Data Driven Sampling (Lecture 7)

Classification

Hongwei YUAN

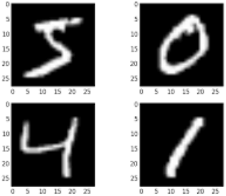
University of Macau

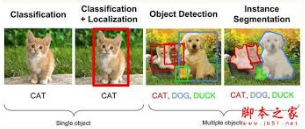
Classification example: customer's credibility

- Bank observes a customer's yearly income and saving, and determines whether he/she is low-risk or high-risk customer according to the yearly income and saving.
- We can model this problem as:
  - customer's yearly income and saving is denoted by random variables X_1 and X_2 respectively;
  - the credibility of customer is modeled by a random variable C with value 0 or 1, where 1 means high-risk customer and 0 means low-risk customer;
  - given (X_1, X_2), we need to find a formula for $P(C = 0|X_1, X_2)$ and $P(C = 1|X_1, X_2)$ to classify customer's credibility, e.g. for any (x_1, x_2)

$$Choose \begin{cases} 1, & P(C = 1|X_1 = x_1, X_2 = x_2) \geq 0.5, \\ 0, & P(C = 1|X_1 = x_1, X_2 = x_2) < 0.5. \end{cases}$$

Classification example:





Two classes: C_1 and C_2

- The observed data $\mathbf{x}$ are in $\mathbb{R}^d$, and belong to the class C_1 or the class C_2.
- For the class C_1, the data density is $p(\mathbf{x}|C_1)$ for the class C_2, the data density is $p(\mathbf{x}|C_2)$
- An example of the above setting: C_1 and C_2 are two car brands, x is the incomes of car customers.

Two classes: C_1 and C_2

- For the class C_i (i=1,2), Likelihood function is

$$L_i(\mathbf{x}) = p(\mathbf{x}|C_i)P(C_i).$$

log likelihood functions is

$$l_i(\mathbf{x}) = \log p(\mathbf{x}|C_i) + \log P(C_i).$$

- Given a data $\mathbf{x} \in \mathbb{R}^d$, it is natural to compare $l_i(\mathbf{x})$ and
  - Choose C_1 if $l_1(\mathbf{x}) \geq l_2(\mathbf{x})$,
  - Choose C_2 if $l_1(\mathbf{x}) < l_2(\mathbf{x})$,
  - The border for classification: $\{x : l_1(x) = l_2(x)\}$, it is often a curve or a line.

Two classes classification: two car brand example

- C_1 and C_2 are two car brands, x is the incomes of car customers.
- For the class C_i (i=1,2), the data density is

$$p(x|C_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}},$$

  where μ_i and σ_i (i=1,2) are all unknown.
- We know

$$l_i(x) = \log P(C_i) - \frac{(x-\mu_i)^2}{2\sigma_i^2} - \frac{1}{2}\log(2\pi) - \log\sigma_i.$$

  In order to compare $l_1(x)$ and $l_2(x)$, one needs to know $P(C_i)$, μ_i and σ_i (i=1,2).

Two classes classification: two car brand example

- We will estimate $P(C_i)$, μ_i and σ_i (i=1,2) by observed data. More precisely, suppose that we observed data

$$\{(x_1, r_1), ..., (x_n, r_n)\}$$

where x_i is the ith customer's income and r_i is the choice of the ith customer, i.e.$r_i = C_1$ or $r_i = C_2$.

- Denote by n_1 the number of customers who chose C_1, and by n_2 the number of customers who chose C_2, we have $n_1 + n_2 = n$. we classify the observed data into two classes, one choosing C_1, the other choosing C_2, i.e.,

$$\{(x_1^1, C_1), ..., (x_{n_1}^1, C_1)\}, \qquad \{(x_1^2, C_2), ..., (x_{n_2}^2, C_2)\}$$

Two classes classification: two car brand example

- We esimate
  - $P(C_1)$ and $P(C_2)$ by

$$\hat{P}(C_1) = \frac{n_1}{n}, \quad \hat{P}(C_2) = \frac{n_2}{n};$$

  - μ_1 and σ_1^2 by

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{n_1} x_i^1}{n_1}, \quad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^{n_1} (x_i^1 - \hat{\mu}_1)^2}{n_1}.$$

  - μ_2 and σ_2^2 by

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{n_2} x_i^2}{n_2}, \quad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^{n_2} (x_i^2 - \hat{\mu}_2)^2}{n_2}.$$

Two classes classification: two car brand example

- Plugging the estimated $\hat{P}(C_i)$, $\hat{\mu}_i$ and $\hat{\sigma}_i$ (i=,2) into the log likelihood function $l_i(x)$ (i=1,2) to obtain an estimated log likelihood function:

$$\hat{l}_i(x) = \log \hat{P}(C_i) - \frac{(x-\hat{\mu}_i)^2}{2\hat{\sigma}_i^2} - \frac{1}{2}\log(2\pi) - \log \hat{\sigma}_i.$$

- Now give a new data $x \in \mathbb{R}$, we can compute its $\hat{l}_1(x)$ and $\hat{l}_2(x)$, and determine its class by
  - if $\hat{l}_1(x) \geq \hat{l}_2(x)$, we claim x belongs to C_1,
  - if $\hat{l}_1(x) < \hat{l}_2(x)$, we claim x belongs to C_2,
  - the border is $\{x : \hat{l}_1(x) = \hat{l}_2(x)\}$, which is often a curve or a line.

General framework for the classification problem

- What we can learn from the previous example:
  - We have seen from the previous slides, we take the two likelihood function $l_1(x)$ and $l_2(x)$ as a criterion to determine which class a given data belongs to.
  - $l_1(x)$ and $l_2(x)$ are called discriminant functions in classification problem.
- General framework for classification problem:
  - K classes: $C_1, ..., C_K$,
  - K discriminant functions: $g_1(x), ..., g_K(x)$,
  - Criterion: for a given data x, if $g_k(x) = \max_{1 \le i \le K} g_i(x)$, we choose C_k as the class of x.
  - We will have $\frac{K(K-1)}{2}$ borders, i.e. $\{x : g_i(x) = g_j(x)\}$ for $i \neq j$.

Linear discriminant

We still consider two classes classification problem. Let the two discriminant functions be $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^d$. Suppose that

$$g_1(\mathbf{x}) = \theta_1^T \mathbf{x} + \theta_{1,0}, \quad g_1(\mathbf{x}) = \theta_2^T \mathbf{x} + \theta_{2,0},$$

where $\theta_i = (\theta_{i,1}, ..., \theta_{i,d})^T$ for $i = 1, 2$.

- If $g_1(\mathbf{x}) \geq g_2(\mathbf{x})$ for $\mathbf{x}$, we choose the class C_1 for $\mathbf{x}$. Otherwise, we choose C_2 for $\mathbf{x}$.
- Define $g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$, then we have
  - If $g(\mathbf{x}) > 0$, choose C_1; if $g(\mathbf{x}) \leq 0$, choose C_2.
  - Since $g(\mathbf{x})$ is linear, we use a plane $g(\mathbf{x}) = 0$ to separate the two classes.

A simple example for linear discriminant

We have a random number $\mathbf{z}$ which is either from $N(\mu_1, \Sigma_1)$ or from $N(\mu_2, \Sigma_2)$, with (here we assume that $P(C_1) = P(C_2) = 1/2$)

$$\mu_1 = [2,2]^T, \quad \mu_2 = [0,0]^T, \quad \Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Assignment 1: Compute log likelihood function $l_1(\mathbf{x}) = -\frac{(x_1-2)^2+(x_2-2)^2}{2} + C$ and $l_2(\mathbf{x}) = -\frac{x_1^2+x_2^2}{2} + C$ with $C = -\log(4\pi)$.
- As shown in the previous slide, we take $l_1(\mathbf{x})$ and $l_2(\mathbf{x})$ as the two discriminant functions. Denote $l(\mathbf{x}) = l_1(\mathbf{x}) - l_2(\mathbf{x}) = 2x_1 + 2x_2 - 4$.
- If $l(\mathbf{z}) > 0$, $\mathbf{z}$ belongs to class 1. If $l(\mathbf{z}) \leq 0$, $\mathbf{z}$ belongs to class 2.

A simple example for linear discriminant

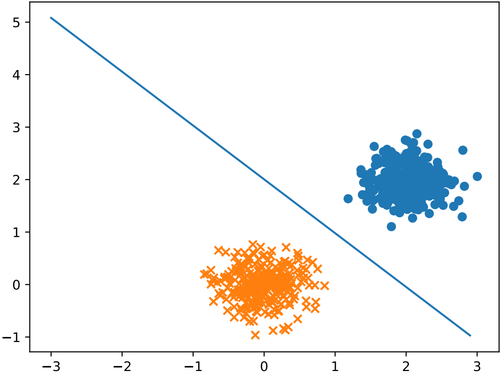
Question: I don't know the sources of these data, but only know their covariances are both identity matrix. How can I classify a given data $\mathbf{z}$?

- You sample n_1 random numbers $\mathbf{x_1}, ..., \mathbf{x_{n_1}}$ from $N(\mu_1, \Sigma_1)$ and another $n_2 = 30$ random numbers $\mathbf{y_1}, ..., \mathbf{y_{n_2}}$ from $N(\mu_2, \Sigma_2)$ for me.
- I use these data to esimate $\mu_\mathbf{1}$ and $\mu_\mathbf{2}$ by

$$\hat{\mu_\mathbf{1}} = \frac{\mathbf{x_1} + ... + \mathbf{x_{n_1}}}{n_1}, \hat{\mu_\mathbf{2}} = \frac{\mathbf{y_1} + ... + \mathbf{y_{n_2}}}{n_2}$$

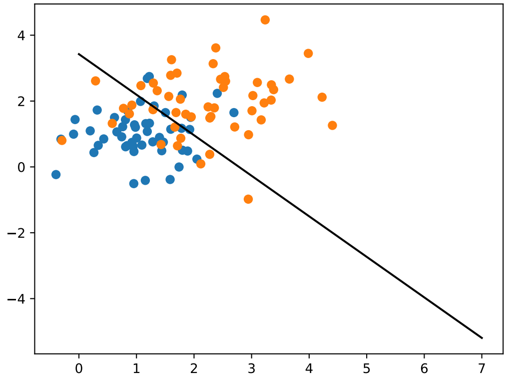
- Replacing $\mu_\mathbf{1}, \mu_\mathbf{2}$ by $\hat{\mu_\mathbf{1}}, \hat{\mu_\mathbf{1}}$ respectively, we get $\hat{l}_1(\mathbf{x}), \hat{l}_2(\mathbf{x})$ and $\hat{l}(\mathbf{x})$.
- (Class Exercise) Show that $\hat{l}(\mathbf{x}) = (\hat{\mu_\mathbf{1}} - \hat{\mu_\mathbf{2}})^T \mathbf{x} + \frac{1}{2}((\hat{\mu_\mathbf{2}})^T \hat{\mu_\mathbf{2}} - (\hat{\mu_\mathbf{1}})^T \hat{\mu_\mathbf{1}})$.

```
import numpy as np
from scipy.stats import cauchy
from scipy.optimize import minimize_scalar
import matplotlib.pyplot as plt
from sympy import *
num1,num2=300,300
np.random.seed(0)
mean_1 = [0, 0]
cov_1 = [[0.1, 0], [0, 0.1]]
mean_2 = [2, 2]
cov_2 = [[0.1, 0], [0, 0.1]]
x1=np.random.multivariate_normal(mean_1, cov_1, num1)
x2=np.random.multivariate_normal(mean_2, cov_2, num2)
plt.scatter(x2[:,0],x2[:,1],marker='o')
plt.scatter(x1[:,0],x1[:,1],marker='x')
#def p(a,b):
#    return a/(a+b)
average_mu1=sum(x1)/num1
average_mu2=sum(x2)/num2
print(average_mu1)
print(average_mu2)
def l1(x):
    return np.dot((x-average_mu1),(x-average_mu1))
def l2(x):
    return np.dot((x-average_mu2),(x-average_mu2))
z_test=(4,5)
if l1(z_test)>l2(z_test):
    print('z_test belongs to class 1')
else: print('z_test belongs to class 2')
x=np.arange(-3,3,0.1)
a=(average_mu2[0]-average_mu1[0])/(average_mu2[1]-average_mu1[1])
b=4/(average_mu2[1]-average_mu1[1])
print(a)
print(b)
y=b-a*x
#print(y)
plt.plot(x,y)
plt.savefig('class.pdf')
plt.show()
```


5
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
mean1 = (1,1)
mean2 = (2,2)
cov1 = np.array([[0.5,0],[0,0.5]])
cov2 = np.array([[1,0],[0,1]])
np.random.seed(10)
x1 = np.random.multivariate_normal(mean1, cov1, (50,), 'raise')
x2 = np.random.multivariate_normal(mean2, cov2, (50,), 'raise')
x = np.r_[x1, x2]
y = [0]*50+[1]*50 #class labels

#SVM
clf = svm.SVC(kernel = 'linear')
clf.fit(x, y)
#build the boundary
w = clf.coef_[0]
xx = np.linspace(0, 7)
a = -w[0] / w[1]
yy = a*xx - (clf.intercept_ / w[1])
plt.scatter(x1[:, 0], x1[:, 1])
plt.scatter(x2[:, 0], x2[:, 1])
plt.plot(xx, yy, 'k-')
plt.savefig("classification graph.pdf")
plt.show()
```


4
2
0
−2
−4
0
1
2
3
4
5
6
7

Logistic regression (1 dimension case)

Let X be a continuous random variable (valued on $\mathbb{R}$), which can be in the class C_1 or C_2 with probability $P(C_1)$ and $P(C_2)$ respectively.

- Let $p(x|C_i)$ be conditional probability density of X given C_i $(i = 1, 2)$.
- Let $p(x)$ be the probability density of X, we have

$$p(x) = p(x|C_1)P(C_1) + p(x|C_2)P(C_2).$$

- Let $P(C_i|X = x)$ be conditional probability of C_i given $X = x$, we simply denote it by $P(C_i|x)$ $(i = 1, 2)$.
- Let $p(C_i, x)$ be the probability density of x and C_i $(i = 1, 2)$.

Logistic regression (1 dimension case)

- One assumes

$$\log \frac{p(x|C_2)}{p(x|C_1)} = \beta_1 x + \beta_0,$$

where β_0 and β_1 is some unknown parameter. By Bayes' rule,

$$\frac{P(C_2|x)}{P(C_1|x)} = \frac{p(C_2,x)/p(x)}{p(C_1,x)/p(x)} = \frac{p(x|C_2)P(C_2)}{p(x|C_1)P(C_1)},$$

which implies

$$\log \frac{P(C_2|x)}{P(C_1|x)} = \beta_1 x + \beta_0', \quad (*)$$

where $\beta_0' = \beta_0 + \log \frac{P(C_2)}{P(C_1)}$.

- By the above deviation, we assume

$$\log \frac{P(C_2|x)}{P(C_1|x)} = \beta_0' + \beta_1 x,$$

it is called logistic regression model with parameters β_0', β_1.

Logistic regression (1 dimension case)

- We have

$$P(C_2|x) + P(C_1|x) = 1. \qquad (**)$$

- By (*) and (**),

$$P(C_2|x) = \frac{e^{\beta_0' + \beta_1 x}}{1 + e^{\beta_0' + \beta_1 x}} = \frac{1}{1 + e^{-\beta_0' - \beta_1 x}}, \qquad P(C_1|x) = \frac{1}{1 + e^{\beta_0' + \beta_1 x}}$$

- In Statistics, for two classes classification problem, we often use 1 and 0 to denote classes. For instance,

$$Y = \begin{cases} 1, & if\ C_2; \\ 0, & if\ C_1. \end{cases}$$

So $P(Y = 1) = P(C_2)$ and $P(Y = 0) = P(C_1)$.

Logistic regression (1 dimension case)

- We can see $P(1|x) = \frac{1}{1+e^{-\beta_0' - \beta_1 x}}$, $P(0|x) = \frac{1}{1+e^{\beta_0' + \beta_1 x}}$.
- Assignment 2: Show that

$$P(y|x) = \frac{1}{1 + e^{(1-2y)(\beta_0' + \beta_1 x)}}, \quad y = 0, 1.$$

Logistic regression (1 dimension case)

Let $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ be a sequence of observed data.

- Let the probabilty of X is $p(x)$ (we shall see that the regression does not depends on the probability of X), then we have

$$p(1, x) = P(1|x)p(x), \qquad p(0, x) = P(0|x)p(x).$$

- The likelihood function of these data are

$$L_n(\beta_0', \beta_1) = \prod_{i=1}^{n} p(y_i, x_i) = \prod_{i=1}^{n} P(y_i|x_i) \prod_{i=1}^{n} p(x_i) \tag{1}$$

- Assignment 2: Show that

$$L_n(\beta_0', \beta_1) = \prod_{i=1}^{n} \frac{1}{1 + e^{(1-2y_i)(\beta_0' + \beta_1 x_i)}} \prod_{i=1}^{n} p(x_i)$$

Logistic regression (1 dimension case)

- The log likelihood function is

$$l_n(\beta_0', \beta_1) = \sum_{i=1}^{n} \log \frac{1}{1 + e^{(1-2y_i)(\beta_0' + \beta_1 x_i)}} + \sum_{i=1}^{n} \log p(x_i).$$

- The MLE is

$$\begin{aligned}(\hat{\beta}_{0,n}', \hat{\beta}_{1,n}) &= argmax_{(\beta_0', \beta_1) \in \mathbb{R}^2} l_n(\beta_0', \beta_1) \\ &= argmax_{(\beta_0', \beta_1) \in \mathbb{R}^2} \sum_{i=1}^{n} \log \frac{1}{1 + e^{(1-2y_i)(\beta_0' + \beta_1 x_i)}} \\ &= argmin_{(\beta_0', \beta_1) \in \mathbb{R}^2} \sum_{i=1}^{n} \log(1 + e^{(1-2y_i)(\beta_0' + \beta_1 x_i)})\end{aligned} \tag{2}$$

Logistic regression (1 dimension case)

- (Class Exercise) $(\hat{\beta}'_{0,n}, \hat{\beta}_{1,n})$ satisfies the following equations:

$$\sum_{i=1}^{n} \frac{1-2y_i}{1+e^{-(1-2y_i)(\beta'_0+\beta_1 x_i)}} = 0, \quad \sum_{i=1}^{n} \frac{(1-2y_i)x_i}{1+e^{-(1-2y_i)(\beta'_0+\beta_1 x_i)}} = 0$$

- Unfortunately, it is hard to solve these equations. We shall learn (stochastic) gradient descent algorithm to solve it later.
- One simplified way is:
  - Find estimates $\hat{P}(1|x)$ and $\hat{P}(0|x)$ of $P(1|x)$ and $P(0|x)$ for a given x.
  - Consider $\log \frac{P(1|x_i)}{P(0|x_i)} = \beta'_0 + \beta_1 x_i$ for $i = 1, ..., n$, then we can estimate β'_0 and β_1 by least square regression.

Logistic regression (multidimensional case)

Let X be a random variable (valued on $\mathbb{R}^d$), and let Y be its label with a value 0 or 1.

- Denote $P(1|x) = P(Y = 1|X = x)$ and $P(0|x) = P(Y = 0|X = x)$ for a given $x = (x_1, ..., x_d) \in \mathbb{R}^d$, then

$$P(0|x) = 1 - P(1|x).$$

- (Class Exercise) We shall take

$$P(1|x) = \frac{1}{1 + e^{-\beta_0' - \beta_1 x_1 - ... - \beta_d x_d}},$$

$$P(0|x) = \frac{1}{1 + e^{\beta_0' + \beta_1 x + ... + \beta_d x_d}}.$$

- We will not enter the details of multidimensional logistic regression.

A case study of Stock market data (SP 500 stock index)

In this case study, we will use the SP 500 stock index from 2001 to 2005 to predict the direction of stock ('up' or 'down'). The data structure is as the following:

- 1250 lines data, every line has: year, day-5, ..., day-1, volume, today, direction
- (day-5,...,day-1) are the change in SP 500 indexes of the previous 5 days, volume is the shares traded in the previous day (in billion unit), today is today's SP 500 index, direction is today's direction (up or down).
- We shall use the logistic model to fit these data as the following:
  direction of today $\sim$ day-5,...,day-1, volume.
- We compare the direction predicted by the model with the known direction of today, to evaluate the model.

	Year	day -5	day -4	day -3	day -2	day -1	Volume	Today	Direction
1	2001	0.381	-0.192	-2.624	-1.055	5.01	1.1913	0.959	Up
2	2001	0.959	0.381	-0.192	-2.624	-1.055	1.2965	1.032	Up
3	2001	1.032	0.959	0.381	-0.192	-2.624	1.4112	-0.623	Down
4	2001	-0.623	1.032	0.959	0.381	-0.192	1.276	0.614	Up
5	2001	0.614	-0.623	1.032	0.959	0.381	1.2057	0.213	Up
6	2001	0.213	0.614	-0.623	1.032	0.959	1.3491	1.392	Up
7	2001	1.392	0.213	0.614	-0.623	1.032	1.445	-0.403	Down
8	2001	-0.403	1.392	0.213	0.614	-0.623	1.4078	0.027	Up
9	2001	0.027	-0.403	1.392	0.213	0.614	1.164	1.303	Up
10	2001	1.303	0.027	-0.403	1.392	0.213	1.2326	0.287	Up
11	2001	0.287	1.303	0.027	-0.403	1.392	1.309	-0.498	Down
12	2001	-0.498	0.287	1.303	0.027	-0.403	1.258	-0.189	Down
13	2001	-0.189	-0.498	0.287	1.303	0.027	1.098	0.68	Up
14	2001	0.68	-0.189	-0.498	0.287	1.303	1.0531	0.701	Up
15	2001	0.701	0.68	-0.189	-0.498	0.287	1.1498	-0.562	Down
16	2001	-0.562	0.701	0.68	-0.189	-0.498	1.2953	0.546	Up
17	2001	0.546	-0.562	0.701	0.68	-0.189	1.1188	-1.747	Down
18	2001	-1.747	0.546	-0.562	0.701	0.68	1.0484	0.359	Up
19	2001	0.359	-1.747	0.546	-0.562	0.701	1.013	-0.151	Down
20	2001	-0.151	0.359	-1.747	0.546	-0.562	1.0596	-0.841	Down
21	2001	-0.841	-0.151	0.359	-1.747	0.546	1.1583	-0.623	Down
22	2001	-0.623	-0.841	-0.151	0.359	-1.747	1.1072	-1.334	Down
23	2001	-1.334	-0.623	-0.841	-0.151	0.359	1.0755	1.183	Up
24	2001	1.183	-1.334	-0.623	-0.841	-0.151	1.0391	-0.865	Down
25	2001	-0.865	1.183	-1.334	-0.623	-0.841	1.0752	-0.218	Down

A case study of Stock market data (SP 500 stock index)

We denote $\mathbf{x} = (\text{day} - 1, ..., \text{day} - 5, \text{volume})$

- Denote $y = 0$ if 'up' and $y = 1$ if 'down', for the i-th line's data $\mathbf{x}_i$,

$$\hat{P}(0|\mathbf{x}_i) = \frac{\sharp(`+' \text{ index of } 5 \text{ days})}{5}, \quad \hat{P}(1|\mathbf{x}_i) = \frac{\sharp(`-' \text{ index of } 5 \text{ days})}{5},$$

- The logistic model is

$$\log \frac{\hat{P}(1|\mathbf{x}_i)}{\hat{P}(0|\mathbf{x}_i)} = \beta^T \mathbf{x}_i + \beta_0$$

where $\beta = (\beta_6, \beta_5, ..., \beta_1)$.

A case study of Stock market data (SP 500 stock index)

The data of the first line: 5 days' indices=(0.381,-0.192,-2.624,-1.055,5.01), volume=1.1913, i.e. $\mathbf{x}_1 = (0.381, -0.192, -2.624, -1.055, 5.01, 1.1913)$.

- $$\hat{P}(0|\mathbf{x}_1) = \frac{\sharp(`+' \text{ index of } 5 \text{ days})}{5} = \frac{2}{5},$$
  $$\hat{P}(1|\mathbf{x}_1) = \frac{\sharp(`-' \text{ index of } 5 \text{ days})}{5} = \frac{3}{5}.$$
- The logistic model for the data of the first line data is
  $$\log \frac{3}{2} = \beta^T \mathbf{x_1} + \beta_0.$$
  where $\beta = (\beta_6, \beta_5, ..., \beta_1)$ and β_0 are coefficients not known.
- Similarly, we can write down 1250 relations as above by all the 1250 lines' data.

A case study of Stock market data (SP 500 stock index)

- Running the glm(Generalized Linear Model) model function in Python, we can get the values of $\hat{\beta} = (\hat{\beta}_6, ..., \hat{\beta}_1, \hat{\beta}_0)$ from the 1250 relations.
- We need to evaluate whether this glm model is good or bad for fitting the SP 500 data, in the following way: for each line's data $\mathbf{x}_i$
  - Compute
    $$a_i = \hat{\beta}^T \mathbf{x_i} + \hat{\beta}_0,$$
    if $a_i > 0$, we take $y_i = 1$, 'down'; if $a_i \leq 0$, we take $y_i = 0$, 'up'.
  - Compare the above decision ('up' or 'down') with the known 'up' or 'down' in the i-th line. If they are the same, the prediction is correct. Otherwise, the prediction is wrong.

A case study of Stock market data (SP 500 stock index)

- Evaluation score=$\frac{\text{correct predictions}}{1250}$.
- If the evaluation score is not high, we can conclude that the model is bad for fitting the data.

A case study of Stock market data (SP 500 stock index)

```
import pandas as pd # pandas includes data structure
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf  # glm is in the package statsmodels
import statsmodels.api as sm
from sklearn.metrics import confusion_matrix    # evaluation of the model, diagonal is correct and off-diagonal is wrong
from pandas import Series, DataFrame

data = pd.read_csv("Smarket.csv",index_col=0)
print(data.columns.values.tolist()) #print the colnames
print(data.shape) #print the size of data
print(data.describe()) #descriptive statistical analysis
data1 = data.drop('Direction',axis=1) #delete the qualitative variable 'Direction'
print(data1)
print(data1.corr()) #compute the correlations among the predictors in a data set
plt.plot(data[['Volume']])
plt.show()

# Using the Generalized Linear Model (GLM) to fit the Logistic regression
model = smf.glm("Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume", family = sm.families.Binomial(), data=data)
result = model.fit()
print(result.summary())
print(np.column_stack((data[["Direction"]].values.flatten(),result.model.endog))) # check the dummy variables !!!down=1

# access the coefficients and p-values
print("Coefficeients")
print(result.params)
print()
print("p-Values") # p-values for the coefficients
print(result.pvalues)
print()
# predict the probability the market will go down
print("the probability of market down")
probs = result.predict()
print(probs[1:10])
print()
print("Dependent variables")
print(result.model.endog_names)
print()
```

A case study of Stock market data (SP 500 stock index)

```
# create a vector of class predictions based on whether the predicted probability of a market
# increase is greater than or less than 0.5
pred = ['Up' if x < 0.5 else 'Down' for x in probs]   # definition

#  produce a confusion matrix in order to determine how many
# observations were correctly or incorrectly classified
print('confusion matrix')
conmat = confusion_matrix(data["Direction"], pred)
print(conmat)
print()

print('correctly predicted probability')
print(sum(np.diag(conmat))/len(data['Direction']))
print()

print('the fraction of days for which the prediction was correct')
print(np.mean(pred == data['Direction']))#the fraction of days for which the prediction was correct
```