CSE 250B: HW3

WEIWEI LI
PID: A53107958
DIS: WED 7PM

1. We have a data set containing 178 data points in 13 dimensions. Treat the first 128 points as a training set, and the rest are test set. We want to construct a method of coordinate descent. The basic ideas of my coordinate descent method are as follows:

Proposed coordinate descent: First, we preprocess the data set by shuffling it, normalize it and add offset points. Since we have 3 classes, so we have a 3 by 14 weight matirx W.We first initialize our weight matrix W by randomly selecting the numbers. Then we calculate the gradient matrix G of our loss function respect to W_{ij} . We find out the largest 10 gradients' coordinate, and update the 10 W_{ij} by the gradient descent method. Then we recalculate our new gradient matrix. We repeat this procedure for M times. We count the number of $L(W_m)$ being greater than $L(W_{m-1})$. If the count number is greater than 5, we halve our gradient descent step size, reset our count number and repeat the process. We stop our process until the loss difference percentage, which is $\frac{|L(W_m)-L(W_{m-1})|}{L(W_{m-1})}$,smaller than 10^{-4}.

In this method, the loss function need to be first-order differentiable in order to do the gradient descent update.

The pseudocode of the proposed algorithm is as follows:

Input: train data set X, train label Y
Output: loss and error
Proposed Coodinate Descent
loss=[ ], error=[ ]
alpha=A
Preprocess data: Shuffling, normalizing data and add offset points.

for m in range M:
  W=weight matrix
  G=gradient matrix
  for k in range $\{10\}$:
    pos = the position ij of 10 largest gradient in matrix G
    W = update the W_{ij} in the previous weight matrix by gradient descent method
    loss += new loss after updating 10 coordinates.

Date: November 20, 2016.

error += new error afrer updating 10 coordinates.
if $loss[m] > loss[m-1]$:
count=count+1
if count>5:
alpha=alpha/2
count=0
if $\frac{|loss[m]-loss[m-1]|}{loss[m-1]} < 10^{-4}$:
break

2. When the loss function is first order differentiable, the gradient descent is convergent to a local minimum. The propsed coordinate descent method converges to the optimal loss.

3. We initialize the weight matrix W by randomly choosing numbers from the normal distribution with mean 0, and standard deviation 1, and use the initial gradient step size $\alpha = 10^3$. Then we perform the proposed coordinate descent method for 20000 times. We found out that after 6784 times, the loss difference percentage is smaller than 10^{-4}, and we get the final loss is approximatly 0.0024758.

We also perform the random-feature coordinate descent method by choosing coordinates ij uniformly at random and then updates W_{ij} using our method. This method stops after 973 times, when the loss difference percentage is smaller than 10^{-4}, and the final loss is 0.072288.

We use L-BFGS solver to calculate the final loss, without regulation. The final loss is 5.0254×10^{-6}, and the final error rate is 0, which are plotted in the figures in red.

The experiment results are as shown in the following figure.

We can clearly see that the loss iteration of the proposed coordinate descent method asymptote to the final loss. The error rate iteration is oscillating in small regions, but overall it is also asymptote to the final loss.

4. There is scope for further improvement in my coordinate descent scheme. In the current method, we recalculate the gradient method after 10 updates of weights. We can improve it by update the gradient matrix after every update of weight, since the gradient matrix changes after every update of weight. We did the current way to save the time.

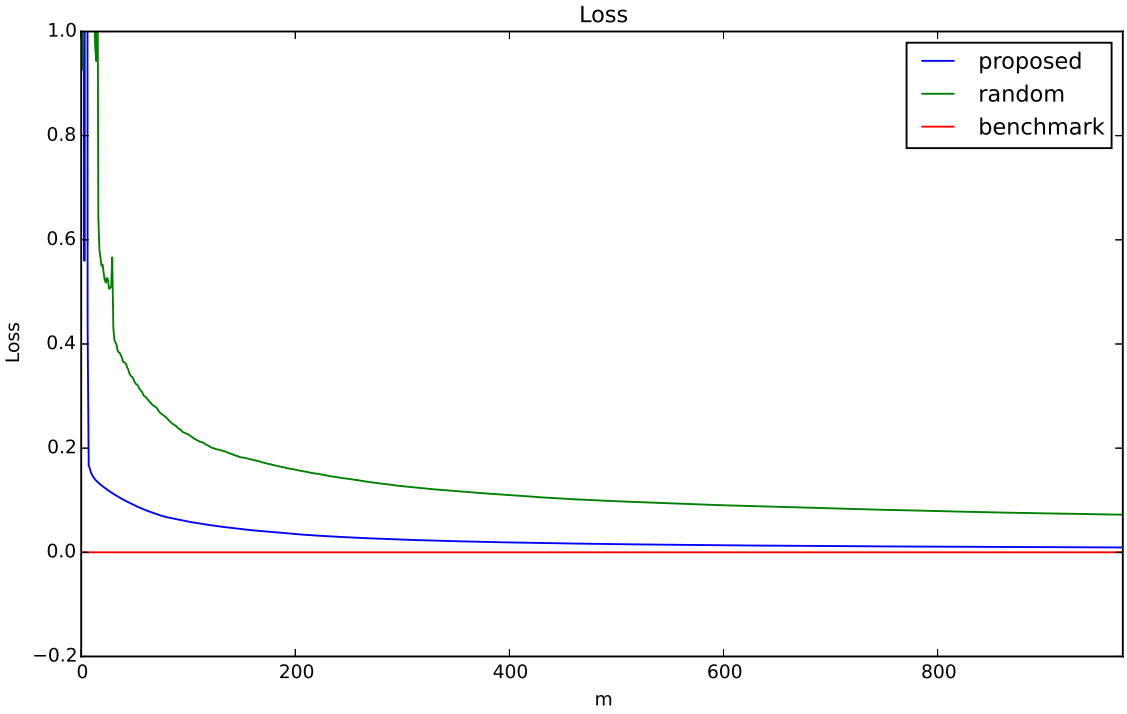

FIGURE 1. loss

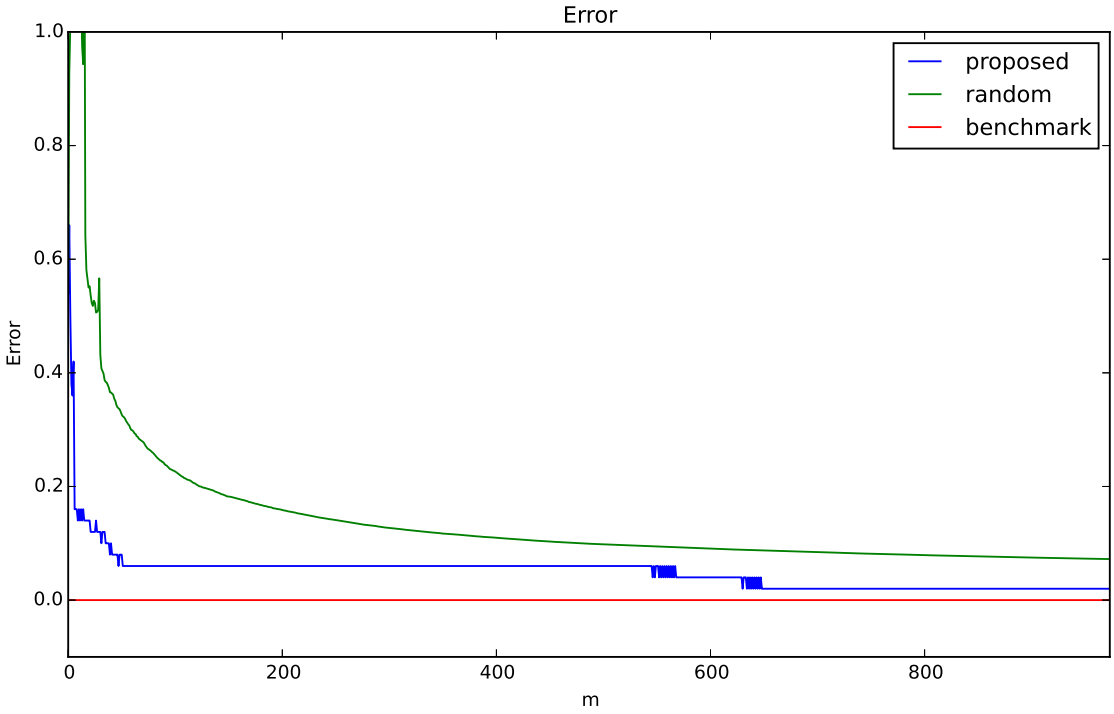

FIGURE 2. error