C5: Quasi-Newton methods

Niclas Börlin

5DA001 Non-linear Optimization

Hessian approximations

- However, provided we use a global strategy, we may replace the Hessian in the Newton equation
  \[ \nabla^2 f(x_k) p = -\nabla f(x_k) \]
  with any positive definite matrix \( B_k \), i.e.
  \[ B_k p = -\nabla f(x_k) \]
  and still get global convergence.
- Some methods calculate Hessian approximations based on first- or second-order derivatives, e.g.
  - **Gauss-Newton**
    \[ B_k = J(x_k)^T J(x_k) \]
  - **Trust-region**
    \[ B_k = \nabla^2 f(x_k) + \lambda I \]
  - **Levenberg-Marquardt**
    \[ B_k = J(x_k)^T J(x_k) + \lambda I \]
- Other methods calculate their Hessian approximations without any derivative information. Two such methods are called Steepest Descent and Quasi-Newton.

Steepest Descent

- The **steepest descent** method uses the simplest Hessian approximation
  \[ B_k = I \Rightarrow p_k = -\nabla f(x_k) \]
  The “calculation” of the search direction is cheap.
- However, the convergence rate is linear with a convergence constant \( C \) bounded from above by
  \[ C_{\sup} = \left( \frac{\kappa(Q) - 1}{\kappa(Q) + 1} \right)^2 \]
  where \( Q = \nabla^2 f(x^*) \) is the true Hessian and \( \kappa(Q) \) is the condition number of \( Q \).
- The convergence rate will be poor even for moderate condition numbers.

<table>
<thead>
<tr>
<th>( \kappa(Q) )</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C )</td>
<td>0</td>
<td>0.6694</td>
<td>0.9608</td>
<td>0.9960</td>
<td>0.9996</td>
</tr>
</tbody>
</table>
- The steepest-descent method should only be used for problems that are known to be well-conditioned.
Example

- Gauss-Newton and the Steepest-descent method with linesearch $\mu = 0.1$ for the antelope problem ($\kappa(Q) \approx 500$):

Quasi-Newton methods

- Quasi-Newton methods use a sequence of symmetric positive definite matrices that approximate the Hessian (or the inverse Hessian).
- For every iteration, the next (inverse) Hessian approximation is calculated by updating the current approximation.
- Each update is constructed to include the curvature information computed in the last step, i.e. the next (inverse) Hessian should behave as the true (inverse) Hessian over the last step.
- This condition is called the Secant Condition.

The Secant Condition

- The one-dimensional Secant method uses the approximation

\[
\begin{align*}
\f''(x_k) & \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}, \\
\f''(x_k)(x_k - x_{k-1}) & \approx f'(x_k) - f'(x_{k-1}).
\end{align*}
\]

- In multiple dimensions this corresponds to

\[
\nabla^2 f(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1}).
\]

- From this we obtain the Secant equation

\[
B_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}).
\]

The Secant Condition

Cont’d

- With substitutions

\[
s_k = x_{k+1} - x_k, y_k = \nabla f(x_{k+1}) - \nabla f(x_k),
\]

we obtain

\[
B_{k+1}s_k = y_k.
\]

- If $H_k = B_k^{-1}$, the secant equation becomes

\[
H_{k+1}y_k = s_k.
\]

- The Hessian approximation $B_k$ has $n^2/2$ independent elements, but the secant equation has only $n$ equations.
- Thus, several updating schemes are possible.
Properties of Quasi-Newton methods

▶ An example of a Quasi-Newton update formula is

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

▶ This formula illustrates several key features with quasi-Newton approximations.
  ▶ The new approximation \( B_{k+1} \) is found by updating the old approximation \( B_k \).
  ▶ As a starting approximation \( B_0 = I \) may be used, but if a better approximation is available at a small cost, it should be used.

Properties of Quasi-Newton methods

Cont'd

▶ The number of operations needed to calculate the search direction depends on the choice of update formula.
  ▶ If a \( B_k \) update formula is used, the solution of the secant equation needs \( O(n^3) \) operations.
  ▶ Update formulas for the cholesky factors \( LL^T = B_k \) exist that reduce the number of operations to \( O(n^2) \).
  ▶ Another \( O(n^2) \) solution is if we use an update formula for \( H_k \). The search direction may then be calculated from

\[ \rho_k = H_k ( - \nabla f_k ). \]

▶ Thus, the total time complexity of one iteration of a Quasi-Newton method can be reduced to \( O(n^2) \) compared to \( O(n^3) \) for e.g. Newton and Trust-region methods.

Properties of Quasi-Newton methods

Cont'd

▶ The secant condition will be satisfied independently of how \( B_k \) is chosen:

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

▶ The new approximation \( B_{k+1} \) can be obtained using \( O(n^2) \) operations since the update only involves vector products.

Properties of Quasi-Newton methods

Cont'd

▶ Common Quasi-Newton methods

BGFS

▶ The most widely used Quasi-Newton formula is known as BFGS (Broyden-Fletcher-Goldfarb-Shanno):

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

▶ BFGS preserves symmetry and positive definiteness if \( y_k^T s_k > 0 \), which may be satisfied with a line search using the Wolfe condition.

▶ The corresponding update formula for \( H_k \) is

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

where \( \rho_k = 1/(y_k^T s_k) \).
Common Quasi-Newton methods

DFP

One of the first Quasi-Newton formulas was DFP (Davidon-Fletcher-Powell):

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

and

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

where \( \rho_k = 1/(y_k^T s_k) \).

Convergence properties

- The BFGS method has super-linear convergence, i.e. faster than linear but slower than quadratic.
- The deficit to quadratically convergent methods usually shows only in the few last iterations.
- Thus, for many practical applications, BFGS converges as fast as a Newton method, i.e. it requires approximately the same number of iterations.
- Since the BFGS search direction can be calculated in \( n^2 \) time vs. \( n^3 \) time for the Newton method, the required execution time is usually substantionally less than Newton.

Example

- Gauss-Newton and BFGS on the antelope problem, no line search.

\[ g(n) = \begin{cases} 2 & 1 \\ 2 & 2 \\ 4 & 2 \\ 6 & 2 \\ 8 & 3 \\ 10 & 4 \\ 12 & 5 \end{cases}, \]

\[ B_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 15 & 105 \\ 105 & 776 \end{bmatrix}, B_2 = \begin{bmatrix} 6 & 31 \\ 31 & 289 \end{bmatrix}, B_3 = \begin{bmatrix} 8 & 45 \\ 45 & 398 \end{bmatrix}, B_4 = \begin{bmatrix} 9 & 66 \\ 66 & 671 \end{bmatrix}, B_5 = \begin{bmatrix} 10 & 72 \\ 72 & 621 \end{bmatrix}, B_6 = \begin{bmatrix} 10 & 78 \\ 78 & 691 \end{bmatrix}, B_7 = \begin{bmatrix} 10 & 80 \\ 80 & 700 \end{bmatrix}, B_8 = \begin{bmatrix} 10 & 75 \\ 75 & 664 \end{bmatrix} \]

\[ \nabla f(x^*) = \begin{bmatrix} 10 \\ 74 \end{bmatrix}, \nabla^2 f(x^*) = \begin{bmatrix} 10 & 74 \\ 74 & 652 \end{bmatrix} \]

Termination rules

- Mathematically, a minimization algorithm should terminate with a solution \( x_k \) satisfying
  - the first order necessary conditions
    \[ \nabla f(x_k) = 0 \]
  - the second order necessary conditions
    \[ \nabla^2 f(x_k) \text{ positive semi-definite} \]

- Due to e.g. finite arithmetic, no algorithm is guaranteed to satisfy this condition in finite time.
- Instead, termination rules based on thresholds on e.g. \( \|\nabla f(x_k)\| \) are used.
Termination rules

Absolute criteria

▶ Consider the absolute termination criteria

\[ \|\nabla f(x_k)\| \leq \epsilon, \]

for some \( \epsilon > 0 \).

▶ This condition is scale dependent, since a change of units in \( f \) would rescale \( \nabla f \) and affect the strength of the condition.

▶ A change of units from e.g. mm to m corresponds to a scaling of \( 10^3 \).

Relative criteria

▶ A small modification of the absolute termination criteria leads the relative termination criteria

\[ \|\nabla f(x_k)\| \leq \epsilon |f(x_k)|, \]

which is not scale dependent.

▶ However, if \( f(x^*) \approx 0 \) the relative test will be difficult or impossible to satisfy due to round-off errors.

▶ A possible combination is

\[ \|\nabla f(x_k)\| \leq \epsilon (1 + |f(x_k)|). \]

▶ This test will behave like an absolute test if \( f(x_k) \approx 0 \) and otherwise like a relative test.

Termination rules

Least squares problems

▶ For least squares problems

\[ \min_x f(x) = \frac{1}{2} r(x)^T r(x), \]

the test

\[ \|f_r\| = \|J(J^T J)^{-1} J^T r\| \leq \epsilon (1 + \|r\|) \]

may be used instead of the gradient test.

▶ Since \( Jp \) and \( r \) belong to the same vector space, the test may be interpreted geometrically.

▶ The ratio

\[ \frac{\|f_r\|}{\|r\|} = \cos \alpha, \]

is related to the angle \( \alpha \) between the residual \( r \) and the tangent plane at \( r(x^*) \).

Termination rules

Least squares problems

▶ For least squares problems

\[ \min_x f(x) = \frac{1}{2} r(x)^T r(x), \]

the test

\[ \|f_r\| = \|J(J^T J)^{-1} J^T r\| \leq \epsilon (1 + \|r\|) \]

may be used instead of the gradient test.

▶ Since \( Jp \) and \( r \) belong to the same vector space, the test may be interpreted geometrically.

▶ Close to the solution the residual will approach orthogonality with the tangent plane, i.e.

\[ \alpha \to \pi/2 \text{ and } \frac{\|f_r\|}{\|r\|} \to 0. \]