

Machine Learning Homework 4

Logistic Regression

- Input
 - n (the number of data points)
 - $mx_1, vx_1, my_1, vy_1, mx_2, vx_2, my_2, vy_2$ (m : mean, v : variance)
- Function
 - Use the Gaussian random number generator in homework 3
 - Generate n data points: $D1 = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where x and y are independently sampled from $N(mx_1, vx_1)$ and $N(my_1, vy_1)$ respectively
 - Generate n data points: $D2 = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where x and y are independently sampled from $N(mx_2, vx_2)$ and $N(my_2, vy_2)$ respectively
 - 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
 - The confusion matrix and the **sensitivity** and **specificity** of the logistic regression applied to the training data D
 - Visualization
 - Plot the ground truth
 - Plot the predict result
 - Gradient descent
 - Newton's method
- Sample input & output (**for reference only**)
 - Case 1: $n = 50, mx_1 = my_1 = 1, mx_2 = my_2 = 10, vx_1 = vy_1 = vx_2 = vy_2 = 2$

Gradient descent:

w:

-78.1766393662
6.7233419236
11.2430677919

Confusion Matrix:

	Predict cluster 1	Predict cluster 2
Is cluster 1	50	0
Is cluster 2	0	50

Sensitivity (Successfully predict cluster 1): 1.00000

Specificity (Successfully predict cluster 2): 1.00000

Newton's method:

w:

-118.3601516394

8.7747332848

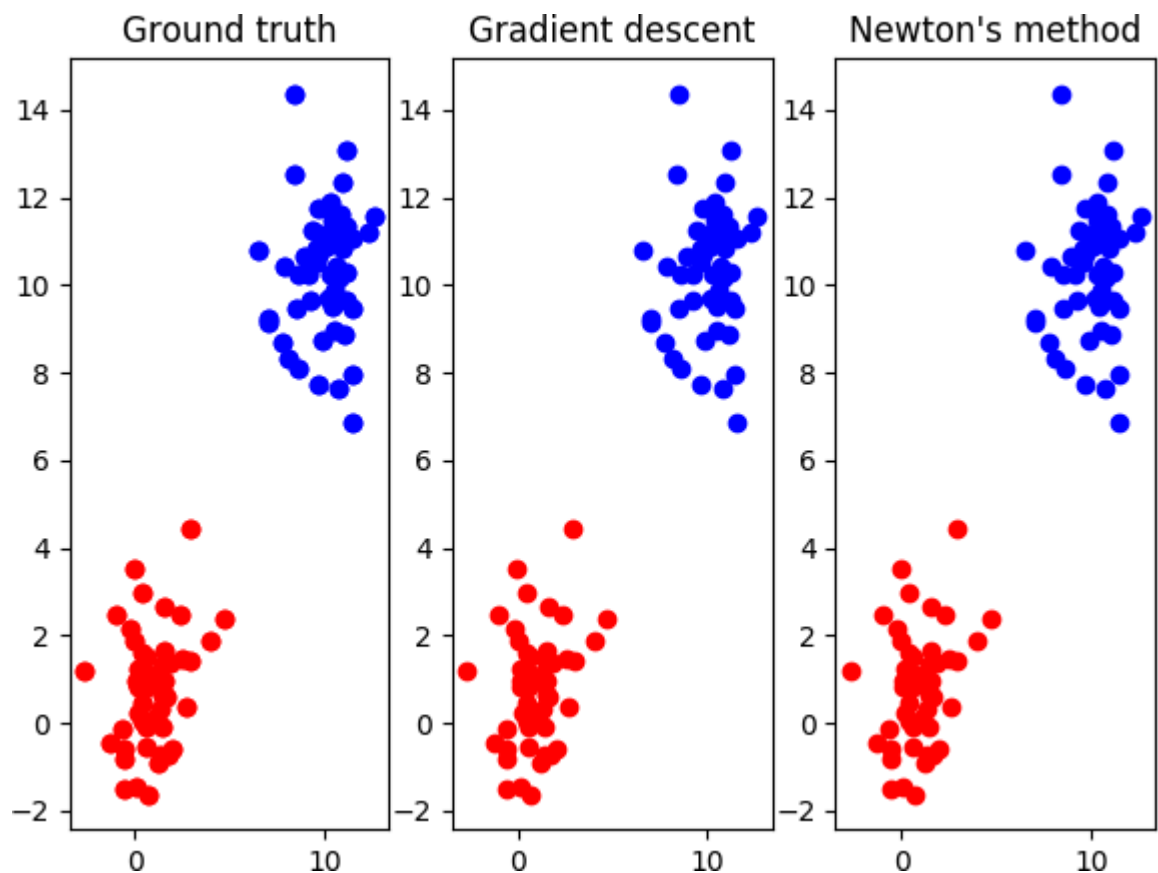
10.1954120077

Confusion Matrix:

	Predict cluster 1	Predict cluster 2
Is cluster 1	50	0
Is cluster 2	0	50

Sensitivity (Successfully predict cluster 1): 1.00000

Specificity (Successfully predict cluster 2): 1.00000



- Case 2: $n = 50$, $m x_1 = m y_1 = 1$, $m x_2 = m y_2 = 3$, $v x_1 = v y_1 = 2$, $v x_2 = v y_2 = 4$

Gradient descent:

w:

-71.1902536008

46.0123814025

54.6803199701

Confusion Matrix:

	Predict cluster 1	Predict cluster 2
Is cluster 1	16	34
Is cluster 2	3	47

Sensitivity (Successfully predict cluster 1): 0.32000

Specificity (Successfully predict cluster 2): 0.94000

Newton's method:

w:

-1.9045831451

0.3940876974

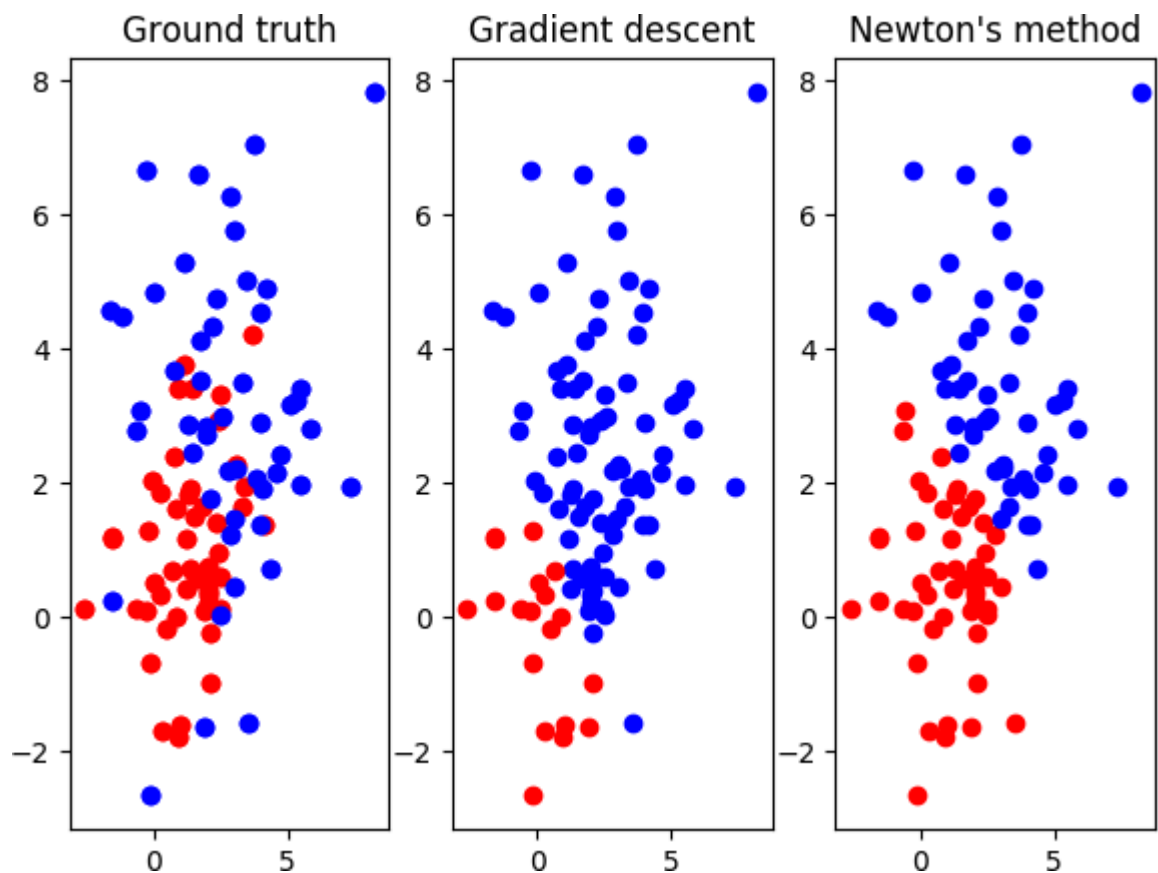
0.5695243849

Confusion Matrix:

	Predict cluster 1	Predict cluster 2
Is cluster 1	40	10
Is cluster 2	10	40

Sensitivity (Successfully predict cluster 1): 0.80000

Specificity (Successfully predict cluster 2): 0.80000



EM Algorithm

- Input
 - MNIST training data and label sets. (Same as HW2)
- Function
 - 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.
 - 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
 - For each digit, output a confusion matrix and the **sensitivity** and **specificity** of the clustering applied to the training data.
 - Print out the imagination of numbers in your classifier
 - Just like before, about the details please refer to HW2
- 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**)

[illegible]

[illegible]

class 9:

[illegible]

Sensitivity (Successfully predict number 0): 0.51038
Specificity (Successfully predict not number 0): 0.99791

Confusion Matrix 1:

	Predict number 1	Predict not number 1
Is number 1	5986	756
Isn't number 1	800	52458

Sensitivity (Successfully predict number 1): 0.88787
Specificity (Successfully predict not number 1): 0.98498

... all other confusion matrices go here ...

Confusion Matrix 9:

	Predict number 9	Predict not number 9
Is number 9	2718	3231
Isn't number 9	5147	48904

Sensitivity (Successfully predict number 9): 0.45688
Specificity (Successfully predict not number 9): 0.90478

Total iteration to converge: 10
Total error rate: 0.508166666666667

Mathematical Derivation

- Given two coins (C_0, C_1) and in each trial, a certain coin is chosen and tossed for three times. There are totally three trial outcomes: $\{HHH, HHT, TTT\}$. Note that the chance of choosing C_0 is k , the chance of C_0 showing H is P_0 , and the chance of C_1 showing H is P_1 . Use EM algorithm to update the estimates of the parameters for just one round. You should show the process step by step with the initial values of parameters: $k = 0.5, P_0 = 0.6, P_1 = 0.1$
- Notes
 - During the demo, you will be required to explain how you arrived at your calculations
 - Upload the handwritten file to e3 (PDF or any image format)

