Introduction to Local and Global Optimization for NLP

Marco Trubian

Dipartimento di Scienze dell'Informazione (DSI)
Università degli Studi di Milano

Optimization

- Concerned with minimization/maximization of mathematical functions
- Often subject to constraints
- Important tool in analyzing physical, economic, chemical and biological systems
- Euler (1707-1783): Nothing at all takes place in the universe in which some rule of the maximum or minimum does not apply.
- Model $\rightarrow$ apply algorithm $\rightarrow$ check solution

Unconstrained optimization

Unconstrained

$$\begin{array}{ll} \min & f(\boldsymbol{x}) \\ s.t. & \boldsymbol{x} \in \mathbb{R}^n \end{array}$$

- f : Objective function
- $\boldsymbol{x}$: Decision variables
- $f : \mathbb{R}^n \to \mathbb{R}$
- f is assumed to be at least twice differentiable
- Examples: $f(x) = 2x^3 - 3x^2$.
- Important application: Data fitting and regression

Constrained optimization

Constrained

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ s.t. \quad & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, k \\ & h_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, h \end{aligned}$$

- $g(\boldsymbol{x})$: nonlinear inequality constraints
- $h(\boldsymbol{x})$: nonlinear equality constraints
- Application: Resource constrained problems, transportation problems

Convex optimization

Convex

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ s.t. \quad & \boldsymbol{x} \in X \subseteq \mathbb{R}^n \end{aligned}$$

- f : f is a convex function
- X is convex: if $\boldsymbol{x}^1, \boldsymbol{x}^2 \in X$ then $(\lambda \boldsymbol{x}^1 + (1-\lambda)\boldsymbol{x}^2) \in X, \ \lambda \in [0,1]$
- Any local solution is global
- In constrained optimization, $h(\boldsymbol{x}) = A\boldsymbol{x} - b$ and $g(\boldsymbol{x})$ are convex
- Application: Controller design

Discrete optimization

Discrete

$$\begin{array}{ll} \min & f(\boldsymbol{x}) \\ s.t. & \boldsymbol{x} \in X \cap \mathbb{Z}^n \end{array}$$

- $\boldsymbol{x}$ can take only discrete values
- e.g. $\boldsymbol{x} \in \{0, 1\}^n$
- Sometimes $\boldsymbol{x}^1 \in \mathbb{R}, \boldsymbol{x}^2 \in \{0, 1\}^n$ implying mixed-integer model
- Application: facility location, routing, combinatorial problems

Stochastic optimization

Stochastic

$$
\begin{aligned}
\min \quad & Ef(\mathbf{x};\omega) \\
s.t. \quad & \mathbf{x} \in \mathbb{R} \\
& c(\mathbf{x};\omega) \geq 0, \omega \in \Omega
\end{aligned}
$$

- Random variable ω belonging to a sample space Ω
- A constraint for each realization of uncertainty ω
- Minimize expected value
- Application: portfolio optimization problem

Our focus

- Smooth nonlinearly constrained optimization problems
- Local solutions (satisfy optimality conditions)
- Main idea:
  1. Check if current point satisfies optimality conditions
  2. If not, obtain new iterate and return to 1.
- Finding new iterate requires using local information

Curve fitting (1)

- Use a curve to fit experimental data
- Measurements $y_1, \ldots, y_m$ at times $t_1, \ldots, t_m$
- Model: $\phi(t, \boldsymbol{x}) = x_1 + x_2 e^{-(x_3 - t)^2/x_4} + x_5 \cos(x_6 t)$
- Model parameters: $x_1, \ldots, x_6$
- How does one set these parameters?

Curve fitting (2)

- Define residuals $r_j = y_j - \phi(t_j, \boldsymbol{x}), \quad j = 1, \ldots, m,$
- Residuals: measure discrepancy between actual and estimated
- Specify performance metric: $f(\boldsymbol{x}) = r_1^2(\boldsymbol{x}) + r_2^2(\boldsymbol{x}) + \ldots + r_m^2(\boldsymbol{x})$

Nonlinear Least Squares

$$\begin{array}{ll} \min & \sum_{j=1}^{m} r_j^2 \\ s.t. & \boldsymbol{x} \in \mathbb{R}^6 \end{array}$$

Weather forcasting: a limited number of parameters and a huge number of measurements

Global and Local Minimizers

- Consider a function $f(\mathbf{x})$ defined on $X \subseteq \mathbb{R}^n$
  Definition A point $\mathbf{x}^* \in X$ is a global minimizer of $f(\mathbf{x})$ if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in X$$

  $f(x) = (x-2)^2$ has a global minimizer $x^* = 2$

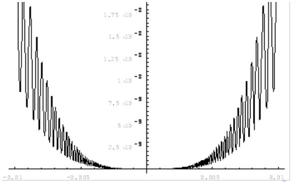
- **Definition** A point $\mathbf{x}^* \in X$ is a local minimizer of $f(\mathbf{x})$ if there exists a neighborhood $N(\mathbf{x}^*)$ of $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in N(\mathbf{x}^*)$$

- $f(x) = \sin(x)$ has local minimizers $x^* = \{3\pi/2 \pm 2n\pi, n \in Z\}$

Strict and Isolated Minimizers

- Consider a function $f(\boldsymbol{x})$ defined on $X \subseteq \mathbb{R}^n$
  Definition A point $\boldsymbol{x}^* \in X$ is a strict local minimizer of $f(\boldsymbol{x})$ if there exists a neighborhood $N(\boldsymbol{x}^*)$ of $\boldsymbol{x}^*$ such that
  $$f(\boldsymbol{x}^*) < f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in N(\boldsymbol{x}^*)$$
- $f(x) = \sin(x)$ has strict local minimizers $x^* = \{3\pi/2 \pm 2n\pi, n \in Z\}$
  Definition A point $\boldsymbol{x}^* \in X$ is a isolated local minimizer of $f(\boldsymbol{x})$ if there exists a neighborhood $N(\boldsymbol{x}^*)$ of $\boldsymbol{x}^*$ such that $\boldsymbol{x}^*$ is the only minimizer in $N(\boldsymbol{x}^*)$
- $f(x) = x^4 \cos(1/x) + 2x^4$ has a strict global minimizer at $x^* = 0$ but it is not isolated



Our focus

- Smooth nonlinearly constrained optimization problems
- Local solutions (satisfy optimality conditions)
- Main idea:
  1. Check if current point satisfies optimality conditions
  2. If not, obtain new iterate and return to 1.
- Finding new iterate requires using local information

The Fondamental Tools: Taylor's Theorems

Theorem (Taylor's First-Order Theorem) Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $\boldsymbol{h} \in \mathbb{R}^n$. Then we have that $f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T\boldsymbol{h} + o(\|\boldsymbol{h}\|^2)$.

Theorem (Taylor's Second-Order Theorem) Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and $\boldsymbol{h} \in \mathbb{R}^n$. Then we have that $f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T\boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^T H(\boldsymbol{x})\boldsymbol{h} + o(\|\boldsymbol{h}\|^3)$.

In order to manipulate any possible kind of function f the algorithms locally approximate it with linear or quadratic models

$$l(\boldsymbol{h}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T\boldsymbol{h}$$

$$m(\boldsymbol{h}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T\boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^T H(\boldsymbol{x})\boldsymbol{h}$$

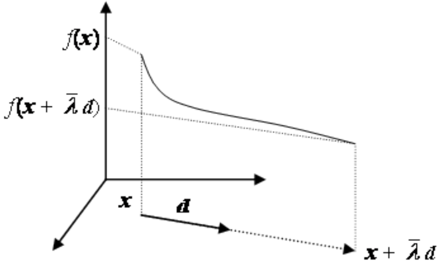
Our focus

- Smooth nonlinearly constrained optimization problems
- Local solutions (satisfy optimality conditions)
- Main idea:
  1. Check if current point satisfies optimality conditions
  2. If not, obtain new iterate and return to 1.
- Finding new iterate requires using local information

Descent directions: where to move

Definition (Descent direction and directional derivative) Given a function $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable in $\boldsymbol{x}$ and a vector $\boldsymbol{d}$

- if exists $\overline{\lambda} > 0$ such that $f(\boldsymbol{x} + \lambda \boldsymbol{d}) < f(\boldsymbol{x})$ for each $0 < \lambda < \overline{\lambda}$ then $\boldsymbol{d}$ is called a descent direction for f in $\boldsymbol{x}$
- $\lim\limits_{\lambda \to 0^+} \frac{f(\boldsymbol{x}+\lambda \boldsymbol{d})-f(\boldsymbol{x})}{\lambda} = \nabla f(\boldsymbol{x})^T \boldsymbol{d}$ is called directional derivative of f in $\boldsymbol{x}$ in the direction $\boldsymbol{d}$



Our focus

- Smooth nonlinearly constrained optimization problems
- Local solutions (satisfy optimality conditions)
- Main idea:
  1. Check if current point satisfies optimality conditions
  2. If not, obtain new iterate and return to 1.
- Finding new iterate requires using local information

Optimality Conditions: when to stop

Theorem (First-Order Necessary Conditions) If $\boldsymbol{x}^*$ is a local minimizer and f is continuously differentiable in a neighborhood of $\boldsymbol{x}^*$, then $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$.

Theorem (Second-Order Necessary Conditions) If $\boldsymbol{x}^*$ is a local minimizer and the hessian matrix $H(\boldsymbol{x})$ is continuous in a neighborhood of $\boldsymbol{x}^*$, then $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$ and $H(\boldsymbol{x}^*)$ is positive semidefinite.

Theorem (Second-Order Sufficient Conditions) Suppose that the hessian matrix $H(\boldsymbol{x})$ is continuous in an neighborhood of $\boldsymbol{x}^*$ and that $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$ and $H(\boldsymbol{x}^*)$ is positive definite. Then $\boldsymbol{x}^*$ is a strict local minimizer of f.

Convex Programming (1)

Definition A square matrix H of order n, is called positive (semi)definite on a set $X \subseteq \mathbb{R}^n$ if for each $\boldsymbol{d} \in X, \boldsymbol{d} \neq \boldsymbol{0}$, then

$$\boldsymbol{d}^T H \boldsymbol{d} > 0 \quad \text{positive definite}$$
$$\boldsymbol{d}^T H \boldsymbol{d} \geq 0 \quad \text{positive semidefinite}$$

Proposition A simmetric matrix H is positive (semi)definite if and only if

- the determinants of all its principal minors are $(\geq) > 0$
- its eigenvalues are $(\geq) > 0$

Proposition Given a simmetric matrix H then
$\lambda_{min} \|\boldsymbol{d}\|^2 \leq \boldsymbol{d}^T H \boldsymbol{d} \leq \lambda_{max} \|\boldsymbol{d}\|^2$, for each $\boldsymbol{d}$,
where λ_{min} := min eigenvalue and λ_{max} := max eigenvalue of H.

Observation If $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$ and the hessian matrix $H(\boldsymbol{x}^*)$ is not positive (semi)definite then $\boldsymbol{x}^*$ is neither a minimum nor a maximum.

Convex Programming (2)

Definition A function $f : \mathbb{R}^n \to \mathbb{R}$ defined on $X \subseteq \mathbb{R}^n$ is convex if X is convex and for each $\boldsymbol{x}, \boldsymbol{y} \in X$ the following relationship holds

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) \text{ for each } \lambda \in [0,1].$$

Proposition Let X be a convex set and let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, then

- f is convex if and only if the hessian matrix $H(\boldsymbol{x})$ is positive semidefinite in $\boldsymbol{x}$, for each $\boldsymbol{x} \in X$.
- if the hessian matrix $H(\boldsymbol{x})$ is positive definite in $\boldsymbol{x}$ then f is strictly convex in a neighbourhood of $\boldsymbol{x}$.

Theorem

- When f is a convex function, then any local minimizer $\boldsymbol{x}^*$ is global.
- If f is a convex differentiable function, then any stationary point $\boldsymbol{x}^*$ is a global minimizer of f.

Quadratic Programming (1)

Optimization algorithms locally approximate f with quadratic models

$$\begin{array}{ll} \min & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \\ t.c. & \boldsymbol{x} \in \mathbb{R}^n \end{array}$$

where Q is a simmetric square matrix of order n.

- $\nabla f(\boldsymbol{x}) = Q\boldsymbol{x} - \boldsymbol{b}$ (a linear function)
- $H(\boldsymbol{x}) = Q$ (a costant matrix)
- $Q\boldsymbol{x} = \boldsymbol{b}$ (first-order necessary conditions)

Quadratic Programming (2)

Optimization algorithms locally approximate f with quadratic models

$$\begin{array}{ll} \min & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \\ t.c. & \boldsymbol{x} \in \mathbb{R}^n \end{array}$$

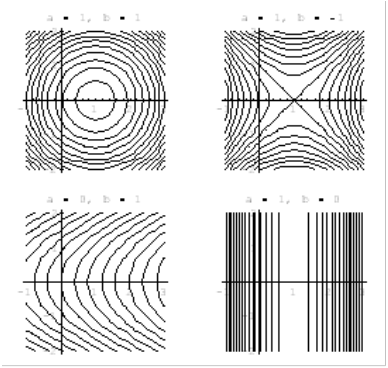
Possible cases:

- Q is not positive semidefinite: f has not minima
- Q is positive definite: $\boldsymbol{x}^* = Q^{-1}\boldsymbol{b}$ is the only global minimizer
- Q is positive semidefinite:
  - Q is not singular: $\boldsymbol{x}^* = Q^{-1}\boldsymbol{b}$ is the only global minimizer
  - Q is singular:
    - no solutions or
    - infinite solutions

Quadratic Programming (3)

Example: $f(x,y) = \frac{1}{2}\left(ax^2 + by^2\right) - x$

$$f(\mathbf{x}) = \frac{1}{2}(x,y)\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} - (x,y)\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



Example of QP: Portfolio Optimization

- $r_i, i = 1, \ldots, n$ r.v.: returns of n possible investiments
- $\mu_i = E[r_i]$ and $\sigma_i^2 = E[(r_i - \mu_i)^2]$ with normal distribution
- $\rho_{ij} = \frac{E[(r_i-\mu_i)(r_j-\mu_j)]}{\sigma_i \sigma_j}$ for $i, j = 1, \ldots, n$ correlations among investiments pairs
- $x_i, i = 1, \ldots, n,$ fraction of budget put into investiment i
- $\sum_{i=1}^n x_i = 1, \quad \boldsymbol{x} \geq \boldsymbol{0},$ all funds are invested
- $R = \sum_{i=1}^n x_i r_i,$ portfolio return
- $E[R] = \sum_{i=1}^n x_i E[r_i] = \boldsymbol{x}^T \boldsymbol{\mu},$ expected return
- $Var[R] = E[(R - E[R])^2] = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_i \sigma_j \rho_{ij} = \boldsymbol{x}^T G \boldsymbol{x}$ where $G_{ij} = \rho_{ij} \sigma_i \sigma_j$ is the symmetric positive semidefinite covariance matrix
- $\max \boldsymbol{x}^T \boldsymbol{\mu} - k \boldsymbol{x}^T G \boldsymbol{x}$, s.t. $\sum_{i=1}^n x_i = 1, \quad \boldsymbol{x} \geq \boldsymbol{0}$.

Algorithm Overview (1)

Continous optimization methods:

1. Given a starting point $\boldsymbol{x}_0$, generate an sequence $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$.
2. Terminate the algorithm, when necessary conditions are satisfied with some accuracy, say $\|\nabla f(\boldsymbol{x}_k)\| \leq \varepsilon$.
3. Monotone algorithms requires that $f(\boldsymbol{x}_k) < f(\boldsymbol{x}_{k-1})$ for all k

How good is an optimization algorithm ?

Local and Global Convergence

An optimization algorithm is *fair* if it converges...

Definition An algorithm is called globally convergent if it converges to a point $\boldsymbol{x}^*$ s.t. $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$ for each starting point $\boldsymbol{x}_0 \in \mathbb{R}^n$

Definition An algorithm is called locally convergent if it converges to a point $\boldsymbol{x}^*$ s.t. $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$ only if the starting point $\boldsymbol{x}_0 \in N(\boldsymbol{x}^*)$

N.B. nothing to do with the convergence to a local or a global optimum!

Rate of Convergence

An optimization algorithm is **good** if it converges rapidly!

- Rate of convergence properties discuss the behavior of an algorithm close to a solution
- How fast does the algorithm converge?

Let $\boldsymbol{x}_k$ be a sequence in $\mathbb{R}^n$ that converges to $\boldsymbol{x}^*$. Convergence is:

- *Q*-linear if $\exists r \in (0,1)$ s.t. $\frac{\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\right\|}{\left\|\boldsymbol{x}_k-\boldsymbol{x}^*\right\|} \leq r, \quad \text{for } k \geq \bar{k}.$
  E.g.: $\{\boldsymbol{x}_k\} = \frac{1}{2^k}$
- *Q*-superlinear if $\lim_{k\to\infty} \frac{\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\right\|}{\left\|\boldsymbol{x}_k-\boldsymbol{x}^*\right\|} = 0.$
  E.g.: $\{\boldsymbol{x}_k\} = \frac{1}{k!}$
- *Q*-quadratic if $\exists C > 0$ s.t. $\frac{\left\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\right\|}{\left\|\boldsymbol{x}_k-\boldsymbol{x}^*\right\|^2} \leq C, \quad \text{for } k \geq \bar{k}.$
  E.g.: $\{\boldsymbol{x}_k\} = \frac{1}{2^{2^k}}$

Q-quadratically $\rightarrow$ Q-superlinearly $\rightarrow$ Q-linearly

Algorithm Overview (2)

Continous optimization methods:

1. Given a starting point $\boldsymbol{x}_0$, generate an sequence $\{\boldsymbol{x}_k\}_{k=0}^{\infty}$.
2. Terminate the algorithm, when necessary conditions are satisfied with some accuracy, say $\|\nabla f(\boldsymbol{x}_k)\| \leq \varepsilon$.
3. Monotone algorithms requires that $f(\boldsymbol{x}_k) < f(\boldsymbol{x}_{k-1})$ for all k

How does one determine $\boldsymbol{x}_k$ given $\boldsymbol{x}_{k-1}$?

Two Main Strategies (1)

- Line search: Given an iterate $\boldsymbol{x}_k$, we first determine a direction $\boldsymbol{h}_k$. Then we determine a stepsize α_k given by

$$\min_{\alpha_k} f(\boldsymbol{x}_k + \alpha_k \boldsymbol{h}_k)$$

  The new iterate is given by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{h}_k$.

- Trust region: Construct a model function m_k using the information at $\boldsymbol{x}_k$. A trust-region radius Δ_k is selected and we obtain a $\boldsymbol{h}_k$ such that

$$\min_{\|\boldsymbol{h}_k\| \leq \Delta_k} m_k(\boldsymbol{h}_k)$$

  If $\boldsymbol{h}_k$ does not produce sufficient descent, we shrink Δ_k and resolve.

Two Strategies(2)

Line search:

1. Select direction
2. Determine stepsize

Trust Region:

1. Select trust region (stepsize)
2. Determine direction

Line Search algorithms

```
Descent Method;
{
    Choose x_0 ∈ ℝ^n;    k := 0;
    While ∇f(x_k) ≠ ∅;
    {
        compute d_k ∈ ℝ^n; /* descent direction */
        compute α_k ∈ ℝ; /* step along d_k */
        x_{k+1} = x_k + α d_k;
        k := k + 1;
    }
}
```

Line Search algorithms

Descent Method;
{
    Choose $\mathbf{x}_0 \in \mathbb{R}^n$; $k := 0$;
    While $\nabla f(\mathbf{x}_k) \neq \emptyset$;
    {
        compute $\mathbf{d}_k \in \mathbb{R}^n$; /* descent direction */
        compute $\alpha_k \in \mathbb{R}$; /* step along $\mathbf{d}_k$ */
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k$;
        $k := k + 1$;
    }
}

Stepsize specification

We assume that in $\boldsymbol{x}$ a descent direction $\boldsymbol{d}$ has been given

Stepsize Problem

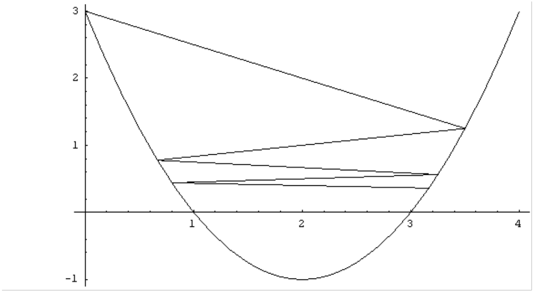
$$\min_{\alpha>0} \phi(\alpha) = f(\boldsymbol{x} + \alpha \boldsymbol{d})$$

- Tradeoff between effort and accuracy
- Global minimizer would be too costly from a computational standpoint
- Exact linesearch: α^* solves $\phi'(\alpha) = \nabla f(\boldsymbol{x} + \alpha \boldsymbol{d})^T \boldsymbol{d} = 0$.
- An inexact linesearch is generally used:
  - Cheap
  - Convergence rate does not rely on exact line search

Observation

The simple decreasing of f is not enough.

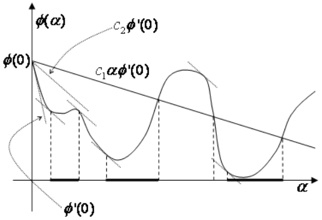
- $\phi(\alpha) = \alpha^2 - 4\alpha + 3$, convex, with $\alpha^* = 2$.
- $\alpha_0 = 0$
- $\{\alpha_k\}$ generated by $\alpha_k = 2 + (-1)^{k+1}(1 + 1/(k+1))$
- $\phi(\alpha_{k+1}) < \phi(\alpha_k)$, con $k = 0, 2, \ldots$.



Wolfe Conditions

To be effective inexact linesearch requires some conditions

- Sufficient Decrease: $f(\boldsymbol{x} + \alpha \boldsymbol{d}) \leq f(\boldsymbol{x}) + c_1 \alpha \nabla f(\boldsymbol{x})^T \boldsymbol{d}, \quad c_1 \in (0, 1)$
$$\phi(\alpha) \leq \phi(0) + \alpha c_1 \phi'(0)$$
- Curvature condition: $\nabla f(\boldsymbol{x} + \alpha \boldsymbol{d})^T \boldsymbol{d} \geq c_2 \nabla f(\boldsymbol{x})^T \boldsymbol{d}, \; c_1 \in (0, 1)$
$$\phi'(\alpha) \geq c_2 \phi(0)$$
- Collectively called Wolfe conditions where $0 < c_1 < c_2 < 1$
- Strong Wolfe conditions introduce a sign constraint on curvature $\left\|\nabla f(\boldsymbol{x} + \alpha \boldsymbol{d})^T \boldsymbol{d}\right\| \leq c_2 \left\|\nabla f(\boldsymbol{x})^T \boldsymbol{d}\right\|$



Existence of stepsize

Proposition Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Let $\boldsymbol{d}$ be a descent direction at $\boldsymbol{x}$ and assume that $\phi(\alpha)$ is bounded below for $\alpha > 0$. Then if $0 < c_1 < c_2 < 1$, there exist steplengths satisfying the (strong) Wolfe conditions.

Looking for α: backtracking line search

Armijo's method;
{
    Choose $\alpha_0 \in \mathbb{R}$; $\alpha := \alpha_0$;
    While $f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) > f(\boldsymbol{x}_k) + \alpha c_1 \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k$;
        $\alpha := \sigma\alpha$; /* backtracking */
    $\alpha^* := \alpha$;
}

$\sigma \approx 0.9$, α_0 often set to 1 for Newton and Quasi-Newton methods

Exact line search for QP

Let us consider a Quadratic Programming model

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$$

- $\phi(\alpha) = f(\boldsymbol{x} + \alpha\boldsymbol{d}) = \frac{1}{2}(\boldsymbol{x} + \alpha\boldsymbol{d})^T Q(\boldsymbol{x} + \alpha\boldsymbol{d}) - \boldsymbol{b}^T(\boldsymbol{x} + \alpha\boldsymbol{d})$
- $\phi'(\alpha) = \nabla f(\boldsymbol{x} + \alpha\boldsymbol{d})^T\boldsymbol{d} = 0.$
- $(Q(\boldsymbol{x} + \alpha\boldsymbol{d}) - \boldsymbol{b})^T\boldsymbol{d} = 0$
- $\alpha^* = -\frac{\boldsymbol{x}^T Q\boldsymbol{d} - \boldsymbol{b}^T\boldsymbol{d}}{\boldsymbol{d}^T Q\boldsymbol{d}} = -\frac{\nabla f(x)^T\boldsymbol{d}}{\boldsymbol{d}^T Q\boldsymbol{d}}$

This result is used in the convergence analysis of optimization algorithms.

On computing α

Other techniques for computing α

- Interpolation: using quadratic or cubic models
- Derivative free techniques, assuming convexity of $\phi(\alpha)$
  - Golden Section Method,
  - Fibonacci Method,
  - Bisection Method.

Convergence of inexact linesearch schemes (1)

If we define θ_k as the angle between $\boldsymbol{d}_k$ and $-\nabla f_k$, then

$$\cos\theta_k = \frac{-\nabla f(\boldsymbol{x})_k^T \boldsymbol{d}_k}{||\nabla f(\boldsymbol{x})_k|| \cdot ||\boldsymbol{d}_k||}$$

Theorem Let $\boldsymbol{d}_k$ be a descent direction and let α_k satisfy the Wolfe conditions. Also assume that f is bounded below on $\mathbb{R}^n$ and continuously differentiable on N which contains the level set $L_f := \{\boldsymbol{x} : f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$, where $\boldsymbol{x}_0$ is the starting point. We also assume that ∇f is Lipschitz continuous on N. Then

$$\sum_{j=0}^{\infty} \cos^2\theta_j ||\nabla f(\boldsymbol{x}_j)||^2 < \infty.$$

Convergence of inexact linesearch schemes (2)

Proof:

- $\nabla f_{k+1}^T \boldsymbol{d}_k \geq c_2 \nabla f_k^T \boldsymbol{d}_k$ (Curvature condition)
- $(\nabla f_{k+1} - \nabla f_k)^T \boldsymbol{d}_k \geq (c_2 - 1) \nabla f_k^T \boldsymbol{d}_k$
- $(\nabla f_{k+1} - \nabla f_k)^T \boldsymbol{d}_k \leq ||\nabla f_{k+1} - \nabla f_k|| \, ||\boldsymbol{d}_k|| \leq \alpha_k L ||\boldsymbol{x}_{k+1} - \boldsymbol{x}_k|| \, ||\boldsymbol{d}_k|| = \alpha_k L ||\boldsymbol{d}_k||^2$ (from Schwartz and Lipschitz inequalities)
- $\alpha_k \geq \frac{c_2 - 1}{L} \frac{\nabla f_k^T \boldsymbol{d}_k}{||\boldsymbol{d}_k||^2}$
- $f_{k+1} \leq f_k + c_1 \alpha_k \boldsymbol{d}_k^T \nabla f_k = f_k - c_1 \frac{1 - c_2}{L} \frac{(\nabla f_k^T \boldsymbol{d}_k)^2}{||\boldsymbol{d}_k||^2}$ (Sufficient Decrease)
- $f_{k+1} \leq f_k - c \cos^2 \theta_k ||\nabla f_k||^2$ where $c = c_1(1 - c_2)/L$
- $f_{k+1} \leq f_0 - c \sum_{j=0}^{k} \cos^2 \theta_j ||\nabla f_j||^2$ (by recursion)
- $c \sum_{j=0}^{k} \cos^2 \theta_j ||\nabla f_j||^2 \leq f_0 - f_{k+1} < \infty$ (by boundness of f)
  Zoutendijk condition

Convergence of inexact linesearch schemes (3)

Zoutendijk condition

$$\sum_{j=0}^{k} \cos^2 \theta_j ||\nabla f_j||^2 < \infty$$

implies

$$\cos^2 \theta_j ||\nabla f(\boldsymbol{x}_j)||^2 \to 0$$

Hence if the algorithm satisfies **also** the angle condition

$$\cos \theta_k \geq \varepsilon > 0$$

then it converges

$$\lim_{k\to\infty} ||\nabla f(\boldsymbol{x}_k)|| = 0$$

Convergence of inexact linesearch schemes (4)

Given a function f bounded below on $\mathbb{R}^n$ and continuously differentiable on N which contains the level set $L_f := \{\boldsymbol{x} : f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$, with ∇f Lipschitz continuous on N then an iterative method,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$$

starting at $\boldsymbol{x}_0$ converges, i.e.

$$\lim_{k \to \infty} ||\nabla f(\boldsymbol{x}_k)|| = 0$$

- if $\boldsymbol{d}_k$ is a descent direction which satisfies the angle condition
- and α_k satisfies the Wolfe conditions
  - Sufficient Decrease: $f(\boldsymbol{x} + \alpha \boldsymbol{d}) \leq f(\boldsymbol{x}) + c_1 \alpha \nabla f(\boldsymbol{x})^T \boldsymbol{d}, \quad c_1 \in (0,1)$
  - Curvature condition: $\nabla f(\boldsymbol{x} + \alpha \boldsymbol{d})^T \boldsymbol{d} \geq c_2 \nabla f(\boldsymbol{x})^T \boldsymbol{d}, \; c_1 \in (0,1)$ where $0 < c_1 < c_2 < 1$