Optimization

Màster de Fonaments de Ciència de Dades

Lecture III. Methods for unconstrained optimization

Gerard Gómez

Preliminaries on optimization methods

- It should be stressed that one hardly can hope to design a **single optimization method capable to solve efficiently all nonlinear optimization problems** – these problems are too diverse

- Methods for numerical solving nonlinear optimization problems are, in their essence, **iterative routines**: a method typically is unable to find exact solution in **finite number** of computations

- What a method generates, is an infinite sequence $\{x_n\}$ of approximate solutions

- Once $\{x_n\}$ has been computed, the next iterate $\{x_{n+1}\}$ is formed, according to certain rules, on the basis of local information of the problem collected along the previous iterates

One-dimensional unconstrained optimization

- Let
  $$f : \mathbb{R} \to \mathbb{R}$$
  be a differentiable function **with a local extremum at** x^*

- As we have already seen, the necessary condition of extrema is: $f'(x^*) = 0$

- So, **the local extrema are solutions of**
  $$f'(x) = 0$$

- This last equation is the one that must be solved, by means of some method to find the roots of a general non-linear equation: $\Phi(x) = 0$

Preliminaries on optimization methods

- Optimization methods can be classified according to the type of local information they use
  - **Zero-order** methods: use only values of the objective and the constraints and do not use their derivatives
  - **First-order** methods: use the values and the gradients of the objective and the constraints
  - **Second-order** methods: use the values, the gradients and the Hessians (i.e., matrices of second-order derivatives) of the objective and the constraints

One-dimensional unconstrained optimization. Summary

1. **Methods in dimension one:**

$$x^{n+1} = F(x^n, x^{n-1}, x^{n-2}, ...), \quad x^n \in \mathbb{R}$$

- **Newton's** method (second order)
- **Secant** method (first order)
- General line search methods
  - **Quadratic** method (zero order)
  - **Cubic** method (first order)
- Line search methods for unimodal functions
  - **Zero-order linear search** (zero order)
  - **Fibonacci** method (zero order)
  - **Golden section** method (zero order)

One-dimensional unconstrained optimization

Newton's method

The idea behind Newton's method is:

- Use a guess x^k for the solution of $f'(x) = 0$. Let the first one be $x^0 \in \mathbb{R}$
- Linearize f' around x^k

$$f'(x) \approx f'(x^k) + f''(x^k)(x - x^k)$$

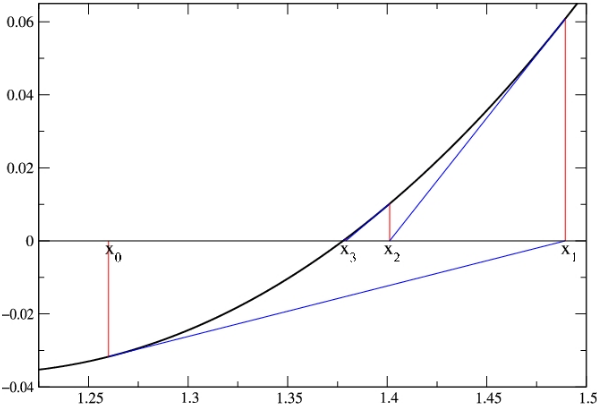
- Solve for the point where the linear function vanishes.

$$f'(x^k) + f''(x^k)(x - x^k) = 0$$

This point is the next guess x^{k+1}

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}, \quad k = 0, 1, 2, \ldots$$

Newton's method



The **question** is to know under which conditions the resulting sequence $\{x^k\}$ formula converges to the solution x^* of our problem.

Newton's method

Lemma

Let $\phi : [a, b] \to T \subset \mathbb{R}$ *with*

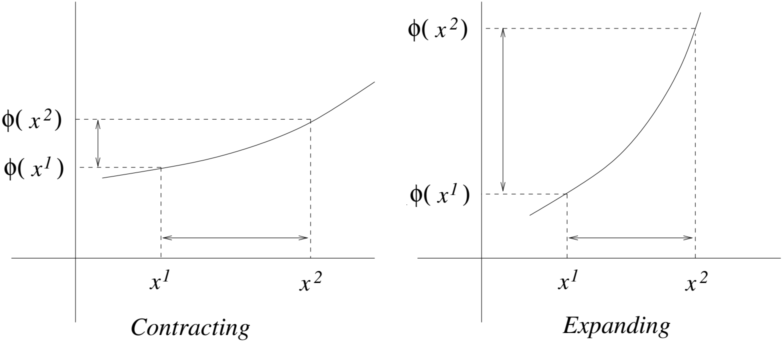
- $T \subset [a, b]$ *be a continuous real-valued function*
- *Contracting condition: it exists* $q \in \mathbb{R}$, $q < 1$, *such that:*

$$\forall x^1, x^2 \in [a, b] \quad \textit{then} \quad |\phi(x^1) - \phi(x^2)| \leq q|x^1 - x^2|$$

Then, if $x^0 \in [a, b]$ *and* $x^{k+1} = \phi(x^k)$ *it follows that:*

1. *There exists a* **unique fixed point** x^* *of* ϕ
2. *For any* $k \geq 0$
$$|x^{k+1} - x^*| \leq q^{k+1}|x^0 - x^*|$$
3. *For any* $x^0 \in [a, b]$ *it follows that* $\{x^k\} \to x^*$

Contracting condition



Newton's method

Proof:

1. Since $\phi(a), \phi(b) \in [a, b]$, the function $F(x) = \phi(x) - x$ satisfies $F(a) = \phi(a) - a \geq 0$ and $F(b) = \phi(b) - b \leq 0$. Since F is continuous, according to Bolzano's theorem, there is, at least, one point x^* such that $F(x^*) = 0$, this is $\phi(x^*) = x^*$

   To see that x^* is unique, assume that there are two distinct fixed points $x_1^* \neq x_2^*$: $\phi(x_i^*) = x_i^*$ for $i = 1, 2$, then

   $$0 < |x_1^* - x_2^*| = |\phi(x_1^*) - \phi(x_2^*)| \leq q|x_1^* - x_2^*|$$

   which is a contradiction, since $q < 1$

2. The inequality holds for $k = 0$ since

   $$|x^1 - x^*| = |\phi(x^0) - \phi(x^*)| \leq q|x^0 - x^*|$$

   Suppose that it holds up to a certain k

   $$|x^k - x^*| \leq q^k|x^0 - x^*|$$

   Then

   $$|x^{k+1} - x^*| = |\phi(x^k) - \phi(x^*)| \leq q|x^k - x^*| \leq q^{k+1}|x^0 - x^*|$$

3. The convergence follows from the inequality, since $q < 1$, so $q^k \to 0$

□

Newton's method

The next lemma deals with **sufficient conditions on ϕ for being a contraction**

Lemma

Suppose that $\phi : [a, b] \to T \subset \mathbb{R}$ with $T \subset [a, b]$ has a continuous derivative on $[a, b]$, ($\phi \in C^1$). If $|\phi'(x)| < 1$ for every $x \in [a, b]$ then ϕ is a contraction

Proof:

Let $x^1, x^2 \in [a, b]$. Then, by the Mean Value Theorem

$$\phi(x^1) = \phi(x^2) + \phi'(\tilde{x})(x^1 - x^2), \quad \tilde{x} \in < x^1, x^2 >$$

where $< x^1, x^2 > \equiv [\min(x^1, x^2), \max(x^1, x^2)]$

Hence

$$|\phi(x^1) - \phi(x^2)| = |\phi'(\tilde{x})| \; |x^1 - x^2|$$

Taking

$$q = \max_{a \leq x \leq b} |\phi'(x)| < 1 \quad \Rightarrow \quad |\phi(x^1) - \phi(x^2)| \leq q \; |x^1 - x^2|, \quad \forall x^1, x^2 \in [a, b]$$

and the Lemma is proved. □

Newton's method

Theorem

Let h, γ be two real valued continuously differentiable functions on $S = [a, b] \subset \mathbb{R}$, and suppose that

1. $h(a)\,h(b) < 0$
2. *For all $x \in S$ the following conditions are satisfied:*
    - $h'(x) > 0$ *(h is monotone increasing on S)*
    - $\gamma(x) > 0$
    - $0 \leq 1 - [\gamma(x)h(x)]' \leq q < 1$

Consider the sequence $\{x^k\}$ defined by

$$x^{k+1} = x^k - h(x^k)\gamma(x^k), \quad k \geq 0$$

with $x^0 \in S$, then $\{x^k\}$ converges to a solution x^* of $h(x) = 0$

Remark: Recall that Newton's method applied to solve $f'(x) = 0$ is:

$$x^0 \in \mathbb{R}, \quad x^{k+1} = x^k - f'(x^k)\frac{1}{f''(x^k)}, \quad k = 0, 1, 2, \ldots$$

Newton's method

Proof:

Define

$$\phi(x) = x - \gamma(x)h(x) \quad \Rightarrow \quad \phi'(x) = 1 - [\gamma(x)h(x)]'$$

By hypothesis, we have

$$0 \leq \phi'(x) \leq q < 1, \quad \forall x \in S$$

so ϕ is monotone nondecreasing on S

The function h is monotone increasing on S and satisties $h(a) < 0$, $h(b) > 0$, hence $\phi(a) = a - \gamma(a)h(a) > a$, and $\phi(b) = b - \gamma(b)h(b) < b$, so

$$a < \phi(x) < b, \quad \forall x \in S = [a, b]$$

Moreover $|\phi'(x)| < 1$ and, by the preceeding Lemma, it follows that ϕ is a contractor on S, so it has a unique fixed point $\overline{x} \in S$ and the sequence

$$x^{k+1} = \phi(x^k) = x^k - \gamma(x^k)h(x^k)$$

converges to $\overline{x}$.

Finally, since $\gamma(x) > 0$, observe that x^* is a fixed point of ϕ if and only if $h(x^*) = 0$, thus $\{x^k\}$ converges to a solution of $h(x) = 0$. □

Newton's method

Now we can state **sufficient conditions for the convergence of Newton's method**

Corollary

Let $h(x) = f'(x)$, $\gamma(x) = 1/f''(x)$ *with* $f \in C^2$ *in* $S = [a, b]$

Assume that h and γ *fulfil the hypotheses of the preceeding Theorem:*

- $h(a)\,h(b) < 0, \quad (f'(a)\,f'(b) < 0)$
- $h'(x) > 0, \quad (f''(x) > 0)$
- $\gamma(x) > 0, \quad (1/f''(x) > 0 \;\Leftrightarrow\; f''(x) > 0)$
- $0 \leq 1 - [\gamma(x)h(x)]' \leq q < 1, \quad (0 \leq f'(x)f'''(x)/(f''(x))^2 \leq q < 1)$

then

$$x^{k+1} = x^k - \gamma(x^k)h(x^k) = x^k - \frac{f'(x^k)}{f''(x^k)} \longrightarrow x^*$$

with $f'(x^*) = 0$

Rates of convergence

- Assume that a method, as applied to a minimization problem P, generates sequence of iterates converging to the solution set X^* of the problem (that can be a set of points)

- The error function $err(x)$ measures the quality of an approximate solution $\boldsymbol{x} \in \mathbb{R}^n$

- There are several choices of the error function. We can use, for instance:
  - The **distance from the approximate solution $x \in \mathbb{R}^n$ to the solution set**
  $$err(\boldsymbol{x}) = inf_{\boldsymbol{x}^* \in X^*} \|\boldsymbol{x} - \boldsymbol{x}^*\|$$
  - Another choice of the error function could be **the residual, in terms of the objective function and the equality constraints $(g_i(x) = 0)$**
  $$err(\boldsymbol{x}) = max\{|f(\boldsymbol{x}) - f^*| \, , |g_1(\boldsymbol{x})|, ..., |g_m(\boldsymbol{x})|\}$$
  f^* being the optimal value of the objective function $(f^* = f(x^*))$

- For a properly chosen error function, convergence of the iterates to the solution set implies that
$$r_n = err(\boldsymbol{x}_n) \to 0$$

Rate of convergence

- In addition to proving convergence of a certain algorithm, it is also important to know the **rate of convergence**.
- We measure the quality of convergence by the rate at which $\{r_n\}$ tends to zero
- Let $\{\boldsymbol{x}^k\}$, with $\boldsymbol{x}^k \in \mathbb{R}^n$ be a sequence that converges to $\boldsymbol{x}^*$ with $\boldsymbol{x}^k \neq \boldsymbol{x}^*$ for all sufficiently large k. **If there exists numbers $p, \alpha \in \mathbb{R}$, with $\alpha \neq 0$, such that**
$$\lim_{k\to\infty} \frac{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^p} = \alpha,$$
then it is said that **the order of convergence of $\{\boldsymbol{x}^k\}$ to $\boldsymbol{x}^*$ is p**, and $\|\boldsymbol{x}^k - \boldsymbol{x}^*\|$ is the error of the kth approximant.
- If $p = 1$ the rate of convergence is said to be **linear**, if $p = 2$ **quadratic** and, in general, if $p > 1$ **superlinear**.

Newton's method convergence

Theorem

Assume that the hypotheses of the last Theorem and Corollary hold (pgs. 12 and 14), and that **the sequence $\{x^k\}$, $x^k \in \mathbb{R}$, generated by Newton's method** *converges to a point x^* that satisfies $h(x^*) = 0$. Then* **the rate of convergence of $\{x^k\}$ towards x^* is quadratic**

Proof:

The point x^* is a solution of $h(x) = 0$ if and only if is a fixed point of

$$\phi(x) = x - \frac{h(x)}{h'(x)}$$

By the Mean Value Theorem

$$x^{k+1} - x^* = \phi(x^k) - \phi(x^*) = \phi'(\xi^k)(x^k - x^*), \quad \xi^k \in < x^k, x^* >$$

If we take into account that

$$\phi'(x) = 1 - \frac{(h'(x))^2 - h(x)h''(x)}{(h'(x))^2} = \frac{h(x)h''(x)}{(h'(x))^2}$$

it follows

$$|x^{k+1} - x^*| = \frac{|h(\xi^k)h''(\xi^k)|}{(h'(\xi^k))^2}|x^k - x^*|$$

Newton's method convergence (cont.)

Since

$$|h(\xi^k)| = |h(\xi^k) - h(x^*)| = |h'(\eta^k)|\ |\xi^k - x^*| \leq |h'(\eta^k)|\ |x^k - x^*|$$

with $\eta^k \in< \xi^k, x^* >$, for the last inequality, we have used that $\xi^k \in< x^k, x^* >$, hence

$$|x^{k+1} - x^*| \leq \frac{|h''(\xi^k)h'(\eta^k)|}{(h'(\xi^k))^2}|x^k - x^*|^2$$

Taking

$$\beta = \sup_x \frac{|h''(x)h'(x)|}{(h'(x))^2}$$

we get

$$|x^{k+1} - x^*| \leq \beta|x^k - x^*|^2$$

□

The secant method

A closely related root-finding method can be obtained by approximating the second derivative $f''(x)$ by

$$f''(x^k) \simeq \frac{f'(x^k) - f'(x^{k-1})}{x^k - x^{k-1}}$$

in Newton's method formula. In this way we get **secant method:**

$$x^{k+1} = x^k - \frac{f'(x^k)(x^k - x^{k-1})}{f'(x^k) - f'(x^{k-1})}$$

If $f''' \neq 0$ then, it can be proved that

$$\lim_{k \to \infty} \frac{|x^{k+1} - x^*|}{|x^k - x^*|^\tau} = \left| \frac{2f''(x^*)}{f'''(x^*)} \right|^{1/\tau}$$

where $\tau = (1 + \sqrt{5})/2 = 1.618... > 1$ is a solution of the equation $t^2 - t - 1 = 0$

Thus (for large values of k) the secant method is **superlinear**

Line search methods

Zero-order line search methods

- We are going to consider numerical methods to **solve the problem**

$$\min_x \{f(x) \ : \ a \leq x \leq b\}, \quad -\infty < a < b < \infty$$

  ***f* being, at least, a continuous function**

- These procedures usually are called **line search methods** and, in general, **use only the values of *f* and not the derivatives**

- Line search methods are a component of almost all usual methods for multidimensional optimization

Polynomial approximation methods: the quadratic method

Let f be the function whose minimum is sought. The **basis of the quadratic method is to approximate f by**

$$\phi(x) = a + bx + cx^2$$

- Suppose that we evaluate f at three points $x_1 < x_2 < x_3$
- Letting $f(x_i) = \phi(x_i)$, $i = 1, 2, 3$ we can solve for the coefficients a, b, c
- The minimum of the quadratic function ϕ (if it has a minimum) can be found analytically by setting $\phi'(x) = 0$, and, for a first approximation of a minimum of f we obtain
$$\tilde{x} = -\frac{b}{2c}$$
- If $c < 0$, the quadratic function is actually a parabola with a maximum and so the point $\tilde{x}$ obtained is **unusable**
- We must assume that $c > 0$. A situation that will ensure that c is positive is
$$f(x_1) > f(x_2), \quad \text{and} \quad f(x_3) > f(x_2)$$
- If these conditions hold we can also ensure that the local minimum of f is between x_1 i x_3

The quadratic method

- Under the above conditions, the minimum of ϕ so found will also satisfy

$$f(x_1) > \phi(\tilde{x}) \quad \text{and} \quad f(x_3) > \phi(\tilde{x})$$

- Now, **consider the four points** $(x_1, f(x_1))$**,** $(x_2, f(x_2))$**,** $(x_3, f(x_3))$**,** $(\tilde{x}, f(\tilde{x}))$

- **Choose as the new** x_2 one of the four points, at which f has been computed, which yielded the **lowest value of** f and let the **new** x_1 **and** x_3 **be the two points adjacent to the new** x_2 from the left and right, respectively. Repeat the iteration

- This algorithm can be **terminated** if either

$$|f(\tilde{x}) - \phi(\tilde{x})| < \epsilon$$

  for some tolerance $\epsilon > 0$, or if estimates of the minimum point in two or more succesive iterations are closer than some predetermined distance

- If $\tilde{x} = x_2$ the algorithm will not evaluate new points, although x_2 may not be a local minimum of f. In such a degenerate case, some perturbations on $\tilde{x}$ are needed in order to proceed with the computations

Exercise

Exercise 4. To be delivered before 11-X-2021 as: `Ex04-YourSurname.pdf`

Let f be a real function on $\mathbb{R}^n$. Also let $x_0 \in \mathbb{R}^n$, $z \in \mathbb{R}^n$,and $\theta \in \mathbb{R}$. Define

$$F(\theta) = f(x_0 + \theta z)$$

and suppose that we are looking for the minimum of F (that is, for the minimum of f in the direction z through the point x_0). Let $x_0 + \theta_1 z$, $x_0 + \theta_2 z$ and $x_0 + \theta_3 z$ be three points where f is evaluated. Show that the minimum predicted by applying the quadratic approximation method is $x_0 + \theta^* z$, where

$$\theta^* = \frac{[\theta_2^2 - \theta_3^2]F(\theta_1) + [\theta_3^2 - \theta_1^2]F(\theta_2) + [\theta_1^2 - \theta_2^2]F(\theta_3)}{2[(\theta_2 - \theta_3)F(\theta_1) + (\theta_3 - \theta_1)F(\theta_2) + (\theta_1 - \theta_2)F(\theta_3)]}$$

and it is indeed the minimum of the parabola passing through the above three points if

$$\frac{(\theta_2 - \theta_3)F(\theta_1) + (\theta_3 - \theta_1)F(\theta_2) + (\theta_1 - \theta_2)F(\theta_3)}{(\theta_2 - \theta_3)(\theta_3 - \theta_1)(\theta_1 - \theta_2)} < 0$$

Polynomial approximation methods: the cubic (first-order) method

In the **cubic method** the function f is approximated by

$$\phi(x) = a + bx + cx^2 + dx^3$$

We will assume that the first derivatives of f can be evaluated

We start at a point x_1 such that $f'(x_1) < 0$. Then we compute $x_2 > x_1$ such that

$$f'(x_2) \geq 0, \quad \text{or} \quad f(x_2) > f(x_1)$$

The coefficients a, b, c and d of the function ϕ can be computed solving the system

$$\begin{aligned} f(x_1) &= a + bx_1 + cx_1^2 + dx_1^3 \\ f'(x_1) &= b + 2cx_1 + 3dx_1^2 \\ f(x_2) &= a + bx_2 + cx_2^2 + dx_2^3 \\ f'(x_2) &= b + 2cx_2 + 3dx_2^2 \end{aligned}$$

The solution of these equations can be found by a simple change of variables. Define

$$z = x - x_1$$

and, instead of f and ϕ, use the functions

$$g(z) = f(x_1 + z), \quad \psi(z) = \phi(x_1 + z)$$

The cubic method

It can be seen that

$$\psi'(z) = g'(0) - \frac{2z}{\lambda}(g'(0) + \alpha) + \frac{z^2}{\lambda^2}(g'(0) + g'(\lambda) + 2\alpha)$$

where $\lambda = x_2 - x_1$ and

$$\alpha = \frac{3(g(0) - g(\lambda))}{\lambda} + g'(0) + g'(\lambda)$$

The point that satisfies $\psi'(z) = \phi'(x_1 + z) = 0$ (minimum of ϕ) is

$$\tilde{z} = \lambda(1 - \beta)$$

where

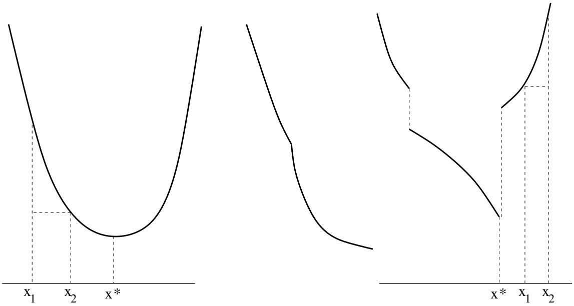
$$\beta = \frac{g'(\lambda) + (\alpha^2 + g'(0)g'(\lambda))^{1/2} - \alpha}{g'(\lambda) - g'(0) + 2(\alpha^2 + g'(0)g'(\lambda))^{1/2}}$$

If $|g'(\tilde{z})| < \epsilon$ the procedure is terminated; otherwise the algorithm must be restarted by a procedure similar to the one of the quadratic method

Unimodal functions

Let $L = [a, b] \subset \mathbb{R}$ be a closed interval. A real-valued function f is said to be **unimodal** on L if there exist x^* $\in L$ such that x^* minimizes f on L, and for any two points $x_1, x_2 \in L$ such that $x_1 < x_2$ we have

$$\begin{aligned} x_2 \leq x^* \quad &\Rightarrow \quad f(x_1) > f(x_2), \\ x^* \leq x_1 \quad &\Rightarrow \quad f(x_2) > f(x_1). \end{aligned}$$



In another words, f is unimodal on $L = [a, b]$ if it possesses a unique local minimum x^* on $[a, b]$, which implies that that f is strictly decreasing in $[a, b]$ to the left of x^* and strictly increasing in $[a, b]$ to the right of x^*

The line search method

The startegy of the **zero-order line search method for unimodal functions** is based in the following. Choose, somehow, two points x_1 and x_2 such that $a < x_1 < x_2 < b$ and compute the values of f at these points.

The basic observation is that:

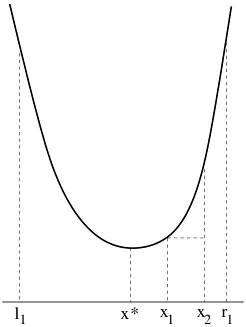
- **If $f(x_1) \leq f(x_2)$, then x^* is to the left of x_2, $(x^* < x_2)$**
- **If $f(x_1) \geq f(x_2)$, then x^* is to the right of x_1, $(x^* > x_1)$**

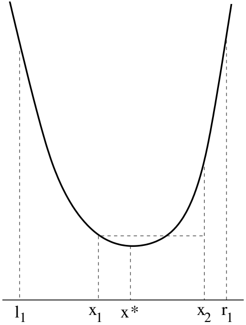
Line search algorithm

Let $L = \{x \mid l_1 \leq x \leq r_1\} = [l_1, r_1]$ and $x_1, x_2 \in L$ two points such that $x_1 < x_2$. We evaluate the unimodal function f at both points: $f(x_1)$ and $f(x_2)$. Then, there are three possibilities:

- If $f(x_1) < f(x_2)$. Since f is unimodal, it follows that either $x^* \leq x_1 < x_2$ or $x_1 \leq x^* \leq x_2$. In both cases $x^* \in [l_1, x_2]$
- If $f(x_1) > f(x_2)$. Since f is unimodal, it follows that $x^* \in [x_1, r_1]$
- If $f(x_1) = f(x_2)$. Since f is unimodal, it follows that $x^* \in [x_1, x_2]$

Line search algorithm

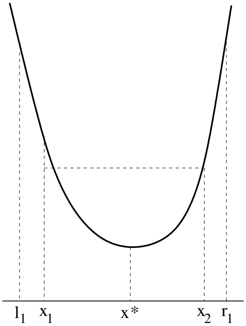




$$f(x_1) < f(x_2)$$

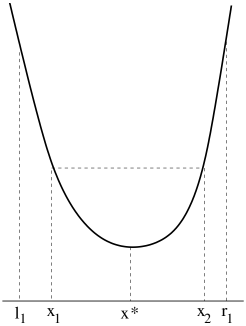
$$x^* \in [l_1, x_2]$$

Line search algorithm



$$f(x_1) > f(x_2)$$

$$x^* \in [x_1, r_1]$$



$$f(x_1) = f(x_2)$$

$$x^* \in [x_1, x_2]$$

The line search method

- In all the cases, after the first two function evaluations, a portion of L to the right of x_2 or the left of x_1 can be eliminated from further search.
- So we have found a new interval $[l_2, r_2]$ such that $x^* \in [l_2, r_2]$. Then we repeat the procedure iteratively
- We can ensure, at least, linear convergence if the lengths of subsequent uncertainty segments tend to 0
- If x_1, x_2 are chosen to split $[l_n, r_n]$ into three equal parts, we ensure $|r_{n+1} - l_{n+1}| = (2/3)|r_n - l_n|$, so

$$|x_n - x^*| \leq \left(\frac{2}{3}\right)^n |b - a|$$

Fibonacci numbers

Fibonacci numbers, F_k, are defined by the following recurrence relation:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_k &= F_{k-1} + F_{k-2}, \quad k = 2, 3, \dots \end{aligned}$$

The first Fibonacci numbers are: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,...

It can be shown that

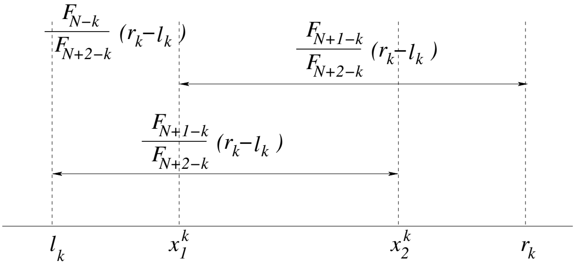
$$\lim_{n\to\infty} \frac{F_{n-1}}{F_n} = \frac{1}{\tau} = \frac{\sqrt{5}-1}{2} = 0.6180339\dots$$

where $\tau = 1.6180339\dots$ is the golden ratio

The Fibonacci method

- Let N be the total number of points at which the unimodal function f will be evaluated. For N function evaluations, the Fibonacci method does $N-1$ interval reductions (iterations)
- Among all the search procedures with N function evaluations, the Fibonacci method minimizes the length of the possible interval remaining after N function evaluations, and containing the sought minimum
- At iteration number k the interval containing x^* is $[l_k, r_k]$
- For $k = 1, 2, ..., N-1$ the function values are compared at the two points

$$x_1^k = l_k + \frac{F_{N-k}}{F_{N+2-k}}(r_k - l_k), \quad x_2^k = l_k + \frac{F_{N-k+1}}{F_{N+2-k}}(r_k - l_k) \qquad (1)$$



$\frac{F_{N-k}}{F_{N+2-k}}(r_k - l_k)$
$\frac{F_{N+1-k}}{F_{N+2-k}}(r_k - l_k)$
$\frac{F_{N+1-k}}{F_{N+2-k}}(r_k - l_k)$
l_k
x_1^k
x_2^k
r_k

The Fibonacci method. Example

- Consider the function $f(x) = (x-3)^2$
- Set $N = 4,\ L = [l_1, r_1] = [0, 10]$
- According to (1)

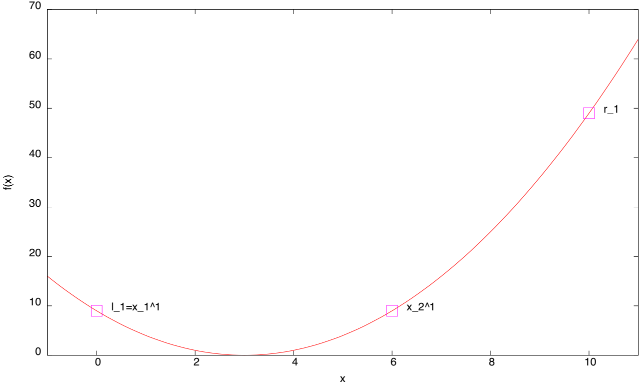
$$x_1^1 \ = \ l_1 + \frac{F_3}{F_5}(r_1 - l_1) = \frac{2}{5}(10-0) = 4, \quad x_2^1 \ = \ l_1 + \frac{F_4}{F_5}(r_1 - l_1) = \frac{3}{5}(10-0) = 6$$

- Compute $f(x_1^1) = 1,\ f(x_2^1) = 9$
- Since f is unimodal and $f(x_1^1) < f(x_2^1)$, then $x^* \in [l_1, x_2]$, so: $[l_2, r_2] = [l_1, x_2^1] = [0, 6]$, this is: $l_2 = 0$ i $r_2 = 6$

i	l_i	x_1^i	$f(x_1^i)$	x_2^i	$f(x_2^i)$	r_i
1	0					10
		4	1	6	9	
2	0					6

The Fibonacci method. Example

$$[l_1, r_1] = [0, 10] \longrightarrow [0, 6] = [x_1^1, x_2^1]$$



The Fibonacci method. Example

- According to (1)

$$x_1^2 \;=\; l_2+\frac{F_2}{F_4}(r_2-l_2) = 0+\frac{1}{3}(6-0) = 2, \qquad x_2^2 \;=\; l_2+\frac{F_3}{F_4}(r_2-l_2) = 0+\frac{2}{3}(6-0) = 4$$

- Note that $x_2^2 = x_1^1$, and that $f(x_1^2) = f(x_2^2) = 1$, so $x^* \in [x_1, x_2]$ and $l_3 = 2$ and $r_3 = 4$

i	l_i	x_1^i	$f(x_1^i)$	x_2^i	$f(x_2^i)$	r_i
2	0					6
		2	1	4	1	
3	2					4

- According to (1)

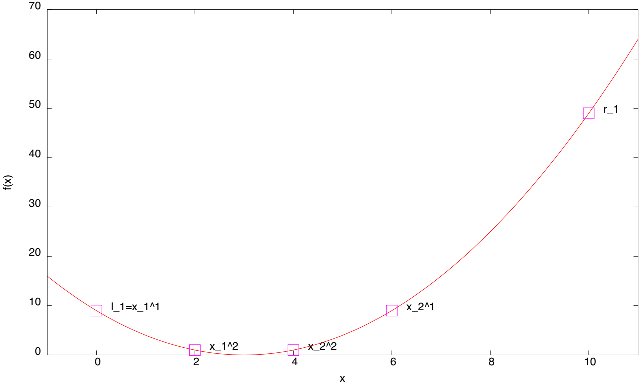
$$x_1^3 \;=\; 2+\frac{F_1}{F_3}(6-2) = 2+\frac{1}{2}(6-2) = 4, \qquad x_2^3 \;=\; 2+\frac{F_2}{F_3}(6-2) = 2+\frac{1}{2}(6-2) = 4$$

- The final interval is $[2, 4]$. Note that

$$r_4 - l_4 = 4 - 2 = \frac{10-0}{5} = \frac{r_1 - l_1}{F_5}$$

The Fibonacci method. Example

$$[l_1, r_1] = [0, 10] \longrightarrow [x_1^1, x_2^1] = [0, 6] \longrightarrow [x_1^2, x_2^2] = [2, 4] \longrightarrow [x_1^3, x_2^3] = [4, 4]$$



The Fibonacci method. Remarks

- Except for $k = 1$ in all the steps of the methods the function f has already been evaluated in a previous iteration at one of the two points

- Note that the points x_1^k and x_2^k are placed symmetrically in the interval $[l_k, r_k]$, since

$$\begin{aligned} x_2^k - l_k &= \frac{F_{N+1-k}}{F_{N+2-k}}(r_k - l_k) = \frac{F_{N+2-k} - F_{N-k}}{F_{N+2-k}}(r_k - l_k) \\ &= r_k - l_k - \frac{F_{N-k}}{F_{N+2-k}}(r_k - l_k) = r_k - x_1^k \end{aligned}$$

- At the last iteration $(k = N - 1)$ formulas (1) give

$$x_1^{N-1} = x_2^{N-2} = l_{N-1} + \frac{1}{2}(r_{N-1} - l_{N-1}),$$

and no further interval reduction is possible

The Fibonacci method. Remarks

- After N function evaluations, the length of the interval containing x^* is

  $$r_N - l_N = \frac{r_1 - l_1}{F_{N+1}}$$

  To see this equality, recall that

  $$r_{k+1} - l_{k+1} = \frac{F_{N+1-k}}{F_{N+2-k}}(r_k - l_k)$$

  So, the product of all the contracting factors from $k = 1$ up to $k = N$ is

  $$\frac{F_{N+1-1}}{F_{N+2-1}}\,\frac{F_{N+1-2}}{F_{N+2-2}}\,\frac{F_{N+1-3}}{F_{N+2-3}}\cdots\cdots\frac{F_{N+1-N+1}}{F_{N+2-N+1}}\,\frac{F_{N+1-N}}{F_{N+2-N}} =$$

  $$= \frac{F_N}{F_{N+1}}\,\frac{F_{N-1}}{F_N}\,\frac{F_{N-2}}{F_{N-1}}\cdots\cdots\frac{F_2}{F_3}\,\frac{F_1}{F_2} = \frac{1}{F_{N+1}}F_1 = \frac{1}{F_{N+1}}$$

  In this way, we can bracket the minimum of any unimodal function

  - within 1% of the starting interval by 11 function evaluations ($F_{12} = 144$)
  - within 0.1% by 16 evaluations ($F_{17} = 1597$)

- Among all the search procedures with N function evaluations, the Fibonacci method minimizes the length of the possible interval remaining after N function evaluations, and containing the sought minimum

The golden section method

- One disadvantage of the Fibonacci method is that the number of function evaluations N must be known prior to starting the search
- This requirement is not necessary in a related technique, called **the golden section method**, which is an approximation of the Fibonacci search
- The golden section method places the points at which the function is to be evaluated by:

$$x_1^{kG} = l_k + \frac{\tau - 1}{\tau}(r_k - l_k), \quad x_2^{kG} = l_k + \frac{1}{\tau}(r_k - l_k)$$

where $\tau = 1.6180339\ldots$ is the golden ratio

The golden search method

- As the Fibonacci method, the golden section method also places the points symmetrically:

$$x_2^{kG} - l_k = \frac{1}{\tau}(r_k - l_k), \quad r_k - x_1^{kG} = r_k - l_k - \frac{\tau - 1}{\tau}(r_k - l_k) = \frac{1}{\tau}(r_k - l_k)$$

- The golden section method reduces the initial interval containing the minimum by a factor $1/\tau^{N-1}$ in front of the factor of the Fibonacci method that is $1/F_{N+1}$.

- It can be shown that

$$\lim_{n \to \infty} \frac{F_{N+1}}{\tau^{N-1}} = \frac{\tau^2}{\sqrt{5}} = 1.17...$$

  Thus, for large N the golden section method yields a final interval that is some 17% larger that the Fibonacci method

n-dimensional unconstrained optimization

Descent methods

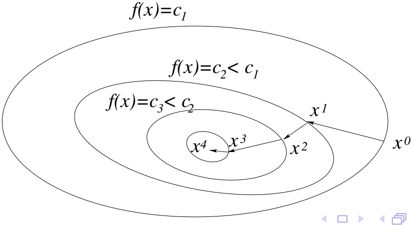
- We consider methods for **unconstrained optimization problems**
- Most of the algorithms for these problems rely on an important idea: the iterative descent
- Let

  $$f : \mathbb{R}^n \longrightarrow \mathbb{R}$$

  be, at least, a continuosly differentiable function. The iterative descent method is:

  - Take an initial guess $\boldsymbol{x}^0 \in \mathbb{R}^n$
  - Generate a **sequence of points x^1, x^2,... such that the value of f is decreased at each iteration**, this is

    $$f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k), \quad k = 0, 1, 2, \ldots$$



Recall that the gradient...

- The gardient of a continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the vectorfield

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x_n} \right)^T$$

- If $\boldsymbol{s} \in \mathbb{R}^n$ is a unitary vector, the directional derivative of $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at a point $\boldsymbol{x} \in \mathbb{R}^n$ in the direction of $\boldsymbol{s}$, which measures the rate of change of the function along $\boldsymbol{s}$, is equal to

$$Df(\boldsymbol{x}, \boldsymbol{s}) = \lim_{\lambda \to 0} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{s}) - f(\boldsymbol{x})}{\lambda} = (\nabla f(\boldsymbol{x}))^T \boldsymbol{s} \in \mathbb{R}$$

- Since the directional derivative is

$$(\nabla f(\boldsymbol{x}))^T \boldsymbol{s} = \|\nabla f(\boldsymbol{x})\| \|\boldsymbol{s}\| \cos\theta = \|\nabla f(\boldsymbol{x})\| \cos\theta$$

  the maximum rate of change of f at the point x occurs when $\cos\theta$ is maximized, this is when $\theta = 0$ and $\theta = \pi$.

- Thus, the greatest increase occurs in the direction of $\nabla f(\boldsymbol{x})$, and the **greatest decrease occurs in the direction of** $-\nabla f(x)$

Gradient methods. Basic principle

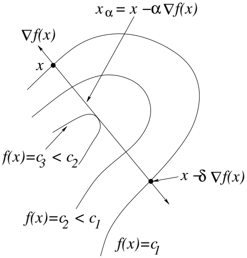
Given $\boldsymbol{x} \in \mathbb{R}^n$ with $\nabla f(\boldsymbol{x}) \neq 0$, consider the half line

$$\boldsymbol{x}_\alpha = \boldsymbol{x} - \alpha \nabla f(\boldsymbol{x}), \quad \alpha \geq 0$$

According to Taylor's formula, and since $\nabla f(\boldsymbol{x})^T \nabla f(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|^2$, we have

$$\begin{aligned} f(\boldsymbol{x}_\alpha) &= f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T (\boldsymbol{x}_\alpha - \boldsymbol{x}) + o(\|\boldsymbol{x}_\alpha - \boldsymbol{x}\|)^1 \\ &= f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \left(-\alpha \nabla f(\boldsymbol{x})\right) + o(\alpha \|\nabla f(\boldsymbol{x})\|) = \\ &= f(\boldsymbol{x}) - \alpha \|\nabla f(\boldsymbol{x})\|^2 + o(\alpha \|\nabla f(\boldsymbol{x})\|) = f(\boldsymbol{x}) - \alpha \|\nabla f(\boldsymbol{x})\|^2 + o(\alpha) \end{aligned}$$

When we are **close** to the minimum, and since $\nabla f(\boldsymbol{x}) \neq 0$, for α within a certain (small enough) positive interval $0 \leq \alpha \leq \delta$, we have: $f(\boldsymbol{x}_\alpha) < f(\boldsymbol{x})$



[1] $g(\alpha) = o(\alpha) \quad \Leftrightarrow \quad \lim_{\alpha \to 0} \frac{g(\alpha)}{\alpha} = 0$

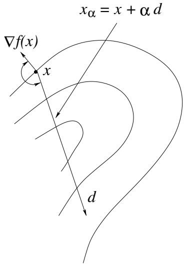
Gradient methods. Basic principle

The above procedure can be generalised. Consider the half line

$$\boldsymbol{x}_\alpha = \boldsymbol{x} + \alpha \boldsymbol{d}, \quad \alpha \geq 0$$

where **the direction $\boldsymbol{d} \in \mathbb{R}^n$ makes an angle with $\nabla f(x)$ between 90^o and 270^o**, this is

$$\nabla f(\boldsymbol{x})^T \boldsymbol{d} < 0$$



The inequality $\nabla f(\boldsymbol{x})^T \boldsymbol{d} < 0$ is known as the **descent condition**

According to Taylor's formula

$$f(\boldsymbol{x}_\alpha) = f(\boldsymbol{x}) + \alpha \nabla f(\boldsymbol{x})^T \boldsymbol{d} + o(\alpha)$$

For positive and small enough values of α ($0 \leq \alpha \leq \delta$), we also have

$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) < f(\boldsymbol{x})$$

General gradient methods

- The **general expression of a gradient method** is

  $$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k, \quad k = 0, 1, \ldots$$

  where, if $\nabla f(\boldsymbol{x}^k) \neq 0$, the direction $\boldsymbol{d}^k$ is chosen so that

  $$\nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k < 0$$

  and the stepsize is $\alpha^k > 0$

- The name "gradient methods" is due to the relation between $\boldsymbol{d}^k$ and $\nabla f(\boldsymbol{x}^k)$

- When $\nabla f(\boldsymbol{x}^k) = 0$ (or $\|\nabla f(\boldsymbol{x}^k)\| \leq$ tolerance) the method stops

- The gradients methods that will be considered are also descent methods, this is, the step size α^k is such that

  $$f(\boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k) < f(\boldsymbol{x}^k), \quad k = 0, 1, \ldots$$

The descent direction $\boldsymbol{d}^k$ for general gradient methods

- There are many possibilities for choosing the direction $\boldsymbol{d}^k$, and also the step size α^k

- We consider **general gradient methods**, $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$, with the following descent direction $\boldsymbol{d}^k = -D^k \nabla f(\boldsymbol{x}^k)$, this is:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k D^k \nabla f(\boldsymbol{x}^k)$$

  where D^k is a positive definite symmetric matrix ($\boldsymbol{z}^T D^k \boldsymbol{z} > 0, \forall \boldsymbol{z} \neq 0$ and $D^T = D$)

- Since

$$\boldsymbol{d}^k = -D^k \nabla f(\boldsymbol{x}^k)$$

  the descent condition $\nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k < 0$ becomes

$$-\nabla f(\boldsymbol{x}^k)^T D^k \nabla f(\boldsymbol{x}^k) < 0 \quad \Leftrightarrow \quad \nabla f(\boldsymbol{x}^k)^T D^k \nabla f(\boldsymbol{x}^k) > 0$$

  which holds, since D^k is positive definite

General gradient methods. Summary

1. **General gradient methods:** $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k D^k \nabla f(\boldsymbol{x}^k)$

   - D^k selection

     - Steepest descent: $D^k = Id$
     - General Newton's method: $D^k = (\nabla^2 f(\boldsymbol{x}^k))^{-1}$
     - Modified Newton's method: $D^k = (\nabla^2 f(\boldsymbol{x}^0))^{-1}$
     - Discretized Newton's method: $D^k \approx (\nabla^2 f(\boldsymbol{x}^k))^{-1}$
     - Diagonally scaled steepest descent. Diagonal approximation to Newton's method:

       $$D^k = diag(d_1^k, ..., d_n^k) \text{ with } d_i^k \approx \left(\partial^2 f(\boldsymbol{x}^k)/\partial x_i^2\right)^{-1}$$

   - α^k selection

     - Constant stepsize
     - Minimization rule*
     - Limited minimization rule*
     - Successive stepsize reduction. Armijo's rule*

   * convergent if $\{\boldsymbol{d}^k\}$ is gradient related to $\{\boldsymbol{x}^k\}$

2. **The Gauss-Newton method** (for the sum of squares of functions)

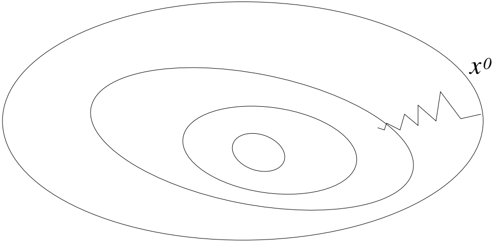
The steepest descent method

- The simplest choice for D^k is

$$D^k = Id, \quad k = 0, 1, \dots \quad \Rightarrow \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k), \quad k = 0, 1, \dots$$

  where I is the identity matrix. In this case the method is known as the **steepest descent method**

- This choice often leads to slow convergence



The steepest descent method

The name "steepest descent" of the above method is due to the following.

Recall that if

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \alpha \mathbf{d}^{k}, \quad \alpha \geq 0$$

then

$$f(\mathbf{x}^{k+1}) = f(\mathbf{x}^{k}) + \alpha \nabla f(\mathbf{x}^{k})^{T} \mathbf{d} + o(\alpha),$$

so the rate of chage of f at $\mathbf{x}^{k}$ is $\alpha \nabla f(\mathbf{x}^{k})^{T} \mathbf{d}$

Consider any unitary direction $\mathbf{d} \in \mathbb{R}^{n}$, $(\|\mathbf{d}\| = 1)$. According to Schwartz inequality[2], the rate of change of f verifies

$$\nabla f(\mathbf{x}^{k})^{T} \mathbf{d} \leq \|\nabla f(\mathbf{x}^{k})\| \; \|\mathbf{d}\| = \|\nabla f(\mathbf{x}^{k})\|$$

If we set

$$\mathbf{d} = \frac{\nabla f(\mathbf{x}^{k})}{\|\nabla f(\mathbf{x}^{k})\|}$$

then

$$\nabla f(\mathbf{x}^{k})^{T} \mathbf{d} = \|\nabla f(\mathbf{x}^{k})\|$$

therefore, $-\nabla f(\mathbf{x}^{k})$ **is the max-rate descending direction of** f **at** $\mathbf{x}^{k}$

[2]

$$|\mathbf{x}^{T}\mathbf{y}| \leq \|\mathbf{x}\|\|\mathbf{y}\|, \quad \text{and} \quad |\mathbf{x}^{T}\mathbf{y}| = \|\mathbf{x}\|\|\mathbf{y}\| \Leftrightarrow \mathbf{x} = \alpha \mathbf{y}$$

The general Newton's method

- The idea of Newton's method is to minimize, at each iteration, the quadratic approximation G of f around the current point $\boldsymbol{x}^k$. This quadratic approximation is given by

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

  By setting the derivative of $G(\boldsymbol{x})$ (with respect to $\boldsymbol{x}$) equal to zero, we get

  $$G'(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^k) + \nabla^2 f(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k) = 0$$

  from which, isolating $\boldsymbol{x}$ and setting $\boldsymbol{x}^{k+1} = \boldsymbol{x}$, we have

  $$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - (\nabla^2 f(\boldsymbol{x}^k))^{-1} \nabla f(\boldsymbol{x}^k)$$

  This is the "pure" Newton iteration ($\alpha^k = 1$)

- The **general Newton's procedure is**

  $$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k (\nabla^2 f(\boldsymbol{x}^k))^{-1} \nabla f(\boldsymbol{x}^k), \quad k = 0, 1, \ldots$$

  so

  $$D^k = -(\nabla^2 f(\boldsymbol{x}^k))^{-1}, \quad k = 0, 1, \ldots$$

  provided $\nabla^2 f(\boldsymbol{x}^k)$ is positive definite (if not some modification must be done)

- Usually the convergence of the method is fast and has not the zig-zagging behavior of the steepest descent method, but requires second derivatives of the function f

The general Newton's method

- **Remark:** Newton's method with $\alpha^k = 1$ determines the minimum of a quadratic positive definite function in ONLY ONE iteration.

  Let
  $$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + a$$
  with Q positive definite. Note that $\nabla^2 f(\boldsymbol{x}) = Q$ is constant.

  Let $\boldsymbol{x}^0$ be an arbitrary point in $\mathbb{R}^n$ and $\boldsymbol{x}^*$ the minimum of f. Then
  $$\nabla f(\boldsymbol{x}^0) = Q\boldsymbol{x}^0 + \boldsymbol{b}, \quad \text{and} \quad \nabla f(\boldsymbol{x}^*) = 0 = Q\boldsymbol{x}^* + \boldsymbol{b}$$
  From these two equations we get
  $$\boldsymbol{x}^* = -Q^{-1}\boldsymbol{b}, \qquad \boldsymbol{x}^0 = Q^{-1}\nabla f(\boldsymbol{x}^0) - Q^{-1}\boldsymbol{b}$$
  and
  $$\boldsymbol{x}^* = \boldsymbol{x}^0 - Q^{-1}\nabla f(\boldsymbol{x}^0) = \boldsymbol{x}^0 - (\nabla^2 f(\boldsymbol{x}^0))^{-1}\nabla f(\boldsymbol{x}^0)$$
  which is the first iteration of Newton's method starting at $\boldsymbol{x}^0$

The general Newton's method

Example Consider the quadratic function

$$f(\boldsymbol{x}) = (x - y + z)^2 + (-x + y + z)^2 + (x + y - z)^2,$$

that, if $\boldsymbol{x} = (x, y, z)$, can be written as

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x}, \quad \text{with} \quad Q = \begin{pmatrix} 6 & -2 & -2 \\ -2 & 6 & -2 \\ -2 & -2 & 6 \end{pmatrix}.$$

Let $\boldsymbol{x}^0 = (1/2, 1, 1/2)^T$, then

$$\nabla f(\boldsymbol{x}^0) = Q\boldsymbol{x}^0 = (0, 4, 0)^T,$$

and

$$\boldsymbol{x}^* = \boldsymbol{x}^0 - Q^{-1}\nabla f(\boldsymbol{x}^0) = \begin{pmatrix} 1/2 \\ 1 \\ 1/2 \end{pmatrix} - \begin{pmatrix} 1/4 & 1/8 & 1/8 \\ 1/8 & 1/4 & 1/8 \\ 1/8 & 1/8 & 1/4 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

So, f has a local (and global) minimum at $(0, 0, 0)^T$

Modified and discretized Newton's methods

- **Modified Newton's method**
  In the general gradient method

  $$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k D^k \nabla f(\boldsymbol{x}^k)$$

  take

  $$D^k = \left(\nabla^2 f(\boldsymbol{x}^0)\right)^{-1}, \quad k = 0, 1, \ldots$$

  provided $\nabla^2 f(\boldsymbol{x}^0))$ is positive definite
  This method is the same as Newton's method except that the Hessian matrix is not computed at each step. A related method recomputes the Hessian matrix every $p > 1$ steps (p not necessarily fixed)
- **Discretized Newton's method**
  In the general gradient method

  $$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k D^k \nabla f(\boldsymbol{x}^k)$$

  take

  $$D^k = \left(H(\boldsymbol{x}^k)\right)^{-1}, \quad k = 0, 1, \ldots$$

  where $H(\boldsymbol{x}^k)$ is a positive definite symmetric approximation of $\nabla^2 f(\boldsymbol{x}^k)$ computed using finite difference approximations of the second derivatives of f (eventually using the values of f')

Diagonally scaled steepest descent: diagonal approximation to Newton's method

- In the general gradient method

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k D^k \nabla f(\boldsymbol{x}^k),$$

the diagonally scaled steepest descent method uses

$$D^k = \begin{pmatrix} d_1^k & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & d_2^k & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & d_{n-1}^k & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & d_n^k \end{pmatrix}, \quad k = 0, 1, \dots$$

where $d_i^k \in \mathbb{R}$ are all positive, thus ensuring that D^k is positive definite

- A popular choice, resulting in a method known as a **diagonal approximation to Newton's method** is to take d_i^k to be an approximation of the inverted second partial derivative of f with respect to x_i, this is

$$d_i^k \approx \left(\frac{\partial^2 f(\boldsymbol{x}^k)}{\partial x_i^2} \right)^{-1}$$

Selecting the stepsize

Some of the most usual rules for choosing the stepsize α^k in a gradient method are:

- **Constant stepsize**

  A fixed stepsize $s > 0$ is selected and

  $$\alpha^k = s, \quad k = 0, 1, \dots$$

  In this simple rule, if the stepsize is too large, probably divergence will occur, while if the stepsize is too small, the rate of convergence may be very slow

- **Minimization rule**

  Take α^k such that the cost function is minimized along the direction $\boldsymbol{d}^k$, that is α^k satisfies

  $$f(\boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k) = \min_{\alpha \geq 0} f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k)$$

- **Limited minimization rule**

  Fix a certain $s > 0$ and choose α^k such that

  $$f(\boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k) = \min_{0 \leq \alpha \leq s} f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k)$$

Remark: The last two rules must be implemented together with an one-dimensional minimization procedure

Selecting the stepsize

- **Successive stepsize reduction**

  In the simplest rule of this type an initial stepsize s is chosen. If

  $$f(\boldsymbol{x}^k + s\boldsymbol{d}^k) < f(\boldsymbol{x}^k)$$

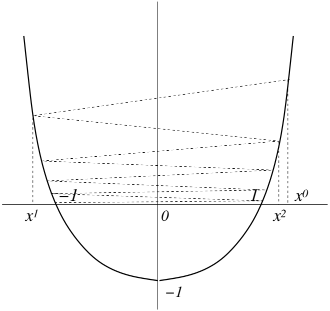
  we take $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + s\boldsymbol{d}^k$ and continue the iterative procedure. If the above condition is not fulfilled the stepsize is reduced, perhaps repeatedly, by a certain factor, until the value of f is improved

  Remark: It may happen that the cost improvement obtained at each iteration may not be substantial enough to guarantee convergence as is shown in the following example

Successive stepsize reduction

Example. Consider the function

$$f(x) = \begin{cases} \dfrac{3(1-x)^2}{4} - 2(1-x), & \text{if} \quad x > 1, \\ \dfrac{3(1+x)^2}{4} - 2(1+x), & \text{if} \quad x < -1, \\ x^2 - 1, & \text{if} \quad -1 \leq x \leq 1. \end{cases}$$



Clearly f is convex, continuously differentiable, is minimized at $x^* = 0$, and

$$f(x) < f(y) \quad \text{if and only if} \quad |x| < |y|.$$

Example (cont.)

The gradient of f is given by

$$\nabla f(x) = \begin{cases} \dfrac{3x}{2} + \dfrac{1}{2}, & \text{if} \quad x > 1, \\ \dfrac{3x}{2} - \dfrac{1}{2}, & \text{if} \quad x < -1, \\ 2x, & \text{if} \quad -1 \le x \le 1. \end{cases}$$

If we take $x > 1$, then

$$x - \nabla f(x) = x - \frac{3x}{2} - \frac{1}{2} = -\left(\frac{x}{2} + \frac{1}{2}\right),$$

from which it can be verified that since $x > 1$, then

$$|x - \nabla f(x)| < |x| \quad \Rightarrow \quad f(x - \nabla f(x)) < f(x)$$

and also

$$x - \nabla f(x) < -1$$

Similarly, if $x < -1$, then

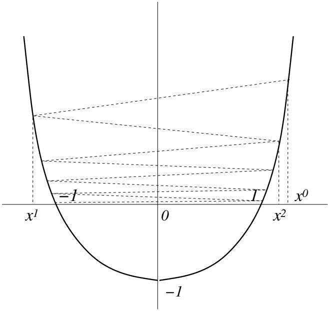
$$f(x - \nabla f(x)) < f(x), \quad \text{and} \quad x - \nabla f(x) > 1$$

Example (cont.)

Consider the steepest descent iteration

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

where the stepsize is successively reduced from an initial stepsize $s = 1$ until descent is obtained



As in the figure, take $x^0 > 1$ (or $|x^0| > 1$), then $|x^1| > 1$, $|x^2| > 1$,..., $|x^k| > 1$ so it cannot converge to the unique minimum $x^* = 0$

Limit points of gradient methods

We want to analize when each limit point $\boldsymbol{x}^*$ of a sequence $\{\boldsymbol{x}^k\}$ generated by a gradient method is a stationary point: $\nabla f(x^*) = 0$

- From Taylor's formula

$$f(\boldsymbol{x}^{k+1}) = f(\boldsymbol{x}^k) + \alpha^k (\nabla f(\boldsymbol{x}^k))^T \boldsymbol{d}^k + o(\alpha^k)$$

  we see that: if the slope of f at $\boldsymbol{x}^k$ along the direction $\boldsymbol{d}^k$ ($\approx$ directional derivative of f at $\boldsymbol{x}^k$ along $\boldsymbol{d}^k$), which is $(\nabla f(\boldsymbol{x}^k))^T \boldsymbol{d}^k$, is large, then the rate of progress of the method will be, in principle, also large

- On the other hand, if the directions $\boldsymbol{d}^k$ tend to become asymptotically orthogonal to the gradient direction

$$\frac{(\nabla f(\boldsymbol{x}^k))^T \boldsymbol{d}^k}{\|\nabla f(\boldsymbol{x}^k)\| \|\boldsymbol{d}^k\|} \to 0$$

  as $\boldsymbol{x}^k$ approaches a nonstationary point, there is a chance that the method will get "stuck" near that point

- To ensure that this does not happen, we consider some non-orthogonality condition on the directions $\boldsymbol{d}^k$, the so called gradient related condition

The gradient related condition

Assume that the direction $\boldsymbol{d}^k$ is obtained as a given function of $\boldsymbol{x}^k$

Definition

We say that the direction sequence $\{\boldsymbol{d}^k\}$ is gradient related to $\{\boldsymbol{x}^k\}$ if the following property holds: For any subsequence $\{\boldsymbol{x}^k\}_{k\in\mathcal{K}}$ of $\{\boldsymbol{x}^k\}$ convergent towards a non-stationary point, the corresponding subsequence $\{\boldsymbol{d}^k\}_{k\in\mathcal{K}}$ is bounded and satisfies

$$\lim_{k\to\infty}\sup_{k\in\mathcal{K}}\nabla f(\boldsymbol{x}^k)^T d^k < 0 \tag{2}$$

- If $\{\boldsymbol{d}^k\}$ is gradient related, it follows that if a subsequence $\{\nabla f(\boldsymbol{x}^k)\}_{k\in\mathcal{K}}$ tends to a nonzero vector, the corresponding sequence of directions $\boldsymbol{d}^k$ is bounded and does not tend to be orthogonal to $\nabla f(\boldsymbol{x}^k)$

- Roughly, this means that $\boldsymbol{d}^k$ does not become "too small" or "too large" relative to $\nabla f(\boldsymbol{x}^k)$, and that the angle between $\nabla f(\boldsymbol{x}^k)$ and $\boldsymbol{d}^k$ does not get "too close" to 90 degrees

Successive stepsize reduction. Armijo rule

The Armijo rule.

- The Armijo rule is essentially the succesive reduction rule suitably modified to eliminate the convergence difficulty shown in the example of page 58
- Fix scalars s, β i σ such that $0 < \beta < 1$ i $0 < \sigma < 1$
- In $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$ take

$$\alpha^k = \beta^{m_k} s$$

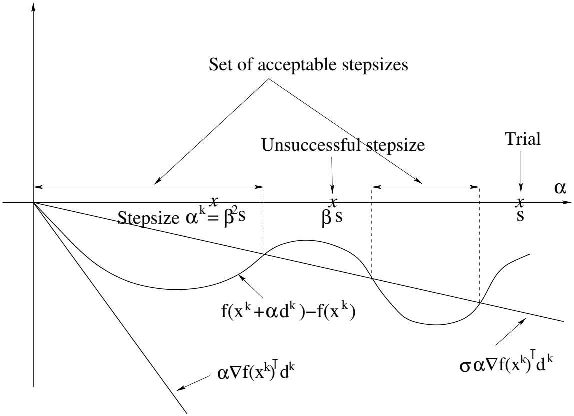
where m_k is the first non-negative integer m for which

$$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^k + \beta^{m_k} s \boldsymbol{d}^k) \geq -\sigma \beta^{m_k} s \nabla f(x^k)^T \boldsymbol{d}^k$$

$$\left(f(\boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k) - f(\boldsymbol{x}^k) \leq \sigma \alpha^k \nabla f(x^k)^T \boldsymbol{d}^k \right)$$

- The above rule means that the stepsizes $\beta^m s$, $m = 0, 1, \ldots$ are tried until the above inequality is satisfied (that guarantees that the cost improvement is large enough) and then we set $m_k = m$
- Usually σ is chosen close to zero, for instance $\sigma \in [10^{-5}, 10^{-1}]$. The reduction factor β is usually chosen between $1/2$ and $1/10$, depending on the confidence we have on the quality on the initial stepsize s

The Armijo rule


Set of acceptable stepsizes
Unsuccessful stepsize
Trial
α
Stepsize $\alpha^k = \beta^2 s$
βs
s
$f(x^k+\alpha d^k)-f(x^k)$
$\alpha\nabla f(x^k)^T d^k$
$\sigma\alpha\nabla f(x^k)^T d^k$

Line search by the Armijo rule: We start with the trial stepsize s and continue with βs, $\beta^2 s$,... until the first time that $\beta^m s$ falls within the sets of stepsizes α satisfying the inequality

$$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) \geq -\sigma\alpha\nabla f(x^k)^T \boldsymbol{d}^k$$

Thus, the cost improvement $f(\boldsymbol{x}^k) - f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k)$ must not be just positive, it must be sufficiently large as to fulfil the above condition

Convergence

The following theorem is the main convergence result of the gradient methods

Theorem

Let $\{\boldsymbol{x}^k\}$ be a sequence generated by a gradient method

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$$

and assume that $\{\boldsymbol{d}^k\}$ is gradient related to $\{\boldsymbol{x}^k\}$, and that α^k is chosen by the Armijo rule.

Then, every limit point of $\{\boldsymbol{x}^k\}$ is a stationary point ($\nabla f(\boldsymbol{x}^*) = 0$)

Proof of the convergence Theorem

Proof

Consider the Armijo rule and, to arrive to a contradiction, assume that $\boldsymbol{x}^*$ is a limit point of $\{\boldsymbol{x}^k\}$ such that $\nabla f(\boldsymbol{x}^*) \neq 0$

- Since $\{f(\boldsymbol{x}^k)\}$ is monotonically non-increasing, then $\{f(\boldsymbol{x}^k)\}$ either converges to a finite value or diverges to $-\infty$
- Since f is continuous, then

  $$\lim_{k\to\infty} f(\boldsymbol{x}^k) = f(\boldsymbol{x}^*)$$

  so, it follows that

  $$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1}) \to 0$$

- By the definition of the Armijo rule, we have

  $$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1}) \geq -\sigma\alpha^k \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k \tag{3}$$

  hence $\alpha^k \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k \to 0$

- Let $\{\boldsymbol{x}^k\}_{k\in\mathcal{K}}$ be a subsequence converging to $\boldsymbol{x}^*$. Since $\{\boldsymbol{d}^k\}$ is gradient related and $\nabla f(\bar{\boldsymbol{x}}) \neq 0$, we have that

  $$\lim_{k\to\infty} \sup_{k\in\mathcal{K}} \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k < 0 \quad \Rightarrow \quad \{\alpha^k\}_{\mathcal{K}} \to 0$$

Proof of the convergence Theorem (cont.)

By the definition of the Armijo rule, we must have for some index $\overline{k} \geq 0$ that

$$f(\boldsymbol{x}^k) - f\left(\boldsymbol{x}^k + \frac{\alpha^k}{\beta} d^k\right) < -\sigma \frac{\alpha^k}{\beta} \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k, \quad \forall k \in \mathcal{K}, k \geq \overline{k} \qquad (4)$$

that is, the initial stepsize s will be reduced at least once for all $k \in \mathcal{K}$, $k \geq \overline{k}$. Denote

$$\boldsymbol{p}^k = \frac{\boldsymbol{d}^k}{\|\boldsymbol{d}^k\|}, \quad \overline{\alpha}^k = \frac{\alpha^k \|\boldsymbol{d}^k\|}{\beta}$$

since $\{\boldsymbol{d}^k\}$ is gradient related, the sequence $\{\|\boldsymbol{d}^k\|\}_{\mathcal{K}}$ is bounded, and it follows that

$$\{\overline{\alpha}^k\}_{\mathcal{K}} \to 0$$

Since $\|\boldsymbol{p}^k\| = 1$ for all $k \in \mathcal{K}$, there exist a subsequence $\{\boldsymbol{p}^k\}_{\overline{\mathcal{K}}}$ of $\{\boldsymbol{p}^k\}_{\mathcal{K}}$ such that

$$\{\boldsymbol{p}^k\}_{\overline{\mathcal{K}}} \to \overline{\boldsymbol{p}}$$

where $\overline{\boldsymbol{p}}$ is some vector with $\|\overline{\boldsymbol{p}}\| = 1$. From equation(4), we have

$$\frac{f(\boldsymbol{x}^k) - f(\boldsymbol{x}^k + \overline{\alpha}^k \boldsymbol{p}^k)}{\overline{\alpha}^k} < -\sigma \nabla f(\boldsymbol{x}^k)^T \boldsymbol{p}^k, \quad \forall k \in \mathcal{K}, k \geq \overline{k} \qquad (5)$$

Proof of the convergence Theorem (cont.)

Using the mean value Theorem, the above relation is written as

$$-\nabla f(\boldsymbol{x}^k + \tilde{\alpha}^k \boldsymbol{p}^k)^T \boldsymbol{p}^k < -\sigma \nabla f(\boldsymbol{x}^k)^T \boldsymbol{p}^k, \quad \forall k \in \mathcal{K}, k \geq \overline{k}$$

where $\tilde{\alpha}^k \in [0, \overline{\alpha}^k]$. Taking limits in the above equation one gets

$$-\nabla f(\overline{\boldsymbol{x}})^T \overline{\boldsymbol{p}} \leq -\sigma \nabla f(\overline{\boldsymbol{x}})^T \overline{\boldsymbol{p}}$$

this is

$$0 \leq (1-\sigma) \nabla f(\overline{\boldsymbol{x}})^T \overline{\boldsymbol{p}}$$

Since $\sigma < 1$, it follows that

$$0 \leq \nabla f(\overline{\boldsymbol{x}})^T \overline{\boldsymbol{p}} \tag{6}$$

On the other hand we have

$$\nabla f(\boldsymbol{x}^k)^T \boldsymbol{p}^k = \frac{\nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k}{\|\boldsymbol{d}^k\|}$$

By taking the limit as $k \in \mathcal{K}$, $k \to \infty$

$$\nabla f(\overline{\boldsymbol{x}})^T \overline{\boldsymbol{p}} \leq \frac{\limsup_{k\to\infty, k\in\mathcal{K}} \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k}{\limsup_{k\to\infty, k\in\mathcal{K}} \|\boldsymbol{d}^k\|} < 0$$

which contradicts (6). This proves the result

Second convergence Theorem

Theorem

Let $\{\boldsymbol{x}^k\}$ be a sequence generated by a gradient method

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$$

and assume that $\{\boldsymbol{d}^k\}$ is gradient related to $\{\boldsymbol{x}^k\}$, and that α^k is chosen by the minimization rule, or the limited minimization rule.
Then, every limit point of $\{\boldsymbol{x}^k\}$ is a stationary point $(\nabla f(\boldsymbol{x}^*) = 0)$

Proof

Consider the minimization rule, and let $\{\boldsymbol{x}^k\}_{\mathcal{K}}$ converge to $\overline{\boldsymbol{x}}$ with $\nabla f(\overline{\boldsymbol{x}}) \neq 0$. Again we have that $\{f(\boldsymbol{x}^k)\}$ decreases monotonically to $f(\overline{\boldsymbol{x}})$. Let $\tilde{\boldsymbol{x}}^{k+1}$ be the point generated from $\boldsymbol{x}^k$ using the Armijo rule, and let $\tilde{\alpha}^k$ be the corresponding stepsize. We have

$$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1}) \geq f(\boldsymbol{x}^k) - f(\tilde{\boldsymbol{x}}^{k+1}) \geq -\sigma\tilde{\alpha}^k \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k$$

By repeating the argument of the previous proof following equation (2), replacing α^k by $\tilde{\alpha}^k$, we can obtain a contradiction. In particular we have

$$\{\tilde{\alpha}^k\}_{\mathcal{K}} \to 0$$

and, by the definition of the Armijo rule, we have for some index $\overline{k} \geq 0$

$$f(\boldsymbol{x}^k) - f\left(\boldsymbol{x}^k + \frac{\alpha^k}{\beta}\boldsymbol{d}^k\right) < -\sigma \frac{\alpha^k}{\beta} \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k, \quad \forall k \in \mathcal{K}, k \geq \overline{k}$$

Proof of the second convergence Theorem (cont.)

Proceeding as earlier, we obtain (4) and (5) with $\overline{\alpha}^k = \tilde{\alpha}^k \|\boldsymbol{d}^k\|/\beta$, and a contradiction

The argument just used establishes that any stepsize rule that gives a larger reduction in cost at each iteration than the Armijo rule inherits its convergence properties. This also proves the proposition for the limited minimization rule

The Gauss-Newton method

- This method is applicable to the problem of minimizing the sum of squares of real valued functions $g_1,\ldots,g_m$.
- Denoting $\boldsymbol{g} = (g_1, \ldots, g_m)^T$ the problem can be written as

  $$\text{minimize} \quad F(\boldsymbol{x})$$

  where

  $$F(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{g}(\boldsymbol{x})\|^2 = \frac{1}{2}\boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{g}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{m} g_i^2(\boldsymbol{x})$$

  with $\boldsymbol{x} \in \mathbb{R}^n$
- This problem can be solved using Newton's method
- To solve this problem, by means of Gauss-Newton method, we use the linealization of $\boldsymbol{g}(\boldsymbol{x})$ around $\boldsymbol{x}^k$:

  $$\boldsymbol{g}(\boldsymbol{x}) \approx \boldsymbol{g}(\boldsymbol{x}^k) + \nabla\boldsymbol{g}(\boldsymbol{x}^k)^T(\boldsymbol{x} - \boldsymbol{x}^k)$$

The Gauss-Newton method

- We need to compute the minimum of $\frac{1}{2}\|\boldsymbol{g}(\boldsymbol{x})\|^2$ using the above approximation, this is, the minimum of

$$\frac{1}{2}\left(\boldsymbol{g}(\boldsymbol{x}^k)+\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k)\right)^T\left(\boldsymbol{g}(\boldsymbol{x}^k)+\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k)\right)=$$

$$\frac{1}{2}\left(\|\boldsymbol{g}(\boldsymbol{x}^k)\|^2+2(\boldsymbol{x}-\boldsymbol{x}^k)^T\nabla\boldsymbol{g}(\boldsymbol{x}^k)\boldsymbol{g}(\boldsymbol{x}^k)+(\boldsymbol{x}-\boldsymbol{x}^k)^T\nabla\boldsymbol{g}(\boldsymbol{x}^k)\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k)\right)$$

- Equating to zero the derivative of this expression, we get

$$\nabla\boldsymbol{g}(\boldsymbol{x}^k)\ \boldsymbol{g}(\boldsymbol{x}^k)+\nabla\boldsymbol{g}(\boldsymbol{x}^k)\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k)=0$$

- If the matrix $\nabla\boldsymbol{g}(\boldsymbol{x}^k)\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T$ is non-singular, then

$$\nabla\boldsymbol{g}(\boldsymbol{x}^k)\ \boldsymbol{g}(\boldsymbol{x}^k)+\nabla\boldsymbol{g}(\boldsymbol{x}^k)\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T(\boldsymbol{x}-\boldsymbol{x}^k)=0 \quad\Rightarrow$$

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\left(\nabla\boldsymbol{g}(\boldsymbol{x}^k)\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T\right)^{-1}\nabla\boldsymbol{g}(\boldsymbol{x}^k)\ \boldsymbol{g}(\boldsymbol{x}^k)$$

Note that since $F(\boldsymbol{x})=(1/2)\boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{g}(\boldsymbol{x})$, then

$$\nabla F(\boldsymbol{x}^k)=\nabla\boldsymbol{g}(\boldsymbol{x}^k)\,\boldsymbol{g}(\boldsymbol{x}^k)\quad\Rightarrow\quad \boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\left(\nabla\boldsymbol{g}(\boldsymbol{x}^k)\nabla\boldsymbol{g}(\boldsymbol{x}^k)^T\right)^{-1}\nabla F(\boldsymbol{x}^k)$$

The Gauss-Newton method (cont.)

- According to the general pattern of gradient methods, we can write Gauss-Newton method as

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k - \alpha^k \left(\nabla \boldsymbol{g}(\boldsymbol{x}^k)\nabla \boldsymbol{g}(\boldsymbol{x}^k)^T\right)^{-1} \nabla \boldsymbol{g}(\boldsymbol{x}^k)\ \boldsymbol{g}(\boldsymbol{x}^k) \\ &= \boldsymbol{x}^k - \alpha^k \left(\nabla \boldsymbol{g}(\boldsymbol{x}^k)\nabla \boldsymbol{g}(\boldsymbol{x}^k)^T\right)^{-1} \nabla F(\boldsymbol{x}^k) \end{aligned}$$

  so

$$D^k = \left(\nabla \boldsymbol{g}(\boldsymbol{x}^k)\nabla \boldsymbol{g}(\boldsymbol{x}^k)^T\right)^{-1}, \quad k = 0, 1, \ldots$$

- We have assumed that $\nabla \boldsymbol{g}(\boldsymbol{x}^k)\nabla \boldsymbol{g}(\boldsymbol{x}^k)^T$ is non-singular. In fact, it will be always positive semidefinite.
- The matrix $\nabla \boldsymbol{g}(\boldsymbol{x}^k)\nabla \boldsymbol{g}(\boldsymbol{x}^k)^T$ is positive definite, and so non-singular, if the matrix $\nabla \boldsymbol{g}(\boldsymbol{x}^k)$ has rang n

The Gauss-Newton method (cont.)

- **Advantage** of Gauss-Newton method over Newton's method: **no second derivatives of g are needed**
- **Disadvantage** of Gauss-Newton method over Newton's method: **convergence is slower**

Applications of the Gauss-Newton method. Least squares problems

Example 1. Model construction (and curve fitting)

We want to estimate n parameters $\boldsymbol{p} \in \mathbb{R}^n$ of a mathematical model $h(\boldsymbol{x}, \boldsymbol{p})$, so that it fits well a physical system $f(\boldsymbol{x})$ based on a set of mesurements.

Assume that:

- $z = f(\boldsymbol{x}) \in \mathbb{R}$ is the physical system's output
- $h(\boldsymbol{x}, \boldsymbol{p}) \in \mathbb{R}$ is a known real value function representing the model
- $\boldsymbol{x} \in \mathbb{R}^p$ is the physical system's input
- $\boldsymbol{p} \in \mathbb{R}^n$ is a vector of unknown parameters

Given a set of m input-output data pairs $(\boldsymbol{x}_1, z_1)$,...,$(\boldsymbol{x}_m, z_m)$ from measurements of the physical system that we tray to model, we want to find the vector of parameters $\boldsymbol{p}$ that matches best the data, in the sense that it mininizes the sum of squared errors

$$\frac{1}{2}\sum_{i=1}^{m} \|z_i - h(\boldsymbol{x}_i, \boldsymbol{p})\|^2$$

so, according to the notation introduced for the Gauss-Newton method:

$$g_i(\boldsymbol{p}) = z_i - h(\boldsymbol{x}_i, \boldsymbol{p}), \quad \boldsymbol{g}(\boldsymbol{p}) = (g_1(\boldsymbol{p}), ..., g_m(\boldsymbol{p})), \quad F(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{g}(\boldsymbol{p})\|^2 = \frac{1}{2}\sum_{i=1}^{m} g_i^2(\boldsymbol{p})$$

Applications of the Gauss-Newton method. Least squares problems

Example 2. Model construction and dynamical system identification

A common model for a single input-output dynamical system is to relate the input sequence $\{x_k\}_{k=1,\ldots,m}$ to the output sequence $\{z_k\}_{k=1,\ldots,m}$, with $x_k,\ z_k \in \mathbb{R}$, by a linear equation of the form

$$\sum_{j=0}^{n} \alpha_j z_{m-j} = \sum_{j=0}^{n} \beta_j x_{m-j}$$

Given a set of m inputs and outputs (x_1, z_1),...,(x_m, z_m) from the true system, we would like to find the set of parameters α_j, β_j, $j = 0, \ldots, n$ that matches best the set of data, in the sense that it minimizes

$$\sum_{k=1}^{m} \left(\sum_{j=0}^{n} \alpha_j z_{k-j} - \sum_{j=0}^{n} \beta_j x_{k-j} \right)^2$$

so, according to the above notation, if $\boldsymbol{y} = (\alpha_0, \alpha_1, \ldots, \alpha_n, \beta_0, \beta_1, \ldots, \beta_n)$

$$g_k(\boldsymbol{y}) = \sum_{j=0}^{n} \alpha_j z_{k-j} - \sum_{j=0}^{n} \beta_j x_{k-j}, \quad F(\boldsymbol{y}) = \frac{1}{2} \sum_{k=1}^{m} g_k^2(\boldsymbol{y})$$

Applications of the Gauss-Newton method. Least squares problems

Example 3. Neural networks

- Their purpose is to model a physical system

$$\boldsymbol{y} \longrightarrow \boldsymbol{z}$$

  by a multistage system with a certain number N of stages (layers), Given a certain input $\boldsymbol{y}$, the output of the physical system is denoted by $\boldsymbol{z}$

- Let $\boldsymbol{y}_0 = (y_0^1, ..., y_0^{n_0})$ be the input of the first stage, and $\boldsymbol{y}_k = (y_k^1, ..., y_k^{n_k})$ the output vector of the system (that has n_k activation units) at the k-th stage

$$\boldsymbol{y}_0 = \begin{pmatrix} y_0^1 \\ \vdots \\ y_0^{n_0} \end{pmatrix} \to \boldsymbol{y}_1 = \begin{pmatrix} y_1^1 \\ \vdots \\ y_1^{n_1} \end{pmatrix} \to \cdots \boldsymbol{y}_k = \begin{pmatrix} y_k^1 \\ \vdots \\ y_k^{n_k} \end{pmatrix} \to \cdots \begin{pmatrix} y_N^1 \\ \vdots \\ y_N^{n_N} \end{pmatrix} = \boldsymbol{z}$$

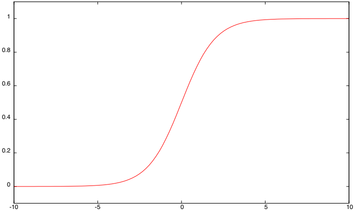
- Note that, in principle, $n_0 \neq n_1 \neq \cdots \neq n_N$

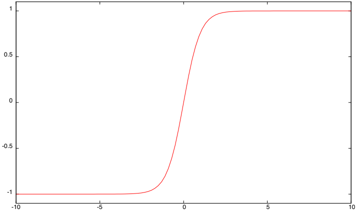
Applications of the Gauss-Newton method. Neural networks

- The k-th stage of the multistage system model consists of n_k "activation units", each of which is given by single input-single output mapping ϕ

$$\mathbf{y}_{k-1} = \begin{pmatrix} y_{k-1}^{1} \\ \vdots \\ y_{k-1}^{n_{k-1}} \end{pmatrix} \rightarrow \cdots \mathbf{y}_k = \begin{pmatrix} y_{k}^{1} \\ \vdots \\ y_{k}^{n_{k}} \end{pmatrix}$$

- Common examples of "activation units" are functions such as:





$$\phi(x) = \frac{1}{1+e^{-x}} \qquad\qquad \phi(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}},$$

whose derivatives are zero when $x \rightarrow \pm\infty$

Applications of the Gauss-Newton method. Neural networks

- In the k stage, the input of any activation unit ϕ is a linear function of the output vector $\boldsymbol{y}_{k-1}$,

$$\boldsymbol{y}_{k-1} = \begin{pmatrix} y^1_{k-1} \\ \vdots \\ y^{n_{k-1}}_{k-1} \end{pmatrix} \to \boldsymbol{y}_k = \begin{pmatrix} y^1_k = \phi\left(u^{0_1}_{k-1} + \sum_{s=1}^{n_{k-1}} u^{s_1}_{k-1} y^s_{k-1}\right) \\ \vdots \\ y^{n_k}_k = \phi\left(u^{0_{n_k}}_{k-1} + \sum_{s=1}^{n_{k-1}} u^{s_{n_k}}_{k-1} y^s_{k-1}\right) \end{pmatrix}$$

and the output of the j-th activation unit is

$$y^j_k = \phi\left(u^{0_j}_{k-1} + \sum_{s=1}^{n_{k-1}} u^{s_j}_{k-1} y^s_{k-1}\right), \quad j = 1, \dots, n_k,$$

where all the coefficients $u^{s_j}_{k-1}$ (weights) are to be determined

Applications of the Gauss-Newton method. Neural networks

- Let $\boldsymbol{u}$ denote the vector of the weights of all the stages

$$\boldsymbol{u} = \{u_k^{s_j} \mid k = 0, ..., N-1, \; s = 0, ..., n_k, \; j = 1, ..., n_{k+1}\}$$

- Using a given activation unit ϕ, for a given vector of weights $\boldsymbol{u}$, and an input vector $\boldsymbol{y}_0$ to the first stage the model produces an unique output vector $\boldsymbol{y}_N$ of the N stage

- Thus, we may view the multistage system defining the neural network as a mapping h, parametrized by $\boldsymbol{u}$, such that

$$h : \; \boldsymbol{y}_0 \longrightarrow h(\boldsymbol{u}, \boldsymbol{y}_0) = \boldsymbol{y}_N$$

Applications of the Gauss-Newton method. Neural networks

- Selecting $\boldsymbol{u}$ appropriartely, we can try to match the mapping of the multistage system with the mapping of the physical system

- A way to do find the optimal weights, that is known as *training the network* can be done as follows:
  1. Use a sample of m input-output pairs $(\boldsymbol{y}_1, \boldsymbol{z}_1)$,..., $(\boldsymbol{y}_m, \boldsymbol{z}_m)$ from the physical system
  2. Minimize, over $\boldsymbol{u}$, the sum of squared errors

$$\frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{z}_i - h(\boldsymbol{u}, \boldsymbol{y}_i)\|^2$$

- For the functions ϕ already given, it is possible to show that with a sufficient number of activation units and a number of stages $N \geq 2$, a multistage system can approximate arbitrarily closely very complex input-output maps

Applications of the Gauss-Newton method. Pattern classification

Example 4. Pattern classification

Consider the problem of classifying objects (persons or situations) based on the values of their characteristics

- Each object is presented with a vector of $\boldsymbol{y}$ features, and we wish to clasify it in one of a certain set with s categories
- For example, the vector $\boldsymbol{y}$ may represent the results of a collection of tests on a medical patient, and we may wish to clasify the patient as being healthy or as having one of several types of illnesses
- A classical pattern classification approach is to assume that **for each category** $j = 1, ..., s$**, we know the probability** $p(j|\boldsymbol{y})$ that an object with feature vector $\boldsymbol{y}$ is of category j
- Then, we may associate an object with feature vector $\boldsymbol{y}$ with the category $j^*(\boldsymbol{y})$ having maximum probability, this is

$$j^*(\boldsymbol{y}) = \arg \max_{j=1,\ldots,s} p(j|\boldsymbol{y})$$

Applications of the Gauss-Newton method. Pattern classification

- Suppose that the probabilities $p(j|\boldsymbol{y})$ are unknown, but instead we have a sample consisting of m object-category pairs: $(j_1, \boldsymbol{y}_1), \ldots, (j_m, \boldsymbol{y}_m)$

  Then we may try to estimate $p(j|\boldsymbol{y})$ based on the following simple fact: *Of all functions $f_j(\boldsymbol{y})$ of $\boldsymbol{y}$, $p(j|\boldsymbol{y})$ is the one that minimizes the value of*

  $(z_j - f_j(\boldsymbol{y}))^2$, where

  $$z_j = \begin{cases} 1 & \textit{if } \boldsymbol{y} \textit{ is of category } j, \\ 0 & \textit{otherwise.} \end{cases}$$

- To compute the estimates of $p(j|\boldsymbol{y})$, for each category $j \in \{1, \ldots, s\}$, we approximate the probability $p(j|\boldsymbol{y})$ by a function $h_j(\boldsymbol{x}_j, \boldsymbol{y})$ that is parametrized by a vector $\boldsymbol{x}_j$.

- The function h_j may be provided, for example, by a neural network (see Example 3).

Applications of the Gauss-Newton method. Pattern classification

- Then, we can obtain $\boldsymbol{x}_j$ by minimizing the least squares function

$$\frac{1}{2}\sum_{i=1}^{m}\left(z_j^i - h_j(\boldsymbol{x}_j, \boldsymbol{y}_i)\right)^2,$$

where

$$z_j^i = \begin{cases} 1 & \text{if } \boldsymbol{y}_i \text{ is of category } j, \\ 0 & \text{otherwise.} \end{cases}$$

- This minimization approximates the minimization of the expected value of $(z_j - f_j(\boldsymbol{y}))^2$.

- Once the optimal parameter vectors $\boldsymbol{x}_j^*$, $j = 1, ..., s$ have been obtained, we can use them to classify a new object with feature vector $\boldsymbol{y}$ according to the rule

$$\text{Estimated object category} = \arg\max_{j=1,\ldots,s} h_j(x_j^*, \boldsymbol{y}).$$