Control Lab
www.a-lab.ee

TALLINN UNIVERSITY OF TECHNOLOGY

ISS0031 Modeling and Identification

Lecture 3: Optimization. Convexity. Newton's Method. Least Squares.

Aleksei Tepljakov, Ph.D.

September 19, 2017

Part I: Optimization. Convexity

General Nonlinear Optimization Problem

The single objective optimization problem can be written as

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{subject to} \begin{cases} c_i(x) & = & 0, & i \in \mathscr{I}, \\ q_j(x) & \leqslant & 0, & j \in \mathscr{E}, \\ \lambda_k \leqslant \; x_k & \leqslant & \xi_k, & k \in \mathscr{K}, \end{cases} \tag{1}$$

where f, c_i and q_j are scalar-valued functions of the variables x, λ_k and ξ_k are scalar bounds, and $\mathscr{I}$,$\mathscr{E}$, and $\mathscr{K}$ are sets of indices.

The linear programming problem is therefore a particular case of the general optimization problem stated above.

Types of Optimization Problems

Based on the search variable set:

- *Continuous*: Solutions belong to an uncountably infinite set, e.g., $x \in \mathbb{R}^n$;
- *Discrete*: Solutions belong to a limited set, e.g., $x \in \{1, 2, 3\}$.

Based on constraints:

- *Unconstrained*: Such problems arise directly in many practical applications, where it is safe to disregard certain constraints;
- *Constrained*: Such problems arise from models that include specific constraints on the variables, e.g., bounds and nonlinear constraints;

To make a transition from an unconstrained problem to a constrained one it is also to modify the objective function by adding penalizing terms.

Numerical Optimization Algorithms

Optimization algorithms are iterative processes, which are started by choosing an initial estimate for the solution, and generate a series of improved estimates until a solution is found. The following properties ARE essential to any good optimization algorithm:

- *Accuracy*. The algorithm should be able to identify a solution with adequate precision without being sensitive to errors in the data or to rounding errors that occur due to limited computational capacity of the implementation.
- *Robustness*. The algorithm should perform well on a wide variety of problems for all reasonable choices of the initial estimates.
- *Efficiency*. The algorithm should not require too much computer time or memory storage.

These goals usually conflict, especially in case of complex optimization problems, so tradeoffs must be sought.

Convexity

Let $S \neq \varnothing, S \subset \mathbb{R}^n$ and $x_1, x_2 \in S$.

Definition 1. *The set* $[x_1, x_2] = \{x | x = \lambda x_1 + (1-\lambda)x_2,$ $0 \leq \lambda \leq 1\}$ *is called a* ***line segment*** *with the endpoints* x_1, x_2*.*

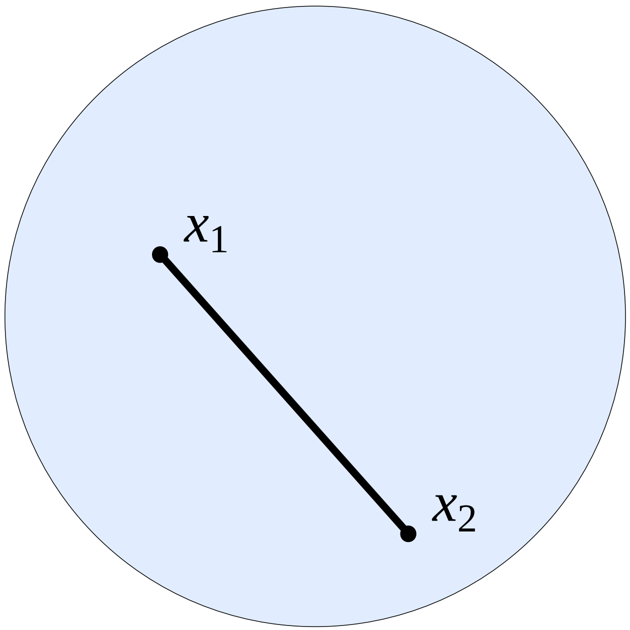
Definition 2. *The set* S *is called a* ***convex set*** *if the line segment joining any pair of points* $x_1, x_2 \in S$ *lies entirely in* S*.*

Definition 3. *The function* f *is called* ***convex*** *if its domain* D *is a convex set and for any two points* $x_1, x_2 \in D$ *there holds*

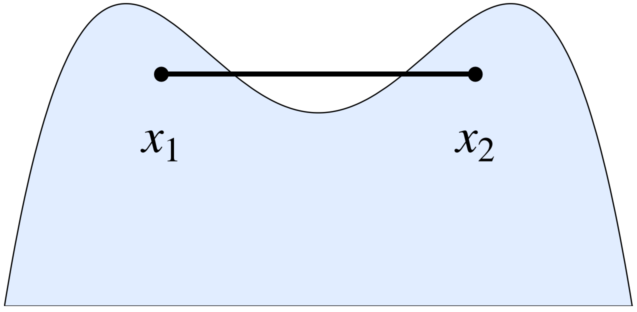
$$f(\lambda x_1 + (1-\lambda)x_2) \leqslant \lambda f(x_1) + (1-\lambda) f(x_2), \forall \lambda \in [0,1]. \quad (2)$$

(That is: The graph of f lies below the straight line connecting $(x_1, f(x_1))$ to $(x_2, f(x_2))$.)

(Non)convex Sets: Examples

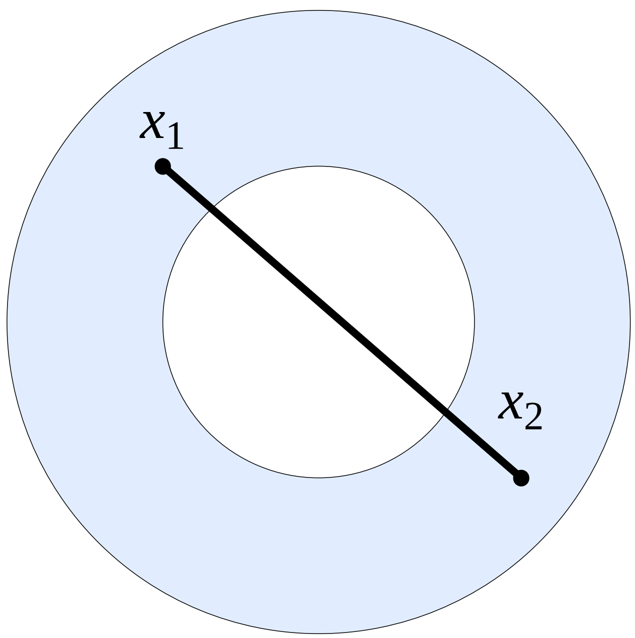


Convex

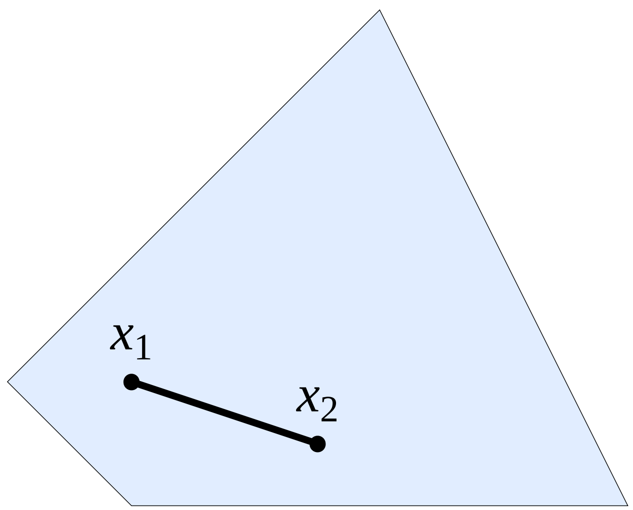


Nonconvex

(Non)convex Sets: Examples (continued)

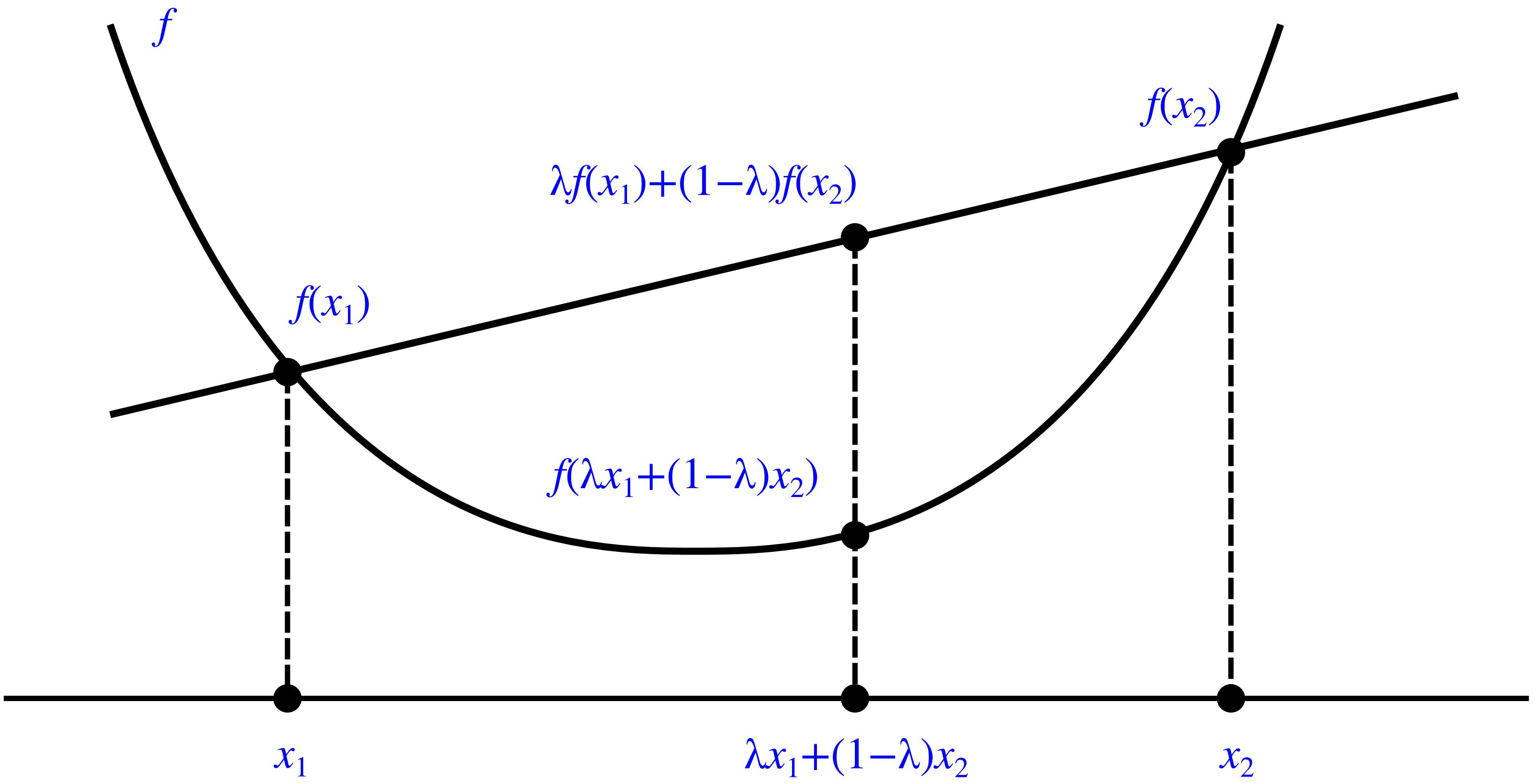


Nonconvex



Convex

Convex Function: Illustration



Convexity: Some Propositions

Proposition 1. *A solution set $\mathbb{L}$ for the linear inequality $a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n \leq b$ is a convex set.*

Proposition 2. *The intersection of any finite number of convex sets is also a convex set.*

Corollary 1. *The solution set of a system of linear equations (inequalities) is a convex set.*

Corollary 2. *The solution set of constraints for linear programming problem (set of feasible solutions) is a convex set.*

Definition 4. *Given a finite number of points $x_1, x_2, \ldots, x_n$ in a real vector space, a* ***convex combination*** *of these points is a point of the form $\alpha_1x_1 + \alpha_2x_2 + \cdots + \alpha_nx_n$, where $\alpha_i \in \mathbb{R}$, $\alpha_i \geq 0$ and $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$.*

Definition 5. *Let x be a convex combination of points from the set S. Then, S is called convex if $x \in S$.*

Convex Optimization Problem

Definition 6. *A point x^* is a **local minimizer** of the function f if there is a neighborhood $\mathcal{N}$ of x^*such that $f(x^*) \leqslant f(x)$ for all $x \in \mathcal{N}$.*

Definition 7. *A point x^* is a global minimizer of the function f if $f(x^*) \leqslant f(x)$ for all x.*

Definition 8. *A convex optimization problem is a problem where all of the constraints are convex functions, and the objective is a convex function.*

Theorem 1. *When f is convex, any local minimizer x^* is a global minimizer of f. If in addition f is differentiable, then any stationary point x^* is a global minimizer of f.*

Linear Programming as a Special Case of Convex Optimization Problem

The linear programming problem can be stated as follows:

$$z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \to \min$$

subject to the constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leqslant b_1 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leqslant b_m \end{aligned}$$

and $x_1 \geqslant 0, x_2 \geqslant 0, \ldots, x_n \geqslant 0$.

Theorem 2. *If a linear programming problem has a solution, then it must occur at a vertex, or corner point, of the feasible set S, associated with the problem. Furthermore, if the objective function z is optimized at two adjacent vertices of S, then it is optimized at every point on the line segment joining these two vertices, in which case there are infinitely many solutions to the problem.*

Part II: Some Practical Optimization Methods

Newton's Method

Consider a problem of locating the root of

$$f(x) = 0, \quad f : \mathbb{R} \to \mathbb{R}. \tag{3}$$

We denote by x_0 the initial estimate for the root, by x_k the current estimate, and by x_{k+1} the next estimate. Then, for the one-dimensional Newton's method we have

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{4}$$

This equation is derived by considering the Taylor series for the function f, i.e.

$$f(x+p) \approx f(x) + pf'(x). \tag{5}$$

Examples: Find the value of $x = \sqrt{23}$; Illustration for the method.

Newton's Method: Example

Consider the one-dimensional problem

$$f(x) = 7x^4 + 3x^3 + 2x^2 + 9x + 4 = 0.$$

Complete iteration of Newton's method:

k	x_k	$f(x_k)$	$\lvert x_k - x^*\rvert$
0	0	$4 \cdot 10^{0}$	$5 \cdot 10^{-1}$
1	−0.4444444444444444	$4 \cdot 10^{-1}$	$7 \cdot 10^{-2}$
2	−0.5063255748934088	$3 \cdot 10^{-2}$	$5 \cdot 10^{-3}$
3	−0.5110092428604380	$2 \cdot 10^{-4}$	$3 \cdot 10^{-5}$
4	−0.5110417864454134	$9 \cdot 10^{-9}$	$2 \cdot 10^{-9}$
5	−0.5110417880368663	0	0

Newton's Method: Convergence

If the sequence $\{x_k\}$ converges, then when x_k is sufficiently close to x^*, we have

$$x_{k+1} - x^* \approx \frac{1}{2}\left(\frac{f''(x^*)}{f'(x^*)}\right)(x_k - x^*)^2, \tag{6}$$

indicating that the error in x_k is approximately squared at every iteration. Consider the following theorem.

Theorem 3. *Assume that the function $f : \mathbb{R} \to \mathbb{R}$ has two continuous derivatives. Let x^* be a zero of f with $f'(x^*) \neq 0$. If $|x_0 - x^*|$ is sufficiently small, then the sequence defined by*

$$x_{k+1} = x_k - f(x_k)/f'(x_k) \tag{7}$$

converges quadratically to x^* with rate constant

$$C = |f''(x^*)/2f'(x^*)|. \tag{8}$$

Newton's Method for Systems of Nonlinear Equations

Suppose we have

$$F(x) = 0, \quad f : \mathbb{R}^n \to \mathbb{R}^n. \tag{9}$$

Then, formula for Newton's methon in n dimensions is

$$x_{k+1} = x - J^{-1}(x)F(x), \tag{10}$$

where $J(x)$ is the Jacobian of F, i.e.

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & & \frac{\partial F_2}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}. \tag{11}$$

Newton's Method for Systems of Nonlinear Equations: Convergence

- Using Newton's method does not guarantee convergence. Therefore, it is practical to set some limit on the number of iterations. If the limit is exceeded and no solution is found one may try to alter the initial estimate and repeat the process.
- For problems in n dimensions, convergence may be assessed by means of the L^2 norm. For a vector $F(x)$ we have

$$\|F(x)\| = \sqrt{F_1^2(x) + F_2^2(x) + \cdots + F_n^2(x)}. \tag{12}$$

Once the iteration satisfies

$$\|F(x)\| < \varepsilon, \tag{13}$$

where ε is a reasonably small scalar number, the algorithm shall stop returning the located vector x.

Newton's Method for Systems of Nonlinear Equations (Example)

Consider a system of nonlinear equations

$$\begin{aligned} F_1(x_1, x_2) &= \sin x_1 + x_2^2 = 0 \\ F_2(x_1, x_2) &= 3x_1 + 5x_2^2 = 0. \end{aligned}$$

The Jacobian is given by

$$J(x_1, x_2) = \begin{bmatrix} \cos x_1 & 2x_2 \\ 3 & 10x_2 \end{bmatrix}.$$

The formula for Newton's method in this case is

$$x_{k+1} = x - \begin{bmatrix} \cos x_1 & 2x_2 \\ 3 & 10x_2 \end{bmatrix}^{-1} \begin{bmatrix} \sin x_1 + x_2^2 \\ 3x_1 + 5x_2^2 \end{bmatrix}.$$

The task is to find $x^* = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathsf{T}$.

Some Issues of Newton's Method

The following list outlines some of the issues encountered with Newton's method.

- *Poor initial estimate*. This directly affects the convergence of the algorithm. If there is a large error in the initial estimate, the method may fail to converge, or converge slowly.
- *Encountering a stationary point*. If the method encounters a stationary point of f, where the derivative is zero, the method will terminate due to division by zero.
- *Overshoot*. If the derivative of f is not well behaved in the vicinity of a root, the method may overshoot, and diverge from the root.

Modifications of Newton's method exist that tackle these problems.

Linear Least Squares: Basics

Least squares methods solve problems of type

$$\min_x r_1^2 + \cdots + r_n^2 = \sum_{i=1}^{n} (y_i - f_i(x, t_i))^2, \quad (14)$$

where (t_i, y_i) is the *observed data points*, f_i is the *modeling function*, and r_i are called *residuals*.

A particularly useful application of least squares fitting exists for systems of equations

$$Ax = b. \quad (15)$$

If there exists no solution for this system, we can instead solve

$$A^{\mathsf{T}} A\hat{x} = A^{\mathsf{T}} b \quad (16)$$

to obtain an approximate solution that minimizes $E = \|Ax - b\|^2$.

Linear Least Squares: Example

Find the closest line to the points $(0,6)$, $(1,0)$, and $(2,0)$.

Solution: The line is given by $y = \alpha t + \beta$. We construct the system

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \alpha \\ \beta \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}}_{b}.$$

This system has no solution. It is obvious, that no straight line goes through these points. If, however, we consider solving

$$A^\mathsf{T} A \hat{x} = A^\mathsf{T} b$$

we have $\hat{\alpha} = -3$, $\hat{\beta} = 5$. The line equation is $y = -3t + 5$.

Questions?

Thank you for your attention!