BCD&ADMM 收敛理论

罗自炎
zyluo@bjtu.edu.cn

AI4M优化方向暑期班 北京大学
2024.7

Lecture 4：数值迭代算法概述

算法概述

数值迭代算法

主要利用问题的函数值信息或者（一阶与\或二阶）导数信息等，按照一定的规则从当前迭代点产生下一个更好的迭代点，直到不能改进为止.这类算法得到的是数值解，即为最优解的近似解.

分类：

模式搜索法

根据函数值变化规律探测函数的下降方向并沿着该方向寻求更优的点.

优点：简单、直观、无需计算目标函数的梯度

缺点：适用于变量较少、约束简单的情形

常见算法：坐标轮转法、Hooke-Jeeves法、Powell共轭方向法、单纯形法

梯度法

利用函数值及其梯度信息等按照一定的搜索策略进行迭代.

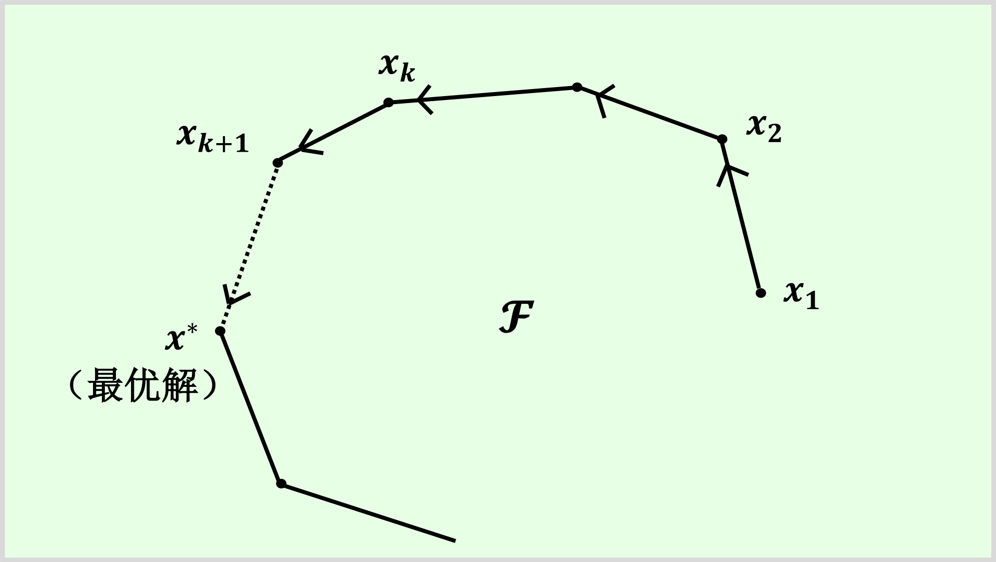
优点：收敛速度快且易建立其理论性质

缺点：要求函数连续性且其梯度易计算

搜索策略：线搜索方法、信赖域方法

算法概述

数值迭代算法



单纯形法搜索轨迹

算法概述

线搜索算法

在算法的每一步迭代中，基于目标函数的梯度信息产生一个**搜索方向**,沿着该方向走一定的距离（**迭代步长**），使得目标函数获得一定程度的下降，使得新的迭代点更靠近最优解.

信赖域算法

在当前迭代点附近建立目标函数的一个近似**二次**模型,利用目标函数在当前点的某邻域内（信赖域）与该二次模型的充分近似，基于该二次模型在该邻域内的最优解产生新的迭代点.

算法概述

数值迭代算法评价指标

- 全局收敛与局部收敛
- 收敛速度与二次终止性
- 稳定性
- 计算复杂性与存储消耗
- 数值效果

算法概述

全局收敛与局部收敛

定义1 设S^*是最优化问题的某种解集合，A为某种算法. 给定集合Y，若对于任意初始点$x_0 \in Y$，算法A产生的迭代点列$\{x_k\}$中任一收敛子列的极限都属于S^*，则称算法A在Y上收敛.

- 若集合Y可以任意选取（无需限定在解集合很小的邻域内），则相应的收敛性为全局收敛性（global convergence）；
- 若集合Y只能取接近解集合的点集, 则相应的收敛性为局部收敛性（local convergence）.

算法概述

解集合

满足某些最优性条件的点构成的集合.

常用解集合

- $S^* = \{x^*: \nabla f(x^*) = 0\}$ 【无约束优化一阶最优性条件】
- $S^* = \left\{x^*: x^* \text{ 满足KKT条件}\right\}$ 【约束优化的一阶最优性条件】
- $S^* = \{x^*: f(x^*) \leq \alpha\}$ α 为某个可接受的目标函数值

算法概述

子序列收敛与全序列收敛

定义2 算法A产生的迭代点列$\{x_k\}$中任一收敛子列的极限点都属于解集S^*，则称为子序列收敛（subsequence convergence），若$x_k \to x^* \in S^*$，则称为全序列收敛(whole sequence convergence).

算法概述

例题1：判断如下点列是否收敛

(1) $\left\{\frac{1}{k}\right\}$ ✓

(2) $\left\{\frac{1}{k!}\right\}$ ✓

(3) $\left\{\frac{1}{2^{2^k}}\right\}$ ✓

(4) $\{\boldsymbol{x}_k\}$, 其中 $\boldsymbol{x}_k = -\boldsymbol{x}_{k-1}^3$

算法概述

收敛速率

定义2 设点列$\{x_k\}$收敛到x^*，若存在实数$p \geq 0,\ r \geq 1$,使得

$$\lim_{k\to+\infty} \frac{\|x_{k+1}-x^*\|}{\|x_k - x^*\|^r} \leq p < +\infty,$$

则称点列$\{x_k\}$以Q-r阶收敛速率收敛到x^*.

Quotient

- Q-次线性收敛： $r = 1,\ p = 1$
- Q-线性收敛： $r = 1,\ p \in (0,1)$
- Q-超线性收敛： $r = 1,\ p = 0$
- Q-二次收敛： $r = 2,\ p > 0$

算法概述

例题2：判断如下点列的收敛速率

(1) $\left\{\frac{1}{k}\right\}$ Q-次线性收敛

(2) $\left\{\frac{1}{2^k}\right\}$ Q-线性收敛

(3) $\left\{\frac{1}{k!}\right\}$ Q-超线性收敛

(4) $\left\{\frac{1}{2^{2^k}}\right\}$ Q-二次收敛

(5) $\{x_k\}$，其中 $x_k = -x_{k-1}^3$ 且 $|x_0| < 1$ Q-3阶收敛

算法概述

收敛速率

定义3 设点列$\{x_k\}$收敛到x^*，若存在实数$\alpha > 0,\ q \in (0,1)$,使得

$$\|x_k - x^*\| \le \alpha q^k, \qquad =: t_k \quad \text{Q-线性收敛到0}$$

则称点列$\{x_k\}$以R−线性收敛到x^*.

Root

若存在实数$\alpha > 0$, 以及收敛到0的正数列$\{q_k\}$,使得

$$\|x_k - x^*\| \le \alpha \prod_{i=1}^{k} q_i\,, \qquad =: t_k \quad \text{Q-超线性收敛到0}$$

则称点列$\{x_k\}$以R−超线性收敛到x^*.

算法概述

Q-收敛速率 $\Longrightarrow$ R-收敛速率

以线性收敛为例：

$$
\begin{aligned}
\|x_k - x^*\| &\le p\|x_{k-1} - x^*\| \\
&\le p^2\|x_{k-2} - x^*\| \\
&\;\vdots \\
&\le p^k\|x_0 - x^*\| \\
&=: \alpha q^k
\end{aligned}
$$

算法概述

迭代次数复杂度

定义4 设某一算法求解 $\min_{x\in\mathbb{R}^n} f(x)$产生的迭代点列$\{x_k\}$收敛到最优解$x^*$，若存在实数$c, r > 0$,使得

$$f(x_k) - f(x^*) \le \frac{c}{k^r} \qquad \forall k > 0$$

则称该算法是$\mathcal{O}\left(\frac{1}{k^r}\right)$阶收敛到$x^*$.

- 若需要算法满足精度$f(x_k) - f(x^*) \le \varepsilon$, 只需令$\frac{c}{k^r} \le \varepsilon$, 从而得到$k \ge \frac{c^{1/r}}{\varepsilon^{1/r}}$.
  因此该优化算法对应的迭代次数复杂度为$N(\varepsilon) = \mathcal{O}\left(\frac{1}{\varepsilon^{1/r}}\right)$
- $\boldsymbol{r}$ 越大, 收敛越快, 复杂度越低, 算法越好.

算法概述

二次终止性

定义5 若某个算法对任意的严格凸二次函数，从任意的初始点出发，都能经有限步迭代达到其极小值点，则称该算法具有**二次终止性**.

- **用二次终止性作为判断算法优劣的原因：**
  - 严格凸二次函数具有某些较好的性质，因此一个好的算法应能够在有限步内达到其极小点.
  - 对于一般的目标函数f, 若在其极小值点x^*处海塞矩阵$\nabla^2 f(x^*)$正定, 则
    $$f(x) = f(x^*) + \nabla f(x^*)^T(x - x^*) + (x - x^*)^T \nabla^2 f(x^*)(x - x^*) + o\left(\|x - x^*\|^2\right)$$
    可以猜想, 对严格凸二次函数好的算法, 对于一般函数f也应有较好性质.

算法概述

空间复杂性与时间复杂性

定义6 描述算法的存储要求和运行时间要求,分为算法的**空间复杂性**和算法的**时间复杂性**. 利用算法需要的**初等运算次数**表示算法的时间复杂性.

- 求解实例I的算法的基本运算总次数$C(I)$是实例输入长度$L(I)$的一个函数,该函数被另一个函数g控制，即存在一个函数g和一个常数α, 使得

$$C(I) \le \alpha g(L(I))$$

若g为多项式函数, 则称该算法对实例I是多项式时间的算法;若对于这类问题的任意实例I是多项式时间的算法,则称该算法为求解这一类问题的**多项式时间算法**.类似地可以定义的**指数时间算法**.

算法概述

多项式时间算法的优势

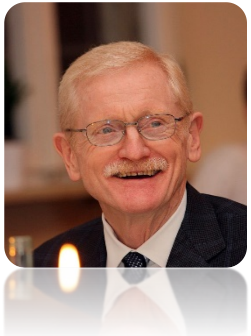
- 随着问题输入规模的增加, 算法的计算量（即算法时间复杂性）呈多项式增长.
- 一个多项式时间算法利用另一个多项式时间算法作为其“子程序”, 构造一个新的复合型算法，则新算法仍是多项式时间算法。

$$\boldsymbol{g(x) = g_1(g_2(x))}$$

算法概述

单纯形法 & 内点法

➢ **单纯形法（Simplex Method）是指数时间算法**



【CPLEX】 https://www.ibm.com/analytics/cplex-optimizer

IBM ILOG CPLEX Optimization Studio（CPLEX）是一个优化软件包.2004年CPLEX的工作获得了首届INFORMS影响力奖.最初由**Robert E. Bixby**开发,并于1988年由CPLEX商业出售,随后ILOG在2009年1月被IBM收购.

➢ **内点法（Interior Point Method）是多项式时间算法**

【Gurobi】http://www.gurobi.com/

在全球著名的专业优化器评比网站Decision Tree for Optimization Software(plato.asu.edu/bench.html)中,Gurobi比其他大规模优化器有明显优势.

Zonghao Gu, Edward Rothberg, Robert Bixby

Thank You !