COMP 540 Assignment #2

Yunda Jia yj32
Yu Wu yw92

February 8, 2020

1 Gradient and Hessian of (θ) for logistic regression(20 points)

- Show that $\frac{\partial g(z)}{\partial z} = g(z)(1 - g(z))$.

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

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

Thus, $\frac{\partial g(z)}{\partial z} = g(z)(1 - g(z))$.

- Derive the gradient of the L2 penalized cost function (θ).

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^i \log(h_\theta(x^i)) + (1-y^i)\log(1-h_\theta(x^i))) + \frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2$$

$$\frac{\partial J(\theta)}{\partial \theta} = -\frac{1}{m}\sum_{i=1}^{m}(y^i \frac{(h_\theta(x^i))'}{h_\theta(x^i)} + (1-y^i)\frac{(1-h_\theta(x^i))'}{1-h_\theta(x^i)}) + \frac{\lambda}{2m}\frac{\partial \theta^T\theta}{\partial \theta}$$

Since

$$\begin{aligned} h_\theta(x^i)' &= x^i h_\theta(x^i)(1-h_\theta(x^i)) \\ (1-h_\theta(x^i))' &= -x^i h_\theta(x^i)(1-h_\theta(x^i)) \\ \frac{\lambda}{2m}\frac{\partial \theta^T\theta}{\partial \theta} &= \frac{\lambda}{2m}2\boldsymbol{\theta} \\ &= \frac{\lambda}{m}\boldsymbol{\theta} \end{aligned}$$

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta} &= -\frac{1}{m}\sum_{i=1}^{m}(y^i x^i h_\theta(x^i)(1-h_\theta(x^i)) + (1-y^i)(-x^i h_\theta(x^i)(1-h_\theta(x^i)))) + \frac{\lambda}{m}\boldsymbol{\theta} \\ &= -\frac{1}{m}\sum_{i=1}^{m}(y^i x^i - y^i x^i h_\theta(x^i) - x^i h_\theta(x^i) + y^i x^i h_\theta(x^i)) + \frac{\lambda}{m}\boldsymbol{\theta} \\ &= \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)x^i + \frac{\lambda}{m}[0, \theta_1, ..., \theta_d]^T\end{aligned}$$

- Derive the vector form of the first derivative of (θ) with respect to θ.

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

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta} &= -\frac{1}{m}(\boldsymbol{y}^T \cdot \frac{1}{g(\boldsymbol{\theta}^T\boldsymbol{x})} \cdot \frac{\partial g(\boldsymbol{\theta}^T\boldsymbol{x})}{\partial(\boldsymbol{\theta}^T\boldsymbol{x})} \cdot \boldsymbol{x} + (1-\boldsymbol{y}^T) \cdot \frac{1}{1-g(\boldsymbol{\theta}^T\boldsymbol{x})} \cdot \frac{\partial(1-g(\boldsymbol{\theta}^T\boldsymbol{x}))}{\partial(\boldsymbol{\theta}^T\boldsymbol{x})} \cdot \boldsymbol{x}) + \frac{\lambda}{m}\boldsymbol{\theta} \\ &= -\frac{1}{m}(\boldsymbol{y}^T \cdot \frac{g(\boldsymbol{\theta}^T\boldsymbol{x})(1-g(\boldsymbol{\theta}^T\boldsymbol{x}))}{g(\boldsymbol{\theta}^T\boldsymbol{x})} \cdot \boldsymbol{x} - (1-\boldsymbol{y}^T) \cdot \frac{-g(\boldsymbol{\theta}^T\boldsymbol{x})(1-g(\boldsymbol{\theta}^T\boldsymbol{x}))}{1-g(\boldsymbol{\theta}^T\boldsymbol{x})}) + \frac{\lambda}{m}\boldsymbol{\theta} \\ &= \frac{1}{m}(\boldsymbol{y}^T \cdot (1-g(\boldsymbol{\theta}^T\boldsymbol{x})) \cdot \boldsymbol{x} + (1-\boldsymbol{y}^T) \cdot g(\boldsymbol{\theta}^T\boldsymbol{x}) \cdot \boldsymbol{x}) + \frac{\lambda}{m}\boldsymbol{\theta} \\ &= \frac{1}{m}(h_\theta(\boldsymbol{x}) - \boldsymbol{y}^T) \cdot \boldsymbol{x} + \frac{\lambda}{m}[0, \theta_1, ..., \theta_d]^T\end{aligned}$$

- Show that the Hessian or second derivative of (θ) can be written as $H = \frac{1}{m}(\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X} + \lambda\boldsymbol{I})$ where

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

First derivative not including regulation part.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)x_{ij}$$

$$\begin{aligned}\frac{\partial^2 J(\theta)}{\partial \theta_{ij}} &= \frac{1}{m}\sum_{i=1}^{m} x_{ij}(\frac{\partial}{\partial \theta_i}h_\theta(x_i)) \\ &= \frac{1}{m}\sum_{i=1}^{m} x_{ij}x_{ik}h_\theta(x_{(i)})(1-h_\theta(x_{(i)}) \\ &= \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X}\end{aligned}$$

Where $\boldsymbol{X} = (X_{ij}, ..., x_{mj})^T$
For regulation part

$$\begin{aligned}\frac{\partial^2 J(\theta)}{\partial \theta_{ij}} &= \frac{\partial}{\partial \theta}\frac{\lambda}{m}\theta^T \\ &= \frac{\lambda}{m}\boldsymbol{I}\end{aligned}$$

So $H = \frac{1}{m}(\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X} + \lambda\boldsymbol{I})$.
For an arbitrary selected non-zero vector $\boldsymbol{u}$:

$$\boldsymbol{u}^T\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X}\boldsymbol{u} = (\boldsymbol{X}\boldsymbol{u})^T\boldsymbol{S}(\boldsymbol{X}\boldsymbol{u})$$

Since $\boldsymbol{S}$ is positive definite, for an arbitrary selected non-zero vector $\boldsymbol{v}$:

$$\boldsymbol{v}\boldsymbol{S}\boldsymbol{v} > 0$$

Assume $\boldsymbol{X}$ is full rank, thus:

$$\begin{aligned}(\boldsymbol{X}\boldsymbol{u})^T\boldsymbol{S}(\boldsymbol{X}\boldsymbol{u}) &= \boldsymbol{u}^T(\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X})\boldsymbol{u} \\ &> 0\end{aligned}$$

So $H = \boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X}$ is always positive.

- Use Netwon's method.
  – State the θ update.
  From Netwon's method

$$\theta_{t+1} = \theta_t - \boldsymbol{H}^{-1}\nabla J(\theta)$$

  – Show the values of θ after the first and second iteration of Newton's method.

```
A = np.array([0, 3, 1, 3, 0, 1, 1, 1]).reshape((4, 2))
#prepend 1
A = np.c_[np.ones(A.shape[0]), A]
y = np.array([1, 1, 0, 0]).reshape(1, 4)
theta = np.array([0, -2, 1]).reshape(3, 1)
reg = 0.07
#grad + hess
h = utils.sigmoid(A @ theta)
m, _ = A.shape
test = h - y.T
grad = 1 / m * A.T@(h - y.T) + reg / m * theta
I = np.identity(grad.shape[0])
HESS = 1 / m * (np.dot(A.T, A) * np.diag(h) * np.diag(1 - h)) + reg / m * I
#Newton's law
hessianInv = np.linalg.inv(HESS)
theta1 = theta - np.dot(hessianInv, grad)
theta2 = theta1 - np.dot(hessianInv, grad)
print(f'After the first iteration is\n {theta1}')
print(f'After the first iteration is\n {theta2}')
```

The results are shown below.

```
#After the first iteration is
 [[-5.95507474]
 [ 0.37364703]
 [ 2.64597488]]
#After the first iteration is
 [[-11.91014948]
 [  2.74729407]
 [  4.29194977]]
```

2 Overfitting and unregularized logistic regression(10 points)

For maximum likelihood $\theta^T X_n > 0$ and $\theta^T X_m < 0$ for $(x_n \in C_1, X_m \in C_2)$.
If $|\theta| \to \infty$

1. $P(C_1|X_n) = \delta(\theta^T x_n) \to 1$
2. $P(C_2|X_m) = 1 - P(C_1|X_m) \to 1$

Which means for the likelihood function, if we have $|\theta| \to \infty$, and we also label all points at two sides (C_1C_2), then we can achieve the maximum of likelihood. Therefore, it will cause overfitting.
The result physically means the $|\theta|$ is not import for finding a decision boundary.
To avoid singular solution, we can add regularization term.

3 Implementing a k-nearest-neighbor classifier(25 points)

- visualize the distance matrix

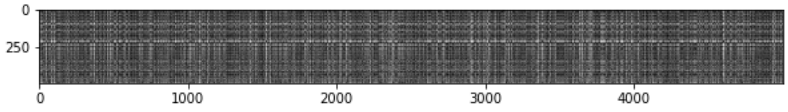


Figure 1: Visualization of distance matrix

Question: Notice the structured patterns in the distance matrix, where some rows or columns are visible brighter. (Note that with the default color scheme black indicates low distances while white indicates high distances.)

- What in the data is the cause behind the distinctly bright rows? - What causes the columns?

Answer:

1.The ith test data are very different to a large number of train data causes the distinctly bright rows.

2.The jth train data are very different to a large number of test data causes the bright columns. If the columns are dim, it is because the jth train data are similar to a large number of test data.

- Compute the majority label
  For k=1, Got 137/500$correct$ => $accuracy$: 0.274000
  For k=5, Got 139/500$correct$ => $accuracy$: 0.278000

- Choosing k by cross validation

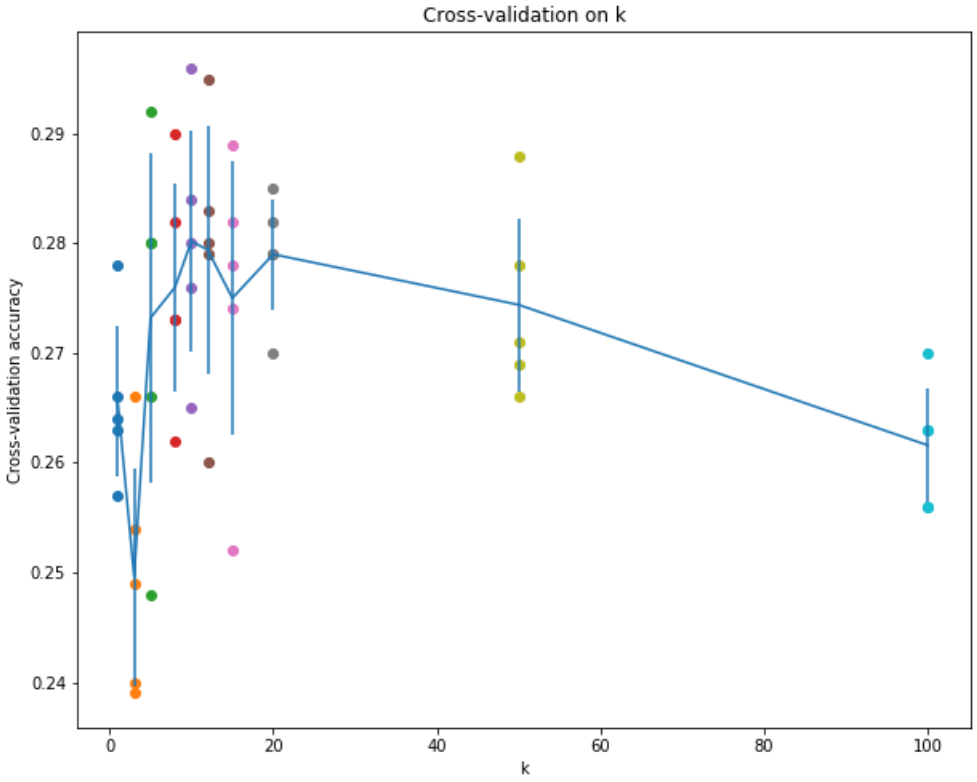


Figure 2: Cross Validation on k

4 Implementing logistic regression(45 points)

4.1 Logistic regression

- Visualizing the dataset

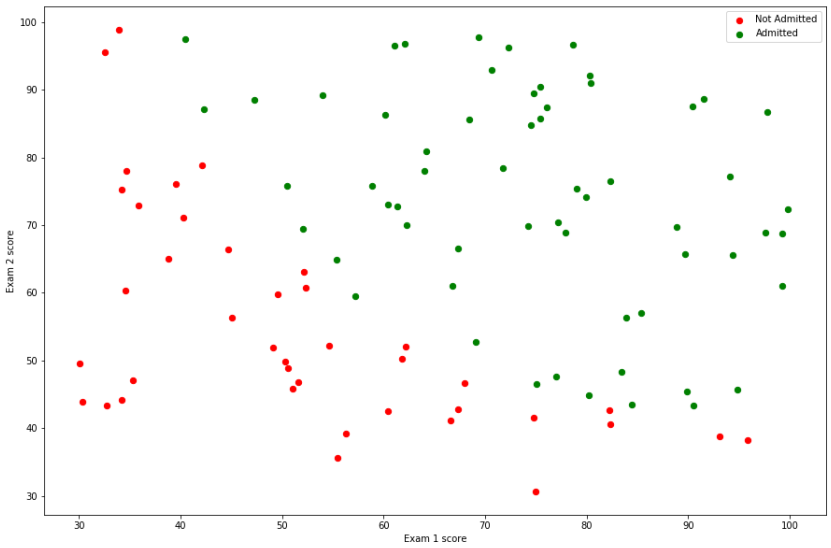


Figure 3: Visualization

- Visualize decision boundary

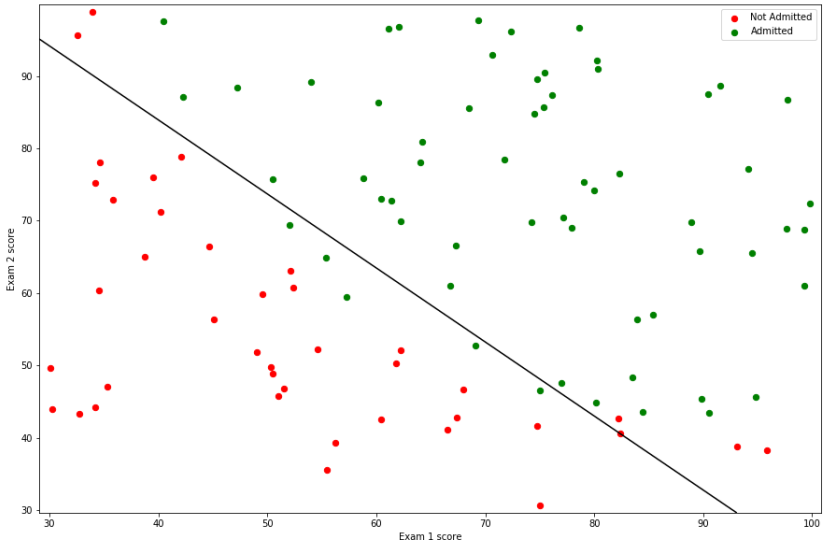


Figure 4: Visualization of decision boundary

4.2 Regularized logistic regression

- Visualizing the data

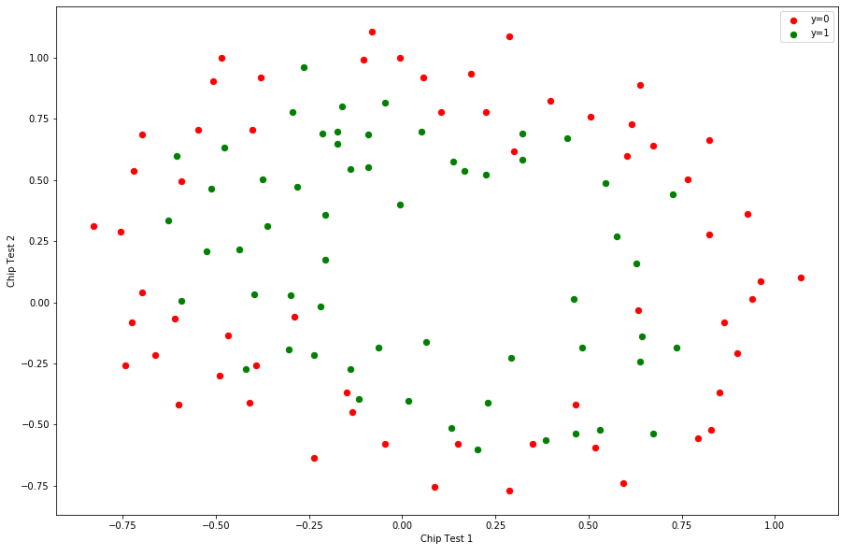


Figure 5: Visualization

- plotting the decision boundary

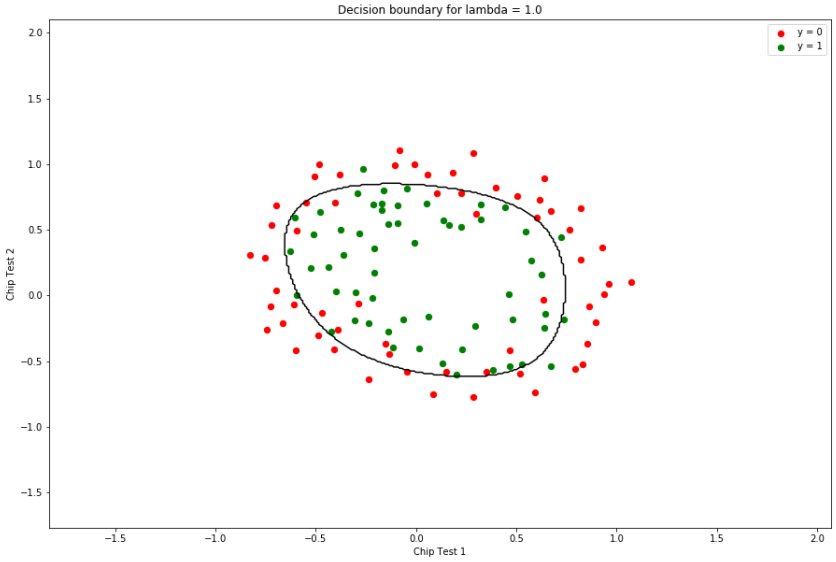
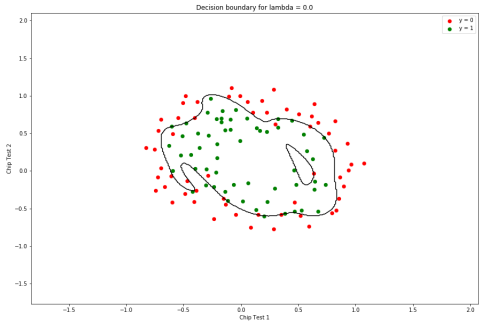
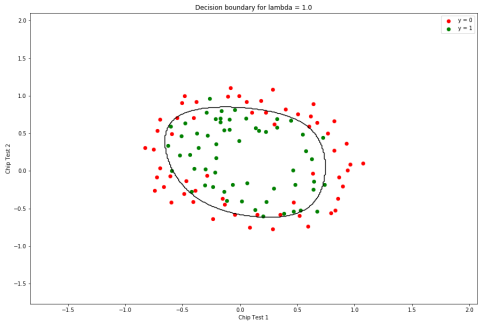


Figure 6: Decision boundary

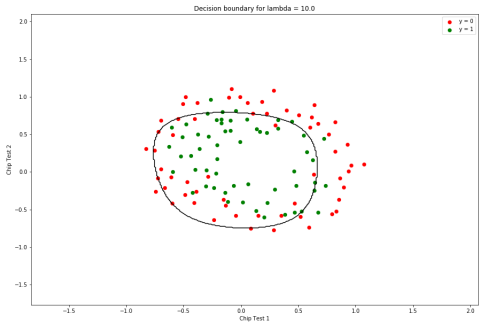
- Varying λ



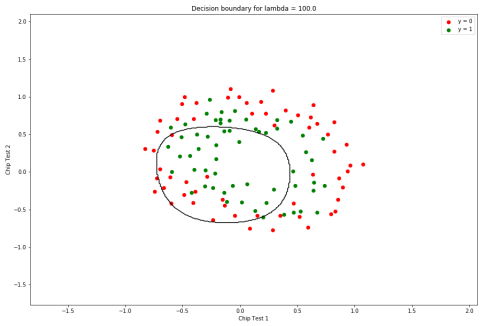
(a) $\lambda = 0.0$



(b) $\lambda = 1.0$



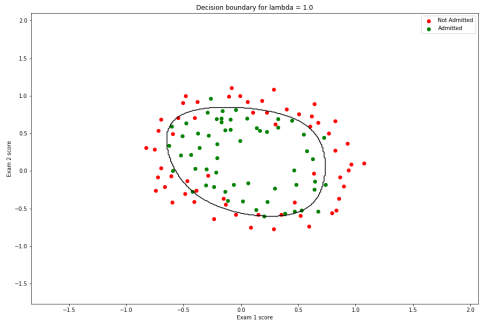
(c) $\lambda = 10.0$



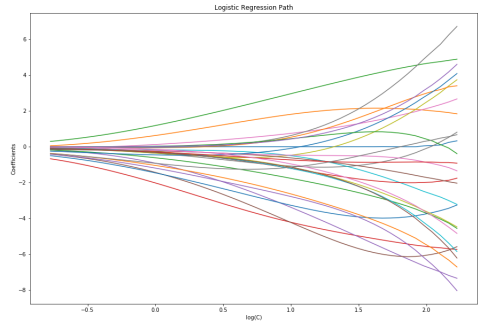
(d) $\lambda = 100.0$

Figure 7: Decision boundary varying λ

- Exploring L1 and L2 penalized logistic regression.
  For sklearn L2 regularization.



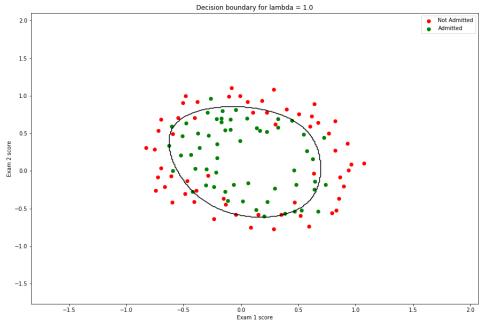
(a) L2 decision boundary



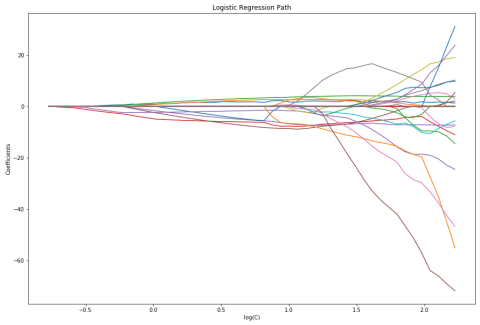
(b) L2 regularization path

Figure 8: Sklearn L2 regularization

For sklearn L1 regularization.



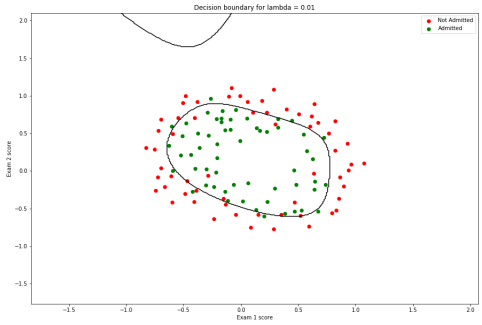
(a) L1 decision boundary



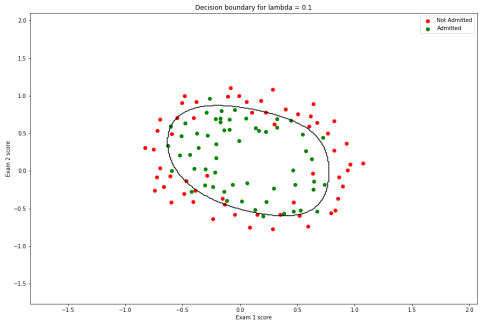
(b) L1 regularization path

Figure 9: Sklearn L1 regularization

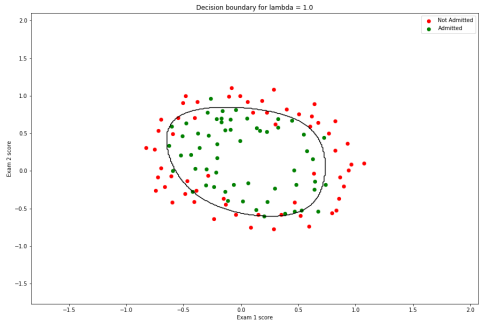
Try varying λ.
For sklearn L2 regularization.



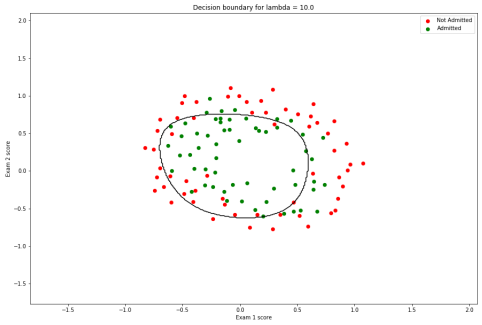
(a) $\lambda = 0.01$



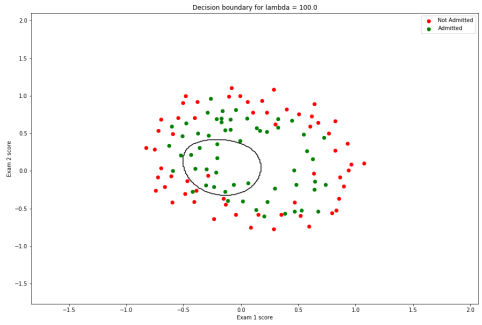
(b) $\lambda = 0.1$



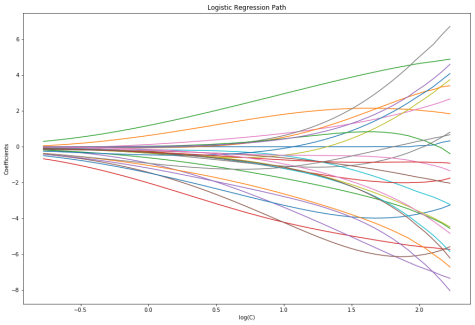
(c) $\lambda = 1.0$



(d) $\lambda = 10.0$



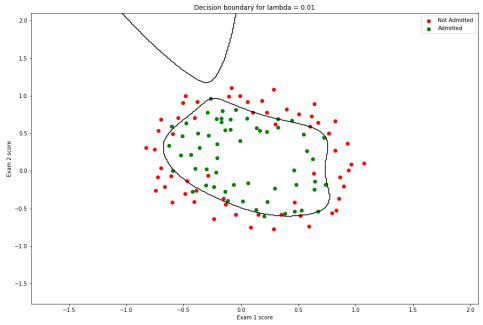
(e) $\lambda = 100.0$



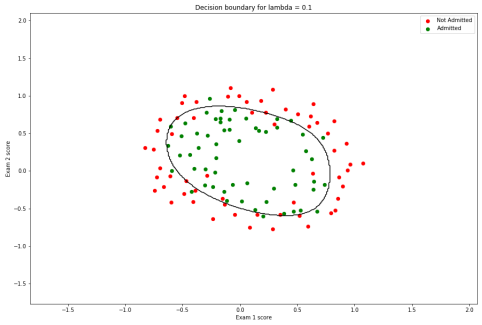
(f) L2 regularization path

Figure 10: Decision boundary varying λ for L2

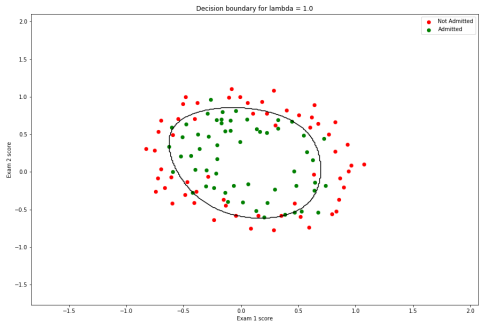
For sklearn L1 regularization.



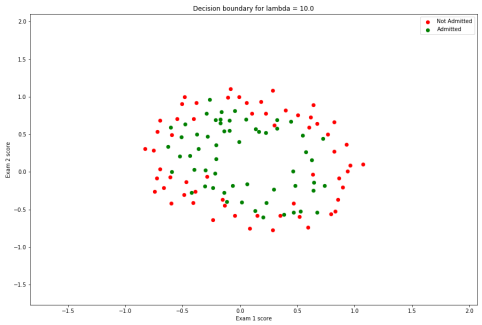
(a) $\lambda = 0.01$



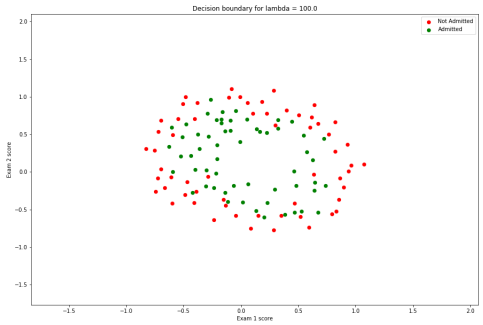
(b) $\lambda = 0.1$



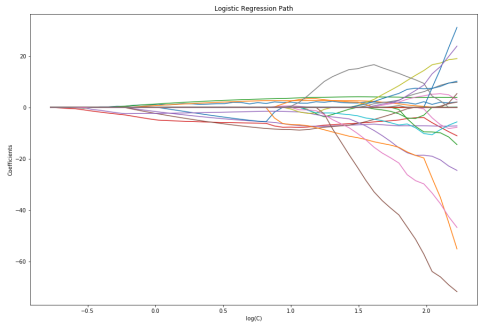
(c) $\lambda = 1.0$



(d) $\lambda = 10.0$



(e) $\lambda = 100.0$



(f) L1 regularization path

Figure 11: Decision boundary varying λ for L1

For both L1 and L2 models, with λ varies from 0.01 to 100.0, change from overfitting to under-fitting. The difference is the regularization path. For L1, the coefficients decreases suddenly when λ increases to 10.0. For L2, the coefficients increases and decreases smoothly.

4.3 Part C: Logistic regression for spam classification

4.3.1 Comment on the model sparsities with L1 and L2 regularization

For L1 regularization, it has more sparse coefficients than L2.
For this data, I would recommend L1 regularization, because for this problem, not all the emails are spam emails. Thus the sparse of L1 will help better to focus on important features.