MATH 3011 运筹学 **秋季 2023**

Lecture 16: 约束优化 罚函数方法

Lecturer: 陈士祥 *Scribes:* 陈士祥

致谢：感谢北京大学文再文老师提供的《最优化方法》参考讲义

1 问题形式

考虑约束优化问题：

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X}. \end{aligned}$$

其中，$\mathcal{X}$ 为 x 的可行域。约束问题相比于无约束问题的困难：

- 约束优化问题中 x 不能随便取值，梯度下降法所得点不一定在可行域内
- 最优解处目标函数的梯度不一定为零向量

为了解决这些困难，考虑使用**罚函数法**将约束优化问题转化为无约束优化问题处理。

2 二次罚函数方法

2.1 等式问题的二次罚函数法

首先考虑简单情形：仅包含等式约束的约束优化问题

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \end{aligned} \tag{16.1}$$

其中 $x \in \mathbb{R}^n$，$\mathcal{E}$ 为等式约束的指标集，$c_i(x)$ 为连续函数。

定义该问题的二次罚函数为:

$$P_E(x, \sigma) = f(x) + \frac{1}{2}\sigma \sum_{i \in \mathcal{E}} c_i^2(x) \tag{16.2}$$

其中等式右端第二项称为**二次罚函数**，$\sigma > 0$ 称为**罚因子**。

- 由于这种罚函数对不满足约束的点进行惩罚，在迭代过程中点列一般处于可行域之外，因此它也被称为**外点罚函数**.

为了直观理解罚函数的作用，我们给出一个例子:

Example 16.1 考虑优化问题

$$\begin{aligned} \min \quad & x+\sqrt{3}y \\ s.t. \quad & x^2+y^2=1 \end{aligned}$$

容易求得最优解为 $\left(-\frac{1}{2},-\frac{\sqrt{3}}{2}\right)^{\mathrm{T}}$，考虑二次罚函数

$$P_E(x,y,\sigma)=x+\sqrt{3}y+\frac{\sigma}{2}\left(x^2+y^2-1\right)^2$$

并在下图中绘制出 $\sigma=1$ 和 $\sigma=10$ 对应的罚函数的等高线. 当 $\sigma=1$ 时，罚函数的最小值大概为 $x\approx-0.6625, y\approx-1.147$. 而当 $\sigma=10$ 时，出现了两个局部最优解。

图 16.1: 取不同的值时二次罚函数 $P_E(x,y,\sigma)$ 的等高线

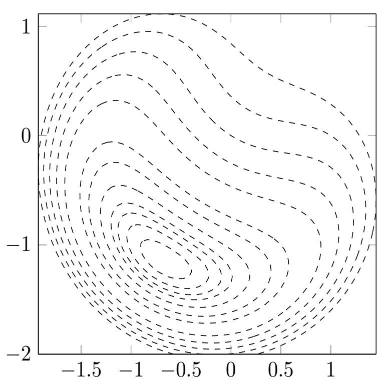

图 16.2: (a) $\sigma=1$

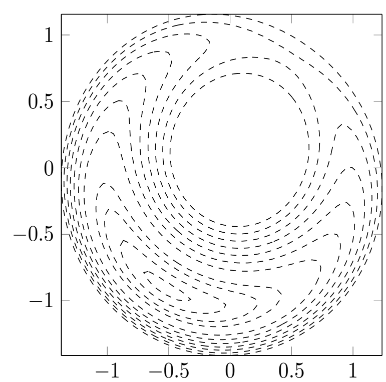

图 16.3: (b) $\sigma=10$

下面这个例子表明，当 σ 选取过小时罚函数可能无下界.

Example 16.2 考虑优化问题

$$\begin{aligned} \min \quad & -x^2+2y^2 \\ s.t. \quad & x=1 \end{aligned}$$

容易求得最优解为 $(1,0)^T$，然而考虑罚函数

$$P_E(x,y,\sigma)=-x^2+2y^2+\frac{\sigma}{2}(x-1)^2$$

对任意的 $\sigma\leq 2$，该罚函数无下界。

上述两个例子表明，罚因子 σ 的选取需要充分大，同时也改变了原问题的性质。

我们从 KKT 条件角度分析原问题和罚函数的性质。

- 原问题的 KKT 条件:

$$\nabla f(x^*) - \sum_{i\in\mathcal{E}} \lambda_i^* \nabla c_i(x^*) = 0$$
$$c_i(x^*) = 0, \quad \forall i \in \mathcal{E}$$

- 添加罚函数项问题的 KKT 条件:

$$\nabla f(x) + \sum_{i\in\mathcal{E}} \sigma c_i(x) \nabla c_i(x) = 0$$

假设两个问题收敛到同一点，对比 KKT 条件 (梯度式)，应有下式成立:

$$\sigma c_i\left(x\right) \approx -\lambda_i^*, \quad \forall i \in \mathcal{E}$$

最优点处乘子 λ^* 固定，为使约束 $c_i(x) = 0$ 成立，需要 $\sigma \to \infty$.

因此，我们有如下的算法。

Algorithm 1 二次罚函数法

给定 $\sigma_1 > 0, x_0, k \leftarrow 1$. 罚因子增长系数 $\rho > 1$.
while 未达到收敛准则 **do**
    以 x^{k-1} 为初始点，求解 $x^k = \arg\min\limits_x P_E\left(x, \sigma_k\right)$
    选取 $\sigma^{k+1} = \rho\sigma_k$.
    $k \leftarrow k+1$
end while

- 考虑罚函数 $P_E(x,\sigma)$ 的海瑟矩阵:

$$\nabla_{xx}^2 P_E(x,\sigma) = \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \sigma c_i(x) \nabla^2 c_i(x) + \sigma \nabla c(x) \nabla c(x)^{\mathrm{T}}$$

- 等号右边的前两项可以使用拉格朗日函数 $L(x,\lambda^*)$ 来近似，即:

$$\nabla_{xx}^2 P_E(x,\sigma) \approx \nabla_{xx}^2 L\left(x,\lambda^*\right) + \sigma \nabla c(x) \nabla c(x)^{\mathrm{T}}$$

- 右边为一个定值矩阵和一个最大特征值趋于正无穷的矩阵，这导致 $\nabla_{xx}^2 P_E(x,\sigma)$ 条件数越来越大，求解子问题的难度也会相应地增加.

- 此时使用梯度类算法求解将会变得非常困难. 若使用牛顿法，则求解牛顿方程本身就是一个非常困难的问题. 因此在实际应用中，我们不可能令罚因子趋于正无穷.

注意事项:

- 选取合适的参数 ρ: 如果 σ_k 太大，则对应的罚函数条件数非常差，导致问题病态。 σ_k 增长过快会使子问题求解困难，σ_k 增长过慢则会增加迭代次数. 另外，也可以自适应地调整 ρ。
- 检测到迭代点发散就应该立即终止迭代并增大罚因子。
- 为保证收敛，子问题求解误差需要趋于零。

作业 16.1 证明如下 *3* 个结论:

结论 1：设 $\sigma_{k+1} > \sigma_k > 0$, 则有 $P_E(x^k, \sigma^k) \leq P_E(x^{k+1}, \sigma^{k+1})$,

$$\sum_{i\in\mathcal{E}} \|c_i(x^k)\|^2 \geq \sum_{i\in\mathcal{E}} \|c_i(x^{k+1})\|^2, \quad f(x^k) \leq f(x^{k+1}).$$

结论 2：设令 $\bar{\mathbf{x}}$ 是原问题(16.1)的最优解，则对任意的 $\sigma^k > 0$ 成立

$$f(\bar{\mathbf{x}}) \geq P_E(x^k, \sigma^k) \geq f(x^k).$$

结论 3：令 $\delta = \sum_{i\in\mathcal{E}} \|c_i(x^k)\|^2$, 则 x^k 也是约束问题

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \sum_{i\in\mathcal{E}} \|c_i(x)\|^2 \leq \delta \end{array}$$

的最优解。

2.2 罚函数法的收敛性结论

下面的定理需要假设每个罚函数 $P_E(x, \sigma_k)$ 都有最小值，并且 $\{x^k\}$ 有极限点。

Theorem 16.1 (二次罚函数法的收敛性 1) 设 x^k 是 $P_E(x, \sigma_k)$ 的全局极小解, σ_k 单调上升趋于无穷, 则 x^k 的每个极限点 x^* 都是原问题的全局极小解.

Proof: 设 $\bar{x}$ 为原问题的极小解. 由 x^k 为 $P_E(x, \sigma_k)$ 的极小解，得 $P_E\left(x^k, \sigma_k\right) \leqslant P_E(\bar{x}, \sigma_k)$，即

$$f\left(x^k\right) + \frac{\sigma_k}{2} \sum_{i\in\mathcal{E}} c_i^2\left(x^k\right) \leqslant f(\bar{x}) + \frac{\sigma_k}{2} \sum_{i\in\mathcal{E}} c_i^2(\bar{x}) = f(\bar{x}) \tag{16.3}$$

整理得:

$$\sum_{i\in\mathcal{E}} c_i^2\left(x^k\right) \leqslant \frac{2}{\sigma_k}\left(f(\bar{x}) - f\left(x^k\right)\right) \tag{16.4}$$

设 x^* 是 x^k 的一个极限点,不妨设 $\{x^k\}$ 的子列 $x^{k_n} \to x^*$。在(16.4)式中令 $k_n \to \infty$,得 $\sum_{i\in\mathcal{E}} c_i^2(x^*) = 0$. 由此易知，$x^*$ 为原问题的可行解，又由(16.3)式知 $f\left(x^k\right) \leqslant f(\bar{x})$，取极限得 $f(x^*) \leqslant f(\bar{x})$，故 x^* 为全局极小解。 ■

由于定理 1 需要每个罚函数 $P_E(x,\sigma_k)$ 解出全局最小值。这个要求比较高。下面的定理给出更弱的情况下的收敛结果，即求解子问题时，只需满足一阶最优条件下的收敛结论。

Theorem 16.2 (二次罚函数法的收敛性 2) *设 $f(x)$ 与 $c_i(x)$ ($i \in \mathcal{E}$) 连续可微，正数序列 $\varepsilon_k \to 0$, $\sigma_k \to +\infty$。在算法 1中，子问题的解 x^{k+1} 满足 $\left\|\nabla_x P_E\left(x^k,\sigma_k\right)\right\| \le \varepsilon_k$，而对 x^k 的任何极限点 x^*，都有 $\{\nabla c_i(x^*), i\in\mathcal{E}\}$ 线性无关，则 x^* 是等式约束最优化问题(16.1) 的 KKT 点，且*

$$\lim_{k\to\infty}\left(-\sigma_k c_i\left(x^k\right)\right) = \lambda_i^*, \quad \forall i \in \mathcal{E}$$

其中 λ_i^* 是约束 $c_i(x^*) = 0$ 对应的拉格朗日乘子.

关于上述定理，我们有如下说明。

- 不管 $\{\nabla c_i(x^*)\}$ 是否线性无关，通过算法 1给出解 x^k 的聚点总是 $\phi(x) = \|c(x)\|^2$ 的一个稳定点. 这说明即便没有找到可行解，我们也找到了使得约束 $c(x) = 0$ 违反度相对较小的一个解.
- 定理 16.2虽然不要求每一个子问题精确求解，但要获得原问题的解，子问题解的精度需要越来越高.

2.3 一般约束问题的二次罚函数

考虑不等式约束问题:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \leqslant 0, i \in \mathcal{I} \end{aligned}$$

定义该问题的二次罚函数为:

$$P_I(x,\sigma) = f(x) + \frac{1}{2}\sigma\sum_{i\in\mathcal{I}}\tilde{c}_i^2(x)$$

其中 $\tilde{c}_i(x)$ 定义为:

$$\tilde{c}_i(x) = \max\{c_i(x), 0\}.$$

即，我们只对违反不等式约束的部分进行惩罚。

注：$h(t) = (\min\{t,0\})^2$ 关于 t 可导，故 $P_I(x,\sigma)$ 梯度存在，所以可以使用梯度类算法求解。

现在考虑一般约束问题:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x)=0, i\in\mathcal{E} \\ & c_i(x)\leqslant 0, i\in\mathcal{I}. \end{aligned} \tag{16.5}$$

定义该问题的二次罚函数为:

$$P_E(x,\sigma)=f(x)+\frac{1}{2}\sigma\left[\sum_{i\in\mathcal{E}}c_i^2(x)+\sum_{i\in\mathcal{I}}\tilde{c}_i^2(x)\right].$$

其中等式右端第二项称为惩罚项，$\tilde{c}_i(x)$ 的定义如(16.5)式，常数 $\sigma>0$ 称为罚因子.

定理 16.1和 16.2对于一般约束问题同样成立。

3 应用举例

3.1 低秩矩阵恢复

某视频网站提供了约 48 万用户对 1 万 7 千多部电影的上亿条评级数据，希望对用户的电影评级进行预测，从而改进用户电影推荐系统，为每个用户更有针对性地推荐影片.

显然每一个用户不可能看过所有的电影，每一部电影也不可能收集到全部用户的评级. 电影评级由用户打分 1 星到 5 星表示，记为取值 1- 5 的整数. 我们将电影评级放在一个矩阵 M 中，矩阵 M 的每一行表示不同用户，每一列表示不同电影. 由于用户只对看过的电影给出自己的评价，矩阵 M 中很多元素是未知的。

$$\begin{array}{c} \\ \text{用户 1} \\ \text{用户 2} \\ \text{用户 3} \\ \text{用户 4} \\ \vdots \\ \text{用户m} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{电影 1} & \text{电影 2} & \text{电影 3} & \text{电影 4} & \cdots & \text{电影n} \end{array} \\ \begin{bmatrix} 4 & ? & ? & 3 & \cdots & ? \\ ? & 2 & 4 & ? & \cdots & ? \\ 3 & ? & ? & ? & \cdots & ? \\ 2 & ? & 5 & ? & \cdots & ? \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ ? & 3 & ? & 4 & \cdots & ? \end{bmatrix} \end{array}$$

该问题在推荐系统、图像处理等方面有着广泛的应用。由于用户对电影的偏好可进行分类，按年龄可分为：年轻人，中年人，老年人；且电影也能分为不同的题材：战争片，悬疑片，言情片等。故这类问题隐含的假设为补全后的矩阵应为低秩的。即矩阵的行与列会有“合作”的特性，故该问题具有别名“collaborative filtering”。除此之外，由于低秩矩阵可分解为两个低秩矩阵的乘积，所以低秩限制下的矩阵补全问题是比较实用的，这样利于储存且有更好的诠释性。

由上述分析可以引出该问题：

- 令 Ω 是矩阵 M 中所有已知评级元素的下标的集合，则该问题可以初步描述为构造一个矩阵 X，使得在给定位置的元素等于已知评级元素，即满足 $X_{ij}=M_{ij}$, $(i,j)\in\Omega$.

- 低秩矩阵恢复（low rank matrix completion）

$$\begin{aligned}\min_{X\in\mathbb{R}^{m\times n}} \quad & \operatorname{rank}(X), \\ \text{s.t.} \quad & X_{ij}=M_{ij},\ (i,j)\in\Omega.\end{aligned} \tag{16.6}$$

  rank(X) 正好是矩阵 X 所有非零奇异值的个数

- 矩阵 X 的核范数（nuclear norm）为矩阵所有奇异值的和，即：$\|X\|_*=\sum_i\sigma_i(X)$，最小化核范数可以近似的看成最小化矩阵的秩，因此我们有如下的优化问题：

$$\begin{aligned}\min_{X\in\mathbb{R}^{m\times n}} \quad & \|X\|_*, \\ \text{s.t.} \quad & X_{ij}=M_{ij},\ (i,j)\in\Omega.\end{aligned} \tag{16.7}$$

对于上述问题，我们引入等式约束的二次罚函数，

$$\min \quad \|X\|_*+\frac{\sigma}{2}\sum_{(i,j)\in\Omega}(X_{ij}-M_{ij})^2$$

令 $\sigma=\frac{1}{\mu}$，即有等价形式的优化问题：

$$\min \quad \mu\|X\|_*+\frac{1}{2}\sum_{(i,j)\in\Omega}(X_{ij}-M_{ij})^2 \tag{16.8}$$

当然，核范数是非光滑函数，我们可以用次梯度法求解上述问题，我们会在后面的课程介绍次梯度法。

Algorithm 2 矩阵补全问题求解的罚函数法

```
给定初值 X^0，最终参数 μ，初始参数 μ_0，因子 γ ∈ (0,1), k ← 1
while μ_k ≥ μ do
  以 X^{k-1} 为初值，μ = μ_k 为正则化参数求解问题(16.8)，得 X^k
  if μ_k = μ then
    停止迭代，输出 X^k
  else
    更新罚因子 μ_{k+1} = max{μ, γμ_k}
    k ← k + 1
  end if
end while
```

4 其他罚函数方法

4.1 精确罚函数方法

- 由于二次罚函数存在数值困难，并且与原问题的解存在误差，故考虑精确罚函数.
- **精确罚函数**，是一种问题求解时不需要令罚因子趋于正无穷（或零）的罚函数. 常用的精确罚函数是 ℓ_1 罚函数.
- 二次罚函数对应的问题是光滑的,ℓ_1 罚函数对应的问题是非光滑的。

定义一般约束优化问题的 ℓ_1 罚函数:

$$P(x,\sigma) = f(x) + \sigma\left[\sum_{i\in\mathcal{E}}|c_i(x)| + \sum_{i\in\mathcal{I}}\tilde{c}_i(x)\right]$$

这里用绝对值代替二次惩罚项，下面的定理揭示了 ℓ_1 罚函数的精确性。

Theorem 16.3 (精确罚函数法的收敛性) *设 x^* 是一般约束优化问题(16.5)的一个严格局部极小解，且满足 KKT 条件，其对应的拉格朗日乘子为 $\lambda_i^*, i\in\mathcal{E}\cup\mathcal{I}$，则当罚因子 $\sigma > \sigma^*$ 时，x^* 也为 $P(x,\sigma)$ 的一个局部极小解，其中*

$$\sigma^* = \|\lambda^*\|_\infty \overset{def}{=} \max_i |\lambda_i^*|.$$

另一方面，存在 $\hat{\sigma} > 0$, 对于 $\sigma \geq \hat{\sigma}$，如果 $\hat{x}$ 是罚函数 $P(x,\sigma)$ 的稳定点. 那么，如果 $\hat{x}$ 是一般约束优化问题(16.5)的可行点，则 $\hat{x}$ 也满足(16.5)的 KKT 条件。

定理 16.3说明对于精确罚函数，罚因子充分大（不是正无穷），原问题的极小值点是 ℓ_1 罚函数的极小值点，这和定理 16.1是有区别的. 反之，罚函数的稳定点若是可行点，在较弱的假设下，则也是约束问题的 KKT 点。

我们有如下的传统精确罚函数方法。

Algorithm 3 精确罚函数法

1: 给定 $\sigma_1 > 0, x_0, k \leftarrow 0$. 罚因子增长系数 $\rho > 1$.
2: **while** 未达到收敛准则 **do**
3: 　以 x^{k-1} 为初始点，求解 $x^k = \arg\min_x \left\{f(x) + \sigma\left[\sum_{i\in\mathcal{E}}|c_i(x)| + \sum_{i\in\mathcal{I}}\tilde{c}_i(x)\right]\right\}$
4: 　选取 $\sigma^{k+1} = \rho\sigma_k$.
5: 　$k \leftarrow k+1$
6: **end while**

- 取 ρ 为固定值是一种在实际中行之有效的方法，然而也可能出现：
  - 初始罚因子过小，迭代次数增加，且最优解可能远离原问题最优解
  - 罚因子过大时子问题求解困难，此时需要适当减小罚因子
- 子问题求解的初始点取法不唯一。一般取上一次子问题求解的最优值点作为下一次子问题求解的起点。

除了 ℓ_1 范数，可以用更一般的范数定义精确罚函数法：

$$P(x,\sigma) = f(x) + \mu \left\| c_{\mathcal{E}}(x) \right\| + \mu \left\| [c_{\mathcal{I}}(x)]^+ \right\|$$

其中，$\|\cdot\|$ 可以是任意的向量范数，$[c_{\mathcal{I}}(x)]^+$ 为向量各分量取 $\max\{0,x\}$。则我们可以推广定理 16.3，将 $\|\cdot\|_\infty$ 替换为 $\|\cdot\|_D$ (这里 $\|\cdot\|_D$ 表示 $\|\cdot\|$ 的对偶范数)。对偶范数的定义如下：

$$\|x\|_D = \max_{\|y\|=1} x^T y.$$

常见的对偶范数：

- $\|\cdot\|_1$ 和 $\|\cdot\|_\infty$ 互为对偶;
- ℓ_2 范数的对偶是它自身.

4.2 精确罚函数的非光滑性

下面说明，精确罚函数必然是非光滑的。
为简化讨论，假设仅有一条等式约束 $c_1(x) = 0$。设罚函数的形式为：

$$P(x,\sigma) = f(x) + \sigma h(c_1(x)).$$

其中，函数 $h : \mathbb{R} \to \mathbb{R}$ 满足 $h(y) \geq 0,\ \forall y \in \mathbb{R}$ 且 $h(0) = 0$.

若函数 h 连续可微，因为 $y = 0$ 是 $h(y)$ 最小值点，则有 $\nabla h(0) = 0$ 成立。故对于 $P(x,\sigma)$ 最优点 x^*，有

$$0 = \nabla P(x^*,\sigma) = \nabla f(x^*) + \sigma \nabla c_1(x^*) \nabla h(c_1(x^*)) = \nabla f(x^*).$$

然而，在约束优化问题中，f 取到最小值时，其梯度不一定为 0。这说明假设 h 连续可微是不正确的，即罚函数项必须是非光滑的。
另一方面，正是罚函数项的非光滑性，克服了原函数在最优点处的梯度，才能在充分大的罚因子下实现精确求解。