

# Logistic Regression

Semester 1, 2021

Ling Luo

# Back to Classification

- In linear regression, we have *continuous features* and a *continuous output variable*
- In classification, we have *continuous/discrete features*, and *discrete output variable*
- Logistic regression?

*classification*

# Regression → Classification

- Translate a regression task into a simple classification task
  - Discretisation: map continuous values onto discrete labels, by setting range of continuous variable that corresponds to each discrete class
  - e.g. predict age group 11-20, 21-30, 31-40 ... instead of the value of age

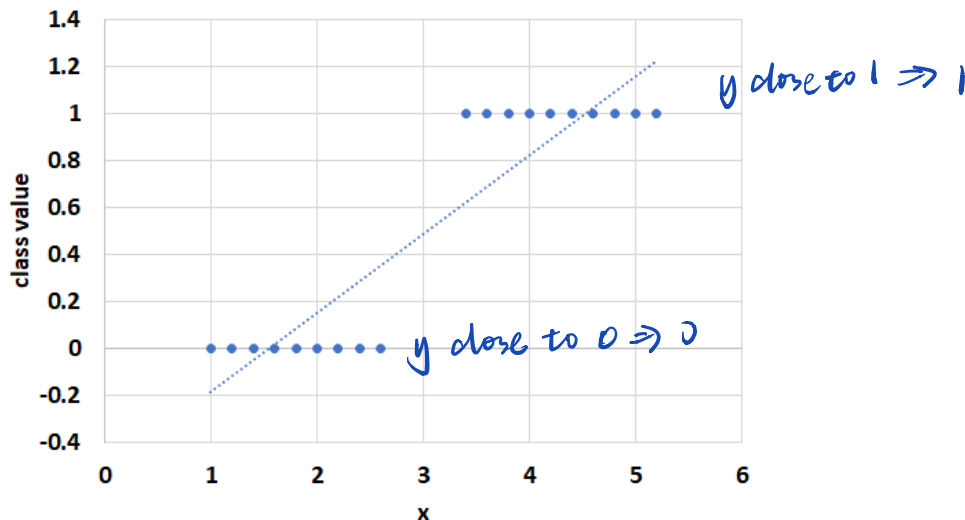
*regression → classification*

*\* map continuous value onto discrete labels.*

# Regression ← Classification

- Translate a classification task into a regression task
  - Binary-class: take class labels 0 and 1 as numeric values
  - Multi-class: build a set of regression tasks
  - Approximates a *numeric membership* for each class

*classification → regression*



# Probabilistic Classification

- Naïve Bayes

$$\hat{c} = \arg \max_{c_j \in \mathcal{C}} P(c_j | \mathbf{x})$$

- We apply Bayes' rule and independent assumption

$$\begin{aligned}\hat{c} &= \arg \max_{c_j \in \mathcal{C}} P(c_j | \mathbf{x}) \\ &= \arg \max_{c_j \in \mathcal{C}} P(\mathbf{x} | c_j) P(c_j) \\ &\approx \arg \max_{c_j \in \mathcal{C}} P(c_j) \prod_k P(x_k | c_j)\end{aligned}$$

$\mathbf{x} = [x_1, x_2, \dots, x_D]$  is one instance with  $D$  attributes

# Probabilistic Classification

- Strategy: attempt to model  $P(c|\mathbf{x})$  directly
- Assuming a 2-class problem (Y/N)
  - $P(c = Y|\mathbf{x})$ : probability of an instance with class Y
  - For an instance with class = Y:  $P(c = Y|\mathbf{x}) = 1$
  - For an instance with class = N:  $P(c = Y|\mathbf{x}) = 0$
- If the numerical attributes of the instance are predictors,  $P(c = Y|\mathbf{x})$  is the target variable:  
*construct a regression model (linear regression)*

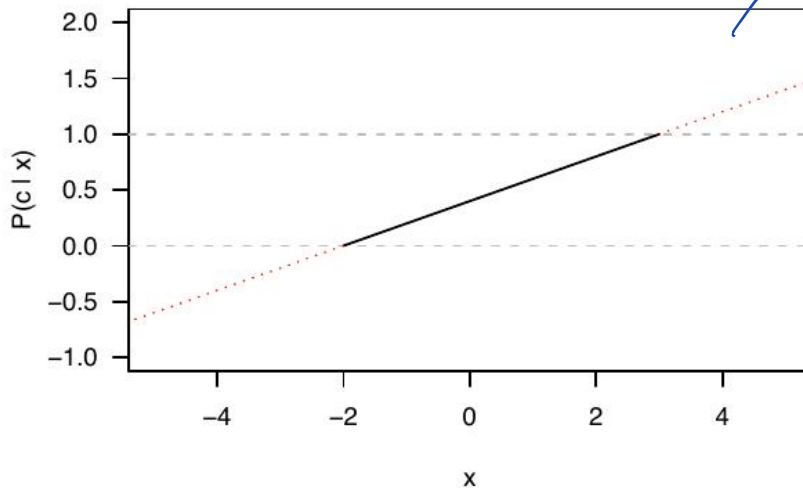
$$\begin{aligned} P(c = Y|\mathbf{x}) &= \boldsymbol{\beta} \cdot \mathbf{x} \\ &= \beta_0 + \beta_1 x_1 + \dots + \beta_D x_D \end{aligned}$$

*$\mathbf{x} = [1, x_1, \dots, x_D]$   
 $\boldsymbol{\beta} = [\beta_0, \beta_1, \dots, \beta_D]$*

# Probabilistic Classification

$$P(c = Y|x) = \beta \cdot x$$

problem: not bounded  
to  $[0,1]$



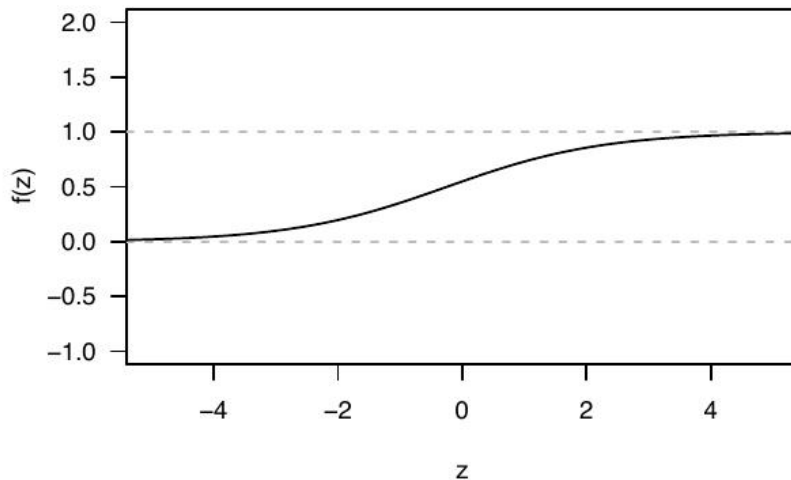
- Linear function, no guarantee that  $\hat{P} \in [0,1]$
- Meaning of probabilities outside  $[0,1]$  is unclear

# Probabilistic Classification

$$P(c = Y|\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{x}}}$$

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

- Logistic function
- $\hat{P} \in [0,1]$  for all  $z$  values  
( $z = \boldsymbol{\beta} \cdot \mathbf{x}$ )





# Log-Linear Model

- Derivation of logistic function
- Consider the following multiplicative formulation

$$P(c|\mathbf{x}) = \gamma_0 \cdot \gamma_1^{x_1} \cdot \dots \cdot \gamma_D^{x_D}$$

$$\log P(c|\mathbf{x}) = \log(\gamma_0 \cdot \gamma_1^{x_1} \cdot \dots \cdot \gamma_D^{x_D})$$

$$\log P(c|\mathbf{x}) = \log \gamma_0 + x_1 \log \gamma_1 + \dots + x_D \log \gamma_D$$



$$\beta_0$$



$$\beta_1$$



$$\beta_D$$

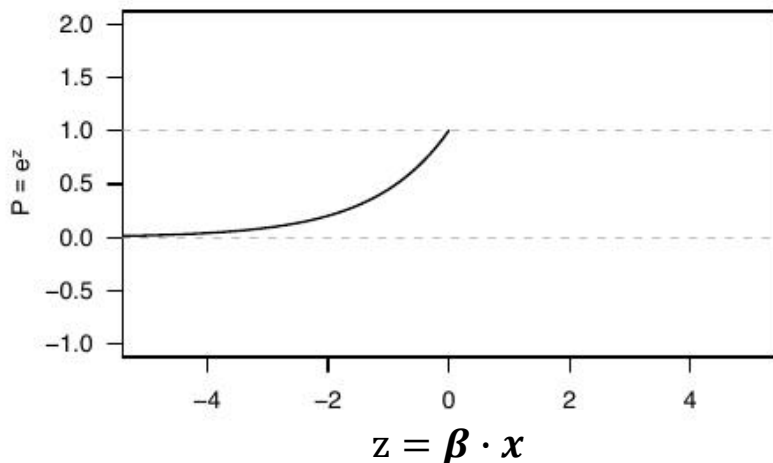
$$\log P(c|\mathbf{x}) = \beta_0 + \beta_1 x_1 + \dots + \beta_D x_D$$

*natural log*

# Log-Linear Model

$$\log P(c|\mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{x}$$

$$P(c|\mathbf{x}) = e^{\boldsymbol{\beta} \cdot \mathbf{x}}$$



- Log-linear function
- Curve is unbalanced
  - Fine <sup>精度</sup>granularity of response as  $P \rightarrow 0$
  - Coarse response as  $P \rightarrow 1$
  - $P > 1$  when  $\boldsymbol{\beta} \cdot \mathbf{x} > 0$

# Balanced P

- Want same response behaviour for high and low  $P$
- Solution: **logit** (i.e. log odds)

$$\text{logit}(P) = \log \frac{P}{1-P}$$

$$\text{logit}(1-P) = \log \frac{1-P}{P} = -\text{logit}(P)$$

$\text{logit } P = \log \frac{P}{1-P} = \beta x$

$\frac{P}{1-P} = e^{\beta x}$

$\frac{P-1+1}{1-P} = e^{\beta x}$

$-1 + \frac{1}{1-P} = e^{\beta x}$

$\frac{1}{1-P} = 1 + e^{\beta x}$

$1-P = \frac{1}{1+e^{\beta x}}$

$P = 1 - \frac{1}{1+e^{\beta x}} = \frac{e^{\beta x}}{1+e^{\beta x}}$

$\frac{1}{1+e^{\beta x}} = \frac{1}{1+e^{\beta x}}$

$\frac{1}{1+e^{\beta x}} = \frac{1}{1+e^{\beta x}}$

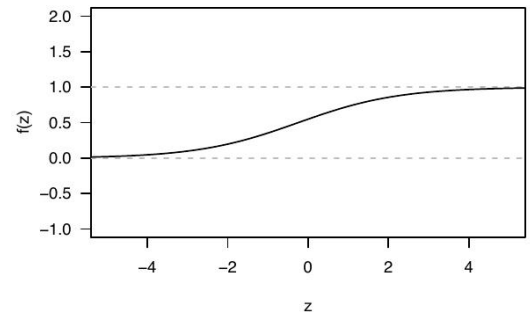
# Logistic Regression

- Use linear regression to model logit  $P$

$$\text{logit } P(c|\mathbf{x}) = \log \frac{P(c|\mathbf{x})}{1 - P(c|\mathbf{x})} = \boldsymbol{\beta} \cdot \mathbf{x}$$

$$P(c|\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{x}}}$$

logistic function



# Training and Prediction

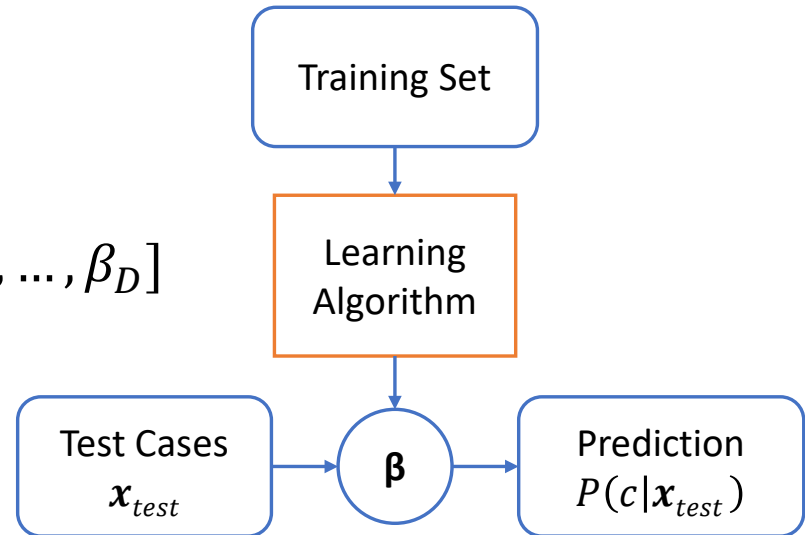
Training set:  $N$  examples

$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N),$   
 $\mathbf{x}_i = [x_{i0}, x_{i1}, x_{i2}, \dots, x_{iD}]$

Find the optimal  $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \dots, \beta_D]$

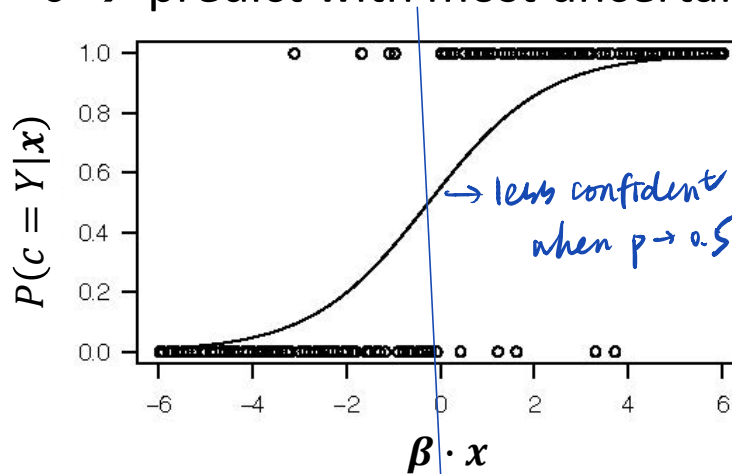
Predict a continuous valued output

$$P(c|\mathbf{x}_{test}) = \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{x}_{test}}}$$



# Training and Prediction

- Most values of  $\beta \cdot x$  lead to  $P \approx 1$  or  $P \approx 0$
- Assuming a 2-class problem (Y/N) *p → 1 / 0. more confident*
  - $\beta \cdot x > 0 \rightarrow \hat{P}(c = Y|x) > 0.5 \rightarrow$  predict **class = Y**
  - $\beta \cdot x < 0 \rightarrow \hat{P}(c = Y|x) < 0.5 \rightarrow$  predict **class = N**
  - $\beta \cdot x \approx 0 \rightarrow$  predict with most uncertainty



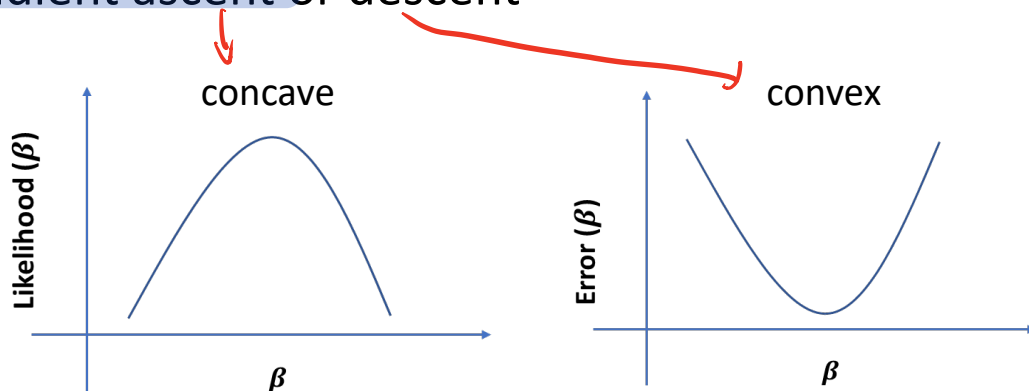
*also indicate confidence*

# Parameter Estimation

- How do we determine  $\beta$ ?
  - We need to define an objective function to optimise.  
Maximise accuracy/likelihood or equivalently minimise error

$$\hat{\beta} = \arg \max_{\beta} \text{Likelihood}(\beta; L, F(T))$$

- Gradient ascent or descent



# Parameter Estimation

- Assuming a 2-class problem (1/0), our aim is to choose  $\beta$ 
  - $\hat{P}(y = 1|x)$  is close to 1, when instance  $x$  has class label 1
  - $\hat{P}(y = 1|x)$  is close to 0, when instance  $x$  has class label 0
- The likelihood for one instance

when  $y=1$ ,  $\max h_p(x)$   
 $y=0$   $\max (1-h_p(x))$

$$P(y = 1|x, \beta) = h_\beta(x) = \frac{1}{1 + e^{-\beta \cdot x}}$$

$$P(y = 0|x, \beta) = 1 - h_\beta(x) = \frac{e^{-\beta \cdot x}}{1 + e^{-\beta \cdot x}}$$

$$\Rightarrow P(y|x, \beta) = (h_\beta(x))^y (1 - h_\beta(x))^{1-y}$$



# Parameter Estimation

- Given  $N$  independent training instances  $\{\mathbf{X}, Y\}$ , we want to choose  $\boldsymbol{\beta}$  to *maximise the likelihood* of observing them

$$\begin{aligned} P(Y|\mathbf{X}, \boldsymbol{\beta}) &= \prod_{i=1}^N P(y_i | \mathbf{x}_i, \boldsymbol{\beta}) \\ &= \prod_{i=1}^N (h_{\boldsymbol{\beta}}(\mathbf{x}_i))^{y_i} (1 - h_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i} \end{aligned}$$

$$\hat{\boldsymbol{\beta}} = \arg \max_{\boldsymbol{\beta}} \prod_{i=1}^N (h_{\boldsymbol{\beta}}(\mathbf{x}_i))^{y_i} (1 - h_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

# Parameter Estimation

- For simplicity, maximise log-likelihood

$$\log P(Y|\mathbf{X}, \boldsymbol{\beta}) = \sum_{i=1}^N y_i \log h_{\boldsymbol{\beta}}(\mathbf{x}_i) + (1 - y_i) \log(1 - h_{\boldsymbol{\beta}}(\mathbf{x}_i))$$

- This is a concave function with regard to  $\beta_k$
- Preferred strategy for choosing  $\boldsymbol{\beta}$  is gradient ascent

In iteration ( $iter + 1$ ) update  $\beta_k$  as:

$$\beta_k^{iter+1} := \beta_k^{iter} + \alpha \frac{\partial \log P(Y|\mathbf{X}, \boldsymbol{\beta}^{iter})}{\partial \beta_k^{iter}}$$

for logistic function  $\rightarrow$  concave  
always have maximum

# Multi-class Classification

*standard logistic regression: binary class*

## Multinomial logistic regression

- Take one class as **pivot**
- For every other class  $c_j$ , build a regression model

*one-vs-rest*

*$C-1$  models.*

Treat class  $c_j$  as class Y, and the pivot class as class N

- End up with  $(|C| - 1)$  different logistic regression models
- Predict the class label according to the one with the highest probability score

# Multi-class Classification

The probability distribution for each instance sums to 1

- For  $|C| - 1$  non-pivot class  $c_j$   
*assume: change pivot class won't affect results.*

$$P(y = c_j | x, \beta) = \frac{e^{\beta^{(c_j)} \cdot x}}{1 + \sum_{c=1}^{|C|-1} e^{\beta^{(c)} \cdot x}}$$

- For pivot class

$$P(y = \text{pivot} | x, \beta) = \frac{1}{1 + \sum_{c=1}^{|C|-1} e^{\beta^{(c)} \cdot x}}$$

3 classes

$c_1, c_2, c_3$

↑  
pivot

2 model


(1)

$$\log \frac{P(y=c_1|x)}{P(y=c_3|x)} = \beta^{(1)} \cdot x$$

(2)

$$\log \frac{P(y=c_2|x)}{P(y=c_3|x)} = \beta^{(2)} \cdot x$$

$$P(y=c_1|x) + P(y=c_2|x) + P(y=c_3|x) = 1$$



$$e^{\beta_{(1)}x} \cdot P(Y=c_2|x) + e^{\beta_{(2)}x} P(Y=c_3|x) + P(Y=c_1|x) = 1$$

$$\Rightarrow P(Y=c_2|x) = \frac{1}{1 + e^{\beta_{(1)}x} + e^{\beta_{(2)}x}}$$

$$P(Y=c_2|x) = \frac{e^{\beta_{(2)}x}}{1 + e^{\beta_{(1)}x} + e^{\beta_{(2)}x}}$$

$$P(Y=c_1|x) = \frac{e^{\beta_{(1)}x}}{1 + e^{\beta_{(1)}x} + e^{\beta_{(2)}x}}$$

# Discussion

## Improvement on Naïve Bayes

- Naïve Bayes: model  $P(c_j|\mathbf{x})$  with Bayes' rule and independent assumption

$$P(c_j|\mathbf{x}) \approx P(c_j) \prod_k P(x_k|c_j)$$

- Logistic regression: model  $P(c_j|\mathbf{x})$  directly using regression (subject to parameter  $\boldsymbol{\beta}$ ), no need to estimate  $P(\mathbf{x}|c_j)$

$$P(c_j|\mathbf{x}, \boldsymbol{\beta}) = \text{logistic}(\boldsymbol{\beta} \cdot \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta} \cdot \mathbf{x}}}$$

*without  
independent assumption*

# Discussion

## Some other points about logistic regression

- Forms a linear decision boundary
- Provides probabilities of the classification result
- Easy to interpret using the learnt parameters

linear decision boundary  $\xrightarrow[\text{mapping}]{\text{logistic function}}$  probability

non-linear :  $\begin{matrix} \text{introduce} \\ \vee \end{matrix}$  polynomial term  $x = [x^1, x^2, x^3, \dots]$

# Summary

- How is logistic regression related to regression?
- How to estimate parameters  $\beta$  by maximising log-likelihood?
- How can logistic regression be extended to multi-class classification?