MATH: Operations Research **2014-15 First Term**

Handout 10: Nonlinear Programming: an Introduction

Instructor: Junfeng Yang *November 17, 2014*

Contents

10.1 Nonlinear Programming Models

Optimization in general form ($f : R^n \to R$, $X \subset R^n$)

$$\begin{aligned} \min \quad & f(x) \\ s.t. \; & x \in X. \end{aligned}$$

Linear programming

$$\begin{aligned} \min \quad & c^T x \\ s.t. \; & Ax = b \\ & x \geq 0. \end{aligned}$$

Nonlinear programming ($f, c_i : R^n \to R$, $i = 1, 2, \ldots, m$)

$$\begin{aligned} \min \quad & f(x) \\ s.t. \; & c_i(x) \leq 0, \, i = 1, 2, \ldots, m, \end{aligned}$$

or

$$\begin{aligned} \min \quad & f(x) \\ s.t. \; & c_i(x) = 0, \, i = 1, 2, \ldots, m_e, \\ & c_i(x) \leq 0, \, i = m_e + 1, 2, \ldots, m, \end{aligned}$$

where at least one of the functions is nonlinear.

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

Convex optimization The optimization problem $\min\{f(x) : s.t. \; x \in X\}$ is called a convex optimization problem if $X \subset R^n$ is a convex set and $f : X \to R$ is a convex function.

A commonly studied form of convex optimization

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & Ax = b, \\ & c_i(x) \leq 0, \, i = 1, 2, \ldots, m, \end{aligned}$$

where all $c_i : R^n \to R$, $i = 1, 2, \ldots, m$, are convex functions.

Objectives for different problems

- Convex optimization: seek a globally optimal solution;
- General nonlinear programming (objective or constraint functions are not known to be convex): global optimization is too ambitious and local optimization is a compromise that has to be taken. Sometimes, only a KKT/stationary point can be guaranteed.

Things to learn

- Theory: optimality conditions, duality theory
- Study of various numerical algorithms (construction of algorithms, convergence and numerical performance, etc.)
- Applications

10.2 Optimality conditions for unconstrained optimization

Let $f : R^n \to R$. We consider unconstrained optimization

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

We focus on the class of continuously differentiable functions, i.e., $f \in C^1(R^n)$. Most algorithms are constructed based on derivatives (gradient, Hession). Direct algorithms, which do not use derivative information, are also very useful in practical applications.

For any $x, d \in R^n$, the Taylor's expansion tells that

$$f(x + td) = f(x) + t\nabla f(x)^T d + o(t),$$

from which we can see that

$$\exists \delta > 0 \text{ such that } f(x + td) < f(x), \ \forall t \in (0, \delta),$$

if and only if $\nabla f(x)^T d < 0$.

Definition 10.1 (descent direction) *Let $f \in C^1(R^n)$ and $x \in R^n$. A vector $d \in R^n$ is called a descent direction of f at x if*

$$\nabla f(x)^T d < 0.$$

Theorem 10.2 (First order necessary condition) *Let $f \in C^1(R^n)$. If $x^* \in R^n$ is a local minimizer of f, then $\nabla f(x^*) = 0$.*

Proof: Since $x^* \in R^n$ is a local minimizer of f, there is no descent direction at x^*, i.e., for any $d \in R^n$, it holds that

$$\nabla f(x^*)^T d \geq 0.$$

Setting $d = -\nabla f(x^*)$ completes the proof. ■

Definition 10.3 (stationary point) *A point $x^* \in R^n$ is said to be a stationary (or critical) point of a differentiable function f if $\nabla f(x^*) = 0$.*

Theorem 10.4 (Second order necessary condition) *Let $f \in C^2(R^n)$, i.e., $f : R^n \to R$ is twice continuously differentiable. If $x^* \in R^n$ is a local minimizer of f, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$.*

Proof: From Taylor's expansion, for any $\alpha > 0$ and $d \in R^n$, it holds that

$$\frac{f(x^* + \alpha d) - f(x^*)}{\alpha^2} = \frac{1}{2} d^T \nabla^2 f(x^*) d + o(1).$$

If there exist $d \in R^n$ such that $d^T \nabla^2 f(x^*) d < 0$, then

$$\frac{1}{2} d^T \nabla^2 f(x^*) d + o(1) < 0$$

for $\alpha > 0$ sufficiently small, in which case $f(x^* + \alpha d) < f(x^*)$. This contradicts to the fact that x^* is a local minimizer of f. Thus, $\nabla^2 f(x^*) \succeq 0$. ■

Theorem 10.5 (Second order sufficient condition) *Let $f \in C^2(R^n)$. If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, then x^* is a strict local minimizer of f.*

Proof: Since $f \in C^2(R^n)$ and $\nabla^2 f(x^*) \succ 0$, there exist $\delta > 0$ such that $\nabla^2 f(x) \succ 0$ in $B(x^*, \delta)$ (open neighborhood of x^* with radius δ). For any $d \in R^n$, $0 < \|d\| < \delta$, it holds that

$$f(x^* + d) = f(x^*) + \frac{1}{2} d^T \nabla^2 f(x^* + \theta d) d,$$

for some $\theta \in (0, 1)$. Since $\nabla^2 f(x^* + \theta d) \succ 0$, it is clear that

$$f(x^* + d) > f(x^*),$$

which implies that x^* is a strict local minimizer of f. ■

For convex function, we have stronger results.

Theorem 10.6 *Let $C \subset R^n$ be a nonempty convex set and $f : C \to R$. Suppose $x^* \in C$ is a local minimizer for $\min_{x \in C} f(x)$. Then*

- *If f is convex, then x^* is also a global minimizer.*
- *If f is strictly convex, then x^* is a unique global minimizer.*

Theorem 10.7 *Let $f : R^n \to R$ be a differentiable convex function. Then x^* is a global minimizer if and only if $\nabla f(x^*) = 0$.*

Proof: Since f is differentiable and convex, for any $x \in R^n$, it holds that

$$f(x) \geq f(x^*) + \nabla f(x^*)^T (x - x^*) = f(x^*).$$

This completes the proof of sufficiency. ■

10.3 Structure of optimization algorithms

Most optimization algorithms are iterative in nature; Starting at an initial point x_0, an algorithm generates a sequence of points $\{x_k : k = 1, 2, \ldots\}$; The sequence is either finite or infinite. If finite, the last point is the solution/stationary point of the problem; If infinite, generally any limit point of the sequence is a solution of the problem; A desirable optimization algorithm should be able to (1) approach a solution point stably when the current point is far away; and (2) converge quickly to a solution when already close to one. These two points corresponds to global convergence and local convergence of an algorithm.

For unconstrained optimization, the structure of an algorithm is generally as follows.

Algorithm 1 (structure of unconstrained optimization algorithm of line search type) *Initialization: provide initial point x_0, algorithmic parameters, etc.*

1. *Find d_k satisfies $\nabla f(x_k)^T d_k < 0$;*
2. *Find $\alpha_k > 0$ such that $f(x_k + \alpha_k d_k) < f(x_k)$.*
3. *Check stopping criterion. If satisfied, stop; otherwise, repeat.*

This is the structure of line search type methods. Trust region type methods are different.

10.4 Step size rules / Line search

Notation. Use subscript k to count iteration number, i.e., a sequence of points generated by an algorithm will be denoted by $\{x_k : k = 1, 2, \ldots\}$. For simplicity the gradient of f at x is sometimes denoted by $g(x)$, i.e., $g(x) = \nabla f(x)$. Thus, sometimes $\nabla f(x_k)$ is denoted by g_k, i.e., $g_k := \nabla f(x_k)$. Also, occasionally $f(x_k)$ is shortened as f_k, i.e., $f_k := f(x_k)$.

Suppose d_k is a descent direction at the current point x_k, i.e., $g_k^T d_k < 0$ and the iteration formula is

$$x_{k+1} = x_k + h_k d_k.$$

There exist a few step size rules to determine h_k.

1. Predetermined: for examples

$$h_k = h > 0 \text{ (constant step)}, \quad h_k = \frac{h}{\sqrt{k+1}}.$$

   (simple, mainly used in gradient method applied to convex and Lipschitz problems.)

2. Exact line search: Find $h_k > 0$ such that

$$h_k = \arg\min_{h \geq 0} f(x_k + hd_k).$$

   Recall that $g_{k+1} = \nabla f(x_{k+1}) = \nabla f(x_k + h_k d_k)$. Consequence: $g_{k+1}^T d_k = 0$, i.e., g_{k+1} is perpendicular to d_k. For gradient method, it holds that $d_k = -g_k$ and thus

$$g_{k+1}^T g_k = 0.$$

   Exact line search is mainly studied theoretically and rarely used in practice.

3. Goldstein-Armijo line search rule. Let α and β be given parameters which satisfy $0 < \alpha < \beta < 1$. The Goldstein-Armijo rule determines a step size h_k such that

$$f(x_{k+1}) \leq f(x_k) + \alpha h_k g_k^T d_k,$$
$$f(x_{k+1}) \geq f(x_k) + \beta h_k g_k^T d_k.$$

   Let $\phi(h) = f(x_k + hd_k)$. The two conditions are equivalent to

$$\phi(h_k) \leq \phi(0) + \alpha\phi'(0)h_k, \quad \text{(suffiicent decrease)}$$
$$\phi(h_k) \geq \phi(0) + \beta\phi'(0)h_k, \quad (h_k \text{ not too small}).$$

   Such h_k exists unless $\phi(h)$ ($h \geq 0$) is unbounded below.

4. Wolfe-Powell line search rule. Let $\gamma \in (\alpha, 1)$. The Wolfe-Powell rule determines a step size h_k such that

$$f(x_{k+1}) \leq f(x_k) + \alpha h_k g_k^T d_k,$$
$$g_{k+1}^T d_k \geq \gamma g_k^T d_k.$$

   The second condition is equivalent to $\phi'(h_k) \geq \gamma\phi'(0)$. Suppose $\hat{h}_k > 0$ satisfies $f(x_k + \hat{h}_k d_k) = f(x_k) + \alpha \hat{h}_k g_k^T d_k$. Then,

$$\hat{h}_k \nabla f(x_k + \theta_k \hat{h}_k d_k)^T d_k = f(x_k + \hat{h}_k d_k) - f(x_k) = \alpha \hat{h}_k g_k^T d_k.$$

   Since $\gamma > \alpha$, the above implies that $h_k := \theta_k \hat{h}_k$ satisfies the second condition.

5. Strong Wolfe-Powell line search rule. The strong Wolfe-Powell rule determines a step size h_k such that

$$
\begin{aligned}
&f(x_{k+1}) \leq f(x_k) + \alpha h_k g_k^T d_k, \\
&|g_{k+1}^T d_k| \leq \gamma |g_k^T d_k|.
\end{aligned}
$$

   Theoretically, $\gamma \to 0$ implies exact line search.

6. Backtracking line search. Let $0 < \delta < 1$. Initialize $h_k = \hat{h} > 0$ (e.g., $\hat{h} = 1$), repeat $h_k = \delta \hat{h}$ until

$$
f(x_{k+1}) \leq f(x_k) + \frac{h_k}{2} g_k^T d_k.
$$

   (easy to be realized and frequently used in practice.)

7. Curvilinear search. Define a curve $\{x_k(h) : h \geq 0\}$ at x_k which satisfies

$$
\frac{\mathrm{d} f(x_k(h))}{\mathrm{d} h}|_{h=0} < 0.
$$

   At iteration k, search along the curve $\{x_k(h) : h \geq 0\}$ and determine $h_k > 0$ such that certain decrease conditions are satisfied.

8. Nonmonotone line search. Let $0 < \delta < 1$. Initialize $h_k = \hat{h} > 0$ (e.g., $\hat{h} = 1$), repeat $h_k = \delta \hat{h}$ until

$$
f(x_{k+1}) \leq C_k + \frac{h_k}{2} g_k^T d_k,
$$

   where $C_k := \max\{f_k, f_{k-1}, \ldots, f_{k-m+1}\}$ and m is a predetermined positive integer.

References

[Yuan-Sun] 袁亚湘、孙文瑜著，科学出版社 最优化理论与方法.