Lecture 5: Newton's method for optimization problems (continued)

Coralia Cartis, Mathematical Institute, University of Oxford

C6.2/B2: Continuous Optimization

Disadvantages of Newton's method for optimization

- in the conditions of local convergence Theorem 9: x^k can get attracted to local maxima or saddle points of f if x^k sufficiently close to such points (as $\nabla^2 f(x^*)$ only required to be nonsingular in Th 9).

  Example: $f : \mathbb{R} \to \mathbb{R}, \;\; f(x) = -x^2$;
  $x^* = 0$ is global maximizer;
  apply Newton starting from $x^0 = 1 \Rightarrow s^0 = -1$ ascent direction and $x^1 = 0$.

- Newton's method may fail to converge at all if x^0 "too far" from solution (outside neighbourhood of local convergence, failure may occur).

  $\longrightarrow$ Newton is not globally convergent for general f.

Disadvantages of Newton's method for optimization

Example of failure of Newton's method to converge globally.

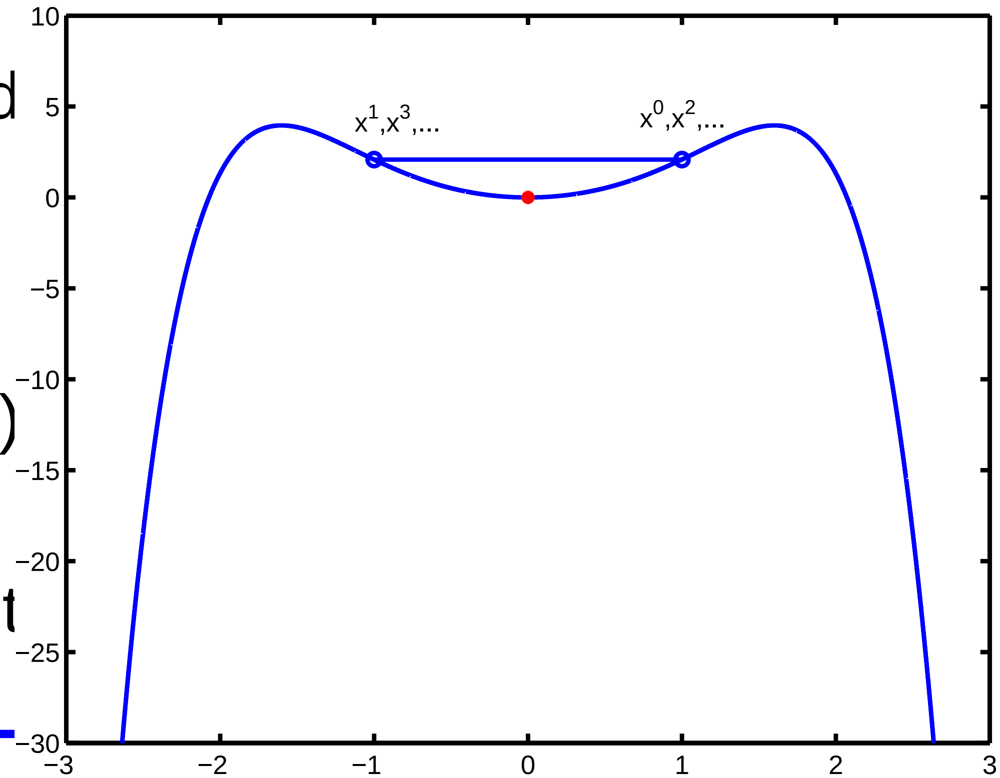
$$f : \mathbb{R} \to \mathbb{R}, \quad f(x) = -\frac{x^6}{6} + \frac{x^4}{4} + 2x^2.$$

$x^* = 0$ local minimizer; $x = \pm\sqrt{(1+\sqrt{17})/2} \approx \pm 1.6$ global max.

Newton's method applied to f, with $x^0 = 1$;
$\Rightarrow$ $x^{2k} = 1$ and $x^{2k+1} = -1$, for all k.

-1 and 1 are not (even) stationary points of f.

Note that s^k descent but we have gone "too far".



Newton's method applied to f.

Damped Newton's method

$\Longrightarrow$ include linesearch in Newton's method: damped Newton.

Damped Newton's method for minimization:

Choose $\epsilon > 0$ and $x^0 \in \mathbb{R}^n$.

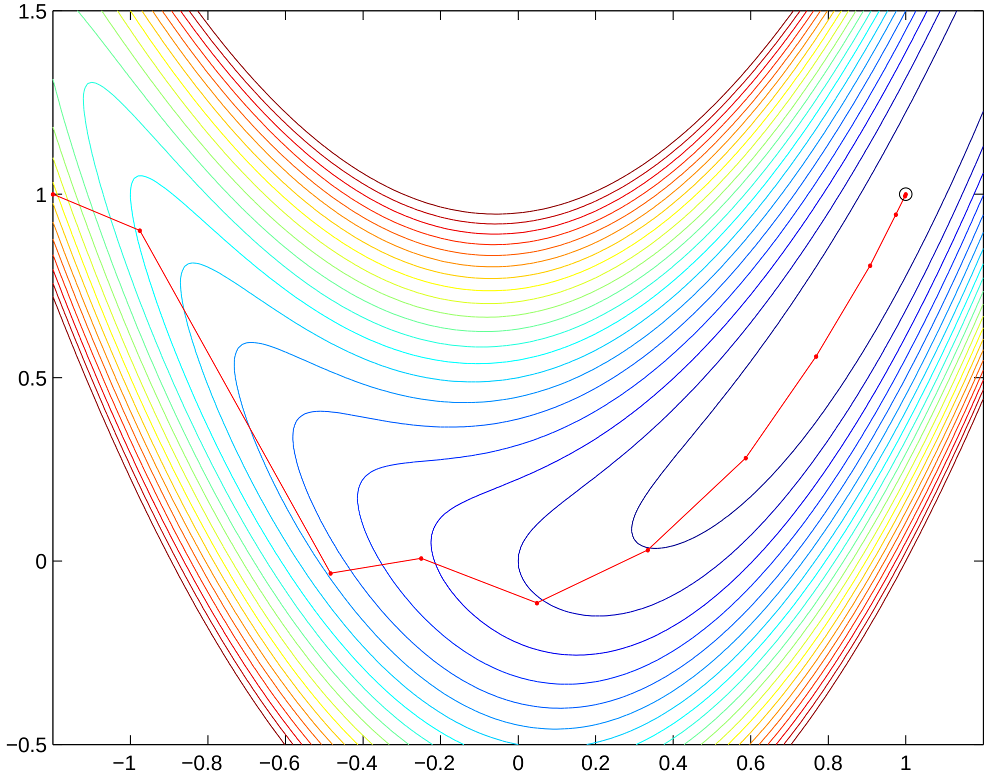
While $\|\nabla f(x^k)\| > \epsilon$, REPEAT:

- solve the linear system $\nabla^2 f(x^k) s^k = -\nabla f(x^k)$.
- set $x^{k+1} = x^k + \alpha^k s^k$, with $\alpha^k \in (0,1]$; $k := k+1$. END.

- Damped Newton's method is a GLM provided $\nabla^2 f(x^k)$ is positive definite so that s^k descent. Then α^k can be computed by exact linesearch, bArmijo, etc.
- if $\alpha^k \to 1$ as $k \to \infty \Longrightarrow$ damped Newton's mthd is locally quadratically convergent.
- (local convergence) Assume $\nabla^2 f$ is Lipschitz cont., and $\nabla^2 f(x^k) \succ 0$. Let $x^k \to x^*$ with $\nabla^2 f(x^*) \succ 0$. Let $s^k =$Newton direction in GLM and bArmijo linesearch have $\beta < 0.5$ and $\alpha_{(0)} = 1$. Then, $\alpha^k = 1$ for all k suff. large and $x^k \to x^*$ quadratically.

Local convergence for damped Newton with bArmijo

$$f(x_1, x_2) = 10(x_2 - x_1^2)^2 + (x_1 - 1)^2; \quad x^* = (1, 1).$$



Damped Newton with bArmijo linesearch applied to the Rosenbrock function f.

- $\beta < 0.5$ and $\alpha_{(0)} = 1$ in bArmijo; $\alpha^k = 1$ for suff. large k.

Global convergence of damped Newton's method

- recall backtracking Armijo (bArmijo) linesearch.

Theorem 10 Let $f \in \mathcal{C}^2(\mathbb{R}^n)$ be bounded below on $\mathbb{R}^n$.
Let ∇f be Lipschitz continuous. Let the eigenvalues of $\nabla^2 f(x^k)$ be positive and uniformly bounded below, away from zero (for all k). Apply damped Newton's method to f with bArmijo linesearch and $\epsilon = 0$. Then

either

there exists $l \geq 0$ such that $\nabla f(x^l) = 0$

or

$\|\nabla f(x^k)\| \to 0$ as $k \to \infty$. $\square$

• Theorem 10 is satisfied if $f \in \mathcal{C}^2$ with ∇f Lipschitz continuous is also strongly convex (i.e., the eigenvalues of $\nabla^2 f(x)$ for all x are positive, bounded below, away from zero). Then s^k is descent for all k.

Global convergence of damped Newton's method ...

Proof of Theorem 10. The conditions of Theorem 4 (Global convergence of GLM with bArmijo linesearch) are satisfied. Thus Th 4 gives that either $\exists\, l \geq 0$ such that $\nabla f(x^l) = 0$ or

$$M_k := \min\left\{\frac{|\nabla f(x^k)^T s^k|}{\|s^k\|}, |\nabla f(x^k)^T s^k|\right\} \longrightarrow 0 \text{ as } k \to \infty. \text{ } (\dagger)$$

Let $\nabla^2 f(x^k) := H_k$. Th assumptions on $f \implies \forall s \in \mathbb{R}^n$, $s \neq 0$,

$$0 < \lambda_{\min} \leq \lambda_{\min}(H_k) \leq \frac{s^T H_k s}{\|s\|^2} \leq \lambda_{\max}(H_k) \leq \lambda_{\max}.$$

$$|\nabla f(x^k)^T s^k| = |\nabla f(x^k)^T H_k^{-1} \nabla f(x^k)| \geq \lambda_{\min}(H_k^{-1}) \|\nabla f(x^k)\|^2$$

$$= \frac{\|\nabla f(x^k)\|^2}{\lambda_{\max}(H_k)} \geq \frac{\|\nabla f(x^k)\|^2}{\lambda_{\max}}.$$

$$\|s^k\|^2 = \nabla f(x^k)^T H_k^{-2} \nabla f(x^k) \leq \lambda_{\max}(H_k^{-2}) \|\nabla f(x^k)\|^2 \leq \lambda_{\min}^{-2} \|\nabla f(x^k)\|^2.$$

$\implies M_k \geq \min\left\{\frac{\lambda_{\min}}{\lambda_{\max}} \|\nabla f(x^k)\|, \frac{1}{\lambda_{\max}} \|\nabla f(x^k)\|^2\right\}$ for all k

$\implies \nabla f(x^k) \longrightarrow 0$ as $k \to \infty$. $\square$

Modified damped Newton methods

If $\nabla^2 f(x^k)$ is not positive definite, it is usual to solve instead

$$\left(\nabla^2 f(x^k) + M^k\right) s^k = -\nabla f(x^k),$$

where

- M^k chosen such that $\nabla^2 f(x^k) + M^k$ is "sufficiently" positive definite.
- $M^k := 0$ when $\nabla^2 f(x^k)$ is "sufficiently" positive definite.

Options:

1. As $\nabla^2 f(x^k)$ is symmetric, we can factor $\nabla^2 f(x^k) = Q^k D^k (Q^k)^\top$, where Q^k is orthogonal and D^k is diagonal, and set

$$\nabla^2 f(x^k) + M^k := Q^k \max(\epsilon I, |D^k|)(Q^k)^\top,$$

for some "small" $\epsilon > 0$. Expensive approach for large problems.

Modified damped Newton methods

2. Estimate $\lambda_{\min}(\nabla^2 f(x^k))$ and set

$$M^k := \max(0, \epsilon - \lambda_{\min}(\nabla^2 f(x^k)))I.$$

Cheaper. Often tried in practice but "biased" (may overemphasize a large negative eigval at the expense of small, positive ones).

3. Modified Cholesky: compute Cholesky factorization

$$\nabla^2 f(x^k) = L^k (L^k)^\top,$$

where L^k is lower triangular matrix. Modify the generated L^k if the factorization is in danger of failing (modify small or negative diagonal pivots, etc.).

Popular in computations.

Other directions for GLMs

Choose/compute B^k to approximate $\nabla^2 f(x^k)$.

Let B^k symmetric, positive definite matrix. Let s^k be defined by

$$B^k s^k = -\nabla f(x^k).$$

Update B^k after the calculation of s^k and α^k.

- $\Longrightarrow$ s^k descent direction;
- $\Longrightarrow$ s^k solves the problem

$$\text{minimize}_{s \in \mathbb{R}^n} \; m_k(s) = f(x^k) + \nabla f(x^k)^T s + \frac{1}{2} s^T B^k s^k.$$

- s^k is a scaled steepest descent direction;
- Theorem 10 (global convergence) continues to hold with $\nabla^2 f(x^k)$ replaced by B^k in the statement and proof.

Approximating the Hessian matrix by finite differences

Approximating the Hessian from gradient vals: $i \in \{1, \ldots, n\}$;

$$[\nabla^2 f(x)]e^i \approx \frac{1}{h}[\nabla f(x + he^i) - \nabla f(x)]$$

Cost of approximating $\nabla^2 f(x)$ is $n + 1$ gradient values.

For all finite-differencing, careful with the choice of h in computations:

- “too large” $h \to$ inaccurate approximations,
- “too small” $h \to$ numerical cancellation errors.

But successful techniques exist for smooth noiseless problems when sufficient function and/or gradient values can be computed.

For noisy problems, use derivative-free optimization methods (if problem size is not too large).

Quasi-Newton methods

Secant approximations for computing $B^k \approx \nabla^2 f(x^k)$

At the start of the GLM, choose B^0 (say, $B^0 := I$). After computing $s^k = -(B^k)^{-1}\nabla f(x^k)$ and $x^{k+1} = x^k + \alpha^k s^k$, compute update B^{k+1} of B^k.

Wish list:

Compute B^{k+1} as a function of already-computed quantities $\nabla f(x^{k+1})$, $\nabla f(x^k)$, ..., $\nabla f(x^0)$, B^k, s^k,

B^{k+1} should be symmetric, nonsingular (pos. def.),

B^{k+1} "close" to B^k, a "cheap" update of B^k, $B^k \to \nabla^2 f(x^k)$, etc.

$\Longrightarrow$ a new class of methods: faster than steepest descent method, cheaper to compute per iteration than Newton's.

For the first wish, choose B^{k+1} to satisfy the secant equation

$$\gamma^k := \nabla f(x^{k+1}) - \nabla f(x^k) = B^{k+1}(x^{k+1} - x^k) = B^{k+1}\alpha^k s^k.$$

Quasi-Newton methods ...

Interpretation of the secant equation:

It is satisfied by $B^{k+1} := \nabla^2 f$ when f is a quadratic function.

The change in gradient contains information about the Hessian.

The gradient change predicted by the current quadratic model

$$\nabla f(x^{k+1}) - \nabla f(x^k) \approx \nabla q(x^k + \alpha^k s^k) - \nabla q(x^k) = -\alpha^k \nabla f(x^k),$$

where $\quad q(x^k + s) = f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^k s$

and $\quad s^k = -(B^k)^{-1} \nabla f(x^k)$.

Want the new quadratic model

$$u(x^k + s) := f(x^k) + \nabla f(x^k)^\top s + \frac{1}{2} s^\top B^{k+1} s$$

to predict correctly the change in gradient γ^k, i.e.,

$$\gamma^k = \nabla f(x^{k+1}) - \nabla f(x^k) = \nabla u(x^{k+1}) - \nabla u(x^k) = B^{k+1}(x^{k+1} - x^k).$$

Quasi-Newton methods ...

Many ways to compute B^{k+1} to satisfy the secant equation.
Trade-off between "wishes" on the list for some of the methods.

Symmetric rank 1 updates. [see Prob Sheet 3]

Set $B^{k+1} := B^k + u^k(u^k)^\top$, for some $u^k \in \mathbb{R}^n$, and all $k \geq 0$.

- B^{k+1} symmetric, "close" to B^k.
- Work per iteration: $\mathcal{O}(n^2)$ (as opposed to the $\mathcal{O}(n^3)$ of Newton), due to Sherman-Morrison-Woodbury formula!

The secant equation $\Longrightarrow u^k = (\gamma^k - B^k\delta^k)/\rho^k$,
where $\delta^k := x^{k+1} - x^k = \alpha^k s^k$, $(\rho^k)^2 := (\gamma^k - B^k\delta^k)^\top \delta^k > 0$.

- B^k may not be positive definite, s^k may not be descent.
- ρ^k may be close to zero leading to large updates.

Other updates: BFGS, DFP, Broyden family, etc.

Quasi-Newton methods ...

BFGS updates.

[see Prob Sheet 3]

- Broyden-Fletcher-Goldfarb-Shanno (independently).

Set $B_{k+1} := B_k + u_k u_k^\top + v_k v_k^\top$, for some $u_k \in \mathbb{R}^n$, $v_k \in \mathbb{R}^n$.

- It is a rank 2 update (if u_k and v_k are linearly independent).
- SWM formula yields $\mathcal{O}(n^2)$ operations/iteration.
- In practice, update the Cholesky factors of B_k (still $\mathcal{O}(n^2)$).

Given $B_k = J_k J_k^\top$, where J_k arbitrary nonsingular, and $\|\cdot\|_F$ Frobenius norm, let J_{k+1} solve

$$\min_J \|J - J_k\|_F \quad \text{subject to} \quad J\delta_k = \gamma_k.$$

$$\Rightarrow \quad B_{k+1} := J_{k+1} J_{k+1}^\top = B_k + u_k u_k^\top + v_k v_k^\top,$$

where $u_k u_k^\top = -B_k \delta_k \delta_k^\top B_k / (\delta_k^\top B_k \delta_k)$, $v_k v_k^\top = \gamma_k \gamma_k^\top / (\gamma_k^\top \delta_k)$.

- Let $J_k := L_k$ the lower triangular Cholesky factor of B_k.

Quasi-Newton methods ...

BFGS updates. (continued)

- Thus B_{k+1} is “close” to B_k.
- B_k symmetric pos. def. $\Rightarrow B_{k+1}$ symmetric pos. def. (provided $(\delta^k)^T\gamma^k > 0$, ensured by say, Wolfe linesearch)
- BFGS method: GLM with $s_k := -B_k^{-1}\nabla f(x_k)$, with B_k updated by BFGS formula on each iteration.
- For global convergence of BFGS method, must use Wolfe linesearch to compute stepsize instead of bArmijo linesearch.
- The BFGS method has local Q-superlinear convergence!
- When applying the BFGS method with exact linesearches, to a strictly convex quadratic function f, then $B_k = \nabla^2 f$ after n iterations.
- Satisfies all the wishes on the wish list! Has been very popular when second derivatives of f are not available.

Appendix: providing derivatives to algorithms

How to compute/provide derivatives to a solver?

- Calculate derivatives by hand when easy/simple objective and constraints; user provides code that computes them.
- Calculate or approximate derivatives automatically:
  - Automatic differentiation: breaks down computer code for evaluating f into elementary arithmetic operations + differentiate by chain rule. Software: ADIFOR, ADOL-C.
  - Symbolic differentiation: manipulate the algebraic expression of f (if available). Software: symbolic packages of MAPLE, MATHEMATICA, MATLAB.
  - Finite differencing $\longrightarrow$ approximate derivatives.

See Nocedal & Wright, Numerical Optimization (2nd edition, 2006) for more details of the above procedures.