# STA 314: Statistical Methods for Machine Learning I

## Lecture 7 - Logistic regression, metrics for binary classification

Xin Bing

Department of Statistical Sciences
University of Toronto

# Review

- In classification, $X \in \mathbb{R}^p$ and $Y \in C = \{0, 1, \ldots, K-1\}$.
- The Bayes rule

$$\arg\max_{k \in C} \mathbb{P}\{Y = k \mid X = x\}, \qquad \forall x \in \mathbb{R}^p$$

  has the smallest expected error rate.

- For binary classification, our goal is to estimate

$$p(x) = \mathbb{P}\{Y = 1 \mid X = x\}, \qquad \forall x \in \mathbb{R}^p.$$

# Logistic Regression

Logistic Regression is a parametric approach that assumes parametric structure on

$$p(X) = \mathbb{P}(Y = 1 \mid X).$$

- It assumes

  $$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

  The function $f(t) = e^t/(1+e^t)$ is called the logistic function. $\beta_0, \ldots, \beta_p$ are the parameters.

- It is easy to see that we always have $0 \leq p(X) \leq 1$.

- Note that $p(X)$ is **NOT** a linear function either in $X$ or in $\beta$.

# Logistic Regression

- A bit of rearrangement gives

$$\underbrace{\frac{p(X)}{1-p(X)}}_{\text{odds}} = e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p},$$

$$\underbrace{\log\left[\frac{p(X)}{1-p(X)}\right]}_{\text{log-odds (a.k.a. logit)}} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

odds $\in [0, \infty)$ and log-odds $\in (-\infty, \infty)$.

- Similar interpretation as linear models.

- How to estimate $\beta_0, \dots, \beta_p$?

# Maximum Likelihood Estimator (MLE)

Given $\mathcal{D}^{train} = \{(x_1, y_1), ..., (x_n, y_n)\}$ with $y_i \in \{0, 1\}$, we estimate the parameters by **maximizing the likelihood** of $\mathcal{D}^{train}$.

## The maximum likelihood principle

The maximum likelihood principle is that we seek the estimates of parameters such that the fitted probability are the closest to the individual's observed outcome.

# Cont'd: MLE under logistic regression

General steps of computing the MLE:

- Write down the likelihood, as always!
- Solve the optimization (maximization) problem.

# Likelihood under Logistic Regression

For simplicity, let us set $\beta_0 = 0$ such that

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

The data consists of $(x_1, y_1), \ldots, (x_n, y_n)$ with

$$y_i \sim \text{Bernoulli}(p(x_i)), \qquad p(x_i) = \frac{e^{x_i^\top \boldsymbol{\beta}}}{1 + e^{x_i^\top \boldsymbol{\beta}}}, \quad 1 \le i \le n.$$

- What is the likelihood of $y_i$?

# Likelihood under Logistic Regression

The likelihood of each data point $(x_i, y_i)$ at any $\boldsymbol{\beta}$ is

$$L_i(\boldsymbol{\beta}) = [p(x_i)]^{y_i}[1 - p(x_i)]^{1-y_i}$$

with

$$p(x_i) = \frac{e^{x_i^\top \boldsymbol{\beta}}}{1 + e^{x_i^\top \boldsymbol{\beta}}}.$$

The joint likelihood of all data points is

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} [p(x_i)]^{y_i}[1 - p(x_i)]^{1-y_i}.$$

# Log-likelihood under Logistic Regression

The log-likelihood at any $\boldsymbol{\beta}$ is

$$\begin{aligned}
\ell(\boldsymbol{\beta}) &= \log\left\{\prod_{i=1}^{n}\left[p(x_i)\right]^{y_i}\left[1-p(x_i)\right]^{1-y_i}\right\} \\
&= \sum_{i=1}^{n}\left[y_i\log(p(x_i)) + (1-y_i)\log(1-p(x_i))\right] \\
&= \sum_{i=1}^{n}\left[y_i\log\left(\frac{p(x_i)}{1-p(x_i)}\right) + \log(1-p(x_i))\right] \\
&= \sum_{i=1}^{n}\left[y_i x_i^\top\boldsymbol{\beta} - \log\left(1+e^{x_i^\top\boldsymbol{\beta}}\right)\right].
\end{aligned}$$

# How to compute the MLE?

How do we maximize the log-likelihood

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[ y_i x_i^\top \boldsymbol{\beta} - \log\left(1 + e^{x_i^\top \boldsymbol{\beta}}\right)\right]$$

for logistic regression?

- No direct solution: taking derivatives of $\ell(\boldsymbol{\beta})$ w.r.t. $\boldsymbol{\beta}$ and setting them to 0 doesn't have an explicit solution.
- Need to use iterative procedure, later...

# Cont'd: MLE under logistic regression

Let

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[ y_i x_i^\top \boldsymbol{\beta} - \log\left(1 + e^{x_i^\top \boldsymbol{\beta}}\right) \right]$$

The estimator $\hat{\boldsymbol{\beta}}$ is called **the Maximum Likelihood Estimator** (MLE).

The MLE has many nice properties!

- Asymp consistent.
- Asymp normal.
- And more......

# Inference under logistic regression

- Z-statistic is similar to t-statistic in regression, and is defined as

$$\frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}, \qquad \forall j \in \{0, 1, \ldots, p\}.$$

- It produces p-value for testing the null hypothesis

$$H_0 : \beta_j = 0 \quad \text{v.s.} \quad H_1 : \beta_j \neq 0.$$

A large (absolute) value of the z-statistic or small p-value indicates evidence against $H_0$.

# Example: Default data

Consider the Default data using balance, income, and student status as predictors.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

| | Coefficient | Std. Error | Z-statistic | P-value |
|---|---|---|---|---|
| `Intercept` | -10.8690 | 0.4923 | -22.08 | $< 0.0001$ |
| `balance` | 0.0057 | 0.0002 | 24.74 | $< 0.0001$ |
| `income` | 0.0030 | 0.0082 | 0.37 | 0.7115 |
| `student[Yes]` | -0.6468 | 0.2362 | -2.74 | 0.0062 |

# Prediction at different levels under logistic regression

Let $\hat{\beta}_0, \dots, \hat{\beta}_p$ be the MLE.

- Prediction of the logit at $x \in \mathbb{R}^p$:

$$\widehat{\text{logit}}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p.$$

- Prediction of $\mathbb{P}(Y = 1 \mid X = x)$:

$$\hat{\mathbb{P}}(Y = 1 \mid X = x) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}$$

- Prediction of $Y$ (i.e. *classification*) at $X = x$:

$$\hat{y} = \begin{cases} 1, & \text{if} \quad \hat{\mathbb{P}}(Y = 1 \mid X = x) \geq 0.5; \\ 0, & \text{otherwise.} \end{cases}$$

# Prediction of $\mathbb{P}(Y = 1 \mid X)$

Consider the Default data with student status as the only feature. What is our estimated probability of default for a student?

To fit the model, we encode student status as 1 for student and 0 otherwise.

| | Coefficient | Std. Error | Z-statistic | P-value |
|---|---|---|---|---|
| Intercept | -3.5041 | 0.0707 | -49.55 | $< 0.0001$ |
| student[Yes] | 0.4049 | 0.1150 | 3.52 | 0.0004 |

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

# Metrics used for evaluating classifiers

In classification, we have several metrics that can be used to evaluate a given classifier.

- The most commonly used metric is the overall classification accuracy.
- For binary classification, there are a few more out there.....

# Logistic Regression on the Default Data

Classify whether or not an individual will default on the basis of credit card balance and student status. The confusion matrix on default data.

| | | *True default status* | | |
|---|---|---|---|---|
| | | No | Yes | Total |
| *Predicted* | No | 9,644 | 252 | 9,896 |
| *default status* | Yes | 23 | 81 | 104 |
| | Total | 9,667 | 333 | 10,000 |

- The training error rate is $(23 + 252)/10000 = 2.75\%$.
- **False positive rate (FPR)**: The fraction of negative examples that are classified as positive: $23/9667 = 0.2\%$ in default data.
- **False negative rate (FNR)**: The fraction of positive examples that are classified as negative: 75.7% in default data.
- For a credit card company that is trying to identify high-risk individuals, an error rate of $252/333 = 75.7\%$ among individuals who default is unacceptable.

# Types of Errors for binary classification

- The false negative rate is too high. How can we modify the LDA rule to lower the FNR?

- The current classifier is based on the rule

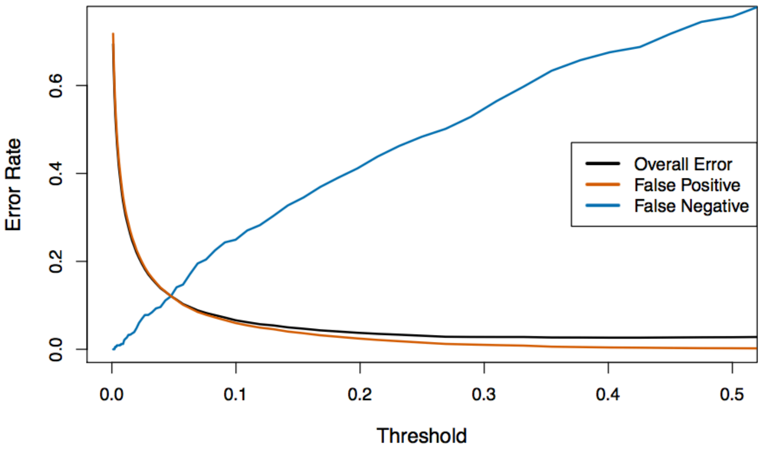
$$\mathbb{P}(\text{default} = \textit{yes} \mid X = x) \geq 0.5.$$

- We can achieve better balance between FPR and FNR by varying the threshold:
  - To lower FNR, we reduce the number of negative predictions. Classify $X = x$ to *yes* if

$$\mathbb{P}\left(Y = \textit{yes} \mid X = x\right) \geq \text{thresh}.$$

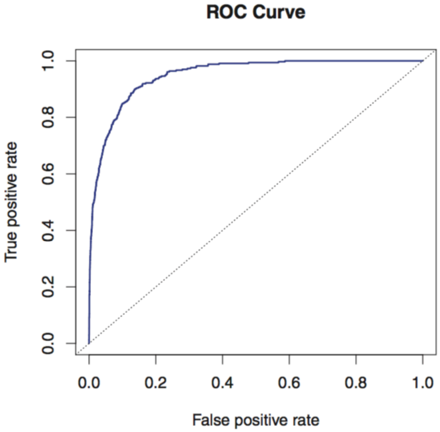
    for some thresh $< 0.5$.

# Trade-off between FPR and FNR



# ROC Curve

The ROC curve is a popular graphic for simultaneously displaying FPR and TPR for all possible thresholds.




The overall performance of a classifier, summarized over all thresholds, is given by the area under the curve (AUC). High AUC is good.

# More metrics in the binary classification

| | | *Predicted class* | | |
|---|---|---|---|---|
| | | − or Null | + or Non-null | Total |
| *True* | − or Null | True Neg. (TN) | False Pos. (FP) | N |
| *class* | + or Non-null | False Neg. (FN) | True Pos. (TP) | P |
| | Total | $N^*$ | $P^*$ | |

| Name | Definition | Synonyms |
|---|---|---|
| False Pos. rate | FP/N | Type I error, 1−Specificity |
| True Pos. rate | TP/P | 1−Type II error, power, sensitivity, recall |
| Pos. Pred. value | $TP/P^*$ | Precision, 1−false discovery proportion |
| Neg. Pred. value | $TN/N^*$ | |

The above also defines **sensitivity** and **specificity**.