

# 最优化方法/实用优化算法主要知识点

2023 年 6 月 22 日

如果不做说明，一般用  $g(x)$  代表  $\nabla f(x)$ ,  $G(x)$  代表  $\nabla^2 f(x)$ ,  $L$  代表 Lagrange 函数.

- 一维搜索
  - 精确一维搜索：解析解（求导数，令导数为0）；迭代解：黄金分割法；
  - 非精确搜索：Wolfe 准则，Armijo 准则（了解公式）
- 无约束最优性条件
  - 一阶必要条件： $\nabla f(x) = 0$ ；会据此求解简单的无约束问题的极小点.
  - 二阶充分条件： $\nabla f(x) = 0$ ,  $\nabla^2 f(x)$  正定；据此可判断一个驻点是不是极小点.  
由于我们遇到的问题维数都比较低，所以往往用霍尔维茨定理来判断正定性更为容易.
  - 下降方向1. 定义；2.  $\nabla f_k^T d < 0$ .
  - 收敛速度：线性，超线性，2阶收敛（定义）
- 无约束优化算法：
- 算法框架（主要是要说清楚搜索方向, 步长）
  - 搜索方向
    - \* 最速下降法:  $d_k = -g_k$ ;
    - \* Newton, 阻尼Newton 法:  $d_k = -G_k^{-1} g_k$  或  $G_k d_k + g_k = 0$ .
    - \* 共轭梯度法:  $d_0 = -g_0$ ,  $d_k = -g_k + \beta d_{k-1}$ , 掌握用FR 公式计算  $\beta$ ;

- \* 拟Newton 法:  $d_k = -H_k g_k$ ,  $H_k$  由拟Newton 修正公式计算, 至少掌握DFP 公式;

– 计算步长:

- \* 精确搜索:

- 写出  $\varphi(\alpha) = f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k)$ . 一些比较简单的问题会利用其导数求出  $\alpha_k$ .
- 对于二次函数, 会利用公式直接求得  $\alpha_k = -\frac{g_k^T d_k}{d_k^T G d_k}$ . (非二次函数不能用此公式求步长)
- 能够用黄金分割法求给定点和给定方向处的精确步长.

- \* 非精确搜索: Wolfe 准则和 Armijo 准则 (知道公式)

– 算法的其它相关问题

- \* 各个算法的优缺点(如计算量, 收敛性, 收敛速度等).
- \* 最速下降法的 Zigzag 现象 (锯齿现象):  $d_{k+1}^T d_k = 0$ ,
- \* 精确搜索  $\Rightarrow g_{k+1}^T d_k = 0$ ,
- \* 共轭方向的定义, 性质
- \* 拟Newton 方程, DFP 的正定继承性,
- \* Newton 法, 共轭方向法, 拟Newton 法的二次终止性.

● 约束优化问题

– 可行方向, 下降方向, 可行下降方向的概念, 即判断方法;

- \* 下降方向 1. 定义; 2.  $\nabla f(x)^T d < 0$ .
- \* 可行方向 1. 定义; 2. 对不等式约束, 满足  $\nabla c_i(x)^T d > 0, \forall$  积极不等式约束  $c_i(x) \geq 0$ .

– 含不等式约束的问题的有效集

- \* 有效集的定义, 能够写出给定点处的有效集及对应的等式约束优化问题;
- \* 有效集在算法中的作用: 将不等式约束转化为等式约束问题.

– 一阶最优性条件 (KKT) 条件;

- \* 能够写出约束优化问题的最优性条件（本课程最重要的知识点之一）并利用其求解简单的约束优化问题.（最优化方法：会用一阶最优性条件完成证明）
  - 等式约束：即高数讲过的Lagrange 乘子法；能用它求出简单的等式约束优化问题的解.
  - 不等式约束: KKT 条件;能够利用它验证一个点是KKT点或求解简单的含不等式约束的问题.
- \* 一般说来, 作业题中如果没有明确要求用某种算法求解约束优化问题, 则用Lagrange 乘子法或KKT条件求解比较方便.
- \* LICQ 约束规范（知道其定义）
- 二阶充分条件（能就简单的问题验证、使用二阶充分条件）
- 约束优化问题的算法
  - 外罚函数法
    - \* 等式约束：外罚函数的表达式，能够利用外罚函数求解简单的问题;
    - \* 不等式约束和一般约束：外罚函数的表达式
  - 内罚函数法（内点法，障碍函数法）
    - \* 倒数障碍函数
    - \* 对数障碍函数
    - \* 会用内罚函数法求解不等式约束优化问题. 同一问题，使用两种障碍函数，其求解难度可能会有很大的差异.
  - 增广Lagrange 函数法(课本上叫乘子法)
    - \* 等式约束：增广Language 函数的表达式，乘子的迭代公式，能够利用增广Lagrange 函数求解简单的问题;
    - \* 不等式约束和一般约束：增广Language 函数的表达式，乘子的迭代公式.
  - 上述三种算法的性质，收敛性（了解）
  - 二次规划的有效集算法（了解，知道一些基本概念，比如有效集，可行方向等）.

- 提示：一般说来，用迭代算法手动求解问题，即使问题比较简单，即使只要求迭代较少次，计算量也是比较大的。因此一定要熟悉每个算法的流程，在复习阶段动手计算一定数量的习题。