

1

(1)

带入得 $(b-w)^2 + (b)^2 + (1-(w+b))^2$
 $= (b^2 - 2bw + w^2) + b^2 + 1 - 2(w+b) + w^2 + 2wb + b^2$
 $= 3b^2 + 2w^2 - 2w - 2b + 1$
所以 $w = \frac{1}{2}, b = \frac{1}{3}$

(2)

$$\begin{aligned} dis^2 &= \sum \frac{(w_E x_i + b_E + y_i)^2}{w_E^2 + 1} \\ &= \frac{b_E^2}{w_E^2 + 1} + \frac{(b_E - w_E)^2}{w_E^2 + 1} + \frac{(1 + b_E + w_E)^2}{w_E^2 + 1} \\ &= \frac{3b_E^2 + 2w_E^2 + 2b_E + 1}{w_E^2 + 1} \end{aligned}$$

分别对 b_E, w_E 求偏导.

对 b 求偏导: $w\bar{x} - \bar{y} + b = 0$

$$b = \frac{1}{3}$$

对 w 求偏导: $\frac{w}{m} \sum_{i=1}^m (x_i^2 - y_i^2) + \frac{w^2 - 1}{m} \sum_{i=1}^m x_i y_i + 2bw\bar{y} - bw^2\bar{x} + b\bar{x} - b^2w = 0$

带入数值得 $w = \frac{\sqrt{13}}{3} - \frac{2}{3} = 0.535$

(3)

$$(w^*, b^*) = \operatorname{argmin}_{w, b} \sum_{i=1}^m \left| \frac{wx_i + b - y_i}{\sqrt{w^2 + 1}} \right|$$

当 $(w^*, b^*) = (0.5, \frac{1}{3})$ 时, $\sum_{i=1}^m \left| \frac{wx_i + b - y_i}{\sqrt{w^2 + 1}} \right| = 0.596$

而当 $(w, b) = (0.5, 0.5)$ 时, $\sum_{i=1}^m \left| \frac{wx_i + b - y_i}{\sqrt{w^2 + 1}} \right| = 0.45 < 0.596$ 所以 $(w^*, b^*) \neq (0.5, \frac{1}{3})$

2

$$\frac{p(y=i|x)}{p(y=K|x)} = \frac{e^{z_i}}{\sum e^{z_i}}$$

3

方法一

```
1 import numpy as np
2 from pandas import read_csv
3 from math import exp
4
5 theta = 0.5
6
7 dataset = read_csv('F:\杂\大二下\机器学习\ch03-data\train_feature.csv')
8 dataset['10'] = [1 for i in range(len(dataset))]
9 ground_truth = read_csv('F:\杂\大二下\机器学习\ch03-
  data\train_target.csv')
```

```

10 # print(dataset)
11 dataMatrix = np.array(dataset[1:])
12 truthMatrix = np.array(ground_truth[1:])
13 betaMatrix = np.dot(np.dot(np.linalg.inv(np.dot(dataMatrix.T, dataMatrix)),
14 dataMatrix.T), truthMatrix)
15
16 val_feature = read_csv('F:\\杂\\大二下\\机器学习\\ch03-data\\val_feature.csv')
17 val_target = read_csv('F:\\杂\\大二下\\机器学习\\ch03-data\\val_target.csv')
18
19 val_feature['10'] = [1 for i in range(len(val_feature))]
20
21 valMatrix = np.array(val_feature)
22 val_Truth = np.array(val_target)
23
24 TP = 0
25 FP = 0
26 TN = 0
27 FN = 0
28
29 z = np.dot(valMatrix, betaMatrix)
30
31 for i in range(len(z)):
32     f = 1 / (1 + exp(-z[i]))
33     if f > theta:
34         if val_Truth[i] == 1:
35             TP += 1
36         else:
37             FP += 1
38     else:
39         if val_Truth[i] == 1:
40             FN += 1
41         else:
42             TN += 1
43
44 print(betaMatrix)
45 print(TP, FP, FN, TN)
46
47 print('准确率: %f 查全率: %f 查准率: %f' % ((TP + TN) / (TP + TN + FP + FN), TP
48 / (TP + FN), TP / (TP + FP)))

```

准确率: 0.740000 查全率: 1.000000 查准率: 0.666667

方法二:

```

1 import numpy as np
2 from pandas import read_csv
3 from numpy import exp
4
5 theta = 0.5
6
7 def p1(x):
8     return x / (1 + x)
9
10
11 dataset = read_csv('F:\\杂\\大二下\\机器学习\\ch03-data\\train_feature.csv')
12 dataset['10'] = [1 for i in range(len(dataset))]
13 ground_truth = read_csv('F:\\杂\\大二下\\机器学习\\ch03-
14 data\\train_target.csv')

```

```

14 dataMatrix = np.array(dataset[:])
15 truthMatrix = np.array(ground_truth[:])
16 val_feature = read_csv('F:\\杂\\大二下\\机器学习\\ch03-data\\val_feature.csv')
17 val_target = read_csv('F:\\杂\\大二下\\机器学习\\ch03-data\\val_target.csv')
18 val_feature['10'] = [1 for i in range(len(val_feature))]
19 valMatrix = np.array(val_feature)
20 val_Truth = np.array(val_target)
21
22
23
24 betaMatrix = np.zeros((11, 1))
25 betaMatrix[10][0] = 1
26
27
28 loop = 0
29
30 while loop < 100:
31     Ans = np.dot(dataMatrix, betaMatrix)
32     tmp1 = np.zeros((11, 11))
33     tmp2 = np.zeros((11, 1))
34     tmp_Matrix = np.zeros((11, 1))
35
36     for i in range(len(dataMatrix)):
37         P1 = p1(exp(Ans[i][0]))
38         tmp_Matrix = dataMatrix[i:i+1, :].T
39         tmp2 -= tmp_Matrix * (truthMatrix[i][0] - P1)
40         tmp1 += np.dot(tmp_Matrix, tmp_Matrix.T) * P1 * (1 - P1)
41
42     betaMatrix -= np.dot(np.linalg.inv(tmp1), tmp2)
43     loop += 1
44
45 print(betaMatrix)
46
47 TP = 0
48 FP = 0
49 TN = 0
50 FN = 0
51
52 z = np.dot(valMatrix, betaMatrix)
53
54 for i in range(len(z)):
55     f = 1 / (1 + exp(-z[i]))
56     if f > theta:
57         if val_Truth[i] == 1:
58             TP += 1
59         else:
60             FP += 1
61     else:
62         if val_Truth[i] == 1:
63             FN += 1
64         else :
65             TN += 1
66 print(TP, FP, TN, FN)
67 print('准确率: %f 查全率: %f 查准率: %f' % ((TP + TN) / (TP + TN + FP + FN), TP
68 / (TP + FN), TP / (TP + FP)))

```

准确率: 1.000000 查全率: 1.000000 查准率: 1.000000