Homework Set 5, CPSC 8420, Fall 2023

Last Name, First Name

Due 12/10/2023, Friday, 11:59PM EST

Problem 1

Recall the classification models we discussed in class: **SVM** and **Logistic Regression**, seems both of them work on binary classification task. However, in real-world applications, multi-classification is everywhere, thus in this problem we explore how to extend vanilla **Logistic Regression** for multi-classification. Assume we have K different classes and the input $\mathbf{x} \in \mathcal{R}^d$, and the probability to each class is defined as:

$$P(Y=k|X=\mathbf{x}) = \frac{exp(\mathbf{w}_k^T\mathbf{x})}{1+\sum_{l=1}^{K-1} exp(\mathbf{w}_l^T\mathbf{x})} \quad for \quad k=1,2,\dots,K-1; P(Y=K|X=\mathbf{x}) = \frac{1}{1+\sum_{l=1}^{K-1} exp(\mathbf{w}_l^T\mathbf{x})} \tag{1}$$

If we define $\mathbf{w}_K = \mathbf{0}$, then we can combine the two cases above as one:

$$P(Y=k|X=\mathbf{x}) = \frac{exp(\mathbf{w}_k^T\mathbf{x})}{1+\sum_{l=1}^{K-1} exp(\mathbf{w}_l^T\mathbf{x})} \quad for \quad k=1,\dots,K \tag{2}$$

1. What and how many parameters are there to be optimized?

   $$\{\mathbf{w}_1, \mathbf{w}_2 ... \mathbf{w}_{k-1}\}$$

   since each $\mathbf{w}_i$ is a d dimension vector, so total $d \times (k-1)$

2. The training data is given as: $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, please simplify the log likelihood function to your best:

$$\begin{aligned} L(\mathbf{w}_1, \dots, \mathbf{w}_{K-1}) &= \sum_{i=1}^{n} lnP(Y=y_i|X=\mathbf{x}_i) & (3) \\ &= \sum_{i=1}^{n} ln \frac{exp(\mathbf{w}_{y_i}^T\mathbf{x}_i)}{1+\sum\limits_{l=1}^{K-1} exp(\mathbf{w}_l^T\mathbf{x}_i)} & (4) \\ &= \sum_{i=1}^{n} (\mathbf{w}_{y_i}^T\mathbf{x}_i - ln(1+\sum_{l=1}^{K-1} exp(\mathbf{w}_l^T\mathbf{x}_i))) & (5) \end{aligned}$$

3. Now please find the gradient of L w.r.t. $\mathbf{w}_k$.

$$\frac{\partial L}{\partial \mathbf{w}_k} = \sum_{i=1}^{n} (\mathbf{I}(y_i == k)\mathbf{x}_i - \frac{exp(\mathbf{w}_k^T \mathbf{x}_i)}{1 + \sum\limits_{l=1}^{K-1} exp(\mathbf{w}_l^T \mathbf{x}_i)} \mathbf{x}_i), y_i = k \tag{6}$$

4. If we add regularization term and formulate new objective function as:

$$f(\mathbf{w}_1, \ldots, \mathbf{w}_{K-1}) = L(\mathbf{w}_1, \ldots, \mathbf{w}_{K-1}) - \frac{\lambda}{2} \sum_{l=1}^{K-1} \|\mathbf{w}_l\|_2^2, \tag{7}$$

now please determine the new gradient.

$$\frac{\partial f}{\partial \mathbf{w}_k} = \sum_{i=1}^{n} (\mathbf{I}(y_i == k)\mathbf{x}_i - \frac{exp(\mathbf{w}_k^T \mathbf{x}_i)}{1 + \sum\limits_{l=1}^{K-1} exp(\mathbf{w}_l^T \mathbf{x}_i)} \mathbf{x}_i) - \lambda \mathbf{w}_k \tag{8}$$

$$\Longrightarrow \frac{\partial f}{\partial \mathbf{W}} = \mathbf{X}^T (\mathbf{I} - \frac{exp(\mathbf{XW})}{\Delta}) - \lambda \mathbf{W} \tag{9}$$

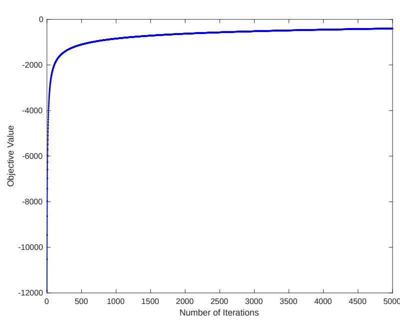
$$\text{where, } \Delta_i = |1 + \sum_j exp(\mathbf{XW})_{i,j},\ \ldots,\ 1 + \sum_j exp(\mathbf{XW})_{i,j}| \tag{10}$$

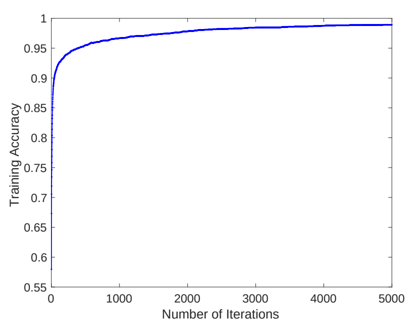
5. You are given *USPS* handwritten recognition digit dataset, with image size 16×16. For each digit (*i.e.* 0,1,...,9) there are 600 training samples in addition to 500 testing ones. You may use: imshow(reshape($\mathbf{x}$,16,16)) to view the image in Matlab. (Non-Matlab user may utilize .txt files to conduct experiments.)

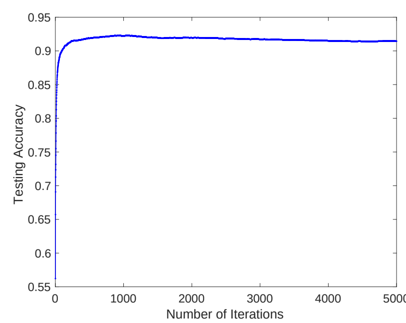
   (a) Please use gradient ascent algorithm (you are expected to complete log_grad.m) to train the model and plot 1) vanilla objective function L in Eq.(**??**); 2) training accuracy and 3) testing accuracy with updates respectively. Also indicate the final testing accuracy score. (Please choose a proper learning rate and stopping criterion). The folder include figures for your reference.

```
function G=log_grad(y, X, B)

% each sum opertaion can be represented as a multiplication of 2 matrix

    % X 6000 x 256
    % y 6000 x 1
    % B 256 x 9

    % G 256 X 9

    % exp(x0w0) exp(x0w1) .... exp(x0w9)
    % exp(x1w0) exp(x1w1) .... exp(x1w9)
    % exp(x2w0) exp(x2w1) .... exp(x2w9)
    % ....
```

```
        % exp(xnw0) exp(xnw1) .... exp(xnw9)
        Mat_EXP = exp(X*B); % 6000 x 9


        % 6000 *1
        % 1/ (\sum exp(x0wi)+1)
        % 1/ (\sum exp(x1wi)+1)
        % 1/ (\sum exp(x2wi)+1)
        % .....
        % 1/ (\sum exp(xnwi)+1)
        Delta = 1./(sum(Mat_EXP, 2)+1); % 6000 * 1


         %Classes = 9;
        Classes = size(B,2);
        Temp=Mat_EXP.*repmat(Delta, 1, Classes); % 6000 * 9

        I = zeros(size(X,1),Classes);

        for k= 1: Classes
            I(:,k) = (y == k);
        end

        Temp = I - Temp;

        G=(X'*Temp);
    end
```







(b) Now if we add the regularization term as Eq.(7), please show the final accuracy when $\lambda = \{0, 1, 10, 100, 200\}$ respectively.

$$\lambda = 0.1, acc = 0.914400$$
$$\lambda = 1, acc = 0.915800$$
$$\lambda = 10, acc = 0.919200$$
$$\lambda = 100, acc = 0.897400$$
$$\lambda = 200, acc = 0.881800$$

(c) What conclusion can we draw from the above experiments? penalty term can avoid overfitting on training data, but too large a penalty term will decrease the accuracy