Convex Optimization Homework

1. Convex Set

(1) Polyhedron Convexity

Problem: Show that a polyhedron $\{x \in \mathbb{R}^n : Ax \leq b\}$ for some $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, is convex.

证明. Let $S = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Let $x, y \in S$ and $\theta \in [0, 1]$. Since $x, y \in S$, we have $Ax \leq b$ and $Ay \leq b$. Consider $z = \theta x + (1 - \theta)y$.

$$Az = A(\theta x + (1 - \theta)y) = \theta Ax + (1 - \theta)Ay$$

Since $\theta \geq 0$ and $1 - \theta \geq 0$, we can combine the inequalities:

$$\theta Ax + (1 - \theta)Ay \leq \theta b + (1 - \theta)b = b$$

Thus $Az \leq b$, which implies $z \in S$. Therefore, the polyhedron is a convex set. □

(2) Epigraph Convexity

Problem: Consider a convex function $f : \mathbb{R}^n \to \mathbb{R}$. Prove that the set (Epigraph) $\{(x, t) \mid f(x) \leq t, x \in \mathbb{R}^n, t \in \mathbb{R}\}$ is convex.

证明. Let $K = \text{epi}(f) = \{(x, t) \mid f(x) \leq t\}$. Let $(x_1, t_1), (x_2, t_2) \in K$ and $\theta \in [0, 1]$. By definition, $f(x_1) \leq t_1$ and $f(x_2) \leq t_2$. By the convexity of f:

$$f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$$

Substituting the inequalities for t:

$$\theta f(x_1) + (1 - \theta)f(x_2) \leq \theta t_1 + (1 - \theta)t_2$$

Let $\bar{x} = \theta x_1 + (1 - \theta)x_2$ and $\bar{t} = \theta t_1 + (1 - \theta)t_2$. The above shows $f(\bar{x}) \leq \bar{t}$. Thus, $(\bar{x}, \bar{t}) \in K$, proving the set is convex. □

(3) Product Constraint

Problem: Show that $\{x \in \mathbb{R}^n_{++} \mid \prod_{i=1}^n x_i \geq 1\}$ is convex. (Hint: If $a, b \geq 0$ and $0 \leq \theta \leq 1$ then $a^\theta b^{(1-\theta)} \leq \theta a + (1-\theta)b$).

证明. The condition $\prod_{i=1}^n x_i \geq 1$ for $x \in \mathbb{R}^n_{++}$ is equivalent to taking the logarithm on both sides:

$$\sum_{i=1}^n \ln(x_i) \geq 0 \iff -\sum_{i=1}^n \ln(x_i) \leq 0$$

Define $g(x) = -\sum_{i=1}^n \ln(x_i)$. Since $h(u) = -\ln(u)$ is a convex function, $g(x)$ is a sum of convex functions and is therefore convex. The set can be written as the 0-sublevel set of the convex function $g(x)$, i.e., $\{x \mid g(x) \leq 0\}$. Sublevel sets of convex functions are convex. □

(4) Distance Ratio

Problem: Show that the set $\{x \mid \|x - a\|_2 \leq \theta \|x - b\|_2\}$ where $a \neq b$ and $0 \leq \theta \leq 1$, is convex.

证明. Square both sides of the inequality (valid since norms are non-negative):

$$\|x - a\|_2^2 \leq \theta^2 \|x - b\|_2^2$$

Expand the squared norms:

$$(x - a)^T (x - a) \leq \theta^2 (x - b)^T (x - b)$$

$$x^T x - 2a^T x + a^T a \leq \theta^2 (x^T x - 2b^T x + b^T b)$$

Rearrange terms to one side:

$$(1 - \theta^2) x^T x - 2(a - \theta^2 b)^T x + (\|a\|^2 - \theta^2 \|b\|^2) \leq 0$$

Since $0 \leq \theta \leq 1$, we have $(1 - \theta^2) \geq 0$. The function $f(x) = (1 - \theta^2)\|x\|^2 + p^T x + q$ is a convex quadratic function (its Hessian is $2(1 - \theta^2) I \succeq 0$). The set is the sublevel set of this convex quadratic function, hence it is convex. □

2. Convex Function

(1) Entropy Function

Problem: Prove that the entropy function $f(x) = -\sum_{i=1}^n x_i \log(x_i)$ with $\mathrm{dom}(f) = \{x \in \mathbb{R}^n_{++} : \sum_{i=1}^n x_i = 1\}$, is strictly concave.

证明. Consider the scalar function $h(t) = -t \log t$ for $t > 0$.

$$h'(t) = -(\log t + 1), \quad h''(t) = -\frac{1}{t}$$

Since $t > 0$, $h''(t) < 0$, so $h(t)$ is strictly concave. The function $f(x) = \sum_{i=1}^n h(x_i)$ is a sum of strictly concave functions, so $f(x)$ is strictly concave on $\mathbb{R}^n_{++}$. The domain constraint $\sum x_i = 1$ is a convex set (affine). Restricting a strictly concave function to a convex set preserves strict concavity. □

(2) Inverse Product

Problem: Show that $f(x_1, x_2) = \frac{1}{x_1 x_2}$ on $\mathbb{R}^2_{++}$ is convex.

证明. Calculate the Hessian matrix $\nabla^2 f(x)$. Gradient: $\nabla f = \begin{bmatrix} -1/(x_1^2 x_2) \\ -1/(x_1 x_2^2) \end{bmatrix}$. Hessian:

$$\nabla^2 f = \begin{bmatrix} \frac{2}{x_1^3 x_2} & \frac{1}{x_1^2 x_2^2} \\ \frac{1}{x_1^2 x_2^2} & \frac{2}{x_1 x_2^3} \end{bmatrix}$$

Check for positive definiteness for $x_1, x_2 > 0$: 1. Trace: $\frac{2}{x_1^3 x_2} + \frac{2}{x_1 x_2^3} > 0$. 2. Determinant:

$$\det(\nabla^2 f) = \frac{4}{x_1^4 x_2^4} - \frac{1}{x_1^4 x_2^4} = \frac{3}{x_1^4 x_2^4} > 0$$

Since the Hessian is positive definite on the domain, f is convex. □

(3) Matrix Inverse Trace

Problem: Show that $f(X) = \mathrm{tr}(X^{-1})$ is convex on $\mathrm{dom}\, f = \mathbb{S}^n_{++}$.

证明. Let $X \in \mathbb{S}^n_{++}$ and $V \in \mathbb{S}^n$. Consider the function restricted to a line $g(t) = f(X + tV)$.

$$g(t) = \mathrm{tr}((X + tV)^{-1})$$

First derivative:

$$\frac{d}{dt}(X + tV)^{-1} = -(X + tV)^{-1} V (X + tV)^{-1}$$
$$g'(t) = -\mathrm{tr}((X + tV)^{-1} V (X + tV)^{-1}) = -\mathrm{tr}((X + tV)^{-2} V)$$

Second derivative at $t = 0$:

$$g''(0) = 2\mathrm{tr}(X^{-1} V X^{-1} V X^{-1}) = 2\mathrm{tr}((X^{-1/2} V X^{-1/2})^2 X^{-1})$$

Since X^{-1} is positive definite and $(X^{-1/2} V X^{-1/2})$ is symmetric, the trace form is essentially weighted squared norm, which is non-negative. Specifically, let $Y = X^{-1/2} V X^{-1/2}$. Then $g''(0) = 2\mathrm{tr}(Y^2 X^{-1})$. This is strictly positive for $V \neq 0$. Thus f is convex. □

3. Dual Problem

(1) Single Inequality Constraint

Problem: Formulate the dual problem of $\min c^T x$ s.t. $f(x) \leq 0$.

Solution: Lagrangian: $L(x, \lambda) = c^T x + \lambda f(x)$ with $\lambda \geq 0$. Lagrange dual function: $g(\lambda) = \inf_x (c^T x + \lambda f(x))$. The dual problem is:

$$\max_{\lambda \geq 0} \quad g(\lambda)$$

(2) General Linear Programming

Problem: Find the dual problem of the LP: $\min c^T x$ s.t. $Gx \leq h, Ax = b$.

Solution: Lagrangian:

$$L(x, \lambda, \nu) = c^T x + \lambda^T (Gx - h) + \nu^T (Ax - b) = (c + G^T \lambda + A^T \nu)^T x - h^T \lambda - b^T \nu$$

Dual function $g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$. This infimum is bounded ($-\infty$) unless the coefficient of x is zero.

$$g(\lambda, \nu) = \begin{cases} -h^T \lambda - b^T \nu & \text{if } G^T \lambda + A^T \nu + c = 0 \\ -\infty & \text{otherwise} \end{cases}$$

Dual Problem:

$$\begin{aligned} \max \quad & -h^T \lambda - b^T \nu \\ \text{s.t.} \quad & G^T \lambda + A^T \nu + c = 0 \\ & \lambda \geq 0 \end{aligned}$$

4. KKT Condition

(1) Two Circle Constraints

Problem: Give the KKT conditions for $\min x_1^2 + x_2^2$ s.t. $(x_1 - 1)^2 + (x_2 - 1)^2 \leq 1$ and $(x_1 - 1)^2 + (x_2 + 1)^2 \leq 1$.

Solution: Let $f_0(x) = x_1^2 + x_2^2$, $g_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 - 1$, $g_2(x) = (x_1 - 1)^2 + (x_2 + 1)^2 - 1$. KKT Conditions:

1. **Stationarity:** $\nabla f_0(x) + \lambda_1 \nabla g_1(x) + \lambda_2 \nabla g_2(x) = 0$

$$\begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} + \lambda_1 \begin{pmatrix} 2(x_1 - 1) \\ 2(x_2 - 1) \end{pmatrix} + \lambda_2 \begin{pmatrix} 2(x_1 - 1) \\ 2(x_2 + 1) \end{pmatrix} = 0$$

2. **Primal Feasibility:** $(x_1 - 1)^2 + (x_2 - 1)^2 \leq 1, \quad (x_1 - 1)^2 + (x_2 + 1)^2 \leq 1$

3. **Dual Feasibility:** $\lambda_1 \geq 0, \quad \lambda_2 \geq 0$

4. **Complementary Slackness:** $\lambda_1 g_1(x) = 0, \quad \lambda_2 g_2(x) = 0$

(2) Equality Constrained Least Squares

Problem: Consider $\min \frac{1}{2}\|Ax-b\|_2^2$ s.t. $Gx = h$. Give KKT conditions and derive expressions for primal solution x^* and dual solution ν^*.

Solution: The Lagrangian is $L(x,\nu) = \frac{1}{2}(Ax-b)^T(Ax-b) + \nu^T(Gx-h)$.

1. KKT Conditions:

- **Stationarity:** $\nabla_x L = A^T(Ax-b) + G^T\nu = 0 \implies A^TAx + G^T\nu = A^Tb$.
- **Primal Feasibility:** $Gx = h$.

Writing this as a block matrix system (KKT System):

$$\begin{bmatrix} A^TA & G^T \\ G & 0 \end{bmatrix} \begin{pmatrix} x^* \\ \nu^* \end{pmatrix} = \begin{pmatrix} A^Tb \\ h \end{pmatrix}$$

2. Derivation of Explicit Solutions: Since $\text{rank}(A) = n$, A^TA is invertible. From the stationarity condition:

$$x^* = (A^TA)^{-1}(A^Tb - G^T\nu^*) = (A^TA)^{-1}A^Tb - (A^TA)^{-1}G^T\nu^*$$

Substitute this into the feasibility condition $Gx^* = h$:

$$G\left[(A^TA)^{-1}A^Tb - (A^TA)^{-1}G^T\nu^*\right] = h$$

$$G(A^TA)^{-1}A^Tb - G(A^TA)^{-1}G^T\nu^* = h$$

Solve for the dual variable ν^* (assuming G has full row rank, $G(A^TA)^{-1}G^T$ is invertible):

$$\nu^* = \left(G(A^TA)^{-1}G^T\right)^{-1}\left(G(A^TA)^{-1}A^Tb - h\right)$$

Finally, substitute ν^* back to obtain the primal solution x^*:

$$x^* = (A^TA)^{-1}A^Tb - (A^TA)^{-1}G^T\left(G(A^TA)^{-1}G^T\right)^{-1}\left(G(A^TA)^{-1}A^Tb - h\right)$$

Note: The first term $(A^TA)^{-1}A^Tb$ is the unconstrained least squares solution, and the second term corrects it to satisfy $Gx = h$.

5. Gradient and Newton Descent

Problem: Minimize $f(x) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$. Use Gradient and Newton method ($\alpha = 0.1, \beta = 0.6$). Draw convergence curve.

Solution: The Python code below implements the Gradient Descent and Newton's Method.

Listing 1: Optimization Code

```
import numpy as np
import matplotlib.pyplot as plt

```

```python
def f(x):
    return (np.exp(x[0] + 3*x[1] - 0.1) +
            np.exp(x[0] - 3*x[1] - 0.1) +
            np.exp(-x[0] - 0.1))

def grad_f(x):
    t1 = np.exp(x[0] + 3*x[1] - 0.1)
    t2 = np.exp(x[0] - 3*x[1] - 0.1)
    t3 = np.exp(-x[0] - 0.1)
    g1 = t1 + t2 - t3
    g2 = 3*t1 - 3*t2
    return np.array([g1, g2])

def hess_f(x):
    t1 = np.exp(x[0] + 3*x[1] - 0.1)
    t2 = np.exp(x[0] - 3*x[1] - 0.1)
    t3 = np.exp(-x[0] - 0.1)
    h11 = t1 + t2 + t3
    h12 = 3*t1 - 3*t2
    h22 = 9*t1 + 9*t2
    return np.array([[h11, h12], [h12, h22]])

def backtracking(x, direction, grad, alpha=0.1, beta=0.6):
    t = 1.0
    while f(x + t*direction) > f(x) + alpha*t*np.dot(grad, direction):
        t *= beta
    return t

def solve_opt(method='gradient', steps=50):
    x = np.array([-1.0, 1.0]) # Initial point
    history = []
    for _ in range(steps):
        history.append(f(x))
        g = grad_f(x)
        if method == 'gradient':
            d = -g
        else:
            H = hess_f(x)
            d = -np.linalg.solve(H, g)

        t = backtracking(x, d, g)
        x = x + t*d
    return history

gd_hist = solve_opt('gradient')
```

```
nm_hist = solve_opt('newton')

plt.plot(gd_hist, label='Gradient␣Descent')
plt.plot(nm_hist, label='Newton␣Method')
plt.yscale('log')
plt.legend()
plt.xlabel('Iteration')
plt.ylabel('f(x)')
plt.show()
```

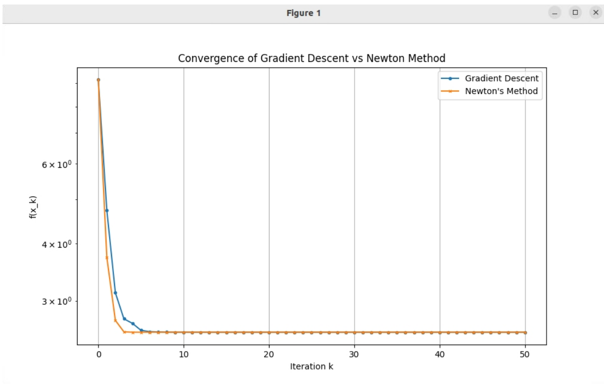

图 1: Convergence of Gradient Descent vs Newton Method