DS 4400

Machine Learning and Data Mining I
Spring 2024

David Liu

Khoury College of Computer Science

Northeastern University

January 30 2024

Announcements

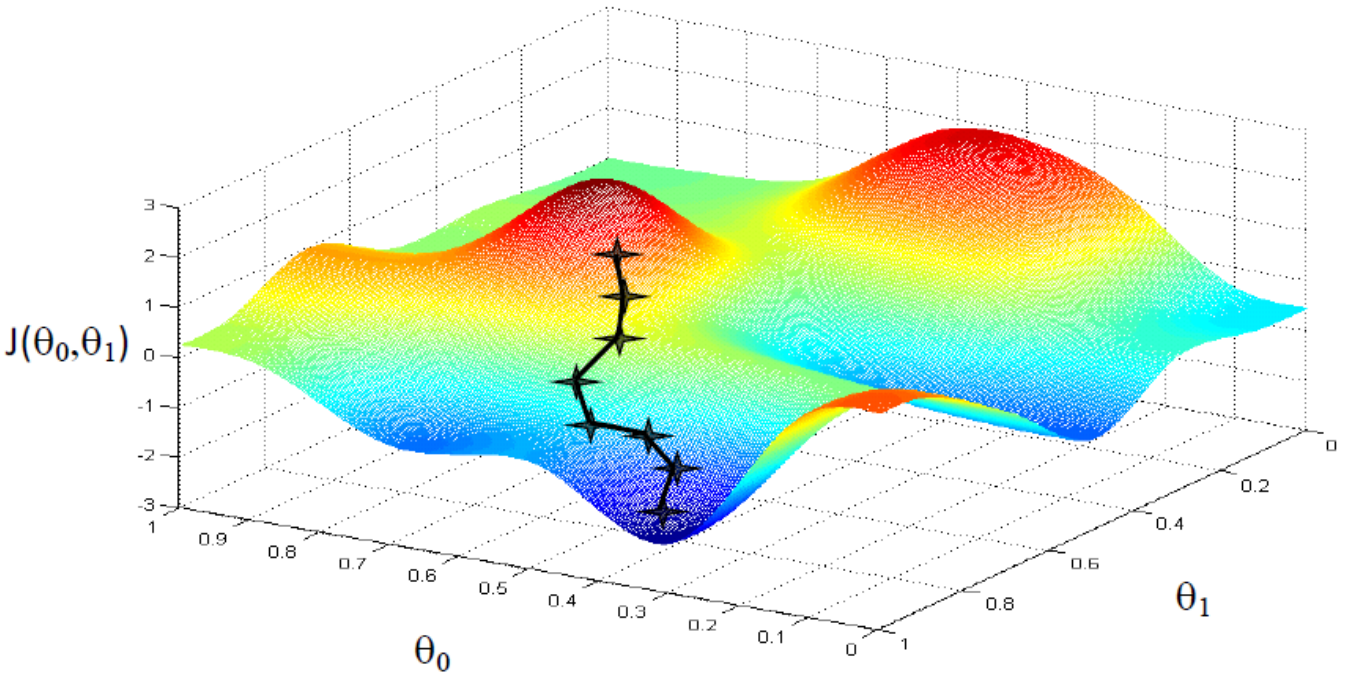
- Will start grading HW 1 after tonight's deadline
- Will release HW 2 this week
- Midterm exam on Friday Feb 23
- Will release further final project guidance this week.

Outline

- Review of Gradient Descent
- Non-linear regression
  - Polynomial regression
  - Cubic, spline regression
- Regularization
  - Ridge regression
  - Lasso regression
- Classification
  - K Nearest Neighbors (kNN)
  - Bias-Variance tradeoff

How to optimize $J(\theta)$?

- Choose initial value for θ
- Until we reach a minimum:
  - Choose a new value for θ to reduce $J(\theta)$



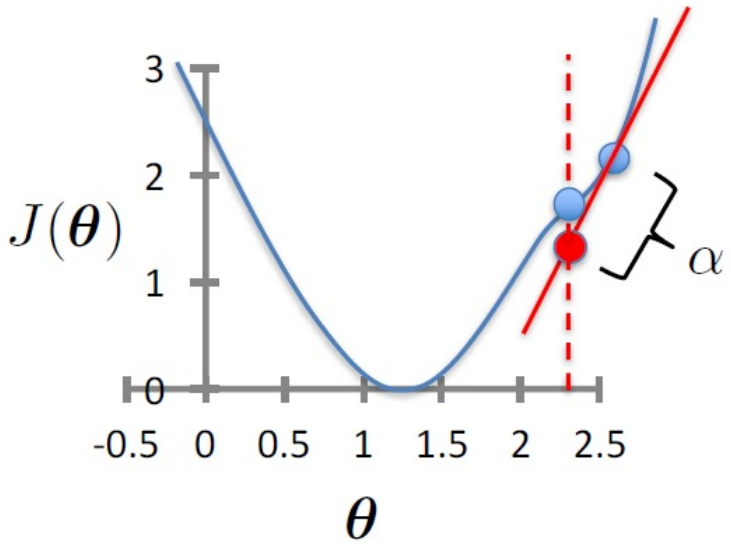
Gradient Descent

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

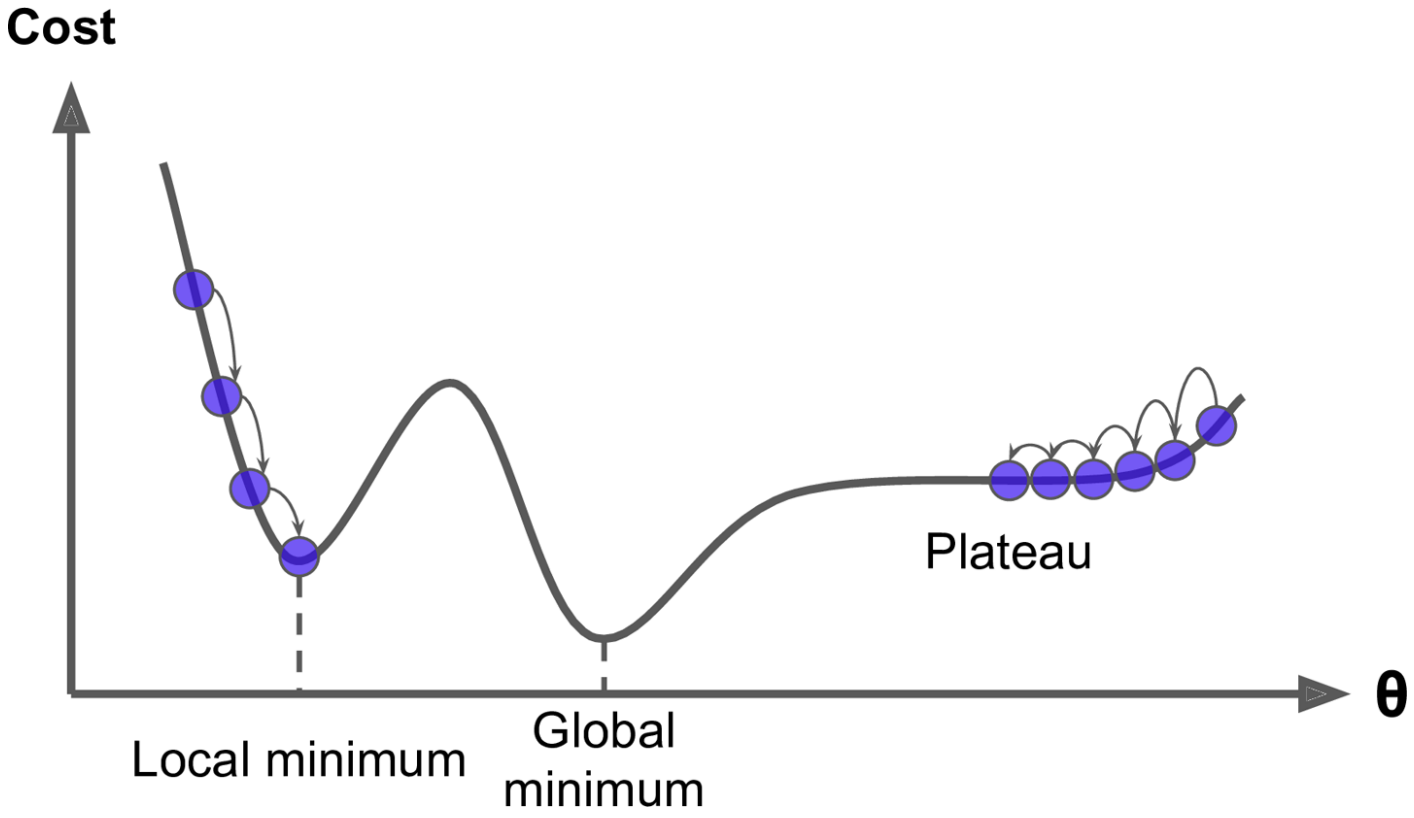
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

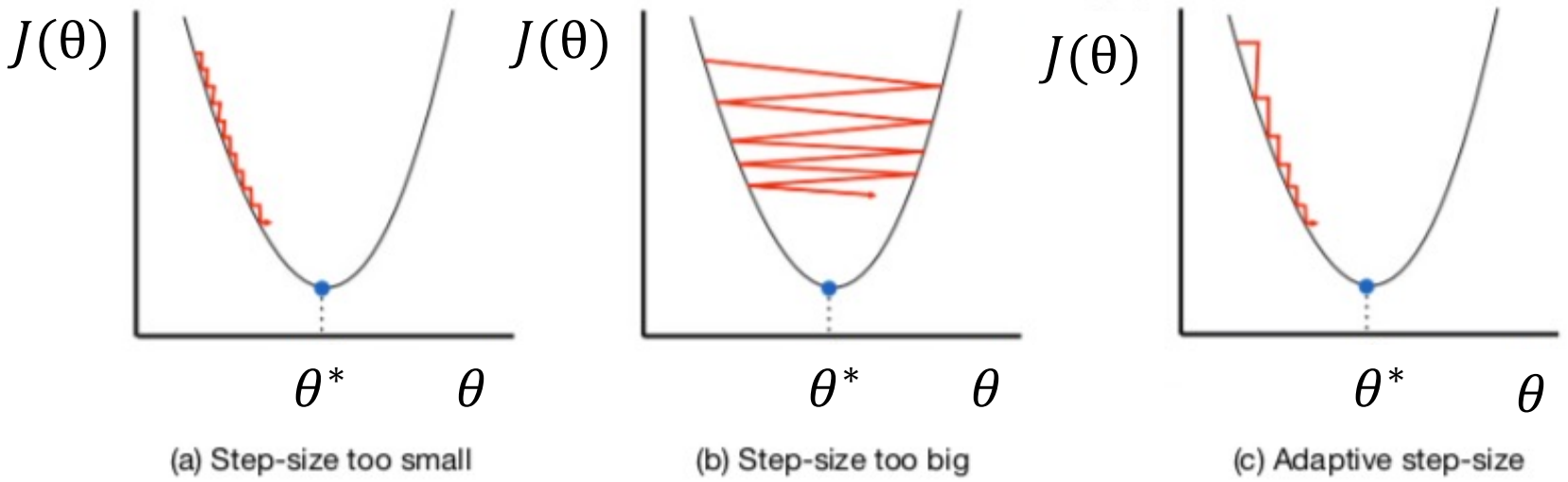
learning rate (small)
e.g., α = 0.05



GD Convergence Issues



Adaptive step size



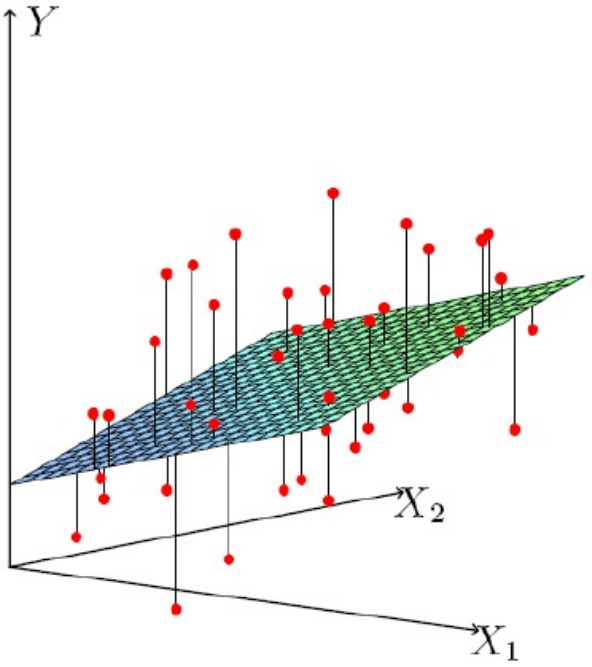
- Start with large step size and reduce over time, adaptively
- Line search method
- Measure how objective decreases

NON-LINEAR REGRESSION

Multiple Linear Regression

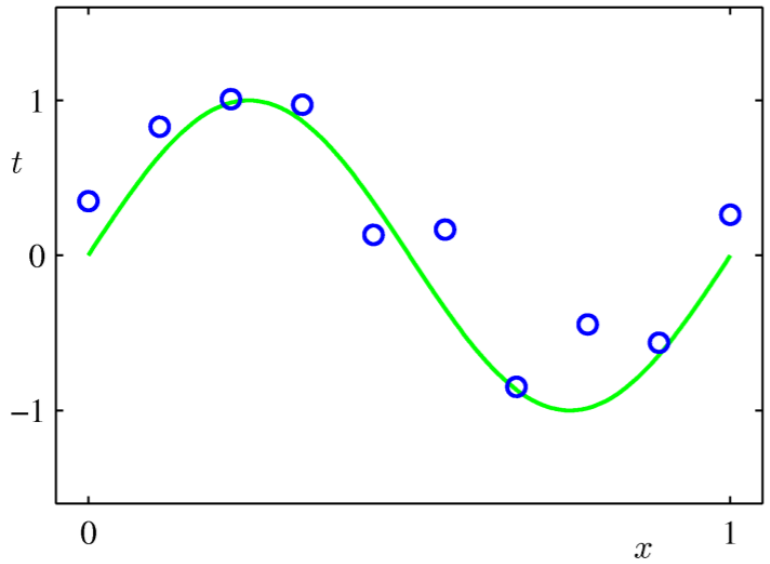
- Dataset: $x_i \in R^d, y_i \in R$
- Hypothesis $h_\theta(x) = \theta^T x$
- MSE $= \frac{1}{N}\sum (\theta^T x_i - y_i)^2$ Loss / cost

$$\theta = (X^\intercal X)^{-1} X^\intercal y$$

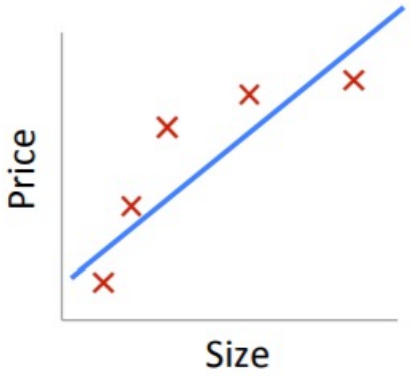


Polynomial Regression

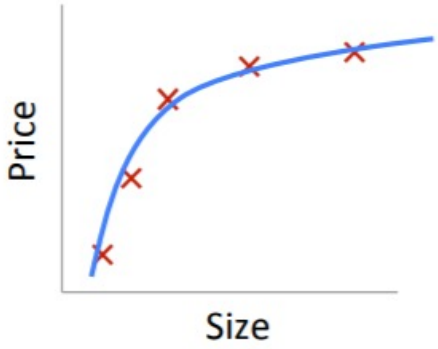
- Polynomial function on single feature
  - $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_p x^p$



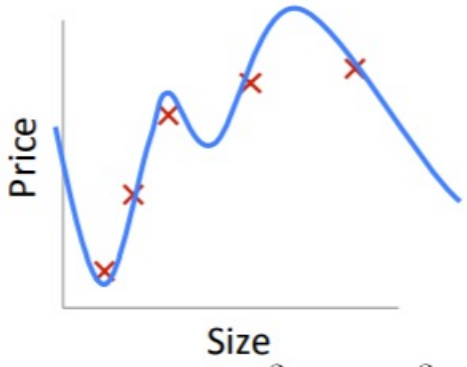
Polynomial Regression



$\theta_0 + \theta_1 x$



$\theta_0 + \theta_1 x + \theta_2 x^2$



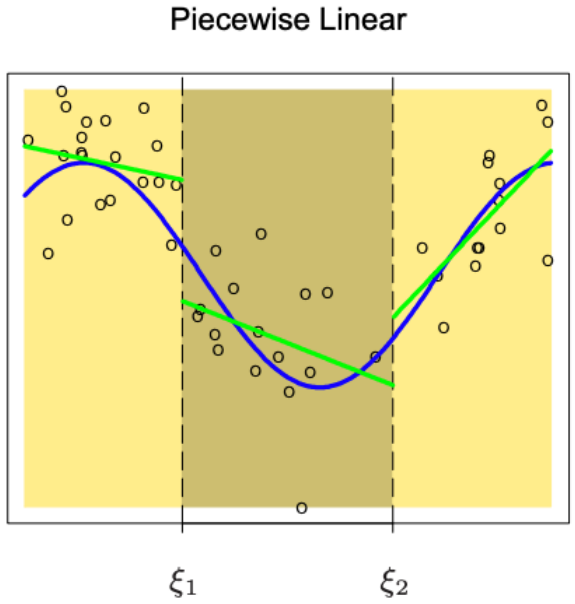
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

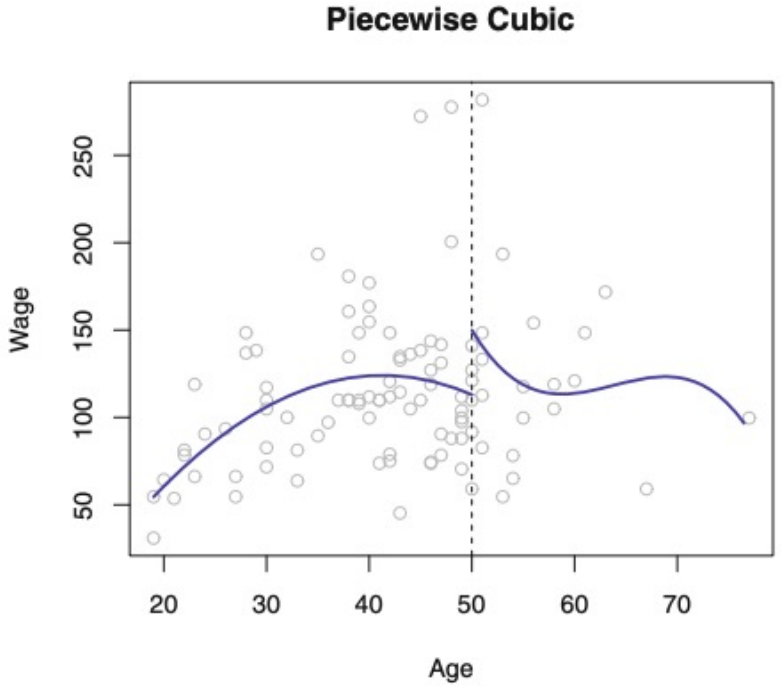
Polynomial Regression Training

- Simple Linear Regression
- $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_p x^p$
- How to train model?

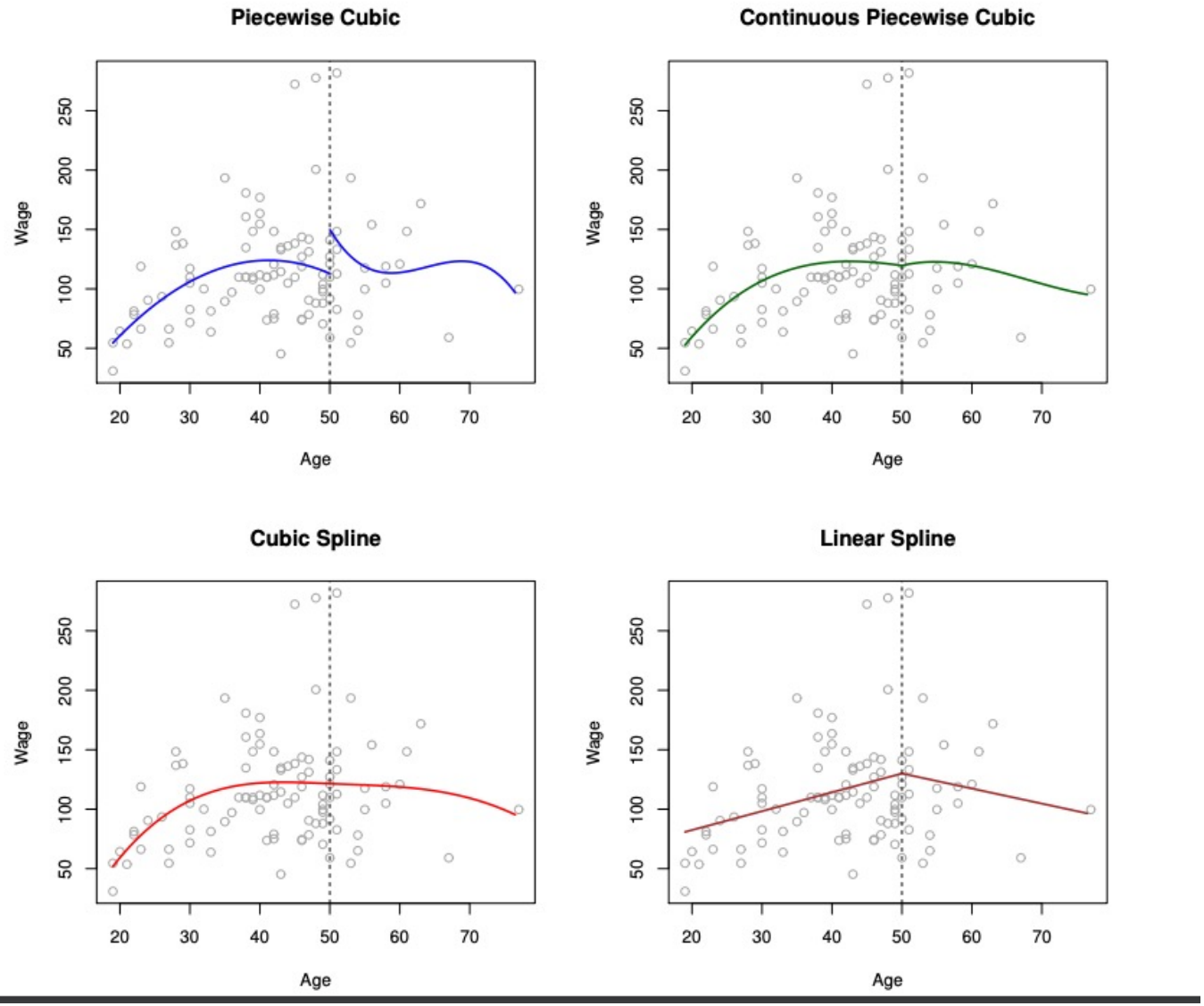
Piecewise Polynomial

- Divide the space into regions
- Polynomial regression on each region
  - Linear piecewise (degree 1), quadratic piecewise (degree 2), cubic piecewise (degree 3)

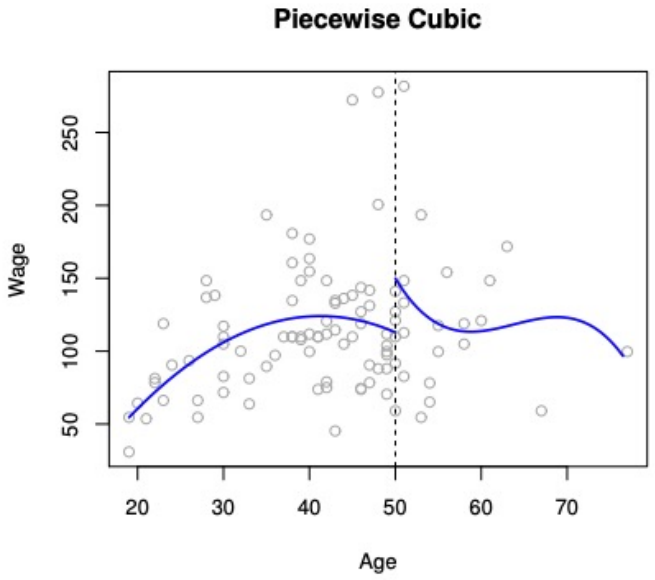


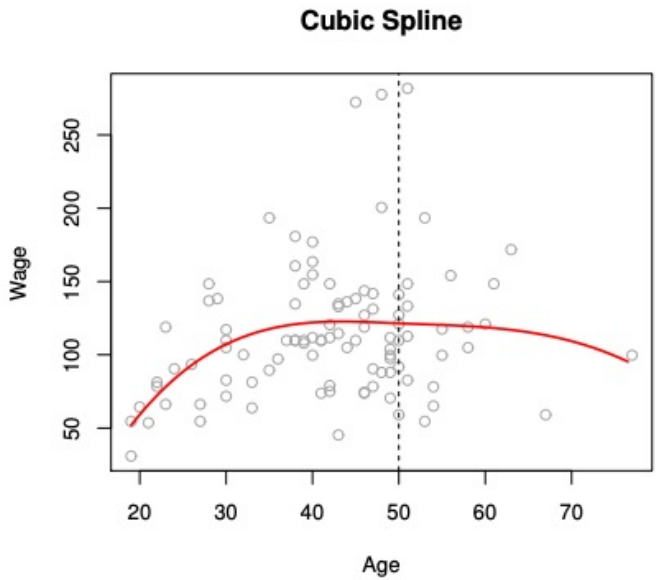


Piecewise and spline regression



Piecewise polynomial vs Regression spline





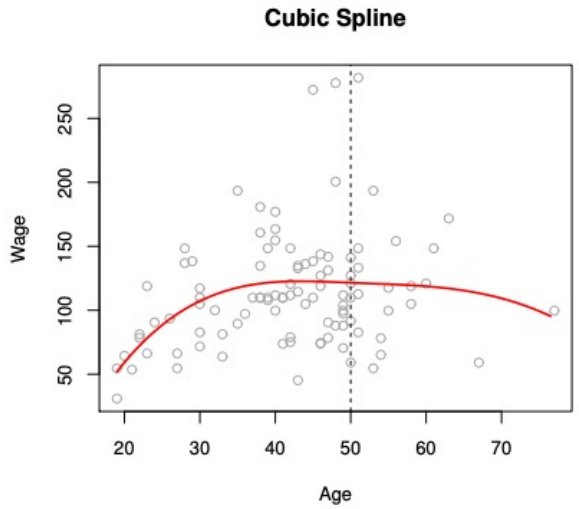
1 break at Age $= 50$

1 knot at Age $= 50$

Definition: Cubic spline

A cubic spline with knots at x-values $\xi_1, \ldots, \xi_K$ is a continuous piecewise cubic polynomial with *continuous derivates* and *continuous second derivatives* at each knot.

Cubic splines

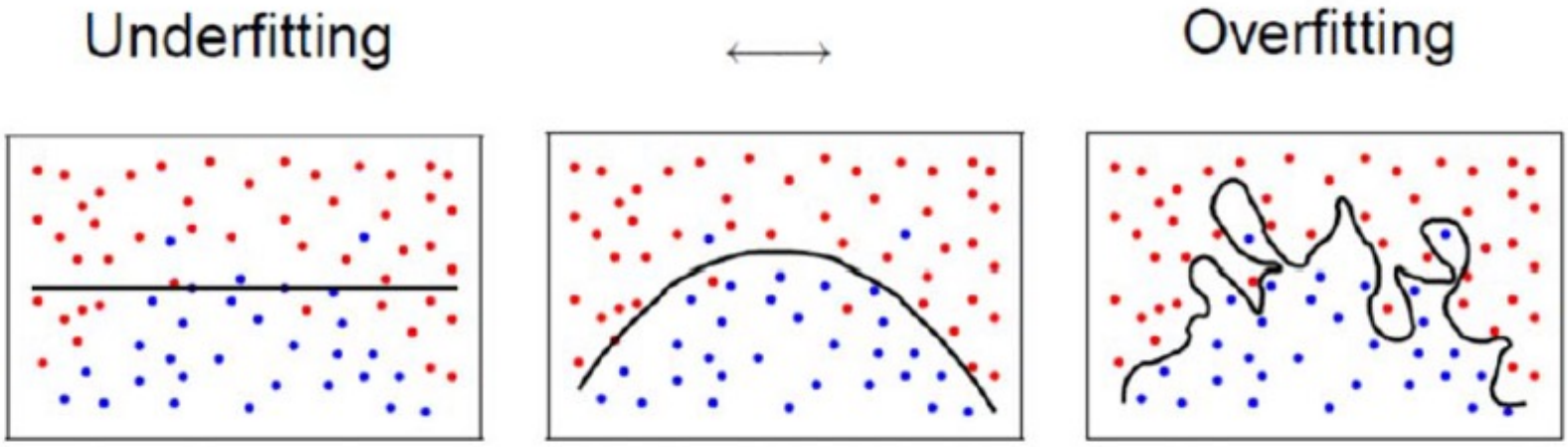


- Turns out, cubic splines are sufficiently flexible to *consistently* estimate smooth regression functions f
- You can use higher-degree splines, but *there's no need to*
- To fit a cubic spline, we just need to pick the knots

Additive Models

- Multiple Linear Regression Model
  - $y_i = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d$
- Additive Models
  - $y_i = \theta_0 + f_1(x_1) + \cdots + f_d(x_d)$
- Can instantiate functions f with:
  - Linear functions:
  - Quadratic:
  - Cubic:

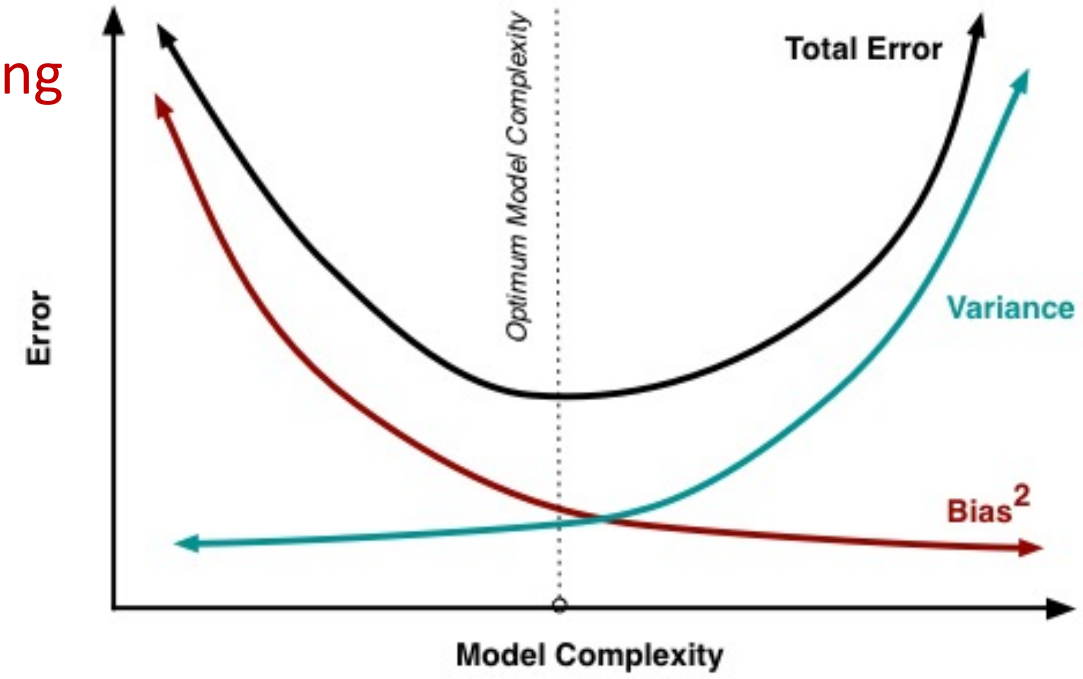
Generalization in ML



- Goal is to generalize well on new testing data
- Risk of overfitting to training data

Bias-Variance Tradeoff

Under-fitting



Over-fitting

- Bias = Difference between estimated and true models
- Variance = Model difference on different training sets

MSE is proportional to Bias + Variance

REGULARIZATION

Regularization

- A method for automatically controlling the complexity of the trained model
- Goals
  - Reduce model complexity
  - Reduce variance
  - Mitigate the bias-variance tradeoff
- Main techniques
  - Modify loss function to account for regularization term (Ridge, Lasso)
  - Perform feature selection and fit model on subset of features

Ridge regression

- Linear regression objective function

$$J(\theta) = \sum_{i=1}^{N} [h_\theta(x_i) - y_i]^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

Ridge regression

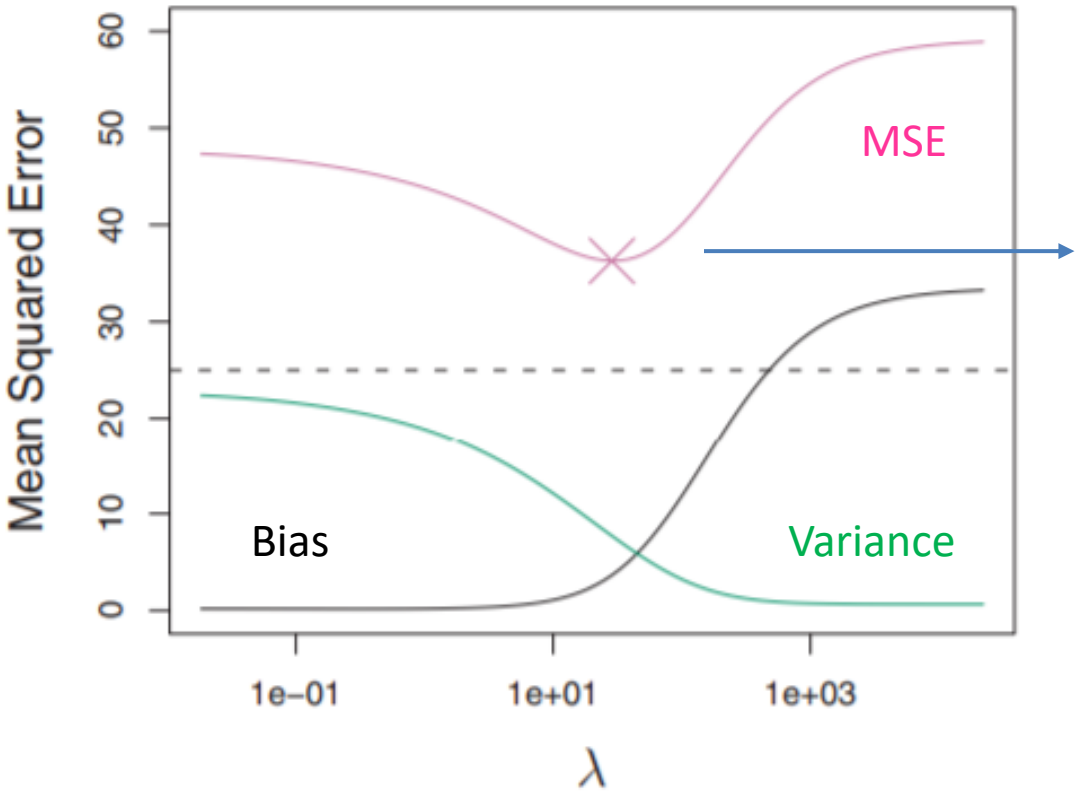
- Linear regression objective function

$$J(\theta) = \underbrace{\frac{1}{2}\sum_{i=1}^{N}[h_\theta(x_i) - y_i]^2}_{\text{model fit to data}} + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2}_{\text{regularization}}$$

  - λ is the regularization parameter ($\lambda \geq 0$)
  - No regularization on θ_0!

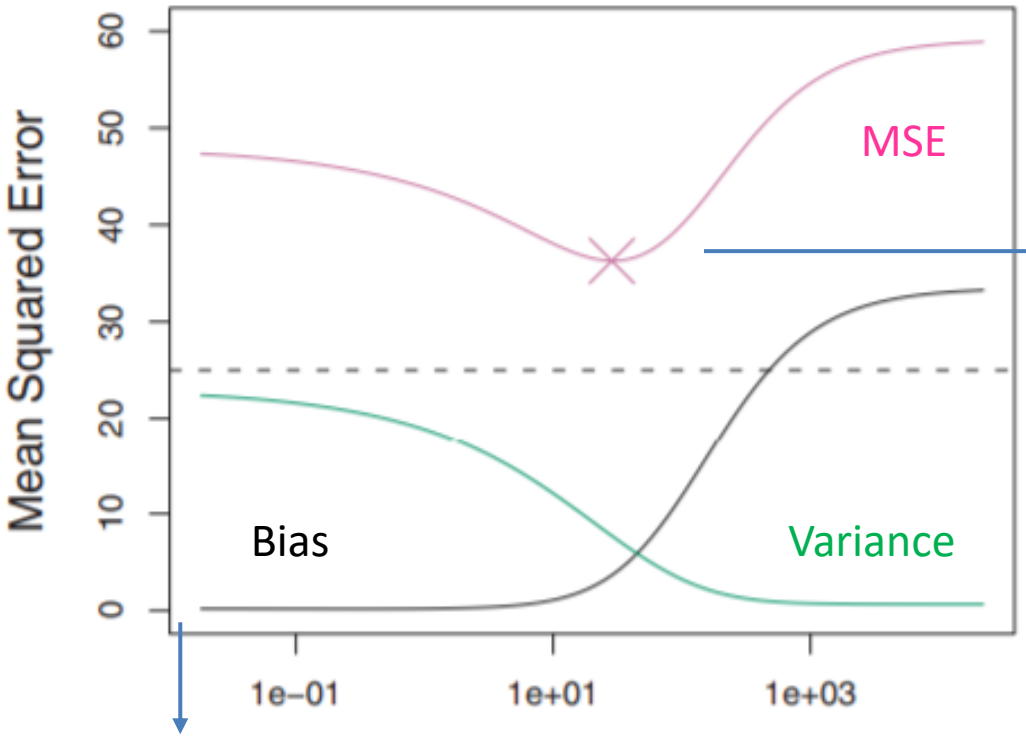
    - If λ = 0, we train linear regression
    - If λ is large, the coefficients will shrink close to 0

Bias-Variance Tradeoff



Optimal
Ridge regression

Bias-Variance Tradeoff

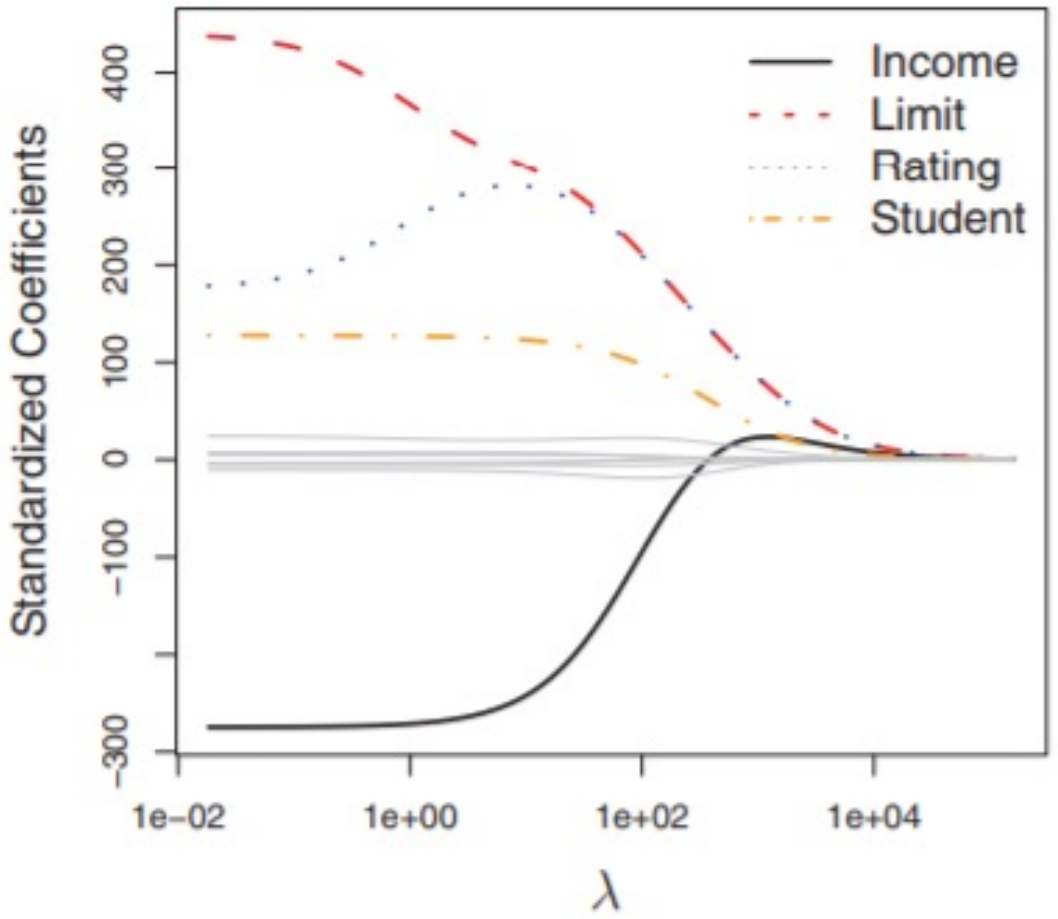


Optimal
Ridge regression

Linear
regression

Reduced model
complexity

Coefficient shrinkage



Predict credit card balance

GD for Ridge Regression

Min Loss

$$J(\theta) = \sum_{i=1}^{N} [h_\theta(x_i) - y_i]^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

GD for Ridge Regression

Min MSE

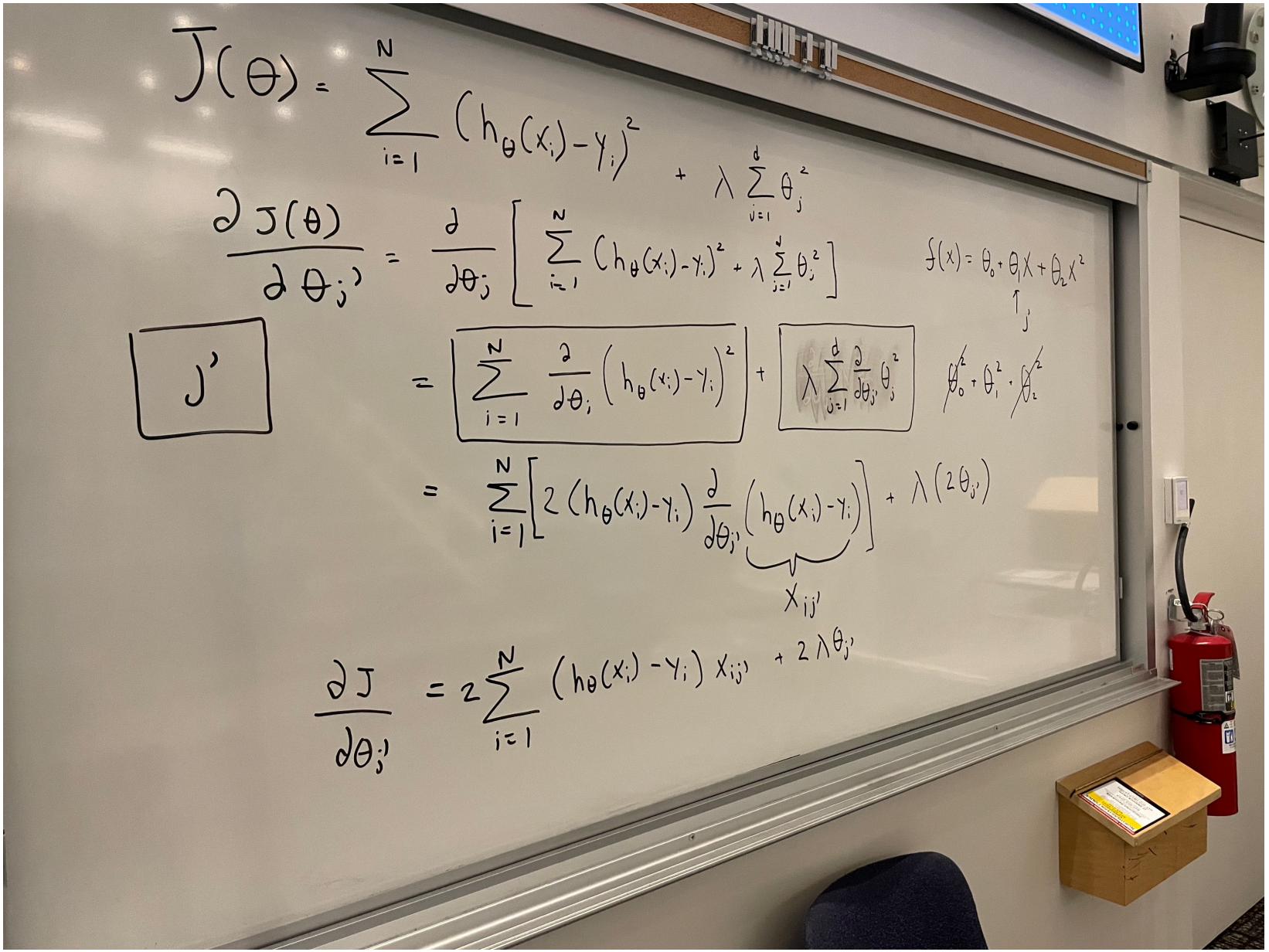
$$J(\theta) = \sum_{i=1}^{N} [h_\theta(x_i) - y_i]^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

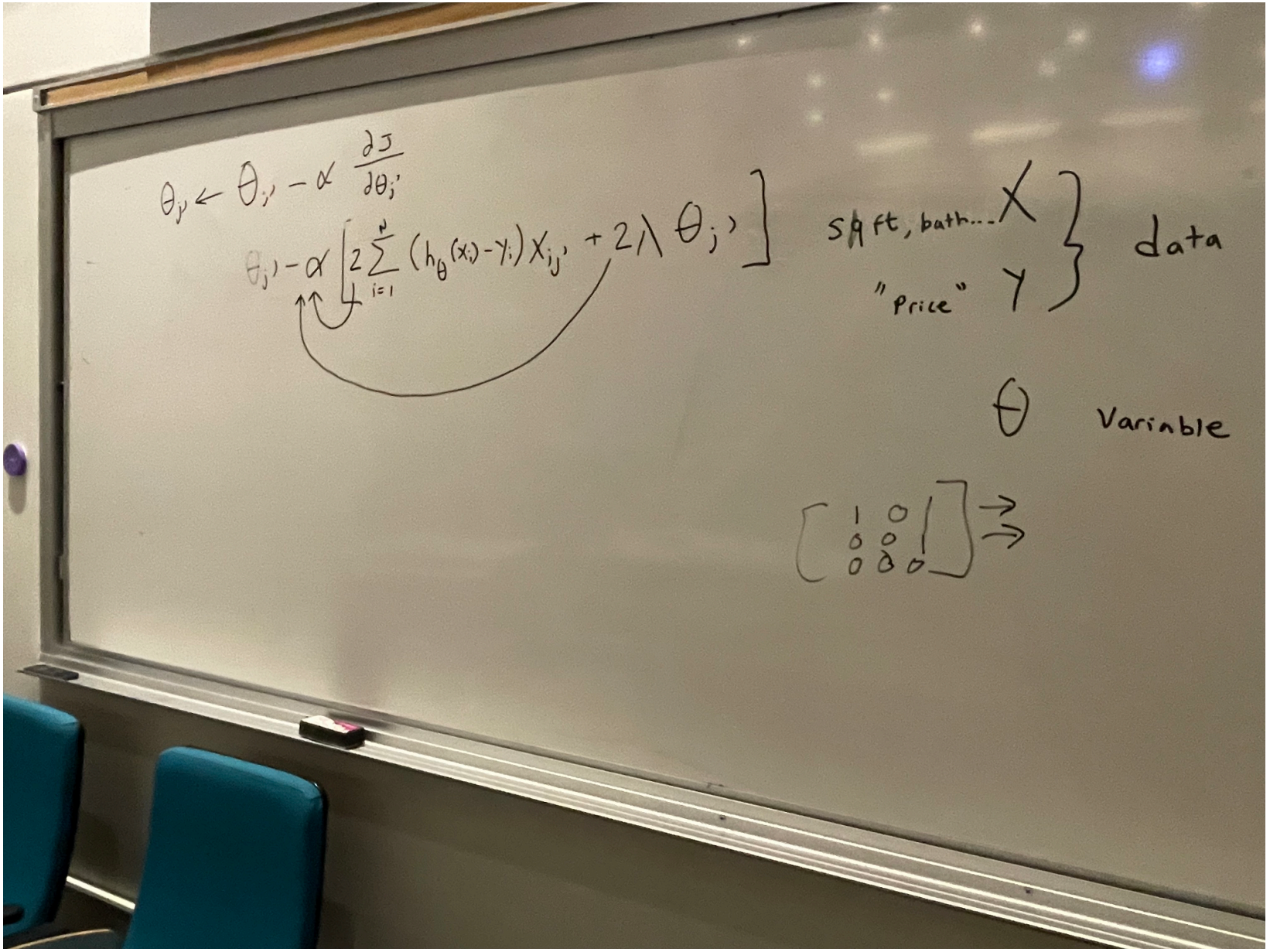
Gradient update: $\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i)$

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij} - \alpha\lambda\theta_j$$

Regularization

$$\theta_j \leftarrow \theta_j(1 - \alpha\lambda) - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$


$J(\theta) = \sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$
$\frac{\partial J(\theta)}{\partial \theta_{j'}} = \frac{\partial}{\partial \theta_{j'}} \left[\sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{d} \theta_j^2 \right]$
$f(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
j'
J'
$= \sum_{i=1}^{N} \frac{\partial}{\partial \theta_{j'}} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{d} \frac{\partial}{\partial \theta_{j'}} \theta_j^2$
$\theta_0^2 + \theta_1^2 + \theta_2^2$
$= \sum_{i=1}^{N} \left[2 (h_\theta(x_i) - y_i) \frac{\partial}{\partial \theta_{j'}} (h_\theta(x_i) - y_i) \right] + \lambda (2\theta_{j'})$
$x_{ij'}$
$\frac{\partial J}{\partial \theta_{j'}} = 2 \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij'} + 2\lambda\theta_{j'}$


$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J}{\partial \theta_j}$
$\theta_j - \alpha \left[2 \sum_{i=1}^{n} (h_\theta(x_i) - y_i) x_{ij} + 2\lambda \theta_j \right]$
sq ft, bath... X
"Price" Y
data
θ Variable

Lasso Regression

$$J(\theta) = \sum_{i=1}^{N} [h_\theta(x_i) - y_i]^2 + \lambda \sum_{j=1}^{d} |\theta_j|$$

- L1 norm for regularization
- Results in sparse coefficients
- Small issue: gradients cannot be computed around 0
  - Can use sub-gradient at 0

Lasso Regression

$$J(\theta) = \underbrace{\sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2}_{\text{Squared Residuals}} + \underbrace{\lambda \sum_{j=1}^{d} |\theta_j|}_{\text{Regularization}}$$

- L1 norm for regularization
- Results in sparse coefficients
- Issue: gradients cannot be computed around 0
- Method of sub-gradient optimization

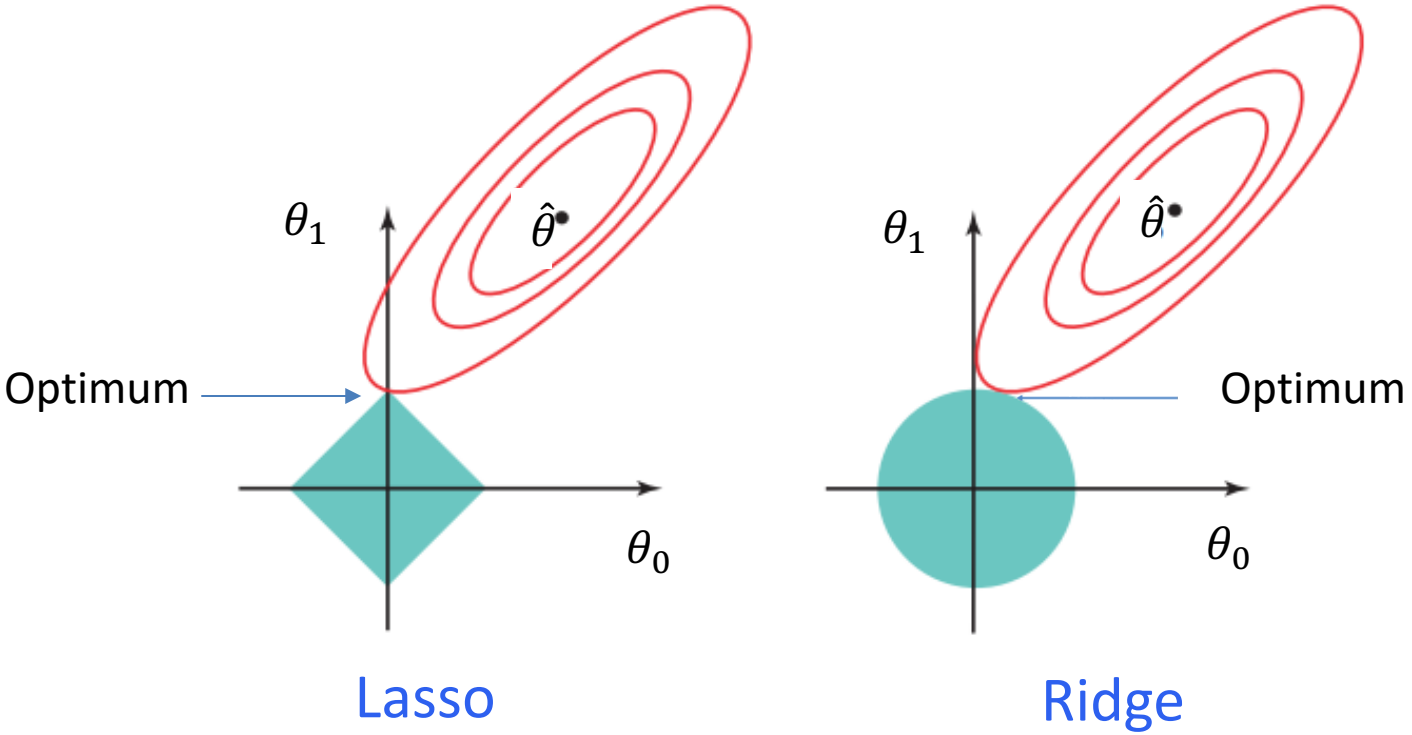
Alternative Formulations

- Ridge
  - L2 Regularization
  - $\min_{\theta} \sum_{i=1}^{N} [h_\theta(x_i) - y_i]^2$ subject to $\sum_{j=1}^{d} |\theta_j|^2 \leq \epsilon$

- Lasso
  - L1 regularization
  - $\min_{\theta} \sum_{i=1}^{N} [h_\theta(x_i) - y_i]^2$ subject to $\sum_{j=1}^{d} |\theta_j| \leq \epsilon$

Lasso vs Ridge

- Ridge shrinks all coefficients
- Lasso sets some coefficients at 0 (sparse solution)
  - Perform feature selection



Ridge vs Lasso

- Both methods can be applied to any loss function (regression or classification)

• Ridge	• Lasso

Ridge vs Lasso

- Both methods can be applied to any loss function (regression or classification)
- In both methods, value of regularization parameter λ needs to be adjusted
- Both reduce model complexity

Ridge	Lasso
+ Differentiable objective	- Gradient descent needs to be adapted
+ Gradient descent converges to global optimum	+ Results in sparse model
- Shrinks all coefficients	+ Can be used for feature selection in large dimensions