Understanding the Logit Model (Continued)

The Logistic Function (Sigmoid)

As derived previously, the **logistic function**, more commonly called the **Sigmoid function** (especially when using base *e*), is the key component that converts the linear combination of predictors (log-odds) into a probability between 0 and 1.

Its standard form (using base *e*) is often represented by the Greek letter sigma (σ):

* σ(t) = e^t / (e^t + 1) = 1 / (1 + e^(-t))

Where:

* t represents the linear combination of the input features and parameters: t = β₀ + β₁x₁ + β₂x₂ + ... + β<0xE2><0x82><0x99>x<0xE2><0x82><0x99> (For a simple case with one explanatory variable x, this is t = β₀ + β₁x).
* σ(t) outputs the estimated probability p = P(Y=1).

**In essence, the Sigmoid function takes any real input t (log-odds, potentially ranging from -∞ to +∞) and maps it to an output probability between 0 and 1.**

The general logistic function providing the probability output can thus be written as:

* p = σ(t) = 1 / (1 + e^-(β₀ + β₁x + ...))

Interpreting the Coefficients (βs)

Understanding the coefficients (β₀, β₁, ...) in logistic regression requires thinking in terms of **log-odds** and **odds**, not directly probability (due to the non-linear transformation by the sigmoid function).

Let's consider an example (using log-base 10 for illustration, as in the slides, though base *e* is standard): Suppose our fitted model is:

log₁₀( p / (1-p) ) = l = -3 + 1\*x₁ + 2\*x₂

Where p is the probability of the event Y=1.

* **Intercept (β₀ = -3):**
  + This is the **log-odds** (base 10) of the event Y=1 when all predictor variables (x₁, x₂) are equal to zero.
  + The corresponding *odds* when x₁=x₂=0 are 10^(-3) = 0.001.
* **Coefficient for x₁ (β₁ = 1):**
  + This means that a **one-unit increase in x₁** (holding x₂ constant) increases the **log-odds** (base 10) of the event Y=1 by **1**.
  + This corresponds to multiplying the **odds** of the event Y=1 by a factor of 10¹ = 10.
  + Note: While the *probability* p also increases, it does **not** increase by a fixed amount or factor; the relationship is non-linear. The simple multiplicative relationship holds for the *odds*.
* **Coefficient for x₂ (β₂ = 2):**
  + A **one-unit increase in x₂** (holding x₁ constant) increases the **log-odds** (base 10) of the event Y=1 by **2**.
  + This corresponds to multiplying the **odds** of the event Y=1 by a factor of 10² = 100.
  + Comparing β₂ and β₁: The effect of x₂ on the *log-odds* is twice as great as the effect of x₁. However, the effect of x₂ on the *odds* (10²=100) is 10 times greater than the effect of x₁ (10¹=10) because of the exponentiation involved in converting log-odds to odds. Again, the effect on the final *probability* p is more complex and non-linear.

**(When using the standard natural logarithm (base *e*), the interpretation is the same, but the multiplicative factor for the odds becomes e^(βᵢ) for a one-unit increase in xᵢ).**

Example: Diabetes Prediction

Let's illustrate Logistic Regression with a simple example: determining whether a person has diabetes based on their blood sugar level reading.

* **Data:** We have data points showing Blood Sugar Level and whether the person is Diabetic (Yes/No). We typically encode the target variable: Diabetes = 1 (Yes), Diabetes = 0 (No).
* **Problem Statement:** Given a specific Blood Sugar value (e.g., 210), what is the **probability** of that person being Diabetic (P(Diabetes=1))?

The initial scatter plot shows that people with lower blood sugar levels tend to be Non-Diabetic (0), and those with higher levels tend to be Diabetic (1). There's a transition zone. Linear regression is unsuitable here because we want to predict a probability between 0 and 1, not a continuous line that extends beyond this range.

Modeling Probability with the Sigmoid Curve

Logistic Regression uses the Sigmoid function to model this probability:

P(Diabetes=1) = σ(β₀ + β₁ \* BloodSugarLevel) = 1 / (1 + e^-(β₀ + β₁ \* BloodSugarLevel))

This function produces an "S"-shaped curve that transitions smoothly from 0 to 1.

* The curve shows the estimated probability of being Diabetic for any given blood sugar level.
* For low blood sugar levels, the probability is close to 0.
* For high blood sugar levels, the probability is close to 1.
* In the transition zone, the probability reflects the uncertainty.

**Challenge:** How do we find the **best-fitting Sigmoid curve**? This means finding the optimal combination of the parameters **β₀ (intercept)** and **β₁ (slope for BloodSugarLevel)** that makes the curve best represent the observed data points.

Finding the Best Fit: Maximum Likelihood Estimation

Unlike Linear Regression which minimizes the Sum of Squared Residuals (SSR/MSE), Logistic Regression typically finds the best parameters using **Maximum Likelihood Estimation (MLE)**.

* **Goal:** Find the parameter values (β₀, β₁) that **maximize the likelihood** (the probability) of observing the actual outcomes (Diabetes = 0 or Diabetes = 1) in our training data, given the model's predicted probabilities.
* **Intuition:**
  + For individuals who *are* Diabetic (Actual=1), we want the model's predicted probability pᵢ = P(Diabetes=1 | BloodSugarᵢ) to be as close to 1 as possible.
  + For individuals who are *not* Diabetic (Actual=0), we want the model's predicted probability pᵢ to be as close to 0 as possible, which means (1 - pᵢ) should be as close to 1 as possible.
* **Likelihood Function (L):** The overall likelihood of observing the entire dataset is the product of the individual probabilities for each data point: L = Product [ pᵢ ] (for all Diabetic patients where Actual=1) \* Product [ 1 - pᵢ ] (for all Non-Diabetic patients where Actual=0)

*Where pᵢ = σ(β₀ + β₁ \* BloodSugarLevelᵢ)*

* **Optimization:** The learning algorithm searches for the values of β₀ and β₁ that make this likelihood product L as large as possible.
* **Example:** The slide shows a calculation of the Likelihood for a specific choice of parameters (β₀ = -10, β₁ = 0.06). The algorithm would try different parameter values to find the ones yielding the highest Likelihood.

Likelihood, Cost Functions, and Gradient Descent

* **Maximizing Likelihood ≈ Minimizing Cost:** In practice, working with the *product* in the Likelihood function is mathematically inconvenient (especially taking derivatives). It's much easier to work with the *sum* of *logarithms*. Maximizing the Likelihood is equivalent to maximizing the **Log-Likelihood**, which in turn is equivalent to **minimizing the negative Log-Likelihood**. This negative Log-Likelihood function serves as the **Cost Function** for Logistic Regression (often called Log Loss or Binary Cross-Entropy).
* **Gradient Descent:** Just like in Linear Regression, **Gradient Descent** is the iterative optimization algorithm used to find the parameter values (β₀, β₁) that minimize this cost function (and thus maximize the likelihood). It calculates the gradient (derivatives) of the cost function and updates the parameters in the opposite direction until convergence. The update rule conceptually remains similar, although the specific derivative calculations differ due to the different cost function.

Revisiting Odds and Probability with the Diabetes Example

Once the model is trained (e.g., finding β₀ = -13.5 and β₁ = 0.06 as in the slide example), we can calculate probabilities:

P(Diabetes=1) = 1 / (1 + e^-(-13.5 + 0.06 \* BloodSugarLevel))

Plugging in different BloodSugarLevel values yields the probabilities shown in the table:

| **Sugar Level** | **Probability P(Diabetes=1)** |
| --- | --- |
| 120 | 0.18% |
| 130 | 9.96% |
| 140 | 16.78% |
| 150 | 26.88% |
| 160 | 40.13% |
| ... | ... |
| 190 | 80.23% |
| 200 | 88.09% |
| 210 | 93.10% |

Notice the non-linear relationship: Increasing blood sugar by 10 points has a much larger effect on the probability in the middle range (e.g., 150 to 160) than at the extremes (e.g., 120 to 130 or 200 to 210). While the *log-odds* change linearly with blood sugar (0.06 per unit increase), the probability follows the sigmoid curve.