学海无涯苦作舟

优化篇

洗衣机

Last Update: 2025 年 4 月 4 日

目录

1 分式规划 (Fractional Programming): Dinkelbach 算法 1

2 Newton's Method 2

3 Damped Newton's Method 3

4 总结 4

1 分式规划 (Fractional Programming): Dinkelbach 算法

考虑如下问题

$$\min_{x\in\mathcal{S}} F(x) = \frac{f(x)}{g(x)} \tag{1}$$

其中 $f(x), g$ 是适当的闭凸函数, $g(x) > 0$.

Algorithm 1 Dinkelbach 算法

Input: 初始值 $\lambda^{(0)} \in \mathbb{R}$, 迭代次数 $k = 0$, 约束条件 $x \in \mathcal{S}$, 精度阈值 $\epsilon > 0$

Output: 最优解 x^* 和最优值 λ^*

repeat

  Step 1: 求解子问题:

$$x^{(k)} = \arg\max_{x\in X}\{f(x) - \lambda^{(k)} g(x)\} \tag{2}$$

  Step 2: 计算:

$$\phi(\lambda^{(k)}) = \max_{x\in X}\{f(x) - \lambda^{(k)} g(x)\} \tag{3}$$

  Step 3: 更新:

$$\lambda^{(k+1)} = \frac{f(x^{(k)})}{g(x^{(k)})} \tag{4}$$

  更新迭代次数: $k \leftarrow k + 1$

until $\phi(\lambda^{(k)}) < \epsilon$

性质 1.1. *ϕ 关于 λ 单调递减: $\lambda_1 < \lambda_2 \Rightarrow \phi(\lambda_1) > \phi(\lambda_2)$.*

性质 1.2. $\lambda = \lambda^* \Leftrightarrow \phi(\lambda) = 0$.

证明. $(\Rightarrow)$: 令 $\lambda = \lambda^* = F(x^*) = \frac{f(x^*)}{g(x^*)}$. $\forall x \in \mathcal{S}, \lambda^* \le \frac{f(x)}{g(x)} \Rightarrow f(x) - \lambda^* g(x) \ge 0$, 因此 x^* 恰好取到 $\phi(\lambda^*)$ 的下界 0.

$(\Leftarrow)$: 假设存在 $\lambda' = F(x')$ 是更优解, 因此 $\lambda' = \frac{f(x')}{g(x')} < \lambda \Rightarrow f(x') - \lambda g(x') < 0$, i.e.$\phi(\lambda) < 0$, 矛盾. □

2025 年 4 月

LectureInformation

Lecture Number: 2
Student: MIStalE
School: Fudan University
Course: Optimization
Date: Fall 2024

2 Newton's Method

假设 $f(x)$ 是 两次连续可微的且 强凸。那么可以用以下二次近似对 $f(x_{k+1})$ 进行描述：

$$f(x_{k+1}) \approx f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{1}{2}(x_{k+1} - x_k)^T \nabla^2 f(x_k)(x_{k+1} - x_k) \tag{5}$$

令 $p = x_{k+1} - x_k$，我们最小化右边的二次近似得到一个近似的最优解 x_{k+1}：

$$x_{k+1} = \arg\min_p \left[f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p \right] + x_k \tag{6}$$

通过求解上式的最优 p，我们可以得到：

$$\nabla f(x_k) + \nabla^2 f(x_k) p = 0. \tag{7}$$

Analysis

由于 $\nabla^2 f(x_k)$ 是正定的，它是非奇异的。因此，牛顿法的更新为：

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k). \tag{8}$$

定理

设 f 是定义在 $\mathbb{R}^n$ 上的两次连续可微的函数。假设以下条件成立：

- f 是参数为 m 的强凸函数：存在 $m > 0$，使得对任何 $x \in \mathbb{R}^n$，有 $\nabla^2 f(x) \succeq mI$。
- $\nabla^2 f$ 是 Lipschitz 连续的，参数为 L：存在 $L > 0$，对任何 $x, y \in \mathbb{R}^n$，有 $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|$。

设 $\{x_k\}_{k\geq 0}$ 为牛顿法生成的序列，且 x^* 是 f 在 $\mathbb{R}^n$ 上的唯一最小值。那么对于任意 $k = 0, 1, \ldots$，不等式

$$\|x_{k+1} - x^*\| \leq \frac{L}{2m}\|x_k - x^*\|^2 \tag{9}$$

成立。此外，如果 $\|x_0 - x^*\| \leq \frac{m}{L}$，则有

$$\|x_k - x^*\| \leq \frac{2m}{L}\left(\frac{1}{2}\right)^{2^k}. \tag{10}$$

3 Damped Newton's Method

尽管牛顿法具有快速收敛的特点，但它并不是一个通用的下降方法。在某些情况下，为了使牛顿法更稳定，我们引入一个步长来进行线搜索，从而得到所谓的阻尼牛顿法。以下是阻尼牛顿法的伪代码表示：

算法

Algorithm 2 算法名称

1: **Input:** 目标函数 $f(x)$，约束条件 $x \in X$，精度阈值 $\epsilon > 0$

2: **Output:** 最优解 x^* 和最优值 λ^*

3: 初始化：选择一个初始值 $\lambda^{(0)} \in \mathbb{R}$，设置迭代次数 $k = 0$

4: **repeat**

5: **Step 1:** 求解子问题：

$$x^{(k)} = \arg\max_{x \in X} \{ f(x) - \lambda^{(k)} g(x) \} \tag{11}$$

6: **Step 2:** 计算：

$$\phi(\lambda^{(k)}) = \max_{x \in X} \{ f(x) - \lambda^{(k)} g(x) \} \tag{12}$$

7: **Step 3:** 更新：

$$\lambda^{(k+1)} = \frac{f(x^{(k)})}{g(x^{(k)})} \tag{13}$$

8: 更新迭代次数：$k \leftarrow k + 1$

9: **until** $\phi(\lambda^{(k)}) < \epsilon$

10: 输出最优解：$x^* = x^{(k)}$，最优值：$\lambda^* = \lambda^{(k)}$

Remark

牛顿法在大规模优化问题中可能需要大量的存储和计算资源，这时我们可以利用稀疏矩阵的特性或使用共轭梯度法来求解线性系统，从而提高计算效率。

4 总结

总结

牛顿法是求解强凸函数最优化问题的有效方法，在初始点足够接近最优解时具有二次收敛的性质。然而，其计算复杂度较高，特别是在 Hessian 矩阵稠密或规模较大时。通过引入步长，阻尼牛顿法增强了牛顿法的鲁棒性，使其在较远的初始点也能稳定收敛。