Name: UTSAV BARAL **Student ID: 25694452**

CS 189: Introduction to Machine Learning

Homework 4

Due: March 31, 2016 at 11:59pm

Problem 1: Ridge Regression

(a) We will first expand the matrix equation given to us, and see how it simplifies. And we will set the gradient, $\nabla J = 0$ and solve for values of w and α which minimize this modified squared loss function. Starting with the cost function: $J(\mathbf{w}, \alpha) = (X\mathbf{w} + \alpha\mathbf{1} - \mathbf{y})^t(X\mathbf{w} + \alpha\mathbf{1} - \mathbf{y}) + \lambda\mathbf{w}^t\mathbf{w}$ We can see right off the bat that the complicated looking part with the transpose of the expression multiplied with itself is the expression describing the squared loss, and the added expression, is how ridge regression attempts to keep variance low by punishing very large weight values, and thus helping with potential overfitting problems. Specifically, we have that:

$$\begin{aligned} J(\mathbf{w}, \alpha) &= (X\mathbf{w} + \alpha\mathbf{1} - \mathbf{y})^t(X\mathbf{w} + \alpha\mathbf{1} - \mathbf{y}) + \lambda\mathbf{w}^t\mathbf{w} \\ &= \begin{bmatrix} X_{11}w_1 + \ldots + X_{1d}w_d + \alpha - y_1 \\ X_{21}w_1 + \ldots + X_{2d}w_d + \alpha - y_2 \\ \vdots \\ X_{n1}w_1 + \ldots + X_{nd}w_d + \alpha - y_n \end{bmatrix}^t \begin{bmatrix} X_{11}w_1 + \ldots + X_{1d}w_d + \alpha - y_1 \\ X_{21}w_1 + \ldots + X_{2d}w_d + \alpha - y_2 \\ \vdots \\ X_{n1}w_1 + \ldots + X_{nd}w_d + \alpha - y_n \end{bmatrix} + (\lambda w_1^2 + \ldots + \lambda w_d) \\ &= \sum_{i=1}^{n} (X_{i1}w_1 + \ldots + X_{id}w_d + \alpha - y_i)^2 \ + (\lambda w_1^2 + \ldots + \lambda w_d) \end{aligned}$$

Now that its actually readable, we can take the partials with respect to $\mathbf{w}$ and α and set equal to zero to find where the grad is 0 and where we have a critical minimum point. We can write the equation again, in terms of a sum of all the row vectors:

$$J = \sum_{i=1}^{n} (X_i\mathbf{w} + \alpha - y_i)^2 \ + \ (\lambda w_1^2 + \ldots + \lambda w_d)$$

$$\frac{\partial J}{\partial \mathbf{w}} = \sum_{i=1}^{n} 2 \begin{bmatrix} X_{i1} \\ \vdots \\ X_{id} \end{bmatrix} (X_i\mathbf{w} + \alpha - y_i) \ + \ 2\lambda \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

Since each X_i row ends up geting transposed, the whole sum can just be written as a big matrix with where the design matrix gets transposed and left multiplied to acheive the sum. We then expand the resulting expression out:

$$\frac{\partial J}{\partial \mathbf{w}} = 2X^tX\mathbf{w} + 2X^t\alpha - 2X^ty \ + \ 2\lambda\mathbf{w}$$

Setting this simplified expression to 0, we then get that:

$$X^t y = X^t X\mathbf{w} + X^t\alpha + \lambda\mathbf{w}$$

Another simplification can be made, since $X^t\alpha$ contains elements that are the column sums of the design matrix times some constant, this must be zero! Since we are given that the mean of x is 0. Thus we can rewrite again, factor the $\mathbf{w}$ vector and get our answer for $\mathbf{w}$:

$$\begin{aligned}
X^t y &= X^t X\mathbf{w} + \lambda\mathbf{w} \\
X^t y &= X^t X\mathbf{w} + (\lambda\mathbf{I})\mathbf{w} \\
X^t y &= (X^t X + \lambda\mathbf{I})\mathbf{w} \\
\therefore \mathbf{w} &= (X^t X + \lambda\mathbf{I})^{-1} X^t y \quad \blacksquare
\end{aligned}$$

Finding the value for α is pretty easy since we have a just a normal partial derivative without complicated vectors, we set the partial = 0 and simplify:

$$\begin{aligned}
\frac{\partial J}{\partial \alpha} &= 2\sum_{i=1}^{n} X_i\mathbf{w} \; + \; 2(n\cdot\alpha) + 2\sum_{i=1}^{n} -y_i = 0 \\
\sum_{i=1}^{n} y_i &= \sum_{i=1}^{n} X_i\mathbf{w} \; + \; (n\cdot\alpha)
\end{aligned}$$

We are treating w as a constant, since this is a partial derivative, and again, since we know that the mean of x is 0, that whole sum just drops out, and we are left with:

$$\begin{aligned}
\sum_{i=1}^{n} y_i &= n\cdot\alpha \\
\therefore \alpha &= \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y} \quad \blacksquare
\end{aligned}$$

(b)

i. See APPENDIX A.

ii. [$\lambda = 106$ to $\lambda = 136$] were where good ranges of hyperparamaters occured. The residual sum os squared on this malication set was...

iii. Here is a plot generated using python (see APPENDIX B for code):

Problem 2: Logistic Regression.

****See APPENDIX C for Code****

Design Matrix ($\mathbf{X}$) = $\begin{bmatrix} 0 & 3 & 1 \\ 1 & 3 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$; Labels($\mathbf{y}$) = $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$; $\mathbf{w}^{(0)} = \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}$

(a.) $R(\mathbf{w}^{(0)}) = 1.9883724141284105$

(b.) $\mu^{(0)} = \begin{bmatrix} 0.95257413 \\ 0.73105858 \\ 0.73105858 \\ 0.26894142 \end{bmatrix}$

(c.) $\mathbf{w}^{(1)} = \begin{bmatrix} -2 \\ 0.94910188 \\ -0.68363271 \end{bmatrix}$

(d.) $R(\mathbf{w}^{(1)}) = 1.720617095621304$

(e.) $\mu^{(1)} = \begin{bmatrix} 0.89693957 \\ 0.54082713 \\ 0.56598026 \\ 0.15000896 \end{bmatrix}$

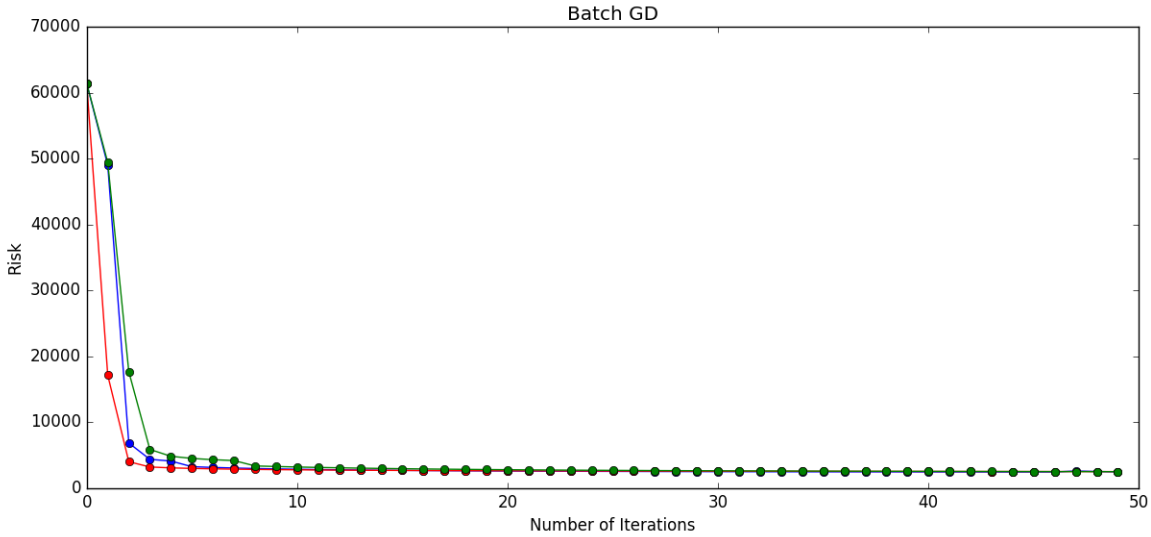
(f.) $\mathbf{w}^{(2)} = \begin{bmatrix} -1.69083609 \\ 1.91981257 \\ -0.83738862 \end{bmatrix}$

(g.) $R(\mathbf{w}^{(2)}) = 1.8546997847922464$

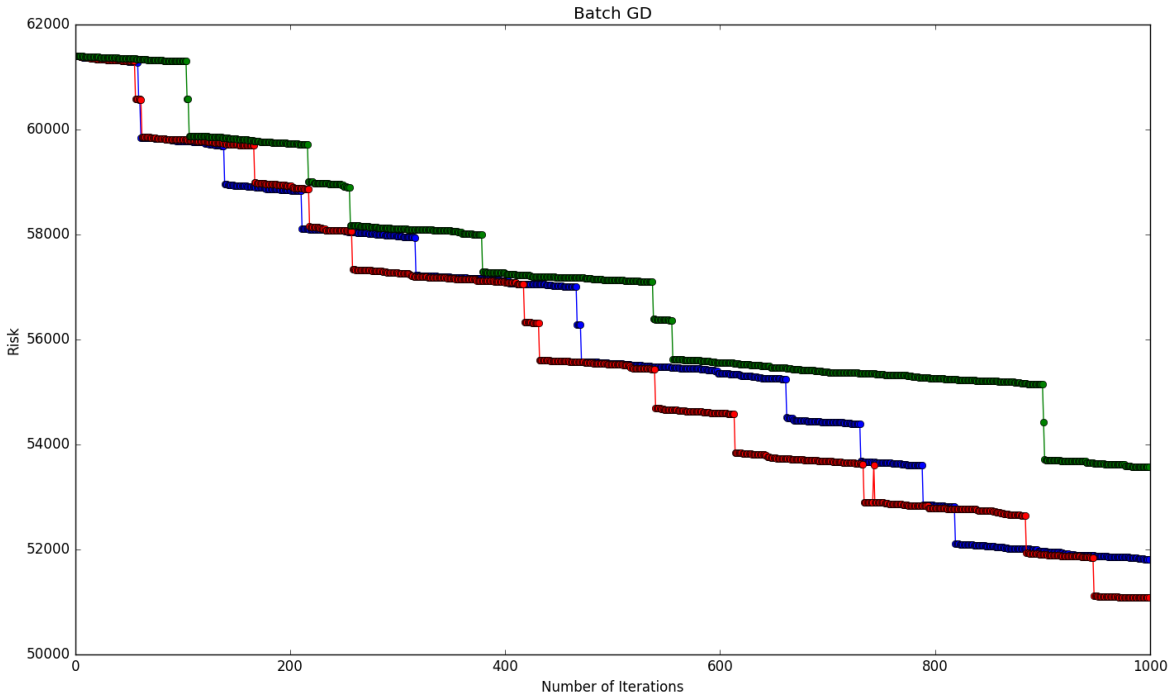
(h.) $\mu^{(2)} = \begin{bmatrix} 0.99276849 \\ 0.96199213 \\ 0.74695242 \\ 0.35242149 \end{bmatrix}$

Problem 3: Spam classification using Logistic Regression.

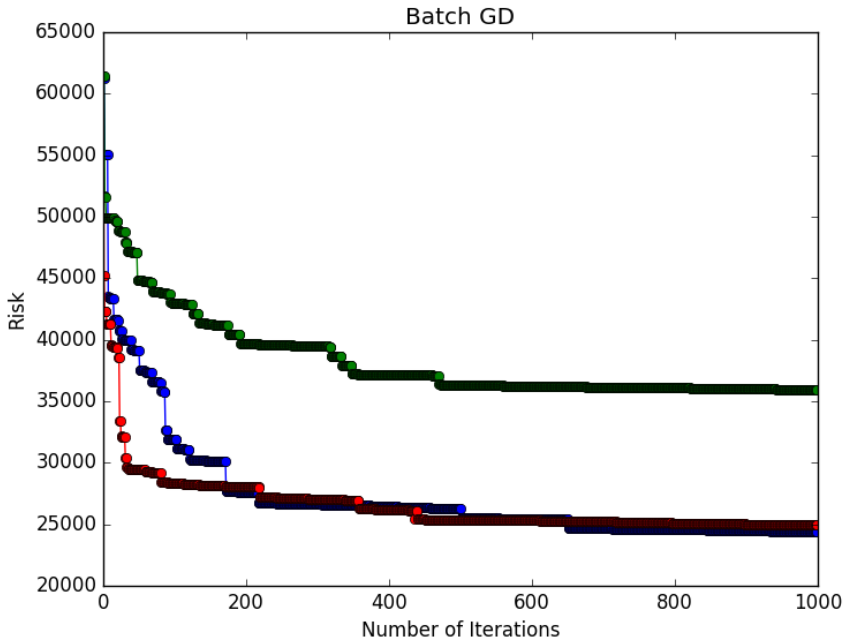
(1.) Batch Gradient Descent, 60 iterations until it started converging:



(2.) Stochastic Gradient Descent,(Title Says Batch on Accident):



(3.) Stochastic Gradient Descent, with Decaying Learning Rate:



(4.)

(5.) Clearly, batch gradient descent worked the best, maybe due to better hyperparameter maniupulation, or possible/probably error in code of the other methods. I will use batch gradient descent with log preprocessing for kaggle. SCORE=0.76422

Problem 4: Revisiting Logistic Regression.

(a) Replacing $\tanh(z)$ with the corresponding function we have that $g(z) = \frac{(2s(2z)+1)-1}{2} = s(2z)$ So we can just simplify this expression:

$$s(2z) = \frac{1}{1+e^{-2z}} = \frac{1}{1+e^{-2z}} \cdot \frac{e^z}{e^z}$$

$$= \frac{e^z}{e^z + e^{-z}} = \frac{2e^z}{2(e^z + e^{-z})} = \frac{e^z - e^{-z} + e^z + e^{-z}}{2(e^z + e^{-z})}$$

$$= \frac{e^z - e^{-z}}{2(e^z + e^{-z})} + \frac{e^z + e^{-z}}{2(e^z + e^{-z})}$$

$$= \frac{e^z - e^{-z}}{2(e^z + e^{-z})} + \frac{1}{2} \quad \blacksquare$$

(b) With simple application of the chain rule, we see that $\frac{dg}{dz} = \frac{1}{2}\frac{dh}{dz} = 2\frac{ds}{dz}$ We can expand this expression to see that:

$$2\frac{ds}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} =$$

(c)

Problem 5: Real World Spam Classification.