STA 35C: Statistical Data Science III

Lecture 8: Classification Basics & Logistic Regression

Dogyoon Song

Spring 2025, UC Davis

Agenda

So far: Regression

Today:

- Classification overview
  - What is classification?
  - How it differs from regression
- Logistic regression
  - Basic ideas
  - Model formulation
  - Prediction with logistic regression
  - Parameter estimation

Classification: Motivation

Classification = Supervised learning to predict *qualitative* (categorical) responses

Examples:

- Email spam vs. non-spam
- Fraudulent transaction vs. legitimate
- Medical diagnosis (multiple possible conditions)
- Handwritten digit classification (0–9)

Key difference from regression:

- Y is a *class label*, not a numeric value
- We often interpret output as the *probability* of a class
- Accuracy metrics differ (e.g., classification error, confusion matrix)

Classification: A visual illustration

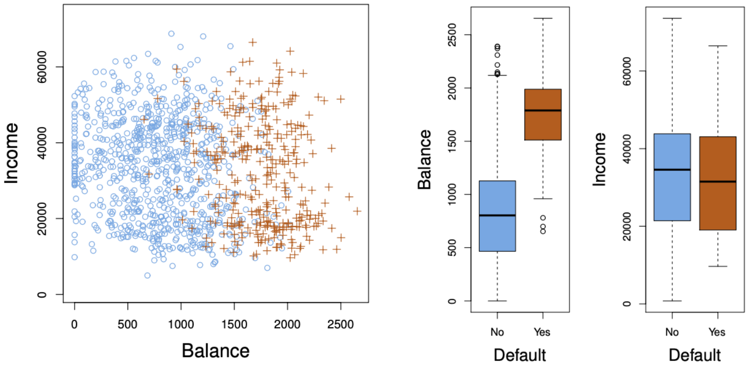


Figure: The `Default` dataset: annual incomes vs. monthly credit card balances. Orange: individuals who defaulted, Blue: those who did not [JWHT21, Figure 4.1].

Goal: Find a rule or bondary that assigns a new point x_{new} to the correct class

Classification setting: Formal description

Goal: Given data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$,

- Assign each $x \in \mathbb{R}^p$ to one of K classes $y \in \{1, \ldots, K\}$
- Learn a model $f : \mathbb{R}^p \to \{1, \ldots, K\}$ that predicts the class Y for given X

Example

- Email text $(X) \to$ spam or not (Y)
- Handwritten image $(X) \to$ digit (Y)
- Patient measurements $(X) \to$ medical condition (Y)

Question: Wait... why not just use regression?

Why not simply use regression methods?

Naive attempt:

- Assign $Y \in \{0, 1\}$ or $\{1, \ldots, K\}$ numerically
- Fit a linear model $Y \approx \beta_0 + \beta_1 X$

Issues with the naive attempt:

- For $K \geq 2$, no natural numeric ordering or distance among classes
- Even with $K = 2$, predictions can fall outside $[0, 1]$ if we interpret $\hat{y}$ as probability

We need a method that respects the *categorical* nature of Y and keeps predicted probabilities in $[0, 1]$

Visual illustration: Linear vs. logistic regression

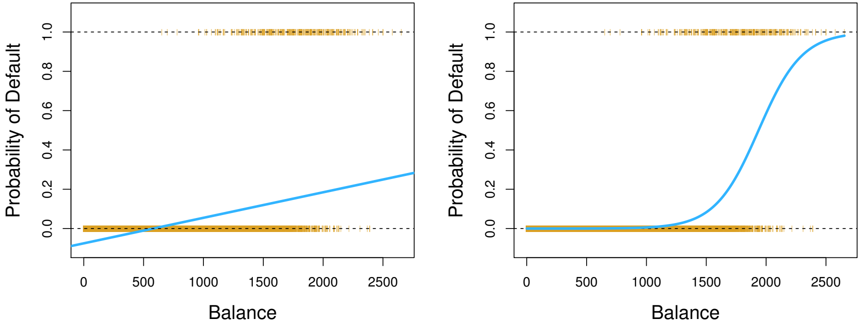


Figure: **Left:** Estimated "probability" of `default` using linear regression. **Right:** Probability estimated via logistic regression [JWHT21, Figure 4.2].

Therefore: Classification-specific methods are typically more appropriate and preferred

Pop-up quiz #1: Classification vs. regression

Question: Which of the following best describes the key difference between *classification* and *regression*?

A) Classification predicts a *continuous* response variable, while regression predicts a *categorical* outcome.

B) Classification deals with *categorical* labels, whereas regression deals with *quantitative* outcomes.

C) Classification uses logistic regression, and regression uses linear regression.

D) Classification cannot produce predictions outside $[0, 1]$, whereas regression can predict any real number.

Roadmap

We will learn two types of classification methods

- **Logistic regression**: a *discriminative* approach that models $P(Y = 1|X)$
- **Generative models**: first model $P(X|Y)$, and then apply Bayes' rule
  - Example: Linear discriminant analysis (LDA), Naive Bayes

Today: Logistic regression with one predictor X ($p = 1$) for binary ($K = 2$) classification

Logistic regression: Basic ideas

For binary classification ($Y \in \{0, 1\}$), let

$$p(x) = \Pr(Y = 1 \mid X = x)$$

- We want a function $f : x \mapsto p(x) \in [0, 1]$
- Then predict

$$Y = \begin{cases} 1 & \text{if } p(x) \geq p^* \text{ (e.g., 0.5),} \\ 0 & \text{otherwise.} \end{cases}$$

How do we get there from linear regression?

- Naive approach: $Y \in \{0, 1\}$; model $Y \approx \beta_0 + \beta_1 X$ can yield $\hat{y} \notin [0, 1]$
- *Odds*: $\frac{p(X)}{1 - p(X)} \in [0, \infty)$
- The *log-odds* (logit): $\log\left(\frac{p(X)}{1 - p(X)}\right) \in (-\infty, \infty)$

Logistic regression: Model formulation

Key idea: Fit a linear model to the *log-odds (logit)*:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

- Positive log-odds $\to p(X) > 0.5$, negative $\to p(X) < 0.5$

Question: How do we write $p(X)$ as a function of β_0, β_1, X?

- Observe that

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \quad \iff \quad \frac{p(X)}{1-p(X)} = \exp(\beta_0 + \beta_1 X)$$

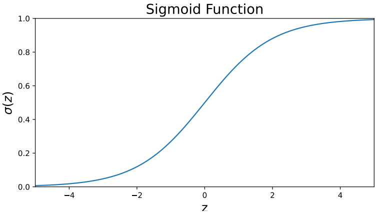
$$\iff \quad p(X) = \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)}$$

Logistic regression model

Logistic regression model:

$$p(X) = \Pr[Y = 1 \mid X] = \sigma(\beta_0 + \beta_1 X) := \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)}$$

- $\sigma : z \mapsto \frac{e^z}{1+e^z}$ is called the logistic function (=sigmoid function)



Interpretation:

- $p(x) \in (0, 1)$ for all x.
- Decision boundary at $\beta_0 + \beta_1 x = 0$, i.e. $p(x) = 0.5$ (or any other threshold p^*)

An example in R: Fraud or not (cont'd)

Scenario: Predict if a transaction is fraud ($Y = 1$) or not ($Y = 0$) using its amount (X)

```
# Simulate toy data:
set.seed(123)
n <- 100
X <- runif(n, 1, 500)  # transaction amount
# true logistic function: p = 1 / [1 + exp(-(-5 + 0.02*X))]
p <- 1 / (1 + exp(-(-5 + 0.02*X)))
Y <- rbinom(n, 1, prob=p)

# Fit logistic regression:
model <- glm(Y ~ X, family=binomial)
summary(model)

# Probability of fraud at X=300:
predict(model, data.frame(X=300), type="response")
```

```
Call:
glm(formula = Y ~ X, family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.084377   1.242746  -4.896 9.79e-07 ***
X            0.024664   0.004923   5.010 5.44e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 138.629  on 99  degrees of freedom
Residual deviance:  53.504  on 98  degrees of freedom
AIC: 57.504

Number of Fisher Scoring iterations: 6

> # Probability of fraud at X=300:
> predict(model, data.frame(X=300), type="response")
        1
0.7883033
```

Check:

- Interpret $\hat{\beta}_0, \hat{\beta}_1$
- $\exp(\hat{\beta}_1)$: how odds change per \$1 increase in the transaction amount

Pop-up quiz #2: Logistic regression coefficients

Scenario: You fit a logistic regression model $\log(\frac{p}{1-p}) = \beta_0 + \beta_1 X$ find $\hat{\beta}_1 = -2.0$

Question: Which of the following interpretations is most accurate?

A) The slope -2.0 is invalid because β_1 must be positive in logistic regression.

B) For each one-unit increase in X, the predicted probability p decreases by 2.

C) For each one-unit increase in X, the *odds* of $Y = 1$ multiply by $e^{-2} \approx 0.14$.

D) If X goes up by 2 units, p becomes exactly zero.

Regression coefficient estimation

Maximum likelihood estimation (MLE):

- Each $Y_i \sim \mathrm{Bernoulli}(p_i)$ where $p_i = \sigma(\beta_0 + \beta_1 x_i)$
- Likelihood function:

$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

- Find $\hat{\beta}_0, \hat{\beta}_1$ that *maximizes* $L(\beta_0, \beta_1)$ (for the given data)

Why not just do non-linear least squares?

- Minimizing $\sum(y_i - p_i)^2$ is feasible, but not consistent with Bernoulli nature of Y
- MLE aligns with the data's distribution, yielding favorable statistical properties

Wrap-up

Today's takeaways:

- *Classification* vs. regression: fundamental differences in Y
- Linear regression on $\{0, 1\}$ can be problematic for classification
- *Logistic regression* models the log-odds as $\beta_0 + \beta_1 x$, ensuring $p \in (0, 1)$
- Parameter estimation via *maximum likelihood* criterion

Next lecture:

- Extending logistic regression to multiple predictors ($p > 1$) & multi-class ($K > 2$)
- Generative models for classification

References

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani.
An Introduction to Statistical Learning: with Applications in R, volume 112 of *Springer Texts in Statistics*.
Springer, New York, NY, 2nd edition, 2021.