Continous optimization
Lecture I - Traditional nonlinear optimization

François Glineur

Université catholique de Louvain - EPL/INMA & CORE
Francois.Glineur@uclouvain.be

Inverse Problems and Optimization in Low Frequency Electromagnetism
Continuous and Discrete Optimization workshop March 3rd 2008

Questions and comments ...

... are more than welcome, at any time !

Slides will be available on the web :
`http://www.core.ucl.ac.be/~glineur/`

References

This lecture is mainly based on a single recent reference

◇ Numerical Optimization, Jorge Nocedal and Stephen J. Wright, Springer, 1999

Motivation

Modelling and decision-making

Help to choose the best decision

$$\left.\begin{array}{rcl} \text{Decision} & \leftrightarrow & \text{vector of variables} \\ \text{Best} & \leftrightarrow & \text{objective function} \\ \text{Constraints} & \leftrightarrow & \text{feasible domain} \end{array}\right\} \Rightarrow \text{Optimization}$$

Use

- ◇ Numerous applications in practice
- ◇ Resolution methods efficient in practice
- ◇ Modelling and solving large-scale problems

Introduction

Applications

- ◇ Planning, management and scheduling

  Supply chain, timetables, crew composition, etc.
- ◇ Design

  Dimensioning, structural optimization, networks
- ◇ Economics and finance

  Portfolio optimization, computation of equilibrium
- ◇ Location analysis and transport

  Facility location, circuit boards, vehicle routing
- ◇ And lots of others ...

Two facets of optimization

- ◇ Modelling

  Translate the problem into mathematical language (sometimes trickier than you might think)

$$\Updownarrow$$

Formulation of an optimization problem

$$\Updownarrow$$

- ◇ Solving

  Develop and implement algorithms that are efficient in *theory* and in *practice*

Close relationship

- ◇ Formulate models that you know how to solve
- ◇ Develop methods applicable to real-world problems

Classical formulation

$$\min_{x \in \mathbb{R}^n} f(x) \text{ such that } x \in X \subseteq \mathbb{R}^n$$

(finite dimension) Often, we define

$$X = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0 \text{ and } h_j(x) = 0 \text{ for } i \in I, j \in J\}$$

Plan for Lecture I - first part

Introduction to continuous optimization

- ◇ An important special case: linear optimization
- ◇ Two paradigms: (traditional) nonlinear vs. convex
- ◇ Fundamentals of unconstrained optimization

Two strategies for unconstrained optimization

- ◇ Line search techniques
  - – Step length selection and convergence
- ◇ Trust-region techniques
  - – Model definition and convergence

Linear optimization: three examples

A. Diet problem

Consider a set of different foods for which you know

◇ Quantities of calories, proteins, glucids, lipids, vitamins contained per unit of weight

◇ Price per unit of weight

Given the nutritional recommendations with respect to daily supply of proteins, glucids, etc, design an optimal, i.e. meeting the constraints with the lowest cost

Formulation

◇ Index i for the food types $(1 \leq i \leq n)$

◇ Index j for the nutritional components $(1 \leq j \leq m)$

◇ Data (per unit of weight) :

$c_i \longrightarrow$ price of food type i,

$a_{ji} \longrightarrow$ amount of component j in food type i,

$b_j \longrightarrow$ daily recommendations for component j

◇ Unknowns:

Quantity x_i of food type i in the optimal diet

Formulation (continued)

This is a linear problem:

$$\min \sum_{i=1}^{n} c_i x_i$$

such that

$$x_i \geq 0 \; \forall i \text{ and } \sum_{i=1}^{n} a_{ji} x_i = b_j \; \forall j$$

Using matrix notations

$$\min c^{\mathrm{T}} x \text{ such that } Ax = b \text{ and } x \geq 0$$

This is a one of the most simple problems, and can be solved for large dimensions (m and $n \approx 10^7$)

B. Assignment problem

Given

- ◇ n workers
- ◇ n tasks to accomplish
- ◇ the amount of time needed for each worker to execute each of the tasks

Assign (bijectively) the n tasks to the n workers so that the total execution time is minimized

This is a discrete problem with an a priori exponential number of potential solutions $(n!)$ $\longrightarrow$ explicit enumeration is impossible in practice

Formulation

First idea: x_i denotes the number of the task assigned to person i (n integer variables between 1 and n)

Problem : how to force a bijection ?

Better formulation:

- ◇ Index i for workers ($1 \leq i \leq n$)
- ◇ Index j for tasks ($1 \leq j \leq n$)
- ◇ Data :

  $a_{ij} \longrightarrow$ duration of task j for worker i
- ◇ Unknowns:

  x_{ij} binary variable $\{0, 1\}$ indicating whether worker i executes task j

Formulation (continued)

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij}$$

such that

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j, \ \sum_{j=1}^{n} x_{ij} = 1 \ \forall i, \quad \text{and } x_{ij} \in \{0,1\} \ \forall i \ \forall j$$

- ◇ Higher number of variables (n^2) $\rightarrow$ more difficult ?
- ◇ Linear problem with integer (binary) variables $\rightarrow$ different algorithms
- ◇ But bijection constraint is simplified

Although it admits an exponential number of potential solutions, this problem can be solved very efficiently !

C. Travelling salesman problem

Given

- ◇ a travelling salesman that has to visit n cities going through each city once and only once
- ◇ the distance (or duration of the journey) between each pair of cities

Find an optimal tour that visits each city once with minimal length (or duration)

Also a discrete and exponential problem

Other application : soldering on circuit boards

Formulation

First idea: x_i describes city visited in position i during the tour (n integer variables between 1 and n)
Problem : how to require that each city is visited ?

Better formulation:

- ◇ Indices i and j for the cities $(1 \leq i, j \leq n)$
- ◇ Data :

  $a_{ij} \longrightarrow$ distance (or journey duration) between i and j
- ◇ Unknowns:

  x_{ij} binary variable $\{0, 1\}$ indicating whether the trip from city i to city j is part of the trip

Formulation (continued)

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij}$$

such that

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j, \ \sum_{j=1}^{n} x_{ij} = 1 \ \forall i, x_{ij} \in \{0, 1\} \ \forall i \ \forall j$$

and $\displaystyle\sum_{i \in S, j \notin S} x_{ij} \geq 1 \, \forall S$ with $S \subseteq \{1, \ldots, n\}, 1 < |S| < n$

- ◇ High (exponential) number of constraints
- ◇ Problem is a lot harder to solve ($n \approx 10^4$)

Algorithms and complexity

Why are these three problems different ?

Three linear problems: a priori among the simplest ... ?

- ◇ A. Diet: continuous variables $\longrightarrow$ linear optimization
- ◇ B. Assignment: discrete variables, exponential number of solutions

  $\longrightarrow$ linear integer optimization (but ...)
- ◇ C. Salesman: discrete variables, exponential number of constraints and solutions

  $\longrightarrow$ linear integer optimization

However, B is not more difficult than A while C is a lot harder than A and B !

Algorithmic complexity

Difficulty of a problem depends on the efficiency of methods that can be applied to solve it

$\Rightarrow$ what is a good algorithm ?

- ◇ Solves the problem (approximately)
- ◇ Until the middle of the 20th century: in finite time (number of elementary operations)
- ◇ Now (computers): in bounded time (depending on the problem size)

  $\longrightarrow$ algorithmic complexity (worst / average case)

Crucial distinction:

polynomial $\leftrightarrow$ exponential complexity

Algorithms for linear optimization

For linear optimization with continuous variables:
very efficient algorithms ($n \approx 10^7$)

◇ Simplex algorithm (Dantzig, 1947)

Exponential complexity but ...

Very efficient in practice

◇ Ellipsoid method (Khachiyan, 1978)

Polynomial complexity but ...

Poor practical performance

◇ Interior-point methods (Karmarkar, 1985)

Polynomial complexity and ...

Very efficient in practice (large-scale problems)

Algorithms for linear optimization (continued)

For linear optimization with discrete variables: algorithms are a lot less efficient, because the problem is intrinsically exponential
(cf. class of *NP-complete* problems)

◇ Linear relaxation (approximation)

◇ Branch and bound

Exponential complexity

$\longrightarrow$ Middle-scale or even small-scale problems ($n \approx 10^2$) can already be intractable

$\longrightarrow$ C is a lot harder to solve than A.

What about the assignment problem B. ?

Why can it be solved efficiently ?
It can be simplified: one can replace variables $x_{ij} \in \{0, 1\}$ by $0 \leq x_{ij} \leq 1$ without changing the optimal value and solutions !
We obtain linear optimization with continuous variables
$\longrightarrow$ Reformulation is sometimes crucial

In general, if one can replace the binary variables by continuous variables with an additional polynomial number of linear constraints, the resulting problem can be solved in polynomial time

Combinatorial/integer/discrete problems are not always difficult !

Nonlinear vs. convex optimization

Why is this course divided in two lectures ?

Linear optimization does not permit satisfactory modelling of all situations $\longrightarrow$ let us look again at

$$\min_{x \in \mathbb{R}^n} f(x) \text{ such that } x \in X \subseteq \mathbb{R}^n$$

where X is defined most of the time by

$$X = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0 \text{ and } h_j(x) = 0 \text{ for } i \in I, j \in J\}$$

and f, g_i and h_j might be nonlinear

A taxonomy

- ◇ Deterministic or stochastic* problem
- ◇ Accurate data or inaccurate/fuzzy* (robustness)
- ◇ Single or multiple* objectives
- ◇ Constrained or unconstrained problem
- ◇ Functions described analytically or using a black box*
- ◇ Continuous functions or not*, differentiable or not
- ◇ General, polynomial, quadratic, linear functions
- ◇ Continuous or discrete* variables

Switch categories: sometimes with *reformulations*

Back to complexity

Discrete sets X can make the problem difficult (with exponential complexity)
but even continuous problems can be difficult!

Consider a *simple* unconstrained minimization

$$\min f(x_1, x_2, \ldots, x_{10})$$

with smooth f (Lipschitz continuous with $L = 2$):

One can show that for any algorithm there exists some functions where at least 10^{20} iterations (function evaluations) are needed to find a solution with accuracy 1% !

Two paradigms

- ◇ Tackle all problems without any efficiency guarantee
  - – Traditional **nonlinear** optimization (this lecture)
  - – (Meta)-Heuristic methods
- ◇ Limit the scope to some classes of problems and get in return an efficiency guarantee
  - – Linear optimization
    - ∗ very fast specialized algorithms
    - ∗ but sometimes too limited in practice
  - – **Convex** optimization (next lecture)

Compromise: generality $\leftrightarrow$ efficiency

Unconstrained optimization

Fundamentals

$$\min_{x \in \mathbb{R}^n} f(x)$$

(Usually) assume f is smooth, bounded below

No other assumption is made on f

Reminder: universal algorithm does not exist!

What is a solution?

- ◇ Global minimizer x^* iff $f(x^*) \leq f(x)\ \forall x$ (but no hope of finding them)
- ◇ Local minimizer x^* iff $f(x^*) \leq f(x)\ \forall x \in \mathcal{N}$ with $\mathcal{N}$ some open neighborhood of x^*
- ◇ Strict local minimizer iff $f(x^*) < f(x)\ \forall x \neq x^* \in \mathcal{N}$
- ◇ Isolated local minimizer iff x^* is the only strict minimizer in some neighborhood of x^*

We have strict inclusions

$$\text{Isolated} \Rightarrow \text{Strict} \Rightarrow \text{Local} \Rightarrow \text{Global}$$

($x^4 \cos(1/x) + 2x^4$ has a strict min. in 0 but not isolated)

Recognizing a local minimum

Main tools (assuming enough smoothness where necessary):

First order:

$f(x+\Delta x) = f(x)+\nabla f(x+\alpha\Delta x)^{\mathrm{T}}\Delta x$ for some $0 < \alpha < 1$

and thus

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^{\mathrm{T}}\Delta x$$

Second order:

$f(x+\Delta x) = f(x)+\nabla f(x)^{\mathrm{T}}\Delta x+\frac{1}{2}(\Delta x)^{\mathrm{T}}\nabla^2 f(x+\alpha\Delta x)\Delta x$

for some $0 < \alpha < 1$ and thus

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^{\mathrm{T}}\Delta x + \frac{1}{2}(\Delta x)^{\mathrm{T}}\nabla^2 f(x)\Delta x$$

Necessary and sufficient conditions

- x^* local minimizer $\Rightarrow \nabla f(x^*) = 0$ (stationary point)
  ($\nabla f(x)$ continuous on neighborhood of x^*)
- x^* local minimizer $\Rightarrow \nabla^2 f(x^*) \succeq 0$ (p.s.d.)
  ($\nabla^2 f(x)$ continuous on neighborhood of x^*)
- $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$ (p.d.)
  $\Rightarrow x^*$ strict local minimizer
  ($\nabla^2 f(x)$ continuous on neighborhood of x^*)
  But no sufficient condition for non-strict minimizer!
- We only focus on local minimizers ; finding global minimizer is in general very difficult (not covered here)

Two strategies

- ◇ Line search
  - – Choose direction p_k
  - – Choose step length α_k solving (approximately)

$$\min_{\alpha>0} \phi(\alpha) = f(x_k + \alpha p_k)$$

- ◇ Trust region
  - – Choose model m_k such that

$$m_k(x_k + p_k) \approx f(x_k + p_k) \text{ around } x_k$$

  - – Choose trust region defined by $\|p_k\| \leq \Delta_k$
  - – Minimize model (approximately) over trust region

Somehow opposite strategies!

Line search

Which line search direction?

- ◇ Descent direction when $\nabla f(x)^{\mathrm{T}} p < 0$
- ◇ What is the best descent direction ?

$$\min_p \nabla f(x)^{\mathrm{T}} p \text{ such that } \|p\| = 1$$

has solution

$$p^S = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

$\Rightarrow$ steepest descent direction

- ◇ Newton direction considering

$$f(x+p) \approx f(x) + \nabla f(x)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 f(x) p = 0$$

$\Rightarrow p^N = -\nabla^2 f(x)^{-1} \nabla f(x)$ (assuming $\nabla^2 f(x)$ p.s.d.)

More about Newton direction

- ◇ $\nabla^2 f(x)$ p.s.d. $\Rightarrow p^N$ is a descent direction
- ◇ Computing second derivatives is potentially expensive or error prone
  $\Rightarrow$ replace $\nabla^2 f(x)$ by approximation B_k

A sound requirement:

$$\nabla f(x_{k+1}) \approx \nabla f(x_k) + \nabla^2 f(x_{k+1})(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|)$$

$$\Rightarrow \quad \nabla^2 f(x_{k+1})(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$\Rightarrow \quad B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$\Rightarrow \quad B_{k+1} s_k = y_k$$

These are called quasi-Newton directions $-B_k^{-1} \nabla f(x_k)$

Quasi-Newton directions

- ◇ Typically, impose symmetry on B_k (mimic Hessian)
- ◇ Update B_k with low-rank perturbation
  - – Symmetric rank one (SR1)

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^{\mathrm{T}}}{(y_k - B_k s_k)^{\mathrm{T}} s_k}$$

  - – BFGS (Broyden-Fletcher-Goldfarb-Shanno)

$$B_{k+1} = B_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k}$$

(rank two, B_k p.d. if B_0 p.d. and $s_k^{\mathrm{T}} y_k > 0$)

- ◇ Equivalent formulae for $H_k = B_k^{-1} \Rightarrow p_k = -H_k \nabla f(x_k)$

Scaling issues

- ◇ Poor scaling can arise from model

$$f(x_1, x_2) = 10^{-4}x_1^3 - 10^5 x_2^2$$

- ◇ Choice of units
- ◇ Diagonal rescaling

$$\hat{x} = Dx \text{ with } D = \operatorname{diag} d_i > 0$$

- ◇ Some methods are sensitive to poor scaling (e.g. steepest descent)
  some others are not (e.g. Newton's method)

$\Rightarrow$ scale-invariance is a desirable property
(usually more difficult for TR than for LS)

Choosing the step length: Wolfe conditions

◇ Sufficient decrease condition (Armijo):

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^{\mathrm{T}} p_k$$

with $0 < c_1 < 1$ (typically 10^{-4})

Always possible to satisfy when $\alpha \to 0$

$\Rightarrow$ we also need ...

◇ Curvature condition

$$\nabla f(x_k + \alpha p_k)^{\mathrm{T}} p_k \geq c_2 \nabla f(x_k)^{\mathrm{T}} p_k$$

with $c_1 < c_2 < 1$ (typically 0.9 for a (quasi)-Newton)

Strong Wolfe condition

Replace curvature condition by

$$\left|\nabla f(x_k + \alpha p_k)^{\mathrm{T}} p_k\right| \leq c_2 \left|\nabla f(x_k)^{\mathrm{T}} p_k\right|$$

Meaning

Recall that

$$\phi(\alpha) = f(x_k + \alpha p_k) \Rightarrow \phi'(\alpha) = \nabla f(x + \alpha p_k)^{\mathrm{T}} p_k$$

- ◇ Sufficient decrease condition forces rate of decrease to be at least $c_1 \phi'(0)$

$$\phi(\alpha) \leq \phi(0) - \alpha c_1 \phi'(0)$$

- ◇ Curvature condition bounds $\phi'(\alpha)$ (strong: $|\phi'(\alpha)|$)

$$\phi'(\alpha) \geq c_2 \phi'(0) \quad (\text{strong: } |\phi'(\alpha)| \leq c_2 \, |\phi'(0)| \,)$$

Existence

Assume

- ◇ p is a descent direction
- ◇ $\phi(\alpha) = f(x_k + \alpha p_k)$ is bounded below for $\alpha > 0$
- ◇ $0 < c_1 < c_2 < 1$

Then there are intervals of step lengths satisfying the Wolfe conditions and the strong Wolfe conditions

There exists a (one-dimensional) search procedure guaranteed to compute a point on this interval

These conditions are scale-invariant

Backtracking

As an alternative to the second curvature condition:
Choose starting $\alpha > 0$ and $0 < \rho < 1$
(e.g. $\alpha = 1$ for (quasi-)Newton)

- While $f(x_k + \alpha p_k) > f(x_k) + c_1 \alpha \nabla f(x_k)^{\mathrm{T}} p_k$
- Update α with $\rho\alpha$

In practice

Good αs can be found by interpolation techniques using

- function values and
- derivatives previously computed

e.g. minimize cubic interpolant based on $\phi(0)$, $\phi'(0)$, $\phi(\alpha^{(i)})$ and $\phi(\alpha^{(i-1)})$ or on $\phi(\alpha^{(i)})$, $\phi(\alpha^{(i-1)})$, $\phi'(\alpha^{(i)})$ and $\phi'(\alpha^{(i-1)})$

Convergence

Define angle θ_k between p_k and $\nabla f(x_k)$ by

$$\cos\theta_k = -\frac{\nabla f(x_k)^{\mathrm{T}} p_k}{\|\nabla f(x_k)\| \, \|p_k\|}$$

Assuming f bounded below, continuously differentiable, p_k descent directions satisfying Wolfe conditions, ∇f is Lipschitz continuous, we have

$$\sum_{k\geq 0} \cos^2\theta_k \, \|\nabla f(x_k)\|^2 < +\infty \quad \text{(Zoutendijk condition)}$$

- ◇ Implies $\cos^2\theta_k \, \|\nabla f(x_k)\|^2 \to 0$
- ◇ If angle bounded away from $\frac{\pi}{2}$ i.e. $\cos\theta_k \geq \delta > 0$ then $\|\nabla f(x_k)\|^2 \to 0$ stationary pt (e.g. steepest descent)

Convergence (continued)

◇ We only get stationary points
since no second-order information is used

◇ (Quasi-)Newton:
assuming

$$\|B_k\| \leq M \text{ and } \left\|B_k^{-1}\right\| \leq M$$

we have

$$\cos\theta_k \geq \frac{1}{M}$$

$\Rightarrow$ convergence when B_k

– are p.d. (to ensure descent property) and
– have bounded condition numbers

Rate of convergence: steepest descent

◇ For a convex quadratic $f(x) = \frac{1}{2}x^{\mathrm{T}}Qx - b^{\mathrm{T}}x$ with exact line searches (and λ_i eigenvalues of $Q \succ 0$)

$$\|x_{k+1} - x^*\|_Q \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \|x_k - x^*\|_Q$$

◇ In general with exact line searches and $\nabla^2 f(x^*) \succ 0$

$$f(x_{k+1}) - f(x^*) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 (f(x_k) - f(x^*))$$

with λ_i eigenvalues of $\nabla^2 f(x^*)$

$\longrightarrow$ linear rate $\longrightarrow$ slow (and inexact is worse)

$\kappa(Q) = 800,\, f(x_0) = 1,\; f(x^*) = 0 \Rightarrow f(x_{1000}) \approx 0.08$

Rate of convergence: general descent

For a general descent direction p_k: if

- $\diamond$ $\nabla^3 f$ continuous
- $\diamond$ $x \to x^*$ such that
- $\diamond$ x^* minimizer with $\nabla f(x) = 0$ and $\nabla^2 f(x^*) \succ 0$
- $\diamond$ α_k satisfies Wolfe with $c_1 \leq \frac{1}{2}$
- $\diamond$ $\lim_{k\to\infty} \left\| \nabla f(x_k) + \nabla^2 f(x_k) p_k \right\| / \|p_k\| = 0$

Then

- $\diamond$ $\alpha_k = 1$ becomes admissible for all $k \geq k_0$
- $\diamond$ $x_k \to x^*$ superlinearly if $\alpha_k = 1$ is chosen $\forall\, k \geq k_0$

$\Rightarrow$ full step $\alpha_k^{(0)} = 1$ must be tried first

Rate of convergence: quasi-Newton

For $p_k = B_k^{-1} \nabla f(x_k)$: if

◇ $\nabla^3 f$ continuous

◇ $x \longrightarrow x^*$ such that

◇ x^* minimizer with $\nabla f(x) = 0$ and $\nabla^2 f(x^*) \succ 0$

◇ $\alpha_k = 1 \ \forall k$

Then

◇ $x_k \longrightarrow x^*$ superlinearly if and only if

$$\lim_{k \to \infty} \frac{\left\| (B_k - \nabla^2 f(x^*)) p_k \right\|}{\| p_k \|} = 0$$

◇ $B_k \longrightarrow \nabla^2 f(x^*)$ not needed ! (only along p_k)

Rate of convergence: Newton

For $p_k = \nabla^2 f(x_k)^{-1} \nabla f(x_k)$: if

- $\diamond$ $\nabla^2 f$ Lipschitz continuous
- $\diamond$ x^* minimizer with $\nabla f(x) = 0$ and $\nabla^2 f(x^*) \succ 0$
- $\diamond$ x_0 sufficiently close to x^*
- $\diamond$ $\alpha_k = 1 \ \forall k$

Then

- $\diamond$ $x_k \longrightarrow x^*$
- $\diamond$ quadratic rate of convergence (cf. previous slide)
- $\diamond$ gradient norms $\|\nabla f(x_k)\|$ quadratically tend to 0

Trust region

Idea 1: a model

◇ Choose model m_k such that

$$m_k(x_k + p_k) \approx f(x_k + p_k) \text{ around } x_k$$

◇ Choose trust region defined by $\|p_k\| \leq \Delta_k$

◇ Minimize model (approximately) over trust region

Which model for trust-region?

◇ Quadratic to ease minimization

$$m_k(x_k + p) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} B_k p$$

$$m_k(x_k + p) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} B_k p$$

Impose model to be exact up to first order

- ◇ Case $B_k = 0$ … (not useful)
  $\Rightarrow$ steepest descent with step length depending on Δ_k
- ◇ Case $B_k = \nabla^2 f(x_k)$
  $\Rightarrow$ second-order model
- ◇ Case $B_k \approx \nabla^2 f(x_k)$ (e.g. SR1 or BFGS)
  $\Rightarrow$ quasi-Newton trust region

Advantage: $\Delta_k \Rightarrow$ minimum exists (even when $B_k \nsucc 0$)

Idea 2: update the trust region

◇ Model (to be trusted on region $\{\|x - x_k\| \leq \Delta_k\}$)

$$m_k(x_k + p_k) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} p_k + \frac{1}{2} p_k^{\mathrm{T}} B_k p_k$$

exact up to first or second order

◇ Trust region radius Δ_k:

– decrease when model is a bad approximation of f

– increase when model is a good approximation of f

◇ Actual criteria depends on p_k according to

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)} = \frac{\text{actual reduction}}{\text{predicted reduction}}$$

Complete algorithm

Given Δ_M, $0 < \Delta_0 \leq \Delta_M$, $0 \leq \eta < \frac{1}{4}$

For $k = 0, 1, 2, \ldots$

- ◇ Obtain p_k by solving (approximately)
$$\min m_k(x_k + p_k) \text{ such that } \|p_k\| \leq \Delta_k$$
- ◇ Compute ρ_k
- ◇ If $\rho_k < \frac{1}{4}$ set $\Delta_{k+1} = \frac{1}{4}\|p_k\|$
  If $\frac{1}{4} \leq \rho_k \leq \frac{3}{4}$ set $\Delta_{k+1} = \Delta_k$
  If $\frac{3}{4} < \rho_k$ set $\Delta_{k+1} = \min\{2\Delta_k, \Delta_M\}$
- ◇ If $\rho_k > \eta$ set $x_{k+1} = x_k + p_k$
  If $\rho_k \leq \eta$ set $x_{k+1} = x_k$

Cauchy point

The Cauchy point is the model minimizer on the steepest descent direction

$$p_k^C = -\tau_k \Delta_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$$

with

$$\tau_k = 1$$

when $\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k) \leq 0$ or

$$\tau_k = \min\{1, \|\nabla f(x_k)\|^3 / (\Delta_k \nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k))$$

when $\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k) > 0$

◇ Can be inside ($\tau_k < 1$) or on the boundary ($\tau_k = 1$)

Convergence result

The Cauchy point achieves the following decrease

$$m_k(x_k) - m_k(x_k + p_k^C) \geq \frac{1}{2} \|\nabla f(x_k)\| \min\{\Delta_k, \frac{\|\nabla f(x_k)\|}{\|B_k\|}\}$$

If one can guarantee a reduction of the same order $\forall k$

$$m_k(x_k) - m_k(x_k + p_k) \geq c_1 \|\nabla f(x_k)\| \min\{\Delta_k, \frac{\|\nabla f(x_k)\|}{\|B_k\|}\}$$

assuming ∇f is continuous, f is bounded below and a uniform bound $\|B_k\| \leq \beta$ $\forall k$ we have

◇ When $\eta = 0$: $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$

◇ When $0 < \eta < \frac{1}{4}$: $\lim_{k\to\infty} \nabla f(x_k) = 0$

Only stationarity is guaranteed

Strategies for computing a valid p_k

- ◇ Stick to p_k^C (but second-order information not used)
- ◇ Dogleg: minimize on path $x_k \longrightarrow x_k + p_k^U \longrightarrow x_k + p_k^B$

$$p_k^U = -\frac{\nabla f(x_k)^{\mathrm{T}} \nabla f(x_k)}{\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k)} \nabla f(x_k)$$

(this is the minimum along $-\nabla f(x_k)$)

$$p^B = -B_k^{-1} \nabla f(x_k)$$

(this is actual model minimizer)

- – path intersects trust region boundary at most once
- – intersection can be computed easily (scalar quadratic)

but this approach requires that B_k is pos. definite

◇ 2D subspace minimization: minimize on x_k+span$\{p_k^C, p_k^U\}$ (can be adapted when B_k is not p.d.)

In all three cases (Cauchy, dogleg, 2D subspace):
Cauchy decrease condition satisfied $\Rightarrow$ global convergence

Plan for Lecture I - second part

Towards constrained optimization

◇ More on unconstrained optimization techniques

- Linear conjugate gradients (very large-scale)
- Nonlinear conjugate gradients (large-scale)
- More on trust-region methods (medium-scale)

◇ Brief overview of constrained optimization techniques

- Optimality conditions
- Penalty methods, barrier methods and sequential quadratic programming (SQP)
- Nonsmooth optimization

Linear conjugate gradients

Motivation

Strictly convex quadratic optimization: when $A \succ 0$

$$\text{Minimize } \Phi(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x \quad \Leftrightarrow \quad \text{Solve } Ax = b$$

optimal x^* unique ; observe $r(x) = Ax - b = \nabla\Phi(x)$
First naive approach: coordinate descent:
minimize successively along axes $\Rightarrow$ not efficient
Better approach: define a set of *conjugate* directions

$$\{p_0, p_1, \ldots, p_l\} \text{ such that } p_i^{\mathrm{T}}Ap_j = 0 \text{ for all } i \neq j$$

Main result: $\Phi(x)$ can be minimized in exactly n steps using a sequence of n conjugate directions

Principle

Start with x_0 and define $x_{k+1} = x_k + \alpha_k p_k$ where α_k defines the exact (one-dimensional) minimizer of $\Phi(x_k + \alpha p_k)$

$$\alpha_k = -\frac{r_k^{\mathrm{T}} p_k}{p_k^{\mathrm{T}} A p_k}$$

$\{x_k\}$ converges to x^* in at most n steps for any x_0

$$x^* = x_0 + \sigma_0 p_0 + \sigma_1 p_1 + \cdots + \sigma_{n-1} p_{n-1}$$

- ◇ Conjugate directions $\Rightarrow$ independent directions
- ◇ $$\sigma_k = \frac{p_k^{\mathrm{T}} A (x^* - x_0)}{p_k^{\mathrm{T}} A p_k}$$
- ◇ $\sigma_k = \alpha_k$ for all k

Geometric interpretation

When A is diagonal, we get coordinate descent

Define $S = [p_0\ p_1 \dots p_{n-1}]$ and consider $x = S\tilde{x}$ to get

$$\tilde{\Phi}(\tilde{x}) = \Phi(S\tilde{x}) = \frac{1}{2}\tilde{x}^{\mathrm{T}}S^{\mathrm{T}}AS\tilde{x} - b^{\mathrm{T}}S\tilde{x}$$

$\Rightarrow$ same problem with $\tilde{b} = S^{\mathrm{T}}b$

and $\tilde{A} = S^{\mathrm{T}}AS$ which is diagonal

We have

$$r_k^{\mathrm{T}}p_i = 0 \text{ for all } 0 \leq i < k$$

and

$$x_k \text{ minimizes } \Phi(x_k) \text{ over } x_0 + \operatorname{span}\{p_0, p_1, \dots, p_{k-1}\}$$

Conjugate gradient

This was about conjugate directions:
what about conjugate gradients?

- ◇ $p_0 = -\nabla f(x_0) = -r_0$
- ◇ $p_k = -r_k + \beta_k p_{k-1}$

chosen such that conjugacy holds, i.e.

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}$$

In practice: $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$ and $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$ (cheap!)
(we also have $r_{k+1} = r_k + \alpha_k A p_l$)

Properties

Assume x^k is not the optimal solution x^*:

- ◇ $r_k^T r_i = 0$ for all $0 \leq i < k$
- ◇ $\text{span}\ \{r_0, r_1, \ldots, r_k\} = \text{span}\ \{p_0, p_1, \ldots, p_k\}$
- ◇ $\text{span}\ \{r_0, r_1, \ldots, r_k\} = \text{span}\ \{r_0, Ar_0, \ldots, A^k r_0\}$
- ◇ $p_k^T A p_i = 0$ for all $0 \leq i < k$

$\Rightarrow$ convergence in (at most) n steps
Gradients are orthogonal, not conjugate (*misnomer*)

Rate of convergence

For large n, we have to stop before n iterations ...

Rate of convergence (continued)

- ◇ If A has only r distinct eigenvalues, $x_r = x^*$
- ◇ If A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

$$\|x_{k+1} - x^*\|_A \leq \frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \|x_0 - x^*\|_A$$

$\rightarrow$ nice behavior for clustered eigenvalues

- ◇ One also has

$$\|x_k - x^*\|_A \leq \left(\frac{\sqrt{\lambda_1/\lambda_n} - 1}{\sqrt{\lambda_1/\lambda_n} + 1} \right)^{2k} \|x_0 - x^*\|_A$$

Preconditioning $x \rightarrow Cx \Leftrightarrow A \rightarrow C^{-T} A C^{-1}$
(ideally $C = L^{\mathrm{T}}$ such that $A = LL^{\mathrm{T}}$)

Nonlinear conjugate gradient

Introduction

$$\min_{x \in \mathbb{R}^n} f(x)$$

where f is no longer a strictly convex quadratic
Principle: slightly modify linear conjugate gradient

◇ Compute α_k with a line search
(instead of exact formula)

◇ Use actual gradient $\nabla f(x_k)$ instead of r_k

$\longrightarrow$ this is the Fletcher-Reeves method

Fletcher-Reeves (continued)

Descent direction?

$$\nabla f(x_k)^{\mathrm{T}} p_k = -\|\nabla f(x_k)\|^2 + \beta_k^{FR} \nabla f(x_k)^{\mathrm{T}} p_{k-1}$$

- If exact line search, second term is 0 $\Rightarrow$ descent
- Strong Wolfe conditions with $c_2 < \frac{1}{2}$ ensure first term dominates $\Rightarrow$ descent

Polak-Ribière method

Simple modification (among many others)

$$\beta_{k+1}^{PR} = \frac{\nabla f(x_{k+1})^{\mathrm{T}}(\nabla f(x_{k+1}) - \nabla f(x_k))}{\|\nabla f(x_k)\|^2}$$

- ◇ Not always descent direction
  (even with strong Wolfe)
- ◇ But with $\beta_{k+1}^{+} = \max\{\beta_{k+1}^{PR}, 0\} \rightarrow$ descent property
  (assuming slightly modified strong Wolfe)

More on trust-region algorithms

Exact minimization

Dogleg and subspace minimization: approximate minimizer by solving one linear system involving B_k

Goal: try to find an exact model minimizer with a little more work (i.e. solving a few more linear systems)

Hope: convergence to a better solution (true minimizer instead of stationary point)

$$\min m(x+p) = f(x) + \nabla f(x)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} B p \text{ s.t. } \|p\| \leq \Delta$$

admits optimal solution p^* iff there exists $\lambda \geq 0$ such that

$$(B+\lambda I)p^* = -\nabla f(x), \; \lambda(\Delta - \|p^*\|) = 0 \text{ and } B + \lambda I \succ 0$$

Exact minimization (continued)

Solving for $\lambda \geq 0$: define

$$p(\lambda) = -(B + \lambda I)^{-1} \nabla f(x) \text{ for } \lambda \text{ sufficiently large}$$

- ◇ Either $\lambda = 0$ with $\|p\| \leq \Delta$
- ◇ Or one looks for $\lambda > 0$ such that $p(\lambda) = \Delta$

$\Rightarrow$ one-dimensional root finding in λ

Assuming (for analysis only) that $B = Q \Lambda Q^{\mathrm{T}}$ one gets

$$p = \sum_{i=1}^{n} \frac{q_i^{\mathrm{T}} \nabla f(x)}{\lambda_i + \lambda} q_i \text{ and } \|p(\lambda)\|^2 = \sum_{i=1}^{n} \frac{(q_i^{\mathrm{T}} \nabla f(x))^2}{(\lambda_i + \lambda)^2}$$

One has $\|p(-\lambda_1)\| \rightarrow +\infty$ decreasing to $\|p(+\infty)\| \rightarrow 0$

Exact minimization (continued)

One can apply Newton's method, usually replacing

$$\|p(\lambda)\| - \Delta = 0$$

by

$$\frac{1}{\|p(\lambda)\|} - \frac{1}{\Delta} = 0$$

High accuracy not needed $\longrightarrow$ two or three iterations enough

Hard case

Problem when $q_1^T \nabla f(x) = 0$: one has then

$$\|p(\lambda)\| < \Delta \text{ for all } \lambda > -\lambda_1$$

Hard case (continued): solution

- ◇ Choose $\lambda = -\lambda_1$
- ◇ Find z such that $(B - \lambda_1 I)z = 0$ and $\|z\| = 1$ (eigenvector of B associated to λ_1)
- ◇ Choose p according to

$$p = \sum_{i:\lambda_i \neq \lambda_1} \frac{q_i^{\mathrm{T}} \nabla f(x)}{\lambda_i + \lambda} q_i + \tau z$$

such that

$$\|p\|^2 = \sum_{i:\lambda_i \neq \lambda_1} \frac{(q_i^{\mathrm{T}} \nabla f(x))^2}{(\lambda_i + \lambda)^2} + \tau^2 = \Delta^2$$

(one-dimensional problem in τ)

Global convergence results

- ◇ Using exact Hessians: $B_k = \nabla^2 f(x_k)$
- ◇ Assuming at each iteration $\|p_k\| \leq \gamma \Delta_k$ and

$$m(x_k) - m(x_k + p_k) \geq c_1 (m(x_k) - m(x_k + p_k^*))$$

  for some $0 < c_1 \leq 1$ and $\gamma > 0$

- ◇ With constant $0 < \eta < \frac{1}{4}$

one has

$$\liminf_{k \to \infty} \|\nabla f(x_k)\| = 0$$

Global convergence results (continued)

But it can get better:
if in addition level set $\{x \mid f(x) \le f(x_0)\}$ is compact

- ◇ Either algorithm terminates at a point satisfying second-order necessary conditions

$$\nabla f(x_k) = 0 \text{ and } \nabla^2 f(x_k) \succeq 0$$

- ◇ Or $\{x_k\}$ has a limit point x^* in the level set satisfying second-order necessary conditions

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \succeq 0$$

Convergence to (some) saddle-points cannot happen!

Constrained optimization

Optimality conditions

$$\min_{x\in\mathbb{R}^n} f(x) \text{ s.t. } c_i(x) = 0, i \in \mathcal{E} \text{ and } c_i(x) \geq 0, i \in \mathcal{I}\}$$

or

$$\min_{x\in\Omega} f(x)$$

with

$$\Omega = \{x \in \mathbb{R}^n \mid c_i(x) = 0, i \in \mathcal{E} \text{ and } c_i(x) \geq 0, i \in \mathcal{I}\}$$

x^* is a local minimizer iff $x^* \in \Omega$ and there exists some neighborhood $\mathcal{N}$ of x^* such that

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

Lagrangian, active set and constraint qualification

Lagrangian $\mathcal{L}(x, \lambda)$ is

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$$

Active set $\mathcal{A}(x)$ is

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} \mid c_i(x) = 0\}$$

Linear independence constraint qualification:
LICQ condition holds at x iff the set of active gradients

$$\{\nabla c_i(x), i \in \mathcal{A}(x)\}$$

is linearly independent $(\Rightarrow \nabla c_i \neq 0 \; \forall i \in \mathcal{A}(x))$

First-order necessary condition

Also called Karush-Kuhn-Tucker conditions (KKT)
Suppose x^* is a local minimizer and LICQ holds at x^*: then there exists a Lagrange multiplier vector λ with components λ_i, $i \in \mathcal{E} \cup \mathcal{I}$ such that

$$\nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla c_i(x^*),$$

$$c_i(x^*) = 0 \ \forall i \in \mathcal{E}, c_i(x^*) \geq 0 \ \forall i \in \mathcal{I},$$
$$\lambda_i^* \geq 0 \ \forall i \in \mathcal{I} \text{ and } \lambda_i^* c_i(x^*) = 0 \ \forall i \in \mathcal{I}$$

◇ First condition is equivalent to $\nabla_x \mathcal{L}(x, \lambda) = 0$ or

$$\nabla f(x^*) = \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i^* \nabla c_i(x^*)$$

since λ_i^* must be zero for each $i \notin \mathcal{A}(x^*)$
Intuitively: $\nabla f(x^*)$ can be nonzero but must be a linear combination of the active constraints gradients

- ◇ Last condition is called complementarity condition: at least one of λ_i^* and $c_i(x^*)$ must be zero $\forall i \in \mathcal{I}$
$\longrightarrow$ nonconvex condition ($\approx$ combinatorial type)
- ◇ There are more practical conditions to replace (LICQ) (constraint qualification $\longrightarrow$ broad literature)
- ◇ There are also second-order necessary or sufficient conditions (not both at the same time)
- ◇ Applied to linear optimization, one finds the standard primal-dual optimality conditions

Constrained optimization techniques

A brief overview

- Penalty methods: solve a sequence of unconstrained problems

$$\min f(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(x)$$

until solution to original problem is obtained

- Exact penalty: solve a single problem with suitable μ

$$\min f(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} |c_i(x)|$$

◇ Barrier methods: solve a sequence of unconstrained problems

$$\min f(x) - \mu \sum_{i \in \mathcal{I}} \log c_i(x)$$

◇ Augmented Lagrangian methods: combine Lagrangian with quadratic penalty

$$\min \mathcal{L}_A(x, \lambda, \mu) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(x)$$

Better numerical properties

- Quadratic programming: very special case

  $\longrightarrow$ tailored algorithms

  – active-set methods

  – interior-point methods

- Sequential quadratic programming techniques: approximate problem (locally) with a quadratic model

  Search direction p_k is solution to

$$\min_p \frac{1}{2} p^{\mathrm{T}} W_k p + \nabla f(x_k)^{\mathrm{T}} p \text{ s.t. } A_k p + c_k = 0$$

  Use merit function to determine step length

And when gradient is not available ?

Two situations

◇ Differentiable function with unknown gradient or too difficult to compute :

– automatic differentiation

– numerical estimation of derivatives

– derivative free methods
(e.g. based on interpolation techniques)

◇ Truly non-differentiable function :

– Dedicated methods for specific problems (nonsmooth optimization), e.g. eigenvalue optimization

– Reformulating the problem can make it differentiable (e.g. minimization of absolute values)

– One can sometimes trade non-differentiability for discrete variables

– In the general case (little or no information about the function), one can try the simplex method of Nelder-Mead (direct search method)

($\neq$ Dantzig's simplex algorithm for linear optimization)

Thanks for you attention