The Steepest Descent, Coordinate Search and the Newton-Raphson Method

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

We continue to consider the unconstrained minimisation problem

\[
\min_{x \in \mathbb{R}^n} f(x).
\]

In Lecture 2 we considered line-search descent methods:

**Algorithm 1** Choose a starting point \( x_0 \in \mathbb{R}^n \) and a tolerance parameter \( \epsilon > 0 \). Set \( k = 0 \).

\( S1 \) If \( \| \nabla f(x_k) \| \leq \epsilon \) then stop and output \( x_k \) as an approximate minimiser.

\( S2 \) Choose a search direction \( d_k \in \mathbb{R}^n \) such that \( \langle \nabla f(x_k), d_k \rangle < 0 \).

\( S3 \) Choose a step size \( \alpha_k > 0 \) such that \( f(x_k + \alpha_k d_k) < f(x_k) \).

\( S4 \) Set \( x_{k+1} := x_k + \alpha_k d_k \), replace \( k \) by \( k + 1 \), and go to \( S1 \).

We proved a convergence result which only required that

- \( d_k \) is a descent direction; \( \langle \nabla f(x_k), d_k \rangle < 0 \),

- a line-search has to be used.

Since we already discussed the issue of choosing a step length \( \alpha_k \) (remember the Wolfe conditions?), we can now concentrate on methods to compute good search directions \( d_k \).
Steepest Descent: This choice of search direction was already motivated and discussed in Example 2 of Lecture 2:

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

- Intuitively appealing.
- Easy to apply, \(-\nabla f(x_k)\) “cheap” to compute.
- \(\theta(-\nabla f(x_k), d_k) \equiv 0\) in this case, and Theorem 2 of Lecture 2 implies convergence.

Regrettably, the method has major disadvantages:

- Badly affected by round-off errors.
- Badly affected by ill-conditioning, convergence can be excruciatingly slow due to excessive zig-zagging.

To illustrate this, let \(x^*\) be a strict local minimiser of \(f\) and suppose that the sufficient first and second order optimality conditions hold, i.e.,

\[ \nabla f(x^*) = 0, \quad D^2 f(x^*) > 0. \]

The second condition implies that the ordered eigenvalues of \(D^2 f(x^*)\) satisfy

\[ \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0. \]

The ratio \(\kappa := \frac{\lambda_1}{\lambda_n}\) is called the condition number of \(D^2 f(x^*)\). If \(\kappa\) is large, then \(x^*\) lies in a "long narrow valley" of \(f\).

Once the steepest descent method enters this valley, it just bounces back and forth without making much progress when \(\kappa\) is large:

**Proposition 1:** Let \(x_0\) be a starting point and let the sequence \((x_k)_{k\in\mathbb{N}}\) be produced by

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

where \(\alpha_k\) corresponds to an exact line-search (see Lecture 2). Then

\[ \|x_{k+1} - x^*\| \simeq \frac{\kappa - 1}{\kappa + 1} \|x_k - x^*\| \]

for all \(k\) large.
Coordinate Search: This method is even simpler, as the search direction cycles through the coordinate axes:

\[ d_k = e_i, \quad i \equiv 1 + k \mod n. \]

- Even cheaper, as \( d_k \) does not have to be computed at all.
- Convergence even worse than steepest descent.

\[ \text{Newton Methods:} \] This approach is motivated by the first order necessary optimality condition \( \nabla f(x^*) = 0 \) and works when \( D^2 f(x) \) is non-singular for \( x \) in a neighbourhood of \( x^* \).

- Idea: replace the nonlinear root-finding problem \( \nabla f(x) = 0 \) by a sequence of linear problems which are easy to solve.
- Linearisation: given \( x_k \), the first order Taylor approximation

\[ x \mapsto \varphi(x) = \nabla f(x_k) + D^2 f(x_k) (x - x_k), \]

approximates the nonlinear (vector valued) function \( x \mapsto \nabla f(x) \) well in a neighbourhood of \( x_k \).

\[ \text{Newton-Raphson method:} \] given a starting point \( x_0 \), apply exact Newton steps

\[ x_{k+1} = x_k + n_f(x_k). \]

- \( n_f(x) \) is a descent direction when \( D^2 f(x) \succ 0 \):

\[ \langle n_f(x), \nabla f(x) \rangle = - (\nabla f(x))^\top (D^2 f(x_k))^{-1} \nabla f(x_k) < 0, \]

since \( D^2 f(x) \succ 0 \Rightarrow (D^2 f(x))^{-1} \succ 0 \). In particular, this happens when \( f \) is strictly convex (see Lecture 1).

- If \( D^2 f(x) \npreceq 0 \) then \( n_f(x) \) may not be a descent direction and the method may converge to any point where \( \nabla f(x) = 0 \), which could be a minimiser, maximiser or saddle point.
Examples can be constructed on which the method cycles through a finite number of points, that is, \( x_{k+j} = x_k \) for some \( k, j \in \mathbb{N} \), and the method does not converge.

However, when \( x_0 \) is chosen sufficiently close to \( x^* \) where the first and second order optimality conditions for a minimiser hold, then the convergence is Q-quadratic, see Theorem 1 below.

Conclusions:

- Newton’s method is great for the minimisation of convex problems (or the maximisation of concave problems).

- Since \( f \) is typically strictly convex in a neighbourhood of a local minimiser \( x^* \), it is great to switch to Newton’s method in the final phase of an algorithm that otherwise relies on a line-search descent method.

Dampened Newton method:

- Uses the following search direction in Algorithm 1,

\[
d_k = \begin{cases} 
  n_f(x_k) & \text{if } \langle n_f(x_k), \nabla f(x_k) \rangle < 0, \\
  -n_f(x_k) & \text{otherwise.}
\end{cases}
\]

- The line-search step length \( \alpha_k \) should asymptotically become 1 (i.e., full Newton step taken) if the fast convergence rate of the Newton-Raphson method is to be picked up.

Example 1: Linear Programming. Consider the linear programming problem

\[
\begin{align*}
\max_{x \in \mathbb{R}^n} c^T x \\
\text{s.t. } Ax &\leq b, \\
x &\geq 0.
\end{align*}
\]

Here \( A \in \mathbb{R}^{m \times n} \) (a \( m \times n \) matrix with linearly independent rows), \( b \in \mathbb{R}^m \) and \( c \in \mathbb{R}^n \) are all given, and \( x \in \mathbb{R}^n \) is the vector of decision variables.

Let \( \mu > 0 \) and \( e := \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^T \).
At the heart of interior-point methods for linear programming lies the solution of the nonlinear system of equations

\[ \begin{align*}
Ax &= b \\ 
A^T y + s &= c \\
XSe &= \mu e \\
x, s &> 0,
\end{align*} \]

where \( x, s \in \mathbb{R}^n \), \( y \in \mathbb{R}^m \), \( X = \text{Diag}(x) \) and \( S = \text{Diag}(s) \) are the diagonal matrices with \( x \) and \( s \) on their diagonals, and where \( x, s > 0 \) means that both vectors have to be component-wise strictly positive.

It can be shown that the system (1)-(4) has a unique solution \((x^*, y^*, s^*)\).

Given a current approximate solution \((x, y, s)\) such that \( x, s > 0 \), we can compute a Newton step \((\Delta x, \Delta y, \Delta s)\) for the unconstrained system (1)-(3) which is obtained by solving the linearised system of equations

\[ \begin{align*}
A\Delta x &= b - Ax \\
A^T \Delta y + \Delta s &= c - A^T y - s \\
S\Delta x + X\Delta s &= \mu e - XSe.
\end{align*} \]

In order to guarantee that (4) continues to be satisfied, we use \((\Delta x, \Delta y, \Delta s)\) as a search direction and determine an updated approximate solution \((x_+, y_+, s_+)\) as follows:

\[ \alpha^* = \sup \{ \alpha > 0 : x + \alpha \Delta x > 0, s + \alpha \Delta s > 0 \}, \]

\[ (x_+, y_+, s_+) = (x, y, s) + \min(1, 0.99\alpha^*) (\Delta x, \Delta y, \Delta s). \]

It can be shown that the resulting sequence of intermediate solutions converges very efficiently to \((x^*, y^*, s^*)\).

\textbf{Theorem 1: Convergence of Newton-Raphson.}

Let \( f \in C^2(\mathbb{R}^n, \mathbb{R}) \) with \( \Lambda \)-Lipschitz continuous Hessian. Let \( x^* \in \mathbb{R}^n \) be such that \( \nabla f(x^*) = 0 \) and \( D^2 f(x^*) \) nonsingular. Then there exists a neighbourhood \( B_\rho(x^*) \) with the property that \( x_0 \in B_\rho(x^*) \) implies \( x_k \in B_\rho(x^*) \) for all \( k \), and \( x_k \to x^* \) \( Q \)-quadratically.
Proof:

- $D^2f(x^*)$ nonsingular, $x \mapsto D^2f(x)$ continuous $\Rightarrow \exists \bar{\rho} > 0$ such that $D^2f(x)$ nonsingular for all $x \in B_{\bar{\rho}}(x^*)$ and $n_f(x)$ well-defined.

- Moreover, $x \mapsto (D^2f(x))^{-1}$ is continuous, thus can choose $\bar{\rho}$ sufficiently small so that

\[ \|(D^2f(x))^{-1}\| \leq 2\|(D^2f(x^*))^{-1}\| =: \beta. \quad (5) \]

- The Newton update implies

\[ (x_{k+1} - x^*) = (x_k - x^*) - (D^2f(x_k))^{-1}\nabla f(x_k). \quad (6) \]

- Lipschitz continuity of $D^2f$ implies

\[
\|S\| \leq \int_{t=0}^{1} \|D^2f(x_k) - D^2f(tx^* + (1 - t)x_k)\|dt \\
\leq \int_{t=0}^{1} \Lambda t \|x_k - x^*\|dt = \frac{\Lambda}{2} \|x_k - x^*\|.
\]

- Substituting this and (5) in (8),

\[
\|x_{k+1} - x^*\| \leq \frac{\beta\Lambda}{2} \|x_k - x^*\|^2. \quad (9)
\]

- Finally, for $\rho := \min(\bar{\rho}, 2(\beta\Lambda)^{-1})$, (9) shows that

\[ x_k \in B_{\rho}(x^*) \Rightarrow x_k \in B_{\rho}(x^*), \]

so that the entire sequence $(x_k)\in\mathbb{N}$ is well defined as long as $x_0 \in B_{\rho}(x^*)$.

- Using $\nabla f(x^*) = 0$, find

\[ \nabla f(x_k) = \nabla f(x_k) - \nabla f(x^*) = \int_{t=0}^{1} D^2f(tx^* + (1 - t)x_k)(x_k - x^*)dt \]

- Substituting into (6),

\[
(x_{k+1} - x^*) = (D^2f(x_k))^{-1}S(x_k - x^*),
\]

where

\[
S := D^2f(x_k) - \int_{t=0}^{1} D^2f(tx^* + (1 - t)x_k)dt \\
= \int_{t=0}^{1} D^2f(x_k) - D^2f(tx^* + (1 - t)x_k)dt.
\]

- Taking norms on both sides of (7),

\[
\|x_{k+1} - x^*\| \leq \|(D^2f(x_k))^{-1}\| \times \|S\| \times \|x_k - x^*\|. \quad (8)
\]

Reading Assignment: Download and read Lecture-Note 3.

Note: From now on all lectures are in Comlab 147.