Introduction to unconstrained optimization - gradient-based methods

Jussi Hakanen
Post-doctoral researcher
jussi.hakanen@jyu.fi

spring 2014
TIES483 Nonlinear optimization
Model algorithm for unconstrained minimization

1) [Test for convergence.] If conditions are satisfied, stop. The solution is $x^h$.

2) [Compute a search direction.] Compute a non-zero vector $d^h \in R^n$ which is the search direction.

3) [Compute a step length.] Compute $\alpha_h > 0$, the step length, for which it holds that $f(x^h + \alpha_h d^h) < f(x^h)$.

4) [Update the estimate for minimum.] Set $x^{h+1} = x^h + \alpha_h d^h$, $h = h + 1$ and go to step 1.

Nelder Mead Simplex


*Don’t confuse with more famous Simplex algorithm for linear optimization*

A direct search method

Operates with $n + 1$ points $x^1, ..., x^{n+1}$ in $R^n$

Also known as the Polytope algorithm because the points can be seen as vertices of a polytope in $R^n$
Nelder Mead Simplex (cont.)

- Initialize: Select $n + 1$ points in $R^n$ and evaluate objective function in the points. Order the points s.t. $f^1 \leq f^2 \leq \cdots \leq f^n \leq f^{n+1}$ (note: $f^j = f(x^j)$).

- New trial point: $x^r = c + \alpha(c - x^{n+1})$, where $c = \frac{1}{n} \sum_{j=1}^{n} x^j$ is the centroid of $n$ best vertices and $\alpha > 0$ is the reflection coefficient. Evaluate $f^r$.
  - $f^1 \leq f^r \leq f^n$: $x^r$ is neither a new best point nor worst point $\rightarrow x^r$ replaces $x^{n+1}$ and move to next iteration.
  - $f^r < f^1$: $x^r$ is the new best point, try to improve $f$ further in that direction $\rightarrow x^e = c + \beta(x^r - c)$, where $\beta > 1$ is the expansion coefficient. If $f^e < f^r$, replace $x^{n+1}$ by $x^e$. Otherwise, replace $x^{n+1}$ by $x^r$.
  - $f^r > f^n$: The polytope is considered to be too large and should be contracted.

- Contraction: $x^c = c + \gamma(x^{n+1} - c)$ if $f^r \geq f^{n+1}$ OR $x^c = c + \gamma(x^r - c)$ if $f^r < f^{n+1}$, where $0 < \gamma < 1$ is the contraction coefficient. If $f^c < \min\{f^r, f^{n+1}\}$ replace $x^{n+1}$ by $x^c$. Otherwise further contraction is carried out.
Rosenbrock function

Have you heard of the Rosenbrock function?

A non-convex function $f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$

Has a global minimum in $x^* = (1,1)^T$, $f(x^*) = 0$ which is located in a narrow, banana-shaped valley

The coefficient of the second term can be adjusted but it does not affect the position of the global minimum

Used to test optimization algorithms
Figure 4i. Solution path of a polytope algorithm on Rosenbrock’s function. The curved lines correspond to lines of equal function value; the linear segments correspond to the movement of the "best" vertex $x_1$ as the solution process progresses. The algorithm was started at the point $(-1.2, 1.0)^T$. Rosenbrock’s function has a unique minimum at the point $(1, 1)^T$, which lies at the base of the banana-shaped valley.
Gradient-based methods

- If gradients are available, they should always be used → improve convergence
- If analytical formulas of the gradients are not available, they can be **numerically approximated** by using finite differences ($h > 0$)
  - Forward difference: $\frac{f(x+h)-f(x)}{h} = f'(x) + O(h)$
  - Central difference: $\frac{f(x+h)-f(x-h)}{2h} = f'(x) + O(h^2)$
  - Central difference is more accurate but requires twice as many function evaluations (in $x + h$ and $x - h$)
- Gradient-based methods usually converge to a local minimum which is **nearest** to the starting point
Examples of gradient-based methods

- Steepest descent
- Newton’s method
- Quasi-Newton method
- Conjugate gradient method
Model algorithm for unconstrained minimization

Let \( x^h \) be the current estimate for \( x^* \)

1) [Test for convergence.] If conditions are satisfied, stop. The solution is \( x^h \).

2) [Compute a search direction.] Compute a non-zero vector \( d^h \in \mathbb{R}^n \) which is the search direction.

3) [Compute a step length.] Compute \( \alpha_h > 0 \), the step length, for which it holds that
\[
 f(x^h + \alpha_h d^h) < f(x^h). 
\]

4) [Update the estimate for minimum.] Set
\[
 x^{h+1} = x^h + \alpha_h d^h, \quad h = h + 1 \text{ and go to step 1.} 
\]


spring 2014 TIES483 Nonlinear optimization
Reminder: Descent direction

**Definition:** Let $f: \mathbb{R}^n \to \mathbb{R}$. A vector $d \in \mathbb{R}^n$ is a **descent direction** for $f$ in $x^* \in \mathbb{R}^n$ if $\exists \delta > 0$ s.t.

$$f(x^* + \lambda d) < f(x^*) \quad \forall \lambda \in (0, \delta].$$

**Result:** Let $f: \mathbb{R}^n \to \mathbb{R}$ be differentiable in $x^*$. If $\exists d \in \mathbb{R}^n$ s.t. $\nabla f(x^*)^T d < 0$ then $d$ is a descent direction for $f$ in $x^*$. 

---

spring 2014

TIES483 Nonlinear optimization
Important steps

Computing a step length $\alpha_h$
- Requirement of $f(x^h + \alpha_h d^h) < f(x^h)$ is **not sufficient**
- E.g. consider a univariate function $f(x) = x^2$ and a sequence $(-1)^h \left( \frac{1}{2} + 2^{-h} \right)$
- Thus, the step length must produce a “sufficient decrease”

Computing a search direction $d^h \in \mathbb{R}^n$
- Function $f(x)$ decreases the most locally in the direction $-\nabla f(x)$
- If $d^h$ is almost **orthogonal** to $\nabla f(x^h)$, then $f$ is almost constant along $d^h \rightarrow$ this must be avoided

When $f$ is bounded from below, a strictly decreasing sequence $\{x^h\}$ converges
- Can converge to a **non-optimal point**
- If the above situations are avoided, the level set $L(f(x^0)) = \{x|f(x) \leq f(x^0)\}$ is closed and bounded and $f$ is twice continuously differentiable, then $\lim_{h \to \infty} \|\nabla f(x^h)\| = 0$
Steepest descent

- Also known gradient descent or gradient method
- Utilizes the first order Taylor series
- We want a direction $d^h$ s.t. $\exists \delta > 0$ for which $f(x^h + \alpha d^h) < f(x^h)$ for all $\alpha \in (0, \delta) \rightarrow \min_{d^h} \frac{f(x^h + \alpha d^h) - f(x^h)}{\alpha} = \min_{d^h} \frac{\alpha \nabla f(x^h)^T d^h}{\alpha}$

We are interested only in the direction, not the length of the vector so we can assume that $\|d^h\| \leq 1$. Therefore by Schwarz inequality $|\nabla f(x^h)^T d^h| \leq \|\nabla f(x^h)\| \|d^h\| \leq \|\nabla f(x^h)\|$

Because $\nabla f(x^h)^T d^h < 0$ we have $-\nabla f(x^h)^T d^h \leq \|\nabla f(x^h)\|$ if and only if $\nabla f(x^h)^T d^h \geq -\|\nabla f(x^h)\| \rightarrow$ the minimum is obtained when $\nabla f(x^h)^T d^h = -\|\nabla f(x^h)\|$

Therefore the objective function decreases fastest in $x^h$ when $d^h = -\nabla f(x^h)/\|\nabla f(x^h)\|$
Steepest descent algorithm

1) Choose the final tolerance $\epsilon > 0$ and a starting point $x^1$. Set $h = 1$.

2) If $\|\nabla f(x^h)\| < \epsilon$ stop.

3) Otherwise, set $d^h = -\nabla f(x^h)$.

4) Let $\alpha_h$ be the solution of problem

$$\min_{\alpha \geq 0} f(x^h + \alpha d^h).$$

5) Set $x^{h+1} = x^h + \alpha_h d^h$, $h = h + 1$ and go to 2).
Steepest descent: example

Figure 4j. Solution path of a steepest-descent algorithm on Rosenbrock’s function. The linear segments correspond to the step taken within a given iteration. Note that the algorithm would have failed in the vicinity of the point $(-0.3, 0.1)^T$ but for the fact that the linear search found, by chance, the second minimum along the search direction. Several hundred iterations were performed close to the new point without any perceptible change in the objective function. The algorithm was terminated after 1000 iterations.

From Gill et al., Practical Optimization, 1981
Steepest descent

\[ f(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \]

From Miettinen: Nonlinear optimization, 2007 (in Finnish)
Newton’s method

- Utilizes the second order Taylor series
  \[ f(x) \approx f(x^h) + \nabla f(x^h)^T(x - x^h) + \frac{1}{2}(x - x^h)^T H(x^h)(x - x^h) \]

- Necessary condition: ”gradient should be zero”. Differentiate the right-hand-side and set it equal to zero
  - \( \nabla f(x^h) + H(x^h)(x - x^h) = 0 \)
  - Assume that \( H(x^h) \) is non-singular →
    \[ x^{h+1} = x^h - H(x^h)^{-1} \nabla f(x^h) \]
  - \( d^h = -H(x^h)^{-1} \nabla f(x^h) \)
Newton’s method: example

\[ f(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \]
Summary

- **Steepest descent**
  - Descent direction: yes
  - Global convergence: yes
  - Local convergence: zig-zag near optimum
  - Computational bottle neck: $\alpha_n$
  - Memory consumption: $O(n)$

- **Newton’s method**
  - Descent direction: only if $H(x^h)^{-1}$ positive definite
  - Global convergence: no
  - Local convergence: yes, good if $f$ quadratic
  - Computational bottle neck: $H(x^h)^{-1}$
  - Memory consumption: $O(n^2)$
Topic of the lectures next week

Mon, Jan 27\textsuperscript{th}: gradient-based methods continue, e.g. quasi-Newton and conjugate gradient

Wed, Jan 29\textsuperscript{th}: demonstration of the methods learnt so far by using Matlab

Study this before the lecture!