# Computational Data Analytics ISYE 6740

## Homework 3

**Submitted by: Arjun Mishra**

Q.1. Gradient Descent for Multiple Linear Regression

(A) (B)

(a) Our optimized parameters:

$$\beta^* = \underset{\beta \in R^p}{\operatorname{argmin}} f(\beta)$$

where $f(\beta) = \frac{1}{2n} \|Y - X\beta\|_2^2$ is the cost function.

To get the gradient:

$$\nabla f(\beta) = \frac{\partial}{\partial \beta} \left( \frac{1}{2n} \|Y - X\beta\|_2^2 \right)$$

$$= \frac{1}{2n} \times -2X^T (Y - X\beta)$$

$$\nabla f(\beta) = \frac{1}{n} X^T (X\beta - Y)$$

(b) We can get the step size from the Lipschitz constant. For $L$:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \|x - y\|_2 \quad \text{(Lipschitz continuity)}$$

can be written as:

$$\|\nabla^2 f(\beta)\| \leq L$$

as $f(\beta)$ is convex & twice differentiable.

(A), (B)

→ $\nabla^2 f(\beta)$ is the Hessian of $f(\beta)$ and is a positive, semidefinite matrix.

∴ The norm is given by the largest eigenvalue

$$L \geq \lambda_{max} \{ \nabla^2 f(\beta) \}$$

Hessian of $f(\beta) = X^T X$

$$L \geq \lambda_{max} \{ X^T X \}$$

& step size $\alpha = 1/L$ :

$$\alpha = \frac{1}{L} \leq \frac{1}{\lambda_{max}(X^T X)}$$

In this equality, the largest possible $\alpha$ :

$$\alpha = \frac{1}{\lambda_{max}(X^T X)}$$

For the given dataset :

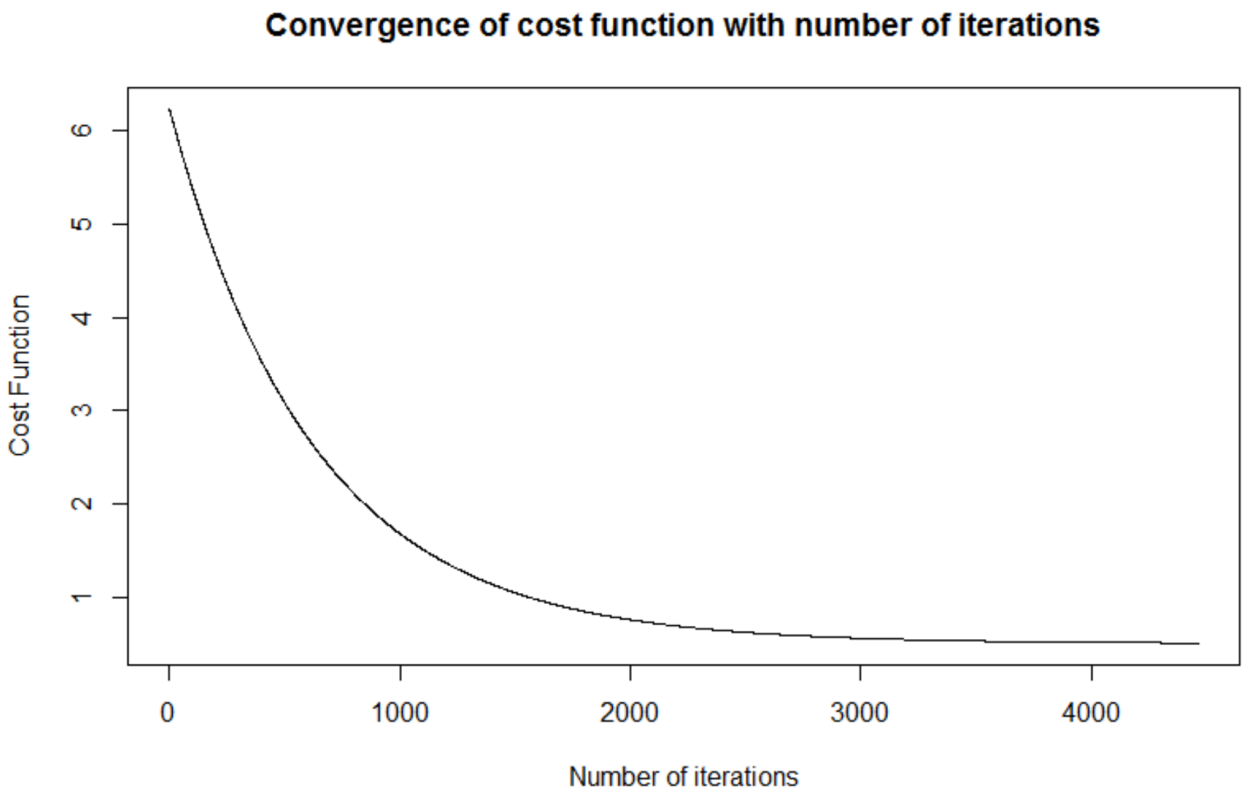
$$X^T X = 1292.95$$

$$\therefore \alpha = 7.73 \times 10^{-4}$$

(C) For stopping the gradient descent algorithm at a suitable stage, I am using the cost function values computed at each stage. I have finally arrived at a reasonable small threshold value of 0.00001 after testing and research. When the absolute difference in the cost function values between iteration (i) and (i-1) is lesser than the threshold, I stop running the iterations. This logic is straightforward – when there isn't a lot of improvement in the cost functions over iterations, stop the loop. If the threshold is small enough, we can get a good approximate minimum of the function in a small amount of time.

(D)

Plot of convergence for the cost function for normal gradient descent algorithm:




(E) The mean squared error between the true regression coefficients and the ones calculated by gradient descent is: 0.001120878

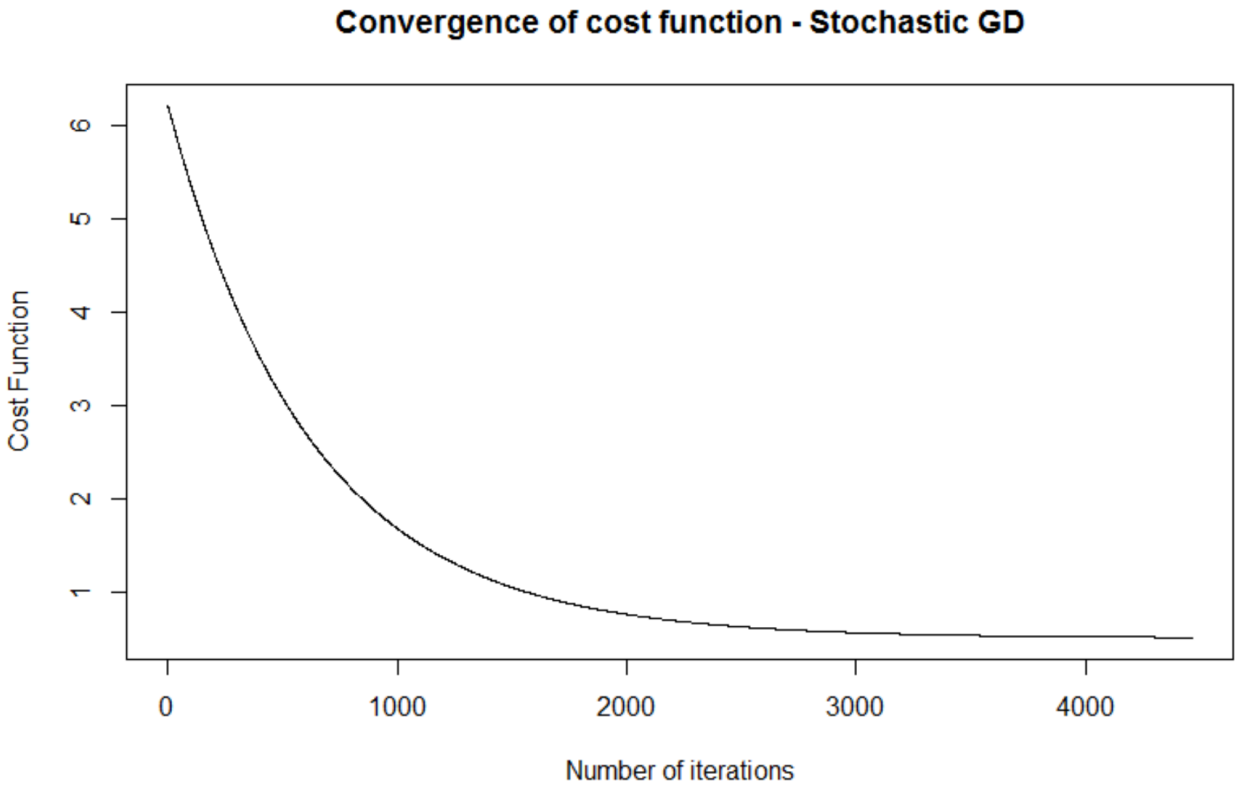
## Q.2. Stochastic Gradient Descent for Multiple Linear Regression

(A)

In stochastic gradient descent, we do not take one run on the whole dataset to update the gradient but just a single observation. First, we reduce the step size by a factor equaling the number of statistical units. At each iteration, the regression coefficients β* are updated based on the gradient at one statistical unit. The dataset is shuffled at each pass.

At the end of each iteration as defined above, the cost function was calculated as the average of the costs evaluated for all the statistical units within that iteration. We have used this average of costs for plotting against the number of iterations. For stopping the algorithm, we have used the absolute difference of the average of the cost function and the same threshold of 0.00001.

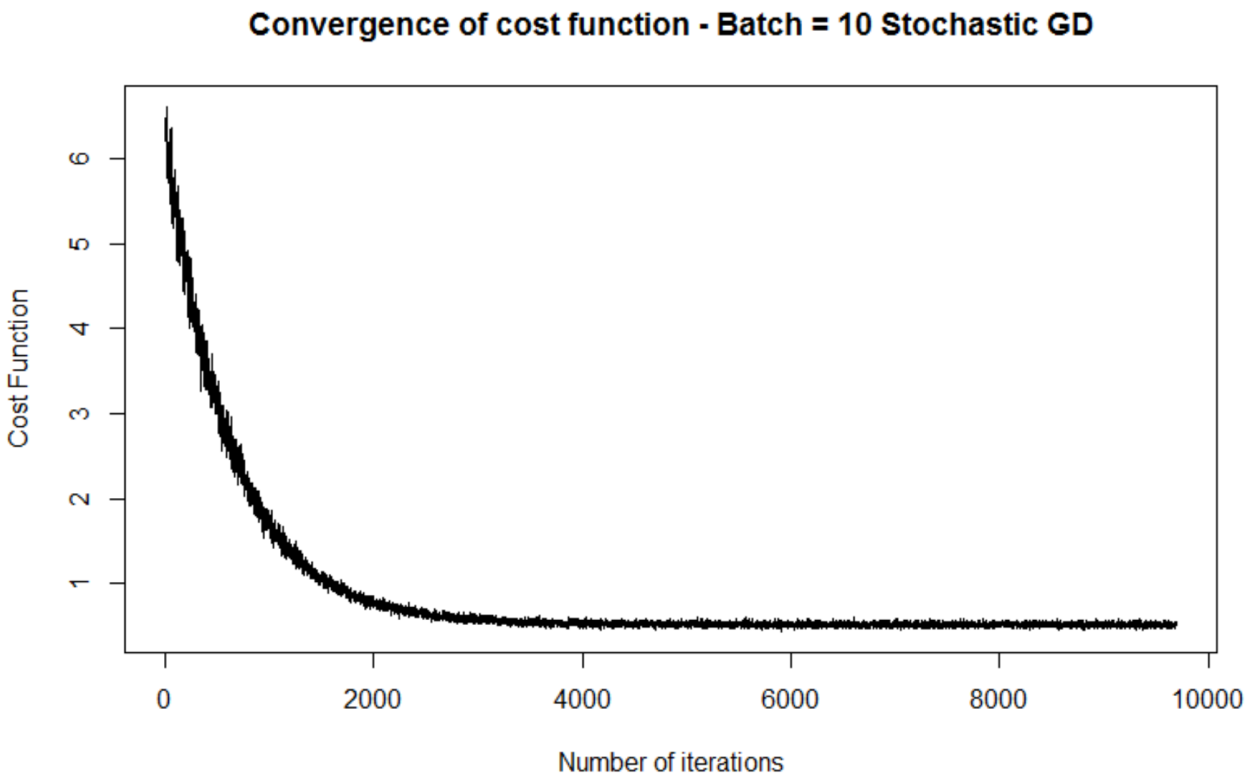
The convergence plot is as below:

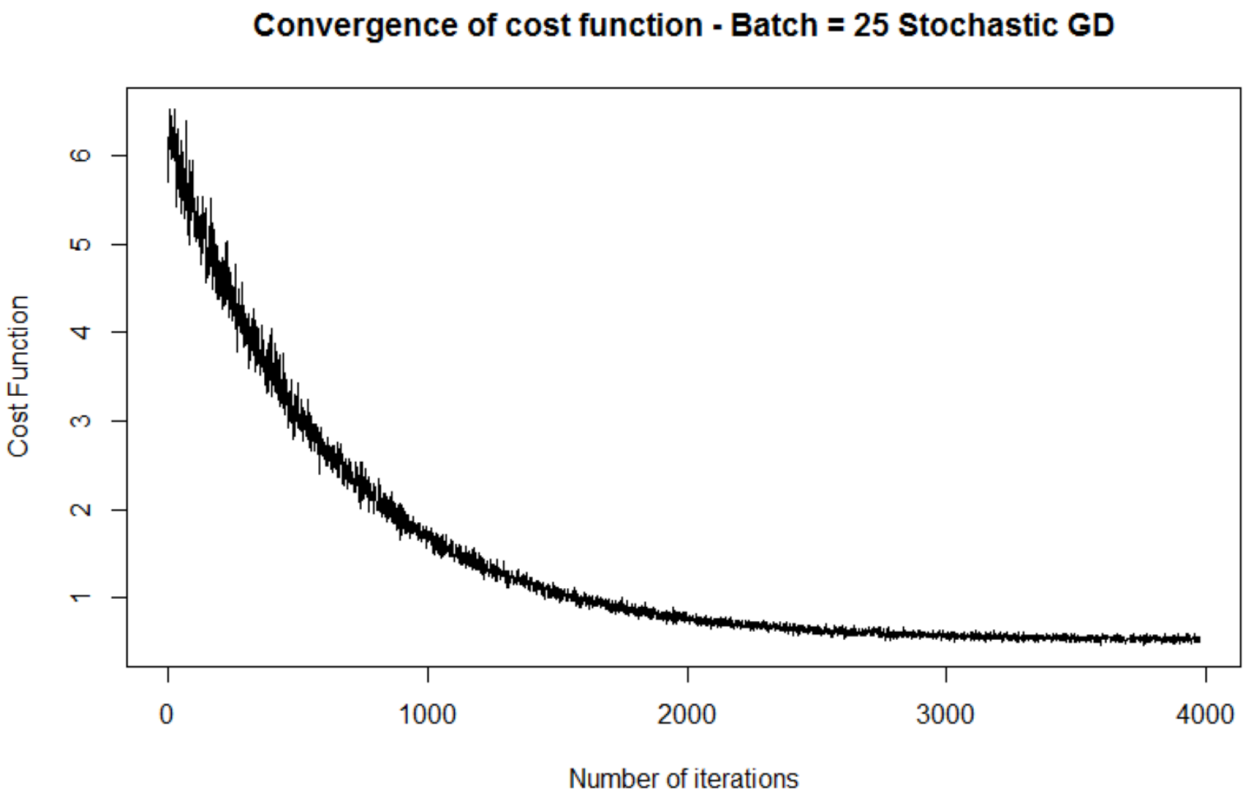


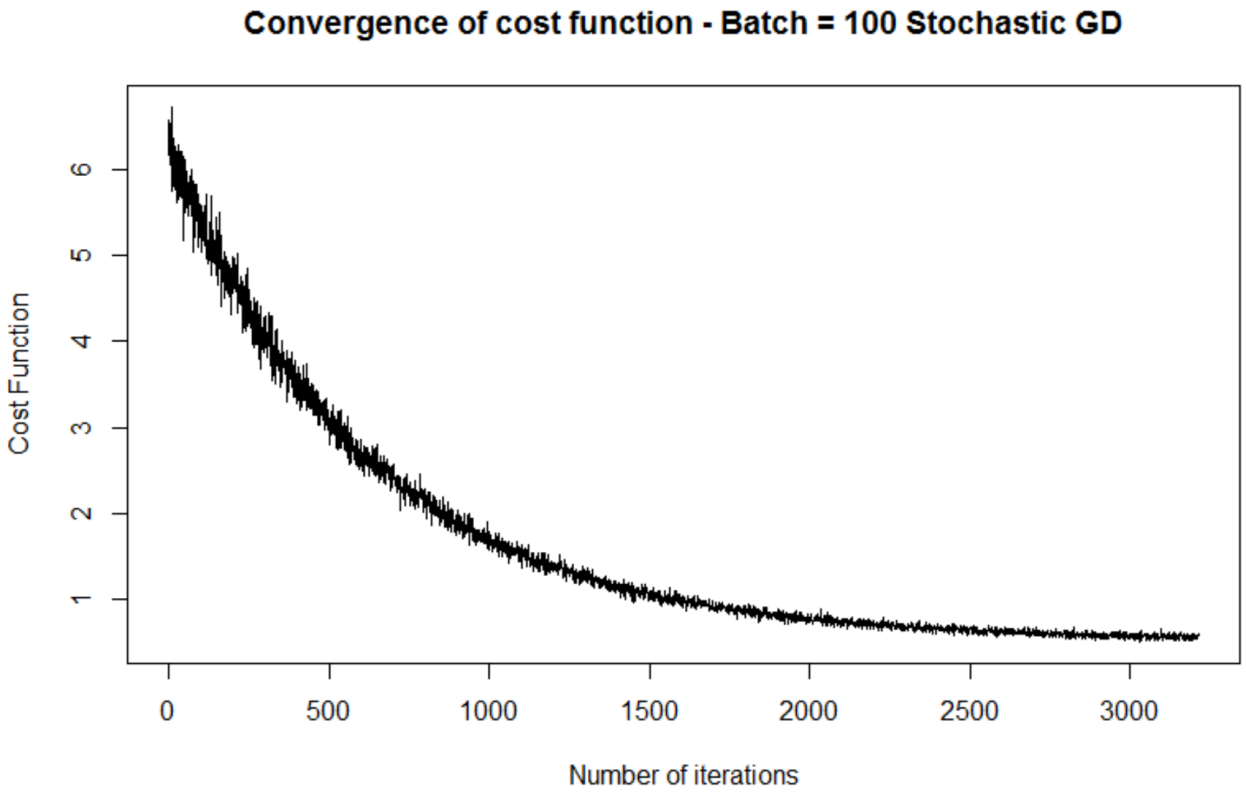

(B)

In mini-batch stochastic gradient descent, instead of using a single observation to calculate the gradient, we use a fixed number of observations randomly selected from the dataset to run each iteration. For this algorithm, the step size was multiplied by the size of the mini-batch, and the regression coefficients β* were updated based on the mean gradient evaluated for one mini-batch at each iteration.
The cost function was calculated as the average of the cost for all mini-batches within one pass. The stopping rule is the same as in the Gradient Descent Algorithm, with the modification that average cost function for each iteration has been used.

Plots of the cost function against the iterations are below. The size of the batches used were 10, 25 and 100:




Convergence of cost function - Batch = 25 Stochastic GD
Cost Function
1
2
3
4
5
6
0
1000
2000
3000
4000
Number of iterations


Convergence of cost function - Batch = 100 Stochastic GD
Cost Function
1
2
3
4
5
6
0
500
1000
1500
2000
2500
3000
Number of iterations

(C) The mean squared error comparing the true regression coefficients and stochastic regression coefficients for each of the runs described above:

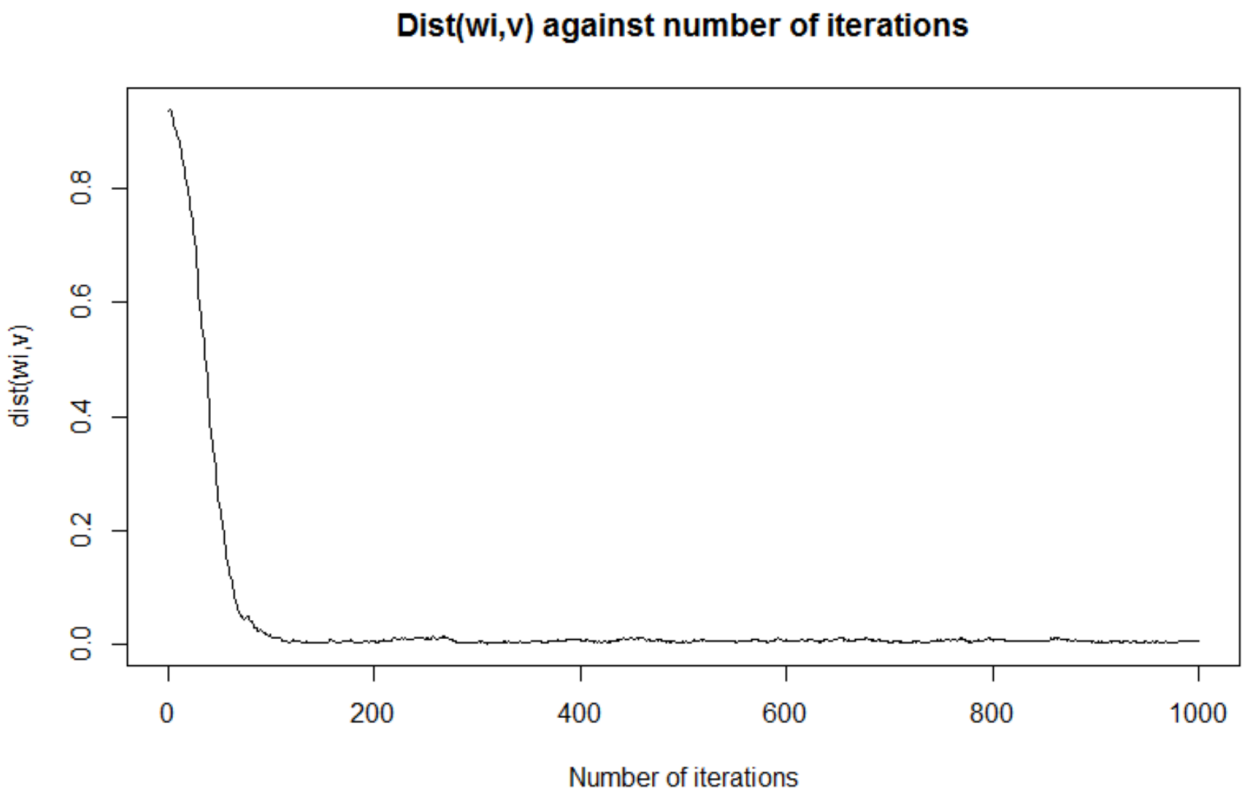
Stochastic Gradient Descent: 0.001121042
Mini-batch SGD (b = 10): 0.001348672
Mini-batch SGD (b = 25): 0.001370626
Mini-batch SGD (b = 100): 0.001295305

## Q.3. Online Principal Component Analysis

(A)

In the first implementation of Oja's algorithm, we use a fixed step size of η = 0.01 and calculate the square of the sine of the angle between W (the output at the end of every iteration) and V (the true eigenvector).

The plot of $dist(W_i, V)$ versus the number of iterations is as below:



(B) The second implementation comprises of changing the step size with every iteration.

$$\eta = \frac{1}{100 + iteration\ number}$$

The plot for the same is below: