计算方法

傅孝明 管理科研楼 1207 室 [1]
E-mail: fuxm@ustc.edu.cn

[1] 数学科学学院 中国科学技术大学

最优化方法

最优化方法

研究求解数学问题最优解的学科，即对于给定的实际问题，从众多的解中选取最优的解．从数学意义上说，最优化方法是一种求函数极值的方法，即在一组条件为等式或不等式的约束下，使选定的目标函数达到极值，即最大值或最小值．

最优化问题

上下班如何规划乘车路线，才能快速又经济地到达公司；旅游中如何选择航班和宾馆，既省钱又能玩得开心．

最优化方法

研究内容

最优化模型的建立、分析、求解及应用，譬如最优解的条件或标准，求解的算法，以及收敛性、时间复杂度分析等．属于计算数学，运筹学，系统工程等领域．

应用领域

随着计算机的快速发展和普及，最优化方法在经济规划、工程设计、生产管理、交通运输、国防安全等领域得到了广泛的应用，发挥着越来越重要的作用．

古老的极值问题，例如阿基米德 (Archimedes) 证明：给定周长，圆所包围的面积为最大，这是欧洲古代城堡几乎都建成圆形的原因之一.

最优化方法成为一门独立的学科是在第二次世界大战前后，由于军事上的需要以及科学技术和生产的迅速发展，许多实际的最优化问题已经无法用古典方法来解决，从而促进了近代最优化方法的产生.

具有代表性的工作有: 以美国的丹齐格 (Dantzig) 和苏联的康托罗维奇 (Kantorovich) 为代表的线性规划; 以美国的库恩 (Kuhn) 和塔克尔 (Tucker) 为代表的非线性规划; 以美国的贝尔曼 (Bellman) 为代表的动态规划; 以苏联的庞特里亚金 (Pontryagin) 为代表的极大值原理等, 这些方法后来都形成体系, 成为很活跃的领域.

近些年来在实际应用应用的驱动下, 譬如信号处理, 机器学习, 推荐系统, 自动驾驶等, 凸优化, 非光滑优化, 整数规划等得到了深入的研究.

§10.1.1 线性规划问题

在生产管理和经营活动中，经常会遇到一类问题：如何合理利用有限的人力、物力、财力等资源，以取到最佳的经济效益.

例 10.1

某工厂在计划期内要安排生产 I、II 两种产品，已知生产每种单位产品所需的设备台数及 A、B 两种原材料的消耗，如表10.1所示.

产品	产品 I	产品 II	现有资源
设备	4 台时/件	2 台时/件	18 台时
原材料 A	4 kg/件	1 kg/件	16 kg
原材料 B	1 kg/件	3 kg/件	12 kg

该工厂每生产一件 I 产品可获利 2 元，每生产一件 II 产品可获利 5 元，问应该如何安排计划使该工厂获利最多?

§10.1.1 线性规划问题

例 10.2

生产某汽车需要用 I、II、III 三种规格的轴各一根，长度规格分别为 1.5、1、0.7 米，它们需要用一种圆钢来制作，圆钢的长度为 4 米．现在要制造 1000 辆这种类型的汽车，问至少需要多少根圆钢以满足生产需求？

§10.1.2 数学模型

最优化问题的数学模型由三个要素组成:

(1) **变量**或称**决策变量**, 即问题中要优化的未知量, 用于描述问题中用数量表示的方案、措施等, 其值可由决策者控制和确定;

(2) **目标函数**, 即决策变量的函数, 按优化目标在这个函数前加上 max 或 min, 表示目标函数欲取最大值或最小值;

(3) **约束条件**, 即刻画决策变量取值时受到的各种资源条件的限制, 通常表示为含决策变量函数的等式或不等式.

§10.1.2 数学模型

如果在优化问题的数学模型中，决策变量的取值是连续的，目标函数是决策变量的线性函数，约束条件是含决策变量的线性等式或不等式，则称它为**线性规划问题**，其一般的形式为

$$\max(\min)\, z = c_1x_1 + c_2x_2 + \cdots + c_nx_n,$$

$$\text{s. t.} \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant (=, \geqslant) b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant (=, \geqslant) b_2, \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant (=, \geqslant) b_m, \\ x_1, x_2, \cdots, x_n \geqslant 0. \end{cases}$$

或简写为

$$\max(\min)\, z = \sum_{i=1}^{n} c_i x_i,$$

$$\text{s. t.} \begin{cases} \sum\limits_{j=1}^{n} a_{ij} x_j \leqslant (=, \geqslant) b_i, \quad i = 1, 2, \cdots m, \\ x_i \geqslant 0, \quad j = 1, 2, \cdots n. \end{cases}$$

§10.1.2 数学模型

在实际问题中，线性规划模型是建立在以下假设基础之上的:

(1) 比例性，指每个决策变量 x_j 在约束条件中和在目标函数中数值变化时，按 x_j 对应的系数 a_{ij} 与价值系数 c_j 严格的成比例变化;

(2) 可加性，指目标函数的总值是各项组成部分值 $c_i x_i$ 之和; 第 i 个约束关系式中各组成部分值之和就是第 i 项资源需求总量. 决策变量是相互独立的，不发生关联，且不允许有交叉;

(3) 可分性，即模型中的变量可以取小数、分数或某一实数;

(4) 确定性，即模型中的参数均为确定的常数.

然而，有些实际问题不符合上述条件，例如每件产品售价 3 元，但批量采购的话，可以打七折. 对于这类不符合线性的条件，在一些合理的假设下，有时可看作近似满足线性条件，也可用线性规划来建模.

§10.2.1 图解法

定义 10.1

称线性规划问题***有解***，是指能找到一组值 $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ 以满足所有约束条件；否则，称该问题***无解***．任意一个满足约束条件的解 $(x_1, x_2, \cdots, x_n)$ 都称为该线性规划问题的一个***可行解***．全部可行解构成的集合称为***可行域***．在可行域中使目标函数值达到最优的可行解称为***最优解***．

§10.2.1 图解法

当线性规划模型中只含 2 个变量时，问题具有明显的几何意义，可通过在平面上作图的方法求解，称为***图解法***.

图解法的基本步骤如下：在平面上建立直角坐标系；图示约束条件，判别是否存在可行域，如果存在，则找出可行域；图示目标函数，寻找最优解.

§10.2.1 图解法

下面通过例10.1来作具体说明. 先将其数学模型重写如下:

$$\max z = 2x_1 + 5x_2,$$

$$\text{s. t.} \begin{cases} 4x_1 + 2x_2 \leqslant 18, & \text{(1a)} \\ 4x_1 + x_2 \leqslant 16, & \text{(1b)} \\ x_1 + 3x_2 \leqslant 12, & \text{(1c)} \\ x_1, x_2 \geqslant 0. & \text{(1d)} \end{cases}$$

§10.2.1 图解法

(1) 以 x_1 为横坐标轴, x_2 为纵坐标轴, 适当选取坐标长度的单位, 建立平面直角坐标系. 由变量的非负约束条件(1d)知, 满足该条件的解均在第 I 象限内.

(2) 依据约束条件, 找出可行域. 约束条件(1a)表示含直线 $4x_1+2x_2=18$ 上的点及其左下方的半平面, 约束条件(1b)、(1c)的含义类似. 同时满足(1a)-(1d)的点构成该线性规划问题的可行域, 如图1所示, 即凸多边形 $OP_1P_2P_3P_4$.

(3) 图示目标函数. 随着 z 的变化, 方程 $z=2x_1+5x_2$ 表示斜率为 $-2/5$ 的一族平行直线, 见图1.

(4) 确定最优解. 因为最优解是可行域中使目标函数值 z 达到最大值的点, 从图1中可看出, 当代表目标函数的那条直线由原点开始向右上方移动时, z 的值逐渐增大, 一直移动目标函数的直线, 直到与约束条件确定的凸多边形相切时停止, 切点所在的位置即为最优解.

§10.2.1 图解法

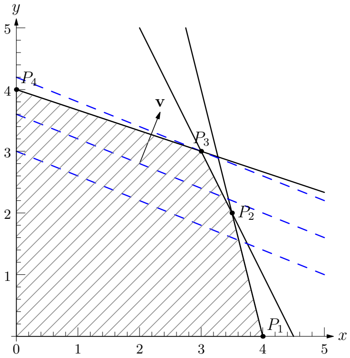

Figure: 图解法

§10.2.1 图解法

不难看出，此问题的最优解是唯一．但是，对于一般的线性规划问题来说，还可能出现下列的情况：

(1) 无穷多最优解．若将目标函数换成 $z = x_1 + 3x_2$，则目标函数的直线族与约束条件(1c)平行，此时线段 P_3P_4 上所有的点都是最优解，即有无穷多最优解．
(2) 无界解．若将约束条件(1a)-(1c)换成 $4x_1 \leqslant 18$，则可行域是无穷区域，即变量 x_2 的取值可无限增大，此时目标函数值也可无限增大，即最优解无界．
(3) 无解，或无可行解．若将约束条件(1a)换成 $4x_1 + 2x_2 \geqslant 24$，则不存在满足所有约束条件的公共区域，可行域是空集，即无解．

当求解结果出现无界解或无解情况时，一般来说是线性规划问题的数学模型有错误，前者缺乏必要的约束条件，后者存在矛盾的约束条件，需重新建模．

§10.2.1 图解法

图解法仅能用于求解具有两个变量的线性规划问题，但是它为求解一般线性规划问题提供了一些启示:

(1) 若线性规划问题的可行域存在，则可行域是凸集.

(2) 若线性规划问题的最优解存在，则最优解或最优解之一(有无穷多解时) 是可行域的某个顶点.

单纯形法的***基本思路***: 先找出可行域的任一顶点，计算在该顶点处的目标函数值，检查周围相邻顶点的目标函数值是否比这个值大，如果是否，则它就是最优解的点或最优解的点之一; 否则，转到比这个点的目标函数值更大的另一顶点. 重复上述过程，直至找到使目标函数值达到最大的顶点为止.

因目标函数和约束条件内容和形式上的差别，线性规划问题的表示形式是多种多样的．因此，为便于后续方法的描述，有必要规定线性规划问题的**标准形式**：

$$\max\ z=\sum_{i=1}^{n}c_ix_i, \tag{2}$$

$$\text{s. t.}\begin{cases}\displaystyle\sum_{j=1}^{n}a_{ij}x_j=b_i,\quad i=1,2,\cdots m, & \text{(3a)}\\ x_j\geqslant 0,\quad j=1,2,\cdots n. & \text{(3b)}\end{cases}$$

在标准形式中，目标函数为求极大值，约束条件全为等式，约束条件右端常数项 b_i 全为非负值，变量 x_j 均取非负值．

§10.2.2 标准形式

对于不符合标准形式的线性规划问题，可通过下列变换进行转化：

(1) 若目标函数为求极小值，即 $\min z = \sum_{i=1}^{n} c_i x_i$，可令 $\tilde{z} = -z$，则原问题可化为 $\max \tilde{z} = \left(-\sum_{i=1}^{n} c_i x_i\right)$，这就与标准形式中的目标函数一致了；

(2) 若约束条件是不等式，有两种情况：一种是约束方程为 "$\leqslant$" 不等式，可在不等式的左端加入一个非负变量；另一种是约束方程为 "$\geqslant$" 不等式，可在不等式的左端减去一个非负变量，新加入的变量称为**松弛变量**，则可将不等式约束化为等式约束.

(3) 若约束条件的右端项 $b_i < 0$，可将等式两端乘以"-1"，则等式右端项必大于零；

(4) 若约束条件 $x_k \leqslant 0$，可令 $x_{k+1} = -x_k$，则化为约束条件 $x_{k+1} \geqslant 0$；

(5) 若存在取值无约束的变量 x_k，可令 $x_k = x_{k+1} - x_{k+2}$，则化为约束条件 $x_{k+1}, x_{k+2} \geqslant 0$.

不难证明，任何形式的线性规划问题都可以化为标准形式.

§10.2.2 标准形式

例 10.3

将下列线性规划问题化为标准形式

$$\min z = x_1 - 4x_2 + x_3,$$

$$\text{s. t.} \begin{cases} -2x_1 - x_2 + 2x_3 \geqslant 7, \\ 3x_1 + 2x_2 - x_3 \leqslant 2, \\ x_1 + 2x_2 - x_3 = -1, \\ x_1 \leqslant 0, x_2 \geqslant 0, x_3\text{无约束}. \end{cases}$$

§10.2.3 基本定理

对于标准形式的线性规划问题，引入有关基的几个概念.

定义 10.2

设 $A=(a_{ij})_{m\times n}$ 为等式约束条件(3a)的系数矩阵，其中 $m<n$，$\mathrm{rank}(A)=m$. 若 B 是 A 的一个 $m\times m$ 满秩子矩阵，则称 B 是线性规划问题的一组*基*. 不失一般性，可设

$$B=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{pmatrix}=(\mathbf{p}_1,\mathbf{p}_2,\cdots,\mathbf{p}_m),$$

其中 B 中的每一个列向量 $\mathbf{p}_j$ 称为*基向量*，与基向量 $\mathbf{p}_j$ 对于的变量 x_j 称为*基变量*. 线性规划问题中除基变量以外的变量称为*非基变量*.

§10.2.3 基本定理

定义 10.3

在等式约束条件(3a)中，若令所有非基变量 $x_{m+1}=x_{m+2}=\cdots=x_n=0$，又因 $\det(B)\neq 0$，知由 m 个约束方程可解出 m 个基变量的唯一解 $\mathbf{x}_B=(x_1,x_2,\cdots,x_m)^{\mathrm{T}}$. 将这个解加上非基变量取 0 的值，有 $\mathbf{x}=(x_1,x_2,\cdots,x_m,0,\cdots,0)^{\mathrm{T}}$，称 $\mathbf{x}$ 为线性规划问题的**基解**.

注意，在线性规划问题求解过程中，基是可以变化的，此时基向量，基变量，基解等也随之变化.

容易看出，在基解中变量取非零值的个数不大于方程个数 m，故基解的总数不超过 $\binom{n}{m}$.

§10.2.3 基本定理

定义 10.4

满足非负约束条件(3b)的基解称为*基可行解*.

定义 10.5

与基可行解对应的基称为*可行基*.

§10.2.3 基本定理

定义 10.6

设 C 是 n 维欧式空间的一个点集, 若对任意的 $\mathbf{x}_1, \mathbf{x}_2 \in C$, 均有

$$\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in C, \quad \forall \lambda \in [0,1],$$

即 $\mathbf{x}_1$ 与 $\mathbf{x}_2$ 连线上的所有点也都在 C 中, 则称 C 是**凸集**.

从直观上看, 凸集没有凹入部分, 其内部没有空洞. 实心圆, 实心球体, 实心立方体等都是凸集, 而圆环就不是凸集.

任何两个凸集的交集是凸集.

定义 10.7

设 $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k$ 是 n 维欧式空间中的点, 若存在实数 $0 \leqslant \lambda_1, \lambda_2, \cdots, \lambda_k \leqslant 1$, 且 $\lambda_1 + \lambda_2 + \cdots + \lambda_k = 1$, 使

$$\mathbf{x} = \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \cdots + \lambda_k \mathbf{x}_k,$$

则称 $\mathbf{x}$ 是 $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k$ 的**凸组合**.
当 $0 < \lambda_i < 1\,(i = 1, 2, \cdots, k)$ 时, 称为**严格凸组合**.

§10.2.3 基本定理

定义 10.8

设 C 是 n 维欧式空间的一个凸集，点 $\mathbf{x} \in C$，若不存在不同的两点 $\mathbf{x}_1 \in C$ 和 $\mathbf{x}_2 \in C$ 使得

$$\mathbf{x} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2, \quad 0 < \lambda < 1,$$

成立，则称 $\mathbf{x}$ 是 C 的一个**顶点**或**极点**.

§10.2.3 基本定理

定理 10.1

若线性规划问题的可行域存在，则它是凸集.

引理 10.1

线性规划问题的可行解 $\mathbf{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 为基可行解的充要条件是 $\mathbf{x}$ 的正分量所对应的系数列向量是线性无关的.

定理 10.2

线性规划问题的基可行解对应于可行域的顶点.

§10.2.3 基本定理

定理 10.3

若线性规划问题的可行域有界，则必存在一个基可行解是最优解.

如果目标函数在多个顶点达到最大值，那么在这些顶点的凸组合上也达到最大值，此时线性规划问题有无穷多最优解.

若可行域无界，则可能无最优解，也可能有最优解. 若有最优解，则也必定在某顶点上取到.

§10.2.3 基本定理

若线性规划问题有最优解，则必可在某顶点上取到．又因可行域的顶点数是有限的，若采用枚举法找出所有基可行解，总数不超过 $\binom{n}{m}$，则通过比较可找到最优解．

当 n, m 的值较大时，该方法的时间复杂度迅速增加，不能满足实际应用的需求．

§10.3.1 迭代原理

一般地，在线性规划问题的等式约束条件(3a)中，方程组的变量数大于方程的个数，即 $n > m$，方程组有无穷多解．结合非负约束条件(3b)，线性规划问题的可行域常构成 $\mathbb{R}^n$ 中的多面体，它由一系列单纯形组合而成．

在 n 维欧式空间中，零维的单纯形是点，一维的单纯形是线段，二维的单纯形是三角形，三维的单纯形是四面体，k 维的单纯形是有 $k+1$ 个顶点的多面体．

§10.3.1 迭代原理

单纯形法采用迭代的策略，基本思路为：先找出一个基可行解，判断其是否为最优解，若为否，则转换到相邻的基可行解，并使目标函数值不断增大，一直找到最优解为止.

1. 确定初始基可行解.
对标准形式的线性规划问题, 在等式约束条件(3a)中, 不失一般性, 总存在一个由单位阵构成的基, 即

$$(\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

令等式约束条件(3a)中所有非基变量等于零, 可找到一个解 $\mathbf{x} = (x_1, x_2, \cdots, x_m, 0, \cdots, 0)^{\mathrm{T}} = (b_1, b_2, \cdots, b_m, 0, \cdots, 0)^{\mathrm{T}}$. 又因 $b_i \geqslant 0$, 故 $\mathbf{x}$ 满足不等式约束条件(3b), 是一个基可行解.

2. 从一个基可行解转换为相邻的基可行解.
   设初始基可行解中的前 m 个为基变量, 即
   $\mathbf{x}^0 = (x_1^0, x_2^0, \cdots, x_m^0, 0, \cdots, 0)^{\mathrm{T}}$, 其中
   $x_i^0 \geqslant 0, i = 1, 2, \cdots, m$. 等式约束条件(3a) 可写成增广矩阵的形式:

$$
\begin{array}{cccccccccc}
\left(\begin{array}{cccccccc|c}
1 & 0 & \cdots & 0 & a_{1,m+1} & \cdots & a_{1j} & \cdots & a_{1n} & b_1 \\
0 & 1 & \cdots & 0 & a_{2,m+1} & \cdots & a_{2j} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & 1 & a_{m,m+1} & \cdots & a_{mj} & \cdots & a_{mn} & b_m
\end{array}\right) \\
\uparrow & \uparrow & & \uparrow & \uparrow & & \uparrow & & \uparrow & \uparrow \\
\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_m & \mathbf{p}_{m+1} & \cdots & \mathbf{p}_j & \cdots & \mathbf{p}_n & \mathbf{b}
\end{array}
$$

由基可行解的定义，知

$$\sum_{i=1}^{m} x_i^0 \mathbf{p}_i = \mathbf{b}.$$

又因 $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$ 构成一组基，故向量 $\mathbf{p}_j$ 可表示为

$$\mathbf{p}_j = \sum_{i=1}^{m} a_{ij} \mathbf{p}_i.$$

对任意的 $\lambda > 0$，从而有

$$\sum_{i=1}^{m} x_i^0 \mathbf{p}_i + \lambda\left(\mathbf{p}_j - \sum_{i=1}^{m} a_{ij}\mathbf{p}_i\right) = \mathbf{b} \Longleftrightarrow \sum_{i=1}^{m}(x_i^0 - \lambda a_{ij})\mathbf{p}_i + \lambda \mathbf{p}_j = \mathbf{b}.$$

§10.3.1 迭代原理

若记

$\mathbf{x}^1 = (x_1^0 - \lambda a_{1j}, x_2^0 - \lambda a_{2j}, \cdots, x_m^0 - \lambda a_{mj}, 0, \cdots, \lambda, \cdots, 0)^{\mathrm{T}}$,

显然 $\mathbf{x}^1$ 也满足等式约束条件(3a). 要使 $\mathbf{x}^1$ 成为一个基可行解, 则 λ 需满足

$$x_i^0 - \lambda a_{ij} \geqslant 0, \quad i = 1, 2, \cdots, m,$$

且其中至少有一个不等式取到等号. 显然, 当 $a_{ij} \leqslant 0$ 时, 上述不等式对 $\lambda > 0$ 总成立. 若取

$$\lambda = \min_i \left\{ \frac{x_i^0}{a_{ij}} \,\middle|\, a_{ij} > 0 \right\} = \frac{x_l^0}{a_{lj}},$$

容易验证 $\mathbf{x}^1$ 是一个基可行解.

§10.3.1 迭代原理

将变量 $x_1, \cdots, x_{l-1}, x_j, x_{l+1}, \cdots, x_m$ 对应的向量和 $\mathbf{b}$ 写成增广矩阵的形式:

$$
\begin{array}{ccccccccc}
\left(\begin{array}{cccccccc|c}
1 & 0 & \cdots & 0 & a_{1j} & 0 & \cdots & 0 & b_1 \\
0 & 1 & \cdots & 0 & a_{2j} & 0 & \cdots & 0 & b_2 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & 1 & a_{l-1,j} & 0 & \cdots & 0 & b_{l-1} \\
0 & 0 & \cdots & 0 & a_{l,j} & 0 & \cdots & 0 & b_l \\
0 & 0 & \cdots & 0 & a_{l+1,j} & 1 & \cdots & 0 & b_{l+1} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & 0 & a_{m,j} & 0 & \cdots & 1 & b_m
\end{array}\right) \\
\begin{array}{ccccccccc}
\uparrow & \uparrow & & \uparrow & \uparrow & \uparrow & & \uparrow & \uparrow \\
\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_{l-1} & \mathbf{p}_j & \mathbf{p}_{l+1} & \cdots & \mathbf{p}_m & \mathbf{b}
\end{array}
\end{array}
$$

§10.3.1 迭代原理

因 $a_{lj} > 0$, 故 $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{l-1}, \mathbf{p}_j, \mathbf{p}_{l+1}, \cdots, \mathbf{p}_m$ 构成一组基. 对上述增广矩阵施行初等行变换: 第 l 行乘以 $(1/a_{lj})$, 再分别乘以 $(-a_{ij})$ 加到第 i 行上去, $i = 1, \cdots, l-1, l+1, \cdots, m$, 得变换后的向量 $\mathbf{b} = (b_1 - \lambda a_{1j}, \cdots, b_{l-1} - \lambda a_{l-1,j}, \lambda, b_{l+1} - \lambda a_{l+1,j}, \cdots, b_m - \lambda a_{mj})^{\mathrm{T}}$. 明显地, $\mathbf{x}^1$ 是与 $\mathbf{x}^0$ 相邻的一个基可行解, 且相应基向量组成的矩阵仍是单位阵.

3. 最优性检验和解的判断.
将基可行解 $\mathbf{x}^0$ 和 $\mathbf{x}^1$ 分别代入目标函数(3)得

$$z^0 = \sum_{i=1}^{m} c_i x_i^0,$$

$$z^1 = \sum_{i=1}^{m} c_i(x_i^0 - \lambda a_{ij}) + \lambda c_j = z^0 + \lambda\left(c_j - \sum_{i=1}^{m} c_i a_{ij}\right).$$

又因 $\lambda > 0$, 故当 $c_j - \sum_{i=1}^{m} c_i a_{ij} > 0$ 时, 就有 $z^1 > z^0$. 若记 $\sigma_j = c_j - \sum_{i=1}^{m} c_i a_{ij}$, 则 σ_j 可用作线性规划问题的解进行最优性检验的指标.

§10.3.1 迭代原理

具体的判别规则如下:

(a) 当所有的 $\sigma_j \leqslant 0$ 时, 现有基可行解的目标函数值比起相邻各基可行解的目标函数值都大, 又因可行域是凸集, 故该基可行解是最优解.

(b) 当所有的 $\sigma_j \leqslant 0$, 且存在某个非基变量 x_k, 使得 $\sigma_k = 0$, 这意味着可以找到另外一个基可行解使目标函数值取到最大值, 从而它们之间连线上的点也都取到最大值, 故线性规划问题存在无穷多解. 反之, 当所有非基变量的 $\sigma_j < 0$ 时, 线性规划问题具有唯一的最优解.

(c) 当存在某个 $\sigma_j > 0$, 且 $\mathbf{p}_j \leqslant \mathbf{0}$, 这意味着对任意 $\lambda > 0$, 均有 $x_i^0 - \lambda a_{ij} \geqslant 0$, 而 λ 的取值可无限增大, z^1 的值也可无限增大, 故线性规划问题有无界解.

对求解结果为无可行解的判别将在后面讨论.

§10.3.2 计算步骤

为检验一个基可行解是否最优，需要将其目标函数值与相邻基可行解的目标函数值进行比较．为了书写规范和便于计算，下面引入一种称为***单纯形表***的表格:

$c_j \to$			c_1	$\cdots$	c_m	$\cdots$	c_j	$\cdots$	c_n
$\mathbf{c_B}$	$\mathbf{x_B}$	$\mathbf{b}$	x_1	$\cdots$	x_m	$\cdots$	x_j	$\cdots$	x_n
c_1	x_1	b_1	1	$\cdots$	0	$\cdots$	a_{1j}	$\cdots$	a_{1n}
c_2	x_2	b_2	0	$\cdots$	0	$\cdots$	a_{2j}	$\cdots$	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$
c_m	x_m	b_m	0	$\cdots$	1	$\cdots$	a_{mj}	$\cdots$	a_{mn}
σ_j			0	$\cdots$	0	$\cdots$	$c_j - \sum_{i=1}^{m} c_i a_{ij}$	$\cdots$	$c_n - \sum_{i=1}^{m} c_i a_{in}$

§10.3.2 计算步骤

其中

- $\mathbf{x_B}$ 列中填入基变量，这里是 $x_1, x_2, \cdots, x_m$;
- $\mathbf{c_B}$ 列中填入目标函数中基变量的相应系数，这里是 $c_1, c_2, \cdots, c_m$;
- $\mathbf{b}$ 列中填入等式约束方程组右端的常数;
- x_j 列中填入等式约束中变量 x_j 对应的系数向量;
- c_j 行中填入目标函数中变量 x_j 的相应系数;
- σ_j 行中填入非基变量 x_j 的检验指标 $c_j - \sum_{i=1}^{m} c_i a_{ij}$.

在迭代计算过程中，每找出一个新的基可行解时，就重新建立一张单纯形表.

单纯形法的计算步骤:

1. 根据线性规划模型的标准形式确定初始可行基和初始基可行解, 建立单纯形表;
2. 计算各非基变量 x_k 的检验指标

$$\sigma_k = c_k - \sum_{i=1}^{m} c_i a_{ik}, \quad k = m+1, m+2, \cdots, n.$$

   若 $\sigma_k \leqslant 0$ 对所有 $k = m+1, m+2, \cdots, n$ 成立, 则已得到最优解, 算法终止. 否则, 进入下一步.
3. 在 $\sigma_k > 0\,(m+1 \leqslant k \leqslant n)$ 中, 若有某个 σ_j 对应的 x_j 的系数向量 $\mathbf{p}_j \leqslant \mathbf{0}$, 则该线性规划问题有无界解, 算法终止. 否则, 进入下一步.

4. 根据 $\max\{\sigma_k\} = \sigma_j$, 确定 x_j 为换入变量, 按公式

$$\lambda = \min_i \left\{ \frac{x_i^0}{a_{ij}} \,\middle|\, a_{ij} > 0 \right\} = \frac{x_l^0}{a_{lj}}$$

   可确定 x_l 为换出变量;

5. 以 a_{lj} 为主元素, 对等式约束条件(3a)的增广矩阵施行初等行变换, 将变量 x_j 对应的列向量

$$\mathbf{p}_j = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{lj} \\ \vdots \\ a_{mj} \end{pmatrix} \Longrightarrow \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{第 } l \text{ 行},$$

   并将 $\mathbf{x_B}$ 列中的 x_l 换为 x_j, 建立新的单纯形表.

6. 重复上述步骤 2-5, 直至算法终止.

§10.3.2 计算步骤

例 10.4

用单纯形法解线性规划问题

$$\max z = 2x_1 + 5x_2$$

$$\begin{cases} 4x_1 + 2x_2 \leqslant 18, \\ 4x_1 + x_2 \leqslant 16, \\ x_1 + 3x_2 \leqslant 12, \\ x_1, x_2 \geqslant 0. \end{cases}$$

§10.3.2 计算步骤

单纯形法计算中的几个问题

- 初始基可行解的寻找. 在前面, 曾假设在等式约束条件(3a)中存在一个单位阵构成的基, 以此求初始基可行解和建立初始单纯形表十分方便. 但是, 有化为标准形式后等式约束条件的系数矩阵中不存在子单位矩阵的例子. 对于这种情形, 可以采用人工变量法或两阶段法来处理.

§10.3.2 计算步骤

- 解的退化. 在按最小比值 λ 来确定换出的基变量时, 可能会出现两个或以上相同的最小比值, 从而使下一个表的基可行解中出现一个或多个基变量等于零的退化解. 该现象出现的原因是模型中存在多余的约束, 使得多个基可行解对应于同一顶点. 当退化解出现时, 就可能出现迭代计算的无限循环, 尽管可能性极其微小. 为避免出现计算的循环, 可采用勃兰特 (Bland) 规则: (a) 当存在多个 $\sigma_j > 0$ 时, 始终选取下标最小的变量作为换入变量; (b) 当计算 λ 值出现两个或以上相同的最小值时, 始终选取下标最小的变量作为换出变量.

§10.3.2 计算步骤

- 无可行解的判别. 当线性规划问题中添加人工变量后, 初始单纯形表中的解因含人工变量, 故实质上是非可行的. 当求解结果出现所有 $\sigma_j \leqslant 0$ 时, 若基变量中仍含有非零的人工变量, 则表明该线性规划问题无可行解.

§10.3.2 计算步骤

在实际应用中，尽管单纯形法非常有效，但在理论上，其算法时间复杂度仍是指数级别的．一个自然的问题是：能否找到算法时间复杂度更低的算法？在此驱动下，产生了*内点法*，其算法时间复杂度是多项式级别的，是求解线性规划问题的另一有效方法．

§10.4.1 数学模型

非线性优化问题

如果最优化问题中的目标函数或约束条件中有关于决策变量的非线性函数，则称它是*非线性优化*问题或*非线性规划*问题.

例 10.5

在半径为 R 的圆内有一内接三角形 $\triangle ABC$，其顶点 A, B, C 可以在圆上移动，求 $\triangle ABC$ 面积的最大值.

§10.4.1 数学模型

例 10.6

现有一块椭球形的石材，其半轴分别为 $a, b, c > 0$. 若要从中加工出一块长方体的石砖，要求长方体各边平行于椭球的半轴且浪费最少，问该如何设计切割方案?

§10.4.1 数学模型

不失一般性, **非线性优化问题**可写成:

$$\min f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n$$

$$\text{s. t.} \begin{cases} c_i(\mathbf{x}) \geqslant 0, & i \in \mathcal{I} \\ c_i(\mathbf{x}) = 0, & i \in \mathcal{E}. \end{cases}$$

其中 $f(\mathbf{x}), c_i(\mathbf{x})$ 是有 n 个自变量的连续实函数, $\mathcal{E}, \mathcal{I}$ 分别是等式约束和不等式约束的有限指标集.

§10.4.1 数学模型

如果非线性优化问题中的目标函数和约束条件中的函数均为凸函数，则称它为***凸优化***问题.

如果目标函数的自变量为离散变量，譬如整数或有限集合中的元素，则称它为***离散优化***. 整数规划问题、旅行商问题、图着色问题、最小生成树等都是离散优化问题.

§10.4.2 函数极值

定义 10.9

设函数 $f : D \to \mathbb{R}$，其中区域 $D \subset \mathbb{R}^n$. 如果对于 D 的内点 $\mathbf{x}^*$, 存在领域 $U(\mathbf{x}^*) \subset D$, 当 $\mathbf{x} \in U(\mathbf{x}^*)$ 时, 有 $f(\mathbf{x}) \geqslant f(\mathbf{x}^*)$ $(f(\mathbf{x}) \leqslant f(\mathbf{x}^*))$, 则称函数 f 在 D 的内点 $\mathbf{x}^*$ 取***局部极小值*** (***局部极大值***), 简称***极小值*** (***极大值***). 如果对 $\mathbf{x} \in U(\mathbf{x}^*)\backslash\{\mathbf{x}^*\}$ 有严格不等式 $f(\mathbf{x}) > f(\mathbf{x}^*)$ $(f(\mathbf{x}) < f(\mathbf{x}^*))$, 则称函数 f 在 $\mathbf{x}^*$ 取***严格局部极小值*** (***严格局部极大值***).

定义 10.10

函数的局部极大值与局部极小值统称为函数的***局部极值***, 简称***极值***, 函数取极值的点称为***极值点***.

§10.4.2 函数极值

定理 10.4

设函数 $f: D \to \mathbb{R}$ 在内点 $\mathbf{x}^*$ 取极值且 $Jf(\mathbf{x}^*)$ 存在，则 $Jf(\mathbf{x}^*) = \mathbf{0}$，即

$$\frac{\partial f}{\partial x_i}(\mathbf{x}^*) = 0, \quad i = 1, 2, \cdots, n.$$

定义 10.11

设函数 $f: D \to \mathbb{R}$, 在 D 中使得 $Jf(\mathbf{x}) = \mathbf{0}$ 的一切内点称为函数 f 的**驻点**.

§10.4.2 函数极值

显然，极值点一定是驻点，但一般来说驻点未必是极值点．若记

$$Jf(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right), Hf(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{pmatrix},$$

分别称 $Jf(\mathbf{x})$ 和 $Hf(\mathbf{x})$ 为函数 f 的 **Jacobian 矩阵**和 **Hessian 矩阵**.

§10.4.2 函数极值

定理 10.5

设函数 $f(\mathbf{x})$ 在 D 上具有连续的二阶偏导数，即 $f(\mathbf{x}) \in C^2(D)$.

(1) (必要性) 若 $f(\mathbf{x})$ 在 $\mathbf{x}^*$ 取局部极小(大)值，则 $J(\mathbf{x}^*) = \mathbf{0}$ 且 $Hf(\mathbf{x}^*)$ 半正(负)定;

(2) (充分性) 若 $Jf(\mathbf{x}^*) = \mathbf{0}$ 且 $Hf(\mathbf{x}^*)$ 正(负)定，则 $f(\mathbf{x})$ 在 $\mathbf{x}^*$ 取得严格局部极小(大)值.

判别函数内部局部极值点的二阶必要条件和充分条件.

§10.4.2 函数极值

定义 10.12

设 $\mathbf{x}^* \in D$, 若对所有 $\mathbf{x} \in D$, 都有 $f(x) \geqslant f(\mathbf{x}^*)$ ($f(x) \leqslant f(\mathbf{x}^*)$), 则称 $\mathbf{x}^*$ 为 $f(\mathbf{x})$ 在 D 上的**全局极小值点** (**全局极大值点**). 若对所有 $\mathbf{x} \in D \backslash \mathbf{x}^*$, 都有 $f(x) > f(\mathbf{x}^*)$ ($f(x) < f(\mathbf{x}^*)$), 则称 $\mathbf{x}^*$ 为 $f(\mathbf{x})$ 在 D 上的**全局严格极小值点** (**全局严格极大值点**)

全局极值点可能在定义域的边界点上取到. 然而, 对于这些边界上的极值点, 并没有实用的判别条件. 因此, 寻找全局极值点是一个相当困难的任务.

幸好, 在许多实际问题中, 局部极值点就已能满足应用的需求, 故非线性优化主要研究寻找局部极值点.

§10.4.2 函数极值

凸优化问题的极值

对于定义在凸集上的凸函数 $f(\mathbf{x})$，可以证明：$f(\mathbf{x})$ 的局部极值点就是全局极值点，且所有极值点构成一个凸集. 进一步，若 $f(\mathbf{x})$ 是严格凸函数，则 $f(\mathbf{x})$ 的全局极值点是唯一的.

§10.4.3 迭代法

求解非线性优化问题的一个自然的想法是: 1) 令 $\nabla f(\mathbf{x}) = \mathbf{0}$, 求出 $f(\mathbf{x})$ 的所有驻点; 2) 对每一驻点, 利用局部极值的充分条件进行判别, 找出所要的解.

对某些较简单的函数, 这样做是可行的, 但是, 对于一般的 n 元函数, 由条件 $\nabla f(\mathbf{x}) = \mathbf{0}$ 得到的方程通常是一个非线性方程组, 求解非常困难. 当函数 $f(\mathbf{x})$ 不可微分时, 就更不能用上述方法了.

因此, 求解非线性优化问题常使用迭代法.

§10.4.3 迭代法

迭代法的基本思路是：为求函数 $f(\mathbf{x})$ 的最优解，首先给定一个初始估计 $\mathbf{x}_0$，然后按某种规则找出比 $\mathbf{x}_0$ 更好的一个解 $\mathbf{x}_1$，即 $f(\mathbf{x}_1) < f(\mathbf{x}_0)$，重复上述过程，得到一个解序列 $\{\mathbf{x}_k\}_{k=0}^{\infty}$. 若这个解序列有极限 $\mathbf{x}^*$，即

$$\lim_{k\to\infty} \|\mathbf{x}_k - \mathbf{x}^*\| = 0,$$

则称它收敛于 $\mathbf{x}^*$.

迭代法的关键是：设计有效的规则，使得所产生的解序列 $\{\mathbf{x}_k\}_{k=0}^{\infty}$ 收敛于该问题的最优解之一.

§10.4.3 迭代法

若由某种算法所产生的解序列 $\{\mathbf{x}_k\}_{k=0}^{\infty}$ 使目标函数 $f(\mathbf{x}_k)$ 逐步减少, 则称它为***下降算法***. 现假定已迭代到点 $\mathbf{x}_k$, 若从该点出发沿任何方向移动都不能使目标函数 $f(\mathbf{x})$ 值下降, 则 $\mathbf{x}_k$ 是一个局部极小点, 迭代停止; 否则, 从该点出发至少存在一个方向使目标函数值有所下降, 则可选择使目标函数值下降的某一方向 $\mathbf{p}_k$ 作为搜索方向, 选择一个合适的步长 λ_k, 得到一个新的迭代点

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{p}_k,$$

其中称 $\mathbf{p}_k$ 为***搜索方向***, λ_k 为***步长***或***步长因子***.

§10.4.3 迭代法

下降迭代算法的计算步骤如下:

1. 选定某一初始点 $\mathbf{x}_0$, 并令 $k \leftarrow 0$;
2. 确定搜索方向 $\mathbf{p}_k$;
3. 从 $\mathbf{x}_k$ 出发, 沿搜索方向 $\mathbf{p}_k$ 求步长 λ_k, 以产生下一个迭代点 $\mathbf{x}_{k+1}$;
4. 检查新点 $\mathbf{x}_{k+1}$ 是否为极小点或近似极小点. 若是, 则算法终止; 否则, 令 $k \leftarrow k+1$, 重复步骤 2-4.

§10.4.3 迭代法

选取搜索方向 $\mathbf{p}_k$ 是最关键的一步, 各种算法的主要差别之一在于确定搜索方向的方法不同.

确定步长 λ_k 亦可采用不同的方法. 最简单的一种是令它等于某一常数, 譬如 $\lambda_k = 1$, 这样做计算简便, 但不能保证使目标函数值下降; 第二种称为可接收点算法, 只要能使目标函数值下降, 可任意选择步长 λ_k; 第三种方法是沿着搜索方向使目标函数值下降最多, 即求解一个子优化问题

$$\min_{\lambda\in\mathbb{R}} f(\mathbf{x}_k + \lambda\mathbf{p}_k),$$

称这一过程为**最优一维搜索**或**精确一维搜索**, 所确定的步长为**最佳步长**.

§10.4.3 迭代法

最优一维搜索有一个重要的性质：在搜索方向上所得到的最优点处，目标函数的梯度和该搜索方向正交.

定理 10.6

设目标函数 $f(\mathbf{x})$ 具有一阶连续偏导数，$\mathbf{x}_{k+1}$ 按如下规则产生：

$$\begin{cases} \lambda_k = \arg\min\limits_{\lambda\in\mathbb{R}} f(\mathbf{x}_k + \lambda\mathbf{p}_k), \\ \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{p}_k, \end{cases}$$

则有

$$\nabla f(\mathbf{x}_{k+1})^{\mathrm{T}}\mathbf{p}_k = 0.$$

§10.4.3 迭代法

定义 10.13

设序列 $\{\mathbf{x}_k\}_{k=0}^{\infty}$ 收敛于 $\mathbf{x}^*$, 若存在常数 $c>0$, $\alpha \geqslant 1$ 及整数 $k_0>0$, 使得当 $k>k_0$ 时, 均有

$$\|\mathbf{x}_{k+1}-\mathbf{x}^*\| \leqslant c\|\mathbf{x}_k-\mathbf{x}^*\|^{\alpha}$$

成立, 则称 $\{\mathbf{x}_k\}_{k=0}^{\infty}$ 的**收敛阶**为 α, 或 $\{\mathbf{x}_k\}_{k=0}^{\infty}$ 是α **阶收敛**. 当 $\alpha=2$ 时, 称为**二阶收敛**; 当 $1<\alpha<2$ 时, 称为**超线性收敛**; 当 $\alpha=1$ 时, 称为**线性收敛**或**一阶收敛**.

通常, 线性收敛速度是比较慢的, 二阶收敛是很快的, 超线性收敛介于它们之间.

若一个算法具有超线性或更高的收敛速度, 一般就认为它是一个很好的算法了.

§10.4.3 迭代法

算法的收敛准则. 常用的有以下几种:

(1) 依据相邻两次迭代的绝对误差

$$\|\mathbf{x}_{k+1}-\mathbf{x}_k\|<\varepsilon_1,\quad |f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)|<\varepsilon_2;$$

(2) 依据相邻两次迭代的相对误差

$$\frac{\|\mathbf{x}_{k+1}-\mathbf{x}_k\|}{\|\mathbf{x}_k\|}<\varepsilon_3,\quad \frac{|f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)|}{|f(\mathbf{x}_k)|}<\varepsilon_4;$$

(3) 依据目标函数梯度的模

$$\|\nabla f(\mathbf{x}_k)\|<\varepsilon_5,$$

其中 $\varepsilon_1,\varepsilon_2,\varepsilon_3,\varepsilon_4,\varepsilon_5$ 为事先设定的精度参数. 此外, 为了防止程序陷入无限循环, 还可设置最大迭代次数 N.

§10.5.1 数学模型

所谓**一维搜索**，又称**线性搜索**，是指单变量函数的最优化，它是多变量函数优化的基础.

在用迭代法求函数的极小值时，经常用到一维搜索，即沿着某一已知方向求目标函数的极小值点. 另外，如果优化问题中的目标函数的导数或偏导数不存在或难以计算时，也常采用搜索的方法求解.

一维搜索的主要步骤：首先确定包含问题最优解的搜索区间；然后采用某种分割技术或插值方法缩小这个区间，进行搜索求解.

§10.5.1 数学模型

定义 10.14

设 $\varphi:\mathbb{R}\mapsto\mathbb{R}$ 是一个连续函数，$\lambda^*\in[0,+\infty)$，且有

$$\varphi(\lambda^*)=\min_{\lambda\geqslant 0}\varphi(\lambda).$$

若存在闭区间 $[a,b]\subset[0,+\infty)$，使 $\lambda^*\in[a,b]$，则称 $[a,b]$ 是一维优化问题 $\min\limits_{\lambda\geqslant 0}\varphi(\lambda)$ 的**搜索区间**.

确定搜索区间的一种简单方法叫进退法，它的基本思想为：从定义域内某一点出发，按一定步长，试图确定某一区间，使得函数值呈现“高 - 低 - 高”的形状. 一个方向不成功，就退回来，再沿相反方向进行搜索.

§10.5.1 数学模型

进退法的计算步骤如下:

1. 设置初始值. $\lambda_0 \in [0, +\infty)$, $h_0 > 0$, 加倍系数 $t > 1$, 常取 $t = 2$, 计算 $\varphi(\lambda_0)$, $k \leftarrow 0$;
2. 比较目标函数值. 令 $\lambda_{k+1} \leftarrow \lambda_k + h_k$, 计算 $\varphi(\lambda_{k+1})$. 若 $\varphi(\lambda_{k+1}) < \varphi(\lambda_k)$, 进入下一步; 否则, 转步骤 4;
3. 更新搜索步长. 令 $h_{k+1} \leftarrow t \times h_k$, $\lambda \leftarrow \lambda_k$, $\lambda_k \leftarrow \lambda_{k+1}$, $k \leftarrow k+1$, 转步骤 2;
4. 反向搜索. 若 $k = 0$, 转换搜索方向, 令 $h_k \leftarrow (-h_k)$, $\lambda_k \leftarrow \lambda_{k+1}$, 转步骤 2; 否则, 停止迭代, 并令

$$a = \min\{\lambda, \lambda_{k+1}\}, \quad b = \max\{\lambda, \lambda_{k+1}\},$$

输出 $[a, b]$.

§10.5.1 数学模型

一维搜索方法主要针对单峰函数在单峰区间上的优化问题.

定义 10.15

设 $\varphi:\mathbb{R}\mapsto\mathbb{R}$ 是一个连续函数, 若存在 $[a,b]$ 及 $\lambda^*\in[a,b]$, 使 $\varphi(\lambda)$ 在 $[a,\lambda^*]$ 上严格递减, 在 $[\lambda^*,b]$ 上严格递增, 则称 $[a,b]$ 是函数 $\varphi(\lambda)$ 的***单峰区间***, $\varphi(\lambda)$ 是区间 $[a,b]$ 上的***单峰函数***.

引理 10.2

设 $\varphi:\mathbb{R}\mapsto\mathbb{R}$ 是一个连续函数, $[a,b]$ 是 $\varphi(\lambda)$ 的单峰区间, $\alpha_1,\alpha_2\in[a,b]$, 且 $\alpha_1<\alpha_2$.

(1) *若 $\varphi(\alpha_1)\leqslant\varphi(\alpha_2)$, 则 $[a,\alpha_2]$ 是 $\varphi(\lambda)$ 的单峰区间;*
(2) *若 $\varphi(\alpha_1)\geqslant\varphi(\alpha_2)$, 则 $[\alpha_1,b]$ 是 $\varphi(\lambda)$ 的单峰区间.*

§10.5.1 数学模型

如果 $\varphi(\lambda)$ 是 $[a,b]$ 上的单峰函数, 则可通过比较 $\varphi(\lambda)$ 的函数值, 来缩小搜索区间. 设 $\varphi(\lambda)=f(\mathbf{x}_k+\lambda\mathbf{p}_k)$, $\varphi(\lambda)$ 是搜索区间 $[a,b]$ 上的单峰函数. 记第 k 次迭代所产生的搜索区间为 $[a_k,b_k]$, 取两个试探点 $\alpha_k,\beta_k\in[a_k,b_k]$, 且 $\alpha_k<\beta_k$, 计算 $\varphi(\alpha_k)$ 和 $\varphi(\beta_k)$, 根据引理10.2, 知

(1) 若 $\varphi(\alpha_k)\leqslant\varphi(\beta_k)$, 则令 $a_{k+1}\leftarrow a_k$, $b_{k+1}\leftarrow\beta_k$;
(2) 若 $\varphi(\alpha_k)>\varphi(\beta_k)$, 则令 $a_{k+1}\leftarrow\alpha_k$, $b_{k+1}\leftarrow b_k$.

反复执行上述步骤, 搜索区间逐渐变小, 当达到所需的精度时, 算法终止.

问题: 如何确定试探点 α_k,β_k 的位置.

§10.5.2 黄金分割法

要求试探点 α_k, β_k 满足以下条件:

(1) α_k 和 β_k 到搜索区间 $[a_k, b_k]$ 的端点距离相等，即

$$b_k - \alpha_k = \beta_k - a_k; \tag{4}$$

(2) 每次迭代，搜索区间长度的缩短率相等，即

$$b_{k+1} - a_{k+1} = \tau(b_k - a_k). \tag{5}$$

将式 (4) 与式 (5) 联立, 可得

$$\alpha_k = a_k + (1-\tau)(b_k - a_k), \tag{6}$$

$$\beta_k = a_k + \tau(b_k - a_k). \tag{7}$$

现考虑 $\varphi(\alpha_k) \leqslant \varphi(\beta_k)$ 的情形, 则新的搜索区间为 $[a_{k+1}, b_{k+1}] = [a_k, \beta_k]$. 为进一步缩短搜索区间, 需取新的试探点 $\alpha_{k+1}, \beta_{k+1}$. 由式 (7) 知,

$$\begin{aligned}
\beta_{k+1} &= a_{k+1} + \tau(b_{k+1} - a_{k+1}) \\
&= a_k + \tau(\beta_k - a_k) \\
&= a_k + \tau[a_k + \tau(b_k - a_k) - a_k] \\
&= a_k + \tau^2(b_k - a_k).
\end{aligned}$$

§10.5.2 黄金分割法

若令 $\tau^2 = 1 - \tau$, 则

$$\beta_{k+1} = a_k + (1-\tau)(b_k - a_k) = \alpha_k.$$

此时, 新的试探点 β_{k+1} 不需要重新计算, 只要取 α_k 即可. 从而在每次迭代中, 仅需取一个新的试探点即可. 类似地, 考虑 $\varphi(\alpha_k) > \varphi(\beta_k)$ 的情形, 由式 (6) 知, 新的试探点 $\alpha_{k+1} = \beta_k$, 它也不需要重新计算.

由于 $\tau^2 = 1 - \tau$, 且 $\tau > 0$, 解得

$$\tau = \frac{-1+\sqrt{5}}{2} \approx 0.618.$$

因 0.618 等于黄金分割率, 故上述方法常称为***黄金分割法***或 ***0.618 法***.

§10.5.2 黄金分割法

黄金分割法的计算步骤如下:

1. 选取初始数据. 确定初始搜索区间 $[a_0, b_0] = [a, b]$ 和精度要求 $\delta > 0$. 计算初始试探点 α_0, β_0, 并令 $k \leftarrow 0$;
2. 比较函数值. 若 $\varphi(\alpha_k) \leqslant \varphi(\beta_k)$, 进入下一步; 否则, 转步骤 4;
3. 若 $\beta_k - a_k \leqslant \delta$, 则停止计算, 输出 α_k; 否则, 令 $a_{k+1} \leftarrow a_k$, $b_{k+1} \leftarrow \beta_k$, $\beta_{k+1} \leftarrow \alpha_k$, $\varphi(\beta_{k+1}) \leftarrow \varphi(\alpha_k)$, $\alpha_{k+1} \leftarrow a_{k+1} + 0.382(b_{k+1} - a_{k+1})$, 计算 $\varphi(\alpha_{k+1})$, 转步骤 5;
4. 若 $k - \alpha_k \leqslant \delta$, 则停止计算, 输出 β_k; 否则, 令 $a_{k+1} \leftarrow \alpha_k$, $b_{k+1} \leftarrow b_k$, $\alpha_{k+1} \leftarrow \beta_k$, $\varphi(\alpha_{k+1}) \leftarrow \varphi(\beta_k)$, $\beta_{k+1} \leftarrow a_{k+1} + 0.618(b_{k+1} - a_{k+1})$, 计算 $\varphi(\beta_{k+1})$, 进入下一步;
5. 令 $k \leftarrow k + 1$, 转步骤 2.

每次迭代后, 搜索区间的缩短率为 τ. 若搜索初始区间为 $[a, b]$, 则经过 n 迭代后, 搜索区间的长度为 $\tau^n(b - a)$, 算法是收敛的, 且收敛速度是线性的.

斐波那契法：搜索区间长度的缩短率不是采用黄金分割数，而是采用斐波那契数列 $\{F_k\}_{k=0}^{\infty}$，即

$$F_0 = F_1 = 1,$$

$$F_{k+1} = F_k + F_{k-1}, \quad k = 1, 2, \cdots .$$

在该方法中，试探点 α_k，β_k 的取法为

$$\alpha_k = a_k + \left(1 - \frac{F_{n-k}}{F_{n-k+1}}\right)(b_k - a_k),$$

$$\beta_k = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k).$$

§10.5.3 斐波那契法

每次缩短率满足

$$b_{k+1} - a_{k+1} = \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k),$$

其中 n 是计算函数值的次数. 若要求经过 n 次迭代后, 所得的区间长度不超过 δ, 即 $b_n - a_n \leqslant \delta$, 那么

$$\begin{aligned} b_n - a_n &= \frac{F_1}{F_2}(b_{n-1} - a_{n-1}) \\ &= \frac{F_1}{F_2} \cdot \frac{F_2}{F_3} \cdots \frac{F_{n-1}}{F_n}(b_1 - a_1) \\ &= \frac{1}{F_n}(b_1 - a_1). \end{aligned}$$

从而有 $F_n \geqslant \dfrac{b_1 - a_1}{\delta}$.

§10.5.3 斐波那契法

斐波那契数列的通项公式

$$F_k = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+1}\right\}, \quad k = 0, 1, 2, \cdots .$$

因此, 对于给定的初始搜索区间 $[a, b]$ 及精度参数 δ, 可先确定 F_n 的下界, 然后利用通项公式求出满足要求的最小 n, 再通过采用与黄金分割法类似的迭代过程, 求出最终的搜索区间.

§10.5.3 斐波那契法

容易看出

$$\lim_{k\to\infty}\frac{F_{k-1}}{F_k}=\frac{\sqrt{5}-1}{2}=\tau,$$

故斐波那契法的区间缩短率与黄金分割法相同，也是线性收敛的. 可以证明: 斐波那契法是用分割方法求一维极小化问题的最优策略，而黄金分割法是近似最优的. 但由于后者简单易行，因而得到了广泛的应用.

§10.5.4 二分法

二分法: 通过计算函数导数值来缩短搜索区间. 设初始搜索区间为 $[a_0, b_0] = [a, b]$, 第 k 步时的搜索区间为 $[a_k, b_k]$, 满足 $\varphi'(a_k) \leqslant 0$, $\varphi'(b_k) \geqslant 0$, 取中点

$$c_k = \frac{a_k + b_k}{2}.$$

若 $\varphi'(c_k) \geqslant 0$, 则令 $a_{k+1} = a_k$, $b_{k+1} = c_k$; 否则, 令 $a_{k+1} = c_k$, $b_{k+1} = b_k$, 从而得到新的搜索区间 $[a_{k+1}, b_{k+1}]$. 重复上述步骤, 直到搜索区间的长度小于事先设定的精度为止. 容易看出, 二分法每次迭代都将区间缩短一半, 故二分法的收敛速度也是线性的.

§10.5.4 插值法

插值法: 在搜索区间中不断用低次 (通常不超过三次) 多项式来近似目标函数, 并逐步用插值多项式的极小点来逼近一维搜索问题

$$\varphi(\lambda^*) = \min_{\lambda \geqslant 0} \varphi(\lambda)$$

的极小点. 当函数具有较好的解析性质时, 插值法比直接方法, 譬如黄金分割法和斐波那契法等, 效果更好.

§10.5.4 插值法

考虑利用一点处的函数值、一阶和二阶导数值来构造二次插值函数. 设二次插值多项式为

$$q(x) = ax^2 + bx + c,$$

则 $q(x)$ 在

$$x = -\frac{b}{2a}$$

处取到极小值, 故可作为计算近似极小点的公式.

§10.5.4 插值法

若已知函数在 λ 处的函数值、一阶和二阶导数值, 即 $\varphi(\lambda), \varphi'(\lambda), \varphi''(\lambda)$, 则 $q(x)$ 需满足插值条件

$$
\begin{aligned}
q(\lambda) &= a\lambda^2 + b\lambda + c = \varphi(\lambda), \\
q'(\lambda) &= 2a\lambda + b = \varphi'(\lambda), \\
q''(\lambda) &= 2a = \varphi''(\lambda).
\end{aligned}
$$

解得 $a = \dfrac{\varphi''(\lambda)}{2}, \quad b = \varphi'(\lambda) - \varphi''(\lambda)\lambda$. 从而有迭代计算公式

$$
\lambda_{k+1} = \lambda_k - \frac{\varphi'(\lambda_k)}{\varphi''(\lambda_k)}, \quad k = 0, 1, \cdots,
$$

称为***一点二次插值法***, 或称为***牛顿法***.

可以证明: 当初始值 λ_0 充分靠近 λ^* 时, 由牛顿法产生的迭代序列 $\{\lambda_k\}_{k=1}^{\infty}$ 是收敛的, 且收敛速度是二阶的.

§10.5.4 插值法

牛顿法的优点是速度速度快, 但是需要用到二阶导数值, 计算代价高. 因此, 一个自然的想法是: 用两个点 λ_1, λ_2 处的函数值及其中一个点的导数值 $\varphi'(\lambda_1)$(或 $\varphi'(\lambda_2)$) 来构造二次插值函数. 通过插值条件并求解, 可得计算公式:

$$\lambda_{k+1} = \lambda_k - \frac{(\lambda_k - \lambda_{k-1})\varphi'(\lambda_k)}{2\left[\varphi'(\lambda_k) - \dfrac{\varphi(\lambda_k) - \varphi(\lambda_{k-1})}{\lambda_k - \lambda_{k-1}}\right]}, \quad k = 1, \cdots,$$

称为***两点二次插值法***, 其收敛速度的阶为 $(1+\sqrt{5})/2 \approx 1.618$. 类似地, 还有三点二次插值法, 二点三次插值法等.

§10.5.5 其他方法

一维搜索过程是非线性优化方法的基本组成部分，前述通过求解一维搜索最优化问题的方法，譬如黄金分割法，牛顿法等，统称为**精确一维搜索**，需要花费很大的计算量．特别地，当迭代点远离问题的解时，精确得求解一个一维子问题通常不是十分有效．

因此，提出了**非精确一维搜索**，即选取步长 λ_k 使得目标函数 $f(\mathbf{x})$ 有可接收的下降量 $f(\mathbf{x}_k)-f(\mathbf{x}_k+\lambda_k\mathbf{p}_k)>\delta$．非精确一维搜索包括 Armijo-Goldstein 方法，Wolfe-Powell 方法等．

§10.5.5 其他方法

在非线性优化中, 一个可用来代替一维线搜索的方法是***信赖域方法*** (Trust-Region Methods), 其基本思想是: 先设定一个可信的步长, 然后利用局部 n 维二次模型寻找最优下降方向和步长. 因步长受到使 Taylor 展开式有效的信赖域限制, 故又称***限步长法***. 信赖域方法具有快速的局部收敛性, 又有理想的总体收敛性, 是一个很好的方法.

§10.6.1 最速下降法

不失一般性，**无约束非线性优化**问题可写成

$$\min f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \tag{8}$$

其中 $f(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ 是有 n 个自变量的连续函数.

求解无约束最优化问题最简单的方法是**最速下降法**，它以负梯度方向作为极小化算法的下降方向，故又称为**梯度法**.

§10.6.1 最速下降法

设函数 $f(\mathbf{x})$ 在 $\mathbf{x}_k$ 附件连续可微，且 $\nabla f(\mathbf{x}_k) \neq 0$. 利用 Taylor 展开式

$$f(\mathbf{x}) = f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^{\mathrm{T}} \nabla f(\mathbf{x}_k) + o(\|\mathbf{x} - \mathbf{x}_k\|)$$

可知，若记 $\mathbf{x} - \mathbf{x}_k = \lambda \mathbf{p}_k$, $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, 则满足 $\mathbf{p}_k^{\mathrm{T}} \mathbf{g}_k < 0$ 的方向 $\mathbf{p}_k$ 都是下降方向. 当 λ 的值固定后，若 $\mathbf{p}_k^{\mathrm{T}} \mathbf{g}_k$ 的值越小，即 $-\mathbf{p}_k^{\mathrm{T}} \mathbf{g}_k$ 的值越大，则函数 $f(\mathbf{x})$ 的值下降的越快. 另外，由 Cauchy-Schwartz 不等式知最优化问题

$$\min_{\mathbf{p}_k \in \mathbb{R}^n} \mathbf{p}_k^{\mathrm{T}} \mathbf{g}_k, \quad \text{subject to } \|\mathbf{p}_k\| = \|\mathbf{g}_k\|,$$

当且仅当 $\mathbf{p}_k = -\mathbf{g}_k$ 时，$\mathbf{p}_k^{\mathrm{T}} \mathbf{g}_k$ 的值最小，称 $-\mathbf{g}_k$ 为最速下降方向.

最速下降法的计算步骤如下:

1. 设定初始值 $\mathbf{x}_0 \in \mathbb{R}^n$ 及精度参数 $\varepsilon > 0$, 并令 $k \leftarrow 0$;
2. 计算梯度 $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, 令 $\mathbf{p}_k = -\mathbf{g}_k$;
3. 若 $\|\mathbf{g}_k\| < \varepsilon$, 则算法终止; 否则, 利用一维搜索求步长因子 λ_k, 使得

$$f(\mathbf{x}_k + \lambda_k \mathbf{p}_k) = \min_{\lambda \geqslant 0} f(\mathbf{x}_k + \lambda \mathbf{p}_k);$$

4. 计算 $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \lambda_k \mathbf{p}_k$, 令 $k \leftarrow k+1$, 转步骤 2.

§10.6.1 最速下降法

可以证明: 若 $f(\mathbf{x}) \in C^1$, 在最速下降法中采用精确一维搜索, 则产生的迭代点序列 $\{\mathbf{x}_k\}_{k=1}^{\infty}$ 的每一个聚点都是驻点. 进一步, 若 $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$ 且 $f(\mathbf{x})$ 在 $\mathbf{x}^*$ 的某一邻域内二次连续可微, 存在 $\varepsilon > 0$ 和 $M > m > 0$, 使得当 $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$ 时, 有

$$m\|\mathbf{y}\|^2 \leqslant \mathbf{y}^{\mathrm{T}} G(\mathbf{x})\mathbf{y} \leqslant M\|\mathbf{y}\|^2, \quad \forall \mathbf{y} \in \mathbb{R}^n,$$

其中 $G(\mathbf{x}) = \nabla^2 f(\mathbf{x})$, 则最速下降法至少是线性收敛的.

§10.6.1 最速下降法

由于最速下降方向是函数的局部性质，对许多实际问题，最速下降法并非“最速下降”，而是下降非常缓慢.

数值实验表明，当目标函数的等值线或面接近于一个圆或球时，最速下降法下降较快；而当目标函数的等值线或面接近于一个扁长的椭圆或椭球时，最速下降法开始几步下降较快，后来就出现锯齿现象，下降十分缓慢. 事实上，因采用一维精确搜索，由定理10.6知

$$\mathbf{g}_{k+1}^{\mathrm{T}}\mathbf{p}_k = \mathbf{p}_{k+1}^{\mathrm{T}}\mathbf{p}_k = 0,$$

即在相邻的两个迭代点上，两个搜索方向是相互正交的，这便是产生锯齿状现象的原因. 当接近极小点时，步长越小，前进越慢.

§10.6.2 牛顿法

最速下降法仅用到了函数 $f(\mathbf{x})$ 的一阶局部信息. 若目标函数 $f(\mathbf{x})$ 是二次连续可微的, 则在 $\mathbf{x}_k$ 附件可用 $f(\mathbf{x})$ 的二次 Taylor 展开式

$$q_k(\mathbf{h}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^{\mathrm{T}}\mathbf{h} + \frac{1}{2}\mathbf{h}^{\mathrm{T}}\nabla^2 f(\mathbf{x}_k)\mathbf{h}$$

来近似 $f(\mathbf{x})$, 其中 $\mathbf{h} = \mathbf{x} - \mathbf{x}_k$. 若矩阵 $\nabla^2 f(\mathbf{x}_k)$ 正定, 则 n 维二次函数 $q_k(\mathbf{x})$ 在

$$\frac{\partial q_k(\mathbf{h})}{\partial \mathbf{h}} = \mathbf{0} \Leftrightarrow \nabla^2 f(\mathbf{x}_k)\mathbf{h} = -\nabla f(\mathbf{x}_k) \Leftrightarrow \mathbf{h} = -\left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k)$$

处取到极小值. 因此, 可设计迭代格式

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k), \quad k = 0, 1, \cdots,$$

称为**牛顿迭代法**.

牛顿迭代法是在椭球范数下的最速下降法. 事实上, 对于局部近似函数 $f(\mathbf{x}_k+\mathbf{h}) \approx f(\mathbf{x}_k)+\mathbf{g}_k^{\mathrm{T}}\mathbf{h}$, 考虑极小化问题

$$\min_{\mathbf{h}\in\mathbb{R}^n} \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{h}}{\|\mathbf{h}\|}$$

的最优解, 其中 $\|\cdot\|$ 是某种向量范数. 当采用 l_2 范数时, 即 $\|\mathbf{x}\|^2=\mathbf{x}^{\mathrm{T}}\mathbf{x}$, 可得

$$\mathbf{h}=-\mathbf{g}_k,$$

所得的方法是最速下降法. 当采用椭球范数时, 即 $\|\mathbf{x}\|^2=\mathbf{x}^{\mathrm{T}}G_k\mathbf{x}$, 可得

$$\mathbf{h}=-G_k^{-1}\mathbf{g}_k,$$

所得的方法是牛顿迭代法.

当 $f(\mathbf{x})$ 是正定二次函数时, $q_k(\mathbf{x})=f(\mathbf{x})$, 故牛顿迭代法一步就可达最优解. 而对于非二次函数, 牛顿迭代法不能保证经过有限次迭代求得最优解, 但由于目标函数在极小点附件可近似于二次函数, 故当初始点靠近极小点时, 牛顿迭代法的收敛速度一般是快的.

可以证明: 若 $f(\mathbf{x})\in C^2$, $\mathbf{x}_k$ 充分靠近 $\mathbf{x}^*$, 如果 $\nabla^2 f(\mathbf{x}^*)$ 正定, 且 $G(x)=\nabla^2 f(\mathbf{x})$ 的每一个元素的函数满足 Lipschitz 条件, 则牛顿迭代法所得的序列 $\{\mathbf{x}_k\}_{k=1}^{\infty}$ 收敛于 $\mathbf{x}^*$, 且具有二阶的收敛速度.

§10.6.2 牛顿法

当初始点远离最优解时, G_k 不一定正定, 牛顿方向不一定是下降方向, 收敛性不能保证, 这说明恒取步长为 1 的牛顿迭代法是不合适的. 因此, 应该引入某种一维搜索来确定步长因子, 这种方法称为***带步长因子的牛顿法***, 计算步骤如下:

1. 设定初始值 $\mathbf{x}_0 \in \mathbb{R}^n$ 及精度参数 $\varepsilon > 0$, 并令 $k \leftarrow 0$;
2. 计算梯度 $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$. 若 $\|\mathbf{g}_k\| < \varepsilon$, 则算法终止; 否则, 进入下一步;
3. 计算矩阵 G_k, 并求解线性方程组 $G_k\mathbf{p} = -\mathbf{g}_k$ 得牛顿方向 $\mathbf{p}_k$;
4. 利用一维搜索求步长因子 λ_k, 使得

$$f(\mathbf{x}_k + \lambda_k\mathbf{p}_k) = \min_{\lambda \geqslant 0} f(\mathbf{x}_k + \lambda\mathbf{p}_k);$$

5. 计算 $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \lambda_k\mathbf{p}_k$, 令 $k \leftarrow k+1$, 转步骤 2.

§10.6.3 共轭方向法

共轭方向法：一种介于最速下降法与牛顿法之间的方法，它仅需利用一阶导数信息，但克服了最速下降法收敛慢的缺点，又避免了存储和计算牛顿法所需要的二阶导数信息.

共轭方向法是从研究二次函数的极小化问题产生的，是一种迭代求解大型稀疏正定线性方程组的有效方法，同时它还可以推广到处理非二次函数的极小化问题.

§10.6.3 共轭方向法

定义 10.16

设 G 是 n 阶正定方阵, $\mathbf{p}_1$, $\mathbf{p}_2$ 是 n 维非零向量, 如果 $\mathbf{p}_1^{\mathrm{T}} G \mathbf{p}_2 = 0$, 则称 $\mathbf{p}_1$ 与 $\mathbf{p}_2$ 是*G-共轭的*. 类似地, 设 $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$ 是一组 n 维非零向量, 如果

$$\mathbf{p}_i^{\mathrm{T}} G \mathbf{p}_j = 0, \quad \forall 1 \leqslant i \neq j \leqslant n,$$

则称 $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$ 是 G-共轭的.

显然, 如果 $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$ 是 G-共轭的, 则它们是线性无关的. 当 $G = I$ 时, G-共轭性就是通常的正交性.

一般共轭方向法的计算步骤如下:

1. 设定初始值 $\mathbf{x}_0 \in \mathbb{R}^n$ 及初始方向 $\mathbf{g}_0$, 计算 $\mathbf{p}_0$, 使得 $\mathbf{p}_0^{\mathrm{T}}\mathbf{g}_0 < 0$, 并令 $k \leftarrow 0$;
2. 计算 λ_k 和 $\mathbf{x}_{k+1}$, 使得

$$f(\mathbf{x}_k + \lambda_k \mathbf{p}_k) = \min_{\lambda \geqslant 0} f(\mathbf{x}_k + \lambda \mathbf{p}_k),$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{p}_k;$$

3. 计算 $\mathbf{p}_{k+1}$, 使得 $\mathbf{p}_{k+1}^{\mathrm{T}} G \mathbf{p}_j = 0$, 其中 $j = 0, 1, 2, \cdots, k$;
4. 令 $k \leftarrow k+1$, 转步骤 2.

§10.6.3 共轭方向法

共轭方向法：只要执行精确一维搜索，就具有二次终止性.

定理 10.7

(共轭方向基本定理) 设 $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}G\mathbf{x} + \mathbf{g}^{\mathrm{T}}\mathbf{x} + c$ 是正定二次函数，其中 G 是 n 阶正定方阵，$\mathbf{g}$ 是 n 维向量，c 是常数，则共轭方向法至多经过 n 步精确线性搜索终止；且每一个 $\mathbf{x}_{k+1}$ 都是 $f(\mathbf{x})$ 在点 $\mathbf{x}_0$ 和方向 $\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_k$ 所张成的线性流形

$$\left\{\mathbf{x} \,\middle|\, \mathbf{x} = \mathbf{x}_0 + \sum_{j=0}^{k} \lambda_j \mathbf{p}_j, \quad \forall \lambda_j \in \mathbb{R}\right\}$$

中的极小点.

在精确线性搜索的条件下，利用 $\mathbf{g}_{k+1} - \mathbf{g}_k = G(\mathbf{x}_{k+1} - \mathbf{x}_k) = \lambda_k G\mathbf{p}_k$ 可知，共轭方向法的梯度 $\mathbf{g}_{k+1}$ 满足

$$\mathbf{g}_{k+1}^{\mathrm{T}}\mathbf{p}_j = 0, \quad j = 0, 1, \cdots, k.$$

§10.6.3 共轭方向法

若记 $\mathbf{x}^* = \mathbf{x}_0 + \sum_{i=0}^{n-1} y_i^* \mathbf{p}_i$, $\mathbf{x} = \mathbf{x}_0 + \sum_{i=0}^{n-1} y_i \mathbf{p}_i$, 则二次函数 $f(\mathbf{x})$ 可改写成

$$q(\mathbf{y}) = \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^{\mathrm{T}} G(\mathbf{x}-\mathbf{x}^*) + \tilde{c} = \frac{1}{2}(\mathbf{y}-\mathbf{y}^*)^{\mathrm{T}} S^{\mathrm{T}} G S(\mathbf{y}-\mathbf{y}^*) + \tilde{c},$$

其中 $S = (\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_{n-1})$. 利用 G-共轭性条件, 得

$$q(\mathbf{y}) = \frac{1}{2}\sum_{i=0}^{n-1}(y_i - y_i^*)^2 d_{ii},$$

其中 $d_{ii} = \mathbf{p}_i^{\mathrm{T}} G \mathbf{p}_i$.

共轭性意味着存在一个恰当的坐标变换 S, 使得 G 在新的坐标系下是一个对角矩阵 $S^T G S$, 新的变量在二次函数中是相互分离的. 于是, 一个共轭方向法是在新坐标系中的一个交替变量法.

§10.6.3 共轭方向法

共轭梯度法是最著名的共轭方向法，它使得最速下降方向具有共轭性，从而提高算法的有效性和可靠性.

由于解正定线性方程组 $A\mathbf{x} = \mathbf{b}$ 等价于极小化一个正定二次函数

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x},$$

共轭梯度法首先是作为解线性方程组的方法被提出来的.

§10.6.3 共轭方向法

考虑正定二次函数

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}} G\mathbf{x} + \mathbf{g}^{\mathrm{T}}\mathbf{x} + c,$$

$f(\mathbf{x})$ 的梯度为 $\nabla f(\mathbf{x}) = G\mathbf{x} + \mathbf{g}$. 共轭梯度法有一个非常特殊的性质: 在构造共轭方向时, $\mathbf{p}_k$ 仅依赖于 $\mathbf{p}_{k-1}$. 这意味着在实现算法时, 可以减少存储量和计算时间, 是一个很好的性质. 另外, 在搜索方向 $\mathbf{p}_k$ 上, 希望函数值是下降的, 故一个自然的选择是

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_{k-1}\mathbf{p}_{k-1},$$

其中 $\mathbf{g}_k = \nabla f(\mathbf{x_k}) = G\mathbf{x_k} + \mathbf{g}$, β_{k-1} 是待定系数.

因 $\mathbf{p}_k$ 与 $\mathbf{p}_{k-1}$ 需满足 G-共轭条件，上式两端左乘以 $\mathbf{p}_{k-1}^{\mathrm{T}}G$，故有

$$\beta_{k-1} = \frac{\mathbf{g}_k^{\mathrm{T}} G \mathbf{p}_{k-1}}{\mathbf{p}_{k-1}^{\mathrm{T}} G \mathbf{p}_{k-1}}, \quad k = 1, 2, \cdots, n-1.$$

对于 $\mathbf{p}_0$，直接取 $\mathbf{p}_0 = -\mathbf{g}_0$. 可以证明 $\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_{n-1}$ 是 G-共轭的，且每一次迭代所产生的搜索方向 $\mathbf{p}_k$ 和梯度方向 $\mathbf{g}_k$ 都包含于

$$\mathcal{K}(\mathbf{g}_0; k) \triangleq \operatorname{span}\{\mathbf{g}_0, G\mathbf{g}_0, \cdots, G^k \mathbf{g}_0\},$$

称 $\mathcal{K}(\mathbf{g}_0; k)$ 为 $\mathbf{g}_0$ 的 k 次 Krylov 子空间.

§10.6.3 共轭方向法

定理 10.8

设 $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}G\mathbf{x} + \mathbf{g}^{\mathrm{T}}\mathbf{x} + c$ 是正定二次函数，其中 G 是 n 阶正定方阵，$\mathbf{g}$ 是 n 维向量，c 是常数，则采用精确线性搜索的共轭梯度法经过 $m \leqslant n$ 步终止，且对任意的 $k \leqslant m$ 有

$$\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_i = 0, \quad i = 0, 1, \cdots, k-1, \tag{9}$$

$$\mathbf{p}_k^{\mathrm{T}}G\mathbf{p}_i = 0, \quad i = 0, 1, \cdots, k-1, \tag{10}$$

$$\mathbf{p}_i^{\mathrm{T}}\mathbf{g}_i = -\mathbf{g}_i^{\mathrm{T}}\mathbf{g}_i, \quad i = 0, 1, \cdots, k-1, \tag{11}$$

$$\mathrm{span}\left\{\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_k\right\} = \mathcal{K}(\mathbf{g}_0; k), \tag{12}$$

$$\mathrm{span}\left\{\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_k\right\} = \mathcal{K}(\mathbf{g}_0; k). \tag{13}$$

注意：$\mathbf{p}_0$ 必须取 $-\mathbf{g}_0$，否则上述定理是不成立的.

§10.6.3 共轭方向法

式 (9) 表明梯度向量序列 $\{\mathbf{g}_i\}$ 是相互正交的，而式 (10) 表明搜索方向序列 $\{\mathbf{p}_i\}$ 才是 G-共轭的. 由式 (11) 知，共轭梯度法的每一步都是值下降的，精确一维搜索的步长 λ_k 可写成:

$$\lambda_k = -\frac{\mathbf{p}_k^{\mathrm{T}}\mathbf{g}_k}{\mathbf{p}_k^{\mathrm{T}}G\mathbf{p}_k} = \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k}{\mathbf{p}_k^{\mathrm{T}}G\mathbf{p}_k}.$$

利用 $G\mathbf{p}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)/\lambda_k$ 及式 (9) 和 (11), 可得计算搜索方向的步长 β_k 几个常用公式

$$\begin{aligned}
\beta_k &= \frac{\mathbf{g}_{k+1}^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{p}_k^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)} \text{ (Crowder-Wolfe 公式)} \\
&= \frac{\mathbf{g}_{k+1}^{\mathrm{T}}\mathbf{g}_{k+1}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k} \text{ (Fletcher-Reeves 公式)} \\
&= \frac{\mathbf{g}_{k+1}^{\mathrm{T}}(\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k} \text{ (Polak-Ribiere-Polyak 公式)} \\
&= -\frac{\mathbf{g}_{k+1}^{\mathrm{T}}\mathbf{g}_{k+1}}{\mathbf{p}_k^{\mathrm{T}}\mathbf{g}_k} \text{ (Dixon 公式).}
\end{aligned}$$

共轭梯度法仅比最速下降法稍微复杂一点, 但却具有二次终止性, 所以共轭梯度法是一个很有效的方法.

§10.6.3 共轭方向法

共轭梯度法可推广至一般的目标函数 $f(\mathbf{x})$, 此时, 步长 λ_k 可通过精确一维搜索或非精确一维搜索得到, 而 β_k 的计算公式保持不变. 但是, 需要注意的是, n 步以后共轭梯度法所产生的搜索方向 $\mathbf{p}_n$ 不再与 $\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_{n-1}$ 满足 G-共轭条件, 故需重新取最速下降方向作为搜索方向, 即

$$\mathbf{p}_{cn} = -\mathbf{g}_{cn}, \quad c = 1, 2, \cdots,$$

这种方法称为***再开始共轭梯度法***.

§10.6.3 共轭方向法

再开始共轭梯度法的计算步骤如下:

1. 设定初始值 $\mathbf{x}_0 \in \mathbb{R}^n$ 及精度参数 $\varepsilon > 0$, 计算 $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, 令 $k \leftarrow 0$;
2. 若 $\|\mathbf{g}_0\| < \varepsilon$, 则算法终止; 否则, 令 $\mathbf{p}_0 \leftarrow -\mathbf{g}_0$;
3. 利用一维搜索求步长因子 λ_k, 使得
$$f(\mathbf{x}_k + \lambda_k \mathbf{p}_k) = \min_{\lambda \geqslant 0} f(\mathbf{x}_k + \lambda \mathbf{p}_k),$$
令 $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \lambda_k \mathbf{p}_k$, 计算 $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$;
4. 若 $\|\mathbf{g}_{k+1}\| < \varepsilon$, 则算法终止; 否则, 进入下一步;
5. 若 $k = n-1$, 令 $\mathbf{x}_0 \leftarrow \mathbf{x}_{k+1}$, 转步骤 1; 否则, 进入下一步;
6. 计算 $\beta_k = \mathbf{g}_{k+1}^{\mathrm{T}} \mathbf{g}_{k+1} / \mathbf{g}_k^{\mathrm{T}} \mathbf{g}_k$, 令 $\mathbf{p}_{k+1} \leftarrow -\mathbf{g}_{k+1} + \beta_k \mathbf{p}_k$, $k \leftarrow k+1$;
7. 若 $\mathbf{p}_k^{\mathrm{T}} \mathbf{g}_k > 0$, 令 $\mathbf{x}_0 \leftarrow \mathbf{x}_k$, 转步骤 1; 否则, 转步骤 3.

当步长因子 λ_k 也可以采用非精确一维搜索时, β_k 的几种计算公式的效果是不同的.

共轭梯度法在一定条件下是收敛的. 可以证明: 若 $f(\mathbf{x})$ 在有界水平集 $L = \{\mathbf{x} \in \mathbb{R}^n \,|\, f(\mathbf{x}) \leqslant f(\mathbf{x}_0)\}$ 上连续可微, 则采用 Fletcher-Reeves 公式和精确一维搜索的共轭梯度法产生的序列 $\{\mathbf{x}_k\}$ 至少有一个聚点是驻点, 即

(1) 当 $\{\mathbf{x}_k\}$ 是有穷点列时, 其最后一个点 $\mathbf{x}^*$ 是 $f(\mathbf{x})$ 的驻点;
(2) 当 $\{\mathbf{x}_k\}$ 是无穷点列时, 它必有极限点, 且其任一极限点都是 $f(\mathbf{x})$ 的驻点.

采用其他公式和搜索策略也有类似的结论.

当 $f(\mathbf{x})$ 是正定二次函时，采用精确一维搜索的共轭梯度法至多在 n 次迭代后终止．而对于一般的非线性函数 $f(\mathbf{x})$，若将 n 次迭代视为一次大的迭代，则共轭梯度法应该与牛顿法有类似的收敛速度．可以证明：设 $f(\mathbf{x}) \in C^3$，且存在常数 $m, M > 0$，使得

$$m\|\mathbf{y}\|^2 \leqslant \mathbf{y}^{\mathrm{T}}\nabla^2 f(\mathbf{x})\mathbf{y} \leqslant M\|\mathbf{y}\|^2, \quad \forall \mathbf{y} \in \mathbb{R}^n, \mathbf{x} \in L,$$

其中 $L = \{\mathbf{x} \in \mathbb{R}^n \,|\, f(\mathbf{x}) \leqslant f(\mathbf{x}_0)\}$ 是有界水平集，则采用 Fletcher-Reeves 公式和精确一维搜索的再开始共轭梯度法产生的序列 $\{\mathbf{x}_k\}$ 是 n 步二阶收敛的，即存在常数 $c > 0$，使得

$$\lim_{k\to\infty} \sup \frac{\|\mathbf{x}_{kn+n} - \mathbf{x}^*\|}{\|\mathbf{x}_{kn} - \mathbf{x}^*\|^2} \leqslant c < \infty.$$

§10.6.4 拟牛顿法

拟牛顿法: 利用目标函数 $f(\mathbf{x})$ 及一阶导数 $\nabla f(\mathbf{x})$ 的信息来构造近似的 Hessian 矩阵, 从而实现加快收敛速度的目标.

设 $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ 是二次连续可微的函数, 则 $f(\mathbf{x})$ 在 $\mathbf{x}_{k+1}$ 附件的二次近似函数为

$$f(\mathbf{x}) \approx f(\mathbf{x}_{k+1}) + \mathbf{g}_{k+1}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}_{k+1}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_{k+1})^{\mathrm{T}} G_{k+1}(\mathbf{x} - \mathbf{x}_{k+1}),$$

其中 $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$, $G_{k+1} = \nabla^2 f(\mathbf{x}_{k+1})$, 从而有

$$\nabla f(\mathbf{x}) \approx \mathbf{g}_{k+1} + G_{k+1}(\mathbf{x} - \mathbf{x}_{k+1}).$$

令 $\mathbf{x} = \mathbf{x}_k$, 记 $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, 则

$$G_{k+1}^{-1}\mathbf{y}_k \approx \mathbf{s}_k.$$

显然, 当 $f(\mathbf{x})$ 是二次函数时, 上述近似关系取等号.

§10.6.4 拟牛顿法

在拟牛顿法中，要求构造出来的 Hessian 矩阵的逆近似 H_{k+1} 满足这种关系，即

$$H_{k+1}\mathbf{y}_k = \mathbf{s}_k, \tag{14}$$

称为***拟牛顿条件***或***拟牛顿方程***. 若记 $B_{k+1} = H_{k+1}^{-1}$，则上述条件也可写为

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k,$$

是关于 Hessian 矩阵的拟牛顿条件.

§10.6.4 拟牛顿法

一般拟牛顿法的计算步骤如下:

1. 设定初始值 $\mathbf{x}_0 \in \mathbb{R}^n$ 及精度参数 $\varepsilon > 0$, $H_0 \in \mathbb{R}^{n\times n}$, 并令 $k \leftarrow 0$;
2. 计算梯度 $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, 若 $\|\mathbf{g}_k\| < \varepsilon$, 则算法终止; 否则, 计算 $\mathbf{p}_k = -H_k\mathbf{g}_k$;
3. 利用一维搜索求步长因子 λ_k, 使得
$$f(\mathbf{x}_k + \lambda_k\mathbf{p}_k) = \min_{\lambda\geqslant 0} f(\mathbf{x}_k + \lambda\mathbf{p}_k);$$
4. 校正 H_k 产生 H_{k+1}, 使得拟牛顿条件 (14) 成立;
5. 计算 $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \lambda_k\mathbf{p}_k$, 令 $k \leftarrow k+1$, 转步骤 2.

类似地, 拟牛顿法也可采用近似 Hessian 矩阵 B_k 进行. 在实际应用中, 初始矩阵 H_0 可取单位阵, 此时, 拟牛顿法的第一次迭代等价于一个最速下降迭代.

§10.6.4 拟牛顿法

拟牛顿法是在椭球范数 $\|\cdot\|_{H_k^{-1}}$ 下的最速下降法，搜索方向

$$\mathbf{p}_k = -H_k\mathbf{g}_k$$

是 $f(\mathbf{x})$ 从 $\mathbf{x}_k$ 点出发的最速下降方向．因在每一次迭代中，度量矩阵 H_k^{-1} 都是变化的，故方法也称为*变尺度法*．

与牛顿法相比，拟牛顿法有以下优点：

(1) 仅需一阶导数信息；
(2) H_k 保持正定，使得方法具有下降性质；
(3) 每次迭代需要 $O(n^2)$ 次乘法运算．

§10.6.4 拟牛顿法

如何校正 H_k 产生 H_{k+1}, 使得拟牛顿条件 (14) 成立, 并保持正定性. 设 H_{k+1} 是 H_k 通过对称秩二校正得到的, 即

$$H_{k+1} = H_k + a\mathbf{u}\mathbf{u}^{\mathrm{T}} + b\mathbf{v}\mathbf{v}^{\mathrm{T}},$$

其中 $a, b \in \mathbb{R}$, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n\times 1}$. 代入拟牛顿条件 (14) 得

$$H_k\mathbf{y}_k + a\mathbf{u}\mathbf{u}^{\mathrm{T}}\mathbf{y}_k + b\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{y}_k = \mathbf{s}_k.$$

显然, 向量 $\mathbf{u}$ 和 $\mathbf{v}$ 不能由上式唯一确定, 但是 $\mathbf{u}$ 和 $\mathbf{v}$ 可取为

$$\mathbf{u} = \mathbf{s}_k, \quad \mathbf{v} = H_k\mathbf{y}_k,$$

这时, 只要 a 和 b 满足 $a\mathbf{u}^{\mathrm{T}}\mathbf{y}_k = 1$, $b\mathbf{v}^{\mathrm{T}}\mathbf{y}_k = -1$, 拟牛顿条件 (14) 就成立了. 解得

$$H_{k+1} = H_k + \frac{\mathbf{s}_k\mathbf{s}_k^{\mathrm{T}}}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k} - \frac{H_k\mathbf{y}_k\mathbf{y}_k^{\mathrm{T}}H_k}{\mathbf{y}_k^{\mathrm{T}}H_k\mathbf{y}_k}, \tag{15}$$

称为 **DFP 校正公式**, 是由 Davidon, Fletcher 和 Powell 发展出来的.

§10.6.4 拟牛顿法

DFP 校正公式具有很多重要的性质:

(1) 校正保持正定性, 故下降性质成立;
(2) 每次迭代需要 $3n^2 + O(n)$ 次乘法运算;
(3) 具有超线性的收敛速度;
(4) 对于凸函数, 当采用精确一维搜索时, 方法具有总体收敛性.

§10.6.4 拟牛顿法

利用 $H_{k+1} \longleftrightarrow B_{k+1}$, $\mathbf{s}_k \longleftrightarrow \mathbf{y}_k$ 之间的对偶关系及矩阵逆的秩一校正公式, 可得

$$H_{k+1} = H_k + \left(1 + \frac{\mathbf{y}_k^{\mathrm{T}} H_k \mathbf{y}_k}{\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k}\right) \frac{\mathbf{s}_k \mathbf{s}_k^{\mathrm{T}}}{\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k} - \frac{\mathbf{s}_k \mathbf{y}_k^{\mathrm{T}} H_k + H_k \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}}}{\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k}, \qquad (16)$$

称为 **BFGS 校正公式**, 是由 Broyden, Fletcher, Goldfarb 和 Shanno 发展出来的. BFGS 校正公式具备与 DFP 校正公式相同的性质, 并且对于凸函数, 当采用非精确一维搜索时, 仍具有总体收敛性. 在实际应用中, BFGS 校正公式的结果也优于 DFP 校正公式, 是最常用的拟牛顿方法之一.

§10.6.4 拟牛顿法

一般地，当初始点靠近极小点时，拟牛顿迭代法在一定条件下是线性收敛的，很多时候甚至是超线性收敛的.

对于凸函数，拟牛顿法还有整体收敛性. 可以证明：当 $f(\mathbf{x})$ 是一致凸的二阶连续可微函数时，采用精确一维搜索和 DFP 校正公式的拟牛顿法整体收敛；当 $f(\mathbf{x})$ 是凸的二阶连续可微函数时，采用非精确一维搜索的 Wolfe-Powell 准则和 BFGS 校正公式的拟牛顿法整体收敛.

§10.6.5 随机梯度下降法

在许多实际应用中，譬如深度学习，压缩感知，低秩矩阵填充等，优化变量的个数可能会很大，如几十万甚至更多，这时即使采用梯度下降法，每一步迭代所需的计算量也是相当可观的. 因此，人们提出了**随机梯度下降法**，基本思想是：梯度向量可以视为期望，而期望可以使用小规模的样本来近似估计. 设优化的目标函数为 $f(\mathbf{x})$，优化变量 $\mathbf{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，先任取它的一个有 m 个元素的子集 $\{x_{i1},x_{i2},\cdots,x_{im}\}$，$m$ 远小于 n，然后计算 $f(\mathbf{x})$ 关于变量 $x_{i1},x_{i2},\cdots,x_{im}$ 的梯度并乘以一个称为学习率的常数 r，其他变量视为常数，执行一次梯度下降. 重复以上步骤，直至算法满足收敛准则.

Thanks for your attention!