COMP 540: Assignment 2

Suguman Bansal (sb55), Harsh Upadhyay (hu3)

February 8, 2016

1

a.

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

b.

Note that $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ where θ is a d vector, and x is a d vector.

$$
\begin{aligned}
NLL(\theta) &= \frac{-1}{m}\Sigma_{i=1}^{m} y^{(i)} \cdot \log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)})) \\
\implies NLL(\theta) &= \frac{-1}{m}\Sigma_{i=1}^{m} NLL(\theta)_i \text{ where } NLL(\theta)_i = y^{(i)} \cdot \log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)})) \\
\implies \frac{\partial(NLL(\theta))}{\partial(\theta)} &= \frac{-1}{m}\Sigma_{i=1}^{m} \frac{\partial(NLL(\theta)_i)}{\partial(\theta)}
\end{aligned}
$$

Now we compute the value of $\frac{\partial(NLL(\theta)_i)}{\partial(\theta)}$.

$$\begin{aligned}
&\frac{\partial(NLL(\theta)_i)}{\partial(\theta)} = y^{(i)} \cdot \frac{\partial(\log(h_\theta(x^{(i)})))}{\partial(\theta)} + (1-y^{(i)}) \cdot \frac{\partial(\log(1-h_\theta(x^{(i)})))}{\partial(\theta)} \\
\implies &\frac{\partial(NLL(\theta)_i)}{\partial(\theta)} = y^{(i)} \cdot \frac{\partial(\log(h_\theta(x^{(i)})))}{\partial(h_\theta(x^{(i)}))} \cdot \frac{\partial(h_\theta(x^{(i)}))}{\partial(\theta^T x^{(i)})} \cdot \frac{\partial(\theta^T x^{(i)})}{\partial(\theta)} \\
&\qquad + (1-y^{(i)}) \cdot \frac{\partial(\log(1-h_\theta(x^{(i)})))}{\partial(h_\theta(x^{(i)}))} \cdot \frac{\partial(h_\theta(x^{(i)}))}{\partial(\theta^T x^{(i)})} \cdot \frac{\partial(\theta^T x^{(i)})}{\partial(\theta)} \\
\implies &\frac{\partial(NLL(\theta)_i)}{\partial(\theta)} = y^{(i)} \cdot \frac{1}{h_\theta(x^{(i)})} \cdot h_\theta(x^{(i)}) \cdot (1-h_\theta(x^{(i)})) \cdot x^{(i)} \\
&\qquad + (1-y^{(i)}) \cdot \frac{-1}{1-h_\theta(x^{(i)})} \cdot h_\theta(x^{(i)}) \cdot (1-h_\theta(x^{(i)})) \cdot x^{(i)} \\
\implies &\frac{\partial(NLL(\theta)_i)}{\partial(\theta)} = (y^{(i)} - h_\theta(x^{(i)})) \cdot x^{(i)}
\end{aligned}$$

Therefore,

$$\frac{\partial(NLL(\theta))}{\partial(\theta)} = -\frac{1}{m}\Sigma_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)})) \cdot x^{(i)} = \frac{1}{m}\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

c.

We are given that $H = X^T S X$, where $S = diag(h_\theta(x^{(1)}) \cdot (1-h_\theta(x^{(1)}), \ldots, h_\theta(x^{(m)}) \cdot (1-h_\theta(x^{(m)}))$. Since we assume, $0 < h_\theta(x^{(i)}) < 1$ for all i, $0 < h_\theta(x^{(i)}) \cdot (1-h_\theta(x^{(i)}) < 1$. Therefore, $S = S_{\surd} S_{\surd}$ where

$$S_{\surd} = diag(\sqrt{h_\theta(x^{(1)}) \cdot (1-h_\theta(x^{(1)})}, \ldots, \sqrt{h_\theta(x^{(m)}) \cdot (1-h_\theta(x^{(m)}))})$$

Additionally, since $S_{\surd}$ is a diagonal matrix, $S_{\surd} = S_{\surd}^T$.

Therefore, $H = X^T S_{\surd}^T S_{\surd} X = (S_{\surd} X)^T (S_{\surd} X)$.

Let us denote the i-th element of the diagonal of $S_{\surd}$ by $f(i)$. We can show that $(S_{\surd} X)_{p,q} = f(p) X_{p,q}$, where $A_{i,j}$ denotes the i,j-th element of A. What this means, is that $S_{\surd} X$ is the matrix obtained by multiplying the i-th row of X with non-zero scalar value $f(i)$. Hence if $C_k = [x_k^{(1)}, \ldots, x_k^{(m)}]$ denotes the k-th column of X, then $C'_k = [x_k^{(1)} \cdot f(1), \ldots, x_k^{(m)} \cdot f(m)]$ denotes the k-th column of $S_{\surd} X$. Since all $f(i)$s are non-zero, it is easy to see that if $C_1, C_2, \ldots, C_d$ are lineraly independent vectors, then so are $C'_1, C'_2, \ldots, C'_d$. Since, X is a full rank matrix, it means that its columns are linearly independent. Therefore, columns of $S_{\surd} X$ are also linearly independent.

We have shown that $H = (S_{\surd} X)^T (S_{\surd} X)$ where columns of $(S_{\surd} X)$ are linearly independent. Therefore, we have shown that H is positive definite.

2

Since θ_{MAP} maximizes the value of $P(\theta) \cdot P(D|\theta)$, the following relation is true

$$P(\theta_{MAP}) \cdot P(D|\theta_{MAP}) \geq P(\theta_{MLE}) \cdot P(D|\theta_{MLE})$$

We know that θ_{MLE} maximizes the value of $P(D|\theta)$, we know that

$$P(D|\theta_{MAP}) \leq P(D|\theta_{MLE})$$

Since probabilities are always greater than equal to 0, from the above two inequalities we get the following:

$$P(\theta_{MAP}) \geq P(\theta_{MLE})$$

Since, $P(\theta) \sim N(0, \alpha^2 I)$, the above implies that

$$\begin{aligned} & P(\theta_{MAP}) \geq P(\theta_{MLE}) \\ \implies & exp(\frac{-||\theta_{MAP}||_2}{2 \cdot \alpha^2}) \geq exp(\frac{-||\theta_{MLE}||_2}{2 \cdot \alpha^2}) \\ \implies & ||\theta_{MAP}||_2 \leq ||\theta_{MLE}||_2 \end{aligned}$$

Hence proven!

3

A

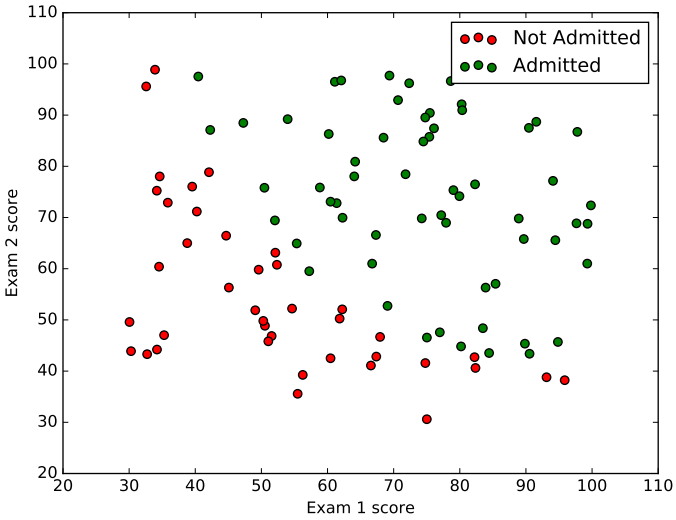

Figure 1: Data plot for admissions based on scores in Exams 1 and 2

Figures 1, 2, and 3 correspond the data set visualization, and the logistic regression model obtained from using functions fmin$_b$fgs and skLearn python packages that were provided to us.

B

Figures 4, 5, 8, 6, 7, 9, 10, 11, 12, and 13 correspond to figures for this problem 3(b). We also observed more zero terms in the Theta learnt by the L1 penalization scheme.

C

As expected, we obtained accuracies in the range of 92%-94% for different feature transforms and regularization schemes.

We observed from the outputs that L1 coefficients had more zero parameters compared to L2, thereby having more sparsity. We also achieved maximum accuracy of 94.44% with L1 regularization and logarithmic transformation of the input. We therefore recommend L1 regularization.

D

Predicted performance of the model was consistently higher for the Mel Cepstral representation in comparison to the FFT representation.

For the FFT representation, the observed accuracy was between 27%-34%. For the Mel Cepstral representation, the onserved accuracy was between 47%-54%. We observed a range of accuracy for successive runs with the same value of λ, since $\mathsf{train_test_split}$ randomly partitions the training and the test sets.

It is difficult to give a definite answer for performace sensitivity to λ due to randomness introduced during the train test split. We tested with λ values equal to 10, 1, 0.1 and 0.01. However, on an average the accuracies reduced as we deviated from $\lambda = 1$. We observed that the range of accuracy was larger as we moved further away from $\lambda = 1$. For example, at $\lambda = 10$, the accuracy range was between 46%-60%.

We observed that of all the music categories, Classical musical was easiest to classify, followed by Pop music. In our experiment, we noticed that Rock was the hardest to classify, followed by Jazz. For these two we noticed a lot of off-diagonal values which were very similar in value.

To study improvements of the performance for such classifiers with respect to λ, we need to fix the training and test data sets instead of randomly splitting the training and test data sets. We can next use cross-validation across a range of λs and select the best one.

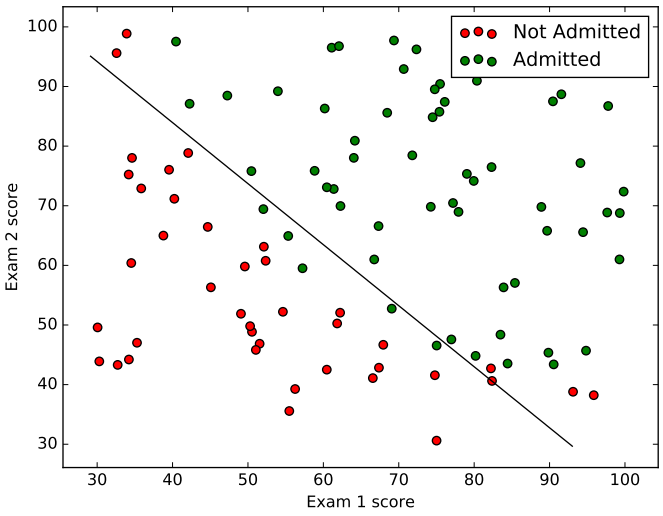

Figure 2: Decision Boundary obtained via fmin_bfgs. The trained model has an accuracy of 89% and predicted an admission probability of 0.776 for scores of 45 and 85 in exams 1 and 2 respectively. Our implementation with all zer0 theta recorded a loss of 0.693 and the gradient of the loss function was recorded as $[-0.1, -12.01, -11.26]^T$. After learning the best theta from fmin-bfgs the cost came down to 0.203

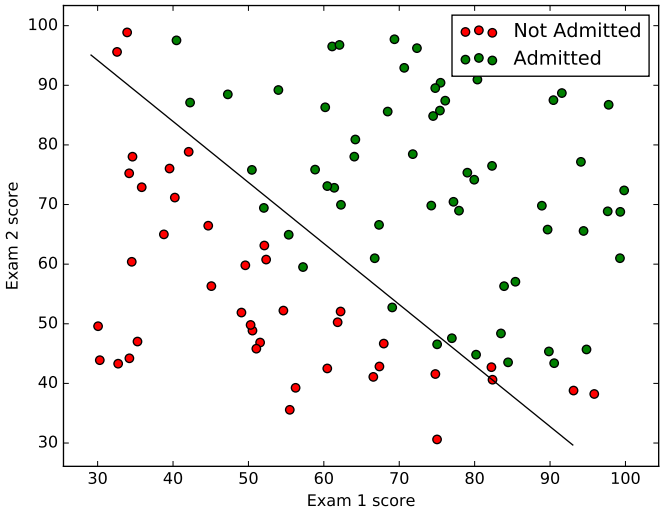

Figure 3: Decision boundary obtained by using sklearn python package. Here too, accuracy of 89% was observed

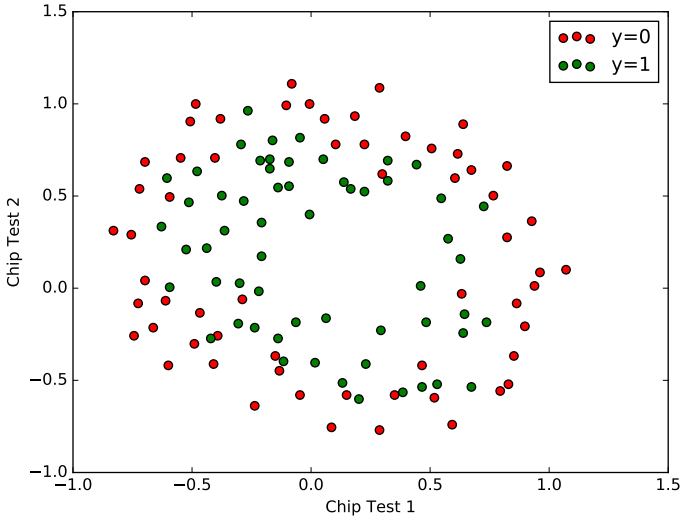

Figure 4: Data plot for the Chip tests

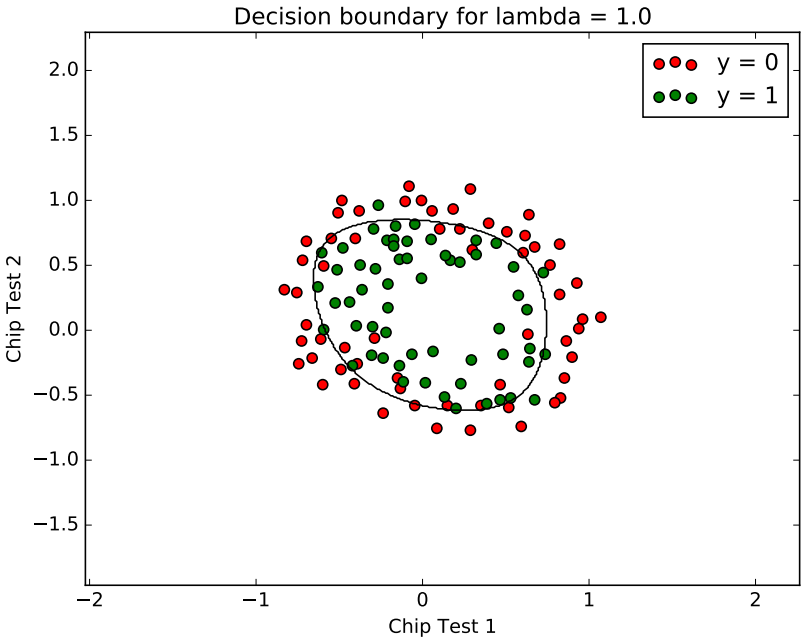

Figure 5: Decision boundary with $\lambda = 1$. As expected, testing on unseen data resulted in an accuracy of 83.05%

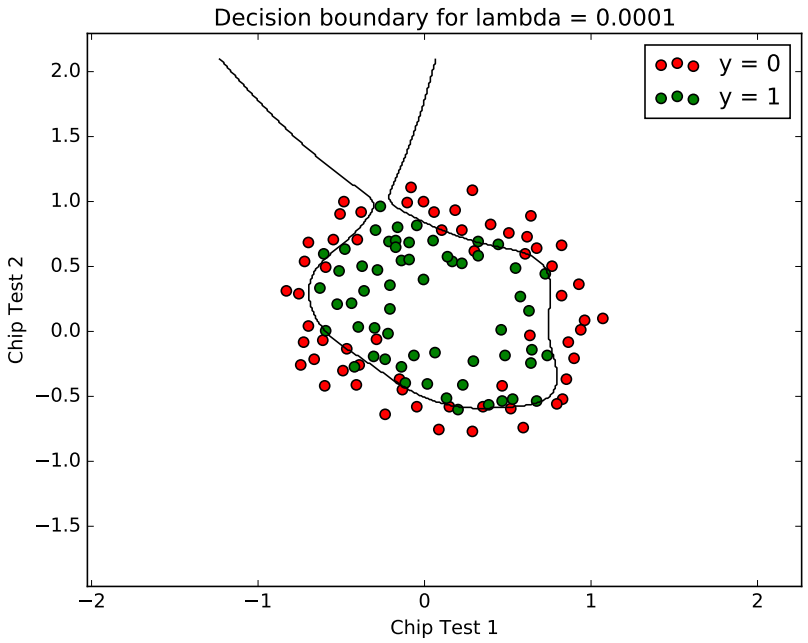

Figure 6: We see that for low λ, we observe overfitting on the training data.

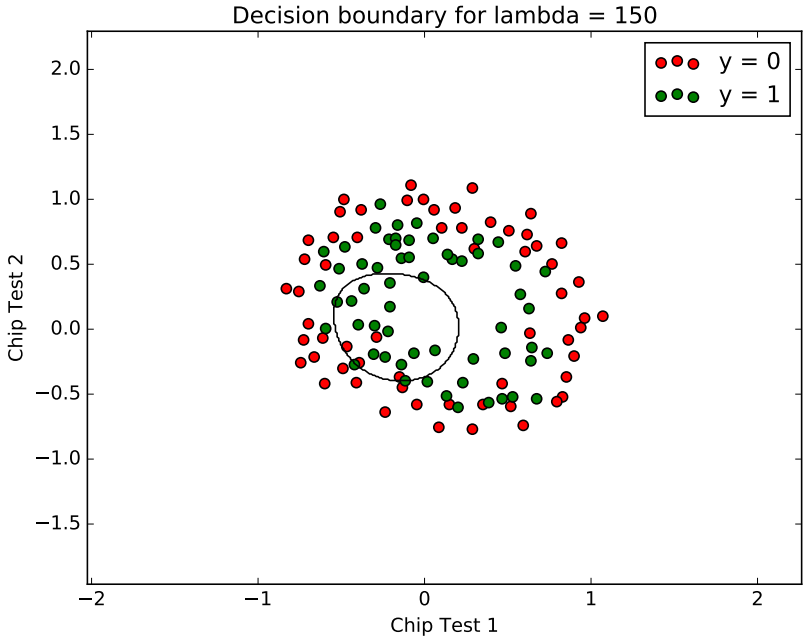

Figure 7: We see that for high λ, we observe underfitting on the training data.

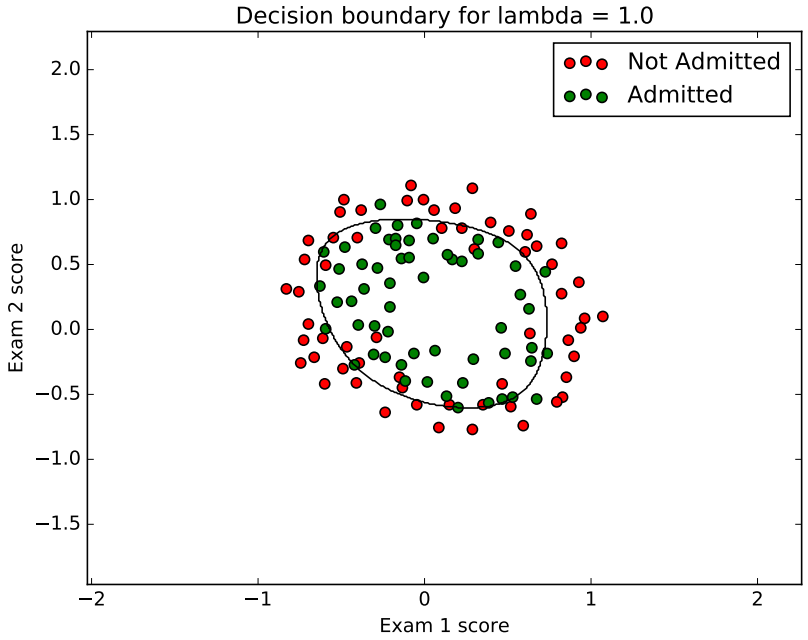

Figure 8: Decision boundary obtained via skLearn using L2 regularization

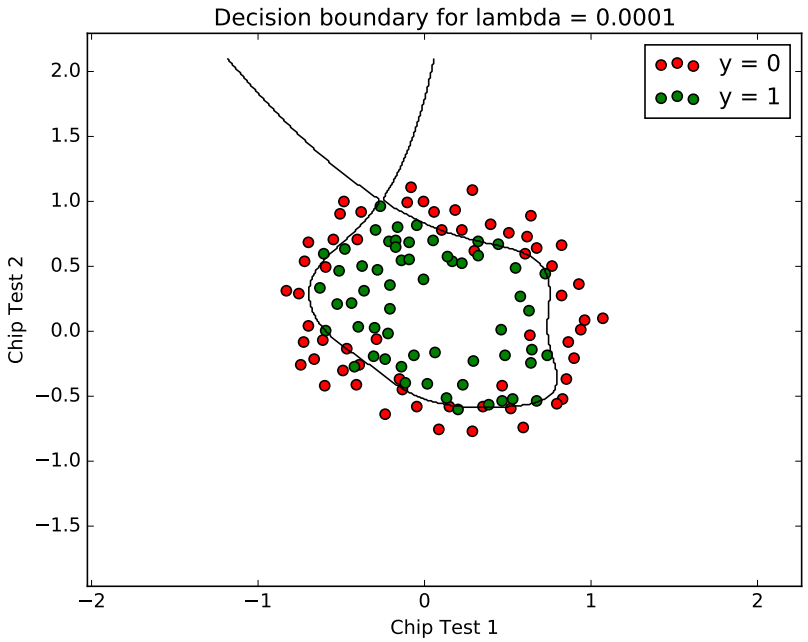

Figure 9: Decision boundary obtained via skLearn using L2 regularization. Here we observe that with a low λ, we see overfitting on the training data

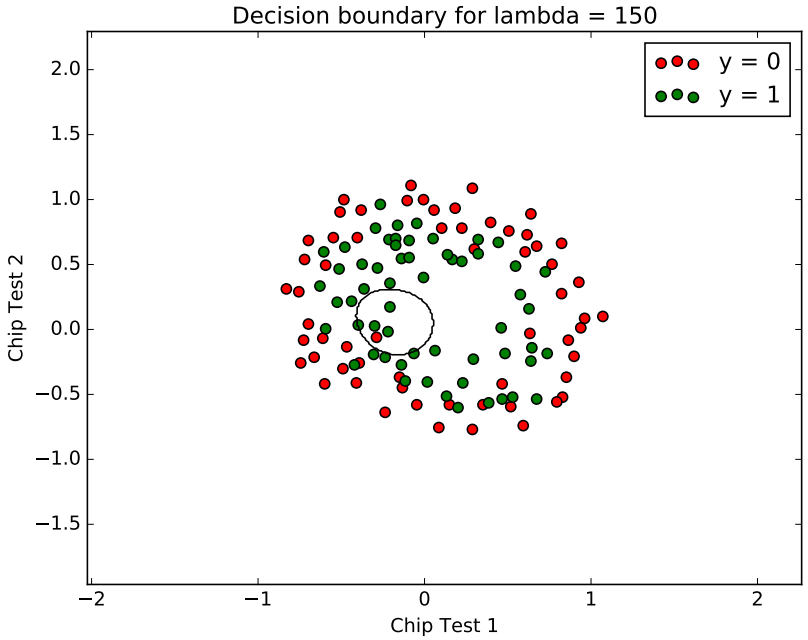

Figure 10: Decision boundary obtained via skLearn using L2 regularization. Here we observe that with a high λ, we see underfitting on the training data

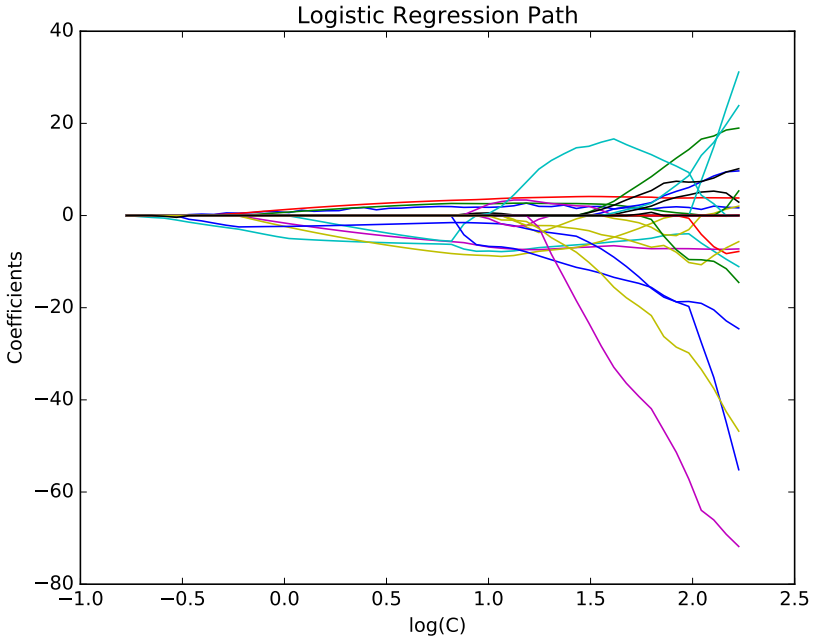

Figure 11: Regularization path with $\lambda = 1$

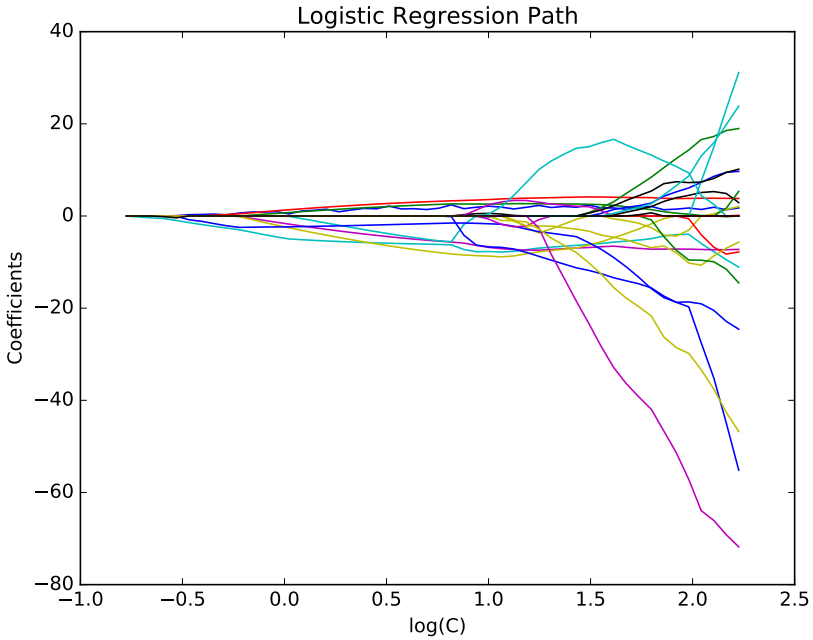

Figure 12: Regularization path with low $\lambda = 0.0001$

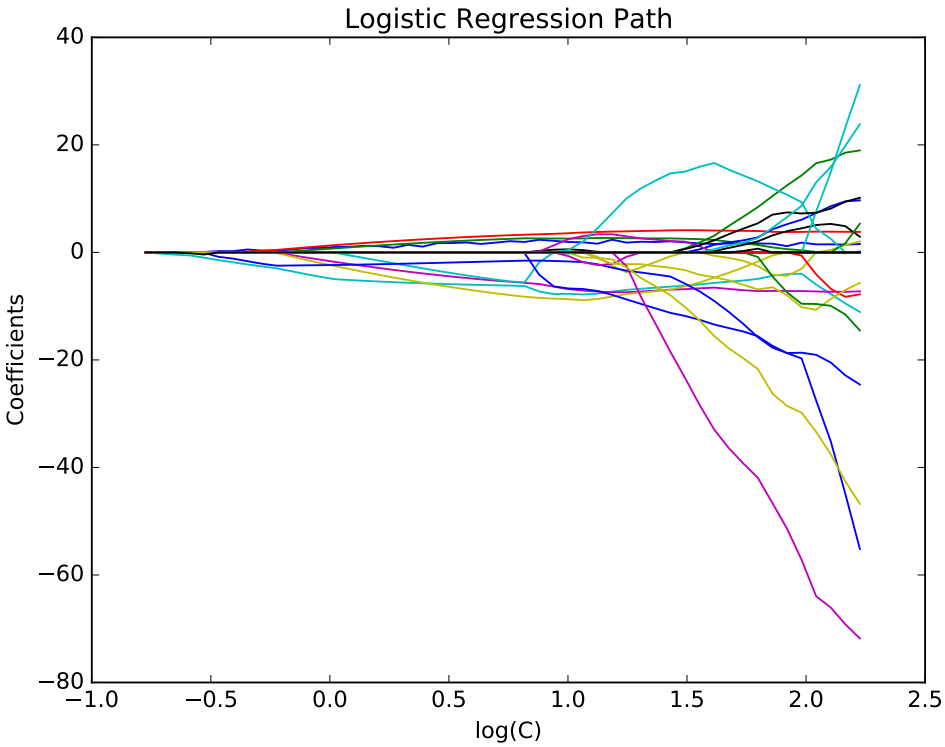

Figure 13: Regularization path with high $\lambda = 150$