Introduction and Lecture 1: Optimization on paper

Antoine Chapel

Introduction

- Hello to all
- Antoine Chapel: `antoine.chapel@sciencespo.fr`
- A few words about math+econ+code
- Combining code and slides
- What about you ?

Introduction

Outline of the two-days session

- Optimization with pen and computer
- Dynamic Programming
- Numerical methods
- Maximum likelihood and logit
- Advanced econometrics: BLP and Rust

Today: Optimization with mostly pen and a bit of computer

- Weierstrass Theorem
- Unconstrained Optimization
- Constrained Optimization
- Introduction to SymPy
- Duality
- Linear Programming

The optimization problem

$$\begin{aligned} \max_{x \in A} \quad & f(x) \\ \text{s.t.} \quad & g(x) = d, \\ & h(x) \leq e \end{aligned}$$

Goal: find x^* such that $f(x^*) \geq f(x)$, $\forall x \in A$ that satisfies the constraints.

Be able to move easily from $\max$ to $\min$: $\max f(x) = \min -f(x)$.

The Weierstrass Theorem

Can we ever be sure that a solution to the optimization problem exists ? Yes.

Weierstrass Theorem:

- Let $D \in \mathbf{R}^n$ be a compact space
- Let $f : D \to \mathbf{R}$ be a continuous function on D

$\Rightarrow$ then, f attains a $\max$ and a $\min$ on D.

Heine Borel theorem: A subset $S \subset \mathbf{R}^n$ is compact if it is closed and bounded

Weierstrass theorem checklist

1. Is my function continuous ?
2. Is the set over which I am optimizing closed ?
3. Can i fit this set inside an open ball ?

If you can answer positively to the three questions, there exists a $\min$ and a $\max$ to your optimization problem.

Careful: these conditions are sufficient, but not necessary. If you cannot answer positively, that does not mean you cannot solve the problem.

Unconstrained Optimization: FOC

$$\max_{x \in \mathbf{R}^n} \quad f(x)$$

$$\max_{(x,y) \in \mathbf{R}^2} \quad f(x,y)$$

First order conditions: equate the gradient to zero.

$$\nabla f = 0$$

$$\Big(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\Big) = (0,0)$$

May yield several candidates (x^c, y^c), and we cannot know if they are local, global, min, max, or saddle points.

Unconstrained Optimization: SOC

Second order conditions: do not be a robot (will prove useful when we do dynamic programming).

If convexity/concavity is not trivial: compute the hessian.

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{pmatrix}$$

If $(\nabla^2 f)\big|_{(x^c,y^c)}$ is positive (negative) definite, then (x^c, y^c) is a strict local minimizer (maximizer).

How to check positive definiteness ? Method 1: If all eigenvalues of A are positive, A is positive definite. Method 2: If the determinants of leading principal minors of A are positive, A is positive definite.

Equality Constrained Optimization

$$\begin{aligned} \max_{x \in \mathbf{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = d \end{aligned}$$

$$\mathcal{L}(x, \lambda) = f(x) + \lambda[d - g(x)]$$

That, you know already. What happens under the hood ?

$$\begin{aligned} & \max_x f(x) + \min_\lambda \lambda(d - g(x)) \\ = & \max_x \min_\lambda f(x) + \lambda[d - g(x)] \end{aligned}$$

We will go into more details when we consider duality.

Equality Constrained Optimization: FOC and SOC

After forming the lagrangian $\mathcal{L}$: FOC:

$$\nabla \mathcal{L} = 0$$

SOC:

$$\nabla^2 \mathcal{L} = \begin{pmatrix} \frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial \lambda^2} & \frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial x \partial \lambda} \\ \frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial \lambda \partial x} & \frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial x^2} \end{pmatrix}$$

Suppose the problem is n-dimensional with m constraints: $\nabla^2 \mathcal{L}$ is a matrix of four blocks:

- $\frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial \lambda^2}$: $(m \times m)$ $\frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial \lambda \partial x}$: $(1 \times n)$
- $\frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial x \partial \lambda}$: $(n \times 1)$ $\frac{\partial^2 \mathcal{L}(x,\lambda)}{\partial \lambda \partial x}$: $(n \times n)$

Equality Constrained Optimization: SOC

SOC: We have n dimensions and m constraints. Take the $(n-m)$ largest leading principal minors of the Hessian. Evaluate these leading principal minors (lpm) at candidate points, starting from the lpm that is smallest in size.

- If their signs alternate, starting from the sign of $(-1)^{m+1}$, the candidate is a local maximizer
- If they all have sign $(-1)^m$, the candidate is a local minimizer

MATH BREAK: SYMPY

Let us apply what we have just seen in Python, without even explicitly using numerical methods.

Linear Programming

Linear Programming is a class of optimization problems that possesses useful properties.

$$\begin{aligned} \max_{x \in \mathbf{R}^n} \quad & x'c \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

For example:

$$\begin{aligned} \max_{x \in \mathbf{R}^2} \quad & 4x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 2, \\ & 3x_1 + x_2 \leq 3 \end{aligned}$$

The objective function and the constraints are linear. Constraints can be equality or inequality. Notice that any equality constraint can be rewritten as two inequality constraints.

A graphical approach:

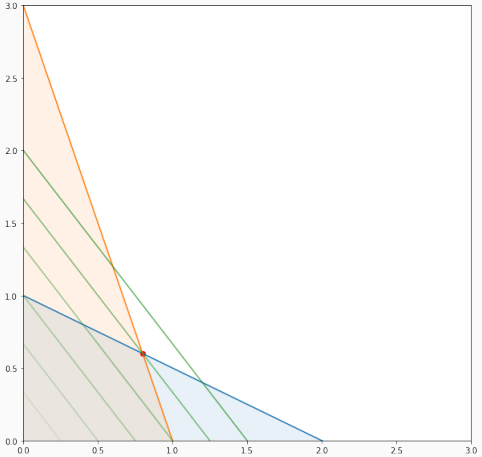

Figure 1: Linear Programming

Linear Programming

What makes linear programming appealing ? Convexity, possibility to enter all parameters of the problem as one vector and one matrix. Most importantly, dedicated algorithms and solvers that allow us to find a fast solution to (very) large problems.

In math+econ+code, you will rely on a commercial, high-performance solver: Gurobi. Today, we will solve a very low-dimensional problem, using the free scipy solver.

Duality

Every optimization problem can be studied from two perspectives: the primal and the dual.

For convex optimization problems (optimization of convex functions over convex sets), if a solution exists, it coincides for the primal and the dual.

If your primal is nonconvex, your dual **is** convex. The solutions do not coincide, but the dual is still informative, as it provides a bound on the value of the primal.

DUALITY

Let us assume the following problem is convex. For example, it could be linear programming. We denote the value of the objective function at the solution V_P.

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, ..., m, \\ & h_i(x) = 0, \quad i = 1, ..., p \end{aligned}$$

The control variable is x, the Lagrangian variables are λ and ν. Rewriting this problem with penalization gives:

$$\min_x f_0(x) + \max_\lambda \sum_{i=1}^{m} \lambda_i f_i(x) + \max_\nu \sum_{i=1}^{p} \nu_i h_i(x)$$

$$\max_\nu \nu \cdot h_i(x) = \begin{cases} +\infty & \text{if } h_i(x) > 0 \quad (\nu^* = +\infty) \\ 0 & \text{if } h_i(x) = 0 \quad (\nu^* = \text{ any } \nu)) \\ +\infty & \text{if } h_i(x) < 0 \quad (\nu^* = -\infty) \end{cases}$$

DUALITY

We write the Lagrangian:

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

So, the primal problem is:

$$V_p = \min_x \max_{(\lambda,\nu)} \mathcal{L}(x, \lambda, \nu)$$

The **Lagrange Dual** is defined as:

$$V_d = \max_{(\lambda,\nu)} \min_x \mathcal{L}(x, \lambda, \nu)$$

Where λ and ν are the control variables, and x is the ”lagrangian” variable.

Duality: Linear Programming example

$$\begin{aligned} \max_{x} \quad & 4x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 - 2 = 0, \\ & 3x_1 + x_2 - 3 = 0 \end{aligned}$$

$$\mathcal{L}(x, \lambda) = 4x_1 + 3x_2 + \lambda_1(2 - x_1 - 2x_2) + \lambda_2(3 - 3x_1 - x_2)$$

Duality: Linear Programming example

$$
\begin{aligned}
V_d &= \min_{\lambda} \max_{x} 4x_1 + 3x_2 + \lambda_1(2 - x_1 - 2x_2) + \lambda_2(3 - 3x_1 - x_2) \\
&= \min_{\lambda} 2\lambda_1 + 3\lambda_2 + \max x_1(4 - \lambda_1 - 3\lambda_2) + x_2(3 - 2\lambda_1 - \lambda_2)
\end{aligned}
$$

Now "think" of λ as the control variable, and x the lagrangian variable.

$$
\begin{aligned}
\min_{\lambda} \quad & 2\lambda_1 + 3\lambda_2 \\
\text{s.t.} \quad & \lambda_1 + 3\lambda_2 = 4, \\
& \lambda_1 + \lambda_2 = 3
\end{aligned}
$$

We have our dual problem. Let us take this to Python and check that the values of V_p and V_d coincide.