Numerical Methods for Nonconvex Optimization

- that are applicable when f is smooth : TODAY
- that are applicable when f is nonsmooth : Lec 13.

So, assume f is smooth (but not necessarily convex).

Two classes of method

LINE SEARCH (Armijo cond + Wolfe cond.)

TRUST REGION : won't discuss. See Nocedal+Wright (NW).

In all cases, looking for local minimizers : global optimization in nonconvex case is HARD!

LINE SEARCH METHODS

1. Newton's method : if $H = \nabla^2 f(x)$ is not positive definite then $d = -H^{-1}\nabla f(x)$ may not be a descent direction:
$d^T \nabla f(x) = -d^T H d$ which could be +ve

What to do ?

(a) compute it anyway*, + use convex comb. of Newton + gradient directions which can choose to be a descent direction.

(b) use Cholesky to factor H, + if it breaks down (which means $H \not\succ 0$), add a multiple of I to H + repeat

(c) use "modified Cholesky factorization" of Gill, Murray+Wright: see N+W. Not available as a standard code in Matlab.

(d) use "eig" to compute eigenvalues of H, + replace negative or zero eigenvalues by small +ve numbers. This results in a direction dominated by the "negative curvature" directions. But it's several times the cost of Cholesky.

*a factorization that exploits symmetry but does not assume that $H \succ 0$ is the LDL^T factorization, where L has 1×1 and 2×2 blocks (Bunch-Parlett-Kaufman) - see the "help \" info in Matlab.

CAN USE BACKTRACKING LINE SEARCH FOR ALL OF THESE.

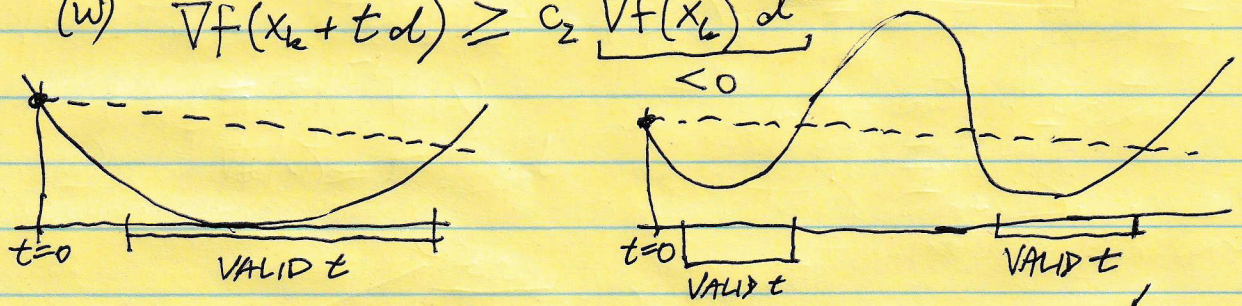
2. Quasi-Newton + Conjugate Gradient Methods.
For these need a more sophisticated "ARMIJO-WOLFE" line search.
Recall the "Armijo condition". Let d = "search direction" Δx.*

(A) $f(x_k + t d) \leq f(x_k) + c_1 t \nabla f(x_k)^T d$

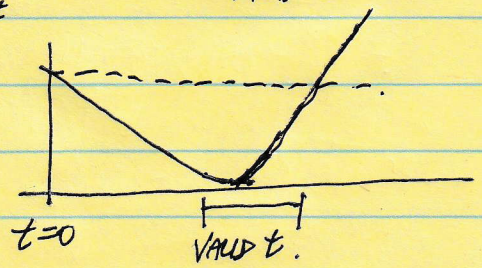
↖ called α earlier, $\in (0, \frac{1}{2})$

New condition: "Wolfe condition"

(W) $\nabla f(x_k + t d) \geq c_2 \underbrace{\nabla f(x_k)^T d}_{<0}$



(A) ensures t not too large
(W) ensures t not too small

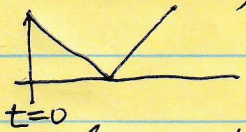


Require $0 < c_1 < c_2 < 1$.

Often, a "strong Wolfe" condition is recommended instead

(SW) $|\nabla f(x_k + t d)| \leq c_2 |\nabla f(x_k)^T d|$ (see N+W)

But this can greatly reduce the range of acceptable steps, makes it potentially impossible if f is non-smooth,



and makes writing a valid line search code MUCH MORE COMPLICATED: NW Ch. 3, section 6 is almost unreadable for this reason.

*Assume d is a DESCENT DIRECTION: $\nabla f(x_k)^T d < 0$.
(from x_k)

A Simple Bracketing Armijo-Wolfe Line Search

Set $t \leftarrow 1$, done $\leftarrow$ false, $\alpha \leftarrow 0$, $\beta \leftarrow \infty$ ← UPPER BOUND.
← LOWER BOUND

while not done

$x \leftarrow x_k + td$ (d is "descent direction")

if $f(x) > f(x_k) + c_1 t \nabla f(x_k)^T d$

$\beta \leftarrow t$: (A) condition violated, t too big

elseif $\nabla f(x)^T d < c_2 \nabla f(x_k)^T d$

$\alpha \leftarrow t$: (W) condition violated, t too small

else

$\alpha \leftarrow t$, $\beta \leftarrow t$, done $\leftarrow$ true

end.

(set up next function evaluation)

if $\beta < \alpha$

$t \leftarrow (\alpha+\beta)/2$ (bisection)

else

$t \leftarrow 2t$ (double)

end

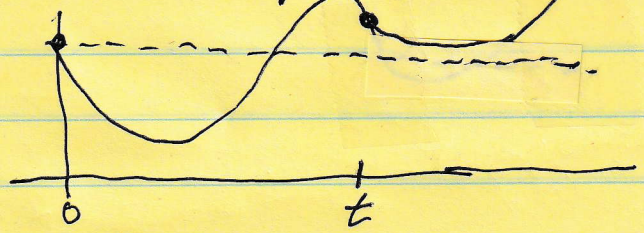
code: linesch_ww.m in HANSO (see my webpage)

"weak Wolfe" (N&W) = "Armijo Wolfe".

Update chosen so $[\alpha, \beta]$ always contains a valid A-W step t.

Essential that violation of 1st condition is checked first

e.g.



both conditions violated.

Update upper bound β, not lower bound α

This idea goes back to Powell (1976).

Thm If f is C^1 and bounded below on $\{x_k + td : t \geq 0\}$, then the line search terminates with a valid A-W step.

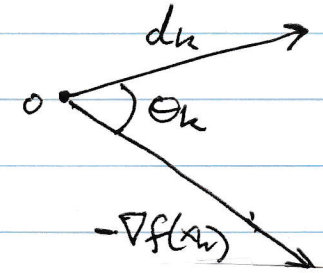
For an analysis under much weaker assumptions on f, e.g. f locally Lipschitz but not nec. differentiable, see LEWIS & OVERTON 2013.

ZOUTENDIJK'S THEOREM

Assume f is bd below, $f \in C^1$ and ∇f is Lipschitz on $\{x : f(x) \le f(x_0)\}$. (Lip const. L)

Define a descent algorithm:

$$x_{k+1} = x_k + t_k d_k$$



e.g. $d_k = -\nabla f(x_k)$, $\theta_k = 0$.

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

where t_k satisfies the Armijo-Wolfe conditions.

Then
$$\sum_{k=0}^{\infty} (\cos\theta_k)^2 \, \|\nabla f(x_k)\|^2 < \infty \qquad (*)$$

so, if $\cos\theta_k \ge \tau > 0$ for all k $\left(\theta_k \le \psi < \frac{\pi}{2}\right)$

we have $\nabla f(x_k) \to 0$.

Pf Write g_k for $\nabla f(x_k)$, f_k for $f(x_k)$.

We have from the Wolfe condition that

$$g_{k+1}^T d_k \ge c_2 g_k^T d_k$$

so
$$(g_{k+1} - g_k)^T d_k \ge (c_2 - 1) g_k^T d_k$$

so
$$L \|t_k d_k\| \, \|d_k\| \ge (c_2 - 1) g_k^T d_k$$

so
$$t_k \ge \frac{c_2 - 1}{L} g_k^T d_k / \|d_k\|^2 \qquad \text{(product of 2 neg. numbers)}$$

Substitute this into the Armijo condition:

lower bd on t_k $\qquad$ <0

$$f_{k+1} \le f_k + c_1 \left[\frac{(c_2-1)}{L} \frac{g_k^T d_k}{\|d_k\|^2} \right] g_k^T d_k \cdot \frac{\|g_k\|^2}{\|g_k\|^2}$$

or

$$f_{k+1} \le f_k - K (\cos\theta_k)^2 \|g_k\|^2$$

$\nwarrow \frac{c_1(1-c_2)}{L}$

Sum over $j \le k$

$$f_{k+1} \le f_0 - K \sum_{j=0}^{k} (\cos\theta_j)^2 \|g_j\|^2$$

Since f is bd below, let $k \to \infty \Rightarrow$ (*).

Zoutendijk: Thm applies to any "Newton-like" method

$$d_k = -H_k^{-1} \nabla f(x_k) \quad \text{as long as } H_k \text{ is}$$

$\rightarrow$ UNIFORMLY POSITIVE DEF.

However, hard to prove this for QN methods — and not true for CG methods.

Rate of convergence of Gradient Method: as in convex case, slow.

_____ of Newton's Method: as before, quadratic under regularity assumptions, but no guarantees can be made re # iters.

QUASI-NEWTON METHODS.

Motivation: Newton's method is $O(n^3)$ work.
Want to update an approx. to [FACTORIZATION OR INVERSE] of $\nabla^2 f(x)$ in $O(n^2)$ time.

How? Make better use of gradient info.
After line search, we have:

x_k $\quad$ $g_k \equiv \nabla f(x_k)$

x_{k+1} $\quad$ $g_{k+1} = \nabla f(x_{k+1})$.

Let $s_k = x_{k+1} - x_k = t_k d_k$

$y_k = g_{k+1} - g_k$

From Fund. Thm. of Calc.,

$$y_k = \int_0^1 \nabla^2 f(x_k + \tau s_k) s_k \, d\tau$$

$$= \underbrace{\left[\int_0^1 \nabla^2 f(x_k + \tau s_k) \, d\tau\right]}_{G_k} s_k$$

G_k "average Hessian along s_k".

So it seems reasonable that our new approx to $\nabla^2 f(x_{k+1})$, say B_{k+1}, should satisfy

$$B_{k+1} s_k = y_k$$

THE SECANT EQUATION.

or, if we are approx $\nabla^2 f(x_{k+1})^{-1}$ by, say, C_k

$$C_{k+1} y_k = s_k$$

Davidon-Fletcher-Powell 1959 1963.

Various choice known: PSB, DFP, BFGS

Broyden-Fletcher-Goldfarb-Shanno 1970

BFGS:

$$C_{k+1} = (I - \gamma_k s_k y_k^T) C_k (I - \gamma_k y_k s_k^T) + \gamma_k s_k s_k^T$$

where $\gamma_k = \frac{1}{s_k^T y_k}$.

Check that $C_{k+1} y_k =$

$$(I - \gamma_k s_k y_k^T) C_k (y_k - \gamma_k y_k s_k^T y_k) + \gamma_k s_k s_k^T y_k$$

$y_k - \gamma_k y_k s_k^T y_k = 0$ ✓.

How much work is needed to compute C_{k+1}? ASK

⟹ HOW DO WE KNOW $s_k^T y_k > 0$? DIRECTLY FROM THE WOLFE CONDITION (see next page).

THM (Powell, 1976)

Suppose 1. $f \in C^2$

2. $\Omega = \{x : f(x) \le f(x_0)\}$ is convex

3. $\exists m, M$ s.t.

STRONG CONVEXITY

$$m\|z\|^2 \le z^T \nabla^2 f(x) z \le M\|z\|^2$$

$\forall z \in \mathbb{R}^n$, $x \in \Omega$

Then $\{x_k\}$ generated by BFGS with Armijo-Wolfe line search satisfies $x_k \to$ UNIQUE LOCAL MINIMIZER OF f. (exists!).

(x^*)

Pf: beautiful, 2 pages + 3 onterdijk. See Nocedal+Wright.

HARD PART IS SHOWING EIGS(C_k) REMAIN BOUNDED AND (AND $\ge \delta > 0$.

Clarifying why $s_k^T y_k > 0$

$g_{k+1}^T d_k \geq c_2 \underbrace{g_k^T d_k}_{<0}$ (Wolfe)

$y_k^T d_k = g_{k+1}^T d_k - g_k^T d_k \geq (c_2 - 1) g_k^T d_k > 0.$

$s_k = t_k d_k$

$\Rightarrow \quad s_k^T y_k = y_k^T s_k \geq (c_2 - 1) g_k^T s_k > 0.$

Why update C_{k+1}, approx to $\nabla^2 f(x_{k+1})^{-1}$, instead of B_{k+1}, approx to $\nabla^2 f(x_{k+1})$. ASK.

Answer: then $d_{k+1} = C_{k+1} \nabla f(x_{k+1})$ is only $O(n^2)$.

Alternative update Cholesky factor of $\nabla^2 f(x_{k+1})$.
See Dennis + Schnabel's textbook. $\rightarrow$ can solve for d_{k+1} in $O(n^2)$ time.
Nice idea, used to be thought to be more stable, but not clear there is any advantage.

Superlinear Convergence (Dennis + Moré).

If we also assume $\nabla^2 f$ is Lipschitz, then $x_k \to x^*$ superlinearly, i.e.

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

Pf: See N&W.

NONCONVEX CASE: Nothing known in theory (i.e. there are pathological counterexamples)
But in practice, always works! Like st. desc.
$\|\nabla f(x_k)\| \to 0$ CLUSTER POINTS ARE STATIONARY
But UNLIKE ST. D. convergence is superlinear.

Methods that are O(n)

LIMITED MEMORY BFGS — for more efficient: O(n) see Noc + Wri. But not sup. conv.

NONLINEAR CG Set $d_0 = -\nabla f(x_0) = -g_0$.

$$x_{k+1} = x_k + t_k d_k$$

$$d_{k+1} = \underbrace{-g_{k+1}}_{(-\nabla f(x_{k+1}))} + \beta_{k+1} d_k$$

AS IN "LINEAR" CG

Fletcher Reeves $\quad \beta_{k+1}^{FR} = \dfrac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$

Polak-Ribiere $\quad \beta_{k+1}^{PR} = \dfrac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}$

In both cases reduce to "linear" CG when f is quadratic.

"Linear CG" ≡ solve $Ax=b$, $A \succ 0$

Theorem If $f(x) = x^T A x - b^T x$ $\quad A \succ 0$

and we use $t_k \leftarrow$ exact line search. (minimizer of quadratic along line)

then CG ≡ BFGS

and terminates in n steps. (More in presence of rounding & ill-conditioning of A).

↖ OR FEWER: #DISTINCT EIGS(A).

When n is large, want good approx in much fewer than n steps.

Convergence is like $\left(1 - \sqrt{\frac{m}{M}}\right)^k$ compared to $\left(1 - \frac{m}{M}\right)^k$ for steepest descent.

Nonlinear Case (Nonquadratic)

To get convergence result for Nonlinear CG, need to use a "strong Wolfe" line search to prove that FR converges — but it's generally inferior to PR. (See Nocedal/Wright)

A variant: CG-PR-FR

$$\beta_{k+1} = \begin{cases} -\beta_{k+1}^{FR} & \text{if } \beta_{k+1}^{PR} < -\beta_{k+1}^{FR} \\ \beta_{k+1}^{PR} & \text{otherwise} \\ \beta_{k+1}^{FR} & \text{if } \beta_{k+1}^{PR} > \beta_{k+1}^{FR} \end{cases}$$

i.e. β_{k+1} is projection of β_{k+1}^{PR} onto $\left[-\beta_{k+1}^{FR}, \beta_{k+1}^{FR}\right]$

is a good compromise: as good or better than PR in practice, same conv. theory as FR.

Recent work: variants that use only weak Wolfe.

Nonlinearly Constrained Optimization

A huge area.
Two big classes of algs.

SQP (sequential quadratic programming)
IP. (interior point)

See N+W.
Software SNOPT
IPOPT.

Margaret Wright will teach a course that treats these in depth in ~~Fall~~ 2029.