stuff on:
Linear and Logistic Regression

maths and concepts

Abstract

linear and logistic regression, two fundamental techniques in statistical learning and machine learning. develop the theory from first principles, derive key results, explore optimization methods, and discuss practical considerations. make connections between statistical and machine learning perspectives.

Contents

Part I
Linear Regression

1 Introduction and Motivation

Regression analysis is one of the most widely used statistical techniques, forming the backbone of predictive modeling in fields ranging from economics and finance to biology and engineering. The term "regression" was coined by Sir Francis Galton in the late 19th century when studying the relationship between heights of parents and their children, observing that children's heights tended to "regress" toward the population mean.

At its core, regression analysis seeks to understand and quantify the relationship between variables. Given a set of input variables (also called predictors, features, independent variables, or covariates), we wish to predict or explain a response variable (also called the outcome, target, or dependent variable). This seemingly simple goal has profound implications for scientific inference, decision-making, and prediction.

1.1 The Regression Problem

Definition 1.1 (Regression Problem)**.** Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^p$ are feature vectors and $y_i \in \mathbb{R}$ are continuous response variables, the **regression problem** seeks to find a function $f : \mathbb{R}^p \to \mathbb{R}$ such that $f(\boldsymbol{x})$ approximates y well for both observed and unobserved data.

The fundamental assumption underlying regression is that there exists some true relationship:

$$y = f(\boldsymbol{x}) + \epsilon \tag{1}$$

where $f(\boldsymbol{x}) = \mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}]$ is the **regression function** (conditional expectation of Y given $\boldsymbol{X}$), and ϵ is random noise with $\mathbb{E}[\epsilon] = 0$.

This decomposition captures a fundamental truth about real-world data: observed outcomes are determined partly by systematic, predictable factors (captured by $f(\boldsymbol{x})$) and partly by random variation (captured by ϵ). The error term ϵ represents measurement error, inherent randomness in the phenomenon, and the effects of variables not included in our model. A key insight is that no matter how sophisticated our model, some irreducible error will always remain—this is the price we pay for the complexity and randomness inherent in natural phenomena.

The goal of regression is twofold: **prediction** (estimating y for new observations) and **inference** (understanding how changes in $\boldsymbol{x}$ affect y). These goals sometimes conflict; a model optimized for prediction may be difficult to interpret, while a highly interpretable model may sacrifice predictive accuracy.

1.2 Why Linear Models?

Linear models assume that:

$$f(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j = \boldsymbol{x}^T \boldsymbol{\beta} \tag{2}$$

Despite their apparent simplicity, linear models remain remarkably useful and are often the first choice in practice. This assumption is motivated by several factors:

1. **Interpretability:** Each coefficient β_j represents the change in $\mathbb{E}[Y]$ for a unit change in x_j, holding other variables constant. This "all else equal" interpretation is precisely what scientists and decision-makers often seek. In medicine, we want to know: "If we increase the dosage by 10mg, how much will the patient's blood pressure change, on average?" Linear models provide direct answers to such questions.

2. **Taylor Approximation:** Any smooth function can be locally approximated by a linear function (first-order Taylor expansion). If the true relationship is reasonably smooth and we're operating in a limited region of the input space, a linear approximation may be quite accurate. This is the mathematical justification for why linear models often work well in practice even when the true relationship is nonlinear.

3. **Computational Tractability:** Linear models admit closed-form solutions and efficient algorithms. The ordinary least squares estimator can be computed in $O(np^2)$ time, and gradient-based methods converge reliably because the loss function is convex. This computational efficiency enables rapid model fitting, cross-validation, and uncertainty quantification.

4. **Statistical Properties:** Under certain conditions, linear estimators have optimal properties (BLUE - Best Linear Unbiased Estimator). The Gauss-Markov theorem guarantees that among all linear unbiased estimators, OLS has minimum variance. This theoretical backing provides confidence in the reliability of our estimates.

5. **Foundation for Advanced Methods:** Understanding linear regression deeply is essential because many advanced techniques—including generalized linear models, mixed effects models, and even neural networks—build upon or generalize linear regression concepts.

2 Mathematical Formulation

2.1 Model Specification

Definition 2.1 (Linear Regression Model)**.** The **linear regression model** specifies:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad i = 1, \ldots, n \tag{3}$$

or in matrix notation:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{4}$$

where:

- $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ is the response vector
- $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ is the design matrix with $\mathbf{X}_{ij} = x_{ij}$ (including intercept column of 1s)
- $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T \in \mathbb{R}^{p+1}$ is the parameter vector
- $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T \in \mathbb{R}^n$ is the error vector

The matrix notation is not merely a convenience—it reveals the deep linear algebraic structure of regression and enables us to leverage powerful tools from matrix theory. The model states that the response vector $\boldsymbol{y}$ is a linear combination of the columns of $\mathbf{X}$, plus noise. Our task is to find the coefficients $\boldsymbol{\beta}$ that best describe this linear combination.

2.2 The Design Matrix

The design matrix $\mathbf{X}$ has the structure:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{5}$$

Each row represents an observation, and each column (after the first) represents a feature. The first column of all 1s corresponds to the intercept term β_0; when we compute $\mathbf{X}\boldsymbol{\beta}$, this column ensures that β_0 is added to every prediction. This is sometimes called the "bias" term in machine learning contexts, as it shifts the entire regression surface up or down.

The design matrix encodes all the information about our predictor variables. Its properties—particularly its rank and the relationships among its columns—fundamentally determine what we can learn from the data. If two columns are identical, we cannot distinguish their individual effects. If a column is a linear combination of others, we face multicollinearity, which inflates the variance of our estimates.

2.3 Geometric Interpretation

Understanding regression geometrically provides profound insight into what least squares actually does.

Remark 2.1 (Column Space Interpretation)**.** The fitted values $\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ lie in the **column space** of $\mathbf{X}$, denoted $\mathcal{C}(\mathbf{X})$. This is a $(p+1)$-dimensional subspace of $\mathbb{R}^n$ (assuming $\mathbf{X}$ has full column rank).

The least squares solution finds the point in $\mathcal{C}(\mathbf{X})$ closest to $\boldsymbol{y}$ in Euclidean distance. Geometrically, $\hat{\boldsymbol{y}}$ is the **orthogonal projection** of $\boldsymbol{y}$ onto $\mathcal{C}(\mathbf{X})$.

Think of it this way: we live in an n-dimensional space (one dimension per observation). The response vector $\boldsymbol{y}$ is a point in this space. The column space $\mathcal{C}(\mathbf{X})$ is a $(p+1)$-dimensional hyperplane passing through the origin (or through any point if we exclude the intercept). Among all points on this hyperplane, we seek the one closest to $\boldsymbol{y}$. Basic geometry tells us this closest point is found by dropping a perpendicular from $\boldsymbol{y}$ to the hyperplane—this perpendicular is precisely the residual vector $\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$.

This geometric view immediately explains several key properties of least squares: the residuals are orthogonal to the fitted values, the residuals are orthogonal to each predictor column, and the total variation in $\boldsymbol{y}$ decomposes into explained variation (along the hyperplane) and unexplained variation (perpendicular to it).

3 Classical Assumptions

The validity of statistical inference in linear regression depends critically on certain assumptions about the data-generating process. These assumptions, known as the Gauss-Markov assumptions, determine when the ordinary least squares estimator has desirable properties. Understanding these assumptions—and knowing when they might be violated—is essential for responsible statistical practice.

3.1 The Gauss-Markov Assumptions

A1. Linearity in Parameters:

$$\mathbb{E}[Y|\mathbf{X}] = \mathbf{X}\boldsymbol{\beta} \tag{6}$$

The conditional expectation of Y given $\mathbf{X}$ is a linear function of $\boldsymbol{\beta}$.

This assumption states that the true relationship between the predictors and the expected response is linear in the parameters. Note that this does *not* require linearity in the original variables—we can include transformed variables like x^2, $\log(x)$, or $x_1 \cdot x_2$ as predictors, and the model remains linear in the parameters $\boldsymbol{\beta}$.

A2. Random Sampling / Exogeneity:

$$\mathbb{E}[\epsilon_i|\mathbf{X}] = 0 \quad \forall i \tag{7}$$

The errors have zero conditional mean given the predictors (strict exogeneity).

This is perhaps the most important assumption for causal interpretation. It requires that the predictors contain no information about the errors—that is, knowing the values of $\mathbf{X}$ tells us nothing about the expected value of ϵ. Violations occur when there are omitted variables correlated with included predictors, when there is reverse causality, or when there is measurement error in the predictors. When this assumption fails, OLS estimates are biased and inconsistent.

A3. No Perfect Multicollinearity:

$$\text{rank}(\mathbf{X}) = p + 1 < n \tag{8}$$

The design matrix has full column rank (no exact linear relationships among predictors).

This is a technical requirement ensuring that $\mathbf{X}^T\mathbf{X}$ is invertible so that the OLS solution exists and is unique. Perfect multicollinearity occurs when one predictor is an exact linear combination of others (e.g., including both temperature in Celsius and Fahrenheit). In practice, we rarely have perfect multicollinearity, but near-multicollinearity (highly correlated predictors) can cause numerical instability and inflated standard errors.

A4. Homoscedasticity:

$$\text{Var}(\epsilon_i|\mathbf{X}) = \sigma^2 \quad \forall i \tag{9}$$

The conditional variance of errors is constant across all observations.

Homoscedasticity means "same scatter"—the spread of errors around the regression line is the same regardless of the predictor values. Violations (heteroscedasticity) are common in practice: for example, income variance typically increases with education level, and prediction errors for stock returns often increase during volatile market periods. Heteroscedasticity doesn't bias the coefficient estimates, but it invalidates the usual standard errors and confidence intervals.

A5. No Autocorrelation:

$$\text{Cov}(\epsilon_i, \epsilon_j|\mathbf{X}) = 0 \quad \forall i \neq j \tag{10}$$

Errors are uncorrelated with each other.

This assumption is particularly relevant for time series and spatial data, where nearby observations often have correlated errors. For cross-sectional data with independent sampling, this assumption typically holds. Autocorrelation, like heteroscedasticity, doesn't bias coefficient estimates but does invalidate standard errors.

Remark 3.1. Assumptions A4 and A5 can be combined as:

$$\text{Var}(\boldsymbol{\epsilon}|\mathbf{X}) = \sigma^2\mathbf{I}_n \tag{11}$$

This is called **spherical errors**.

3.2 Additional Assumption for Inference

For hypothesis testing and confidence intervals, we often add:

A6. Normality:

$$\boldsymbol{\epsilon}|\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_n) \tag{12}$$

Errors are normally distributed.

Under A1-A6, we have the **Classical Normal Linear Regression Model**:

$$\boldsymbol{y}|\mathbf{X} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n) \tag{13}$$

4 Ordinary Least Squares (OLS) Estimation

4.1 The Least Squares Criterion

The method of least squares, developed by Carl Friedrich Gauss and Adrien-Marie Legendre in the early 19th century, remains the most common approach to fitting linear regression models. The intuition is straightforward: we want to find the line (or hyperplane) that makes the smallest total "error" in predicting our response values.

But why squared errors? Several compelling reasons justify this choice:

- **Mathematical tractability:** Squared errors yield a smooth, differentiable objective function with a unique global minimum (assuming $\mathbf{X}$ has full rank).
- **Statistical optimality:** Under normally distributed errors, minimizing squared errors is equivalent to maximum likelihood estimation.
- **Geometric interpretation:** Minimizing squared errors corresponds to finding the orthogonal projection onto the column space of $\mathbf{X}$.
- **Penalizing large errors:** Squaring amplifies the penalty for large deviations, encouraging the model to avoid egregious mispredictions.

Definition 4.1 (Residual Sum of Squares)**.** The **Residual Sum of Squares (RSS)** or **Sum of Squared Errors (SSE)** is:

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) \tag{14}$$

The RSS is a quadratic function of $\boldsymbol{\beta}$—it forms a paraboloid in parameter space. This convexity guarantees that any local minimum is the global minimum, and that gradient-based optimization methods will converge to the optimal solution.

The OLS estimator minimizes the RSS:

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} \text{RSS}(\boldsymbol{\beta}) \tag{15}$$

4.2 Derivation of the Normal Equations

The derivation of the OLS estimator is a beautiful application of multivariate calculus. We differentiate the RSS with respect to the parameter vector and set the result equal to zero.

Theorem 4.1 (OLS Estimator)**.** If $\mathbf{X}^T\mathbf{X}$ is invertible, the OLS estimator is:

$$\boxed{\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}} \tag{16}$$

Proof. Expand the RSS:

$$\begin{aligned}
\text{RSS}(\boldsymbol{\beta}) &= (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) && (17)\\
&= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} && (18)\\
&= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} && (19)
\end{aligned}$$

Taking the gradient with respect to $\boldsymbol{\beta}$:

$$\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^T\boldsymbol{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \tag{20}$$

Setting equal to zero (first-order condition):

$$\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\boldsymbol{y} \tag{21}$$

These are the **normal equations**. If $\mathbf{X}^T\mathbf{X}$ is invertible:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} \tag{22}$$

The Hessian is $\frac{\partial^2 \text{RSS}}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^T} = 2\mathbf{X}^T\mathbf{X}$, which is positive semi-definite, confirming this is a minimum. □

The matrix $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is called the **Moore-Penrose pseudoinverse** of $\mathbf{X}$ (when $\mathbf{X}$ has full column rank). It provides the "best" way to invert a non-square matrix in the least squares sense.

The normal equations $\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\boldsymbol{y}$ have a beautiful interpretation: they state that the residual vector $\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ must be orthogonal to every column of $\mathbf{X}$. This orthogonality condition is both necessary and sufficient for optimality.

4.3 The Hat Matrix and Projection

The hat matrix provides deep geometric insight into the mechanics of least squares regression.

Definition 4.2 (Hat Matrix)**.** The **hat matrix** (or projection matrix) is:

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{23}$$

It "puts the hat on $\boldsymbol{y}$": $\hat{\boldsymbol{y}} = \mathbf{H}\boldsymbol{y}$.

The hat matrix linearly transforms the observed responses into fitted values. Each fitted value $\hat{y}_i$ is a weighted average of all observed responses, with weights determined by the row $\mathbf{H}_{i\cdot}$. The diagonal element h_{ii} measures how much observation i influences its own fitted value—this is the **leverage** of observation i.

Proposition 4.2 (Properties of the Hat Matrix)**.** The hat matrix $\mathbf{H}$ satisfies:

1. **Symmetric:** $\mathbf{H}^T = \mathbf{H}$
2. **Idempotent:** $\mathbf{H}^2 = \mathbf{H}$
3. **Trace:** $\text{tr}(\mathbf{H}) = p + 1$ (number of parameters)
4. **Eigenvalues:** All eigenvalues are 0 or 1
5. **Range:** $\mathcal{R}(\mathbf{H}) = \mathcal{C}(\mathbf{X})$

The idempotency property ($\mathbf{H}^2 = \mathbf{H}$) confirms that $\mathbf{H}$ is a projection matrix. Applying the projection twice gives the same result as applying it once—once you're on the subspace, you stay there. This is the defining characteristic of projection operators.

Proof of idempotency.

$$\mathbf{H}^2 = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \cdot \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{24}$$

$$= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\underbrace{(\mathbf{X}^T\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1}}_{\mathbf{I}}\mathbf{X}^T \tag{25}$$

$$= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \mathbf{H} \tag{26}$$

□

4.4 Residuals and the Residual Maker Matrix

Just as the hat matrix creates fitted values, the residual maker matrix creates residuals.

Definition 4.3 (Residuals)**.** The vector of residuals is:

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}} = (\mathbf{I} - \mathbf{H})\boldsymbol{y} = \mathbf{M}\boldsymbol{y} \tag{27}$$

where $\mathbf{M} = \mathbf{I} - \mathbf{H}$ is the **residual maker matrix** (or annihilator matrix).

The residuals represent the portion of the response that our model fails to explain. They contain valuable information for diagnosing model problems: patterns in residuals may reveal non-linearity, heteroscedasticity, or outliers.

Proposition 4.3 (Properties of the Residual Maker Matrix)**.**

1. $\mathbf{M}$ is symmetric and idempotent
2. $\mathbf{MX} = \mathbf{0}$ (residuals are orthogonal to predictors)
3. $\text{tr}(\mathbf{M}) = n - p - 1$ (degrees of freedom for residuals)
4. $\mathbf{HM} = \mathbf{MH} = \mathbf{0}$

The property $\mathbf{MX} = \mathbf{0}$ is crucial: it states that residuals are orthogonal to the column space of $\mathbf{X}$. This is the geometric manifestation of the normal equations. The trace of $\mathbf{M}$ equals $n - p - 1$, which represents the degrees of freedom for error—the amount of "information" left over after estimating $p + 1$ parameters from n observations.

Remark 4.1 (Orthogonality Conditions)**.** The OLS estimator satisfies:

$$\mathbf{X}^T\hat{\boldsymbol{\epsilon}} = \mathbf{X}^T(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0} \tag{28}$$

This means:

- $\sum_{i=1}^{n} \hat{\epsilon}_i = 0$ (residuals sum to zero, from the intercept column)
- $\sum_{i=1}^{n} x_{ij}\hat{\epsilon}_i = 0$ for each predictor j (residuals uncorrelated with predictors)

5 Statistical Properties of OLS Estimators

5.1 Unbiasedness

Theorem 5.1 (Unbiasedness of OLS)**.** Under assumptions A1-A3, the OLS estimator is unbiased:

$$\mathbb{E}[\hat{\boldsymbol{\beta}}|\mathbf{X}] = \boldsymbol{\beta} \tag{29}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\hat{\boldsymbol{\beta}}|\mathbf{X}] &= \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}|\mathbf{X}] && (30)\\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbb{E}[\boldsymbol{y}|\mathbf{X}] && (31)\\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \quad (\text{since } \mathbb{E}[\boldsymbol{y}|\mathbf{X}] = \mathbf{X}\boldsymbol{\beta}) && (32)\\
&= \boldsymbol{\beta} && (33)
\end{aligned}$$

□

5.2 Variance-Covariance Matrix

Knowing that the OLS estimator is unbiased tells us that on average, across many samples, we'll get the right answer. But how much will our estimates vary from sample to sample? The variance-covariance matrix quantifies this uncertainty.

Theorem 5.2 (Variance of OLS Estimator)**.** Under assumptions A1-A5:

$$\text{Var}(\hat{\boldsymbol{\beta}}|\mathbf{X}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \tag{34}$$

This result is profound. The variance of our estimates depends on two factors: the noise level σ^2 (more noise means more uncertainty) and the structure of our predictors through $(\mathbf{X}^T\mathbf{X})^{-1}$. The matrix $\mathbf{X}^T\mathbf{X}$ is sometimes called the **information matrix** because it captures how much information the predictors provide about the parameters.

Proof.

$$\begin{aligned}
\text{Var}(\hat{\boldsymbol{\beta}}|\mathbf{X}) &= \text{Var}((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}|\mathbf{X}) && (35)\\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\text{Var}(\boldsymbol{y}|\mathbf{X})\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} && (36)\\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\sigma^2\mathbf{I})\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} && (37)\\
&= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} && (38)\\
&= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} && (39)
\end{aligned}$$

□

The diagonal elements of $\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$ give the variances of individual coefficient estimates, while the off-diagonal elements give the covariances. High correlation between predictors inflates these variances—this is the mathematical manifestation of multicollinearity.

5.3 The Gauss-Markov Theorem

The Gauss-Markov theorem is one of the crown jewels of statistical theory. It provides a powerful justification for using OLS: among a large class of estimators, OLS is optimal.

Theorem 5.3 (Gauss-Markov)**.** Under assumptions A1-A5, the OLS estimator $\hat{\boldsymbol{\beta}}$ is the **Best Linear Unbiased Estimator (BLUE)**. That is, among all linear unbiased estimators of $\boldsymbol{\beta}$, OLS has the smallest variance.

Let's unpack this statement carefully:

- **Linear:** The estimator is a linear function of the response $\boldsymbol{y}$.
- **Unbiased:** The expected value of the estimator equals the true parameter.
- **Best:** Among all estimators satisfying the above two properties, OLS has the smallest variance (in the matrix sense—no other linear unbiased estimator has smaller variance for any linear combination of parameters).

The theorem does *not* say that OLS is the best estimator overall. Biased estimators (like ridge regression) can sometimes achieve lower mean squared error by trading a small amount of bias for a large reduction in variance. And nonlinear estimators are not covered by the theorem at all.

Proof. Let $\tilde{\boldsymbol{\beta}} = \mathbf{C}\boldsymbol{y}$ be any other linear unbiased estimator. Write $\mathbf{C} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T + \mathbf{D}$ for some matrix $\mathbf{D}$.

For unbiasedness:

$$\mathbb{E}[\tilde{\boldsymbol{\beta}}|\mathbf{X}] = \mathbf{C}\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta} \implies \mathbf{C}\mathbf{X} = \mathbf{I} \tag{40}$$

This requires $\mathbf{D}\mathbf{X} = \mathbf{0}$.

The variance of $\tilde{\boldsymbol{\beta}}$:

$$\begin{aligned}
\text{Var}(\tilde{\boldsymbol{\beta}}|\mathbf{X}) &= \sigma^2\mathbf{C}\mathbf{C}^T && (41)\\
&= \sigma^2[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T + \mathbf{D}][(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T + \mathbf{D}]^T && (42)\\
&= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} + \sigma^2\mathbf{D}\mathbf{D}^T && (43)
\end{aligned}$$

where cross terms vanish because $\mathbf{D}\mathbf{X} = \mathbf{0}$.

Since $\mathbf{D}\mathbf{D}^T$ is positive semi-definite:

$$\text{Var}(\tilde{\boldsymbol{\beta}}|\mathbf{X}) \geq \text{Var}(\hat{\boldsymbol{\beta}}|\mathbf{X}) \tag{44}$$

with equality if and only if $\mathbf{D} = \mathbf{0}$. □

5.4 Estimation of σ^2

To conduct inference, we need to estimate the unknown error variance σ^2. We cannot simply compute the sample variance of the residuals because residuals systematically underestimate the true errors (the fitted line passes closer to the observed points than the true line does).

Theorem 5.4 (Unbiased Estimator of σ^2)**.** Under A1-A5, an unbiased estimator of σ^2 is:

$$\hat{\sigma}^2 = s^2 = \frac{\text{RSS}}{n-p-1} = \frac{\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}}{n-p-1} = \frac{\sum_{i=1}^n \hat{\epsilon}_i^2}{n-p-1} \tag{45}$$

The denominator $n - p - 1$ is the **degrees of freedom** for error. We divide by this rather than n because we've "used up" $p + 1$ degrees of freedom in estimating the coefficients. This correction ensures unbiasedness and accounts for the fact that residuals are constrained to satisfy $p + 1$ orthogonality conditions.

Proof. We need to show $\mathbb{E}[\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}|\mathbf{X}] = (n-p-1)\sigma^2$.

Note that $\hat{\boldsymbol{\epsilon}} = \mathbf{M}\boldsymbol{y} = \mathbf{M}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) = \mathbf{M}\boldsymbol{\epsilon}$ (since $\mathbf{M}\mathbf{X} = \mathbf{0}$).

$$\begin{aligned}
\mathbb{E}[\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}|\mathbf{X}] &= \mathbb{E}[\boldsymbol{\epsilon}^T\mathbf{M}^T\mathbf{M}\boldsymbol{\epsilon}|\mathbf{X}] = \mathbb{E}[\boldsymbol{\epsilon}^T\mathbf{M}\boldsymbol{\epsilon}|\mathbf{X}] && (46)\\
&= \mathbb{E}[\text{tr}(\boldsymbol{\epsilon}^T\mathbf{M}\boldsymbol{\epsilon})|\mathbf{X}] = \mathbb{E}[\text{tr}(\mathbf{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T)|\mathbf{X}] && (47)\\
&= \text{tr}(\mathbf{M}\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T|\mathbf{X}]) = \text{tr}(\mathbf{M}\sigma^2\mathbf{I}) && (48)\\
&= \sigma^2\text{tr}(\mathbf{M}) = \sigma^2(n-p-1) && (49)
\end{aligned}$$

□

6 Distribution Theory and Inference

With the groundwork of estimation in place, we now turn to inference: testing hypotheses and constructing confidence intervals. These require knowledge of the sampling distributions of our estimators, which in turn require the normality assumption.

6.1 Distribution Under Normality

Under the full Classical Normal Linear Model (A1-A6), we obtain exact (finite-sample) distributional results.

Theorem 6.1 (Distribution of OLS Estimator)**.**

$$\hat{\boldsymbol{\beta}}|\mathbf{X} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}) \tag{50}$$

This follows because $\hat{\boldsymbol{\beta}}$ is a linear transformation of the normally distributed $\boldsymbol{y}$, and linear transformations preserve normality. Each individual coefficient $\hat{\beta}_j$ is normally distributed with mean β_j and variance $\sigma^2[(\mathbf{X}^T\mathbf{X})^{-1}]_{jj}$.

Theorem 6.2 (Distribution of RSS)**.**

$$\frac{\text{RSS}}{\sigma^2} = \frac{\hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}}{\sigma^2} \sim \chi^2_{n-p-1} \tag{51}$$

and RSS is independent of $\hat{\boldsymbol{\beta}}$.

6.2 Hypothesis Testing

6.2.1 Testing Individual Coefficients

To test $H_0 : \beta_j = \beta_j^0$ versus $H_1 : \beta_j \neq \beta_j^0$:

$$t = \frac{\hat{\beta}_j - \beta_j^0}{\text{SE}(\hat{\beta}_j)} = \frac{\hat{\beta}_j - \beta_j^0}{s\sqrt{[(\mathbf{X}^T\mathbf{X})^{-1}]_{jj}}} \sim t_{n-p-1} \tag{52}$$

under H_0.

6.2.2 Testing Linear Restrictions (F-test)

Consider testing $H_0 : \mathbf{R}\boldsymbol{\beta} = \boldsymbol{r}$ where $\mathbf{R}$ is a $q \times (p+1)$ matrix of restrictions.

Theorem 6.3 (F-test)**.** Under H_0:

$$F = \frac{(\mathbf{R}\hat{\boldsymbol{\beta}} - \boldsymbol{r})^T[\mathbf{R}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{R}^T]^{-1}(\mathbf{R}\hat{\boldsymbol{\beta}} - \boldsymbol{r})/q}{s^2} \sim F_{q,n-p-1} \tag{53}$$

6.2.3 Testing Overall Significance

To test $H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$ (all slopes zero):

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)} = \frac{R^2/p}{(1-R^2)/(n-p-1)} \sim F_{p,n-p-1} \tag{54}$$

6.3 Confidence Intervals

6.3.1 Individual Coefficient

A $(1-\alpha)$ confidence interval for β_j:

$$\hat{\beta}_j \pm t_{\alpha/2,n-p-1} \cdot \text{SE}(\hat{\beta}_j) \tag{55}$$

6.3.2 Confidence Region for $\boldsymbol{\beta}$

The joint $(1-\alpha)$ confidence region:

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \mathbf{X}^T \mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq (p+1) s^2 F_{\alpha, p+1, n-p-1} \tag{56}$$

This is an ellipsoid centered at $\hat{\boldsymbol{\beta}}$.

7 Model Evaluation and Diagnostics

7.1 Decomposition of Variance

Definition 7.1 (Sum of Squares Decomposition)**.**

$$\underbrace{\sum_{i=1}^{n} (y_i - \bar{y})^2}_{\text{TSS}} = \underbrace{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}_{\text{ESS/RegSS}} + \underbrace{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}_{\text{RSS}} \tag{57}$$

where:

- TSS = Total Sum of Squares (total variation in y)
- ESS = Explained Sum of Squares (variation explained by model)
- RSS = Residual Sum of Squares (unexplained variation)

7.2 Coefficient of Determination

Definition 7.2 (R^2)**.**

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} = \frac{\text{ESS}}{\text{TSS}} = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{58}$$

Proposition 7.1 (Properties of R^2)**.**

1. $0 \leq R^2 \leq 1$
2. R^2 equals the squared correlation between y and $\hat{y}$: $R^2 = \text{Corr}(y, \hat{y})^2$
3. Adding variables never decreases R^2
4. R^2 does not penalize model complexity

7.3 Adjusted R^2

To penalize for model complexity:

$$\bar{R}^2 = 1 - \frac{\text{RSS}/(n-p-1)}{\text{TSS}/(n-1)} = 1 - (1 - R^2) \frac{n-1}{n-p-1} \tag{59}$$

7.4 Information Criteria

Definition 7.3 (AIC and BIC)**.**

$$\text{AIC} = n \ln \left(\frac{\text{RSS}}{n} \right) + 2(p+2) \tag{60}$$

$$\text{BIC} = n \ln \left(\frac{\text{RSS}}{n} \right) + (p+2) \ln(n) \tag{61}$$

Lower values indicate better models. BIC penalizes complexity more heavily for $n > 8$.

7.5 Residual Analysis

7.5.1 Standardized and Studentized Residuals

Definition 7.4.

$$\text{Standardized residual: } r_i = \frac{\hat{\epsilon}_i}{s\sqrt{1-h_{ii}}} \tag{62}$$

$$\text{Studentized residual: } t_i = \frac{\hat{\epsilon}_i}{s_{(i)}\sqrt{1-h_{ii}}} \tag{63}$$

where h_{ii} is the i-th diagonal element of $\mathbf{H}$ (leverage), and $s_{(i)}$ is the standard error estimate computed without observation i.

7.5.2 Leverage

The leverage h_{ii} measures how far $\boldsymbol{x}_i$ is from the center of the predictor space:

$$h_{ii} = \boldsymbol{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{x}_i \tag{64}$$

Properties:

- $\frac{1}{n} \le h_{ii} \le 1$
- $\sum_{i=1}^{n} h_{ii} = p+1$
- Average leverage: $\bar{h} = \frac{p+1}{n}$

High leverage points: $h_{ii} > 2\bar{h}$ or $h_{ii} > 3\bar{h}$.

7.5.3 Cook's Distance

Measures the influence of observation i on all fitted values:

$$D_i = \frac{(\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(i)})^T(\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(i)})}{(p+1)s^2} = \frac{r_i^2}{p+1} \cdot \frac{h_{ii}}{1-h_{ii}} \tag{65}$$

Rule of thumb: $D_i > 1$ or $D_i > 4/n$ suggests influential observation.

8 Optimization Methods

8.1 Gradient Descent

8.1.1 Batch Gradient Descent

For the MSE loss $J(\boldsymbol{\beta}) = \frac{1}{2n}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2$:

$$\nabla_{\boldsymbol{\beta}} J = -\frac{1}{n}\mathbf{X}^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) \tag{66}$$

Update rule:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \alpha\nabla_{\boldsymbol{\beta}} J = \boldsymbol{\beta}^{(t)} + \frac{\alpha}{n}\mathbf{X}^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}^{(t)}) \tag{67}$$

8.1.2 Stochastic Gradient Descent (SGD)

Update using a single randomly chosen observation:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \alpha(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}^{(t)})\boldsymbol{x}_i \tag{68}$$

8.1.3 Mini-batch Gradient Descent

Update using a batch B of size b:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \frac{\alpha}{b} \sum_{i \in B} (y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}^{(t)}) \boldsymbol{x}_i \tag{69}$$

8.1.4 Convergence Analysis

Theorem 8.1 (Convergence of Gradient Descent)**.** For the quadratic loss in linear regression, gradient descent converges if:

$$0 < \alpha < \frac{2}{\lambda_{\max}(\mathbf{X}^T \mathbf{X}/n)} \tag{70}$$

where $\lambda_{\max}$ is the largest eigenvalue.

The convergence rate is:

$$\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\| \leq \left(1 - \frac{\lambda_{\min}}{\lambda_{\max}}\right)^t \|\boldsymbol{\beta}^{(0)} - \boldsymbol{\beta}^*\| \tag{71}$$

8.2 Normal Equation vs. Gradient Descent

Aspect	Normal Equation	Gradient Descent
Complexity	$O(np^2 + p^3)$	$O(knp)$ per iteration
Memory	$O(p^2)$ for $\mathbf{X}^T\mathbf{X}$	$O(np)$ or $O(p)$ for SGD
Exact solution	Yes	Approximate
Works when $p > n$	No	Yes (with regularization)
Hyperparameters	None	Learning rate α
Scalability	Poor for large p	Good for large n and p

Table 1: Comparison of optimization methods

9 Regularization

When p is large or features are correlated, regularization improves estimation.

9.1 Ridge Regression (L2 Regularization)

Definition 9.1 (Ridge Regression)**.**

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\} \tag{72}$$

Theorem 9.1 (Ridge Solution)**.**

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \boldsymbol{y} \tag{73}$$

Proposition 9.2 (Properties of Ridge)**.**

1. Biased but lower variance: Bias^2 increases, Var decreases with λ
2. Always invertible (even when $p > n$)
3. Shrinks coefficients toward zero but doesn't set them exactly to zero
4. Equivalent to MAP estimation with Gaussian prior: $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I})$

9.2 Lasso Regression (L1 Regularization)

Definition 9.2 (Lasso)**.**

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\} \tag{74}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$.

Key property: Lasso produces **sparse** solutions (some $\hat{\beta}_j = 0$ exactly), performing automatic feature selection.

9.3 Elastic Net

Combines L1 and L2 penalties:

$$\hat{\boldsymbol{\beta}}_{\text{EN}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \right\} \tag{75}$$

9.4 Bias-Variance Trade-off

Theorem 9.3 (Bias-Variance Decomposition)**.** For any estimator $\hat{f}(\boldsymbol{x})$:

$$\mathbb{E}[(y - \hat{f}(\boldsymbol{x}))^2] = \underbrace{\text{Var}(\hat{f}(\boldsymbol{x}))}_{\text{Variance}} + \underbrace{[\text{Bias}(\hat{f}(\boldsymbol{x}))]^2}_{\text{Bias}^2} + \underbrace{\sigma^2}_{\text{Irreducible}} \tag{76}$$

Regularization introduces bias but reduces variance, potentially lowering total prediction error.

10 Extensions of Linear Regression

10.1 Polynomial Regression

Transform features: if x is a scalar, create $\boldsymbol{x}' = (1, x, x^2, \ldots, x^d)^T$.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d + \epsilon \tag{77}$$

This is still **linear in parameters** $\boldsymbol{\beta}$, so all linear regression theory applies.

10.2 Interaction Terms

Model interactions between predictors:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon \tag{78}$$

The effect of x_1 on y now depends on the value of x_2:

$$\frac{\partial \mathbb{E}[y]}{\partial x_1} = \beta_1 + \beta_3 x_2 \tag{79}$$

10.3 Generalized Least Squares (GLS)

When $\text{Var}(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{\Omega}$ (non-spherical errors):

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathbf{X}^T \boldsymbol{\Omega}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{\Omega}^{-1} \boldsymbol{y} \tag{80}$$

GLS is BLUE when the error covariance structure is known.

10.4 Weighted Least Squares (WLS)

Special case of GLS when $\boldsymbol{\Omega} = \text{diag}(w_1^{-1}, \ldots, w_n^{-1})$:

$$\hat{\boldsymbol{\beta}}_{\text{WLS}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} w_i (y_i - \boldsymbol{x}_i^T \boldsymbol{\beta})^2 \tag{81}$$

11 Multicollinearity: Deep Dive

11.1 Definition and Detection

Definition 11.1 (Multicollinearity)**.** Multicollinearity exists when predictor variables are highly correlated, i.e., when there exist constants $c_1, \ldots, c_p$ (not all zero) such that:

$$\sum_{j=1}^{p} c_j x_{ij} \approx 0 \quad \forall i \tag{82}$$

11.1.1 Variance Inflation Factor (VIF)

Definition 11.2 (VIF)**.** For predictor x_j:

$$\text{VIF}_j = \frac{1}{1 - R_j^2} \tag{83}$$

where R_j^2 is the R^2 from regressing x_j on all other predictors.

Proposition 11.1.

$$\text{Var}(\hat{\beta}_j) = \frac{\sigma^2}{(n-1)\text{Var}(x_j)} \cdot \text{VIF}_j \tag{84}$$

Thus VIF directly measures how much the variance of $\hat{\beta}_j$ is inflated due to multicollinearity.

11.2 Mathematical Consequences

When $\mathbf{X}^T\mathbf{X}$ is nearly singular:

- Eigenvalues span a wide range (high condition number)
- Small changes in $\boldsymbol{y}$ cause large changes in $\hat{\boldsymbol{\beta}}$
- Standard errors of coefficients are inflated

Definition 11.3 (Condition Number)**.**

$$\kappa(\mathbf{X}^T\mathbf{X}) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{85}$$

A condition number > 30 indicates serious multicollinearity.

Part II
Logistic Regression

12 Introduction to Classification

While linear regression handles continuous outcomes, many real-world problems involve categorical outcomes: Will a customer churn? Is this email spam? Does a patient have a disease? Will

a loan default? These are **classification** problems, and they require fundamentally different modeling approaches.

Classification is ubiquitous in modern applications. Medical diagnosis systems classify patients as healthy or diseased. Credit scoring models classify applicants as creditworthy or risky. Spam filters classify emails. Image recognition systems classify objects. In all these cases, we seek to predict a categorical outcome from observed features.

12.1 The Classification Problem

Definition 12.1 (Binary Classification)**.** Given data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$, find a function $f : \mathbb{R}^p \to \{0, 1\}$ that accurately predicts the class label.

The binary classification problem appears deceptively similar to regression: we have inputs and outputs, and we seek a predictive function. However, the discrete nature of the output fundamentally changes the problem. We can't simply minimize squared errors because the output space is discrete. Instead, classification typically proceeds in two stages: first estimate the probability of each class, then make a decision based on these probabilities.

This probabilistic approach has several advantages. It provides calibrated uncertainty estimates ("I'm 90% confident this is spam"), allows for different decision thresholds based on costs (missing a cancer diagnosis is worse than a false alarm), and enables principled combination of multiple information sources.

12.2 Why Not Linear Regression for Classification?

A natural first thought might be: "Why not just use linear regression with $y \in \{0, 1\}$?" This approach, sometimes called the **Linear Probability Model (LPM)**, does see some use but has fundamental problems.

If we model $y \in \{0, 1\}$ using linear regression:

$$\hat{y} = \boldsymbol{x}^T \hat{\boldsymbol{\beta}} \tag{86}$$

Problems:

1. **Predictions can be < 0 or > 1 (not valid probabilities):** For extreme values of $\boldsymbol{x}$, the linear model can predict negative probabilities or probabilities greater than one, which are nonsensical. This is not just a theoretical concern—it happens routinely in practice.

2. **Heteroscedasticity:** $\text{Var}(y|x) = p(1-p)$ depends on x. When y is binary, its conditional variance is $\pi(1 - \pi)$ where $\pi = P(y = 1|\boldsymbol{x})$. This variance is maximized when $\pi = 0.5$ and approaches zero as π approaches 0 or 1. Since π depends on $\boldsymbol{x}$, the variance is not constant—homoscedasticity is inherently violated.

3. **Non-normal errors (Bernoulli distribution):** The errors cannot be normally distributed because y can only take two values. This invalidates the standard inference procedures (t-tests, F-tests) that rely on normality.

4. **Decision boundary is not optimal:** The linear probability model doesn't optimize any sensible classification criterion. It minimizes squared error, but for classification, we typically care about maximizing likelihood or minimizing classification error.

These problems motivate the development of logistic regression, which addresses all of them by modeling probabilities directly using an appropriate functional form.

13 The Logistic Regression Model

Logistic regression is the workhorse of binary classification. It models the probability of the positive class as a function of the predictors, ensuring that predictions are always valid probabilities between 0 and 1. The model is simple enough to be interpretable yet flexible enough to capture many real-world relationships.

13.1 Model Specification

The key insight of logistic regression is to model the *probability* of the outcome rather than the outcome itself, using a function that naturally constrains outputs to the interval $(0, 1)$.

Definition 13.1 (Logistic Regression Model)**.** The probability that $y = 1$ given $\boldsymbol{x}$ is modeled as:

$$P(Y = 1|\boldsymbol{x}) = \pi(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{x}^T\boldsymbol{\beta}}} = \frac{e^{\boldsymbol{x}^T\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}^T\boldsymbol{\beta}}} \tag{87}$$

Equivalently:

$$P(Y = 0|\boldsymbol{x}) = 1 - \pi(\boldsymbol{x}) = \frac{1}{1 + e^{\boldsymbol{x}^T\boldsymbol{\beta}}} \tag{88}$$

The model combines a linear predictor $\boldsymbol{x}^T\boldsymbol{\beta}$ (just as in linear regression) with a nonlinear transformation (the sigmoid function) that maps the entire real line to the unit interval. This elegant construction preserves the interpretability of linear models while respecting the constraints of probability.

13.2 The Sigmoid (Logistic) Function

The sigmoid function is the mathematical heart of logistic regression. Understanding its properties is essential for understanding the model.

Definition 13.2 (Sigmoid Function)**.**

$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z} \tag{89}$$

The sigmoid function has a characteristic S-shaped curve that smoothly transitions from 0 to 1. It was originally studied in the context of population growth models and has since found applications throughout statistics, machine learning, and neural networks.

Proposition 13.1 (Properties of the Sigmoid)**.**

1. **Range:** $\sigma : \mathbb{R} \to (0, 1)$ — outputs are always valid probabilities
2. **Symmetry:** $\sigma(-z) = 1 - \sigma(z)$ — the function is symmetric about the point $(0, 0.5)$
3. **Derivative:** $\sigma'(z) = \sigma(z)(1-\sigma(z))$ — a remarkably simple form that facilitates optimization
4. **Limits:** $\lim_{z\to-\infty} \sigma(z) = 0$, $\lim_{z\to+\infty} \sigma(z) = 1$ — extreme inputs give extreme probabilities
5. **Inverse:** $\sigma^{-1}(p) = \ln\left(\frac{p}{1-p}\right) = \text{logit}(p)$ — the logit function inverts the sigmoid

The derivative property is particularly important. The fact that $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ means the gradient can be computed efficiently from the function value itself, without additional computation. This makes gradient-based optimization very efficient.

Proof of derivative.

$$\frac{d\sigma}{dz} = \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) = \frac{e^{-z}}{(1+e^{-z})^2} \tag{90}$$

$$= \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}} = \sigma(z) \cdot \frac{1}{1+e^{z}} \tag{91}$$

$$= \sigma(z)(1-\sigma(z)) \tag{92}$$

□

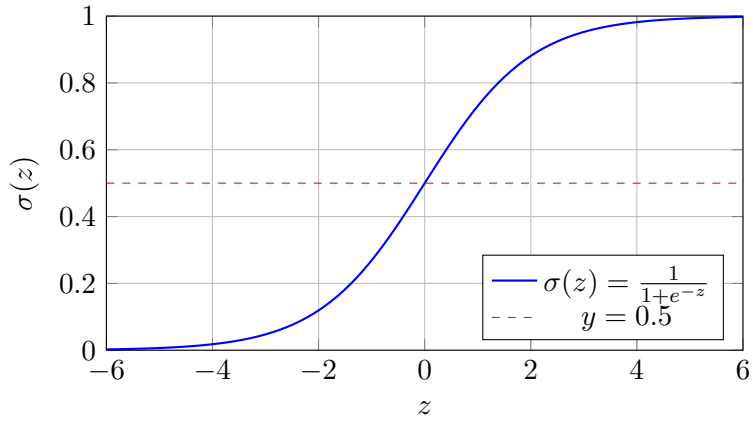

Figure 1: The sigmoid function transforms any real number into a probability between 0 and 1. The steepest part of the curve is near $z = 0$, where small changes in the input cause large changes in the output probability.

13.3 Log-Odds (Logit) Representation

An alternative and highly interpretable way to express the logistic regression model is through log-odds.

Definition 13.3 (Odds and Log-Odds)**.**

$$\text{Odds} = \frac{P(Y=1|\boldsymbol{x})}{P(Y=0|\boldsymbol{x})} = \frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})} \tag{93}$$

$$\text{Log-odds (Logit)} = \ln\left(\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}\right) = \boldsymbol{x}^T\boldsymbol{\beta} \tag{94}$$

The logistic regression model assumes the **log-odds is linear** in the predictors:

$$\text{logit}(\pi(\boldsymbol{x})) = \ln\left(\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{95}$$

13.4 Connection to Generalized Linear Models

Logistic regression is a **Generalized Linear Model (GLM)** with:

- **Random component:** $Y_i \sim \text{Bernoulli}(\pi_i)$
- **Systematic component:** $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$
- **Link function:** $g(\pi) = \text{logit}(\pi) = \ln\left(\frac{\pi}{1-\pi}\right)$

The logit is the **canonical link** for the Bernoulli distribution.

14 Maximum Likelihood Estimation

14.1 The Likelihood Function

For independent observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} P(Y_i = y_i|\boldsymbol{x}_i) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \tag{96}$$

where $\pi_i = \pi(\boldsymbol{x}_i) = \sigma(\boldsymbol{x}_i^T\boldsymbol{\beta})$.

14.2 The Log-Likelihood

$$\ell(\boldsymbol{\beta}) = \ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \ln(\pi_i) + (1-y_i)\ln(1-\pi_i)] \tag{97}$$

Using $\pi_i = \sigma(\boldsymbol{x}_i^T\boldsymbol{\beta})$ and properties of sigmoid:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \ln\left(\frac{\pi_i}{1-\pi_i}\right) + \ln(1-\pi_i) \right] \tag{98}$$

$$= \sum_{i=1}^{n} \left[y_i \boldsymbol{x}_i^T\boldsymbol{\beta} - \ln(1+e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}) \right] \tag{99}$$

14.3 The Cross-Entropy Loss

The **negative log-likelihood** (cross-entropy loss) is:

$$\boxed{J(\boldsymbol{\beta}) = -\frac{1}{n}\ell(\boldsymbol{\beta}) = -\frac{1}{n}\sum_{i=1}^{n} [y_i \ln(\pi_i) + (1-y_i)\ln(1-\pi_i)]} \tag{100}$$

This loss function has an elegant interpretation: it heavily penalizes confident wrong predictions. If the model predicts $\pi = 0.99$ (highly confident of class 1) but the true label is $y = 0$, the term $\ln(1-0.99) = \ln(0.01) \approx -4.6$ contributes a large positive value to the loss. Conversely, confident correct predictions contribute very little to the loss.

Remark 14.1 (Information-Theoretic Interpretation)**.** Cross-entropy measures the "distance" between the true distribution p (with $y \in \{0,1\}$) and predicted distribution q (with probabilities $\pi, 1-\pi$):

$$H(p,q) = -\mathbb{E}_p[\ln q] = -[p\ln q + (1-p)\ln(1-q)] \tag{101}$$

From information theory, cross-entropy quantifies the expected number of bits needed to encode data from distribution p using a code optimized for distribution q. Minimizing cross-entropy encourages the model's predicted distribution to match the empirical distribution of labels.

14.4 Gradient of the Log-Likelihood

To maximize the log-likelihood (or equivalently, minimize the cross-entropy loss), we need its gradient with respect to the parameters.

Theorem 14.1 (Score Function)**.** The gradient (score) of the log-likelihood is:

$$\nabla_{\boldsymbol{\beta}}\ell = \sum_{i=1}^{n}(y_i - \pi_i)\boldsymbol{x}_i = \mathbf{X}^T(\boldsymbol{y} - \boldsymbol{\pi}) \tag{102}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)^T$.

Proof. For a single observation:

$$\frac{\partial \ell_i}{\partial \boldsymbol{\beta}} = \frac{\partial}{\partial \boldsymbol{\beta}} \left[y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \ln(1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}) \right] \tag{103}$$

$$= y_i \boldsymbol{x}_i - \frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}} \boldsymbol{x}_i \tag{104}$$

$$= y_i \boldsymbol{x}_i - \pi_i \boldsymbol{x}_i = (y_i - \pi_i) \boldsymbol{x}_i \tag{105}$$

Summing over all observations gives the result. □

Remark 14.2. The gradient has the same form as in linear regression! The difference is that $\pi_i = \sigma(\boldsymbol{x}_i^T \boldsymbol{\beta})$ is nonlinear in $\boldsymbol{\beta}$. This beautiful correspondence is not coincidental—it reflects the deeper connection through generalized linear models, where logistic regression is the natural choice for binary outcomes.

The gradient $(y_i - \pi_i)\boldsymbol{x}_i$ has an intuitive interpretation: each observation contributes to the gradient proportional to its prediction error $(y_i - \pi_i)$ weighted by its feature vector $\boldsymbol{x}_i$. If $y_i = 1$ but π_i is small (underpredicting), the contribution is positive, pushing the coefficients to increase the prediction. If $y_i = 0$ but π_i is large (overpredicting), the contribution is negative.

14.5 Hessian of the Log-Likelihood

The Hessian matrix (second derivative) tells us about the curvature of the log-likelihood function and is essential for Newton-type optimization methods and for computing standard errors.

Theorem 14.2 (Hessian/Information Matrix)**.** The Hessian is:

$$\mathbf{H} = \nabla^2_{\boldsymbol{\beta}} \ell = -\sum_{i=1}^{n} \pi_i (1 - \pi_i) \boldsymbol{x}_i \boldsymbol{x}_i^T = -\mathbf{X}^T \mathbf{W} \mathbf{X} \tag{106}$$

where $\mathbf{W} = \text{diag}(\pi_1(1 - \pi_1), \ldots, \pi_n(1 - \pi_n))$.

Proof.

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\sum_{i=1}^{n} \frac{\partial \pi_i}{\partial \boldsymbol{\beta}} \boldsymbol{x}_i^T \tag{107}$$

$$= -\sum_{i=1}^{n} \pi_i (1 - \pi_i) \boldsymbol{x}_i \boldsymbol{x}_i^T \tag{108}$$

using $\frac{\partial \pi_i}{\partial \boldsymbol{\beta}} = \pi_i(1 - \pi_i)\boldsymbol{x}_i$. □

Notice that the weights $w_i = \pi_i(1 - \pi_i)$ are largest when $\pi_i = 0.5$ (maximum uncertainty) and smallest when π_i is near 0 or 1 (high confidence). Observations where the model is uncertain contribute most to determining the parameters—this makes intuitive sense, as observations that are clearly in one class or another provide less information about where the decision boundary should be.

Corollary 14.3. The Hessian is negative semi-definite, so the log-likelihood is concave. This guarantees that any local maximum is a global maximum.

This concavity is crucial: it means that gradient-based optimization will always find the global optimum, regardless of initialization. There are no local maxima to get trapped in. This is a significant advantage over many other machine learning models.

14.6 Fisher Information

The **Fisher Information matrix** quantifies how much information the data provides about the parameters:

$$\mathcal{I}(\boldsymbol{\beta}) = -\mathbb{E}[\mathbf{H}] = \mathbf{X}^T\mathbf{W}\mathbf{X} \tag{109}$$

The Fisher Information appears in the asymptotic distribution of the MLE and determines the precision of our parameter estimates.

Asymptotically:

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathcal{I}^{-1}) \tag{110}$$

This result is the foundation for inference in logistic regression. It tells us that for large samples, the MLE is approximately normally distributed, centered at the true parameter value, with variance determined by the inverse Fisher Information. This enables us to construct confidence intervals and perform hypothesis tests.

15 Optimization Algorithms

Unlike linear regression, the logistic regression MLE has no closed-form solution. The log-likelihood is a nonlinear function of $\boldsymbol{\beta}$, and we must resort to iterative optimization methods. Fortunately, the concavity of the log-likelihood ensures that these methods converge to the global optimum.

15.1 Newton-Raphson Method

The Newton-Raphson method uses second-order information (the Hessian) to take optimal steps toward the maximum. The key idea is to approximate the log-likelihood locally by a quadratic function and move to its maximum.

The Newton-Raphson update:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - [\mathbf{H}^{(t)}]^{-1}\nabla_{\boldsymbol{\beta}}\ell^{(t)} \tag{111}$$

Substituting our expressions for the gradient and Hessian:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + (\mathbf{X}^T\mathbf{W}^{(t)}\mathbf{X})^{-1}\mathbf{X}^T(\boldsymbol{y} - \boldsymbol{\pi}^{(t)}) \tag{112}$$

Newton-Raphson typically converges very quickly—often in fewer than 10 iterations—because it uses curvature information to take appropriately sized steps. Near the optimum, it exhibits quadratic convergence, meaning the number of correct digits roughly doubles with each iteration.

15.2 Iteratively Reweighted Least Squares (IRLS)

A beautiful reformulation of Newton-Raphson reveals a connection to weighted least squares. This is the IRLS algorithm.

The Newton-Raphson update can be rewritten as:

$$\boldsymbol{\beta}^{(t+1)} = (\mathbf{X}^T\mathbf{W}^{(t)}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}^{(t)}\boldsymbol{z}^{(t)} \tag{113}$$

where $\boldsymbol{z}^{(t)} = \mathbf{X}\boldsymbol{\beta}^{(t)} + (\mathbf{W}^{(t)})^{-1}(\boldsymbol{y} - \boldsymbol{\pi}^{(t)})$ is the **working response**.

This is a weighted least squares problem solved at each iteration! The algorithm repeatedly solves weighted linear regressions, updating the weights and working response at each step. This insight connects logistic regression to linear regression and allows us to leverage efficient linear algebra routines.

[H] IRLS for Logistic Regression

1. Initialize $\boldsymbol{\beta}^{(0)}$
2. Repeat until convergence:
   (a) Compute $\pi_i^{(t)} = \sigma(\boldsymbol{x}_i^T \boldsymbol{\beta}^{(t)})$
   (b) Compute weights: $w_i^{(t)} = \pi_i^{(t)}(1 - \pi_i^{(t)})$
   (c) Compute working response: $z_i^{(t)} = \boldsymbol{x}_i^T \boldsymbol{\beta}^{(t)} + \frac{y_i - \pi_i^{(t)}}{w_i^{(t)}}$
   (d) Solve: $\boldsymbol{\beta}^{(t+1)} = (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}^{(t)} \boldsymbol{z}^{(t)}$

15.3 Gradient Descent

Update rule:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \alpha \sum_{i=1}^{n} (y_i - \pi_i^{(t)}) \boldsymbol{x}_i \tag{114}$$

For the loss function $J(\boldsymbol{\beta}) = -\frac{1}{n}\ell(\boldsymbol{\beta})$:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \alpha \nabla J = \boldsymbol{\beta}^{(t)} + \frac{\alpha}{n} \mathbf{X}^T (\boldsymbol{y} - \boldsymbol{\pi}^{(t)}) \tag{115}$$

15.4 Stochastic Gradient Descent

Update using single observation:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \alpha (y_i - \pi_i^{(t)}) \boldsymbol{x}_i \tag{116}$$

16 Interpretation of Coefficients

16.1 Effect on Log-Odds

Theorem 16.1 (Interpretation of β_j)**.** A one-unit increase in x_j (holding other variables constant) increases the log-odds by β_j:

$$\ln\left(\frac{\pi(\boldsymbol{x} + \boldsymbol{e}_j)}{1 - \pi(\boldsymbol{x} + \boldsymbol{e}_j)}\right) - \ln\left(\frac{\pi(\boldsymbol{x})}{1 - \pi(\boldsymbol{x})}\right) = \beta_j \tag{117}$$

where $\boldsymbol{e}_j$ is the j-th standard basis vector.

16.2 Odds Ratio

Definition 16.1 (Odds Ratio)**.** The **odds ratio** associated with a one-unit increase in x_j is:

$$\text{OR}_j = e^{\beta_j} = \frac{\text{Odds}(Y = 1 | x_j + 1)}{\text{Odds}(Y = 1 | x_j)} \tag{118}$$

Interpretation:

- OR > 1: increasing x_j increases odds of $Y = 1$
- OR < 1: increasing x_j decreases odds of $Y = 1$
- OR $= 1$: x_j has no effect ($\beta_j = 0$)

16.3 Marginal Effects

The effect of x_j on the **probability** is not constant:

$$\frac{\partial \pi}{\partial x_j} = \frac{\partial \sigma(\boldsymbol{x}^T\boldsymbol{\beta})}{\partial x_j} = \sigma(\boldsymbol{x}^T\boldsymbol{\beta})(1 - \sigma(\boldsymbol{x}^T\boldsymbol{\beta})) \cdot \beta_j = \pi(1-\pi)\beta_j \tag{119}$$

This is maximized when $\pi = 0.5$ (at the decision boundary).

Definition 16.2 (Average Marginal Effect)**.**

$$\text{AME}_j = \frac{1}{n}\sum_{i=1}^{n} \pi_i(1-\pi_i)\beta_j \tag{120}$$

17 Model Evaluation

17.1 Classification Metrics

Definition 17.1 (Confusion Matrix)**.** For threshold τ (typically 0.5):

		Predicted	
		0	1
2*Actual	0	TN	FP
	1	FN	TP

Key metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{121}$$

$$\text{Precision} = \frac{TP}{TP + FP} = P(\text{Actual} = 1 | \text{Predicted} = 1) \tag{122}$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} = P(\text{Predicted} = 1 | \text{Actual} = 1) \tag{123}$$

$$\text{Specificity} = \frac{TN}{TN + FP} = P(\text{Predicted} = 0 | \text{Actual} = 0) \tag{124}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{125}$$

17.2 ROC Curve and AUC

Definition 17.2 (ROC Curve)**.** The **Receiver Operating Characteristic (ROC) curve** plots:

- x-axis: False Positive Rate (FPR) $= 1 - \text{Specificity} = \frac{FP}{FP+TN}$
- y-axis: True Positive Rate (TPR) $= \text{Recall} = \frac{TP}{TP+FN}$

as the classification threshold varies from 0 to 1.

Definition 17.3 (AUC)**.** The **Area Under the ROC Curve (AUC)** equals:

$$\text{AUC} = P(\pi(\boldsymbol{x}_{\text{pos}}) > \pi(\boldsymbol{x}_{\text{neg}})) \tag{126}$$

the probability that a randomly chosen positive example has higher predicted probability than a randomly chosen negative example.

This probabilistic interpretation makes AUC particularly intuitive: it measures the model's ability to rank positive examples above negative examples. A model with AUC = 0.8 will, 80% of the time, assign a higher probability to a randomly selected positive case than to a randomly selected negative case.

Interpretation:

- AUC = 0.5: random guessing (no discriminative ability)
- AUC = 1.0: perfect classification (complete separation)
- AUC $\approx$ 0.7-0.8: acceptable discrimination
- AUC $\approx$ 0.8-0.9: good discrimination
- AUC $>$ 0.9: excellent discrimination

AUC has several advantages as an evaluation metric: it is threshold-independent, it summarizes performance across all possible operating points, and it is insensitive to class imbalance (unlike accuracy).

17.3 Likelihood-Based Measures

While classification metrics evaluate the final predictions, likelihood-based measures evaluate the quality of the probability estimates themselves.

17.3.1 Deviance

The deviance generalizes the residual sum of squares to non-normal likelihoods.

Definition 17.4 (Deviance)**.**

$$D = -2\ell(\hat{\boldsymbol{\beta}}) = -2\sum_{i=1}^{n}[y_i \ln(\hat{\pi}_i) + (1-y_i)\ln(1-\hat{\pi}_i)] \tag{127}$$

Lower deviance indicates better fit. The deviance is analogous to the RSS in linear regression and plays a central role in model comparison.

For comparing nested models:

$$\Delta D = D_{\text{reduced}} - D_{\text{full}} \sim \chi^2_{\Delta p} \tag{128}$$

under H_0 that the additional parameters are zero. This is the likelihood ratio test statistic.

17.3.2 Pseudo-R^2 Measures

Unlike linear regression, logistic regression has no single universally accepted R^2 measure. Several "pseudo-R^2" measures have been proposed, each with different interpretations.

Definition 17.5 (McFadden's R^2)**.**

$$R^2_{\text{McFadden}} = 1 - \frac{\ell(\hat{\boldsymbol{\beta}})}{\ell(\mathbf{0})} \tag{129}$$

where $\ell(\mathbf{0})$ is the log-likelihood of the null model (intercept only).

McFadden's R^2 measures the proportional improvement in log-likelihood relative to the null model. Values between 0.2 and 0.4 are often considered satisfactory. Note that McFadden's R^2 can never reach 1.0 (unlike the linear regression R^2), so direct comparison of magnitudes is not appropriate.

18 Hypothesis Testing

Statistical inference in logistic regression relies on asymptotic (large-sample) theory. Three classical test procedures are available, all asymptotically equivalent but with different computational and practical properties.

18.1 Wald Test

The Wald test is the simplest and most commonly reported test, based on the asymptotic normality of the MLE.

For testing $H_0 : \beta_j = 0$:

$$z = \frac{\hat{\beta}_j}{\text{SE}(\hat{\beta}_j)} \xrightarrow{d} \mathcal{N}(0, 1) \tag{130}$$

where $\text{SE}(\hat{\beta}_j) = \sqrt{[(\mathbf{X}^T\hat{\mathbf{W}}\mathbf{X})^{-1}]_{jj}}$.

The Wald test is computationally convenient because it only requires fitting the full model. However, it can be unreliable when the sample size is small or when the true parameter is far from zero (the standard error estimate may be poor in these cases).

18.2 Likelihood Ratio Test

The likelihood ratio test compares the maximized likelihoods of nested models.

For testing $H_0 : \boldsymbol{\beta}_S = \mathbf{0}$ (subset of coefficients):

$$\Lambda = -2[\ell(\hat{\boldsymbol{\beta}}_{\text{reduced}}) - \ell(\hat{\boldsymbol{\beta}}_{\text{full}})] \xrightarrow{d} \chi^2_q \tag{131}$$

where q is the number of parameters being tested.

The likelihood ratio test is generally preferred over the Wald test because it has better small-sample properties and is invariant to reparameterization. However, it requires fitting both the full and reduced models.

18.3 Score Test (Lagrange Multiplier Test)

The score test evaluates whether the gradient of the log-likelihood is significantly different from zero when evaluated at the null hypothesis.

$$S = \left(\frac{\partial \ell}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}_0}\right)^T \mathcal{I}^{-1}(\boldsymbol{\beta}_0) \left(\frac{\partial \ell}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}_0}\right) \xrightarrow{d} \chi^2_q \tag{132}$$

Advantage: only requires estimation under H_0. This makes the score test computationally attractive when the alternative model is expensive to fit, such as when testing whether to add many variables to a model.

19 Regularization in Logistic Regression

Just as in linear regression, regularization helps prevent overfitting and handles multicollinearity in logistic regression. The ideas are the same, but applied to the log-likelihood rather than the sum of squared errors.

19.1 L2 Regularization (Ridge)

Ridge logistic regression adds an L2 penalty to the log-likelihood:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \underset{\boldsymbol{\beta}}{\text{argmax}} \left\{ \ell(\boldsymbol{\beta}) - \frac{\lambda}{2} \|\boldsymbol{\beta}\|_2^2 \right\} \tag{133}$$

Gradient:

$$\nabla = \mathbf{X}^T(\boldsymbol{y} - \boldsymbol{\pi}) - \lambda\boldsymbol{\beta} \tag{134}$$

The penalty shrinks coefficients toward zero, reducing variance at the cost of some bias. Ridge regularization is particularly useful when predictors are correlated or when p is large relative to n. It also ensures that the optimization problem has a unique solution even when the classes are perfectly separable.

19.2 L1 Regularization (Lasso)

Lasso logistic regression uses an L1 penalty:

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\text{argmax}} \left\{ \ell(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_1 \right\} \tag{135}$$

Produces sparse models (feature selection). The L1 penalty drives some coefficients exactly to zero, automatically selecting a subset of relevant features. This is valuable for interpretability and when many features are believed to be irrelevant. However, the L1 penalty makes the optimization problem non-smooth, requiring specialized algorithms like coordinate descent.

19.3 Elastic Net

Elastic net combines L1 and L2 penalties:

$$\hat{\boldsymbol{\beta}}_{\text{EN}} = \underset{\boldsymbol{\beta}}{\text{argmax}} \left\{ \ell(\boldsymbol{\beta}) - \lambda_1 \|\boldsymbol{\beta}\|_1 - \frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2 \right\} \tag{136}$$

Elastic net inherits the sparsity of lasso while also handling correlated predictors better (lasso tends to arbitrarily select one of a group of correlated predictors, while elastic net includes or excludes them together).

20 Multiclass Extensions

While binary classification is the most common application of logistic regression, many real-world problems involve more than two classes. Two main approaches extend logistic regression to handle multiple classes.

20.1 Multinomial Logistic Regression (Softmax)

Multinomial logistic regression directly generalizes binary logistic regression to $K > 2$ classes.

For K classes, model:

$$P(Y = k|\boldsymbol{x}) = \frac{e^{\boldsymbol{x}^T\boldsymbol{\beta}_k}}{\sum_{j=1}^{K} e^{\boldsymbol{x}^T\boldsymbol{\beta}_j}} \tag{137}$$

This is the **softmax function**:

$$\text{softmax}(\boldsymbol{z})_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \tag{138}$$

The softmax function is a smooth approximation to the argmax function. It converts a vector of real numbers into a probability distribution, with larger inputs receiving larger probabilities. The "temperature" of the softmax can be controlled by scaling the inputs: dividing by a small number makes the distribution more peaked (closer to argmax), while dividing by a large number makes it more uniform.

Properties:

- $\sum_{k=1}^{K} P(Y = k|\boldsymbol{x}) = 1$ — probabilities sum to one by construction
- Reduces to logistic regression when $K = 2$
- The model has $K \times (p + 1)$ parameters, but only $(K - 1) \times (p + 1)$ are identifiable (we typically set $\boldsymbol{\beta}_K = \mathbf{0}$ as a reference)

Multinomial logistic regression is trained by maximizing the multinomial log-likelihood, which is a straightforward extension of the binary cross-entropy.

20.2 One-vs-Rest (OvR)

An alternative approach trains K separate binary classifiers, each distinguishing one class from all others combined.

Train K binary classifiers, each distinguishing class k from all others:

$$P(Y = k|\boldsymbol{x}) \propto \sigma(\boldsymbol{x}^T \boldsymbol{\beta}_k) \tag{139}$$

Prediction: $\hat{y} = \text{argmax}_k \, \sigma(\boldsymbol{x}^T \boldsymbol{\beta}_k)$

OvR is simple to implement and can use any binary classifier. However, it has drawbacks: the K classifiers are trained on different (imbalanced) datasets, and the predicted probabilities from different classifiers are not directly comparable. Multinomial logistic regression is generally preferred when a principled probabilistic model is desired.

21 Decision Boundaries

Understanding the geometry of decision boundaries provides intuition about what logistic regression can and cannot model.

21.1 Linear Decision Boundary

The decision boundary is the set of points where the model is equally uncertain between classes—where the predicted probability equals 0.5.

The decision boundary (where $P(Y = 1|\boldsymbol{x}) = 0.5$) is:

$$\boldsymbol{x}^T \boldsymbol{\beta} = 0 \quad \Leftrightarrow \quad \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0 \tag{140}$$

This is a **hyperplane** in $\mathbb{R}^p$. In two dimensions, it's a line; in three dimensions, a plane. The key insight is that standard logistic regression can only produce linear decision boundaries—it cannot capture curved boundaries without feature engineering.

21.2 Geometric Interpretation

The coefficients have a clear geometric interpretation:

- The normal vector to the hyperplane is $\boldsymbol{\beta}_{1:p} = (\beta_1, \ldots, \beta_p)^T$ — this vector is perpendicular to the decision boundary

- Distance from origin: $\frac{|\beta_0|}{\|\boldsymbol{\beta}_{1:p}\|}$ — the intercept controls how far the boundary is from the origin
- $\boldsymbol{\beta}$ points toward the region where $P(Y = 1|\boldsymbol{x}) > 0.5$ — the positive class region

Moving in the direction of $\boldsymbol{\beta}$ increases the predicted probability of class 1. The magnitude of the coefficients controls how quickly probabilities change as we move away from the boundary: larger coefficients mean sharper transitions.

21.3 Non-linear Decision Boundaries

By including polynomial or interaction terms, logistic regression can model non-linear boundaries:

$$\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 \tag{141}$$

The decision boundary becomes a conic section (circle, ellipse, parabola, hyperbola). With higher-order polynomials, arbitrarily complex boundaries can be approximated. However, this requires manual feature engineering and can lead to overfitting if not regularized.

This is the same principle as polynomial regression: the model remains linear in the parameters even though it's nonlinear in the original features. Modern approaches like kernel methods and neural networks automate this feature construction process.

22 Comparison: Linear vs. Logistic Regression

Despite their different applications, linear and logistic regression share deep structural similarities. Both are members of the generalized linear model (GLM) family, differing only in the choice of link function and error distribution.

Aspect	**Linear Regression**	**Logistic Regression**
Response type	Continuous $y \in \mathbb{R}$	Binary $y \in \{0, 1\}$
Model	$\mathbb{E}[Y\|\boldsymbol{x}] = \boldsymbol{x}^T\boldsymbol{\beta}$	$\ln \frac{P(Y=1\|\boldsymbol{x})}{P(Y=0\|\boldsymbol{x})} = \boldsymbol{x}^T\boldsymbol{\beta}$
Distribution	$Y \sim \mathcal{N}(\boldsymbol{x}^T\boldsymbol{\beta}, \sigma^2)$	$Y \sim \text{Bernoulli}(\sigma(\boldsymbol{x}^T\boldsymbol{\beta}))$
Link function	Identity	Logit
Loss function	MSE	Cross-entropy
Estimation	OLS (closed form)	MLE (iterative)
Coefficient interpretation	Change in $\mathbb{E}[Y]$	Change in log-odds

Table 2: Comparison of linear and logistic regression. Both models use a linear predictor $\boldsymbol{x}^T\boldsymbol{\beta}$, but they differ in how this predictor relates to the response.

The gradient of the loss function has the same form in both cases: $\mathbf{X}^T(\boldsymbol{y} - \hat{\boldsymbol{y}})$, where $\hat{\boldsymbol{y}}$ is the vector of fitted values. In linear regression, $\hat{\boldsymbol{y}} = \mathbf{X}\boldsymbol{\beta}$; in logistic regression, $\hat{\boldsymbol{y}} = \boldsymbol{\pi} = \sigma(\mathbf{X}\boldsymbol{\beta})$. This beautiful correspondence reflects the underlying GLM structure.

23 Practical Considerations

Real-world application of logistic regression requires attention to several practical issues that can affect model performance and reliability.

23.1 Complete Separation

When a hyperplane perfectly separates classes, MLE does not exist (coefficients $\to \pm\infty$).

Complete (or perfect) separation occurs when the two classes can be perfectly distinguished by a linear boundary. While this might seem like a good situation, it causes the likelihood to be maximized at infinity—the model wants to make increasingly confident predictions, pushing coefficients toward $\pm\infty$.

Signs of separation include: coefficients with extremely large magnitudes, standard errors that are orders of magnitude larger than the coefficients, and convergence warnings from the optimization algorithm.

Solutions:

- **Regularization (L1 or L2):** Penalizing large coefficients prevents them from diverging. This is the most common and practical solution.
- **Firth's penalized likelihood:** Adds a specific penalty based on the Jeffreys prior, which has been shown to reduce bias in small samples and handle separation.
- **Exact logistic regression:** Uses conditional inference to compute exact (rather than asymptotic) p-values, avoiding the divergence problem entirely. Computationally intensive for large datasets.

23.2 Class Imbalance

Class imbalance occurs when one class is much more frequent than the other—a common situation in fraud detection, rare disease diagnosis, and anomaly detection.

When $P(Y = 1) \ll P(Y = 0)$:

- **Accuracy is misleading:** A model that predicts the majority class for every observation can achieve high accuracy while being completely useless. If 99% of observations are negative, predicting "negative" always gives 99% accuracy.
- **Use AUC, F1, precision-recall curves:** These metrics are more informative for imbalanced data. AUC measures discrimination regardless of class proportions. F1 balances precision and recall.
- **Consider resampling:** Oversampling the minority class (e.g., SMOTE) or undersampling the majority class can create a more balanced training set.
- **Adjust classification threshold:** Instead of using 0.5, choose a threshold that optimizes your specific objective (e.g., maximizing F1 or achieving a target sensitivity).
- **Use class weights in the loss function:** Weight the minority class more heavily in the loss function, so misclassifying minority examples incurs a larger penalty.

23.3 Feature Scaling

While not strictly necessary (the model is invariant to linear transformations of individual features), scaling features to have similar ranges improves:

- **Convergence speed of gradient descent:** When features are on vastly different scales, the loss surface is elongated, causing gradient descent to take many small zigzagging steps. Scaling creates a more spherical loss surface.
- **Numerical stability:** Very large or very small feature values can cause numerical overflow or underflow in the exponential calculations.

- **Interpretability of regularization:** When features are on the same scale, the regularization penalty treats all coefficients equally. Without scaling, coefficients for large-scale features are penalized more heavily.

Common scaling methods include standardization (subtract mean, divide by standard deviation) and min-max scaling (rescale to $[0, 1]$).

24 Summary of Key Equations

This section collects the most important equations from both parts of this document for quick reference.

24.1 Linear Regression

$$\text{Model:} \quad \boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{142}$$

$$\text{OLS:} \quad \hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y} \tag{143}$$

$$\text{Variance:} \quad \text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \tag{144}$$

$$\text{MSE:} \quad \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{145}$$

$$R^2\text{:} \quad R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{146}$$

24.2 Logistic Regression

$$\text{Model:} \quad P(Y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{x}^T\boldsymbol{\beta}}} \tag{147}$$

$$\text{Log-odds:} \quad \text{logit}(\pi) = \ln\frac{\pi}{1-\pi} = \boldsymbol{x}^T\boldsymbol{\beta} \tag{148}$$

$$\text{Loss:} \quad J = -\frac{1}{n}\sum_{i=1}^{n}[y_i \ln \pi_i + (1 - y_i)\ln(1 - \pi_i)] \tag{149}$$

$$\text{Gradient:} \quad \nabla J = -\frac{1}{n}\mathbf{X}^T(\boldsymbol{y} - \boldsymbol{\pi}) \tag{150}$$

$$\text{Hessian:} \quad \mathbf{H} = \frac{1}{n}\mathbf{X}^T\mathbf{W}\mathbf{X} \tag{151}$$

25 Conclusion

Linear and logistic regression are foundational techniques that every data scientist and statistician must master. Despite the proliferation of more complex methods—random forests, gradient boosting, neural networks—these classical models remain invaluable for their interpretability, computational efficiency, and theoretical guarantees.

Linear regression provides the essential framework for understanding relationships between continuous variables. Its elegant mathematics—the projection interpretation, the Gauss-Markov theorem, the clean distributional results under normality—offer deep insights that transfer to more advanced methods.

Logistic regression extends these ideas to classification, demonstrating how the generalized linear model framework accommodates different response types. The same linear predictor $\boldsymbol{x}^T\boldsymbol{\beta}$ appears in both models, transformed appropriately for the response distribution.

Understanding these models deeply—their assumptions, their geometry, their optimization, their inference procedures—provides the foundation for all of statistical learning. When a complex model fails, the path forward often involves returning to these simpler models to diagnose the problem. When interpretability is paramount, these models are often the right choice.

We encourage readers to not just memorize the formulas, but to understand the underlying principles: why least squares makes sense geometrically, why the sigmoid function is natural for probabilities, why maximum likelihood produces good estimators, and when each model's assumptions are likely to hold or fail. This conceptual understanding will serve you well throughout your statistical career.

26 Further Reading

For readers wishing to deepen their understanding, we recommend the following texts:

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. — A comprehensive treatment of statistical learning methods, freely available online.

2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer. — A more accessible introduction, also freely available online, with R code examples.

3. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. — Emphasizes the Bayesian perspective and connections to neural networks.

4. Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. Wiley. — Excellent coverage of GLMs with a statistical focus.

5. McCullagh, P., & Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall. — The classic reference on GLMs, mathematically rigorous.

6. Gelman, A., & Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press. — Practical guidance with real data examples.