Statistical Machine Learning: Classification With Logistic Regression

Yikun Zhang

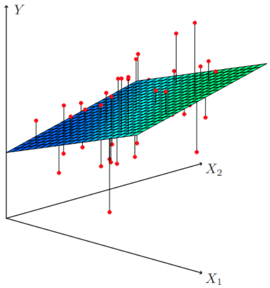
Department of Statistics,
University of Washington

School of Mathematics, University of Birmingham
October 20, 2025

Department of STATISTICS

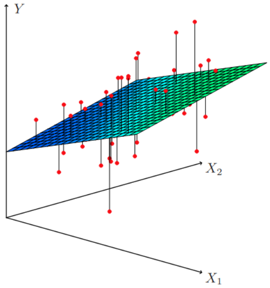
UNIVERSITY *of* WASHINGTON

Last lecture's content is based on **Chapter 3** of *"An Introduction to Statistical Learning with Applications in Python"* (Gareth et al. 2023; https://www.statlearning.com/).



1. Simple and multiple linear regression: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$.
2. Estimation: $\arg\min_{\beta_0,\ldots,\beta_p \in \mathbb{R}} \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_p X_{ip}\right)^2$.

Review on Last Lecture



3. Model assessment and variable selection:
   - F-test for $H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0$.
   - Forward and backward selection via Akaike information criteria (AIC) and Bayesian information criterion (BIC).
   - Assess the model fit by R^2 and residual standard error.
4. Dummy variable for qualitative predictors, interaction and nonlinear predictors, outliers, collinearity, etc.

Outline of Today's Lecture

1. Regression v.s. Classification
2. Drawback of Linear Regression for Classification
3. Logistic Regression
   - Modeling, Interpretation, and Estimation
   - Gradient Ascent and Iteratively Reweighted Least Squares
4. Multinomial Logistic Regression

Outline of Today's Lecture

1. Regression v.s. Classification
2. Drawback of Linear Regression for Classification
3. Logistic Regression
    - Modeling, Interpretation, and Estimation
    - Gradient Ascent and Iteratively Reweighted Least Squares
4. Multinomial Logistic Regression

- Today's lecture content is based on
  - **Chapters 4.1-4.3** of "*An Introduction to Statistical Learning with Applications in Python*" (Gareth et al. 2023; https://www.statlearning.com/);
  - **Chapter 4.4** in "*The Elements of Statistical Learning*" (Hastie et al. 2009; https://hastie.su.domains/ElemStatLearn/).

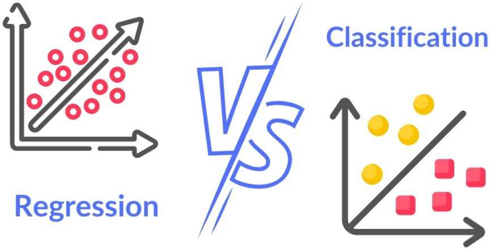
Regression v.s. Classification

Regression and classification tasks mainly fall into the *supervised* learning domain.

- Observed data: $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$ with a feature vector $\boldsymbol{X}_i = (X_{i1}, ..., X_{ip})^T \in \mathbb{R}^p$ and a response variable Y_i.

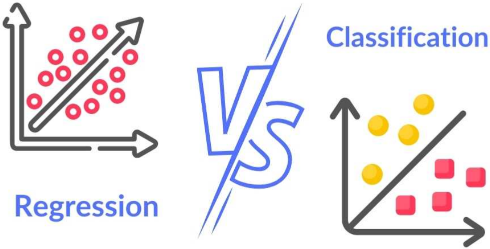
Regression and classification tasks mainly fall into the *supervised* learning domain.

- Observed data: $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$ with a feature vector $\boldsymbol{X}_i = (X_{i1}, ..., X_{ip})^T \in \mathbb{R}^p$ and a response variable Y_i.
- Regression: Y_i's are quantitative (*e.g.*, age, income, price).
- **Classification:** Y_i's are qualitative/categorical.



Regression and classification tasks mainly fall into the *supervised* learning domain.

- Observed data: $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^{n}$ with a feature vector $\boldsymbol{X}_i = (X_{i1}, ..., X_{ip})^T \in \mathbb{R}^p$ and a response variable Y_i.
- Regression: Y_i's are quantitative (*e.g.*, age, income, price).
- **Classification:** Y_i's are qualitative/categorical.



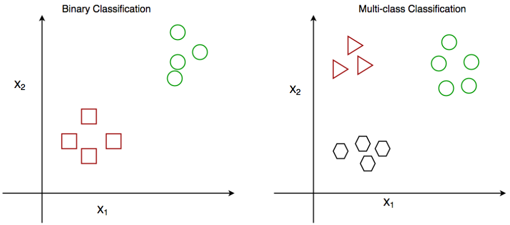
► For classification, we often encode $Y_i \in \{C_1, ..., C_K\}$ as $Y_i \in \{0, 1, ..., K-1\}$.

- eye color $\in \{\texttt{black, blue, green}\} \to \{0, 1, 2\}$.

Objective of Classification Tasks

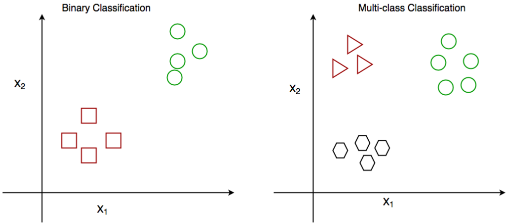
Objective of Classification Tasks

- **Prediction:** Given a feature vector $\boldsymbol{X}_{\text{new}} = \boldsymbol{x}_{\text{new}} \in \mathbb{R}^p$, predict its value for Y_{new}.



Objective of Classification Tasks

- **Prediction:** Given a feature vector $\boldsymbol{X}_{\text{new}} = \boldsymbol{x}_{\text{new}} \in \mathbb{R}^p$, predict its value for Y_{new}.



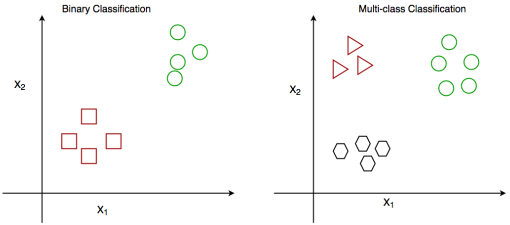
- **Interpretability:** We are more interested in predicting the probability

$$\mathbb{P}(Y_{\text{new}} | \boldsymbol{X}_{\text{new}} = \boldsymbol{x}_{\text{new}}).$$

Modeling $\mathbb{P}(Y = k | \boldsymbol{X} = \boldsymbol{x})$ for $k = 0, 1, ..., K-1$ becomes the key component of (discriminative) classification methods!

Objective of Classification Tasks

- **Prediction:** Given a feature vector $\boldsymbol{X}_{\text{new}} = \boldsymbol{x}_{\text{new}} \in \mathbb{R}^p$, predict its value for Y_{new}.



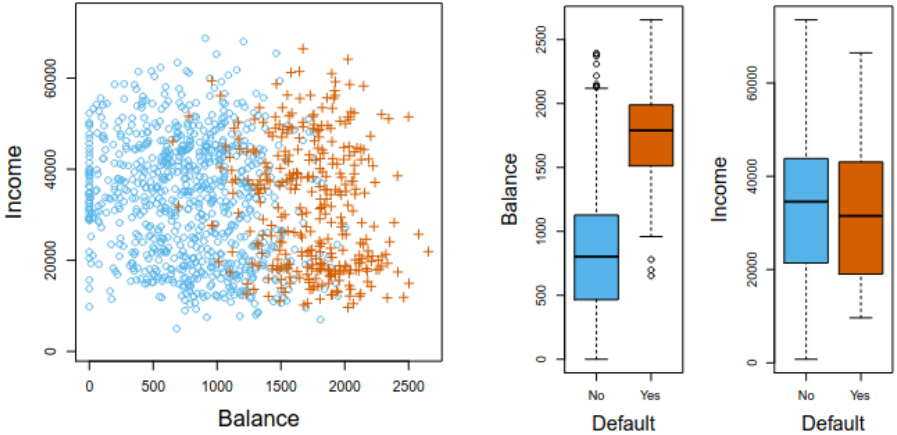
- **Interpretability:** We are more interested in predicting the probability

$$\mathbb{P}(Y_{\text{new}} | \boldsymbol{X}_{\text{new}} = \boldsymbol{x}_{\text{new}}).$$

Modeling $\mathbb{P}(Y = k|\boldsymbol{X} = \boldsymbol{x})$ for $k = 0, 1, ..., K - 1$ becomes the key component of (discriminative) classification methods!

▶ Today, we focus on the logistic regression model, which formulates $\mathbb{P}(Y = k|\boldsymbol{X} = \boldsymbol{x})$ in a generalized linear way.

Data Example: Credit Card Default



- Response $Y_i \in \{$No, Yes$\}$: whether an individual will default on his or her credit card payment.
- Features $\boldsymbol{X}_i = (X_{i1}, X_{i2}, X_{i3})$: *annual income*, *monthly credit card balance*, and *student status* (Yes/No).

Encode $Y_i \in \{\texttt{No}, \texttt{Yes}\}$ by $Y_i = \begin{cases} 0 & \text{if } \texttt{No}, \\ 1 & \text{if } \texttt{Yes}. \end{cases}$

Can We Use Linear Regression?

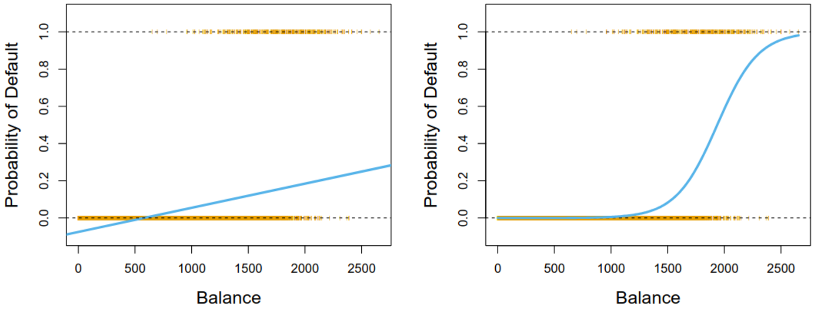
Encode $Y_i \in \{\texttt{No}, \texttt{Yes}\}$ by $Y_i = \begin{cases} 0 & \text{if } \texttt{No}, \\ 1 & \text{if } \texttt{Yes}. \end{cases}$

- $\mathbb{P}(Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}) = \mathbb{E}(Y_i | \boldsymbol{X}_i = \boldsymbol{x})$, so linear regression is mathematically valid for binary classification.
- Predict Yes if $\widehat{Y}_{\text{new}} > 0.5$, which becomes *linear discriminant analysis* in next lecture.

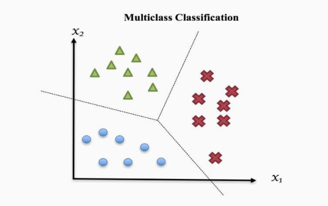
Can We Use Linear Regression?

Encode $Y_i \in \{\texttt{No}, \texttt{Yes}\}$ by $Y_i = \begin{cases} 0 & \text{if } \texttt{No}, \\ 1 & \text{if } \texttt{Yes}. \end{cases}$

- $\mathbb{P}(Y_i = 1|\boldsymbol{X}_i = \boldsymbol{x}) = \mathbb{E}(Y_i|\boldsymbol{X}_i = \boldsymbol{x})$, so linear regression is mathematically valid for binary classification.
- Predict Yes if $\widehat{Y}_{\text{new}} > 0.5$, which becomes *linear discriminant analysis* in next lecture.



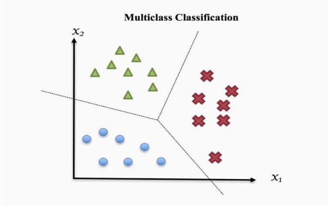
▶ **Issue I:** Linear regression might produce probabilities beyond $[0, 1]$!!



Consider a multi-class classification problem

$$Y_i = \begin{cases} 0 & \text{if } \texttt{Assistant Professor}, \\ 1 & \text{if } \texttt{Associate Professor}, \\ 2 & \text{if } \texttt{Full Professor}. \end{cases}$$

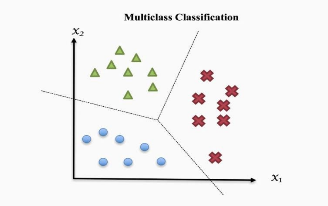
Can We Use Linear Regression?



Consider a multi-class classification problem

$$Y_i = \begin{cases} 0 & \text{if } \texttt{Assistant Professor}, \\ 1 & \text{if } \texttt{Associate Professor}, \\ 2 & \text{if } \texttt{Full Professor}. \end{cases} \qquad Y_i = \begin{cases} 0 & \text{if } \texttt{Full Professor}, \\ 1 & \text{if } \texttt{Assistant Professor}, \\ 2 & \text{if } \texttt{Associate Professor}. \end{cases}$$

Can We Use Linear Regression?



Consider a multi-class classification problem

$$Y_i = \begin{cases} 0 & \text{if } \texttt{Assistant Professor}, \\ 1 & \text{if } \texttt{Associate Professor}, \\ 2 & \text{if } \texttt{Full Professor}. \end{cases} \qquad Y_i = \begin{cases} 0 & \text{if } \texttt{Full Professor}, \\ 1 & \text{if } \texttt{Assistant Professor}, \\ 2 & \text{if } \texttt{Associate Professor}. \end{cases}$$

- Any encoding suggests an **ordering**.
- Assume the gap between class 0 and 1 is **similar** to the gap between class 1 and 2.

► **Issue II:** Different encodings of Y_i lead to fundamentally different linear models and predictions.

For a binary classification problem $Y_i \in \{0, 1\}$, a direct linear regression has its issue:

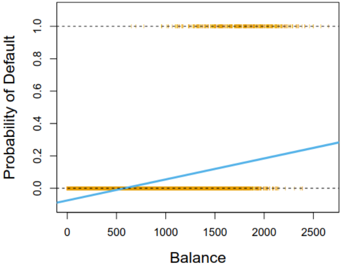
$$\mathbb{P}(Y_i = 1|\boldsymbol{X}_i) = \mathbb{E}(Y_i|\boldsymbol{X}_i) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = \boldsymbol{\beta}^T \boldsymbol{Z}_i,$$

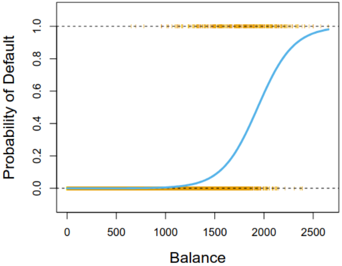
where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T \in \mathbb{R}^{p+1}$ and $\boldsymbol{Z}_i = (1, X_{i1}, ..., X_{ip})^T \in \mathbb{R}^{p+1}$.

For a binary classification problem $Y_i \in \{0,1\}$, a direct linear regression has its issue:

$$\mathbb{P}(Y_i = 1|\boldsymbol{X}_i) = \mathbb{E}(Y_i|\boldsymbol{X}_i) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = \boldsymbol{\beta}^T \boldsymbol{Z}_i,$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T \in \mathbb{R}^{p+1}$ and $\boldsymbol{Z}_i = (1, X_{i1}, ..., X_{ip})^T \in \mathbb{R}^{p+1}$.





Logistic regression assumes the form

$$\mathbb{P}(Y_i = 1|\boldsymbol{X}_i) = \frac{\exp\left(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}\right)}{1 + \exp\left(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}\right)} = \frac{\exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)},$$

where $x \mapsto \exp(x) = e^x$ is the exponential function with $\exp(1) = e \approx 2.71828$.

$$p(\boldsymbol{X}_i) := \mathbb{P}(Y_i = 1|\boldsymbol{X}_i) = \frac{\exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)} \quad \text{with} \quad \boldsymbol{\beta}, \boldsymbol{Z}_i = (1, X_{i1}, ..., X_{ip})^T \in \mathbb{R}^{p+1}.$$

Logistic Regression: Interpretation

$$p(\boldsymbol{X}_i) := \mathbb{P}(Y_i = 1|\boldsymbol{X}_i) = \frac{\exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)} \quad \text{with} \quad \boldsymbol{\beta}, \boldsymbol{Z}_i = (1, X_{i1}, ..., X_{ip})^T \in \mathbb{R}^{p+1}.$$

Some algebra implies that

$$\text{logit}(p(\boldsymbol{X}_i)) := \log\left(\frac{p(\boldsymbol{X}_i)}{1 - p(\boldsymbol{X}_i)}\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = \boldsymbol{\beta}^T \boldsymbol{Z}_i.$$

- $x \mapsto \text{logit}(p) = \log\left(\frac{p}{1-p}\right)$ is the *logit* function or the *log odds* when $p \in (0, 1)$.

Logistic Regression: Interpretation

$$p(\boldsymbol{X}_i) := \mathbb{P}(Y_i = 1|\boldsymbol{X}_i) = \frac{\exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)} \quad \text{with} \quad \boldsymbol{\beta}, \boldsymbol{Z}_i = (1, X_{i1}, ..., X_{ip})^T \in \mathbb{R}^{p+1}.$$

Some algebra implies that

$$\text{logit}(p(\boldsymbol{X}_i)) := \log\left(\frac{p(\boldsymbol{X}_i)}{1 - p(\boldsymbol{X}_i)}\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = \boldsymbol{\beta}^T \boldsymbol{Z}_i.$$

- $x \mapsto \text{logit}(p) = \log\left(\frac{p}{1-p}\right)$ is the *logit* function or the *log odds* when $p \in (0, 1)$.
- **Poisson regression** (in next lecture): When $Y_i \in \{0, 1, ...\}$ and is assumed to follow a Poisson distribution,

$$\log\left(\mathbb{E}(Y_i|\boldsymbol{X}_i)\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = \boldsymbol{\beta}^T \boldsymbol{Z}_i.$$

- **Generalized linear model:** $\eta\left(\mathbb{E}(Y_i|\boldsymbol{X}_i)\right) = \boldsymbol{\beta}^T \boldsymbol{Z}_i$ based on a pre-specified *link* function $x \mapsto \eta(x)$.

From the observed data $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \{0,1\}$, we define a **likelihood function**

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^n p(\boldsymbol{X}_i)^{Y_i} \left[1 - p(\boldsymbol{X}_i)\right]^{1-Y_i}.$$

- **Maximum likelihood estimation:** Find $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}$ to maximize $\mathcal{L}(\boldsymbol{\beta})$.

From the observed data $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \{0,1\}$, we define a **likelihood function**

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^n p(\boldsymbol{X}_i)^{Y_i} \left[1 - p(\boldsymbol{X}_i)\right]^{1-Y_i}.$$

- **Maximum likelihood estimation:** Find $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}$ to maximize $\mathcal{L}(\boldsymbol{\beta})$.
  - $\mathcal{L}(\boldsymbol{\beta})$ quantifies the probability of seeing the data under a statistical model.

From the observed data $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \{0,1\}$, we define a **likelihood function**

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^n p(\boldsymbol{X}_i)^{Y_i} \left[1 - p(\boldsymbol{X}_i)\right]^{1-Y_i}.$$

- **Maximum likelihood estimation:** Find $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}$ to maximize $\mathcal{L}(\boldsymbol{\beta})$.
  - $\mathcal{L}(\boldsymbol{\beta})$ quantifies the probability of seeing the data under a statistical model.
  - Maximizing $\mathcal{L}(\boldsymbol{\beta})$ ensures the predicted probability $\widehat{p}(\boldsymbol{X}_i)$ to be close to Y_i.
  - For logistic regression, the log-likelihood function is

$$\ell(\boldsymbol{\beta}) = \log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^n \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right]\right\}.$$

Logistic Regression: Estimation

From the observed data $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n \subset \mathbb{R}^p \times \{0,1\}$, we define a **likelihood function**

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^n p(\boldsymbol{X}_i)^{Y_i} \left[1 - p(\boldsymbol{X}_i)\right]^{1-Y_i}.$$

▶ **Maximum likelihood estimation:** Find $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}$ to maximize $\mathcal{L}(\boldsymbol{\beta})$.

- $\mathcal{L}(\boldsymbol{\beta})$ quantifies the probability of seeing the data under a statistical model.
- Maximizing $\mathcal{L}(\boldsymbol{\beta})$ ensures the predicted probability $\widehat{p}(\boldsymbol{X}_i)$ to be close to Y_i.
- For logistic regression, the log-likelihood function is

$$\ell(\boldsymbol{\beta}) = \log \mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^n \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right]\right\}.$$

▶ **Difficulty:** Unlike linear regression, there are no closed-form solutions for $\widehat{\boldsymbol{\beta}}$ when maximizing $\ell(\boldsymbol{\beta})$!

Gradient Ascent For Logistic Regression

$$\widehat{\boldsymbol{\beta}} = \mathop{\arg\max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \mathop{\arg\max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right] \right\}.$$

A common method for solving an unconstrained optimization problem is to use the *gradient ascent* iterative algorithm:

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} + \gamma \cdot \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots \tag{1}$$

Gradient Ascent For Logistic Regression

$$\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right]\right\}.$$

A common method for solving an unconstrained optimization problem is to use the *gradient ascent* iterative algorithm:

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} + \gamma \cdot \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots \tag{1}$$

- $\gamma > 0$ is the step size (or learning rate), and the gradient is given by

$$\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[Y_i - \frac{\exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)} \right] \boldsymbol{Z}_i = \sum_{i=1}^{n} \left[Y_i - p(\boldsymbol{X}_i) \right] \boldsymbol{Z}_i \in \mathbb{R}^{p+1}.$$

$$\widehat{\boldsymbol{\beta}} = \mathop{\arg\max}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \mathop{\arg\max}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right]\right\}.$$

A common method for solving an unconstrained optimization problem is to use the *gradient ascent* iterative algorithm:

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} + \gamma \cdot \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \dots \tag{1}$$

- $\gamma > 0$ is the step size (or learning rate), and the gradient is given by

$$\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[Y_i - \frac{\exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)} \right] \boldsymbol{Z}_i = \sum_{i=1}^{n} \left[Y_i - p(\boldsymbol{X}_i)\right] \boldsymbol{Z}_i \in \mathbb{R}^{p+1}.$$

- Iterate (1) until convergence, *e.g.*, $\left\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^{(t-1)}\right\|_2 < \epsilon = 10^{-8}$, and take $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$.

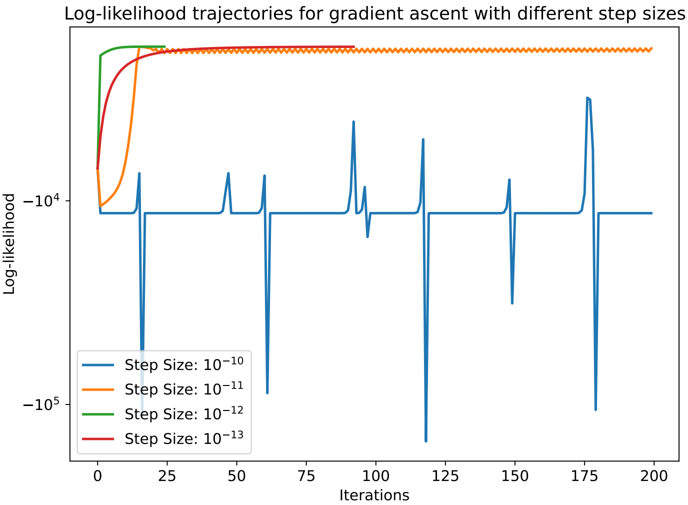
Practicality of Gradient Ascent

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} + \gamma \cdot \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots$$

- **Question:** How do we choose the step size $\gamma > 0$ in practice?

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} + \gamma \cdot \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots$$

- **Question:** How do we choose the step size $\gamma > 0$ in practice?



$$\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right]\right\}.$$

The objective function $\ell(\boldsymbol{\beta})$ is concave, and its globally optimal solution $\widehat{\boldsymbol{\beta}}$ satisfies

$$\nabla_{\boldsymbol{\beta}} \ell(\widehat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \left[Y_i - \widehat{p}(\boldsymbol{X}_i)\right] \boldsymbol{Z}_i = \mathbf{0} \quad \text{with} \quad \widehat{p}(\boldsymbol{X}_i) = \frac{\exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}.$$

Newton-Raphson Method for Logistic Regression

$$\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right] \right\}.$$

The objective function $\ell(\boldsymbol{\beta})$ is concave, and its globally optimal solution $\widehat{\boldsymbol{\beta}}$ satisfies

$$\nabla_{\boldsymbol{\beta}} \ell(\widehat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \left[Y_i - \widehat{p}(\boldsymbol{X}_i)\right] \boldsymbol{Z}_i = \mathbf{0} \quad \text{with} \quad \widehat{p}(\boldsymbol{X}_i) = \frac{\exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}.$$

To find the solution/root of $\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \mathbf{0}$, we use the *Newton-Raphson* algorithm.

Newton-Raphson Method for Logistic Regression

$$\widehat{\boldsymbol{\beta}} = \mathop{\arg\max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \mathop{\arg\max}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right] \right\}.$$

The objective function $\ell(\boldsymbol{\beta})$ is concave, and its globally optimal solution $\widehat{\boldsymbol{\beta}}$ satisfies

$$\nabla_{\boldsymbol{\beta}} \ell(\widehat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \left[Y_i - \widehat{p}(\boldsymbol{X}_i)\right] \boldsymbol{Z}_i = \boldsymbol{0} \quad \text{with} \quad \widehat{p}(\boldsymbol{X}_i) = \frac{\exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}.$$

To find the solution/root of $\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \boldsymbol{0}$, we use the *Newton-Raphson* algorithm.

- The rationale is based on Taylor's approximation:

$$\underbrace{\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})}_{\text{set to } 0} = \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) + \nabla_{\boldsymbol{\beta}}^2 \ell(\boldsymbol{\beta}^{(t-1)}) \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t-1)}\right) + \underbrace{o\left(\left\|\boldsymbol{\beta} - \boldsymbol{\beta}^{(t-1)}\right\|_2\right)}_{\text{negligible}}.$$

Newton-Raphson Method for Logistic Regression

$$\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \ell(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \left\{ Y_i \cdot \boldsymbol{\beta}^T \boldsymbol{Z}_i - \log\left[1 + \exp\left(\boldsymbol{\beta}^T \boldsymbol{Z}_i\right)\right]\right\}.$$

The objective function $\ell(\boldsymbol{\beta})$ is concave, and its globally optimal solution $\widehat{\boldsymbol{\beta}}$ satisfies

$$\nabla_{\boldsymbol{\beta}} \ell(\widehat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \left[Y_i - \widehat{p}(\boldsymbol{X}_i)\right] \boldsymbol{Z}_i = \boldsymbol{0} \quad \text{with} \quad \widehat{p}(\boldsymbol{X}_i) = \frac{\exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}{1 + \exp\left(\widehat{\boldsymbol{\beta}}^T \boldsymbol{Z}_i\right)}.$$

To find the solution/root of $\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \boldsymbol{0}$, we use the *Newton-Raphson* algorithm.

- The rationale is based on Taylor's approximation:

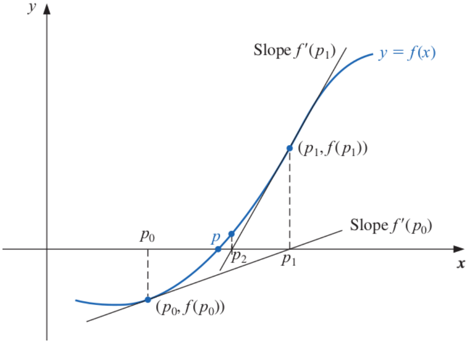
$$\underbrace{\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})}_{\text{set to } 0} = \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) + \nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}^{(t-1)}) \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(t-1)}\right) + \underbrace{o\left(\left\|\boldsymbol{\beta} - \boldsymbol{\beta}^{(t-1)}\right\|_2\right)}_{\text{negligible}}.$$

$$\Longrightarrow \boldsymbol{\beta} \approx \boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \left[\nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}^{(t-1)})\right]^{-1} \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots$$

Newton-Raphson Method for Logistic Regression

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} - \left[\nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}^{(t-1)})\right]^{-1} \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots$$

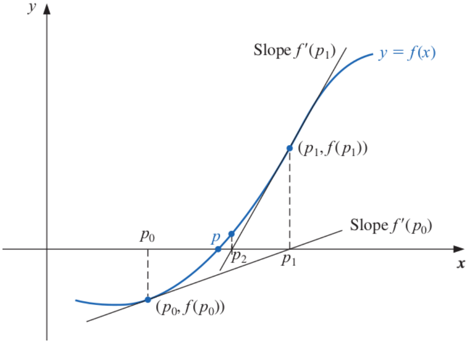
An illustration of Newton-Raphson method for solving the root of $f(p) = 0$ (Burden and Faires, 2011):



Newton-Raphson Method for Logistic Regression

$$\boldsymbol{\beta}^{(t)} \leftarrow \boldsymbol{\beta}^{(t-1)} - \left[\nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}^{(t-1)})\right]^{-1} \nabla_{\boldsymbol{\beta}} \ell\left(\boldsymbol{\beta}^{(t-1)}\right) \quad \text{for} \quad t = 1, 2, \ldots$$

An illustration of Newton-Raphson method for solving the root of $f(p) = 0$ (Burden and Faires, 2011):



Given $p(\mathbf{X}_i) = \frac{\exp(\boldsymbol{\beta}^T \mathbf{Z}_i)}{1+\exp(\boldsymbol{\beta}^T \mathbf{Z}_i)}$, we have

$$\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^n \left[Y_i - p(\mathbf{X}_i)\right] \mathbf{Z}_i \quad \text{and} \quad \nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = -\sum_{i=1}^n p(\mathbf{X}_i) \left[1 - p(\mathbf{X}_i)\right] \mathbf{Z}_i \mathbf{Z}_i^T \in \mathbb{R}^{(p+1)\times(p+1)}.$$

Iteratively Reweighted Least Squares (IRLS)

$$\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} [Y_i - p(\boldsymbol{X}_i)] \boldsymbol{Z}_i \quad \text{and} \quad \nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = -\sum_{i=1}^{n} p(\boldsymbol{X}_i) [1 - p(\boldsymbol{X}_i)] \boldsymbol{Z}_i \boldsymbol{Z}_i^T \in \mathbb{R}^{(p+1)\times(p+1)}.$$

- $\mathbb{Y} = (Y_1, ..., Y_n)^T$, $\Pi = (p(\boldsymbol{X}_1), ..., p(\boldsymbol{X}_n))^T \in \mathbb{R}^n$, and $\mathbb{Z} = (\boldsymbol{Z}_1, ..., \boldsymbol{Z}_n)^T \in \mathbb{R}^{n\times(p+1)}$;
- $\mathbb{W} = \mathtt{Diag}\left(p(\boldsymbol{X}_1)[1 - p(\boldsymbol{X}_1)], ..., p(\boldsymbol{X}_n)[1 - p(\boldsymbol{X}_n)]\right) \in \mathbb{R}^{n\times n}$.

Iteratively Reweighted Least Squares (IRLS)

$$\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} [Y_i - p(\boldsymbol{X}_i)] \boldsymbol{Z}_i \quad \text{and} \quad \nabla_{\boldsymbol{\beta}}^2 \ell(\boldsymbol{\beta}) = -\sum_{i=1}^{n} p(\boldsymbol{X}_i) [1 - p(\boldsymbol{X}_i)] \boldsymbol{Z}_i \boldsymbol{Z}_i^T \in \mathbb{R}^{(p+1)\times(p+1)}.$$

- $\mathbb{Y} = (Y_1, ..., Y_n)^T$, $\Pi = (p(\boldsymbol{X}_1), ..., p(\boldsymbol{X}_n))^T \in \mathbb{R}^n$, and $\mathbb{Z} = (\boldsymbol{Z}_1, ..., \boldsymbol{Z}_n)^T \in \mathbb{R}^{n\times(p+1)}$;
- $\mathbb{W} = \mathrm{Diag}\left(p(\boldsymbol{X}_1)[1 - p(\boldsymbol{X}_1)], ..., p(\boldsymbol{X}_n)[1 - p(\boldsymbol{X}_n)]\right) \in \mathbb{R}^{n\times n}$.

$$\Longrightarrow \nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \mathbb{Z}^T (\mathbb{Y} - \Pi) \quad \text{and} \quad \nabla_{\boldsymbol{\beta}}^2 \ell(\boldsymbol{\beta}) = -\mathbb{Z}^T \mathbb{W} \mathbb{Z}.$$

Iteratively Reweighted Least Squares (IRLS)

$$\nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} [Y_i - p(\boldsymbol{X}_i)] \boldsymbol{Z}_i \quad \text{and} \quad \nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = -\sum_{i=1}^{n} p(\boldsymbol{X}_i)[1 - p(\boldsymbol{X}_i)] \boldsymbol{Z}_i \boldsymbol{Z}_i^T \in \mathbb{R}^{(p+1)\times(p+1)}.$$

- $\mathbb{Y} = (Y_1, ..., Y_n)^T$, $\Pi = (p(\boldsymbol{X}_1), ..., p(\boldsymbol{X}_n))^T \in \mathbb{R}^n$, and $\mathbb{Z} = (\boldsymbol{Z}_1, ..., \boldsymbol{Z}_n)^T \in \mathbb{R}^{n\times(p+1)}$;
- $\mathbb{W} = \text{Diag}\left(p(\boldsymbol{X}_1)[1 - p(\boldsymbol{X}_1)], ..., p(\boldsymbol{X}_n)[1 - p(\boldsymbol{X}_n)]\right) \in \mathbb{R}^{n\times n}$.

$$\Longrightarrow \nabla_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \mathbb{Z}^T(\mathbb{Y} - \Pi) \quad \text{and} \quad \nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = -\mathbb{Z}^T\mathbb{W}\mathbb{Z}.$$

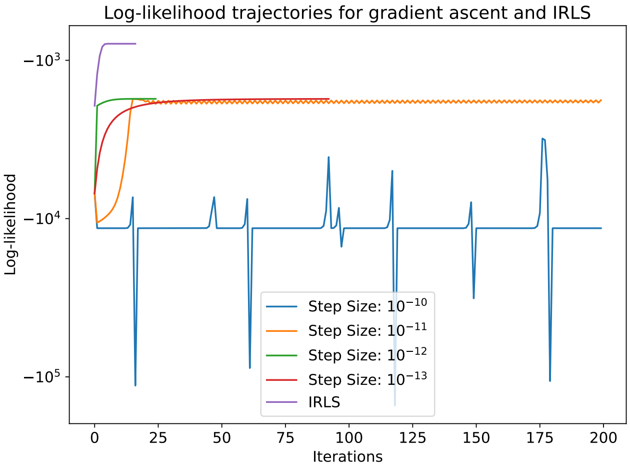
The Newton iterative step becomes

$$\begin{aligned} \boldsymbol{\beta}^{(t)} &= \boldsymbol{\beta}^{(t-1)} + \left(\mathbb{Z}^T\mathbb{W}\mathbb{Z}\right)^{-1} \mathbb{Z}^T(\mathbb{Y} - \Pi) \\ &= \left(\mathbb{Z}^T\mathbb{W}\mathbb{Z}\right)^{-1} \mathbb{Z}^T\mathbb{W} \underbrace{\left[\mathbb{Z}\boldsymbol{\beta}^{(t-1)} + \mathbb{W}^{-1}(\mathbb{Y} - \Pi)\right]}_{:= \text{"adjusted response" } \mathbb{V} \text{ depends on } t}. \end{aligned}$$

▶ This algorithm is known as the ***iterratively reweighted least squares*** **(IRLS)**:

$$\boldsymbol{\beta}^{(t)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} (\mathbb{V} - \mathbb{Z}\boldsymbol{\beta})^T \mathbb{W} (\mathbb{V} - \mathbb{Z}\boldsymbol{\beta}).$$

Comparisons Between Gradient Ascent and IRLS Algorithms



- IRLS converges in fewer iterations than gradient ascent.
- However, each IRLS iteration is more expensive due to inverting $\nabla^2_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})$, whose time complexity is $O(p^3)$!

For a multi-class classification problem with $Y_i \in \{0, 1, , ..., K-1\}$, it assumes

$$\mathbb{P}(Y_i = k|\boldsymbol{X}_i) = \frac{\texttt{exp}\left(\beta_{k0} + \beta_{k1}X_{i1} + \cdots + \beta_{kp}X_{ip}\right)}{\sum_{j=0}^{K-1} \texttt{exp}\left(\beta_{j0} + \beta_{j1}X_{i1} + \cdots + \beta_{jp}X_{ip}\right)} \quad \text{for} \quad k = 0, 1, ..., K-1.$$

- This is known as the *softmax* encoding (*i.e.*, a smooth approximation to the "arg max" function).

▶ **Interpretation:** The log odds ratio between the k-th and k'-th classes is

$$\texttt{log}\left(\frac{\mathbb{P}(Y_i = k|\boldsymbol{X}_i)}{\mathbb{P}(Y_i = k'|\boldsymbol{X}_i)}\right) = (\beta_{k0} - \beta_{k'0}) + (\beta_{k1} - \beta_{k'1})X_{i1} + \cdots + (\beta_{kp} - \beta_{k'p})X_{ip}$$

for $k, k' \in \{0, 1, ..., K-1\}$.

Assignment and Next Lecture

- **Assignment:**
  - Implement gradient ascent and IRLS algorithms for logistic regression on the “Default” dataset: https://colab.research.google.com/drive/1iO3MkZnyz9Rb4FduthSNuYHYXlD7HrNo?usp=sharing.

- **Assignment:**
  - Implement gradient ascent and IRLS algorithms for logistic regression on the "Default" dataset: https://colab.research.google.com/drive/1iO3MkZnyz9Rb4FduthSNuYHYXlD7HrNo?usp=sharing.

- **Next Lecture:**
  - Logistic regression is a **discriminative** model

$$\mathbb{P}(Y|\boldsymbol{X}=\boldsymbol{x}) = \frac{\exp\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\right)}{1+\exp\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\right)}.$$

  - **Generative** models instead model $\mathbb{P}(\boldsymbol{X}|Y=y)$ and apply Bayes' theorem for $\mathbb{P}(Y|\boldsymbol{X}=\boldsymbol{x})$, *e.g.*, linear discriminant analysis, naive Bayes, K-nearest neighbors.

Assignment and Next Lecture

- **Assignment:**
  - Implement gradient ascent and IRLS algorithms for logistic regression on the "Default" dataset: https://colab.research.google.com/drive/1iO3MkZnyz9Rb4FduthSNuYHYXlD7HrNo?usp=sharing.

- **Next Lecture:**
  - Logistic regression is a **discriminative** model

$$\mathbb{P}(Y|\boldsymbol{X}=\boldsymbol{x}) = \frac{\texttt{exp}\left(\beta_0+\beta_1 X_1+\cdots+\beta_p X_p\right)}{1+\texttt{exp}\left(\beta_0+\beta_1 X_1+\cdots+\beta_p X_p\right)}.$$

  - **Generative** models instead model $\mathbb{P}(\boldsymbol{X}|Y=y)$ and apply Bayes' theorem for $\mathbb{P}(Y|\boldsymbol{X}=\boldsymbol{x})$, *e.g.*, linear discriminant analysis, naive Bayes, K-nearest neighbors.
  - Generalized linear model, *e.g.*, Poisson regression.
  - Density estimation through classification.

Assignment and Next Lecture

▶ **Assignment:**

- Implement gradient ascent and IRLS algorithms for logistic regression on the "Default" dataset: https://colab.research.google.com/drive/1iO3MkZnyz9Rb4FduthSNuYHYXlD7HrNo?usp=sharing.

▶ **Next Lecture:**

- Logistic regression is a **discriminative** model

$$\mathbb{P}(Y|\boldsymbol{X}=\boldsymbol{x}) = \frac{\exp\left(\beta_0+\beta_1 X_1+\cdots+\beta_p X_p\right)}{1+\exp\left(\beta_0+\beta_1 X_1+\cdots+\beta_p X_p\right)}.$$

- **Generative** models instead model $\mathbb{P}(\boldsymbol{X}|Y=y)$ and apply Bayes' theorem for $\mathbb{P}(Y|\boldsymbol{X}=\boldsymbol{x})$, *e.g.*, linear discriminant analysis, naive Bayes, K-nearest neighbors.
- Generalized linear model, *e.g.*, Poisson regression.
- Density estimation through classification.

Thank you!

Reference

R. Burden and J. Faires. *Numerical Analysis*. Cengage Learning, 9th edition, 2011.

J. Gareth, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor. *An Introduction to Statistical Learning: With Applications in Python*. Springer International Publishing: Cham, Switzerland, 2023.

T. Hastie, R. Tibshirani, J. Friedman, et al. *The Elements of Statistical Learning*. Springer series in statistics New-York, 2009.