Mathematic Analysis with Matlab

Lecture 13: Non-Linear Programming

Lecturer: *Dr*. Lian-Sheng Wang

Fall Semester 2022

Contact: lswang@xmu.edu.cn

Outline

1. Non-Linear Programming: with Constraints
2. Non-Linear Programming: without Constraint
3. Course Project

Quadratic Programming: the problem

- Given following problem:
- A company is going to invest 5000$ on two projects **A** and **B**. Given x_1 and x_2 are the amount of money that will be allocated to project **A** and **B** respectively. The annual profits for **A** and **B** are 70% and 66% respectively. Meanwhile, the risk of loss is related to a function $0.02x_1^2 + 0.01x_2^2 + 0.04(x_1 + x_2)^2$
- **Question: build a plan to maximize the expected profit, while minimizing the possible loss**

1. Target1: maximize $1.7 * x_1 + 1.66 * x_2$
2. Target2: minimize $0.02x_1^2 + 0.01x_2^2 + 0.04(x_1 + x_2)^2$

Linear Programming: modeling (1)

1. Target1: maximize $1.7x_1 + 1.66x_2$
2. Target2: minimize $0.02x_1^2 + 0.01x_2^2 + 0.04(x_1 + x_2)^2$
3. Explicit constraint: $x_1 + x_2 \leq 5000$
4. Implicit constraint: $x_1 \geq 0$, $x_2 \geq 0$

- Q: How to merge two conflicting targets??
- A: Depends on your consideration
- Half-by-half (0.5 for each) is a lazy trade-off
- Then the target is:

  $\min\ 0.5(0.02x_1^2 + 0.01x_2^2 + 0.04(x_1 + x_2)^2) - 0.5(1.7x_1 + 1.66x_2)$

- re-organize it as

$$\min\ 0.03x_1^2 + 0.025x_2^2 + 0.04x_1x_2 - 0.85x_1 - 0.83x_2$$

Linear Programming: modeling (2)

- Complete model for the problem:

$$\min\ 0.03x_1^2 + 0.025x_2^2 + 0.04x_1x_2 - 0.85x_1 - 0.83x_2$$

$$\text{s.b.t} \begin{cases} x_1 + x_2 \leq 5000 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

- Standard Quadratic Optimization form:

$$\min\ \frac{1}{2}x^T Hx + c^T x$$
$$\text{s.b.t.}\ Ax \preccurlyeq b$$
$$Aeqx = Beq$$
$$lb \preccurlyeq x \preccurlyeq ub$$

$$H = \begin{bmatrix} 0.06 & 0.04 \\ 0.04 & 0.05 \end{bmatrix}, c = [\text{-}0.85\ \text{-}0.83], A = [1\ 1], b = [5000]$$

$$Aeq = [], Beq = [], lb = [0\ 0], ub = [].$$

- Solve it by Matlab: [x, fval]=quadprog(H,c,A,b,Aeq,Beq,lb,ub)

Alternative solution (1)

$$\min\ 0.03x_1^2 + 0.025x_2^2 + 0.04x_1x_2 - 0.85x_1 - 0.83x_2$$
$$\text{subject to} \begin{cases} x_1 + x_2 \leq 5000 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

$$H = \begin{bmatrix} 0.06 & 0.04 \\ 0.04 & 0.05 \end{bmatrix}, c = [\text{-}0.85\ \text{-}0.83], A = [1\ 1], b = [5000]$$
$$Aeq = [], Beq = [], lb = [0\ 0], ub = [].$$

- Step 1: define a matlab function
  1. function f=minTarget(x)
  2. f=0.03*x(1)^2+0.025*x(2)^2+0.04*x(1)*x(2)-0.85*x(1)-0.83*x(2);
  3. end

```
H = [0.06, 0.04;0.04 0.05];
C = [-0.85 -0.83];
Ae = [1 1];
be = [5000];
[x fval] = quadprog(H, C, [],[], Ae, be, [], [])
f1 = x(1)*1.7+x(2)*1.66
```

Listing 1: Treat A as equation constraint

```
H = [0.06, 0.04;0.04 0.05];
C = [-0.85 -0.83];
A = [1 1];
b = [5000];
[x fval] = quadprog(H, C, A, b, [], [], [], [])
f1 = x(1)*1.7+x(2)*1.66
```

Listing 2: Treat A as inequation constraint

Alternative solution (2)

- Step 1: define a matlab function

```
1 function f=minTarget(x)
2   f=0.03*x(1)^2+0.025*x(2)^2+0.04*x(1)*x(2)-0.85*x(1)-0.83*x(2);
3 end
```

- Step 2: call 'fmincon'

```
1 x0=[1000, 1000];
2 A=[1 1]; b=5000;
3 lb=zeros(2, 1);
4 [x, fval, flag]=fmincon(@minTarget,x0,A,b,[],[],lb,[])
5 f1=1.70*x(1)+1.66*x(2)
6 f2=0.02*x(1)^2+0.01*x(2)^2+0.04*(x(1)+x(2))^2
```

- Comments: no need to write out Hessian matrix explicitly. 'fmincon' is feasible for higher order (> 2) problem

Summary over **optimization problem** with constraint

$$\min\ f(x)$$
$$\text{subject to}\ \begin{cases} Ax \leq B \\ Aeq{\cdot}x = Beq \\ C(x) \leq 0 \\ Ceq(x) = 0 \\ lb \preceq x \preceq ub \end{cases}$$

- Step 1: define a matlab function

```
function f=minTarget(x)
  f=f(x);
end
```

```
function [g1,g2,...]=nonCon(x)
  g1=g1(x);
  g2=g2(x);
  ...;
end
```

- Step 2: call ‘fmincon’
  1. [x, fval]=fmincon(@minTarget,x0,A,b,Aeq,Beq,lb,ub,@nonCon)

Solve optimization problem with **fmincon**

$$\min\ x_1^2 + x_2^2 + 8$$

$$\text{subject to} \begin{cases} x_1^2 - x_2 \geq 0 \\ -x_1 - x_2^2 + 2 = 0 \\ x_1, x_2 \geq 0 \end{cases}$$

- Step 1: define a matlab function

```
function f=mTarget(x)
    f=x(1)^2+x(2)^2+8;
end
```

```
function [g1,g2]=nonCon(x)
    g1=-x(1)^2+x(2);
    g2=-x(1)-x(2)^2+2;
end
```

[Step 2: call 'fmincon']

```
options=optimset;
[x, fval]=fmincon(@mTarget,rand(2,1),[],[],[],[],zeros(2,1),[],@nonCon,
    options)
```

An alternative solution: Lagrange-multiplier (a reminder)

$$\text{extreme } f(x)$$

$$\text{subject to } \begin{cases} g1(x) = 0 \\ g2(x) = 0 \\ \dots \end{cases}$$

- Step 1: define function L
  1. L=f(x)-λ_1g1(x)-λ_2g2(x)+...;
- Step 2: take partial derivative on L: $\frac{\partial L}{\partial x_1}$,...
- Step 3: solve the equations by 'solve(...)';
- Comments:
  1. It is feasible when all functions are continuous and differentiable
  2. All constraints are equations
  3. But not necessarily convex

Outline

1. Non-Linear Programming: with Constraints

2. Non-Linear Programming: without Constraint

3. Course Project

Min f(x): unconstrained

- Case 1: single variable
  1. [x,val]=fminbnd(@f, lb, ub)
- Case 2: multiple variables
  1. [x,val,exitflag]=fminsearch(@f, x0, options)
  2. [x,val,exitflag]=fminunc(@f,x0,options)
  3. help optimset;

Min f(x): example (1)

- Given

$$f(x) = 2x_1^3 + 4x_1x_2^3 - 10x_1x_2 + x_2^2 \tag{1}$$

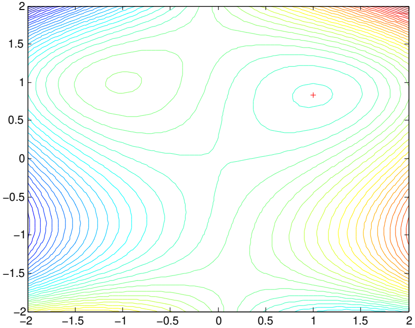
- Step 1:

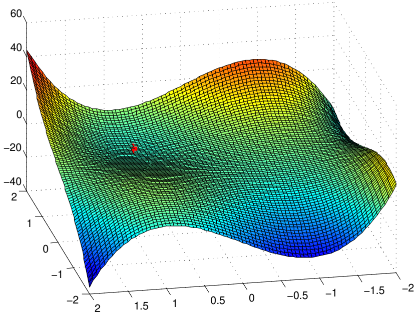
```
function f=minF(x)
  f=2*x(1)^3+4*x(1)*x(2)^3-10*x(1)*x(2)+x(2)^2
end
```

- Step 2:

```
[x, fval]=fminsearch(@minF,[0,0]);
```

Min f(x): example (1)

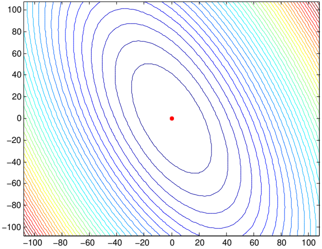


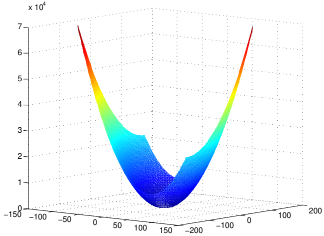


Min f(x): example (2)

$$f(x) = 3x_1^2 + 2x_1x_2 + x_2^2 \tag{2}$$

- Steps:
  1. f='3*x(1)^2+2*x(1)*x(2)+x(2)^2'
  2. [x, fval]=fminunc(f,zeros(1,2));





SUMT: the idea

$$\min\ f(x)$$

$$\text{subject to} \begin{cases} g_i(x) \leq 0, & i = 1, \cdots, r \\ h_i(x) \geq 0, & i = 1, \cdots, s \\ k_i(x) = 0, & i = 1, \cdots, t \end{cases}$$

- Step 1: construct a function P(x, M):

$$= f(x) + M\sum_{i=1}^{r} max(g_i(x), 0) - M\sum_{i=1}^{r} min(h_i(x), 0) + M\sum_{i=1}^{r} |k_i(x)|,$$

where M is a constant with big value

- Step 2: fminunc(P,x0)
- A way converts problem with constraints to a problem with no constraint
- This approach is called “**S**equential **U**nconstrained **M**inimization **T**echnique”

SUMT: the example

$$\min\ x_1^2 + x_2^2 + 8$$

$$\text{subject to } \begin{cases} x_1^2 - x_2 \geq 0 \\ -x_1 - x_2^2 + 2 = 0 \\ x_1, x_2 \geq 0 \end{cases}$$

[Step 1: construct a function:]

```
function g=minP(x)
  M = 50000;
  f = x(1)^2+x(2)^2+8;
  g = f-M*min(x(1),0)-M*min(x(2),0)-M*min(x(1)^2-x(2),0)+M*abs(-x(1)-x
     (2)^2+2);
end
```

[Step 2:]

```
   [x, fval]=fminunc(@minP,rand(2,1))
```

Exercise-1 (1)

- There are 5000$ to invest on two projects A and B, the profit are 20% and 16% respectively
- The risk of loss is given by $2{x_A}^2 + {x_B}^2 + (x_A + x_B)^2$
- Problem: how to set x_A and x_B to maximize the profit while minimizing the possible loss
- Given the loss factor (weight) as 0.2

Exercise-1 (2)

- Maximize: $1.2 \cdot x_A + 1.16 \cdot x_B$
- Minimize: $2{x_A}^2 + {x_B}^2 + (x_A + x_B)^2$
- Constraint: $x_A + x_B = 5000$
- Factor: 0.2

$$\Downarrow$$

- Minimize: $0.2 \cdot (2{x_A}^2 + {x_B}^2 + (x_A + x_B)^2) - 1.2 \cdot x_A - 1.16 \cdot x_B$
- Constraint: $x_A + x_B = 5000$

Exercise-1 (3)

- Minimize: $0.2{\cdot}(2{x_A}^2 + {x_B}^2 + (x_A + x_B)^2) - 1.2{\cdot}x_A - 1.16{\cdot}x_B$
- Constraint: $x_A + x_B = 5000$

$$\Downarrow$$

- Minimize: $0.6{\cdot}{x_A}^2 + 0.4{\cdot}{x_B}^2 + 0.4{\cdot}x_A{\cdot}x_B - 1.2{\cdot}x_A - 1.16{\cdot}x_B$
- Constraint: $x_A + x_B = 5000$

Exercise-1 (4)

- Maximize: $0.6 \cdot {x_A}^2 + 0.4 \cdot {x_B}^2 + 0.4 \cdot x_A \cdot x_B - 1.2 \cdot x_A - 1.16 \cdot x_B$
- Constraint: $x_A + x_B = 5000$

$$\Downarrow$$

$$H = \begin{bmatrix} 1.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix}$$

$$f = [-1.2, \quad -1.16]$$

$$Aeq = [1 \quad 1]$$

$$Beq = [5000]$$

Exercise-1 (5)

$$H = \begin{bmatrix} 1.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix}$$

$$f = [-1.2, -1.16]$$

$$Aeq = [1 \quad 1]$$

$$Beq = [5000]$$

```
H   = [1.2 0.4;0.4 0.8]
f   = [-1.2 -1.16]
Aeq = [1 1]
Beq = 5000
lb  = [0 0]
[x, fval] = quadprog(H, f,[], [], Aeq, Beq
    , lb,[])
```

- Answers: $x_A = 1667.3$, $x_B = 3332.7$

Outline

1. Non-Linear Programming: with Constraints
2. Non-Linear Programming: without Constraint
3. Course Project

- Two problems
- One is about gradient descent
- Another is about quadratic programming
- Requirements:
  - Solutions and reports (in English)
  - No cheating!!
  - Deadline: 2023/January/15, 23:59

Wish you all great success in your study and future career!!