最优化理论和方法

第8讲

最速下降法、BB法

第3章 无约束问题算法(I)

一、 最速下降法

二、 BB方法

给定无约束问题

$$\min_{x\in R^n} f(x) \qquad (3.1)$$

下降算法的结构：$x_{k+1} = x_k + \alpha_k d_k$

（1）步长α_k计算方法：Armijo法，Wolfe-Powell法

（2）下降方向d_k的计算方法，在后面的几章里来介绍.

下降方向的计算是整个最优化方法的核心,不同计算方式对应不同的最优化算法,相应算法的收敛性理论和数值效果有很大区别

思 考

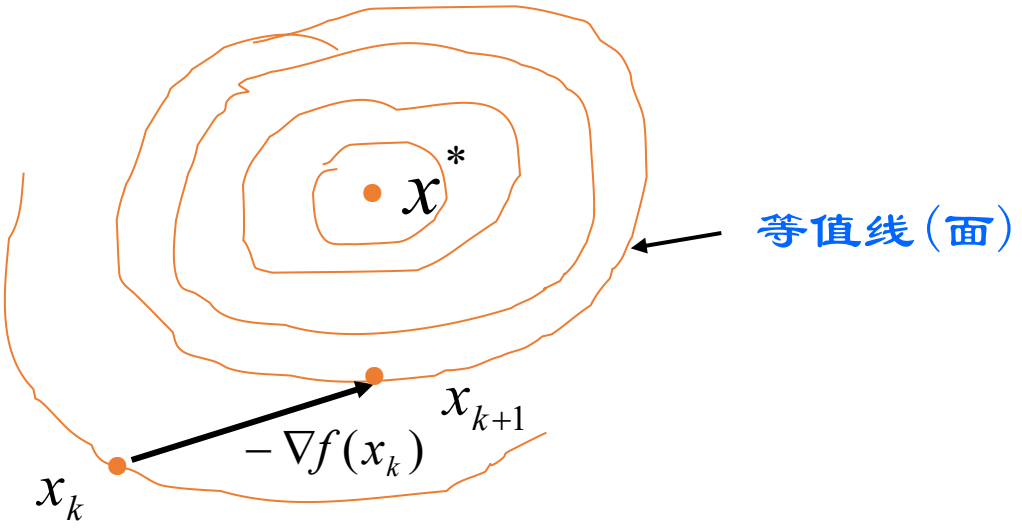
下降方向d_k如何选取?

一、 最速下降法

$$d_k = -\nabla f(x_k)$$

最古老的优化方法，十九世纪中叶由Cauchy提出

1、 思想 ：每次沿负梯度方向进行搜索



负梯度方向也称为最速下降方向：

事实上，对任意$p \in R^n$且$\| p \| = 1$,

由Cauchy - Schwarz不等式得

$$\nabla f(x_k)^T P \geq - \| \nabla f(x_k) \| \cdot \| P \| = - \| \nabla f(x_k) \|$$

当取$p = \dfrac{- \nabla f(x_k)}{\| \nabla f(x_k) \|}$时等号成立，即 $p = \dfrac{- \nabla f(x_k)}{\| \nabla f(x_k) \|}$是下列问题的解

$$\min_{\|p\|=1} \nabla f(x_k)^T P$$

以负梯度为搜索方向的算法称为最速下降法

2、 算法步骤

算法3.1 (最速下降法)

步1 给定初始点$x_0 \in R^n$, 精度$\varepsilon > 0$.令$k = 0$;

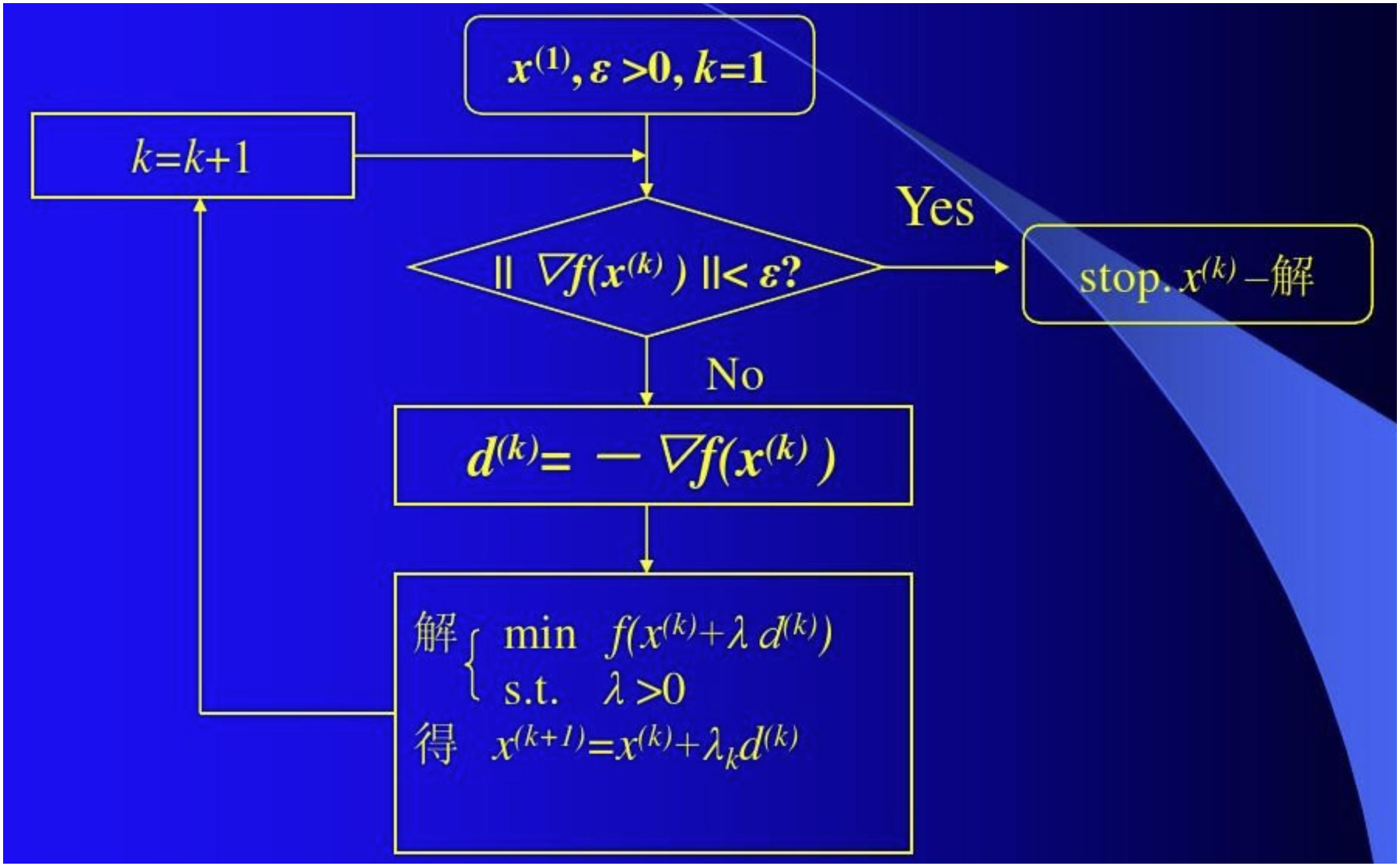
步2 若$\| \nabla f(x_k) \| \leq \varepsilon$, 则得解$x_k$, 算法终止.否则

计算$d_k = -\nabla f(x_k)$, 然后转步3;

步3 由线性搜索计算步长α_k;

步4 令$x_{k+1} = x_k + \alpha_k d_k, k := k+1$, 转步2.

最速下降算法流程图


$x^{(1)}, \varepsilon > 0, k=1$
$k=k+1$
$\| \nabla f(x^{(k)}) \| < \varepsilon ?$
Yes
stop. $x^{(k)}$ –解
No
$d^{(k)} = -\nabla f(x^{(k)})$
解 { min $f(x^{(k)}+\lambda d^{(k)})$
s.t. $\lambda > 0$
得 $x^{(k+1)}=x^{(k)}+\lambda_k d^{(k)}$

例3.1.1 取初始点$x^{(0)}=(2,\ 1)^T$. 采用精确搜索的最速下降法求解下面的最优化问题：

$$\min f(x)=\frac{1}{2}x_1^2+x_2^2$$

解 函数 f 的梯度为：$\nabla f(x)=\begin{pmatrix}x_1\\2x_2\end{pmatrix}, Q=\nabla^2 f(x)=\begin{pmatrix}1&0\\0&2\end{pmatrix}$

搜索方向 $d=-\nabla f(x)=\begin{pmatrix}-x_1\\-2x_2\end{pmatrix}$

采用精确搜索极小化二次函数的步长公式为

$$\alpha=-\frac{\nabla f(x)^T d}{d^T Q d}=\frac{x_1^2+4x_2^2}{x_2^2+8x_2^2}$$

迭代公式为

$$x^{(k+1)}=x^{(k)}+\alpha_k d^{(k)}=x^{(k)}-\frac{(x_1^{(k)})^2+4(x_2^{(k)})^2}{(x_1^{(k)})^2+8(x_2^{(k)})^2}\begin{pmatrix}x_1^{(k)}\\2x_2^{(k)}\end{pmatrix}$$

由上面的迭代公式不难计算出最速下降法产生的点列为

$$x^{(k)}=\left(\frac{1}{3}\right)^{k}\begin{pmatrix}2\\(-1)^{k}\end{pmatrix},k=0,1,\cdots$$

容易看出上面的序列是收敛的,并且$x^{(k)}\to(0,\ 0)^{T}=x^{*}$

从上面的例子看到,对于简单的二元二次函数极小化问题,最速下降法在有限次迭代并没有求出其精确最优解,但能以较慢的速度无限接近最优解.

事实上，上面的例子刻画了最速下降法的所有收敛特征，以及最多线性的收敛速度。

3、 最速下降法的收敛性

全局收敛性

由于最速下降法的搜索方向与负梯度方向一致,即$\theta_k = 0$,且

$$\| \nabla f(x_k) \| = \| d_k \|$$

所以,由定理2.4.1 - 2.4.3,我们很容易得到最速下降算法的全局收敛性.

定理3.1.1 设假设2.4.1的条件成立,那么采用精确搜索, 或Armijo搜索或Wolfe - Powell搜索的最速下降法产生的迭代序列$\{x_k\}$满足

$$\lim_{k\to\infty} \| \nabla f(x_k) \| = 0$$

由前面的例子看到,最速下降法的收敛速度至多是线性的,具体见下面的两个定理.

收敛速度估计

下面的定理给出了其求解严格凸二次函数极小化问题的收敛速度估计.定理证明可袁亚湘[27, 定理3.1.4]

定理3.1.2 设矩阵Q对称正定,$q \in R^n$.记$\lambda_{\max}$和$\lambda_{\min}$分别是Q的最大和最小特征值 ,$\kappa = \dfrac{\lambda_{\max}}{\lambda_{\min}}$.考察如下二次函数极小 化问题：

$$\min f(x) = \frac{1}{2} x^T Q x + q^T x$$

则由采用精确搜索的最 速下降法产生的点列$\{x_k\}$满足

$$\| x_{k+1} - x^* \|_Q \leq \left(\frac{\kappa - 1}{\kappa + 1} \right) \| x_k - x^* \|_Q \qquad (3.2)$$

其中x^*是问题的唯一解,$\| x \|_Q = \left(x^T Q x \right)^{\frac{1}{2}}$

$$\| x_{k+1} - x^* \|_Q \leq \left(\frac{\kappa - 1}{\kappa + 1} \right) \| x_k - x^* \|_Q \qquad (3.2)$$

对于二次函数,由于$\nabla f(x) = Qx + q$且在x^*处

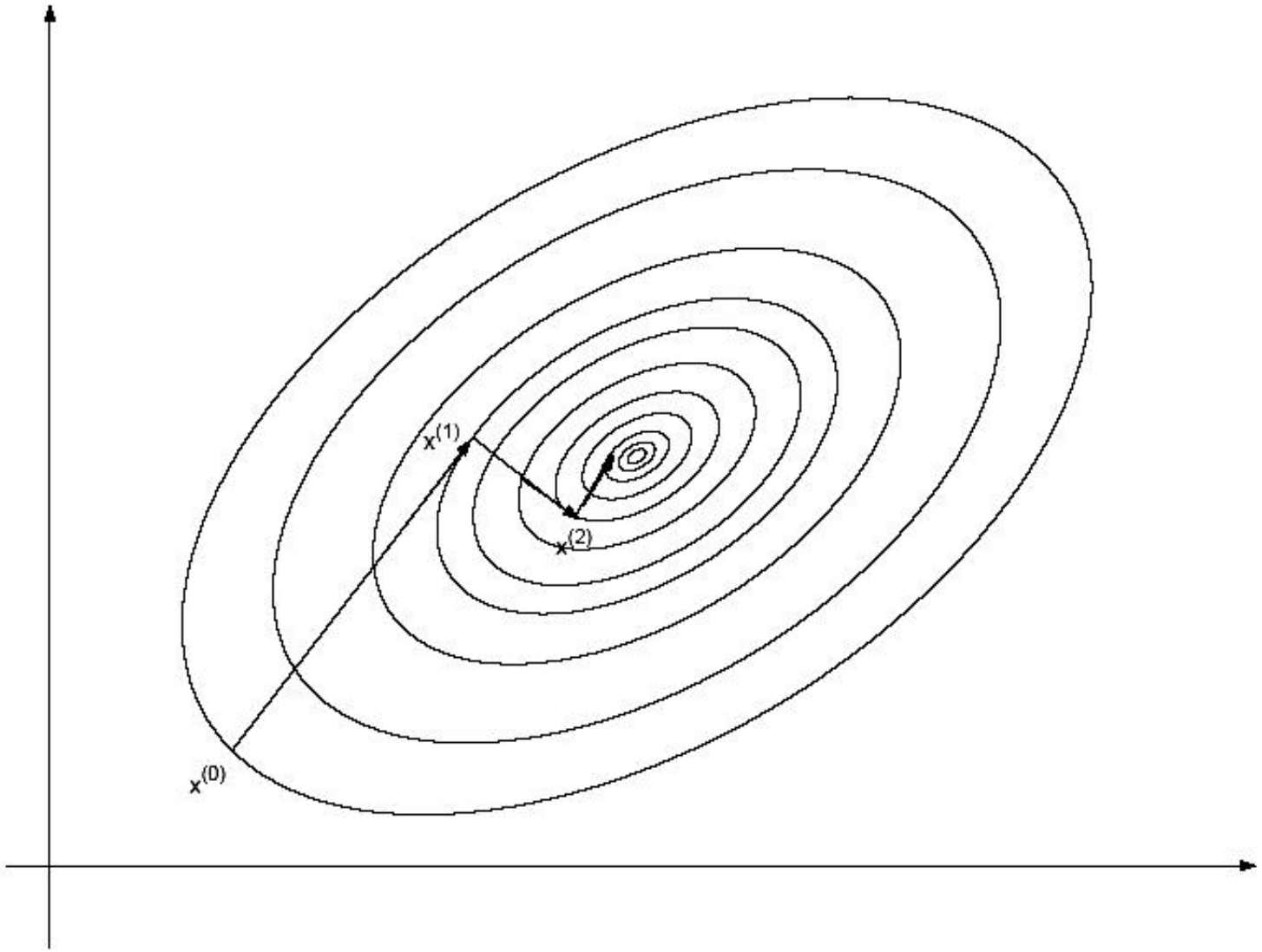
$$\nabla f(x^*) = Qx^* + q = 0$$

则 $$f(x) - f(x^*) = \frac{1}{2}(x - x^*)^{\mathrm{T}} Q(x - x^*) = \frac{1}{2} \| x - x^* \|_Q^2$$

所以(3.2)可以改写成

$$f(x_{k+1}) - f(x^*) \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^2 [f(x_k) - f(x^*)]$$

由收敛速度估计式(3.2)看到,最速下降的收敛速度与矩阵Q的条件数κ有关,当κ接近于1,最速下降收敛很快,特别,当$\kappa = 1$即Q的所有特征值相等时,算法只需一次迭代即可求出最优解. 而当κ较大时(Q接近病态),算法收敛很慢.



从上图可以看出，最速下降法 具有锯齿现象

对一般的非二次函数有下面的收敛速度估计：

定理3.1.3 设函数f二次连续可微，若采用精确搜索的最速下降法产生的点列$\{x_k\}$收敛到问题(3.1)的解x^*，且$\nabla^2 f(x^*)$正定，则有估计

$$f(x_{k+1}) - f(x^*) \le \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 [f(x_k) - f(x^*)] + o[f(x_k) - f(x^*)]$$

其中$\kappa = \dfrac{\lambda_{\max}}{\lambda_{\min}}$，且$\lambda_{\max}$和$\lambda_{\min}$分别是$\nabla^2 f(x^*)$的最大和最小特征值.

定理的证明参见文献[19, 定理3. 4]

由上面的分析可知，最速下降法的收敛速度比较慢，通常将其用在某些算法的初始阶段求较好的初始点或作为某些算法的间插步.

你们上机1作业采用三种步长搜索方法，对最速下降法进行了数值计算，有什么结论和体会？

最速下降方向+最好步长

效果最好吗？

两点步长梯度法(BB法)

Barzilai 和 Borwein (1988) 提出两点步长梯度法，其基本思想是利用迭代当前点以及前一点的信息来确定步长因子. 迭代公式 $x_{k+1} = x_k - \alpha_k g_k$ 可以看成是

$$x_{k+1} = x_k - D_k g_k,$$

其中 $D_k = \alpha_k I$ 是一个矩阵. 为了使矩阵 D_k 具有拟牛顿性质 (拟牛顿法将在第五章讨论), 计算 α_k 使得

$$\min \|s_{k-1} - D_k y_{k-1}\|,$$

$$s_{k-1} = x_k - x_{k-1}, \quad y_{k-1} = g_k - g_{k-1}.$$

$$\alpha_k = s_{k-1}^T y_{k-1} / \|y_{k-1}\|^2 \qquad (3.1.37a)$$

步长采用上一步的最佳步长

$$x_k + \alpha_k(-g_k) = x_{k-1} + \alpha_k(-g_{k-1}),$$
$$g_k := -\nabla f(x_k)$$

算法 3.1.7 (两点步长梯度法)

步 1　给出 $x_0 \in R^n$, $0 \leqslant \varepsilon \ll 1$, $k := 0$;

步 2　计算 $d_k = -g_k$; 如果 $\|g_k\| \leqslant \varepsilon$, 则停止;

步 3　如果 $k = 0$, 利用一维搜索求 α_0; 否则, 利用 (3.1.37a) 或 (3.1.37b) 计算 α_k.

步 4　$x_{k+1} = x_k + \alpha_k d_k$;

步 5　$k := k + 1$, 转步 2.　□

$$\alpha_k = s_{k-1}^T y_{k-1} / \|y_{k-1}\|^2 \tag{3.1.37a}$$

$$\alpha_k = \|s_{k-1}\|^2 / s_{k-1}^T y_{k-1}. \tag{3.1.37b}$$

超线性收敛！

$$s_{k-1} = x_k - x_{k-1}, \quad y_{k-1} = g_k - g_{k-1}.$$

IMA Journal of Numerical Analysis (1988) **8,** 141–148

Two-Point Step Size Gradient Methods

JONATHAN BARZILAI

School of Business Administration, Dalhousie University, Halifax, N.S., Canada

AND

JONATHAN M. BORWEIN

Department of Mathematics, Statistics and Computing Science, Dalhousie University, Halifax, N.S., Canada

[Received 24 September 1986 and in revised form 14 April 1987]

We derive two-point step sizes for the steepest-descent method by approximating the secant equation. At the cost of storage of an extra iterate and gradient, these algorithms achieve better performance and cheaper computation than the classical steepest-descent method. We indicate a convergence analysis of the method in the two-dimensional quadratic case. The behaviour is highly remarkable and the analysis entirely nonstandard.

数值实验比较

- 练习Matlab编程，分别编写最速下降法,BB法和Newton法的程序，计算习题三的11，以及Rosenbock函数的极小值点：

$$\min f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

初始点分别取（1.2，1.2）和（-1.2，1）。

列出每一步的步长。比较三种算法的优劣。