CS189/CS289A – Spring 2017 — Homework 4

Yaoyang Zhang, SID 3032114788

Problem 1

(a) According to the lecture note, the gradient of the cost function is

$$\nabla_w J = -X^T(y - s(Xw)) + 2\lambda w$$

where $s(\gamma) = \frac{1}{1+e^{-\gamma}}$ is the sigmoid function.

(b) According to the lecture note, the Hessian of the cost function is

$$\nabla_w^2 J = X^T \Omega X + \lambda I$$

where

$$\Omega = \begin{bmatrix} s_1(1-s_1) & 0 & \cdots & 0 \\ 0 & s_2(1-s_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_n(1-s_n) \end{bmatrix}$$

and $s_i = s(w^T X_i)$

(c) The update equation of Newton's method is

$$w \leftarrow w + (X^T \Omega X + \lambda I)^{-1}(X^T(y - s(Xw)) - 2\lambda w)$$

(d) (i)

$$s^{(0)} = [0.0474, 0, 0, 0]^T$$

(ii)

$$w^{(1)} = [16.2857, 11.7852, 4.2617]^T$$

(iii)

$$s^{(1)} = [0, 0.5267 \times 10^{-24}, 0, 0]^T$$

(iv)

$$w^{(2)} = [-2.0000, 73.9291, 24.3097]^T$$

Problem 2

(a) The cost function can be written as

$$J(w) = (Xw - y)^T(Xw - y) + \lambda \sum_{i=1}^{d} |w_i| = nw^T w - 2w^T X^T y - y^T y + \lambda \sum_{i=1}^{d} |w_i|$$

$$= (y^T y) + \sum_{i=1}^{d} (nw_i^2 + \lambda |w_i| - 2w_i X_{*i}^T y)$$

In this case

$$g(y) = y^T y$$

and

$$f(X_{*i}, w_i, y, \lambda) = nw_i^2 + \lambda |w_i| - 2w_i X_{*i}^T y$$

(b) If $w_i^* > 0$,

$$\frac{\partial J}{\partial w_i} = 2nw_i^* + \lambda - 2X_{*i}^T y = 0$$

$$w_i^* = \frac{X_{*i}^T y}{n} - \frac{\lambda}{2n} > 0$$

(c) If $w_i^* < 0$,

$$\frac{\partial J}{\partial w_i} = 2nw_i^* - \lambda - 2X_{*i}^T y = 0$$

$$w_i^* = \frac{X_{*i}^T y}{n} + \frac{\lambda}{2n} < 0$$

(d) The condition under which $w_i^* = 0$ is

$$\frac{X_{*i}^T y}{n} - \frac{\lambda}{2n} < 0$$

and

$$\frac{X_{*i}^T y}{n} + \frac{\lambda}{2n} > 0$$

or

$$-\frac{\lambda}{2} < X_{*i}^T y < \frac{\lambda}{2}$$

(e) In ridge regression,

$$f(X_{*i}, w_i, y, \lambda) = nw_i^2 + 2\lambda w_i - 2w_i X_{*i}^T y$$

Set derivative to 0

$$\frac{\partial J}{\partial w_i} = 2nw_i^* + 2\lambda - 2X_{*i}^T y = 0$$

$$w_i^* = \frac{X_{*i}^T y - \lambda}{n}$$

If we let $w_i^* = 0$

$$X_{*i}^T y = \lambda$$

In LASSO, the condition under which $w_i^* = 0$ is an interval while in Ridge regression, the condition is one value. Thus, w_i is more likely to be set to 0 in LASSO than in Ridge regression.

Problem 3

(a)

$$\nabla|w|^4 = 4|w|^2 w$$

Let $v = Xw - y$, apply chain rule

$$\nabla|Xw - y|^4 = 4|Xw - y|^2 X^T(Xw - y)$$

(b) Gradient of the cost function is

$$\nabla_w J = 4|Xw - y|^2 X^T(Xw - y) + 2\lambda w$$

Hessian of the cost function is

$$\nabla_w^2 J = 12|Xw - y|^2 X^T X + 2\lambda I$$

Since $X^T X$ is symmetric and positive semidefinite and λ is positive, $\nabla_w^2 J$ will be positive definite and thus the optimal solution will be unique. The optimal solution can be acquired by setting the gradient to 0

$$4|Xw^* - y|^2 X^T(Xw^* - y) + 2\lambda w^* = 0$$

The optimal solution w^* can be rewritten as

$$w^* = -\frac{2}{\lambda}|Xw^* - y|^2 X^T(Xw^* - y) = X^T(\frac{2}{\lambda}|Xw^* - y|^2(y - Xw^*)) = \sum_{i=1}^{n} a_i X_i$$

where $a_i = (\frac{2}{\lambda}|Xw^* - y|^2(y - Xw^*))_i$ or

$$a = \frac{2}{\lambda}|Xw^* - y|^2(y - Xw^*)$$

(c) If L is convex, then the optimal solution can be acquired by setting the gradient of the cost function to 0. Let $v_i = w^T X_i$, apply chain rule

$$\nabla_w J = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial L(v_i, y_i)}{\partial v_i} X_i + 2\lambda w^* = 0$$

Note that $\frac{\partial L(v_i, y_i)}{\partial v_i}$ are scalars, thus the optimal solution can be expressed as

$$w^* = -\frac{1}{2\lambda n}\sum_{i=1}^{n} \frac{\partial L(w^T X_i, y_i)}{\partial (w^T X_i)} X_i = \sum_{i=1}^{n} a_i X_i$$

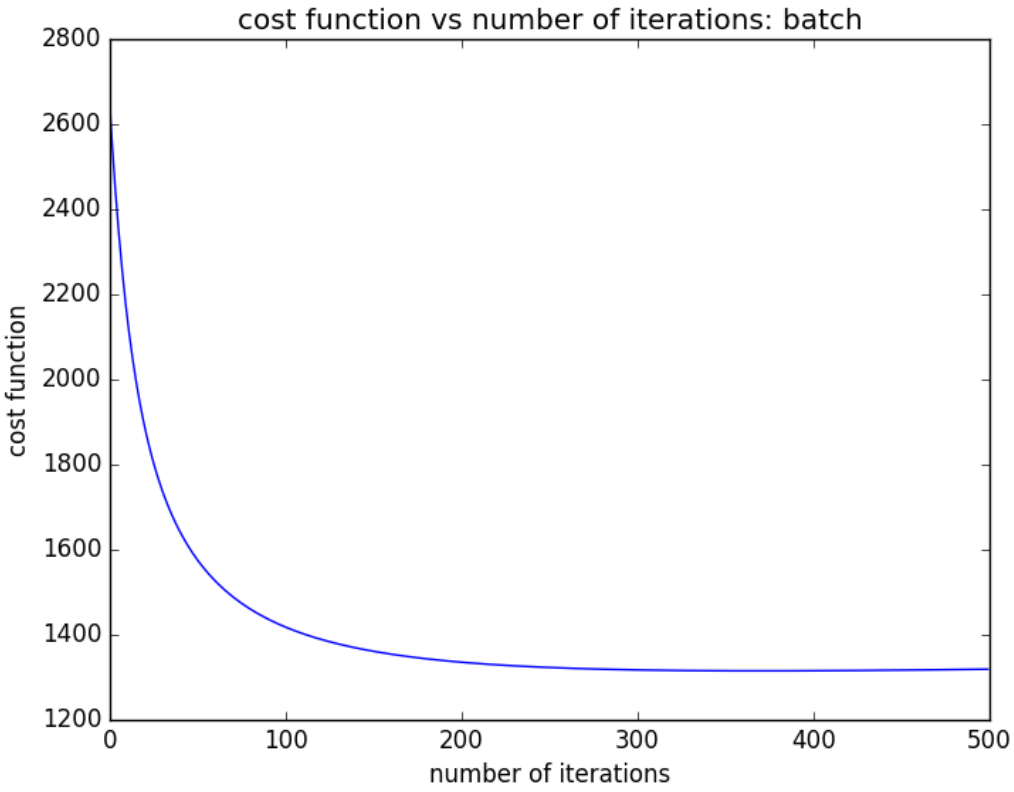
If L is not convex, then the above solution is not necessary optimal, thus we cannot guarantee the existence of such form for the optimal solution.

Problem 4

(a) According to lecture notes, the batch gradient descent update equation for logistic regression with l_2 regularization is

$$w \leftarrow w + \epsilon(X^T(y - s(Xw)) - 2\lambda w)$$

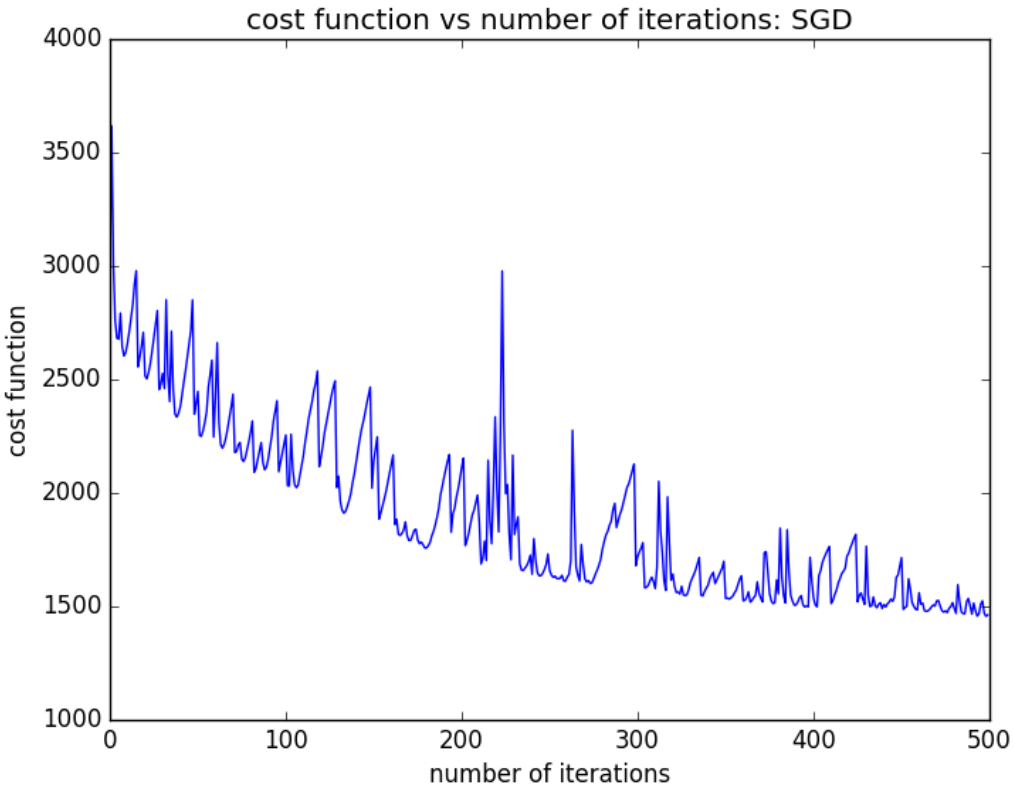
where $s(\gamma) = \frac{1}{1+e^{-\gamma}}$ is the sigmoid function.
Parameters: $\lambda = 0.0001, \epsilon = 0.001$



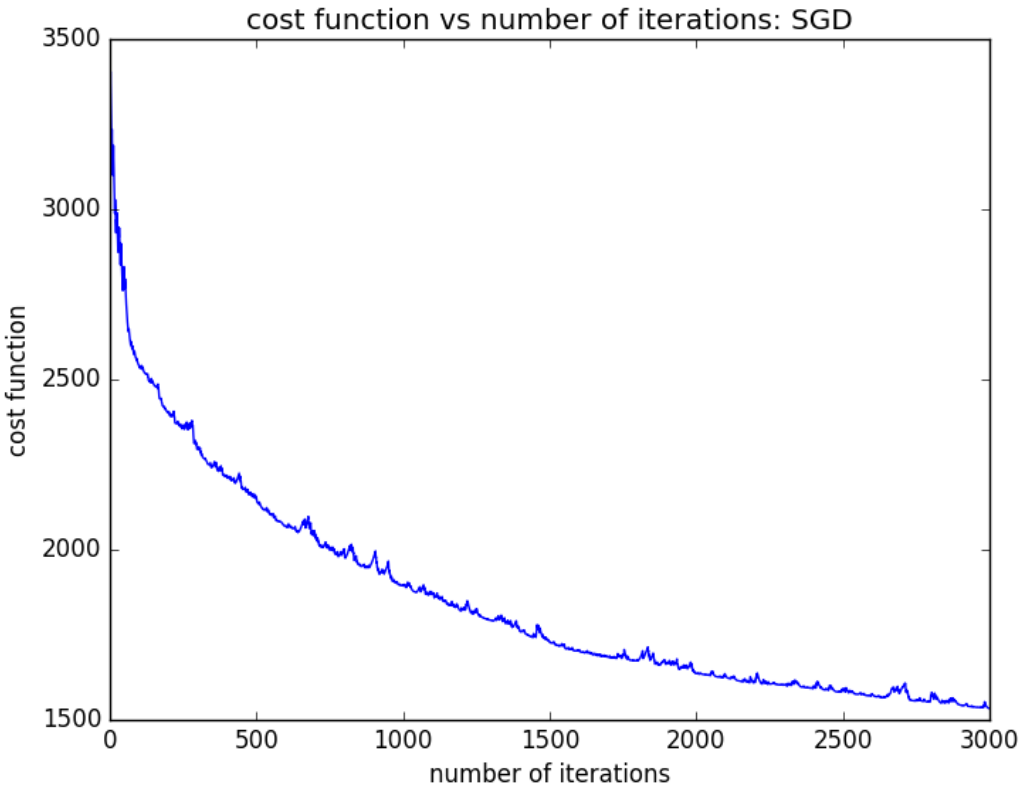
(b) The stochastic gradient descent update equation for logistic regression with l_2 regularization is

$$w \leftarrow w + \epsilon((y_i - s(w^T X_i))X_i - 2\lambda w)$$

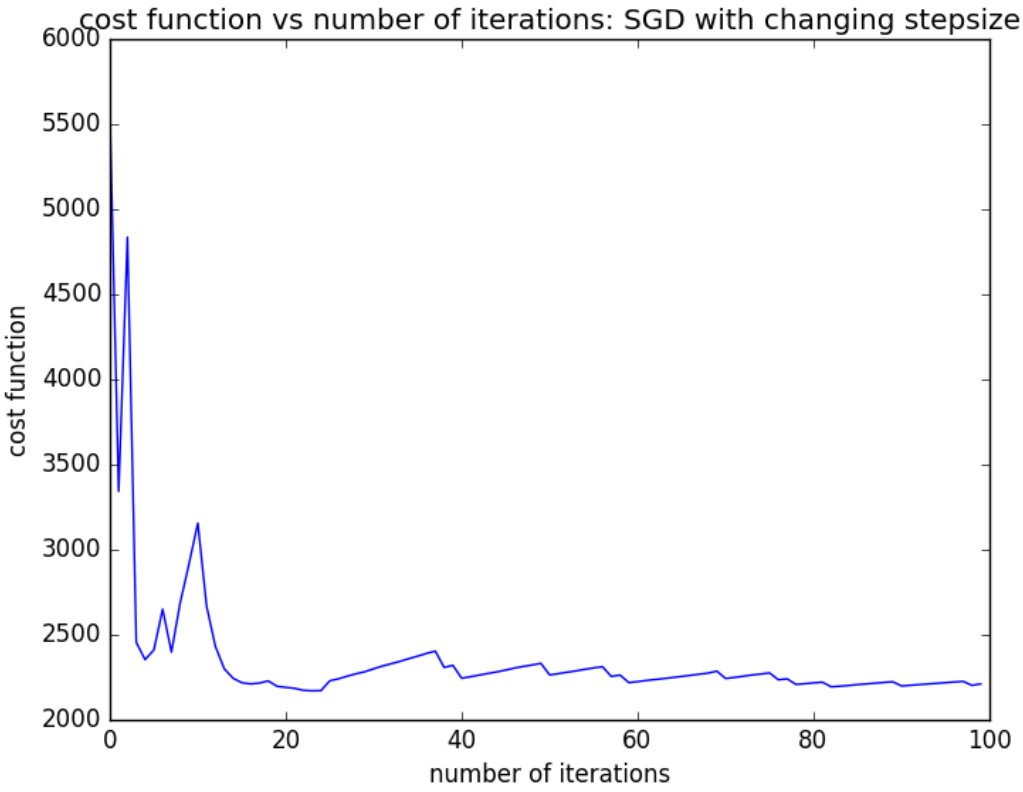
where i is a random index of the training samples at each iteration.
Parameters: $\lambda = 0.0001, \epsilon = 1$



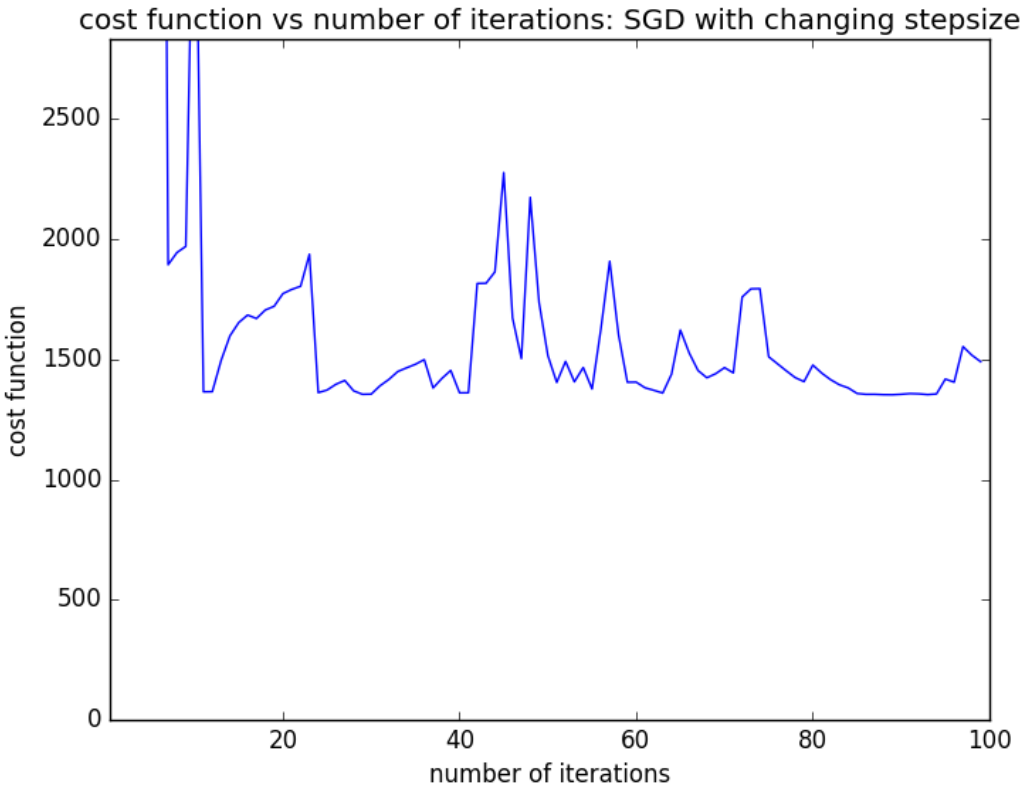
We can observe that the cost function fluctuate a lot. This is due to the randomness of how we calculate the gradient at each iteration. To reduce fluctuation, I chose a smaller $\epsilon = 0.1$. Fluctuation is significantly reduced, but it takes more iterations to converge since we reduce the step size. But SGD is still a lot faster than batch method.



(c) I set $\epsilon = \frac{10}{t}$ and $\lambda = 0.0001$



We can see that it converges much faster than SGD. However, after several steps, the cost value barely drops but obviously it is not the optimal value (compared to the results from SGD, the optimal value should be somewhere around 1500, since we are using the same λ). This is due to the fast decay of step size. After several iterations, the step size becomes very close to 0 so the algorithm is having a hard time to find the optimal value. So I tried to increase $\epsilon = \frac{50}{t}$. This time it gets closer to the optimal value and it is really fast. Compared to a constant ϵ, a decaying step size can get close to the optimal value very fast, but it may not be able to find the optimal value if the step size becomes too small before it gets close enough to the optimal value



(d) For Kaggle, I used the batch method with $\lambda = 0, \epsilon = 0.5$ and ran 200,000 iterations. My score is 0.97581, my name on Kaggle is YaoyangZhang.

Problem 5

Since the spam hits usually appear around midnight, if he uses the amount of time after midnight as a new feature, the spam classes tend to appear around 0 and the maximum value. Thus in the original feature space, a linear classifier performs very bad. What he could do is to instead use x, the time after (or before) noon as a new feature (thus all the spams are located in the most positive or most negative regions) and to use an additional $|x|^2$ feature to lift the sample points so that they can by linearly separated in the new feature space.