310.

Lecture 0: Introduction

0.1. Formulation

variant of "Math Programming"

0.1.1 ⊛ Formulation of Nonlinear Programming

- Numerical Optimization

⊛ Sometimes called "continuous constrained optimization".

(NLP)

$$\min_x f(x) \quad (= \max -f(x)) \rightarrow \text{"objective"}$$

$$\left.\begin{aligned} \text{s.t.} \quad & c_i(x) = 0 \; ; \; i \in \mathcal{E} \\ & c_i(x) = 0 \; ; \; i \in \mathcal{I} \end{aligned}\right\} \rightarrow \text{"constraints"}$$

We Assume: ① $f: \mathbb{R}^n \to \mathbb{R}$ and $c_i: \mathbb{R}^n \to \mathbb{R}; i \in \mathcal{E} \cup \mathcal{I}$ are twice continuously differentiable

② $f(x), \{c_i(x)\}, \nabla_x f(x), \{\nabla_x c_i(x)\}_{i \in \mathcal{E} \cup \mathcal{I}}$ are computable and sometimes even $\nabla^2_{xx} f(x)$ and $\{\nabla^2_{xx} c_i(x)\}_{i \in \mathcal{E} \cup \mathcal{I}}$

Minai Anitesou Notes

0.1.2 Abreviation : NLP

Why study it? It either represents or approximates distinguished application problems.

Mihai Anitescu Notes

0. Introduction

0.2. Example Problems

0.2.1 A. Exactly Representable

1.) Maximum likelihood Problems

$$\min_{\theta} -L(\{x_i\}|\theta) = \left(\min -\log(L)\right)$$

or

Here $L(\{x_i\}|\theta) = f(x_1, \ldots x_n|\theta) \overset{\text{if i.i.d}}{=} \prod_{i=1}^{N_0} f(x_i|\theta)$

2) Constrained Max Likelihood Problems

$$V = \theta_n I + \sum_{i=1}^{n-1} \theta_i Z_i Z_i'$$

Z_i is an $n \times q_i$ matrix

X: fixed ~~set of~~ matrix of predictors

$$L(\theta, \alpha; y) = -\frac{1}{2} \log \det(V) - \frac{1}{2}(y - X\alpha)' V^{-1} (y - X\alpha)$$

Mihai Anitescu Notes

0.2.2 $\max_{\theta, \alpha \in \mathbb{R}^n} L(\theta, \alpha; y)$

$\theta_i \geq 0 \; ; \; i = 1, 2 .. m$

Observation: almost certainly $\theta_m > 0$ at solution.

$\log(\det(V))$ acts as a "barrier" for θ_m

We will see this again in interior-point methods

3) Transportation problems: Example 1.3 in book → and many other management science problems.

4) Nonlinear least squares (if time)

$$\min_\theta f(\theta) = \sum_{i=1}^{n} \{ y_i - g(x_i, \theta) \}^2$$

Mihai Anitescu Notes

0.2.3 B. Converted to NLP by approximation

4.)

$$\min_{u(t)\in\mathcal{U}} = \int_0^T f(t, x(t), u(t))\,dt$$

$$\dot{x}(t) = g(t, x(t), u(t)) \;;\; x(0) = x_0$$

and path constraints

e.g. $g_i(x(t)) \le 0 \;;\; i = 1, 2 \ldots$

or $x_j(t) \in [a_j, b_j]$, etc

$\mathcal{U}$: admissible controls

e.g.: $\mathcal{U} : \{ u : [0,T] \to \mathbb{R}^m \mid \text{measurable} \}$

$\mathcal{U} : \{ u : [0,T] \to \bigtimes_{i=1}^{m} [a_i, b_i] ; \text{measurable} \}$

Discretize

$$\min_{\{u(t_i)\}_{i=1}^{N}} \sum_{i=1}^{N} f(t_i, x(t_i), u(t_i))$$

$$x(t_n) = x(t_{n-1}) + \Delta t\, g(t_{n-1}, \tilde{x}(t_n), u(t_n))$$

$$x_j(t_n) \in [a_j, b_j]$$

Mihai Anitescu Notes

02.4 Here $\tilde{x}(t_n) = x(t_{n-1})$: Forward Euler

$\tilde{x}(t_n) = x(t_n)$: Backward Euler

<u>Example</u>: Building thermal Control

f: cost; g: dynamics $\begin{cases} \text{heat equation} \\ \text{furnace} \end{cases}$

path constraints: thermal comfort constraints

5) Stochastic Programming

If no time: Take it from same reference.

If time:

$$\min_{x, \{y(\omega)\}} E_\Omega [f(x, y(\omega), \omega)]$$

$$g_1(x) \leq 0 \qquad \text{(vector notation)}$$

$$g_2(x, y(\omega), \omega) \leq 0 \qquad \forall \omega \in \Omega$$

Mihai Anitescu Notes

0.2.5.

If Ω is discrete and finite $\rightarrow$ NLP

If Ω is ∞ $\Rightarrow$ SAA: sample average approximation

first-stage decision

$$\min_{x, \{y(\omega_i)\}_{i=1}^{N_s}} \frac{1}{N_s} \sum f(x, y(\omega_i), \omega_i)$$

second-stage decision

$$\begin{cases} g_1(x) \leq 0 \\ g_2(x, y(\omega_i), \omega_i) \leq 0 \end{cases}$$

$\downarrow$ "$x^N \rightarrow x^*$" $\rightarrow$ see Shapiro.

It is again an NLP

Example: Optimal control of building system under ambient temperature uncertainty

Mihai Anitescu Notes

0.3 Why use Matlab?

Matlab is a language environment for computational Math

0.3.1 Why use Matlab?

→ A: extensive set of math libraries, UNIVERSE, PDE, optim, SP, ODE...

Alternatives: C
Fortran → B: productivity
Java
⋮

This is not a complete test set, but results very hard to come by.

I will "paraphrase" Funk

Really bad ECON :-)

$$\text{Productivity} = \frac{1}{\text{Relative effort} * \text{Time to Solution}}$$

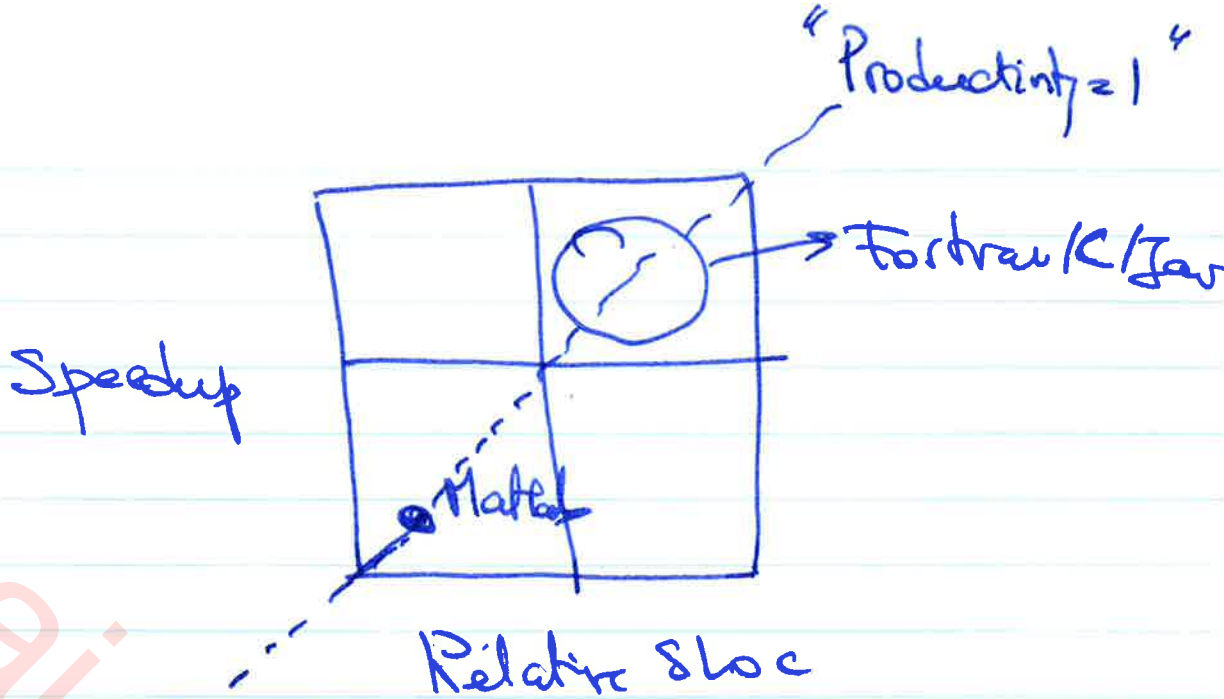
$$\frac{1}{\text{Time-to-solution}} = \text{Speedup}$$

$$\cancel{\approx \frac{\text{Speedup}}{\text{Rel}}}$$

Relative effort = SLOC "Source Line of Code"

Mihai Anitescu Notes

0.3.2



Therefore: $\text{Productivity} = \dfrac{\text{Speedup}}{\text{Relative SLOC}}$

For that test: about even...

$\Longleftrightarrow$

~~So Matlab provides a slow solution but with small developer time~~

~~If run twice~~

{ If run twice $\Rightarrow$ Productivity from C doubles!! (SLOC is same) }

Mihai Antiescu Notes

0.3.3 When need a solution for a problem you do not solve too often in a short amount of time → MATLAB.

Mihai Anitescu Notes

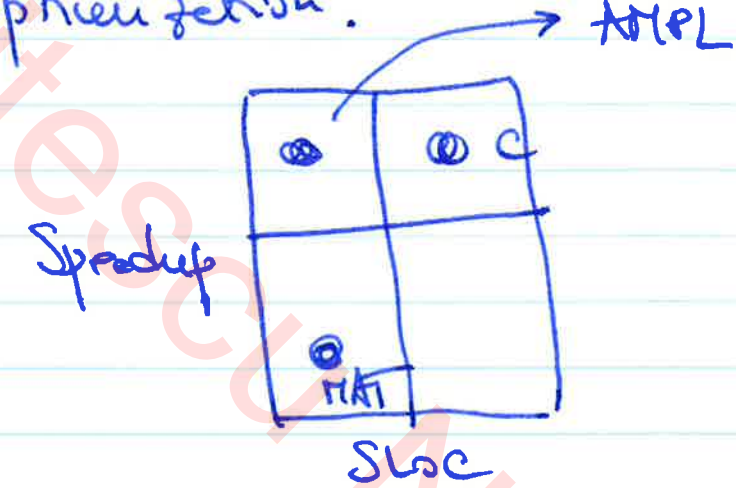
0. INTRODUCTION

0.4. When and why use modeling language

0.4.1 AMPL: A modeling language for Mathematical Programming → distribut-

an example

AMPL approaches natural language for optimization.



....

but not general!!

~~No PDE support..~~

~~Does no~~

Advantages → Derivatives are automatic

SLOC is small; Speedup-k

~~But Not general~~ ~~[illegible]~~ (~~No PDE support~~)

Mihai Anitescu Notes

0.4.2

Disadvantages

a) Not general. No PDE/ODE supp

b) Only fundamental functions gradients

So max likelihood in trouble..

Cannot compute $\nabla_\theta \log(\det V(\theta))$!!

Mihai Anitescu Notes

Should be 0.2 0. INTRO

0.5. Special cases of nonlinear programming

0.5.1 A) When $\mathcal{E} \cup \mathcal{I} = \emptyset$: unconstrained optimization

~~0.5.2~~ B) $\mathcal{I} = \emptyset$; $f = 0$ => Nonlinear Equation

C) f, c linear: Linear Programming

D) $f, \{c_i\}_{i \in \mathcal{I}}$ convex ; $\{c_i\}_{i \in \mathcal{E}}$ linear

convex programming

E) f : convex quadratic ; $\{c\}$ linear

convex quadratic programming

Mihai Anitescu Notes

0. INTRODUCTION.

0.6. What does it mean "I solve NLP"

0.6.1 CONTEXT. My aim is in local, iterative algorithms. What can I hope for?

Define "feasible set"

$$\mathcal{F} = \{ x \in \mathbb{R}^n \mid c_i(x) \leq 0 \quad i \in \mathcal{I}, \quad c_i(x) = 0 \quad i \in \mathcal{E} \}$$

NOTE: If $\mathcal{I} \cup \mathcal{E} = \emptyset \Rightarrow \mathcal{F} = \mathbb{R}^n$

ASSUME: $\mathcal{F} \neq \emptyset$.

A) Global Minimizer

$\mathcal{F} \ni x^*$ is GM if $f(x) \leq f(x^*)$ $\forall x \in \mathcal{F}$

B) Local Minimizer

$\mathcal{F} \ni x^*$ is LM if $\exists\, \mathcal{N}(x^*)$ s.t.

$$f(x^*) \leq f(x), \quad \forall x \in \mathcal{N}(x^*) \cap \mathcal{F}$$

Mihai Anitescu Notes

0.6.2 C) ~~A point~~ Isolated Local Minimizer

$\mathcal{F} \ni x^*$ is ILM if $\exists\, \mathcal{N}(x^*)$ s.t x^* is only LM in $\mathcal{N}(x^*) \cap \mathcal{F}$

D) Strict local minimizer

$\mathcal{F} \ni x^*$ is SLM if $\exists\, \mathcal{N}(x^*)$ s.t

$f(x^*) < f(x) \quad \forall x^* \neq x \in \mathcal{N}(x^*) \cap \mathcal{F}$

GM $\Rightarrow$ LM, ~~ILM, SLM~~ ;

ILM

SLM

GM $\not\Rightarrow$ ILM, SLM e.g. $\min\ x+y$ s.t $x+y=1$, $x \geq 0, y \geq 0$

$\frac{1}{2}, \frac{1}{2}$ is neither strict nor isolated

SLM $\not\Rightarrow$ ILM (see example book) $\boxed{x^4 \cos\frac{1}{x} + 2x^4}$

ILM $\Rightarrow$ SLM (I think, prove!)

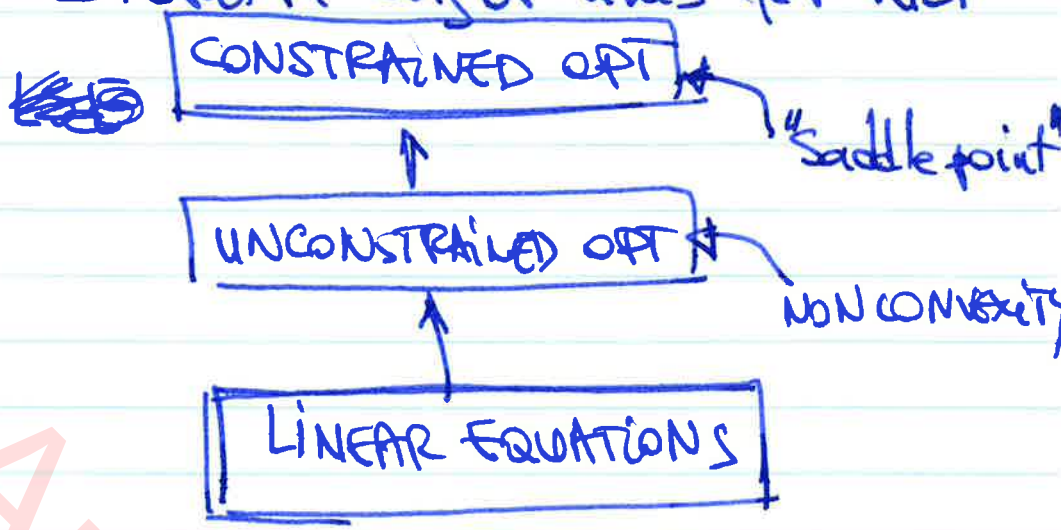
Mihai Antescu Notes

0. INTRODUCTION

0.7 GOALS AND SCOPE.

0.7.1 SCOPE: Iterative algorithms for NLP

Loosely



GOALS: Cannot hope to reach A, in general.

GOAL A: If started from a vicinity of x^*, converge to it "LOCAL CONVERGENCE"

LM, ILM, SLM

GOAL B:

From wherever I start my algorithm, I converge to a LM "GLOBAL CONVERGENCE"

Mihai Anitescu Notes

0.7.2

For unconstrained optimization, I will get close

For constrained optimization, constraints must "help".

It is weak, but it is perhaps amazing what good results one gets...

Mihai Anitescu Notes