Xin-She Yang

Introduction to

Algorithms for Data Mining and Machine Learning

Introduction to Algorithms for Data Mining and Machine Learning

Xin-She Yang
Middlesex University
School of Science and Technology
London, United Kingdom

ACADEMIC PRESS
An imprint of Elsevier

Academic Press is an imprint of Elsevier
125 London Wall, London EC2Y 5AS, United Kingdom
525 B Street, Suite 1650, San Diego, CA 92101, United States
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

ISBN: 978-0-12-817216-2

Publisher: Candice Janco
Acquisition Editor: J. Scott Bentley
Editorial Project Manager: Michael Lutz
Production Project Manager: Nilesh Kumar Shah
Designer: Miles Hitchen

Typeset by VTeX

Contents

Introduction to optimization

1

Contents

This book introduces the most fundamentals and algorithms related to optimization, data mining, and machine learning. The main requirement is some understanding of high-school mathematics and basic calculus; however, we will review and introduce some of the mathematical foundations in the first two chapters.

1.1 Algorithms

An algorithm is an iterative, step-by-step procedure for computation. The detailed procedure can be a simple description, an equation, or a series of descriptions in combination with equations. Finding the roots of a polynomial, checking if a natural number is a prime number, and generating random numbers are all algorithms.

1.1.1 Essence of an algorithm

In essence, an algorithm can be written as an iterative equation or a set of iterative equations. For example, to find a square root of $a > 0$, we can use the following iterative equation:

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right), \tag{1.1}$$

where k is the iteration counter ($k = 0, 1, 2, \ldots$) starting with a random guess $x_0 = 1$.

Introduction to Algorithms for Data Mining and Machine Learning. https://doi.org/10.1016/B978-0-12-817216-2.00008-9

Example 1

As an example, if $x_0 = 1$ and $a = 4$, then we have

$$x_1 = \frac{1}{2}(1 + \frac{4}{1}) = 2.5. \tag{1.2}$$

Similarly, we have

$$x_2 = \frac{1}{2}(2.5 + \frac{4}{2.5}) = 2.05, \quad x_3 = \frac{1}{2}(2.05 + \frac{4}{2.05}) \approx 2.0061, \tag{1.3}$$

$$x_4 \approx 2.00000927, \tag{1.4}$$

which is very close to the true value of $\sqrt{4} = 2$. The accuracy of this iterative formula or algorithm is high because it achieves the accuracy of five decimal places after four iterations.

The convergence is very quick if we start from different initial values such as $x_0 = 10$ and even $x_0 = 100$. However, for an obvious reason, we cannot start with $x_0 = 0$ due to division by zero.

Find the root of $x = \sqrt{a}$ is equivalent to solving the equation

$$f(x) = x^2 - a = 0, \tag{1.5}$$

which is again equivalent to finding the roots of a polynomial $f(x)$. We know that Newton's root-finding algorithm can be written as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \tag{1.6}$$

where $f'(x)$ is the first derivative or gradient of $f(x)$. In this case, we have $f'(x) = 2x$. Thus, Newton's formula becomes

$$x_{k+1} = x_k - \frac{(x_k^2 - a)}{2x_k}, \tag{1.7}$$

which can be written as

$$x_{k+1} = (x_k - \frac{x_k}{2}) + \frac{a}{2x_k} = \frac{1}{2}\big(x_k + \frac{a}{x_k}\big). \tag{1.8}$$

This is exactly what we have in Eq. (1.1).

Newton's method has rigorous mathematical foundations, which has a guaranteed convergence under certain conditions. However, in general, Eq. (1.6) is more general, and the gradient information $f'(x)$ is needed. In addition, for the formula to be valid, we must have $f'(x) \neq 0$.

1.1.2 Issues with algorithms

The advantage of the algorithm given in Eq. (1.1) is that it converges very quickly. However, careful readers may have asked: we know that $\sqrt{4} = \pm 2$, how can we find the other root -2 in addition to $+2$?

Even if we use different initial value $x_0 = 10$ or $x_0 = 0.5$, we can only reach $x_* = 2$, not -2.

What happens if we start with $x_0 < 0$? From $x_0 = -1$, we have

$$x_1 = \frac{1}{2}(-1 + \frac{4}{-1}) = -2.5, \quad \boldsymbol{x}_2 = \frac{1}{2}(-2.5 + \frac{4}{-2.5}) = -2.05, \tag{1.9}$$

$$x_3 \approx -2.0061, \quad x_4 \approx -2.00000927, \tag{1.10}$$

which is approaching -2 very quickly. If we start from $x_0 = -10$ or $x_0 = -0.5$, then we can always get $x_* = -2$, not $+2$.

This highlights a key issue here: the final solution seems to depend on the initial starting point for this algorithm, which is true for many algorithms.

Now the relevant question is: how do we know where to start to get a particular solution? The general short answer is "we do not know". Thus, some knowledge of the problem under consideration or an educated guess may be useful to find the final solution.

In fact, most algorithms may depend on the initial configuration, and such algorithms are often carrying out search moves locally. Thus, this type of algorithm is often referred to as local search. A good algorithm should be able to "forget" its initial configuration though such algorithms may not exist at all for most types of problems.

What we need in general is the global search, which attempts to find final solutions that are less sensitive to the initial starting point(s).

Another important issue in our discussions is that the gradient information $f'(x)$ is necessary for some algorithms such as Newton's method given in Eq. (1.6). This poses certain requirements on the smoothness of the function $f(x)$. For example, we know that $|x|$ is not differentiable at $x = 0$. Thus, we cannot directly use Newton's method to find the roots of $f(x) = |x|x^2 - a = 0$ for $a > 0$. Some modifications are needed.

There are other issues related to algorithms such as the setting of parameters, the slow rate of convergence, condition numbers, and iteration structures. All these make algorithm designs and usage somehow challenging, and we will discuss these issues in more detail later in this book.

1.1.3 Types of algorithms

An algorithm can only do a specific computation task (at most a class of computational tasks), and no algorithms can do all the tasks. Thus, algorithms can be classified due to their purposes. An algorithm to find roots of a polynomial belongs to root-finding algorithms, whereas an algorithm for ranking a set of numbers belongs to sorting algorithms. There are many classes of algorithms for different purposes. Even for the same purpose such as sorting, there are many different algorithms such as the merge sort, bubble sort, quicksort, and others.

We can also categorize algorithms in terms of their characteristics. The root-finding algorithms we just introduced are deterministic algorithms because the final solutions are exactly the same if we start from the same initial guess. We obtain the same set of solutions every time we run the algorithm. On the other hand, we may introduce some randomization into the algorithm, for example, using purely random initial points. Every time we run the algorithm, we use a new random initial guess. In this case, the algorithm can have some nondeterministic nature, and such algorithms are referred to as stochastic. Sometimes, using randomness may be advantageous. For example, in the example of $\sqrt{4} = \pm 2$ using Eq. (1.1), random initial values (both positive and negative) can allow the algorithm to find both roots. In fact, a major trend in the modern metaheuristics is using some randomization to suit different purposes.

For algorithms to be introduced in this book, we are mainly concerned with algorithms for data mining, optimization, and machine learning. We use a relatively unified approach to link algorithms in data mining and machine learning to algorithms for optimization.

1.2 Optimization

Optimization is everywhere, from engineering design to business planning. After all, time and resources are limited, and optimal use of such valuable resources is crucial. In addition, designs of products have to maximize the performance, sustainability, and energy efficiency and to minimize the costs. Therefore, optimization is important for many applications.

1.2.1 A simple example

Let us start with a very simple example to design a container with volume capacity $V_0 = 10$ m^3. As the main cost is related to the cost of materials, the main aim is to minimize the total surface area S.

The first thing we have to decide is the shape of the container (cylinder, cubic, sphere or ellipsoid, or more complex geometry). For simplicity, let us start with a cylindrical shape with radius r and height h (see Fig. 1.1).

The total surface area of a cylinder is

$$S = 2(\pi r^2) + 2\pi r h, \tag{1.11}$$

and the volume is

$$V = \pi r^2 h. \tag{1.12}$$

There are only two design variables r and h and one objective function S to be minimized. Obviously, if there is no capacity constraint, then we can choose not to build the container, and then the cost of materials is zero for $r = 0$ and $h = 0$. However,

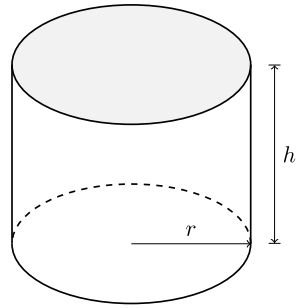

Figure 1.1 Design of a cylindric container.

the constraint requirement means that we have to build a container with fixed volume $V_0 = \pi r^2 h = 10\ \mathrm{m}^3$. Therefore, this optimization problem can be written as

$$\text{minimize} \quad S = 2\pi r^2 + 2\pi r h, \tag{1.13}$$

subject to the equality constraint

$$\pi r^2 h = V_0 = 10. \tag{1.14}$$

To solve this problem, we can first try to use the equality constraint to reduce the number of design variables by solving h. So we have

$$h = \frac{V_0}{\pi r^2}. \tag{1.15}$$

Substituting it into (1.13), we get

$$\begin{aligned} S &= 2\pi r^2 + 2\pi r h \\ &= 2\pi r^2 + 2\pi r \frac{V_0}{\pi r^2} = 2\pi r^2 + \frac{2V_0}{r}. \end{aligned} \tag{1.16}$$

This is a univariate function. From basic calculus we know that the minimum or maximum can occur at the stationary point, where the first derivative is zero, that is,

$$\frac{\mathrm{d}S}{\mathrm{d}r} = 4\pi r - \frac{2V_0}{r^2} = 0, \tag{1.17}$$

which gives

$$r^3 = \frac{V_0}{2\pi}, \quad \text{or} \quad r = \sqrt[3]{\frac{V_0}{2\pi}}. \tag{1.18}$$

Thus, the height is

$$\frac{h}{r} = \frac{V_0/(\pi r^2)}{r} = \frac{V_0}{\pi r^3} = 2. \tag{1.19}$$

This means that the height is twice the radius: $h = 2r$. Thus, the minimum surface is

$$S_* = 2\pi r^2 + 2\pi rh = 2\pi r^2 + 2\pi r(2r) = 6\pi r^2$$
$$= 6\pi\Big(\frac{V_0}{2\pi}\Big)^{2/3} = \frac{6\pi}{\sqrt[3]{4\pi^2}} V_0^{2/3}. \tag{1.20}$$

For $V_0 = 10$, we have

$$r = \sqrt[3]{\frac{V_0}{(2\pi)}} = \sqrt[3]{\frac{10}{2\pi}} \approx 1.1675, \quad h = 2r = 2.335,$$

and the total surface area

$$S_* = 2\pi r^2 + 2\pi rh \approx 25.69.$$

It is worth pointing out that this optimal solution is based on the assumption or requirement to design a cylindrical container. If we decide to use a sphere with radius R, we know that its volume and surface area is

$$V_0 = \frac{4\pi}{3} R^3, \quad S = 4\pi R^2. \tag{1.21}$$

We can solve R directly

$$R^3 = \frac{3V_0}{4\pi}, \quad \text{or } R = \sqrt[3]{\frac{3V_0}{4\pi}}, \tag{1.22}$$

which gives the surface area

$$S = 4\pi\Big(\frac{3V_0}{4\pi}\Big)^{2/3} = \frac{4\pi\sqrt[3]{9}}{\sqrt[3]{16\pi^2}} V_0^{2/3}. \tag{1.23}$$

Since $6\pi/\sqrt[3]{4\pi^2} \approx 5.5358$ and $4\pi\sqrt[3]{9}/\sqrt[3]{16\pi^2} \approx 4.83598$, we have $S < S_*$, that is, the surface area of a sphere is smaller than the minimum surface area of a cylinder with the same volume. In fact, for the same $V_0 = 10$, we have

$$S(\text{sphere}) = \frac{4\pi\sqrt[3]{9}}{\sqrt[3]{16\pi^2}} V_0^{2/3} \approx 22.47, \tag{1.24}$$

which is smaller than $S_* = 25.69$ for a cylinder.

This highlights the importance of the choice of design type (here in terms of shape) before we can do any truly useful optimization. Obviously, there are many other factors that can influence the choice of design, including the manufacturability of the design, stability of the structure, ease of installation, space availability, and so on. For a container, in most applications, a cylinder may be much easier to produce than a sphere, and thus the overall cost may be lower in practice. Though there are so many factors to be considered in engineering design, for the purpose of optimization, here we will only focus on the improvement and optimization of a design with well-posed mathematical formulations.

1.2.2 General formulation of optimization

Whatever the real-world applications may be, it is usually possible to formulate an optimization problem in a generic form [49,53,160]. All optimization problems with explicit objectives can in general be expressed as a nonlinearly constrained optimization problem

$$\begin{aligned} &\text{maximize/minimize} \quad f(\boldsymbol{x}), \;\; \boldsymbol{x} = (x_1, x_2, \ldots, x_D)^T \in \mathbb{R}^D, \\ &\text{subject to } \phi_j(\boldsymbol{x}) = 0 \;\; (j = 1, 2, \ldots, M), \\ &\qquad\qquad \psi_k(\boldsymbol{x}) \leq 0 \;\; (k = 1, \ldots, N), \end{aligned} \tag{1.25}$$

where $f(\boldsymbol{x})$, $\phi_j(\boldsymbol{x})$, and $\psi_k(\boldsymbol{x})$ are scalar functions of the design vector $\boldsymbol{x}$. Here the components x_i of $\boldsymbol{x} = (x_1, \ldots, x_D)^T$ are called design or decision variables, and they can be either continuous, discrete, or a mixture of these two. The vector $\boldsymbol{x}$ is often called the decision vector, which varies in a D-dimensional space $\mathbb{R}^D$.

It is worth pointing out that we use a column vector here for $\boldsymbol{x}$ (thus with transpose T). We can also use a row vector $\boldsymbol{x} = (x_1, \ldots, x_D)$ and the results will be the same. Different textbooks may use slightly different formulations. Once we are aware of such minor variations, it should cause no difficulty or confusion.

In addition, the function $f(\boldsymbol{x})$ is called the objective function or cost function, $\phi_j(\boldsymbol{x})$ are constraints in terms of M equalities, and $\psi_k(\boldsymbol{x})$ are constraints written as N inequalities. So there are $M + N$ constraints in total. The optimization problem formulated here is a nonlinear constrained problem. Here the inequalities $\psi_k(\boldsymbol{x}) \leq 0$ are written as "less than", and they can also be written as "greater than" via a simple transformation by multiplying both sides by -1.

The space spanned by the decision variables is called the search space $\mathbb{R}^D$, whereas the space formed by the values of the objective function is called the objective or response space, and sometimes the landscape. The optimization problem essentially maps the domain $\mathbb{R}^D$ or the space of decision variables into the solution space $\mathbb{R}$ (or the real axis in general).

The objective function $f(\boldsymbol{x})$ can be either linear or nonlinear. If the constraints ϕ_j and ψ_k are all linear, it becomes a linearly constrained problem. Furthermore, when ϕ_j, ψ_k, and the objective function $f(\boldsymbol{x})$ are all linear, then it becomes a linear programming problem [35]. If the objective is at most quadratic with linear constraints, then it is called a quadratic programming problem. If all the values of the decision variables can be only integers, then this type of linear programming is called integer programming or integer linear programming.

On the other hand, if no constraints are specified and thus x_i can take any values in the real axis (or any integers), then the optimization problem is referred to as an unconstrained optimization problem.

As a very simple example of optimization problems without any constraints, we discuss the search of the maxima or minima of a univariate function.

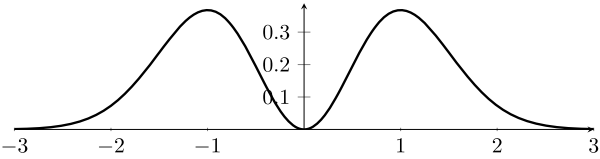

Figure 1.2 A simple multimodal function $f(x) = x^2 \mathrm{e}^{-x^2}$.

Example 2

For example, to find the maximum of a univariate function $f(x)$

$$f(x) = x^2 e^{-x^2}, \qquad -\infty < x < \infty, \tag{1.26}$$

is a simple unconstrained problem, whereas the following problem is a simple constrained minimization problem:

$$f(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2, \qquad (x_1, x_2) \in \mathbb{R}^2, \tag{1.27}$$

subject to

$$x_1 \geq 1, \qquad x_2 - 2 = 0. \tag{1.28}$$

It is worth pointing out that the objectives are explicitly known in all the optimization problems to be discussed in this book. However, in reality, it is often difficult to quantify what we want to achieve, but we still try to optimize certain things such as the degree of enjoyment or service quality on holiday. In other cases, it may be impossible to write the objective function in any explicit form mathematically.

From basic calculus we know that, for a given curve described by $f(x)$, its gradient $f'(x)$ describes the rate of change. When $f'(x) = 0$, the curve has a horizontal tangent at that particular point. This means that it becomes a point of special interest. In fact, the maximum or minimum of a curve occurs at

$$f'(x_*) = 0, \tag{1.29}$$

which is a critical condition or stationary condition. The solution x_* to this equation corresponds to a stationary point, and there may be multiple stationary points for a given curve.

To see if it is a maximum or minimum at $x = x_*$, we have to use the information of its second derivative $f''(x)$. In fact, $f''(x_*) > 0$ corresponds to a minimum, whereas $f''(x_*) < 0$ corresponds to a maximum. Let us see a concrete example.

Example 3

To find the minimum of $f(x) = x^2 \mathrm{e}^{-x^2}$ (see Fig. 1.2), we have the stationary condition $f'(x) = 0$ or

$$f'(x) = 2x \times \mathrm{e}^{-x^2} + x^2 \times (-2x)\mathrm{e}^{-x^2} = 2(x - x^3)\mathrm{e}^{-x^2} = 0.$$

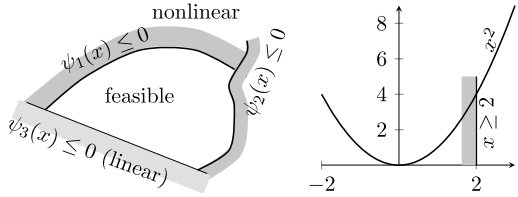

Figure 1.3 (a) Feasible domain with nonlinear inequality constraints $\psi_1(x)$ and $\psi_2(x)$ (left) and linear inequality constraint $\psi_3(x)$. (b) An example with an objective of $f(x) = x^2$ subject to $x \ge 2$ (right).

As $e^{-x^2} > 0$, we have

$$x(1 - x^2) = 0, \quad \text{or} \quad x = 0 \ \text{ and } \ x = \pm 1.$$

The second derivative is given by

$$f''(x) = 2e^{-x^2}(1 - 5x^2 + 2x^4),$$

which is an even function with respect to x.

So at $x = \pm 1$, $f''(\pm 1) = 2[1 - 5(\pm 1)^2 + 2(\pm 1)^4]e^{-(\pm 1)^2} = -4e^{-1} < 0$. Thus, there are two maxima that occur at $x_* = \pm 1$ with $f_{\max} = e^{-1}$. At $x = 0$, we have $f''(0) = 2 > 0$, thus the minimum of $f(x)$ occurs at $x_* = 0$ with $f_{\min}(0) = 0$.

Whatever the objective is, we have to evaluate it many times. In most cases, the evaluations of the objective functions consume a substantial amount of computational power (which costs money) and design time. Any efficient algorithm that can reduce the number of objective evaluations saves both time and money.

In mathematical programming, there are many important concepts, and we will first introduce a few related concepts: feasible solutions, optimality criteria, the strong local optimum, and weak local optimum.

1.2.3 Feasible solution

A point $\boldsymbol{x}$ that satisfies all the constraints is called a feasible point and thus is a feasible solution to the problem. The set of all feasible points is called the feasible region (see Fig. 1.3).

For example, we know that the domain $f(x) = x^2$ consists of all real numbers. If we want to minimize $f(x)$ without any constraint, all solutions such as $x = -1$, $x = 1$, and $x = 0$ are feasible. In fact, the feasible region is the whole real axis. Obviously, $x = 0$ corresponds to $f(0) = 0$ as the true minimum.

However, if we want to find the minimum of $f(x) = x^2$ subject to $x \ge 2$, then it becomes a constrained optimization problem. The points such as $x = 1$ and $x = 0$ are no longer feasible because they do not satisfy $x \ge 2$. In this case the feasible solutions are all the points that satisfy $x \ge 2$. So $x = 2$, $x = 100$, and $x = 10^8$ are all feasible. It is obvious that the minimum occurs at $x = 2$ with $f(2) = 2^2 = 4$, that is, the optimal solution for this problem occurs at the boundary point $x = 2$ (see Fig. 1.3).

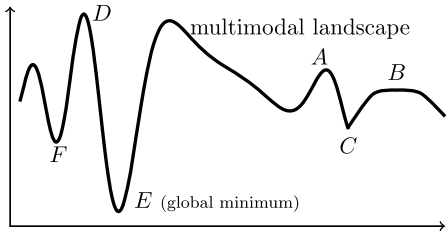

Figure 1.4 Local optima, weak optima, and global optimality.

1.2.4 Optimality criteria

A point $\boldsymbol{x}_*$ is called a strong local maximum of the nonlinearly constrained optimization problem if $f(\boldsymbol{x})$ is defined in a δ-neighborhood $N(\boldsymbol{x}_*, \delta)$ and satisfies $f(\boldsymbol{x}_*) > f(\boldsymbol{u})$ for $\boldsymbol{u} \in N(\boldsymbol{x}_*, \delta)$, where $\delta > 0$ and $\boldsymbol{u} \neq \boldsymbol{x}_*$. If $\boldsymbol{x}_*$ is not a strong local maximum, then the inclusion of equality in the condition $f(\boldsymbol{x}_*) \geq f(\boldsymbol{u})$ for all $\boldsymbol{u} \in N(\boldsymbol{x}_*, \delta)$ defines the point $\boldsymbol{x}_*$ as a weak local maximum (see Fig. 1.4). The local minima can be defined in a similar manner when $>$ and $\geq$ are replaced by $<$ and $\leq$, respectively.

Fig. 1.4 shows various local maxima and minima. Point A is a strong local maximum, whereas point B is a weak local maximum because there are many (in fact, infinite) different values of $\boldsymbol{x}$ that will lead to the same value of $f(\boldsymbol{x}_*)$. Point D is the global maximum, and point E is the global minimum. In addition, point F is a strong local minimum. However, point C is a strong local minimum, but it has a discontinuity in $f'(\boldsymbol{x}_*)$. So the stationary condition for this point $f'(\boldsymbol{x}_*) = 0$ is not valid. We will not deal with these types of minima or maxima in detail.

As we briefly mentioned before, for a smooth curve $f(x)$, optimal solutions usually occur at stationary points where $f'(x) = 0$. This is not always the case because optimal solutions can also occur at the boundary, as we have seen in the previous example of minimizing $f(x) = x^2$ subject to $x \geq 2$. In our present discussion, we will assume that both $f(\boldsymbol{x})$ and $f'(\boldsymbol{x})$ are always continuous or $f(\boldsymbol{x})$ is everywhere twice continuously differentiable. Obviously, the information of $f'(x)$ is not sufficient to determine whether a stationary point is a local maximum or minimum. Thus, higher-order derivatives such as $f''(x)$ are needed, but we do not make any assumption at this stage. We will further discuss this in detail in the next section.

1.3 Unconstrained optimization

Optimization problems can be classified as either unconstrained or constrained. Unconstrained optimization problems can in turn be subdivided into univariate and multivariate problems.

1.3.1 Univariate functions

The simplest optimization problem without any constraints is probably the search for the maxima or minima of a univariate function $f(x)$. For unconstrained optimization problems, the optimality occurs at the critical points given by the stationary condition $f'(x) = 0$.

However, this stationary condition is just a necessary condition, but it is not a sufficient condition. If $f'(x_*) = 0$ and $f''(x_*) > 0$, it is a local minimum. Conversely, if $f'(x_*) = 0$ and $f''(x_*) < 0$, then it is a local maximum. However, if $f'(x_*) = 0$ and $f''(x_*) = 0$, care should be taken because $f''(x)$ may be indefinite (both positive and negative) when $x \to x_*$, then x_* corresponds to a saddle point.

For example, for $f(x) = x^3$, we have

$$f'(x) = 3x^2, \quad f''(x) = 6x. \tag{1.30}$$

The stationary condition $f'(x) = 3x^2 = 0$ gives $x_* = 0$. However, we also have

$$f''(x_*) = f''(0) = 0.$$

In fact, $f(x) = x^3$ has a saddle point $x_* = 0$ because $f'(0) = 0$ but f'' changes sign from $f''(0+) > 0$ to $f''(0-) < 0$ as x moves from positive to negative.

Example 4

For example, to find the maximum or minimum of a univariate function

$$f(x) = 3x^4 - 4x^3 - 12x^2 + 9, \qquad -\infty < x < \infty,$$

we first have to find its stationary points x_* when the first derivative $f'(x)$ is zero, that is,

$$f'(x) = 12x^3 - 12x^2 - 24x = 12(x^3 - x^2 - 2x) = 0.$$

Since $f'(x) = 12(x^3 - x^2 - 2x) = 12x(x+1)(x-2) = 0$, we have

$$x_* = -1, \quad x_* = 2, \quad x_* = 0.$$

The second derivative of $f(x)$ is simply

$$f''(x) = 36x^2 - 24x - 24.$$

From the basic calculus we know that the maximum requires $f''(x_*) \le 0$ whereas the minimum requires $f''(x_*) \ge 0$.

At $x_* = -1$, we have

$$f''(-1) = 36(-1)^2 - 24(-1) - 24 = 36 > 0,$$

so this point corresponds to a local minimum

$$f(-1) = 3(-1)^4 - 4(-1)^3 - 12(-1)^2 + 9 = 4.$$

Similarly, at $x_* = 2$, $f''(x_*) = 72 > 0$, and thus we have another local minimum

$$f(x_*) = -23.$$

However, at $x_* = 0$, we have $f'(0) = -24 < 0$, which corresponds to a local maximum $f(0) = 9$. However, this maximum is not a global maximum because the global maxima for $f(x)$ occur at $x = \pm\infty$.

The global minimum occurs at $x_* = 2$ with $f(2) = -23$.

The maximization of a function $f(x)$ can be converted into the minimization of $A - f(x)$, where A is usually a large positive number (though $A = 0$ will do). For example, we know the maximum of $f(x) = e^{-x^2}$, $x \in (-\infty, \infty)$, is 1 at $x_* = 0$. This problem can be converted to the minimization of $-f(x)$. For this reason, the optimization problems can be expressed as either minimization or maximization depending on the context and convenience of formulations.

In fact, in the optimization literature, some books formulate all the optimization problems in terms of maximization, whereas others write these problems in terms of minimization, though they are in essence dealing with the same problems.

1.3.2 Multivariate functions

We can extend the optimization procedure for univariate functions to multivariate functions using partial derivatives and relevant conditions. Let us start with the example

$$\text{minimize} \quad f(x, y) = x^2 + y^2, \quad x, y \in \mathbb{R}. \tag{1.31}$$

It is obvious that $x = 0$ and $y = 0$ is a minimum solution because $f(0, 0) = 0$. The question is how to solve this problem formally. We can extend the stationary condition to partial derivatives, and we have $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$. In this case, we have

$$\frac{\partial f}{\partial x} = 2x + 0 = 0, \quad \frac{\partial f}{\partial y} = 0 + 2y = 0. \tag{1.32}$$

The solution is obviously $x_* = 0$ and $y_* = 0$.

Now how do we know that it corresponds to a maximum or minimum? If we try to use the second derivatives, we have four different partial derivatives such as f_{xx} and f_{yy}, and which one should we use? In fact, we need to define the Hessian matrix from these second partial derivatives, and we have

$$\boldsymbol{H} = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}. \tag{1.33}$$

Since

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}, \tag{1.34}$$

we can conclude that the Hessian matrix is always symmetric. In the case of $f(x, y) = x^2 + y^2$, it is easy to check that the Hessian matrix is

$$\boldsymbol{H} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}. \tag{1.35}$$

Mathematically speaking, if $\boldsymbol{H}$ is positive definite, then the stationary point (x_*, y_*) corresponds to a local minimum. Similarly, if $\boldsymbol{H}$ is negative definite, then the stationary point corresponds to a maximum. The definiteness of a symmetric matrix is controlled by its eigenvalues. For this simple diagonal matrix $\boldsymbol{H}$, its eigenvalues are its two diagonal entries 2 and 2. As both eigenvalues are positive, this matrix is positive definite. Since the Hessian matrix here does not involve any x or y, it is always positive definite in the whole search domain $(x, y) \in \mathbb{R}^2$, so we can conclude that the solution at point $(0, 0)$ is the global minimum.

Obviously, this is a particular case. In general, the Hessian matrix depends on the independent variables, but the definiteness test conditions still apply. That is, positive definiteness of a stationary point means a local minimum. Alternatively, for bivariate functions, we can define the determinant of the Hessian matrix in Eq. (1.33) as

$$\Delta = \det(\boldsymbol{H}) = f_{xx} f_{yy} - (f_{xy})^2. \tag{1.36}$$

At the stationary point (x_*, y_*), if $\Delta > 0$ and $f_{xx} > 0$, then (x_*, y_*) is a local minimum. If $\Delta > 0$ but $f_{xx} < 0$, then it is a local maximum. If $\Delta = 0$, then it is inconclusive, and we have to use other information such as higher-order derivatives. However, if $\Delta < 0$, then it is a saddle point. A saddle point is a special point where a local minimum occurs along one direction, whereas the maximum occurs along another (orthogonal) direction.

Example 5

To minimize $f(x, y) = (x - 1)^2 + x^2 y^2$, we have

$$\frac{\partial f}{\partial x} = 2(x - 1) + 2xy^2 = 0, \quad \frac{\partial f}{\partial y} = 0 + 2x^2 y = 0. \tag{1.37}$$

The second condition gives $y = 0$ or $x = 0$. Substituting $y = 0$ into the first condition, we have $x = 1$. However, $x = 0$ does not satisfy the first condition. Therefore, we have a solution $x_* = 1$ and $y_* = 0$.

For our example with $f = (x - 1)^2 + x^2 y^2$, we have

$$\frac{\partial^2 f}{\partial x^2} = 2y^2 + 2, \quad \frac{\partial^2 f}{\partial x \partial y} = 4xy, \quad \frac{\partial^2 f}{\partial y \partial x} = 4xy, \quad \frac{\partial^2 f}{\partial y^2} = 2x^2, \tag{1.38}$$

and thus we have

$$\boldsymbol{H} = \begin{pmatrix} 2y^2 + 2 & 4xy \\ 4xy & 2x^2 \end{pmatrix}. \tag{1.39}$$

At the stationary point $(x_*, y_*) = (1, 0)$, the Hessian matrix becomes

$$\boldsymbol{H} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

which is positive definite because its double eigenvalues 2 are positive. Alternatively, we have $\Delta = 4 > 0$ and $f_{xx} = 2 > 0$. Therefore, $(1, 0)$ is a local minimum.

In fact, for a multivariate function $f(x_1, x_2, \ldots, x_n)$ in an n-dimensional space, the stationary condition can be extended to

$$\boldsymbol{G} = \nabla f = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n})^T = 0, \tag{1.40}$$

where $\boldsymbol{G}$ is called the gradient vector. The second derivative test becomes the definiteness of the Hessian matrix

$$\boldsymbol{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{pmatrix}. \tag{1.41}$$

At the stationary point defined by $\boldsymbol{G} = \nabla f = 0$, the positive definiteness of $\boldsymbol{H}$ gives a local minimum, whereas the negative definiteness corresponds to a local maximum. In essence, the eigenvalues of the Hessian matrix $\boldsymbol{H}$ determine the local behavior of the function. As we mentioned before, if $\boldsymbol{H}$ is positive semidefinite, then it corresponds to a local minimum.

1.4 Nonlinear constrained optimization

As most real-world problems are nonlinear, nonlinear mathematical programming forms an important part of mathematical optimization methods. A broad class of nonlinear programming problems is about the minimization or maximization of $f(\boldsymbol{x})$ subject to no constraints, and another important class is the minimization of a quadratic objective function subject to nonlinear constraints. There are many other nonlinear programming problems as well.

Nonlinear programming problems are often classified according to the convexity of the defining functions. An interesting property of a convex function f is that the

vanishing of the gradient $\nabla f(\boldsymbol{x}_*) = 0$ guarantees that the point x_* is a global minimum or maximum of f. We will introduce the concept of convexity in the next chapter. If a function is not convex or concave, then it is much more difficult to find its global minima or maxima.

1.4.1 Penalty method

For the simple function optimization with equality and inequality constraints, a common method is the penalty method. For the optimization problem

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}), \ \ \boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n, \\ &\text{subject to } \phi_i(\boldsymbol{x}) = 0, \ (i = 1, \ldots, M), \quad \psi_j(\boldsymbol{x}) \le 0, \ \ (j = 1, \ldots, N), \end{aligned} \tag{1.42}$$

the idea is to define a penalty function so that the constrained problem is transformed into an unconstrained problem. Now we define

$$\Pi(\boldsymbol{x}, \mu_i, \nu_j) = f(\boldsymbol{x}) + \sum_{i=1}^{M} \mu_i \phi_i^2(\boldsymbol{x}) + \sum_{j=1}^{N} \nu_j \max\{0, \psi_j(\boldsymbol{x})\}^2, \tag{1.43}$$

where $\mu_i \gg 1$ and $\nu_j \ge 0$.

For example, let us solve the following minimization problem:

$$\text{minimize} \quad f(x) = 40(x-1)^2, \quad x \in \mathbb{R}, \quad \text{subject to} \quad g(x) = x - a \ge 0, \tag{1.44}$$

where a is a given value. Obviously, without this constraint, the minimum value occurs at $x = 1$ with $f_{\min} = 0$. If $a < 1$, then the constraint will not affect the result. However, if $a > 1$, then the minimum should occur at the boundary $x = a$ (which can be obtained by inspecting or visualizing the objective function and the constraint). Now we can define a penalty function $\Pi(x)$ using a penalty parameter $\mu \gg 1$. We have

$$\Pi(x, \mu) = f(x) + \mu[g(x)]^2 = 40(x-1)^2 + \mu(x-a)^2, \tag{1.45}$$

which converts the original constrained optimization problem into an unconstrained problem. From the stationarity condition $\Pi'(x) = 0$ we have

$$80(x-1) - 2\mu(x-a) = 0, \quad \text{or} \quad x_* = \frac{40 - \mu a}{40 - \mu}. \tag{1.46}$$

For a particular case $a = 1$, we have $x_* = 1$, and the result does not depend on μ. However, in the case of $a > 1$ (say, $a = 5$), the result will depend on μ. When $a = 5$ and $\mu = 100$, we have $x_* = 40 - 100 \times 5/40 - 100 = 7.6667$. If $\mu = 1000$, then this gives $50 - 1000 * 5/40 - 1000 = 5.1667$. Both values are far from the exact solution $x_{\text{true}} = a = 5$. If we use $\mu = 10^4$, then we have $x_* \approx 5.0167$. Similarly, for $\mu = 10^5$, we have $x_* \approx 5.00167$. This clearly demonstrates that the solution in general depends on μ. However, it is very difficult to use extremely large values without causing extra computational difficulties.

Ideally, the formulation using the penalty method should be properly designed so that the results will not depend on the penalty coefficient, or at least the dependence should be sufficiently weak.

1.4.2 Lagrange multipliers

Another powerful method without the limitation of using large μ is the method of Lagrange multipliers. Suppose we want to minimize a function $f(\boldsymbol{x})$:

$$\text{minimize } f(\boldsymbol{x}), \qquad \boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n, \tag{1.47}$$

subject to the nonlinear equality constraint

$$h(\boldsymbol{x}) = 0. \tag{1.48}$$

Then we can combine the objective function $f(\boldsymbol{x})$ with the equality to form the new function, called the Lagrangian,

$$\Pi = f(\boldsymbol{x}) + \lambda h(\boldsymbol{x}), \tag{1.49}$$

where λ is the Lagrange multiplier, which is an unknown scalar to be determined.

This again converts the constrained optimization into an unconstrained problem for $\Pi(\boldsymbol{x})$, which is the beauty of this method. If we have M equalities

$$h_j(\boldsymbol{x}) = 0 \qquad (j = 1, \ldots, M), \tag{1.50}$$

then we need M Lagrange multipliers λ_j $(j = 1, \ldots, M)$. We thus have

$$\Pi(x, \lambda_j) = f(\boldsymbol{x}) + \sum_{j=1}^{M} \lambda_j h_j(\boldsymbol{x}). \tag{1.51}$$

The requirement of stationary conditions leads to

$$\frac{\partial \Pi}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_{j=1}^{M} \lambda_j \frac{\partial h_j}{\partial x_i} \ (i = 1, \ldots, n), \quad \frac{\partial \Pi}{\partial \lambda_j} = h_j = 0 \ (j = 1, \ldots, M). \tag{1.52}$$

These $M + n$ equations determine the n-component $\boldsymbol{x}$ and M Lagrange multipliers. As $\frac{\partial \Pi}{\partial g_j} = \lambda_j$, we can consider λ_j as the rate of the change of Π as a functional of h_j.

Example 6

For the well-known monkey surface $f(x, y) = x^3 - 3xy^2$, the function does not have a unique maximum or minimum. In fact, the point $x = y = 0$ is a saddle point. However, if we impose an extra equality $x - y^2 = 1$, we can formulate an optimization problem as

$$\text{minimize } f(x, y) = x^3 - 3xy^2, \ (x, y) \in \mathbb{R}^2,$$

subject to

$$h(x, y) = x - y^2 = 1.$$

Now we can define

$$\Phi = f(x, y) + \lambda h(x, y) = x^3 - 3xy^2 + \lambda(x - y^2 - 1).$$

The stationary conditions become

$$\frac{\partial \Phi}{\partial x} = 3x^2 - 3y^2 + \lambda = 0, \quad \frac{\partial \Phi}{\partial y} = 0 - 6xy + (-2\lambda y) = 0,$$
$$\frac{\partial \Phi}{\partial \lambda} = x - y^2 - 1 = 0.$$

The second condition $-6xy - 2\lambda y = -2y(3x + \lambda) = 0$ implies that $y = 0$ or $\lambda = -3x$.

- If $y = 0$, then the third condition $x - y^2 - 1 = 0$ gives $x = 1$. The first condition $3x^2 + 3y^2 - \lambda = 0$ leads to $\lambda = -3$. Therefore, $x = 1$ and $y = 0$ is an optimal solution with $f_{\min} = 1$. It is straightforward to verify that this solution corresponds to a minimum (not a maximum).
- If $\lambda = -3x$, then the first condition becomes $3x^2 - 3y^2 - 3x = 0$. Substituting $x = y^2 + 1$ (from the third condition), we have

$$3(y^2 + 1)^2 - 3y^2 - 3(y^2 + 1) = 0, \quad \text{or} \quad 3(y^4 + 2) = 0.$$

This equation has no solution in the real domain. Therefore, the optimality occurs at $(1, 0)$ with $f_{\min} = 1$.

1.4.3 Karush–Kuhn–Tucker conditions

There is a counterpart of the Lagrange multipliers for nonlinear optimization with inequality constraints. The Karush–Kuhn–Tucker (KKT) conditions concern the requirement for a solution to be optimal in nonlinear programming [111].

Let us know focus on the nonlinear optimization problem

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}), \ \boldsymbol{x} \in \mathbb{R}^n, \\ &\text{subject to } \phi_i(\boldsymbol{x}) = 0 \ (i = 1, \ldots, M), \quad \psi_j(\boldsymbol{x}) \le 0 \ (j = 1, \ldots, N). \end{aligned} \tag{1.53}$$

If all the functions are continuously differentiable at a local minimum $\boldsymbol{x}_*$, then there exist constants $\lambda_0, \lambda_1, \ldots, \lambda_q$ and $\mu_1, \ldots, \mu_p$ such that

$$\lambda_0 \nabla f(\boldsymbol{x}_*) + \sum_{i=1}^{M} \mu_i \nabla \phi_i(\boldsymbol{x}_*) + \sum_{j=1}^{N} \lambda_j \nabla \psi_j(\boldsymbol{x}_*) = 0, \tag{1.54}$$

$$\psi_j(\boldsymbol{x}_*) \le 0, \quad \lambda_j \psi_j(\boldsymbol{x}_*) = 0 \ (j = 1, 2, \ldots, N), \tag{1.55}$$

where $\lambda_j \geq 0$ $(i = 0, 1, \ldots, N)$. The constants satisfy $\sum_{j=0}^{N} \lambda_j + \sum_{i=1}^{M} |\mu_i| \geq 0$. This is essentially a generalized method of the Lagrange multipliers. However, there is a possibility of degeneracy when $\lambda_0 = 0$ under certain conditions.

It is worth pointing out that such KKT conditions can be useful to prove theorems and sometimes useful to gain insight into certain types of problems. However, they are not really helpful in practice in the sense that they do not give any indication where the optimal solutions may lie in the search domain so as to guide the search process.

Optimization problems, especially highly nonlinear multimodal problems, are usually difficult to solve. However, if we are mainly concerned about local optimal or suboptimal solutions (not necessarily about global optimal solutions), there are relatively efficient methods such as interior-point methods, trust-region methods, the simplex method, sequential quadratic programming, and swarm intelligence-based methods [151]. All these methods have been implemented in a diverse range of software packages. Interested readers can refer to more advanced literature.

1.5 Notes on software

Though there many different algorithms for optimization, most software packages and programming languages have some sort of optimization capabilities due to the popularity and relevance of optimization in many applications. For example, Wikipedia has some extensive lists of

- optimization software,[1]
- data mining and machine learning,[2]
- deep learning software.[3]

There is a huge list of software packages and internet resources; it requires a lengthy book to cover most of it, which is not our intention here. Interested readers can refer to them for more detail.

[1] https://en.wikipedia.org/wiki/List_of_optimization_software.
[2] https://en.wikipedia.org/wiki/Category:Data_mining_and_machine_learning_software.
[3] https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software.