Quasi-Newton methods:
Symmetric rank 1 (SR1)
Broyden–Fletcher–Goldfarb–Shanno (BFGS)
Limited memory BFGS (L-BFGS)

February 6, 2014
Roadmap

<table>
<thead>
<tr>
<th></th>
<th>Line-search</th>
<th>Trust-region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>$p_k = -\nabla f_k + \beta_k p_{k-1}$</td>
<td></td>
</tr>
<tr>
<td>Newton</td>
<td>$p_k = -[\nabla^2 f_k]^{-1} \nabla f_k$</td>
<td>$m_k(p) = f_k + \nabla f_k^T p + 0.5 p^T \nabla^2 f_k p$</td>
</tr>
<tr>
<td>q-Newton</td>
<td>$p_k = -B_k^{-1} \nabla f_k$</td>
<td>$m_k(p) = f_k + \nabla f_k^T p + 0.5 p^T B_k p$</td>
</tr>
</tbody>
</table>

Today:
How to select $B_k \approx \nabla^2 f_k$ for fast convergence of algorithms?

Tomorrow:
How to cheaply approximately solve Newton’s subproblem & preserve fast convergence?
Idea

Taylor:

$$\nabla f(x_{k+1}) - \nabla f(x_k) = \frac{\nabla^2 f(x_k + t(x_{k+1} - x_k))}{x_{k+1} - x_k} \approx B_{k+1}$$

Secant equation

$$s_k := x_{k+1} - x_k$$
$$y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$$
$$B_{k+1}s_k = y_k$$

$n$ equations in $(n + 1)n/2$ unknowns!
SR1 method

Ansatz:

\[ B_{k+1} = B_k + \sigma vv^T, \quad \sigma = \pm 1 \]
SR1 method

\[
B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}, \quad \text{when } (y_k - B_k s_k)^T s_k \neq 0
\]

Sherman–Morrison

\[
\tilde{A} = A + ab^T \quad \implies \quad \tilde{A}^{-1} = A^{-1} - \frac{A^{-1}ab^TA^{-1}A^{-1}}{1 + b^TA^{-1}a},
\]

provided \( A, \tilde{A} \) are non-singular.

\[
H_{k+1} = B_{k+1}^{-1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}
\]
SR1: cases

1. If \((y_k - B_k s_k)^T s_k \neq 0\) use SR1 update formula
2. If \(y_k = B_k s_k \implies B_{k+1} = B_k\)
3. If \(y_k \neq B_k s_k\) and \((y_k - B_k s_k)^T s_k = 0 \implies\) no SR1 update! Skip the update in this case \((B_{k+1} = B_k)\)

Note:

- \(B_k\) does not have to be positive definite \(\implies\)
  - Good Hessian approximation for non-convex problems
  - Better suited for TR than LS approach
SR1: convergence

**Theorem 6.1, N&W**

Exact in $n$ steps on convex quadratic functions (if updates never skipped)
SR1: convergence

**Theorem 6.2, N&W**

- \( x_k \to x^* \)
- \( \nabla^2 f \) is bounded & Lipschitz
- \( |(y_k - B_k s_k)^T s_k| \geq r ||y_k - B_k s_k|| ||s_k|| \) (updates never skipped)
- \( \{s_k\}\)-uniformly linearly independent

Then

\[
\lim_{k \to \infty} ||B_k - \nabla^2 f(x^*)|| = 0
\]
SR1: convergence

**Theorem 6.7, N&W**

Under restrictive assumptions (unique stationary point in the vicinity of the algorithmic iterations; updates never skipped; $B_k$ bounded; etc) $\Rightarrow$ convergence with $n$-step superlinear rate.

Good convergence in practice.
Rank-two update algorithms

Idea:

\[
\begin{align*}
\min_{B \in \mathbb{R}^{n \times n}} & \quad \| B - B_k \|, \\
\text{subject to} & \quad Bs_k = y_k, \\
& \quad B = B^T.
\end{align*}
\]

or

\[
\begin{align*}
\min_{H \in \mathbb{R}^{n \times n}} & \quad \| H - H_k \|, \\
\text{subject to} & \quad Hy_k = s_k, \\
& \quad H = H^T.
\end{align*}
\]
Rank-two update algorithms

Idea:

\[
\begin{align*}
\text{minimize} & \quad \| B - B_k \|, \\
\text{subject to} & \quad Bs_k = y_k, \\
& \quad B = B^T.
\end{align*}
\]

Take scaled Frobenius matrix norm:

\[
\| B - B_k \|^2 = \text{Tr}[W(B - B_k)W(B - B_k)^T],
\]

\[
W = W^T, \ W \succ 0, \ Wy_k = s_k.
\]

Result: Davidon–Fletcher–Powell (DFP) algorithm

\[
B_{k+1} = (I - \rho_k y_k s_k^T)B_k(I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T,
\]

\[
\rho_k = (y_k^T s_k)^{-1}.
\]
Davidon–Fletcher–Powell algorithm:

\[
B_{k+1} = (I - \rho_k y_k s_k^T)B_k(I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T,
\]

\[
H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k},
\]

\[
\rho_k = (y_k^T s_k)^{-1}.
\]

Rank-two update algorithms

Idea:

\[
\text{minimize } H \in \mathbb{R}^{n \times n} \| H - H_k \|, \\
\text{subject to } Hs_k = y_k, \\
H = H^T.
\]

Take scaled Frobenius matrix norm:

\[
\| H - H_k \|^2 = \text{Tr}[W(H - H_k)W(H - H_k)^T],
\]

\[W = W^T, \quad W \succ 0, \quad Ws_k = y_k.\]

Result: Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm

\[
H_{k+1} = (I - \rho_k s_k y_k^T)H_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T,
\]

\[
\rho_k = (y_k^T s_k)^{-1}.
\]
Broyden–Fletcher–Goldfarb–Shanno algorithm:

\[ H_{k+1} = (I - \rho_k s_k y_k^T)H_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \]

\[ B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \]

\[ \rho_k = (y_k^T s_k)^{-1}. \]
Linesearch framework:

Wolfe curvature condition:

\[ \nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k \]

\[ \Downarrow \]

\[ [\nabla f_{k+1}^T - \nabla f_k]^T p_k \geq (c_2 - 1) \nabla f_k^T p_k > 0. \]

Quasi-Newton methods: Symmetric rank 1 (SR1) Broyden–Fletcher–Goldfarb–Shanno (BFGS) Limited memory BFGS (L-BFGS)
BFGS/DFP + Wolfe linesearch:

Suppose $H_k \succ 0$.

Wolfe curvature condition $\implies y_k^T s_k > 0$.

$\forall z \in \mathbb{R}^n, z \neq 0$:

$$z^T H_{k+1} z = [(I - \rho_k y_k s_k^T)z_k]^T H_k [(I - \rho_k y_k s_k^T)z_k] + [s_k^T z]^2 / (y_k^T s_k) \geq 0.$$ 

If $s_k^T z \neq 0$ $\implies z^T H_{k+1} z^T \geq [s_k^T z]^2 / (y_k^T s_k) > 0$.

If $s_k^T z = 0$ $\implies z^T H_{k+1} z = z^T H_k z > 0$.

Therefore $H_{k+1} \succ 0$! (Same for DFP)
BFGS/DFP + Wolfe linesearch:

\[ B_0 \succ 0 \quad (H_0 \succ 0) \quad + \text{ Wolfe linesearch} \]
\[ \downarrow \]
\[ B_k \succ 0 \quad (H_k \succ 0), \quad \forall k \]
\[ \downarrow \]
\[ p_k = -B_k^{-1} \nabla f_k = -H_k \nabla f_k \text{ is a descent direction (} \nabla f_k \neq 0). \]
BFGS: global convergence

**Theorem 6.4, N&W**

Exact in $n$ steps on convex quadratic functions; if $B_0 = I \iff$ same iterates as CG.
BFGS: global convergence

**Theorem 6.5, N&W**

1. $B_0 \succ 0$, $B_0 = B_0^T$
2. $f \in C^2$
3. $\exists m, M > 0: m\|z\|^2 \leq z^T \nabla^2 f(x)z \leq M\|z\|^2$

Then $x_k \rightarrow x^*$: $\nabla f(x^*) = 0$ & $\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty$. 

Quasi-Newton methods: Symmetric rank 1 (SR1) Broyden–Fletcher–Goldfarb–Shanno (BFGS) Limited memory BFGS (L-BFGS)

February 6, 2014 19 / 25
BFGS: local convergence

Theorem 6.6, N&W

1. $f \in C^2$
2. $\nabla f$ - Lipschitz
3. $\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty$.

Then $\lim_{k \to \infty} \|x_{k+1} - x^*\| / \|x_k - x^*\| = 0$. 
Presented quasi-Newton algorithms:

Good for small to medium size optimization problems with dense (or costly to compute) Hessian matrices

- Factorization/system solve cost for $[\nabla^2 f_k]^{-1} \nabla f_k$: $O(n^3)$
- Update + multiplication with $H_k$: $O(n^2)$
Limited-memory quasi-Newton algorithms:

**Idea**

- Storage of $B_k$ (or $H_k$): $O(n^2)$
- Instead: store $m \ll n$ pairs $(s_k, y_k)$: $O(2mn)$
Example: L-BFGS

\[ H_{k+1}q = [I - \rho_k y_k s_k^T]^T H_k[I - \rho_k y_k s_k^T]q + \rho_k s_k s_k^T q \]
\[ = [I - \rho_k s_k y_k^T] H_k[q - \{\rho_k s_k^T q\} y_k] + \{\rho_k s_k^T q\} s_k. \]

Algorithm (only one pair is stored)

1: \( \alpha_k := \rho_k s_k^T q \)
2: \( q := q - \alpha_k y_k \)
3: \( r := H_k q \)
4: \( \beta_k := \rho_k y_k^T r \)
5: \( r := r + (\alpha_k - \beta_k) s_k \)
Example: L-BFGS

1: \( q := \nabla f_{k+1} \)
2: for \( i = k, k - 1, \ldots, k - m + 1 \) do
3: \( \alpha_i := \rho_i s_i^T q \)
4: \( q := q - \alpha_i y_i \)
5: end for
6: \( r := H_k^0 q \)
7: for \( i = k - m + 1, \ldots, k - 1, k \) do
8: \( \beta_i := \rho_i y_i^T r \)
9: \( r := r + (\alpha_i - \beta_i) s_i \)
10: end for
11: Result: \( r \approx H_{k+1} \nabla f_{k+1} \)
Storage and step complexity revisited:

<table>
<thead>
<tr>
<th>Method</th>
<th>Storage</th>
<th>Step complexity</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>linear</td>
</tr>
<tr>
<td>BFGS/DFP/SR1</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>superlinear</td>
</tr>
<tr>
<td>L-BFGS</td>
<td>$O(mn)$</td>
<td>$O(mn)$</td>
<td>???</td>
</tr>
<tr>
<td>Newton (full Hess)</td>
<td>$O(n^2)$</td>
<td>$O(n^3)$</td>
<td>quadratic</td>
</tr>
<tr>
<td>Newton (sparse Hess)</td>
<td>$O(n) - O(n^2)$</td>
<td>$O(n) - O(n^3)$</td>
<td>quadratic</td>
</tr>
</tbody>
</table>

Note: step complexity does not account for complexities in evaluating function/derivatives!!!