ORIE 6326: Convex Optimization

Quasi-Newton Methods

Professor Udell

Operations Research and Information Engineering
Cornell

April 10, 2017

Slides on steepest descent and analysis of Newton's method adapted from Stanford EE364a; slides on BFGS adapted from UCLA EE236C

Outline

Steepest descent method

normalized steepest descent direction (at x, for norm $\|\cdot\|$):

$$\Delta x_{\mathrm{nsd}} = \mathrm{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}$$

interpretation:

- for small v, $f(x+v) \approx f(x) + \nabla f(x)^T v$
- Δx_{nsd} is unit-norm step with most negative directional derivative

(unnormalized) steepest descent direction

$$\Delta x_{\mathrm{sd}} = \|\nabla f(x)\|_* \Delta x_{\mathrm{nsd}}$$

satisfies $\nabla f(x)^T \Delta x_{\mathrm{sd}} = -\|\nabla f(x)\|_*^2$

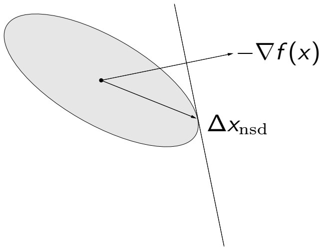
steepest descent method

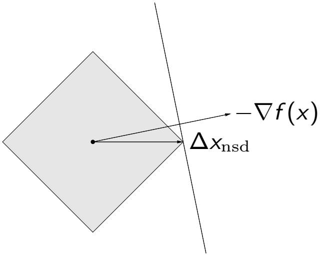
- general descent method with $\Delta x = \Delta x_{\mathrm{sd}}$
- convergence properties similar to gradient descent

examples

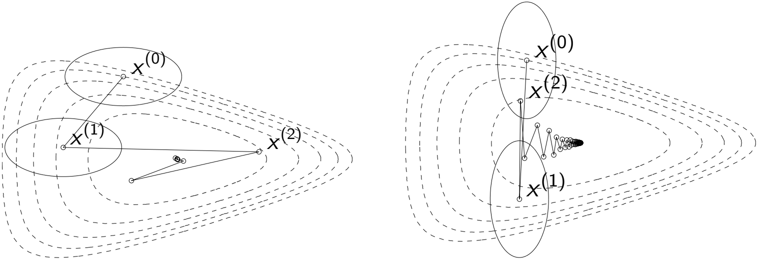
- Euclidean norm: $\Delta x_{\rm sd} = -\nabla f(x)$
- quadratic norm $\|x\|_B = (x^T B x)^{1/2}$ ($B \in \mathbf{S}^n_{++}$):
  $\Delta x_{\rm sd} = -B^{-1}\nabla f(x)$
- ℓ_1-norm: $\Delta x_{\rm sd} = -(\partial f(x)/\partial x_i)e_i$, where
  $|\partial f(x)/\partial x_i| = \|\nabla f(x)\|_\infty$

unit balls and normalized steepest descent directions for a quadratic norm and the ℓ_1-norm:





choice of norm for steepest descent



- steepest descent with backtracking line search for two quadratic norms
- ellipses show $\{x \mid \|x - x^{(k)}\|B = 1\}$
- equivalent interpretation of steepest descent with quadratic norm $\|\cdot\|_B$: gradient descent after change of variables $\bar{x} = B^{1/2}x$

shows choice of B has strong effect on speed of convergence

Outline

Recap: convergence analysis for gradient descent

$$\text{minimize} \quad f(x)$$

recall: we say (twice-differentiable) f is α-strongly convex and β-smooth if

$$\alpha I \preceq \nabla^2 f(x) \preceq \beta I$$

recall: if f is α-strongly convex and β-smooth, gradient descent converges linearly

$$f(x^{(k)}) - p^\star \leq \frac{\beta c^k}{2} \|x^{(0)} - x^\star\|^2,$$

where $c = (\frac{\kappa-1}{\kappa+1})^2$, $\kappa = \frac{\beta}{\alpha} \geq 1$ is condition number $\implies$ want $\kappa \approx 1$

Recap: convergence analysis for gradient descent

$$\text{minimize} \quad f(x)$$

recall: we say (twice-differentiable) f is α-strongly convex and β-smooth if

$$\alpha I \preceq \nabla^2 f(x) \preceq \beta I$$

recall: if f is α-strongly convex and β-smooth, gradient descent converges linearly

$$f(x^{(k)}) - p^\star \leq \frac{\beta c^k}{2} \|x^{(0)} - x^\star\|^2,$$

where $c = (\frac{\kappa-1}{\kappa+1})^2$, $\kappa = \frac{\beta}{\alpha} \geq 1$ is condition number $\implies$ want $\kappa \approx 1$
idea: can we minimize another function with $\kappa \approx 1$ whose solution will tell us the minimizer of f?

Preconditioning

for $D \succ 0$, the two problems

$$\text{minimize} \quad f(x) \quad \text{and} \quad \text{minimize} \quad f(Dz)$$

have solutions related by $x^\star = Dz^\star$

- gradient of $f(Dz)$ is $D^T \nabla f(Dz)$
- the second derivative (Hessian) of $f(Dz)$ is $D^T \nabla^2 f(Dz) D$

a gradient step on $f(Dz)$ with step-size $t > 0$ is

$$\begin{aligned} z^+ &= z - tD^T \nabla f(Dz) \\ Dz^+ &= Dz - tDD^T \nabla f(Dz) \\ x^+ &= x - tDD^T \nabla f(x) \end{aligned}$$

from prev analysis, we know gradient descent on z converges fastest if

$$\begin{aligned} D^T \nabla^2 f(Dz) D &\approx I \\ D &\approx (\nabla^2 f(Dz))^{-1/2} \end{aligned}$$

Approximate inverse Hessian

$B = DD^T$ is called the **approximate inverse Hessian**

can fix B or update it at every iteration:

- if B is constant: called **preconditioned** method (*e.g.*, preconditioned conjugate gradient)
- if B is updated: called **(quasi)-Newton** method

how to choose B? want

- $B \approx \nabla^2 f(x)^{-1}$
- easy to compute (and update) B
- fast to multiply by B

Outline

Newton step

$$\Delta x_{\mathrm{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

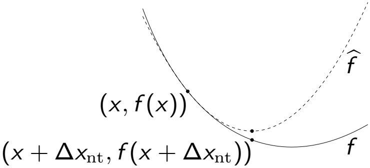
interpretations

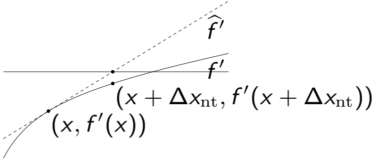
- $x + \Delta x_{\mathrm{nt}}$ minimizes second order approximation

$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

- $x + \Delta x_{\mathrm{nt}}$ solves linearized optimality condition

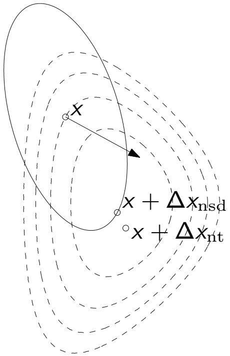
$$\nabla f(x+v) \approx \nabla \widehat{f}(x+v) = \nabla f(x) + \nabla^2 f(x) v = 0$$





- $\Delta x_{\rm nt}$ is steepest descent direction at x in local Hessian norm

$$\|u\|_{\nabla^2 f(x)} = \left(u^T \nabla^2 f(x) u\right)^{1/2}$$



dashed lines are contour lines of f; ellipse is
$\{x + v \mid v^T \nabla^2 f(x) v = 1\}$
arrow shows $-\nabla f(x)$

Newton decrement

$$\lambda(x) = \left(\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)\right)^{1/2}$$

a measure of the proximity of x to $x^\star$

properties

- gives an estimate of $f(x) - p^\star$, using quadratic approximation $\widehat{f}$:

$$f(x) - \inf_y \widehat{f}(y) = \frac{1}{2}\lambda(x)^2$$

- equal to the norm of the Newton step in the quadratic Hessian norm

$$\lambda(x) = \left(\Delta x_{\mathrm{nt}}^T \nabla^2 f(x) \Delta x_{\mathrm{nt}}\right)^{1/2}$$

- directional derivative in the Newton direction: $\nabla f(x)^T \Delta x_{\mathrm{nt}} = -\lambda(x)^2$
- affine invariant (unlike $\|\nabla f(x)\|_2$)

Newton's method

Algorithm 1 Newton's method.

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.

repeat

1. **Compute the Newton step and decrement.**
   $\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$.
2. **Stopping criterion. quit** if $\lambda^2/2 \le \epsilon$.
3. **Line search.** Choose step size t by backtracking line search.
4. **Update.** $x := x + t\Delta x_{\mathrm{nt}}$.

---

affine invariant, *i.e.*, independent of linear changes of coordinates:

Newton iterates for $\tilde{f}(y) = f(Ty)$ starting at $y^{(0)} = T^{-1}x^{(0)}$ are

$$y^{(k)} = T^{-1}x^{(k)}$$

Outline

Quasi-Newton method

Algorithm 2 Quasi-Newton method.

given a starting point $x \in \mathbf{dom}\, f$, $H \succ 0$, tolerance $\epsilon > 0$.

repeat

1. **Compute the step and decrement.**
   $\Delta x_{\mathrm{qn}} := -H^{-1}\nabla f(x); \quad \lambda^2 := \nabla f(x)^T H^{-1} \nabla f(x)$.
2. **Stopping criterion. quit** if $\lambda^2/2 \leq \epsilon$.
3. **Line search.** Choose step size t by backtracking line search.
4. **Update** x**.** $x := x + t\Delta x$.
5. **Update** H**.** Depends on specific method.

- Quasi-Newton methods defined by choice of H update
- can store and update $B = H^{-1}$ instead

Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

BFGS update

$$H^+ = H + \frac{yy^T}{y^Ts} - \frac{Hss^TH}{s^THs}$$

where $s = x^+ - x$, $y = \nabla f(x^+) - \nabla f(x)$

Inverse update

$$B^+ = \left(I - \frac{sy^T}{y^Ts}\right) B \left(I - \frac{ys^T}{y^Ts}\right) + \frac{ss^T}{y^Ts}$$

- note $y^Ts > 0$ if f is strongly convex (monotonicity of gradient)
- update to H is rank 2
- cost of update or inverse update is $O(n^2)$

BFGS update preserves positive definiteness

if $y^T s > 0$, then BFGS update preserves positive definiteness

proof: from inverse update, for any $v \in \mathbf{R}^n$

$$\begin{aligned} v^T B^+ v &= v^T \left((I - \frac{sy^T}{y^T s}) B (I - \frac{ys^T}{y^T s}) + \frac{ss^T}{y^T s} \right) v \\ &= (v - \frac{s^T v}{y^T s} y)^T B (v - \frac{s^T v}{y^T s} y) + \frac{(v^T s)^2}{y^T s} \end{aligned}$$

- first term is nonnegative because $B \succ 0$
- second term is nonnegative because $y^T s > 0$

BFGS update preserves positive definiteness

if $y^T s > 0$, then BFGS update preserves positive definiteness

proof: from inverse update, for any $v \in \mathbf{R}^n$

$$\begin{aligned} v^T B^+ v &= v^T \left((I - \frac{sy^T}{y^T s}) B (I - \frac{ys^T}{y^T s}) + \frac{ss^T}{y^T s} \right) v \\ &= (v - \frac{s^T v}{y^T s} y)^T B (v - \frac{s^T v}{y^T s} y) + \frac{(v^T s)^2}{y^T s} \end{aligned}$$

- first term is nonnegative because $B \succ 0$
- second term is nonnegative because $y^T s > 0$

can show $\Delta x_{\mathrm{qn}} = -H^{-1} \nabla f(x) > 0$, so Δx_{qn} is a descent direction:

- second term is 0 iff $v^T s = 0$
- first term is 0 if in addition $v = 0$
- taking $v = \nabla f(x)$, $s = x - x^-$, we see $\Delta x_{\mathrm{qn}} = -H^{-1} \nabla f(x) > 0$ unless $\nabla f(x) = 0$, *i.e.*, x is optimal

Secant condition

the BFGS update satisfies the **secant condition** $H^+ s = y$, *i.e.*,

$$H^+(x^+ - x) = \nabla f(x^+) - \nabla f(x)$$

Interpretation: define the second-order approximat at x^+

$$\hat{f}(z) = f(x^+) + \nabla f(x^+)^T (z - x^+) + \frac{1}{2}(z - x^+) H(z - x^+)$$

secant condition ensures gradient of $\hat{f}$ agrees with f at x:

$$\begin{aligned} \nabla \hat{f}(x) &= \nabla f(x^+) + H(x - x^+) \\ &= \nabla f(x) \end{aligned}$$

Secant method

for $f : \mathbf{R} \to \mathbf{R}$, BFGS with unit step size gives the secant method

$$x^+ = x - \frac{f'(x)}{H}, \qquad H = \frac{f'(x) - f'(x^-)}{x - x^-}$$

Limited memory quasi-Newton methods

main disadvantage of quasi-Newton method: need to store H or B

Limited-memory BFGS (L-BFGS): don't store B explicitly!

- instead, store the m (say, $m = 30$) most recent values of

$$s_j = x^{(j)} - x^{(j-1)}, \qquad y_j = \nabla f(x^{(j)}) - \nabla f(x^{(j-1)})$$

- evaluate $\delta x = B_k \nabla f(x^{(k)})$ recursively, using

$$B_j = \left(I - \frac{s_j y_j^T}{y_j^T s_j}\right) B_{j-1} \left(I - \frac{y_j s_j^T}{y_j^T s_j}\right) + \frac{s_j s_j^T}{y_j^T s_j}$$

  assuming $B_{k-m} = I$

Limited memory quasi-Newton methods

main disadvantage of quasi-Newton method: need to store H or B

Limited-memory BFGS (L-BFGS): don't store B explicitly!

- instead, store the m (say, $m = 30$) most recent values of

$$s_j = x^{(j)} - x^{(j-1)}, \qquad y_j = \nabla f(x^{(j)}) - \nabla f(x^{(j-1)})$$

- evaluate $\delta x = B_k \nabla f(x^{(k)})$ recursively, using

$$B_j = \left(I - \frac{s_j y_j^T}{y_j^T s_j}\right) B_{j-1} \left(I - \frac{y_j s_j^T}{y_j^T s_j}\right) + \frac{s_j s_j^T}{y_j^T s_j}$$

  assuming $B_{k-m} = I$
- advantage: for each update, just apply rank 1 + diagonal matrix to vector!
- cost per update is $O(n)$; cost per iteration is $O(mn)$
- storage is $O(mn)$

L-BFGS: interpretations

- only remember curvature of Hessian on active subspace

$$S_k = \mathrm{span}\{s_k, \ldots, s_{k-m}\}$$

- hope: locally, $\nabla f(x^{(k)})$ will approximately lie in active subspace

$$\nabla f(x^{(k)}) = g^S + g^{S^\perp}, \quad g^S \in S_k,\ g^{S^\perp} \text{ small}$$

- L-BFGS assumes $B_k \sim I$ on $S^\perp$, so $B_k g^{S^\perp} \approx g^{S^\perp}$;
  if $g^{S^\perp}$ is small, it shouldn't matter much.

Outline

Convergence of Quasi-Newton methods

Global convergence. if f is strongly convex, Newton method or BFGS with backtracking line search converge to solution $x^\star$ for any initial x and $H \succ 0$.

Local convergence of BFGS. if f is strongly convex and $\nabla^2 f(x)$ is Lipshitz continuous, local convergence is **superlinear**:
for k sufficiently large,

$$\|x^{k+1} - x^\star\|_2 \leq c_k \|x^{(k)} - x^\star\|_2 \to 0$$

where $c_k \to 0$

Local convergence of Newton method. if f is strongly convex and $\nabla^2 f(x)$ is Lipshitz continuous, local convergence is **quadratic**:
for k sufficiently large,

$$\|x^{k+1} - x^\star\|_2 \leq c_k \|x^{(k)} - x^\star\|_2^2 \to 0$$

Classical convergence analysis of Newton method

assumptions

- f strongly convex on S with constant α
- $\nabla^2 f$ is Lipschitz continuous on S, with constant $\gamma > 0$:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq \gamma \|x - y\|_2$$

  (γ measures how well f can be approximated by a quadratic function)

outline: there exist constants $\eta \in (0, \alpha^2/\gamma)$, $\mu > 0$ such that

- if $\|\nabla f(x)\|_2 \geq \eta$, then $f(x^{(k+1)}) - f(x^{(k)}) \leq -\mu$
- if $\|\nabla f(x)\|_2 < \eta$, then

$$\frac{\gamma}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{\gamma}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2$$

damped Newton phase ($\|\nabla f(x)\|_2 \geq \eta$)

- most iterations require backtracking steps
- function value decreases by at least γ
- if $p^\star > -\infty$, this phase ends after at most $(f(x^{(0)}) - p^\star)/\gamma$ iterations

quadratically convergent phase ($\|\nabla f(x)\|_2 < \eta$)

- all iterations use step size $t = 1$
- $\|\nabla f(x)\|_2$ converges to zero quadratically: if $\|\nabla f(x^{(k)})\|_2 < \eta$

$$\frac{\beta}{2\alpha^2}\|\nabla f(x^l)\|_2 \leq \left(\frac{\beta}{2\alpha^2}\|\nabla f(x^k)\|_2\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{2^{l-k}}, \qquad l \geq k$$

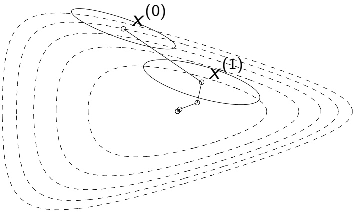
conclusion: number of iterations until $f(x) - p^\star \le \epsilon$ is bounded above by

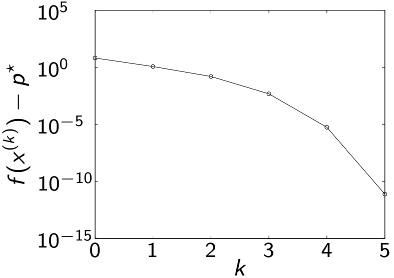
$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(\epsilon_0/\epsilon)$$

- μ, ϵ_0 are constants that depend on α, γ, $x^{(0)}$
- second term is small (of the order of 6) and almost constant for practical purposes
- in practice, constants α, γ (hence μ, ϵ_0) are usually unknown
- provides qualitative insight in convergence properties (*i.e.*, explains two algorithm phases)

Examples

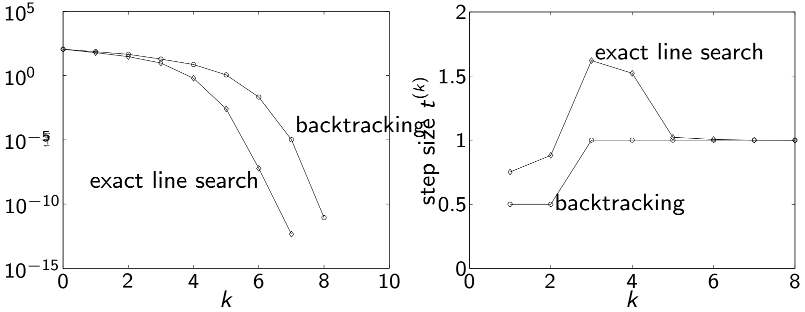
example in $\mathbf{R}^2$





- backtracking parameters $\alpha = 0.1$, $\beta = 0.7$
- converges in only 5 steps
- quadratic local convergence

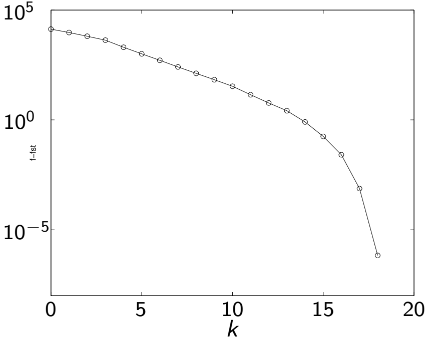
example in $\mathbf{R}^{100}$



- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$
- backtracking line search almost as fast as exact l.s. (and much simpler)
- clearly shows two phases in algorithm

example in $\mathbf{R}^{10000}$ (with sparse a_i)

$$f(x) = -\sum_{i=1}^{10000} \log(1 - x_i^2) - \sum_{i=1}^{100000} \log(b_i - a_i^T x)$$



- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$.
- performance similar as for small examples

Self-concordance

shortcomings of classical convergence analysis

- depends on unknown constants (α, γ, $\ldots$)
- bound is not affinely invariant, although Newton's method is

convergence analysis via self-concordance (Nesterov and Nemirovski)

- does not depend on any unknown constants
- gives affine-invariant bound
- applies to special class of convex functions ('self-concordant' functions)
- developed to analyze polynomial-time interior-point methods for convex optimization

Self-concordant functions

definition

- convex $f : \mathbf{R} \to \mathbf{R}$ is self-concordant if $|f'''(x)| \leq 2f''(x)^{3/2}$ for all $x \in \mathbf{dom}\, f$
- $f : \mathbf{R}^n \to \mathbf{R}$ is self-concordant if $g(t) = f(x + tv)$ is self-concordant for all $x \in \mathbf{dom}\, f$, $v \in \mathbf{R}^n$

examples on R

- linear and quadratic functions
- negative logarithm $f(x) = -\log x$
- negative entropy plus negative logarithm: $f(x) = x \log x - \log x$

affine invariance: if $f : \mathbf{R} \to \mathbf{R}$ is strongly convex, then $\tilde{f}(y) = f(ay + b)$ is strongly convex:

$$\tilde{f}'''(y) = a^3 f'''(ay + b), \qquad \tilde{f}''(y) = a^2 f''(ay + b)$$

Self-concordant calculus

properties

- preserved under positive scaling $\alpha \geq 1$, and sum
- preserved under composition with affine function
- if g is convex with $\mathbf{dom}\, g = \mathbf{R}_{++}$ and $|g'''(x)| \leq 3g''(x)/x$ then

$$f(x) = \log(-g(x)) - \log x$$

  is self-concordant

examples: properties can be used to show that the following are strongly convex

- $f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$ on $\{x \mid a_i^T x < b_i,\ i = 1, \ldots, m\}$
- $f(X) = -\log\det X$ on $\mathbf{S}_{++}^n$
- $f(x) = -\log(y^2 - x^T x)$ on $\{(x, y) \mid \|x\|_2 < y\}$

Convergence analysis for self-concordant functions

summary: there exist constants $\eta \in (0, 1/4]$, $\gamma > 0$ such that

- if $\lambda(x) > \eta$, then

$$f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$$

- if $\lambda(x) \leq \eta$, then

$$2\lambda(x^{(k+1)}) \leq \left(2\lambda(x^{(k)})\right)^2$$

(η and γ only depend on backtracking parameters α, β)

complexity bound: number of Newton iterations bounded by

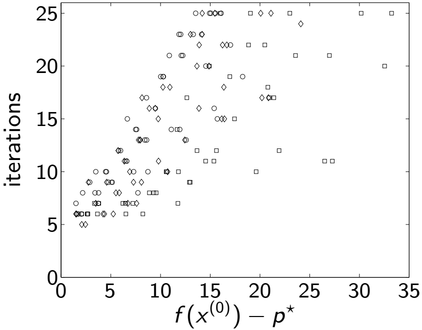
$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(1/\epsilon)$$

for $\alpha = 0.1$, $\beta = 0.8$, $\epsilon = 10^{-10}$, bound evaluates to
$375(f(x^{(0)}) - p^\star) + 6$

numerical example: 150 randomly generated instances of

$$\text{minimize} \quad f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$$

$\bigcirc$: $m = 100$, $n = 50$
$\square$: $m = 1000$, $n = 500$
$\diamondsuit$: $m = 1000$, $n = 50$



- number of iterations much smaller than $375(f(x^{(0)}) - p^\star) + 6$
- bound of the form $c(f(x^{(0)}) - p^\star) + 6$ with smaller c (empirically) valid

Implementation

main effort in each iteration: evaluate derivatives and solve Newton system

$$H\Delta x = -g$$

where $H = \nabla^2 f(x)$, $g = \nabla f(x)$

via Cholesky factorization

$$H = LL^T, \qquad \Delta x_{\mathrm{nt}} = -L^{-T}L^{-1}g, \qquad \lambda(x) = \|L^{-1}g\|_2$$

- cost $(1/3)n^3$ flops for unstructured system
- cost $\ll (1/3)n^3$ if H sparse, banded

example of dense Newton system with structure

$$f(x) = \sum_{i=1}^{n} \psi_i(x_i) + \psi_0(Ax+b), \qquad H = D + A^T H_0 A$$

- assume $A \in \mathbf{R}^{p \times n}$, dense, with $p \ll n$
- D diagonal with diagonal elements $\psi_i''(x_i)$; $H_0 = \nabla^2 \psi_0(Ax+b)$

method 1: form H, solve via dense Cholesky factorization: (cost $(1/3)n^3$)

method 2 factor $H_0 = L_0 L_0^T$; write Newton system as

$$D\Delta x + A^T L_0 w = -g, \qquad L_0^T A \Delta x - w = 0$$

eliminate Δx from first equation; compute w and Δx from

$$(I + L_0^T A D^{-1} A^T L_0) w = -L_0^T A D^{-1} g, \qquad D\Delta x = -g - A^T L_0 w$$

cost: $2p^2 n$ (dominated by computation of $L_0^T A D^{-1} A^T L_0$)

References

- Nocedal and Wright, Numerical Optimization
- Lieven Vandenberghe, UCLA EE236C
- Stephen Boyd, Stanford EE364a