

# Computational Data Analysis

## Machine Learning

**Yao Xie, Ph.D.**

*Associate Professor*

Harold R. and Mary Anne Nash Early Career Professor  
H. Milton Stewart School of Industrial and Systems  
Engineering

Cross Validation



# Motivating example: real estate agent

Data was collected on 100 homes recently sold in a city. It consisted of the sale price, house size, the number of bedrooms, the number of bathrooms, the lot size, and the annual real estate tax. Use price as the response variable and determine which of the these factors should be included in the regression model.

The screenshot shows a REDFIN search results page for zip code 30342. At the top, there's a search bar with '30342' and a 'Save Search' button. To the right are 'Tools' and a user profile for 'Chrystal'. Below the search bar is a map of the area, with a red outline highlighting a specific neighborhood. A 'Selected Home' is marked with a green icon on the map. On the right side of the map, there's a large image of a two-story house at 10 Battle Ridge Dr, Atlanta, GA 30342, listed for \$599,900. Below the image, the listing details are: HOA \$300, Status Active, \$/Sq. Ft. \$160, Year Built 1988, Lot Size 0.61 Acres. To the left of the house image, it says 'Pam O'Connor-Smith saw this home REDFIN Agent'. Below the house image is a short description: "Spacious home at the entrance to the community. There is no foyer entry, you are immediately in a sunning area. There is also a s..." followed by a 'More' link. At the bottom of the page is a table showing three more homes:

| Address                  | Location     | Price     | Beds | Baths | Sq.Ft. | \$/Sq.Ft. | Days |
|--------------------------|--------------|-----------|------|-------|--------|-----------|------|
| 10 Battle Ridge Dr       | Battle Creek | \$599,900 | 5    | 4     | 3,742  | \$160     | 54   |
| 4650 Windsor Gate Ct     | Windsor A... | \$625,000 | 5    | 4.5   | 3,564  | \$175     | 167  |
| 245 Mystic Ridge Hl #245 | Mystic Ridge | \$450,000 | 4    | 3     | -      | -         | 1    |

# Linear Regression Model

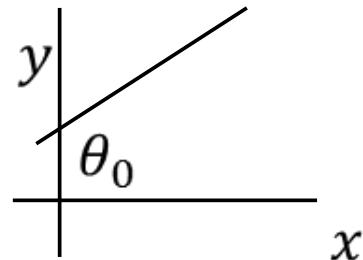
- Assume  $y$  is a linear function of  $x$  (features) plus noise  $\epsilon$

$$y = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n + \epsilon$$

- where  $\epsilon$  is an error term of unmodeled effects or random noise
- Let  $\theta = (\theta_0, \theta_1, \dots, \theta_n)^\top$ , and augment data by one dimension

$$x \leftarrow (1, x)^\top$$

- Then  $y = \theta^\top x + \epsilon$



# Matrix version of the gradient

- $\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m} \sum_i^m y^i x^i + \frac{2}{m} \sum_i^m x^i x^{i\top} \theta = 0$

- Equivalent to

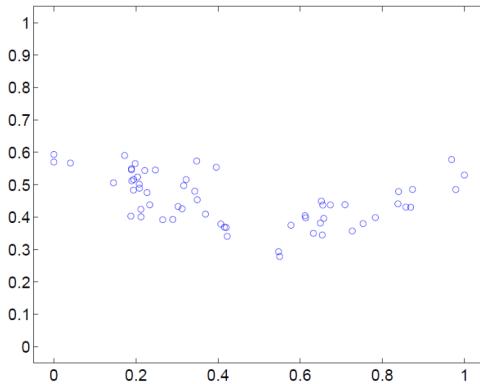
$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m} (x^1 \dots, x^m) (y^1 \dots, y^m)^\top + \frac{2}{m} (x^1, \dots, x^m) (x^1, \dots, x^m)^\top \theta = 0$$

- Define  $X = (x^1, x^2, \dots, x^m)$ ,  $y = (y^1, y^2, \dots, y^m)^\top$ , gradient becomes

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m} Xy + \frac{2}{m} XX^\top \theta$$

$$\Rightarrow \hat{\theta} = (XX^\top)^{-1}Xy$$

# Nonlinear regression



- Want to fit a polynomial regression model

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^n + \epsilon$$

- Let  $\tilde{x} = (1, x, x^2, \dots, x^n)^\top$  and  $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)^\top$

$$y = \theta^\top \tilde{x}$$

# Least mean square method

- Given  $m$  data points, find  $\theta$  that minimizes the mean square error

$$\hat{\theta} = \operatorname{argmin}_{\theta} L(\theta) = \frac{1}{m} \sum_{i=1}^m (y^i - \theta^\top \tilde{x}^i)^2$$

- Our usual trick: set gradient to 0 and find parameter

$$\begin{aligned}\frac{\partial L(\theta)}{\partial \theta} &= -\frac{2}{m} \sum_i^m (y^i - \theta^\top \tilde{x}^i) \tilde{x}^i = 0 \\ \Leftrightarrow -\frac{2}{m} \sum_i^m y^i \tilde{x}^i + \frac{2}{m} \sum_i^m \tilde{x}^i \tilde{x}^{i\top} \theta &= 0\end{aligned}$$

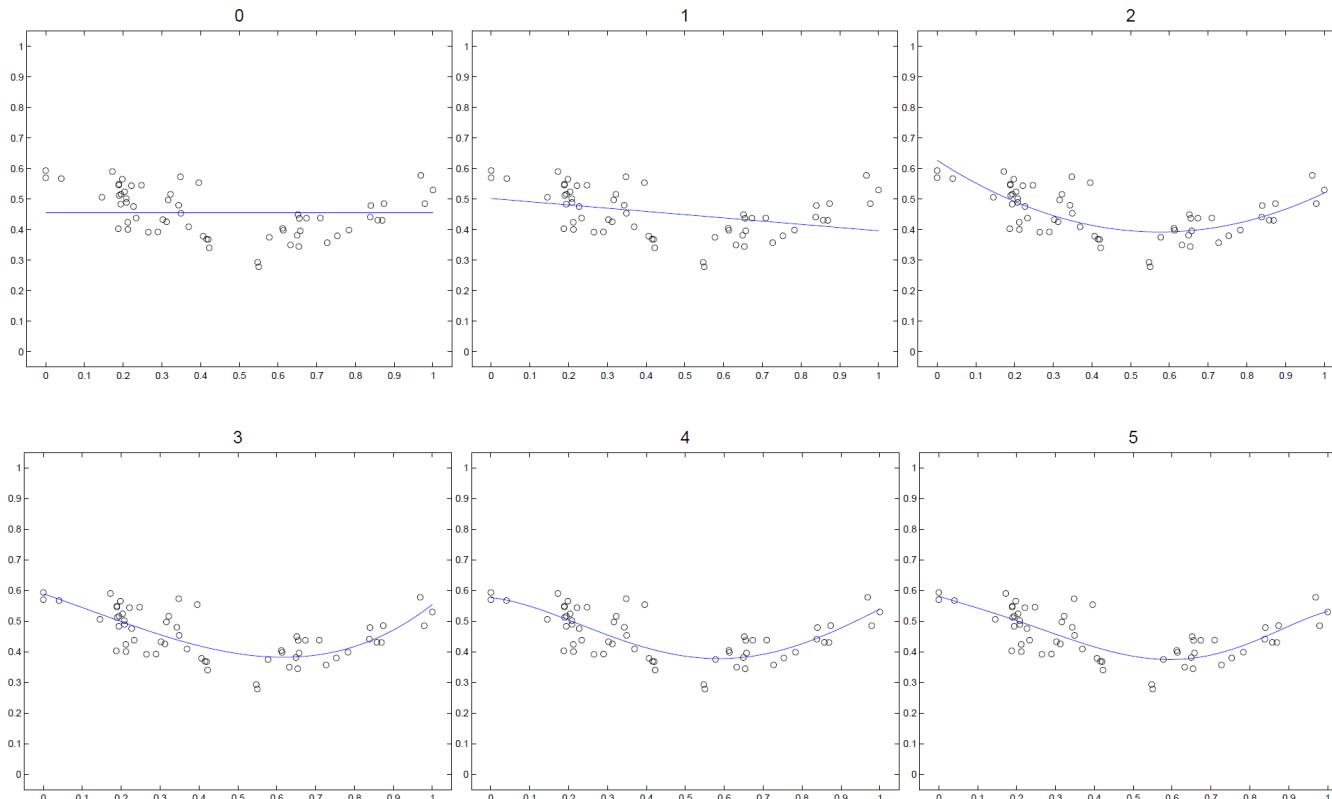
# Matrix version of the gradient

- Define  $\tilde{X} = (\tilde{x}^{(1)}, \tilde{x}^{(2)}, \dots, \tilde{x}^{(m)})$ ,  $y = (y^{(1)}, y^{(2)}, \dots, y^{(m)})^\top$ , gradient becomes

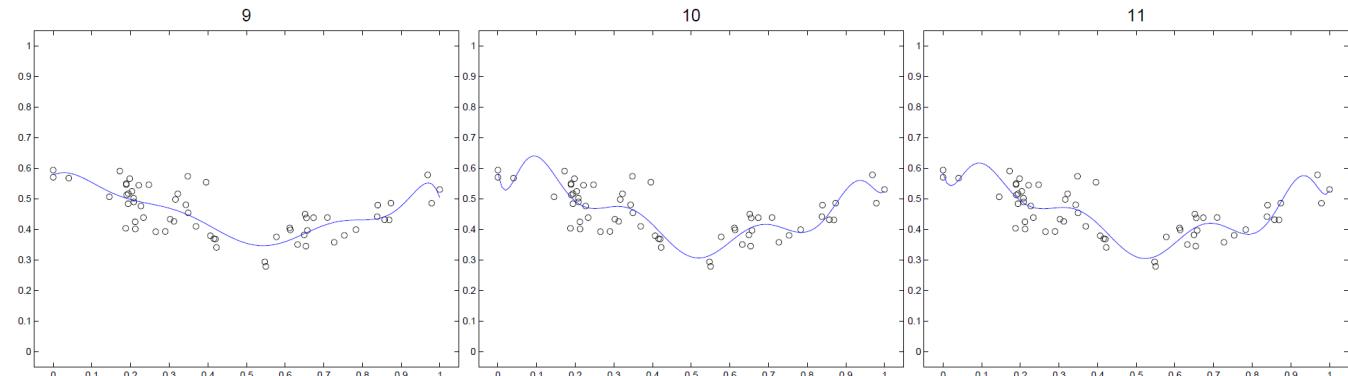
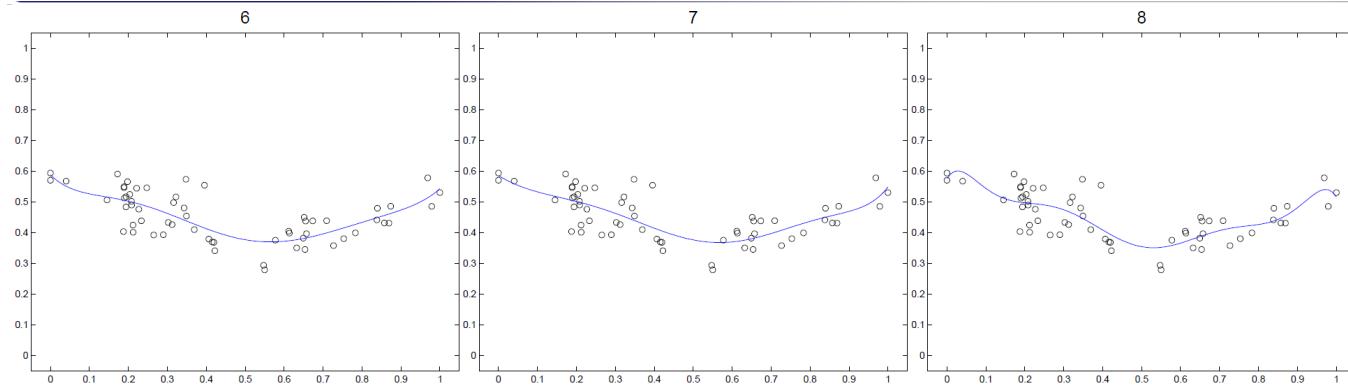
$$\begin{aligned}\frac{\partial L(\theta)}{\partial \theta} &= -\frac{2}{m} \tilde{X}y + \frac{2}{m} \tilde{X}\tilde{X}^\top \theta = 0 \\ \Rightarrow \hat{\theta} &= (\tilde{X}\tilde{X}^\top)^{-1} \tilde{X}y\end{aligned}$$

- Note that  $\tilde{x} = (1, x, x^2, \dots, x^n)^\top$
- If we choose a different maximal degree  $n$  for the polynomial, the solution will be different.

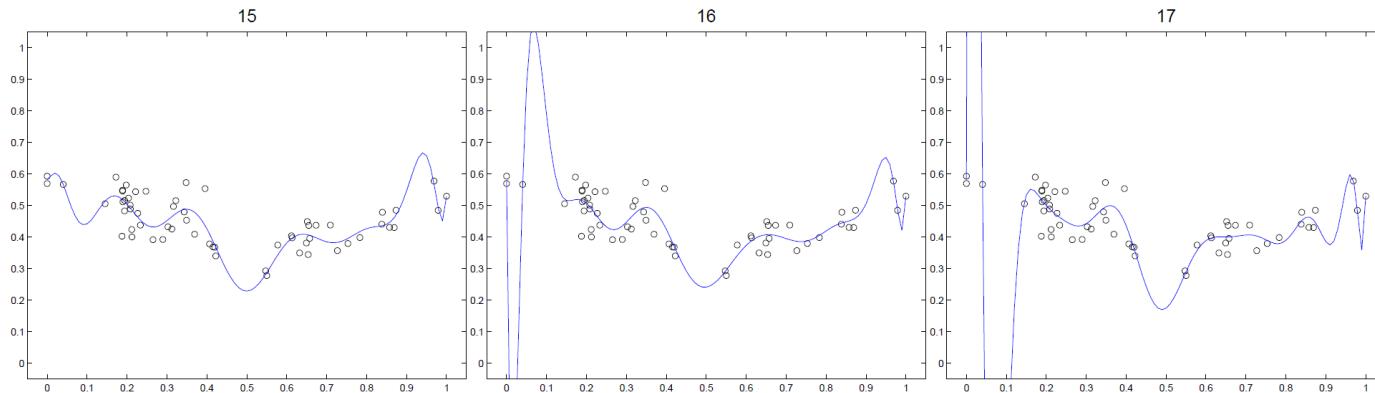
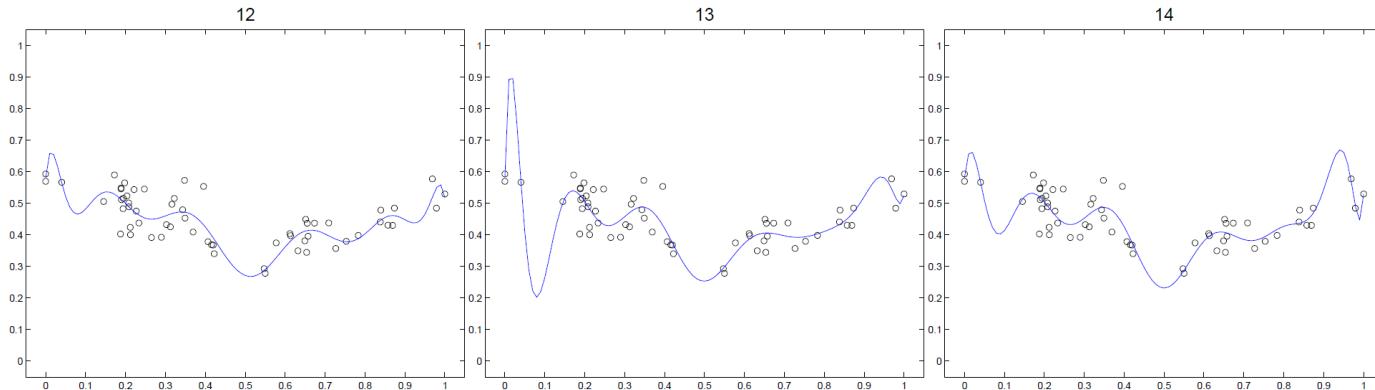
# Increasing the maximal degree



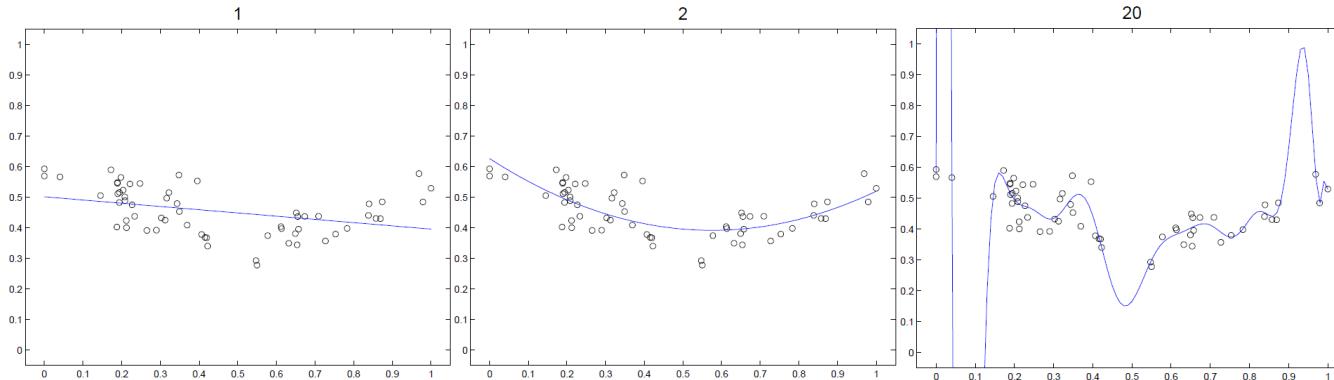
# Increasing the maximal degree



# Increasing the maximal degree



# Which one is better?



- Can we increase the maximal polynomial degree to very large, such that the curve passes through all training points?
- The optimization does not prevent us from doing that

# When maximal degree is very large

Define  $\tilde{X} = (\tilde{x}^{(1)}, \tilde{x}^{(2)}, \dots, \tilde{x}^{(m)})$ ,  $y = (y^{(1)}, y^{(2)}, \dots, y^{(m)})^\top$ , set gradient to zero,

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m} \tilde{X}y + \frac{2}{m} \tilde{X}\tilde{X}^\top \theta = 0$$

$$\Rightarrow \tilde{X}\tilde{X}^\top \theta = \tilde{X}y$$

Each  $\tilde{x} = (1, x, x^2, \dots, x^n)^\top$  is a vector of polynomial features, the size of  $\tilde{X}$  is  $n \times m$ , and  $\tilde{X}\tilde{X}^\top$  is  $n \times n$

When  $n > m$ ,

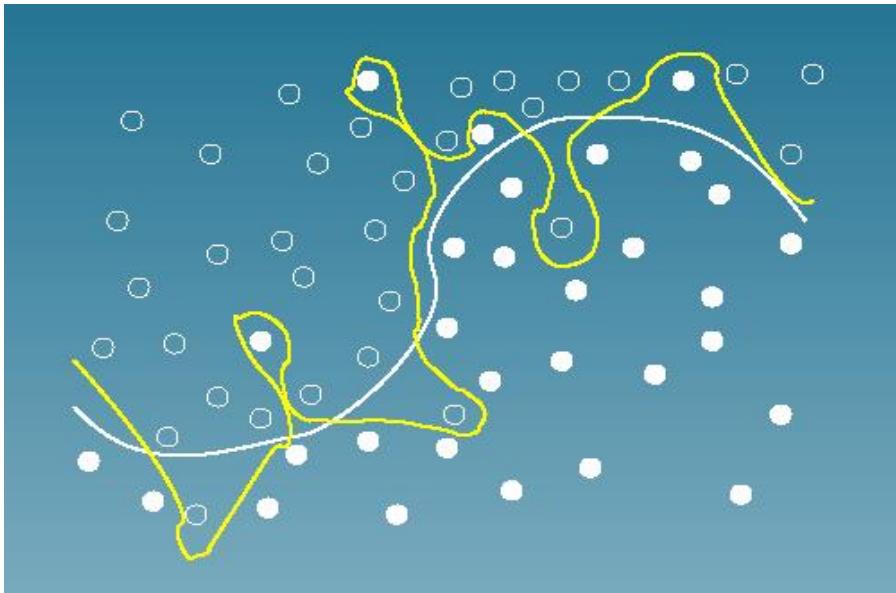
$\tilde{X}\tilde{X}^\top$  is not invertible; there are multiple solutions  $\theta$  which give zero objective

$$L(\theta) = \frac{1}{m} \sum_{i=1}^m (y^i - \theta^\top \tilde{x}^i)^2$$

# Classification with polynomials

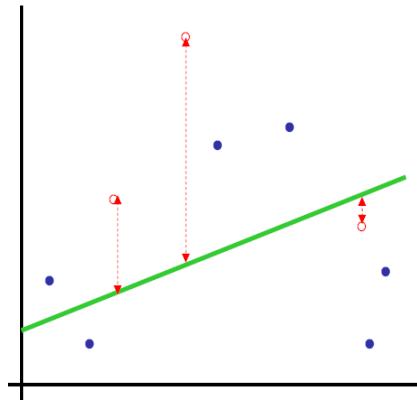
- Eg. Logistic regression with polynomial features

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^\top \tilde{x})}$$

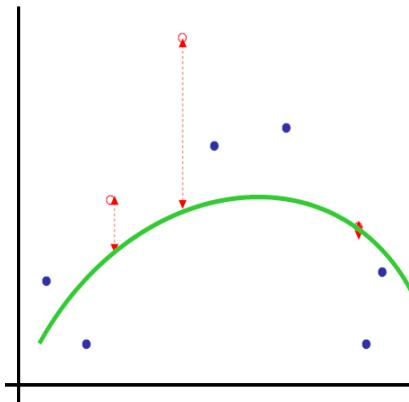


# Overfitting/Underfitting

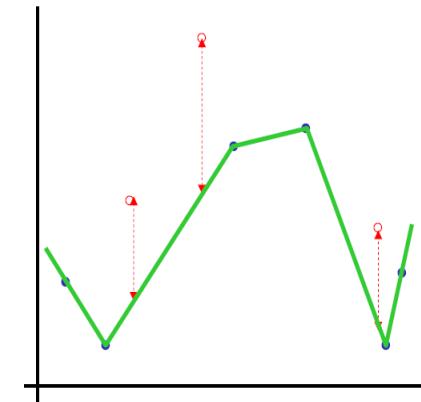
- Blue points: training data points, Red points: test data points
- The fit in the middle panel achieves a balance of small error in both training and test points.



Underfitting



Good fit



Overfitting

# What is the problem?

- Given  $m$  training data points  $D = \{(\tilde{x}^i, y^i)\}$ , find  $\theta$  that minimizes the mean square error

$$\hat{\theta} = \operatorname{argmin}_{\theta} \hat{L}(\theta) := \frac{1}{m} \sum_{i=1}^m (y^i - \theta^\top \tilde{x}^i)^2$$

- But we really want to minimize the error for unseen "test" data points, or with respect to the entire distribution of data

$$\theta^* = \operatorname{argmin}_{\theta} L(\theta) := \mathbb{E}_{(\tilde{x}, y) \sim P(\tilde{x}, y)} [(y - \theta^\top \tilde{x})^2]$$

- It is the finite number training point that creates the problem

# Decomposition of expected loss

- Estimate your function from a finite data set  $D$

$$\hat{f} = \operatorname{argmin}_f \hat{L}(f) := \frac{1}{m} \sum_{i=1}^m (y^i - f(x^i))^2$$

- $\hat{f}$  is a random function, generally different for different data set and depend on distribution of training data  $D$
- Expected loss of  $\hat{f}$

$$L(\hat{f}) := \mathbb{E}_D \mathbb{E}_{(x,y)} \left[ (y - \hat{f}(x))^2 \right]$$

$(x, y)$  denotes test data.

- Bias-variance decomposition

$$\text{Expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

# What is the best we can do?

The expected squared loss is

$$\begin{aligned} L(\hat{f}) &:= \mathbb{E}_D \mathbb{E}_{(x,y)} \left[ (y - \hat{f}(x))^2 \right] \\ &= \mathbb{E}_D \left[ \int \int (y - \hat{f}(x))^2 p(x, y) dx dy \right] \end{aligned}$$

Our goal is to choose  $\hat{f}(x)$  that minimize  $L(\hat{f})$ . Calculus of variations

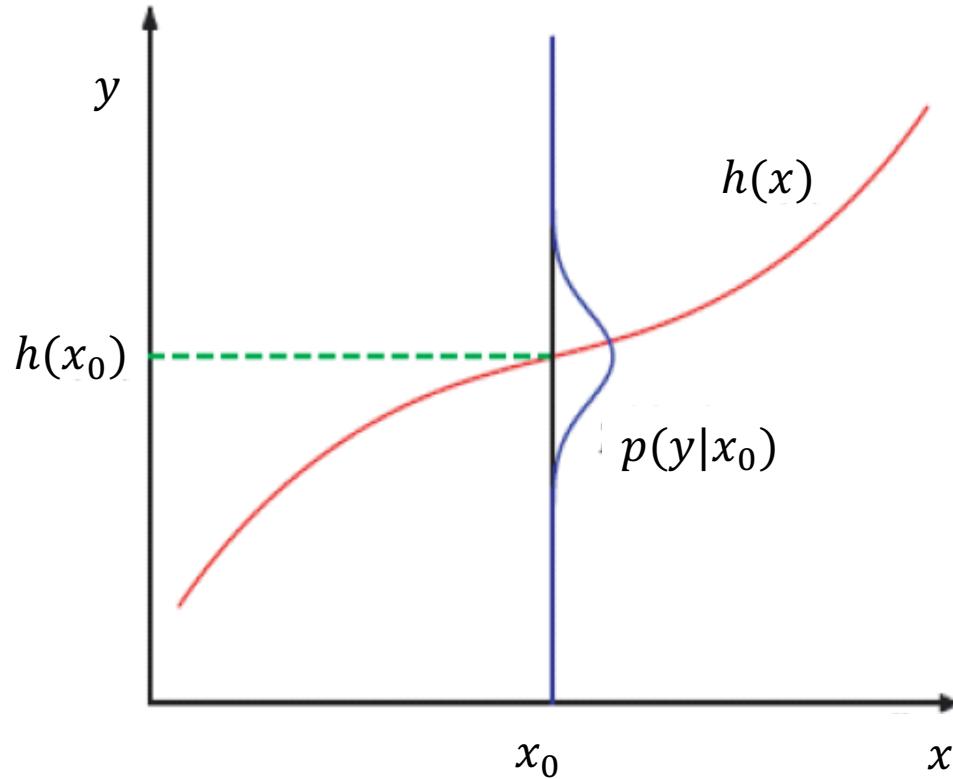
$$\frac{\partial \mathcal{A}}{\partial f(x)} = 2 \int (y - f(x)) p(x, y) dy = 0$$

$$\Leftrightarrow \int f(x) p(x, y) dy = \int y p(x, y) dy$$

$$\Leftrightarrow h(x) := \int \frac{y p(x, y)}{p(x)} dy = \int y p(y|x) dt = \mathbb{E}_{y|x}[y] = \mathbb{E}[y|x]$$

# The best predictor is the expected value

The best you can do is  $h(x) = \mathbb{E}_{y|x}[y]$ : the expected value of  $y$  given a particular  $x$



# Noise term in the decomposition

$h(x) = \mathbb{E}(y|x)$  is the **optimal** predictor, and  $\hat{f}(x)$  our actual predictor, decompose the error a bit

$$\begin{aligned}\mathbb{E}_D \mathbb{E}_{(x,y)} \left[ (y - \hat{f}(x))^2 \right] &= \mathbb{E}_D \left[ \int \int (y - h(x) + h(x) - \hat{f}(x))^2 p(x,y) dx dy \right] \\ &= \mathbb{E}_D \left[ \int \int \left( (\hat{f}(x) - h(x))^2 + 2(\hat{f}(x) - h(x))(h(x) - y) \right. \right. \\ &\quad \left. \left. + (h(x) - y)^2 \right) p(x,y) dx dy \right] \\ &= \mathbb{E}_D \left[ \int (\hat{f}(x) - h(x))^2 p(x) dx \right] + \int \int (h(x) - y)^2 p(x,y) dx dy\end{aligned}$$



Will decompose further



Noise term. can not do better than this.  
a lower bound of the expected loss

# Bias-variance decomposition

$\hat{f}(x)$  is a random function, generally different for different dataset  $D$

$\mathbb{E}_D[\hat{f}(x)]$  : expected value of  $\hat{f}(x)$  with respect to random dataset

$$\mathbb{E}_D \left[ \int (\hat{f}(x) - h(x))^2 p(x) dx \right] = \mathbb{E}_D \mathbb{E}_x \left[ (\hat{f}(x) - h(x))^2 \right]$$

$$= \mathbb{E}_x \mathbb{E}_D \left[ (\hat{f}(x) - \mathbb{E}_D[\hat{f}(x)] + \mathbb{E}_D[\hat{f}(x)] - h(x))^2 \right]$$

$$\begin{aligned} &= \mathbb{E}_x \mathbb{E}_D \left[ (\hat{f}(x) - \mathbb{E}_D[\hat{f}(x)])^2 \right] + \mathbb{E}_x \mathbb{E}_D \left[ (\mathbb{E}_D[\hat{f}(x)] - h(x))^2 \right] \\ &\quad - 2 \mathbb{E}_x \mathbb{E}_D \left[ (\hat{f}(x) - \mathbb{E}_D[\hat{f}(x)]) (\mathbb{E}_D[\hat{f}(x)] - h(x)) \right] \end{aligned}$$

$$= \underbrace{\mathbb{E}_x \mathbb{E}_D \left[ (\hat{f}(x) - \mathbb{E}_D[\hat{f}(x)])^2 \right]}_{\text{Variance}} + \underbrace{\mathbb{E}_x \left[ (\mathbb{E}_D[\hat{f}(x)] - h(x))^2 \right]}_{\text{Bias}^2}$$

Variance

Bias<sup>2</sup>

# Overall decomposition of expected loss

Putting things together

$$\text{Expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

In formula

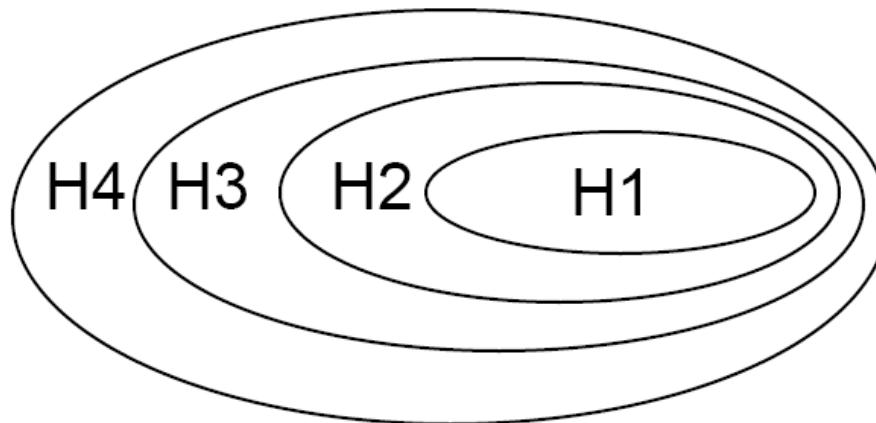
$$\begin{aligned} & \mathbb{E}_D \mathbb{E}_{(x,y)} \left[ (y - \hat{f}(x))^2 \right] \\ &= \mathbb{E}_x \left[ \left( \mathbb{E}_D [\hat{f}(x)] - h(x) \right)^2 \right] (\text{bias}^2) \\ &+ \mathbb{E}_x \mathbb{E}_D \left[ \left( \hat{f}(x) - \mathbb{E}_D [\hat{f}(x)] \right)^2 \right] (\text{variance}) \\ &+ \mathbb{E}_{(x,y)} [(h(x) - y)^2] (\text{noise}) \end{aligned}$$

Key quantities

- $\hat{f}(x)$ : actual predictor
- $\mathbb{E}_D [\hat{f}(x)]$ : expected predictor
- $h(x) = \mathbb{E}(y|x)$ : **optimal** predictor

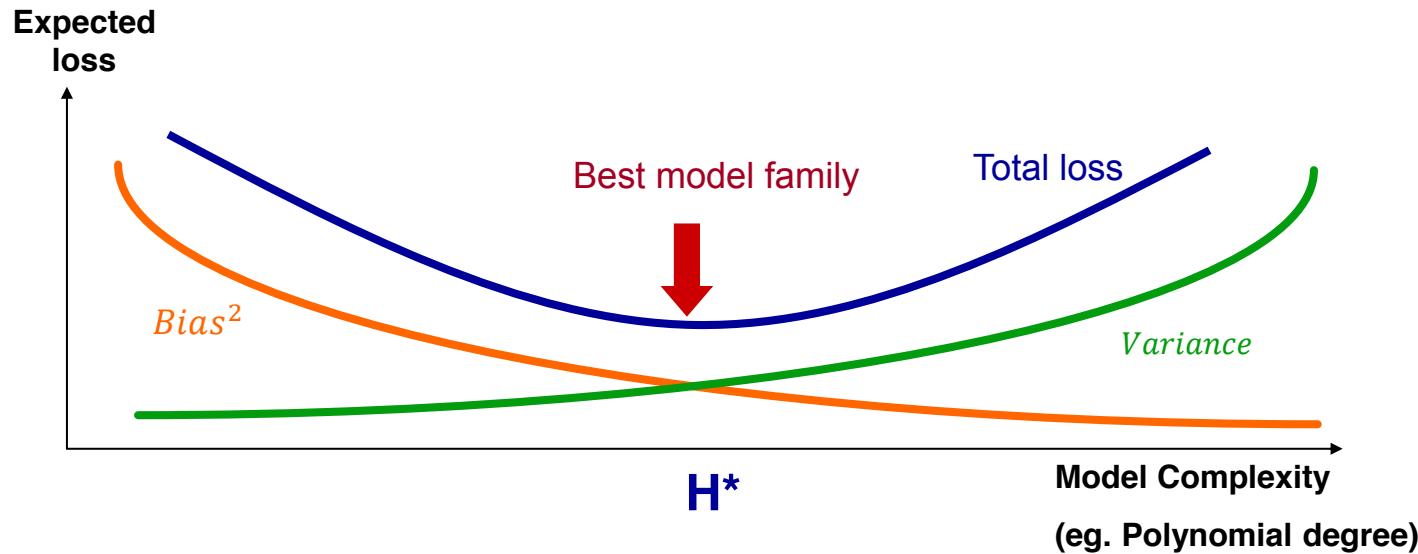
# Model space

- Which model space should we choose?
- The more complex the model, the large the model space
- Eg. Polynomial function of degree 1, 2, ... corresponds to space H1, H2 ...



# Intuition of model selection

Find the right model family s.t. expected loss becomes minimum



# Other things that control model complexity

Eg. In the case of linear models  $y = \langle w, x \rangle + b$ , one wants to make  $\|w\|$  a controlled parameter

$$\|w\| < C$$

- $H_C$  the linear model function family satisfying the constraint
- The larger the  $C$ , the larger the model family

Eg. the larger the regularization parameter  $\lambda$ , the smaller the model family

- $J(w) = \sum_i (w^\top x_i - y_i)^2 + \lambda \|w\|_2^2$
- $J(w) = \sum_i (w^\top x_i - y_i)^2 + \lambda \|w\|_1$

Eg. Early stopping in boosting

# Ridge regression

Given  $m$  data points, find  $\theta$  that minimizes the **regularized** mean square error

$$\theta^r = \operatorname{argmin}_{\theta} L(\theta) = \frac{1}{m} \sum_{i=1}^m (y^i - \theta^\top x^i)^2 + \lambda \|\theta\|^2$$

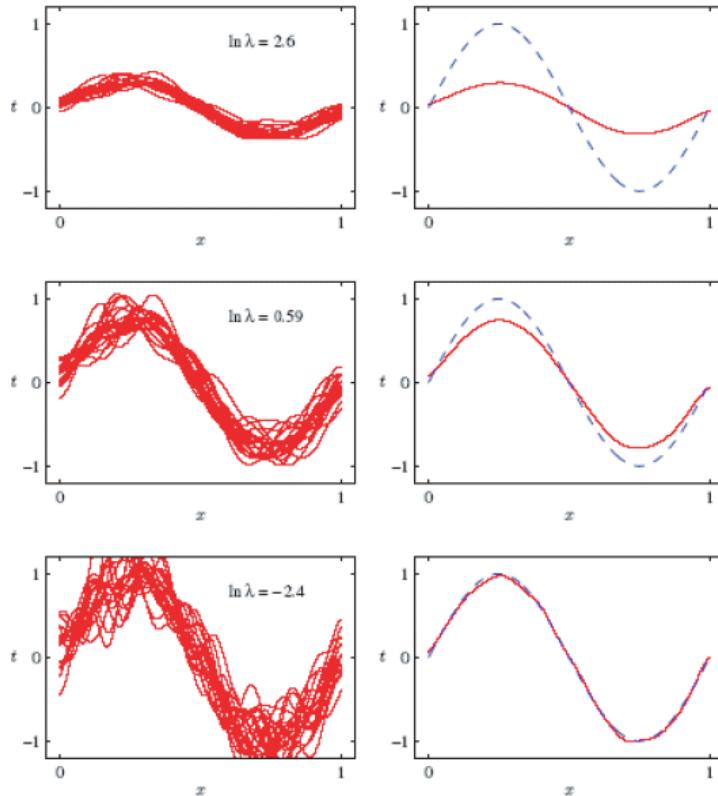
gradient becomes

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m} X y + \frac{2}{m} X X^\top \theta + \frac{2\lambda}{m} \theta = 0$$

$$\Rightarrow \theta^r = (X X^\top + \lambda I)^{-1} X y$$

If we choose a different  $\lambda$ , the solution will be different.

# Experiment with bias-variance tradeoff



- $\lambda$  is a "regularization" terms in LR, the smaller the  $\lambda$ , is more complex the model
  - Simple (highly regularized) models have low variance but high bias.
  - Complex models have low bias but high variance.
- The actual  $\mathbb{E}_D$  can not be computed
- You are inspecting an empirical average over 100 training set.

# Cross-Validation

## $K$ -fold cross-validation (CV)

For each fold  $i$ :

- Set aside  $\alpha \cdot m$  samples of  $D$  (where  $m = |D|$ ) as the **held-out data**. They will be used to evaluate the error
- Fit a model  $f_i(x)$  to the remaining  $(1 - \alpha) \cdot m$  samples in  $D$
- Calculate the error of the model  $f_i(x)$  on the held-out data.

Repeat the above  **$K$  times**, choosing a **different** held-out data set each time, and the errors are averaged over the folds.

For the polynomial degree with the lowest score, we use all of  $D$  to find the parameter values for  $f(x)$ .

# Cross-Validation

- Eg. Want to select the maximal degree of polynomial
- 5-fold cross-validation (blank: training; red: test)

Data:    1               ...                  $m$

Fold 1:      $\Rightarrow f_1(x)$      $\Rightarrow \text{error 1}$

Fold 2:      $\Rightarrow f_2(x)$      $\Rightarrow \text{error 2}$

Fold 3:      $\Rightarrow f_3(x)$      $\Rightarrow \text{error 3}$

Fold 4:      $\Rightarrow f_4(x)$      $\Rightarrow \text{error 4}$

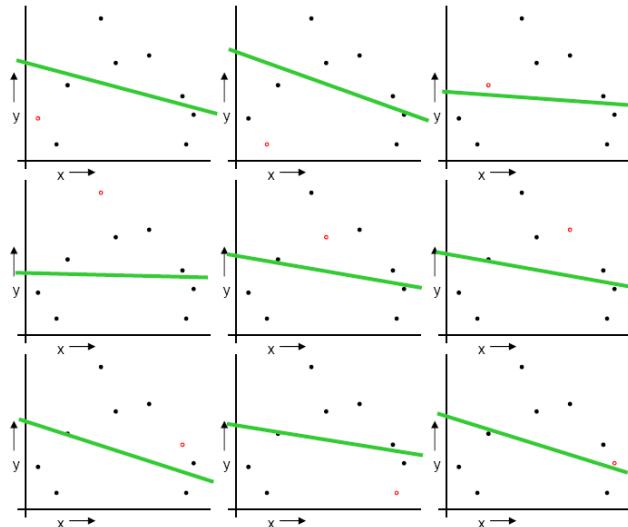
Fold 5:      $\Rightarrow f_5(x)$      $\Rightarrow \text{error 5}$

average = cross validation error

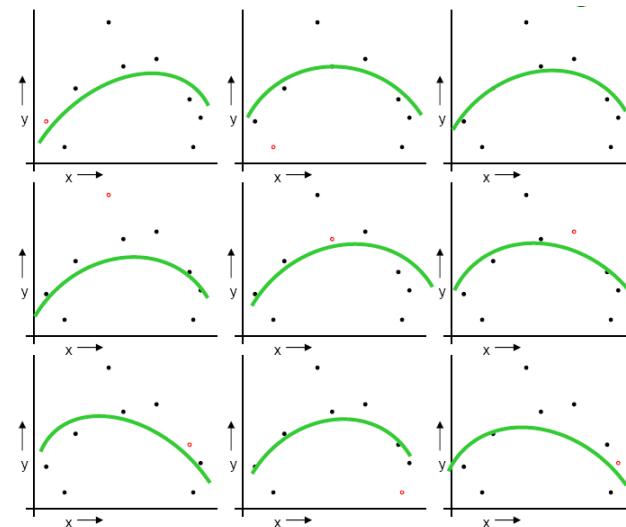
- Important: test data  $i$  is **not** used to fit model  $f_i(x)$

# Example:

- When  $\alpha = 1/N$ , the algorithm is known as Leave-One-Out-Cross-Validation (LOOCV)



$$MSELOOCV(M_1)=2.12$$



$$MSELOOCV(M_2)=0.962$$

# Practical issues for K-fold CV

How to decide the values for  $K$  (or  $\alpha$ )

- Commonly used  $K = 10$  or ( $\alpha = 0.1$ ).
- Large  $K$  makes it time-consuming.
- Bias-variance trade-off
  - Large  $K$  usually leads to low bias. In principle,  $LOOCV$  provides an almost unbiased estimate of the generalization ability of a classifier, but it can also have high variance.
  - Small  $K$  can reduce variance, but will lead to under-use of data, and causing high-bias.

One important point is that the test data  $D_{test}$  is never used in CV, because doing so would result in overly (indeed dishonest) optimistic accuracy rates during the testing phase.

# Ridge regression

Given  $m$  data points, find  $\theta$  that minimizes the **regularized** mean square error

$$\theta^r = \operatorname{argmin}_{\theta} L(\theta) = \frac{1}{m} \sum_{i=1}^m (y^i - \theta^\top x^i)^2 + \lambda \|\theta\|^2$$

gradient becomes

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m} X y + \frac{2}{m} X X^\top \theta + 2\lambda \theta = 0$$

$$\Rightarrow \theta^r = \left( \frac{1}{m} X X^\top + \lambda I \right)^{-1} \left( \frac{1}{m} X y \right)$$

If we choose a different  $\lambda$ , the solution will be different.

# Example: Cross validation error curve for ridge regression

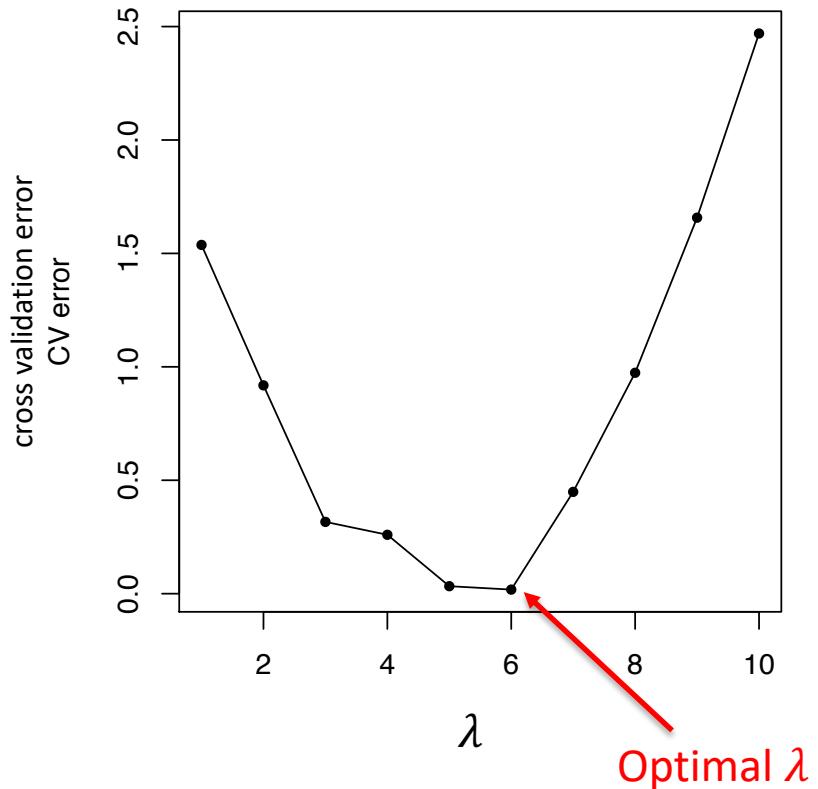
$$\min_{\theta} L(\theta) = \frac{1}{m} \sum_{i=1}^m (y^i - \theta^\top x^i)^2 + \lambda \|\theta\|^2$$

$$\theta^r = \left( \frac{1}{m} X X^T + \lambda I \right)^{-1} \left( \frac{1}{m} X y \right)$$

Error (for each fold, L samples) =

$$\frac{1}{L} \sum_{l=1}^L (\tilde{y}^l - (\theta^r)^T \tilde{x}^l)^2$$

CV Error = Average of error over all folds



# Regularization in maximum likelihood

Regularize the likelihood objective (also known as penalized likelihood, shrinkage, smoothing, etc.)

$$\hat{\theta}_{shrinkage} = \arg \max_{\theta} [l(\theta; D) - \lambda \|\theta\|]$$

where  $\lambda > 0$  and  $\|\theta\|$  might be the  $L_1$  or  $L_2$  norm.

The choice of norm has an effect

- using the  $L_2$  norm pulls directly towards the origin,
- while using the  $L_1$  norm pulls towards the coordinate axes, i.e it tries to set some of the coordinates to 0.

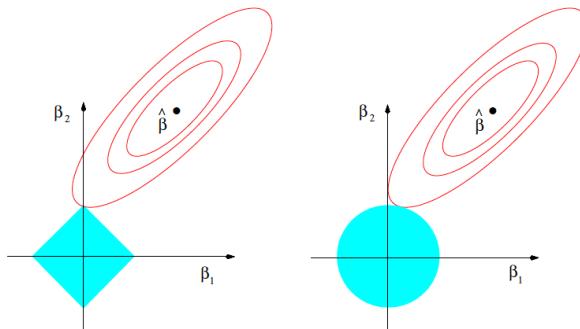
# Lasso for variable selection

- Least square method

$$\hat{\theta} = \operatorname{argmin}_{\theta} L(\theta) = \frac{1}{m} \sum_i^m (y^i - \theta^\top x^i)^2$$

- Encourage  $\theta$  to be sparse, i.e., having very few number of non-zeros, while having a good “fit”

$$\hat{\theta} = \operatorname{argmin}_{\theta} L(\theta) = \frac{1}{m} \sum_i^m (y^i - \theta^\top x^i)^2 + \|\theta\|_1$$



# Lasso

The lasso estimate

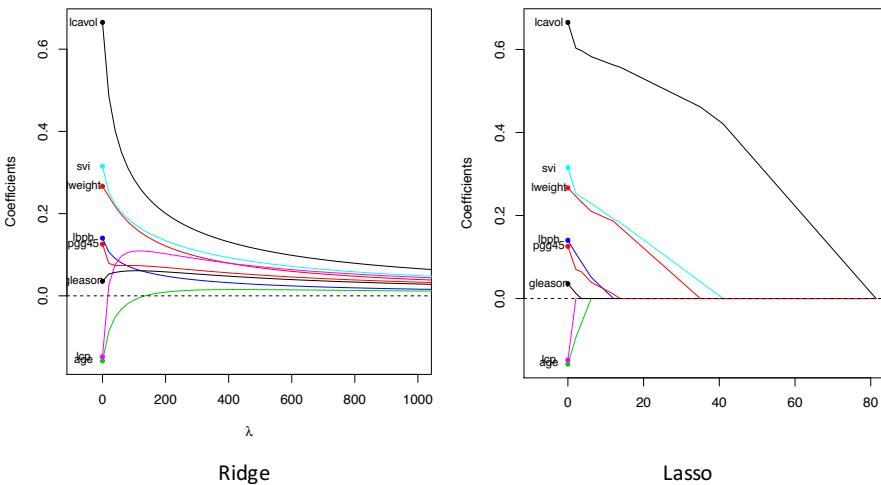
$$\min_{\beta} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

- ▶ The penalty term  $\|\beta\|_2^2$  is replaced by  $\ell_1$  norm  $\|\beta\|_1$
- ▶  $\lambda$  controls the strength of the penalty
- ▶ Lasso is able to perform model selection in the linear model
- ▶ As  $\lambda$  increases, more coefficients are set to 0 (less variables are selected)
- ▶ Among the nonzero coefficients, more shrinkage is employed

R. Tibshirani, 1996, Regression shrinkage and selection via the lasso.

# Example: prostate data

- ▶ We are interested in the level of prostate-specific antigen (PSA) elevated in men who has prostate cancer.
- ▶ Measure PSA on  $n = 97$  patients,  $p = 8$  clinical variables



- ▶ If we want the 3 leading factors, we report “cancer volume”, “seminal sesicle invasion”, “prostate weight”

