Nonlinear Optimization: Introduction

INSEAD, Spring 2006

Jean-Philippe Vert

Ecole des Mines de Paris

`Jean-Philippe.Vert@mines.org`

Optimization problem

$$\begin{array}{rl} \textit{minimize} & f(x) \\ \textit{subject to} & x \in \mathcal{X} \end{array}$$

- x: a *decision variable*
- $\mathcal{X}$: the *constraint set*, i.e., available decisions
- $f : \mathcal{X} \to \mathbb{R}$: the *cost* or *objective* function

We want to find an *optimal decision*, i.e., an $x^* \in \mathcal{X}$ such that:

$$f(x^*) \leq f(x), \quad \forall x \in \mathcal{X}.$$

Where do optimization problems arise?

- *Economics*: Consumer theory / supplier theory
- *Finance*: Optimal hedging / pricing
- *Statistics*: data fitting, regression, pattern recognition
- *Science / Engineering*: Aerospace, product design, data mining
- *Other Business decisions*: scheduling, production, organizational decisions
- *Government*: Military applications, fund allocation, etc
- *Other Personal decisions*: Sports, on-field decisions, player acquisition, marketing

Example

- *Portfolio optimization*
  - variables: amount invested in different assets
  - constraint: budget, max/min investment per asset, minimum return
  - objective: overall risk, or return variance
- *Device sizing in electronic circuits*
  - variables: device width and lengths
  - constraints: manufacturing limits, max area
  - power consumption
- *Data fitting*
  - variable: model parameters
  - contraints: prior information, parameter limits
  - objective: measure of misfit, prediction error

Classification of optimization problems

- *$\mathcal{X}$ is discrete/finite*, e.g. *integer programming* (scheduling, route planning)
- *$\mathcal{X} \subset \mathbb{R}^d$ is continuous*
  - *Linear programming (LP)*: f is linear and $\mathcal{X}$ is a polyhedron specified by linear equalities and inequalities.
  - *Quadratic programming (QP)*: f is convex quadratic and $\mathcal{X}$ is a polyhedron specified by linear equalities and inequalities.
  - *Convex programming*: f is a convex function and X is a convex set.
  - *Nonlinear programming*: f is nonlinear, and/or $\mathcal{X}$ is specified by nonlinear equalities and inequalities.

Solving optimization problems

- General optimization problem
  - *very difficult* to solve
  - methods involve some *compromise*, e.g., very long computation time, or not always finding the solution
- *Exceptions*: certain problem classes can be solved efficiently and reliably
  - least-square problems
  - linear programming problems
  - convex optimization problems

Least squares

$$\textit{minimize} \quad \| Ax - b \|_2^2$$

- analytical solution $x^* = \left(A^\top A\right)^{-1} A^\top b$
- reliable and efficient algorithms and softwares
- a mature technology
- least-square problems are easy to recognize

Linear programming

$$\begin{aligned} \textit{minimize} \quad & c^\top x \\ \textit{subject to} \quad & a_i^\top x \leq b_i\,, \quad i = 1, \ldots, m \end{aligned}$$

- no analytical formula for solution
- reliable and efficient algorithms and software (simplex algorithm, interior-point methods)
- a mature technology
- not as easy to recognize as least-square problems
- a few standard tricks to convert problems into linear programs

Convex optimization

$$\begin{array}{rl} \textit{minimize} & f(x) \\ \textit{subject to} & g_i(x) \leq b_i\,, \quad i = 1, \ldots, m \end{array}$$

- f and g_i are convex:

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y) \quad \text{if} \alpha, \beta \geq 0\,, \alpha + \beta = 1\,.$$

- include least-squares and linear programming
- no analytical solution
- reliable and efficient algorithms
- almost a technology
- often difficult to recognize

Brief history

- 1947: simplex algorithm for linear programming (Dantzig)
- 1960s: early interior-point methods (Fiacco & McCormick, Dikin, ...)
- 1970s: ellipsoid method and other subgradient methods
- 1980s: polynomial-time interior-point methods for linear programming (Karmarkar 1984)
- late 1980s-now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)

The optimization process (1/2)

- *Formulate* real life problems into mathematical models
  - Study the environment and clearly understand the problem
  - Formulate the problem using verbal description
  - Define notations for parameters and decision variables
  - Construct a model using mathematical expressions
  - Collect necessary data; Transform the raw data to parameter values
- *Implement* the model and solution algorithms using a computer : analyze the models and develop efficient procedures to obtain best solutions

The optimization process (2/2)

- *Interpret* computer solutions and perform sensitivity analysis
- *Implementation*: put the knowledge gained from the solution to work
- *Monitor* the validity and effectiveness of the model and *update* it when necessary

What you will learn

- *Models* - the art: How we choose to represent real problems
- *Theory* - the science: What we know about different classes of models; e.g. necessary and sufficient conditions for optimality
- *Algorithms* - the tools: How we apply the theory to robustly and efficiently solve powerful models

Course outline

- Modeling and reformulating
- Optimality conditions
- Duality theory and sensitivity analysis
- Algorithms for unconstrained problems
- Algorithms for linearly constrained problems
- Algorithms for convex problems
- Applications

Practical informations

- Course web page:

  http://cbio.ensmp.fr/˜vert/teaching/2006insead

  - schedule
  - slides
  - references, links..
- $10 \times 2h$
- Grading: weekly homework (50%), article presentation (10%), final exam (40%)