

## 4. 非线性约束的最优化问题

朱天宇

### 1 等式约束的情形

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{x}) = 0 \end{aligned} \quad (1)$$

这里要求  $\mathbf{c}(\mathbf{x}) = (c_1(\mathbf{x}), c_2(\mathbf{x}), \dots, c_m(\mathbf{x}))^\top$ 。

**最优性条件** 记  $\mathbf{A}(\mathbf{x}) = [\nabla \mathbf{c}(\mathbf{x})]^\top = (\nabla c_1(\mathbf{x}), \dots, \nabla c_m(\mathbf{x}))^\top$ ，那么  $\mathbf{x}$  是等式约束问题1的 K-T 点当且仅当存在乘子  $\lambda \in \mathbb{R}^m$ ，使得

$$\nabla f(\mathbf{x}) - \mathbf{A}(\mathbf{x})^\top \lambda = 0$$

且  $\mathbf{x}$  是一个可行点，即  $\mathbf{c}(\mathbf{x}) = 0$ 。讨论这个式子可以关注一下方程的维度。其实际上就是问题1的拉格朗日函数  $L(\mathbf{x}, \lambda)$  的导数

$$\begin{cases} \nabla_{\mathbf{x}} L = \nabla f(\mathbf{x}) - \mathbf{A}(\mathbf{x})^\top \lambda = 0 \\ \nabla_{\lambda} L = -\mathbf{c}(\mathbf{x}) = 0 \end{cases}$$

这是一个无约束的非线性方程组，可以通过 Newton-Raphson 方法求出解。

记  $\mathbf{x}, \lambda$  的计算增量分别为  $\delta_{\mathbf{x}}, \delta_{\lambda}$ ，计算

$$\begin{pmatrix} \mathbf{W}(\mathbf{x}, \lambda) & -\mathbf{A}(\mathbf{x})^\top \\ -\mathbf{A}(\mathbf{x}) & 0 \end{pmatrix} \begin{pmatrix} \delta_{\mathbf{x}} \\ \delta_{\lambda} \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}) - \mathbf{A}(\mathbf{x})^\top \lambda \\ -\mathbf{c}(\mathbf{x}) \end{pmatrix} \quad (2)$$

其中， $\mathbf{W}(\mathbf{x}, \lambda) = \nabla^2 f(\mathbf{x}) - \sum_{i=1}^m \lambda_i \nabla^2 c_i(\mathbf{x})$ ，表示目标函数和约束的二阶导数； $\lambda_{\mathbf{x}}, \delta_{\lambda}$  是迭代方向。这是一个典型的牛顿迭代的步骤。是一维情况的高维延伸。

$$\mathbf{x}^{k+1} - \mathbf{x}^k = \frac{g(\mathbf{x}^k)}{g'(\mathbf{x}^k)}$$

因此，定义价值函数

$$\psi(\mathbf{x}, \lambda) = \|\nabla f(\mathbf{x}) - \mathbf{A}(\mathbf{x})^\top \lambda\|^2 + \|\mathbf{c}(\mathbf{x})\|^2 \quad (3)$$

显然， $\psi(\mathbf{x}, \lambda)$  是一个关于拉格朗日-牛顿方法的下降函数。即满足

$$\nabla \psi(\mathbf{x}, \lambda)^\top \begin{pmatrix} \delta_{\mathbf{x}} \\ \delta_{\lambda} \end{pmatrix} = -2\psi(\mathbf{x}, \lambda) \leq 0$$

#### 1.1 Lagrange-Newton 法

---

**Algorithm 1** 拉格朗日-牛顿法

---

```
1: 给定  $\mathbf{x}^0, k \leftarrow 0, \lambda \in \mathbb{R}^m, \beta \in (0, 1), \epsilon \geq 0$ 
2: while  $\psi(\mathbf{x}^k, \lambda^k) > \epsilon$  do
3:   求解公式2，获得下降方向  $(\delta_{\mathbf{x}^k}, \delta_{\lambda^k})$ ，令  $\alpha_k \leftarrow 1$ 
4:   while  $\psi(\mathbf{x}^k + \alpha_k \delta_{\mathbf{x}^k}, \lambda^k + \alpha_k \delta_{\lambda^k}) > (1 - \beta \alpha_k) \psi(\mathbf{x}^k, \lambda^k)$  do
5:      $\alpha_k = \frac{1}{4} \alpha_k$ 
6:   end while
7:    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \alpha_k \delta_{\mathbf{x}^k}, \lambda^{k+1} \leftarrow \lambda^k + \alpha_k \delta_{\lambda^k}, k \leftarrow k + 1$ 
8: end while
```

---

这个算法首先计算价值函数3，只要价值函数的值大于一个极小的值，就进行从第三行开始的循环。通过公式2来获取下降方向，并且通过回退的方法求出步长，依次进行迭代更新。

## 2 含有不等式约束的情形

### 2.1 逐步二次规划方法

Sequential Quadratic Programming Method, SQP, 逐步二次规划。

可以将公式2写成如下形式

$$\begin{cases} W(x, \lambda) \delta_x + \nabla f(x) = A(x)^\top (\lambda + \delta_\lambda) \\ c(x) + A(x) \delta_x = 0 \end{cases} \quad (4)$$

对比二次规划的问题和 KT 方程组的对应关系, 可以发现, 根据最优性条件可知,  $\delta_{x^k}$  是下列 **二次规划问题** 的 K-T 点:

$$\begin{aligned} \min \quad & \frac{1}{2} d^\top W(x^k, \lambda^k) d + \nabla f(x^k)^\top d \\ \text{s.t.} \quad & c(x^k) + A(x^k) d = 0 \end{aligned} \quad (5)$$

拉格朗日牛顿方法可以理解为逐步求解上述等式约束二次规划的方法。

设  $d^k$  是二次规划问题5的最优解, 那么就可以迭代更新求出问题。

$$x^{k+1} = x^k + \alpha_k d^k$$

设  $\bar{\lambda}^k$  是问题5对应的拉格朗日乘子向量, 相当于4第一个等式中的  $(\lambda + \delta_\lambda)$ 。那么对于  $k \geq 1$  有

$$\lambda^{k+1} = \lambda^k + \alpha_k (\bar{\lambda}^k - \lambda^k)$$

其中  $\alpha_k$  是第  $k$  次的迭代步长。

### 2.2 扩展到不等式约束的情况

考虑一般形式的非线性约束的最优化问题。

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, \quad i \in \mathcal{E} = 1, \dots, m_e \\ & c_i(x) \geq 0, \quad i \in \mathcal{I} = m_e + 1, \dots, m \end{aligned} \quad (6)$$

类似的, 在第  $k$  次迭代中, 求解二次规划子问题:

$$\begin{aligned} \min \quad & \frac{1}{2} d^\top W_k d + g^k d \\ \text{s.t.} \quad & c_i(x^k) + a_i(x^k)^\top d = 0, i \in \mathcal{E} \\ & c_i(x^k) + a_i(x^k)^\top d \geq 0, i \in \mathcal{I} \end{aligned} \quad (7)$$

在这里,  $W_k$  是原问题的拉格朗日函数的 Hessian 阵或者是近似:

$$\begin{aligned} g^k &= \nabla f(x^k) \\ A(x^k) &= (a_1(x^k), \dots, a_m(x^k))^\top = [\nabla c(x^k)]^\top \end{aligned}$$

记子问题7的解为  $d^k$ , 拉格朗日乘子向量为  $\bar{\lambda}^k$ , 那么就有

$$\begin{cases} W_k d^k + g^k = A(x^k)^\top \bar{\lambda}^k \\ \bar{\lambda}_i^k \geq 0, i \in \mathcal{I} \\ c(x^k) + A(x^k) d^k = 0 \end{cases} \quad (8)$$

这个式子和4有相同的格式。  $d^k$  是这个问题的解。同样的, 构造一个价值函数 (罚函数)。定义的方法有多种。例如  $L_1$  罚函数:

$$P(x, \sigma) = f(x) + \sigma \left( \sum_{i=1}^{m_e} |c_i(x)| + \sum_{i=m_e+1}^m |c_i(x)_-| \right)$$

其中,

$$\begin{cases} c_i(x)_- = c_i(x), & i \in \mathcal{E} \\ c_i(x)_- = \min\{0, c_i(x)\}, & i \in \mathcal{I} \end{cases}$$

## 2.3 Han(1977)

一个简单的二次规划算法框架：

---

### Algorithm 2 Han(1977)

---

- 1: 给定  $\mathbf{x}^0, k \leftarrow 0, \mathbf{W}_0 \in \mathbb{R}^{n \times n}, \sigma > 0, \rho \in (0, 1), \epsilon \geq 0$
- 2: 求解子问题8, 获取  $\mathbf{d}^k$
- 3: **while**  $\mathbf{d}^k > \epsilon$  **do**
- 4:   求取适当的  $\alpha_k \in [0, \rho]$ , 使得

$$P(\mathbf{x}^k + \alpha_k \mathbf{d}^k, \sigma) \leq \min_{0 \leq \alpha \leq \rho} P(\mathbf{x}^k + \alpha \mathbf{d}^k, \sigma) + \epsilon_k$$

- 5:    $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \alpha_k \mathbf{d}^k$ , 更新  $\mathbf{W}_{k+1}$ ,  $k \leftarrow k + 1$
  - 6:   求解子问题8, 获取  $\mathbf{d}^k$
  - 7: **end while**
- 

这是一个比较早期的算法框架，现代有很多对此改进的方法，主要改进点是算法第 4 行的部分，更快的获取一个更好的步长。

## 3 罚函数法

对于前述的一般形式的非线性约束的最优化问题。

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E} = 1, \dots, m_e \\ & c_i(\mathbf{x}) \geq 0, \quad i \in \mathcal{I} = m_e + 1, \dots, m \end{aligned} \tag{9}$$

罚函数方法构造关于目标函数  $f(\mathbf{x})$  与约束方程  $\mathbf{c}(\mathbf{x})$  的拥有 **惩罚性质** 的函数：

$$P(\mathbf{x}) = P(f(\mathbf{x}), \mathbf{c}(\mathbf{x}))$$

“**惩罚性质**” 如果一个点是问题的可行点  $\mathbf{x} \in S$ , 那么有  $P(\mathbf{x}) = f(\mathbf{x})$ ; 而当约束条件被破坏的时候,  $P(\mathbf{x}) > f(\mathbf{x})$ 。

比如前述定义的  $\mathbf{c}(\mathbf{x})_-$ ：

$$\begin{cases} c_i(\mathbf{x})_- = c_i(\mathbf{x}), & i \in \mathcal{E} \\ c_i(\mathbf{x})_- = \min\{0, c_i(\mathbf{x})\}, & i \in \mathcal{I} \end{cases}$$

如果一个点是可行点，那么其等式约束、不等式约束都被满足。因此  $\mathbf{c}(\mathbf{x})_-$  的等式约束部分就和问题的等式约束一致；不等式约束部分 ( $c_i(\mathbf{x}) \geq 0$ ) 则为 0 与  $\mathbf{c}(\mathbf{x})$  的较小值。

因此罚函数可以取目标函数与罚项之和：

$$P(\mathbf{x}) = f(\mathbf{x}) + \phi(\mathbf{c}(\mathbf{x})_-)$$

罚项需要满足

$$\phi(0) = 0, \quad \lim_{\|\mathbf{c}\| \rightarrow \infty} \phi(\mathbf{c}) = +\infty$$

且最好和方向没有关系。也就是“径向函数”

### Courant 罚函数

$$P_\sigma(\mathbf{x}) = f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})_-\|_2^2$$

在这里  $\sigma$  是罚因子。

记  $\mathbf{x}(\sigma)$  是无约束问题  $\min_{\mathbf{x} \in \mathbb{R}^n} P_\sigma(\mathbf{x})$  的最优解，且同时也是原问题9的可行点，那么  $\mathbf{x}(\sigma)$  也是该问题的最优解。因为只要可行，那么罚项就等于 0。

如果罚因子  $\sigma$  充分大，那么在优化的时候就会更充分地考虑  $\|\mathbf{c}(\mathbf{x})_-\|^2$ ，此项更加敏感，从而可以保证可行性；如果  $\sigma$  一开始就很大，那么解罚函数的数值层面就比较困难。因此在实际计算中， $\sigma$  都是逐渐增大的。

### 3.1 罚函数方法的迭代步骤

---

**Algorithm 3** 罚函数法框架

---

1: 任选初始点  $\mathbf{x}^0$ , 给定罚因子  $\sigma_0 > 0, \beta > 1, \epsilon > 0, k \leftarrow 0$

2: **while true do**

3:   求解罚函数的无约束问题的极小点, 记为  $\mathbf{x}(\sigma_k)$ , 即

$$\mathbf{x}(\sigma_k) = \arg \min_{\mathbf{x} \in \mathbb{R}^n} P_{\sigma_k}(\mathbf{x})$$

4:   **if**  $\|\mathbf{c}(\mathbf{x}(\sigma_k))\| < \epsilon$  **then**

5:     算法结束, 取  $\mathbf{x}(\sigma_k)$  作为原问题的近似最优解

6:   **else**

7:      $\mathbf{x}^{k+1} \leftarrow \mathbf{x}(\sigma_k), \sigma_{k+1} \leftarrow \beta \sigma_k, k \leftarrow k + 1$

8:   **end if**

9: **end while**

---