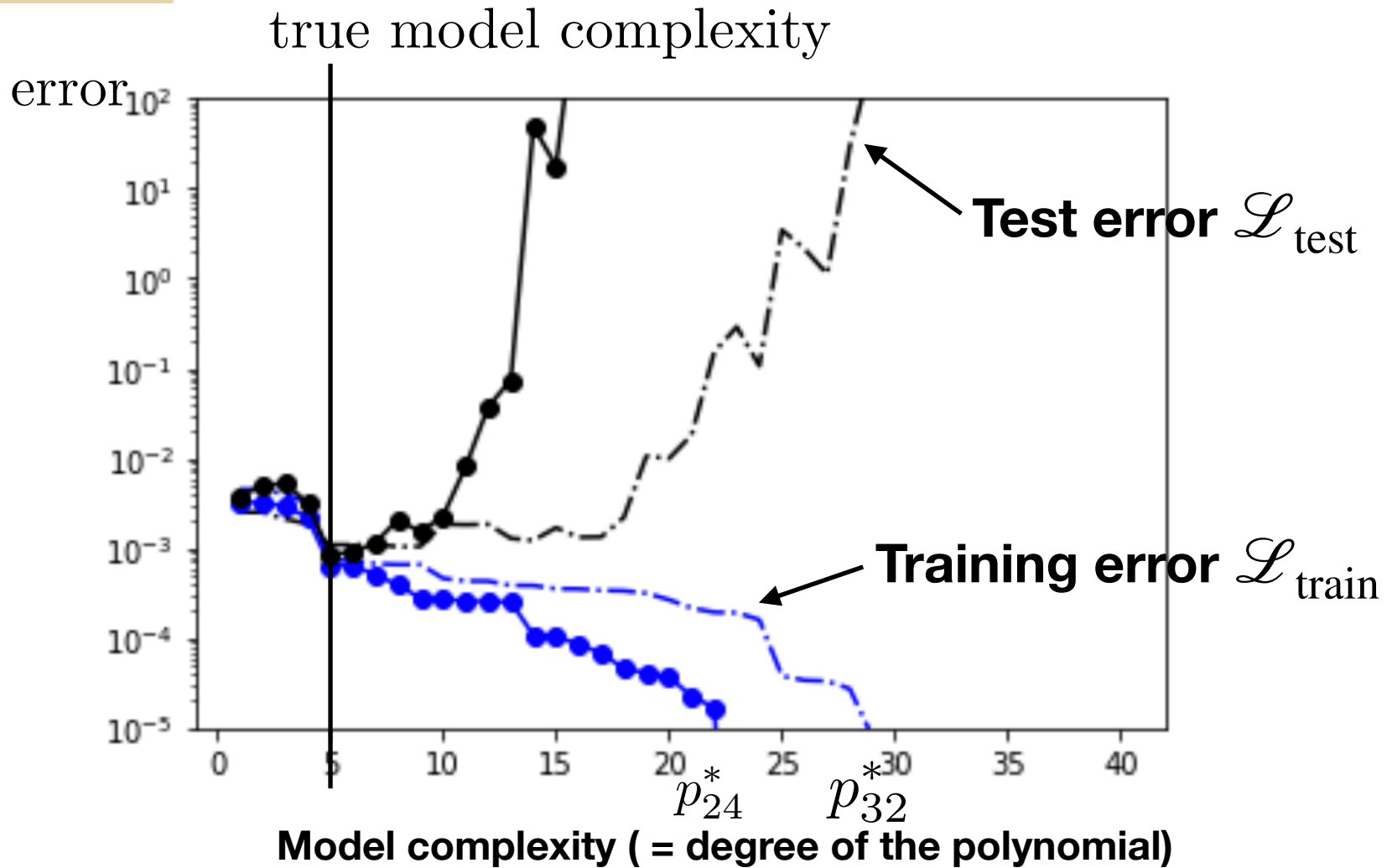
- When model complexity is high (higher than the optimal predictor  $\eta(x)$ )
  - Bias of our predictor,  $(\eta(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)])^2$ , is small
  - Variance of our predictor,  $\mathbb{E}_{\mathcal{D}}[(\mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - \hat{f}_{\mathcal{D}}(x))^2]$ , is large
  - If we have more samples, then
    - Bias
    - Variance
    - Because Variance is dominating, overall test error

- let us first fix sample size  $N=30$ , collect one dataset of size  $N$  i.i.d. from a distribution, and fix one training set  $S_{\text{train}}$  and test set  $S_{\text{test}}$  via 80/20 split
  - then we run multiple validations and plot the computed MSEs for all values of  $p$  that we are interested in
- true model complexity



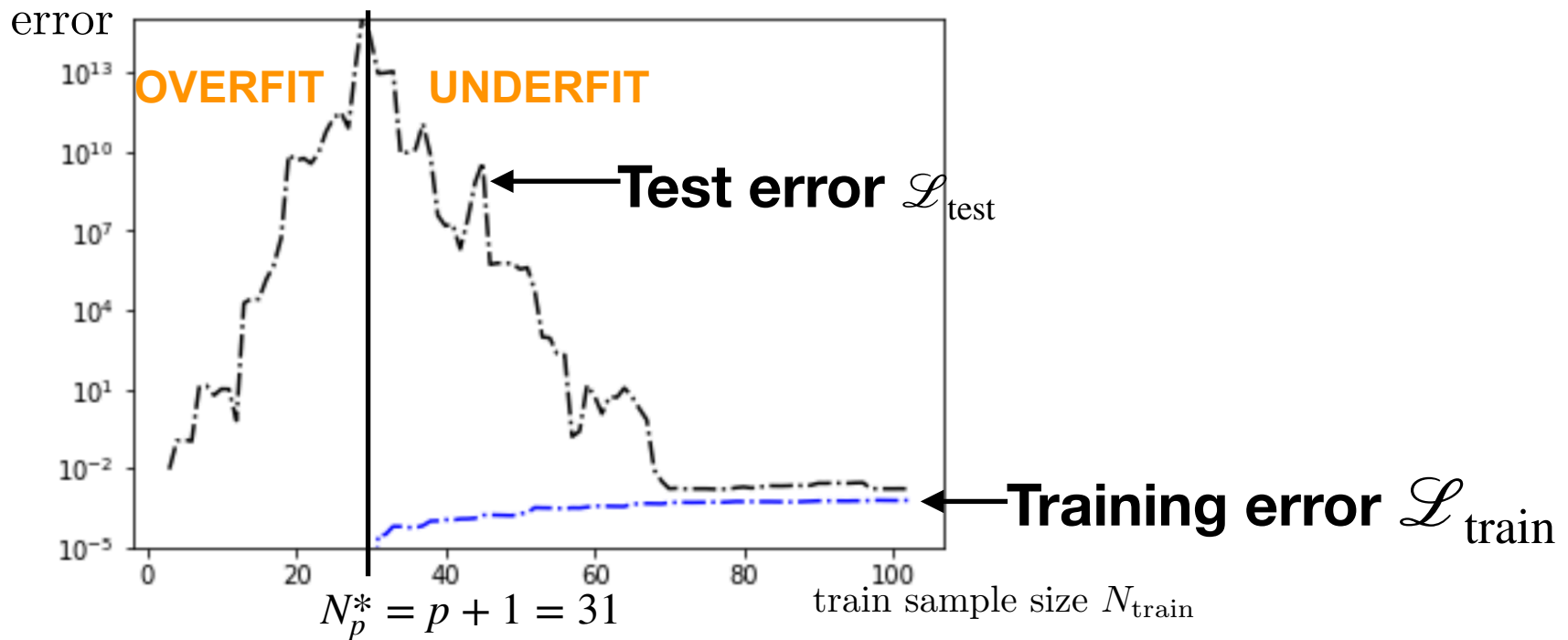
- Given sample size  $N$  there is a threshold,  $p_N^*$ , where training error is zero
- Training error is **always** monotonically non-increasing
- Test error has a trend of going down and then up, but fluctuates

- let us now repeat the process changing the sample size to **N=40**, and see how the curves change



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

- let us now fix predictor model complexity  $p=30$ , collect multiple datasets by starting with 3 samples and adding one sample at a time to the training set, but keeping a large enough test set fixed
- then we plot the computed MSEs for all values of train sample size  $N_{train}$  that we are interested in



- There is a threshold,  $N_p^*$ , below which training error is zero (extreme overfit)
- Below this threshold, test error is meaningless, as we are overfitting and there are multiple predictors with zero training error some of which have very large test error
- Test error tends to decrease
- Training error tends to increase

# Bias-variance tradeoff for linear models

If  $Y_i = X_i^T w^* + \epsilon_i$  and  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

$$\mathbf{y} = \mathbf{X}w^* + \epsilon$$

$$\widehat{w}_{\text{MLE}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} =$$
$$=$$

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

$$\hat{f}_{\mathcal{D}}(x) = x^T \widehat{w}_{\text{MLE}} =$$

# Bias-variance tradeoff for linear models

If  $Y_i = X_i^T w^* + \epsilon_i$  and  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

$$\mathbf{y} = \mathbf{X}w^* + \epsilon$$

$$\begin{aligned}\widehat{w}_{\text{MLE}} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{X}w^* + \epsilon) \\ &= w^* + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon\end{aligned}$$

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

$$\widehat{f}_{\mathcal{D}}(x) = x^T \widehat{w}_{\text{MLE}} = x^T w^* + x^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

- Irreducible error:  $\mathbb{E}_{X,Y}[(Y - \eta(x))^2 | X = x] =$
- Bias squared:  $(\eta(x) - \mathbb{E}_{\mathcal{D}}[\widehat{f}_{\mathcal{D}}(x)])^2 =$   
(is independent of the sample size!)



# Bias-variance tradeoff for linear models

If  $Y_i = X_i^T w^* + \epsilon_i$  and  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

$$\widehat{w}_{\text{MLE}} = w^* + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\eta(x) = x^T w^*$$

$$\hat{f}_{\mathcal{D}}(x) = x^T w^* + x^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

- Variance:  $\mathbb{E}_{\mathcal{D}} \left[ \left( \hat{f}_{\mathcal{D}}(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 \right] =$

# Bias-variance tradeoff for linear models

If  $Y_i = X_i^T w^* + \epsilon_i$  and  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

$$\widehat{w}_{\text{MLE}} = w^* + (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

$$\eta(x) = x^T w^*$$

$$\hat{f}_{\mathcal{D}}(x) = x^T w^* + x^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon$$

- Variance:  $\mathbb{E}_{\mathcal{D}} \left[ \left( \hat{f}_{\mathcal{D}}(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] \right)^2 \right] = \mathbb{E}_{\mathcal{D}} [x^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \epsilon \epsilon^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x]$   
 $= \sigma^2 \mathbb{E}_{\mathcal{D}} [x^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} x]$   
 $= \sigma^2 x^T \mathbb{E}_{\mathcal{D}} [(\mathbf{X}^T \mathbf{X})^{-1}] x$
- To analyze this, let's assume that  $X_i \sim \mathcal{N}(0, \mathbf{I})$  and number of samples,  $n$ , is large enough such that  $\mathbf{X}^T \mathbf{X} = n\mathbf{I}$  with high probability and  $\mathbb{E}[(\mathbf{X}^T \mathbf{X})^{-1}] \simeq \frac{1}{n} \mathbf{I}$ , then
  - Variance is  $\frac{\sigma^2 x^T x}{n}$ , and decreases with increasing sample size  $n$

# Regularization

---

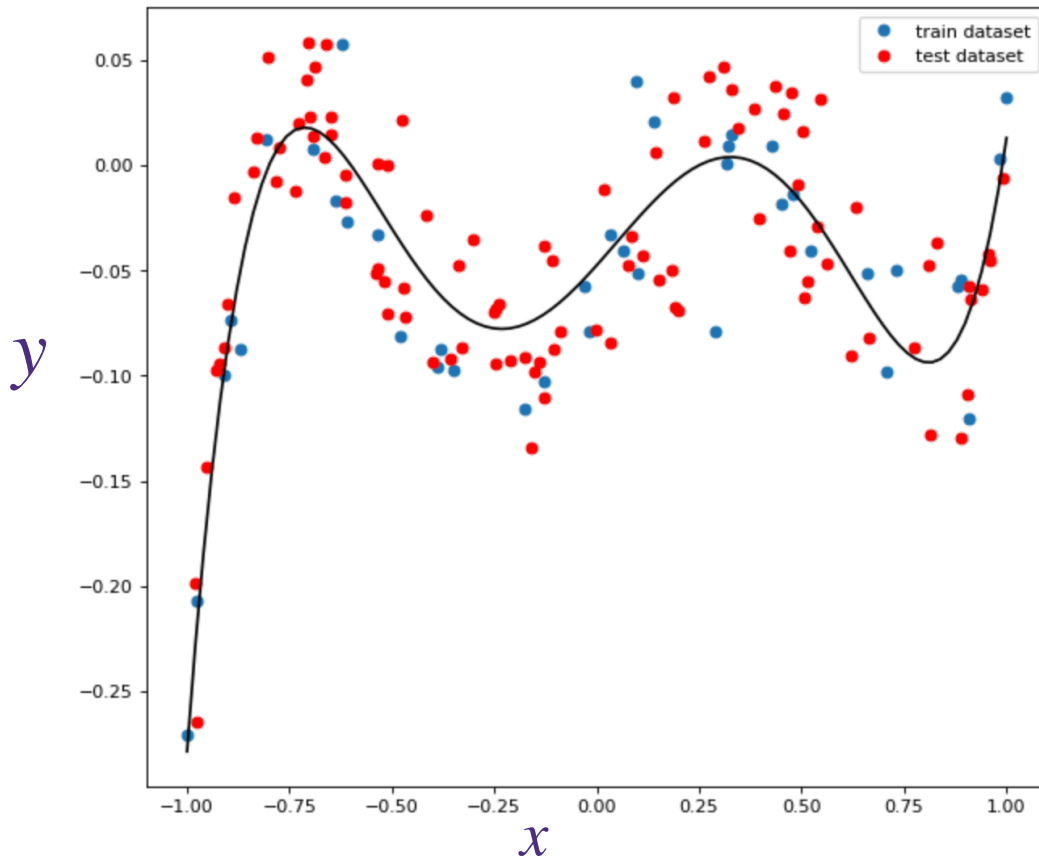


# Recap: bias-variance tradeoff

- Consider 100 training examples and 100 test examples i.i.d. drawn from degree-5 polynomial features

$$x_i \sim \text{Uniform}[-1, 1], y_i \sim f_{w^*}(x_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

$$f_w(x_i) = b^* + w_1^* x_i + w_2^* (x_i)^2 + w_3^* (x_i)^3 + w_4^* (x_i)^4 + w_5^* (x_i)^5$$

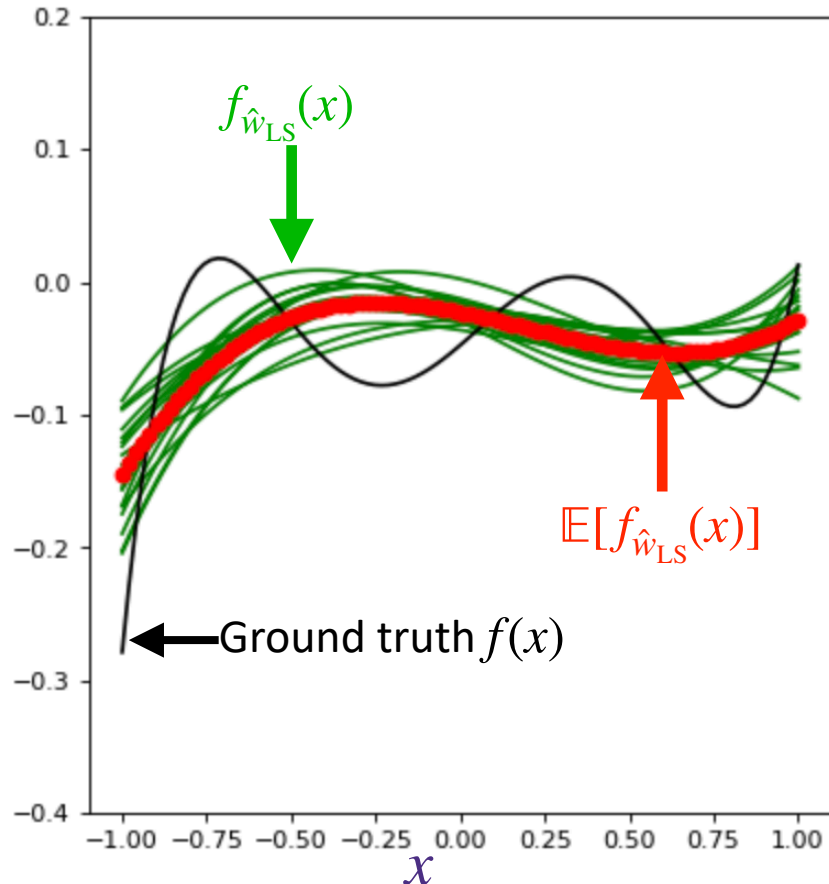


This is a linear model with features  $h(x_i) = (x_i, (x_i)^2, (x_i)^3, (x_i)^4, (x_i)^5)$

# Recap: bias-variance tradeoff

With degree-3 polynomials, we underfit

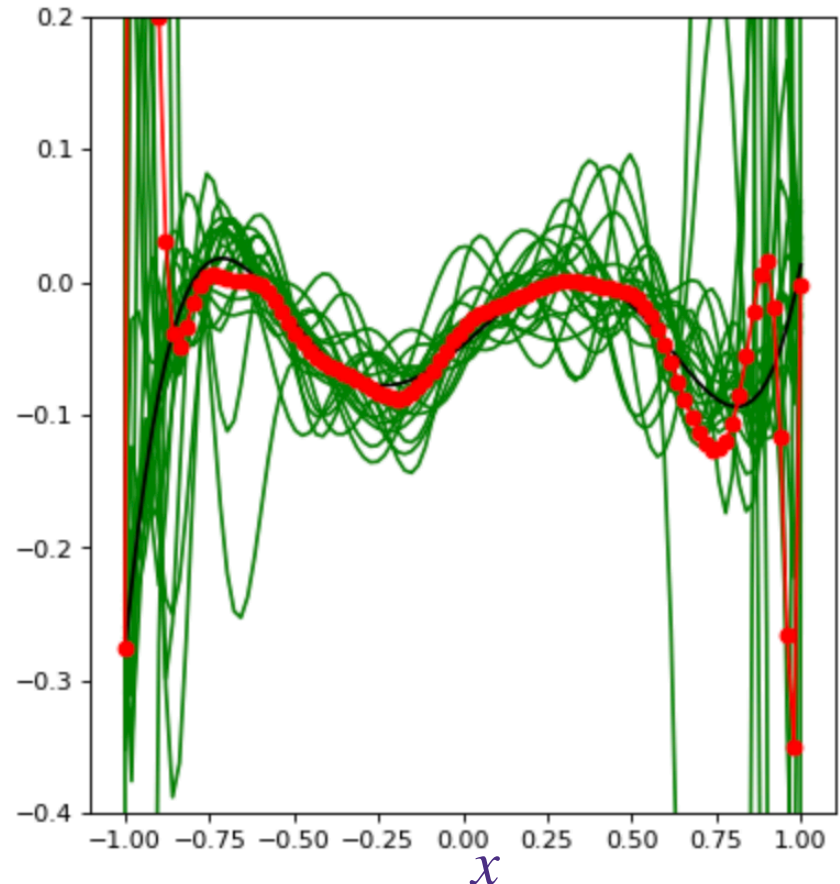
$$\hat{f}_{\hat{w}_{LS}}(x)$$



current train error = 0.0036791644380554187  
current test error = 0.0037962529988410953

With degree-20 polynomials, we overfit

$$\hat{f}_{\hat{w}_{LS}}(x)$$



0.0005421686349568773  
0.14210029429557927

# Sensitivity: how to detect overfitting

- 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  $x_j$
- 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  $b$
- 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)

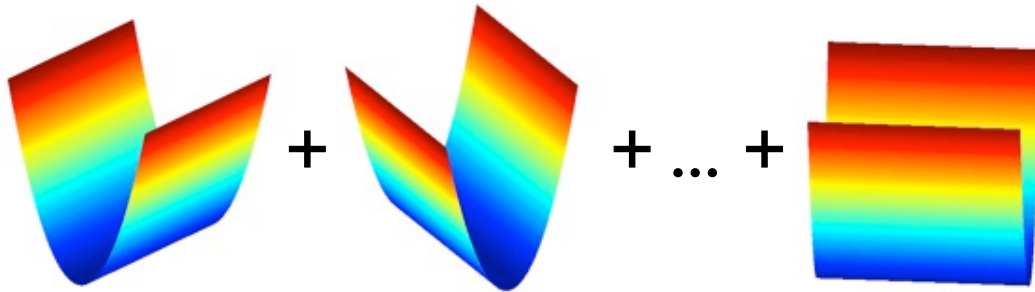
$$\hat{w}_{ridge} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2$$

# Ridge Regression

---

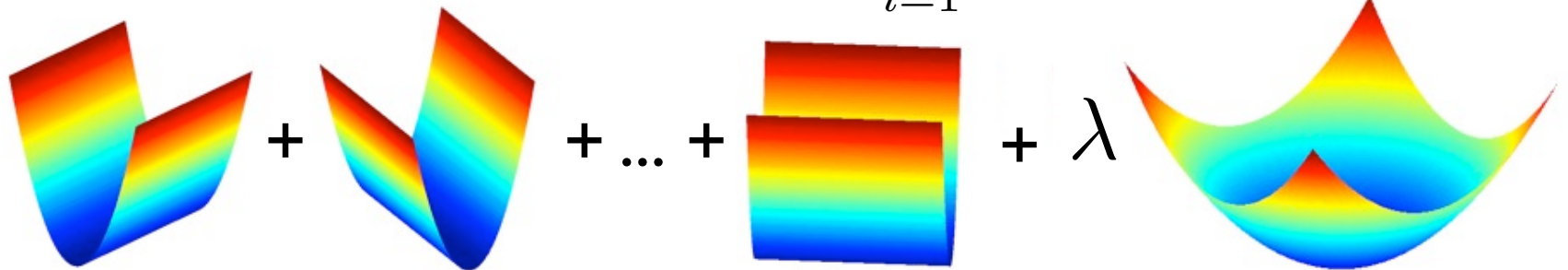
- (Original) Least squares objective:

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



- 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

---

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



# Shrinkage Properties

---

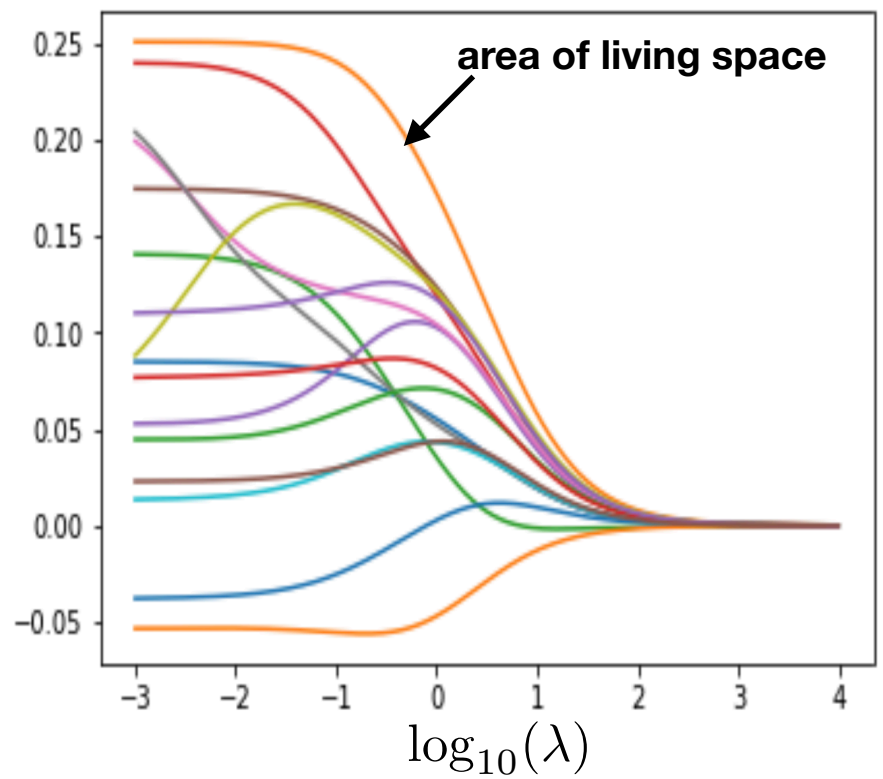
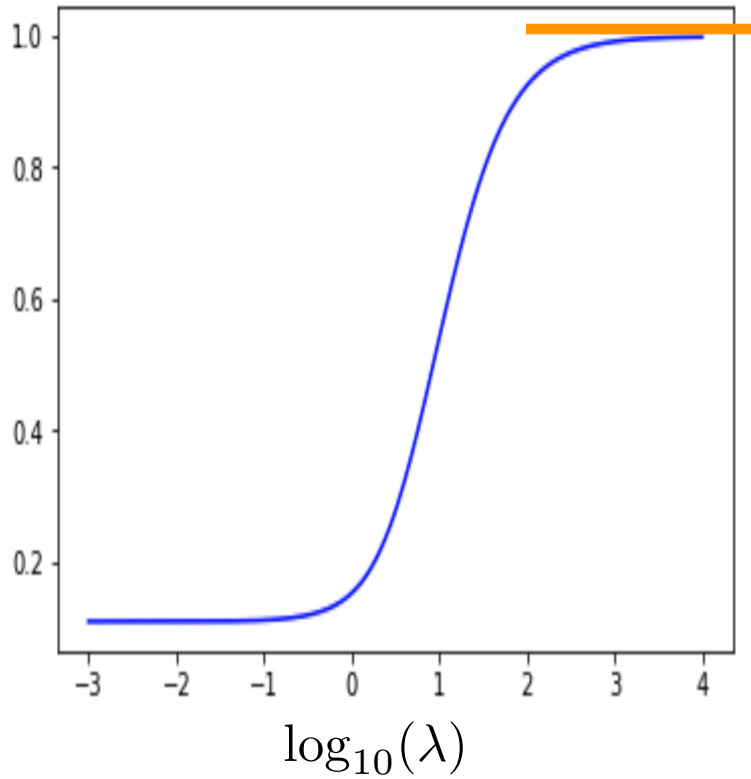
$$\begin{aligned}\hat{w}_{ridge} &= \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda \|w\|_2^2 \\ &= (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{y}\end{aligned}$$

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

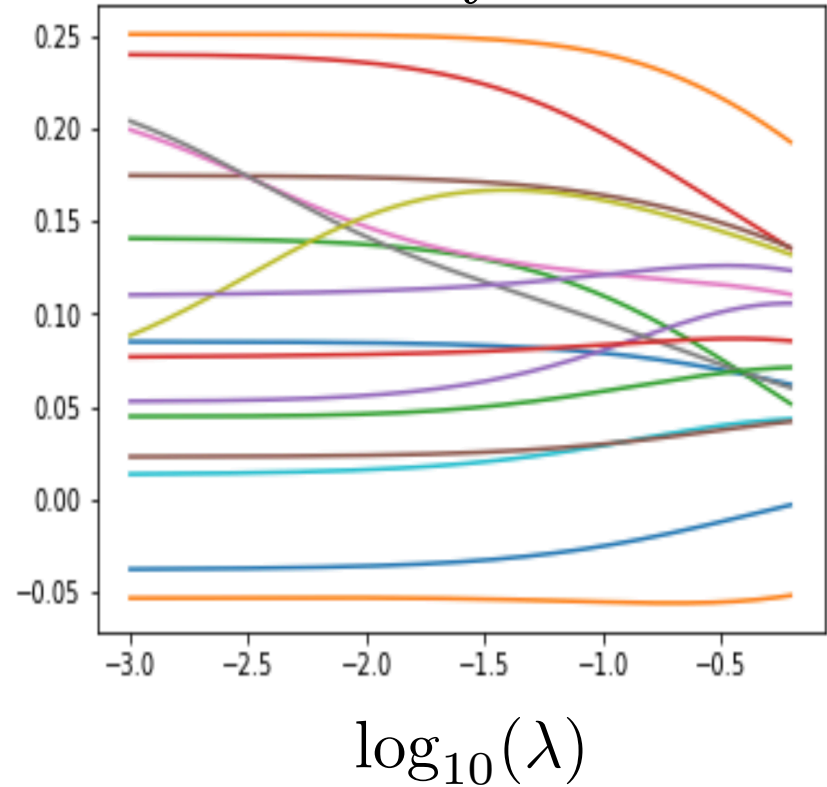
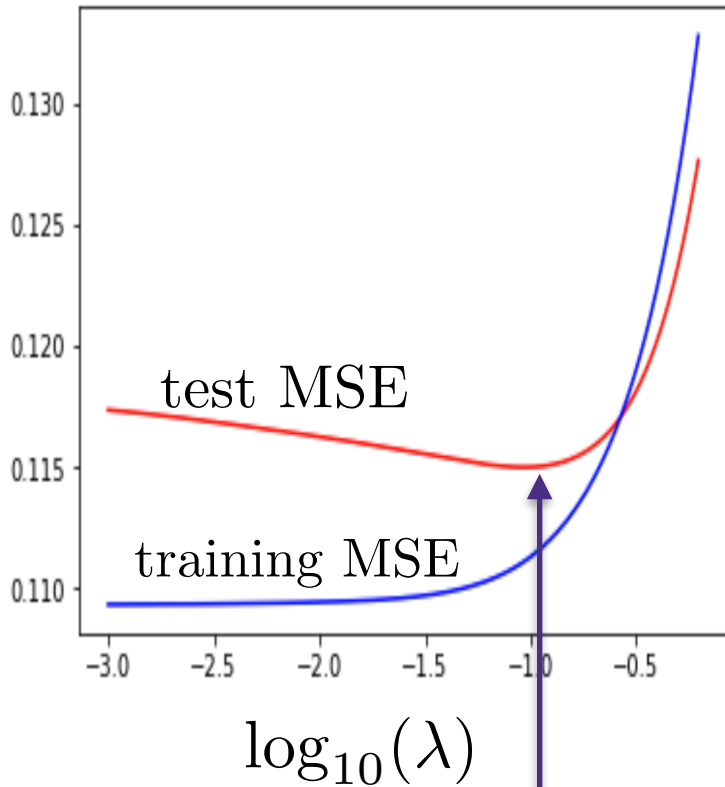
$w_i$ 's



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

$w_i$ 's



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

# 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

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

# 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

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

# 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

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

# 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

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

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

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

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

Irreduc. Error

Bias-squared

Variance

# Bias-Variance Properties

- 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

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

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

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

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

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

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



# 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

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

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

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

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

(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

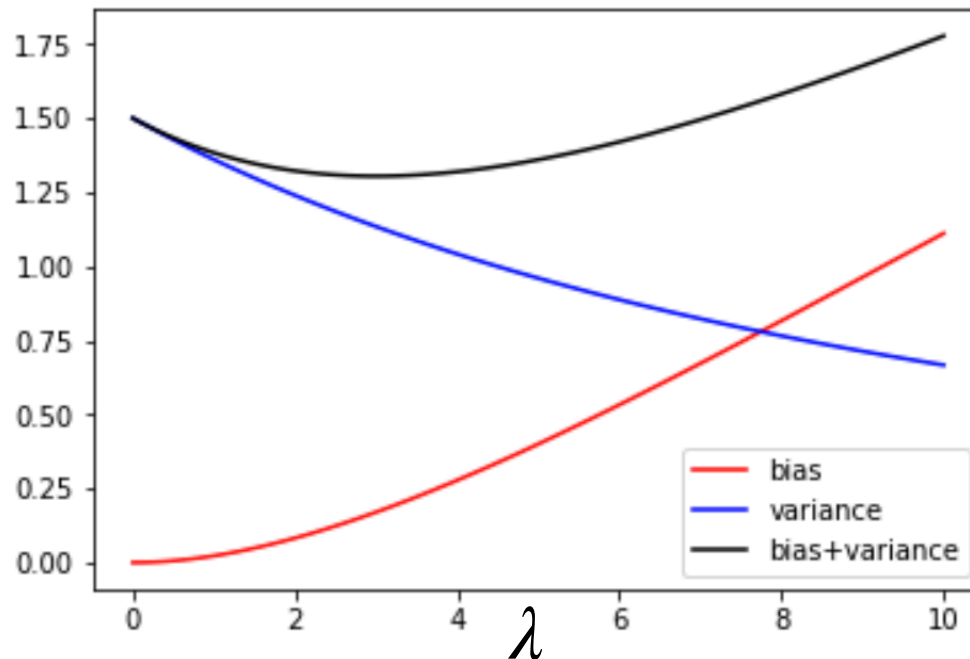
Variance

# Bias-Variance Properties

- 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

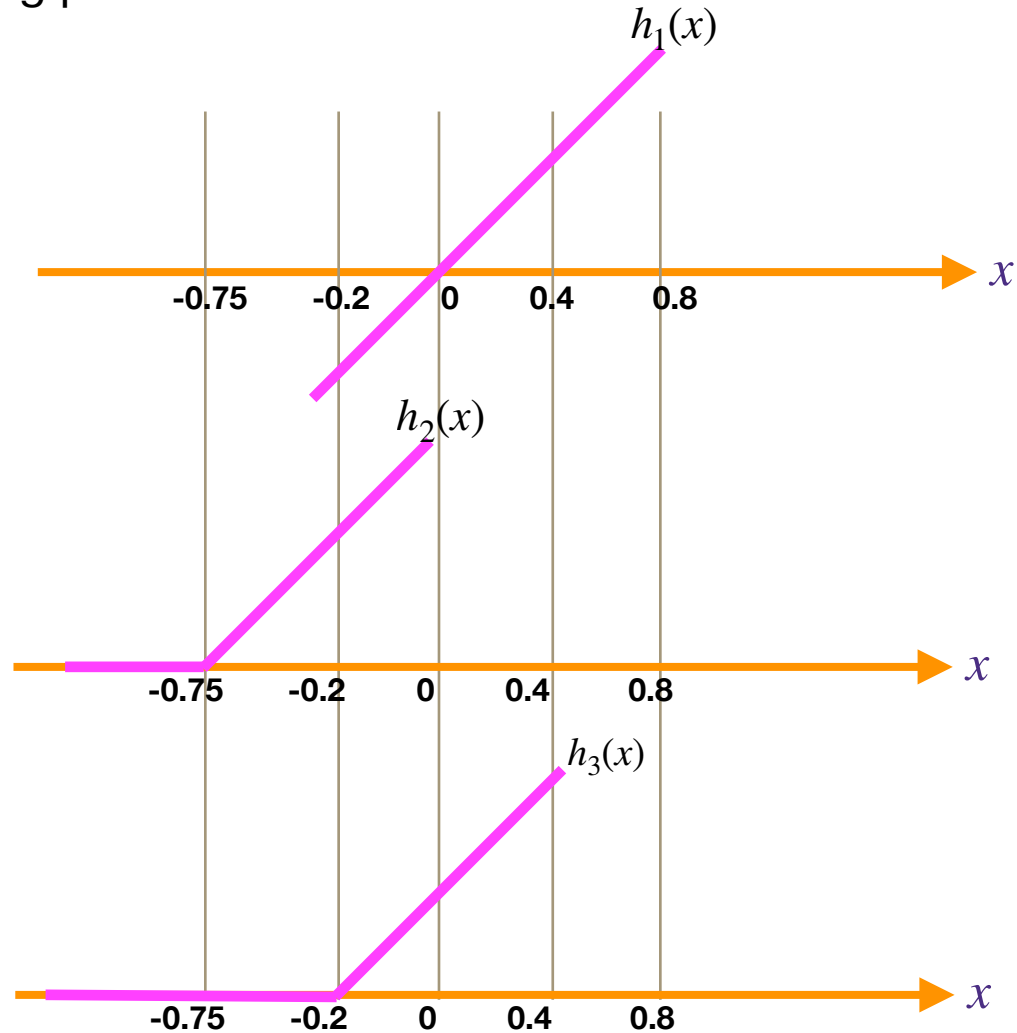
- $L_2$  penalized least-squares regression
- Regularization parameter trades off model complexity with training error
- Never regularize the offset!

# 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

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

$$[a]^+ \triangleq \max\{a, 0\}$$

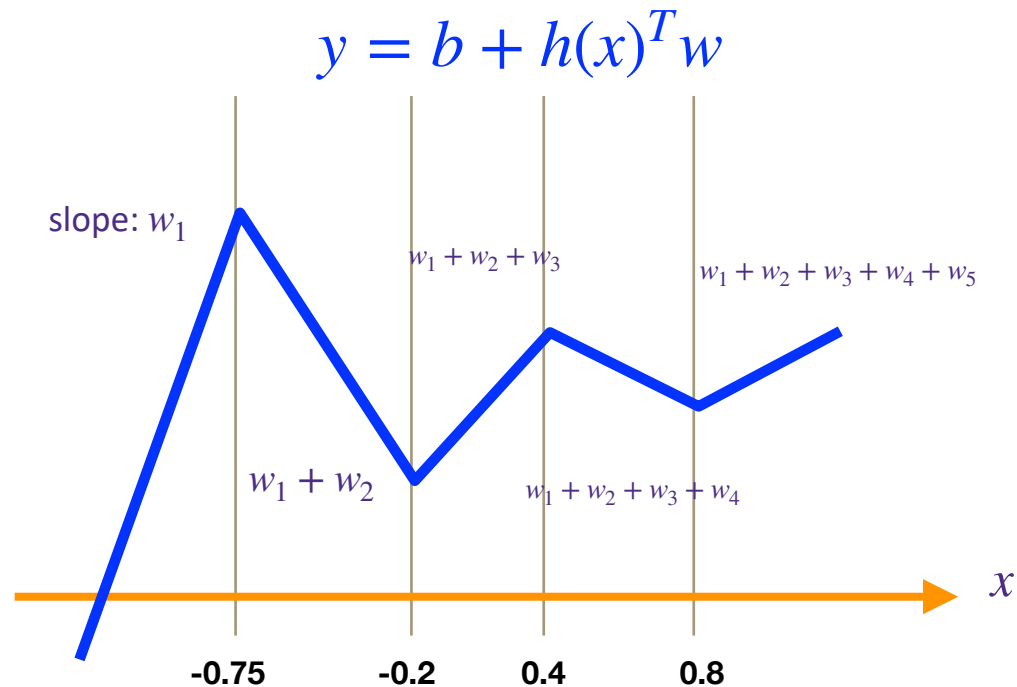


# Example: piecewise linear fit

- 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

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

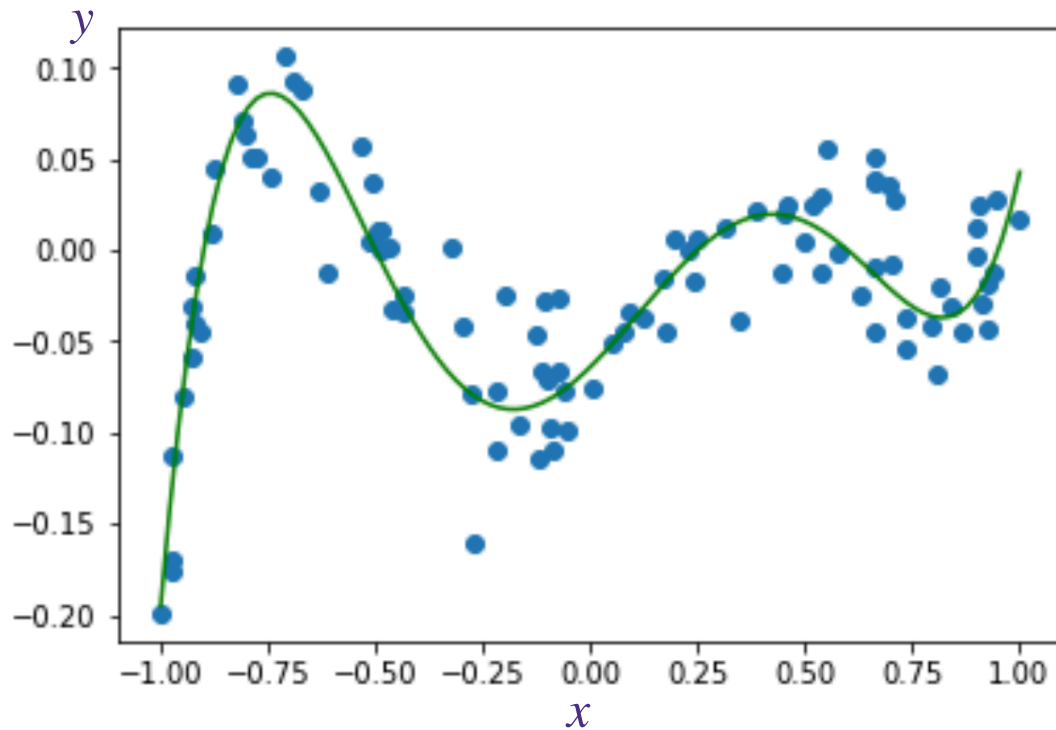
$$[a]^+ \triangleq \max\{a, 0\}$$



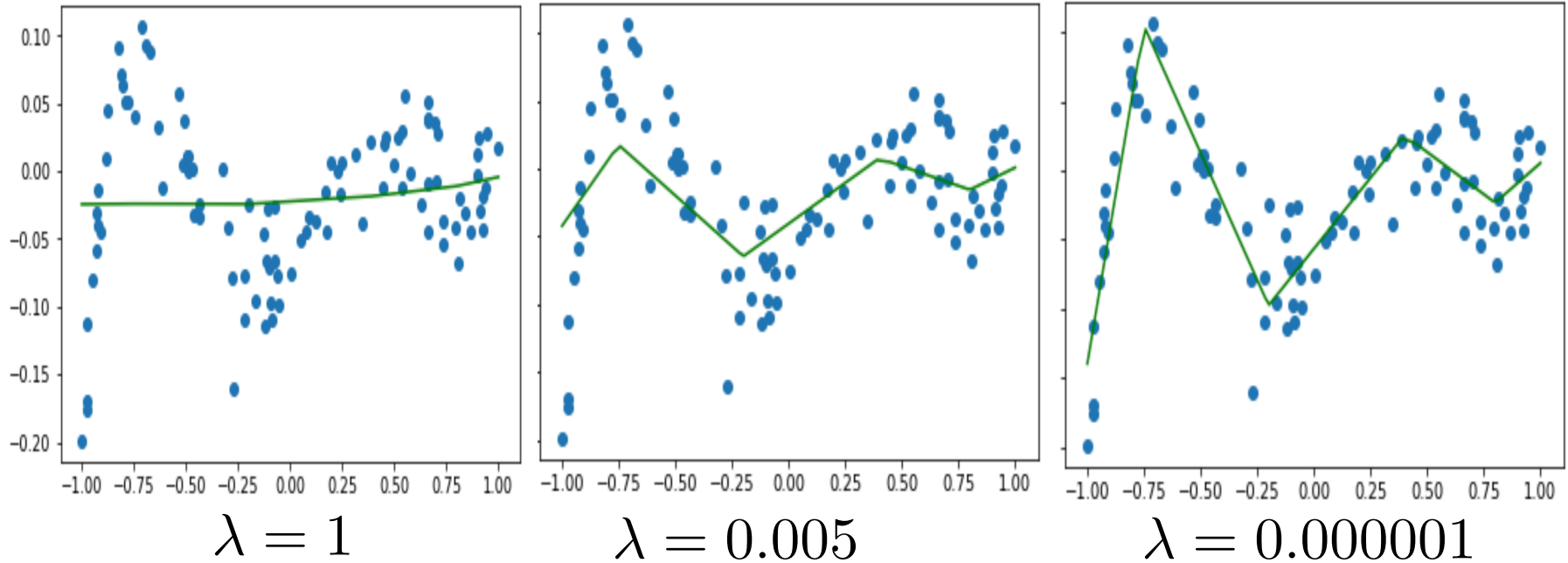
**the weights capture the change in the slopes**

# 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

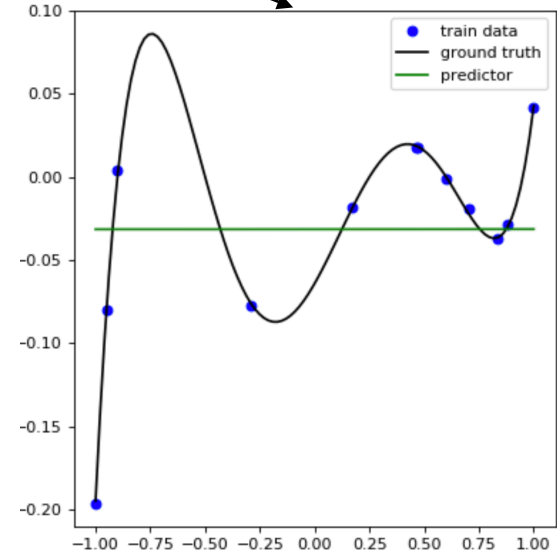
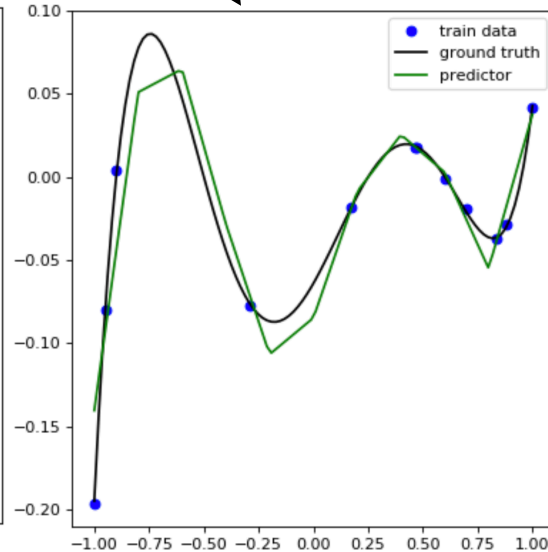
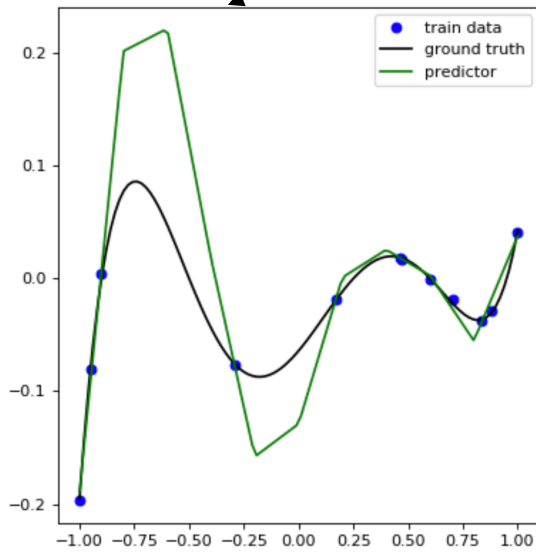
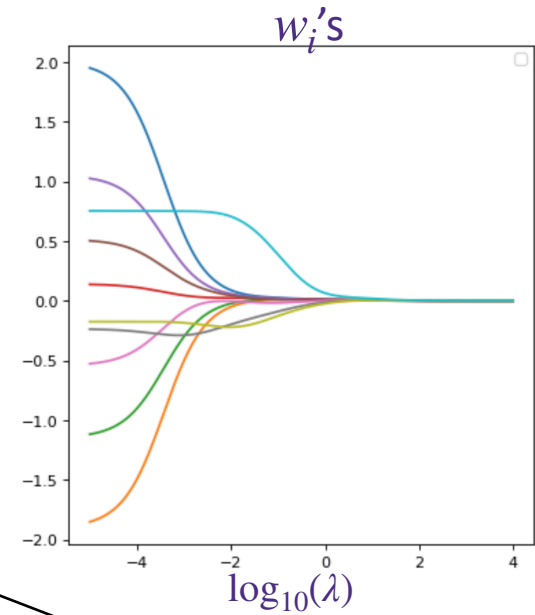
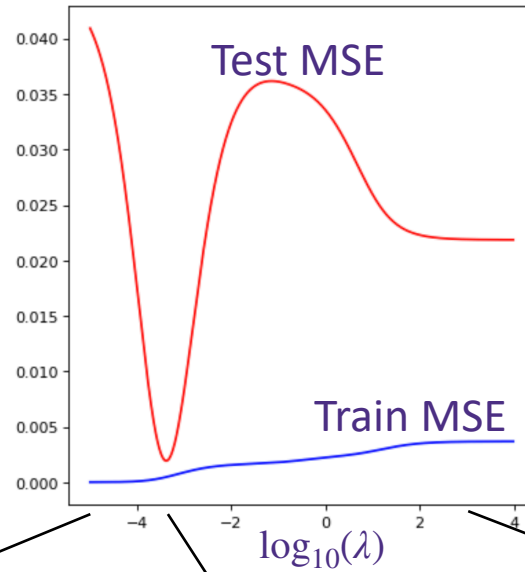


# Example: piecewise linear fit (ridge regression)



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

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





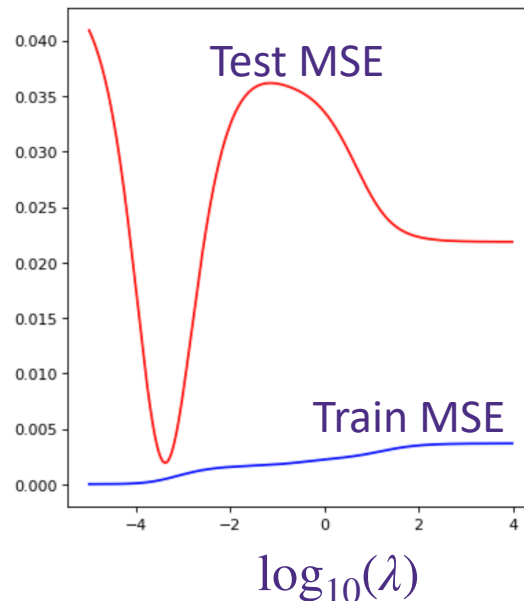
# Model selection using Cross-validation

---



# How... How... How????????

- > Ridge regression:  
How do we pick the regularization constant  $\lambda$ ...
- > Polynomial features:  
How do we pick the number of basis functions...
- > We could use the test data, but...





# (LOO) Leave-one-out cross validation

---

- > Consider a validation set with 1 example:
  - $\mathcal{D}$  : training data
  - $\mathcal{D} \setminus j$  : training data with  $j$ -th data point  $(x_j, y_j)$  moved to validation set
- > Learn model  $f_{\mathcal{D} \setminus j}$  with  $\mathcal{D} \setminus j$  dataset
- > The squared error on predicting  $y_j$ :  $(y_j - f_{\mathcal{D} \setminus j}(x_j))^2$

is an unbiased estimate of the **true error**

$$\text{error}_{\text{true}}(f_{\mathcal{D} \setminus j}) = \mathbb{E}_{(x,y) \sim P_{x,y}} [(y - f_{\mathcal{D} \setminus j}(x))^2]$$

but, variance of  $(y_j - f_{\mathcal{D} \setminus j}(x_j))^2$  is too large

# (LOO) Leave-one-out cross validation

- > Consider a validation set with 1 example:
  - $\mathcal{D}$  : training data
  - $\mathcal{D} \setminus j$  : training data with  $j$ -th data point  $(x_j, y_j)$  moved to validation set
- > Learn model  $f_{\mathcal{D} \setminus j}$  with  $\mathcal{D} \setminus j$  dataset

> The squared error on predicting  $y_j$ :  $(y_j - f_{\mathcal{D} \setminus j}(x_j))^2$

is an unbiased estimate of the **true error**

$$\text{error}_{\text{true}}(f_{\mathcal{D} \setminus j}) = \mathbb{E}_{(x,y) \sim P_{x,y}} [(y - f_{\mathcal{D} \setminus j}(x))^2]$$

but variance of  $(y_j - f_{\mathcal{D} \setminus j}(x_j))^2$  is too large, so instead

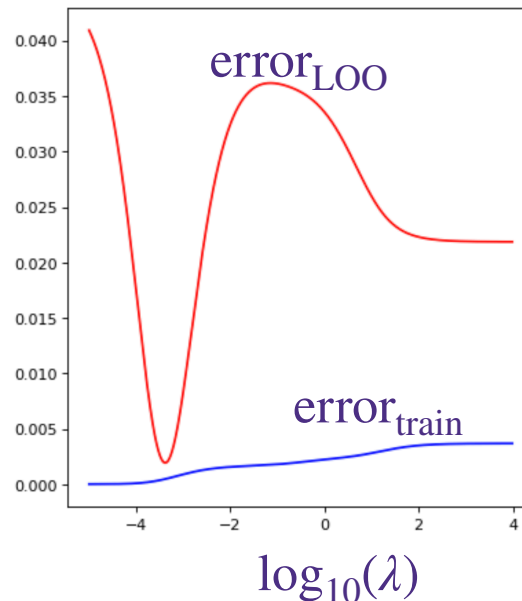
- > **LOO cross validation:** Average over all data points  $j$ :
  - Train  $n$  times:  
for each data point you leave out, learn a new classifier  $f_{\mathcal{D} \setminus j}$

- **Estimate the true error as:**

$$\text{error}_{LOO} = \frac{1}{n} \sum_{j=1}^n (y_j - f_{\mathcal{D} \setminus j}(x_j))^2$$

# LOO cross validation is (almost) unbiased estimate!

- > When computing LOOCV error, we only use  $n - 1$  data points to train
  - So it's not estimate of true error of learning with  $n$  data points
  - Usually pessimistic – learning with less data typically gives worse answer. (Leads to an over estimation of the error)
- > LOO is almost unbiased! Use LOO error for model selection!!!
  - **E.g., picking  $\lambda$**



# Computational cost of LOO

---

- > Suppose you have 100,000 data points
- > say, you implemented a fast version of your learning algorithm
  - Learns in only 1 second
- > Computing LOO will take about 1 day!!

# Use $k$ -fold cross validation

> Randomly divide training data into  $k$  equal parts

-  $\mathcal{D}_1, \dots, \mathcal{D}_k$

$$\mathcal{D} = \mathcal{D}_1 \mathcal{D}_2 \mathcal{D}_3 \mathcal{D}_4 \mathcal{D}_5$$

> For each  $i$

- Learn model  $f_{\mathcal{D} \setminus \mathcal{D}_i}$  using data point not in  $\mathcal{D}_i$

- Estimate error of  $f_{\mathcal{D} \setminus \mathcal{D}_i}$  on validation set  $\mathcal{D}_i$ :

$$\text{error}_{\mathcal{D}_i} = \frac{1}{|\mathcal{D}_i|} \sum_{(x_j, y_j) \in \mathcal{D}_i} (y_j - f_{\mathcal{D} \setminus \mathcal{D}_i}(x_j))^2$$

$$f_{\mathcal{D} \setminus \mathcal{D}_3}$$

Train	Train	Validation	Train	Train
-------	-------	------------	-------	-------

>

>



# Use $k$ -fold cross validation

> Randomly divide training data into  $k$  equal parts

–  $\mathcal{D}_1, \dots, \mathcal{D}_k$

> For each  $i$

– Learn model  $f_{\mathcal{D} \setminus \mathcal{D}_i}$  using data point not in  $\mathcal{D}_i$

– Estimate error of  $f_{\mathcal{D} \setminus \mathcal{D}_i}$  on validation set  $\mathcal{D}_i$ :

$$\text{error}_{\mathcal{D}_i} = \frac{1}{|\mathcal{D}_i|} \sum_{(x_j, y_j) \in \mathcal{D}_i} (y_j - f_{\mathcal{D} \setminus \mathcal{D}_i}(x_j))^2$$

>  $k$ -fold cross validation error is average over data splits:

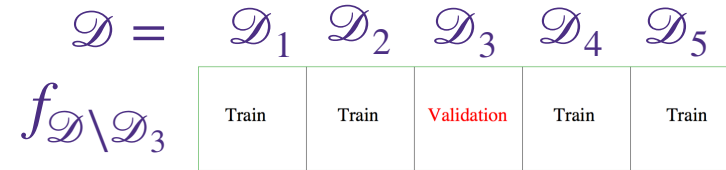
$$\text{error}_{k\text{-fold}} = \frac{1}{k} \sum_{i=1}^k \text{error}_{\mathcal{D}_i}$$

>  $k$ -fold cross validation properties:

– Much faster to compute than LOO as  $k \ll n$

– More (pessimistically) biased – using much less data, only  $n - \frac{n}{k}$

– Usually,  $k = 10$

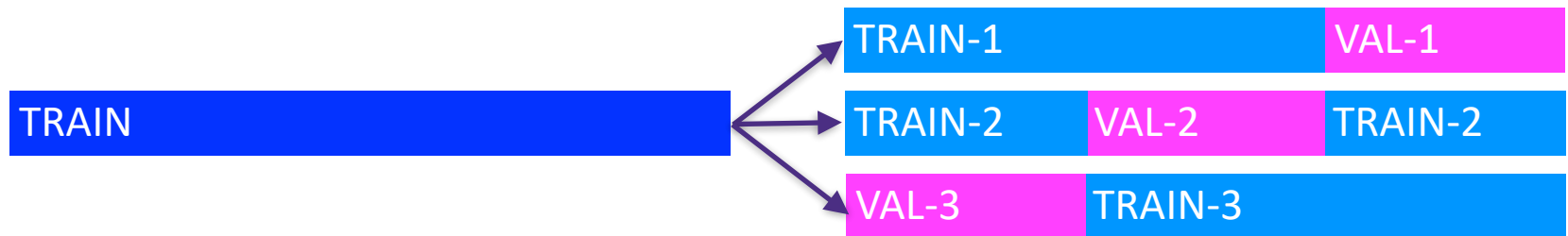


# Recap

- > Given a dataset, begin by splitting into



- > Model selection: Use k-fold cross-validation on **TRAIN** to train predictor and choose hyper-parameters such as  $\lambda$



- > Model assessment: Use **TEST** to assess the accuracy of the model you output
  - **Never ever ever ever ever train or choose parameters based on the test data**

# Model selection using cross validation

---

> **For**  $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$

> **For**  $j \in \{1, \dots, k\}$

>

$$\hat{w}_{\lambda, \text{Train}-j} \leftarrow \arg \min_w \sum_{i \in \text{Train}-j} (y_i - w^T x_i)^2 + \lambda \|w\|_2^2$$

>

$$\hat{\lambda} \leftarrow \arg \min_{\lambda} \frac{1}{k} \sum_{j=1}^k \sum_{i \in \text{Val}-j} (y_i - \hat{w}_{\lambda, \text{Train}-j}^T x_i)^2$$

# Example 1

---

- > You wish to predict the stock price of zoom.us given historical stock price data  $y_i$ 's (for each  $i$ -th day) and the historical news articles  $x_i$ 's
- > You use all daily stock price up to Jan 1, 2020 as **TRAIN** and Jan 2, 2020 - April 13, 2020 as **TEST**
- > What's wrong with this procedure?

# Example 2

---

- > Given 10,000-dimensional data and  $n$  examples, we pick a subset of 50 dimensions that have the highest correlation with labels in the training set:

50 indices  $j$  that have largest

$$\frac{|\sum_{i=1}^n x_{i,j} y_i|}{\sqrt{\sum_{i=1}^n x_{i,j}^2}}$$

- > After picking our 50 features, we then use CV with the training set to train ridge regression with regularization  $\lambda$
- > What's wrong with this procedure?

# Recap

---

## > Learning is...

- Collect some data
  - > E.g., housing info and sale price
- Randomly split dataset into TRAIN, VAL, and TEST
  - > E.g., 80%, 10%, and 10%, respectively
- Choose a hypothesis class or model
  - > E.g., linear with non-linear transformations
- Choose a loss function
  - > E.g., least squares with ridge regression penalty on TRAIN
- **Choose an optimization procedure**
  - > E.g., set derivative to zero to obtain estimator, **cross-validation on VAL to pick num. features and amount of regularization**
- Justifying the accuracy of the estimate
  - > E.g., report TEST error

# Simple variable selection: LASSO for sparse regression

---



# Sparsity

---

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

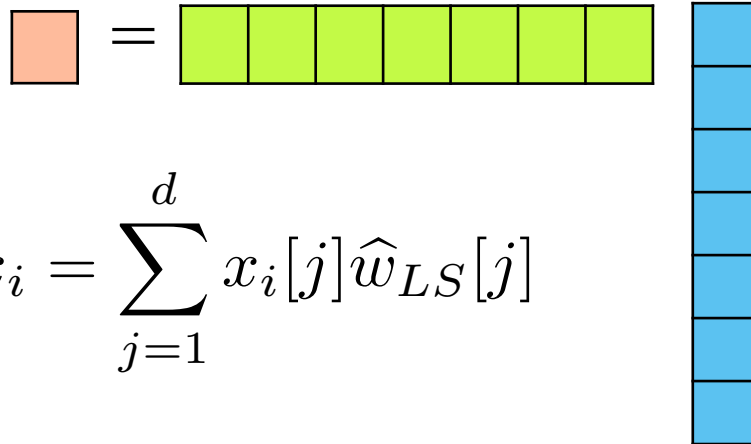
- Vector  $w$  is **sparse**, if many entries are zero



# 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  $\text{size}(w) = 100$  Billion, each prediction  $w^T x$  is expensive:
    - If  $w$  is sparse, prediction computation only depends on number of non-zeros in  $w$



$$\hat{y}_i = \hat{w}_{LS}^T x_i = \sum_{j=1}^d x_i[j] \hat{w}_{LS}[j]$$

# 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?



Lot size	Dishwasher
Single Family	Garbage disposal
Year built	Microwave
Last sold price	Range / Oven
Last sale price/sqft	Refrigerator
Finished sqft	Washer
Unfinished sqft	Dryer
Finished basement sqft	Laundry location
# floors	Heating type
Flooring types	Jetted Tub
Parking type	Deck
Parking amount	Fenced Yard
Cooling	Lawn
Heating	Garden
Exterior materials	Sprinkler System
Roof type	
Structure style	

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

# Finding best subset: Exhaustive

---

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

# Finding best subset: Greedy

---

## **Forward stepwise:**

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

## **Backward stepwise:**

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

## **Combining forward and backward steps:**

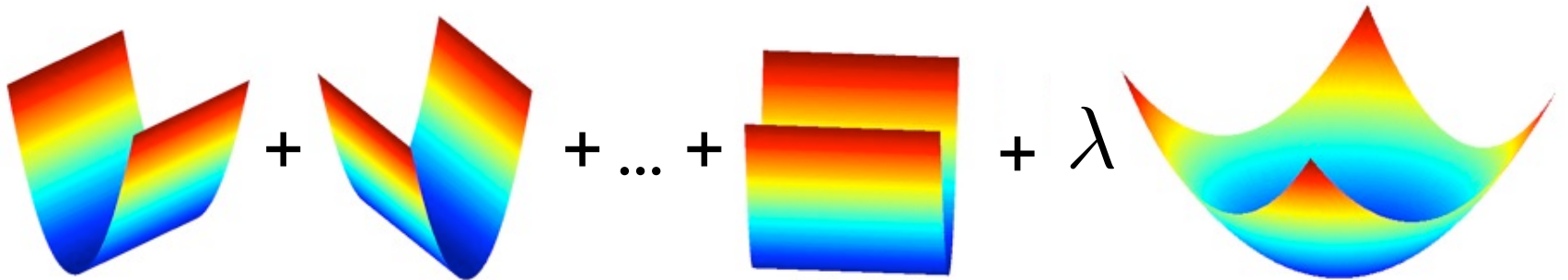
In forward algorithm, insert steps to remove features no longer as important

*Lots of other variants, too.*

# Finding best subset: Regularize

Ridge regression makes coefficients small

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

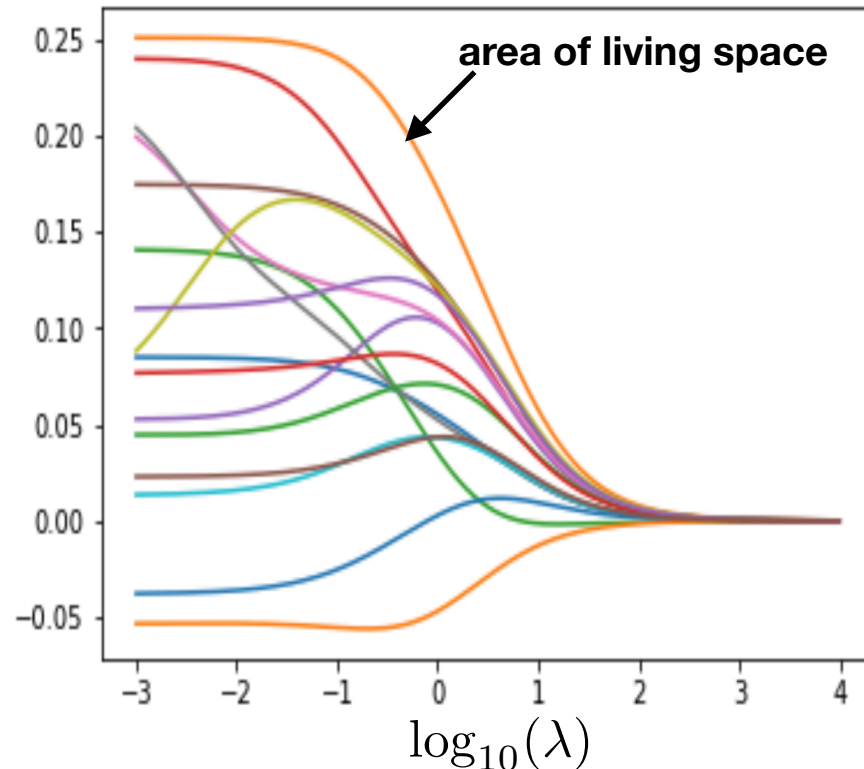


# Finding best subset: Regularize

Ridge regression makes coefficients small

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

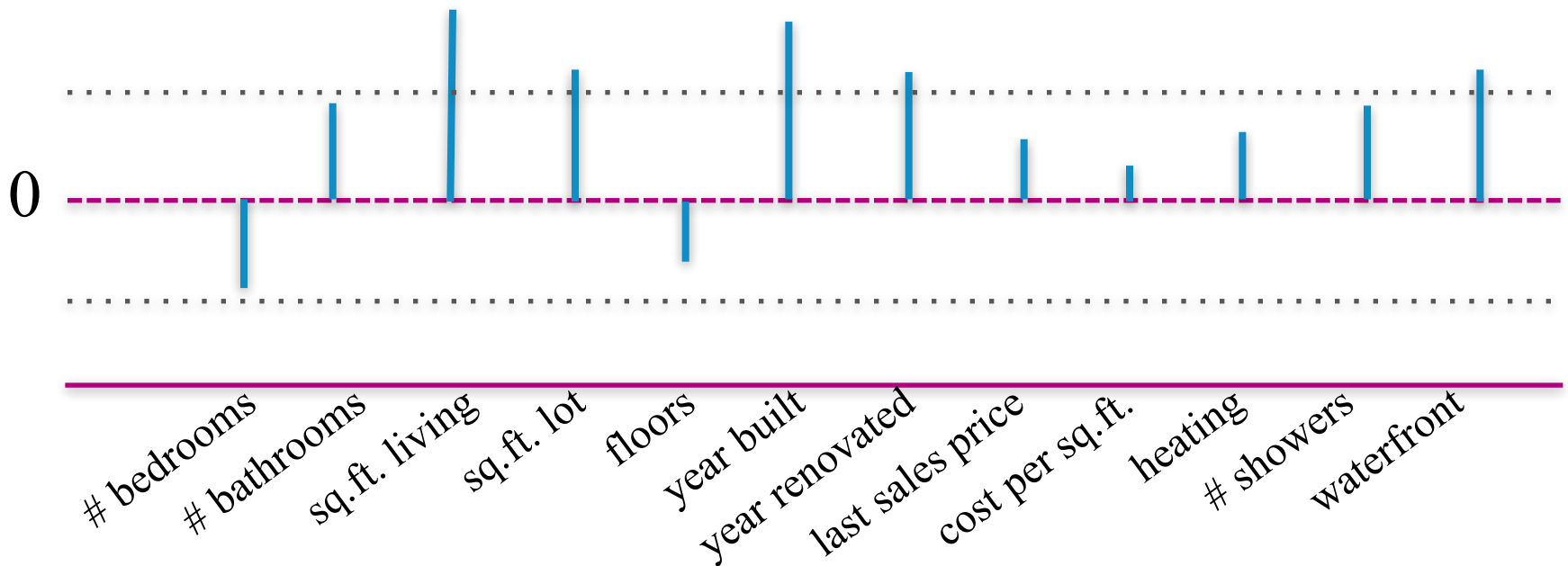
$w_i$ 's



# Thresholded Ridge Regression

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

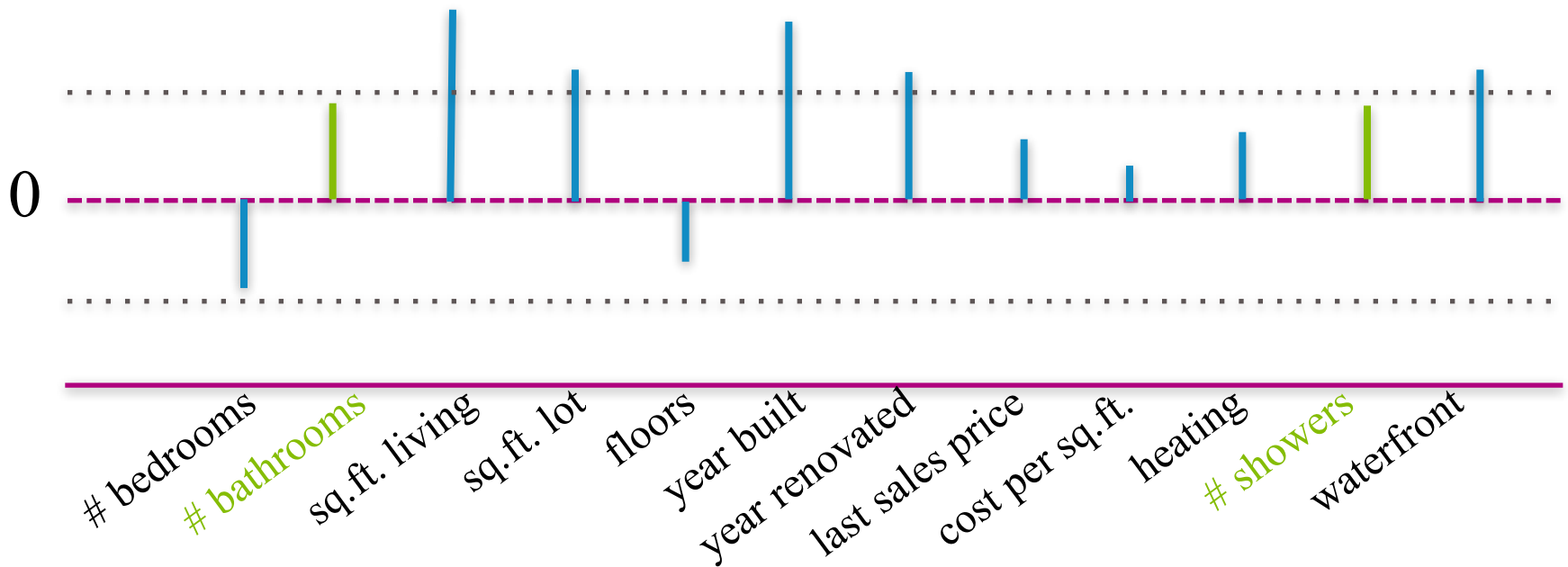
Why don't we just set **small** ridge coefficients to 0?



# 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)

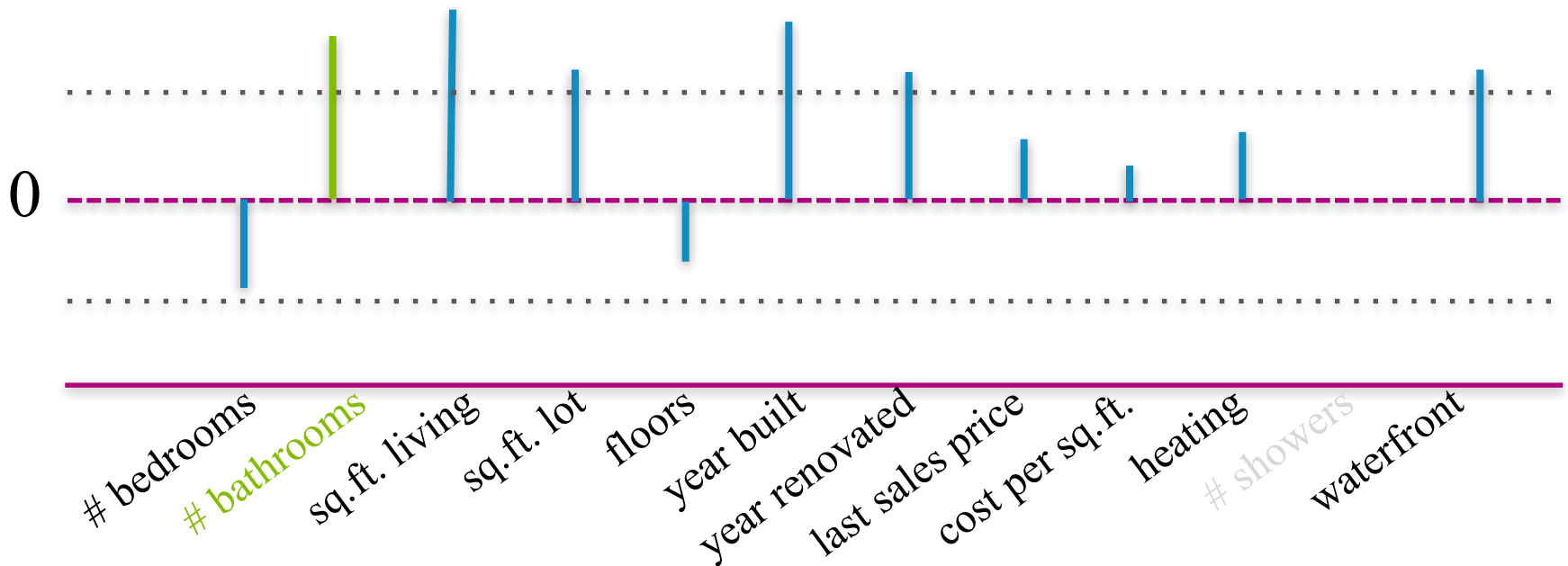




# Thresholded Ridge Regression

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

What if we **didn't** include showers? Weight on bathrooms increases!



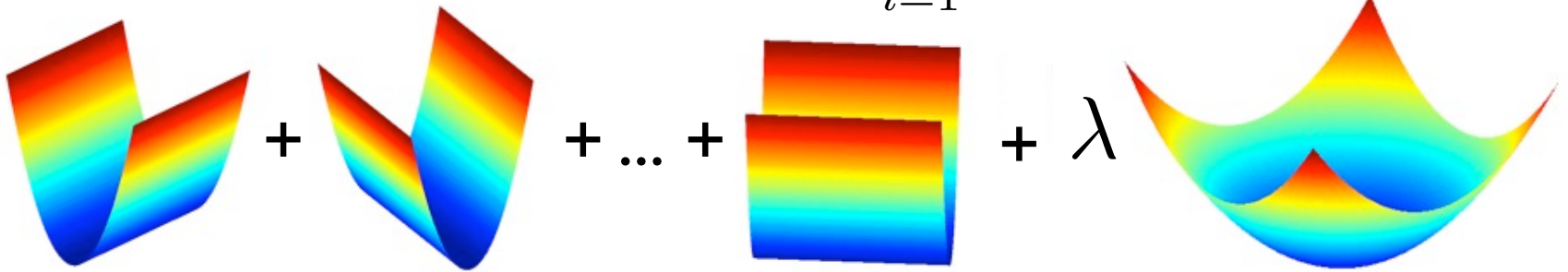
Can another regularizer perform selection automatically?

# Recall Ridge Regression

---

- Ridge Regression objective:

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



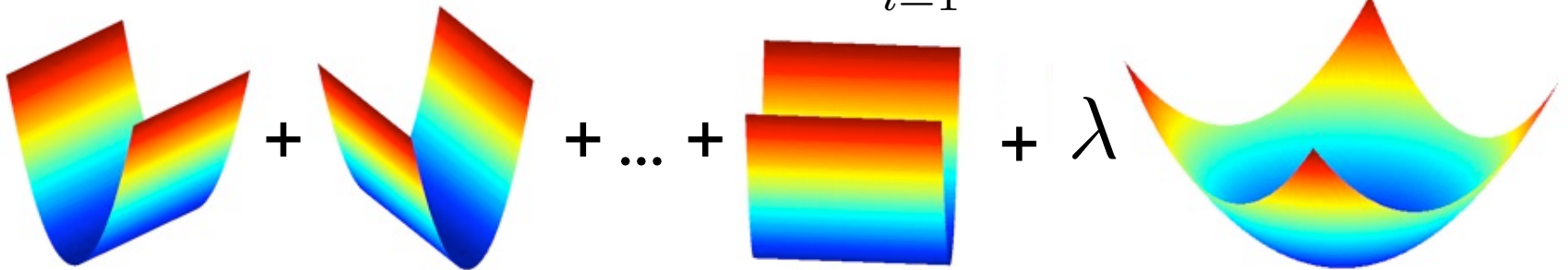
$$\|w\|_p = \left( \sum_{i=1}^d |w|^p \right)^{1/p}$$

# Ridge vs. Lasso Regression

---

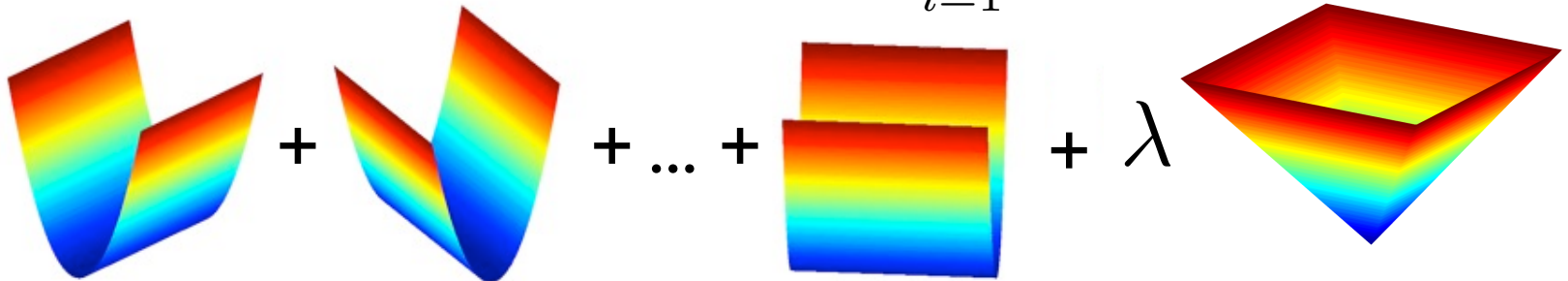
- Ridge Regression objective:

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



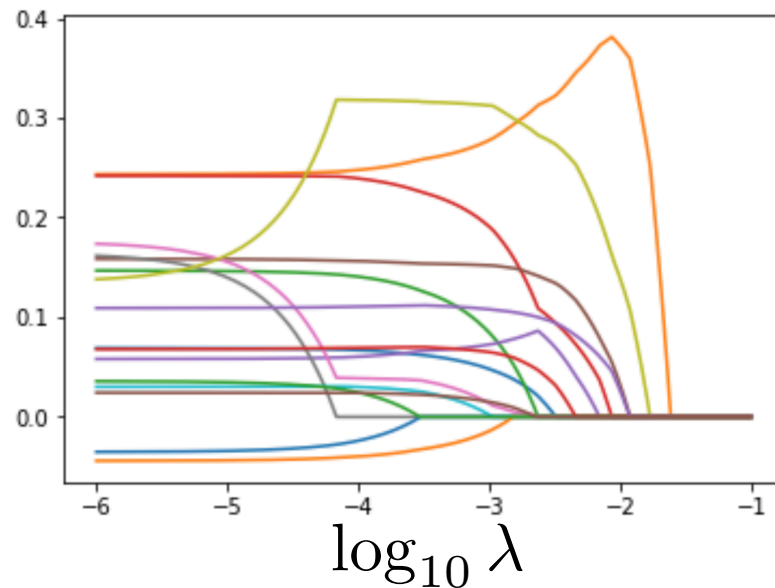
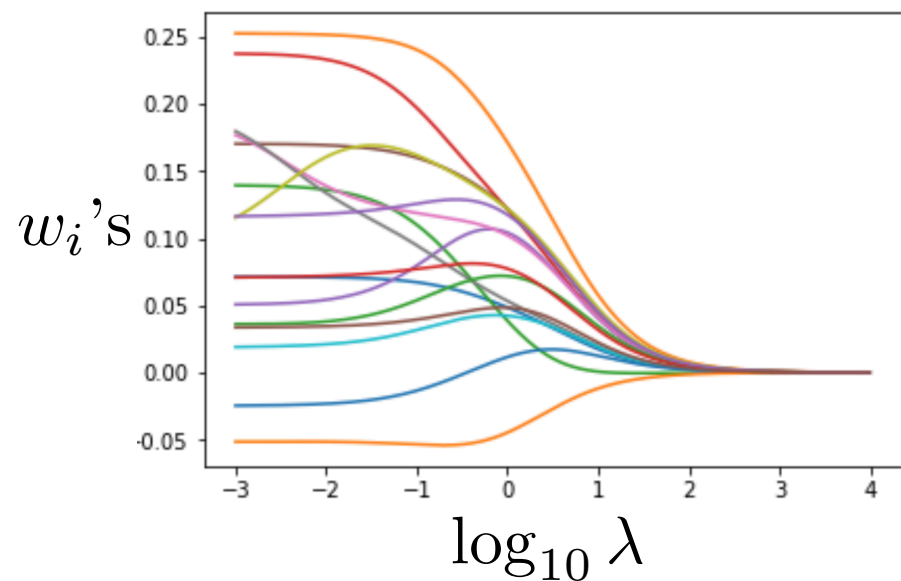
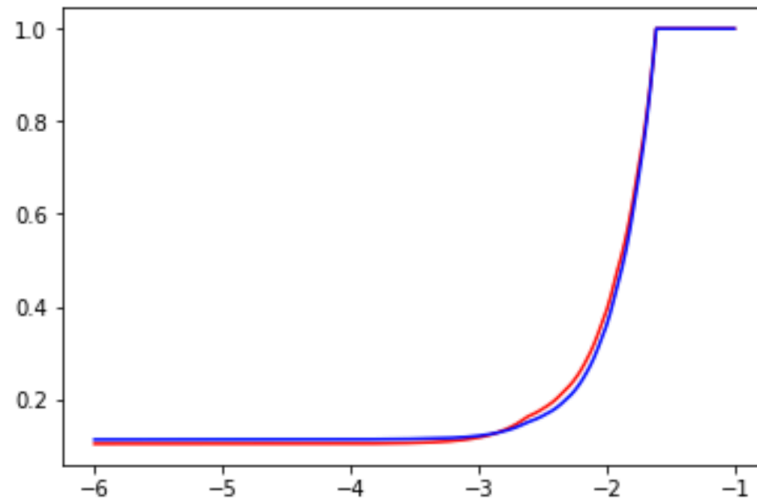
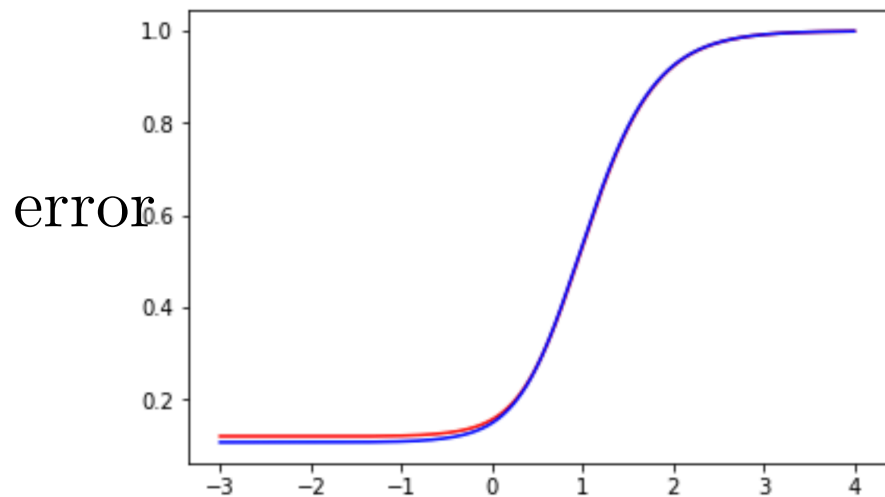
- Lasso objective:

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



# Example: house price with 16 features

test error is red and train error is blue



Ridge regression

Lasso regression

# Lasso regression naturally gives sparse features

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

# Example: piecewise-linear fit

- We use Lasso on the piece-wise linear example

$$h_0(x) = 1$$

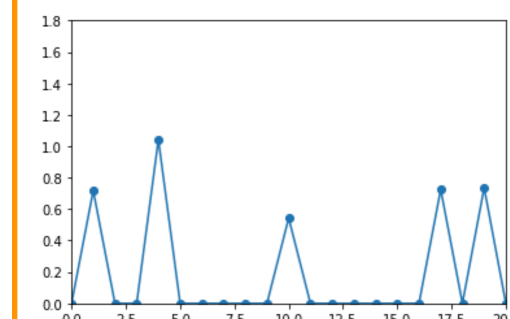
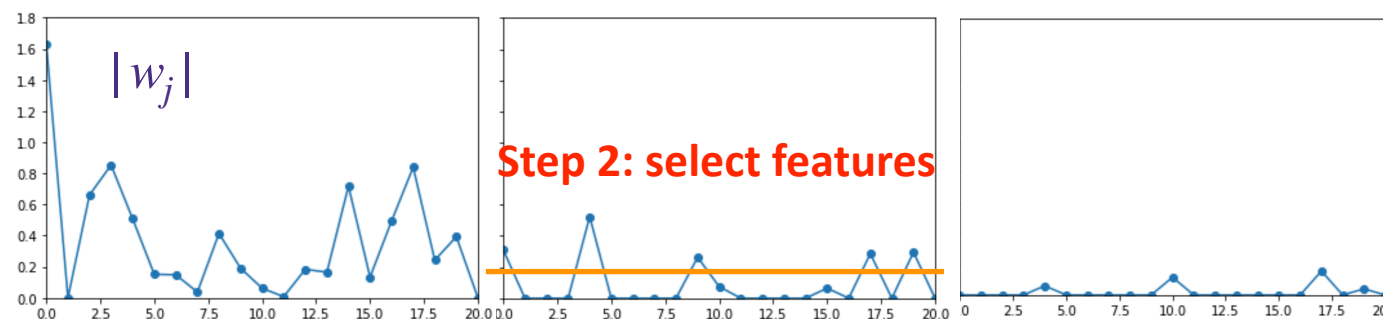
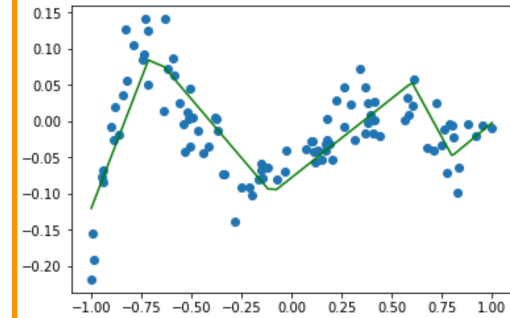
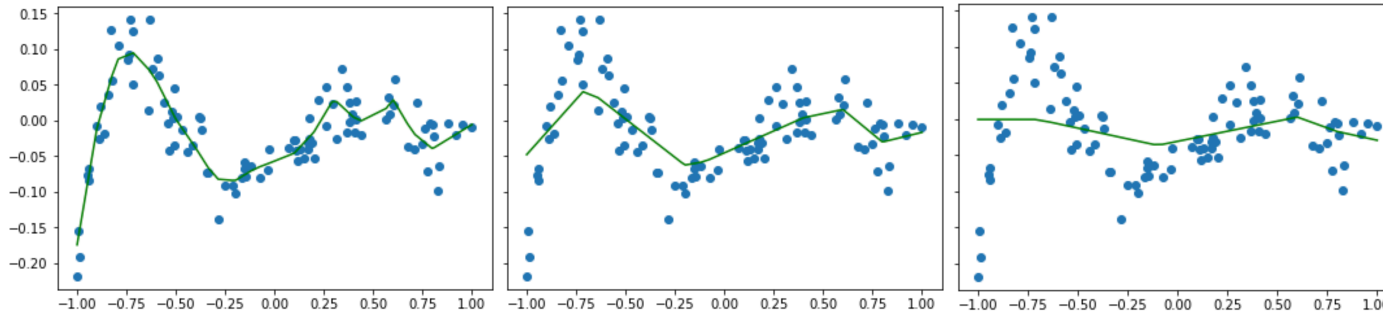
$$h_i(x) = [x + 1.1 - 0.1i]^+$$

Step 1: find optimal  $\lambda^*$

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

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

# Penalized Least Squares

---

$$\text{Ridge : } r(w) = \|w\|_2^2 \qquad \text{Lasso : } r(w) = \|w\|_1$$

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

# Penalized Least Squares

---

$$\text{Ridge : } r(w) = \|w\|_2^2 \quad \text{Lasso : } r(w) = \|w\|_1$$

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

For any  $\lambda \geq 0$  for which  $\hat{w}_r$  achieves the minimum, there exists a  $\mu \geq 0$  such that

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

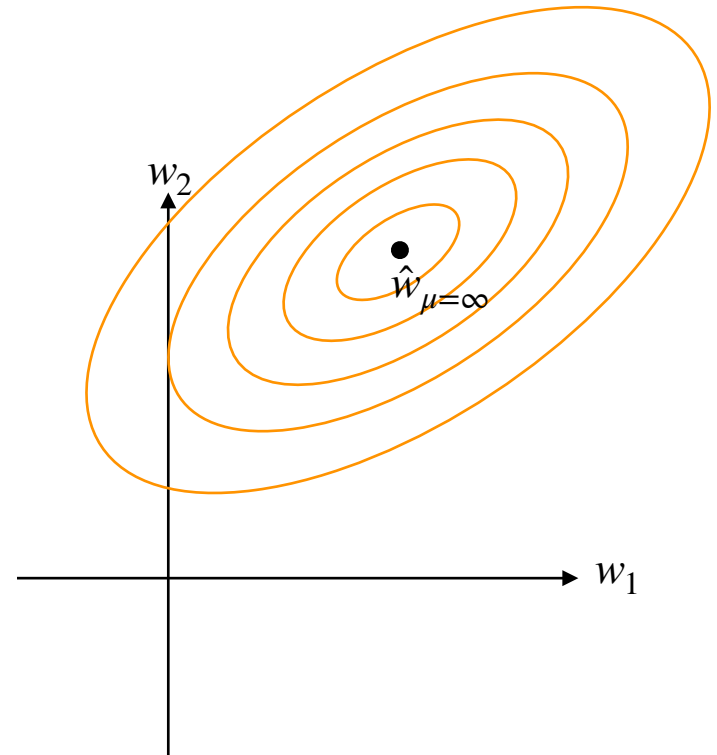


# Why does Lasso give sparse solutions?

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

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

- 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  $\hat{w}_{\mu=\infty} = \hat{w}_{\text{LS}}$



# Why does Lasso give sparse solutions?

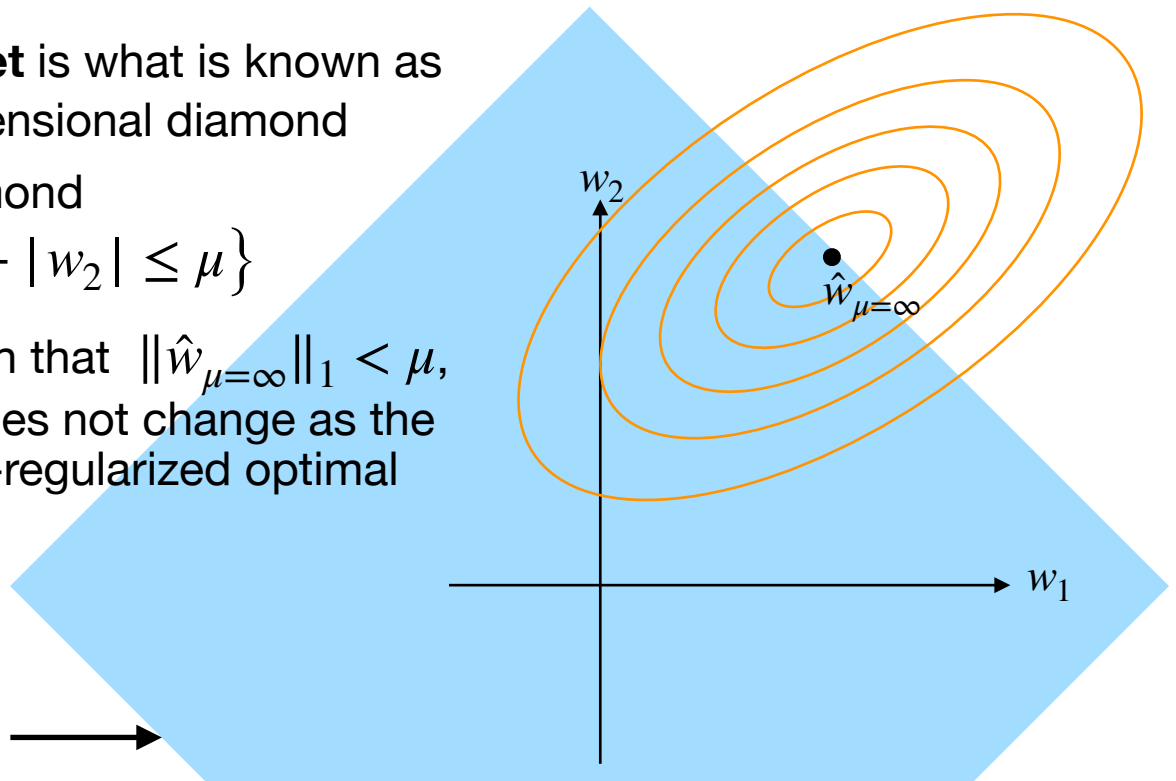
$$\text{minimize}_w \sum_{i=1}^n (w^T x_i - y_i)^2$$

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

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

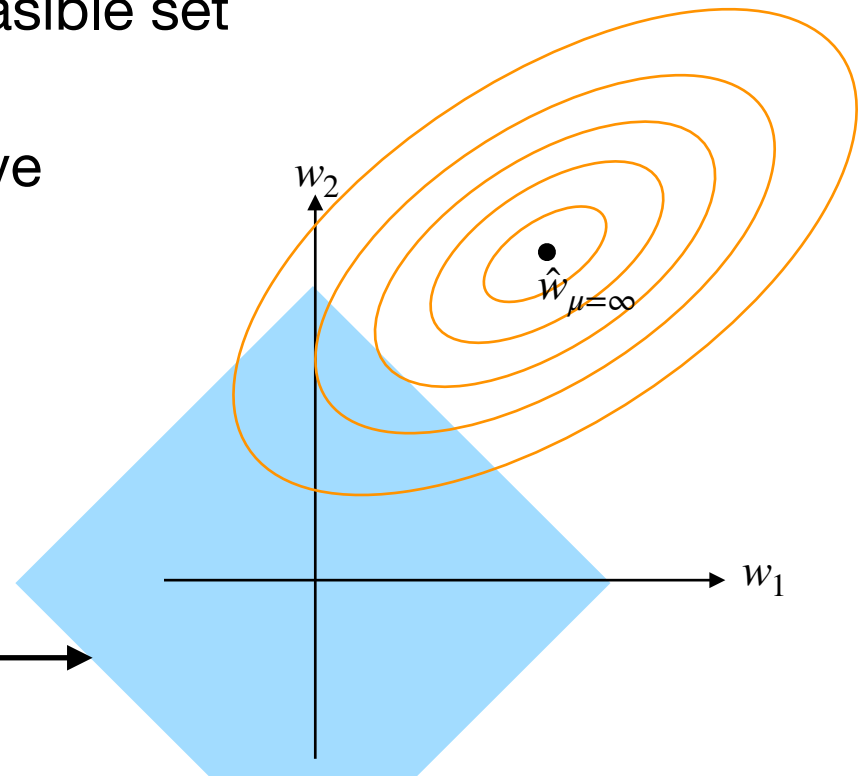
# Why does Lasso give sparse solutions?

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

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

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

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

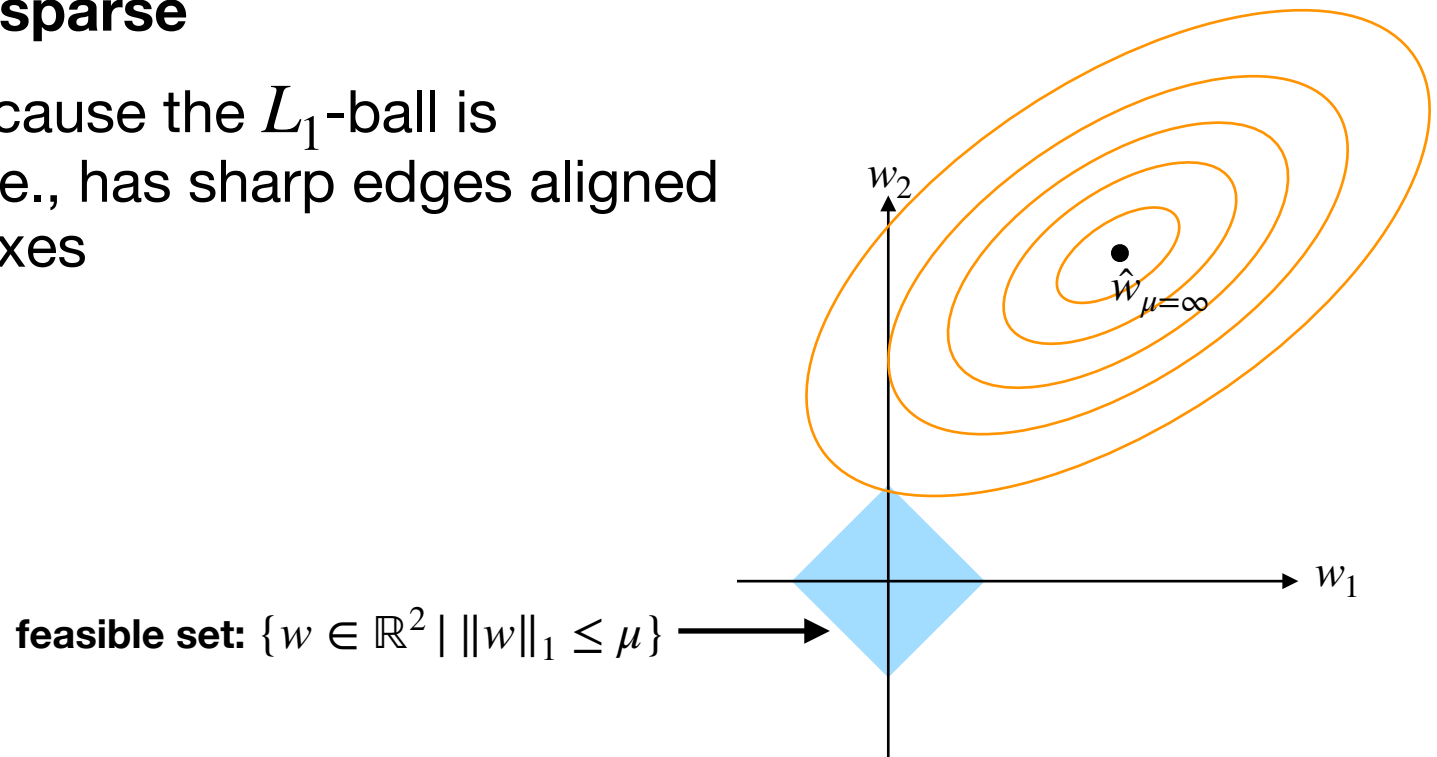


# Why does Lasso give sparse solutions?

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

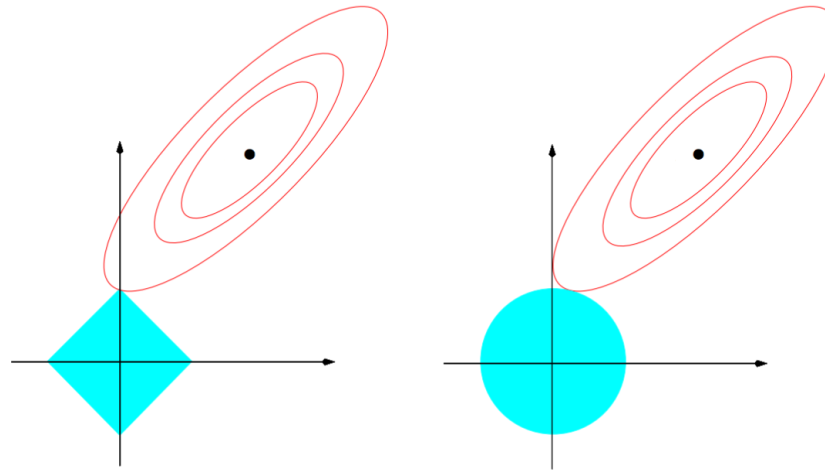
$$\text{subject to } \|w\|_1 \leq \mu$$

- 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



# Penalized Least Squares

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



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

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

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

$$\text{subject to } \|w\|_2^2 \leq \mu$$

# Questions?

---