Numerical Methods / NonConvex Optimization
- that are applicable when $f$ is smooth but nonconvex: TODAY
- that """" $f$ is nonsmooth / nonconvex: NEXT WEEK

Assume $f$ smooth (but not necessarily convex).

Two classes of methods.
LINE SEARCH (Armijo + Weak or Strong Wolfe)
TEST REGION — won’t discuss, see N+W (Nocedal & Wright)

LINE SEARCH Methods
1. Newton’s Method.
   Can continue to use backtracking line search.
   but need to modify $H = V^2 f(x)$ if $V$ is not positive definite — because otherwise $d = -H^{-1} Vf(x)$ may not be a descent direction: $d^T Vf(x) = -d^T H d$

   How?
   (a) Use chol, repeat with adding multiple of $I$ to $H$
   if necessary
   (b) "Modified Cholesky" factorization of $GMW$.
      See N+W. A bit complicated, not built-in to Matlab.
   (c) Use eig. A bit expensive.
   (d) Use $H = \text{symmetric indef. fact} = L L^T$
      where $B$ has $1 \times 1$ and $2 \times 2$ blocks.
      (Burach-Friedt-Kaufman)

2. QN and CG Methods (Quasi-Newton + Conjugate gradient).
   We need a more sophisticated line search.
Weak Wolfe line search

Much less complicated than "strong Wolfe"!

And, useful in non-smooth functions as well as smooth.

Recall we want to meet that (ARMijo condition)

\[ f(x_0 + t d) \leq f(x_0) + c_1 t \nabla f(x_0)^T d \quad \text{(so don't go too far)} \]

New condition (WEAK WOLFE)

\[ \nabla f(x_0 + t d)^T d \geq c_2 \frac{\nabla f(x_0)^T d}{d} \quad \text{(so go far enough)} \]

Two scenarios:

To ensure existence of such point in smooth case, we assume \( c_2 > c_1 > 0 \).

\[ \text{Normally take small} \]

\[ \text{not too small, otherwise too demanding} \]

In non-smooth case, might like to take \( c_2 = 0 \),

so \( \nabla f(x_0 + t d) \) changes sign.

But this is too restrictive in smooth case -

for example, can prevent superlinear convergence [TF65].

We will update an interval \([x, \beta]\) that brackets a point satisfying the W.W. condition.

Initially \( x = 0, \beta = \infty \).

[* Strong Wolfe condition is \( |\nabla f(x_0 + t d)| \leq c_2 |\nabla f(x_0)^T d| \). See NW. *]
Weisz Wolfe LS Alg. $t \leq 1$

While not done

$x \leftarrow x_0 + t d$

if $f(x) > f(x_0) + c_1 t \nabla f(x_0)^T d$

$\beta \leftarrow t$ \hspace{1cm} % 1st condition violated, go to for

else if $\nabla f(x)^T d < c_2 \nabla f(x_0)^T d$

$x \leftarrow t$ \hspace{1cm} % 2nd condition violated, not far enough

else

$2 \leq t$

$\beta \leftarrow t$

end. STOP

To set up next function evaluation

if $\beta < 2$

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

else

$t \leftarrow 2t$

end.

It is crucial that violation of the 1st condition is checked 1st.

Both conditions violated.
Update $\beta$, NOT $x$. 

$\Delta$
ZOUTENDIJK'S THEOREM

Assume \( f \) is ad below, \( f \in C^1 \) and \( \nabla f \in \text{Lip}(1) \),
on \( \{ x : f(x) \leq f(x_0) \} \).

Define a descent algorithm:

\[ x_{k+1} = x_k + t_k d_k \]

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

\[ c_k \theta_k = -\nabla f(x_k)^T d_k \]
\[ \nabla f(x_k) \| d_k \| \]

where the satisfy the Weak Wolfe+Armijo conditions.

\[ \sum_{k=0}^{\infty} \left( \cos \theta_k \right)^2 \| \nabla f(x_k) \|^2 < \infty \] (4)

\( \Rightarrow \cos \theta_k \geq \gamma > 0 \) for all \( k \)
\( (\theta_k < \psi < \frac{\pi}{2}) \)

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

If \( \theta_k \in [0, \pi] \), \( \theta_k \in [0, \pi] \).
We have from the Wolfe condition that

\[ g_k^T d_k \geq C_2 g_k^T \]

\[ (g_{k+1} - g_k)^T d_k \geq (C_2 - 1) g_k^T d_k \]

\[ - \| g_k \| d_k \| d_k \| \geq (C_2 - 1) g_k^T d_k \]

\[ t_k \geq \frac{C_2 - 1}{\| d_k \|^2} \text{ (product of 2 neg. ineq.)} \]
Substitute this into the Armijo condition:

\[ f_{k+1} \leq f_k + c_1 (c_2 - 1) \frac{g_k^T \Delta x_k}{\|\Delta x_k\|^2} \frac{g_k^T d_k}{\|d_k\|^2} \|g_k - h_k\|^2 \]

or

\[ f_{k+1} \leq f_k - K \left( \cos \theta_k \right)^2 \|g_k\|^2 \frac{c_1 (1 - \theta_k)}{L} \]

sum over \( j \leq k \)

\[ f_{k+1} \leq f_0 - K \sum_{j=0}^{k} \left( \cos \theta_j \right)^2 \|g_j\|^2 \]

since \( f \) is bd below, let \( k \to \infty \Rightarrow (\star) \)

Furthermore, this applies to any "Newton-like" method

\[ d_k = H_k^{-1} \nabla f(x_k) \]

as long as \( H_k \) is uniformly positive def.

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

Rate of convergence of gradient method: as in convex case, slow.

\[ \frac{1}{2} \frac{\|\nabla f(x_k)\|^2}{\lambda_{\min} \lambda_{\max}} \]

Rate of convergence of Newton's Method: as before, quadratic under regularity assumption, but no guarantees can be made on iterates.
**QUASI-NEWTON METHODS.**

Motivation: Newton's method is $O(n^3)$ work.

Want to update an approx. to $\mathbf{V}^2 f(x)$ in $O(n^2)$ time.

How? Make better use of gradient info.

After line search, we have:

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

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

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

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

From Fund. Theor. of Calc.,

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

$$= \left[ \int_0^1 \nabla^2 f(x_k + r s_k) \, dr \right] s_k$$

$G_k$ "average Hessian along $s_k$".

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

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

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

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

THE SECANT EQUATION.
Various choice known: BFGS, DFP, BFGS

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

where \( y_k = \frac{1}{s_k} s_k^T y_k \).

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

\[ \text{How much work is needed to compute } C_{k+1}? \text{ Ask} \]

\[ \rightarrow \text{How do we know } s_k^T y_k > 0? \text{ Directly from the Wolfe condition (see next page).} \]

\[ \text{THM (Powell, 1976)} \]

1. \( f \in C^2 \)
2. \( \mathcal{D} = \{ x : f(x) \leq f(x_0) \} \) is convex
3. \( \exists m, M > 0 \)

\[ \text{Strong convexity} \quad m \| z \|^2 \leq z^T \nabla^2 f(x) z \leq M \| z \|^2 \]

\[ \forall z \in \mathbb{R}^n, x \in \mathcal{D} \]

Then \( \{ x_k \} \) generated by BFGS with Armijo-Wolfe line search

satifies \( x_k \to \text{unique local minimizer of } f \) (exists).

If: beautiful, 2 pages, Zoutendijk

See Nocedal+Wright

[Hard part is showing \( \text{Eigs}(C_k) \) remain bounded and \( \gamma > 0 \).]
Claiming why $s_n y_n > 0$

\[ g_{k+1}^T d_k \geq c_r g_k^T d_k \]  \hspace{1cm} (Wolfe)

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

\[ s_k = t_k d_k \]

\[ s_k^T y_k = y_k^T s_k \geq (c_r - 1) g_k^T s_k > 0. \]
Superlinear Convergence (Dennis & Moré).

If vector sequence \( \mathbf{x}_k \) is Lipschitz, then

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

**Proof:** see now.

**Nonconvex Case:** Nothing known in theory. But in practice, always works! Like at deep.

**Methods that are \( O(n) \)**

**Limited Memory BFGS** — far more efficient: \( O(n) \)

**Nonlinear CG**

\( \mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{d}_k \)

\( \mathbf{d}_{k+1} = -\mathbf{V}(\mathbf{x}_k) \mathbf{d}_k \)

**Fletcher-Reeves**

\[
\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_{k}^T \mathbf{g}_{k}}
\]

**Polak-Ribière**

\[
\beta_{k+1} = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_{k}^T \mathbf{g}_{k}}
\]

In both cases reduce to "linear" CG when quadratic.
"Linear CG" \( \equiv \) solve \( Ax = b, A \succ 0 \)

\[ f(x) = x^TAx - b^Tx \quad A \succ 0 \]

and we use the exact line search. (minimum of quadratic along line)

then CG \( \equiv \) BFGS

at terminates in \( n \) steps. (More in presence of rounding or fewer \#distinct eigv(A), ill-conditioning of A)

when \( n \) is large, want good approximation of original in fewer than \( n \) steps:

Convergence is like \( (1 - \frac{\mu}{m})^k \text{ vs } (1 - \frac{\mu}{\mu A})^k \text{ in steepest descent.} \)

\[ \text{Nonlinear Case (nonquadratic)} \]

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 FR. (See Nocedal)

A variant: CG FR FR

\[ \beta_{k+1} = \begin{cases} -\beta_k & \text{if } \beta_{k+1} < -\beta_k \\ \beta_{k+1} + \beta_{k+1} & \text{otherwise} \\ \beta_{k+1} & \text{if } \beta_{k+1} > \beta_{k+1} \end{cases} \]

i.e. \( \beta_{k+1} \) is projection of \( \beta_{k+1} \) onto \( [-\beta_{k+1}, \beta_{k+1}] \)

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

Recent work: variants that use only weak Wolfe,
Nonlinearly Constrained Optimization

A huge area.
Two big classes of algo.

SQP
IP

See N+W.
Software SNOPT
IPOPT

Margaret Wright will teach a course that treats these in depth in Fall 2016.