# Comp 540 Machine Learning

Yue He, Peng Yang

January 2019

On my honor, I have neither given nor received any unauthorized aid on this assignment.

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

**Exercise 1.** *Proof.* Since we know that $g(z) = \frac{1}{1+e^{-z}}$, then

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

□

**Exercise 2.** *Proof.* Since in logistic regression, we know the following,

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+exp(-\theta^T x)}$$

then, we know,

$$p(y^{(i)}|x^{(i)}, \theta) = h_\theta(x^{(i)})^{y(i)}(1-h_\theta(x^{(i)}))^{1-y^{(i)}},\ i = 1, \ldots, m$$

Then we can get the likelihood function,

$$\begin{aligned}
L(\theta) &= \prod_i p(y^{(i)}|x^{(i)}, \theta) \\
&= \prod_i h_\theta(x^{(i)})^{y(i)}(1-h_\theta(x^{(i)}))^{1-y^{(i)}}
\end{aligned}$$

By taking the negative *log* on both side,

$$NNL(\theta) = -\sum_{i}^{m} \left\{ y^{(i)} log h_\theta(x^{(i)}) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)})) \right\}$$

Then we calculate,

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

* □

**Exercise 3.** *Proof.* Note that $H = X^T S X$,

$$S = diag(h_\theta(x^{(1)})(1 - h_\theta(x^{(1)})), \ldots, h_\theta(x^{(m)})(1 - h_\theta(x^{(m)})))$$

where $0 < h_\theta(x^{(i)}) < 1 \rightarrow 0 < (1 - h_\theta(x^{(i)})) < 1$. Also, we know that $X$ is full rank. For any non-zero vector $v_{m*1}$,

$$v^T H v = \sum_{i}^{m} S^{(i)} (v^{(i)} \sum_{k}^{m} x_{ki} x_{kj})^2 > 0$$

. Thus, $H$ is positive defined.

□

# 2 Properties of L2 regularized logistic regression

**Exercise 4.** **False** Since we know that

$$\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_k} = H + \frac{\lambda}{m}, \quad if \quad j = k$$

$$\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_k} = H, \quad if \quad j \neq k$$

where H is Hessian matrix and it is positive defined. Thus, when $\lambda > 0$, $\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_k} > 0$, we know $J(\theta)$ is a convex. $J(\theta)$ has one global optimal solution.

**Exercise 5.** **False** In L2 regularized logistic regression, the penalty term is actually a prior with normal distribution; thus, $\hat{\theta}$ won't be sparse. However, if using lasso regularized logistic regression, $\hat{\theta}$ would be sparse.

**Exercise 6. True** When $\lambda = 0$, we minimize the loss function $J(\theta)$, which may lead to over-fitting the model. Thus, some coefficients $\theta_j$ might become infinite to fit every single response variable.
For example, from Figure 1, we see that the two kinds of points are linearly separated. Based on the logistic model, we know

$$p(y = 1|x) = \frac{1}{1 + \exp \theta_0 + \theta_1 x_1 + \theta_2 x_2} \tag{1}$$

Thus, when $\lambda = 0$ and there is no penalty on $\theta$; Furthermore, $\theta_2$ goes to $\infty$, which can make a perfect classification.

**Exercise 7. False** If the training data is linearly separable, the increasing of $\lambda$ won't have an influence of the first term of $J(\theta)$. Similarly, based on the Figure 1 and equation (1), once $\theta_2 > 0$, could make a perfect classification which means the first part of $J(\theta)$ would not change with the increasing of $\lambda$.

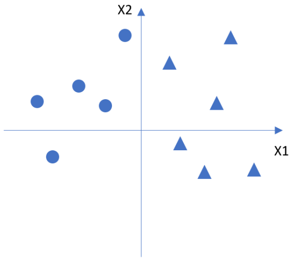

Figure 1:

# 3 Implementing a k-nearest-neighbor classifier

## 3.1 Distance matrix computation with two loops

## 3.2 Compute majority label

## 3.3 Distance matrix computation with one loop

## 3.4 Distance matrix computation with no loops

Two loop version took 19.844659 seconds One loop version took 66.737828 seconds No loop version took 0.301162 seconds

### 3.5 Choosing k by cross validation

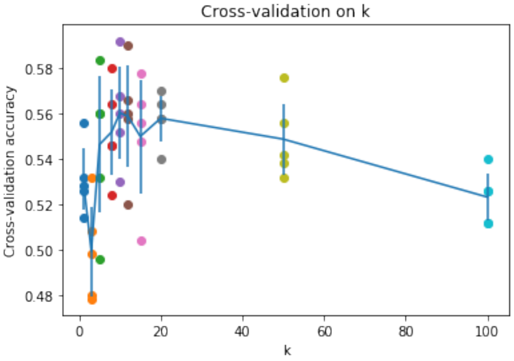

Figure 2:

k = 1, accuracy = 2.000000 k = 1, accuracy = 2.000000 k = 1, accuracy = 2.000000 k = 1, accuracy = 2.000000 k = 1, accuracy = 2.000000 k = 3, accuracy = 1.008000 k = 3, accuracy = 1.028000 k = 3, accuracy = 1.050000 k = 3, accuracy = 1.046000 k = 3, accuracy = 1.100000 k = 5, accuracy = 0.856000 k = 5, accuracy = 0.890000 k = 5, accuracy = 0.906000 k = 5, accuracy = 0.906000 k = 5, accuracy = 0.976000 k = 8, accuracy = 0.750000 k = 8, accuracy = 0.814000 k = 8, accuracy = 0.794000 k = 8, accuracy = 0.804000 k = 8, accuracy = 0.808000 k = 10, accuracy = 0.732000 k = 10, accuracy = 0.792000 k = 10, accuracy = 0.738000 k = 10, accuracy = 0.764000 k = 10, accuracy = 0.770000 k = 12, accuracy = 0.712000 k = 12, accuracy = 0.774000 k = 12, accuracy = 0.720000 k = 12, accuracy = 0.730000 k = 12, accuracy = 0.742000 k = 15, accuracy = 0.666000 k = 15, accuracy = 0.718000 k = 15, accuracy = 0.710000 k = 15, accuracy = 0.698000 k = 15, accuracy = 0.722000 k = 20, accuracy = 0.662000 k = 20, accuracy = 0.686000 k = 20, accuracy = 0.666000 k = 20, accuracy = 0.656000 k = 20, accuracy = 0.682000 k = 50, accuracy = 0.582000 k = 50, accuracy = 0.600000 k = 50, accuracy = 0.608000 k = 50, accuracy = 0.594000 k = 50, accuracy = 0.586000 k = 100, accuracy = 0.540000 k = 100, accuracy = 0.566000 k = 100, accuracy = 0.548000 k = 100, accuracy = 0.562000 k = 100, accuracy = 0.568000

# 4 Implementing logistic regression

### 4.1 Visualizing the dataset

First, visualizing the dataset

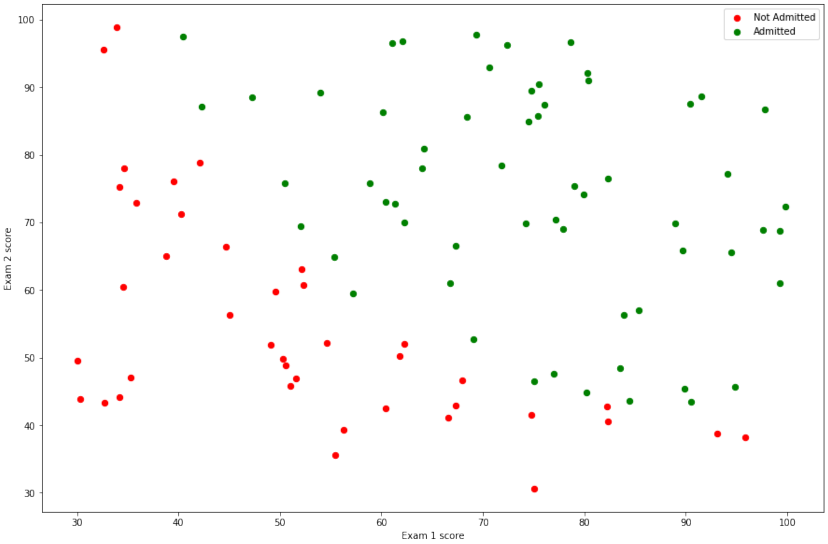

Figure 3:

## 4.2 Prediction using a logistic regression model

Loss on all-zeros theta vector (should be around 0.693) = 0.6931
Gradient of loss w.r.t. 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 = 0.7762
Accuracy on the training set = 0.8900

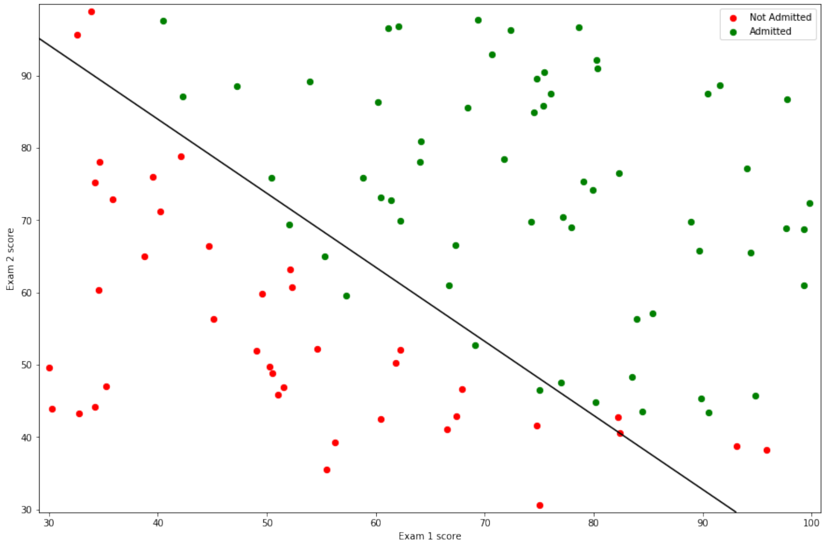

Figure 4:

## 4.3 Regularized logistic regression

### 4.3.1 Visualizing the data

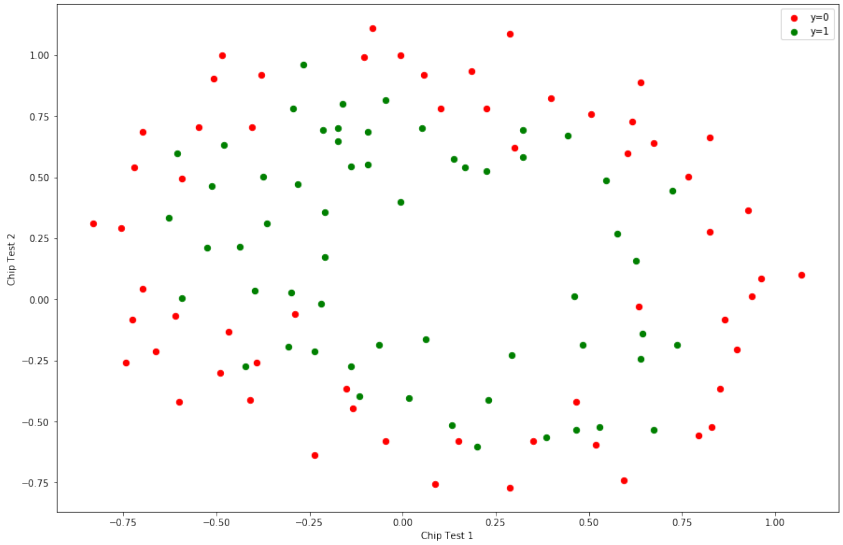

Figure 5:

### 4.3.2 Plotting the decision boundary

Accuracy on the training set = 0.8305

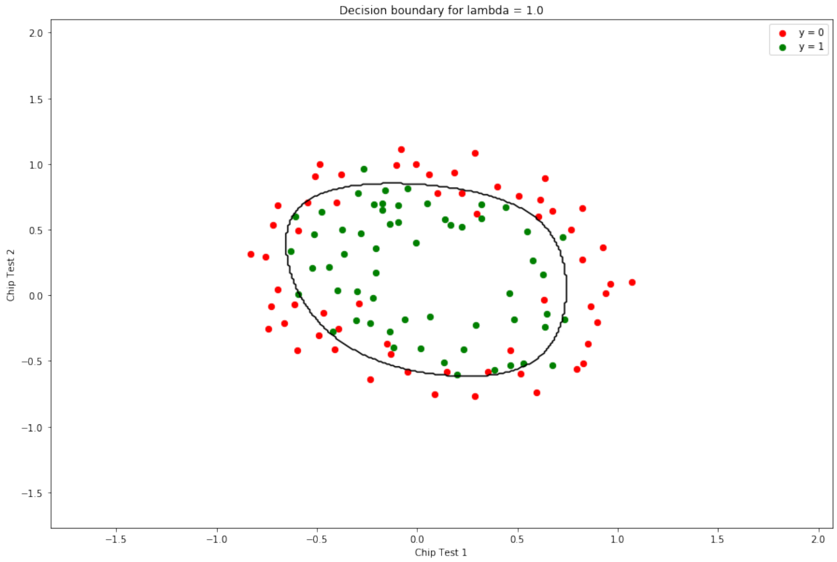

Figure 6: Figure4.4

### 4.4 Varying $\lambda$

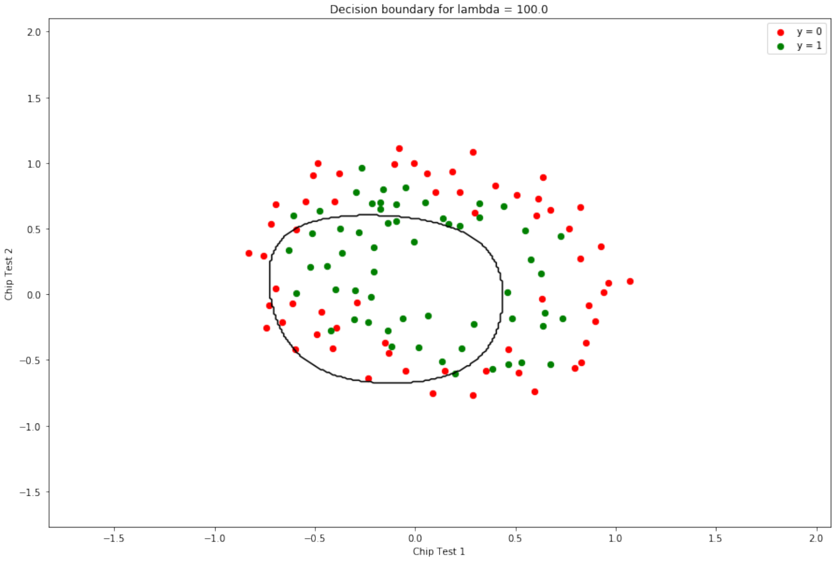

Figure 7:

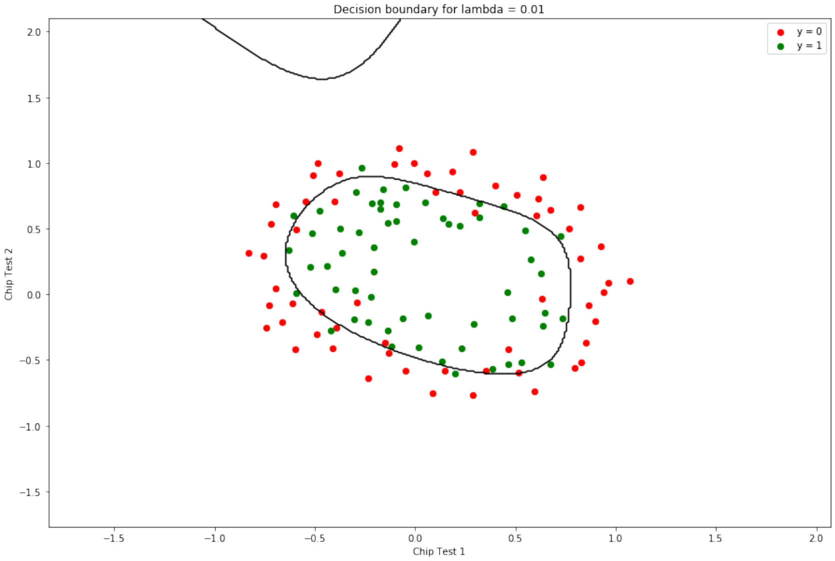

Figure 8:

In general, L1 has more zero cofficient than L2.

### 4.5 Fitting regularized logistic regression models (L2 and L1)

For different $\lambda$, we get the best *lambda* by using cross-validation.
For L2 Penalty experiments

best lambda = 0.100. Accuracy on set aside test set for std = 0.9297.

best lambda = 0.600. Accuracy on set aside test set for logt = 0.9434.

best lambda = 1.100. Accuracy on set aside test set for bin = 0.9277.

For L1 Penalty experiments

best lambda = 4.100. Accuracy on set aside test set for std = 0.9225.

best lambda = 1.600. Accuracy on set aside test set for logt = 0.9440.

best lambda = 3.600. Accuracy on set aside test set for bin = 0.9258.

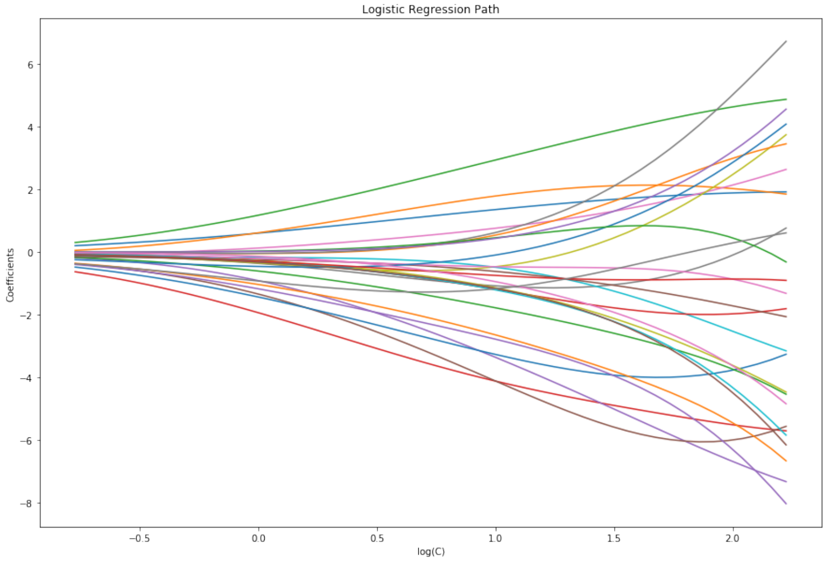

Figure 9:

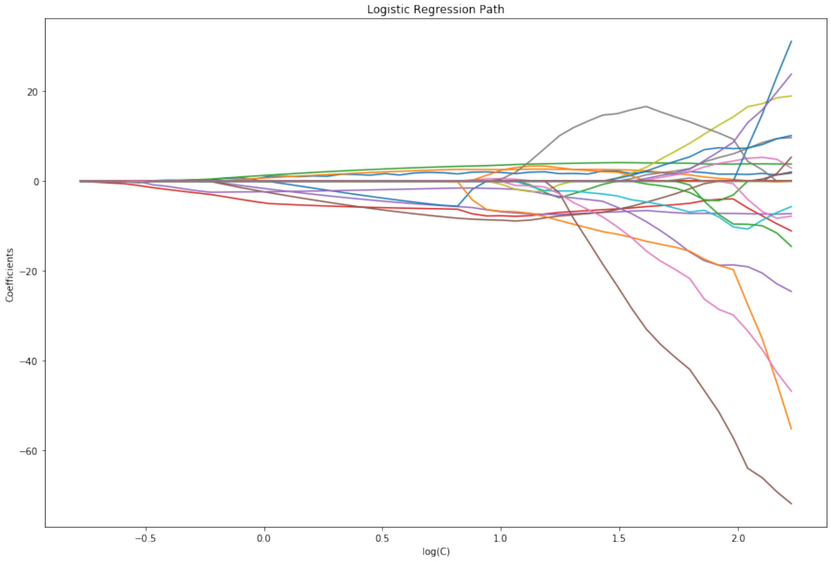

Figure 10:

Figure9 shows the L2 regularization path, while Figure10 shows the L1 regularization path. In L1, lambda needs to be large.L1 is more sparse. Considering that we have 57 features, we only need the most important ones. So I choose L1