Numerical Optimisation:
Large scale methods

Department of Computer Sc
Centre for Medical Image Com
Centre for Inverse Proble
University College London

Lecture 9

- Hessian solve: Line search and trust region methods require factorisation of the Hessian. For large scale it is infeasible and has to be performed using large scale techniques such as sparse factorisations or iterative methods.

-
been developed, where the Hessian approximation can be stored using only few vectors (slow conver
Approximated Hessians preserving spa

- Special structure properties of the objective function like *partial separability* i.e. the function can be decomposed into a sum of simpler functions each depending only on a small subspace of $\mathbb{R}^n$.

Solve the Newton step system

$$\underbrace{\nabla^2 f_k}_{:=A} p_k^{\mathrm{iN}} = \underbrace{-\nabla f_k}_{b}$$

usin i

hand

Implementation can be done matrix free i.e. the He
need to be calculated or stored explicitly, we only re
which executes the Hessian matrix vector produ

Question: How does the inexact solve impact on the local convergence of the Newton methods?

Most of the termination rules for iterative methods are based on the residual

$$r_k = \nabla^2 f_k p_k^{\mathrm{iN}} + \nabla f_k,$$

where p_k^{iN} is the inexact Newton step.

Usua

(STOP)

where $\{\eta_k\}$ is some sequence $0 < \eta_k$

For the moment we assume that step of length
i.e. globalisation strategies do not interfere with the inexact-Newton step.

Suppose $\nabla^2 f(x)$ exists and is continuous in the neighbourhood of a minimiser $x^\star$, with $\nabla^2 f(x^\star)$ positive definite.
Consider the inexact Newton iteration with step length $\alpha_k = 1$

$x_{k+1} = x_k + p_k$, with a starting point x_0 sufficiently close to $x^\star$,
term ... me
cons

The

$$\|\nabla^2 f(x^\star)(x_{k+1} - x^\star)\| \leq \hat{\eta}\|$$

for some constant $\hat{\eta}\ :\ \eta < \hat{\eta} < 1$.

Remark: This result provides convergence for $\{\eta_k\}$ bounded away from 1.

Proof idea convergence (superlinear):

Continuity of $\nabla^2 f(x)$ in a neighbourhood $\mathcal{N}(x^\star)$ of $x^\star$ implies

$$\nabla f(x_k) = \nabla^2 f(x^\star)(x_k - x^\star) + o(\|x_k - x^\star\|),$$

thus show instead $\|\nabla f(x_{k+1})\| \leq \hat{\eta}\|\nabla f(x_k)\|$.

Continuity and positive definiteness of $^2 f(x)$ in $(x^\star)$ implies $\exists L \in$

From Taylor theorem and continuity of $^2 f(x)$ in $(x^\star)$ we have

$$\begin{aligned}
\nabla f(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k) p_k + \int_0^1 \quad] p_k dt \\
&= \nabla f(x_k) + \nabla^2 f(x_k) p_k + o(\\
&= \nabla f(x_k) - (\nabla f(x_k) - r_k) + o(\|\nabla f(x_k)\|) = r_k + o(\|\nabla f(x_k)\|)
\end{aligned}$$

$$\|\nabla f(x_{k+1})\| \leq \eta_k \|\nabla f(x_k)\| + o(\|\nabla f(x_k)\|) \leq (\eta_k + o(1))\|\nabla f(x_k)\|$$

with $\eta_k = o(1)$, $\leq o(\|\nabla f(x_k)\|)$.

Proof idea convergence (quadratic):

Continuity of $\nabla^2 f(x)$ in a neighbourhood $\mathcal{N}(x^\star)$ of $x^\star$ implies

$$\nabla f(x_k) = \nabla^2 f(x^\star)(x_k - x^\star) + o(\|x_k - x^\star\|),$$

thus show instead $\|\nabla f(x_{k+1})\| \le \hat{\eta}\|\nabla f(x_k)\|$.

Continuity and positive definiteness of $^2 f(x)$ in $(x^\star)$ implies
$\exists L \in$

From Taylor theorem and Lipschitz continuity of $^2 f(x)$ in $(x^\star)$

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k) p_k + \int_0^1 \quad] p_k \, dt$$

$$= \nabla f(x_k) + \nabla^2 f(x_k) p_k + \mathcal{O}($$

$$= \nabla f(x_k) - (\nabla f(x_k) - r_k) + \mathcal{O}(\|\nabla f(x_k)\|^2) = r_k + \mathcal{O}(\|\nabla f(x_k)\|^2)$$

with $\eta_k = \mathcal{O}(\|\nabla f(x_k)\|)$

$$\|\nabla f(x_{k+1})\| \le \eta_k \|\nabla f(x_k)\| + \mathcal{O}(\|\nabla f(x_k)\|^2) \le \mathcal{O}(\|\nabla f(x_k)\|^2).$$

Theorem: superlinear (quadratic) convergence

Suppose $\nabla^2 f(x)$ exists and is continuous in the neighbourhood of a minimiser $x^\star$, with $\nabla^2 f(x^\star)$ positive definite.
Let the sequence $\{x_k\}$ generated by the inexact Newton iteration with step length $\alpha_k = 1$, $x_{k+1} = x_k + p_k$ with stopping (iN-STOP) and x_0
suffi

The

If in addition $\nabla^2 f(x)$ is Lipschitz continuou
$\eta_k = \mathcal{O}(\|\nabla f_k\|)$, then the convergence is quad

Remark: To obtain superlinear convergence we can set e.g. $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$. The choice $\eta_k = \min(0.5, \|\nabla f_k\|)$ would yield quadratic convergence.

Also called *truncated Newton method*. The key differences to standard Newton line search method:

- Solve the Newton step with CG with initial guess 0 and the termination criterium (iN-STOP) with the suitable choice of η_k, e.g. $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$ for superlinear convergence.

-

- Away from the solution $x^\star$ the Hes
  definite. Therefore, we terminate CG whe
  of non-positive curvature is generated.
  guarantees that the produced search dire
  direction and preserves the fast pure Newton convergence rate provided $\alpha_k = 1$ is used whenever it satisfies the acceptance criteria.

Weakness: Performance when Hessian is nearly singular.

Use a special CG variant to solve the quadratic trust region model problem

$$\min_{p\in\mathbb{R}^n} m_k(p) = f_k + p^{\mathrm{T}}\nabla f_k + \frac{1}{2}p^{\mathrm{T}}\nabla^2 f_k p \quad \text{subject to } \|p\| \le \Delta_k.$$

Mod

-
  e.g. $\eta_k = \min(0.5, \sqrt{\|\nabla f_k\|})$ for supe
- If CG generates direction of non-positive cu
  i.e. $d_j^{\mathrm{T}}\nabla^2 f_k d_j \le 0$, stop and return
  minimises $m_k(p_k)$ along d_j and satisfies $\|p_k\| = \Delta_k$.

- If the current iterate violates the trust region constraint i.e. $\|z_{j+1}\| \ge \Delta_k$, stop and return $p_k = z_j + \tau d_j$, $\tau \ge 0$ which satisfies $\|p_k\| = \Delta_k$.

The initialisation $z_0 = 0$ is crucial:

- Whenever $\|r_k\| \geq \varepsilon_k$, the algorithm terminates at a point p_k for which $m_k(p_k) \leq m_k(p_k^C)$ that is when the reduction in the model is at least that of the Cauchy point.
  - If $d_0^{\mathrm{T}} B_k d_0 = \nabla f_0^{\mathrm{T}} B_0 \nabla f_0 \leq 0$, the first **if** is activated and the algorithm returns the Cauchy point $p = -(\Delta_0/\|\nabla f_0\|)\nabla f_0$.
  - Otherwise, the algorithm defines

$$\mathrm{T} \qquad \mathrm{T} \qquad .$$

    - t steps ensure that the final p_k satisfies $m_k(p_k)$ $m_k(z_1)$.
    - When $\|z_1\| \geq \Delta_0$, the second terminates at the Cauchy point.

Therefore, it is globally convergent.

- $\|z_{k+1}\| > \|z_k\| > \cdots > \|z_1\|$ as a consequence of the initialisation $z_0 = 0$. Thus we can stop as soon as the boundary of trust region has been reached, because no further iterates giving a lower value of m_k will lie inside the trust region.

- Preconditioning can be used, but requires change of trust region definition, which can be reformulated in the standard form in terms of a variable $\hat{p} = Dp$ and modified $\hat{g}_k = D^{-\mathrm{T}}\nabla f_k$ and $\hat{B}_k = D^{-\mathrm{T}}(\nabla^2 f_k)D^{-1}$. Of particular interest is incomplete Cholesky factorisation (Algorithm 7.3 in Nocedal and Wright).

-

  replaced by Lanczos method (which can be s
  generalisation of CG which works for indefi
  more computationally expensive) for wh
  exact trust region can be applied to compute a direction to quickly move away from stationary points which are not minimisers.

Recall the BFGS formula

$$H_{k+1} = (I - \frac{s_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}) H_k (I - \frac{y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}) + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} \qquad \text{(BFGS)}$$

with $s_k = x_{k+1} - x_k = \alpha_k p_k$, $y_k = \nabla f_{k+1} - \nabla f_k$. Application of BFGS Hessian approximation can be efficiently implemented stori

The l

total n

approximation to the last $m \ll n$. After th

pair in the list makes space for the new pair

Same strategy can be applied to the other quasi-N (including updating B_k for use with e.g. trust region methods rather than line search methods which require H_k).

Application: large, non-sparse Hessians.
Convergence: often linear convergence rate.

Consider the *memoryless* BFGS

$$H_{k+1} = (I - \frac{s_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k})(I - \frac{y_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}) + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}$$

i.e. the previous Hessian is reset to identity, $H_k = I$.

If the m

line se

$$p_{k+1} = -H_{k+1} \nabla f_{k+1} = -\nabla f \quad \frac{\quad k \quad\quad k+1}{}$$

which is exactly the Hestens-Stiefel formula, wh

Polak-Ribiere when $\nabla f_{k+1}^{\mathrm{T}} p_k = 0$

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k)}{p_k^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k)}, \quad \beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^{\mathrm{T}} \nabla f_k}.$$

Compact representation of BFGS update

Let B_0 be symmetric positive definite and assume that the vector pairs $\{s_i, y_i\}_{i=0}^{k-1}$ satisfy $s_i^{\mathrm{T}} y_i > 0$. Applying k BFGS updates with these vector pairs to B_0 yields

- In limited memory version we replace the columns or diagonal entries in the matrices cyclically (keeping m last columns).
- Since the dimension of the middle matrix is small, the factorisation cost is negligible.
- Cost of an update: $2mn + \mathcal{O}(m^3)$
- Cost of $B_k v$: $(4m+1)n + \mathcal{O}(m^3)$, (for $B_0 = \delta_k I$)
-

- Similar compact representation can be de
- Compact representation can also be derived as

$$B_k = B_0 + (Y_k - B_0 S_k)(D_k + L_k + L_k^T - S_k^T B_0 S_k)^{-1}(Y_k - B_0 S_k)^T$$

with S_k, Y_k, D_k, L_k as before. The inverse formula for H_k can be obtained by swapping $B \leftrightarrow H$, $s \leftrightarrow y$, however limited memory SR-1 can be less effective than BFGS.

We require the quasi-Newton approximation to the Hessian B_k to has the same (or similar) sparsity pattern as the true Hessian. Suppose that we know which components of the Hessian are non-zero

$$\Omega = \ (i,j) : [\ ^2 f(x)]_{ij} \neq 0 \text{ for some point } x \text{ in the domain of } f\} ,$$

and su

spar

solut

$$\min_{B} \quad \|B - B_k\|_F^2 = \quad (i,j)$$

$$\text{subject to} \quad Bs_k = y_k, \ B = B \quad /$$

It can be shown that the solution of this problem can be obtained solving an $n \times n$ linear system with sparsity pattern Ω. B_{k+1} is not guaranteed to be positive definite. The new B_{k+1} can be used within a trust region.

Unfortunately, this approach has several drawbacks, it is not scale invariant under linear transformations and the performance is disappointing. The fundamental weakness is that the closeness in Frobenius norm is an inadequate model and the produced approximations can be poor.

An alternative approach is to relax the secant equation making sure that it is approximately satisfied at the m last steps (as oppo

subject to $\quad B = B^{\mathrm{T}},\ B_{ij}$ /

with S_k, Y_k containing the last m of s_i

This convex optimisation problem has a solution but it is not easy to compute. Furthermore, it can produce singular and poorly conditioned Hessian approximations. Even though it frequently outperforms the previous approach, its performance is still not impressive for large scale problems.

Partially separable functions

An unconstrained optimisation problem is **separable** if the objective function $f : \mathbb{R}^n \to \mathbb{R}$ can be decomposed in a sum of independent functions e.g.

$$f(x) = f_1(x_1, x_3) + f_2(x_2, x_4, x_6) + f_3(x_5).$$

The optimal value can be found optimising each function inde

In ma $\to \mathbb{R}$ is not se

component functions. Each such component h

it only changes in a small number of directions while f

directions is remains constant. We call such funct

separable.

All functions which have a sparse Hessian are partially separable, but there are many partially separable functions with dense Hessians. Partial separability allows for economical representation and effective quasi-Newton updating.

Consider an objective function $f : \mathbb{R}^n \to \mathbb{R}$

$$f(x) = \sum_{i=1}^{\ell} f_i(x),$$

where each f_i depends only on a few components of x. For such f_i, its gra

For th

$$\nabla f(x) = \sum_{i=1}^{\ell} \nabla f_i(x), \quad \nabla^2 f$$

thus we can maintain an quasi-Newton approximation to each individual component Hessian $\nabla^2 f_i(x)$ instead of approximating the entire Hessian $\nabla^2 f(x)$.

Consider a partially separable objective function

$$f(x) = \underbrace{(x_1 - x_3^2)^2}_{f_1(x)} + \underbrace{(x_2 - x_4^2)^2}_{f_2(x)} + \underbrace{(x_3 - x_2^2)^2}_{f_3(x)} + \underbrace{(x_4 - x_1^2)^2}_{f_4(x)}.$$

Each

Den

$$x^{[1]} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \quad U_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad x^{[1]} \quad -z_2^2)^2.$$

Then $f_1(x) = \phi(U_1 x)$ and using chain rule we o

$$\nabla f_1(x) = U_1^{\mathrm{T}} \nabla \phi_1(U_1 x), \quad \nabla^2 f_1(x) = U_1^{\mathrm{T}} \nabla^2 \phi_1(U_1 x) U_1.$$

For the Hessians $\nabla^2\phi_1$ and $\nabla^2 f_1$ we have

$$\nabla^2\phi_1(U_1 x) = \begin{bmatrix} 2 & -4x_3 \\ -4x_3 & 12x_3^2 - 4x_1 \end{bmatrix}, \quad \nabla^2 f_1(x) = \begin{bmatrix} 2 & 0 & -4x_3 & 0 \\ 0 & 0 & 0 & 0 \\ -4x_3 & 0 & 12x_3^2 - 4x_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Idea: maintain quasi-Newton approximation $B^{[1]}$ to 2×2 Hessian $\nabla^2\phi_1$ and lift it up to $\nabla^2 f_1$.

Afte

and we use BFGS or SR-1 updating to obtain the new approximation $B_{k+1}^{[1]}$ of the small, dense Hess

it back using

$$\nabla^2 f_1(x) \approx U_1^{\mathrm{T}} B_{k+1} U_1.$$

We do the same for all component functions and we obtain

$$\nabla^2 f \approx B = \sum_{i=1}^{\ell} U_i^{\mathrm{T}} B^{[1]} U_i.$$

- The approximated Hessian may be used in trust region algorithm, obtaining an approximate solution to

$$B_k p_p = -\nabla f_k.$$

B_k does not need to be assembled explicitly but conjugate gradient method can be used and the products $B_k v$ can be

-

Then each respective component Hessia

approximated by the iterative method (s

few directions) and the so obtained full Hess

is usually much better than one obtained by a quasi-Newton method applied to the problem ignoring the partially separable structure (large Hessian requires a lot of directions to approximate the curvature).

- It is not always possible for BFGS to update the partial Hessian $B^{[1]}$, as the curvature condition $(s^{[1]})^{\mathrm{T}} y^{[1]} > 0$ may not be satisfied even if the full Hessian is at least positive semidefinite. This can be overcome applying SR-1 update to the component Hessians, which proved effective in practice.

-

  Newton step.

- Another problem is the difficulty of identify separable structure of a function. The perfo quasi-Newton methods is satisfactory provided that we find the *finest* partially separable decomposition.