

Machine Learning Homework 6

Hairui Yin

1.

(a) Given the formula $f(x)$

$$f(x) = \frac{1}{2}x^T Px + q^T x + \log(e^{-2x_1} + e^{-x_2})$$

The gradient is computed as follows:

$$\begin{cases} \nabla(\frac{1}{2}x^T Px) = Px \\ \nabla(q^T x) = q \\ \nabla(\log(e^{-2x_1} + e^{-x_2})) = \frac{1}{e^{-2x_1} + e^{-x_2}} \begin{bmatrix} -2e^{-2x_1} \\ -e^{-x_2} \end{bmatrix} \end{cases}$$

Thus, the gradient $\nabla f(x)$ is

$$\nabla f(x) = Px + q + \frac{1}{e^{-2x_1} + e^{-x_2}} \begin{bmatrix} -2e^{-2x_1} \\ -e^{-x_2} \end{bmatrix}$$

The code is as follows:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.optimize import minimize_scalar
4
5 def f(x, P, q):
6     return 0.5 * x.T @ p @ x + q.T @ x + \
7         np.log(np.exp(-2 * x[0]) + np.exp(-x[1]))
8 def grad_f(x, P, q):
9     grad_quadratic = P @ x + q
10    exp_terms = np.array([np.exp(-2 * x[0]), np.exp(-x[1])])
11    grad_logarithmic = -np.array([2 * exp_terms[0], exp_terms[1]]) / \
12        np.sum(exp_terms)
13    return grad_quadratic + grad_logarithmic
```

where we define the function of $f(x)$ and its gradient. Then implement gradient descent with exact line search, using the `minimize_scalar()` function to find exactly the minimum of $\phi(\alpha)$,

```

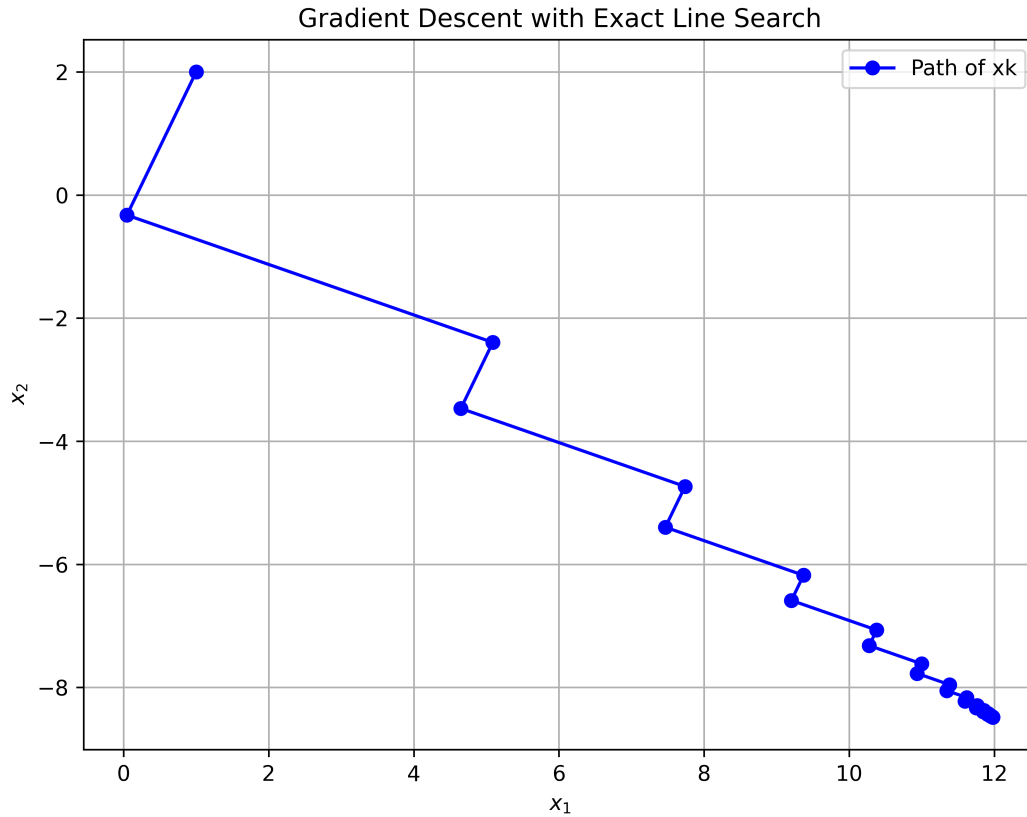
1 def gradient_descent_exact_line_search(x0, P, q, tol=1e-2, max_iter=100):
2     xk = x0
3     path = [xk] # To store the sequence of solutions
4
5     for _ in range(max_iter):
6         grad = grad_f(xk, P, q)
7         if np.linalg.norm(grad) < tol:
8             break
9
10        # Exact line search: minimize f(xk - alpha * grad)
11        # Objective function for line search along direction -grad
12        def phi(alpha):
13            return f(xk - alpha * grad, P, q)
14
15        # Approximate exact line search (find alpha that minimizes phi)
16        alpha = minimize_scalar(phi).x
17
18        # Update xk
19        xk = xk - alpha * grad
20        path.append(xk)
21
22    return np.array(path)

```

Then input parameter P, q and initial point x_0 , we finally have result under stopping condition $\|\nabla f(x)\| \leq 10^{-2}$ is

$$\begin{cases} x_1 = 11.97772588 \\ x_2 = -8.48475201 \\ y^* = -24.74991683 \end{cases}$$

And the sequence of solutions x^k are plotted as follows:



(b) In question (b), we update α by $\alpha^* = \beta$ if satisfying the condition that $f(x_k + \alpha \cdot d_k) \leq f(x_k) + \alpha \cdot \gamma(\nabla^T \nabla)$. The implement gradient descent with backtracking line search is

```

1 def gradient_descent_backtracking(x0, P, q, \
2     alpha_init=0.15, gamma=0.7, beta=0.8, tol=1e-6, max_iter=100):
3     xk = x0
4     path = [xk] # To store the sequence of solutions
5
6     for _ in range(max_iter):
7         grad = grad_f(xk, P, q)
8         if np.linalg.norm(grad) < tol:
9             break
10
11         # Backtracking line search
12         alpha = alpha_init
13         while f(xk - alpha * grad, P, q) > \
14             f(xk, P, q) - gamma * alpha * np.dot(grad.T, grad):
15             alpha *= beta
16
17         # Update xk

```

```

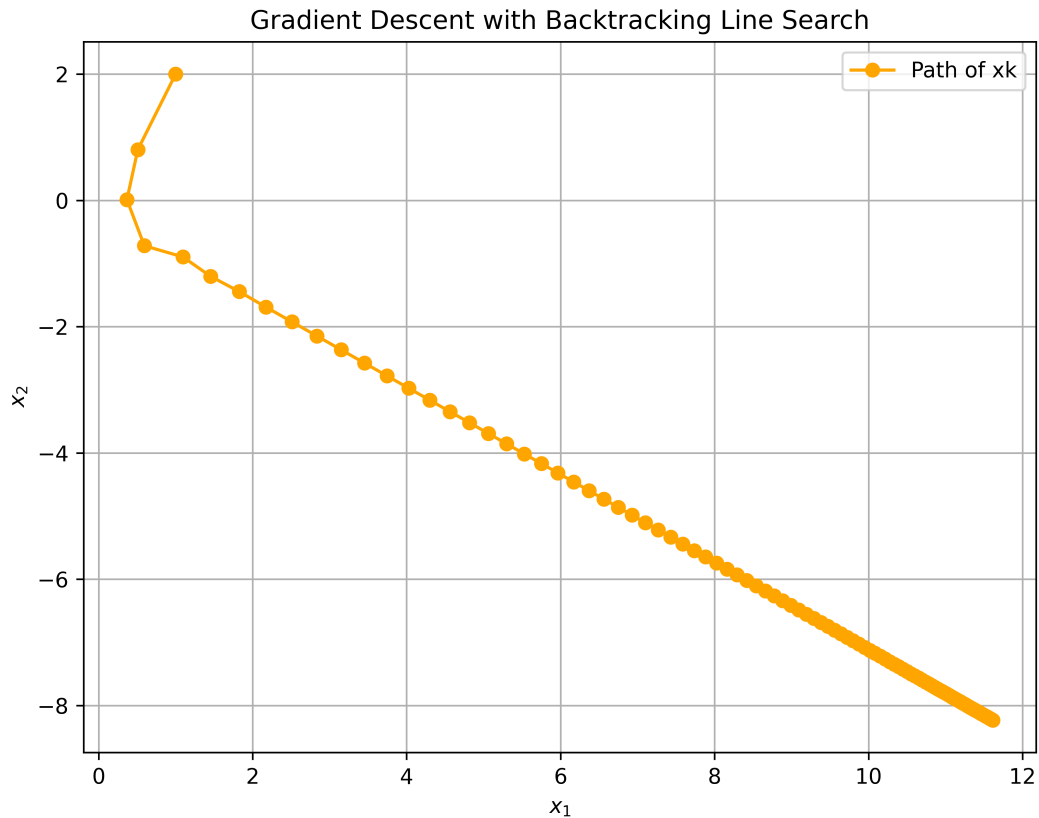
18         xk = xk - alpha * grad
19         path.append(xk)
20
21     return np.array(path)

```

Then input parameter P, q and initial point x_0 , we finally have result under stopping condition $\|\nabla f(x)\| \leq 10^{-2}$ is

$$\begin{cases} x_1 = 11.61378187 \\ x_2 = -8.23235066 \\ y^* = -24.74979329 \end{cases}$$

And the sequence of solutions x^k are plotted as follows:



2.

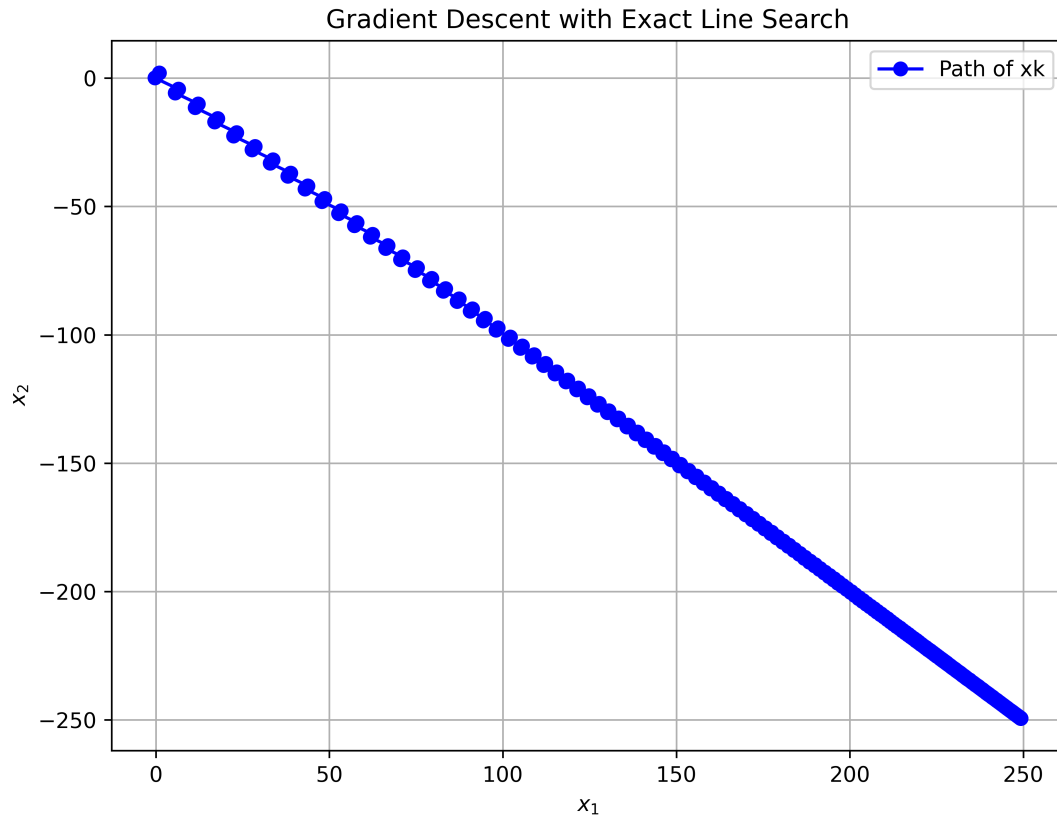
(a) Changing the matrix P to

$$\begin{bmatrix} 5.005 & 4.995 \\ 4.995 & 5.005 \end{bmatrix}$$

And following the same step in question 1 in exact line search, input parameter P, q and initial point x_0 , we finally have result under stopping condition $\|\nabla f(x)\| \leq 10^{-2}$ is

$$\begin{cases} x_1 = 249.2628884 \\ x_2 = -249.36318387 \\ y^* = -625.02028059 \end{cases}$$

And the sequence of solutions x^k are plotted as follows:



(a) Changing the matrix P to

$$\begin{bmatrix} 5.005 & 4.995 \\ 4.995 & 5.005 \end{bmatrix}$$

And following the same step in question 1 in exact line search, input parameter P, q and initial point x_0 , we finally have result under stopping condition $\|\nabla f(x)\| \leq 10^{-2}$ is

$$\begin{cases} x_1 = 249.24321348 \\ x_2 = -249.34321348 \\ y^* = -625.02000453 \end{cases}$$

And the sequence of solutions x^k are plotted as follows:

