CS257 Linear and Convex Optimization Homework 6

Xiangyuan Xue (521030910387)

According to gradient descent algorithm with const step size, the function is as follow:

```
def gd_const_ss(fp, x0, stepsize, tol=1e-5, maxiter=100000):
    x_traces = [np.array(x0)]
    x = np.array(x0)
    for _ in range(maxiter):
        grad = fp(x)
        if np.linalg.norm(grad) < tol:
            break
        x = x - stepsize * grad
        x_traces.append(np.array(x))
    return x_traces
```

1. (a) $L_{\min} = \lambda_{\max}(\boldsymbol{Q}) = \max\{1, \lambda\}$

   (b) Outputs are as follow:

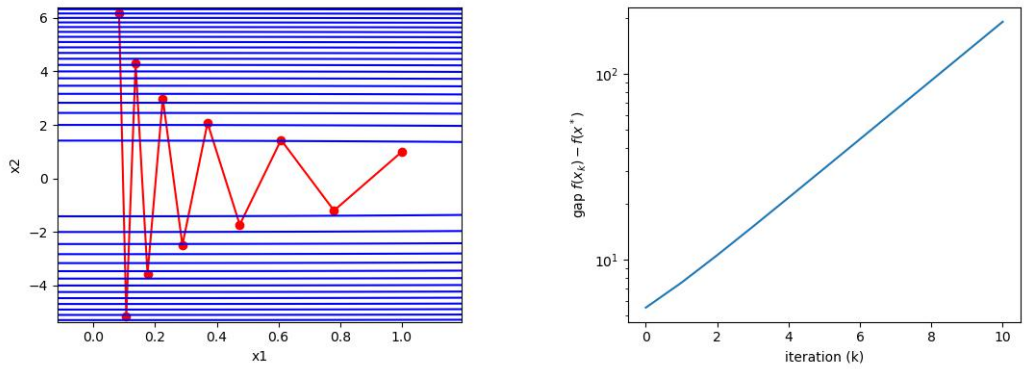
```
gamma = 10.000
    stepsize = 0.220
        iterations: 100000
        state: diverge
    stepsize = 0.100
        iterations: 110
        state: converge
    stepsize = 0.010
        iterations: 1146
        state: converge
    stepsize = 0.001
        iterations: 11508
        state: converge
```

   Therefore, we describe the results in the following table:

gamma	stepsize	state	iterations
10	0.22	diverge	100000
10	0.1	converge	110
10	0.01	converge	1146
10	0.001	converge	11508

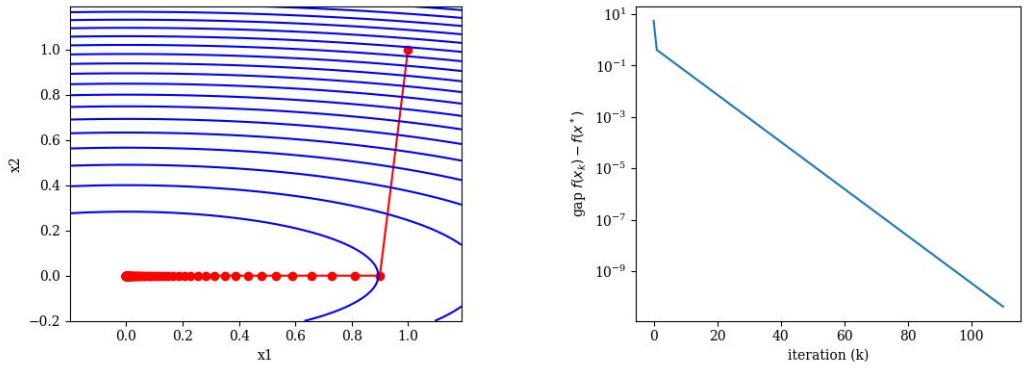
Table 1: Results for $\gamma = 10$

Moreover, 2D trajectory of the sequence $\boldsymbol{x}_k$ and the function values $f(\boldsymbol{x}_k)$ are shown in the following figures:



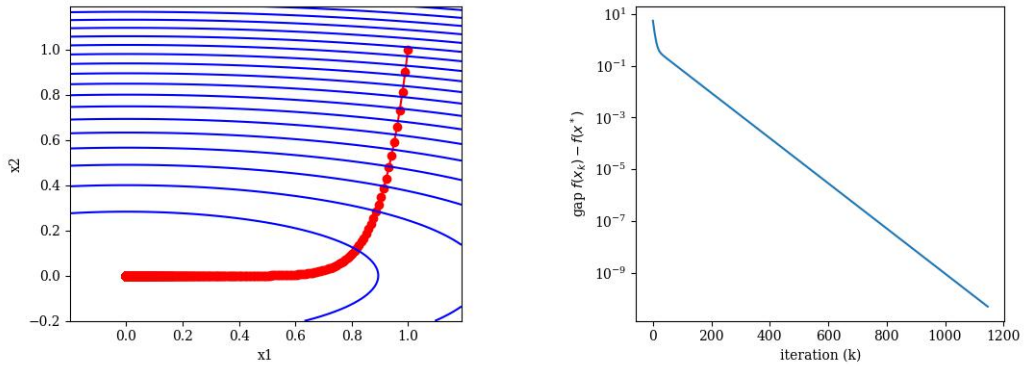
(a) trajectory (b) error

Figure 1: trajectory and error for stepsize = 0.22



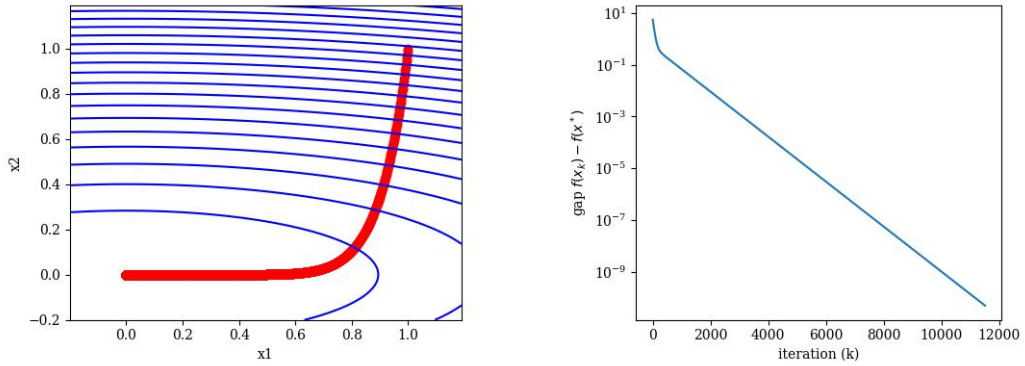
(a) trajectory (b) error

Figure 2: trajectory and error for stepsize = 0.1



(a) trajectory (b) error

Figure 3: trajectory and error for stepsize = 0.01



(a) trajectory (b) error

Figure 4: trajectory and error for stepsize = 0.001

(c) Outputs are as follow:

```
stepsize = 1.000
    gamma = 1.000
        iterations: 1
        state: converge
    gamma = 0.100
        iterations: 88
        state: converge
    gamma = 0.010
        iterations: 688
        state: converge
    gamma = 0.001
        iterations: 4603
        state: converge
```

Therefore, we describe the results in the following table:

gamma	stepsize	state	iterations
1	1	converge	1
0.1	1	converge	88
0.01	1	converge	688
0.001	1	converge	4603

Table 2: Results for stepsize = 1

We can see that as γ decreases, the number of iterations increases.

2. The object function can be written as:

$$f(\boldsymbol{w}) = (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T\boldsymbol{y}$$

Thus its gradient can be written as:

$$\nabla f(\boldsymbol{w}) = 2\left(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^T\boldsymbol{X}\right)$$

Using mathematical software we know that:

$$\lambda_{\max}(\boldsymbol{X}^T\boldsymbol{X}) \approx 185.27$$

So we claim that $f(\boldsymbol{w})$ is 200-smooth and let stepsize $= \frac{1}{200}$. Outputs are as follow:

```
method: gradient descent
    iterations: 1275
    state: converge
    solution: [ 1.22170661 -0.21469307  0.15549204 -0.4586777
   1.18537706  0.00613317]
    value: 13.295569
```

In Homework 5, the solution is as follow:

```
Least square solution: [ 1.22170662 -0.21469307  0.15549204 -0.4586777
      1.18537706  0.00613317]
Least square error: 13.295569218196665
```

By solving the normal equation, the solution is as follow:

```
method: normal equation
    solution: [ 1.22170662 -0.21469307  0.15549204 -0.4586777
   1.18537706  0.00613317]
    value: 13.295569
```

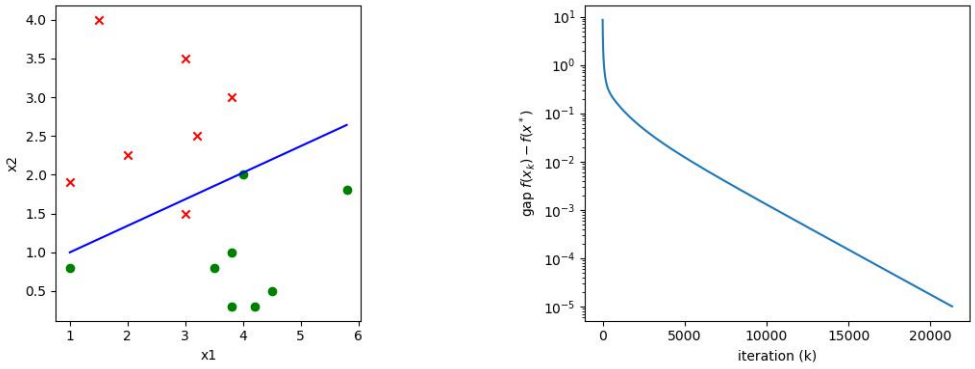
These solutions are actually the same.

3. Let stepsize = 0.01 and initial $\boldsymbol{w}_0 = (0, 0, 0)^T$. Outputs are as follow:

```
iterations: 21322
state: converge
solution: [-1.73186226  5.05432727 -3.31093323]
accuracy: 93.33%
```

Therefore, the optimal $\boldsymbol{w}^* \approx (-1.73, 5.05, -3.31)^T$ and the accuracy is about 93.33%. The missing point is unavoidable because the dataset is not linearly separable indeed.

Visualization is shown in the following figure:



(a) visualization (b) gradient descent

Figure 5: logistic regression on toy dataset