COMP 540 Homework 2

Peiguang Wang, Xinran Zhou

Due: February 2, 2018

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

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

   Proof.
   $$\frac{\partial g(z)}{\partial z} = -\frac{e^{-z}}{(1-e^{-z})} = \frac{1}{1+e^{-z}}(1-\frac{1}{1+e^{-z}}) = g(z)(1-g(z))$$

   □

2. Using the previous result and the chain rule of calculus, derive the following expression for the gradient of the negative log likelihood function $NLL(\theta)$ for logistic regression.
   $$\frac{\partial}{\partial\theta}NNL(\theta) = \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

   Proof. Let h_θ denote $h_\theta(x^{(i)})$. Then the $NLL(\theta)$ is:
   $$NLL(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}log(h_\theta) + (1-y^{(i)})log(1-h_\theta)$$

   Then
   $$\frac{\partial}{\partial(\theta x)}NNL = -\sum_{i=1}^{m} y^{(i)}\frac{h_\theta(1-h_\theta)}{h_\theta} + (-1)(1-y^{(i)}\frac{h_\theta(1-h_\theta)}{1-h_\theta})$$

   Simplify the equation above
   $$\frac{\partial}{\partial(\theta x)}NNL = -\sum_{i=1}^{m} y^{(i)}(1-h_\theta) + (y^{(i)}-1)h_\theta = -\sum_{i=1}^{m} y^{(i)} - h_\theta$$

   Derive $\frac{\partial}{\partial(\theta)}NNL$ from $\frac{\partial}{\partial(\theta x)}NNL$:
   $$\frac{\partial NNL}{\partial\theta} = \frac{\partial NLL}{\partial(\theta x)}\frac{\partial(\theta x)}{\partial\theta} = \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

   □

3. The Hessian or second derivative of the $NLL(\theta)$ can be written as $H = X^T S X$ where
   $$S = diag(h_\theta(x^{(1)})(1-h_\theta(x^{(1)})), ..., h_\theta(x^{(m)})(1-h_\theta(x^{(m)})))$$

   Show that H is positive definite. You may assume that $0 < h_\theta(x^{(i)}) < 1$, so the elements of S are strictly positive and that X is full rank.

   Proof. For any given vector u, compute $u^T H u$:
   $$u^T H u = u^T X^T S X u$$

   Let vector $v = (v_1, v_2, ...v_m)$ denote Xu. since X is full rank, then v is not a zero vector.
   $$u^T H u = v^T S v = \sum_{i=1} m v_i^2 S_{ii}$$

   Since the elements of S are strictly positive, then $u^T H u > 0$. So H is positive definite. □

Part 2: Properties of L2 regularized logistic regression

1. (True or False) $J(\theta)$ has multiple locally optimal solutions.

   Solution. This statement is **False**. Since $J(\theta)$ is a convex function, it only have one global optimal point.

2. (True or False) Let $\theta^\star = argmin_\theta J(\theta)$ be a global optimum. $\theta^\star$ is sparse (has many zero entries).

   Solution. This statement is **False**. Since in this regression problem, we use L2 regularization, it won't make $\theta^\star$ become sparse. L2 norm will make θ have small values. If L1 norm is used, then $\theta^\star$ will become sparse.

3. (True or False) If the training data is linearly separable, then some coefficients θ_j might become infinite if $\lambda = 0$.

   Solution. This statement is **True**. When the maximum likelihood solution occurs, sigmoid function is equal to 0.5, which means that $\theta_j^T x = 0$, the magnitude of θ_j goes to infinity.

4. (True of False) The first term of $J(\theta)$ always increases as we increase λ.

   Solution. This statement is **True**. When adding λ, the cross-entropy loss

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

   will become larger to prevent overfitting.

Part 3: Implementing a k-nearest-neighbor classifier

1. Distance matrix computation with two loops.

   Solution. To solve the problem using two loops, we just compute the distance between the i^{th} test set example and the j^{th} training example, and put that value into $dists(i, j)$.
   1. In the image, each row corresponds to the distances between a specific test example to all the training examples. If a point is bright, it means that the distance is larger than others. So the distinctly bright rows in the image means that: for those test examples, the distance between those and all the training examples are large. Those test samples are more different than others when compared to training examples. Maybe these test examples aren't from any of the classes. Or maybe the test examples are hard to classify.
   2. Similarly, the columns is caused by the fact that, the training examples are different from almost all the test examples.

2. Computer majority label.

   Solution. Result for $k = 1$: Got 137 / 500 correct => accuracy: 0.274000.
   Result for $k = 5$: Got 139 / 500 correct => accuracy: 0.278000.
   We can see that there is a slightly better performance for $k = 5$than with $k = 1$.

3. Distance matrix computation with one loop.

   Solution. Compared to two loops, we compute one entire training data set each time instead of a single example. The result shows that our result is the same as that using two loops. Difference was: 0.000000. Good! The distance matrices are the same

4. Distance matrix computation with no loop.

Solution. The idea comes for the simple equation $(a-b)^2 = a^2 + b^2 - 2ab$. For two matrix we can use the similar method, so we need to do one matrix multiplication, and then do two broadcast sums, which are the square of the two matrix.
The result is same compared to the former methods. We can compare how fast the implementations are. The result is that:
Two loop version took 884.739000 seconds
One loop version took 104.597000 seconds
No loop version took 12.769000 seconds

5. Choosing k by cross validation.

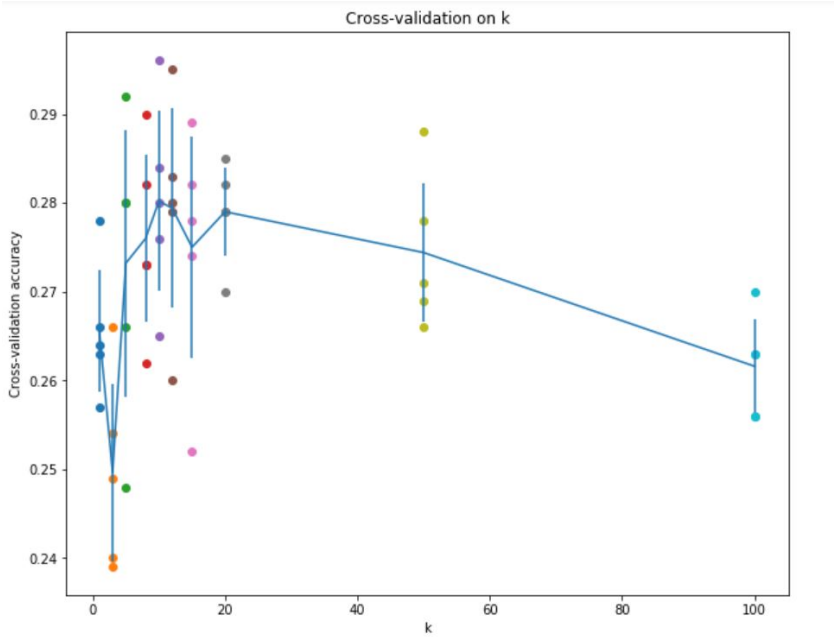

Figure 1: Choosing k by crossvalidation on the CIFAR-10 dataset

Solution. by applying cross-validation, we can get a set of accuracies for all folds. And by changing the value of k, we can also have different accuracies which corresponding to different k.
According to the plot, we choose $k = 8$ as our best value for k and then use this value to retrain the model. We get:
Got 141 / 500 correct => accuracy: 0.282000.
It shows that when $k = 8$ the result is better than other values we have used.

Part 4: Implementing logistic regression

Problem 3A3: Prediction using a logistic regression model

The decision boundaries drawn by our logistic regression program and sklearn are shown in figure 2. Left is the linear boundary drawn by our program. Right is the linear boundary drawn by sklearn package. Results of these two methods are similar.

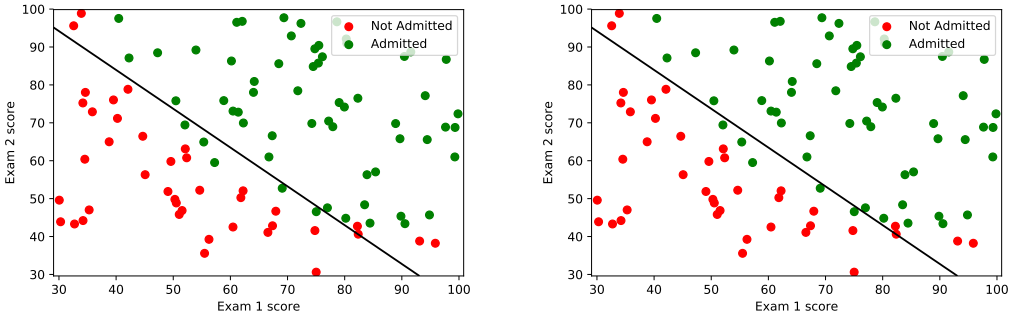

Figure 2: Linear boundaries drawn by our program and by sklearn. Left is the linear boundary drawn by our program. Right is the linear boundary drawn by sklearn package. Results of these two methods are similar.

Problem 3B3: Varying λ

Adjusting different λ, you will see different models. When λ is very large, the model becomes underfitting. When λ is very small(close to zero), the model becomes overfitting. Figure 3 shows results under different λ. Left shows the underfitting model ($\lambda = 100$). Right shows the overfitting model ($\lambda =$ 1e-6).

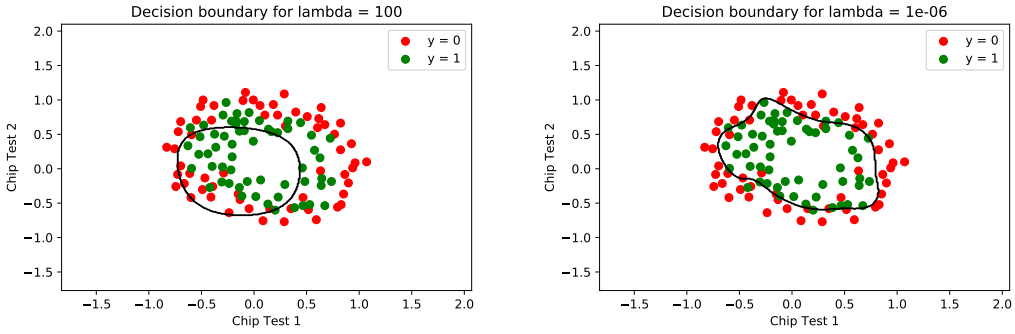

Figure 3: Linear boundaries drawn by our program and by sklearn. Left shows the underfitting model ($\lambda = 100$). Right shows the overfitting model ($\lambda =$ 1e-6).

Problem 3B4: Exploring L1 and L2 penalized logistic regression

Compare L1 and L2 penalized logistic regression with different λ. The results are shown in table 1

λ	L1 Loss	L2 Loss	L1 non-zero θ	L2 non-zero θ
0.01	0.2912	0.3167	19	28
0.1	0.3364	0.3538	14	28
0.3	0.3573	0.3946	8	28
1	0.4382	0.4684	7	28
3	0.6137	0.5496	3	28
10	0.6931	0.6216	0	28

Table 1: Loss and number of non-zero parameters when varying λ

As we can see from the table. We can draw some conclusions:

- When λ increases, the loss of both models increases.

- L1 penalty term makes the model sparse. The larger the λ is, the less non-zero parameters in the trained model. However, L2 penalty term doesn't have this property.
- Loss in L1 model increases more quickly than L2 model.

Fitting regularized logistic regression models (L2 and L1)

1. Do this for every λ in the swept range and return the lambda that yields the highest accuracy.

   Solution. Please see the results in table 2

Regularization	Feature	Best λ	Accuracy
L2	standard feature	0.1	0.9297
L2	log-transformed	0.6	0.9434
L2	binarized feature	1.6	0.9284
L1	standard feature	4.6	0.9219
L1	log-transformed	1.6	0.9440
L1	binarized feature	3.6	0.9258

Table 2: Loss and number of non-zero parameters when varying λ

2. Comment on the model sparsities with L1 and L2 regularization. Which class of models will you recommend for this data set and why?

   Solution. Please see model sparsity table in table 3. The results under L2 regularization are always dense. When using L1 regularization, results in log-transformed features are more sparse than standard features. And results with binarized features are more sparse than log-transformed features.

	# of zeros (std)	# of zeros (log)	# of zeros(bin)
L1 regularization	4	15	20
L2 regularization	0	0	0

Table 3: Loss and number of non-zero parameters when varying λ

I will recommend the model using log transformed feature with L1 regularization($\lambda = 1.6$). Because the accuracy is the highest. And also has many zero parameters and therefore the model is less complex.