JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Algorithms for Data Science

Optimization: Quadratic Programming

Quadratic Programming

Quadratic programming (QP) involves optimizing a quadratic objective function subject to linear constraints.

- **Standard Form:**

$$\text{Minimize } \frac{1}{2}x^T Q x + c^T x \text{ s.t. } Ax \leq b, x \geq 0$$

- **Where:**
  - Q : Symmetric positive semidefinite matrix, representing quadratic interactions between variables..
  - c : Coefficients for linear terms in the objective function.

$Ax \leq b$ Linear constraints limiting the solution space.

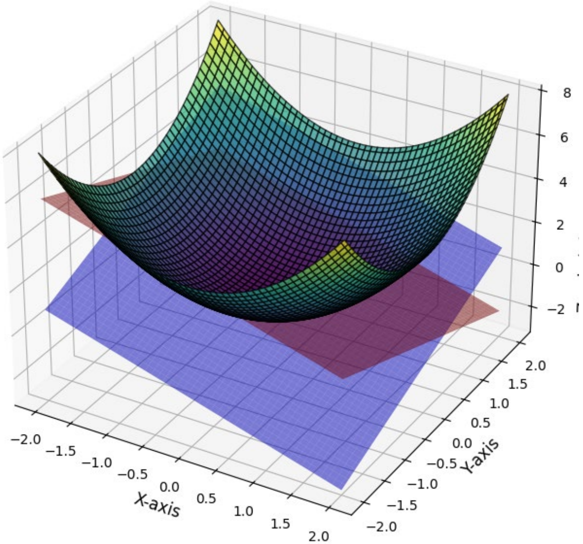


QP: Mathematical Formulation

- **Objective Function:**
  - **Quadratic Term:** $\frac{1}{2}x^TQx$ representing contributions of individual variables
  - **Linear Term:** $c^T x$ representing interactions between variables.

- **Constraints:**
  - **Linear Inequalities:** $Ax \leq$ limiting the solution space.
  - **Non-negativity:** $x \geq 0$ ensuring realistic solutions in practical problems.

- **Feasible Region:**
  Defined by the constraints, forming a convex polyhedron for convex QP problems.



Solving QP Problems

Interior Point Method

- Utilizes feasible region’s structure for efficient optimization.
- Operates by traversing the interior of the feasible region.

Active-Set Method

- Iteratively solves QP problems by focusing on a subset of active constraints.

Complexity

Convex problems can be solved in polynomial time based on their structures nature and depend on problem nize (n) and constraints (m)

- Scales efficiently for high-dimensional problems.
- Scales well for problems with sparse constraints.

QP in Python

Example:

$$\text{Minimize } \frac{1}{2}x_1^2 + x_2^2 - x_1x_2 - 3x_1 - 2x_2$$

s.t.

$$x_1 + x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

Mathematical Formulation:

- **Quadratic Term:** $\frac{1}{2}x^TQx$

$$Q = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix}$$

- **Linear Term:** c^Tx

$$c = \begin{bmatrix} -3 \\ -2 \end{bmatrix}$$

- **Constraints:**

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

QP in Python (cont.)

```
from cvxopt import matrix, solvers

# Define Q (scaled for CVXOPT) and c
Q = matrix([[1.0, -0.5], [-0.5, 2.0]]) # Scaled Q
matrix
c = matrix([-3.0, -2.0])  # Linear coefficients

# Inequality constraints Gx <= h
G = matrix([[1.0, 1.0], [-1.0, 0.0], [0.0, -1.0]], (3,
2), 'd')
h = matrix([2.0, 0.0, 0.0], (3, 1),
'd')

# Solve the QP problem
sol = solvers.qp(Q, c, G, h)

# Print the optimal solution and value
print("Optimal Solution:", sol['x'])
print("Optimal Value:", sol['primal objective'])
```

Solving a quadratic program

Mathematical Formulation:

- **Quadratic Term:** $\frac{1}{2}x^TQx$

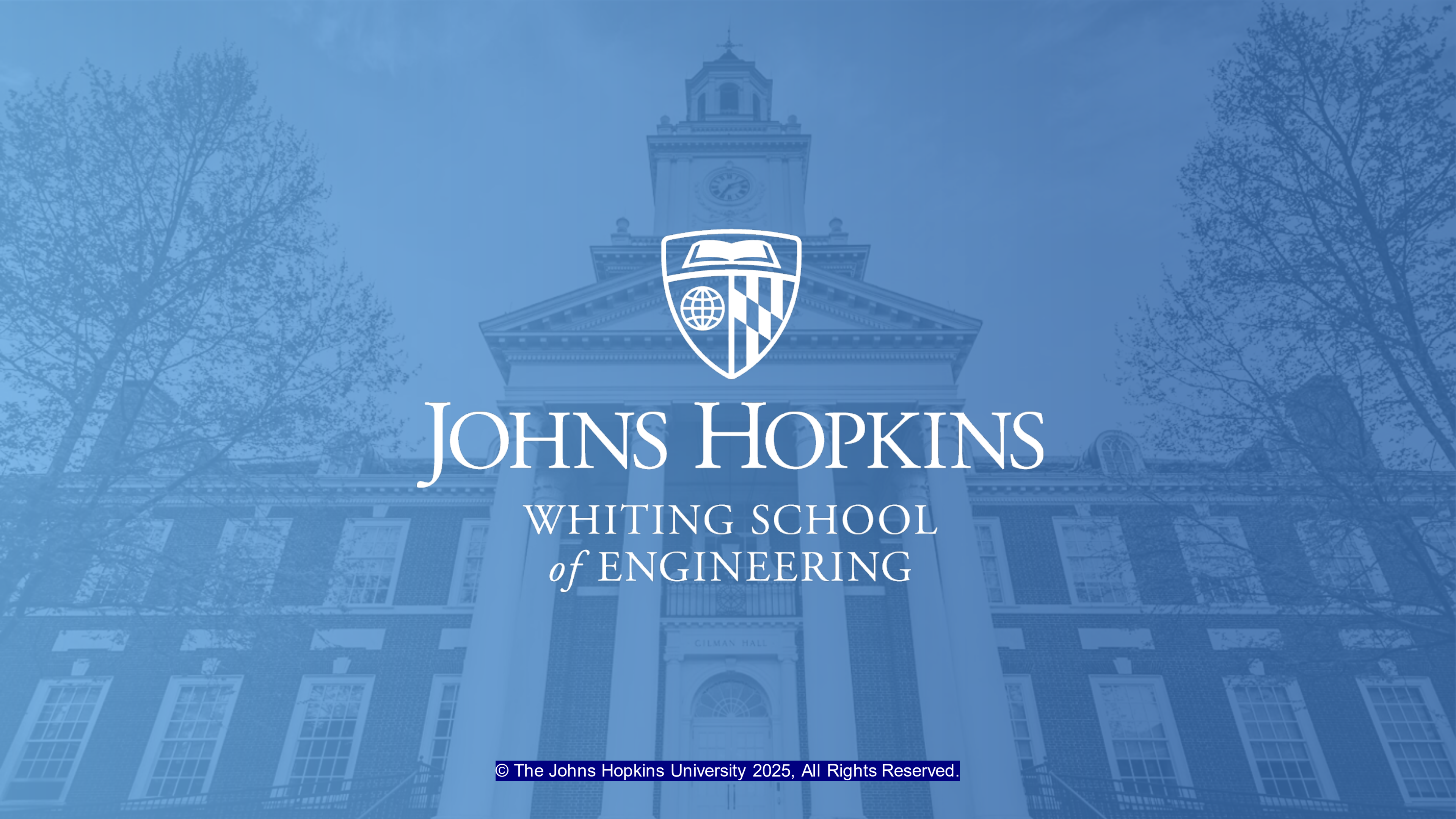
$$Q = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix}$$

- **Linear Term:** c^Tx

$$c = \begin{bmatrix} -3 \\ -2 \end{bmatrix}$$

- **Constraints:**

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING