CS 289 Homework 4

SID: 24978491

March 6, 2015

Problem 1:

1. The negative log likelihood function is:

$$l\left(\beta\right)=\lambda\left\|\beta\right\|_2^2-\sum_{i=1}^{n}\left[y_i log\mu_i+\left(1-y_i\right)log\left(1-\mu_i\right)\right]$$

$$=\lambda\beta^T\beta+\sum_{i=1}^{n}\left[log\left(1+exp\left(-\beta^Tx_i\right)\right)+\left(1-y_i\right)\beta^Tx_i\right]$$

Based on multivariate calculus, the gradient of the negative log likelyhood function is:

$$\nabla_\beta l\left(\beta\right)=2\lambda\beta+\sum_{i=1}^{n}\left[\frac{-exp\left(-\beta^Tx_i\right)x_i}{1+exp\left(-\beta^Tx_i\right)}+\left(1-y_i\right)x_i\right]$$

2. The Hessian of the negative log likelihood should be:

$$\nabla_\beta^2 l\left(\beta\right)=2\lambda I+\sum_{i=1}^{n}\frac{exp\left(-\beta^Tx_i\right)x_ix_i^T}{\left(1+exp\left(-\beta^Tx_i\right)\right)^2}$$

3. The update equation for newton's method is:

$$\beta^{(n+1)}=\beta^{(n)}-\nabla_\beta^2 l\left(\beta^{(n)}\right)^{-1}\nabla_\beta l\left(\beta^{(n)}\right)$$

$$=\beta^{(n)}-\left(2\lambda I-\sum_{i=1}^{n}exp\left(-\beta^{(n)T}x_i\right)x_ix_i^T\right)^{-1}\left(2\lambda\beta^{(n)}+\sum_{i=1}^{n}\left[\frac{-x_i}{1+exp\left(-\beta^{(n)T}x_i\right)}+\left(1-y_i\right)x_i\right]\right)$$

4. When $\lambda=0.07$, $\beta^{(0)}=\left[\begin{array}{ccc}-2 & 1 & 0\end{array}\right]^T$,

a) The value of $\mu^{(0)}$can be calculated by:

$$\mu_i^{(0)}=\frac{1}{1+exp\left(-\beta^Tx_i\right)}$$

If the number 1 is appended to the right of the desigen matrix, the value of $\mu^{(0)}$ is:

$$\mu^{(0)}=\left[\begin{array}{cccc}0.9526 & 0.7311 & 0.7311 & 0.2689\end{array}\right]^T$$

b) The value of β after one iteration is:

$$\beta^{(1)} = \left[\begin{array}{ccc} -0.3836 & 1.4043 & -2.2842 \end{array}\right]^T$$

c) The value of $\mu^{(1)}$is

$$\mu^{(1)} = \left[\begin{array}{cccc} 0.8731 & 0.8238 & 0.2932 & 0.2918 \end{array}\right]^T$$

d) The value of $\beta^{(1)}$ is:

$$\beta^{(2)} = \left[\begin{array}{ccc} -0.5122 & 1.4527 & -2.1627 \end{array}\right]^T$$

Problem 2:

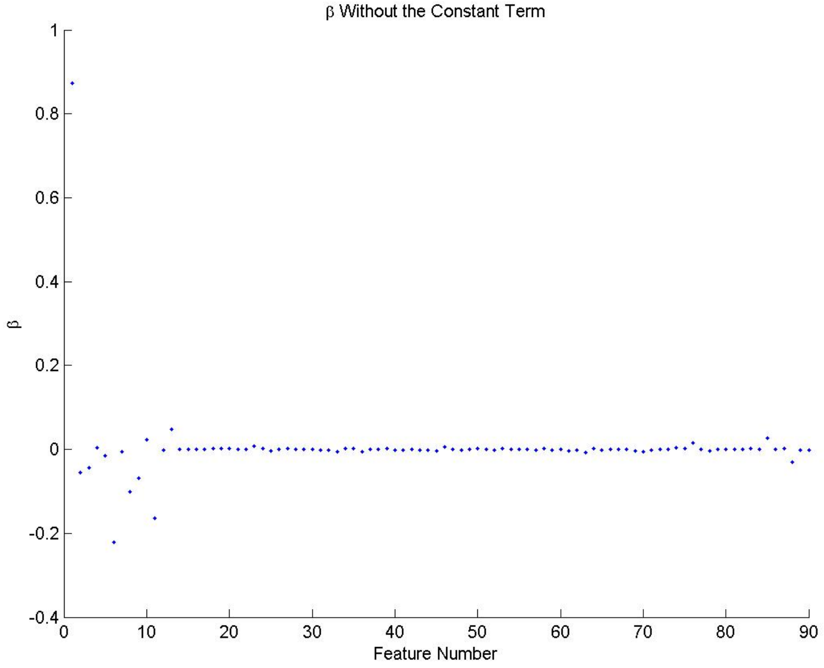
1. The linear regression with least squares is implemented in Matlab. The code for linear regression is in Problem2.m.

2. The trained model is tested on the test set. The Residual Sum of Square is:

$$RSS = 4.6696 \times 10^6$$

The minimum value of the predited year is 2045.5, and the minimum of the predicted year is 1953.9. The range of the data does not make sense. The year of the song cannot be 2045.

3. The regression coefficients β (without the constant term, which is 1951.1 and much larger than the other coefficients) are plotted in the following graph.



4. Linear regression is not a reasonable method to predict the year of a song. First, the range of the prediction should not exceed the current year. Moreover, we can see that in linear regression, most of the coefficients are closed to zero which means the significance of the corresponding features is relatively low. Thus, linear regression is not a reasonable method for this problem.

Problem 3:

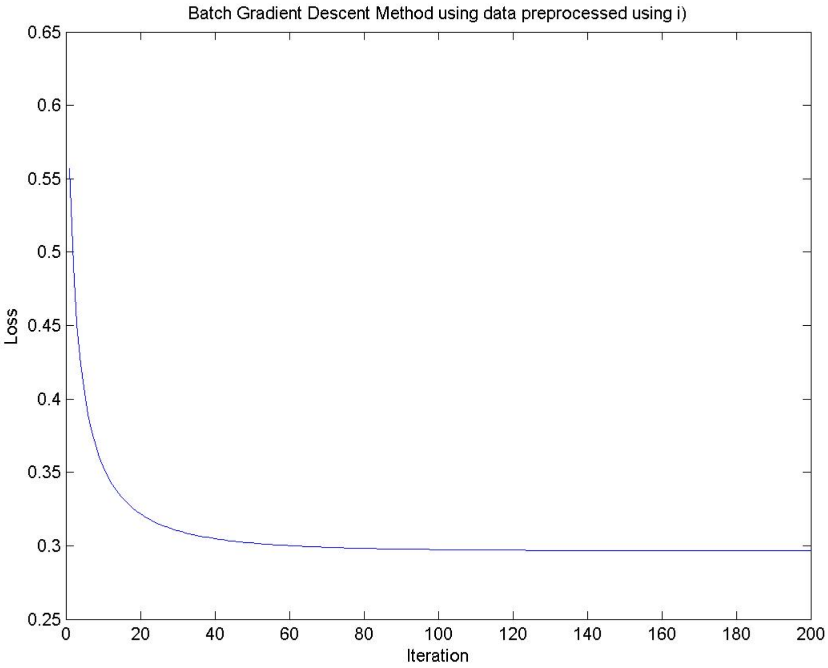
1. To minimize the logistic function, it is same to maximize the negative log likelihood function. The batch gradient descent equations for logistic regression with $l_2$regularization is:

$$\beta^{(n+1)} = \beta^{(n)} - \alpha \nabla_\beta l\left(\beta\right)$$

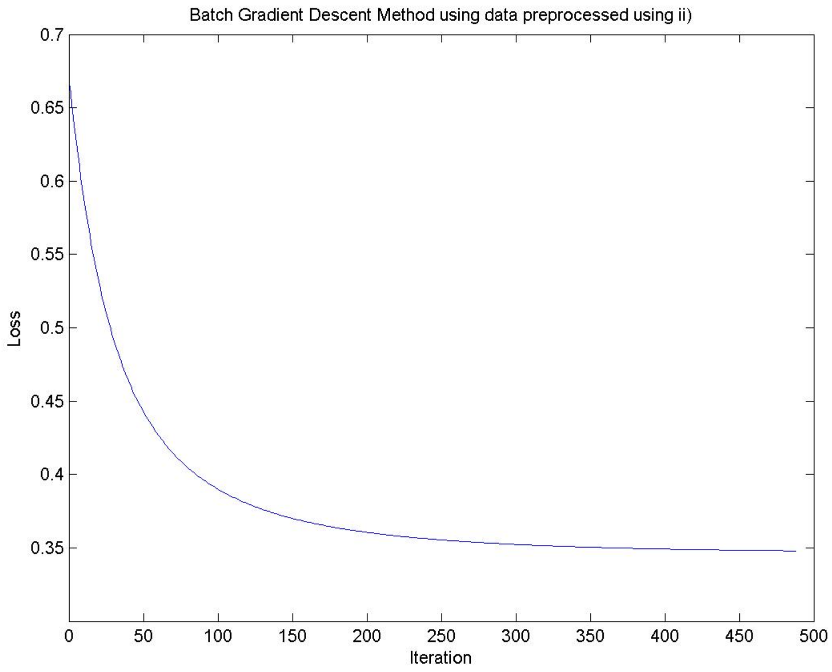
where α is the step size and the gradient $\nabla_\beta l\left(\beta\right)$ is similar to that in Problem 1:

$$\nabla_\beta l\left(\beta\right) = 2\lambda\beta + \frac{1}{n}\sum_{i=1}^{n}\left[\frac{-exp\left(-\beta^T x_i\right)x_i}{1+exp\left(-\beta^T x_i\right)} + \left(1-y_i\right)x_i\right]$$

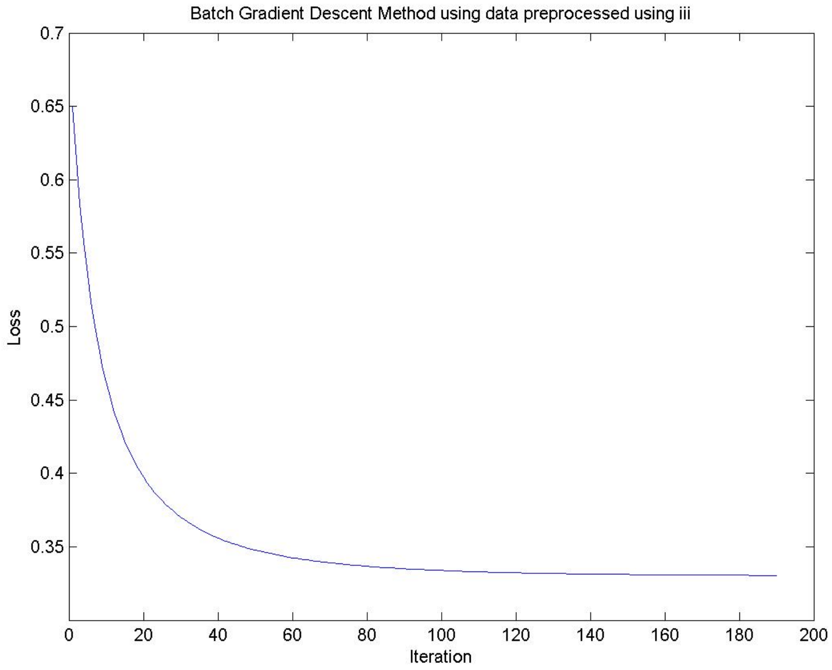
i) When the data is preprocessed using the rule defined in i), the parameter of the logistic regression is $\lambda = 0.01$, and $\alpha = 0.05$, the loss of the batch gradient method is plotted in the following graph.



ii) When the data is preprocessed using ii), the loss in different iteration for batch gradient descent method is plotted in the following graph.



iii) When the data is preprocessed using iii), the loss in different iteration for batch gradient descent method is plotted in the following graph.



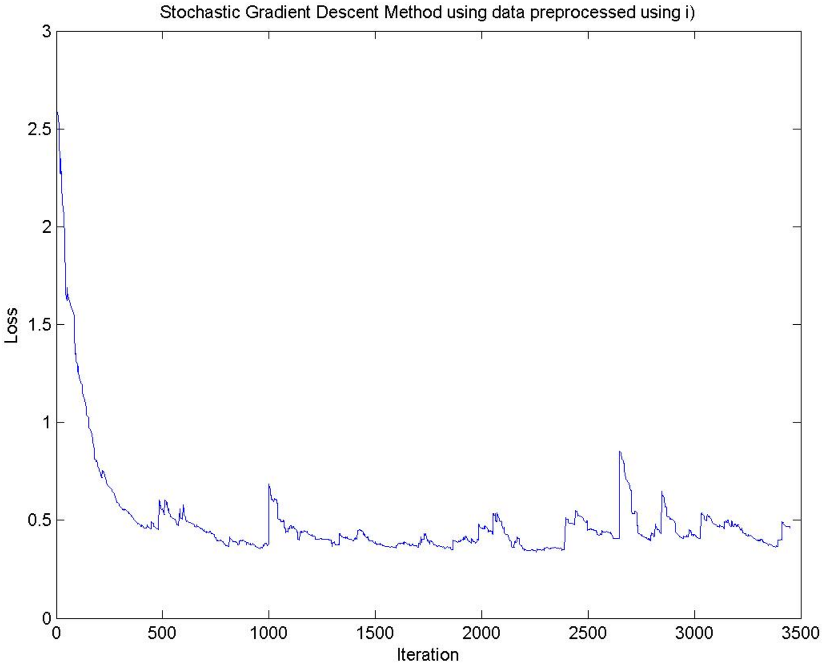
2. To minimize the logistic function, it is same to maximize the negative log likelihood function. The stochastic

gradient descent equations for logistic regression with $l_2$regularization is:

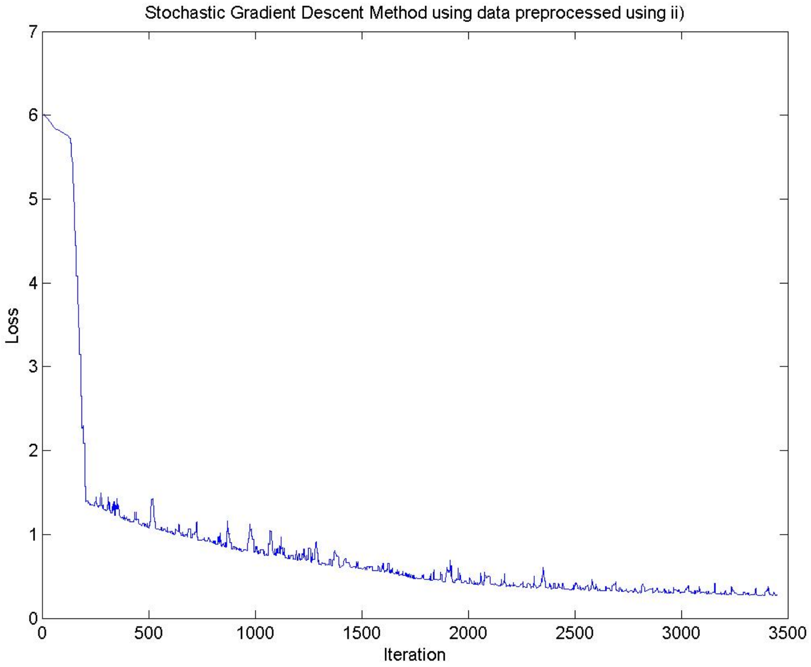
$$\beta^{(n+1)} = \beta^{(n)} - \alpha\left(2\lambda\beta + \left(\frac{-exp\left(-\beta^T x_i\right)x_i}{1+exp\left(-\beta^T x_i\right)} + \left(1-y_i\right)x_i\right)\right)$$

where α is the step size and i is randomly picked from the training set.

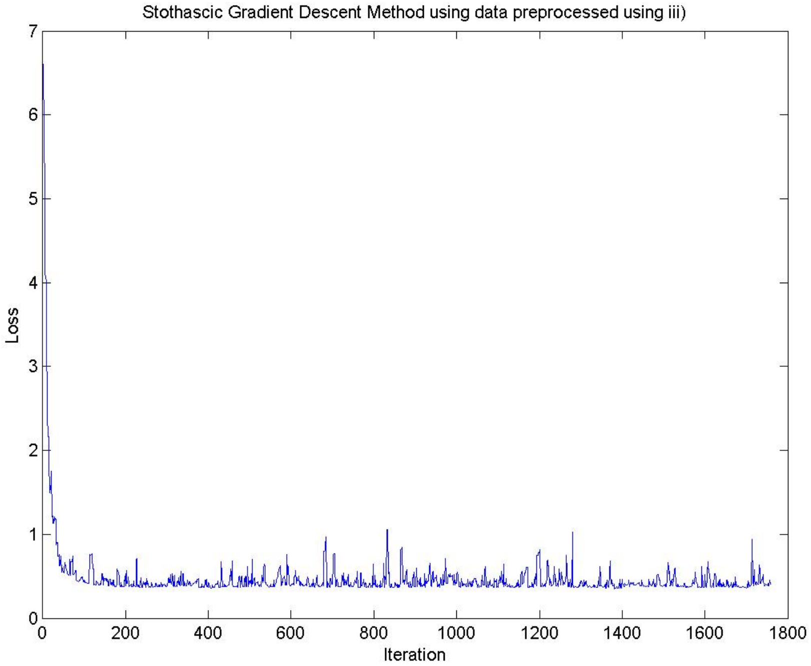
i) When the data is preprocessed using the rule defined in i), the parameter of the logistic regression is $\lambda = 0.01$, and $\alpha = 0.1$, the loss of the batch gradient method is plotted in the following graph.



ii) When the data is preprocessed using ii), the loss in different iteration for batch gradient descent method is plotted in the following graph.The parameter of the logistic regression is $\lambda = 0.01$, and $\alpha = 0.005$



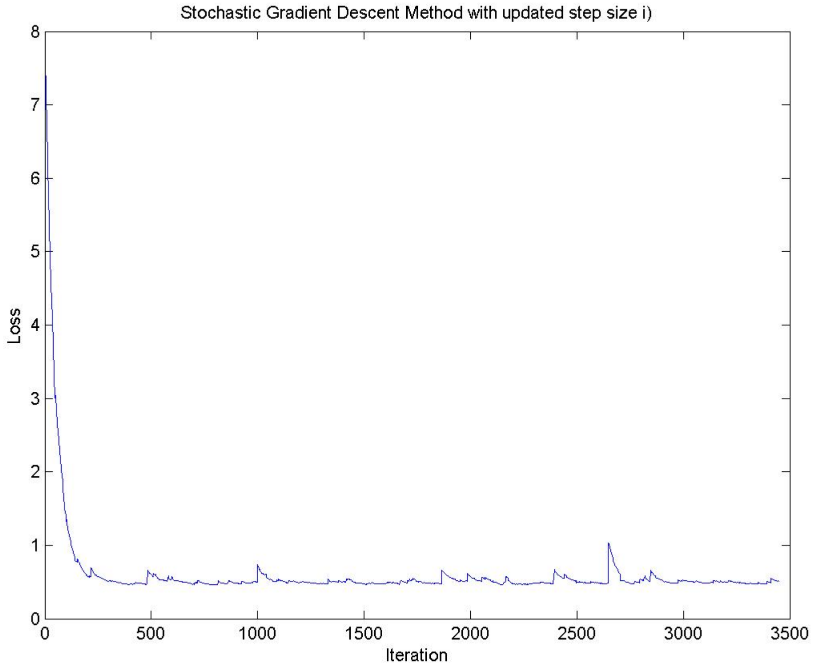
iii) When the data is preprocessed using iii), the loss in different iteration for batch gradient descent method is plotted in the following graph.The parameter of the logistic regression is $\lambda = 0.01$, and $\alpha = 0.3$.



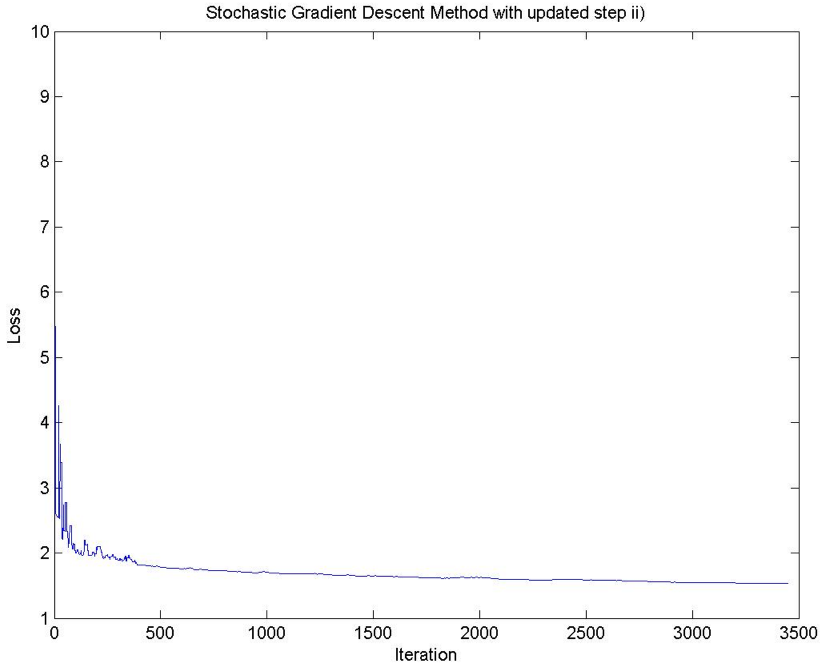
In stochastic gradient descent method, the loss function has oscillation while the loss of batch gradient method is smooth. In 1, when we calculate the gradient, all data in the training set are used by taken the gradient of the loss with respect to them and then taking the average. However, in stochastic gradient method, in each iteration, we only use one data points, so the loss may increase in some iterations since the gradient we calculated is not the actual gradient of the loss function.

3. This strategy is better than having constant step size. The reason for that with the step size $\alpha = \frac{1}{i}$ where i is the iteration number, we can ensure that the algorithm will converge to the local optima. Intuitively, when we are close to the local optima, we will slow down by shrink the step size so we will never miss the optima. Thus, it is better than the stochastic gradient method.

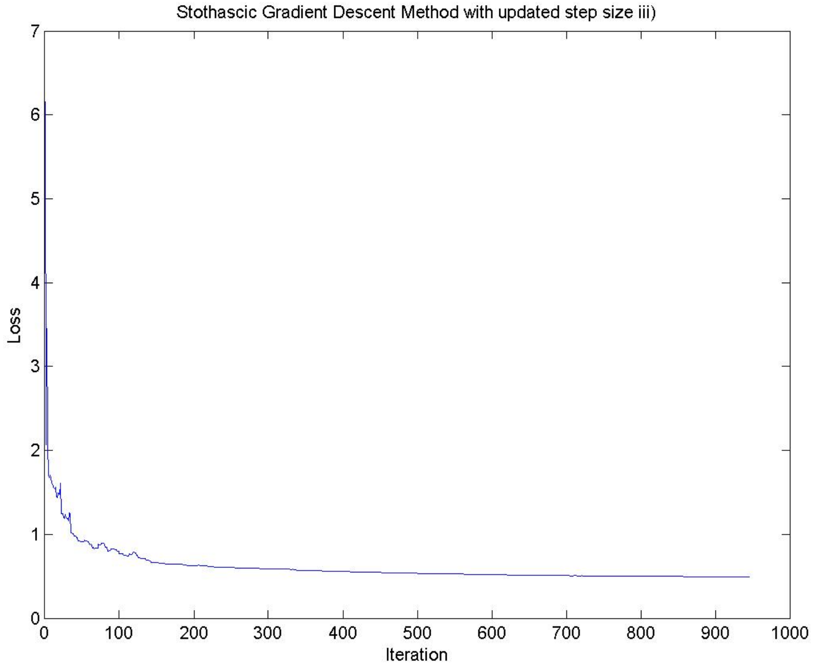
i) When the data is preprocessed using the rule defined in i), the parameter of the logistic regression is $\lambda = 0.1$, and $\alpha = 0.05$, the loss of the batch gradient method is plotted in the following graph.



ii) When the data is preprocessed using ii), the loss in different iteration for batch gradient descent method is plotted in the following graph.The parameter of the logistic regression is $\lambda = 0.01$, and $\alpha = 1$



iii) When the data is preprocessed using iii), the loss in different iteration for batch gradient descent method is plotted in the following graph.The parameter of the logistic regression is $\lambda = 0.01$, and $\alpha = 5$.


Stothascic Gradient Descent Method with updated step size iii)
Loss
Iteration