Quasi-Newton Methods

Lecture 4, Continuous Optimisation
Oxford University Computing Laboratory, HT 2006
Notes by Dr Raphael Hauser (hauser@comlab.ox.ac.uk)
Recall from Lecture 3

Steepest-descent direction $d_k := -\nabla f(x_k)$.

- Takes $\sim n$ function evaluations (of $f$) to compute.
- Q-linear convergence.
Newton-Raphson direction  

\[ d_k = n_f(x_k) := -\left( D^2 f(x_k) \right)^{-1} \nabla f(x_k). \]

- Takes \( \sim n \) function evaluations to compute \( \nabla f(x_k) \) and \( \sim n^2 \) function evaluations to compute \( D^2 f(x_k) \).

- Once these matrices have been computed it takes \( O(n^3) \) computer operations to solve the following linear system for \( d_k \),

\[ D^2 f(x_k) d_k = -\nabla f(x_k). \]

- Q-quadratic convergence.
Ideally, one would like a search-direction that combines the cheapness of $-\nabla f(x_k)$ with the fast convergence of $n_f(x_k)$.

In reality, we need to strike a balance between work per iteration and convergence speed.

Quasi-Newton methods are clever mechanisms that achieve such a balance.
Let $C(f)$ be the cost of one function evaluation of $f$. Then the following shows the trade-off between computational cost and convergence speed,

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost per iteration</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steepest descent</td>
<td>$O(nC(f))$</td>
<td>Q-linear</td>
</tr>
<tr>
<td>Quasi-Newton</td>
<td>$O(n^2 + nC(f))$</td>
<td>Q-superlinear</td>
</tr>
<tr>
<td>Newton-Raphson</td>
<td>$O(n^3 + n^2C(f))$</td>
<td>Q-quadratic</td>
</tr>
</tbody>
</table>
Motivation of Quasi-Newton Updates:

The Newton-Raphson step is defined by

\[ x_{k+1} - x_k = n_f(x_k) = -\left(D^2f(x_k)\right)^{-1}\nabla f(x_k). \]

Assume an approximation \( B_k \approx D^2f(x_k) \) of the Hessian is available. Then an approximate Newton-Raphson step is given by the quasi-Newton update

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

This update is well-defined when \( B_k \) is nonsingular, and in particular when \( B_k \) is positive definite symmetric.
In this case the update is also motivated by the fact that

\[ x_k + d_k = x_k - B_k^{-1} \nabla f(x_k). \]

is the global minimiser of the following *quadratic model* of \( f \),

\[ p(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} (x - x_k)^\top B_k (x - x_k). \]
• $B_k$ is only an approximation of $D^2f(x_k)$. Therefore we use $d_k$ as a search direction rather than an exact update.

• A line-search then yields a new quasi-Newton iterate

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

• Q-N algorithms specify methods for cheaply computing a new approximate Hessian $B_{k+1} \simeq D^2f(x_{k+1})$. This computation should only use the quantities $B_k, \nabla f(x_k)$ and $\nabla f(x_{k+1})$. 
Algorithm 1: Generic Quasi-Newton Method.

**S0** Choose a starting point $x_0 \in \mathbb{R}^n$, a nonsingular $B_0 \in S^n$ (often the choice is $B_0 = I$), and a termination tolerance $\epsilon > 0$. Set $k = 0$.

**S1** If $\| \nabla f(x_k) \| \leq \epsilon$ then stop and output $x_k$ as an approximate local minimiser of $f$. Else go to **S2**.

**S2** Compute the quasi-Newton search direction $d_k = -B_k^{-1} \nabla f(x_k)$. 
S3 Perform a practical line-search for the minimisation of $\phi(\alpha) = f(x_k + \alpha d_k)$: find a step length $\alpha_k$ that satisfies the Wolfe conditions and compute the new iterate $x_{k+1} = x_k + \alpha_k d_k$.

S4 Compute the new approximate Hessian $B_{k+1}$ according to the specified rule.

S5 Replace $k$ by $k + 1$ and go to S1.
A Wish List of Properties of $B_k$

P1: $B_k$ should be nonsingular, so that $S2$ is well-defined.

P2: $B_k$ should be such that $d_k$ is a descent direction, so that $S3$ is well-defined.

P3: $B_k$ should be symmetric, as Hessians are symmetric matrices.
Properties P1–P3 can be satisfied by requiring that $B_k$ be positive definite symmetric: P1 and P3 are trivially true, and P2 follows from

$$\langle \nabla f(x_k), d_k \rangle = -\nabla f(x_k)^\top B_k^{-1} \nabla f(x_k) < 0,$$

unless $\nabla f(x_k) = 0$.

This also avoids that the quasi-Newton method gets attracted to any point but a local minimiser.

**Question:** Is this a problem when $D^2 f(x_k) \neq 0$?
The wish-list continues . . .

P4: \( B_{k+1} \) should be computable by “recycling” the quantities

\[
\nabla f(x_{k+1}), \nabla f(x_k), \ldots, \nabla f(x_0), d_k, \alpha_k \text{ and possibly } B_k.
\]

Crucial observation: the gradient change

\[
\gamma_k := \nabla f(x_{k+1}) - \nabla f(x_k)
\]

yields information about the Hessian change \( D^2 f(x_{k+1}) - D^2 f(x_k) \).

Let \( \delta_k := \alpha_k d_k \) be the chosen update.
The search direction $d_k$ was motivated by the fact that the gradient change predicted by the quadratic model

$$p(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2}(x - x_k)^T B_k(x - x_k)$$

is

$$\nabla f(x_k + d_k) - \nabla f(x_k) \approx \nabla p(x_k + d_k) - \nabla p(x_k)$$

$$= \nabla f(x_k) + B_k d_k - \nabla f(x_k)$$

$$= -\nabla f(x_k).$$

(1)

In other words, it is predicted that $x_k + d_k$ is exactly a stationary point of $f$. 
But $p$ is only a \textit{locally} valid model of $f$ and the new iterate $x_{k+1}$ is obtained via a line search.

The true gradient change

$$\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

differs from the prediction (1).
A clever way to incorporate \( \gamma_k \) into the Hessian approximations is to choose \( B_{k+1} \) so that the quadratic model

\[
h(x) = f(x_k) + \langle \nabla f(x_k), (x - x_k) \rangle + \frac{1}{2} (x - x_k) B_{k+1} (x - x)
\]

would have correctly predicted the observed gradient change:

\[
\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k) = \nabla h(x_{k+1}) - \nabla h(x_k) = \nabla f(x_k) + B_{k+1} \delta_k - \nabla f(x_k)
\]

In other words, \( B_{k+1} \) should be chosen such that

\[
B_{k+1} \delta_k = \gamma_k \tag{2}
\]

holds true. (2) is called the secant condition.
The wish-list continues . . .

P5: $B_{k+1}$ should be close to $B_k$ in a well-defined sense, so that $B_k$ can converge to $D^2 f(x^*)$ and $d_k$ is allowed to become the Newton-Raphson step asymptotically.

A straightforward idea to define a notion of closeness is by use of a matrix norm: $d(B_{k+1}, B_k) = \|B_{k+1} - B_k\|$. However, it is often more useful to characterise closeness by keeping the rank of $B_{k+1} - B_k$ as low as possible.
Low rank updates will automatically guarantee that the last property on our wish list is satisfied as well:

P6: The choice of $B_k$ should be such that the overall work per iteration is at most of order $O(n^2)$, to gain a substantial speed-up over the $O(n^3)$ computer operations needed to perform a Newton-Raphson step.
Symmetric Rank-1 Updates (SR1)

The method we are about to describe satisfies some but not all of the properties P1–P6.

P3 and P5 can be satisfied by requiring that $B_{k+1}$ is a rank-1 update of $B_k$: we want to select some vector $u$ and set

$$B_{k+1} = B_k + uu^T.$$  \hspace{2cm} (3)

If $B_0$ is symmetric, this guarantees that $B_k$ is symmetric for all $k$, and rank($B_{k+1} - B_k$) = 1.
The choice of $u$ is fixed when P4 is satisfied through the secant condition

$$B_{k+1} \delta_k = \gamma_k,$$  \hspace{1cm} (4)

where $\delta_k = x_{k+1} - x_k = \alpha_k d_k$ and $\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ as before.

Multiplying (3) by $\delta_k$ and substituting the result into (4), we find

$$(u^T \delta_k) u = \gamma_k - B_k \delta_k.$$  \hspace{1cm} (5)

Multiplying the transpose of this equation by $\delta_k$, we obtain

$$(u^T \delta_k)^2 = (\gamma_k - B_k \delta_k)^T \delta_k.$$  \hspace{1cm} (6)
Equation (5) shows that

\[ u = \frac{\gamma_k - B_k \delta_k}{u^T \delta_k}. \]

Therefore, (3) and (6) imply that the updating rule should be as follows,

\[
B_{k+1} = B_k + \frac{(\gamma_k - B_k \delta_k)(\gamma_k - B_k \delta_k)^T}{(u^T \delta_k)^2} (u^T \delta_k) \]

\[ = B_k + \frac{(\gamma_k - B_k \delta_k)(\gamma_k - B_k \delta_k)^T}{(\gamma_k - B_k \delta_k)^T \delta_k}. \] (7)

Note that since \( \gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k) \) and \( \delta_k = \alpha_k d_k \), we can compute the SR1 update from the “recycled” information referred to in P4.
When $B_{k+1}$ is computed via the updating rule (7) Algorithm 1 is called the *symmetric rank 1 method* (or SR1).

This method was independently suggested by Broyden, Davidson, Fiacco-McCormick, Murtagh-Sargent, and Wolfe in 1967-69.

The updates of the SR1 method are very simple to compute, but they have the drawback that $B_k$ is not always positive definite and $d_k$ might not always be defined or be a descent direction.

Moreover, $(\gamma_k - B_k \delta_k)^T \delta$ can be close to zero which leads to very large updates.
What about property P6?

- Once $d_k$ is known, computing $\alpha_k, x_{k+1}, \nabla f(x_{k+1}), \gamma_k$ and $\delta_k$ is very cheap.

- The total work for computing the updated matrix $B_{k+1}$ from $B_k$ and $d_k$ is of order $O(n^2)$.

- However, in order to compute $d_k$ we need to solve the linear system of equations

$$B_k d_k = -\nabla f(x_k),$$

which takes $O(n^3)$ time!
A way out of the dilemma . . .

**Theorem 1: Sherman–Morrison–Woodbury formula.** If $B \in \mathbb{R}^{n \times n}$ and $U, V \in \mathbb{R}^{n \times p}$ are matrices then

$$(B + UV^T)^{-1} = B^{-1} - B^{-1}U(I + V^TB^{-1}U)^{-1}V^TB^{-1}. $$

See the new problem set for a proof.
The usefulness of this formula is quickly understood:

- Suppose we knew \( H_k = B_k^{-1} \). Then, applying the Sherman-Morrison-Woodbury formula to \( B_+ = B_{k+1}, B = B_k, U = u = (\gamma_k - B_k \delta_k) \) and \( V = U^\top \) (that is, \( p = 1 \) in this case), we find

\[
H_{k+1} = (B_+)^{-1} = B^{-1} - B^{-1} u \left( 1 + u^\top B^{-1} u \right)^{-1} u^\top B^{-1} = H_k + \frac{(\delta_k - H_k \gamma_k)(\delta_k - H_k \gamma_k)^\top}{(\delta_k - H_k \gamma_k)^\top \gamma_k}.
\]
• Thus, $H_{k+1}$ is just a rank 1 update of $H_k$.

• Since we assumed $H_k$ known, computing $d_k = -H_k \nabla f(x_k)$ now takes only $O(n^2)$ work.

• Furthermore, $H_{k+1}$ is computed from $H_k$ in $O(n^2)$ time.
If the algorithm is started with $B_0 = I$, then $H_0 = I$ is known, and every iteration takes $O(n^2)$ work. $B_k$ need not be formed.

It is possible to analyse the local convergence of the SR1 method and show that the method converges superlinearly in a neighbourhood of a local minimiser of $f$.

Thus, if the SR1 method is properly implemented, it can combine convergence speeds similar to those of the Newton-Raphson method with a lower complexity.

However, $B_k$ is not guaranteed to stay positive definite, so P2 is not satisfied!
The Broyden-Fletcher-Goldfarb-Shanno Method:

BFGS updates are defined by

\[ B_{k+1} = B_k + \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k}. \]

- Rank-2 updates.
- Has all the properties of SR1, but stays positive definite if \( B_0 \succ 0 \).
- The most successful and widely used quasi-Newton method.
- Motivation more difficult, see Lecture Notes 4.
Reading Assignment: Download and read Lecture-Note 4.