

Chapter 483

Quadratic Programming

Introduction

Quadratic programming maximizes (or minimizes) a quadratic objective function subject to one or more constraints. The technique finds broad use in operations research and is occasionally of use in statistical work.

The mathematical representation of the quadratic programming (QP) problem is

Maximize

$$z = \mathbf{CX} + \frac{1}{2}\mathbf{X}'\mathbf{HX} \quad \text{or} \quad z = \mathbf{CX} + \mathbf{X}'\mathbf{DX}$$

subject to

$$\mathbf{AX} \leq \mathbf{b}, \mathbf{X} \geq \mathbf{0}$$

where

$$\mathbf{X} = (x_1, x_2, \dots, x_n)'$$

$$\mathbf{C} = (c_1, c_2, \dots, c_n)$$

$$\mathbf{b} = (b_1, b_2, \dots, b_m)'$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{n1} & \cdots & h_{nn} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & d_{nn} \end{bmatrix}$$

The symmetric matrix \mathbf{H} is often called the Hessian. The upper-triangular matrix \mathbf{D} is constructed from \mathbf{H} using

$$\mathbf{D} = \begin{pmatrix} 2h_{11} & \cdots & h_{1i} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 2h_{ii} & \cdots & h_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 2h_{nn} \end{pmatrix}$$

The x_i 's are the *decision variables* (the unknowns), the first equation is called the *objective function* and the m inequalities (and equalities) are called *constraints*. The constraint bounds, the b_i 's, are often called *right-hand sides* (RHS).

Quadratic Programming

NCSS solves a particular quadratic program using a primal active set method available in the *Extreme Optimization* mathematical subroutine package.

Example

We will solve the following problem using NCSS:

Minimize

$$z = x_1 - 2x_2 + 4x_3 + x_1^2 + 2x_2^2 + 3x_3^2 + x_1x_3$$

subject to

$$3x_1 + 4x_2 - 2x_3 \leq 10$$

$$-3x_1 + 2x_2 + x_3 \geq 2$$

$$2x_1 + 3x_2 + 4x_3 = 5$$

$$0 \leq x_1 \leq 5$$

$$1 \leq x_2 \leq 5$$

$$0 \leq x_3 \leq 5$$

The solution (see Example 1 below) is $x_1 = 0.290$, $x_2 = 1.413$, and $x_3 = 0.045$, which results in $z = 1.741$.

Data Structure

This technique requires a special data format which will be discussed under the *Specifications* tab. Here is the way the above example would be entered. It is stored in the dataset QP.

QP dataset

Type	Logic	RHS	X1	X2	X3	D1	D2	D3
O			1	-2	4	1	0	1
C	<	10	3	4	-2		2	0
C	>	2	-3	2	1			3
C	=	5	2	3	4			
L			0	1	0			
U			5	5	5			

Procedure Options

This section describes the options available in this procedure.

Specifications Tab

Set the specifications for the analysis.

Optimum Type

Type of Optimum

Specify whether to find the minimum or the maximum of the objective function in the constrained region.

Coefficients of the Objective Function, Constraints, and Bounds

Row Type Column

The constrained minimization problem is specified on the spreadsheet. In this column, you indicate the type of information (objective function, constraint, upper bound, or lower bound) that is on each row by entering one of the four letters: O, C, U, or L. Note that the order of the rows does not matter.

The possible cell entries in this column are

- **O**
This row contains the objective function defined by the coefficients (costs).
- **C**
This row contains a constraint defined by logic (<, >, or =), the RHS, and the coefficients.
- **L**
This row contains the optional lower bound for each variable.
- **U**
This row contains the optional upper bound for each variable.

Linear Portion of Objective Function, Constraints, and Bounds

Variables Columns

Specify the columns containing the coefficients of the variables. Each column corresponds to a variable in the constraints and objective function. Each row represents an individual constraint, the objective function, or upper or lower bounds. The program interprets the rows according to the corresponding values of the Row Type column.

The coefficients can be either positive or negative. Note that blanks are treated as zeros.

If you to enter upper bounds or lower bounds, enter them as rows: one for the upper bounds and another for the lower bounds.

For example, consider a problem to minimize the weighted sum of two variables X and Y subject to several constraints. Here is the generic formulation.

Quadratic Programming

Minimize: $X + 2Y + 2X^2 + 2XY + 2Y^2$

s.t.

$$X - Y \leq 3$$

$$X + 2Y \geq 6$$

$$3X - 4Y = 10$$

with bounds

$$4 \leq X \leq 20$$

$$2 \leq Y \leq 22$$

This is entered in seven columns on the spreadsheet as follows (note that H is in the 6th and 7th columns)

C1	C2	C3	C4	C5	C6	C7
O			1	2	4	2
C	<	3	1	-1	2	4
C	>	6	1	2		
C	=	10	3	-4		
L			4	2		
U			20	22		

Labels of Constraints Column

Specify a column containing a label for each constraint (*Row Type* = 'C'). These labels are used to make the output of this procedure easier to interpret.

Labels in non-constraint rows are ignored.

Quadratic Portion of Objective Function

Input Type of Quadratic Terms

Specify how you want to enter the quadratic portion of the objective function. The possibilities are as a Hessian matrix or as the coefficients of the quadratic equation.

- Hessian Matrix**
 A quadratic-programming objective function is often specified with the Hessian matrix. This option lets you specify the columns containing the Hessian matrix.
- Quadratic Coefficients Columns**
 This option lets you specify the coefficients of the quadratic terms in an upper-triangular matrix format. This is often easier than trying to specify the Hessian matrix.

Hessian Matrix Columns

Enter the columns containing the Hessian matrix. The Hessian matrix is a square, symmetric matrix whose dimension are equal to the number of variables (not constraints). Hence, the order of the variables assigned to the rows must be the same as the order of the variables assigned to the columns. This order must also match the order of the variables in the constraints.

Quadratic Programming

Deriving the Hessian Matrix

The quadratic portion of the objective function contains squared and cross-product terms with associated coefficients. An example is

$$3x^2 + 2y^2 + 4z^2 + xy + 2xz + 3yz$$

The Hessian matrix is written using matrix notation as $1/2 X'HX$, where $X' = (x_1, x_2, \dots)$ and H is symmetric and positive semi-definite. The Hessian can be obtained from the quadratic terms by writing down the cross-product term coefficients and doubling the squared-term coefficients.

For the example above, the Hessian is

	x	y	z
x	6	1	2
y	1	4	3
z	2	3	8

In a general three-variable problem, the quadratic portion is

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz.$$

The corresponding 3x3 Hessian matrix is

2a	d	e
d	2b	f
e	f	2c

Quadratic Coefficients Columns

Enter the columns containing the quadratic coefficients arranged in a matrix format. The order of the variables assigned to the rows must be the same as the order of the variables assigned to the columns. This order must also match the order of the variables in the constraints.

Deriving the Quadratic Coefficients Matrix

The quadratic portion of the objective function contains squared and cross-product terms with associated coefficients. An example is

$$3x^2 + 2y^2 + 4z^2 + xy + 2xz + 3yz$$

The quadratic coefficients matrix is constructed by writing the squared terms on the diagonal and the cross-product terms appropriately about the diagonal. The lower portion may be left blank.

For the example above, the quadratic coefficients matrix is

	x	y	z
x	3	1	2
y		2	3
z			4

In a general three-variable problem, the quadratic portion is

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz.$$

The corresponding 3x3 quadratic coefficient matrix is

a	d	e
	b	f
		c

Quadratic Programming

Logic and Constraint Bounds (RHS)

Logic Column

Specify the column containing the logic values for the constraints. Note that the cells in this column are only used for constraints (rows whose *Row Type* is "C").

Possible Logic Values

- <
For less than or equal. You can also use "LE" or "<=".
Example: $X1 + X2 \leq 6$.
- >
For greater than or equal. You can also use "GE" or ">=".
Example: $X1 + X2 \geq 4$.
- =
For equal. You can also use "EQ".
Example: $X1 + X2 = 3$.

Constraint Bounds (RHS) Column

Specify the bound (RHS = right-hand-side) of each of the constraints, one per row.

See the Example in the *Variables Columns* help.

Reports Tab

Select Reports

Objective Function and Solution – Quadratic Portion of the Objective Function

Indicate which reports you want to view.

Report Options

Variable Names

This option lets you select whether to display only variable names, variable labels, or both.

Precision

Specify the precision of numbers in the report. Single precision will display seven-place accuracy, while double precision will display thirteen-place accuracy.

Report Options – Decimal Places

Input Coefficients – Calculated Values

These options let you designate the number of decimal places to be displayed for each type of variable.

Quadratic Programming

Example 1 – Quadratic Programming

This section presents an example of how to run the data presented in the example given above. The data are contained in the QP database. Here is the specification of the problem.

Minimize

$$z = x_1 - 2x_2 + 4x_3 + x_1^2 + 2x_2^2 + 3x_3^2 + x_1x_3$$

subject to

$$3x_1 + 4x_2 - 2x_3 \leq 10$$

$$-3x_1 + 2x_2 + x_3 \geq 2$$

$$2x_1 + 3x_2 + 4x_3 = 5$$

$$0 \leq x_1 \leq 5$$

$$1 \leq x_2 \leq 5$$

$$0 \leq x_3 \leq 5$$

You may follow along here by making the appropriate entries or load the completed template **Example 1** by clicking on Open Example Template from the File menu of the *Quadratic Programming* window.

1 Open the QP dataset.

- From the **File** menu of the NCSS Data window, select **Open Example Data**.
- Click on the file **QP.NCSS**.
- Click **Open**.

2 Open the Quadratic Programming window.

- Using the Analysis menu or the Procedure Navigator, find and select the **Quadratic Programming** procedure.
- On the menus, select **File**, then **New Template**. This will fill the procedure with the default template.

3 Specify the problem.

- On the Quadratic Programming window, select the **Specifications tab**.
- Set **Type of Optimum** to **Minimum**.
- Double-click in the **Row Type Column** text box. This will bring up the column selection window.
- Select **Type** from the list of columns and then click **Ok**. “Type” will appear in this box.
- Double-click in the **Variables Columns** text box. This will bring up the column selection window.
- Select **X1-X3** from the list of columns and then click **Ok**. “X1-X3” will appear in this box.
- Double-click in the **Labels of Constraints Column** text box. This will bring up the column selection window.
- Select **CLabel** from the list of columns and then click **Ok**. “CLabel” will appear in this box.
- Set **Input Type of Quadratic Terms** to **Quadratic Coefficients** since this problem uses the D matrix.
- Double-click in the **Quadratic Coefficients Columns** text box. This will bring up the column selection window.
- Select **D1-D3** from the list of columns and then click **Ok**. “D1-D3” will appear in this box.
- Double-click in the **Logic Column** text box. This will bring up the column selection window.
- Select **Logic** from the list of columns and then click **Ok**. “Logic” will appear in this box.
- Double-click in the **Constraint Bounds (RHS) Column** text box. This will bring up the variable selection window.
- Select column **RHS** from the list of columns and then click **Ok**. “RHS” will appear in this box.

4 Run the procedure.

- From the **Run** menu, select **Run Procedure**. Alternatively, just click the green Run button.

Quadratic Programming

Objective Function and Solution for Minimum

Objective Function and Solution for Minimum

Variable	Objective Function Coefficient	Value at Minimum	Lower Bound	Upper Bound
X1	1.0	0.290	0.0	5.0
X2	-2.0	1.413	1.0	5.0
X3	4.0	0.045	0.0	5.0
Minimum of Objective Function		1.741		

Solution Status: The optimization model is optimal.

This report lists the linear portion of the objective function coefficients, the values of the variables at the minimum (that is, the solution), and the lower and upper bounds if specified. It also shows the value of the objective function at the solution as well as the status of the algorithm when it terminated.

Constraints

Constraints

Label, Logic	X1	X2	X3	RHS
Con1, \leq	3.0	4.0	-2.0	10.0
Con2, \geq	-3.0	2.0	1.0	2.0
Con3, =	2.0	3.0	4.0	5.0

This report presents the coefficients of the constraints as they were input.

Values of Constraints at Solution for Minimum

Values of Constraints at Solution for Minimum

Row, Logic	RHS	RHS at Solution
Con1, \leq	10.0	6.435
Con2, \geq	2.0	2.000
Con3, =	5.0	5.000

This report presents the right hand side of each constraint along with its value at the optimal values of the variables.

Hessian Matrix

Hessian Matrix

Variables	X1	X2	X3
X1	2.0	0.0	1.0
X2	0.0	4.0	0.0
X3	1.0	0.0	6.0

This report shows the Hessian matrix calculated from the D matrix that was input.

Quadratic Portion of the Objective Function

Quadratic Portion of the Objective Function

Variables	X1	X2	X3
X1	1.0	0.0	1.0
X2		2.0	0.0
X3			3.0

This report shows the coefficients of the quadratic portion of the objective function presented in matrix format.