Programming for Economists

Exercise Class 7
Problem Set 3

Jonas Theodor Ø. Schmidt

UNIVERSITY OF COPENHAGEN

Introduction to Optimization

- Objective: Find x find that minimizes $f(x)$

- We are familiar with solving optimization problems by applying analytical methods, utilizing first and second order conditions

- Now, we will learn how to apply numerical methods

- The standard approach is to use minimize. Should our aim be maximize, this can be achieved by minimizing the negative of the function

Grid-Search

- Method: Explore various values of x to find one that minimizes $f(x)$

- Advantages:
  - Provides a basic idea of the function's behavior
  - Avoids getting trapped in non-global minima

- Disadvantages:
  - High computational demand, particularly with increasing dimensions
  - Limited to evaluating points within the predefined grid
  - Precision of the solution is directly tied to the fineness of the grid

Solver

- Method: Iteratively searches for the minimum of $f(x)$ by exploring various values of x and refining guesses according to $f(x)$'s evaluation

- Mechanics: Algorithm dictates which x values to test, except for the initial guess which is user-defined

- Advantages:
  - More efficient and less resource-intensive compared to grid search
  - Yields a more accurate solution

- Disadvantages:
  - Outcome may be influenced by the choice of the starting point
  - Risk of not reaching any solution

Constrained optimization

- Approaches
  - Utilize grid search or specific solvers designed for constrained problems
  - Alternatively, modify the objective function to include penalties for constraint violations to use unconstrained optimization

- Advantages of Penalty Methods
  - Directs the solver back within bounds when it strays outside constraints

- Drawbacks of Penalty Methods
  - Potential to create additional local minima, further complicating the search for the global minima

Solvers

- Choice of solver depends on problem specifics: constrained or unconstrained

- Unconstrained Optimization
  - BFGS: Fast with gradients/Hessians. Ideal for efficient, derivative-informed problems
  - Nelder-Mead: Robust, suitable for derivative-less, complex problems. Slower but reliable

- Constrained Optimization
  - SLSQP: Quick, effective with gradient/Hessian info. Good for problems with direct constraints
  - Penalty Methods: Combine unconstrained solvers with penalties for constraint violations. Versatile for complex constraints

Break

Questions & comments?