



THE GEORGE  
WASHINGTON  
UNIVERSITY

WASHINGTON, DC

# CS 4364/6364

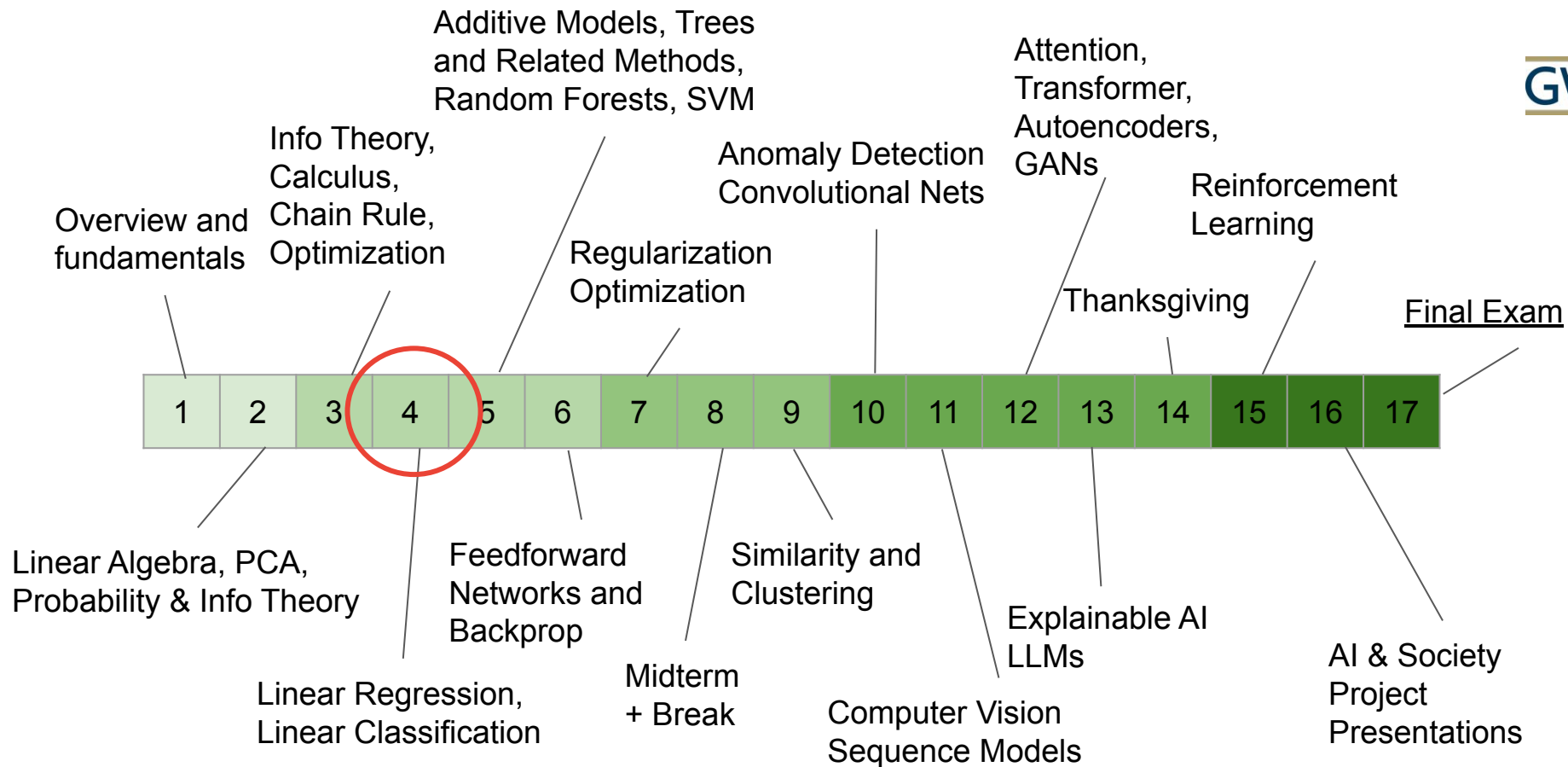
# Machine Learning

Fall Semester 9/14/2023

Lecture 7.

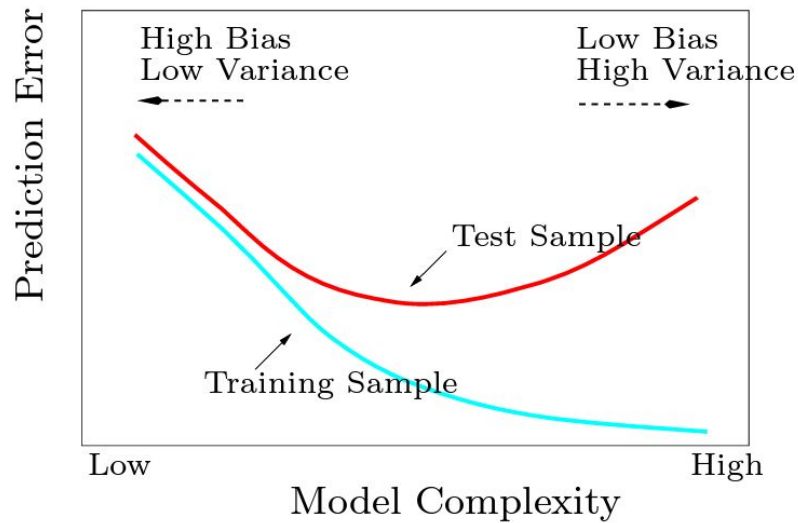
Linear Classification

John Sipple  
jsipple@gwu.edu



# Bias and Variance Tradeoff

# Model Complexity and Bias and Variance



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

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

# Linear Classification

# Classification Problem

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

$$\mathbf{X}^T = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$$

with binary input labels for  $K$  classes

$$\mathbf{Y} \in \{0, 1\}^{N \times K}$$

where

$$y_{i,k} = \begin{cases} 1 & \mathbf{x}_i \in \text{class } k \\ 0 & \text{otherwise} \end{cases}$$

Each example is denoted as a pair  $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)$

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

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

# Ecoli Dataset for Classification

$\mathbf{X}$								$\mathbf{Y}$					
$\mathbf{x}_1$	$\mathbf{x}_2$	$\dots$			$\mathbf{x}_p$			$\mathbf{y}_1$	$\mathbf{y}_2$	$\dots$		$\mathbf{x}_K$	
mcg	gvh	lip	chg	aac	alm2	alm2		cp	im	pp	imU	om	
$\mathbf{x}_1$	0.12	0.43	0.48	0.5	0.63	0.7	0.74	0	1	0	0	0	$\mathbf{y}_1$
	0.31	0.5	0.48	0.5	0.57	0.84	0.85	0	1	0	0	0	
	0.57	0.54	0.48	0.5	0.37	0.28	0.33	1	0	0	0	0	
	0.34	0.51	0.48	0.5	0.44	0.37	0.46	1	0	0	0	0	
	0.63	0.5	0.48	0.5	0.59	0.85	0.86	0	1	0	0	0	$\dots$
$\dots$	0.32	0.42	0.48	0.5	0.35	0.28	0.38	1	0	0	0	0	
	0.17	0.52	0.48	0.5	0.49	0.37	0.46	1	0	0	0	0	
	0.38	0.3	0.48	0.5	0.43	0.29	0.39	1	0	0	0	0	
	0.43	0.37	0.48	0.5	0.53	0.35	0.44	1	0	0	0	0	
	0.76	0.41	0.48	0.5	0.5	0.59	0.62	0	0	0	1	0	
	0.61	0.66	0.48	0.5	0.46	0.87	0.88	0	1	0	0	0	
$\mathbf{x}_N$	0.34	0.49	0.48	0.5	0.58	0.85	0.8	0	1	0	0	0	$\mathbf{y}_N$

$x_{4,3}$  points to the value 0.48 in the 4th row, 3rd column.  
 $y_{5,2}$  points to the value 1 in the 5th row, 10th column.

# Multiclass vs. Multilabel Classification

## Multiclass Classification:

- Exclusive Membership (**Reptile, Mammal, Fish**, etc.)

$$\text{if } y_{i,k} = 1, \text{ then } \forall_{\kappa \in K, \kappa \neq k} y_{i,\kappa} = 0$$

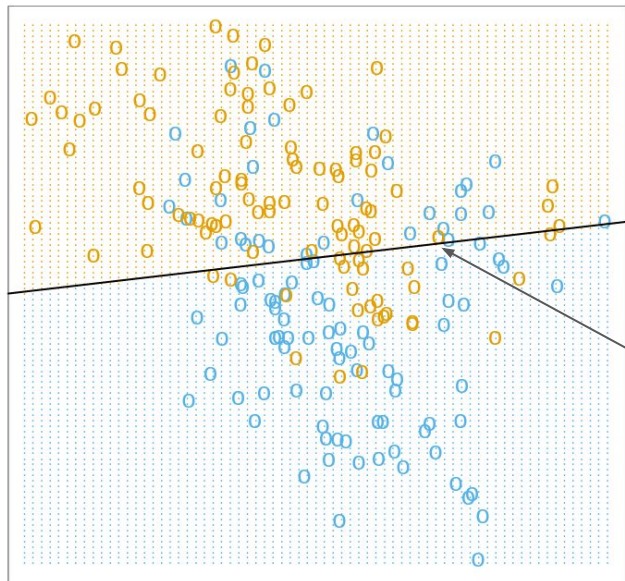
## Multilabel Classification:

- Multiple Memberships Possible: (**Feline, Tiger, Lion, Canine, Dog, Wolf**)

$$\forall_{k \in K} y_{i,k} \in \{0, 1\}$$



# Simple Linear Binary Classification in 2 Dimensions



**FIGURE 2.1.** A classification example in two dimensions. The classes are coded as a binary variable (**BLUE** = 0, **ORANGE** = 1), and then fit by linear regression. The line is the decision boundary defined by  $x^T \hat{\beta} = 0.5$ . The orange shaded region denotes that part of input space classified as **ORANGE**, while the blue region is classified as **BLUE**.

**Hypothesis:**  
Linear Decision Boundary

# Apply Linear Regression to Classification

From Linear Regression Recall (Lecture 6):

$$L(\boldsymbol{\beta}) = \sum_{i=1}^N (y_i - \sum_{j=1}^p x_{i,j} \beta_j)^2$$

In vector notation:

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

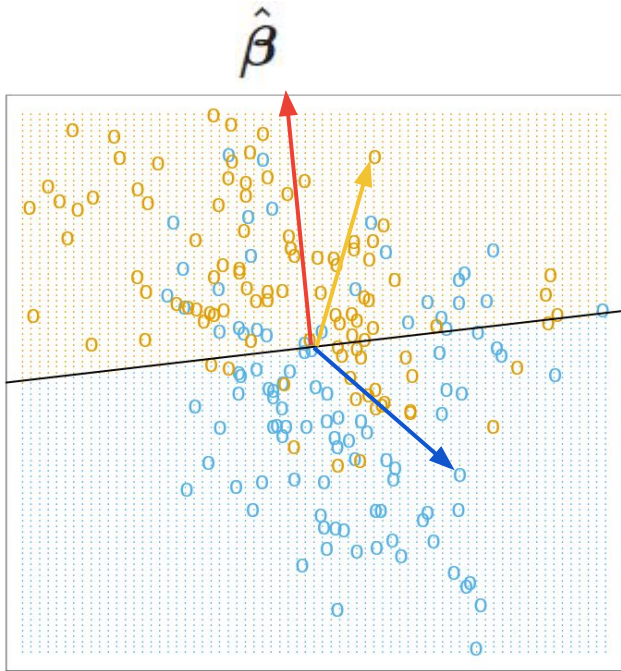
Derivative w.r.t.  $\boldsymbol{\beta}$ :

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

Then our parameters  $\hat{\boldsymbol{\beta}}$  are simply:

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

# Simple Linear Binary Classification in 2 Dimensions



For training, assign  $y_i = 1$  if **Orange**, and 0 if **Blue**

Then the Prediction Decision Boundary is

$$f(x) = \text{Orange if } \{x : x^\top \hat{\beta} > 0\}$$

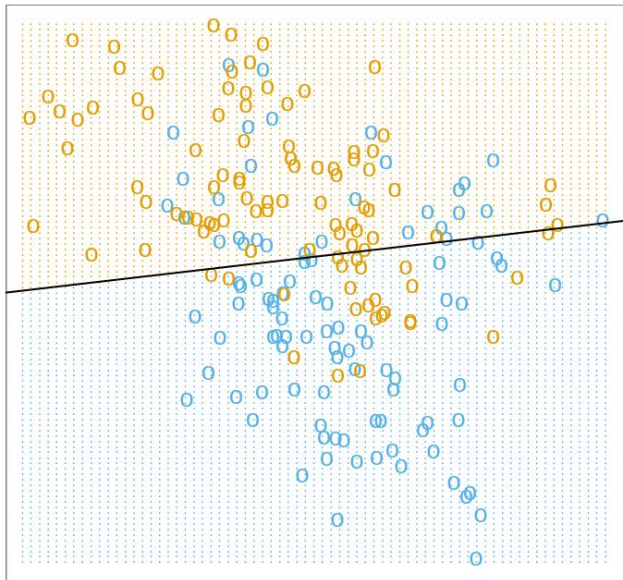
$$f(x) = \text{Blue if } \{x : x^\top \hat{\beta} \leq 0\}$$

Note this is a dot product ranging in  $(-\infty, \infty)$ :

$$x^\top \hat{\beta}$$

Where **acute** angles between  $x$  and  $\beta$  are **Orange**, and **obtuse** angles are **Blue**

# Limitations of Linear Regression for Classification



Prediction results should represent a confidence aspect as a probability:

- Nearly 1 for points well above the decision boundary
- Nearly 0 for points well below the decision boundary
- Around 0.5 for points near the decision boundary

Linear Regression is not bounded

- May yield predictions  $> 1$  and  $< 0$

# Logit Transformation

Recall the logistic sigmoid transformation from Lecture 3:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

For two classes a popular model for estimating the posterior probability:

$$\Pr(y_i = \text{Orange} | \mathbf{x} = \mathbf{x}_i) = \frac{\exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}$$

$$\Pr(y_i = \text{Blue} | \mathbf{x} = \mathbf{x}_i) = \frac{1}{1 + \exp(\boldsymbol{\beta}^\top \mathbf{x}_i)}$$

# Logit Transformation

Apply the logit transformation:  $\log \frac{p}{1-p}$ :

$$\log \frac{\Pr(y_i = \text{Orange} | \mathbf{x} = \mathbf{x}_i)}{\Pr(y_i = \text{Blue} | \mathbf{x} = \mathbf{x}_i)} = \boldsymbol{\beta}^\top \mathbf{x}_i$$

Decision boundary is at equal odds:

$$\log \frac{\Pr(y_i = \text{Orange} | \mathbf{x} = \mathbf{x}_i)}{\Pr(y_i = \text{Blue} | \mathbf{x} = \mathbf{x}_i)} = \log \frac{0.5}{0.5} = 0$$

Which is a hyperplane defined by:  $\{\mathbf{x} | \boldsymbol{\beta}^\top \mathbf{x} = 0\}$

# Logistic Regression with $K$ Classes

For multiclass classification with  $K \geq 2$ :

Our binary logistic regression formulation:

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

Now is rewritten for  $K$  classes:

$$\hat{\mathbf{B}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$$
$$\hat{\mathbf{Y}} = \mathbf{X} \hat{\mathbf{B}} = \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$$
$$\hat{\mathbf{B}} = [\beta_1, \beta_2, \dots, \beta_{K-1}]^\top$$



# Logistic Regression with $K$ Classes

$$\begin{aligned}\log \frac{\Pr(y_i = 1 | \mathbf{x} = \mathbf{x}_i)}{\Pr(y_i = K | \mathbf{x} = \mathbf{x}_i)} &= \beta_1^\top \mathbf{x}_i \\ \log \frac{\Pr(y_i = 2 | \mathbf{x} = \mathbf{x}_i)}{\Pr(y_i = K | \mathbf{x} = \mathbf{x}_i)} &= \beta_2^\top \mathbf{x}_i \\ &\dots \\ \log \frac{\Pr(y_i = K - 1 | \mathbf{x} = \mathbf{x}_i)}{\Pr(y_i = K | \mathbf{x} = \mathbf{x}_i)} &= \beta_{K-1}^\top \mathbf{x}_i\end{aligned}$$

Note that only  $K - 1$  logit transformations exist reflecting the sum of probabilities is 1.



# Logistic Regression with $K$ Classes

for  $k = 1, \dots, K - 1$  :

$$\Pr(y_i = k | \mathbf{x} = \mathbf{x}_i) = \frac{\exp(\boldsymbol{\beta}_k^\top \mathbf{x}_i)}{1 + \sum_{l=1}^{K-1} \exp(\boldsymbol{\beta}_l^\top \mathbf{x}_i)}$$

for  $k = K$  :

$$\Pr(y_i = K | \mathbf{x} = \mathbf{x}_i) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\boldsymbol{\beta}_l^\top \mathbf{x}_i)}$$

which sum to 1

# Fitting regression models

# General form Log-likelihood for $N$ Observations

$$\ell(\theta) = \sum_{i=1} N \log p_{g_i}(x_i; \theta)$$

where

$$p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$$

# Log-likelihood for two classes

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^N \{y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))\} \\ &= \sum_{i=1}^N \{y_i \beta^\top x_i - \log(1 + e^{\beta^\top x_i})\}\end{aligned}$$

where  $\beta = \{\beta_{10}, \beta_1\}$

# Computing the gradient and the Hessian

Gradient of the log-likelihood(first derivative):

$$\nabla_{\beta} \ell = \frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^N x_i (y_i - p(x_i; \beta)) = 0$$

Hessian (2nd derivative) for the Newton-Raphson optimization:

$$H_{\beta}(\ell) = \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^{\top}} = \sum_{i=1}^N x_i x_i^{\top} p(x_i; \beta)(1 - p(x_i; \beta))$$

# Newton-Raphson Optimization

Recall from Lecture 5:

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} - H_{\boldsymbol{\beta}}(\ell(\boldsymbol{\beta}^{old}))^{-1} \nabla_{\boldsymbol{\beta}}(\ell(\boldsymbol{\beta}^{old}))$$

Which is derived from a 2nd-order Taylor approximation.

$$f(\mathbf{x}) \approx f(\mathbf{x}^{(0)}) + (\mathbf{x} - \mathbf{x}^{(0)})^{\top} \nabla_{\mathbf{x}} f(\mathbf{x}^{(0)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(0)})^{\top} \mathbf{H}(f)(\mathbf{x}^{(0)}) (\mathbf{x} - \mathbf{x}^{(0)})$$
$$\mathbf{x}^{*} = \mathbf{x}^{(0)} - \mathbf{H}(f)(\mathbf{x}^{(0)})^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}^{(0)})$$

# Maximizing the log-likelihood with Newton Raphson

$$\beta^{new} = \beta^{old} - \left( \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^\top} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

where the derivatives are evaluated at  $\beta^{old}$

# Rewriting in matrix notation

$$\nabla_{\beta} \ell = \frac{\partial \ell(\beta)}{\partial \beta} = \mathbf{X}^{\top} (\mathbf{y} - \mathbf{p})$$

$$H_{\beta}(\ell) = \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^{\top}} = -\mathbf{X}^{\top} \mathbf{W} \mathbf{X}$$

where:

- $\mathbf{X}$  is the  $N \times (p + 1)$  matrix of  $x_i$
- $\mathbf{p}$  is a vector of probabilities, with elements  $p(x_i; \beta^{old})$
- $\mathbf{W}$  is an  $N \times N$  diagonal matrix with the  $i$ th element  $p(x_i; \beta^{old})(1 - p(x_i; \beta^{old}))$



# Newton-Raphson Optimization Step in Matrix Form

$$\begin{aligned}\beta^{new} &= \beta^{old} + (X^T W X)^{-1} X^T (y - p) \\ &= (X^T W X)^{-1} X^T W (X \beta^{old} + W^{-1} (y - p)) \\ &= (X^T W X)^{-1} X^T W z\end{aligned}$$

where:

- $z = (X \beta^{old} + W^{-1} (y - p))$

Solving iteratively:

$$\beta^{new} \leftarrow \arg \min (z - X \beta)^T W (z - X \beta)$$

# Readings for next lecture

Tree-Based Methods: Hastie 9.2

Bootstrap Methods: Hastie 7.11

Bagging: Hastie 8.7

Boosting Methods: 10.1

Random Forests: 15.1-15.3