comp540 Homework2

Xiaoye Steven Sun (xs6), Wanyi Ye (wy13)

February 2nd 2018

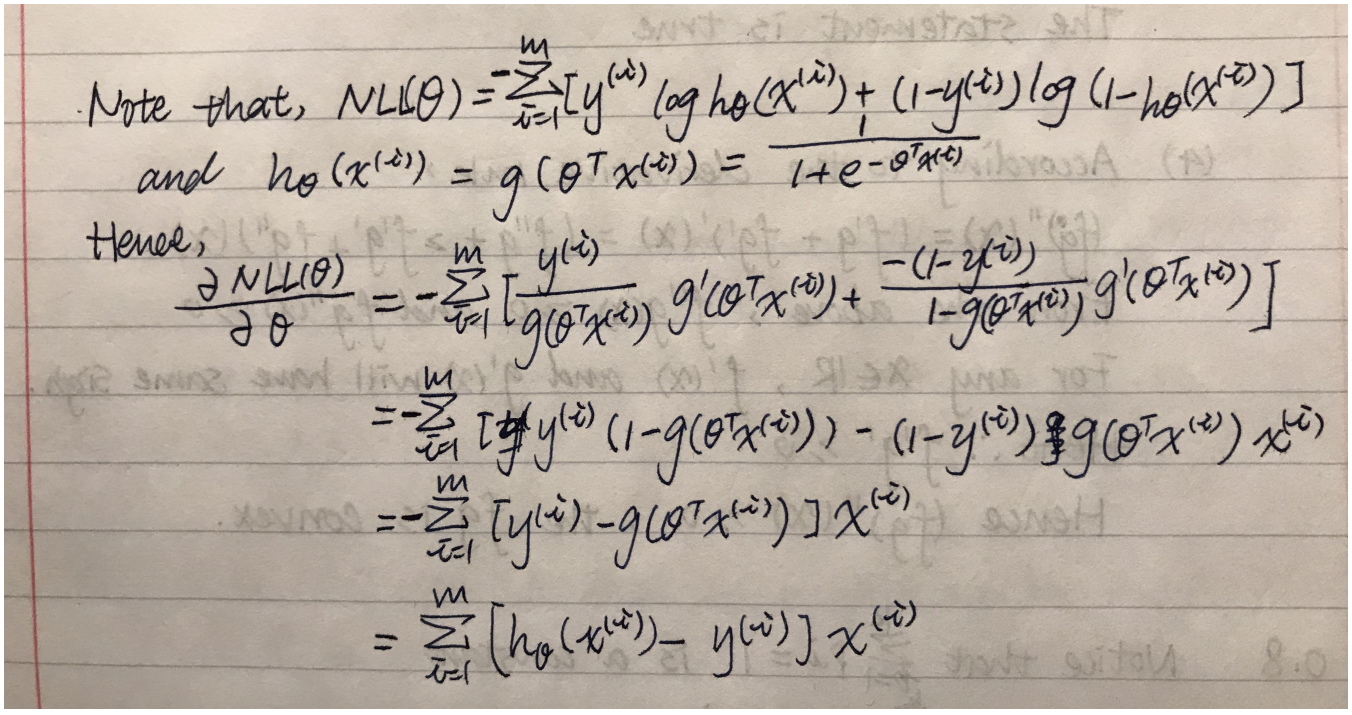
1 Gradient and Hessian of *NLL(θ)* for logistic regression

1.1 Proof

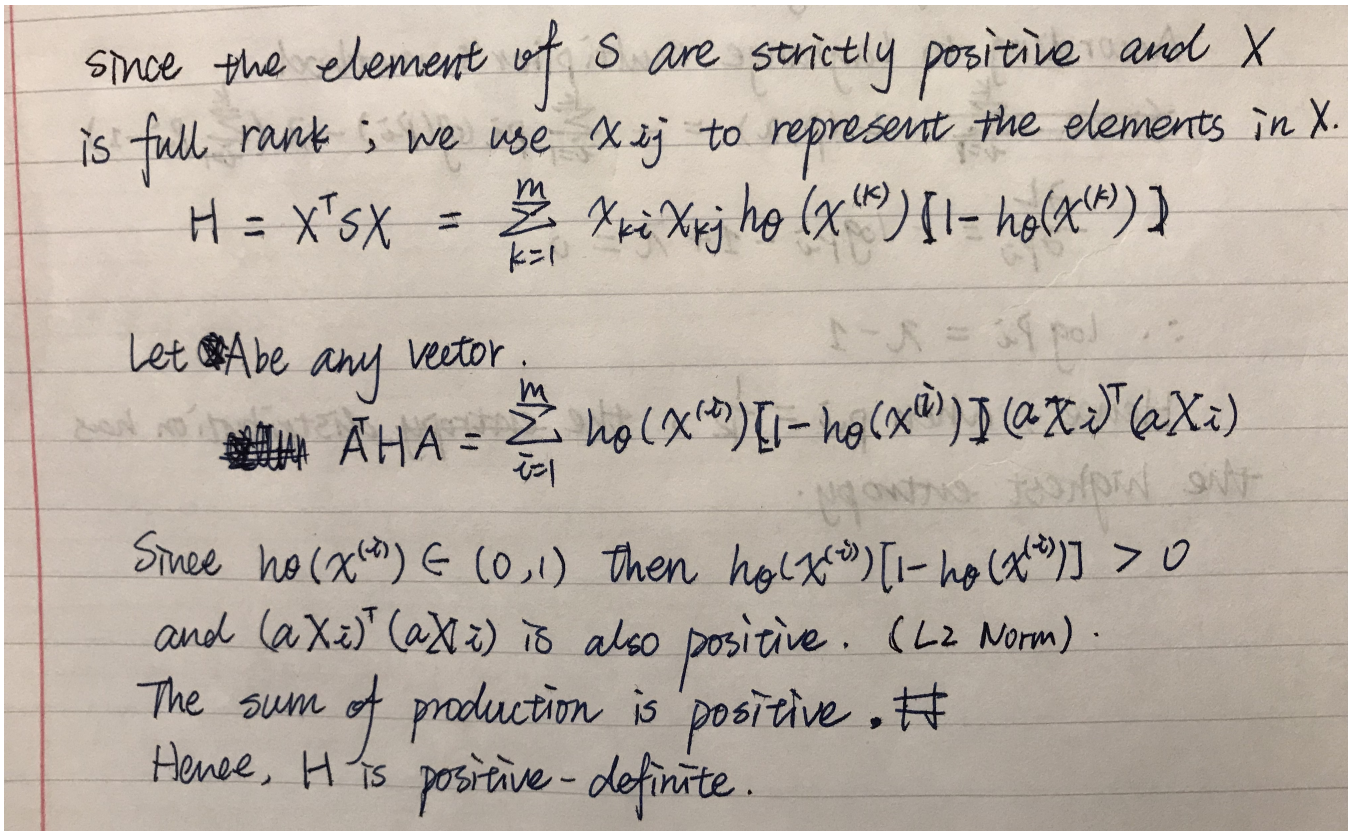
Calculus gives us that $g'(z) = \frac{e^{-z}}{(1+e^{-z})^2}$. From there the result follows by simple algebraic manipulation:

$$\begin{aligned} g'(z) &= \frac{e^{-z}}{(1+e^{-z})^2} \\ &= (\frac{e^{-z}}{1+e^{-z}})(\frac{1}{1+e^{-z}}) \\ &= (\frac{e^{-z}}{1+e^{-z}} + \frac{1}{1+e^{-z}} - \frac{1}{1+e^{-z}})(\frac{1}{1+e^{-z}}) \\ &= (\frac{1+e^{-z}}{1+e^{-z}} - \frac{1}{1+e^{-z}})(\frac{1}{1+e^{-z}}) \\ &= g(z)(1-g(z)) \end{aligned}$$

1.2 Proof



1.3 Proof



2 Properties of L2 regularized logistic regression

2.1 False

Note that

$$\begin{aligned} J(\theta) &= \sum_{i=1}^{m}[y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2 \\ &= NLL(\theta) + \frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2 \end{aligned}$$

Since we already proved that the Hessian of $NLL(\theta)$ is positive definite (Problem 1.3)
When $\lambda >= 1$, $J(\theta)$ has only one global optimal solution.

2.2 False

In general, the L2 Norm is smooth. Unlike the L1 Norm which the tend to be sparse.

2.3 True

If θ separates the data, and the data set is positive, multiplied by positive number will only increase the θ.

2.4 False

Increasing the λ will increase the effect of regularization, which, will actually decrease the fitting. But if a data set is linearly separable, increasing the λ may not affect the θ^*. In this case, the first term should remain the same.

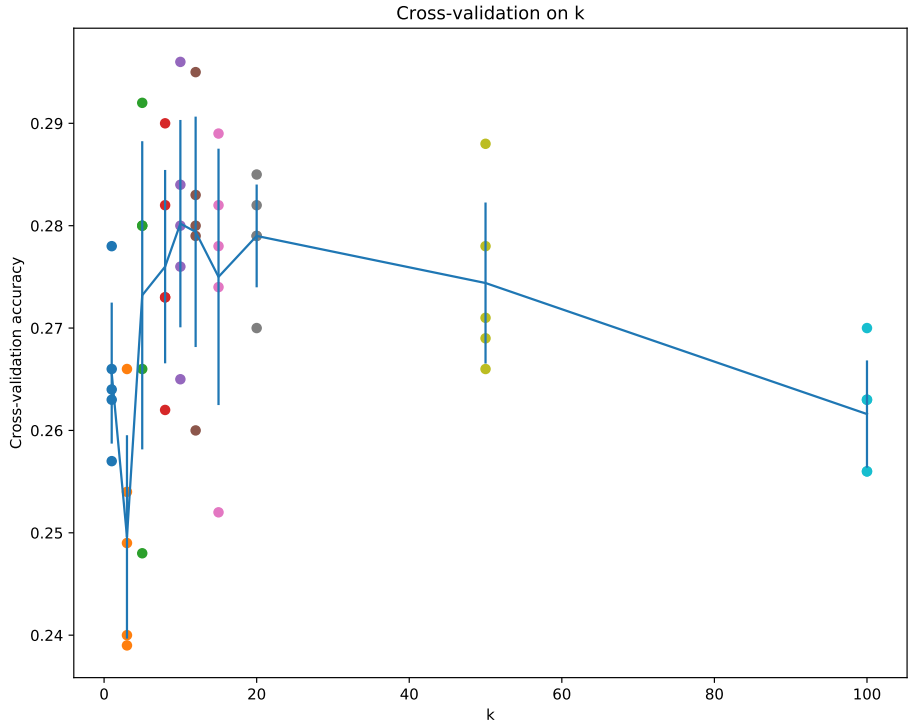


Figure 1: Accuracy with different k values

3 Implementing a k-nearest-neighbor classifier

See `k_nearest_neighbor.py` and `knn.ipynb`.
Two loop version took 21.048373 seconds.
One loop version took 57.990544 seconds.
No loop version took 0.295317 seconds.

See Figure 1 for the best k. Best k=10, Got 141 / 500 correct => accuracy: 0.282000

4 Implementing logistic regression

4.1 Logistic regression

See `logreg.ipynb`, `utils.py` and `logistic_regressor.py`
Loss on all-zeros theta vector (should be around 0.693) = 0.69314718056
Gradient of loss wrt all-zeros theta vector (should be around [-0.1, -12.01, -11.26]) = [-0.1 -12.00921659 -11.26284221]
For a student with 45 on exam 1 and 85 on exam 2, the probability of admission = 0.776246678481
Accuracy on the training set = 89

See Figure 2 for the decision boundary.

4.2 Regularized logistic regression

See `logreg_reg.ipynb` and `logistic_regressor.py`.

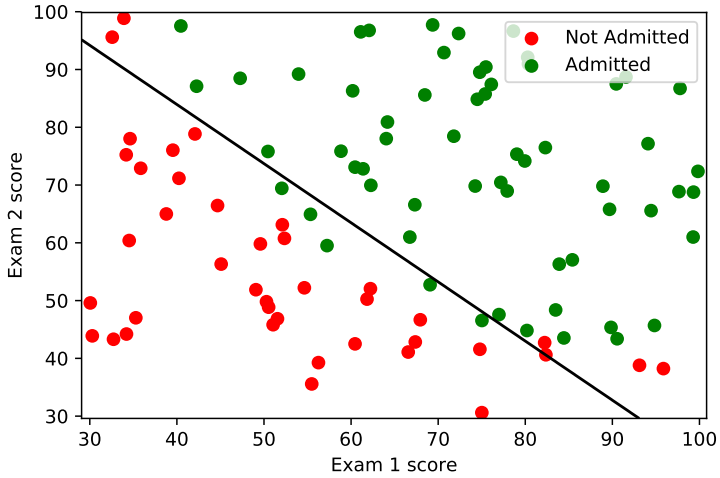
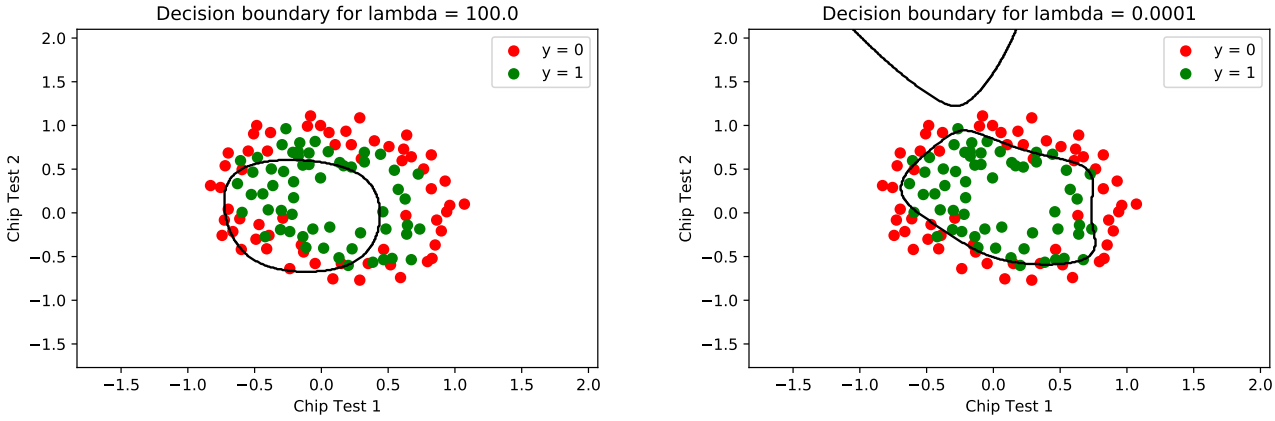


Figure 2: Decision boundary



(a) under-fitted model, λ=100.0

(b) over-fitted model, λ=0.0001

Figure 3: Examples of under-fitted and over-fitted model

See Figure 3 for the examples of underfitted and overfitted models.

See Figure 4. As λ decreases ($\log(c)$ increases), L1 regularization has more zero coefficients and shows smaller loss comparing with L2 (Table 1). In general, L1 has more zero coefficients than L2.

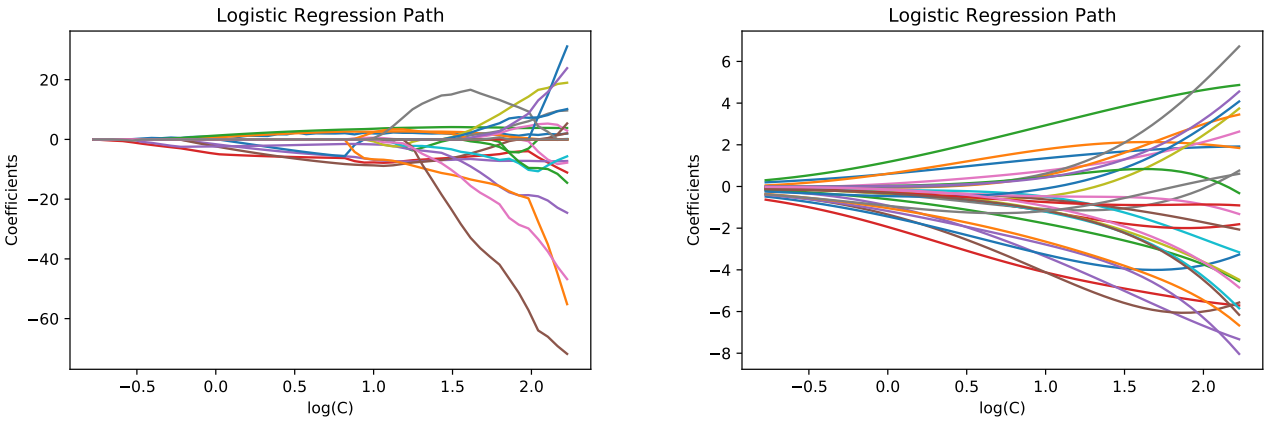
4.3 Logistic regression for spam classification

See `logreg_spam.ipynb` and `utils.py`.

Table 2 shows the L1 and L2 accuracy under different data pre-processing method. `logt` shows the best performance while `std` and `bin` are similar. This is because the data value distributes in very wide range so that taking the `log` makes the data to be distributed more uniformly, which is good for classification. `bin` is not as good as `logt` since it loses too much information from the data. L1 usually requires larger λ. The coefficients in L1 are more sparse.

λ	L1 loss	L2 loss
100	0.69314718056	0.68061702032
1	0.438134830783	0.46843403006
0.01	0.291083349561	0.316699562052
0.0001	0.248154730959	0.286727466561

Table 1: L1 and L2 loss comparison under different λ



(a) L1 regularization path

(b) L2 regularization path

Figure 4: L1 and L2 regularization regularization paths

pre-processing	L1 accuracy	L1 best λ	L2 accuracy	L2 best λ
std	0.921875	4.6	0.9296875	0.1
logt	0.944010416667	1.6	0.943359375	0.6
bin	0.92578125	3.6	0.928385416667	1.6

Table 2: L1 and L2 accuracy comparison under different pre-processing scheme