Nonlinear Programming in MATLAB

Zhibin Deng
Feb. 28, 2013
Contents

- Optimization Tool Box in MATLAB
  - Scope
  - GUI
- `fmincon` function for constrained nonlinear programming problems
  - Input arguments
  - Output interpretation
- CVX package for convex optimization problems
Optimization Tool Box in MATLAB

- Minimization (solving minimization problems)
  - linprog: linear programming problems
  - quadprog: quadratic programming problems
  - bintprog: binary integer programming problems
  - fminbnd: minimum of single-variable function
  - fseminf: minimum of semi-infinite constrained multivariable function
  - fmincon: minimum of constrained multivariable function
- ... 
- Equation solving
- Curve fitting
GUI for optimization tool box
- Type command “optimtool” in command window.

Problem setup
- Select solver and algorithm
- Specify objective function
- Specify constraints
- Specify options
- Run solver and check the output
\[ \min f(x) = \left(x_1^2 + x_2^2 - 1\right)^2 \]

s.t. \[-1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1\]
GUI for Optimization Tool Demo

- Select solver and algorithm
  - “fmincon”
  - “Active set”
- Specify objective function
  - “@(x) (x(1)^2+x(2)^2-1)^2”
  - Starting point “[1; 1]”
- Specify constraints
  - Aeq=[]; beq=[];
  - Lower=[-1; -1]; Upper=[1; 1];
- Run
Output

- Objective function value: $1.334051452011463\times 10^{-9}$

- $x(1) = -0.7070938676480343$
- $x(2) = -0.7070938676480343$

- `fmincon` stopped because the predicted change in the objective function is less than the default value of the function tolerance and constraints are satisfied to within the default value of the constraint tolerance.
Change options
- Set function tolerance to “$1e-10$”
- Rerun the problem
- Objective function value: $1.22662822906332e-14$

\[
\begin{align*}
  x(1) & : -0.7071067420293595 \\
  x(2) & : -0.7071067420293595
\end{align*}
\]

- `fmincon` stopped because the size of the current search direction is less than twice the default value of the step size tolerance.
fmincon solves the problems having the following form

\[
\begin{align*}
\text{min} & \quad f(x) \\
A \cdot x & \leq b \quad \text{linear inequalities} \\
A_{eq} \cdot x & = b_{eq} \quad \text{linear equalities} \\
\text{s.t.} & \quad l_b \leq x \leq u_b \quad \text{lower and upper bounds} \\
& \quad c(x) \leq 0 \quad \text{nonlinear inequalities} \\
& \quad c_{eq}(x) = 0 \quad \text{nonlinear equalities}
\end{align*}
\]
The syntax for fmincon

\[ x, fval, exitflag = \text{fmincon}(\text{objfun}, x_0, A, b, Aeq, beq, lb, ub, \nonlcon, \text{options}); \]

- \( x \): optimal solution; \( fval \): optimal value; \( exitflag \): exit condition
- \( \text{objfun} \): objective function (usually written in a separate M file)
- \( x_0 \): starting point (can be infeasible)
- \( A \): matrix for linear inequalities; \( b \): RHS vector for linear inequalities
- \( Aeq \): matrix for linear equalities; \( beq \): RHS vector for linear equalities
- \( lb \): lower bounds; \( ub \): upper bounds
- \( \nonlcon \): \( [c, ceq] = \text{constraintfunction}(x) \)
Construct Nonlinear Objective and Constraint Functions for fmincon

\[ \min \quad f(x) = (x_1^2 + x_2^2 - 1)^2 \]

\[ -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1, \]

s.t. \[ x_1 + x_2 \geq 1 \]

\[ x_1x_2 \geq \frac{1}{2}, x_2 \geq x_1^2, x_1 \geq x_2^2 \]

\[ A = [-1,-1]; b = -1; \]

\[ lb = [-1; -1]; ub = [1; 1]; \]

\[ c(x) = \begin{bmatrix} \frac{1}{2} - x_1x_2 \\ 2x_1 - x_2 \\ x_1^2 - x_2 \\ x_2^2 - x_1 \end{bmatrix}; ceq(x) = []; \]

% myobj.m
function f=myobj(x)
f = (x(1)^2+x(2)^2-1)^2;

% mycon.m
function [c, ceq]=mycon(x)
c=[1/2-x(1)*x(2);
\quad x(1)^2-x(2);
\quad x(2)^2-x(1)]; % nonlinear inequalities c(x) <= 0;
ceq=[]; % nonlinear equalities ceq(x) = 0;

% main file for fmincon
[x,fval] = fmincon(@myobj,x0,A,b,[],[],lb,ub, @mycon,options);
Construct objective functions with parameters

\[
\begin{align*}
\min & \quad f(x) = \left( x_1^2 + x_2^2 - a \right)^2 \\
-1 & \leq x_1 \leq 1, -1 \leq x_2 \leq 1, \\
s.t. & \quad x_1 + x_2 \geq 1 \\
& \quad x_1x_2 \geq \frac{1}{2}, x_2 \geq x_1^2, x_1 \geq x_2^2
\end{align*}
\]

% myobj.m
function f=myobj(x, a)
f = (x(1)^2+x(2)^2-a)^2;
% main file for fmincon
a = 1;
[x,fval] = fmincon(@(x) myobj(x,a),xo,A,b,[],[],lb,ub, @mycon,options);
Provide gradient information could accelerate the solver and improve the accuracy.

\[ f(x) = \left( x_1^2 + x_2^2 - a \right)^2 \Rightarrow \nabla f(x) = \left[ 4x_1 \left( x_1^2 + x_2^2 - a \right), \quad 4x_2 \left( x_1^2 + x_2^2 - a \right) \right] \]

\[ c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \\ c_3(x) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} - x_1x_2 \\ x_1^2 - x_2 \\ x_2^2 - x_1 \end{bmatrix} \Rightarrow \nabla c(x) = \begin{bmatrix} \frac{\partial c_1(x)}{\partial x_1} & \frac{\partial c_2(x)}{\partial x_1} & \frac{\partial c_3(x)}{\partial x_1} \\ \frac{\partial c_1(x)}{\partial x_2} & \frac{\partial c_2(x)}{\partial x_2} & \frac{\partial c_3(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -x_2 & 2x_1 & -1 \\ -x_1 & -1 & 2x_2 \end{bmatrix} \]
% objective function with gradient information
function [f,G]=myobj(x,a)
    f = (x(1)^2+x(2)^2-a)^2;
    % Gradient of objective function
    if nargout > 1
        G = [4*x(1)*(x(1)^2+x(2)^2-a),
             4*x(2)*(x(1)^2+x(2)^2-a)];
    end

% constraint function with gradient information
function [c, ceq, DC, DCeq]=mycon(x)
c=[1/2-x(1)*x(2);
   x(1)^2-x(2);
   x(2)^2-x(1)]; % nonlinear inequalities c(x)
ceq=[]; % nonlinear equalities ceq(x) = 0;
% gradient of contraint function
if nargout > 2
    DC=[-x(2), -x(1);
        2*x(1), -1;
        -1, 2*x(2)]';
    DCeq=[];
end
Results could be wrong
- Sometimes, `fmincon` find a local maximum instead of local minimum!
- Different algorithms or starting points could return different results.
- It’s unstable for non-differentiable objective or constraint functions.
- For NLP, `fmincon` does not guarantee to return the global minimum.
Better formulation for your problem
- Continuous and differentiable
- Convex

Different starting points

Different solvers and algorithms
- Select the solver appropriate for your problems
- Provide gradient or Hessian information if possible
Convex Optimization

\[
\min f(x) \\
s.t. \ g_i(x) \leq 0 \ \text{for} \ i = 1, \ldots, m.
\]

where \( f(x) \) and \( g_i(x) \) are convex functions.

- **Global** optimal solution is guaranteed by theory
- **Stable** algorithm for well-posed problems
CVX – Modeling Systems

- Less strict modeling syntax
  - What you saw is what you get
- Transform the problem into standard form (LP, SDP or SOCP) automatically
- Return the solver’s status (optimal, infeasible etc.)
- Transforms the solution back to original form
Constrained least square problem

\[
\min \| Ax - b \|
\]

s.t. \( x^T x \leq 1 \)

where

\[
A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\]
% CVX least square \( ||Ax-b|| \) demo
A = [1 2 3; 2 4 6; 7 8 9]; % matrix A;
b = [1; 1; 1]; % right-hand side vector b;

cvx_begin % start of CVX
    variable x(3); % declare variables
    minimize( norm(A*x-b) ); % declare objective function; note to use parentheses
    subject to % can be omitted;
        starting of constraints
            x'*x <= 1; % or norm(x)^2 <= 1
cvx_end % end of CVX
CVX – Correlation Matrix Verification

Given three random variables $A$, $B$ and $C$ with the correlation coefficients $\rho_{AB}$, $\rho_{AC}$ and $\rho_{BC}$, respectively. Suppose we know from some prior knowledge (e.g., empirical results of experiments) that $-0.2 \leq \rho_{AB} \leq -0.1$ and $0.4 \leq \rho_{BC} \leq 0.5$. What are the smallest and largest values that $\rho_{AC}$ can take?

Hint

The correlation coefficients are valid if and only if

$$\begin{bmatrix} 1 & \rho_{AB} & \rho_{AC} \\ \rho_{AB} & 1 & \rho_{BC} \\ \rho_{AC} & \rho_{BC} & 1 \end{bmatrix} \succeq 0$$
CVX – Correlation Matrix Verification

SDP formulation

The above problem can be formulated as following problem:

\[
\begin{align*}
\text{Min/Max} & \quad \rho_{AC} \\
\text{s.t.} & \quad -0.2 \leq \rho_{AB} \leq -0.1 \\
& \quad 0.4 \leq \rho_{BC} \leq 0.5 \\
& \quad \rho_{AA} = \rho_{BB} = \rho_{CC} = 1 \\
\begin{bmatrix}
\rho_{AA} & \rho_{AB} & \rho_{AC} \\
\rho_{AB} & \rho_{BB} & \rho_{BC} \\
\rho_{AC} & \rho_{BC} & \rho_{CC}
\end{bmatrix} & \in \mathcal{S}_+^3
\end{align*}
\]
% CVX correlation matrix verification

\begin{verbatim}
% CVX correlation matrix verification
cvx_begin

% set precision to be BEST
cvx_precision best;

% select solver as SeDuMi instead of SDPT3
cvx_solver sedumi;

% declare variable matrix rho
variable rho(3,3) symmetric;

% specifying objective function
minimize rho(3,1);

% start of constraints
subject to
    rho(1,2) <= -0.1;
    rho(1,2) >= -0.2;
    rho(2,3) <= 0.5;
    rho(2,3) >= 0.4;
    rho(1,1) == 1;  % note equal "==" not "="
    rho(2,2) == 1;
    rho(3,3) == 1;
    rho == semidefinite(3);  % matrix rho is positive

semidefinite

cvx_end
\end{verbatim}
The problem was proposed by Pierre de Fermat in 17th century. Given three points $a$, $b$ and $c$ on the $\mathbb{R}^2$ plane, find the point in the plane that minimizes the total distance to the three given points. The solution method was found by Torricelli, hence known as Torricelli point.

Figure: Torricelli Point Problem
CVX – Torricelli Point Problem

Hint

\[ t_1 \geq \| x - a \|_2 \iff \begin{bmatrix} x - a \\ t_1 \end{bmatrix} \in \mathcal{L}^3, \]

\[ t_2 \geq \| x - b \|_2 \iff \begin{bmatrix} x - b \\ t_2 \end{bmatrix} \in \mathcal{L}^3, \]

\[ t_3 \geq \| x - c \|_2 \iff \begin{bmatrix} x - c \\ t_3 \end{bmatrix} \in \mathcal{L}^3. \]

SOCP Formulation

\[ \begin{align*}
\text{Min} & \quad t_1 + t_2 + t_3 \\
\text{s.t.} & \quad \begin{bmatrix} x - a \\ t_1 \end{bmatrix} \in \mathcal{L}^3, \quad \begin{bmatrix} x - b \\ t_2 \end{bmatrix} \in \mathcal{L}^3, \quad \begin{bmatrix} x - c \\ t_3 \end{bmatrix} \in \mathcal{L}^3
\end{align*} \]
% CVX Torricelli Point Problem
a=[0;2]; b=[-1;0]; c=[3;0]; % location of three points

cvx_begin
  cvx_precision best;
  cvx_solver sedumi;
  variables t(3) x(2); % declare multiple variables
  minimize ( sum(t) );
  subject to
    {x-a, t(1)} <In> lorentz(2); % SOC constraint
    {x-b, t(2)} <In> lorentz(2); % note the dimension
    {x-c, t(3)} <In> lorentz(2);

cvx_end

%% One more straight forward formulation

cvx_begin
  cvx_precision best;
  cvx_solver sdpt3;
  variable x(2)
  minimize ( sum(norms( x*ones(1,3) - [a,b,c] )) );

cvx_end
END

THANK YOU