Logistic Regression

Semester 1, 2021

Ling Luo

Back to Classification

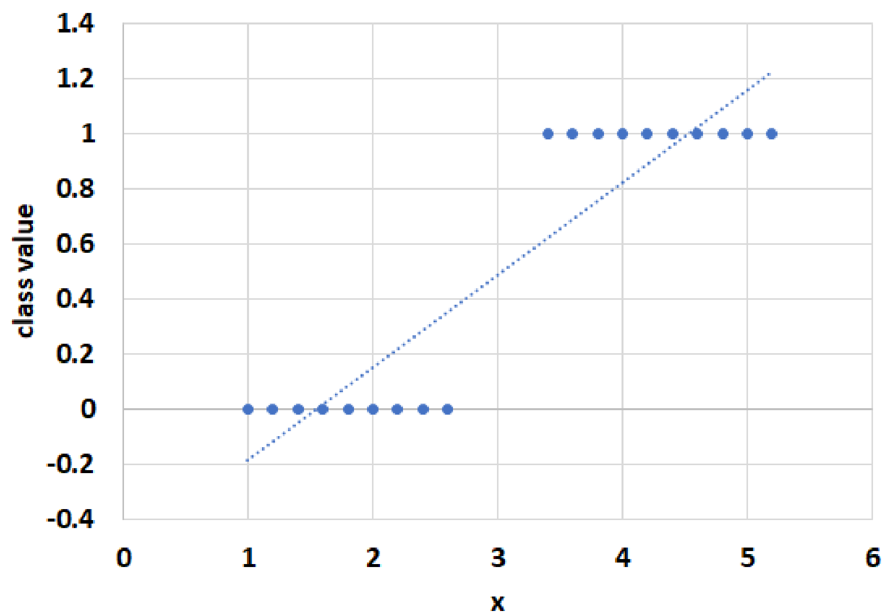
- In linear regression, we have *continuous features* and a *continuous output variable*
- In classification, we have *continuous/discrete features*, and *discrete output variable*
- Logistic regression?

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

- 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



Probabilistic Classification

- Naïve Bayes

$$\hat{c} = \underset{c_j \in C}{\arg\max}\, P(c_j|\boldsymbol{x})$$

- We apply Bayes' rule and independent assumption

$$\begin{aligned}\hat{c} &= \underset{c_j \in C}{\arg\max}\, P(c_j|\boldsymbol{x}) \\ &= \underset{c_j \in C}{\arg\max}\, P(\boldsymbol{x}|c_j)P(c_j) \\ &\approx \underset{c_j \in C}{\arg\max}\, P(c_j)\prod_k P\left(x_k|c_j\right)\end{aligned}$$

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

Probabilistic Classification

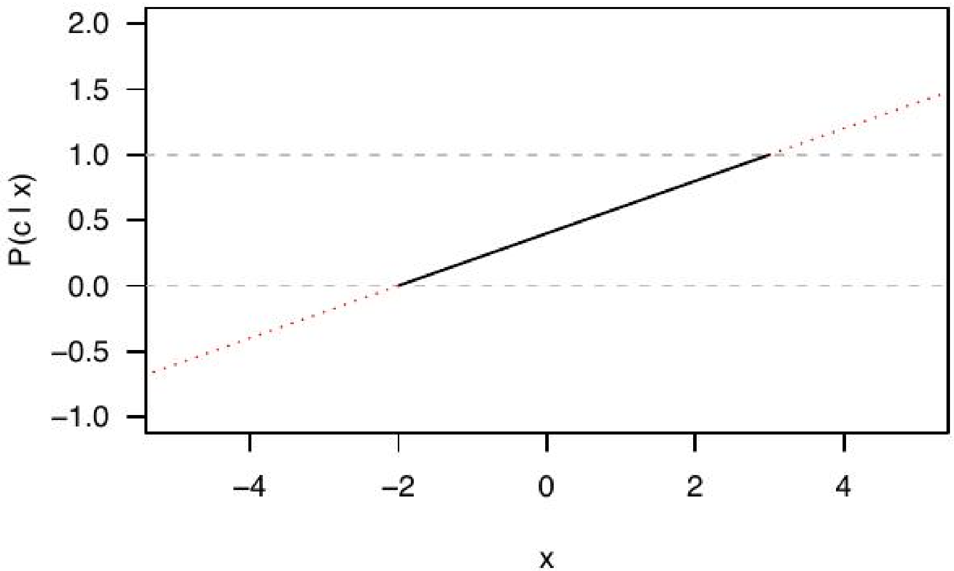
- Strategy: attempt to model $P(c|\boldsymbol{x})$ directly
- Assuming a 2-class problem (Y/N)
  - $P(c = Y|\boldsymbol{x})$: probability of an instance with class *Y*
  - For an instance with class = *Y*: $P(c = Y|\boldsymbol{x}) = 1$
  - For an instance with class = *N*: $P(c = Y|\boldsymbol{x}) =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

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

$$= \beta_0 + \beta_1 x_1 + \ldots + \beta_D x_D$$

Probabilistic Classification

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

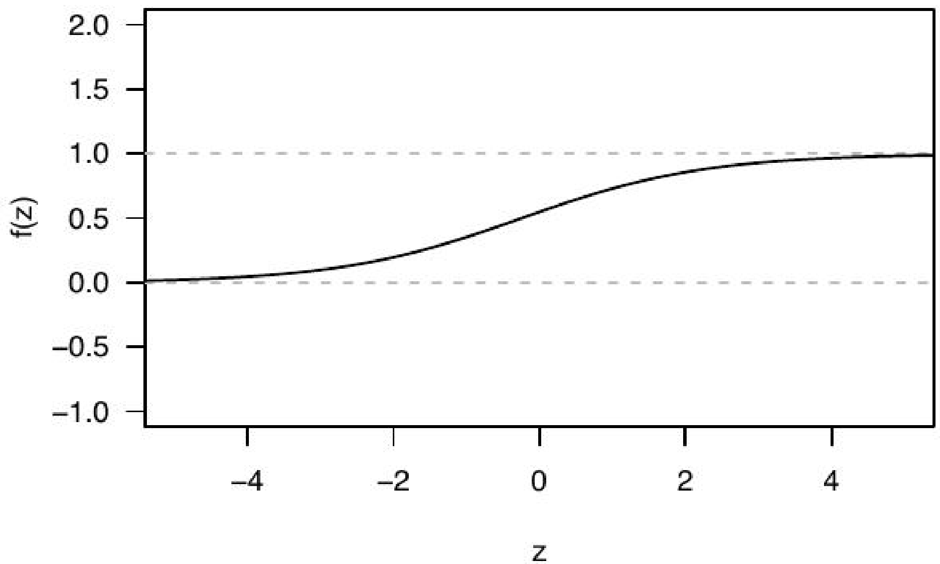


- Linear function, no guarantee that $\hat{P} \in [0,1]$
- Meaning of probabilities outside [0,1] is unclear

Probabilistic Classification

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

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



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

Log-Linear Model

- Derivation of logistic function
- Consider the following multiplicative formulation

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

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

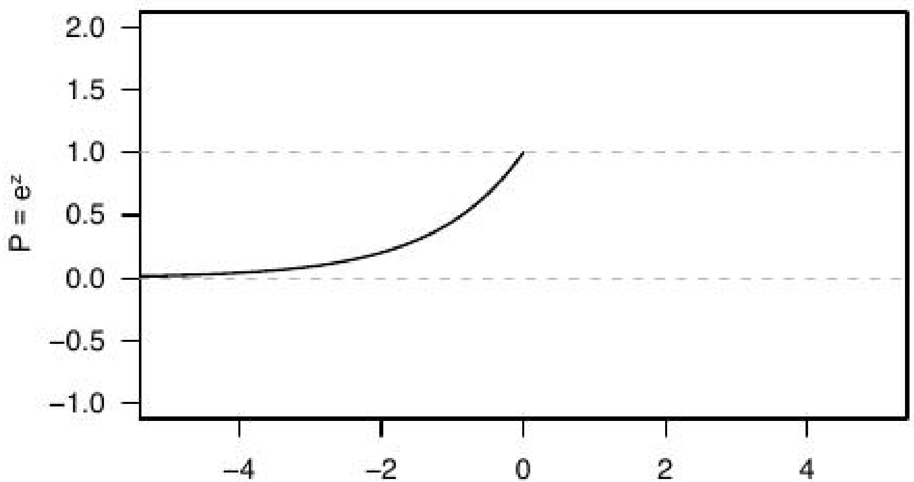
$$\log P(c|\boldsymbol{x}) = \underbrace{\log \gamma_0}_{\beta_0} + x_1 \underbrace{log \gamma_1}_{\beta_1} + \cdots + x_D \underbrace{log \gamma_D}_{\beta_D}$$

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

Log-Linear Model

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

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



$$\mathrm{z} = \boldsymbol{\beta} \cdot \boldsymbol{x}$$

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

Balanced P

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

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

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

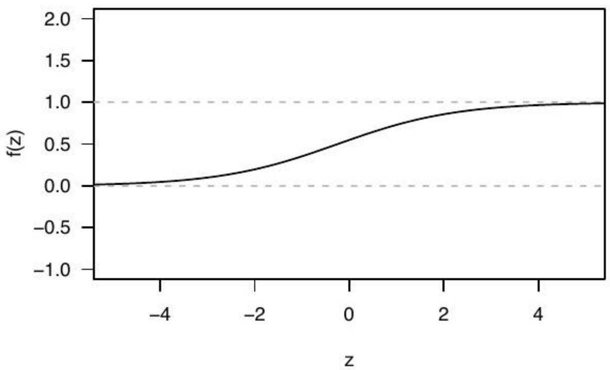
Logistic Regression

- Use linear regression to model logit P

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

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

logistic function



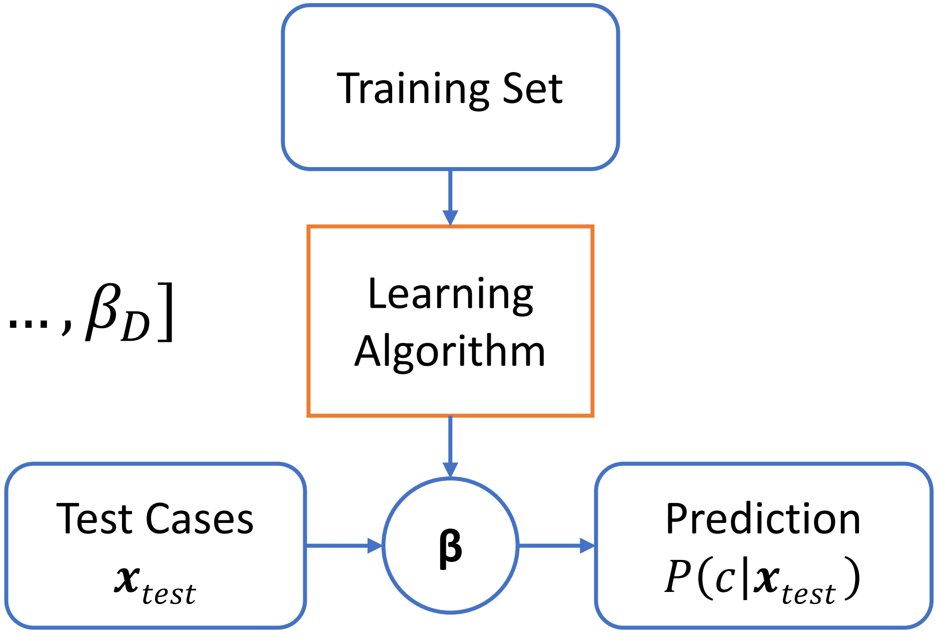
Training and Prediction

Training set: *N* examples
$(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)$,
$\boldsymbol{x}_i = [x_{i0}, x_{i1}, x_{i2}, \ldots, x_{iD}]$

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

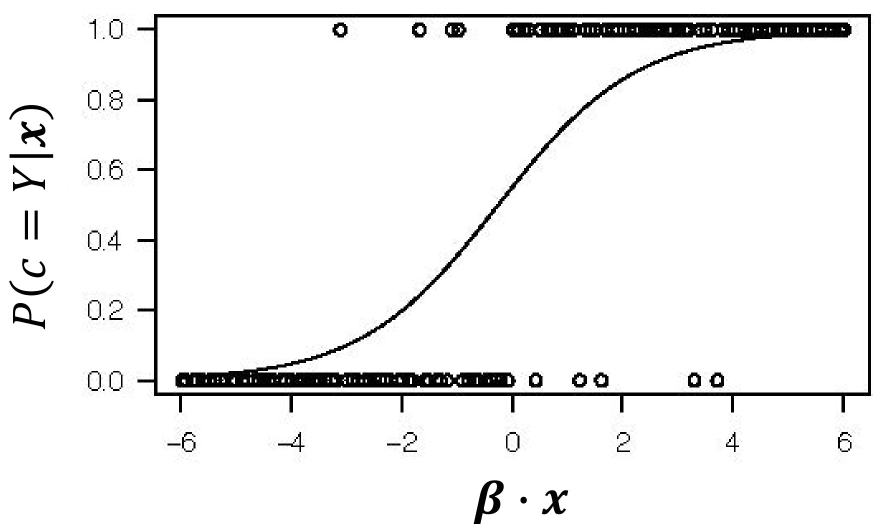
Predict a continuous valued output

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



Training and Prediction

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

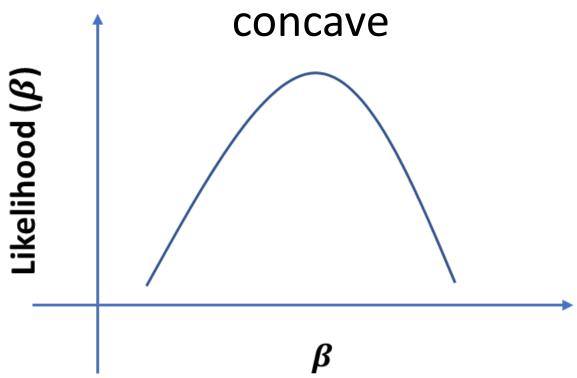


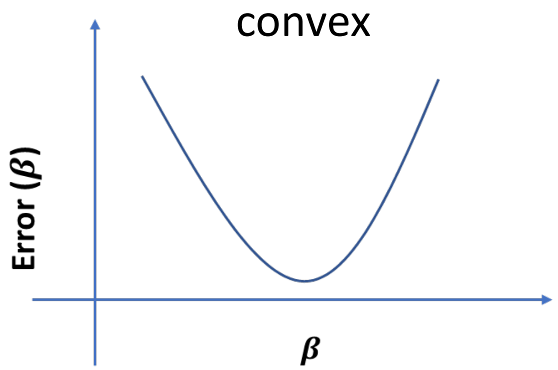
Parameter Estimation

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

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

  - Gradient ascent or descent





Parameter Estimation

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

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

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

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

Parameter Estimation

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

$$\mathrm{P}(Y|\boldsymbol{X},\boldsymbol{\beta}) = \prod_{i=1}^{N} \mathrm{P}(y_i|\boldsymbol{x}_i,\boldsymbol{\beta})$$

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

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

Parameter Estimation

- For simplicity, maximise log-likelihood

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

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

  In iteration $(iter + 1)$ update β_k as:

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

Multi-class Classification

Multinomial logistic regression

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

  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

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

- For pivot class

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

Discussion

Improvement on Naïve Bayes

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

$$P(c_j|\boldsymbol{x}) \approx P(c_j)\prod_k P\left(x_k|c_j\right)$$

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

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

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

Summary

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