

# STA 314: Statistical Methods for Machine Learning I

## Lecture - Logistic Regression in Binary Classification

Xin Bing

Department of Statistical Sciences  
University of Toronto

- In classification,  $X \in \mathcal{X}$  and  $Y \in C = \{0, 1, \dots, K - 1\}$ .
- The Bayes rule

$$f^*(\mathbf{x}) = \arg \max_{k \in C} \mathbb{P}\{Y = k \mid X = \mathbf{x}\}, \quad \forall \mathbf{x} \in \mathcal{X}$$

has the smallest expected error rate.

- For binary classification, our goal is to estimate

$$\begin{aligned} p(\mathbf{x}) &:= \mathbb{P}(Y = 1 \mid X = \mathbf{x}) \\ &= \mathbb{E}[\mathbf{Y} \mid \mathbf{X} = \mathbf{x}], \quad \forall \mathbf{x} \in \mathcal{X} \end{aligned}$$

# Logistic Regression

Logistic Regression is a parametric approach that postulates parametric structure on the function  $p : \mathcal{X} \mapsto [0, 1]$ .

- It is assumed that

$$p(\mathbf{x}) := p(\mathbf{x}; \boldsymbol{\beta}) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}, \quad \forall \mathbf{x} \in \mathcal{X}.$$

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

- We always have  $0 \leq p(\mathbf{x}) \leq 1$ .
- Note that  $p(\mathbf{x}; \boldsymbol{\beta})$  is **NOT** a linear function either in  $\mathbf{x}$  or in  $\boldsymbol{\beta}$ .

# Logistic Regression

- A bit of rearrangement gives

$$\underbrace{\frac{p(\mathbf{x})}{1-p(\mathbf{x})}}_{\text{odds}} = e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p},$$
$$\underbrace{\log \left[ \frac{p(\mathbf{x})}{1-p(\mathbf{x})} \right]}_{\text{log-odds (a.k.a. logit)}} = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p.$$

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

- Similar interpretation as linear models<sup>1</sup>

---

<sup>1</sup>Each  $\beta_j$  represents the change of log-odds for one unit increase in  $X_j$  (with other features held fixed).

Our interests:

- **Prediction**: for any  $\mathbf{x}_0 \in \mathcal{X}$ , classify its corresponding label  $y_0$ .
- **Estimation**: how to estimate the vector of  $\beta$  by using our training data?

# Prediction at **different levels** under logistic regression

Let  $\hat{\beta} = (\hat{\beta}_0, \dots, \hat{\beta}_p)$  be any estimates of  $\beta$ .

- Prediction of **the logit** at  $\mathbf{x} \in \mathcal{X}$ :

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

- Prediction of **the conditional probability**  $p(\mathbf{x}) = \mathbb{P}(Y = 1 | X = \mathbf{x})$ :

$$\hat{p}(\mathbf{x}) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_p x_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_p x_p}}$$

- Classify **the label**  $Y$  at  $X = \mathbf{x}$ :

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

# Maximum Likelihood Estimator (MLE)

Given  $\mathcal{D}^{train} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$  with  $y_i \in \{0, 1\}$ , we estimate the parameters by **maximizing the likelihood** of  $\mathcal{D}^{train}$ .

## The maximum likelihood principle

We seek the estimates of parameters such that the fitted probability are the closest to the individual's observed outcome.

# Computation of the MLE under Logistic Regression

General steps of computing the MLE:

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



# Likelihood under Logistic Regression

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

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

The data consists of  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$  with

$$y_i \sim \text{Bernoulli}(p(\mathbf{x}_i)), \quad p(\mathbf{x}_i) = \frac{e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}, \quad 1 \leq i \leq n.$$

- What is the likelihood of  $y_i$ ?

# Likelihood under Logistic Regression

The likelihood of each data point  $(\mathbf{x}_i, y_i)$  at any  $\beta$  is

$$L(\beta; \mathbf{x}_i, y_i) = [p(\mathbf{x}_i)]^{y_i} [1 - p(\mathbf{x}_i)]^{1-y_i}$$

with

$$p(\mathbf{x}_i) = \frac{e^{\mathbf{x}_i^\top \beta}}{1 + e^{\mathbf{x}_i^\top \beta}}.$$

The joint likelihood of all data points is

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

# Log-likelihood under Logistic Regression

The log-likelihood at any  $\beta$  is

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

# How to compute the MLE?

How do we maximize the log-likelihood

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

for logistic regression?

- It is equivalent to minimize  $-\ell(\beta)$  over  $\beta$ .
- No direct solution: taking derivatives of  $\ell(\beta)$  w.r.t.  $\beta$  and setting them to 0 doesn't have an explicit solution.
- Need to use iterative procedure.

# Why MLE?

The MLE, whenever they can be computed, has many nice properties!

- Asymp. consistent

$$\hat{\beta} - \beta \rightarrow 0, \quad \text{in probability as } n \rightarrow \infty.$$

- Asymp. normal

$$\sqrt{n}(\hat{\beta} - \beta) \rightarrow N(0, \Sigma) \quad \text{in distribution as } n \rightarrow \infty.$$

- Asymp. efficient:

$\Sigma$  is the “smallest” among all asymptotic unbiased estimators.

**Any downsides?** computation, model misspecification ...

# Inference under logistic regression

Let  $\hat{\beta}$  be the MLE of  $\beta$ .

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

$$\frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}, \quad \forall j \in \{0, 1, \dots, p\}$$

where  $SE(\hat{\beta}_j)$  is the asymp. variance of  $\hat{\beta}_j$  (equal to  $\hat{\Sigma}_{jj}/n$  in the previous slide).

- 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

Suppose that we are interested in predicting

*the probability of default for a given customer*

by using **student status** as the only feature.

By encoding  $x_i = 1$  {the  $i$ th customer is student} and,  $y_i = 1$  if default happens and 0 otherwise. Fit the logistic regression model

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

# Prediction of $p(x)$

$$p(x) = \mathbb{P}(Y = 1 \mid X = x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}.$$

The fitted maximum likelihood estimates of  $\beta_0$  and  $\beta_1$  satisfy:

	Coefficient	Std.Error	Z-statistic	P-value
Intercept	-3.5	0.071	-49.55	<0.0001
student[Yes]	0.405	0.115	3.52	0.0004

$$\hat{p}(x = 1) = \hat{\mathbb{P}}(\text{default} \mid \text{student}) = \frac{e^{-3.5 + 0.071 \times 1}}{1 + e^{-3.5 + 0.071 \times 1}} \approx 0.043$$

$$\hat{p}(x = 0) = \hat{\mathbb{P}}(\text{default} \mid \text{non-student}) = \frac{e^{-3.5 + 0.071 \times 1}}{1 + e^{-3.5 + 0.071 \times 1}} \approx 0.029$$



## Example: Default data

Consider using more predictors: **balance**( $X_1$ ), **income**( $X_2$ ), and **student status**( $X_3$ ).

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

The maximum likelihood estimates yield:

	Coefficient	Std.Error	Z-statistic	P-value
Intercept	-10.87	0.492	-22.08	<0.0001
balance	0.006	0.0002	24.74	<0.0001
income	0.003	0.0082	0.37	0.712
student[Yes]	-0.647	0.2362	-2.74	0.0062

*Question:* how does the coefficient of student status changes?

# 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.....

## Cont'd example: the Default Data

- Classify whether or not an individual will default on the basis of credit card balance and student status.
- The confusion matrix** of fitted logistic regression

		<i>True default status</i>		
		No	Yes	Total
<i>Predicted default status</i>	No	9,644	252	9,896
	Yes	23	81	104
Total		9,667	333	10,000

- The training error rate is  $(23 + 252)/10000 = 2.75\%$ .

# Type of Errors for binary classification

		<i>True default status</i>		
		No	Yes	Total
<i>Predicted default status</i>	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

1. **False positive rate (FPR)**: The fraction of negative examples that are classified as positive:  $23/9667 = 0.2\%$  in default data.
2. **False negative rate (FNR)**: The fraction of positive examples that are classified as negative:  $252/333 = 75.7\%$  in default data.<sup>2</sup>

---

<sup>2</sup>For a credit card company that is trying to identify high-risk individuals, the error rate 75.7% among individuals who default is unacceptable.

# Control the false negative rate

**Q:** How to modify the logistic classifier to lower the false negative rate?

the fraction of **positive** examples as **negative**

the fraction of **default** examples classified as **non-default**

- The current classifier is based on the rule

$$\begin{aligned} \hat{y}_i &= 1 \quad (\text{default}), & \text{if } \hat{\mathbb{P}}(\text{default} = \text{yes} \mid X = \mathbf{x}_i) \geq 0.5 \\ \hat{y}_i &= 0 \quad (\text{non-default}), & \text{otherwise.} \end{aligned}$$

# Control the false negative rate

- To lower FNR, we reduce the number of negative predictions.  
Classify  $X = \mathbf{x}$  to yes if

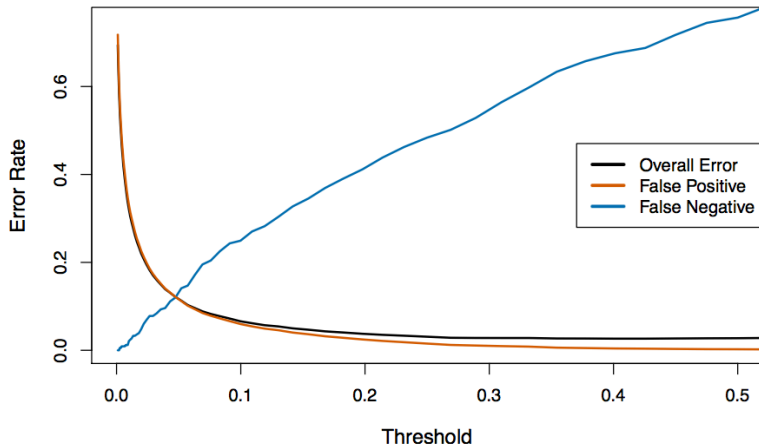
$$\hat{\mathbb{P}}(Y = \text{yes} \mid X = \mathbf{x}) \geq t.$$

for some  $0 \leq t < 0.5$ .

- ▶ Why starts with  $t = 0.5$ ?
- ▶ What happens for  $t = 0$ ?
- ▶ What happens for  $t = 1$ ?

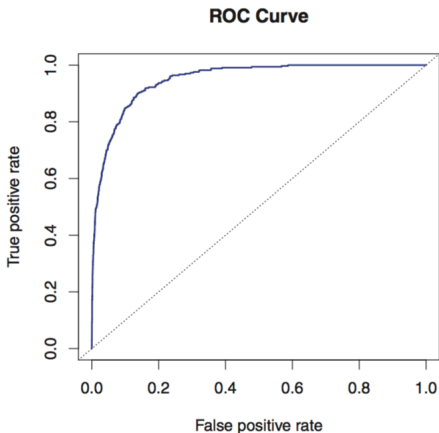
# Trade-off between FPR and FNR

We can achieve better balance between FPR and FNR by varying the threshold  $t$ :



# ROC Curve

The **ROC curve** is a popular graphic for simultaneously displaying FPR and  $\text{TPR} = 1 - \text{FNR}$  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

		<i>Predicted class</i>		
		– or Null	+ or Non-null	Total
<i>True class</i>	– or Null	True Neg. (TN)	False Pos. (FP)	N
	+ 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**.