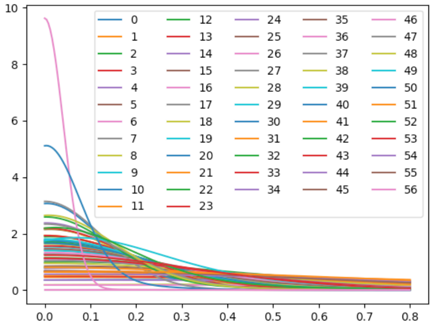
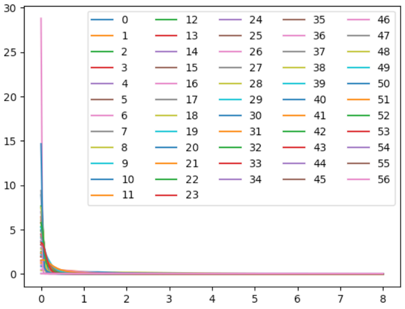
Hot Topics in Machine Learning (HWS17) Assignment 1: Logistic Regression

Chen Zongqi

1.Dataset Statistics

a) From this figure, we can see that there is one highest feature (pink line, 46) which is word edu and the second highest one is feature (blue, 50) which is character "(". All of these features are positive half normal distribution and their mean is when x = 0. Most of them has small variance and after x = 0.6, density values are very small.





b) We use z-score to do the normalization. Firstly, we calculate each feature mean and standard variance and then use the example value minus feature mean then divided by feature standard variance. Finally, we get the z-score.

c) Compared with the figure in a), the different between each feature line is smaller than without doing z-score normalization. The highest density value is under 1.90 and far smaller than the number in a) near 9. Also in x range(5, 6), there is a pink tiny wave in Figure c-2.

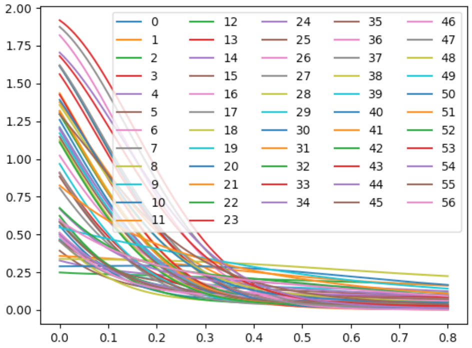


Figure 1-c-1

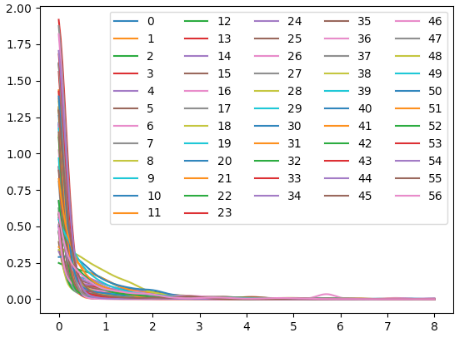


Figure 1-c-2

2. Maximum Likelihood Estimation

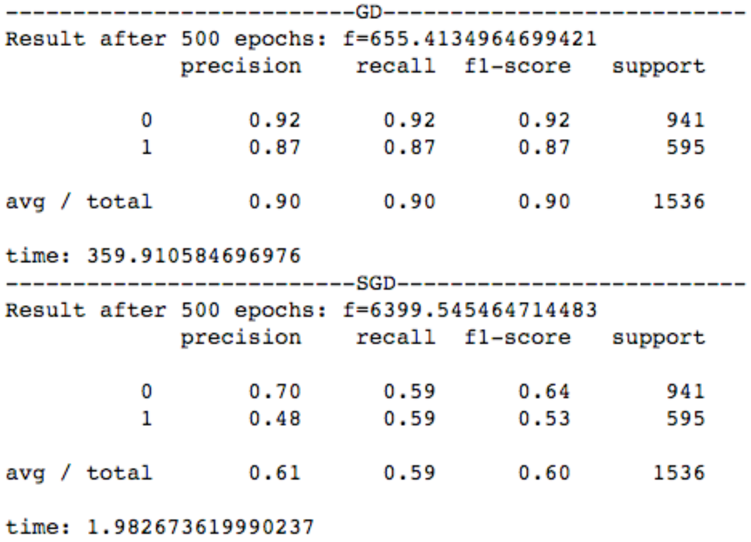
a) If maximum likelihood is calculated by z-score normalized data, then no matter how rescaling or shifting operates, the ML related value doesn't change, because rescaling and shifting do not change data intern relationship. The reason we need to use z-score normalization is that z-score can keep the data original intern relationship and also change the value around 0, for example if the dataset has 1 million data and each data is large number, may caused memory issue, then after z-score normalization may solve this problem.

b) Done

c) Done, no loop!

d) Done

e) Compare with GD and SGD, the first different is speed. For GD is far slower than SGD and GD till the end still not converges, but SGD are faster to converge and faster calculate, but its precision and F score is too low. See Figure 2-e



```
--------------------------GD---------------------------
Result after 500 epochs: f=655.4134964699421
             precision    recall  f1-score   support

          0       0.92      0.92      0.92       941
          1       0.87      0.87      0.87       595

avg / total       0.90      0.90      0.90      1536

time: 359.910584696976
--------------------------SGD--------------------------
Result after 500 epochs: f=6399.545464714483
             precision    recall  f1-score   support

          0       0.70      0.59      0.64       941
          1       0.48      0.59      0.53       595

avg / total       0.61      0.59      0.60      1536

time: 1.982673619990237
```

Figure 2-e

3. Prediction

Using GD leaning weights (wz_gd), we get the Figure 3 below, for precision and recall curve this curve is quite good, when recall nearly equals 0 and precision nearly equals 1.0. When recall nearly equals 1.0, precision reaches 0.42.

For weight vector, I use argmax and argmin to pick the highest and lowest weight value, see Figure 3-2. For GD I find the highest weight value word is 3d and lowest weight value word is hp. “3d” may be the characteristic of spam and normal email seldom to use it. I also compare GD with SGD and I find quite interesting thing is that SGD considers word hp is most important to detect spam. However, SGD is not so accuracy, so for the result of GD, I think it is intuitive.

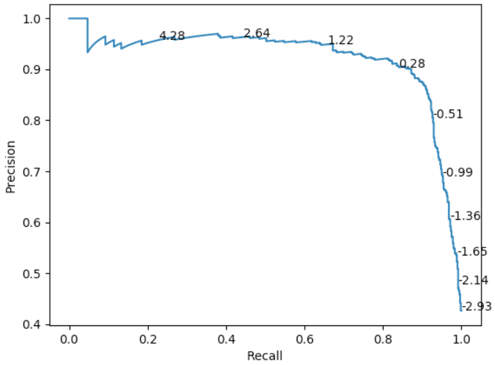
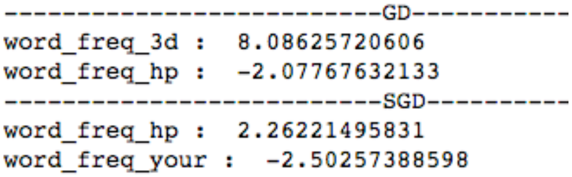


Figure 3-1

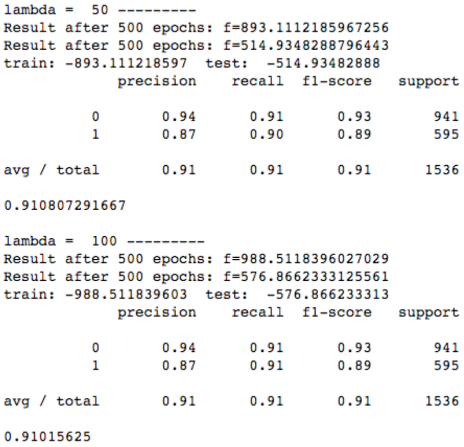


```
--------------------------GD-----------
word_freq_3d :  8.08625720606
word_freq_hp :  -2.07767632133
--------------------------SGD----------
word_freq_hp :  2.26221495831
word_freq_your :  -2.50257388598
```

Figure 3-2

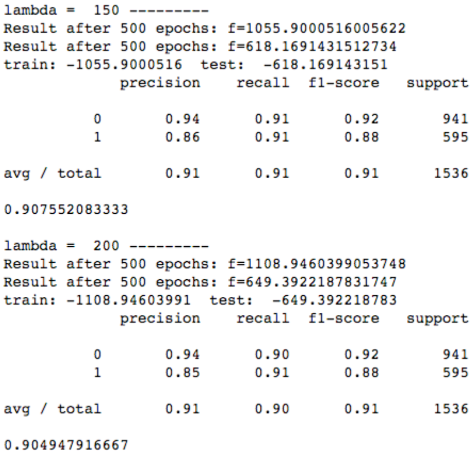
4. Maximum Aposteriori Estimation

a) Done
b) I set four groups, which lambda equals 50, 100, 150 and 200. See the results below, Figure 3-1 and Figure 3-2. I find when lambda increases from 50 to 200, its accuracy decreases from 0.910 to 0.904. I think the reason is that bigger lambda can avoid the model over fit training data. So that’s why when lambda increased and its accuracy decreased.



```
lambda =  50 ---------
Result after 500 epochs: f=893.1112185967256
Result after 500 epochs: f=514.9348288796443
train: -893.111218597  test:  -514.93482888
             precision    recall  f1-score   support

          0       0.94      0.91      0.93       941
          1       0.87      0.90      0.89       595

avg / total       0.91      0.91      0.91      1536

0.910807291667

lambda =  100 ---------
Result after 500 epochs: f=988.5118396027029
Result after 500 epochs: f=576.8662333125561
train: -988.511839603  test:  -576.866233313
             precision    recall  f1-score   support

          0       0.94      0.91      0.93       941
          1       0.87      0.91      0.89       595

avg / total       0.91      0.91      0.91      1536

0.91015625
```

Figure 3-1



```
lambda =  150 ---------
Result after 500 epochs: f=1055.9000516005622
Result after 500 epochs: f=618.1691431512734
train: -1055.9000516  test:  -618.169143151
             precision    recall  f1-score   support

          0       0.94      0.91      0.92       941
          1       0.86      0.91      0.88       595

avg / total       0.91      0.91      0.91      1536

0.907552083333

lambda =  200 ---------
Result after 500 epochs: f=1108.9460399053748
Result after 500 epochs: f=649.3922187831747
train: -1108.94603991  test:  -649.392218783
             precision    recall  f1-score   support

          0       0.94      0.90      0.92       941
          1       0.85      0.91      0.88       595

avg / total       0.91      0.90      0.91      1536

0.904947916667
```

Figure 3-2

c) I use wz_gd_l2 to plus 1 and times 100 as lambda to calculate. And find accuracy decreased.