Contents

MIE377 (Winter 2024) - Laboratory 1

The purpose of this program is to introduce the functions 'linprog' and 'quadprog' through two financial examples

TA: David Islip Instructor: Roy H. Kwon


```
% Name: Aaryan Nagpal
% Student ID: 1007792596

clc
clear all
format short

% Program Start
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

PART 1: linprog

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Suppose a bank has liabilities at the end of the next three years:
%
% Year 1: $12,000   Year2: $18,000      Year 3: $20,000
%
% and the bank wishes to meet these liabilities by buying units of three
% bonds
%
% Bond      Price       Yearly Coupon       Maturity Year
% 1         $102        $5                  1
% 2         $99         $3.5                2
% 3         $98         $3.5                3
%
% All bonds have a face value of $100 (i.e., F_i = 100 for i = 1,2,3). The
% bank must meet or exceed their liability for each year end. How many
% units of each bond should the bank purchase to meet the liabilities with
% minimum cost? Solve using 'linprog'.
%
%-------------------------------------------------------------------------
% Formulate the Linear Program
%-------------------------------------------------------------------------
% MATLAB requires the input in the following format
%
%   min     c' x
%   s.t.    A x <= b
%           Aeq x = beq
%           lb <= x <= ub
%
% And, based on our problem, we have that
%
%   min     P_1 x_1 + P_2 x_2 + P_3 x_3
%   s.t.    (F_1 + c_1) x_1 + c_2 x_2 + c_3 x_3 >= L_1
%                     (F_2 + c_2) x_2 + c_3 x_3 >= L_2
%                               (F_3 + c_3) x_3 >= L_3
%           x_1 >= 0, x_2 >= 0, x_3 >= 0
%
%-------------------------------------------------------------------------
% Solve using 'linprog'
%-------------------------------------------------------------------------

% Constraints - Each row corresponds to each scheduled liability
```

```
%Making the A and b matrix entries negative to get them in form required by
%linprog, obtained by using the constraint equations above
A = -[105 3.5 3.5;
    0 103.5 3.5;
    0 0 103.5];

% Our liabilities for years 1, 2, and 3
b = -[12000; 18000; 20000];

% Cost vector (price of the bonds)
c = [102 99 98];

% Disallow short-selling by setting the lower bound to 0 for all 3 bond
% quantities
lb = [0, 0, 0];

% Note: We do not have equality constraints, nor do we have an upper bound
% on our decision variable x. Therefore, we can leave the entries blank
% '[]' in the linprog function for Aeq, beq, and ub.

% Solve this problem using 'linprog'
x = linprog(c, A, b, [], [], lb, [] );
%display(x)

%displaying the optimized bond quantities outputted by linprog
for i = 1:length(x)
    fprintf('The number of bonds of type bond %d requied are %.4f\n', i ,x(i,1))
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Optimal solution found.

The number of bonds of type bond 1 requied are 102.2652
The number of bonds of type bond 2 requied are 167.3785
The number of bonds of type bond 3 requied are 193.2367
```

PART 2: Quadprog

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Suppose we wish to find the minimum variance portfolio from a combination
% of stocks.
%
% Use the weekly price data provided in the file ?lab1_prices.mat? to
% produce estimates of the covariance and expected weekly return. Find the
% minimum variance portfolio subject to no shortselling and an expected
% weekly return of 0.25%. Formulate the problem as a quadratic program and
% solve using 'quadprog'.
%
%-------------------------------------------------------------------------
% Formulate the Quadratic Program
%-------------------------------------------------------------------------
% MATLAB requires the input in the following format
%
%   min     (1/2) x' H x + c' x
%   s.t.    A x <= b
%           Aeq x = beq
%           lb <= x <= ub
%
% And, based on our problem, we have that
%
%   min     x' Q x
%   s.t.    mu' x >= 0.0025
%           sum( x ) = 1
%           x >= 0
%
```

```
%------------------------------------------------------------------------
% Solve using 'Quadprog'
%------------------------------------------------------------------------
% Load the historical price data for 50 assets
load('lab1_prices.mat')

%the rets were computed using for loops before the practical session, thus
%I am keeping the loops, everything else is done the way it was done in the
%practical demo.
% Compute the returns
rets = zeros(size(prices,1)-1,size(prices,2));
for i = 1:size(prices,2)
    for t=2:size(prices,1)
        rets(t-1,i) = (prices(t,i)-prices(t-1,i))/prices(t-1,i);
    end
end

% Estimate the geometric mean and the covariance matrix
%Using the geomean function on the computed rets to obtain a geometric mean
%return for all assets
mu = geomean(rets+1)-1;

%As explained in the practical since rets = [r_it], we can just use the cov
%function to obtain the covariance matrix
Q  = cov(rets);

% Disallow shortselling by setting a lower bound of 0 for all asset
% weights. The dimensions of the vector of zeros is the same as the number
% of assets in the prices matrix.
lb = zeros(1,size(prices,2));

% Add the expected return constraint
% Taking -A to be expected returns matrix and setting it less than equal to
% -0.0025 (required return) to obtain the constraint in the form
% required by quadprog
A = -(mu);
b = -0.0025;

%constrain weights to sum to 1
% Creating Aeq, a vector of ones and setting beq equal to 1 to obtain the
% second constraint as required by quadprog. The dimension for Aeq is set
% to be equal to the number of assets in the prices matrix
Aeq = ones(1,size(prices,2));
beq = 1;

% It might be useful to increase the tolerance of 'quadprog'
options = optimoptions( 'quadprog', 'TolFun', 1e-9 );

% Note: We do not have a linear component of our objective function, nor do
% we have an upper bound on our decision variable x. Therefore, we can
% leave the corresponding entry blank '[]' in the quadprog function for c
% and ub.

% Solve this problem using 'quadprog'
x = quadprog( 2 * Q, [], A, b, Aeq, beq, lb, [], [], options );
%display(x)

%displaying the optimized asset weights outputted by quadprog
for i = 1:size(prices,2)
    fprintf('Weight for asset %d is %.4f \n',i, x(i,1))
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program End
```

```
Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
```

```
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

Weight for asset 1 is 0.0188
Weight for asset 2 is 0.0000
Weight for asset 3 is 0.0000
Weight for asset 4 is 0.0000
Weight for asset 5 is 0.0000
Weight for asset 6 is 0.0000
Weight for asset 7 is 0.0074
Weight for asset 8 is 0.0000
Weight for asset 9 is 0.0529
Weight for asset 10 is 0.0000
Weight for asset 11 is 0.1157
Weight for asset 12 is 0.0896
Weight for asset 13 is 0.0000
Weight for asset 14 is 0.0302
Weight for asset 15 is 0.0000
Weight for asset 16 is 0.0000
Weight for asset 17 is 0.0044
Weight for asset 18 is 0.0000
Weight for asset 19 is 0.0000
Weight for asset 20 is 0.0000
Weight for asset 21 is 0.0000
Weight for asset 22 is 0.0000
Weight for asset 23 is 0.0000
Weight for asset 24 is 0.0213
Weight for asset 25 is 0.0460
Weight for asset 26 is 0.0000
Weight for asset 27 is 0.1352
Weight for asset 28 is 0.0000
Weight for asset 29 is 0.0106
Weight for asset 30 is 0.0000
Weight for asset 31 is 0.0000
Weight for asset 32 is 0.0341
Weight for asset 33 is 0.0000
Weight for asset 34 is 0.0000
Weight for asset 35 is 0.0879
Weight for asset 36 is 0.1142
Weight for asset 37 is 0.1439
Weight for asset 38 is 0.0000
Weight for asset 39 is 0.0000
Weight for asset 40 is 0.0000
Weight for asset 41 is 0.0204
Weight for asset 42 is 0.0000
Weight for asset 43 is 0.0000
Weight for asset 44 is 0.0582
Weight for asset 45 is 0.0000
Weight for asset 46 is 0.0080
Weight for asset 47 is 0.0010
Weight for asset 48 is 0.0000
Weight for asset 49 is 0.0000
Weight for asset 50 is 0.0000
```

Published with MATLAB® R2023b