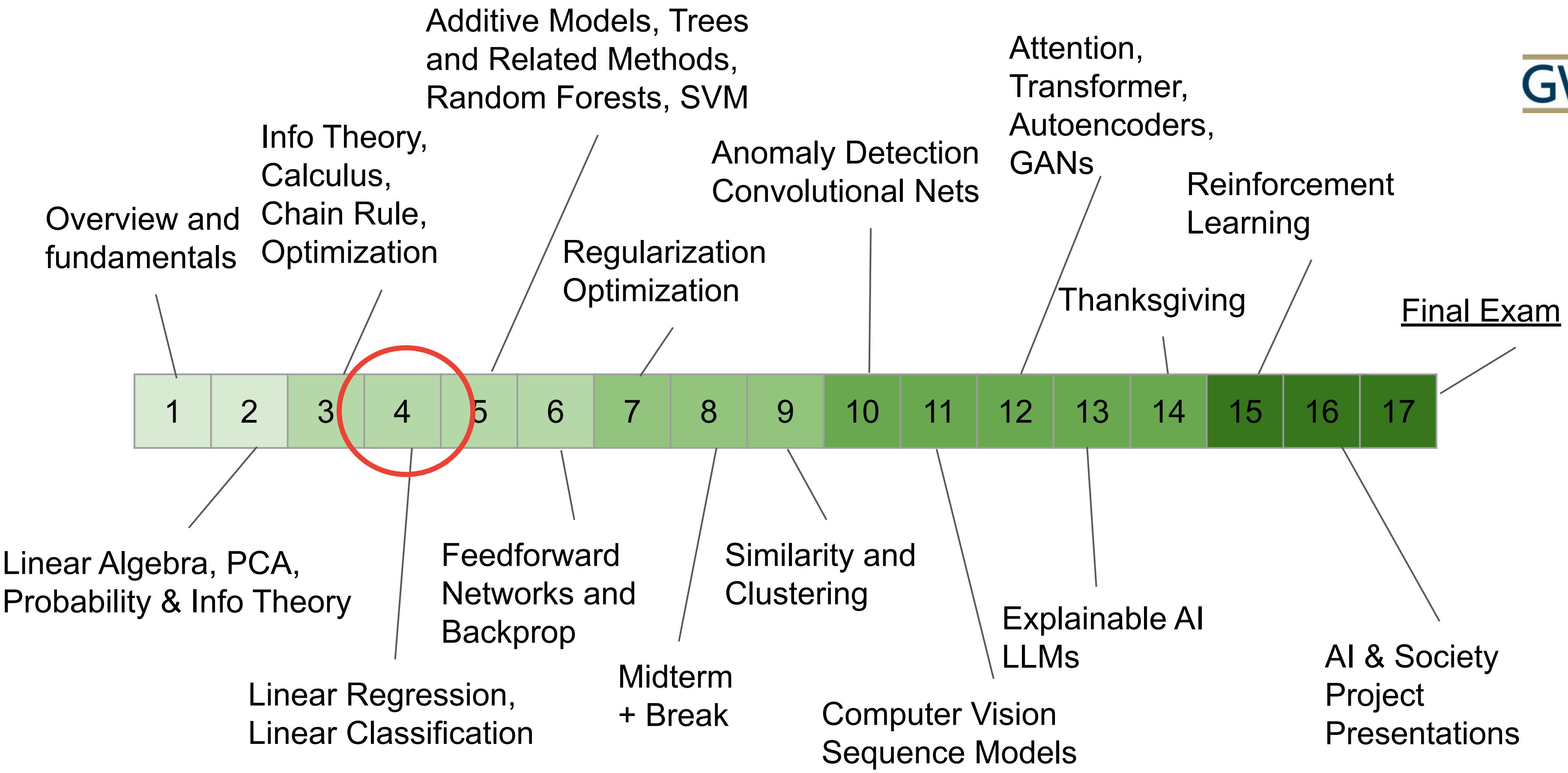
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

CS 4364/6364

Machine Learning

Fall Semester 9/14/2023
Lecture 7.
Linear Classification

John Sipple
jsipple@gwu.edu


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

Bias and Variance Tradeoff

Model Complexity and Bias and Variance

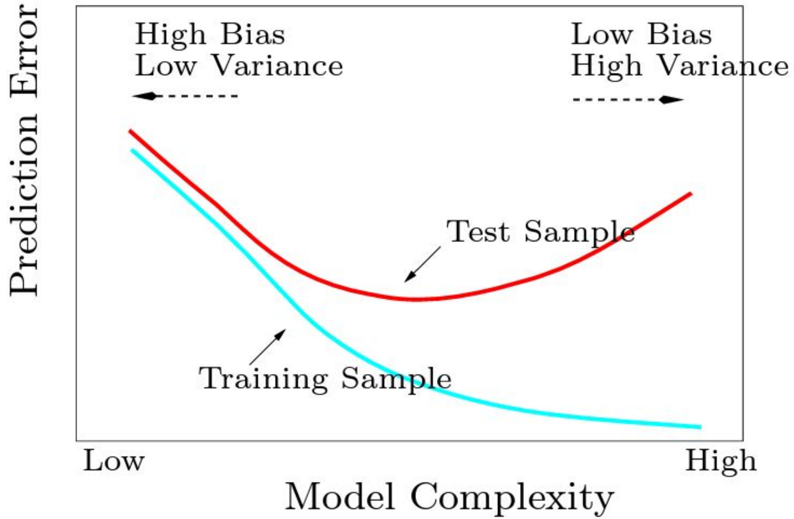


FIGURE 2.11. *Test and training error as a function of model complexity.*

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

Linear Classification

Classification Problem

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

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

with binary input labels for K classes

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

where

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

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

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

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

Ecoli Dataset for Classification

$$\overbrace{\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_p}^{\mathbf{X}} \qquad \overbrace{\mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{x}_K}^{\mathbf{Y}}$$

	mcg	gvh	lip	chg	aac	alm2	alm2	cp	im	pp	imU	om	
$\boldsymbol{x}_1$	0.12	0.43	0.48	0.5	0.63	0.7	0.74	0	1	0	0	0	$\boldsymbol{y}_1$
	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	
$\cdots$	0.63	0.5	0.48	0.5	0.59	0.85	0.86	0	1	0	0	0	$\cdots$
	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	
$\boldsymbol{x}_N$	0.61	0.66	0.48	0.5	0.46	0.87	0.88	0	1	0	0	0	$\boldsymbol{y}_N$
	0.34	0.49	0.48	0.5	0.58	0.85	0.8	0	1	0	0	0	

$x_{4,3}$ $\quad$ $y_{5,2}$

Source: https://archive.ics.uci.edu/ml/datasets/ecoli

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

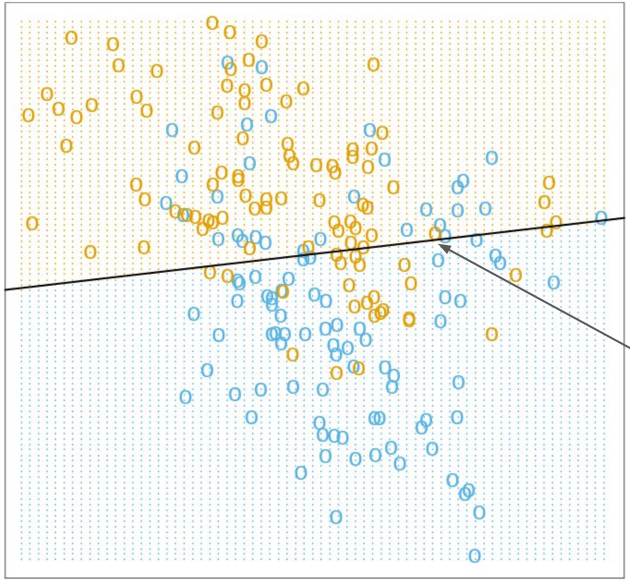


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by $x^T\hat{\beta} = 0.5$. The orange shaded region denotes that part of input space classified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

$X \in \mathbb{R}^{N \times p}$

$\beta \in \mathbb{R}^{p \times 1}$

$y \in \mathbb{R}^{N \times 1}$

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

In vector notation:

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

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

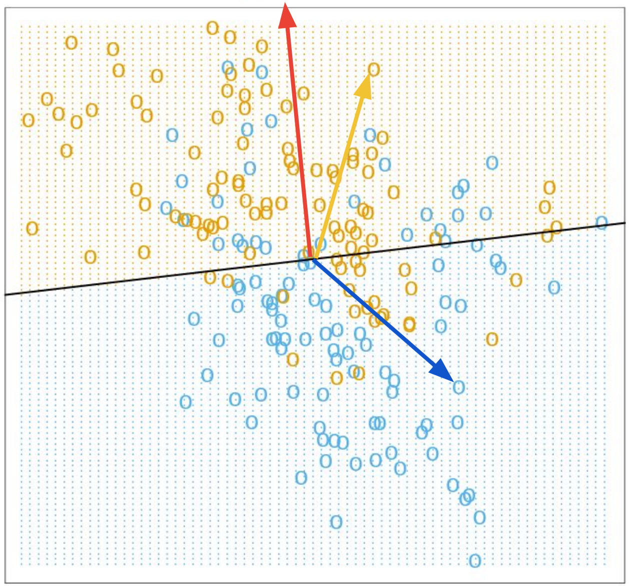
$$\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) = \mathbf{X}^{\intercal}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

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

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

Simple Linear Binary Classification in 2 Dimensions

$\hat{\beta}$



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

Then the Prediction Decision Boundary is

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

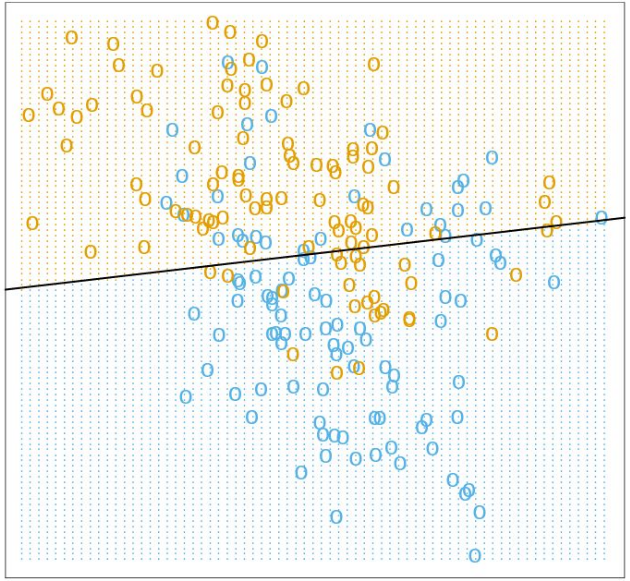
$$f(x) = \text{Blue if } \{x : x^{\top}\hat{\beta} \leq 0\}$$

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

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

Where **acute** angles between $\boldsymbol{x}$ and $\boldsymbol{\beta}$ 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 probablity:

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

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

Logit Transformation

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

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

Decision boundary is at equal odds:

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

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

Logistic Regression with K Classes

For multiclass classification with $K \geq 2$:

Our binary logistic regresion formulation:

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

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

Now is rewritten for K classes:

$$\hat{\boldsymbol{B}} = (\mathbf{X}^\intercal \boldsymbol{X})^{-1} \mathbf{X}^\intercal \mathbf{Y}$$

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{B}} = \mathbf{X}(\mathbf{X}^\intercal \boldsymbol{X})^{-1} \mathbf{X}^\intercal \mathbf{Y}$$

$$\hat{\boldsymbol{B}} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{K-1}]^\intercal$$

Logistic Regression with K Classes

$$\log \frac{\Pr(y_i = 1|\boldsymbol{x} = \boldsymbol{x}_i)}{\Pr(y_i = K|\boldsymbol{x} = \boldsymbol{x}_i)} = \boldsymbol{\beta}_1^\intercal \boldsymbol{x}_i$$

$$\log \frac{\Pr(y_i = 2|\boldsymbol{x} = \boldsymbol{x}_i)}{\Pr(y_i = K|\boldsymbol{x} = \boldsymbol{x}_i)} = \boldsymbol{\beta}_2^\intercal \boldsymbol{x}_i$$

$$\cdots$$

$$\log \frac{\Pr(y_i = K-1|\boldsymbol{x} = \boldsymbol{x}_i)}{\Pr(y_i = K|\boldsymbol{x} = \boldsymbol{x}_i)} = \boldsymbol{\beta}_{K-1}^\intercal \boldsymbol{x}_i$$

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

Logistic Regression with K Classes

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

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

for $k = K$:

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

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›

$$\ell(\beta) = \sum_{i=1}^{N} \left\{ y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta)) \right\}$$

$$= \sum_{i=1}^{N} \left\{ y_i \beta^\top x_i - \log(1 + e^{\beta^\top x_i}) \right\}$$

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 deriviative) for the Newton-Raphson optimization:

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

Newton-Raphson Optimization

Recall from Lecture 5:

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

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

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^{(0)}) + (\boldsymbol{x} - \boldsymbol{x}^{(0)})^{\intercal}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(0)}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{(0)})^{\intercal}\boldsymbol{H}(f)(\boldsymbol{x}^{(0)})(\boldsymbol{x} - \boldsymbol{x}^{(0)})$$

$$\boldsymbol{x}* = \boldsymbol{x}^{(0)} - \boldsymbol{H}(f)(\boldsymbol{x}^{(0)})^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(\mathbf{0})})$$

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 β^{old}

Rewriting in matrix notation

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

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

where:

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

Newton-Raphson Optimization Step in Matrix Form

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

where:

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

Solving iteratively:

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

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