Optimization of CIS

ECE 802

Tutorial 2 – Quadratic programming

L. Tziovani and A. Astolfi

Quadratic programming: A quadratic programming problem is a problem in which the objective function is quadratic and the constraints are linear.

Example 1 (a): Find the minimum of

$$f(x) = \frac{1}{2}{x_1}^2 + {x_2}^2 - x_1 x_2 - 2x_1 - 6x_2$$

Subject to

$$x_1 + x_2 \leq 2$$

$$-x_1 + 2x_2 \leq 2$$

$$2x_1 + x_2 \leq 3$$

Solve the problem using the Matlab optimization solver quadprog (https://www.mathworks.com/help/optim/ug/quadprog.html)

quadprog

Quadratic programming

Solver for quadratic objective functions with linear constraints.

quadprog finds a minimum for a problem specified by

$$\min_x \frac{1}{2} x^T H x + f^T x \text{ such that } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ lb \leq x \leq ub. \end{cases}$$

H, A, and Aeq are matrices, and f, b, beq, lb, ub, and x are vectors.

Solution: In quadprog syntax, this problem is to minimize

$$f(x) = \frac{1}{2} x^T H x + l^T x$$

with

$$H = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$: Hessian matrix of f

$$l = \begin{bmatrix} -2 \\ -6 \end{bmatrix}$$: Coefficients of the linear terms

Note: The Hessian matrix of a function $f: \mathbb{R}^n \to \mathbb{R}$ is

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\[2.2ex] \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\[2.2ex] \vdots & \vdots & \ddots & \vdots \\[2.2ex] \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Matlab code:

```
H = [1 -1; -1 2];
f = [-2; -6];
A = [1 1; -1 2; 2 1];
b = [2; 2; 3];
lb=[-inf -inf];
ub=[inf inf];
Aeq=[];
beq=[];

[x1,fval1,exitflag] =  quadprog(H,f,A,b,Aeq,beq,lb,ub);
```

Examine the final point, function value (fvall), and exit flag:

x1=[0.6667, 1.3333], f(x)=-8.22, exitflag=1

An exit flag of 1 means the result is a local minimizer. <u>Because H is a positive definite matrix, this problem is convex, so the minimizer is a global minimizer</u>. Confirm that H is positive definite by checking its eigenvalues:

eig(H)=[0.38; 2.61]

Example 1 (b): For Example 1 (a), compute the unconstrained global minimizer.

```
H = [1 -1; -1 2];
f = [-2; -6];
A = [];
b = [];
lb=[-inf -inf];
ub=[inf inf];
Aeq=[];
beq=[];

[x2,fval2,exitflag] =  quadprog(H,f,A,b,Aeq,beq,lb,ub);
```

x2=[10, 8], f(x)=-34, exitflag=1

Example 2: Find the minimum of

$$f(x) = -\frac{1}{2}{x_1}^2 + {x_2}^2 - x_1x_2 - 2x_1 - 6x_2$$

subject to

$$x_1 + x_2 \le 2$$

$$-x_1 + 2x_2 \le 2$$

$$2x_1 + x_2 \le 3$$

Matlab code:

```
H = [-1 -1; -1 2];
f = [-2; -6];
A = [1 1; -1 2; 2 1];
b = [2; 2; 3];
lb=[-inf -inf];
ub=[inf inf];
Aeq=[];
beq=[];

[x3,fval3,exitflag] = quadprog(H,f,A,b);

eigenvalues=eig(H);
```

Solution

```
The problem is non-convex.


x =

    1.2000
    1.3750


fval =

  -11.1294


exitflag =

    -6


eigenvalues =

   -1.3028
    2.3028
```

An exit flag of -6 (negative eigenvalues) means that the problem is non-convex. As a result the solution is a local minimizer and not the global minimizer (The exitflag options are described on the website of the solver).

Example 3: Consider a linear regression problem of the form:

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

Using as cost function the sum of the squared error estimate the parameters a_0, a_1, a_2, a_3, a_4 for the noisy dataset data11.mat and plot (a) the noisy data and (b) the estimated model.

Least-Squares Fitting: Fitting requires a parametric model that relates the response data to the predictor data which depend on one or more coefficients. The result of the fitting process is an estimate of the model coefficients. To obtain the estimated coefficients, the least-squares method minimizes the cumulative square of residuals. The residual for the ith data point r_i is defined as the difference between the observed value y_i and the fitted value $\widehat{y_i}$, and it is identified as the error associated with the data, that is

$$r_i = y_i - \widehat{y_i}$$

The cumulative square of the residuals is given by

$$S = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2$$

where n is the number of data points included in the fit and S is the sum of the squared errors.

<u>Linear Least-Squares</u>: A linear model is defined as a model that is linear in the coefficients. For example polynomials are linear. To illustrate the linear least-squares fitting process, suppose you have n data points that can be modeled by a first-degree polynomial, that is

$$\widehat{y_i} = p_1 x_i + p_2$$

$$S = \sum_{i=1}^{n} (y_i - (p_1 x_i + p_2))^2$$

<u>Solution</u>: Objective function:

$$\min S = \sum_{i=1}^{N} \left(Y_i - (a_0 + a_1 x_i + a_2 {x_i}^2 + a_3 {x_i}^3 + a_4 {x_i}^4)\right)^2$$

where x_i and Y_i are the points of the observed response and are known. a_0, a_1, a_2, a_3, a_4 are the parameters of the polynomial and are variables of this optimization problem.

<u>Reformulation:</u>

$$\min S = \sum_{i=1}^{N} (t_i)^2$$

subject to:

$$Y_i - (a_0 + a_1 x_i + a_2 {x_i}^2 + a_3 {x_i}^3 + a_4 {x_i}^4) = t_i \qquad \forall i \in N$$

or

$$-t_i - a_0 - a_1 x_i - a_2 {x_i}^2 - a_3 {x_i}^3 - a_4 {x_i}^4 = -Y_i \qquad \forall i \in N$$

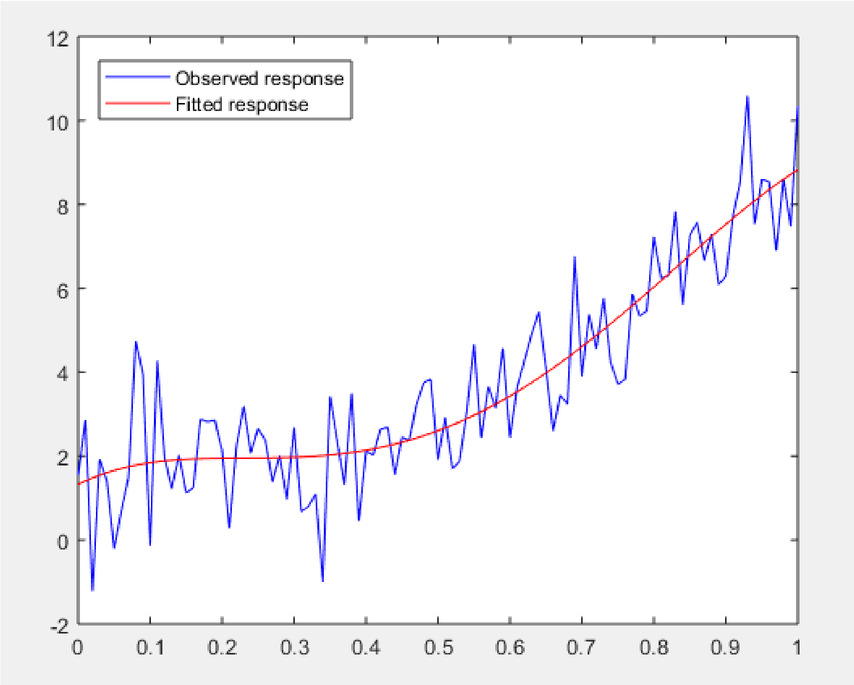
where t_i are new variables. There are N+5 variables and N constraints where N is the total number of points. The resulting formulation is a quadratic program (quadratic objective with linear constraints).

Matlab code:

```
clear all;
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%read the data
datall_struct=load('datall');
data_x=datall_struct.x';
data_y=datall_struct.y';
N=length(data_x); %length of the points
%variables: t1,t2....tN, +a0+a1+a2+a3+a4  (N=5)
%% *********************************************
%Set the objective function
%Hessian matrix
 H=zeros(N+5,N+5);
 for i=1:N
     H(i,i)=2;
 end
 f=zeros(1,N+5); %There are no linear terms
 %% *********************************************
 %Set the bounds
  lb=ones(1,N+5)*inf*-1;
  ub=ones(1,N+5)*inf;
 %% *********************************************
 %There are no inequality constraints
  A=[];
 b=[];
  %% *********************************************
 %equality constraints
  Aeq=zeros(N,N+5); %N constraints with N+5 variables
  for i=1:N
    Aeq(i,i)=-1;
    Aeq(i,N+1)= -1;
    Aeq(i,N+2)= -data_x(i);
    Aeq(i,N+3)= -data_x(i)^2;
    Aeq(i,N+4)= -data_x(i)^3;
    Aeq(i,N+5)= -data_x(i)^4;
  end
 for i=1:N
  beq(i)=-1*data_y(i);
 end
%call the solver
x = quadprog(H,f,A,b,Aeq,beq,lb,ub);
%values of the coefficients
a0_var=x(N+1);
a1_var=x(N+2);
a2_var=x(N+3);
a3_var=x(N+4);
a4_var=x(N+5);
%estimated output
for i=1:N
   Y_estimated(i)=a0_var+a1_var*data_x(i) +a2_var*data_x(i)^2  +a3_var*data_x(i)^3  +a4_var*data_x(i)^4 ;
end;
%plots
 figure(1)
plot(data_x,data_y,'b')
hold on;
plot(data_x,Y_estimated,'r')
legend('Observed response ','Fitted response ')
```

Solution:

a=[1.3257 8.438 -39.873 73.517 -34.581]



<u>Homework</u>: Consider a power system with the following eight committed units:

1 GT unit: C(P) = 710 + 60P + 0.37P^2 $€/\mathrm{h}$, $5 \leq P \leq 30\ MW$

3 ST units: C(P) = 670 + 40P + 0.4P^2 $€/\mathrm{h}$, $25 \leq P \leq 70\ MW$

2 ICE units: C(P) = 150 + 38P + 0.28P^2 $€/\mathrm{h}$, $7 \leq P \leq 28\ MW$

2 ST units: C(P) = 870 + 37P + 0.18P^2 $€/\mathrm{h}$, $50 \leq P \leq 140\ MW$

Find the produced power (P) of each generating unit to minimize the total operational cost of the system, while satisfying the power balance and the generation constraints. Solve the problem for load demand= 418 MW, and for load demand= 86.9 MW.