Optimization through School Geometry

G V V Sharma*

Contents

***Abstract*—This book provides an introduction to optimization based on the NCERT textbooks from Class 6-12. Links to sample Python codes are available in the text.**

Download python codes using

```
svn co https://github.com/gadepall/school/trunk/
    ncert/optimization/codes
```

1 Constrained Optimization

1. Express the problem of finding the distance of the point $\mathbf{P} = \begin{pmatrix} 3 \\ -5 \end{pmatrix}$ from the line

$$L: \quad \begin{pmatrix} 3 & -4 \end{pmatrix} \mathbf{x} = 26 \tag{1.1.1}$$

as an optimization problem.
Solution: The given problem can be expressed as

$$\min_{\mathbf{x}} g(\mathbf{x}) = \left\|\mathbf{x} - \mathbf{P}\right\|^2 \tag{1.1.2}$$

$$\text{s.t.} \quad \mathbf{n}^T\mathbf{x} = c \tag{1.1.3}$$

where

$$\mathbf{n} = \begin{pmatrix} 3 \\ -4 \end{pmatrix} \tag{1.1.4}$$

$$c = 26 \tag{1.1.5}$$

2. Explain Problem 1.1 through a plot and find a graphical solution.
3. Solve (1.1.2) using cvxpy.
Solution: The following code yields

```
codes/line_dist_cvx.py
```

$$\mathbf{x}_{\min} = \begin{pmatrix} 2.64 \\ -4.52 \end{pmatrix}, \tag{1.3.1}$$

$$g\left(\mathbf{x}_{\min}\right) = 0.6 \tag{1.3.2}$$

4. Convert (1.1.2) to an *unconstrained* optimization problem.
Solution: L in (1.1.1) can be expressed in terms of the direction vector $\mathbf{m}$ as

$$\mathbf{x} = \mathbf{A} + \lambda\mathbf{m}, \tag{1.4.1}$$

where $\mathbf{A}$ is any point on the line and

$$\mathbf{m}^T\mathbf{n} = 0 \tag{1.4.2}$$

Substituting (1.4.1) in (1.1.2), an unconstrained optimization problem

$$\min_{\lambda} f(\lambda) = \left\|\mathbf{A} + \lambda\mathbf{m} - \mathbf{P}\right\|^2 \tag{1.4.3}$$

is obtained.
5. Solve (1.4.3).
Solution:

$$f(\lambda) = \left(\lambda\mathbf{m} + \mathbf{A} - \mathbf{P}\right)^T\left(\lambda\mathbf{m} + \mathbf{A} - \mathbf{P}\right) \tag{1.5.1}$$

$$= \lambda^2\left\|\mathbf{m}\right\|^2 + 2\lambda\mathbf{m}^T\left(\mathbf{A} - \mathbf{P}\right) + \left\|\mathbf{A} - \mathbf{P}\right\|^2 \tag{1.5.2}$$

$$\because f^{(2)}\lambda = 2\left\|\mathbf{m}\right\|^2 > 0 \tag{1.5.3}$$

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

the minimum value of $f(\lambda)$ is obtained when

$$f^{(1)}(\lambda) = 2\lambda \|\mathbf{m}\|^2 + 2\mathbf{m}^T (\mathbf{A} - \mathbf{P}) = 0 \tag{1.5.4}$$

$$\implies \lambda_{\min} = -\frac{\mathbf{m}^T (\mathbf{A} - \mathbf{P})}{\|\mathbf{m}\|^2} \tag{1.5.5}$$

Choosing $\mathbf{A}$ such that

$$\mathbf{m}^T (\mathbf{A} - \mathbf{P}) = 0, \tag{1.5.6}$$

substituting in (1.5.5),

$$\lambda_{\min} = 0 \quad \text{and} \tag{1.5.7}$$

$$\mathbf{A} - \mathbf{P} = \mu \mathbf{n} \tag{1.5.8}$$

for some constant μ. (1.5.8) is a consequence of (1.4.2) and (1.5.6). Also, from (1.5.8),

$$\mathbf{n}^T (\mathbf{A} - \mathbf{P}) = \mu \|\mathbf{n}\|^2 \tag{1.5.9}$$

$$\implies \mu = \frac{\mathbf{n}^T \mathbf{A} - \mathbf{n}^T \mathbf{P}}{\|\mathbf{n}\|^2} = \frac{c - \mathbf{n}^T \mathbf{P}}{\|\mathbf{n}\|^2} \tag{1.5.10}$$

from (1.1.3). Substituting $\lambda_{\min} = 0$ in (1.4.3),

$$\min_{\lambda} f(\lambda) = \|\mathbf{A} - \mathbf{P}\|^2 = \mu^2 \|\mathbf{n}\|^2 \tag{1.5.11}$$

upon substituting from (1.5.8). The distance between $\mathbf{P}$ and L is then obtained from (1.5.11) as

$$\|\mathbf{A} - \mathbf{P}\| = |\mu| \|\mathbf{n}\| \tag{1.5.12}$$

$$= \frac{\left|\mathbf{n}^T \mathbf{P} - c\right|}{\|\mathbf{n}\|} \tag{1.5.13}$$

after substituting for μ from (1.5.10). Using the corresponding values from Problem (1.1) in (1.5.13),

$$\min_{\lambda} f(\lambda) = 0.6 \tag{1.5.14}$$

2 Convex Function

1. The following python script plots

$$f(\lambda) = a\lambda^2 + b\lambda + d \tag{2.1.1}$$

for

$$a = \|\mathbf{m}\|^2 > 0 \tag{2.1.2}$$

$$b = \mathbf{m}^T (\mathbf{A} - \mathbf{P}) \tag{2.1.3}$$

$$c = \|\mathbf{A} - \mathbf{P}\|^2 \tag{2.1.4}$$

where $\mathbf{A}$ is the intercept of the line L in (1.1.1) on the x-axis and the points

$$\mathbf{U} = \begin{pmatrix} \lambda_1 \\ f(\lambda_1) \end{pmatrix}, \mathbf{V} = \begin{pmatrix} \lambda_2 \\ f(\lambda_2) \end{pmatrix} \tag{2.1.5}$$

$$\mathbf{X} = \begin{pmatrix} t\lambda_1 + (1-t)\lambda_2 \\ f[t\lambda_1 + (1-t)\lambda_2] \end{pmatrix}, \tag{2.1.6}$$

$$\mathbf{Y} = \begin{pmatrix} t\lambda_1 + (1-t)\lambda_2 \\ tf(\lambda_1) + (1-t) f(\lambda_2) \end{pmatrix} \tag{2.1.7}$$

for

$$\lambda_1 = -3, \lambda_2 = 4, t = 0.3 \tag{2.1.8}$$

in Fig. 2.1. Geometrically, this means that any point $\mathbf{Y}$ between the points $\mathbf{U}, \mathbf{V}$ on the line UV is always above the point $\mathbf{X}$ on the curve $f(\lambda)$. Such a function f is defined to be *convex* function

```
codes/optimization/1.2.py
```

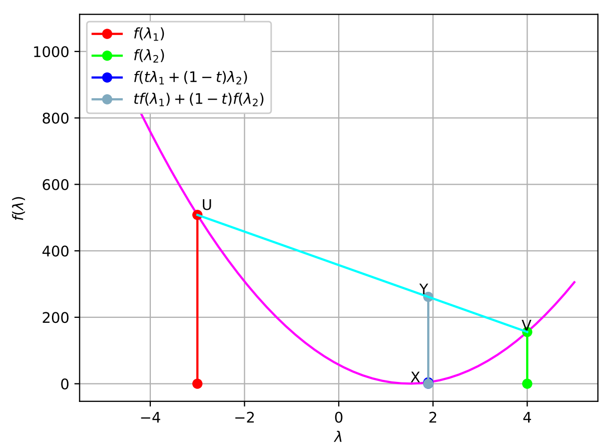

Fig. 2.1: $f(\lambda)$ versus λ

.

2. Show that

$$f[t\lambda_1 + (1-t)\lambda_2] \le tf(\lambda_1) + (1-t) f(\lambda_2) \tag{2.2.1}$$

for $0 < t < 1$. This is true for any convex function.

3. Show that

$$(2.2.1) \implies f^{(2)}(\lambda) > 0 \tag{2.3.1}$$

4. Show that a covex function has a unique minimum.

3 Gradient Descent

1. Find a numerical solution for (2.1.1)
Solution: A numerical solution for (2.1.1) is obtained as

$$\lambda_{n+1} = \lambda_n - \mu f'(\lambda_n) \tag{3.1.1}$$
$$= \lambda_n - \mu(2a\lambda_n + b) \tag{3.1.2}$$

where λ_0 is an inital guess and μ is a variable parameter. The choice of these parameters is very important since they decide how fast the algorithm converges.
2. Write a program to implement (3.1.2).
Solution: Download and execute

```
codes/optimization/gd.py
```

3. Find a closed form solution for λ_n in (3.1.2) using the one sided Z transform.
4. Find the condition for which (3.1.2) converges, i.e.

$$\lim_{n\to\infty} |\lambda_{n+1} - \lambda_n| = 0 \tag{3.4.1}$$

4 Lagrange Multipliers

1. Find

$$\min_{\mathbf{x}} g(\mathbf{x}) = \|\mathbf{x} - \mathbf{P}\|^2 = r^2 \tag{4.1.1}$$
$$\text{s.t.} \quad h(\mathbf{x}) = \mathbf{n}^T\mathbf{x} - c = 0 \tag{4.1.2}$$

by plotting the circles $g(\mathbf{x})$ for different values of r along with the line $g(\mathbf{x})$.
Solution: The following code plots Fig. 4.1

```
codes/concirc.py
```

2. By solving the quadratic equation obtained from (4.1.1), show that

$$\min_{\mathbf{x}} r = \frac{3}{5}, \mathbf{x}_{\min} = \mathbf{Q} = \begin{pmatrix} 2.64 \\ -4.52 \end{pmatrix} \tag{4.2.1}$$

In Fig. 4.1, it can be seen that $\mathbf{Q}$ is the point of contact of the line L with the circle of minimum radius $r = \frac{3}{5}$.
3. Show that

$$\nabla h(\mathbf{x}) = \begin{pmatrix} 3 \\ -4 \end{pmatrix} = \mathbf{n} \tag{4.3.1}$$

where

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{pmatrix} \tag{4.3.2}$$

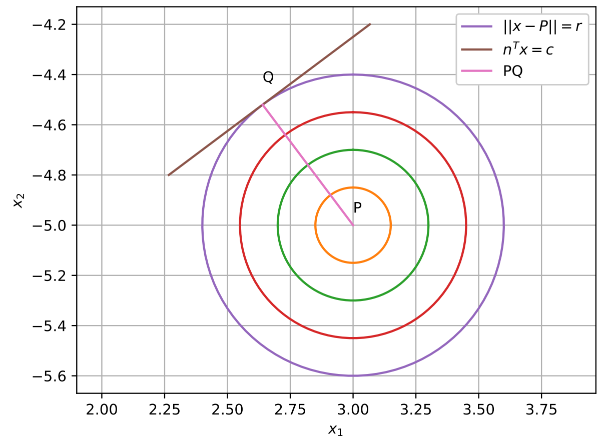

Fig. 4.1: Finding $\min_{\mathbf{x}} g(\mathbf{x})$

.

4. Show that

$$\nabla g(\mathbf{x}) = 2\left\{\mathbf{x} - \begin{pmatrix} 3 \\ -5 \end{pmatrix}\right\} = 2\{\mathbf{x} - \mathbf{P}\} \tag{4.4.1}$$

5. From Fig. 4.1, show that

$$\nabla g(\mathbf{Q}) = \lambda \nabla h(\mathbf{Q}), \tag{4.5.1}$$

Solution: In Fig. 4.1, PQ is the normal to the line L, represented by $h(\mathbf{x})$. $\because$ the normal vector of L is in the same direction as PQ, for some constant k,

$$(\mathbf{Q} - \mathbf{P}) = k\mathbf{n} \tag{4.5.2}$$

which is the same as (4.5.1) after substituting from (4.3.1). and (4.4.1).
6. Use (4.5.1) and $\mathbf{h}(\mathbf{Q}) = 0$ from (4.1.2) to obtain $\mathbf{Q}$.
Solution: From the given equations, we obtain

$$(\mathbf{Q} - \mathbf{P}) - \lambda\mathbf{n} = 0 \tag{4.6.1}$$
$$\mathbf{n}^T\mathbf{Q} - c = 0 \tag{4.6.2}$$

which can be simplifed to obtain

$$\begin{pmatrix} \mathbf{I} & -\mathbf{n} \\ \mathbf{n}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Q} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{P} \\ c \end{pmatrix} \tag{4.6.3}$$

The following code computes the solution to (4.6.3)

```
codes/lagmul.py
```

7. Define

$$C(\mathbf{x}, \lambda) = g(\mathbf{x}) - \lambda h(\mathbf{x}) \tag{4.7.1}$$

and show that $\mathbf{Q}$ can also be obtained by solving the equations

$$\nabla C(\mathbf{x}, \lambda) = 0. \tag{4.7.2}$$

What is the sign of λ? C is known as the Lagrangian and the above technique is known as the Method of Lagrange Multipliers.

8. Obtain $\mathbf{Q}$ using gradient descent.
Solution:

```
codes/gd_lagrange.py
```

5 Quadratic Programming

1. An apache helicopter of the enemy is flying along the curve given by

$$y = x^2 + 7 \tag{5.1.1}$$

A soldier, placed at

$$\mathbf{P} = \begin{pmatrix} 3 \\ 7 \end{pmatrix}. \tag{5.1.2}$$

wants to shoot the heicopter when it is nearest to him. Express this as an optimization problem.
Solution: The given problem can be expressed as

$$\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{P}\|^2 \tag{5.1.3}$$

$$\text{s.t. } \mathbf{x}^T\mathbf{V}\mathbf{x} + \mathbf{u}^T\mathbf{x} + d = 0 \tag{5.1.4}$$

where

$$\mathbf{V} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{5.1.5}$$

$$\mathbf{u} = -\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{5.1.6}$$

$$d = 7 \tag{5.1.7}$$

2. Show that the constraint in 5.1.3 is nonconvex.
3. Show that the following *relaxation* makes (5.1.3) a convex optimization problem.

$$\min_{\mathbf{x}} (\mathbf{x} - \mathbf{P})^T (\mathbf{x} - \mathbf{P}) \tag{5.3.1}$$

$$\text{s.t. } \mathbf{x}^T\mathbf{V}\mathbf{x} + \mathbf{u}^T\mathbf{x} \le 0 \tag{5.3.2}$$

4. Solve (5.3.1) using cvxpy.
Solution: The following code yields the minimum distance as 2.236 and the nearest point on the curve as

$$\mathbf{Q} = \begin{pmatrix} 1 \\ 8 \end{pmatrix} \tag{5.4.1}$$

```
codes/qp_cvx.py
```

5. Solve (5.3.1) using the method of Lagrange multipliers.
6. Graphically verify the solution to Problem 5.1.
Solution: The following code plots Fig. 5.6

```
codes/qp_parab.py
```

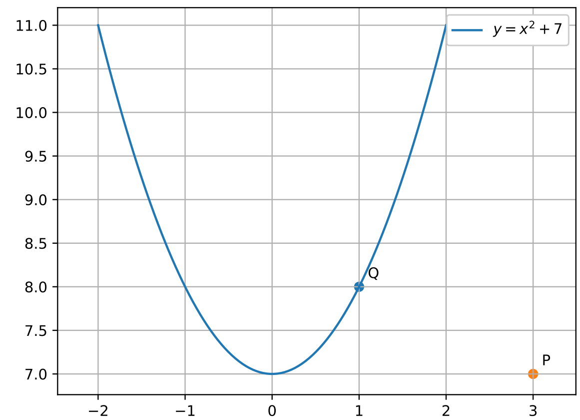

Fig. 5.6: $\mathbf{Q}$ is closest to $\mathbf{P}$

.

7. Solve (5.3.1) using gradient descent.

6 Semi Definite Programming

1. Express the problem of finding the point on the curve

$$x^2 = 2y \tag{6.1.1}$$

nearest to the point

$$\mathbf{P} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}. \tag{6.1.2}$$

as an optimization problem.
Solution: The given problem can be expressed as

$$\min_{\mathbf{x}} \mathbf{x}^T\mathbf{Q}_0\mathbf{x} + \mathbf{q}_0^T\mathbf{x} + c_0 \tag{6.1.3}$$

$$\text{s.t. } \mathbf{x}^T\mathbf{Q}_1\mathbf{x} + \mathbf{q}_1^T\mathbf{x} + c_1 \le 0 \tag{6.1.4}$$

where

$$\mathbf{Q}_0 = \mathbf{I}, \mathbf{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{6.1.5}$$

$$\mathbf{q}_0 = -2\mathbf{P}, \mathbf{q}_1 = -2\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6.1.6}$$

$$c_0 = \|\mathbf{P}\|^2, c_1 = 0 \tag{6.1.7}$$

2. Show that (6.1.3) is equivalent to

$$\min_{\mathbf{x},\theta} \theta$$

$$\text{s.t. } \begin{pmatrix} \mathbf{I} & \mathbf{M}_0\mathbf{x} \\ \mathbf{x}^T\mathbf{M}_0^T & -c_0 - q_0^T\mathbf{x} + \theta \end{pmatrix} \geq 0 \tag{6.2.1}$$

$$\begin{pmatrix} \mathbf{I} & \mathbf{M}_1\mathbf{x} \\ \mathbf{x}^T\mathbf{M}_1^T & -c_1 - q_1^T\mathbf{x} \end{pmatrix} \geq 0$$

where

$$\mathbf{Q}_i = \mathbf{M}_i^T\mathbf{M}_i, i = 0, 1 \tag{6.2.2}$$

3. Solve (6.2.1) using cvxpy.
4. Graphically verify the solution to Problem 6.1.
5. Solve (6.1.3) using the method of Lagrange multipliers.