1-4 迭代算法与收敛速度

最优化方法

最优化方法(算法)主要包含以下两类：解析法与迭代法

- 解析法是直接给出优化问题最优解的显式表达式的方法。比如，通过最优性条件建立方程并求解。现实生活中只有极少数最优化问题可用解析法求解。
- 迭代法是求解最优化问题的解一般方法，其基本思想：给定最优解的一个**初始估计**，记为$\boldsymbol{x}_0$，方法产生一个**逐步改善**的**有限或无限**的迭代序列$\{\boldsymbol{x}_k\}$，
  - 序列为有限时，最后一个点是极值点；
  - 若为无限时，其任意一个聚点是极值点。

  在对最优解的估计满足**指定的精度要求**时停止迭代。

无约束优化算法迭代格式

基本格式:

- **步1:** 给定初始迭代点$\boldsymbol{x}_0$,置$k=0$;
- **步2:** 若$\boldsymbol{x}_k$满足终止条件, 输出结果并停止迭代;
- **步3:** 确定一个改善$\boldsymbol{x}_k$的修正量$\Delta\boldsymbol{x}_k$;
- **步4:** 令$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\Delta\boldsymbol{x}_k$, 置$k:=k+1$后转**步**2。

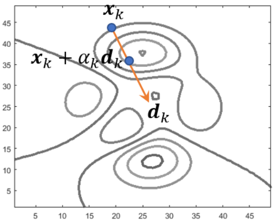
关键要素:

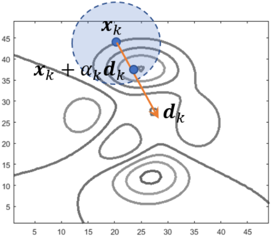
- 初始解选取
- 迭代点优劣比较(评价函数- 目标函数(无约束) 罚函数(约束))
- 修正量确定(通常由某个搜索方向$\boldsymbol{d}_k$与步长构成, 即$\Delta\boldsymbol{x}_k=\alpha_k\boldsymbol{d}_k$)
- 终止条件
- 收敛速度

线搜索方法与信赖域法

修正量的确定是优化迭代算法中的核心关键要素，不同的步长和下降方向选取方法对应不同的优化算法。根据二者确定的先后顺序，可将优化算法分为：

- 线搜索方法 先确定搜索方向$\boldsymbol{d}_k$, 后沿着$\boldsymbol{d}_k$方向确定步长α_k。
- 信赖域方法 先确定步长范围，再同时确定搜索方向$\boldsymbol{d}_k$与步长α_k。





算法全局收敛与局部收敛

定义1.4.1： 设$\boldsymbol{x}^*$为优化问题的极值点. 若算法产生点列$\{\boldsymbol{x}_k\}$满足:

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

则称该算法为**收敛的**。

- 若算法对于任意给定的初始点均收敛, 则称该算法具有**全局收敛性**或**总体收敛性**;
- 若算法仅在当初始点接近$\boldsymbol{x}^*$时迭代点列收敛到$\boldsymbol{x}^*$, 则称该算法具有**局部收敛性**。

注意: 一个收敛的算法产生的迭代点列通常满足：

- 函数值单调下降(每一步始终选取下降方向)
- 或者点列到最优解距离单调下降。

收敛速度

定义1.4.2： 设向量序列$\{\boldsymbol{x}_k\} \subset \mathbb{R}^n$收敛于$\boldsymbol{x}^*$，定义误差序列：$\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}^*$ 如果存在常数a和r，使得:

$$\lim_{k\to\infty} \frac{\|\boldsymbol{e}_{k+1}\|}{\|\boldsymbol{e}_k\|^r} = \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^r} = a$$

成立，则称序列$\{\boldsymbol{x}_k\}$以a为因子r阶收敛于$\boldsymbol{x}^*$。

常见收敛阶：

- $r = 1$(线性$(0 < a < 1)$或超线性收敛$(a = 0)$)
- $r = 2$(二阶收敛)

线性收敛

线性收敛的阶数$r = 1$, 存在以下三种情形：

- 线性收敛$(0 < a < 1)$
  当k充分大时, 有:
  $$\|\boldsymbol{e}_{k+1}\| \approx a\|\boldsymbol{e}_k\|$$
  考虑等号情形, 初始误差为1：
  - 取$a = 0.5$, 则误差序列为：$1, 0.5, 0.25, 0.125, 0.0625, \ldots$
  - 取$a = 0.1$, 则误差序列为：$1, 0.1, 0.01, 0.001, \ldots$

  可见a越小收敛越快.
- 超线性收敛$(a = 0)$,
- 次线性收敛$(a = 1)$

多数最优化方法具有超线性收敛特性。注意: 所有$r > 1$的收敛都属于超线性收敛。

二阶收敛速度

二次收敛的阶数为$r=2$

初始误差为0.1,$a=1$, 误差序列如下:

0.1, 0.01, 0.0001, 0.00000001,...

每迭代一步精度数量级增加一倍。

例1.4.1: (线性收敛与二阶收敛速度比较) 假定当前迭代点$\boldsymbol{x}_k$满足$\|\boldsymbol{x}_k-\boldsymbol{x}^*\|\le 0.003$, 当$k$足够大时分别有:

$$\frac{\|\boldsymbol{e}_{k+1}\|}{\|\boldsymbol{e}_k\|}\approx\frac{1}{2} \quad 和 \quad \frac{\|\boldsymbol{e}_{k+1}\|}{\|\boldsymbol{e}_k\|^2}\approx\frac{1}{2}$$

问: 若要达到$\|\boldsymbol{e}_k\|\le 10^{-9}$的精度, 分别采用具有线性收敛速度与二阶收敛速度算法, 各需多少次迭代? (22次和2次)

二次终止性

正定二次函数:

- 在非线性目标函数中, 正定二次函数是凸函数，具有很好的性质，如光滑性和唯一极小点。
- 另外，在极小点附近，一般函数可以用正定二次函数很好地近似。
- 能否有效地求得正定二次函数的极小点，是检验算法好坏的标准之一。

二次终止性:

- 一个算法从任意初始点出发, 若能够在有限步内找到正定二次函数极小点, 则称该算法具有二次终止性。
- 共轭梯度法具有二次终止性。

例题

例题1.4.2 考虑序列：$\{a^k\}, 0 < a < 1$的收敛速度。

分析： 由于$a^k \to 0$，于是

$$\lim_{k\to\infty} \frac{a^{k+1}}{a^k} = a < 1$$

故序列$\{a^k\}$线性收敛于零。

例题1.4.3 考虑序列：$\{a^{2^k}\}, 0 < |a| < 1$的收敛速度。

分析： 由于$a^{2^k} \to 0$，于是

$$\lim_{k\to\infty} \frac{a^{2^{k+1}}}{(a^{2^k})^2} = 1$$

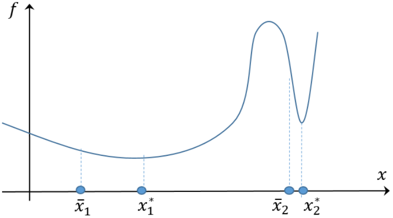
故序列$\{a^{2^k}\}$2阶收敛于零。

终止条件1：梯度变化

根据一阶最优性条件：$\|\nabla f(\boldsymbol{x}_k)\| = 0$，给定很小的精度$\epsilon > 0$，若以下条件成立

$$\|\nabla f(\boldsymbol{x}_k)\| \le \epsilon$$

则终止算法迭代。



缺陷： 依赖于函数极小点邻域内的性质。当前迭代点离最优解仍然很远,但梯度值小，出现过早终止。迭代点离最优解很近，但梯度值依然很大，无法终止。

终止条件2：点列变化

若点列至最优点的误差满足以下条件：

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \le \epsilon$$

则终止算法迭代。通常$\boldsymbol{x}^*$未知，以上准则无法实用。

对具有超线性收敛速度的方法，可证明如下结论：

$$\lim_{k\to\infty} \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|} = 1$$

故采用如下准则判断停机

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| \le \epsilon$$

终止条件3：函数值变化

类似地，若点列函数值与最优点函数值之间误差满足

$$|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^*)| \leq \epsilon$$

若$f(\boldsymbol{x})$二次连续可微时，有：

$$|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| = O(\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2)$$

故若以下条件满足：

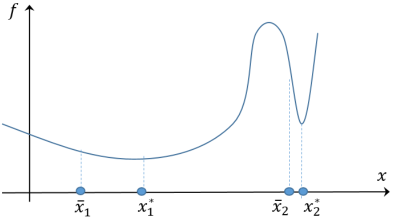
$$|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| \leq \epsilon$$

则终止算法迭代。

组合终止条件

注意： 在一些情况下，单一准则不能确保满意精度近似解，如

- $\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|$较小，$|f(\boldsymbol{x}_{k+1})-f(\boldsymbol{x}_k)|$依然较大，$\boldsymbol{x}_k$离$\boldsymbol{x}^*$可能依然很远
- $|f(\boldsymbol{x}_{k+1})-f(\boldsymbol{x}_k)|$较小，$\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|$依然较大，$\boldsymbol{x}_k$也离$\boldsymbol{x}^*$可能依然很远



针对以上情形，可采用组合方式来确定停止准则。