

ML Homework4

Due Date: 2021/04/26 (MON.) 23:55

TAs' email address: jhhlab.tw@gmail.com

Description :

1. Logistic regression

- Input:
 1. N (number of data points)
 2. $mx_1, vx_1, my_1, vy_1, mx_2, vx_2, my_2, vy_2$ (m : mean, v : variance)
- Function:
 1. Generate n data point: $D1 = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where x and y are independently sampled from $N(mx_1, vx_1)$ and $N(my_1, vy_1)$ respectively.
 2. Generate n data point: $D2 = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where x and y are independently sampled from $N(mx_2, vx_2)$ and $N(my_2, vy_2)$ respectively.
 3. Use Logistic regression to separate $D1$ and $D2$. You should implement both Newton's and steepest gradient descent method during optimization.
 - In other words, when the Hessian is singular, use steepest descent for instead. You should come up with a reasonable rule to determine convergence.(a simple run out of the loop should be used as the ultimatum)
- Output:
 1. The confusion matrix and the **sensitivity** and **specificity** of the logistic regression applied to the training data D .
 2. Visualization
 - Plot the ground truth
 - Plot the predict result
 - Gradient descent
 - Newton's method

Use the Gaussian random number generator in homework 3.

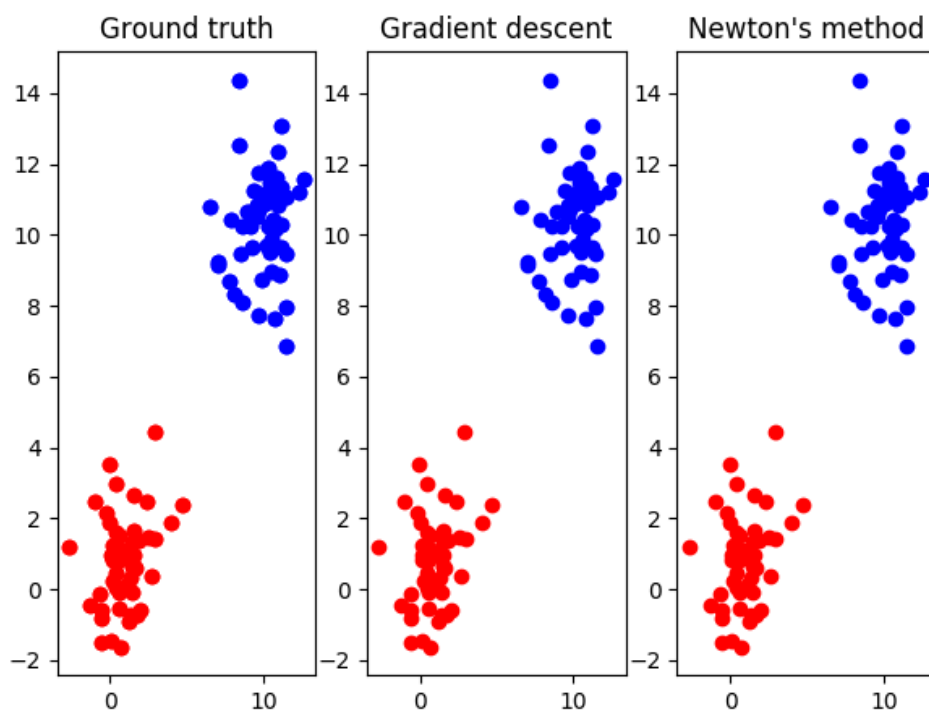
- Sample input & output (for reference only)
 - Case 1: $N = 50, mx_1 = my_1 = 1, mx_2 = my_2 = 10, vx_1 = vy_1 = vx_2 = vy_2 = 2$

1	Gradient descent:
2	
3	w:

```

4  -78.1766393662
5    6.7233419236
6   11.2430677919
7
8  Confusion Matrix:
9              Predict cluster 1 Predict cluster 2
10 Is cluster 1          50           0
11 Is cluster 2           0          50
12
13 Sensitivity (Successfully predict cluster 1): 1.00000
14 Specificity (Successfully predict cluster 2): 1.00000
15
16 -----
17 Newton's method:
18
19 w:
20 -118.3601516394
21   8.7747332848
22  10.1954120077
23
24 Confusion Matrix:
25              Predict cluster 1 Predict cluster 2
26 Is cluster 1          50           0
27 Is cluster 2           0          50
28
29 Sensitivity (Successfully predict cluster 1): 1.00000
30 Specificity (Successfully predict cluster 2): 1.00000

```

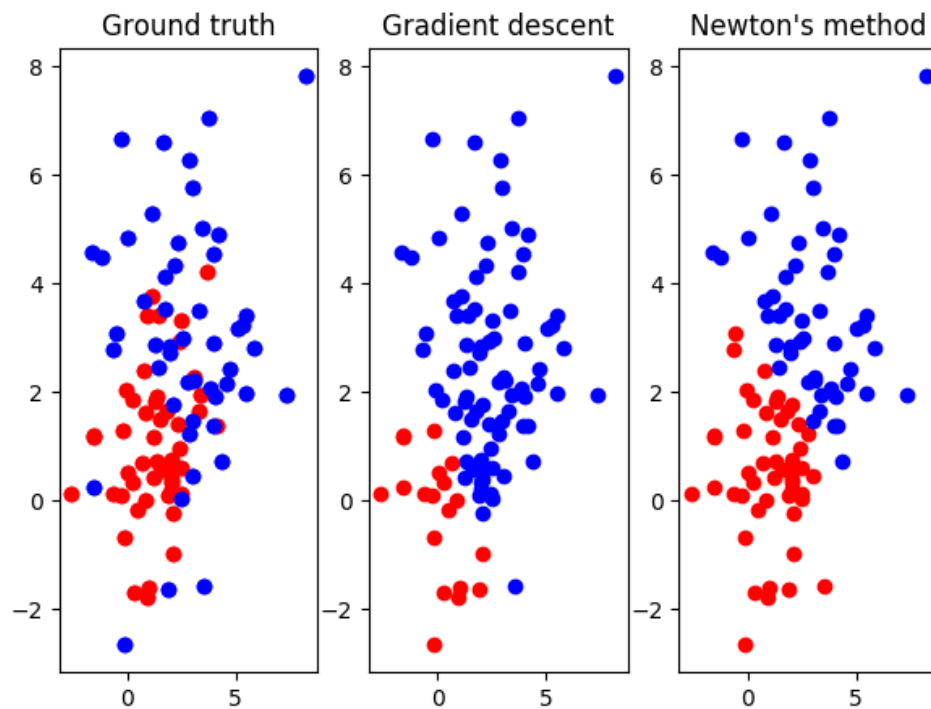


- Case 2: $N = 50, mx_1 = my_1 = 1, mx_2 = my_2 = 3, vx_1 = vy_1 = 2, vx_2 = vy_2 = 4$

```

1 Gradient descent:
2
3 w:
4 -71.1902536008
5 46.0123814025
6 54.6803199701
7
8 Confusion Matrix:
9           Predict cluster 1 Predict cluster 2
10 Is cluster 1          16          34
11 Is cluster 2           3          47
12
13 Sensitivity (Successfully predict cluster 1): 0.32000
14 Specificity (Successfully predict cluster 2): 0.94000
15
16 -----
17 Newton's method:
18
19 w:
20 -1.9045831451
21 0.3940876974
22 0.5695243849
23
24 Confusion Matrix:
25           Predict cluster 1 Predict cluster 2
26 Is cluster 1          40          10
27 Is cluster 2          10          40
28
29 Sensitivity (Successfully predict cluster 1): 0.80000
30 Specificity (Successfully predict cluster 2): 0.80000

```



2. EM algorithm

- Input: [MNIST training](#) data and label sets. (Same as HW02)
 - Function:
 1. Binning the gray level value into **two bins**. Treating all pixels as random variables following Bernoulli distributions. Note that each pixel follows a different Binomial distribution independent to others.
 2. Use EM algorithm to cluster each image into ten groups. You should come up with a reasonable rule to determine convergence. (a simple run out of the loop should be used as the ultimatum)
 - Output:
 1. For each digit, output a confusion matrix and the **sensitivity** and **specificity** of the clustering applied to the training data.
 2. Print out the imagination of numbers in your classifier
 - Just like before, about the details please refer to HW02
 - Hint: The algorithm is a kind of unsupervised learning, so the labels are not used during training. But you can use these labels to help you to figure out which class belongs to which number.
- In other words, you should find a way to assign label to each class which you classified **before you compute the confusion matrix**
- Sample input & output (**for reference only**)

```

1 class 0:
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

[illegible]

```
53 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
54 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
56 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
57 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
58 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
59 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
60
61 ... all other unlabeled imagination of numbers goes here ...
62
63 class 9:
64 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
65 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
66 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
67 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
68 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
69 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
70 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0
71 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
72 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
73 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
74 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
75 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
76 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0
77 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
78 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
79 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
80 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
81 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
82 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0
83 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
84 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0
85 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0
86 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
87 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
88 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
89 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
90 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
91 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
92
93 No. of Iteration: 1, Difference: 3176.579389514846
94
95 -----
96
97 class 0:
98 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
99 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
101 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

[illegible]

[illegible]

158

```
159 No. of Iteration: 10, Difference: 19.89546432548733
```

160

161

162

163

```
164 labeled class 0:
```

[illegible]

193

```
194 labeled class 1:
```

[illegible]

[illegible]

```

249 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
250 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
251 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
252 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
253 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
254 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
255
256 -----
257
258 Confusion Matrix 0:
259             Predict number 0 Predict not number 0
260 Is number 0           3023           2900
261 Isn't number 0        113           53964
262
263 Sensitivity (Successfully predict number 0)      : 0.51038
264 Specificity (Successfully predict not number 0): 0.99791
265
266 -----
267
268 Confusion Matrix 1:
269             Predict number 1 Predict not number 1
270 Is number 1           5986           756
271 Isn't number 1        800           52458
272
273 Sensitivity (Successfully predict number 1)      : 0.88787
274 Specificity (Successfully predict not number 1): 0.98498
275
276 -----
277
278 ... all other confusion matrix goes here ...
279
280 -----
281
282 Confusion Matrix 9:
283             Predict number 9 Predict not number 9
284 Is number 9           2718           3231
285 Isn't number 9        5147           48904
286
287 Sensitivity (Successfully predict number 9)      : 0.45688
288 Specificity (Successfully predict not number 9): 0.90478
289
290 Total iteration to converge: 10
291 Total error rate: 0.5081666666666667

```