Nonlinear Optimization: Lecture I

Ya-xiang Yuan

Inst of Computational Mathematics and Scientific/Engineering Computing
Chinese Academy of Sciences, P. O. Box 2719, Beijing 100080, China
Email: yyx@lsec.cc.ac.cn

Optimization and Applications, Beijing, Sept 18-Oct 15, 2006

Outline

Outline

Nonlinear Optimization Problem

Nonlinear optimization problems:

$$\begin{array}{lll} \min & f(x) & \\ \text{subject to} & c_i(x) = 0, & i = 1, ..., m_e; \\ & c_i(x) \geq 0, & i = m_e + 1, ..., m. \end{array}$$

where, $f(x)$, and $c_i(x)$ $(i = 1, ..., m)$ are continuously differentiable functions from $\Re^n$ to $\Re$. At least one of the functions $f(x)$, and $c_i(x)$ $(i = 1, ..., m)$ is nonlinear.

Homework:

NLP-Q1. What is the most important character of nonlinear optimization?

Outline

Problem Classification - Unconstrained

$m = 0$ (Unconstrained Optimization)

- Quadratic minimization / Linear Least Squares:

$$f(x) = Q(x) = ||Ax - b||_2^2$$

- Nonlinear Least Squares: $f(x) = \|R(x)\|_2^2 = \sum_{i=1}^{N} r_i^2(x)$
- General Norm Minimization (Approximation)
  $f(x) = \|R(x)\|_p \qquad (e.g.\ p = 1, \infty)$
- convex unconstrained optimization: $f(x)$ convex.

Problem Classification - Equality Constrained

$m = m_e > 0$ (Equality Constrained Optimization)

- $f(x)$ quadratic, $c_i(x)$ linear: Equality constrained QP.
- $f(x)$ nonlinear, $c_i(x)$ linear: Linear equality constrained optimization.
- nonlinear equality constrained optimization.

Problem Classification - General Constrained

$m > m_e \geq 0$ (General Nonlinear Constrained Optimization)

- Quadratic programming: $f(x)$ quadratic and $c_i(x)$ linear.
- Linearly constrained nonlinear programming: $c_i(x)$ linear.
- general nonlinear constrained problems.

Outline

Different Aspect of NLP

- Theory
- Method
- Algorithm
- Software
- Application

Theory (Understanding the Problem)

- Optimal Conditions,
- Penalty Function,
- Dual Theory,
- Sensitivity,
- Solvability,
- Complexity,
- ...

Method (How to Solve the Problem)

- Steepest Descent Method,
- Newton's Method,
- Conjugate Gradient Method,
- Sequential QP Method,
- Interior Point Method,
- ... ,

Algorithm (Realization of Method)

Examples:

- inexact line search Newton's method
- an interior point algorithm for box constrained problems.
- a trust region algorithm for nonlinear LS.

Software (Implementation of Algorithms)

- LANCELOT (Conn et al. 1992, Augmented Lagrangian)
- MINOS (Murtagh and Saunders 1982, Augmented Lagrangian)
- SNOP (Gill et al. 2002, SQP)
- NPSOL (Schittkowski 1983, SQP)
- KNITRO (Byrd et al. 2000, Interior Point)
- LOGO (Vanderbei and Shanno 1999, Interior Point)
- IPOPT (Vächter and Biegler 2004, Interior Point)
- NLPSPR (Betts and Frank 1994, Interior Point)
- FilterSQP (Fletcher and Leyffer 1998, Filter SQP)
- ...

Applications (Not simply applying softwares!)

Special techniques can be applied because

- Special Structure
- Special linear/nonlinear properties

Applications in

- Science: (Information, Biology, Physics, ...)
- Finance:
- Transportation
- Defence

Outline

Steepest Descent

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

Steepest Descent Direction:

$$x_{k+1} = x_k + \alpha_k(-\nabla f(x_k))$$

exact line search, α_k^*: $f(x_k + \alpha_k^* d_k) = \min_{\alpha>0} f(x_k + \alpha d_k)$

Inexact line search, α_k satisfies

$$f(x_k + \alpha_k d_k) \leq f(x_k) + b_1 \alpha_k d_k^T \nabla f(x_k)$$

$$d_k^T \nabla f(x_k + \alpha_k d_k) \geq b_2 d_k^T \nabla f(x_k)$$

$b_1 \in (0, 1)$, $b_2 \in (b_1, 1)$.

Convergence of SD

Assume $\|\nabla^2 f(x)\| \leq M$.

$$f(x_k) - f(x_k + \alpha_k^* d_k) \geq \frac{1}{2M}[d_k^T \nabla f(x_k)/\|d_k\|_2]^2$$

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{b_1(1-b_2)}{M}[d_k^T \nabla f(x_k)/\|d_k\|_2]^2$$

Either $f(x_k) \to -\infty$ or

$$\begin{aligned} \sum_{k=1}^{\infty} \|\nabla f(x_k)\|_2^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle &= \sum_{k=1}^{\infty}[d_k^T \nabla f(x_k)/\|d_k\|_2]^2 \\ &\leq \infty \end{aligned}$$

For SD Method, $\cos\langle d_k, -\nabla f(x_k)\rangle = 1$. $\|\nabla f(x_k)\| \to 0$.

Convergence Rate of SD

Assume that $f(x)$ is convex quadratic. $f(x) = \frac{1}{2}x^T Hx$, H is a symmetric positive definite matrix. ($g_k = \nabla f(x_k)$)

$$\frac{f(x_k) - f(x_k + \alpha_k^*(-g_k))}{f(x_k)} = \frac{(x_k^T H^2 x_k)^2}{x_k^T H^3 x_k x_k^T H x_k} \geq \frac{\lambda_1(H)\lambda_n(H)}{(\lambda_1(H) + \lambda_n(H))^2}$$

where λ_1, λ_n: largest eigenvalue and the smallest eigenvalues.

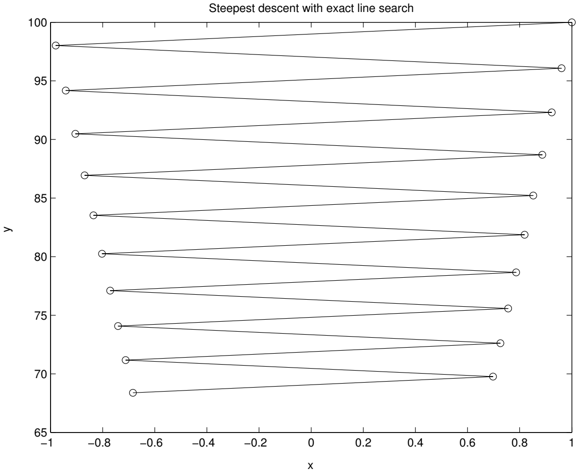
$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} \leq \left(\frac{\mu - 1}{\mu + 1}\right)^2 < 1.$$

$\mu = \lambda_1(H)/\lambda_n(H)$

Linear Convergence!

Exact line search + steepest descent gives bad results

A typical behavior of the steepest descent method is illustrated in the following picture where 20 iterates are plotted for the objective function $f(x, y) = 100x^2 + y^2$, starting at the initial point (1, 100).



BB method

The main idea of Barzilai and Borwein's

$$x_{k+1} = x_k - D_k g_k, \qquad \text{where } D_k = \alpha_k I.$$

Quasi-Newton property:

$$\min \|s_{k-1} - D_k y_{k-1}\|_2$$

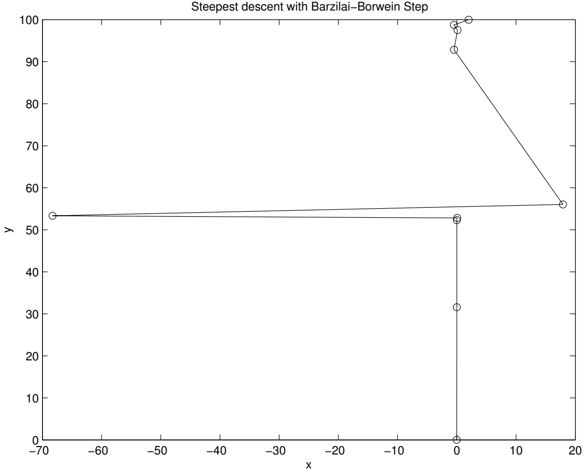
$$\min \|D_k^{-1} s_{k-1} - y_{k-1}\|_2$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

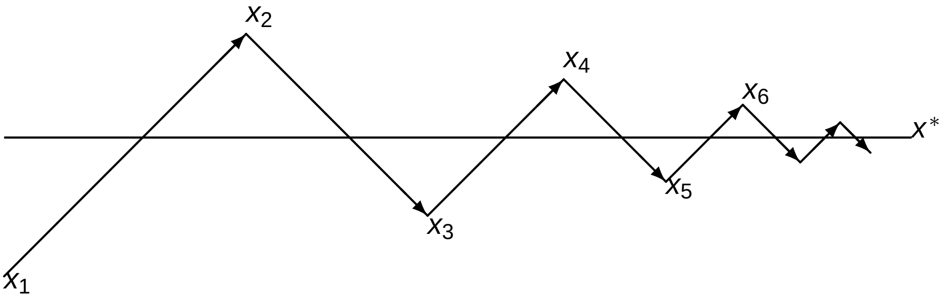
$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|_2^2}, \qquad \alpha_k = \frac{\|s_{k-1}\|_2^2}{s_{k-1}^T y_{k-1}} \quad (\alpha_{k-1}^* \quad !)$$

THEOREM *If $f(x)$ is a strictly quadratic convex function with 2 variables. The BB method gives that $\|g_k\| \le \eta \lambda^{-(\sqrt{2})^k}$, where $\lambda = \sigma_1(H)/\sigma_2(H)$, η is a constant independent of k.*

The behavior of the BB method for $f(x, y) = 100x^2 + y^2$, starting at the initial point (1, 100).



Cauchy Step too long



Stepsizes for Gradient Method

- SS1 (Short Step I)

$$\alpha_k^{SS1} = \gamma_1 \alpha_k^*$$

- Short Step II

$$\alpha_k^{SS2} = \begin{cases} \gamma_2 \alpha_k^*, & \text{if } k \text{ is odd;} \\ \alpha_k^*, & \text{if } k \text{ is even,} \end{cases}$$

- AM (Alternating Minimization)

$$\alpha_k^{AM} = \begin{cases} \dfrac{g_k^T g_k}{g_k^T H g_k}, & \text{if } k \text{ is odd;} \\ \dfrac{g_k^T H g_k}{g_k^T H^2 g_k}, & \text{if } k \text{ is even.} \end{cases}$$

- RAND (Random Step)

$$\alpha_k^{RAND} = \theta_k \alpha_k^*,$$

where θ_k is randomly chosen in $[0, 2]$.

A New Step Size

Question: *finite termination for two dimensional quadratic problems?*

$$\alpha_k^Y = \frac{2}{\sqrt{(1/\alpha_{k-1}^* - 1/\alpha_k^*)^2 + 4\|g_k\|_2^2/\|\|s_{k-1}\|_2^2} + 1/\alpha_{k-1}^* + 1/\alpha_k^*}$$

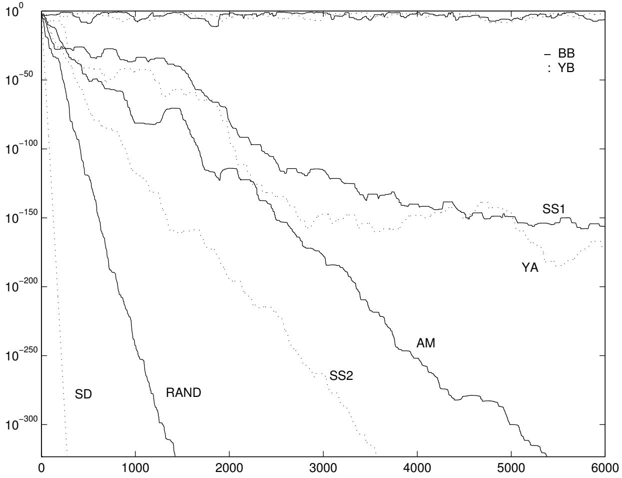
- Algorithm YA

$$\alpha_k^{YA} = \begin{cases} \alpha_k^*, & \text{if } k \text{ is odd;} \\ \alpha_k^Y, & \text{if } k \text{ is even} \end{cases}$$

- Algorithm YB

$$\alpha_k^{YB} = \begin{cases} \alpha_k^*, & \text{if } mod(k,3) \neq 0; \\ \alpha_k^Y, & \text{if } mod(k,3) = 0 \end{cases}$$

$$\xi_k = \frac{\min\{\sum_{j=0}^{L-1} |g_{k+j}^{(i)}|;\ i = 1, \ldots, n\}}{\max\{\sum_{j=0}^{L-1} |g_{k+j}^{(i)}|;\ i = 1, \ldots, n\}}$$



Outline

Conjugate Gradient Method:

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

Idea: (Conjugate Property) $d_k^T \nabla^2 f(x) d_j = 0$

Leading Choices for β_k

- Fletcher-Reeves: $\frac{\|g_{k+1}\|_2^2}{\|g_k\|_2^2}$
- Polak-Ribiere-Polyak: $\frac{g_{k+1}^T(g_{k+1}-g_k)}{\|g_k\|_2^2}$
- Henstenes-Stiefel: $\frac{g_{k+1}^T(g_{k+1}-g_k)}{d_k^T y_k}$
- Dai-Yuan: $\frac{\|g_{k+1}\|_2^2}{d_k^T y_k}$

Convergence of CG Methods

- For convex quadratic functions. Exact line searches imply
  - $d_k^T H d_j = 0 \quad (j = 1, ..., k-1)$
  - $g_k^T d_j = 0 \quad (j = 1, ..., k-1)$
  - $g_k^T g_j = 0 \quad (j = 1, ..., k-1)$
- For general nonlinear functions, analysis techniques:
  - $\cos\langle d_k, -g_k \rangle = \frac{-g_k^T d_k}{\|g_k\| \|d_k\|}$
  - Lower bound for $|g_k^T d_k|$
  - upper bound for $\|d_k\|$

Example 1: For exact line search: $|g_k^T d_k| = \|g_k\|^2$

Example 2: For F-R method: $\frac{d_{k+1}}{\|g_{k+1}\|^2} = -\frac{g_{k+1}}{\|g_{k+1}\|^2} + \frac{d_k}{\|g_k\|^2}$

Modification of β_k for inexact line searches?
Genearl coefficient for g_{k+1} instead of 1 ?

Reason: *Conjugacy is not necessary a good property when line search is not exact !*

Aim:

- Making d_{k+1} heading to the solution for two dimensional problem
- For general n dimensional problems, making the angle between d_{k+1} and $x^* - x_{k+1}$ as small as possible

Example (Stoer and Yuan): $d_{k+1} =$

$$\frac{(g_{k+1}^T y_k g_{k+1}^T s_k - s_k^T y_k \|g_{k+1}\|_2^2) g_{k+1} + (g_{k+1}^T y_k \|g_{k+1}\|_2^2 - \rho_k g_{k+1}^T s_k) s_k}{\rho_k s_k^T y_k - (g_{k+1}^T y_k)^2}$$

$$\rho_k \approx g_k^T \nabla^2 f(x_k) g_k$$

Outline

Newton's Method

$$\nabla f(x) = 0$$

$$\nabla f(x_k) + \nabla^2 f(x_k) d = 0$$

Newton's direction

$$d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Advantage of Newton's method

- Quadratic Convergence
- Easy to implement

Outline

Quasi-Newton

Newton's Method: $d_k = -(\nabla^2 f(x_k))^{-1} g_k$

Quasi-Newton Method

$$d_k = -(B_k)^{-1} g_k$$

B_k satisfies the Quasi-Newton Equation:

$$B_k s_{k-1} = y_{k-1}$$

Normally, B_k positive definite: "Variable Metric Method". (steepest descent direction if we use the norm $\|d\|_{B_k} = \sqrt{d^T B_k d}$)

Major Quasi-Newton Formulae

DFP (Davidon-Fletcher-Powell)

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

BFGS (Broyden-Fletcher-Goldfarb-Shanno)

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

SR1 (Symmetric Rank 1)

$$B_{k+1}^{(\mathrm{SR1})} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

Broyden Family: $B_{k+1}(\theta) = (1-\theta) B_{k+1}^{(\mathrm{BFGS})} + \theta B_{k+1}^{(\mathrm{DFP})}$

Properties of Quasi-Newton Method

- Invariance
- Conjugacy
- Quadratic Termination
- Least Changes

Convergence Properties of Quasi-Newton Methods

Theorem(Powell, 1971) *The DFP method converges for uniformly convex functions if exact line searches are used.*

This theorem can be extended to all Broyden's family due to Dixon.

Theorem(Powell, 1976) *The BFGS method converges for general convex functions if inexact line searches are used.*

The above theorem has been generalized to Broden's convex family ($\theta \in [0, 1]$) except DFP ($\theta = 1$). (Byrd, Nocedal, Yuan)

Without convexity, even BFGS fails to converge (Dai)

Question
Whether DFP with inexact LS converges for convex funtions?

Convergence Rate for quasi-Newton Method

Superlinearly Convergence:

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$$

For $n > 1$, least Q-Order is one, namely we can not prove

$$\|x_{k+1} - x^*\| = O(\|x_k - x^*\|^{1+\epsilon})$$

for any given $\epsilon > 0$.

Let $\quad \epsilon_k = \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|}$.

Question

Could we prove $\epsilon_k = O(\|x_{k-n} - x^*\|)$?

Limited Memory Quasi-Newton

BFGS represented as:($H_k = B_k^{-1}$)

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}$$

Limited BFGS formular:

- $H_k^{(0)} = \frac{s_k^T y_k}{y_k^T y_k} I$
- $H_k^{(j+1)} = H_{k+1}^{BFGS}(H_k^{(j)}, s_{k-m+j}, y_{k-m+j}), \quad (j = 0, 1, ..., m)$
- $H_{k+1} = H_k^{(m+1)}$

$m = 0$ + exact line search $\rightarrow$ conjugate gradient direction (DY)

Advantage: Only need to store $(s_{k-j}, y_{k-i})(i = 0, 1, ..., m)$.

Outline

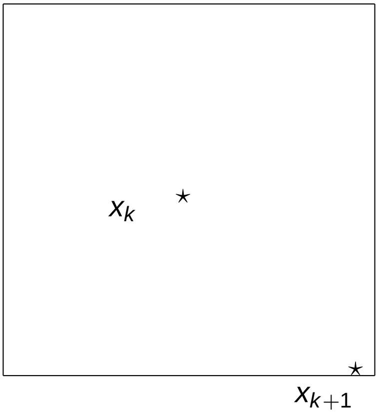
Trust Region

Main Idea: No line search,
“search” in a REGION

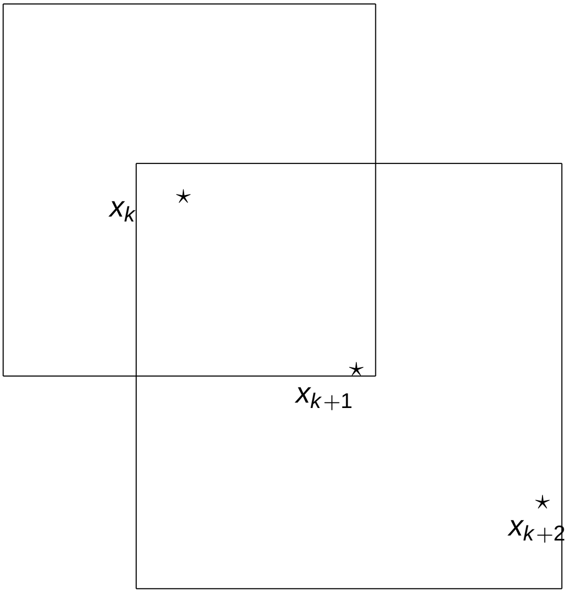
- Current Iterate: x_k
- Trust Region: R_k
- Trial Step: s_k

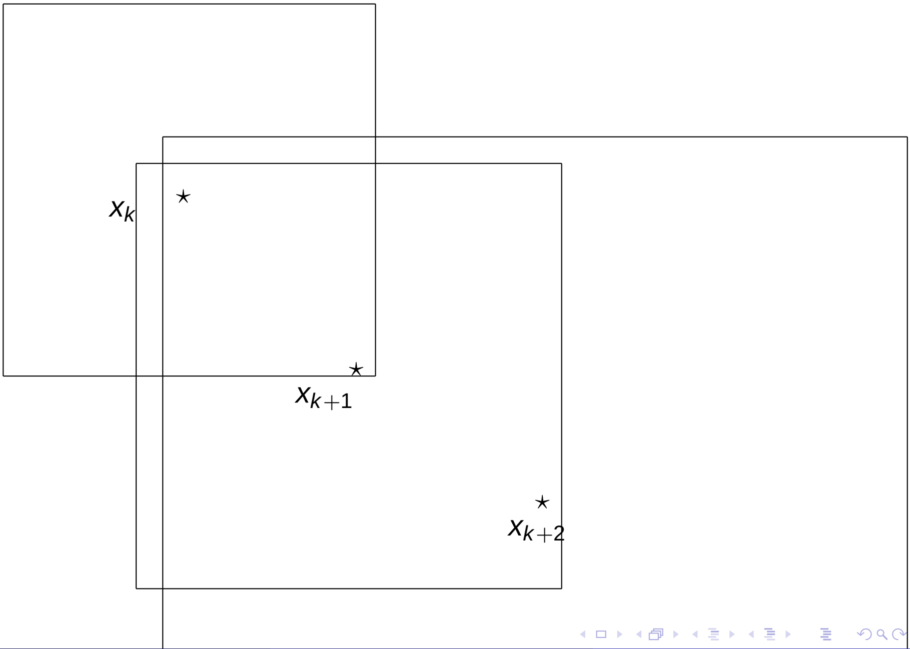
Motivation:

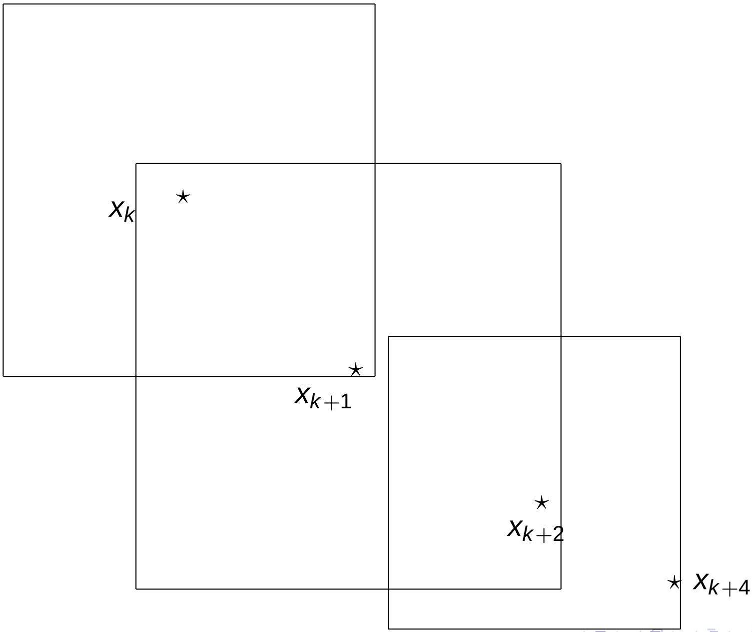
- Region better than a line
- When a search direction is not good, it is likely that the model is not accurate.



x_k
x_{k+1}


x_k
x_{k+1}
x_{k+2}


x_k
x_{k+1}
x_{k+2}


x_k
x_{k+1}
x_{k+2}
x_{k+4}

Key Contents of a Trust Region Algorithm

- Computing the trial step
- Judging the trial step
- updating the trust region
- modifying the approximate model

Key References:

- Powell(70s) – Early Papers on TR.
- Moré(1983) – Survey
- Conn, Gould, Toint(2000) – Monograph

Levenberg-Marquardt Method

$$\min_{x \in \Re^n} ||F(x)||_2^2$$

where $F(x) = (f_1(x), ...f_m(x))^T$.
The Gauss-Newton Step: $d_k = -(A(x_k)^T)^+ F(x_k)$
The Levenberg-Marquardt method

$$d_k = -(A(x_k)A(x_k)^T + \lambda_k I)^{-1} A(x_k) F(x_k)$$

which solves

$$\min_{d \in \Re^n} ||F(x_k) + A(x_k)^T d||_2^2 + \lambda_k ||d||_2^2$$

$$\min_{d \in \Re^n} ||F(x_k) + A(x_k)^T d||_2^2$$

$$s.t. \qquad ||d||_2 \leq \|d_k\|_2$$

Outline

A model TR for Unconstrained Optimization

– Trial Step s_k: Solving subproblem

$$\min_{d\in\Re^n} g_k^T s + \frac{1}{2} s^T B_k s = \phi_k(s)$$

$$s.t. \qquad ||s||_2 \le \Delta_k$$

where $g_k = \nabla f(x_k)$

Predicted Reduction:

$$Pred_k = \phi_k(0) - \phi_k(s_k).$$

Actually Reduction:

$$Ared_k = f(x_k) - f(x_k + s_k).$$

A Model Algorithm - Algorithm Descriptions

Step 1 Given $x_1 \in \Re^n$, $\Delta_1 > 0$, $\epsilon \geq 0$, $B_1 \in \Re^{n\times n}$symmetric;
$0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \leq \tau_0 \leq \tau_2 < 1$, $\tau_2 > 0$, $k := 1$.

Step 2 If $||g_k||_2 \leq \epsilon$ then stop; Compute s_k.

Step 3 Compute $r_k = Ared_k/Pred_k$ Let

$$x_{k+1} = \begin{cases} x_k & \text{if } r_k \leq \tau_0 \text{ ,} \\ x_k + s_k & \text{otherwise ;} \end{cases}$$

Choose Δ_{k+1} that satisfies

$$\Delta_{k+1} \in \begin{cases} [\tau_3||s_k||_2,\ \tau_4\Delta_k] & \text{if } r_k < \tau_2, \\ [\Delta_k,\ \tau_1\Delta_k] & \text{otherwise;} \end{cases}$$

Step 4 Update B_{k+1};
$k := k + 1$; go to Step 2.

Outline

Trust Region Subproblem

$$\min_{d\in\Re^n} g^T s + \frac{1}{2} s^T B s = \phi(s)$$

$$s.\,t. \qquad ||s||_2 \leq \Delta$$

THEOREM (Moré and Sorensen, 1983) s^* *is a solution of the Trust Region Subproblem TRS if and only if there exists* $\lambda^* \geq 0$ *such that*

$$(B + \lambda^* I)s^* = -g$$

and that $B + \lambda^* I$ *is positive semi-definite,* $||s^*||_2 \leq \Delta$ *and*

$$\lambda^*(\Delta - ||s^*||_2) = 0.$$

Exact solution of TRS

- $s^* = -B^{-1}g$
- $s^* = -(B + \lambda^* I)^{-1}g$
- $s^* = -(B + \lambda^* I)^{+}g + v$ ("Hard Case")

Define $d(\lambda) = -(B + \lambda I)^{-1}g$
Need: $\|d(\lambda)\| = \Delta$ or

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

- $1/\|d(\lambda)\|$ is a "linear-like" function.
- Newton's Method can be used.

Inexact Solution of TRS

Motivation:

- sufficient to ensure convergence
- save computations
- Newton's step near solution

Main approaches

- dog-leg method (Powell, Dennis, Mei)
  Cauchy point and Newton Step
- 2-Dimensional Search
  $$Span\{-g, -B^{-1}g\}$$
- Truncated CG method(Toint, Steihaug)
  precondition and truncation
- Semi-definite program approach (Rendl and Wolkowicz)

2 Dimensional Minimizer of TRS

THEOREM (Yuan, 1996)*Let s^* be the exact solution of trust region subproblem, and s^*_{2D} minimizes* $\min g^T s + \frac{1}{2} s^T Bs$ *subject to* $s \in Span\{-g, -B^{-1}g\}, \quad \|s\| \leq \Delta$. *There exists no positive constant c*

$$\frac{\phi(0) - \phi(s^*_{2D})}{\phi(0) - \phi(s^*)} \geq c.$$

for all $B > 0$.

EXAMPLE $g = (-1, -\epsilon, -\epsilon^3)^T$, and

$$B = \begin{pmatrix} \epsilon^{-3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon^3 \end{pmatrix},$$

where $\epsilon > 0$ is a very small positive number.

$$\frac{\phi(0) - \phi(s^*_{2D})}{\phi(0) - \phi(s^*)} = O(\epsilon)$$

Truncated CG Step

THEOREM (Yuan, 2000) *Assume* $B > 0$*, let* s^* *be the exact solution of trust region subproblem, and* s^*_{CG} *is the truncated CG step by Steihaug's algorithm, then*

$$\frac{\phi(0) - \phi(s^*_{CG})}{\phi(0) - \phi(s^*)} \geq \frac{1}{2}.$$

Idea of Proof

- $\cos < -g, s^*_{CG} > \geq \cos < -g, s^* >$
- Define CG path
- $d_k \in S_k = Span\{g, Bg, B^2 g, ..., B^{k-1} g\}$
- $Span\{g, (B + \lambda I)g, (B + \lambda I)^2 g, ..., (B + \lambda I)^{k-1} g\} = S_k$
- exact solution expressed as CG solution

Outline

Combining TR with Backtracking

TR+LS: (Nocedal +Yuan, 1991)
If s_k is not accepable by TR, $x_{k+1} = x_k + \delta^P s_k$ $(\delta \in (0,1), P > 1)$

Motivations:
– Trust Region Trial Step is descent
– Subproblem is difficult to solve

- Under the Framework of TR method
- Back-tracking if trial step is unacceptable
- nice convergence results as TR
- numerical results better than pure TR.

Homework
NLP-Q2. What is the main difference between LS and TR?
NLP-Q3. What is the similarity between LS and TR?