Lecture 7

Unconstrained minimization

General scheme

$$x_{k+1} = x_k + t_k d_k, \quad k = 0, 1, 2, \ldots$$

The steepest descent method

$$d_k = -\nabla f(x_k), \quad \underbrace{t_k : \min_{t \geq 0} f(x_k + t d_k)}_{\text{exact line search}}$$

The Newton method

$$d_k = -H(x_k)^{-1} \nabla f(x_k), \quad t_k = 1 \text{ or}$$

$$t_k : \approx \min_{t \geq 0} f(x_k + t d_k)$$

$$\updownarrow$$

$$H(x_k) d_k = -\nabla f(x_k)$$

$$\updownarrow$$

d_k is the minimizer of the quadratic function:

$$q_k(d) = f(x_k) + d^T \nabla f(x_k) + \frac{1}{2} d^T H(x_k) d$$

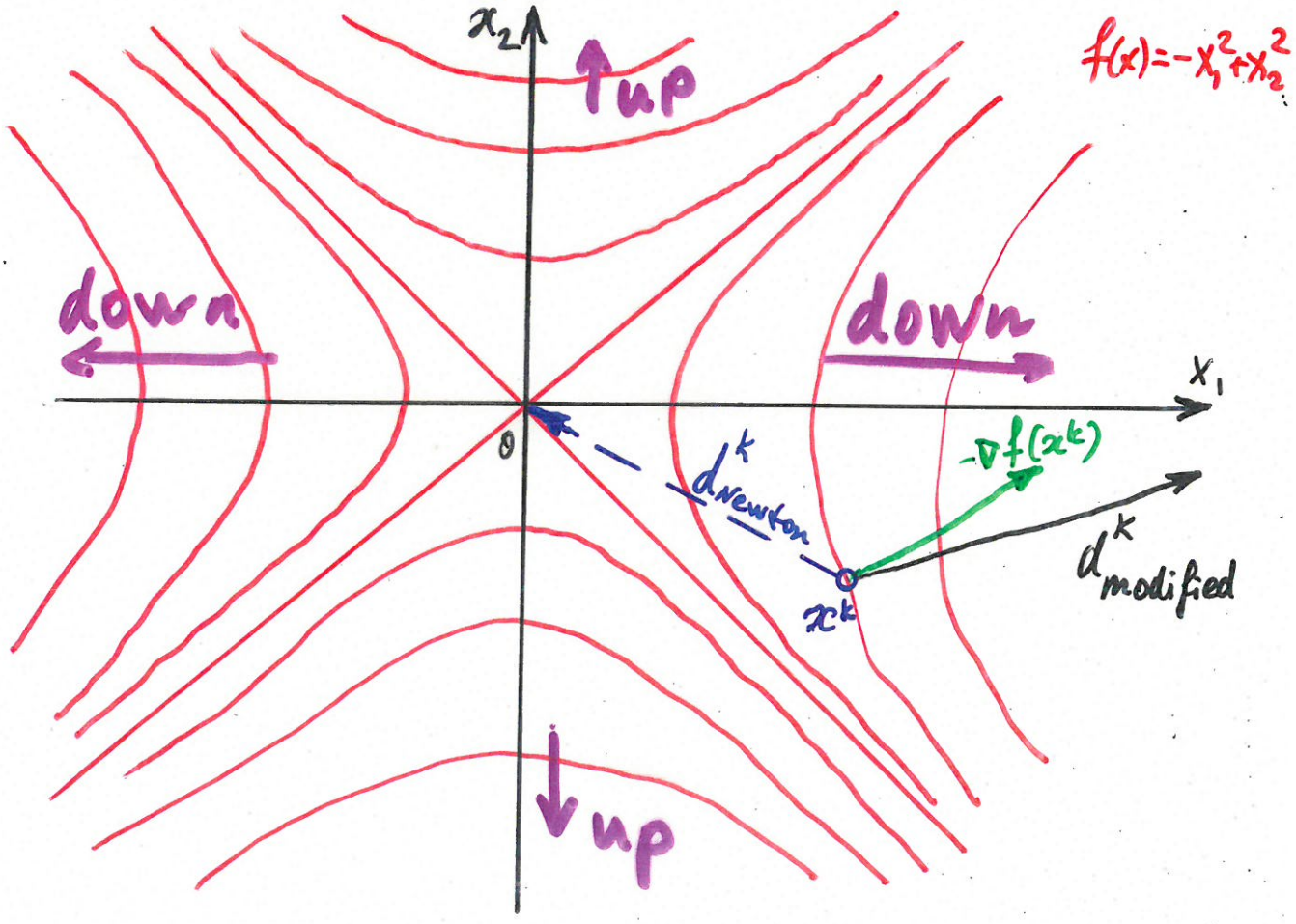
provided that $H(x_k) > 0$

$$\min_{x \in R^n} f(x)$$

$$x^{k+1} = x^k + t_k \cdot d^k$$

$$d^k = -\nabla f(x^k)$$ the steepest descent method

$$d^k = -H(x^k)^{-1} \nabla f(x^k)$$ Newton's method



Modified Newton's method

$$Q_k^{mod}(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \\ + \frac{1}{2} (x - x^k)^T G^k (x - x^k)$$

$$\left.\begin{array}{l} G_k > 0 \\ \min\limits_x Q_k^{mod}(x) \end{array}\right\} \Rightarrow d_k^{mod} = -(G^k)^{-1} \nabla f(x^k)$$

$$H(x^k) > 0 \Rightarrow G^k = H(x^k)$$

otherwise $\Rightarrow$ choose $G^k > 0$

$$\nabla f(x^k)^T d_{mod}^k < 0$$

$$-\underbrace{\left[\nabla f(x^k)^T (G^k)^{-1} \nabla f(x^k)\right]}_{\geq 0} < 0$$

How to choose G^k?

The Levenberg-Marquardt method

$$x^{k+1} = x^k + t_k d^k$$

$$d^k = -\left(G^k\right)^{-1} \nabla f(x^k) \iff G^k d^k = -\nabla f(x^k)$$

Algorithm of computing G^k

1. Compute the Hessian $H(x^k)$.
2. Compute λ_k, the smallest eigenvalue of $H(x^k)$.
3. If $\lambda_k > 0$, set $\gamma_k = 0$, otherwise ($\lambda_k \le 0$), choose $\gamma_k > -\lambda_k$.
4. Compute $G^k = H(x^k) + \gamma_k I$, where

$$I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}. \Longrightarrow \boxed{G^k > 0}$$

Proof. Let λ_{min} and λ_{max} be the smallest and the largest eigenvalues of a symmetric matrix $A \in R^{n \times n}$. Then $\lambda_{min}\|v\|^2 \le v^T A v \le \lambda_{max}\|v\|^2$, $\forall\, v \in R^n$. Hence

$$v^T G^k v = \underbrace{v^T H(x^k) v}_{\ge \lambda_k \|v\|^2} + \underbrace{v^T(\gamma_k I) v}_{= \gamma_k \|v\|^2} \ge \underbrace{(\lambda_k + \gamma_k)}_{>0}\|v\|^2 > 0, \quad \forall\, v \ne 0.$$

Example 10.4 Levenberg - Marquardt method

Determine a search direction using Marquardt's modification for the problem

$$\min \; f(x_1, x_2) = \frac{1}{12}x_1^4 - x_1^2 + x_2^2 - 2x_1x_2 - 2x_1$$

Use $\mathbf{x}^{(0)} = (1\ 0)^T$ *as a starting point.*

Solution:

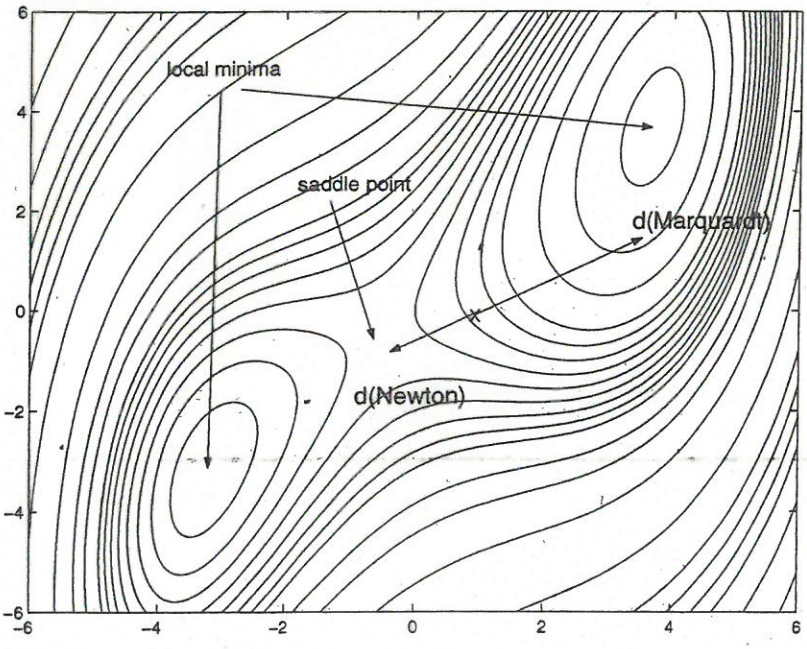
The Hessian matrix is $\mathbf{H}(\mathbf{x}) = \begin{pmatrix} x_1^2 - 2 & -2 \\ -2 & 2 \end{pmatrix}$ *and in the point* $\mathbf{x}^{(0)}$ *it is indefinite because the eigenvalues are* $\lambda_1 = -2$ *and* $\lambda_2 = 3$. *The search direction in Newton's method becomes*

$$\mathbf{d}^{(0)} = \begin{pmatrix} -1.89 \\ -0.89 \end{pmatrix}$$

and this is an ascent direction. A line search in this direction would give $t^{(0)} = 0$. *Using Marquardt's modification, we add* $\lambda\mathbf{I}$, *where we can choose for example,* $\lambda = 2.5$. *The new matrix is positive definite as the eigenvalues are* $\lambda_1 = 0.5$ *and* $\lambda_2 = 5.5$. *The search direction becomes*

$$\mathbf{d}^{(0)} = \begin{pmatrix} 7.45 \\ 3.76 \end{pmatrix}$$

which is a descent direction. Figure 10.3 illustrates how Marquardt's modification rotates the search direction (from Newton's method) and becomes a descent direction. The Newton direction provides an ascent direction because the Hessian matrix is indefinite in the given point.



Quasi-Newton methods

Algorithm

Given: $x_0 \in \mathbb{R}^n$ and $B_0 \in \mathbb{R}^{n \times n}$: $B_0 > 0$.

compute $\nabla f(x_0)$

For $k = 0, 1, 2, \ldots$ do

1. Stop if $\nabla f(x_k) = 0$.
2. Set $d_k = -B_k^{-1} \nabla f(x_k)$
3. Find t_k: $f(x_k + t_k d_k) \approx \min_{t \geq 0} f(x_k + t d_k)$
4. Set $x_{k+1} = x_k + t_k d_k$, $s_k = x_{k+1} - x_k$.
5. Compute $\nabla f(x_{k+1})$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
6. Update B using a quasi-Newton formula

$$B_{k+1} = B_k + \Delta B(B_k, s_k, y_k)$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS) updating formula:

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

Properties

1. If $B_0 > 0$ and $s_k^T y_k > 0 \quad \forall k \geq 0$, then $B_k > 0 \quad \forall k \geq 0$.

   Comment: $s_k^T y_k = \underbrace{s_k^T (\nabla f(x_{k+1}) - \nabla f(x_k))}_{\approx 0} \cong -s_k^T \nabla f(x_k)$

2. Fast convergence $x_k \to x_*$

3. No second derivative is involved

4. Other nice properties

Conjugate gradient methods

$$\min_{x \in R^n} q(x) \equiv \frac{1}{2} x^T A x + b^T x + c$$

Assumption: $A > 0$

x_* is a global min $\iff$ $\nabla q(x_*) = 0$ $\iff$ $Ax_* + b = 0$

Def. Nonzero vectors $d_i \in R^n$, $i = 0, 1, \dots, m$ are called conjugate if

$$d_i^T A d_j = 0, \quad \forall\, i \neq j$$

Properties

1. $d_i^T A d_i > 0$
2. Conjugate directions $d_0, \dots, d_m$ are linarly independent: $(\alpha_0 d_0 + \dots + \alpha_m d_m = 0) \Rightarrow (\alpha_0 = \dots = \alpha_m = 0)$

   Proof. Suppose, to the contrary, that $\alpha_i \neq 0$. Then

   $$d_i^T A (\alpha_0 d_0 + \dots + \alpha_m d_m) = \underbrace{\alpha_i}_{>0} \underbrace{d_i^T A d_i}_{>0} + \sum_{\substack{0 \le j \le m \\ j \neq i}} \alpha_j \underbrace{d_i^T A d_j}_{=0} > 0$$

   This contradicts the assumption that $\alpha_0 d_0 + \dots + \alpha_m d_m = 0$
3. If $d_0, \dots, d_{n-1}$ is a complete set of eigenvectors of A, then they are conjugate. (Exercise: prove it)

Th. Let n vectors $d_0, d_1, \dots, d_{n-1}$ be conjugate with respect to $A > 0$. Suppose that the exact line search is used in

$$x_{k+1} = x_k + t_k d_k, \quad k = 0, 1, 2, \dots,$$

$$f(x_{k+1}) = \min_t f(x_k + t d_k),$$

then $\nabla f(x_{k+1}) \perp d_0, \dots, d_k$, $\forall x_0 \in R^n$.

Remark. The exact line search implies $\nabla f(x_{k+1}) \perp d_k$

Proof (of Th.) By induction in k.

For $k=1$, $d_0^T \nabla f(x_1) = 0$.

Suppose that $d_i^T \nabla f(x_{k+1}) = 0$, $\forall\, i \le k$.

Then $d_{k+1}^T \nabla f(x_{k+2}) = 0$ due to the exact line search.

$$\nabla f(x_{k+2}) = \nabla f(x_{k+1}) + A(t_{k+1} d_{k+1})$$

[because $\nabla f(u) = \nabla f(v) + A(u - v)$]

For $i \le k$: $d_i^T \nabla f(x_{k+2}) = \underbrace{d_i^T \nabla f(x_{k+1})}_{=0 \ (i \le k)} + t_{k+1} \underbrace{d_i^T A d_{k+1}}_{=0}$

$$= 0$$

Corollary. $\forall x_0 \in R^n$, $\exists k \le n$ such that

$$\nabla f(x_k) = 0 \Rightarrow x_k = x_*$$

Q: How to generate conjugate directions?

Conjugate gradient (CG) methods: quadratic case

$$d_k = -\nabla q(x_k) + \beta_k d_{k-1}$$

$$\beta_k = \begin{cases} 0, & \text{if } k=0 \\ \dfrac{\|\nabla q(x_k)\|^2}{\|\nabla q(x_{k-1})\|^2}, & \text{otherwise } (k>0) \end{cases}$$

(Fletcher – Reeves formula)

$\Rightarrow d_0 = -\nabla q(x_0)$

Properties

1. $d_0, \ldots, d_k$ are conjugate directions
2. $\nabla q(x_{k+1}) \perp \nabla q(x_i)$, $i = 0, 1, \ldots, k$
3. If $B_0 = I$ in BFGS, then it is identical to CG

   Exercise: Prove that $x_2^{CG} = x_2^{BFGS}$)

CG methods: non-quadratic case

$$\min_{x \in R^n} f(x)$$

Algorithm (Fletcher – Reeves)

Given $x_0 \in R^n$

Compute $\nabla f(x_0)$. Set $\beta_0 = 0$ and $d_{-1} = 0$.

For $k = 0, 1, 2, \ldots$ do

1. Stop if $\nabla f(x_k) = 0$.
2. Set $d_k = -\nabla f(x_k) + \beta_k d_{k-1}$
3. Find t_k such that $f(x_k + t_k d_k) = \min_{t>0} f(x_k + t d_k)$
4. Set $x_{k+1} = x_k + t_k d_k$ and compute $\nabla f(x_{k+1})$
5. If $k+1$ is a multiple of n, set $\beta_{k+1} = 0$ (restart).

   Otherwise, set $\beta_{k+1} = \dfrac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$.

Properties

1. Fast convergence $x_k \to x_*$.
2. Low memory requirements (a few n-vectors)
3. No second derivatives involved.