CSE 251A: Machine learning Winter 2021

Programming project 2 — Coordinate Descent

Name: Xuanyu Chen

PID: A53306915

I. Description

Coordinate descent is a simple and useful approach to solve an unconstrained optimization problem. Coordinate descent algorithms solve optimization problems by successively performing approximate minimization along coordinate directions. If f is a k-dimensional function, we can minimize f by successively minimizing each of the individual dimensions of f while holding the values of f in the other dimensions fixed. The primary advantage of this approach is that it takes an arbitrarily complex k-dimensional problem and reduces it to a collection of k one-dimensional problems.

II. Pseudocode

Algorithm 1 Coordinate Descent

Input:

α: a positive constant, the step-size for every update in one coordinate

θ: threshold

N: maximum number of iterations

F: bool, if False means the loss value is not optimal, True means optimal

Output: vector w and optimal loss value $f(w^t)$

Procedure COORDINATEDESCENT

set $t \leftarrow 0$ and choose $w^0 \in \mathbb{R}^d$

$loss \leftarrow f(w^0)$

while $(F == False)$ & $(t < N)$:

    choose index $i_t \in \{1, 2, \dots, d\}$ which is $argmax_{i_t} abs([\nabla f(w^t)]_{i_t})$

    $I.append(i_t)$

    $w^{t+1} \leftarrow w^t - \alpha[\nabla f(w^t)]_{i_t}$;

    $newloss \leftarrow f(w^{t+1})$

    while $(loss - newloss) < \theta$:

        choose index $i_t \in \{1, 2, \dots, d\}$ & $i_t \notin I$ which is $argmax_{i_t}\text{abs}([\nabla f(w^t)]_{i_t})$

        $I.append(i_t)$

        $w^{t+1} \leftarrow w^t - \alpha_t[\nabla f(w^t)]_{i_t}$;

        $newloss \leftarrow f(w^{t+1})$

        if $len(I) == len(w)$:

            $F \leftarrow True$

            break

$loss \leftarrow newloss$

$t \leftarrow t + 1$;

III. Experimental results

The optimal log-loss calculated based on the LogisticRegression function (with parameter C = 50) from scikit-learn package is 0.02102.

Here are the results calculated by my Coordinate Descent algorithm:

Table 1.

Max-Iteration	500	1500	5000	7000	9000
alpha	0.1	0.1	0.1	0.1	0.1
threshold	1e-3	1e-4	1e-5	1e-6	1e-6
Log-loss	0.475	0.158	0.0589	0.048	0.0386
iterations	403	1500	4622	7000	9000

Figure 1. (First 100 iterations)

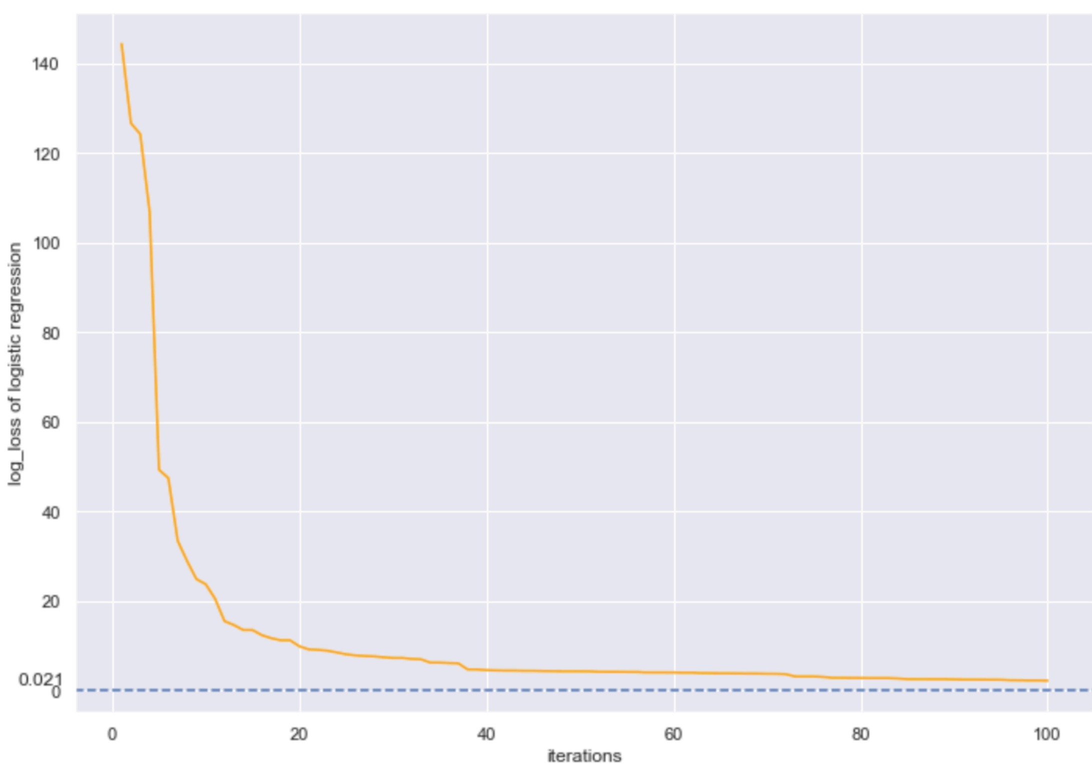
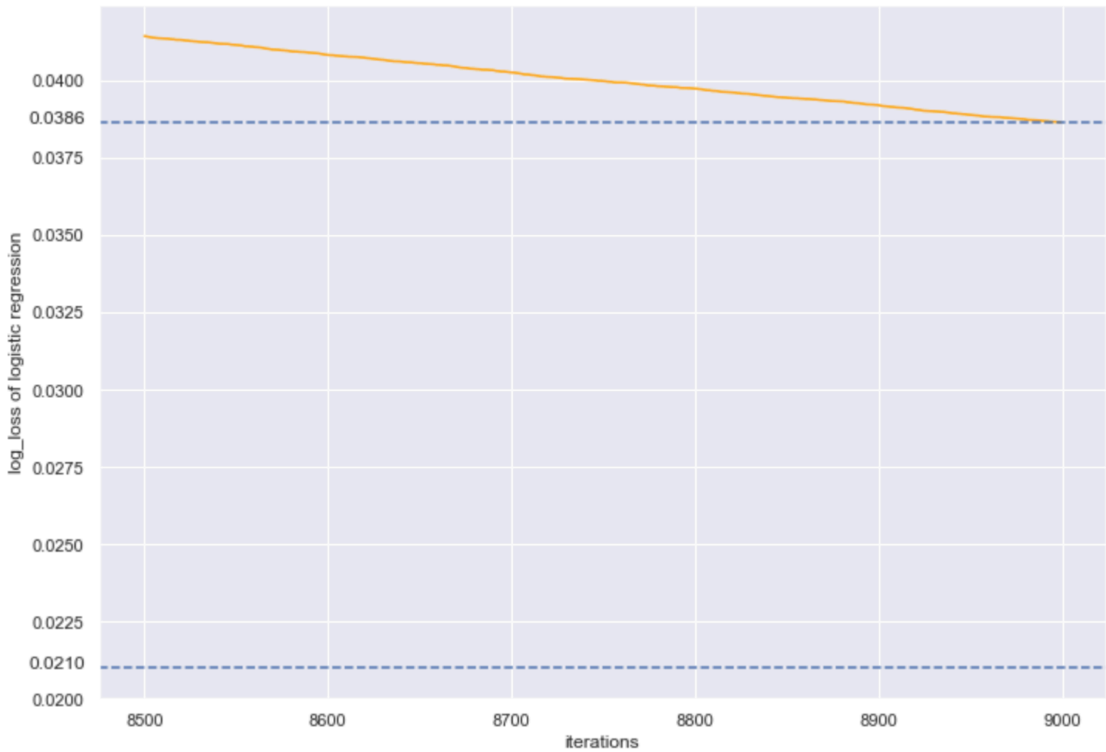


Figure 2. (Last 500 iterations)



IV. Critical evaluation

(1) In my Coordinate Descent algorithm, I choose the index of coordinate which maximizes the first derivative of function f. I also tried to use the method of choosing the index of coordinate randomly. It turns out that the difference between these two methods is not such big like I imagined before. In the future, I may try to use other smarter method to select the coordinate index so that $L(w^t)$ can approach to L^* more quickly and better.

(2) Through my experiment, the "initial starting point" and the original data influence my algorithm a lot. If I use a vector containing all zeros as my initial weight, then it turns out that my final outcome is pretty bad. Moreover, I use MinMaxScaler() function to standardize the original data so that $L(w^t)$ can approach to L^* better.

(3) For every coordinate-dimension update, my step size is constant. Intuitively, the update step size will decrease as iteration goes on. I have read a paper about *Adam Optimization Algorithm* before and it inspired me that my future improvements could include updating step size in real time.