Data Science for Mathematicians
Lesson 6: Logistic Regression and Generalized Linear Models

February 25, 2026

Abstract

This lecture introduces logistic regression as the foundational algorithm for binary classification, serving as a crucial bridge from the continuous domain of linear regression to the discrete domain of classification. We begin by conducting a rigorous analysis of the principled failures of ordinary least squares for binary classification tasks, thereby motivating the necessity of a probabilistic modeling approach. The logistic regression model is then constructed from first principles, starting with the concept of odds and the logit transformation, from which we derive the sigmoid function that maps the linear predictor to a valid probability space.

The lecture will demonstrate how the principle of maximum likelihood estimation, when applied to Bernoulli-distributed data, naturally gives rise to the binary cross-entropy loss function. We will provide an information-theoretic interpretation of this loss and prove its convexity with respect to the model parameters—a critical property that guarantees the convergence of gradient-based optimization methods to a unique global minimum. A full, step-by-step derivation of the gradient of the loss function will be presented, highlighting its elegant mathematical structure.

Finally, we will situate logistic regression within the powerful and unifying framework of generalized linear models. This abstraction provides a robust system for interpreting model parameters in terms of odds ratios and for understanding the relationship between logistic regression, linear regression, and a broader family of statistical models.

Contents

1 Beyond Linear Boundaries

We have focused extensively on the problem of regression, where the goal is to predict a continuous target variable, $y \in \mathbb{R}$. The cornerstone of our approach has been the linear regression model, which posits a linear relationship between the features and the response. For a dataset with n observations and p features, we can write the model in its vectorized form as:

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

where $\mathbf{y} \in \mathbb{R}^n$ is the vector of responses, $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ is the design matrix (with an intercept column), $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ is the vector of model parameters, and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a vector of error terms.

Our primary statistical assumption on this model, which justifies the use of ordinary least squares (OLS), is that the errors are independent and identically distributed according to a Gaussian distribution with zero mean and constant variance, i.e., $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. This implies a probabilistic model for the response variable itself:

$$Y_i \mid \mathbf{x}_i; \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}^T \mathbf{x}_i, \sigma^2).$$

We showed that the OLS solution, $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, can be understood geometrically as the projection of $\mathbf{y}$ onto the column space of $\mathbf{X}$. In Week 5, we revisited this problem from an optimization perspective, framing it as the minimization of a loss function—the mean squared error (MSE)—and solving for $\hat{\boldsymbol{\beta}}$ using iterative algorithms like gradient descent.

1.1 The Classification Problem

We now pivot from predicting continuous quantities to a new class of problems central to data science: **classification**. In classification, the target variable y is not continuous but discrete, representing a category or class label. For this lecture, we will focus on the simplest, yet most common, variant: **binary classification**, where the target variable can take one of two values, which we will canonically encode as $y_i \in \{0, 1\}$.

This framework applies to a vast array of real-world problems:

- **Medical diagnosis:** Given patient data (e.g., age, blood pressure, tumor size), predict whether a tumor is malignant ($y = 1$) or benign ($y = 0$).
- **Spam detection:** Given the content and metadata of an email, classify it as spam ($y = 1$) or not spam ($y = 0$).
- **Fraud detection:** Given transaction details, determine if a credit card transaction is fraudulent ($y = 1$) or legitimate ($y = 0$).

The fundamental task is to learn a function that maps an input feature vector $\mathbf{x} \in \mathbb{R}^p$ to a predicted class label $\hat{y} \in \{0, 1\}$.

1.2 Why Linear Regression is Unsuitable for Classification

Given our success with linear regression, a natural first thought is to apply it directly to the classification problem. We could simply treat the binary labels $\{0, 1\}$ as numerical values and fit a model $\hat{y} = \boldsymbol{\beta}^T\mathbf{x}$ using OLS. A prediction could then be made by thresholding: if $\hat{y} > 0.5$, predict class 1; otherwise, predict class 0. While this approach may seem plausible, it is fundamentally flawed for several rigorous mathematical and statistical reasons.

The output of a linear model, $\hat{y} = \boldsymbol{\beta}^T\mathbf{x}$, is an unbounded real number, i.e., $\hat{y} \in (-\infty, \infty)$. However, in classification, we are fundamentally interested in the *probability* of an observation belonging to a particular class, $\mathbb{P}(Y = 1|X = \mathbf{x})$. Probabilities are, by definition, constrained to the interval $[0, 1]$. Interpreting the unbounded output of a linear model as a probability is mathematically incoherent. The model can easily produce predictions like 1.5 or -0.3, which have no probabilistic meaning. This alone is a critical failure of the modeling approach.

A core assumption of OLS is that the error terms, $\epsilon_i = y_i - \boldsymbol{\beta}^T\mathbf{x}_i$, have a constant variance, $\text{Var}(\epsilon_i) = \sigma^2$, for all values of the predictors $\mathbf{x}_i$. This property is known as homoscedasticity. In the case of a binary response variable, this assumption is systematically violated.

Let $p_i = \mathbb{P}(Y_i = 1|\mathbf{x}_i)$. The response variable Y_i follows a Bernoulli distribution, $Y_i \sim \text{Bernoulli}(p_i)$. The mean and variance of a Bernoulli random variable are as follows:

$$\mathbb{E}[Y_i|\mathbf{x}_i] = p_i,$$

$$\text{Var}(Y_i|\mathbf{x}_i) = p_i(1 - p_i).$$

If our linear model were a reasonable approximation, we would have $p_i \approx \boldsymbol{\beta}^T\mathbf{x}_i$. The variance of the response (and thus the error) would be $\text{Var}(Y_i|\mathbf{x}_i) \approx (\boldsymbol{\beta}^T\mathbf{x}_i)(1 - \boldsymbol{\beta}^T\mathbf{x}_i)$. This variance is clearly not constant; it is a quadratic function of the predicted mean. This violation of homoscedasticity, known as heteroscedasticity, invalidates the statistical inference procedures associated with OLS, such as the calculation of standard errors, confidence intervals, and p-values for the coefficients.

The probabilistic justification for using least squares is that it is the maximum likelihood estimator under the assumption that the errors are normally distributed. For a binary response $y_i \in \{0, 1\}$, the error term $\epsilon_i = y_i - \boldsymbol{\beta}^T \mathbf{x}_i$ can only take on two values for any given $\mathbf{x}_i$: $1 - \boldsymbol{\beta}^T \mathbf{x}_i$ (when $y_i = 1$) and $-\boldsymbol{\beta}^T \mathbf{x}_i$ (when $y_i = 0$). An error term that can only take two values cannot possibly be drawn from a continuous, bell-shaped Gaussian distribution. This fundamental mismatch between the assumed and actual error distributions means that the entire statistical foundation of OLS is inappropriate for this problem.

1.3 Logistics Regression Modeling

The failures of linear regression for classification tasks are not minor issues to be ignored; they are fundamental violations of the model's underlying assumptions. This forces us to abandon the idea of modeling the discrete outcome y directly. Instead, the key quantity to model is the conditional probability of the outcome: $p(\mathbf{x}) \equiv \mathbb{P}(Y = 1 | X = \mathbf{x})$.

This leads us to our central objective: to find a function that takes the linear combination of our features, $\boldsymbol{\beta}^T \mathbf{x}$, and transforms it into a valid probability. This is precisely the role of **Logistic Regression**. It retains the linear core that makes linear models so powerful and interpretable, but wraps it in a non-linear function to correctly handle the probabilistic nature of classification.

Our starting point is the linear predictor, which we will denote as η:

$$\eta = \boldsymbol{\beta}^T \mathbf{x} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p.$$

As established, η can take any value on the real line, $\eta \in (-\infty, \infty)$. Our goal is to find a principled transformation that maps this unbounded value to a probability $p(\mathbf{x}) \in [0, 1]$. We will build this transformation in two logical steps.

Next, we introduce the concept of the **odds ratio**. For an event with probability p, the odds are defined as the ratio of the probability of the event occurring to the probability of it not occurring.

$$\text{odds} = \frac{p}{1-p}.$$

Example 1.1. If the probability of a horse winning a race is $p = 0.8$, the odds of winning are $\frac{0.8}{1-0.8} = \frac{0.8}{0.2} = 4$. We would say the odds are "4 to 1 in favor" of winning. Conversely, if the probability of winning is $p = 0.2$, the odds are $\frac{0.2}{0.8} = 0.25$, or "1 to 4 against" winning (or "4 to 1 against").

The odds ratio provides a useful transformation of the probability scale. While p is constrained to $(0, 1)$, the odds can take any value in $(0, \infty)$. This transformation has successfully removed the upper bound of 1, but the lower bound of 0 remains.

To remove the remaining lower bound and map our quantity to the entire real line, we take the natural logarithm of the odds. This gives the **log-odds**, which is the core of our transformation.

Definition 1.2 (Logit function)**.** The logit function logit: $(0, 1) \to \mathbb{R}$ is defined by

$$\operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right).$$

Proposition 1.3. *The logit function* logit *is a strictly increasing bijection satisfying* $\operatorname{logit}(1/2) = 0$, $\operatorname{logit}(p) \to -\infty$ *as* $p \to 0^+$, *and* $\operatorname{logit}(p) \to +\infty$ *as* $p \to 1^-$.

Proof. Since log is strictly increasing and $p \mapsto p/(1-p)$ is strictly increasing on $(0, 1)$, their composition logit is strictly increasing. As $p \to 0^+$, the odds $p/(1-p) \to 0^+$, so $\operatorname{logit}(p) \to -\infty$. As $p \to 1^-$, the odds $p/(1-p) \to +\infty$, so $\operatorname{logit}(p) \to +\infty$. At $p = 1/2$, the odds equal 1, giving $\operatorname{logit}(1/2) = \log 1 = 0$. By the intermediate value theorem, logit surjects onto $\mathbb{R}$; strict monotonicity implies injectivity. Hence logit is a bijection $(0, 1) \to \mathbb{R}$. □

Example 1.4. For $p = 0.9$, the odds are $\frac{0.9}{0.1} = 9$ and the log-odds are $\operatorname{logit}(0.9) = \log 9 \approx 2.197$. For $p = 0.1$, the odds are $\frac{0.1}{0.9} \approx 0.111$ and the log-odds are $\operatorname{logit}(0.1) \approx -2.197$. The symmetry $\operatorname{logit}(p) = -\operatorname{logit}(1-p)$ reflects that $p = 0.5$ maps to log-odds of zero, the point of maximum uncertainty.

We now have a quantity, the log-odds, that has the same range as our linear predictor $\eta = \boldsymbol{\beta}^T\mathbf{x}$. This allows us to make the central modeling assumption of logistic regression: we assume that the *log-odds of the outcome is a linear function of the predictors.*

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \boldsymbol{\beta}^T\mathbf{x}.$$

This is the fundamental equation of the logistic regression model. It is a linear model, but not for the probability itself; it is a linear model for the *logit-transformed* probability.

The core assumption gives us a relationship between $p(\mathbf{x})$ and $\boldsymbol{\beta}^T\mathbf{x}$, but it is implicit. To use this model for prediction, we need to express $p(\mathbf{x})$ explicitly as a function of $\boldsymbol{\beta}^T\mathbf{x}$. We can achieve this by algebraically inverting the logit function.

Let $\eta = \boldsymbol{\beta}^T\mathbf{x}$. We start with the core assumption:

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \eta.$$

To solve for $p(\mathbf{x})$, we first exponentiate both sides to undo the logarithm:

$$\frac{p(\mathbf{x})}{1-p(\mathbf{x})} = e^{\eta}.$$

Now, we solve for $p(\mathbf{x})$:

$$\begin{aligned} p(\mathbf{x}) &= e^{\eta}(1-p(\mathbf{x})) \\ p(\mathbf{x}) &= e^{\eta} - e^{\eta}p(\mathbf{x}) \\ p(\mathbf{x}) + e^{\eta}p(\mathbf{x}) &= e^{\eta} \\ p(\mathbf{x})(1+e^{\eta}) &= e^{\eta} \\ p(\mathbf{x}) &= \frac{e^{\eta}}{1+e^{\eta}}. \end{aligned}$$

This form is perfectly valid. However, it is conventional to divide both the numerator and the denominator by e^{η} to express the function in terms of $e^{-\eta}$:

$$p(\mathbf{x}) = \frac{e^{\eta}/e^{\eta}}{(1+e^{\eta})/e^{\eta}} = \frac{1}{1/e^{\eta}+1} = \frac{1}{1+e^{-\eta}}.$$

Substituting $\eta = \boldsymbol{\beta}^T\mathbf{x}$ back in, we arrive at the explicit form of our model for the probability:

$$p(\mathbf{x}) = \frac{1}{1+e^{-\boldsymbol{\beta}^T\mathbf{x}}}.$$

This S-shaped function is known as the **sigmoid** or **logistic** function.

Definition 1.5 (Sigmoid function)**.** The sigmoid function $\sigma\colon \mathbb{R} \to (0,1)$ is defined as

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

Using this notation, the logistic regression model is concisely written as $p(\mathbf{x}) = \sigma(\boldsymbol{\beta}^T\mathbf{x})$.

Example 1.6. We compute $\sigma(0) = \dfrac{1}{1+e^0} = 0.5$, corresponding to maximum uncertainty. At $z = 2$, $\sigma(2) = \dfrac{1}{1+e^{-2}} \approx 0.880$, and at $z = -2$, $\sigma(-2) \approx 0.119$. In a logistic regression model, if the linear predictor $\boldsymbol{\beta}^T\mathbf{x} = 2$, the model assigns a probability of approximately 88% to class 1.

Proposition 1.7. *The sigmoid function σ is strictly increasing and satisfies:*

1. *$\sigma(z) \to 0$ as $z \to -\infty$ and $\sigma(z) \to 1$ as $z \to +\infty$;*
2. *$\sigma(0) = 1/2$;*
3. *$\sigma(-z) = 1 - \sigma(z)$ for all $z \in \mathbb{R}$.*

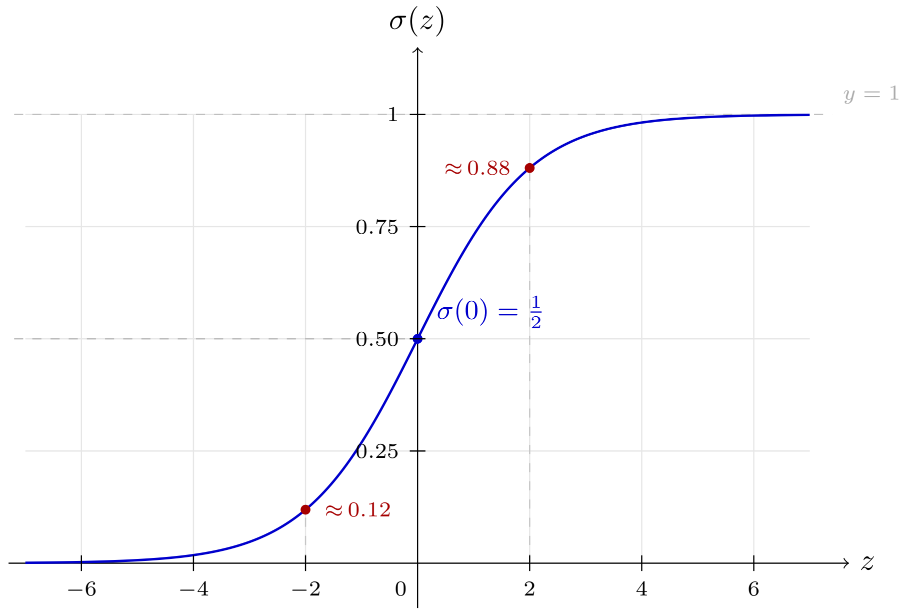

Figure 1: The sigmoid function $\sigma(z) = 1/(1 + e^{-z})$, mapping the entire real line to the open interval $(0, 1)$. The curve passes through $\sigma(0) = 1/2$ (maximum uncertainty) and approaches the horizontal asymptotes 0 and 1 as $z \to -\infty$ and $z \to +\infty$, respectively. The marked points at $z = \pm 2$ illustrate the rapid transition from low to high probability around the origin.

Proof. Since $e^{-z} > 0$ for all $z \in \mathbb{R}$, we have $\sigma(z) = 1/(1+e^{-z}) \in (0, 1)$. Strict monotonicity follows from Theorem 1.8: $\sigma'(z) = \sigma(z)(1 - \sigma(z)) > 0$ since $\sigma(z) \in (0, 1)$. As $z \to +\infty$, $e^{-z} \to 0$, so $\sigma(z) \to 1$; as $z \to -\infty$, $e^{-z} \to +\infty$, so $\sigma(z) \to 0$. At $z = 0$, we have $e^0 = 1$, giving $\sigma(0) = 1/2$. The anti-symmetry identity follows by direct computation:

$$\sigma(-z) = \frac{1}{1 + e^{z}} = 1 - \frac{e^{z}}{1 + e^{z}} = 1 - \frac{1}{1 + e^{-z}} = 1 - \sigma(z). \qquad \square$$

Figure 1 illustrates the characteristic S-shape of the sigmoid function. Its bounded range $(0, 1)$ ensures that model outputs are always valid probabilities, resolving the central flaw of linear regression identified in Section 1.2. The steep but smooth transition around $z = 0$ reflects the model's region of greatest uncertainty, while the flat tails near the asymptotes correspond to confident predictions. This smooth, differentiable shape is essential: it guarantees that the gradient of the loss function with respect to $\boldsymbol{\beta}$ is well-defined everywhere, enabling gradient descent to train the model.

One of the most elegant and useful properties of the sigmoid function is its derivative, which can be expressed in terms of the function itself. This property will be indispensable when we derive the gradient for our optimization algorithm.

Proposition 1.8. *The derivative of the sigmoid function σ with respect to z satisfies*

$$\frac{\mathrm{d}}{\mathrm{d}z}\sigma(z) = \sigma(z)(1 - \sigma(z)).$$

Proof. We apply the quotient rule with $u(z) = 1$ and $v(z) = 1 + e^{-z}$, so $u'(z) = 0$ and $v'(z) = -e^{-z}$:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}z}\sigma(z) &= \frac{u'(z)v(z) - u(z)v'(z)}{[v(z)]^2} \\ &= \frac{0 \cdot (1 + e^{-z}) - 1 \cdot (-e^{-z})}{(1 + e^{-z})^2} \\ &= \frac{e^{-z}}{(1 + e^{-z})^2}. \end{aligned}$$

To show this equals $\sigma(z)(1-\sigma(z))$, we add and subtract 1 in the numerator:

$$\begin{aligned}\frac{e^{-z}}{(1+e^{-z})^2} &= \frac{1+e^{-z}-1}{(1+e^{-z})^2}\\ &= \frac{1+e^{-z}}{(1+e^{-z})^2} - \frac{1}{(1+e^{-z})^2}\\ &= \frac{1}{1+e^{-z}} - \left(\frac{1}{1+e^{-z}}\right)^2\\ &= \sigma(z) - [\sigma(z)]^2\\ &= \sigma(z)(1-\sigma(z)).\end{aligned}$$

This completes the proof. □

2 Parameter Estimation via Maximum Likelihood

Having defined the model $p(\mathbf{x};\boldsymbol{\beta}) = \sigma(\boldsymbol{\beta}^T\mathbf{x})$, we turn to parameter estimation. Because Y_i is Bernoulli-distributed rather than Gaussian, minimizing squared errors is no longer the appropriate criterion. Applying maximum likelihood estimation to the Bernoulli likelihood yields the correct objective function for logistic regression.

Since $Y_i \sim \text{Bernoulli}(p_i)$ with $p_i = \sigma(\boldsymbol{\beta}^T\mathbf{x}_i)$, the probability of a single observation $(\mathbf{x}_i, y_i)$ under the model is

$$\mathbb{P}(Y_i = y_i \mid \mathbf{x}_i;\, \boldsymbol{\beta}) = p_i^{y_i}(1-p_i)^{1-y_i}.$$

Under the i.i.d. assumption, the likelihood of the dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ factors as

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n p_i^{y_i}(1-p_i)^{1-y_i},$$

and taking the natural logarithm yields the log-likelihood

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^n \big[y_i \log\sigma(\boldsymbol{\beta}^T\mathbf{x}_i) + (1-y_i)\log\big(1-\sigma(\boldsymbol{\beta}^T\mathbf{x}_i)\big)\big].$$

Negating the log-likelihood converts the MLE maximization into the standard minimization form $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$.

Definition 2.1. The **binary cross-entropy (BCE)** loss, also called the **log loss**, is the negative log-likelihood of the Bernoulli model:

$$J(\boldsymbol{\beta}) = -\ell(\boldsymbol{\beta}) = -\sum_{i=1}^n \big[y_i \log p_i + (1-y_i)\log(1-p_i)\big],$$

where $p_i = \sigma(\boldsymbol{\beta}^T\mathbf{x}_i)$. The per-sample contribution is $J_i(\boldsymbol{\beta}) = -[y_i \log p_i + (1-y_i)\log(1-p_i)]$.

Example 2.2. With true label $y_i = 1$ and predicted probability $p_i = 0.9$, the per-sample loss is $J_i = -\log(0.9) \approx 0.105$. With $p_i = 0.1$, it rises to $J_i = -\log(0.1) \approx 2.303$: a confident but wrong prediction incurs a severe penalty.

Because $-\log$ is strictly decreasing and diverges as its argument approaches 0, the BCE loss penalizes confident wrong predictions asymptotically while assigning negligible loss to confident correct ones.

The name *cross-entropy* comes from information theory. For a single observation with true label y_i, the empirical distribution P on $\{0,1\}$ is deterministic ($P(Y = y_i) = 1$), while the model provides a predicted distribution $Q = (1-p_i,\, p_i)$. The cross-entropy

$$H(P,Q) = -\sum_{k\in\{0,1\}} P(Y=k)\log Q(Y=k) = -\big[y_i\log p_i + (1-y_i)\log(1-p_i)\big] = J_i(\boldsymbol{\beta})$$

measures the average coding cost when using Q to encode outcomes drawn from P. Minimizing the BCE loss is therefore equivalent to minimizing the information-theoretic dissimilarity between the model's predicted distribution and the empirical distribution of the observed labels.

2.1 Convexity of the Loss Function

One of the most important theoretical properties of logistic regression is that its loss function is convex.

Theorem 2.3. *The binary cross-entropy loss function $J(\boldsymbol{\beta})$ is a convex function with respect to the parameter vector $\boldsymbol{\beta}$.*

Proof. A function is convex if its Hessian matrix (the matrix of second partial derivatives) is positive semi-definite. The loss function is $J(\boldsymbol{\beta}) = \sum_{i=1}^{n} J_i(\boldsymbol{\beta})$. Since the sum of convex functions is convex, we only need to show that the Hessian of a single loss term, $\nabla^2 J_i(\boldsymbol{\beta})$, is positive semi-definite.

In the next section, we will derive the gradient of the loss function as $\nabla J(\boldsymbol{\beta}) = \sum_{i=1}^{n}(p_i - y_i)\mathbf{x}_i$. Taking a second derivative with respect to $\boldsymbol{\beta}$ requires differentiating this expression again. Using the chain rule, one can show that the Hessian matrix takes the form

$$\nabla^2 J(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}^T}\left(\sum_{i=1}^{n}(\sigma(\boldsymbol{\beta}^T\mathbf{x}_i) - y_i)\mathbf{x}_i\right) = \sum_{i=1}^{n}\sigma(\boldsymbol{\beta}^T\mathbf{x}_i)(1-\sigma(\boldsymbol{\beta}^T\mathbf{x}_i))\mathbf{x}_i\mathbf{x}_i^T.$$

We analyze this expression. For any vector $\mathbf{v} \in \mathbb{R}^{p+1}$, consider the quadratic form $\mathbf{v}^T(\nabla^2 J(\boldsymbol{\beta}))\mathbf{v}$:

$$\mathbf{v}^T(\nabla^2 J(\boldsymbol{\beta}))\mathbf{v} = \mathbf{v}^T\left(\sum_{i=1}^{n}\sigma_i(1-\sigma_i)\mathbf{x}_i\mathbf{x}_i^T\right)\mathbf{v} = \sum_{i=1}^{n}\sigma_i(1-\sigma_i)\mathbf{v}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}.$$

We can rewrite $\mathbf{v}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}$ as $(\mathbf{x}_i^T\mathbf{v})^T(\mathbf{x}_i^T\mathbf{v}) = (\mathbf{x}_i^T\mathbf{v})^2$. Since $\mathbf{x}_i^T\mathbf{v}$ is a scalar, its square is always non-negative.

$$\mathbf{v}^T(\nabla^2 J(\boldsymbol{\beta}))\mathbf{v} = \sum_{i=1}^{n}\sigma_i(1-\sigma_i)(\mathbf{x}_i^T\mathbf{v})^2.$$

The term $\sigma_i(1-\sigma_i)$ is also always non-negative, since $p_i = \sigma_i$ is a probability in $[0,1]$. Therefore, the Hessian is a sum of positive semi-definite matrices (specifically, outer products scaled by non-negative scalars). The sum of positive semi-definite matrices is also positive semi-definite. This proves that $J(\boldsymbol{\beta})$ is a convex function of $\boldsymbol{\beta}$. $\square$

The implication of this theorem is profound: it guarantees that the loss surface does not have any local minima that are not also global minima. Therefore, an iterative optimization algorithm like gradient descent, if run for long enough with a suitable learning rate, is guaranteed to converge to the unique, global minimum of the loss function. This is a significant advantage over many more complex models, such as neural networks, which typically have highly non-convex loss landscapes riddled with suboptimal local minima.

2.2 Gradient Descent for Logistic Regression

We have now fully formulated our optimization problem: find the parameter vector $\boldsymbol{\beta}$ that minimizes the convex binary cross-entropy loss function.

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}}\left(-\sum_{i=1}^{n}\left[y_i\log p_i + (1-y_i)\log(1-p_i)\right]\right).$$

Unlike the OLS problem, there is no closed-form analytical solution for $\boldsymbol{\beta}^*$ (i.e., no equivalent of the normal equations). We must therefore rely on an iterative numerical optimization algorithm to find the solution. The natural choice, building on our work from Week 5, is gradient descent.

The gradient descent algorithm provides a simple and powerful way to find the minimum of a differentiable function. Starting from an initial guess $\boldsymbol{\beta}^{(0)}$, we iteratively update the parameters by taking a small step in the direction of the negative gradient of the loss function. The update rule for iteration t takes the form

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \eta\nabla J(\boldsymbol{\beta}^{(t)}),$$

where η is the learning rate, a hyperparameter that controls the step size. The only component we need to implement this algorithm is the gradient of our loss function, $\nabla J(\boldsymbol{\beta})$.

We will now derive the partial derivative of the loss function $J(\boldsymbol{\beta})$ with respect to a single parameter β_j. This derivation is a classic application of the chain rule and showcases the elegance of the mathematical structure we have built. Let $J(\boldsymbol{\beta}) = \sum_{i=1}^{n} J_i(\boldsymbol{\beta})$, where

$$J_i(\boldsymbol{\beta}) = -[y_i \log p_i + (1 - y_i)\log(1 - p_i)].$$

Let $p_i = \sigma(z_i)$ and

$$z_i = \boldsymbol{\beta}^T \mathbf{x}_i = \sum_{k=0}^{p} \beta_k x_{ik}.$$

We need to compute $\frac{\partial J}{\partial \beta_j}$. Since the derivative of a sum is the sum of derivatives, we have

$$\frac{\partial J}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial J_i}{\partial \beta_j}.$$

We apply the chain rule to find the derivative for a single term J_i:

$$\frac{\partial J_i}{\partial \beta_j} = \frac{\partial J_i}{\partial p_i} \cdot \frac{\partial p_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial \beta_j}.$$

We compute each of the three components:

1. **Derivative of the loss with respect to the prediction (p_i):**

$$\begin{aligned} \frac{\partial J_i}{\partial p_i} &= \frac{\partial}{\partial p_i}\left(-[y_i \log p_i + (1 - y_i)\log(1 - p_i)]\right) \\ &= -\left[\frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i}\right] = \frac{-(y_i(1 - p_i) - (1 - y_i)p_i)}{p_i(1 - p_i)} = \frac{p_i - y_i}{p_i(1 - p_i)}. \end{aligned}$$

2. **Derivative of the sigmoid function (p_i) with respect to its input (z_i):** This follows from Theorem 1.8.

$$\frac{\partial p_i}{\partial z_i} = \frac{\mathrm{d}\sigma(z_i)}{\mathrm{d}z_i} = \sigma(z_i)(1 - \sigma(z_i)) = p_i(1 - p_i).$$

3. **Derivative of the linear predictor (z_i) with respect to the parameter (β_j):**

$$\frac{\partial z_i}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\left(\sum_{k=0}^{p} \beta_k x_{ik}\right) = x_{ij}.$$

Now, we multiply these three components together:

$$\frac{\partial J_i}{\partial \beta_j} = \left(\frac{p_i - y_i}{p_i(1 - p_i)}\right) \cdot (p_i(1 - p_i)) \cdot x_{ij}.$$

A remarkable cancellation occurs. The term $p_i(1 - p_i)$ in the denominator from the first part is exactly cancelled by the same term from the derivative of the sigmoid function. This leaves us with an exceptionally simple result:

$$\frac{\partial J_i}{\partial \beta_j} = (p_i - y_i)x_{ij}.$$

Summing over all data points gives the final partial derivative for the total loss:

$$\frac{\partial J}{\partial \beta_j} = \sum_{i=1}^{n}(p_i - y_i)x_{ij} = \sum_{i=1}^{n}(\sigma(\boldsymbol{\beta}^T \mathbf{x}_i) - y_i)x_{ij}.$$

This derivation is a cornerstone of understanding how logistic regression is trained.

2.3 The Vectorized Gradient

We can express the full gradient vector, $\nabla J(\boldsymbol{\beta}) \in \mathbb{R}^{p+1}$, in a compact vectorized form. Let $\mathbf{p}$ be the $n \times 1$ column vector of predicted probabilities, and $\mathbf{y}$ be the $n \times 1$ column vector of true labels. The gradient equals

$$\nabla J(\boldsymbol{\beta}) = \mathbf{X}^T(\mathbf{p} - \mathbf{y}),$$

where $\mathbf{X}$ is the $n \times (p+1)$ design matrix. This form is not only notationally clean but is also essential for efficient implementation in numerical computing environments.

We compare this gradient to the MSE gradient for OLS,

$$\nabla J_{\text{OLS}}(\boldsymbol{\beta}) = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}).$$

Both share the same fundamental structure: $\mathbf{X}^T(\text{prediction} - \text{truth})$. This is not a coincidence. It is a deep property that arises when using a loss function derived from an exponential family distribution paired with its canonical link function, a direct consequence of the principled construction of our model.

Example 2.4. Consider a medical screening problem with a single predictor x (a standardized biomarker level) and a binary outcome y (disease present or absent). We are given $n = 6$ observations.

i	1	2	3	4	5	6
x_i	-2	-1	0	1	2	3
y_i	0	0	0	1	1	1

We fit a logistic regression model with an intercept: $p(x) = \sigma(\beta_0 + \beta_1 x)$, where σ is the sigmoid function. Augmenting each observation with a leading 1 for the intercept, the design matrix and label vector are

$$\mathbf{X} = \begin{pmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

We initialize $\boldsymbol{\beta}^{(0)} = (0, 0)^T$ and set the learning rate $\eta = 0.1$.

Iteration 0. With $\boldsymbol{\beta}^{(0)} = (0, 0)^T$, every linear predictor is $z_i = 0 + 0 \cdot x_i = 0$, so every predicted probability is $p_i = \sigma(0) = 0.5$. The residual vector is

$$\mathbf{p} - \mathbf{y} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{pmatrix}.$$

The gradient is computed via $\nabla J = \mathbf{X}^T(\mathbf{p} - \mathbf{y})$:

$$\nabla J = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 0 \\ -4.5 \end{pmatrix}.$$

The zero first component reflects the balanced dataset; the negative second component indicates that β_1 should increase, consistent with the positive association between the biomarker and disease presence. We update the parameters:

$$\boldsymbol{\beta}^{(1)} = \boldsymbol{\beta}^{(0)} - \eta \nabla J = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0.1 \begin{pmatrix} 0 \\ -4.5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.45 \end{pmatrix}.$$

The single matrix product subsumes both partial derivatives, illustrating the computational advantage of the vectorized gradient.

Iteration 1. With $\boldsymbol{\beta}^{(1)} = (0, 0.45)^T$, the linear predictors are $z_i = 0.45x_i$, giving

$$\mathbf{z} = (-0.90,\ -0.45,\ 0,\ 0.45,\ 0.90,\ 1.35)^T.$$

Applying the sigmoid function yields the predicted probabilities

$$\mathbf{p} \approx (0.289,\ 0.389,\ 0.500,\ 0.611,\ 0.711,\ 0.794)^T.$$

The residual vector $\mathbf{p} - \mathbf{y}$ is approximately $(0.289, 0.389, 0.500, -0.389, -0.289, -0.206)^T$. The gradient is

$$\nabla J = \mathbf{X}^T(\mathbf{p} - \mathbf{y}) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 0.289 \\ 0.389 \\ 0.500 \\ -0.389 \\ -0.289 \\ -0.206 \end{pmatrix} \approx \begin{pmatrix} 0.294 \\ -2.334 \end{pmatrix}.$$

The parameter update is

$$\boldsymbol{\beta}^{(2)} = \begin{pmatrix} 0 \\ 0.45 \end{pmatrix} - 0.1 \begin{pmatrix} 0.294 \\ -2.334 \end{pmatrix} = \begin{pmatrix} -0.029 \\ 0.683 \end{pmatrix}.$$

The small negative shift in β_0 adjusts the decision boundary, while β_1 continues to grow, sharpening the model's discrimination between the two classes.

Tracking the loss. We compute the binary cross-entropy loss at each iteration to verify convergence. At iteration 0, with $p_i = 0.5$ for all i, the loss is $J(\boldsymbol{\beta}^{(0)}) = -6\log(0.5) = 6\log 2 \approx 4.159$. The following table records the loss over the first several iterations.

t	0	1	2	3	4	5	10
$J(\boldsymbol{\beta}^{(t)})$	4.159	2.592	2.035	1.743	1.556	1.423	1.068

Each iteration reduces the loss, consistent with the convexity guarantee established earlier. Figure 2 visualizes this monotone decrease over 20 iterations of gradient descent. The steep initial drop reflects rapid improvement from the uninformative initialization $\boldsymbol{\beta}^{(0)} = \mathbf{0}$, while the flattening tail indicates approach to the optimum.

Continuing this process for many iterations, the parameters stabilize near $\hat{\boldsymbol{\beta}} \approx (-0.41, 1.55)^T$. The fitted model is

$$\hat{p}(x) = \sigma(-0.41 + 1.55\,x) = \frac{1}{1 + e^{-(-0.41+1.55\,x)}}.$$

To convert the model's probabilistic output into a class label, we apply a decision threshold, conventionally set at 0.5. The predicted label is

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{p}(x) \geq 0.5, \\ 0 & \text{if } \hat{p}(x) < 0.5. \end{cases}$$

Since $\sigma(z) = 0.5$ if and only if $z = 0$, the decision boundary occurs where $\beta_0 + \beta_1 x = 0$, i.e., at $x^* = -\beta_0/\beta_1 \approx 0.41/1.55 \approx 0.26$. Observations with $x > 0.26$ are classified as positive.

Applying the fitted model $\hat{p}(x) = \sigma(-0.41 + 1.55\,x)$ to each training observation, the predictions and classifications at threshold 0.5 are as follows.

i	1	2	3	4	5	6
x_i	-2	-1	0	1	2	3
y_i	0	0	0	1	1	1
$\hat{p}(x_i)$	0.03	0.12	0.40	0.76	0.94	0.99
$\hat{y}_i$	0	0	0	1	1	1

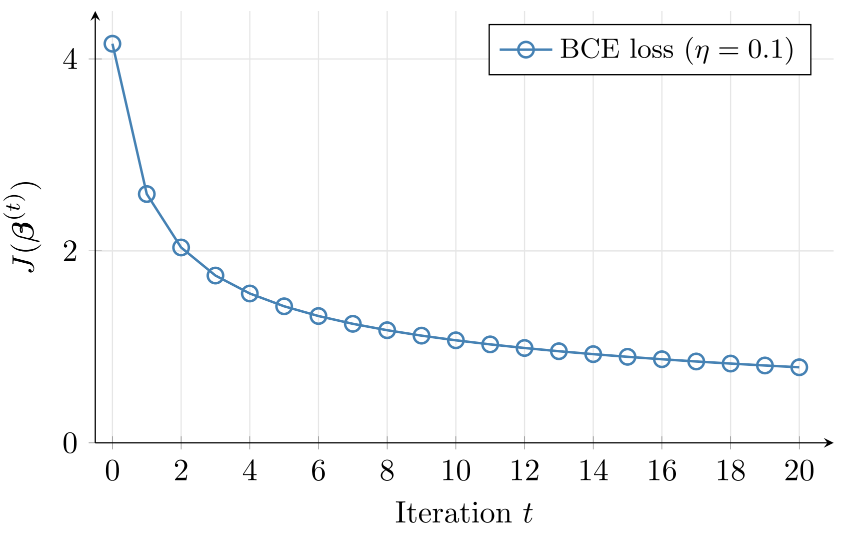

Figure 2: Binary cross-entropy loss $J(\boldsymbol{\beta}^{(t)})$ over 20 iterations of gradient descent with learning rate $\eta = 0.1$ for the medical screening model. The loss decreases monotonically from $J \approx 4.16$ at initialization to $J \approx 0.79$ at iteration 20.

The model correctly classifies all six observations, assigning high confidence to the extreme points and lower confidence near the decision boundary.

The confusion matrix for the fitted model is as follows:

	Predicted 1	Predicted 0
Actual 1	3	0
Actual 0	0	3

We read off TP = 3, TN = 3, FP = 0, and FN = 0. The performance metrics are

$$\text{Accuracy} = \frac{3+3}{6} = 1.0, \qquad \text{Precision} = \frac{3}{3+0} = 1.0, \qquad \text{Recall} = \frac{3}{3+0} = 1.0,$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot 1.0 \cdot 1.0}{1.0 + 1.0} = 1.0.$$

The model achieves perfect performance on the training set. While this is expected for a linearly separable toy dataset with only six points, real-world performance must be assessed on held-out test data to guard against overfitting.

Example 2.5. A financial institution models loan approval ($y = 1$ for approved, $y = 0$ for denied) using two standardized predictors: annual income (x_1) and credit history length (x_2). The training data consists of $n = 6$ past applicants.

i	1	2	3	4	5	6
x_{1i}	-2	-1	0	1	0	2
x_{2i}	-1	-2	-1	1	2	1
y_i	0	0	0	1	1	1

We fit the model $p(\mathbf{x}) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$ via gradient descent. Augmenting each observation with a leading 1 for the intercept, the design matrix and label vector are

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & -1 \\ 1 & -1 & -2 \\ 1 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 1 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

We initialize $\boldsymbol{\beta}^{(0)} = (0,0,0)^T$ and set the learning rate $\eta = 0.1$.

Iteration 0. With $\boldsymbol{\beta}^{(0)} = (0,0,0)^T$, the vector of linear predictors is $\mathbf{z} = \mathbf{X}\boldsymbol{\beta}^{(0)} = \mathbf{0}$, so every predicted probability is $p_i = \sigma(0) = 0.5$. The residual vector is

$$\mathbf{p} - \mathbf{y} = (0.5,\ 0.5,\ 0.5,\ -0.5,\ -0.5,\ -0.5)^T.$$

The gradient is computed via the vectorized formula $\nabla J = \mathbf{X}^T(\mathbf{p} - \mathbf{y})$:

$$\nabla J = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 0 & 2 \\ -1 & -2 & -1 & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 0 \\ -3.0 \\ -4.0 \end{pmatrix}.$$

The zero first component reflects the balanced dataset. The negative second and third components indicate that both β_1 and β_2 should increase, consistent with the positive association between both predictors and loan approval. We update the parameters:

$$\boldsymbol{\beta}^{(1)} = \boldsymbol{\beta}^{(0)} - \eta \nabla J = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - 0.1 \begin{pmatrix} 0 \\ -3.0 \\ -4.0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.30 \\ 0.40 \end{pmatrix}.$$

Iteration 1. With $\boldsymbol{\beta}^{(1)} = (0, 0.30, 0.40)^T$, the linear predictors are

$$\mathbf{z} = \mathbf{X}\boldsymbol{\beta}^{(1)} = \begin{pmatrix} 1 & -2 & -1 \\ 1 & -1 & -2 \\ 1 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0.30 \\ 0.40 \end{pmatrix} = \begin{pmatrix} -1.00 \\ -1.10 \\ -0.40 \\ 0.70 \\ 0.80 \\ 1.00 \end{pmatrix}.$$

Applying the sigmoid function element-wise yields the predicted probabilities

$$\mathbf{p} = \sigma(\mathbf{z}) \approx (0.269,\ 0.250,\ 0.401,\ 0.668,\ 0.690,\ 0.731)^T.$$

The residual vector is $\mathbf{p} - \mathbf{y} \approx (0.269,\ 0.250,\ 0.401,\ -0.332,\ -0.310,\ -0.269)^T$. The gradient is

$$\nabla J = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 0 & 2 \\ -1 & -2 & -1 & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 0.269 \\ 0.250 \\ 0.401 \\ -0.332 \\ -0.310 \\ -0.269 \end{pmatrix} \approx \begin{pmatrix} 0.009 \\ -1.657 \\ -2.391 \end{pmatrix}.$$

The parameter update is

$$\boldsymbol{\beta}^{(2)} = \begin{pmatrix} 0 \\ 0.30 \\ 0.40 \end{pmatrix} - 0.1 \begin{pmatrix} 0.009 \\ -1.657 \\ -2.391 \end{pmatrix} = \begin{pmatrix} -0.001 \\ 0.466 \\ 0.639 \end{pmatrix}.$$

Both β_1 and β_2 continue to grow, sharpening the model's discrimination between the two classes. The small negative shift in β_0 adjusts the intercept to account for the slight asymmetry introduced by the updated slopes.

Tracking the loss. We compute the binary cross-entropy loss at each iteration to verify convergence. At iteration 0, with $p_i = 0.5$ for all i, the loss is $J(\boldsymbol{\beta}^{(0)}) = -6\log(0.5) = 6\log 2 \approx 4.159$. The following table records the loss over the first several iterations.

Each iteration reduces the loss, consistent with the convexity guarantee established earlier. Figure 3 visualizes this monotone decrease over 20 iterations of gradient descent. The steep initial drop reflects rapid improvement from the uninformative initialization $\boldsymbol{\beta}^{(0)} = \mathbf{0}$, while the flattening tail indicates approach to the optimum. Continuing this process, the parameters stabilize near $\hat{\boldsymbol{\beta}} \approx (-0.10,\ 1.10,\ 1.80)^T$. The fitted model is

$$\hat{p}(\mathbf{x}) = \sigma(-0.10 + 1.10\,x_1 + 1.80\,x_2).$$

t	0	1	2	3	4	5	10
$J(\boldsymbol{\beta}^{(t)})$	4.159	2.201	1.495	1.146	0.938	0.799	0.473

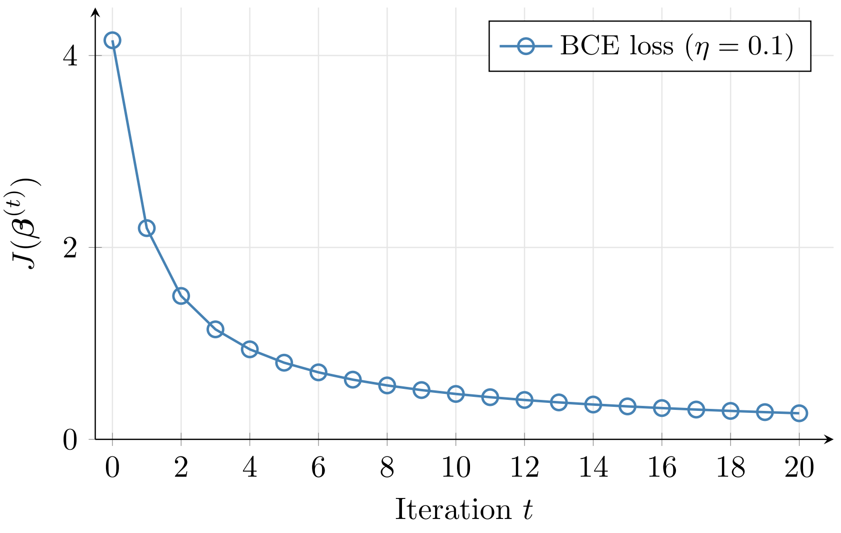

Figure 3: Binary cross-entropy loss $J(\boldsymbol{\beta}^{(t)})$ over 20 iterations of gradient descent with learning rate $\eta = 0.1$ for the loan approval model. The loss decreases monotonically from $J \approx 4.16$ at initialization to $J \approx 0.27$ at iteration 20, illustrating the convergence guarantee provided by convexity.

Interpreting the coefficients. We interpret the fitted parameters using the odds ratio framework. The odds ratio for x_1 is $e^{1.10} \approx 3.00$: for a one-unit increase in standardized income, the odds of loan approval are multiplied by approximately 3, holding credit history constant. The odds ratio for x_2 is $e^{1.80} \approx 6.05$: a one-unit increase in standardized credit history multiplies the odds of approval by approximately 6. Both coefficients are positive, confirming that higher income and longer credit history are associated with increased odds of approval. The larger magnitude of $\hat{\beta}_2$ indicates that credit history has a stronger influence on the model's predictions than income.

Predicting on new applicants. We apply the fitted model to 8 new applicants with known true outcomes from a held-out test set.

i	1	2	3	4	5	6	7	8
x_{1i}	-1.5	0.5	-0.5	1.0	0.5	-0.5	1.5	0.0
x_{2i}	-0.5	-1.0	0.5	0.5	1.0	1.5	1.0	0.0
z_i	-2.65	-1.35	0.25	1.90	2.25	2.05	3.35	-0.10
$\hat{p}_i$	0.066	0.206	0.562	0.870	0.905	0.886	0.966	0.475
$\hat{y}_i$	0	0	1	1	1	1	1	0
y_i	0	0	0	1	1	1	1	1

Evaluating classification performance. The confusion matrix for the test set is as follows.

	Predicted 1	Predicted 0
Actual 1	4	1
Actual 0	1	2

We read off TP = 4, TN = 2, FP = 1, and FN = 1. The performance metrics are

$$\text{Accuracy} = \frac{4+2}{8} = 0.750, \qquad \text{Precision} = \frac{4}{4+1} = 0.800, \qquad \text{Recall} = \frac{4}{4+1} = 0.800,$$

$$F_1 = \frac{2 \cdot 0.800 \cdot 0.800}{0.800 + 0.800} = 0.800.$$

The two misclassifications highlight important practical considerations. Applicant 3 (a false positive) has $\hat{p}_3 = 0.562$, just above the 0.5 threshold, illustrating the sensitivity of the decision boundary for borderline cases. Applicant 8 (a false negative) has $\hat{p}_8 = 0.475$, just below the threshold, and was denied despite ultimately being creditworthy. Such near-boundary errors motivate the inclusion of additional predictors or the careful tuning of the decision threshold to balance precision and recall for the application at hand.

3 Generalized Linear Models

Once we have used gradient descent to find the optimal parameter vector $\hat{\boldsymbol{\beta}}$, we must interpret what these coefficients mean. In OLS, the interpretation is direct: β_j is the expected change in the response variable y for a one-unit increase in the predictor x_j, holding all other predictors constant.

In logistic regression, the relationship is not as direct because our model is linear on the log-odds scale, not the probability scale. Therefore, the coefficient β_j represents the additive change in the log-odds of the outcome for a one-unit increase in x_j.

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \cdots + \beta_j x_j + \cdots + \beta_p x_p.$$

If we increase x_j by one unit to $x_j + 1$, the new log-odds become

$$\log\left(\frac{p'}{1-p'}\right) = \beta_0 + \cdots + \beta_j(x_j + 1) + \cdots + \beta_p x_p = \log\left(\frac{p}{1-p}\right) + \beta_j.$$

So, a one-unit increase in x_j increases the log-odds by β_j.

3.1 The Odds Ratio Interpretation

While mathematically precise, an interpretation in terms of log-odds is difficult to interpret directly. We obtain a more interpretable quantity by exponentiating the coefficient. This relationship implies

$$\log\left(\frac{\text{odds}'}{\text{odds}}\right) = \beta_j \implies \frac{\text{odds}'}{\text{odds}} = e^{\beta_j}.$$

This gives a multiplicative interpretation in terms of the odds. The quantity e^{β_j} is called the **odds ratio (OR)** associated with the predictor x_j.

In general, for a one-unit increase in the predictor x_j, the odds of the outcome occurring ($y = 1$) are multiplied by a factor of e^{β_j}, holding all other predictors constant.

- If $\beta_j > 0$, then $e^{\beta_j} > 1$, and an increase in x_j is associated with an *increase* in the odds of the outcome.
- If $\beta_j < 0$, then $e^{\beta_j} < 1$, and an increase in x_j is associated with a *decrease* in the odds of the outcome.
- If $\beta_j = 0$, then $e^{\beta_j} = 1$, and the predictor x_j has no effect on the odds of the outcome.

Example 3.1. Suppose a medical study fits a logistic regression model to predict the 10-year risk of heart disease ($y = 1$). One of the predictors is a binary variable for smoking ($x_{\text{smoke}} = 1$ for smokers, 0 for non-smokers). The fitted model yields a coefficient $\hat{\beta}_{\text{smoke}} = 0.693$.

To interpret this, we calculate the odds ratio: $OR = e^{0.693} \approx 2.0$. The odds of developing heart disease for a smoker are approximately 2 times the odds for a non-smoker, after controlling for other variables in the model.

The odds ratio interpretation illustrates a broader pattern: the link function determines how we interpret the coefficients $\boldsymbol{\beta}$. In logistic regression the logit link yields multiplicative effects on the odds; in OLS the identity link yields additive effects on the mean. This observation suggests that both models share a common structure, differing only in their choice of response distribution and link function. We now formalize this idea.

3.2 The Exponential Family

For the unified framework to work, we require the response distribution to belong to a specific class.

Definition 3.2 (One-parameter exponential family)**.** A probability distribution belongs to the **one-parameter exponential family** if its density (or mass function) can be written as

$$f(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right\},$$

where θ is the **natural parameter**, $b(\theta)$ is the **log-partition function**, $\phi > 0$ is a dispersion parameter, and $c(y,\phi)$ is a normalization term. Key properties follow from the log-partition function: $\mathbb{E}[Y] = b'(\theta)$ and $\mathrm{Var}(Y) = \phi\, b''(\theta)$.

Example 3.3 (Bernoulli and Gaussian as exponential family members)**.** We verify that the two distributions used so far are members.

- *Bernoulli.* For $Y \sim \mathrm{Bernoulli}(p)$, the probability mass function is $p^y(1-p)^{1-y}$. We rewrite it as

  $$\exp\bigl\{y \log p + (1-y)\log(1-p)\bigr\} = \exp\left\{y \log\frac{p}{1-p} + \log(1-p)\right\}.$$

  Setting $\theta = \log\bigl(p/(1-p)\bigr)$ (the log-odds), we obtain $p = e^\theta/(1+e^\theta)$ and $b(\theta) = \log(1+e^\theta)$, with $\phi = 1$ and $c(y,\phi) = 0$.

- *Gaussian.* For $Y \sim \mathcal{N}(\mu,\sigma^2)$, the density is $(2\pi\sigma^2)^{-1/2}\exp\{-(y-\mu)^2/(2\sigma^2)\}$. Expanding the square gives

  $$\exp\left\{\frac{y\mu - \mu^2/2}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\right\}.$$

  Here $\theta = \mu$, $b(\theta) = \theta^2/2$, $\phi = \sigma^2$, and $c(y,\phi) = -y^2/(2\phi) - \frac{1}{2}\log(2\pi\phi)$.

3.3 Generalized Linear Models

Linear regression and logistic regression are both instances of a single model family: **generalized linear models (GLMs)**. The GLM framework encompasses many statistical models under a single theoretical structure.

Any GLM is defined by three interacting components:

1. **Random component:** The response variable Y_i follows a distribution from the exponential family (Definition 3.2 above). The choice of distribution determines the support of Y_i and the form of the variance function $\mathrm{Var}(Y_i) = \phi\, b''(\theta_i)$.

2. **Systematic component:** A linear predictor $\eta_i = \boldsymbol{\beta}^T\mathbf{x}_i$ combines the explanatory variables. This component is identical across all GLMs.

3. **Link function:** A monotonic, differentiable function g connects the conditional mean $\mu_i = \mathbb{E}[Y_i]$ to the linear predictor:

   $$g(\mu_i) = \eta_i.$$

   The link function must map the range of μ_i to the full real line $\mathbb{R}$. This requirement constrains which links are appropriate for each distribution: the identity maps $\mathbb{R} \to \mathbb{R}$ for Gaussian means, the logit maps $(0,1) \to \mathbb{R}$ for Bernoulli probabilities, and the log maps $(0,\infty) \to \mathbb{R}$ for Poisson means.

Remark 3.4 (Canonical link). Each exponential family distribution has a distinguished link function called the **canonical link**, defined by $g(\mu_i) = \theta_i$, where θ_i is the natural parameter. Since $\mu = b'(\theta)$, the canonical link is $g = (b')^{-1}$. For the Bernoulli distribution, the natural parameter is $\theta = \log\bigl(p/(1-p)\bigr)$, so the canonical link is the logit. For the Gaussian, $\theta = \mu$, so the canonical link is the identity. Using the canonical link simplifies the score equations and guarantees that the sufficient statistic $\mathbf{X}^T\mathbf{y}$ appears directly in the likelihood, which explains the elegant gradient form $\mathbf{X}^T(\mathbf{p} - \mathbf{y})$ we derived in Section 2.

Table 1: Three instances of the GLM framework. All three share the same systematic component $\eta_i = \boldsymbol{\beta}^T \mathbf{x}_i$ but differ in the response distribution, the range of the mean, and the corresponding canonical link.

Component	**OLS**	**Logistic Regression**	**Poisson Regression**
Random	$Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$	$Y_i \sim \text{Bernoulli}(p_i)$	$Y_i \sim \text{Poisson}(\lambda_i)$
Range of μ_i	$\mathbb{R}$	$(0, 1)$	$(0, \infty)$
Canonical link	Identity: $g(\mu_i) = \mu_i$	Logit: $g(p_i) = \log \frac{p_i}{1 - p_i}$	Log: $g(\lambda_i) = \log \lambda_i$

3.4 Situating Our Models within the GLM Framework

Using this framework, we can see that linear regression and logistic regression are two specific instances of a GLM, differing only in their random component and link function. Table 1 adds a third instance—Poisson regression for count data—to illustrate the framework's generality.

To move from linear to logistic regression, we change the assumed distribution of the response from Gaussian to Bernoulli. This change constrains the mean to $(0, 1)$, which necessitates the logit link. The new distribution in turn yields a new likelihood and a new loss function (cross-entropy). The same reasoning applies when we move to Poisson regression for count data: the mean λ_i must be positive, so we use the log link.

Example 3.5 (Poisson regression for count data)**.** A manufacturing engineer models the number of defects Y_i on a circuit board as a function of two predictors: board area x_{i1} (in cm^2) and soldering temperature x_{i2} (in °C). We assume $Y_i \sim \text{Poisson}(\lambda_i)$ with the log link:

$$\log \lambda_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}.$$

Suppose the fitted model yields $\hat{\beta}_0 = -1.386$, $\hat{\beta}_1 = 0.030$, and $\hat{\beta}_2 = 0.008$. To interpret the coefficients, we exponentiate: $e^{\hat{\beta}_1} = e^{0.030} \approx 1.030$. Each additional cm^2 of board area multiplies the expected defect count by approximately 1.03, a 3% increase. For a board with $x_1 = 50$ cm^2 and $x_2 = 250\,°\text{C}$, the predicted mean defect count is

$$\hat{\lambda} = \exp(-1.386 + 0.030 \times 50 + 0.008 \times 250) = \exp(2.114) \approx 8.28.$$

Remark 3.6 (Iterative estimation). For GLMs with a non-identity link, no closed-form solution for $\hat{\boldsymbol{\beta}}$ exists. Fitting requires iterative numerical methods such as the gradient descent algorithm developed in Section 2, or Newton–Raphson and iteratively reweighted least squares (IRLS). The choice of canonical link simplifies these algorithms because the score equations take a particularly clean form.

4 Conclusion

In this lecture, we have made the critical transition from regression to classification. We began by establishing that naively applying linear regression to a binary classification problem fails on principled grounds: the Gaussian error assumption is violated, predicted values escape $[0, 1]$, and the decision boundary lacks a probabilistic interpretation. This motivated a shift in our modelling paradigm: instead of predicting the outcome directly, we must predict the *probability* of the outcome.

This probabilistic approach led us to construct the logistic regression model from the ground up, deriving every component from first principles. We then situated logistic regression within the generalized linear model framework, revealing it as one instance of a broader family of models. The key results are collected below.

Logistic regression model. The sigmoid function maps the linear predictor to a valid probability:

$$p(\mathbf{x}) = \sigma(\boldsymbol{\beta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \mathbf{x}}}, \qquad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

The binary cross-entropy loss and its gradient are

$$J(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \big[y_i \log p_i + (1 - y_i) \log(1 - p_i) \big], \qquad \nabla J(\boldsymbol{\beta}) = \mathbf{X}^T (\mathbf{p} - \mathbf{y}),$$

where $p_i = \sigma(\boldsymbol{\beta}^T \mathbf{x}_i)$. The loss is convex, so gradient descent converges to the unique global minimum.

Classification rule. Given a threshold τ (conventionally $\tau = 0.5$), an observation $\mathbf{x}$ is classified as positive when the predicted probability exceeds the threshold:

$$\hat{y} = \begin{cases} 1 & \text{if } \sigma(\boldsymbol{\beta}^T\mathbf{x}) \geq \tau, \\ 0 & \text{if } \sigma(\boldsymbol{\beta}^T\mathbf{x}) < \tau. \end{cases}$$

Since σ is strictly increasing and $\sigma(0) = 1/2$, the default threshold $\tau = 0.5$ places the decision boundary at $\boldsymbol{\beta}^T\mathbf{x} = 0$.

Coefficient interpretation. Each coefficient has a direct odds-ratio interpretation: e^{β_j} gives the multiplicative change in the odds of the positive class for a unit increase in feature x_j, with all other features held constant.

Generalized linear model framework. Every GLM consists of three components:

1. a *random component*—a distribution from the exponential family,
$$f(y; \theta, \phi) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y, \phi)\right\},$$
2. a *systematic component*—the linear predictor $\eta_i = \boldsymbol{\beta}^T\mathbf{x}_i$,
3. a *link function* g such that $g(\mu_i) = \eta_i$, mapping the mean to the real line.

The three models developed in this course instantiate the framework as follows:

	Linear regression	**Logistic regression**	**Poisson regression**
Distribution	$Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$	$Y_i \sim \text{Bernoulli}(p_i)$	$Y_i \sim \text{Poisson}(\lambda_i)$
Mean range	$\mathbb{R}$	$(0, 1)$	$(0, \infty)$
Canonical link	$g(\mu) = \mu$	$g(p) = \log \frac{p}{1-p}$	$g(\lambda) = \log \lambda$
Loss (neg. log-lik.)	MSE	Cross-entropy	Poisson deviance

Together, these results demonstrate how a principled probabilistic approach—choosing a distributional assumption, deriving the loss via maximum likelihood, and connecting the mean to the linear predictor through a canonical link—provides both a rigorous foundation for classification and a unified perspective on the statistical models developed throughout this course.