Optimization in higher dimensions

- Theoretical aspects
- Gradient descent methods
- Newton's method
- Other methods

Towards Newton's method

⋆ the anti-gradient direction $d = -\nabla f(x)$: the best asymptotic descent direction
⋆ that does not mean it is the best choice in all applications!
⋆ other descent directions exist: any direction such that $d \cdot \nabla f(x) < 0$ is a descent direction.

Examples:

- $d = -\frac{\partial f}{\partial x_i}(x) e_i$
- $d = -D\nabla f(x)$, where D is a diagonal matrix with positive entries
- $d = -A\nabla f(x)$ (or $-A^{-1}\nabla f(x)$) where A is a positive-definite matrix

Why these work?

$$f(x+td) = f(x) + t\nabla f(x) \cdot d + o(t) = f(x) - t\underbrace{(\nabla f(x))^T A \nabla f(x)}_{\geq 0} + o(t)$$

Recall Wolfe's condition

$\star$ $m_1, m_2 \in (0,1)$ are chosen constants
$\star$ d is a descent direction at x: $d \cdot \nabla f(x) < 0$, $q(t) = f(x+td)$
$\star$ recall that $q'(0) = \nabla f(x) \cdot d < 0$

a) $\frac{q(t)-q(0)}{t} \leq m_1 q'(0)$ and $q'(t) \geq m_2 q'(0)$ (then we have a good t)
b) $\frac{q(t)-q(0)}{t} > m_1 q'(0)$ (then t is too big)
c) $\frac{q(t)-q(0)}{t} \leq m_1 q'(0)$ and $q'(t) < m_2 q'(0)$ (then t is too small)

$\star$ Interpretation of $q'(t) \geq m_2 q'(0)$: the slope should be "less negative" at the next point
$\star$ If $x_{i+1} = x_i + t_i d_i$ with t_i verifying the above then:

$$\nabla f(x_{k+1}) \cdot d_k \geq m_2 \nabla f(x_k) \cdot d_k.$$

$\star$ define θ_k as the angle between d_k and $-\nabla f(x_k)$:

$$\cos \theta_k = \frac{-\nabla f(x_k) \cdot d_k}{|\nabla f(x_k)||d_k|}.$$

Zoutendijk condition

Theorem 19

Consider the iteration $x_{i+1} = x_i + t_i d_i$ where $d_i \cdot \nabla f(x_i) < 0$ and t_i verifies the Wolfe conditions. Suppose that f is of class C^1 on $\mathbb{R}^n$ and is bounded from below. Assume also that ∇f is L-Lipschitz, i.e.

$$|\nabla f(x) - \nabla f(y)| \leq L|x-y|, \text{ for all } x, y \in \mathbb{R}^n.$$

Then

$$\sum_{k\geq 0} \cos^2\theta_k |\nabla f(x_k)|^2 < \infty.$$

⋆ the proof is rather straightforward (in the Notes)
⋆ Immediate consequence: if $d_i = -\nabla f(x_i)$ then $\theta_i = 0$ and $|\nabla f(x_i)| \to 0$.
⋆ if the descent direction is chosen such that θ_k is bounded away from 90°, i.e. $\cos\theta_k \geq \delta > 0$ then $|\nabla f_k| \to 0$.

The basic Newton Method

$\star$ as in the 1D case, look at the second order Taylor expansion

$$f(x+h) = f(x) + \nabla f(x) \cdot h + \frac{1}{2} h^T D^2 f(x) h + o(|h|^2)$$

The basic Newton Method

⋆ as in the 1D case, look at the second order Taylor expansion

$$f(x+h) \approx f(x) + \nabla f(x) \cdot h + \frac{1}{2} h^T D^2 f(x) h$$

⋆ then minimize the quadratic function in order to find the new iterate

$$\min_h \left(f(x) + \nabla f(x) \cdot h + \frac{1}{2} h^T D^2 f(x) h \right)$$

$$D^2 f(x) h + \nabla f(x) = 0 \Longrightarrow h = -[D^2 f(x)]^{-1} \nabla f(x)$$

Algorithm 7 (Newton's method)

Given a starting point x_0 run the recurrence

$$x_{i+1} = x_i - [D^2 f(x_i)]^{-1} \nabla f(x_i).$$

Remarks

Inconvenients:

- the method is not necessarily well-defined: is $D^2 f(x_i)$ invertible at x_i?
- the Taylor expansion is local: are we sure that $[D^2 f(x_i)]^{-1}\nabla f(x_i)$ is small?
- is the value of the function decreasing: $f(x_{i+1}) < f(x_i)$?
- is $d = [D^2 f(x_i)]^{-1}\nabla f(x_i)$ a descent direction? Yes, if $D^2 f(x_i)$ is positive-definite!
- note that $[D^2 f(x_i)]^{-1}\nabla f(x_i)$ implies the resolution of a linear system (recall that for large matrices we NEVER compute inverses!) - this might be costly if the number of variables is large

Advantage: when the method converges, the convergence is quadratic!

Theorem 20 (Quadratic convergence: Newton method)

If x^* is a non-degenerate minimizer for the function $f : \mathbb{R}^n \to \mathbb{R}$, i.e. $D^2 f(x^*)$ is positive definite, and the starting point x_0 is close enough to the optimum x^* then Newton's algorithm converges quadratically to x^*.

Newton-Rhapson Method

⋆ another point of view: solve nonlinear systems

$$\begin{cases} g_1(x_1,...,x_n) & = & 0 \\ \vdots & \ddots & \vdots \\ g_n(x_1,...,x_n) & = & 0 \end{cases}$$

⋆ denote $g(x) = (g_1(x),...,g_n(x))$ and $Dg(x) = (\frac{\partial g_i}{\partial x_j})$ (the Jacobian matrix)
⋆ the Newton iteration

$$x_{n+1} = x_n - (Dg(x_n))^{-1}g(x)$$

converges to a zero x^* of g quadratically provided that x_0 is close to x^* and $Dg(x^*)$ is non-degenerate.

⋆ note that the Newton method corresponds to the Newton-Rhapson method applied for finding the zeros of $g = \nabla f$

Fixing Newton's method

1. Use a line-search procedure. If $D^2 f(x)$ is positive definite then the Newton direction $d = -(D^2 f(x))^{-1} \nabla f(x)$ is a descent direction.

Proposition 21 (Newton with line-search)

Let f be a C^2 function and α-convex function. Let x_0 be such that the level set $S = \{x : f(x) \leq f(x_0)\}$ is bounded. Then the Newton method with Wolfe line-search converges to the unique global minimizer of f.

Proof: A lower bound for $\cos \theta_k$ can be found in terms of the eigenvalues of $D^2 f(x)$. The sequence of iterates converges to a critical point. Convergence is not quadratic if the step t is smaller than 1!

2. Variable metric methods. Any positive definite matrix A defines a new metric. There are choices of A for which convergence towards the minimum may be faster.

Discussion

$$f(x+d) \approx f(x) + \nabla f(x) \cdot d = f(x) + d^T \nabla f(x)$$

Minimize the first order approx. in the unit ball $B = \{d : d^T d \leq 1\}$ or equivalently, minimize

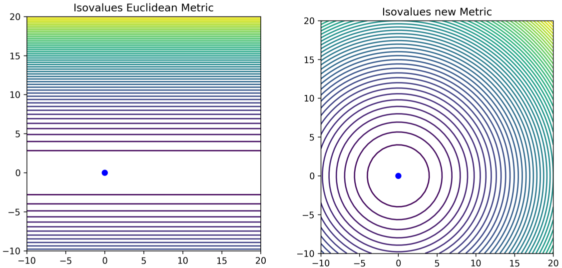
$$d \mapsto d^T \nabla f(x) + \frac{1}{2} d^T d$$

in order to get the optimal, anti-gradient direction

$$d^* = -\nabla f(x)$$

Remark: Note that the gradient method is the same as the Newton method when the Hessian $D^2 f(x)$ is the identity matrix.

Discussion: change the metric



let A be a symmetric positive-definite matrix

$$f(x+d) \approx f(x) + \nabla f(x) \cdot d = f(x) + d^T \nabla f(x)$$

Minimize the first order approx. in the unit ball $B = \{d : d^T A d \leq 1\}$ or equivalently, minimize

$$d \mapsto d^T \nabla f(x) + \frac{1}{2} d^T A d$$

in order to get the optimal direction

$$d = -A^{-1} \nabla f(x)$$

What metric to choose?

⋆ For $f(x) = \frac{1}{2}x^T Ax - b^T x$ change the variable to $\xi = A^{1/2}x$
⋆ Recall that $A^{1/2} = P^{-1}\sqrt{D}P$ where $A = P^{-1}DP$ is a diagonalization of A.
⋆ Then denote $g(\xi) = f(x) = f(A^{-1/2}\xi) = \frac{1}{2}\xi^T\xi - b^T A^{-1/2}\xi$ and note that this function is well conditioned
⋆ Write the GD algorithm for $\xi \mapsto f(A^{-1/2}\xi)$:

$$\xi_{n+1} = \xi_n - t\nabla g(\xi_n)$$

$$\xi_{n+1} = \xi_n - tA^{-1/2}\nabla f(A^{-1/2}\xi_n)$$

Then multiplying by $A^{-1/2}$ we get

$$x_{n+1} = x_n - tA^{-1}\nabla f(x_n).$$

Choosing the descent direction $-A^{-1}\nabla f(x)$ is equivalent to performing a GD step in the new metric!

General algorithm

incorporating all previous algorithms...

Algorithm 8 (Generic Variable Metric method)

Choose the starting point x_0

Iteration *i:*

- *compute* $f(x_i), \nabla f(x_i)$ *and eventually* $D^2 f(x_i)$
- *choose a symmetric positive-definite matrix* A_i*: compute the new direction*
$$d_i = -A_i^{-1} \nabla f(x_i)$$
- *perform a line-search from* x_i *in the direction* d_i *giving a new iterate*
$$x_{i+1} = x_i + t_i d_i = x_i - t_i A_i^{-1} \nabla f(x_i).$$

$\star$ $A_i = \mathrm{Id}$ gives the Gradient Descent method

$\star$ $A_i = D^2 f(x_i)$ gives the Newton method with line search (only when $D^2 f(x_i)$ is positive-definite)

$\star$ such an algorithm will converge to a critical point provided the set $\{f(x) \leq f(x_0)\}$ is bounded. The key point is that line-search guarantees descent: $f(x_{i+1}) < f(x_i)$ when not at a critical point

Modified Newton method

Idea: Choose A_i based on $D^2 f(x_i)$ by eventually changing the Hessian matrix to make it positive definite

1. Choose a threshold $\delta > 0$ and compute the spectral decomposition
$$D^2 f(x_i) = U_i D_i U_i^T.$$
If a diagonal value of D_i is smaller than δ then replace it with δ.
$\longrightarrow$ Large arithmetic cost: $2n^3$ to $4n^3$ arithmetic operations
2. Levenberg-Marquardt modification: $A_i = D^2 f(x_i) + \varepsilon Id$. Choose ε such that A_i is positive definite by using a bisection scheme.

   Test the positive-definiteness using the Cholesky Factorization: $A_i = LDL^T$ - arithmetic cost: $n^3/6$
3. Use a modified Cholesky factorization so that the resulting diagonal matrix has entries bigger than $\delta > 0$.

$\star$ all these techniques are too costly for large n
$\star$ we lose quadratic convergence as soon as $A_i \neq D^2 f(x_i)$ or the corresponding line-search step is smaller than 1

Conclusion: Newton's method

- quadratic convergence when we start close to a non-degenerate minimizer
- in order to guarantee convergence in general a line-search procedure should be used
- if $D^2 f(x_i)$ is not positive-definite then multiple ways exist to "correct the algorithm" but they are all costly: $O(n^3)$
- a linear system should be solved at each iteration
- the cost becomes too big if n is very large

Optimization in higher dimensions

- Theoretical aspects
- Gradient descent methods
- Newton's method
- Other methods

Gauss-Newton Method

⋆ non-linear least squares: assume $m \geq n$

$$f(x) = \sum_{j=1}^{m} r_j(x)^2$$

⋆ define the Jacobian matrix

$$J(x) = \begin{pmatrix} \frac{\partial r_1}{\partial x_1} & \dots & \frac{\partial r_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \dots & \frac{\partial r_m}{\partial x_n} \end{pmatrix}$$

⋆ note that $\nabla f(x) = 2(J(x))^T r$ where $r = (r_1, ..., r_m)$
⋆ Hessian computation: $D^2 f(x) = 2J(x)^T J(x) +$ something small...
⋆ choose to approximate the Hessian by $2J(x)^T J(x)$ which is positive definite when J is of maximal rank
⋆ Therefore we get the Gauss-Newton method

$$x_{i+1} = x_i - \gamma_i (J(x_i)^T J(x_i))^{-1} J^T(x_i) r(x_i)$$

where either $\gamma_i = 1$ or a line-search is performed
⋆ as before, if $-(J(x_i)^T J(x_i))^{-1} J^T(x_i) r(x_i)$ is not a descent direction, one may try to "fix the method"

Example 1

$\star$ the Rosenbrock function: $f(x) = 100(y - x^2)^2 + (1-x)^2 \Longrightarrow$
$r_1 = 10(y-x)^2,\ r_2 = (1-x)$

$\star$ $J(x) = \begin{pmatrix} -20x & 10 \\ -1 & 0 \end{pmatrix}$

$\star$ true Hessian vs Gauss-Newton approx:

$$H(x) = \begin{pmatrix} 1200x^2 - 400y + 2 & -400x \\ -400x & 200 \end{pmatrix}$$

$$2J^TJ = \begin{pmatrix} 800x^2 + 2 & -400x \\ -400x & 200 \end{pmatrix}$$

$\star$ Numerically this converges very fast, using only gradient information

Example 2: Triangulations

Suppose you know the coordinates (x_i, y_i) of three antennas and the distances d_i of a cellphone to these antennas, find the coordinates (x_0, y_0) of the cellphone.

⋆ least squares formulation:

$$f(x,y) = \sum_{i=1}^3 r_i^2, \quad r_i(x,y) = d_i - \sqrt{(x-x_i)^2+(y-y_i)^2}.$$

⋆ Gauss-Newton generally converges faster than GD here

Further examples

$\star$ Other important applications: least squares are often used when fitting models to data

$$f(x) = \sum_{i=1}^{m} r_i(x)^2 = \sum_{i=1}^{m} (y(s_i, x) - y_i)^2$$

where $y(s, x)$ is a non-linear function

$\star$ find parameters of a population model: exponential model, logistic model

$\star$ find parameters for a temperature model: $T(t) = A\sin(wt + \phi) + C$

Nelder-Mead method

⋆ simplex algorithm, gradient free

Algorithm 9 (Nelder-Mead method)

Current test points $x_1, ..., x_{n+1} \in \mathbb{R}^n$

1. **Order**: *relabel points such that* $f(x_1) \leq ... \leq f(x_{n+1})$
2. *Compute centroid* x_0 *of points* $x_1, ..., x_n$
3. **Reflection**: *compute* $x_r = x_0 + \alpha(x_0 - x_{n+1})$ *with* $\alpha > 0$. *If* $f(x_1) \leq f(x_r) < f(x_n)$ *then replace* x_{n+1} *by* x_r *and go to Step 1*
4. **Expansion**: *if* $f(x_r) < f(x_1)$ *compute* $x_e = x_0 + \gamma(x_r - x_0)$ *with* $\gamma > 1$.
   If $f(x_e) < f(x_r)$ *replace* x_{n+1} *by* x_e *and go to Step 1*
   Else *replace* x_{n+1} *by* x_r *and go to Step 1*
5. **Contraction**: *If* $f(x_r) \geq f(x_n)$ *then compute* $x_c = x_0 + \rho(x_{n+1} - x_0)$ *with* $\rho \in (0, 0.5]$. *If* $f(x_c) < f(x_{n+1})$ *then replace* x_{n+1} *by* x_c *and go to Step 1*
6. **Shrink:** *Replace all points except* x_1 *by* $x_i = x_1 + \sigma(x_i - x_1)$. *Go to Step 1*

⋆ Standard parameters: $\alpha = 1, \gamma = 2, \rho = 1/2, \sigma = 1/2$.

⋆ Termination criterion: Simplex too small, variation of f small, etc.