Lecture Notes – Part 9

Unconstrained Nonlinear Programming

1 Introduction

Nonlinear programming (NLP) is somewhat more complicated than linear programming (LP). We start the discussion from the most simplified version – unconstrained NLP, i.e. NLP with no constraints. Assume that $f(\mathbf{x})$ is a nonlinear function of vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ defined over the domain $\mathcal{D} \subseteq \mathbf{R}^n$. Consider an NLP problem

$$\min_{\mathbf{x}\in\mathcal{D}} \text{ (or } \max_{\mathbf{x}\in\mathcal{D}}) \ f(\mathbf{x}).$$

If $\mathcal{D} \equiv \mathbf{R}^n$, then we have an unconstrained NLP problem

$$\min \text{ (or } \max) \ f(\mathbf{x}),$$

where no constraints are placed on the decision variables $\mathbf{x}$.
Although most of the content in this chapter is largely related to calculus you may have learnt, we address more closely the question of how to actually find an optimal solution instead of how to recognise one.

To carefully specify what sort of NLP problems we will consider, let's start by discussing *convexity* for minimisation NLP (or *concavity* for maximisation NLP).

2 Convexity

In this section, we briefly consider what kind of situations could make an optimisation problem particularly hard to solve. We will need the following important definitions.

2.1 Global minimum

A point $\mathbf{x}^*$ is called a *global minimiser* or a *global minimum point* of a function $f(\mathbf{x})$ if

$$\mathbf{x}^* \in \mathcal{D}, \text{ and}$$

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for any } \mathbf{x} \in \mathcal{D}.$$

The value $f(\mathbf{x}^*)$ is called a *global minimum value* of $f(\mathbf{x})$.
Similarly defined is a *strict global minimiser* or a *strict global minimum point*, where

$$f(\mathbf{x}^*) < f(\mathbf{x}) \text{ for any } \mathbf{x} \in \mathcal{D} \setminus \{\mathbf{x}^*\}.$$

2.2 Local minimum

A point $\mathbf{x}^*$ is called a *local minimiser* or a *local minimum point* of a function $f(\mathbf{x})$ if

$$\mathbf{x}^* \in \mathcal{D},$$

and there exists an $\varepsilon > 0$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for any } \mathbf{x} \in \mathcal{D} \text{ satisfying}$$

$$0 < \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon.$$

The value $f(\mathbf{x}^*)$ is called a *local minimum value* of $f(\mathbf{x})$.

Similarly defined is a *strict local minimiser* or a *strict local minimum point*, where

$$f(\mathbf{x}^*) < f(\mathbf{x}) \text{ for any } \mathbf{x} \in \mathcal{D} \text{ satisfying}$$

$$0 < \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon.$$

It is possible for a function to have

- both global and local minimisers;
- neither global nor local minimisers;
- a local minimiser and yet no global minimiser;
- multiple global minimisers.

The basic notion of the solution techniques for the minimisation NLP we will introduce later is to move from some solution "downhill" to a better solution. These methods are guaranteed to find a local minimum, but in some cases there could be many different local minima such that the same algorithm with different starting points will end up with different local minima. And in most cases we are not able to ensure if the obtained local minimum is the global minimum.
Opposite to the convexity, global minimum and local minimum for the minimisation NLP, the *concavity*, *global maximum* and *local maximum* for the maximisation NLP can be similarly defined.

Now we limit ourselves to a specific type of NLP problems – minimising a *convex function* (or maximising a *concave function*) over a *convex set*.

2.3 Convex function

A function $f(\mathbf{x})$ is called *convex* if for any two points (or vectors) $\mathbf{x}_1 \in \mathcal{D}$ and $\mathbf{x}_2 \in \mathcal{D}$ and for any $\alpha \in [0,1]$ we have

$$f\big(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2\big) \leq \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2).$$

Function $f(\mathbf{x})$ is called *strictly convex* if for any $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathcal{D}$ and for any $\alpha \in (0,1)$ we have

$$f\big(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2\big) < \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2).$$

Assume that $f(\mathbf{x})$ has continuous second-order partial derivatives. At each point $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, we denote by

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)^T$$

the *gradient* of $f(\mathbf{x})$ at point $\mathbf{x}$, and by

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{pmatrix}$$

the *Hessian matrix* for $f(\mathbf{x})$ at point $\mathbf{x}$. Note that the Hessian is a symmetric matrix since if $f(\mathbf{x})$ has second-order partial derivatives at point $\mathbf{x}$ we have for $1 \leq i, j \leq n$

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} = \frac{\partial^2 f(\mathbf{x})}{\partial x_j \partial x_i}.$$

Definition. For each $i = 1, \ldots, n$, the i^{th} *principal minor*(s) of an $n \times n$ matrix is the determinant of any $i \times i$ matrix obtained by deleting $(n - i)$ row(s) and the corresponding $(n - i)$ column(s)[1] of the matrix.

Example 1. For the matrix

$$\mathbf{A} = \begin{pmatrix} -2 & -1 \\ -1 & 4 \end{pmatrix}.$$

The 1^{st} principal minors are -2 and 4.
The 2^{nd} principal minor is $|\mathbf{A}| = (-2)(4) - (-1)(-1) = -9$.
For any matrix, the 1^{st} principal minors are just the diagonal entries of the matrix.

By applying the theorem stated below, the Hessian matrix can be used to determine whether a function $f(\mathbf{x})$ is convex.

Theorem 1. Suppose that $f(\mathbf{x})$ has continuous second-order partial derivatives at each point $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathcal{D}$. Then $f(\mathbf{x})$ is a convex function on $\mathcal{D}$ if and only if for each $\mathbf{x} \in \mathcal{D}$ all principal minors of its Hessian are nonnegative.

Example 2. The Hessian matrix of the function $f(\mathbf{x}) = x_1^2 + 2x_1x_2 + x_2^2$ at any point $\mathbf{x} = (x_1, x_2) \in \mathbf{R}^2$ is

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

The 1^{st} principal minors are $2 > 0$ and $2 > 0$. The 2^{nd} principal minor is $2 \times 2 - 2 \times 2 = 0$. So, Theorem 1 shows that $f(\mathbf{x})$ is a convex function on $\mathbf{R}^2$.

[1]That is, if three rows 1, 3 and 4 are the deleted rows, then the corresponding three columns to be deleted are columns 1, 3 and 4.

2.4 Concave function

A function $f(\mathbf{x})$ is called concave if for any two points (or vectors) $\mathbf{x}_1 \in \mathcal{D}$ and $\mathbf{x}_2 \in \mathcal{D}$ and for any $\alpha \in [0,1]$ we have

$$f\big(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2\big) \geq \alpha f(\mathbf{x}_1) + (1-\alpha)f(\mathbf{x}_2).$$

Function $f(\mathbf{x})$ is called *strictly concave* if for any $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathcal{D}$ and for any $\alpha \in (0,1)$ we have

$$f\big(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2\big) > \alpha f(\mathbf{x}_1) + (1-\alpha)f(\mathbf{x}_2).$$

Note:

- It is apparent that a function $f(\mathbf{x})$ is (strictly) concave if and only if the function $-f(\mathbf{x})$ is (strictly) convex.
- If $f(\mathbf{x})$ is a linear function, i.e. $f(\mathbf{x}) = \mathbf{c}^T\mathbf{x}$ for some constant vector $\mathbf{c}$, then

  $$f\big(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2\big) = \alpha f(\mathbf{x}_1) + (1-\alpha)f(\mathbf{x}_2).$$

  Thus, linear functions are both convex and concave. But they are neither strictly convex nor strictly concave.

Theorem 2. Suppose that $f(\mathbf{x})$ has continuous second-order partial derivatives at each point $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathcal{D}$. Then $f(\mathbf{x})$ is a concave function on $\mathcal{D}$ if and only if for each $\mathbf{x} \in \mathcal{D}$ and each $k = 1, \ldots, n$ all nonzero k^{th} principal minors of its Hessian matrix have the same sign as $(-1)^k$.

Example 3. The Hessian of the function $f(\mathbf{x}) = -3x_1^2 + 4x_1x_2 - 2x_2^2$ at any point $\mathbf{x} = (x_1, x_2) \in \mathbf{R}^2$ is

$$\nabla^2(\mathbf{x}) = \begin{pmatrix} -6 & 4 \\ 4 & -4 \end{pmatrix}$$

The 1^{st} principal minors are $-6 < 0$ and $-4 < 0$. The 2^{nd} principal minor is $(-6) \times (-4) - (4) \times (4) = 8 > 0$. Theorem 2 shows that $f(\mathbf{x})$ is a concave function on $\mathbf{R}^2$.

Now we discuss the domain, i.e. feasible region, of the considered objective function.

2.5 Convex set

Recall that the set $\mathbf{S}$ is called convex if for any two $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{S}$ and any $\alpha \in (0, 1)$ we have $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \in \mathbf{S}$.[2]
Note:

- The feasible set we consider in LP

$$A\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

  is convex.[3]

- If $g(\mathbf{x})$ is a convex function, then the set $\mathbf{S} = \{\mathbf{x} : g(\mathbf{x}) \leq c\}$ for any constant c (if existing) is convex.

- If $g(\mathbf{x})$ is a convex function, then the set

$$\mathbf{S} = \{\mathbf{u} = (\mathbf{x}|y) = (x_1, x_2, \cdots, x_n, y) : y \geq g(\mathbf{x})\}$$

  is a convex set of $\mathbf{R}^{n+1}$. If you "colour in" above the graph of a convex function, then you get a convex set.

[2] Please refer to Chapter 2.
[3] However, the set of integers satisfying the above conditions is not a convex set.

Theorem 3. If $f(\mathbf{x})$ is a convex function and $\mathbf{S}$ is a convex set, then any local minimum of the minimisation NLP

$$\begin{aligned} &\min \ f(\mathbf{x}) \\ &s.t. \ \mathbf{x} \in \mathbf{S} \end{aligned}$$

is also a global minimum. If $f(\mathbf{x})$ is a strictly convex function, then the global minimum will be unique.[4]

3 Types of Optimality Conditions

3.1 One-dimensional case

Assume that a single-variable function $f(x)$ is defined and has continuous second-order derivatives on $\mathbf{R}$. Then for any point x^*, the Taylor's theorem states that for "small" value δ we have

$$f(x^* + \delta) = f(x^*) + f'(x^*)\delta + \frac{1}{2}f''(x^*)\delta^2 + o(\delta^2),$$

where $o(\delta^2)$ indicates a term that goes to zero faster than δ^2 does as $\delta \to 0$. In other words,

$$\lim_{\delta \to 0} \frac{o(\delta^2)}{\delta^2} = 0.$$

This is a formal way of denoting that this remainder term gets very small if δ is close to zero, and is "dominated" by the other terms.

The Taylor's formula leads to the following necessary condition and sufficient condition for x^* to be a local minimum of $f(x)$.

[4]The similar result applies to the concave function and maximisation NLP.

First-order condition: If x^* is a local minimum of $f(x)$, then $f'(x^*) = 0$.
This condition is also referred to as a **necessary condition** for a local minimum, since it must happen in order for x^* being a local minimum. But if $f'(x^*) = 0$, we don't know for sure whether we have a local minimum. Thus, it is not a sufficient condition, since it does not guarantee that x^* will be a local minimum.

Second-order condition: If $f'(x^*) = 0$ and $f''(x^*) > 0$, then x^* is a local minimum of $f(x)$.
This condition is also referred to as a **sufficient condition** for a local minimum.

3.2 Multi-dimensional case

Assume that an n-variable function $f(\mathbf{x})$ is defined and has continuous second-order partial derivatives on $\mathbf{R}^n$. Then for any point $\mathbf{x}^*$, the Taylor's theorem states that for "small" deviation $\mathbf{d} = (d_1, d_2, \cdots, d_n)^T$ we have the formula

$$f(\mathbf{x}^*+\mathbf{d}) = f(\mathbf{x}^*)+\sum_{i=1}^{n} \frac{\partial f(\mathbf{x}^*)}{\partial x_i} d_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 f(\mathbf{x}^*)}{\partial x_i \partial x_j} d_i d_j + o(\|\mathbf{d}\|^2)$$

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

The optimality conditions for the n-dimensional case is shown as follows.

3.2.1 First-order optimality condition

If $\mathbf{x}^*$ is a local minimum of $f(\mathbf{x})$, then $\nabla f(\mathbf{x}^*) = \mathbf{0}$.

This condition is also referred to as a **necessary condition** for a local minimum since it must happen in order for $\mathbf{x}^*$ being a local minimum. But if $\nabla f(\mathbf{x}^*) = \mathbf{0}$, we have no idea about whether we have a local minimum. It is not a sufficient condition since it does not guarantee that $\mathbf{x}^*$ will be a local minimum.

We call a point $\mathbf{x}^*$, where $\nabla f(\mathbf{x}^*) = \mathbf{0}$, a stationary point of $f(\mathbf{x})$. For unconstrained NLP problems, all local minima are stationary points.

Example 4. Consider the function $f(\mathbf{x}) = f(x_1, x_2) = x_1^2 - x_1x_2 + x_2^2 - 3x_2$. Then we have

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}\right)^T = (2x_1 - x_2, -x_1 + 2x_2 - 3).$$

Solving the system of equations $\nabla f(\mathbf{x}) = \mathbf{0}$ gives $\mathbf{x}^* = (1, 2)^T$, which is a stationary point of $f(\mathbf{x})$.

3.2.2 Second-order optimality condition

Recall that in one-dimensional case the sufficient condition for a local minimum is "$f'(x^*) = 0$ and $f''(x^*) > 0$". If $f''(x^*) = 0$, then further investigations are necessary.

In the n-dimensional case, the second-order derivative is generalised to the Hessian matrix. If $\mathbf{x}^*$ is a stationary point, then the Taylor's formula at $\mathbf{x}^*$ gives an approximation

$$f(\mathbf{x}^* + \mathbf{d}) \cong f(\mathbf{x}^*) + \frac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{x}^*)\mathbf{d}.$$

If $f(\mathbf{x}^* + \mathbf{d}) \geq f(\mathbf{x}^*)$, i.e. $\mathbf{x}^*$ is a local minimum, then we have $\mathbf{d}^T\nabla^2 f(\mathbf{x}^*)\mathbf{d} \geq 0$ for any $\mathbf{d}$.

Definition: An $n \times n$ symmetric matrix $\mathbf{A}$ is called

- *positive definite* if $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for any n-dimensional vector $\mathbf{x} \neq \mathbf{0}$,
- *positive semidefinite* if $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ for any n-dimensional vector $\mathbf{x}$,
- *negative definite* if $\mathbf{x}^T\mathbf{A}\mathbf{x} < 0$ for any n-dimensional vector $\mathbf{x} \neq \mathbf{0}$,
- *negative semidefinite* if $\mathbf{x}^T\mathbf{A}\mathbf{x} \leq 0$ for any n-dimensional vector $\mathbf{x}$,
- *indefinite* otherwise, i.e. $\mathbf{x}^T\mathbf{A}\mathbf{x}$ has positive and negative values for different $\mathbf{x}$.

Example 5. Let $\mathbf{I}$ be the $n \times n$ identity matrix. Then for any nonzero n-dimensional vector $\mathbf{x}$, we have

$$\mathbf{x}^T\mathbf{I}\mathbf{x} = \mathbf{x}^T\mathbf{x} = x_1^2 + x_2^2 + \cdots + x_n^2 > 0.$$

Hence, $\mathbf{I}$ is positive definite.

Example 6. The matrix $\mathbf{A} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ gives

$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (x_1 - x_2)^2 \geq 0.$$

for all $\mathbf{x} \in \mathbf{R}^2$. So $\mathbf{A}$ is positive semidefinite. Note that for any vector $\mathbf{x} = (x_1, x_1)^T$ we have $\mathbf{x}^T\mathbf{A}\mathbf{x} = 0$. Hence, $\mathbf{A}$ is not positive definite.

Definition: Assume that $\mathbf{A}$ is an $n \times n$ matrix. A nonzero n-dimensional vector $\mathbf{x}$ is called an *eigenvector* of $\mathbf{A}$ if it satisfies the equality $\mathbf{Ax} = \lambda\mathbf{x}$ for some scalar λ. The scalar λ is called an *eigenvalue* of $\mathbf{A}$.
The eigenvalues of $\mathbf{A}$ can be found by solving the characteristic equation

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0,$$

where "det" indicates the determinant.

Theorem 4. A symmetric matrix $\mathbf{A}$ is positive definite if and only if all its eigenvalues are positive.

Now we turn back to the optimality condition.

Theorem 5. (**Second-order necessary condition**)
If $\mathbf{x}^*$ is a local minimum for an unconstrained NLP problem $\min f(\mathbf{x})$, then

(i) $\nabla f(\mathbf{x}^*) = \mathbf{0}$, and

(ii) $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite.

Theorem 6. (**Second-order sufficient condition**)
If

(i) $\nabla f(\mathbf{x}^*) = \mathbf{0}$, and

(ii) $\nabla^2 f(\mathbf{x}^*)$ is positive definite,

then $\mathbf{x}^*$ is a local minimum for the unconstrained NLP problem $\min f(\mathbf{x})$.

Example 7. Consider the function $f(\mathbf{x}) = f(x_1, x_2) = x_1^2 - x_1x_2 + x_2^2 - 3x_2$ in Example 4. The function has a stationary point $\mathbf{x}^* = (1, 2)^T$. Its Hessian matrix is

$$\nabla^2 f(\mathbf{x}) = \nabla^2 f(x_1, x_2) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

Thus, we have $\nabla^2 f(\mathbf{x}^*) = \nabla^2 f(1, 2) = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$.
The eigenvalues of this matrix are found by solving the equation

$$\det(\nabla^2 f(1, 2) - \lambda \mathbf{I}) = \begin{vmatrix} 2 - \lambda & -1 \\ -1 & 2 - \lambda \end{vmatrix}$$

$$= (2 - \lambda)^2 - 1 = (3 - \lambda)(1 - \lambda) = 0.$$

$$\Rightarrow \quad \lambda_1 = 3, \quad \lambda_2 = 1.$$

Since both eigenvalues are positive, the Hessian matrix $\nabla^2 f(1, 2)$ is positive definite. Hence, by the second-order sufficient condition the point $\mathbf{x}^* = (1, 2)^T$ is a local minimum of $f(\mathbf{x})$.

4 Convexity Revisited

Recall that a function $f(\mathbf{x})$ is called *convex* if for any two points $\mathbf{x}_1$ and $\mathbf{x}_2$ in its domain and for any $\alpha \in [0, 1]$ we have

$$f\big(\alpha \mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2\big) \leq \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2).$$

And a function $f(\mathbf{x})$ is called *strictly convex* if for any $\mathbf{x}_1 \neq \mathbf{x}_2$ in its domain and for any $\alpha \in (0, 1)$ we have

$$f\big(\alpha \mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2\big) < \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2).$$

Theorem 7. Consider a function $f(\mathbf{x})$ defined in a convex domain. Then

(i) (Necessary condition for convexity) If $f(\mathbf{x})$ is convex, then $\nabla^2 f(\mathbf{x})$ is positive semidefinite everywhere in its domain.

(ii) (Sufficient condition for strict convexity) Function $f(\mathbf{x})$ is strictly convex if its Hessian matrix $\nabla^2 f(\mathbf{x})$ is positive definite for all $\mathbf{x}$ in its domain.

In Example 7, the Hessian matrix of function $f(\mathbf{x}) = x_1^2 - x_1 x_2 + x_2^2 - 3x_2$ has positive eigenvalues, which do not depend on $\mathbf{x}$, so it is positive definite everywhere. Hence, $f(\mathbf{x})$ is strictly convex. Furthermore, Theorem 3 implies that the local minimum $\mathbf{x}^* = (1, 2)^T$ is a global minimum and a unique global minimum of $f(\mathbf{x})$.

5 Gradient Methods

It is obvious that finding stationary point(s) is the very first step of the solving of unconstrained NLP problems. Finding a stationary point for some NLP problems could be easy, but it may not be the case in many other ones. In this section, we introduce an important class of solution procedures, which cope with unconstrained NLP problems by applying the aforementioned optimality conditions and approximating a stationary point of the nonlinear objective function, which is usually complicated. Once again, consider the unconstrained minimisation NLP problem

$$\min \ f(\mathbf{x}). \tag{1}$$

Let $\mathbf{x}_0$ be an initial approximation to the solution of (1). Assume that $\mathbf{x}_0$ is not a stationary point of $f(\mathbf{x})$. Then we choose an initial direction $\mathbf{d}_0$, and an arbitrary scalar value α. For a sufficient small value of α, Taylor's theorem gives

$$\begin{aligned} f(\mathbf{x}_0 + \alpha\mathbf{d}_0) &= f(\mathbf{x}_0) + (\nabla f(\mathbf{x}_0))^T(\alpha\mathbf{d}_0) + o(\alpha\|\mathbf{d}_0\|) \\ &= f(\mathbf{x}_0) + \alpha(\nabla f(\mathbf{x}_0))^T\mathbf{d}_0 + o(\alpha\|\mathbf{d}_0\|). \end{aligned}$$

If the direction $\mathbf{d}_0$ has been chosen in such a way that $(\nabla f(\mathbf{x}_0))^T\mathbf{d}_0 < 0$, then for a sufficiently small positive α (such that $o(\alpha\|\mathbf{d}_0\|) \approx 0$) we have

$$f(\mathbf{x}_0 + \alpha\mathbf{d}_0) < f(\mathbf{x}_0).$$

Finding the value α_0 such that

$$f(\mathbf{x}_0 + \alpha_0\mathbf{d}_0) = \min_{\alpha>0} f(\mathbf{x}_0 + \alpha\mathbf{d}_0)$$

gives the point $\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0\mathbf{d}_0$ to be a better approximation solution to (1). This procedure can be repeated until we find a "good enough" approximation solution to (1).

The basic idea of the gradient methods is that we choose a starting point $\mathbf{x}_0$ and then at each iteration $k \geq 1$

1. choose a direction $\mathbf{d}_k$ such that $(\nabla f(\mathbf{x}_k))^T\mathbf{d}_k < 0$, and
2. find the value α_k satisfying

$$f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) = \min_{\alpha>0} f(\mathbf{x}_k + \alpha\mathbf{d}_k), \text{ and}$$

3. take $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k$ as a better approximation solution to (1).

The value α_k is then called the *step size* of iteration k.
In many algorithms of this category, the direction $\mathbf{d}_k$ is chosen by taking a symmetric and positive definite matrix $\mathcal{D}_k$ and then calculating

$$\mathbf{d}_k = -\mathcal{D}_k \nabla f(\mathbf{x}_k).$$

This direction $\mathbf{d}_k$ satisfies the required condition since

$$(\nabla f(\mathbf{x}_k))^T \mathbf{d}_k = (\nabla f(\mathbf{x}_k))^T (-\mathcal{D}_k \nabla f(\mathbf{x}_k)) = -(\nabla f(\mathbf{x}_k))^T \mathcal{D}_k \nabla f(\mathbf{x}_k) < 0.$$

Now we introduce two algorithms which fall into the category of gradient methods.

5.1 Steepest descent method

At each iteration $k \geq 1$, the steepest descent method chooses $\mathcal{D}_k = \mathbf{I}$, the $n \times n$ identity matrix, which is positive definite. So the direction chosen is

$$\mathbf{d}_k = -\mathbf{I} \cdot \nabla f(\mathbf{x}_k) = -\nabla f(\mathbf{x}_k),$$

and this is a descent direction.[5] Actually it is the direction in which the function $f(\mathbf{x})$ decreases most rapidly when moving from $\mathbf{x}_k$.

Algorithm for Steepest Descent Method

Step 0. Choose a starting point $\mathbf{x}_0$, and a small positive scalar ϵ. Set $k = 0$.

[5]Notice that in order to simplify the presentation we do not set the "direction" as a normalised vector (unit vector).

Step 1. If $\|\nabla f(\mathbf{x}_k)\| < \epsilon$, then STOP: $\mathbf{x}_k$ is a satisfactory approximate minimum of $f(\mathbf{x})$.[6] Otherwise, set

$$\mathbf{d}_k = -\nabla f(\mathbf{x}_k).$$

Step 2. Choose the step size α_k by solving the one-dimensional problem

$$\min_{\alpha>0} g(\alpha) = \min_{\alpha>0} f(\mathbf{x}_k + \alpha \mathbf{d}_k).$$

Let $\alpha_k = \arg\min_{\alpha>0} g(\alpha)$, and then $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.
Set $k = k + 1$ and go to **Step 1**.

Example 8. Consider the unconstrained NLP problem

$$\min f(x_1, x_2) = x_1^2 - x_1 x_2 + x_2^2 - 3x_2.$$

Although its stationary point can be easily calculated, we demonstrate the steepest descent method to find an approximation solution with $\epsilon = 0.5$ and a starting point at the origin.

Solution. We have

$$\nabla f(x_1, x_2) = (2x_1 - x_2,\ -x_1 + 2x_2 - 3)^T.$$

Iteration 0.

Step 0. Let $\mathbf{x}_0 = (0, 0)^T$.
Step 1. $\nabla f(\mathbf{x}_0) = (0, -3)^T$, and $\|\nabla f(\mathbf{x}_0)\| = \sqrt{0^2 + (-3)^2} = 3 > 0.5 = \epsilon$.
Let $\mathbf{d}_0 = -\nabla f(\mathbf{x}_0) = (0,\ 3)^T$. Go to Step 2.

[6] The point $\mathbf{x}_k$ is a satisfactory approximate stationary point since $\nabla f(\mathbf{x}_k) \approx 0$.

Step 2. To find the the step size value α_0, we need to solve the problem

$$\min_{\alpha>0} g(\alpha) = \min_{\alpha>0} f(\mathbf{x}_0 + \alpha \mathbf{d}_0).$$

Then we have

$$\mathbf{x}_0 + \alpha \mathbf{d}_0 = (0,\, 0)^T + \alpha(0,\, 3)^T = (0,\, 3\alpha)^T$$

$$\Rightarrow \;\; g(\alpha) = f(\mathbf{x}_0 + \alpha \mathbf{d}_0) = f(0, 3\alpha) = 9\alpha^2 - 9\alpha$$

$$\frac{dg(\alpha)}{d\alpha} = 9(2\alpha - 1) = 0 \;\; \Rightarrow \;\; \alpha_0 = \frac{1}{2} > 0$$

Hence,

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 = (0, 0)^T + \frac{1}{2}(0, 3)^T = \Big(0,\, \frac{3}{2}\Big)^T.$$

Iteration 1.

Step 1. $\nabla f(\mathbf{x}_1) = (-\frac{3}{2},\, 0)^T$, and $\|\nabla f(\mathbf{x}_1)\| = \sqrt{(-\frac{3}{2})^2 + 0^2} = \frac{3}{2} > \epsilon = 0.5$.
So let $\mathbf{d}_1 = -\nabla f(\mathbf{x}_1) = \big(\frac{3}{2},\, 0\big)^T$ and go to Step 2.

Step 2. To find the the step size value α_1, we need to solve the problem

$$\min_{\alpha>0} g(\alpha) = \min_{\alpha>0} f(\mathbf{x}_1 + \alpha \mathbf{d}_1).$$

Then we have

$$\mathbf{x}_1 + \alpha \mathbf{d}_1 = \Big(0,\, \frac{3}{2}\Big)^T + \alpha\Big(\frac{3}{2},\, 0\Big)^T = \Big(\frac{3\alpha}{2},\, \frac{3}{2}\Big)^T$$

$$\Rightarrow \;\; g(\alpha) = f(\mathbf{x}_1 + \alpha \mathbf{d}_1) = f\Big(\frac{3\alpha}{2},\, \frac{3}{2}\Big) = \frac{9}{4}(\alpha^2 - \alpha - 1)$$

$$\frac{dg(\alpha)}{d\alpha} = \frac{9}{4}(2\alpha - 1) = 0 \;\Rightarrow\; \alpha_1 = \frac{1}{2} > 0$$

Hence,

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1 \mathbf{d}_1 = \Big(0,\, \frac{3}{2}\Big)^T + \frac{1}{2}\Big(\frac{3}{2},\, 0\Big)^T = \Big(\frac{3}{4},\, \frac{3}{2}\Big)^T.$$

Iteration 2.

Step 1. $\nabla f(\mathbf{x}_2) = \big(0, -\frac{3}{4}\big)^T$, and $\|\nabla f(\mathbf{x}_2)\| = \frac{3}{4} > \epsilon = 0.5$.
So let $\mathbf{d}_2 = -\nabla f(\mathbf{x}_2) = \big(0,\, \frac{3}{4}\big)^T$ and go to Step 2.

Step 2. To find the the step size value α_2, we need to solve the problem

$$\min_{\alpha>0} g(\alpha) = \min_{\alpha>0} f(\mathbf{x}_2 + \alpha \mathbf{d}_2).$$

Then we have

$$\mathbf{x}_2 + \alpha \mathbf{d}_2 = \Big(\frac{3}{4},\, \frac{3}{2}\Big)^T + \alpha\big(0,\, \frac{3}{4}\big)^T = \left(\frac{3}{4},\, \frac{3}{2} + \frac{3\alpha}{4}\right)^T$$

$$\Rightarrow\; g(\alpha) = f(\mathbf{x}_2 + \alpha \mathbf{d}_2) = f\Big(\frac{3}{4},\, \frac{3}{2} + \frac{3\alpha}{4}\Big) = \frac{9}{16}(\alpha^2 - \alpha - 5)$$

$$\frac{dg(\alpha)}{d\alpha} = \frac{9}{16}(2\alpha - 1) = 0 \;\Rightarrow\; \alpha_2 = \frac{1}{2} > 0$$

Hence,

$$\mathbf{x}_3 = \mathbf{x}_2 + \alpha_2 \mathbf{d}_2 = \Big(\frac{3}{4},\, \frac{3}{2}\Big)^T + \frac{1}{2}\Big(0,\, \frac{3}{4}\Big)^T = \Big(\frac{3}{4},\, \frac{15}{8}\Big).$$

Iteration 3.

Step 1. $\nabla f(\mathbf{x}_3) = (-\frac{3}{8},\, 0)^T$, and $\|\nabla f(\mathbf{x}_3)\| = \frac{3}{8} = 0.375 < \epsilon = 0.5$. Stop and declare that $\mathbf{x}_3 = \left(\frac{3}{4},\, \frac{15}{8}\right)$ is a satisfactory approximation solution.[7]

[7] Recall that the optimum to this minimisation NLP occurs at $\mathbf{x}^* = (1, 2)^T$.

Steepest descent method was invented in the nineteenth century by Cauchy. The advantages of this method is that it does not require

- the computation of second-order derivatives,
- solving a system of equations to compute the search direction, and
- storing matrices.

Its disadvantage is the slow rate of convergence. As a result, even though the cost per iteration is low, the overall cost of generating an approximation solution is high.

5.2 Newton's method

At each iteration k, Newton's method takes

$$\mathcal{D}_k = (\nabla^2 f(\mathbf{x}_k))^{-1}.$$

The idea of the method is that at each iteration instead of finding a minimum of $f(\mathbf{x})$ we find a minimum of the quadratic approximation of $f(\mathbf{x})$ around the current point $\mathbf{x}_k$

$$f(\mathbf{x}) \approx f(\mathbf{x}_k)+(\nabla f(\mathbf{x}_k))^T(\mathbf{x}-\mathbf{x}_k)+\frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^T\nabla^2 f(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) = g(\mathbf{x}).$$

The minimum of $g(\mathbf{x})$ occurs when the gradient of this quadratic is zero, i.e. the next approximation can be taken as the solution of the vector equation

$$\nabla g(\mathbf{x}) = \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) = \mathbf{0}$$

Solving this vector equation and setting the solution $\mathbf{x}$ to $\mathbf{x}_{k+1}$ gives

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k))^{-1} \nabla f(\mathbf{x}_k),$$

which is equivalent to setting

$$\alpha_k = 1 \quad \text{and} \quad \mathcal{D}_k = (\nabla^2 f(\mathbf{x}_k))^{-1}.$$

Algorithm for Newton's Method

Step 0. Choose a starting point $\mathbf{x}_0$ and a small positive scalar ϵ. Let $k = 0$.

Step 1. If $\|\nabla f(\mathbf{x}_k)\| < \epsilon$, then STOP: $\mathbf{x}_k$ is a satisfactory approximate minimum of $f(\mathbf{x})$. Otherwise, let

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k))^{-1} \nabla f(\mathbf{x}_k).$$

Step 2. Set the $k = k + 1$ and go to **Step 1**.

Note that Newton's method selects a step size $\alpha_k = 1$ at each iteration k.

Example 9. Consider again the unconstrained NLP problem

$$\min f(x_1, x_2) = x_1^2 - x_1 x_2 + x_2^2 - 3x_2$$

We demonstrate Newton's method to find an approximation solution with $\epsilon = 0.5$ and a starting point at the origin.

Solution. We have

$$\nabla f(x_1, x_2) = (2x_1 - x_2,\ -x_1 + 2x_2 - 3)^T.$$

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

Iteration 0.

Step 0. Let $\mathbf{x}_0 = (0, 0)^T$ and $k = 0$.

Step 1.

$$\nabla f(\mathbf{x}_0) = (0, -3)^T \quad \Rightarrow \quad \|\nabla f(\mathbf{x}_0)\| = \sqrt{0^2 + (-3)^2} = 3 > \epsilon = 0.5$$

$$\nabla^2 f(\mathbf{x}_0) = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \quad \Rightarrow \quad (\nabla^2 f(\mathbf{x}_0))^{-1} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}$$

$$(\nabla^2 f(\mathbf{x}_0))^{-1} \nabla f(\mathbf{x}_0) = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} 0 \\ -3 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$$

Thus,

$$\mathbf{x}_1 = \mathbf{x}_0 - (\nabla^2 f(\mathbf{x}_0))^{-1} \nabla f(\mathbf{x}_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Step 2. Set $k = 0 + 1 = 1$.

Iteration 1.

Step 1.

$$\nabla f(\mathbf{x}_1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \Rightarrow \quad \|\nabla f(\mathbf{x}_1)\| = 0 < \epsilon = 0.5$$

Stop and declare that a satisfactory approximation solution is found.

Newton's method usually converges much more rapidly than the steepest descent method, but sometimes it could fail. Besides,

the calculation of the matrix inverse $(\nabla^2 f(\mathbf{x}))^{-1}$ is hard work. Hence, there have been several modified methods, which will be introduced in the advanced subject "Nonlinear Methods in Quantitative Management".

Further reading: Section 11.1–11.7 in the reference book "Operations Research: Applications and Algorithms" (Winston, 2004)