Optimization and Data Science

Lecture 13: Quasi-Newton Methods

Prof. Dr. Thomas Slawig

Kiel University - CAU Kiel
Dep. of Computer Science

Summer 2020

Contents

1. Quasi-Newton Methods
   - Recall: Globalized Newton method
   - Finite-Difference Approximation of the Hessian
   - Basis of Quasi-Newton Methods: Secant Method
   - Quasi-Newton methods: Hessian Updates

Contents

1 Quasi-Newton Methods

- Recall: Globalized Newton method
- Finite-Difference Approximation of the Hessian
- Basis of Quasi-Newton Methods: Secant Method
- Quasi-Newton methods: Hessian Updates

Globalized Newton method

Algorithm (Globalized Newton method):

1. Fix some parameter $c > 0$.
2. Choose initial guess $x_0 \in \mathbb{R}^n$.
3. For $k = 0, 1, \dots$:
    1. Compute Newton direction d_k, i.e., solve
    $$\nabla^2 f(x_k) d_k = -\nabla f(x_k),$$
    2. If Newton direction is not gradient-related, i.e., if
    $$-\frac{\nabla f(x_k)^\top d_k}{\|\nabla f(x_k)\| \|d_k\|} < c,$$
    set $d_k = -\nabla f(x_k)$.
    3. Choose an efficient step-size $\rho_k > 0$.
    4. Set $x_{k+1} = x_k + \rho_k d_k$.

    until a stopping criterion is satisfied.

Properties of globalized Newton method

- Under some assumptions (see last lecture), we obtain Q-superlinear convergence of the sequence of iterates, i.e., there exists $(q_k)_{k\in\mathbb{N}}$, $q_k \to 0$, with

$$\|x_{k+1} - x^*\| \leq q_k \|x_k - x^*\| \text{ for all } k \in \mathbb{N}.$$

- But we have higher effort (than, e.g., for the gradient method):
- In every Newton step we have to ...
  - evaluate the gradient:

$$\mathcal{O}(n) \times \text{Effort } (f).$$

  - evaluate the Hessian matrix:

$$\mathcal{O}(n^2) \times \text{Effort } (f).$$

  - solve the linear system:

$\mathcal{O}(n^3)$ operations for a dense matrix (less for a sparse matrix).

Contents

1. Quasi-Newton Methods
   - Recall: Globalized Newton method
   - Finite-Difference Approximation of the Hessian
   - Basis of Quasi-Newton Methods: Secant Method
   - Quasi-Newton methods: Hessian Updates

Finite-Difference Approximation of the Hessian

- Components of the gradient $\nabla f(x)$ can be approximated by

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x+he_i)-f(x)}{h}, \quad i=1,\ldots,n, \ \text{ with } h>0 \text{ fixed.}$$

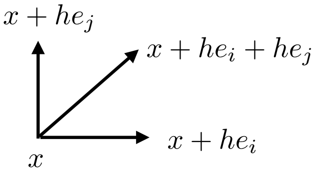
$\rightsquigarrow$ n additional evaluations of f.

- Components of the Hessian $\nabla^2 f(x)$ can be approximated by

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \approx \frac{f(x+he_j+he_i)-f(x+he_i)-f(x+he_j)+f(x)}{h^2}, \quad i,j=1,\ldots,n,$$

again with fixed $h>0$.

- Hessian symmetric $\rightsquigarrow$ $\frac{n(n+1)}{2}$ additional evaluations of f.
- Used approximation introduces a special direction in the derivative approximation ...
- ... maybe not good.

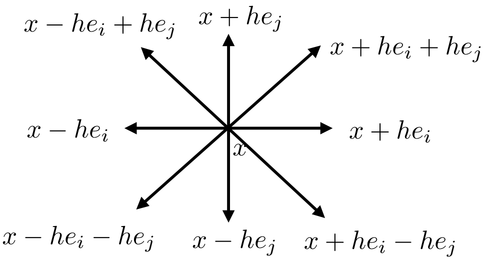


Better Finite-Difference Approximation of the Hessian

- Central approximation for gradient:

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x+he_i)-f(x-he_i)}{2h}.$$

$\rightsquigarrow$ $2n$ additional evaluations of f.



- Same idea for the Hessian:

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \approx \frac{\partial}{\partial x_i}\frac{f(x+he_j)-f(x-he_j)}{2h}$$
$$\approx \frac{1}{2h}\left(\frac{f(x+he_j+he_i)-f(x-he_j+he_i)}{2h} - \frac{f(x+he_j-he_i)-f(x-he_j-he_i)}{2h}\right).$$

- Hessian symmetric $\rightsquigarrow 4\frac{n(n+1)}{2} = 2n(n+1)$ additional evaluations of f.

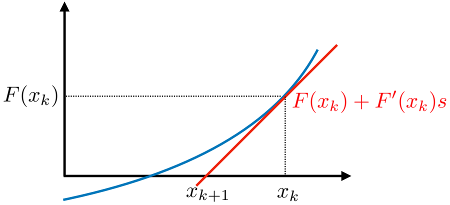
Contents

1. Quasi-Newton Methods
   - Recall: Globalized Newton method
   - Finite-Difference Approximation of the Hessian
   - Basis of Quasi-Newton Methods: Secant Method
   - Quasi-Newton methods: Hessian Updates

There is a more efficient way to approximate a derivative

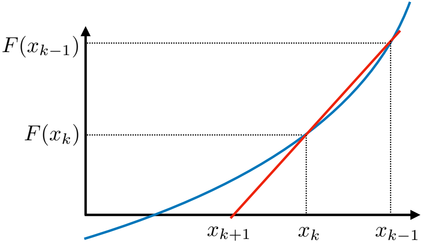
- Back to 1-D Newton: find zero of nonlinear function $F : \mathbb{R} \to \mathbb{R}$ using the tangent at x_k:

$$F(x_k) + F'(x_k)(x_{k+1} - x_k) = 0$$
$$\Leftrightarrow \quad F'(x_k) d_k = -F(x_k).$$



- Tangent can be approximated by secant, 1-D:

$$F'(x_k) \approx \frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}}$$



- ... or (also for $F : \mathbb{R}^n \to \mathbb{R}^n$):

$$F'(x_k)(x_k - x_{k-1}) \approx F(x_k) - F(x_{k-1}),$$

- ... where $F(x_k) \in \mathbb{R}^{n \times n}$ is now a matrix.

Approximation of the derivative using the secant equation

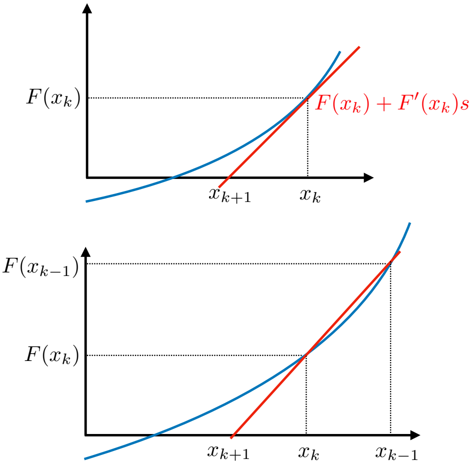
- Idea: In the Newton method

$$F'(x_k)d_k = -F(x_k)$$
$$x_{k+1} = x_k + d_k,$$

- ... replace $F'(x_k)$ by a matrix $B_k \in \mathbb{R}^{n\times n}$ that satisfies
- ... the **secant** or **Quasi-Newton equation**:

$$B_k(x_k - x_{k-1}) = F(x_k) - F(x_{k-1}).$$

- How to construct the matrices $B_k, k = 1, \ldots,$ in an easy and efficient way?



Broyden update

- We want that the matrices B_k satisfy the Quasi-Newton or secant equation:

$$B_k(x_k - x_{k-1}) = F(x_k) - F(x_{k-1}), \quad k = 0, 1, \ldots$$

- We realize this by an easily computable and cheap, additive update

$$B_k := B_{k-1} + U_k, \quad k = 1, 2, \ldots, \text{ with } B_0 \text{ given},$$

- ... with the **Broyden update**

$$U_k := \frac{(y_k - B_{k-1}s_k)s_k^\top}{s_k^\top s_k} \text{ where } y_k := F(x_k) - F(x_{k-1}), s_k := x_k - x_{k-1}.$$

- A matrix $vs^\top = (v_i s_j)_{i,j=1}^n \in \mathbb{R}^{n \times n}$ is called **dyadic product** of v and s.

Dyadic product

- Broyden update

$$U_k := \frac{(y_k - B_{k-1} s_k) s_k^\top}{s_k^\top s_k}$$

- is a dyadic product

$$v s^\top = (v_i s_j)_{i,j=1}^n = \begin{pmatrix} v_1 s_1 & \cdots & v_1 s_n \\ \vdots & & \vdots \\ v_n s_1 & \cdots & v_n s_n \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

- It has rank $= 1$, thus the Broyden update is called a **rank 1-update**.
- Obviously, it is not necessarily symmetric.
- Evaluation requires only n^2 operations.

Broyden update satisfies secant equation

- **Broyden update**

$$U_k := \frac{(y_k - B_{k-1}s_k)s_k^\top}{s_k^\top s_k}, \quad y_k := F(x_k) - F(x_{k-1}), s_k := x_k - x_{k-1}.$$

- Using the Broyden update, the matrix B_k satisfies the secant equation

$$B_k s_k = (B_{k-1} + U_k)s_k = \left(B_{k-1} + \frac{(y_k - B_{k-1}s_k)s_k^\top}{s_k^\top s_k}\right) s_k$$

$$= B_{k-1}s_k + \frac{(y_k - B_{k-1}s_k)s_k^\top s_k}{s_k^\top s_k} = B_{k-1}s_k + y_k - B_{k-1}s_k = y_k.$$

Minimizing property of Broyden update

- The Broyden update is the minimal change to B_{k-1} that preserves the secant equation:

$$U_k = \operatorname{argmin}\left\{\|V\| : V \in \mathbb{R}^{n\times n}, (B_{k-1}+V)s_k = y_k\right\}$$

- where $\|\cdot\|$ is a matrix norm that satisfies

$$\begin{aligned} \|AB\| &\le \|A\|\|B\| & \forall A, B \in \mathbb{R}^{n\times n} \\ \left\|\frac{xx^\top}{x^\top x}\right\| &\le 1 & \forall x \in \mathbb{R}^n. \end{aligned}$$

- If $V \in \mathbb{R}^{n\times n}$ is any other matrix with $(B_{k-1}+V)s_k = y_k$, then

$$\|U_k\| = \left\|\frac{(y_k - B_{k-1}s_k)s_k^\top}{s_k^\top s_k}\right\| = \left\|\frac{(Vs_k)s_k^\top}{s_k^\top s_k}\right\| = \left\|\frac{V(s_k s_k^\top)}{s_k^\top s_k}\right\| = \left\|V\frac{s_k s_k^\top}{s_k^\top s_k}\right\| \le \|V\|\left\|\frac{s_k s_k^\top}{s_k^\top s_k}\right\| \le \|V\|.$$

Contents

1 Quasi-Newton Methods
- Recall: Globalized Newton method
- Finite-Difference Approximation of the Hessian
- Basis of Quasi-Newton Methods: Secant Method
- Quasi-Newton methods: Hessian Updates

Approximation of the Hessian using the secant equation

- We extend the above idea for optimization: $F := \nabla f$:
- In the Newton method

$$\begin{aligned} \nabla^2 f(x_k) d_k &= -\nabla f(x_k) \\ x_{k+1} &= x_k + d_k, \end{aligned}$$

- ... replace $\nabla^2 f(x_k)$ by a matrix $H_k \in \mathbb{R}^{n \times n}$ that satisfies
- ... the **secant** or **Quasi-Newton equation**:

$$H_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}).$$

- We realize this by an easily computable and cheap, additive update

$$H_k := H_{k-1} + U_k, \quad k = 1, 2, \dots, \text{ with } H_0 \text{ given},$$

Updates that preserve symmetry and positive-definiteness

- For the Hessian we want to preserve symmetry (since the Hessian is symmetric)
- ... and (under some assumptions) also positive-definiteness.
- For this purpose we need a **rank 2-update**.
- There are several update formulas.
- Most prominent: **BFGS** (Broyden-Fletcher-Goldfarb-Shanno) update:

$$U_k := \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{H_{k-1} s_k (H_{k-1} s_k)^\top}{s_k^\top H_{k-1} s_k}, \quad y_k := \nabla f(x_k) - \nabla f(x_{k-1}), s_k := x_k - x_{k-1}.$$

- Other well-known update formula: DFP update (Davidon-Fletcher-Powell).
- For H_0, we may once evaluate or approximate the Hessian $\nabla^2 f(x_0)$ ...
- ... or even take $H_0 = I$.
- Under some assumptions, globalized Quasi-Newton methods (as the BFGS method) are superlinear convergent.

Globalized Quasi-Newton method

Algorithm (Globalized Quasi-Newton method):

1. Fix some parameter $c > 0$.
2. Choose initial guess $x_0 \in \mathbb{R}^n$ and initial matrix H_0.
3. For $k = 0, 1, \dots$:
   1. Compute direction d_k, i.e., solve
   $$H_k d_k = -\nabla f(x_k),$$
   2. If direction is not gradient-related, ... (as above) set $d_k = -\nabla f(x_k)$.
   3. Choose an efficient step-size $\rho_k > 0$.
   4. Set $x_{k+1} = x_k + \rho_k d_k$.
   5. Update Hessian approximation (e.g., by BFGS update):
   $$H_{k+1} := H_k + U_{k+1}, \text{ using } y_{k+1} = \nabla f(x_{k+1}) - \nabla f(x_k), s_{k+1} = x_{k+1} - x_k.$$

   until a stopping criterion is satisfied.

Comparison: Effort of Newton vs. Quasi-Newton methods

In every step we have to ...

- evaluate the gradient:

$$\mathcal{O}(n) \ \times \ \text{Effort } (f).$$

- Newton: evaluate the Hessian matrix:

$$\mathcal{O}(n^2) \ \times \ \text{Effort } (f).$$

- Quasi-Newton: update the Hessian approximation:

$$\mathcal{O}(n^2) \text{ operations, independent of the effort for } f.$$

- solve the linear system:

$$\mathcal{O}(n^3) \text{ operations for a dense matrix (less for a sparse matrix)}$$

One step further: inverse updates

- In the Newton method

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k)$$

- … we could formally write

$$d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k).$$

- Idea: Update inverse matrix directly:

Lemma (Sherman-Morrison-Woodbury formula)

Let $B \in \mathbb{R}^{n \times n}$ be invertible and $u, v \in \mathbb{R}^n$. Then, $B + U$ with $U = uv^\top$ is invertible if and only if

$$\sigma := 1 + v^\top B^{-1} u \neq 0.$$

Then, we have

$$(B+U)^{-1} = B^{-1} - \frac{1}{\sigma} B^{-1} U B^{-1}.$$

Inverse updates

- The above formula can be applied twice.
- We then get also a rank-2 update for the BFGS update (and other rank-2 update formulas).
- Setting $\hat{H}_k := H_k^{-1}$, these updates then satisfy

$$\hat{H}_k := \hat{H}_{k-1} + \hat{U}_k$$

- ... with, for the inverse **BFGS** update:

$$\hat{U}_k := \frac{(s_k - \hat{H}_{k-1} y_k) s_k^\top + s_k (s_k - \hat{H}_{k-1} y_k)^\top}{y_k^\top s_k} - \frac{(s_k - \hat{H}_k y_k)^\top y_k}{(y_k^\top s_k)^2} s_k s_k^\top, \qquad (1)$$

with y_k, s_k as above.

Globalized Quasi-Newton method using inverse update

Algorithm (Globalized Quasi-Newton method, inverse update):

1. Fix some parameter $c > 0$.
2. Choose initial guess $x_0 \in \mathbb{R}^n$ and initial matrix H_0.
3. For $k = 0, 1, \ldots$:
   1. Compute direction
      $$d_k = -\hat{H}_k \nabla f(x_k),$$
   2. If direction is not gradient-related, ... (as above) set $d_k = -\nabla f(x_k)$.
   3. Choose an efficient step-size $\rho_k > 0$.
   4. Set $x_{k+1} = x_k + \rho_k d_k$.
   5. Update inverse Hessian approximation:
      $$\hat{H}_{k+1} := \hat{H}_k + \hat{U}_{k+1}, \text{ using } y_{k+1} = \nabla f(x_{k+1}) - \nabla f(x_k), s_{k+1} = x_{k+1} - x_k.$$

   until a stopping criterion is satisfied.

Comparison: Effort of Newton vs. Quasi-Newton methods

In every step we have to ...

- evaluate the gradient:

$$\mathcal{O}(n) \times \text{Effort } (f).$$

- Newton: evaluate the Hessian matrix:

$$\mathcal{O}(n^2) \times \text{Effort } (f).$$

- Quasi-Newton: update the (inverse) Hessian approximation:

$$\mathcal{O}(n^2) \text{ operations, independent of the effort for } f.$$

- solve the linear system:

$$\mathcal{O}(n^3) \text{ operations for a dense matrix (less for a sparse matrix)}$$

- or: use inverse update and matrix-vector multiplication:

$$\mathcal{O}(n^2) \text{ operations for a dense matrix (less for a sparse matrix)}$$

What is important

- The idea of Quasi-Newton methods is to approximate the Hessian (or inverse Hessian) iteratively by an additive rank-two update.
- The effort of $\mathcal{O}(n^2)$ function evaluations for an approximation of the Hessian by finite differences is avoided.
- The idea is to replace the tangent in the Newton method by a secant, using only already computed values.
- Inverse updates can even more reduce the effort of solving the linear system in every Quasi-Newton iteration.
- Quasi-Newton methods retain the superlinear convergence property of the Newton method under some assumptions.