The Dogleg and Steihaug Methods

Lecture 7, Continuous Optimisation
Oxford University Computing Laboratory, HT 2005
Dr Raphael Hauser (hauser@comlab.ox.ac.uk)

Variants of Trust-Region Methods:

0. Different choices of trust region R_k, for example using balls defined by the norms $\|\cdot\|_1$ or $\|\cdot\|_\infty$. Not further pursued.

I. Choosing the model function m_k. We chose
$$m_k(x) = f(x_k) + \nabla f(x_k)^\mathsf{T}(x - x_k) + \frac{1}{2}(x - x_k)^\mathsf{T} B_k (x - x_k).$$
Leaves choice in determining B_k. Further discussed below.

II. Approximate calculation of
$$y_{k+1} \approx \arg\min_{y \in R_k} m_k(y). \tag{1}$$
Further discussed below.

I. Choosing the Model Function

Trust-Region Newton Methods:

If the problem dimension is not too large, the choice
$$B_k = D^2 f(x_k)$$
is reasonable and leads to the 2nd order Taylor model
$$m_k(x) = f(x_k) + \nabla f(x_k)^\mathsf{T}(x - x_k) + \frac{1}{2}(x - x_k)^\mathsf{T} D^2 f(x_k)(x - x_k).$$

Methods based on this choice of model function are called *trust-region Newton methods*.

In a neighbourhood of a strict local minimiser TR-Newton methods take the full Newton-Raphson step and have therefore Q-quadratic convergence.

Trust-region Newton methods are not simply the Newton-Raphson method with an additional step-size restriction!

- TR-Newton is a descent method, whereas this is not guaranteed for Newton-Raphson.

- In TR-Newton, usually $y_{k+1} - x_k \not\sim -(D^2 f(x_k))^{-1}\nabla f(x_k)$, as y_{k+1} is not obtained via a line search but by optimising (1).

- In TR-Newton the update y_{k+1} is well-defined even when $D^2 f(x_k)$ is singular.

Trust-Region Quasi-Newton Methods:

When the problem dimension n is large, the natural choice for the model function m_k is to use quasi-Newton updates for the approximate Hessians B_k.

Such methods are called *trust-region quasi-Newton*.

In a neighbourhood of a strict local minimiser TR-quasi-Newton methods take the full quasi-Newton step and have therefore Q-superlinear convergence.

Differences between TR quasi-Newton and quasi-Newton line-search:

- In TR-quasi-Newton $B_k \not\succ 0$ is no problem, whereas in quasi-Newton line-search it prevents the quasi-Newton update $-B_k^{-1}\nabla f(x_k)$ from being a descent direction.

- In TR-Newton the update y_{k+1} is well-defined even when B_k is singular, while $-B_k^{-1}\nabla f(x_k)$ is not defined.

- In TR-quasi-Newton, usually $y_{k+1} - x_k \not\sim -B_k^{-1}\nabla f(x_k)$, as y_{k+1} is not obtained via a line search but by optimising (1).

II. Solving the Trust-Region Subproblem

The Dogleg Method:

This method is very simple and cheap to compute, but it works only when $B_k \succ 0$. Therefore, BFGS updates for B_k are a good, but the method is not applicable for SR1 updates.

Motivation: let

$$x(\Delta) := \arg \min_{\{x\in\mathbb{R}^n : \|x-x_k\|\leq\Delta\}} m_k(x).$$

If $B_k \succ 0$ then $\Delta \mapsto x(\Delta)$ describes a curvilinear path from $x(0) = x_k$ to the exact minimiser of the unconstrained problem $\min_{x\in\mathbb{R}^n} m_k(x)$, that is, to the quasi-Newton point

$$y_k^{qn} = x_k - B_k^{-1}\nabla f(x_k).$$

Idea:

- Replace the curvilinear path $\Delta \mapsto x(\Delta)$ by a polygonal path $\tau \mapsto y(\tau)$.

- Determine y_{k+1} as the minimiser of $m_k(y)$ among the points on the path $\{y(\tau) : \tau \geq 0\}$.

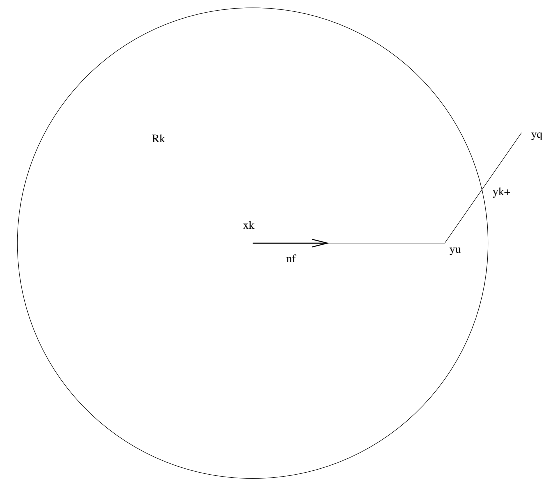
The simplest and most interesting version of such a method works with a polygon consisting of just two line segments, which reminds some people of the leg of a dog.

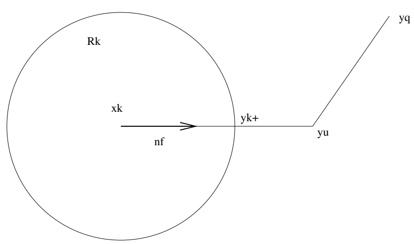
The "knee" of this leg is located at the steepest descent minimiser $y_k^u = x_k - \alpha_k^u \nabla f(x_k)$, where α_k^u is as in Lecture 6.

In Lecture 6 we saw that unless x_k is a stationary point, we have

$$y_k^u = x_k - \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^{\mathsf{T}} B_k \nabla f(x_k)} \nabla f(x_k).$$

From y_k^u the dogleg path continues along a straight line segment to the quasi-Newton minimiser y_k^{qn}.





The dogleg path is thus described by

$$y(\tau) = \begin{cases} x_k + \tau(y_k^u - x_k) & \text{for } \tau \in [0,1], \\ y_k^u + (1-\tau)(y_k^{qn} - y_k^u) & \text{for } \tau \in [1,2]. \end{cases} \tag{2}$$

Lemma 1: Let $B_k \succ 0$. Then

i) the model function m_k is strictly decreasing along the path $y(\tau)$,

ii) $\|y(\tau) - x_k\|$ is strictly increasing along the path $y(\tau)$,

iii) if $\Delta \geq \|B_k^{-1} \nabla f(x_k)\|$ then $y(\Delta) = y_k^{qn}$,

iv) if $\Delta \leq \|B_k^{-1} \nabla f(x_k)\|$ then $\|y(\Delta) - x_k\| = \Delta$,

v) the two paths $x(\Delta)$ and $y(\tau)$ have first order contact at x_k, that is, the derivatives at $\Delta = 0$ are co-linear:

$$\lim_{\Delta\to 0+} \frac{x(\Delta) - x_k}{\Delta} = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \sim \frac{-\|\nabla f(x_k)\|^2}{\nabla f(x_k)^\mathsf{T} B_k \nabla f(x_k)} \nabla f(x_k)$$
$$= \lim_{\tau\to 0+} \frac{y(\tau) - y(0)}{\tau}.$$

Proof: See Problem Set 4. □

Parts i) and ii) of the Lemma show that the dogleg minimiser y_{k+1} is easy to compute:

- If $y_k^{qn} \in R_k$ then $y_{k+1} = y_k^{qn}$.
- Otherwise y_{k+1} is the unique intersection point of the dogleg path with the trust-region boundary ∂R_k.

Algorithm 1: Dogleg Point.

compute y_k^u

if $\|y_k^u - x_k\| \geq \Delta_k$ stop with $y_{k+1} = x_k + \frac{\Delta_k}{\|y_k^u - x_k\|}(y_k^u - x_k)$ (*)

compute y_k^{qn}

if $\|y_k^{qn} - x_k\| \leq \Delta_k$ stop with $y_{k+1} = y_k^{qn}$

else begin

find τ^* s.t. $\|y_k^u + \tau^*(y_k^{qn} - y_k^u) - x_k\| = \Delta_k$

stop with $y_{k+1} = y_k^u + \tau^*(y_k^{qn} - y_k^u)$

end

Comments:

- If the algorithm stops in (*) then the dogleg minimiser lies on the first part of the leg and equals the Cauchy point.

- Otherwise the dogleg minimiser lies on the second part of the leg and is better than the Cauchy point.

- Therefore, we have $m_k(y_{k+1}) \leq m_k(y_k^c)$ as required for the convergence theorem of Lecture 6.

Steihaug's Method:

This is the most widely used method for the approximate solution of the trust-region subproblem.

The method works for quadratic models m_k defined by an arbitrary symmetric B_k. Positive definiteness is therefore not required and SR1 updates can be used for B_k.

It has all the good properties of the dogleg method and more ...

Idea:

- Draw the polygon traced by the iterates $x_k = z_0, z_1, \dots, z_j, \dots$ obtained by applying the conjugate gradient algorithm to the minimisation of the quadratic function $m_k(x)$ for as long as the updates are defined, i.e., as long as $d_j^\top B_k d_j > 0$.

- This terminates in the quasi-Newton point $z_n = y_k^{qn}$, unless $d_j^\top B_k d_j \leq 0$. In the second case, continue to draw the polygon from z_j to infinity along d_j, as m_k can be pushed to $-\infty$ along that path.

- Minimise m_k along this polygon and select y_{k+1} as the minimiser.

The polygon is constructed so that $m_k(z)$ decreases along its path, while Theorem 1 below shows that $\|z - x_k\|$ increases.

Therefore, if the polygon ends at $z_n \in R_k$ then $y_{k+1} = z_n$, and otherwise y_{k+1} is the unique point where the polygon crosses the boundary ∂R_k of the trust region.

Stated more formally, Steighaug's method proceeds as follows, where we made use of the identity $\nabla m_k(x_k) = \nabla f(x_k)$:

Algorithm 2: Steihaug

S0 choose tolerance $\epsilon > 0$, set $z_0 = x_k$, $d_0 = -\nabla f(x_k)$

S1 For $j = 0, \dots, n-1$ repeat

if $d_j^\top B_k d_j \leq 0$

find $\tau^* \geq 0$ s.t. $\|z_j + \tau^* d_j - x_k\| = \Delta_k$

stop with $y_{k+1} = z_j + \tau^* d_j$

else

$z_{j+1} := z_j + \tau_j d_j$, where $\tau_j := \arg\min_{\tau \geq 0} m_k(z_j + \tau d_j)$

if $\|z_{j+1} - x_k\| \geq \Delta_k$

find $\tau^* \geq 0$ s.t. $\|z_j + \tau^* d_j - x_k\| = \Delta_k$

stop with $y_{k+1} = z_j + \tau^* d_j$

end

if $\|\nabla m_k(z_{j+1})\| \leq \epsilon$ stop with $y_{k+1} = z_{j+1}$ (*)

compute $d_{j+1} = -\nabla m_k(z_{j+1}) + \frac{\|\nabla m_k(z_{j+1})\|^2}{\|\nabla m_k(z_j)\|^2} d_j$

end

end

Comments:

- Algorithm 2 stops with $y_{k+1} = z_n$ in (*) after iteration $n-1$ at the latest: in this case $d_j^\top B_k d_j > 0$ for $j = 0, \dots, n-1$, which implies $B_k \succ 0$ and $\nabla m_k(z_n) = 0$.
- Furthermore, since $d_0 = -\nabla f(x_k)$, the algorithm stops at the Cauchy point $y_{k+1} = y_k^c$ if it stops in iteration 0.
- If the algorithm stops later then $m_k(y_{k+1}) < m_k(y_k^c)$.
- The convergence theorem of Lecture 6 is applicable.

Theorem 1: Let the conjugate gradient algorithm be applied to the minimisation of $m_k(x)$ with starting point $z_0 = x_k$, and suppose that $d_j^\top B_k d_j > 0$ for $j = 0, \dots, i$. Then we have

$$0 = \|z_0 - x_k\| \leq \|z_1 - x_k\| \leq \cdots \leq \|z_i - x_k\|.$$

Proof:

- The restriction of B_k to $\text{span}\{d_0, \dots, d_i\}$ is a positive definite operator,

$$\Big(\sum_{j=0}^{i} \lambda_j d_j\Big)^\top B_k \Big(\sum_{j=0}^{i} \lambda_j d_j\Big) = \sum_{j=0}^{i} \lambda_j^2 d_j^\top B_k d_j > 0,$$

where we used the B_k-conjugacy property $d_j^\top B_k d_l = 0 \; \forall\, j \neq l$.

- Therefore, up to iteration i all the properties we derived for the conjugate gradient algorithm remain valid.

- Since $z_j - x_k = \sum_{l=0}^{j-1} \tau_l d_l$ for $(j = 1, \ldots, i)$, we have
$$\|z_{j+1} - x_k\|^2 = \|z_j - x_k\|^2 + \sum_{l=0}^{j-1} \tau_j \tau_l d_j^{\mathsf{T}} d_l.$$
Moreover, $\tau_j > 0$ for all j.

- Therefore, it suffices to show that $d_j^{\mathsf{T}} d_l > 0$ for all $l \leq j$.

- For $j = 0$ this is trivially true. We can thus assume that the claim holds for $j-1$ and proceed by induction.

- For $l < j$ have
$$d_j^{\mathsf{T}} d_l = -\nabla m_k(z_j)^{\mathsf{T}} d_l + \frac{\|\nabla m_k(z_j)\|^2}{\|\nabla m_k(z_{j-1})\|^2} d_{j-1}^{\mathsf{T}} d_l.$$

- The second term on the right-hand side is positive because of the induction hypothesis, and it was established in the proof of Lemma 2.3 from Lecture 5 that the first term is zero.

- Furthermore, if $l = j$ then we have of course $d_j^{\mathsf{T}} d_l > 0$. $\square$

Reading Assignment: Lecture-Note 7.