PS5841

Data Science in Finance & Insurance

Logistic Regression

Yubo Wang

Autumn 2022

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Classification

- Classifies an observation to a specific class

$$f: R^p \to \{1, \dots, K\}$$

- Model class probabilities directly
  - Tolerance functions, GLM, …
- Model class probabilities via Bayes theorem
  - NB, DA, …
- Separate feature spaces
  - Perceptron, MMC, SVM, Tree, KNN …

Model Probabilities: Tolerance Distribution

- Model probability π using a cumulative probability distribution

$$\pi(t) = \int_{-\infty}^{t} f(s)ds$$

- tolerance distribution $f(s)$

$$f(s) \geq 0$$

$$\int_{-\infty}^{+\infty} f(s)ds = 1$$

Logit Model

- Use this tolerance distribution

$$f(s) = \frac{\beta_1 \exp(\beta_0 + \beta_1 s)}{[1 + \exp(\beta_0 + \beta_1 s)]^2}$$

- then

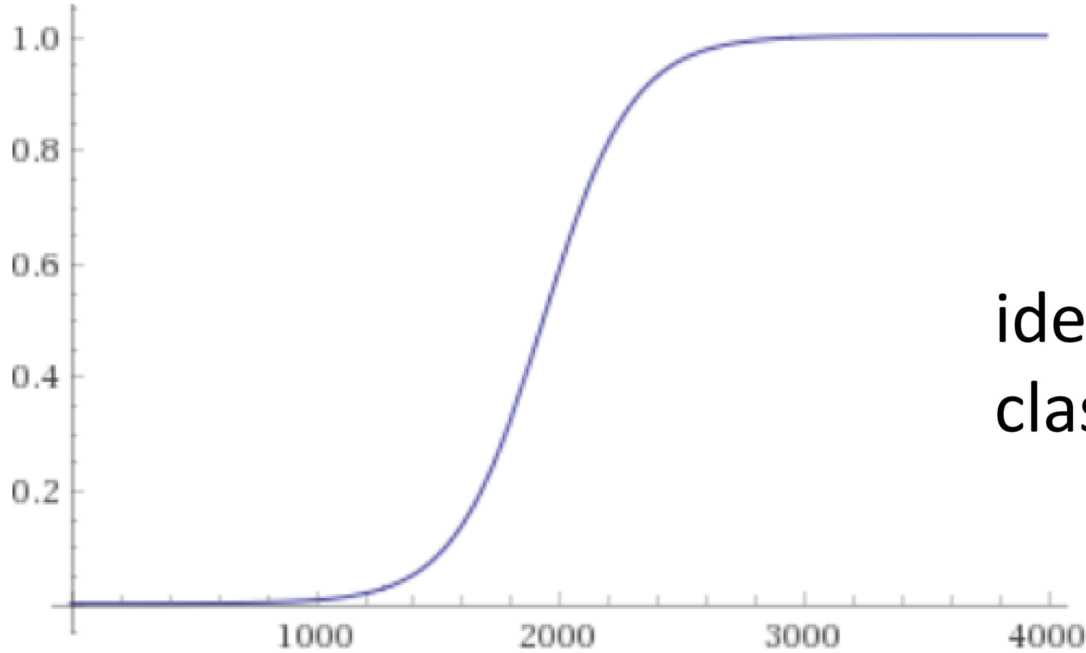
$$\pi(t) = \int_{-\infty}^{t} f(s)ds = \frac{\exp(\beta_0 + \beta_1 t)}{1 + \exp(\beta_0 + \beta_1 t)}$$

- and

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

Profile of π

$$\pi(t) = \frac{\exp(-10.6513 + 0.0055t)}{1 + \exp\left(-10.6513 + 0.0055t\right)}$$



ideal for binary classification

Probit Model

- Use the normal distribution as the tolerance distribution

$$\pi(t) = \Phi\left(\frac{t-\mu}{\sigma}\right)$$

$$\Phi^{-1}(\pi(t)) = -\frac{\mu}{\sigma} + \frac{1}{\sigma}t = \beta_0 + \beta_1 t$$

Binary Classification – basic case

- Example

	amount of study (hours)	passed exam PA (Yes/No)
candidate 1		
⋮		
⋮		
⋮		
candidate n		

Binary Classification – basic case

- n independent RV $Y_i \sim Ber(\pi_i)$

$$\Pr(Y_i = 1) = \pi_i, \Pr(Y_i = 0) = 1 - \pi_i$$

- Likelihood

$$L = \prod_{i:y_i=1} \pi_i \prod_{i:y_i=0} (1 - \pi_i) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i}$$

$$= \exp\left[\sum_{i=1}^{n} y_i \log(\pi_i) + \sum_{i=1}^{n} (1 - y_i) \log(1 - \pi_i)\right]$$

$$= \exp\left[\sum_{i=1}^{n} y_i \log\left(\frac{\pi_i}{1 - \pi_i}\right) + \sum_{i=1}^{n} \log(1 - \pi_i)\right]$$

Logistic Regression

- Logit model

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

- Logit model linear in features

$$\operatorname{logit}\pi_i = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \boldsymbol{x}_i^T\boldsymbol{\beta}$$

- Useful results

$$\pi_i = \pi_i(\boldsymbol{x}_i) = \frac{\exp\left(\boldsymbol{x}_i^T\boldsymbol{\beta}\right)}{1+\exp\left(\boldsymbol{x}_i^T\boldsymbol{\beta}\right)}$$

$$\log(1-\pi_i) = -\log[1+\exp\left(\boldsymbol{x}_i^T\boldsymbol{\beta}\right)]$$

Estimation

- Likelihood

$$l = \sum_{i=1}^{n} y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \log[1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})], \qquad y_i \in \{0,1\}$$

  - (Stochastic) Gradient Descent
  - Newton Raphson

- (Binary) Cross Entropy / Likelihood Ratio

$$-\sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i), \qquad y_i \in \{0,1\}$$

Prediction & Classification

$$\hat{\pi}_i = \frac{\exp\left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}\right)}{1 + \exp\left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}}\right)}$$

- A one-unit increase in x_j is associated with an increase in the log odds by $\hat{\beta}_j$ units.
- Classify $\boldsymbol{x}_i$ to class 1 if $\hat{\pi}_i > \pi^*$, otherwise to class 0
- Linear decision boundary
  - $\text{logit}(\pi)$ is an increasing function in π,

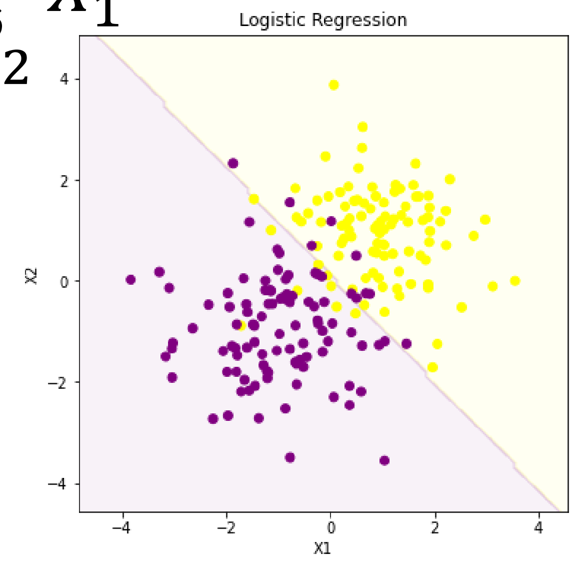
$$\pi > \pi^* \Leftrightarrow \text{logit}(\pi) = \boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}} > \text{logit}(\pi^*)$$

Example: 2D Features

- Let π^* be the classification threshold
  - e.g. $\pi^* = 0.5$

$$X_2 = \frac{\text{logit}(\pi^*) - \hat{\beta}_0}{\hat{\beta}_2} - \frac{\hat{\beta}_1}{\hat{\beta}_2} X_1$$

- Classification when $\hat{\beta}_2 > 0$?



Coding: Logistic Regression

- R

```
base::glm(family=binomial)
```

- Python

```
sklearn.linear_model.LogisticRegression()
```

  - penalty

```
statsmodels.api.Logit()
statsmodels.formula.api.logit()
```

  - encoding

```
statsmodels.formula.api.glm(
    family = statsmodels.genmod.families.Binomial() )
```

Model Probabilities: GLM

- $Y_i \sim Ber(\pi_i)$

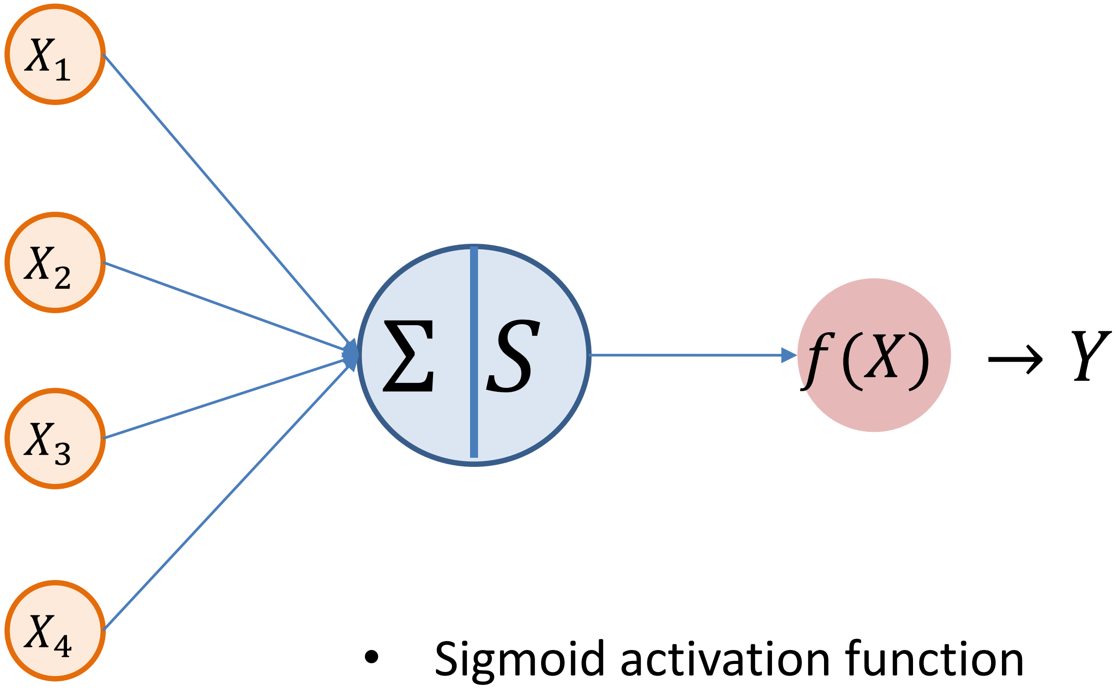
$$E(Y_i) = \pi_i, Var(Y_i) = \pi_i(1 - \pi_i)$$

- GLM with link function $g() = \text{logit}()$

$$g[E(Y_i)] = g(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \boldsymbol{x}_i^T \boldsymbol{\beta}$$

$$\pi_i(\boldsymbol{x}_i) = g^{-1}(\boldsymbol{x}_i^T \boldsymbol{\beta}) = \frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})}$$

NN: Logistic Regression



- Sigmoid activation function

$$Y = f(X) = \frac{1}{1 + \exp\left[-\left(\omega_0 + \omega_1 X_1 + \cdots + \omega_p X_p\right)\right]}$$

- binary cross entropy loss function

Binary Classification – general case

- Example

	amount of study (hours)	cohort size (no. of candidates)	passed exam PA (no. of candidates)
cohort 1		n_1	y_1
⋮		⋮	⋮
⋮		⋮	⋮
⋮		⋮	⋮
cohort N		n_N	y_N

Binary Classification – general case

- N independent RV

  $Y_i \sim Bin(n_i, \pi_i)$ for the i-th subgroup/stratum

  $$\Pr(Y_i = y_i) = \binom{n_i}{y_i} \pi_i^{y_i}(1-\pi_i)^{n_i - y_i}, y_i = 0,1,\dots,n_i$$

- Likelihood

$$L = \exp\left[\sum_{i=1}^{n} y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + n_i \sum_{i=1}^{n} \log(1-\pi_i) + \sum_{i=1}^{n} \log\binom{n_i}{y_i}\right]$$

Coding: General Logistic Regression

- R

```
base::glm(family=binomial)
```

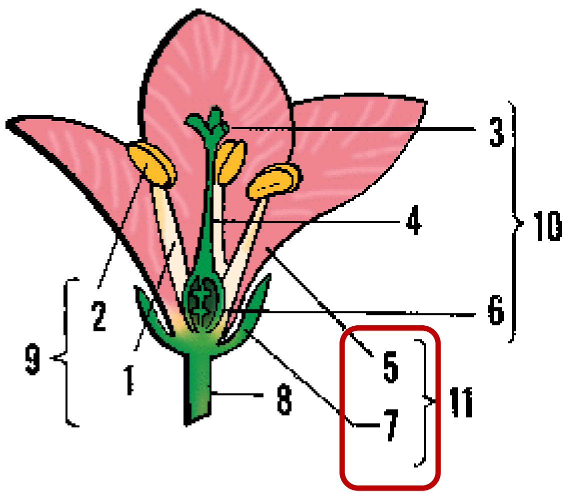
- Python

```
statsmodels.formula.api.glm('y + n_y ~ …',
    data = data,
    family = statsmodels.genmod.families.Binomial() )
```

Multiclass Classification when Y is nominal

- Example (iris)

Illustration of *flower*



Noun

cross section of flower 1b: *1* filament, *2* anther, *3* stigma, *4* style, *5* petal, *6* ovary, *7* sepal, *8* pedicel, *9* stamen, *10* pistil, *11* perianth

- Class
  - Iris-Setosa
  - Iris-Versicolour
  - Iris-Virginica
- Features
  - Petal length & width
  - Sepal length & width

Multiclass Classification when Y is nominal

- Multinomial / Nominal Logistic Regression
  - Reference level, e.g. $k = 1$

$$\text{logit}(\pi_k) = \log\left(\frac{\pi_k}{\pi_1}\right) = \boldsymbol{x}^T\boldsymbol{\beta}_k, \qquad k = 2, \dots, K$$

$$\hat{\pi}_1 = \frac{1}{1 + \sum_2^K \exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)}$$

$$\hat{\pi}_k = \hat{\pi}_1 \exp(\boldsymbol{x}^T\boldsymbol{\beta}_k) = \frac{\exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)}{1 + \sum_2^K \exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)}, \qquad k = 2, \dots, K$$

$$\sum_{k=1}^{K} \hat{\pi}_k = 1$$

Coding: General Logistic Regression

- R

nnet:multnom()

  – Produces $\boldsymbol{x}^T \boldsymbol{\beta}_k = \log\left(\frac{\pi_k}{\pi_{referene}}\right)$

- Python

statsmodels.api.MNLogit(y, X).fit(method = …)

  – Produces $\boldsymbol{x}^T \boldsymbol{\beta}_k = \log\left(\frac{\pi_k}{\pi_{k-1}}\right)$

sklearn.linear_model.LogisticRegression(
    multi_class =multinomial, penalty = ‘none’)

  – uses softmax coding

Softmax Coding

- Treat all classes symmetrically

$$\Pr(Y = k|X = x) = \pi_k(x) = \frac{\exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)}{\sum_{l=1}^{K}\exp(\boldsymbol{x}^T\boldsymbol{\beta}_l)}$$

$$\log\left(\frac{\Pr(Y = k|X = x)}{\Pr(Y = k'|X = x)}\right) = \log\left(\frac{\pi_k(x)}{\pi_{k'}(x)}\right) = \boldsymbol{x}^T(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k'})$$

That was

Maximum Likelihood Estimation

- Log-likelihood

$$l = \sum_{i=1}^{n} l_i = \sum_{i=1}^{n} y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + \sum_{i=1}^{n} \log(1-\pi_i)$$

$$= \sum_{i=1}^{n} y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \log[1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})]$$

- Normal equations

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial l_i}{\partial \beta_j} = \sum_{i=1}^{n} (y_i - \pi_i) x_{ij} = 0, \qquad x_{i0} = 1$$