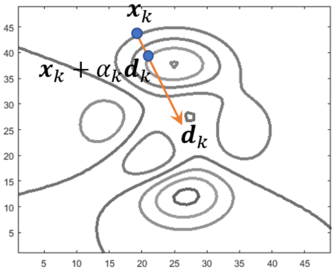
第3章 一维线性搜索方法

线性搜索

在迭代优化算法中，若当前迭代点为$\boldsymbol{x}_k$，该点处的搜索方向为$\boldsymbol{d}_k$，确定步长α_k来获得更优的一下个迭代点$\boldsymbol{x}_{k+1}$的问题称为**线性搜索问题**。



更一般地，可以将**线性搜索**理解为关于单变量函数的优化方法，也称为**一维搜索**，是多变量函数最优化方法的基础。

本章主要学习内容

3-1 精确线性搜索

精确线性搜索

从迭代点$\boldsymbol{x}_k$出发，沿着搜索方向$\boldsymbol{d}_k$确定关于步长函数：

$$\varphi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k),$$

并通过极小化该函数获得**最优步长因子**α_k，即

$$\varphi(\alpha_k) = \min_{\alpha>0} \varphi(\alpha) = \min_{\alpha>0} f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k). \tag{3.1.1}$$

以上获得步长的方法称为精确线性搜索。

若f为一阶连续可微，根据一阶最优性定理有:

$$\varphi'(\alpha) = \nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)^T \boldsymbol{d}_k = 0$$

解以上关于α的方程来确定最优步长：

精确线性搜索

此时显然有:

$$\nabla f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k)^T \boldsymbol{d}_k = \boldsymbol{g}_{k+1}^T \boldsymbol{d}_k = 0.$$

当前迭代点处梯度方向$\boldsymbol{g}_{k+1}$与前一个迭代点搜索方向$\boldsymbol{d}_k$互相垂直。

算法3.1.1 精确线性搜索算法

步1 给出$\boldsymbol{x}_0 \in \mathbb{R}^n$, $0 \le \varepsilon \ll 1$, $k := 0$.

步2 计算$\nabla f(\boldsymbol{x}_k)$.如果$\|\nabla f(\boldsymbol{x}_k)\| \le \varepsilon$, 停止.

步3 计算下降方向$\boldsymbol{d}_k$.

步4 计算步长因子α_k,使得:

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) = \min_{\alpha \ge 0} f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k).$$

步5 令$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$,转步2.

函数值下降估计

定理3.1.1 单步迭代函数值下降界估计.

设$\boldsymbol{d}_k$是下降方向，α_k是精确线性搜索的步长因子。若存在常数$M > 0$，使得对所有$\alpha > 0$,

$$\|\nabla^2 f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)\| \le M, \quad \forall k, \tag{3.1.2}$$

则

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) \ge \frac{1}{2M}\|\boldsymbol{g}_k\|^2 \cos\theta_k. \tag{3.1.3}$$

其中：$\theta_k =< \boldsymbol{d}_k, -\boldsymbol{g}_k >$表示向量$\boldsymbol{d}_k$与$-\boldsymbol{g}_k$之间的夹角，即

$$\cos\theta_k = -\frac{\boldsymbol{d}_k^T \boldsymbol{g}_k}{\|\boldsymbol{d}_k\|\|\boldsymbol{g}_k\|}. \tag{3.1.4}$$

定理3.1.1证明

证明： 由假设可知对任意$\alpha > 0$有：

$$\begin{aligned} f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) &= f(\boldsymbol{x}_k) + \alpha \boldsymbol{g}_k^T \boldsymbol{d}_k + \frac{1}{2}\alpha^2 \boldsymbol{d}_k^T \nabla^2 f(\boldsymbol{x}_k + \theta\alpha \boldsymbol{d}_k)\boldsymbol{d}_k, \ (0 < \theta < 1) \\ &\le f(\boldsymbol{x}_k) + \alpha \boldsymbol{g}_k^T \boldsymbol{d}_k + \frac{1}{2}\alpha^2 M \|\boldsymbol{d}_k\|^2. \end{aligned}$$

令$\bar{\alpha} = -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{M\|\boldsymbol{d}_k\|^2}$. 由于$\alpha_k$是精确线性搜索步长因子，故有:

$$\begin{aligned} f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) &\ge f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \bar{\alpha}\boldsymbol{d}_k) \\ &\ge -\bar{\alpha}\boldsymbol{g}_k^T \boldsymbol{d}_k - \frac{1}{2}\bar{\alpha}^2 M \|\boldsymbol{d}_k\|^2 \\ &= \frac{1}{2}\frac{(\boldsymbol{g}_k^T \boldsymbol{d}_k)^2}{M\|\boldsymbol{d}_k\|^2} = \frac{1}{2M}\|\boldsymbol{g}_k\|^2 \frac{(\boldsymbol{g}_k^T \boldsymbol{d}_k)^2}{\|\boldsymbol{g}_k\|^2\|\boldsymbol{d}_k\|^2} \\ &= \frac{1}{2M}\|\boldsymbol{g}_k\|^2 \cos^2\theta_k. \end{aligned}$$

结论得证。 ■

定理3.1.2 精确线性收敛性.

设梯度$\boldsymbol{g}(\boldsymbol{x})$在水平集$L=\{\boldsymbol{x}\in\mathbb{R}^n|f(\boldsymbol{x})\leq f(\boldsymbol{x}_0)\}$上存在且一致连续，采用精确线性搜索的**算法**3.1.1产生的方向$\boldsymbol{d}_k$与$-\boldsymbol{g}_k$的夹角θ_k满足:

$$\theta_k\leq\frac{\pi}{2}-\mu,\ 对某个\ \mu>0,$$

则或者对某个有限的N有$\boldsymbol{g}_N=0$，或者$f(\boldsymbol{x}_k)\to-\infty$,或者$\boldsymbol{g}_k\to0$.

证明： 假定对所有k，$\boldsymbol{g}_k\neq0$，$f(\boldsymbol{x}_k)$下有界。由于$\{f(\boldsymbol{x}_k)\}$单调下降，故极限存在，因而

$$f(\boldsymbol{x}_k)-f(\boldsymbol{x}_{k+1})\to0. \tag{a}$$

反证法 假定$\boldsymbol{g}_k\to0$不成立，则存在常数$\varepsilon>0$和一个子序列使得$\|\boldsymbol{g}_k\|\geq\varepsilon$. 从而

$$-\frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|}=\|\boldsymbol{g}_k\|\cos\theta_k\geq\varepsilon\sin\mu\triangleq\varepsilon_1 \tag{b}$$

证明续： 又

$$f(\boldsymbol{x}_k+\alpha\boldsymbol{d}_k)=f(\boldsymbol{x}_k)+\alpha\boldsymbol{g}(\boldsymbol{\xi}_k)^T\boldsymbol{d}_k=f(\boldsymbol{x}_k)+\alpha\boldsymbol{g}_k^T\boldsymbol{d}_k+\alpha[\boldsymbol{g}(\boldsymbol{\xi}_k)-\boldsymbol{g}_k]^T\boldsymbol{d}_k$$

$$\leq f(\boldsymbol{x}_k)+\alpha\|\boldsymbol{d}_k\|\left(\frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|}+\|\boldsymbol{g}(\boldsymbol{\xi}_k)-\boldsymbol{g}_k\|\right), \qquad \text{(c)}$$

其中$\boldsymbol{\xi}_k$在$\boldsymbol{x}_k$与$\boldsymbol{x}_k+\alpha\boldsymbol{d}_k$之间。

由于$\boldsymbol{g}$在水平集L上一致连续，故存在$\bar{\alpha}$, 使得当$0\leq\alpha\|\boldsymbol{d}_k\|\leq\bar{\alpha}$时，

$$\|g(\boldsymbol{\xi}_k)-\boldsymbol{g}_k\|\leq\frac{1}{2}\varepsilon_1. \qquad \text{(d)}$$

依次利用前面(b)(c)(d)三式得:

$$f(\boldsymbol{x}_k+\bar{\alpha}\frac{\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|})\leq f(\boldsymbol{x}_k)+\bar{\alpha}\left(\frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|}+\frac{1}{2}\varepsilon_1\right)\leq f(\boldsymbol{x}_k)-\frac{1}{2}\bar{\alpha}\varepsilon_1.$$

定理3.1.2证明

证明续： 从而由精确线性搜索可得：

$$f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k + \bar{\alpha}\frac{\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|}) \le f(\boldsymbol{x}_k) - \frac{1}{2}\bar{\alpha}\varepsilon_1.$$

这与(a)矛盾。从而有$\boldsymbol{g}_k \to 0$. 结论得证。 ■

线性搜索迭代算法

线性搜索迭代算法分成两个阶段:

- 第一阶段: 确定包含理想的步长因子(或问题最优解)的**初始搜索区间**;
- 第二阶段: 采用某种**分割技术或插值方法**缩小这个区间。

试探法/分割法(无导数): 0.618法与Fibonacci法

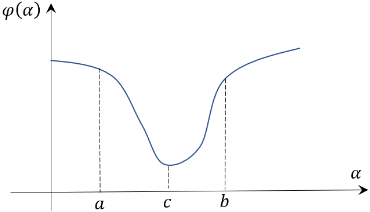
- **基本思想**: 通过取试探点和比较函数值来确定极小点所在近似区间。仅需计算函数值且不涉及导数，又称直接法。
- **适用情形**: 适用于非光滑及导数表达式复杂或写不出的种种情形。要求所考虑区间上的目标函数是单峰函数。

初始搜索区间确定

进退法: 从一点出发，按一定步长，试图确定出函数值呈现“高–底–高”的三点，即

$$\varphi(a) \geq \varphi(c) \leq \varphi(b),$$

这里$a \leq c \leq b$.



进退法(1)-函数值

具体做法1： 初始点$\alpha_0 > 0$，初始步长$h_0 > 0$

- **情形1：**

$$\varphi(\alpha_0 + h_0) \leq \varphi(\alpha_0),$$

下一步试探点$\alpha_1 = \alpha_0 + h_0$，加大步长，再向前搜索，直到目标函数上升为止。

- **情形2:**

$$\varphi(\alpha_0 + h_0) > \varphi(\alpha_0),$$

则下一试探点仍以α_0为出发点，沿反方向同样搜索，直到目标函数上升就停止。

进退法(2)-导数值

具体做法2： 在包含极小点α^*的区间$[a,b]$的端点处，

$$\varphi'(a) \leq 0, \varphi'(b) \geq 0.$$

给定步长$h \geq 0$, 取初始点$\alpha_0 \geq 0$.

- 若$\varphi'(\alpha_0) \leq 0$, 则取

$$\alpha_1 = \alpha_0 + h,$$

- 若$\varphi'(\alpha_0) \geq 0$, 则取

$$\alpha_1 = \alpha_0 - h.$$

其余过程与上述方法类似。

进退法步骤

算法3.1.2 – 进退法

步1 选取初始数据。$\alpha_0 \in [0,\infty)$, $h_0 > 0$, 加倍系数$t > 1$ (e.g.$t = 2$), 计算$\varphi(\alpha_0)$, $k := 0$.

步2 比较目标函数值.令$\alpha_{k+1} = \alpha_k + h_k$, 计算$\varphi_{k+1} = \varphi(\alpha_{k+1})$, 若$\varphi_{k+1} < \varphi_k$, 转步3, 否则转步4.

步3 加大搜索步长. 令$h_{k+1} := th_k$, $\alpha := \alpha_k$, $\alpha_k := \alpha_{k+1}$, $\varphi_k := \varphi_{k+1}$, $k := k+1$, 转步2.

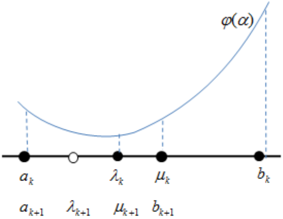
步4 反向搜索. 若$k = 0$, 转换搜索方向, 令$h_k := -h_k$, $\alpha := \alpha_{k+1}$, 转步2; 否则,停止迭代,令

$$a = \min\{\alpha, \alpha_{k+1}\},\ b = \max\{\alpha, \alpha_{k+1}\},$$

输出$[a,b]$, 停止.

0.618法

设包含极小点α^*的初始搜索区间为$[a,b]$, 设$\varphi(\alpha) = f(\boldsymbol{x}_k+\alpha\boldsymbol{d}_k)$, 在$[a,b]$上是单峰函数。



0.618法(也称为黄金分割法)的**基本思想**: 在搜索区间$[a,b]$上选取两个对称点λ,μ且$\lambda<\mu$, 比较两点处的函数值$\varphi(\lambda)$和$\varphi(\mu)$来决定删除左半区间$[a,\lambda)$或者右半区间$(\mu,b]$. 删除后的区间长度是原区间长度的0.618倍。新区间保留两个对称点中的一点，再选一个对称点，比较两个新对称点处的函数值. 重复这个过程, 最后确定出极小点α^*。

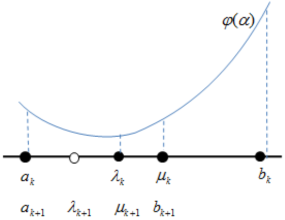
0.618法

记$a_0 = a, b_0 = b$，区间$[a_0, b_0]$经k次缩短后变为$[a_k, b_k]$．选取两个试探点λ_k和μ_k需满足下列条件:

$$b_k - \lambda_k = \mu_k - a_k. \tag{3.1.5}$$

$$b_{k+1} - a_{k+1} = \tau(b_k - a_k) \tag{3.1.6}$$

第一个条件表示试探点要对称分布，第二个条件表示新区间是老区间的τ倍:



0.618法

情形1： $\varphi(\lambda_k) \le \varphi(\mu_k)$ 删除右端点

由于

$$b_k - \lambda_k = \mu_k - a_k \tag{3.1.7}$$

$$\mu_k - a_k = \tau(b_k - a_k) \tag{3.1.8}$$

于是得到

$$\lambda_k = a_k + (1-\tau)(b_k - a_k) \tag{3.1.9}$$

$$\mu_k = a_k + \tau(b_k - a_k) \tag{3.1.10}$$

删掉右半区间$(\mu_k, b_k]$, 保留$[a_k, \mu_k]$, 新搜索区间为

$$[a_{k+1}, b_{k+1}] = [a_k, \mu_k] \tag{3.1.11}$$

0.618法

为进一步缩短区间，由(3.1.10)与(3.1.11), 取试探点μ_{k+1}如下：

$$\begin{aligned}
\mu_{k+1} &= a_{k+1} + \tau(b_{k+1} - a_{k+1}) \\
&= a_k + \tau(\mu_k - a_k) \\
&= a_k + \tau(a_k + \tau(b_k - a_k) - a_k) \\
&= a_k + \tau^2(b_k - a_k).
\end{aligned} \tag{3.1.12}$$

若令$\tau^2 = 1 - \tau$, 取$\tau = \frac{\sqrt{5}-1}{2} \approx 0.618$, 则

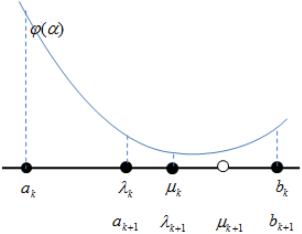
$$\mu_{k+1} = a_k + (1-\tau)(b_k - a_k) = \lambda_k. \tag{3.1.13}$$

$$\lambda_{k+1} = a_{k+1} + (1-\tau)(b_{k+1} - a_{k+1}) \tag{3.1.14}$$

新试探点μ_{k+1}无需重新计算，取λ_k即可。

0.618法

情形2：$\varphi(\lambda_k) > \varphi(\mu_k)$ 删除左端点



新的试探点$\lambda_{k+1} = \mu_k$,不需要重新计算。删去左半区间$[a_k, \lambda_k)$,保留$[\lambda_k, b_k]$, 新的搜索区间为$[a_{k+1}, b_{k+1}] = [\lambda_k, b_k]$.
令

$$\lambda_{k+1} = \mu_k,\ \mu_{k+1} = a_{k+1} + \tau(b_{k+1} - a_{k+1}).$$

再比较$\varphi(\lambda_{k+1})$和$\varphi(\mu_{k+1})$. 重复上述过程，直到$b_{k+1} - a_{k+1} \leq \varepsilon$。

0.618法：迭代步数与精度

试探点计算公式可写成:

$$\lambda_k = a_k + 0.382(b_k - a_k), \tag{3.1.15}$$

$$\mu_k = a_k + 0.618(b_k - a_k). \tag{3.1.16}$$

若要求最后区间长度不超过δ, 即$b_n - a_n \le \delta$, 由于

$$\frac{b_n - a_n}{b_0 - a_0} = (0.618)^n$$

则迭代步数n'为满足$\frac{\delta}{b_0 - a_0} \ge (0.618)^n$的最小正整数, 即有：

$$n' = \left\lfloor \log\left(\frac{\delta}{b_0 - a_0}\right) \Big/ \log(0.618) \right\rfloor + 1 \tag{3.1.17}$$

算法3.1.3 – 0.618法

步1 选取初始数据。确定初始搜索区间$[a_1, b_1]$和精度要求$\delta > 0$. 计算最初两个试探点λ_1, μ_1,

$$\lambda_1 = a_1 + 0.382(b_1 - a_1), \mu_1 = a_1 + 0.618(b_1 - a_1).$$

计算$\varphi(\lambda_1)$和$\varphi(\mu_1)$，令$k = 1$.

步2 比较目标函数值. 若$\varphi(\lambda_k) > \varphi(\mu_k)$, 转步3; 否则转步4.

步3 若$b_k - \lambda_k \le \delta$, 则停止计算，输出$\mu_k$; 否则，令

$$a_{k+1} := \lambda_k,\ b_{k+1} := b_k,\ \lambda_{k+1} := \mu_k,$$
$$\varphi(\lambda_{k+1}) := \varphi(\mu_k),\ \mu_{k+1} := a_{k+1} + 0.618(b_{k+1} - a_{k+1}).$$

计算$\varphi(\mu_{k+1})$, 转步5.

步4 若$\mu_k - a_k \le \delta$, 则停止计算，输出λ_k; 否则，令

$$a_{k+1} := a_k,\ b_{k+1} := \mu_k,\ \mu_{k+1} := \lambda_k,$$
$$\varphi(\mu_{k+1}) := \varphi(\lambda_k),\ \lambda_{k+1} := a_{k+1} + 0.382(b_{k+1} - a_{k+1}).$$

计算$\varphi(\lambda_{k+1})$, 转步5.

步5 $k := k + 1$, 转步2.

Fibonacci法

给定函数值计算的次数n，Fibonacci法中的计算公式为：

$$\begin{aligned}\lambda_k &= a_k + (1 - \frac{F_{n-k}}{F_{n-k+1}})(b_k - a_k) \\ &= a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k),\ k = 1, \cdots, n-1. \end{aligned} \tag{3.1.18}$$

$$\mu_k = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k),\ k = 1, \cdots, n-1. \tag{3.1.19}$$

其中Fibonacci数列满足：

$$\begin{aligned} &F_0 = F_1 = 1, \\ &F_{k+1} = F_k + F_{k-1}, k = 1, 2, \cdots. \end{aligned} \tag{3.1.20}$$

Fibonacci法搜索区间长度的缩短率采用Fibonacci数，而非黄金分割数。

Fibonacci法

注意: $\frac{F_{n-k}}{F_{n-k+1}}$相当于黄金分割法(3.1.9),(3.1.10)中的τ, 每次缩短率满足

$$b_{k+1}-a_{k+1}=\frac{F_{n-k}}{F_{n-k+1}}(b_k-a_k). \tag{3.1.21}$$

若要求最后区间的长度不超过δ, 即$b_n-a_n\leq\delta$,

$$\begin{aligned} b_n-a_n &= \frac{F_1}{F_2}(b_{n-1}-a_{n-1})=\frac{F_1}{F_2}\cdot\frac{F_2}{F_3}\cdots\frac{F_{n-1}}{F_n}(b_1-a_1) \\ &= \frac{1}{F_n}(b_1-a_1)\geq\delta \end{aligned} \tag{3.1.22}$$

迭代步数n可取满足如下不等式的最小整数:

$$F_n\geq\frac{b_1-a_1}{\delta}. \tag{3.1.23}$$

Fibonacci法三个步骤：

(1) 给出最终区间长度的上界δ;

(2) 根据上式求出Fibonacci数F_n;

(3) 再根据F_n确定出n, 一直进行到第n个搜索点。

Fibonacci算法与0.618法几乎完全相同,可以证明

$$\lim_{k\to\infty}\frac{F_{k-1}}{F_k}=\frac{\sqrt{5}-1}{2}=\tau. \tag{3.1.24}$$

Fibonacci法是分割方法求一维极小化问题的最优策略。0.618法是近似最优的，由于0.618法简单易行，因而应用更广泛。

二分法

二分法是一种最简单的分割方法，其基本思想是通过计算函数导数值来缩短搜索区间。设初始区间为$[a_1, b_1]$，第k步时的搜索区间为$[a_k, b_k]$，满足

$$\varphi'(a_k) \leq 0,\ \varphi'(b_k) \geq 0.$$

取中点$c_k = \frac{1}{2}(a_k + b_k)$，

- 若$\varphi'(c_k) \geq 0$，则令$a_{k+1} = a_k$，$b_{k+1} = c_k$；
- 若$\varphi'(c_k) \leq 0$，则令$a_{k+1} = c_k$，$b_{k+1} = b_k$，从而得到新的搜索区间$[a_{k+1}, b_{k+1}]$.

依此进行，直到搜索区间的长度小于预定的误差为止。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，收敛比为$\frac{1}{2}$。