Lecture 16

Smooth unconstrained minimization (part two)

- Newton's method (follow'd)
- quasi-Newton methods
- conjugate gradients
- self-concordance & Newton's method

Newton's method

given starting point $x \in \textbf{dom}\, f$
repeat
1. *Compute Newton direction* $v = -\nabla^2 f(x)^{-1} \nabla f(x)$
2. *Line search.* Choose a step size t
3. *Update.* $x := x + tv$

until stopping criterion is satisfied

(also called *damped* or *guarded* Newton method)

- global convergence with backtracking or exact line search
- quadratic local convergence
(hence, stopping criterion not an issue)

Affine invariance of Newton method

- use new coords $x = T\bar{x}$, $\det T \neq 0$
- apply Newton to $g(\bar{x}) = f(T\bar{x})$
- then $x^{(k)} = T\bar{x}^{(k)}$

e.g., Newton method not affected by variable scaling (cf. gradient, steepest descent)

Convergence analysis

assumptions: $mI \preceq \nabla^2 f(x) \preceq MI$ and Lipschitz condition

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

results: two phases

1. damped Newton phase: $\|\nabla f(x)\| \geq \eta_1$:

$$f(x^+) \leq f(x) - \eta_2,$$

hence

$$\#\text{iterations} \leq \frac{f(x^{(0)}) - f^\star}{\eta_2}$$

2. quadratically convergent phase: $\|\nabla f(x)\| < \eta_1$

$$\#\text{iterations} \leq \log_2 \log_2(\epsilon_0/\epsilon)$$

total #iterations bounded by

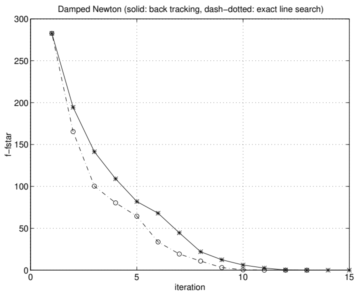
$$\frac{f(x^{(0)}) - f^\star}{\eta_2} + \log_2 \log_2(\epsilon_0/\epsilon)$$

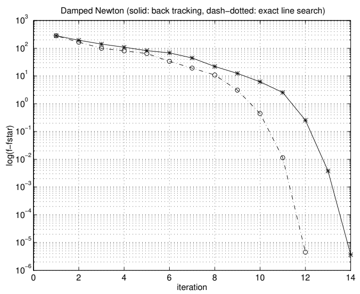
η_1, η_2, ϵ_0 depend on m, M, L
(and α, β for backtracking)

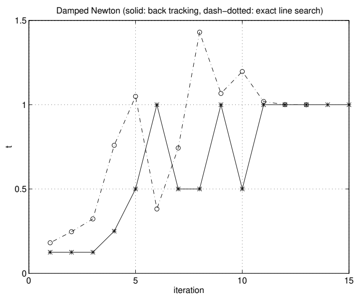
Numerical example: Newton method

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

$m = 100$, $n = 50$







solid line: backtracking ($\beta = 0.5$, $\alpha = 0.2$)
dashed line: exact line search
(iters are more expensive than gradient method)

Quasi-Newton methods

idea: replace $\nabla^2 f(x)$ by approximation H

given starting point $x \in \textbf{dom}\, f$, $H \succ 0$
repeat
1. *Compute quasi-Newton direction.* $v = -H^{-1}\nabla f(x)$
2. *Line search.* Choose a step size t
3. *Update* H.
4. *Update* x. $x := x + tv$

until stopping criterion is satisfied

many update rules $H \rightarrow H^+$, which all satisfy:

- $H = H^T \succ 0$
- secant condition:

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

- $H^{-1}\nabla f(x)$ more easily computed (than $\nabla^2 f(x)^{-1}\nabla f(x)$)

advantages (compared to Newton method)

- don't need to evaluate $\nabla^2 f(x)$
- v can be computed in $O(n^2)$ operations

disadvantage (compared to Newton method)

- local convergence fast, but not quadratic

quasi-Newton methods

- converge in n steps for (cvx) quadratic f on $\mathbf{R}^n$
- widely used in practice

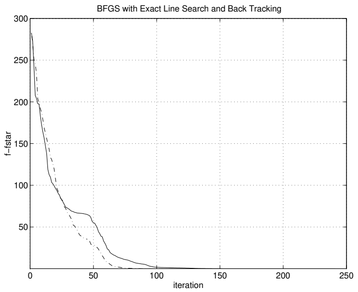
most common is Broyden-Fletcher-Goldfarb-Shanno (BFGS):

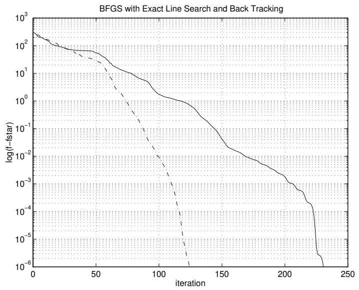
$$\begin{aligned} y &= \nabla f(x^+) - \nabla f(x) \\ s &= x^+ - x \\ H^+ &= H + \frac{yy^T}{y^T s} - \frac{Hss^T H}{s^T H s} \end{aligned}$$

Numerical example: BFGS method

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

$m = 100$, $n = 50$





solid line: backtracking ($\beta = 0.5$, $\alpha = 0.2$)

dashed line: exact line search

when comparing with Newton method, remember:

- BFGS is $O(n^2)$ per iter (plus finding ∇f)
- Newton is $O(n^3)$ per iter (plus finding ∇f, $\nabla^2 f$)

Conjugate gradients method

given starting point $x \in \textbf{dom}\, f$; $\alpha = 0$, $v = 0$
repeat
1. *Compute search direction.* $v := -\nabla f(x) + \alpha t v$.
2. *Line search.* Choose $t := \textbf{argmin}_{t>0} f(x + tv)$.
3. *Update x and α.*

$$y := \nabla f(x+tv) - \nabla f(x),$$
$$\alpha := y^T \nabla f(x+tv)/ty^T v,$$
$$x := x + tv.$$

until stopping criterion is satisfied.

– same as BFGS with H reset to I each iteration

– v is linear comb of current and previous gradient

– converges exactly in n steps for (cvx) quadratic f (at least in theory)

– fast, but not quadratic, local convergence

advantage: no need to form or store H
(hence can be used for very large problems)

disadvantages:

– can work very poorly in practice (numerical issues)

– not affinely invariant

change of coordinates ('preconditioning') can improve practical performance

application:
solving $Ax = b$, with very large, sparse $A \succ 0$

– use CG to minimize $(1/2)x^T Ax - b^T x$

– requires n evals of $\nabla f(x) = Ax - b$ (in theory)

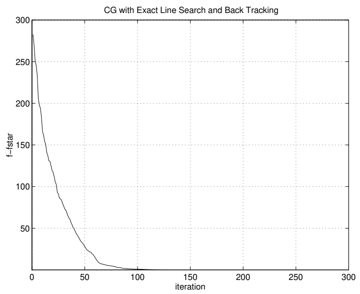
– much faster than direct method
(if lucky, or with good preconditioning)

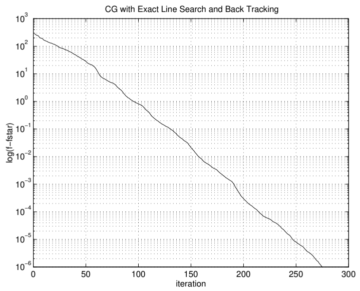
Numerical example: CG method

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

$m = 100$, $n = 50$

CG (with exact line search)





remember: each CG iteration is very cheap
(finding ∇f, a few inner products, line search)

Self-concordance: motivation

drawbacks of classical analysis of Newton's method:

- Newton's method is affinely invariant, but convergence analysis is not
- never know m, M, L in practice
- m, M, L can depend on starting point

Nesterov & Nemirovsky's analysis of Newton's method

- is affinely invariant
- involves no unknown constants
- is valid for many (but not all) functions f

Self-concordance: definition

(Nesterov & Nemirovsky)

- function $f : \mathbf{R} \to \mathbf{R}$ is self-concordant if

$$|f'''(t)| \leq 2f''(t)^{3/2}$$

- $f : \mathbf{R}^n \to \mathbf{R}$ is self-concordant if restriction to arbitrary line is self-concordant

SC condition

- limits third derivative in terms of second
- is affinely invariant: f is SC $\Longleftrightarrow$ $f(Tx)$ is SC

examples of self-concordant fcts

- linear, (convex) quadratic functions
- $-\log x$ on $\{x|x > 0\}$
- $-\log\det X$ on $\{X|X = X^T \succ 0\}$
- $-\log(t^2 - x^T x)$ on $\{\ (x,t) \mid \|x\| < t\ \}$

simple properties:

- affine transformation of domain: f SC $\Longrightarrow$ $g(z) = f(Az+b)$ SC
- sums: f, $\tilde{f}$ SC $\Longrightarrow$ $f + \tilde{f}$ SC
- scaling: f SC, $\alpha \geq 1 \Longrightarrow \alpha f$ SC

hence, *e.g.*,

- $-\Sigma_{i=1}\log(b_i - a_i^T x)$ is SC
- $-\log\det\left(F_0 + x_1 F_1 + \cdots + x_n F_n\right)$ is SC

Convergence analysis via SC

Newton method, with backtracking or exact line search:

$$\#\text{iterations} \leq \frac{f(x^{(0)}) - f^\star}{\eta_2} + \log_2 \log_2(2/\epsilon)$$

where η_2 depends only on backtracking parameters:

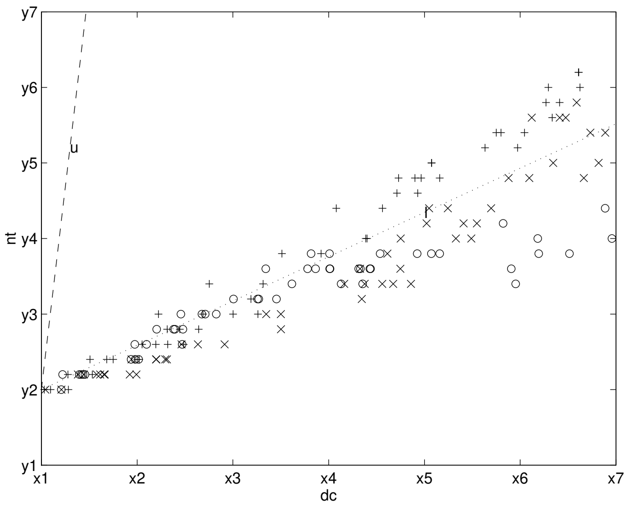
$$\eta_2 = \beta \frac{\alpha(1/2 - \alpha)^2}{5 - 2\alpha}$$

e.g., for $\alpha = 0.2$, $\beta = 0.7$, we have $1/\eta_2 \approx 365$

(a more refined analysis yields smaller bound)

example:

$$f(x) = \log\det\left(F_0 + x_1 F_1 + \cdots + x_n F_n\right)^{-1}$$



conclusion:

- $f(x^{(0)}) - f^\star$ gives upper bound on #iterations
- $f(x^{(0)}) - f^\star$ is also good measure in practice