Lecture 5

Finite-Dimensional Constrained Optimization

Randall Romero Aguilar, PhD

This draft: September 19, 2018

Universidad de Costa Rica
SP6534 - Economía Computacional

Table of contents

Introduction

Applications

Constrained optimization problems are ubiquitous in economics:

- Firm maximizes profit subject to resource constraints
- Firm minimizes cost of producing specified output
- Consumer maximizes utility subject to budget constraint

Definitions

- In the finite-dimensional constrained optimization problem, one is given a real-valued function f defined on $X \subset \Re^n$ and asked to find an $x^* \in X$ such that $f(x^*) \geq f(x)$ for all $x \in X$.
- We denote this problem

$$\max_{x \in X} f(x)$$

- We call f the objective function, X the feasible set, and x^*, if it exists, a maximum or optimum.
- We focus on maximization - to solve a minimization problem, simply maximize the negative of the objective.

We say that $x^* \in X$ is a ...

- maximum of f in X if $f(x^*) \geq f(x)$ for all $x \in X$.
- strict maximum of f in X if $f(x^*) > f(x)$ for all $x \in X$, $x \neq x^*$.
- local maximum of f in X if $f(x^*) \geq f(x)$ for all $x \in X$ in some neighborhood of x^*.
- strict local maximum of f in X if $f(x^*) > f(x)$ for all $x \in X$, $x \neq x^*$, in some neighborhood of x^*.

Weierstrass Extreme Value Theorem

- If f is continuous on a nonempty, closed, and bounded set X, then f attains a maximum in X.
- The following examples illustrate the role of the assumptions.
- The function $f(x) = x$ has no maximum on $X = \Re$: f is continuous and X is closed, but not bounded.
- The function $f(x) = x$ has no maximum on $X = [0, 1)$: f is continuous and X bounded, but not closed.
- The function

$$f(x) = \begin{cases} 1 - x & x \in (0, 1] \\ 0 & x = 0 \end{cases}$$

has no maximum on $X = [0, 1]$: X is closed and bounded, but f is not continuous.

Equality Constrained Optimization

Definition

The canonical equality-constrained optimization problem takes the form

$$\begin{array}{ll} \max & f(x) \\ \text{s.t.} & g(x) = c \end{array}$$

where $f : \Re^n \mapsto \Re$ and $g : \Re^n \mapsto \Re^m$ are continuously differentiable functions, f is concave, g is convex, and $c \in \Re^m$.

Theorem of Lagrange

- Theorem of Lagrange: A vector x^* maximizes $f(x)$ subject to $g(x) = c$ if, and only if, there is a vector $\lambda^* \in \Re^m$ such that (x^*, λ^*) maximizes the Lagrangian

$$L(x, \lambda) \equiv f(x) + \lambda'(c - g(x)).$$

- In particular, x^* and λ^* must simultaneously satisfy

$$0 = \frac{\partial L}{\partial x}(x^*, \lambda^*) = f'(x^*) - {\lambda^*}' g'(x^*)$$

$$0 = \frac{\partial L}{\partial \lambda}(x^*, \lambda^*) = c - g(x^*)$$

Envelope Theorem

- The λ_i^* are called shadow prices.
- The Envelope Theorem asserts that under mild assumptions,

$$\frac{\partial f^*}{\partial c_i} = \lambda_i^*,$$

where f^* is the optimal value of the objective.

Example 1:

Minimization problem

- Consider

$$\begin{aligned} \min \quad & 2 - x_1^2 - x_2^2 \\ \text{s.t.} \quad & x_1 + x_2 = k \end{aligned}$$

- The Lagrangian for this problem is

$$L(x_1, x_2, \lambda) = 2 - x_1^2 - x_2^2 + \lambda(k - x_1 - x_2).$$

- The first-order conditions are

$$0 = \frac{\partial L}{\partial x_1}(x_1, x_2, \lambda) = -2x_1 - \lambda$$

$$0 = \frac{\partial L}{\partial x_2}(x_1, x_2, \lambda) = -2x_2 - \lambda$$

$$0 = \frac{\partial L}{\partial \lambda}(x_1, x_2, \lambda) = k - x_1 - x_2.$$

- Solving these conditions yield

$$x_1 = x_2 = k/2 \text{ and } \lambda = -k$$

- Thus, the optimal value is

$$f^* = 2 - \left(\frac{k}{2}\right)^2 - \left(\frac{k}{2}\right)^2 = 2 - \frac{k^2}{2}.$$

- Note that

$$\frac{df^*}{dk} = -k = \lambda^*$$

as guaranteed by the Envelope Theorem.

Example 2:

Maximization problem

Consider

$$\begin{array}{ll} \max & -x_1^2 - 2x_2^2 - 2x_1x_2 + 18 \\ \text{s.t.} & x_1 - x_2 = 1 \end{array}$$

The Lagrangian for this problem is

$$L(x_1, x_2, \lambda) = -x_1^2 - 2x_2^2 - 2x_1x_2 + 18 + \lambda(1 - x_1 + x_2).$$

The first-order conditions are

$$\begin{aligned} \frac{\partial L}{\partial x_1}(x_1, x_2, \lambda) &= -2x_1 - 2x_2 - \lambda = 0 \\ \frac{\partial L}{\partial x_2}(x_1, x_2, \lambda) &= -2x_1 - 4x_2 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda}(x_1, x_2, \lambda) &= 1 - x_1 + x_2 = 0. \end{aligned}$$

Solving yields $x_1 = 0.6, x_2 = -0.4, \lambda = -0.4.$

Example 3:

A firm problem

- A firm produces a single output using two inputs according to the production function $q = x_1^\alpha x_2^{1-\alpha}$, where $0 < \alpha < 1$.
- The inputs may be bought at competitive wages w_1 and w_2.
- What is the minimum cost of producing output q?
- The firm's optimization problem is

$$\begin{aligned} \min \quad & w_1 x_1 + w_2 x_2 \\ \text{s.t.} \quad & x_1^\alpha x_2^{1-\alpha} = q. \end{aligned}$$

- The Lagrangian for this problem is

$$L(x_1, x_2, \lambda) = w_1 x_1 + w_2 x_2 + \lambda(q - x_1^{\alpha} x_2^{1-\alpha}).$$

- The first-order conditions are

$$0 = \frac{\partial L}{\partial x_1}(x_1, x_2, \lambda) = w_1 - \lambda \alpha x_1^{\alpha-1} x_2^{1-\alpha}$$

$$0 = \frac{\partial L}{\partial x_2}(x_1, x_2, \lambda) = w_2 - \lambda(1-\alpha) x_1^{\alpha} x_2^{-\alpha}$$

$$0 = \frac{\partial L}{\partial \lambda}(x_1, x_2, \lambda) = q - x_1^{\alpha} x_2^{1-\alpha}.$$

- From the first two conditions, we obtain

$$\frac{w_1}{w_2} = \frac{\alpha x_2}{(1-\alpha)x_1}$$

- This implies

$$x_2 = \frac{1-\alpha}{\alpha}\frac{w_1}{w_2}x_1$$

- Substituting into production constraint and solving yields

$$x_1 = q\left(\tfrac{w_2\alpha}{w_1(1-\alpha)}\right)^{1-\alpha}$$

$$x_2 = q\left(\tfrac{w_2\alpha}{w_1(1-\alpha)}\right)^{-\alpha}$$

- After additional algebraic manipulations

$$\lambda = \left(\frac{w_1}{\alpha}\right)^{\alpha} \left(\frac{w_2}{1-\alpha}\right)^{1-\alpha}$$

- The shadow price is the firm's marginal cost of production.

General Constrained Optimization

The most general constrained finite-dimensional optimization problem that we consider takes the form

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq b \\ & x \geq 0 \end{aligned}$$

where $f : \Re^n \mapsto \Re$ and $g : \Re^n \mapsto \Re^m$ are continuously differentiable, f is concave, g is convex, and $b \in \Re^m$.

Think of the optimization problem as follows:

- There are n economic activities.
- The level of activity j is denoted x_j.
- Activity level x_j is inherently nonnegative.
- $f(x)$ is benefit received from activities x.
- Activities require use of m resources.
- An amount $g_i(x)$ of resource i is required to sustain activity x.
- A limited amount b_i of resource i available.
- Optimizer seeks activity vector $x \geq 0$ that maximize benefit $f(x)$ subject to resource availability $g(x) \leq b$.

- Karush-Kuhn-Tucker Theorem: A vector x maximizes $f(x)$ subject to $g(x) \leq b$ and $x \geq 0$ if, and only if, there is a vector $\lambda \in \Re^m$ such that for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$

$$x_j \geq 0 \perp f'_j(x) - \sum_i \lambda_i g'_{ij}(x) \leq 0$$

$$\lambda_i \geq 0 \perp g_i(x) \leq b_i.$$

where $f'_j \equiv \dfrac{\partial f}{\partial x_j}$ and $g'_{ij} \equiv \dfrac{\partial g_i}{\partial x_j}$.

- Here, "$\perp$" indicates complementarity: both inequalities must hold, and at least one must hold as a strict equality.
- If f is strictly concave, g is convex, and x and λ satisfy these conditions, then x is unique.

Consider the problem $\max f(x)$ subject to $a \leq x \leq b$:

$$L = f(x) + \lambda(x-a) + \mu(b-x) \qquad \Rightarrow$$

$$f'(x) + \lambda - \mu = 0$$

$$\lambda \geq 0 \qquad x - a \geq 0 \qquad \lambda(x-a) = 0$$

$$\mu \geq 0 \qquad b - x \geq 0 \qquad \mu(b-x) = 0$$

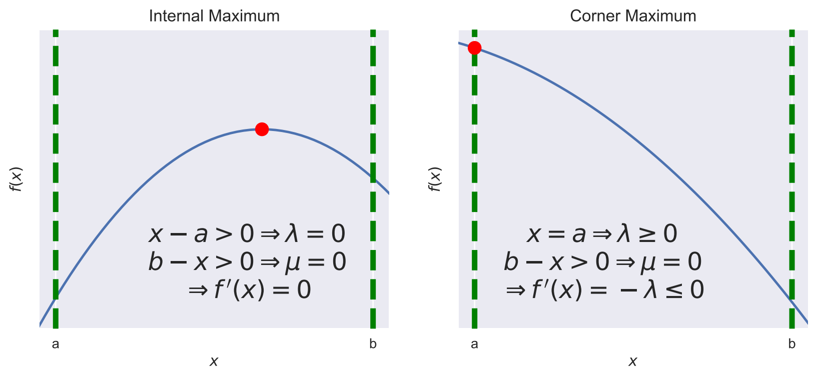

Figure 1: Bound Constrained Optimization

- The λ_i are called shadow prices.
- The Envelope Theorem asserts that under mild assumptions,

$$\frac{\partial f^*}{\partial b_i} = \lambda_i,$$

where f^* is the optimal value of the objective.
- Thus, λ_i is the implicit marginal cost of resource i and

$$MB_j(x) = f_j'(x) - \sum_i \lambda_i g_{ij}'(x)$$

is the net marginal economic benefit of activity j, which equals the explicit marginal benefit of activity j less the implicit marginal cost of resources required for activity j.

- The K-K-T complementarity conditions typically admit an arbitrage interpretation in economic and finance applications:

$x_j \geq 0$	activity levels are nonnegative
$MB_j \leq 0$	otherwise, raise benefit by raising x_j
$x_j > 0 \Rightarrow MB_j \geq 0$	otherwise, raise benefit by lowering x_j
$MB_j < 0 \Rightarrow x_j = 0$	avoid unbeneficial activities
$\lambda_i \geq 0$	shadow price of resource is nonnegative
$g_i(x) \leq b_i$	resource use cannot exceed availability
$\lambda_i > 0 \Rightarrow g_i(x) = b_i$	valuable resources should not be wasted
$g_i(x) < b_i \Rightarrow \lambda_i = 0$	surplus resources have no value

Example 4:

A firm in two markets

- A firm can sell a fixed quantity q in two distinct markets with inverse demand curves

$$p_i = \alpha_i - \frac{\beta_i}{2} q_i$$

where q_i is quantity sold and p_i is price in market i.

- How much should it sell in each market to maximize revenue?
- The firm's optimization problem is

$$\begin{array}{ll} \max & \alpha_1 q_1 - \frac{\beta_1}{2} q_1^2 + \alpha_2 q_2 - \frac{\beta_2}{2} q_2^2 \\ \text{s.t.} & q_1 + q_2 \leq q \\ & q_1 \geq 0, q_2 \geq 0 \end{array}$$

- The K-K-T conditions for this problem are

$$q_1 \geq 0 \perp \alpha_1 - \beta_1 q_1 - \lambda \leq 0$$

$$q_2 \geq 0 \perp \alpha_2 - \beta_2 q_2 - \lambda \leq 0$$

$$\lambda \geq 0 \perp q_1 + q_2 \leq q.$$

- Objective concave, constraint linear, so K-K-T conditions are necessary and sufficient.
- Answer:

$$q_i = \frac{\alpha_i - \alpha_j + q\beta_j}{\beta_1 + \beta_2}, \quad i \neq j.$$

  provided $q \geq \max\{\frac{\alpha_1 - \alpha_2}{\beta_1}, \frac{\alpha_2 - \alpha_1}{\beta_2}\}$.

The scipy.optimize.minimize function

- Algorithms for solving constrained optimization problems can be quite involved, so we will not discuss them in this course.
- We will, however, illustrate how to use `scipy.optimize` module function `minimize`.
- `scipy.optimize.minimize` solves the canonical constrained minimization problem:

$$\min f(x) \quad \text{subject to}$$
$$g_i(x) \geq 0, \qquad i = 1, \ldots, m$$
$$h_j(x) = 0, \qquad j = 1, \ldots, p$$
$$a \leq x \leq b$$

where $x \in \Re^n$.

minimize: calling protocol

```
minimize(fun, #objective function
         x0, #n-vector initial guess
         args=(), #extra arguments for function
         method=None, #type of solver
         bounds=None, #bounds for variables
         constraints=()) #constraints
```

Constraints are passed as a tuple of dictionaries:

```
cons = ({'type': 'eq', 'fun': h1}, ...
        {'type': 'eq', 'fun': hp},
        {'type': 'ineq', 'fun': g1}, ...
        {'type': 'ineq', 'fun': gm})
```

while bounds are passed as a tuple of lower-upper pairs:

```
bnds = ((a1, b1), ..., (an, bn))
```

minimize: output

Output: an object with these attributes (among others)

x	the solution of the optimization
fun	value of objective function
message	description of the cause of the termination
nfev	number of function evaluations
nit	number of iteration by the optimizer
success	True if solution found

Example 5:

Using scipy.optimize.minimize

To solve

$$\begin{aligned} \max \quad & -x_0^2 - (x_1 - 1)^2 - 3x_0 + 1 \\ \text{s.t.} \quad & 4x_0 + x_1 \leq 0.5 \\ & x_0^2 + x_1 \leq 2.0 \\ & x_0 \geq 0, x_1 \geq 0 \end{aligned}$$

starting from guess $(x_0, x_1) = (0, 1)$ execute the script

```
from scipy.optimize import minimize
def f(x):
    return x[0]**2 + (x[1]-1)**2 + 3*x[0] - 2

cons = ({'type': 'ineq',
         'fun': lambda x: 0.5 - 4*x[0] - x[1]},
        {'type': 'ineq',
         'fun': lambda x: 2.0 - x[0]**2 - x[0]*x[1]})

bnds = ((0, None), (0, None))
res = minimize(f,  [0.0, 1.0], method='SLSQP',
               bounds=bnds, constraints=cons)
```

This should produce

```
fun: 1.0078716929461423e-09
message: 'Optimization terminated successfully.'
nfev: 148
nit: 79
status: 0
success: True
x: array([1.    , 1.0001])
```