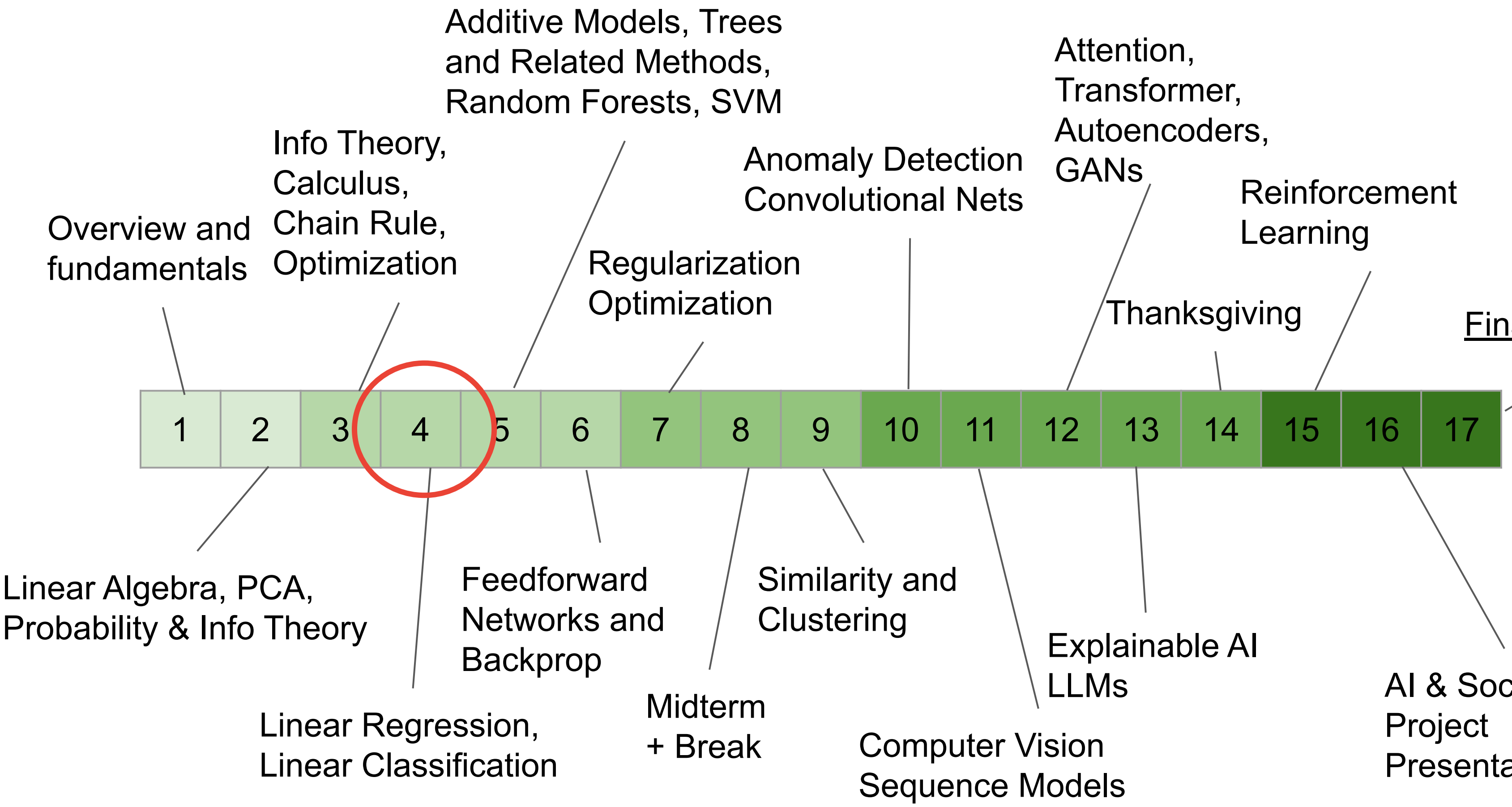
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

CS 4364/6364

Machine Learning

Fall Semester 9/12/2023
Lecture 6.
Linear Regression

John Sipple
jsipple@gwu.edu


Overview and fundamentals
1
Linear Algebra, PCA, Probability & Info Theory
2
Info Theory, Calculus, Chain Rule, Optimization
3
Linear Regression, Linear Classification
4
Additive Models, Trees and Related Methods, Random Forests, SVM
5
Feedforward Networks and Backprop
6
Regularization Optimization
7
Midterm + Break
8
Similarity and Clustering
9
Anomaly Detection Convolutional Nets
10
Computer Vision Sequence Models
11
Attention, Transformer, Autoencoders, GANs
12
Explainable AI LLMs
13
Thanksgiving
14
Reinforcement Learning
15
AI & Society Project Presentations
16
17
Final Exam

Regression Problem

Input matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$ with N examples and p dimensions

$$\mathbf{X}^\intercal = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\}$$

with input labels $\mathbf{y} \in \mathbb{R}^N$

Each example is denoted as a pair $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)$

Given a new point $\boldsymbol{x}$, we would like to predict y:

$$\hat{y} = f(\boldsymbol{x})$$

Example Data Set for Regression

$$\overbrace{\hphantom{\text{MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude Longitude}}}^{\mathbf{X}}$$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$					$\mathbf{x}_p$	$\mathbf{y}$	
	MedInc	**HouseAge**	**AveRooms**	**AveBedrms**	**Population**	**AveOccup**	**Latitude**	**Longitude**	**MedHouseVal**	
$\boldsymbol{x}_1$	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526	y_1
$\boldsymbol{x}_2$	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585	y_2
	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521	
	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413	
$\boldsymbol{x}_N$	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422	y_N

$x_{3,2}$

Types of input features

Quantitative inputs

Transformation of an input via log, square root, square

Polynomial (i.e. basis expansion)

Numeric coding of qualitative values

Feature crosses or interactions

Hypothesis

We can predict or approximate $\mathbf{y}$ with a linear function:

$$\hat{\mathbf{y}} = f(X) = \beta_0\mathbf{x}_0 + \beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 + \cdots + \beta_p\mathbf{x}_p = \beta_0 + \sum_{j=1}^{p} \mathbf{x}_j\beta_j$$

where $\mathbf{x}_0 = \mathbf{1}_p$

And redefine $\mathbf{X}$ slightly:

$$\mathbf{X}^\intercal = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\} \in \mathbb{R}^{N\times(p+1)}$$

Given vector $\hat{\boldsymbol{\beta}}$ we can make predictions $\hat{\mathbf{y}}$

$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

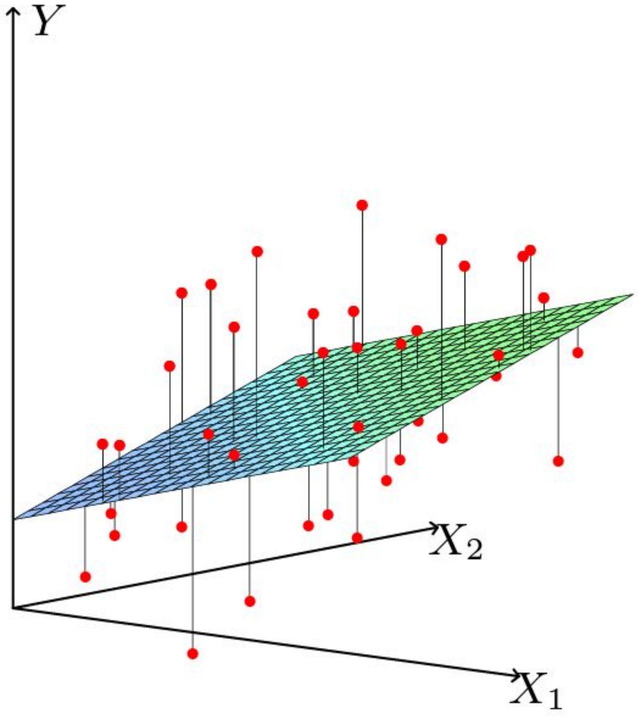


FIGURE 3.1. *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.*

Linear Regression with Least Squares

Optimization

Find the best w that minimizes the Euclidean, L_2, loss with respect to $\boldsymbol{\beta}$

$$L(\boldsymbol{\beta}) = N||\mathbf{y} - f(\mathbf{X})||_2^2 = \sum_{i=1}^{N}(y_i - f(x_i))^2 = \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\beta_j)^2$$

Least Squares Optimization

Rewrite in matrix form and simplify the equation

$$
\begin{aligned}
L(\boldsymbol{\beta}) &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\
&= \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} - (\mathbf{X}\boldsymbol{\beta})^\top\mathbf{y} + (\mathbf{X}\boldsymbol{\beta})^\top\mathbf{X}\boldsymbol{\beta} \\
&= \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} - ((\mathbf{X}\boldsymbol{\beta})^\top\mathbf{y})^\top + \boldsymbol{\beta}^\top\mathbf{X}\mathbf{X}^\top\boldsymbol{\beta} \\
&= \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} - \mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\mathbf{X}\mathbf{X}^\top\boldsymbol{\beta} \\
&= \mathbf{y}^\top\mathbf{y} - 2\mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\mathbf{X}\mathbf{X}^\top\boldsymbol{\beta}
\end{aligned}
$$

Least Squares Optimization

Solve for the Derivative:

$$\nabla_{\beta} L(\boldsymbol{\beta}) = -2\boldsymbol{X}^{\top}\mathbf{y} - 2\mathbf{X}^{\top}\mathbf{X}\beta = -2\mathbf{X}^{\top}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Set the derivative to 0:

$$\mathbf{X}^{\top}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

Least Squares Optimization

Solve for $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\intercal}\boldsymbol{X})^{-1}\mathbf{X}^{\intercal}\mathbf{y}$$

Place $\boldsymbol{\beta}$ in the original equation, and predict $\mathbf{y}$:

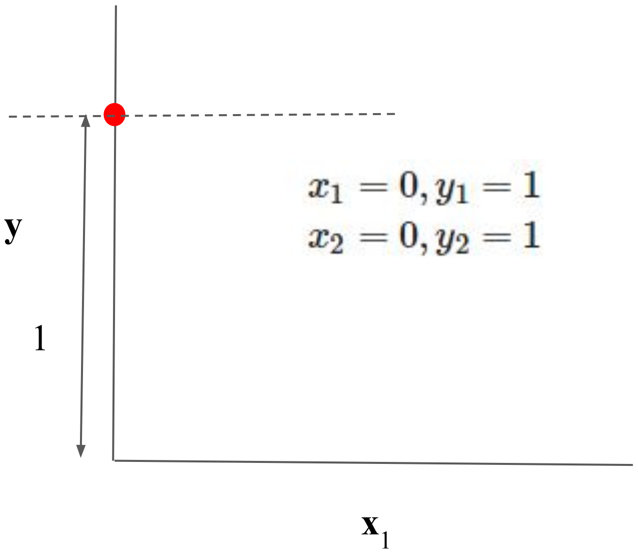
$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^{\intercal}\boldsymbol{X})^{-1}\mathbf{X}^{\intercal}\mathbf{y}$$

Regularization

Least squares solution is not stable

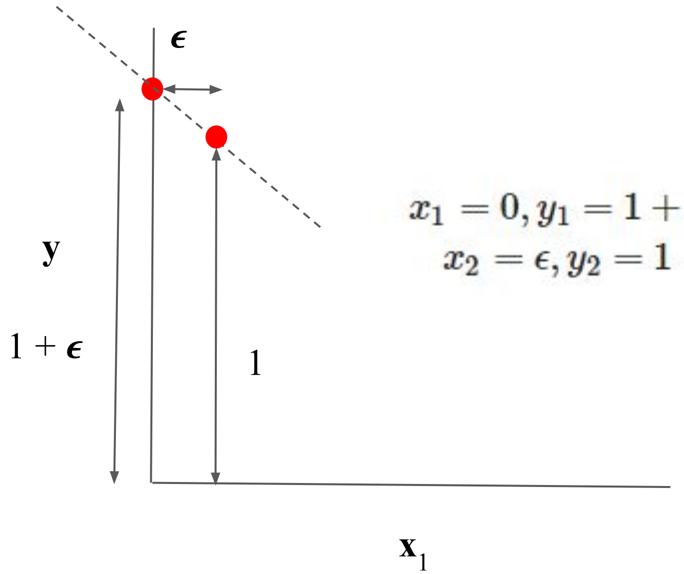
- Small perturbation in the input leads to a dramatic change in the output
- Form of overfitting

Two Simple, Contrived Examples



$$x_1 = 0, y_1 = 1$$
$$x_2 = 0, y_2 = 1$$

$$\beta = (1, 0)$$



$$x_1 = 0, y_1 = 1 + \epsilon$$
$$x_2 = \epsilon, y_2 = 1$$

$$\beta = (1 + \epsilon, -1)$$

Linear Regression with Regularization

Ridge Regression (L2 Regularization)

Fluctuation of values tend to cause instability, so favor smaller values for β

Add a penalty term λ to large β coefficients:

$$\hat{\beta}^{ridge} = \arg\min_{\beta}\left\{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\}$$

Ridge Regression (L2 Regularization)

Include the penalty term in our L2 Loss function:

$$L(\boldsymbol{\beta}, \lambda) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\intercal(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\intercal\boldsymbol{\beta}$$

Derive the gradient and set to 0:

$$\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta}, \lambda) = 0$$

Then solve for updated coefficients:

$$\hat{\boldsymbol{\beta}}^{ridge} = (\mathbf{X}^\intercal \boldsymbol{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\intercal\mathbf{y}$$

Lasso Regression (L1 Regularization)

Lasso Regression: Replace L2 Penalty in Ridge with the L1 Penalty

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Adds a nonlinearity, and there is no closed-form solution

- Solved via Quadratic Programming or Coordinate Descent

Unlike Ridge, Lasso sets small coefficients to 0

Ridge vs. Lasso Regression

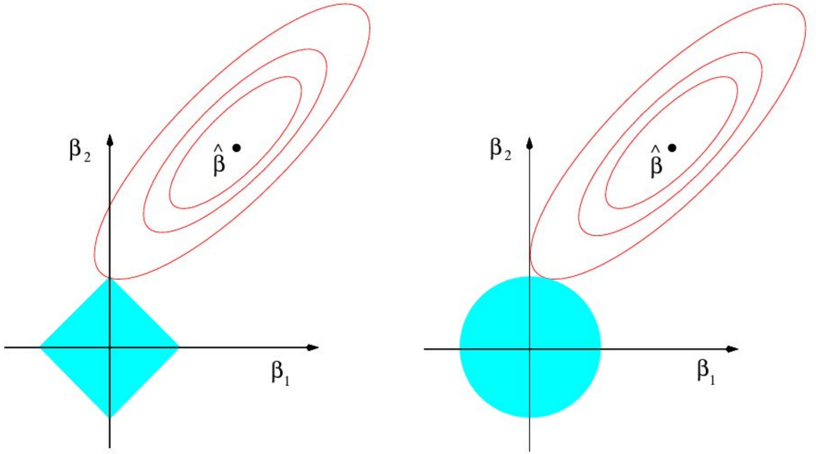


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Bias and Variance Tradeoff

Model Complexity and Bias and Variance

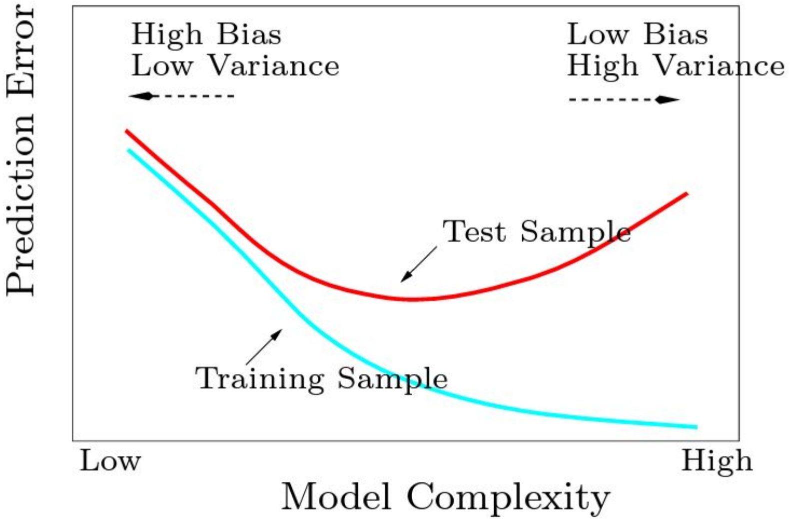


FIGURE 2.11. *Test and training error as a function of model complexity.*

$$EPE = \text{irreducible error} + \text{bias} + \text{variance}$$