COMP 540 Statistical Machine Learning HW2

Xiang Zhou (xz58) and Guangyuan Yu (gy12)

February 6, 2017

1 Gradient and Hessian of $NLL(\theta)$ for logistic regression

1.1

- Given $g(z) = \frac{1}{1+e^{-z}}$

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

1.2

- We know $NLL(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)\right)$,
  $\frac{\partial}{\partial\theta}NLL(\theta) = \frac{\partial NLL(\theta)}{\partial h_\theta(x^{(i)})}\frac{\partial h_\theta(x^{(i)})}{\partial\theta}$,
  $h_\theta(x^{(i)}) = g(\theta^T x)$
  and given $\frac{\partial g(z)}{\partial z} = g(z)\cdot(1-g(z))$ from above, we have:

$$\begin{aligned}\frac{\partial}{\partial\theta}NLL(\theta) &= -\frac{1}{m}\sum_{i=1}^{m}\left(\frac{y^{(i)}}{h_\theta(x^{(i)})} - \frac{(1-y^{(i)})}{1-h_\theta(x^{(i)})}\right)x^{(i)}h_\theta(x^{(i)})(1-h_\theta(x^{(i)})) \\ &= -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}(1-h_\theta(x^{(i)})) - (1-y^{(i)})h_\theta(x^{(i)}))x^{(i)} \\ &= \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}\end{aligned}$$

1.3

- To prove that H is positive definite, we want to prove that for any nonzero vector $\mathbf{a}$, $\mathbf{a}^T H\mathbf{a} > 0$.

$$\begin{aligned}\mathbf{a}^T H\mathbf{a} &= \mathbf{a}^T X^T S X\mathbf{a} \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{m} a_i x_{ki}(h_\theta(x^{(i)}))(1-h_\theta(x^{(i)}))x_{kj}a_j \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{m} a_i x_{ki}x_{kj}a_j(h_\theta(x^{(i)}))(1-h_\theta(x^{(i)}))\end{aligned}$$

since $i = j, a_i = a_j, x_{ki} = x_{kj}$

$$\mathbf{a}^T H\mathbf{a} = \sum_{i=1}^{d}\sum_{k=1}^{m} a_i^2 x_{ki}^2(h_\theta(x^{(i)}))(1-h_\theta(x^{(i)}))$$

Given $(a_i x_{ki})^2 > 0, (h_\theta(x^{(i)}))(1-h_\theta(x^{(i)}) > 0$, we can prove $\mathbf{a}^T H\mathbf{a} > 0$.
Thus, we have proved that H is positive definite.

2 Regularizing logistic regression

2.1

-

$$\begin{aligned}\theta_{MLE} &= \arg\max_\theta \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) \\ &= \arg\max_\theta \prod_{i=1}^{m} g(\theta^T x^{(i)})\end{aligned}$$

Therefore, we have:

$$\begin{aligned}\prod_{i=1}^{m} g(\theta_{MLE}^T x^{(i)}) &\geq \prod_{i=1}^{m} g(\theta^T x^{(i)}) \\ \prod_{i=1}^{m} \frac{g(\theta_{MLE}^T x^{(i)})}{g(\theta^T x^{(i)})} &\geq 1 \\ \prod_{i=1}^{m} \frac{g(\theta_{MLE}^T x^{(i)})}{g(\theta_{MAP}^T x^{(i)})} &\geq 1\end{aligned}$$

Similarly, we have:

$$\prod_{i=1}^{m} P(\theta_{MAP}) g(\theta_{MAP}^T x^{(i)}) \geq \prod_{i=1}^{m} P(\theta) g(\theta^T x^{(i)})$$

$$\frac{P(\theta_{MAP})}{P(\theta)} \geq \prod_{i=1}^{m} \frac{g(\theta^T x^{(i)})}{g(\theta^T x^{(i)})}$$

$$\frac{P(\theta_{MAP})}{P(\theta_{MLE})} \geq \prod_{i=1}^{m} \frac{g(\theta_{MLE}^T x^{(i)})}{g(\theta_{MAP}^T x^{(i)})} \geq 1$$

Thus, $P(\theta_{MAP}) \geq P(\theta_{MLE})$. Since, $P(\theta)$ is $N(0, \alpha^2 I)$,

$$\frac{1}{\sqrt{2\pi}^{-k}\sqrt{|\alpha^2 I|}} exp(-\frac{\theta_{MAP}^T \theta_{MAP}}{2\alpha^2}) \geq \frac{1}{\sqrt{2\pi}^{-k}\sqrt{|\alpha^2 I|}} exp(-\frac{\theta_{MLE}^T \theta_{MLE}}{2\alpha^2})$$

$$-\theta_{MAP}^T \theta_{MAP} \geq \theta_{MLE}^T \theta_{MLE}$$

$$\|\theta_{MAP}\|_2 \leq \|\theta_{MLE}\|_2$$

The proof is done.

3 Inplementing a k-nearest-neighbot classifier

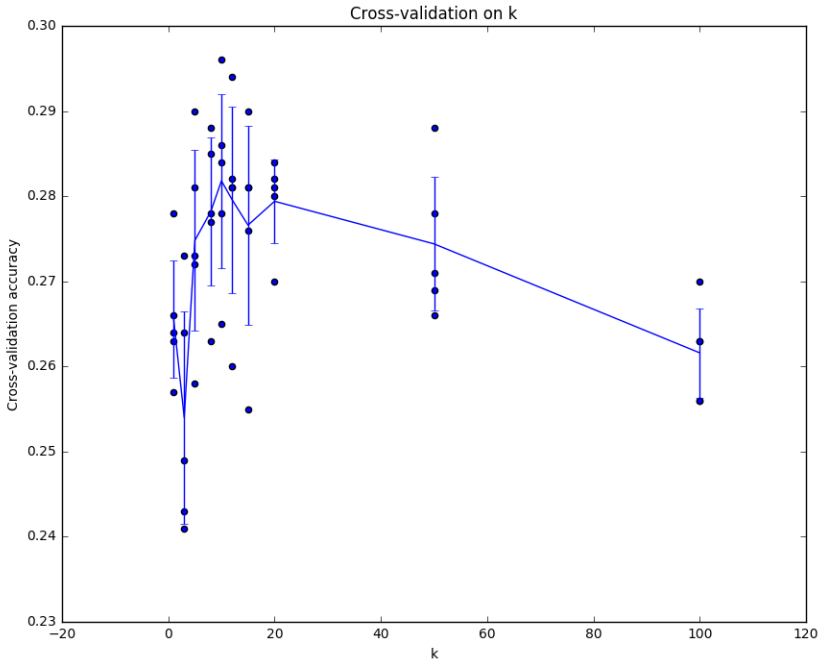
When $k = 1$, we get an *accuracy* $= 137/500 = 0.274000$. When $k = 5$, we get an *accuracy* $= 142/500 = 0.284000$. When we compare one-loop with two-loop, the matrix difference is zero. When we compare no-loop with two-loop, the matrix difference is zero. bright rows: It means this test data has low similarity with the majority of the training data. bright columns : It means this training data has low similarity with the majority of the test data.

3.1 Speeding up distance computations

Two loop version took 37.736882 seconds. One loop version took 55.912858 seconds. No loop version took 0.340704 seconds.

3.2 Choosing k by cross-validation

Figure 1: Choosing k by crossvalidation on the CIFAR-10 dataset

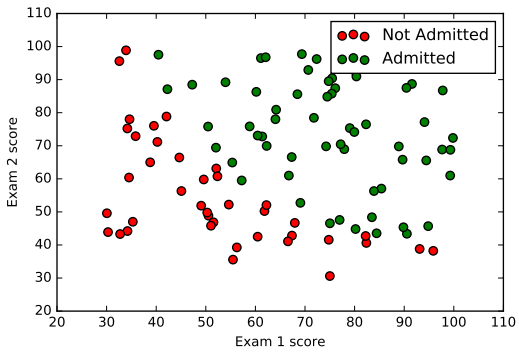


We think the best k is $k = 3$ and we get $accuracy = 139/500 = 0.278000$

4 Implementing logistic regression

4.1 Visualizing the dataset

Figure 2: The training data



4.2 3A1. Implementing logistic regression the sigmoid funciton

Yes we had tested sigmoid function, it is correct.

4.3 3A2 Cost function and gradient of logistic regression

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]
Optimization terminated successfully.
Current function value: 0.203498
Iterations: 19
Function evaluations: 20
Gradient evaluations: 20
Theta found by fmin_bfgs: [−25.160569450.206229630.20146073]
Final loss = 0.203497702351

4.4 Learning parameters using fmin_bfgs

Figure 3: The decision boundary

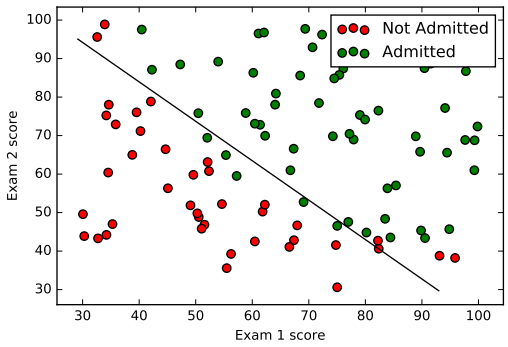
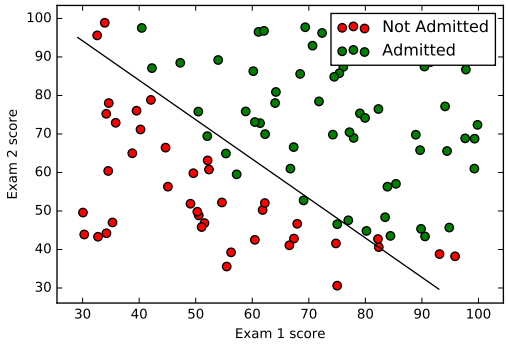


Figure 4: The decision boundary from sklearn



Theta found by sklearn: [[-25.15293066 0.20616459 0.20140349]]

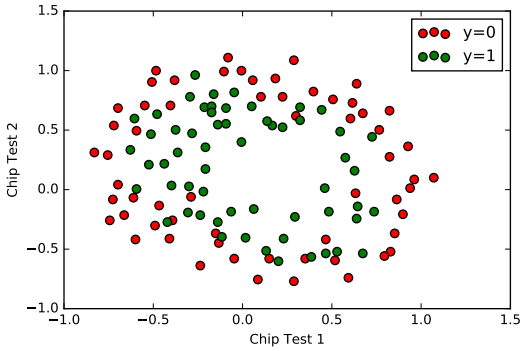
4.5 3A3 Prediction using a logistic regression model

The student with 45/85 score will be admitted with a probability of 0.776246678481.
The accuracy on the training set is 0.89.

5 3part B Regularized logistic regression

5.1 visualizing the data

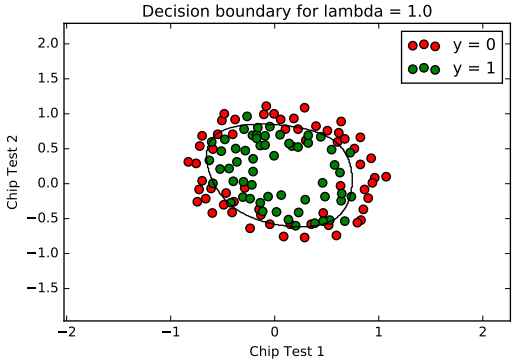
Figure 5: Plot of training data



5.2 3B1 Cost function and gradient for regularized logistic

Accuracy on the training set = 0.830508474576

Figure 6: Traning data with decision boundary for lambda=1



5.3 3B2 prediction using the model

Accuracy on the training set = 0.830508474576

5.4 3B3 varing lambda

Figure 7: lambda=0.1

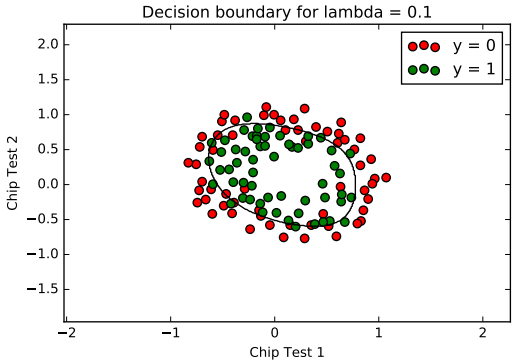
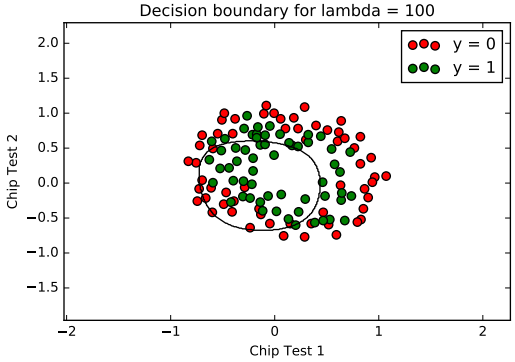


Figure 8: lambda=100



5.5 3B4 Exploring L1 and L2 penalized logistic regression

Figure 9: L1sklearn with reg=1

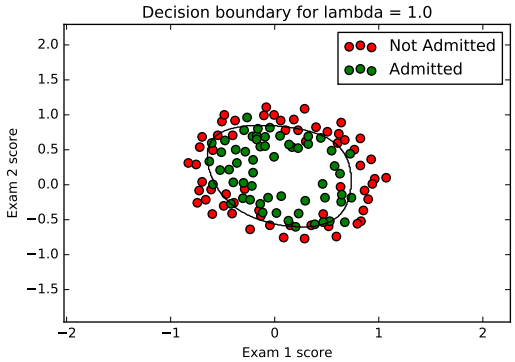


Figure 10: L1sklearn with reg=0.1

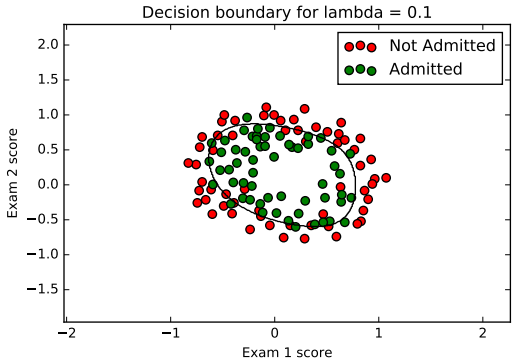
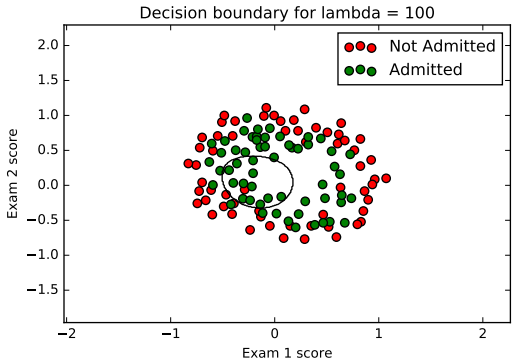


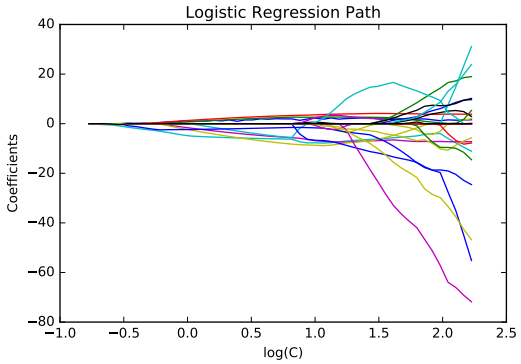
Figure 11: L1sklearn with reg=100



5.5.1 with reg=100

L2sklearn
Theta found by sklearn with L2 reg: [0.00468635 -0.01726848 0.0064196 -0.05402665 -0.01327551 -0.03727145 -0.01821195 -0.00761037 -0.00885306 -0.02224573 -0.04288369 -0.00238585 -0.01393196 -0.00354828 -0.04072376 -0.02078577 -0.00467203 -0.00354978 -0.00624894 -0.00500393 -0.03153159 -0.03381515 -0.00108319 -0.00694192 -0.0003945 -0.00788595 -0.00157683 -0.04058858] Loss with sklearn theta: 0.68061702032
L1
Theta found by sklearn with L1 reg: [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. 0. 0.]
Loss with sklearn theta: 0.69314718056

Figure 12: path

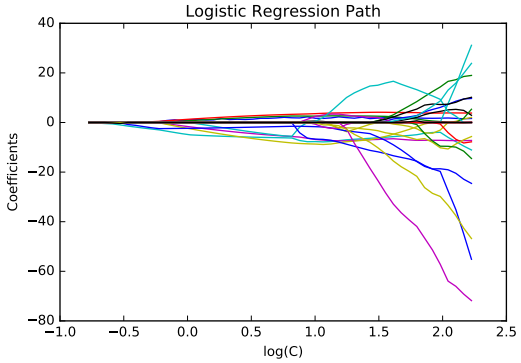


5.5.2 with reg=1

L2sklearn
Theta found by sklearn with L2 reg: [1.1421394 0.60141117 1.16712554 -1.87160974 -0.91574144 -1.26966693 0.12658629 -0.3686536 -0.34511687 -0.17368655 -1.42387465 -0.04870064 -0.60646669 -0.26935562 -1.16303832 -0.24327026 -0.20702143 -0.04326335 -0.28028058 -0.286921 -0.46908732 -1.03633961 0.02914775 -0.29263743 0.01728096 -0.32898422 -0.13801971 -0.93196832]
Loss with sklearn theta: 0.46843403006

L1sklearn
Theta found by sklearn with L1 reg: [1.86965269 0.68661649 1.28041683 -4.86256834 -1.6218069 -2.34227548 0. 0. 0. 0. 0. 0. 0. 0. -2.36743001 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Loss with sklearn theta: 0.438146984954

Figure 13: path

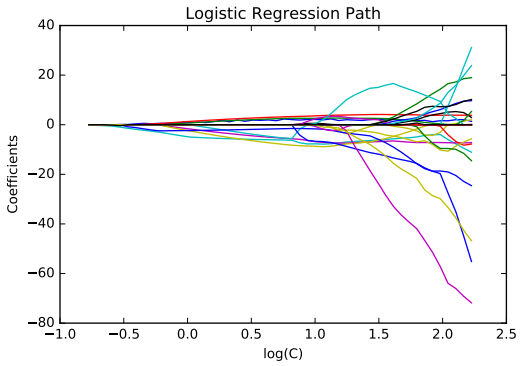


5.5.3 with reg=0.1

L2sklearn
Theta found by sklearn with L2 reg: [2.65855183 1.76427994 2.91364412 -4.03385629 -3.34849756 -4.0181188 0.76777199 -1.08648166 -0.47195071 -0.4774888 -3.27598952 0.54686285 -1.80180787 -1.17932445 -2.79104067 -0.62127841 -0.4711418 0.61454641 -1.14697992 -1.20796935 -0.10569617 -2.66246949 0.45857402 -0.76144039 0.43744164 -1.17502213 -0.93753591 -1.20049576] Loss with

sklearn theta: 0.353830932899
L1
Theta found by sklearn with L1 reg: [4.00273583 2.56793635 3.56332248 -7.68544357 -6.81244292 -8.6654482 0.59001851 -0.20079995 0. 0. 0. 2.44529584 0. 0. -1.70280933 0. 0. 0.36834145 -0.66643549 0. 0. -6.7197063 0. 0. 0. 0. -0.05987662 0.]
Loss with sklearn theta: 0.336434280508

Figure 14: path



comment: L1: When reg is increasing, the loss is increasing. The number of non-zero coefficients are decreasing, that means large reg makes more zero coefficients. L2:When reg is increasing, the loss is increasing. Usually L2 won't give zero coefficients and we can see the value of theta is decreasing with larger reg.

6 3PART C Logistic regression for spam classification

L2 Penalty experiments ————
best_lambda = 0.1
Accuracy on set aside test set for std = 0.9296875

best_lambda = 0.6
Accuracy on set aside test set for logt = 0.943359375

best_lambda = 1.1
Accuracy on set aside test set for bin = 0.927734375

L1 Penalty experiments ————
best_lambda = 4.6
Accuracy on set aside test set for std = 0.921875

best_lambda = 1.6
Accuracy on set aside test set for logt = 0.944010416667

best_lambda = 3.6
Accuracy on set aside test set for bin = 0.92578125

Both L1 and L2 have best performance on logt with accuracy around 0.94. For L2, when best lambda is increasing, the theat is decreasing. L2 always give non-zero coefficients, while L1 give some zero coefficients.This is because L1 regularization forces coefficients to go to zero more often. L1 has larger sparsity than L2. And we suggest to use L1 regularization because this model gives few parameters which is simper. We can get rid of irrelevant features and reduce the chance of overfitting.