Practical Optimization Algorithms

徐 翔

数学科学学院
浙江大学

June 7, 2022

第十二讲: 罚函数和增广拉格朗日法 (Penalty and Augmented Lagrangian Methods)

简介(Introduction)

Outline

- 在求解约束优化问题时，很多方法都是把约束条件加入到目标函数中。

Outline

- 在求解约束优化问题时，很多方法都是把约束条件加入到目标函数中。
- 三种方式：二次罚函数(quadratic penalty)
  非光滑精确罚函数 (nonsmooth exact penalty)
  增广拉格朗日法或者乘子法
  (augmented Lagrangian method or method of multiplier)

Outline

- 在求解约束优化问题时，很多方法都是把约束条件加入到目标函数中。
- 三种方式：二次罚函数(quadratic penalty)
  非光滑精确罚函数 (nonsmooth exact penalty)
  增广拉格朗日法或者乘子法
  (augmented Lagrangian method or method of multiplier)
- 简单介绍log-barrier method，用log项阻止迭代过程靠近边界，是非线性规划内点法的基础。

二次罚函数(Quadratic Penalty): Motivation

首先考虑等式约束问题

二次罚函数(Quadratic Penalty): Motivation

首先考虑等式约束问题

- 一般形式

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, i \in \mathcal{E}$$

二次罚函数(Quadratic Penalty): Motivation

首先考虑等式约束问题

- 一般形式

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, i \in \mathcal{E}$$

- 相应的二次罚函数形式为

$$Q(x;\mu) = f(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i^2(x),$$

其中$\mu > 0$是惩罚参数(penalty parameter).

二次罚函数(Quadratic Penalty): Motivation

首先考虑等式约束问题

- 一般形式

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, i \in \mathcal{E}$$

- 相应的二次罚函数形式为

$$Q(x;\mu) = f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x),$$

其中$\mu > 0$是惩罚参数(penalty parameter).

- 当$\mu \to \infty$，我们逐渐提高对违反约束条件的惩罚。

二次罚函数(Quadratic Penalty): Motivation

首先考虑等式约束问题

- 一般形式

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, i \in \mathcal{E}$$

- 相应的二次罚函数形式为

$$Q(x;\mu) = f(x) + \frac{\mu}{2} \sum_{i\in\mathcal{E}} c_i^2(x),$$

其中$\mu > 0$是惩罚参数(penalty parameter).

- 当$\mu \to \infty$，我们逐渐提高对违反约束条件的惩罚。
- 因此直观地，我们可以考虑$\mu_k \to \infty$，求解$Q(x;\mu_k)$得到x_k。

二次罚函数(Quadratic Penalty): Motivation

首先考虑等式约束问题

- 一般形式

$$\min_x f(x) \quad \text{subject to } c_i(x) = 0, i \in \mathcal{E}$$

- 相应的二次罚函数形式为

$$Q(x;\mu) = f(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i^2(x),$$

  其中$\mu > 0$是惩罚参数(penalty parameter).

- 当$\mu \to \infty$，我们逐渐提高对违反约束条件的惩罚。
- 因此直观地，我们可以考虑$\mu_k \to \infty$，求解$Q(x;\mu_k)$得到x_k。
- 我们可以使用无约束优化问题的方法求解$Q(x;\mu_k)$. 而且可以把 x_{k-1}当成很好的初值。选择好的 $\{\mu_k\}$序列以及好的初值，可以使得在每次迭代中只需要很小的计算量。

二次罚函数(Quadratic Penalty): Motivation

例1： $\min x_1 + x_2$ subject to $x_1^2 + x_2^2 - 2 = 0.$

二次罚函数(Quadratic Penalty): Motivation

$$\text{例1：} \quad \min x_1 + x_2 \quad \text{subject to } x_1^2 + x_2^2 - 2 = 0.$$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

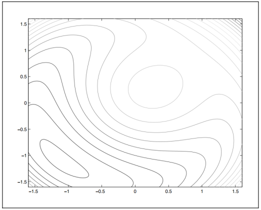
$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$

二次罚函数(Quadratic Penalty): Motivation

$$例1: \quad \min x_1 + x_2 \quad \text{subject to } x_1^2 + x_2^2 - 2 = 0.$$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$

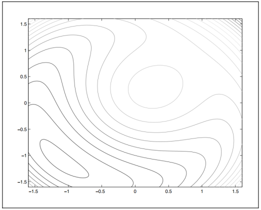


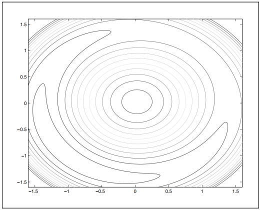
二次罚函数(Quadratic Penalty): Motivation

例1： $\min x_1 + x_2 \quad \text{subject to } x_1^2 + x_2^2 - 2 = 0.$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$



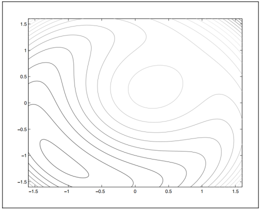


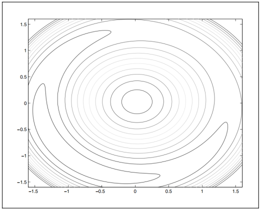
二次罚函数(Quadratic Penalty): Motivation

例1： $\min x_1 + x_2$ subject to $x_1^2 + x_2^2 - 2 = 0.$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$





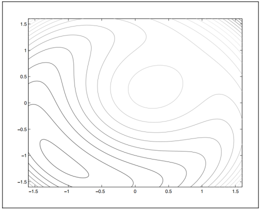
- 在左图中，有一个局部极小值$(-1.1,-1.1)^T$,

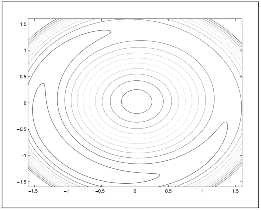
二次罚函数(Quadratic Penalty): Motivation

例1： $\min x_1 + x_2$ subject to $x_1^2 + x_2^2 - 2 = 0.$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$





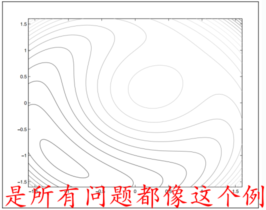
- 在左图中，有一个局部极小值$(-1.1,-1.1)^T$,
- 在右图中，局部极小值更靠近$(-1,-1)^T$.

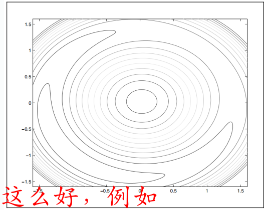
二次罚函数(Quadratic Penalty): Motivation

例1： $\min x_1 + x_2$ subject to $x_1^2 + x_2^2 - 2 = 0.$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$





- 在左图中，有一个局部极小值$(-1.1,-1.1)^T$,
- 在右图中，局部极小值更靠近$(-1,-1)^T$.

不是所有问题都像这个例子这么好，例如

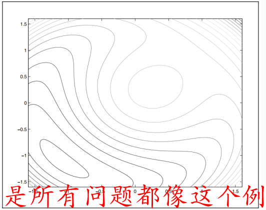
$$\min -5x_1^2 + x_2^2 \quad \text{subject to } x_1 = 1$$

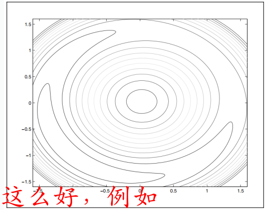
二次罚函数(Quadratic Penalty): Motivation

$$\text{例1：}\quad \min x_1 + x_2 \quad \text{subject to } x_1^2 + x_2^2 - 2 = 0.$$

精确解是$(-1,-1)^T$, 对应的二次罚函数是

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}(x_1^2 + x_2^2 - 2)^2.$$





- 在左图中，有一个局部极小值$(-1.1,-1.1)^T$,
- 在右图中，局部极小值更靠近$(-1,-1)^T$.

不是所有问题都像这个例子这么好，例如

$$\min -5x_1^2 + x_2^2 \quad \text{subject to } x_1 = 1$$

解是$(1,0)$，但是当$\mu < 10$时，罚函数都是无界的。因此对$\mu < 10$时，算法都是发散的。这种缺陷是广泛存在的。

二次罚函数: Algorithmic Framework

二次罚函数: Algorithmic Framework

- 对于一般的约束优化问题

$$\min_{x} \quad \text{subject to } c_i(x) = 0,\ i \in \mathcal{E},\ c_i(x) \geq 0\ i \in \mathcal{I}.$$

二次罚函数: Algorithmic Framework

- 对于一般的约束优化问题

$$\min_x \quad \text{subject to } c_i(x) = 0,\ i \in \mathcal{E},\ c_i(x) \geq 0\ i \in \mathcal{I}.$$

- 定义相关的二次罚函数

$$Q(x;\mu) = f(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i(x)^2 + \frac{\mu}{2}\sum_{i\in\mathcal{I}}([c_i(x)]^-)^2$$

这里$[y]^- = \max(-y, 0)$.

二次罚函数: Algorithmic Framework

- 对于一般的约束优化问题

$$\min_x \quad \text{subject to } c_i(x) = 0,\ i \in \mathcal{E},\ c_i(x) \geq 0\ i \in \mathcal{I}.$$

- 定义相关的二次罚函数

$$Q(x;\mu) = f(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i(x)^2 + \frac{\mu}{2}\sum_{i\in\mathcal{I}}([c_i(x)]^-)^2$$

这里$[y]^- = \max(-y, 0)$.

- Q可能没有目标函数光滑，有可能不能二阶可导。

二次罚函数: Algorithmic Framework

Algorithm 1: 二次罚函数算法

给定 $\mu_0 > 0$, 非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
  寻找$Q(\cdot;\mu_k)$的近似解x_k, 使得$\|\nabla_x Q(\cdot;\mu_k)\| \le \tau_k$;
  if 满足了收敛性测试
    stop $x^* \approx x_k$;
  end(if)
  选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择一个新的初始值x_{k+1}^s;
end(for)

二次罚函数: Algorithmic Framework

Algorithm 1: 二次罚函数算法

给定 $\mu_0 > 0$, 非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
  寻找$Q(\cdot;\mu_k)$的近似解x_k, 使得$\|\nabla_x Q(\cdot;\mu_k)\| \le \tau_k$;
  if 满足了收敛性测试
    stop $x^* \approx x_k$;
  end(if)
  选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择一个新的初始值x_{k+1}^s;
end(for)

- $\{\mu_k\}$可以根据罚函数求解的困难程度自适应地选择。

二次罚函数: Algorithmic Framework

Algorithm 1: 二次罚函数算法

给定 $\mu_0 > 0$, 非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
    寻找$Q(\cdot;\mu_k)$的近似解x_k, 使得$\|\nabla_x Q(\cdot;\mu_k)\| \le \tau_k$;
    if 满足了收敛性测试
        stop $x^* \approx x_k$;
    end(if)
    选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
    选择一个新的初始值x_{k+1}^s;
end(for)

- $\{\mu_k\}$可以根据罚函数求解的困难程度自适应地选择。如果对μ_k求解比较困难，那么可以选个小一点 的$\mu_{k+1} = 1.5\mu_k$，

二次罚函数: Algorithmic Framework

Algorithm 1: 二次罚函数算法

给定 $\mu_0 > 0$, 非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
  寻找$Q(\cdot; \mu_k)$的近似解x_k, 使得$\|\nabla_x Q(\cdot; \mu_k)\| \le \tau_k$;
  if 满足了收敛性测试
    stop $x^* \approx x_k$;
  end(if)
  选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择一个新的初始值x_{k+1}^s;
end(for)

- $\{\mu_k\}$可以根据罚函数求解的困难程度自适应地选择。如果对μ_k求解比较困难，那么可以选个小一点 的$\mu_{k+1} = 1.5\mu_k$，如果比较容易求解，那么可以选个比较激进的$\mu_{k+1} = 10\mu_k$。

二次罚函数: Algorithmic Framework

Algorithm 1: 二次罚函数算法

给定 $\mu_0 > 0$, 非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
    寻找$Q(\cdot;\mu_k)$的近似解x_k, 使得$\|\nabla_x Q(\cdot;\mu_k)\| \le \tau_k$;
    if 满足了收敛性测试
        stop $x^* \approx x_k$;
    end(if)
    选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
    选择一个新的初始值x_{k+1}^s;
end(for)

- $\{\mu_k\}$可以根据罚函数求解的困难程度自适应地选择。如果对μ_k求解比较困难，那么可以选个小一点 的$\mu_{k+1} = 1.5\mu_k$，如果比较容易求解，那么可以选个比较激进的$\mu_{k+1} = 10\mu_k$。
- τ_k的选取只要求$\tau_k \to 0$，就可以保证在迭代过程中越来越逼近精确解。

二次罚函数：Algorithmic Framework

Algorithm 1：二次罚函数算法

给定 $\mu_0 > 0$，非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
  寻找$Q(\cdot;\mu_k)$的近似解x_k，使得$\|\nabla_x Q(\cdot;\mu_k)\| \le \tau_k$;
  if 满足了收敛性测试
    stop $x^* \approx x_k$;
  end(if)
  选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择一个新的初始值x_{k+1}^s;
end(for)

- $\{\mu_k\}$可以根据罚函数求解的困难程度自适应地选择。如果对μ_k求解比较困难，那么可以选个小一点 的$\mu_{k+1} = 1.5\mu_k$，如果比较容易求解，那么可以选个比较激进的$\mu_{k+1} = 10\mu_k$。
- τ_k的选取只要求$\tau_k \to 0$，就可以保证在迭代过程中越来越逼近精确解。
- 当μ_k比较小的时候，迭代的解可能会远离可行域，$\|\nabla_x Q(\cdot;\mu_k)\| \le \tau_k$停止测试不一定能够满足。

二次罚函数: Algorithmic Framework

Algorithm 1: 二次罚函数算法

给定 $\mu_0 > 0$, 非负的 $\tau_k \to 0$ 以及一个初始点x_0^s;
for $k = 0, 1, 2, \cdots$
  寻找$Q(\cdot; \mu_k)$的近似解x_k, 使得$\|\nabla_x Q(\cdot; \mu_k)\| \le \tau_k$;
  if 满足了收敛性测试
    stop $x^* \approx x_k$;
  end(if)
  选择一个新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择一个新的初始值x_{k+1}^s;
end(for)

- $\{\mu_k\}$可以根据罚函数求解的困难程度自适应地选择。如果对μ_k求解比较困难，那么可以选个小一点 的$\mu_{k+1} = 1.5\mu_k$，如果比较容易求解，那么可以选个比较激进的$\mu_{k+1} = 10\mu_k$。
- τ_k的选取只要求$\tau_k \to 0$，就可以保证在迭代过程中越来越逼近精确解。
- 当μ_k比较小的时候，迭代的解可能会远离可行域，$\|\nabla_x Q(\cdot; \mu_k)\| \le \tau_k$停止测试不一定能够满足。 一个常用的策略是，当发现违反约束下降不够快，或者迭代不收敛时，需要加大μ_k.

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx} Q(x;\mu_k)$的条件数会变的很大。

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx} Q(x;\mu_k)$的条件数会变的很大。
- 这个性质会使得拟牛顿法、共轭梯度法失效。

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx} Q(x;\mu_k)$的条件数会变的很大。
- 这个性质会使得拟牛顿法、共轭梯度法失效。虽然牛顿法本身不受坏条件数的影响，但是当μ_k很大时，也会遇到另外的困难

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx} Q(x;\mu_k)$的条件数会变的很大。
- 这个性质会使得拟牛顿法、共轭梯度法失效。虽然牛顿法本身不受坏条件数的影响，但是当μ_k很大时，也会遇到另外的困难
  - 首先，坏条件数可能会影响求解牛顿步，

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx}Q(x;\mu_k)$的条件数会变的很大。
- 这个性质会使得拟牛顿法、共轭梯度法失效。虽然牛顿法本身不受坏条件数的影响，但是当μ_k很大时，也会遇到另外的困难
    - 首先，坏条件数可能会影响求解牛顿步，但这个不是致命的，可以通过转化牛顿方程克服。

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx} Q(x;\mu_k)$的条件数会变的很大。
- 这个性质会使得拟牛顿法、共轭梯度法失效。虽然牛顿法本身不受坏条件数的影响，但是当μ_k很大时，也会遇到另外的困难
    - 首先，坏条件数可能会影响求解牛顿步，但这个不是致命的，可以通过转化牛顿方程克服。
    - 当x接近$Q(x;\mu_k)$的极小值时，其等高线围的区域是香蕉形状而不是椭圆形状。但是牛顿法最适用的局部展开 区域是椭圆区域，这说明可能牛顿法可能收敛会变慢。

二次罚函数: Algorithmic Framework

- 当只有等式约束时，$Q(x;\mu_k)$是光滑的，无约束优化问题的算法都可以使用。
- 当μ_k变大时，优化过程会越来越困难，除非我们有好的技巧计算搜索方向。
- 原因之一是当$x \to x_k$时，$\nabla^2_{xx}Q(x;\mu_k)$的条件数会变的很大。
- 这个性质会使得拟牛顿法、共轭梯度法失效。虽然牛顿法本身不受坏条件数的影响，但是当μ_k很大时，也会遇到另外的困难
    - 首先，坏条件数可能会影响求解牛顿步，但这个不是致命的，可以通过转化牛顿方程克服。
    - 当x接近$Q(x;\mu_k)$的极小值时，其等高线围的区域是香蕉形状而不是椭圆形状。但是牛顿法最适用的局部展开 区域是椭圆区域，这说明可能牛顿法可能收敛会变慢。这个一般可以通过选择好的初始值来克服， 比如可以选取$x^s_{k+1} = x_k$ 并且μ_{k+1}只比μ_k大一些。

二次罚函数: Convergence Analysis

假设$Q(x,\mu_k)$仅有有限个极小子。首先我们看只有等式约束的情形

二次罚函数: Convergence Analysis

假设$Q(x,\mu_k)$仅有有限个极小子。首先我们看只有等式约束的情形

定理1：（精确求解$Q(x;\mu_k)$）

假设x_k是$Q(x;\mu_k)$的精确的全局极小子，且$\mu_k \to \infty$。则$\{x_k\}$的每个极限点x^*是原极小化问题的全局解。

二次罚函数: Convergence Analysis

假设$Q(x, \mu_k)$仅有有限个极小子。首先我们看只有等式约束的情形

定理1：（精确求解$Q(x; \mu_k)$）

假设x_k是$Q(x; \mu_k)$的精确的全局极小子，且$\mu_k \to \infty$。则$\{x_k\}$的每个极限点x^*是原极小化问题的全局解。

证明：

- 假设$\bar{x}$是原问题的全局解，则$f(\bar{x}) \le f(x)$，对于任意可行域内的点$c_i(x) = 0,\ i \in \mathcal{E}$.

二次罚函数：Convergence Analysis

假设$Q(x,\mu_k)$仅有有限个极小子。首先我们看只有等式约束的情形

定理1：（精确求解$Q(x;\mu_k)$）

假设x_k是$Q(x;\mu_k)$的精确的全局极小子，且$\mu_k \to \infty$。则$\{x_k\}$的每个极限点x^*是原极小化问题的全局解。

证明：

- 假设$\bar{x}$是原问题的全局解，则$f(\bar{x}) \leq f(x)$，对于任意可行域内的点$c_i(x) = 0,\ i \in \mathcal{E}$.
- 由于x_k是每个子问题$Q(x;\mu_k)$的全局极小解，我们有$Q(x_k;\mu_k) \leq Q(\bar{x};\mu_k)$, 即

二次罚函数：Convergence Analysis

假设$Q(x,\mu_k)$仅有有限个极小子。首先我们看只有等式约束的情形

定理1：（精确求解$Q(x;\mu_k)$）

假设x_k是$Q(x;\mu_k)$的精确的全局极小子，且$\mu_k \to \infty$。则$\{x_k\}$的每个极限点x^*是原极小化问题的全局解。

证明：

- 假设$\bar{x}$是原问题的全局解，则$f(\bar{x}) \le f(x)$，对于任意可行域内的点$c_i(x) = 0,\ i \in \mathcal{E}$.
- 由于x_k是每个子问题$Q(x;\mu_k)$的全局极小解，我们有$Q(x_k;\mu_k) \le Q(\bar{x};\mu_k)$，即

$$f(x_k) + \frac{\mu_k}{2}\sum_{i\in\mathcal{E}} c_i^2(x_k) \le f(\bar{x}) + \frac{\mu_k}{2}\sum_{i\in\mathcal{E}} c_i^2(\bar{x}) = f(\bar{x})$$

二次罚函数: Convergence Analysis

假设$Q(x,\mu_k)$仅有有限个极小子。首先我们看只有等式约束的情形

定理1：（精确求解$Q(x;\mu_k)$）

假设x_k是$Q(x;\mu_k)$的精确的全局极小子，且$\mu_k \to \infty$。则$\{x_k\}$的每个极限点x^*是原极小化问题的全局解。

证明：

- 假设$\bar{x}$是原问题的全局解，则$f(\bar{x}) \le f(x)$，对于任意可行域内的点$c_i(x) = 0,\ i \in \mathcal{E}$.
- 由于x_k是每个子问题$Q(x;\mu_k)$的全局极小解，我们有$Q(x_k;\mu_k) \le Q(\bar{x};\mu_k)$，即

$$f(x_k) + \frac{\mu_k}{2}\sum_{i\in\mathcal{E}} c_i^2(x_k) \le f(\bar{x}) + \frac{\mu_k}{2}\sum_{i\in\mathcal{E}} c_i^2(\bar{x}) = f(\bar{x})$$

- 进而可以得到

$$\sum_{i\in\mathcal{E}} c_i^2(x_k) \le \frac{2}{\mu_k}[f(\bar{x}) - f(x_k)].$$

二次罚函数: Convergence Analysis

二次罚函数: Convergence Analysis

- 假设$\{x_k\}$有一个极限点x^*，即存在一个无限子列$\mathcal{K}$, 有$\lim_{k\in\mathcal{K}} x_k = x^*$

二次罚函数: Convergence Analysis

- 假设$\{x_k\}$有一个极限点x^*，即存在一个无限子列$\mathcal{K}$, 有$\lim_{k\in\mathcal{K}} x_k = x^*$
- 对上式两边取极限$k \in \mathcal{K}$，得到

$$\sum_{i\in\mathcal{E}} c_i^2(x^*) = \lim_{k\in\mathcal{K}} \sum_{i\in\mathcal{E}} c_i^2(x_k) \leq \lim_{k\in\mathcal{K}} \frac{2}{\mu_k}[f(\bar{x}) - f(x_k)] = 0.$$

二次罚函数: Convergence Analysis

- 假设$\{x_k\}$有一个极限点x^*，即存在一个无限子列$\mathcal{K}$, 有$\lim\limits_{k\in\mathcal{K}} x_k = x^*$
- 对上式两边取极限$k \in \mathcal{K}$，得到

$$\sum_{i\in\mathcal{E}} c_i^2(x^*) = \lim_{k\in\mathcal{K}} \sum_{i\in\mathcal{E}} c_i^2(x_k) \leq \lim_{k\in\mathcal{K}} \frac{2}{\mu_k}[f(\bar{x}) - f(x_k)] = 0.$$

即 $c_i(x^*) = 0$，x^*落在可行域内。

二次罚函数: Convergence Analysis

- 假设$\{x_k\}$有一个极限点x^*，即存在一个无限子列$\mathcal{K}$, 有$\lim_{k\in\mathcal{K}} x_k = x^*$
- 对上式两边取极限$k \in \mathcal{K}$，得到

$$\sum_{i\in\mathcal{E}} c_i^2(x^*) = \lim_{k\in\mathcal{K}} \sum_{i\in\mathcal{E}} c_i^2(x_k) \le \lim_{k\in\mathcal{K}} \frac{2}{\mu_k}[f(\bar{x}) - f(x_k)] = 0.$$

即 $c_i(x^*) = 0$，x^*落在可行域内。
- 再对

$$f(x_k) \le Q(x, \mu_k) \le Q(\bar{x}, \mu_k) = f(\bar{x})$$

两边同时取极限可以得到$f(x^*) \le f(\bar{x})$。

二次罚函数: Convergence Analysis

- 假设$\{x_k\}$有一个极限点x^*，即存在一个无限子列$\mathcal{K}$, 有$\lim_{k\in\mathcal{K}} x_k = x^*$
- 对上式两边取极限$k\in\mathcal{K}$，得到

$$\sum_{i\in\mathcal{E}} c_i^2(x^*) = \lim_{k\in\mathcal{K}} \sum_{i\in\mathcal{E}} c_i^2(x_k) \le \lim_{k\in\mathcal{K}} \frac{2}{\mu_k}[f(\bar{x}) - f(x_k)] = 0.$$

即 $c_i(x^*) = 0$，x^*落在可行域内。

- 再对

$$f(x_k) \le Q(x, \mu_k) \le Q(\bar{x}, \mu_k) = f(\bar{x})$$

两边同时取极限可以得到$f(x^*) \le f(\bar{x})$。

- 由于假设$\bar{x}$是全局极小子，可以得到x^*也是一个全局极小子。

二次罚函数: Convergence Analysis

定理2:(不精确求解$Q(x;\mu_k)$)

假设在Algorithm 1中满足 $\tau_k \to 0$, $\mu_k \uparrow \infty$。$\{x_k\}$的一个极限点x^*.

二次罚函数: Convergence Analysis

定理2:(不精确求解$Q(x;\mu_k)$)

假设在Algorithm 1中满足 $\tau_k \to 0$, $\mu_k \uparrow \infty$。$\{x_k\}$的一个极限点x^*.

1. 如果x^*是不可行点，则x^*是$\|c(x)\|^2$的一个稳定点。

二次罚函数: Convergence Analysis

定理2:(不精确求解$Q(x;\mu_k)$)

假设在Algorithm 1中满足 $\tau_k \to 0$, $\mu_k \uparrow \infty$。$\{x_k\}$的一个极限点x^*.

1. 如果x^*是不可行点，则x^*是$\|c(x)\|^2$的一个稳定点。
2. 如果x^*是可行点，并且$\nabla c_i(x^*)$是线性无关的，则x^*是一个KKT点。 对于这种点x^*，我们有对于任意的子集序列$\mathcal{K}$，$\lim_{k\in\mathcal{K}} x_k = x^*$, 则

$$\lim_{k\in\mathcal{K}} -\mu_k c_i(x_k) = \lambda_i^*, \text{ for all } i \in \mathcal{E},$$

这里λ^*是那些满足KKT条件中的等式约束乘子向量。

二次罚函数: Convergence Analysis

定理2:(不精确求解$Q(x;\mu_k)$)

假设在Algorithm 1中满足 $\tau_k \to 0$, $\mu_k \uparrow \infty$。$\{x_k\}$的一个极限点x^*.

1. 如果x^*是不可行点，则x^*是$\|c(x)\|^2$的一个稳定点。
2. 如果x^*是可行点，并且$\nabla c_i(x^*)$是线性无关的，则x^*是一个KKT点。 对于这种点x^*，我们有对于任意的子集序列$\mathcal{K}$，$\lim_{k\in\mathcal{K}} x_k = x^*$, 则

$$\lim_{k\in\mathcal{K}} -\mu_k c_i(x_k) = \lambda_i^*, \text{ for all } i \in \mathcal{E},$$

这里λ^*是那些满足KKT条件中的等式约束乘子向量。

Proof

二次罚函数: Convergence Analysis

定理2:(不精确求解$Q(x;\mu_k)$)

假设在Algorithm 1中满足 $\tau_k \to 0$, $\mu_k \uparrow \infty$。$\{x_k\}$的一个极限点x^*.

1. 如果x^*是不可行点，则x^*是$\|c(x)\|^2$的一个稳定点。
2. 如果x^*是可行点，并且$\nabla c_i(x^*)$是线性无关的，则x^*是一个KKT点。 对于这种点x^*，我们有对于任意的子集序列$\mathcal{K}$，$\lim_{k\in\mathcal{K}} x_k = x^*$, 则

$$\lim_{k\in\mathcal{K}} -\mu_k c_i(x_k) = \lambda_i^*, \text{ for all } i \in \mathcal{E},$$

这里λ^*是那些满足KKT条件中的等式约束乘子向量。

Proof

- 首先对$Q(x;\mu_k)$ 求导可得：

$$\nabla_x Q(x_k;\mu_k) = \nabla f(x_k) + \sum_{i\in\mathcal{E}} \mu_k c_i(x_k)\nabla c_i(x_k)$$

二次罚函数: Convergence Analysis

二次罚函数: Convergence Analysis

- 对应的终止条件是

$$\|\nabla f(x_k) + \sum_{i\in\mathcal{E}} \mu_k c_i(x_k) \nabla c_i(x_k)\| \leq \tau_k$$

二次罚函数: Convergence Analysis

- 对应的终止条件是

$$\|\nabla f(x_k) + \sum_{i\in\mathcal{E}} \mu_k c_i(x_k)\nabla c_i(x_k)\| \le \tau_k$$

- 利用三角不等式可以得到

$$\|\sum_{i\in\mathcal{E}} c_i(x_k)\nabla c_i(x_k)\| \le \frac{1}{\mu_k}[\tau_k + \|\nabla f(x_k)\|].$$

二次罚函数：Convergence Analysis

- 对应的终止条件是

$$\left\|\nabla f(x_k) + \sum_{i\in\mathcal{E}} \mu_k c_i(x_k)\nabla c_i(x_k)\right\| \le \tau_k$$

- 利用三角不等式可以得到

$$\left\|\sum_{i\in\mathcal{E}} c_i(x_k)\nabla c_i(x_k)\right\| \le \frac{1}{\mu_k}[\tau_k + \|\nabla f(x_k)\|].$$

- 假设x^*是x_k的某一个极限点，即存在一个无限子列$\mathcal{K}$, 使得$\lim_{k\in\mathcal{K}} x_k = x^*$。两边同时取极限可以得到

$$\sum_{i\in\mathcal{E}} c_i(x^*)\nabla c_i(x^*) = 0$$

二次罚函数: Convergence Analysis

- 对应的终止条件是

$$\|\nabla f(x_k) + \sum_{i\in\mathcal{E}} \mu_k c_i(x_k)\nabla c_i(x_k)\| \le \tau_k$$

- 利用三角不等式可以得到

$$\|\sum_{i\in\mathcal{E}} c_i(x_k)\nabla c_i(x_k)\| \le \frac{1}{\mu_k}[\tau_k + \|\nabla f(x_k)\|].$$

- 假设x^*是x_k的某一个极限点，即存在一个无限子列$\mathcal{K}$, 使得$\lim_{k\in\mathcal{K}} x_k = x^*$。两边同时取极限可以得到

$$\sum_{i\in\mathcal{E}} c_i(x^*)\nabla c_i(x^*) = 0$$

- 如果x^*不在可行域，则$c_i(x^*) \neq 0$（即 $\nabla c_i(x^*)$线性相关），则可以推出x^*是$\|c(x)\|_2^2$的一个稳定点。

二次罚函数: Convergence Analysis

二次罚函数: Convergence Analysis

- 如果 $\nabla c_i(x^*)$线性无关，则可以推出$c_i(x^*) = 0$ 对于$\forall i \in \mathcal{E}$. x^*是可行的。

二次罚函数: Convergence Analysis

- 如果 $\nabla c_i(x^*)$线性无关，则可以推出$c_i(x^*) = 0$ 对于$\forall i \in \mathcal{E}$. x^*是可行的。
- 记$A(x)^T = [\nabla c_i(x)]_{i\in\mathcal{E}}$, $\lambda_k = -\mu_k c(x_k)$, 则有

$$A(x_k)^T \lambda_k = \nabla f(x_k) - \nabla_x Q(x_k;\mu_k), \quad \|\nabla_x Q(x_k;\mu_k)\| \le \tau_k.$$

二次罚函数: Convergence Analysis

- 如果 $\nabla c_i(x^*)$线性无关，则可以推出$c_i(x^*) = 0$ 对于$\forall i \in \mathcal{E}$. x^*是可行的。
- 记$A(x)^T = [\nabla c_i(x)]_{i\in\mathcal{E}}$, $\lambda_k = -\mu_k c(x_k)$, 则有

$$A(x_k)^T\lambda_k = \nabla f(x_k) - \nabla_x Q(x_k;\mu_k), \quad \|\nabla_x Q(x_k;\mu_k)\| \le \tau_k.$$

- 当k足够大时，我们可以得到$A(x_k)$是满秩的，这样AA^T是可逆的，可以得到

$$\lambda_k = [A(x_k)A(x_k)^T]^{-1}A(x_k)[\nabla f(x_k) - \nabla_x Q(x_k;\mu_k)]$$

二次罚函数: Convergence Analysis

- 如果 $\nabla c_i(x^*)$线性无关，则可以推出$c_i(x^*) = 0$ 对于$\forall i \in \mathcal{E}$. x^*是可行的。
- 记$A(x)^T = [\nabla c_i(x)]_{i\in\mathcal{E}}$, $\lambda_k = -\mu_k c(x_k)$, 则有

$$A(x_k)^T \lambda_k = \nabla f(x_k) - \nabla_x Q(x_k;\mu_k), \quad \|\nabla_x Q(x_k;\mu_k)\| \leq \tau_k.$$

- 当k足够大时，我们可以得到$A(x_k)$是满秩的，这样AA^T是可逆的，可以得到

$$\lambda_k = [A(x_k)A(x_k)^T]^{-1} A(x_k)[\nabla f(x_k) - \nabla_x Q(x_k;\mu_k)]$$

- 两边同时取极限可以得到

$$\lim_{k\in\mathcal{K}} \lambda_k = \lambda^* = [A(x^*)A(x^*)^T]^{-1} A(x^*)\nabla f(x^*)$$

二次罚函数: Convergence Analysis

- 如果 $\nabla c_i(x^*)$线性无关，则可以推出$c_i(x^*) = 0$ 对于$\forall i \in \mathcal{E}$. x^*是可行的。
- 记$A(x)^T = [\nabla c_i(x)]_{i\in\mathcal{E}}$, $\lambda_k = -\mu_k c(x_k)$, 则有

$$A(x_k)^T \lambda_k = \nabla f(x_k) - \nabla_x Q(x_k;\mu_k), \quad \|\nabla_x Q(x_k;\mu_k)\| \leq \tau_k.$$

- 当k足够大时，我们可以得到$A(x_k)$是满秩的，这样AA^T是可逆的，可以得到

$$\lambda_k = [A(x_k)A(x_k)^T]^{-1} A(x_k)[\nabla f(x_k) - \nabla_x Q(x_k;\mu_k)]$$

- 两边同时取极限可以得到

$$\lim_{k\in\mathcal{K}} \lambda_k = \lambda^* = [A(x^*)A(x^*)^T]^{-1} A(x^*)\nabla f(x^*)$$

- 对终止条件$\|\nabla f(x_k) + \sum_{i\in\mathcal{E}} \mu_k c_i(x_k)\nabla c_i(x_k)\| \leq \tau_k$两边也同时取极限

$$\nabla f(x^*) - A(x^*)^T \lambda^* = 0.$$

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx} Q(x, \mu_k)$的条件数。

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx}Q(x,\mu_k)$的条件数。

- 首先$Q(x;\mu_k)$的Hessian阵为:

$$\nabla^2_{xx}Q(x;\mu_k) = \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) + \mu_k A(x)^T A(x)$$

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx}Q(x,\mu_k)$的条件数。

- 首先$Q(x;\mu_k)$的Hessian阵为:

$$\nabla^2_{xx}Q(x;\mu_k) = \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) + \mu_k A(x)^T A(x)$$

- 当$x \to x^*$时，前两项就是拉格朗日函数的Hessian

$$\nabla^2_{xx}Q(x;\mu_k) \approx \nabla^2_{xx}\mathcal{L}(x,\lambda^*) + \mu_k A(x)^T A(x).$$

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx}Q(x,\mu_k)$的条件数。

- 首先$Q(x;\mu_k)$的Hessian阵为:

$$\nabla^2_{xx}Q(x;\mu_k)=\nabla^2 f(x)+\sum_{i\in\mathcal{E}}\mu_k c_i(x)\nabla^2 c_i(x)+\mu_k A(x)^T A(x)$$

- 当$x\to x^*$时，前两项就是拉格朗日函数的Hessian

$$\nabla^2_{xx}Q(x;\mu_k)\approx\nabla^2_{xx}\mathcal{L}(x,\lambda^*)+\mu_k A(x)^T A(x).$$

- 通常来说等式约束的个数会小于变量个数，$|\mathcal{E}|<n$，因此第二项是奇异。

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx}Q(x,\mu_k)$的条件数。

- 首先$Q(x;\mu_k)$的Hessian阵为:

$$\nabla^2_{xx}Q(x;\mu_k) = \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) + \mu_k A(x)^T A(x)$$

- 当$x \to x^*$时，前两项就是拉格朗日函数的Hessian

$$\nabla^2_{xx}Q(x;\mu_k) \approx \nabla^2_{xx}\mathcal{L}(x,\lambda^*) + \mu_k A(x)^T A(x).$$

- 通常来说等式约束的个数会小于变量个数，$|\mathcal{E}| < n$，因此第二项是奇异。
- 因此总矩阵的有些特征值会趋于一个常数，另外有一些是$\mathcal{O}(\mu_k)$。当μ_k很大时，总矩阵的条件数就会很大。

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx}Q(x,\mu_k)$的条件数。

- 首先$Q(x;\mu_k)$的Hessian阵为:

$$\nabla^2_{xx}Q(x;\mu_k)=\nabla^2 f(x)+\sum_{i\in\mathcal{E}}\mu_k c_i(x)\nabla^2 c_i(x)+\mu_k A(x)^T A(x)$$

- 当$x\to x^*$时，前两项就是拉格朗日函数的Hessian

$$\nabla^2_{xx}Q(x;\mu_k)\approx\nabla^2_{xx}\mathcal{L}(x,\lambda^*)+\mu_k A(x)^T A(x).$$

- 通常来说等式约束的个数会小于变量个数，$|\mathcal{E}|<n$，因此第二项是奇异。
- 因此总矩阵的有些特征值会趋于一个常数，另外有一些是$\mathcal{O}(\mu_k)$。当μ_k很大时，总矩阵的条件数就会很大。
- 条件数很大的一个后果是，无法求解很好的牛顿步，即求解

$$p^N=-(\nabla^2_{xx}Q(x;\mu_k))^{-1}\nabla_x Q(x;\mu_k) \tag{15.1}$$

二次罚函数: Ill-Conditioning

下面我们检查一下$\nabla^2_{xx}Q(x,\mu_k)$的条件数。

- 首先$Q(x;\mu_k)$的Hessian阵为:

$$\nabla^2_{xx}Q(x;\mu_k)=\nabla^2 f(x)+\sum_{i\in\mathcal{E}}\mu_k c_i(x)\nabla^2 c_i(x)+\mu_k A(x)^T A(x)$$

- 当$x\to x^*$时，前两项就是拉格朗日函数的Hessian

$$\nabla^2_{xx}Q(x;\mu_k)\approx\nabla^2_{xx}\mathcal{L}(x,\lambda^*)+\mu_k A(x)^T A(x).$$

- 通常来说等式约束的个数会小于变量个数，$|\mathcal{E}|<n$，因此第二项是奇异。
- 因此总矩阵的有些特征值会趋于一个常数，另外有一些是$\mathcal{O}(\mu_k)$。当μ_k很大时，总矩阵的条件数就会很大。
- 条件数很大的一个后果是，无法求解很好的牛顿步，即求解

$$p^N=-(\nabla^2_{xx}Q(x;\mu_k))^{-1}\nabla_x Q(x;\mu_k) \tag{15.1}$$

- 一般来说很难求解，除非有好的预条件子。

二次罚函数: Transformation

通过引入变量$\zeta = \mu A(x)p$

二次罚函数: Transformation

通过引入变量$\zeta = \mu A(x)p$

- 则(p, ζ)满足

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) & A(x)^T \\ A(x) & -(\mu_k)^{-1} I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x, \mu_k) \\ 0 \end{bmatrix} \tag{15.2}$$

二次罚函数: Transformation

通过引入变量$\zeta = \mu A(x)p$

- 则(p, ζ)满足

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) & A(x)^T \\ A(x) & -(\mu_k)^{-1} I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x, \mu_k) \\ 0 \end{bmatrix} \tag{15.2}$$

- 当x靠近x^*时，上述系数矩阵没有大特征值$\mathcal{O}(\mu_k)$，因此是(15.1)的well-conditioned变换.

二次罚函数: TRANSFORMATION

通过引入变量$\zeta=\mu A(x)p$

- 则(p,ζ)满足

$$\begin{bmatrix} \nabla^2 f(x)+\sum_{i\in\mathcal{E}}\mu_k c_i(x)\nabla^2 c_i(x) & A(x)^T \\ A(x) & -(\mu_k)^{-1}I \end{bmatrix}\begin{bmatrix} p \\ \zeta \end{bmatrix}=\begin{bmatrix} -\nabla_x Q(x,\mu_k) \\ 0 \end{bmatrix} \tag{15.2}$$

- 当x靠近x^*时，上述系数矩阵没有大特征值$\mathcal{O}(\mu_k)$，因此是(15.1)的well-conditioned变换.
- 但是两个系统都不能很好地求解牛顿方向p，因为在(15.2)中$\mu_k c_i(x)$并不能很好地逼近$-\lambda_i^*$，即使x_k已经逼近$Q(x;\mu_k)$的极小子。

二次罚函数: Transformation

通过引入变量$\zeta = \mu A(x)p$

- 则(p, ζ)满足

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x) \nabla^2 c_i(x) & A(x)^T \\ A(x) & -(\mu_k)^{-1} I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x, \mu_k) \\ 0 \end{bmatrix} \tag{15.2}$$

- 当x靠近x^*时，上述系数矩阵没有大特征值$\mathcal{O}(\mu_k)$，因此是(15.1)的well-conditioned变换.
- 但是两个系统都不能很好地求解牛顿方向p，因为在(15.2)中$\mu_k c_i(x)$并不能很好地逼近$-\lambda_i^*$，即使x_k已经逼近$Q(x; \mu_k)$的极小子。
- 这个事实有可能是由于基于p的二次函数模型并不能很好地逼近$Q(\cdot; \mu_k)$，因此牛顿步可能本质上就不是一个很好的搜索方向。

二次罚函数：Transformation

通过引入变量$\zeta = \mu A(x) p$

- 则(p, ζ)满足

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x) \nabla^2 c_i(x) & A(x)^T \\ A(x) & -(\mu_k)^{-1} I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x, \mu_k) \\ 0 \end{bmatrix} \tag{15.2}$$

- 当x靠近x^*时，上述系数矩阵没有大特征值$\mathcal{O}(\mu_k)$，因此是(15.1)的well-conditioned变换.
- 但是两个系统都不能很好地求解牛顿方向p，因为在(15.2)中$\mu_k c_i(x)$并不能很好地逼近$-\lambda_i^*$，即使x_k已经逼近$Q(x;\mu_k)$的极小子。
- 这个事实有可能是由于基于p的二次函数模型并不能很好地逼近$Q(\cdot;\mu_k)$，因此牛顿步可能本质上就不是一个很好的搜索方向。
- 如果要求解(15.2)，即求解一个$n + |\mathcal{E}|$的线性方程组。这里μ_k^{-1}可以看做是正则化参数，用于帮助解决A是秩亏时带来的迭代矩阵有可能是 奇异的困难。

二次罚函数: TRANSFORMATION

通过引入变量$\zeta = \mu A(x)p$

- 则(p,ζ)满足

$$\begin{bmatrix} \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \mu_k c_i(x)\nabla^2 c_i(x) & A(x)^T \\ A(x) & -(\mu_k)^{-1} I \end{bmatrix} \begin{bmatrix} p \\ \zeta \end{bmatrix} = \begin{bmatrix} -\nabla_x Q(x,\mu_k) \\ 0 \end{bmatrix} \tag{15.2}$$

- 当x靠近x^*时，上述系数矩阵没有大特征值$\mathcal{O}(\mu_k)$，因此是(15.1)的well-conditioned变换.
- 但是两个系统都不能很好地求解牛顿方向p，因为在(15.2)中$\mu_k c_i(x)$并不能很好地逼近$-\lambda_i^*$，即使x_k已经逼近$Q(x;\mu_k)$的极小子。
- 这个事实有可能是由于基于p的二次函数模型并不能很好地逼近$Q(\cdot;\mu_k)$，因此牛顿步可能本质上就不是一个很好的搜索方向。
- 如果要求解(15.2)，即求解一个$n+|\mathcal{E}|$的线性方程组。这里μ_k^{-1}可以看做是正则化参数，用于帮助解决A是秩亏时带来的迭代矩阵有可能是 奇异的困难。
- 如果μ_k不能很快趋于无穷大，则无法到达超线性收敛。

非光滑罚函数

- 如果选取合适的惩罚参数，使得一次最优化就可以得到原问题的精确解。

非光滑罚函数

- 如果选取合适的惩罚参数，使得一次最优化就可以得到原问题的精确解。
- 二次罚函数不是精确的，对于任意的$\mu > 0$，其解都不是原问题的解。

非光滑罚函数

- 如果选取合适的惩罚参数，使得一次最优化就可以得到原问题的精确解。
- 二次罚函数不是精确的，对于任意的$\mu > 0$，其解都不是原问题的解。
- 考虑ℓ_1 罚函数

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^-. \tag{15.3}$$

其中$[y]^- = \max(0, -y)$.

定理3：ℓ_1是精确罚函数

假设x^*是原问题的一个精确解满足KKT条件。则x^*也是$\phi_1(x;\mu)$在$\mu > \mu^*$的一个局部解，并且

$$\mu^* = \|\lambda^*\|_\infty = \max_{i\in\mathcal{E}\cup\mathcal{I}} |\lambda_i^*|. \tag{15.4}$$

此外，如果对$\mu > \mu^*$满足二阶充分性条件，则x^*还是严格的局部极小子。

非光滑罚函数

例：

$$\min x \quad \text{subject to } x \geq 1$$

非光滑罚函数

例：

$$\min x \quad \text{subject to } x \geq 1$$

则对应的非光滑罚函数为

$$\phi_1(x;\mu) = x + \mu[x-1]^- = \begin{cases} (1-\mu)x + \mu & \text{if } x \leq 1, \\ x & \text{if } x > 1. \end{cases}$$

非光滑罚函数

例：

$$\min x \quad \text{subject to } x \geq 1$$

则对应的非光滑罚函数为

$$\phi_1(x;\mu) = x + \mu[x-1]^- = \begin{cases} (1-\mu)x + \mu & \text{if } x \leq 1, \\ x & \text{if } x > 1. \end{cases}$$

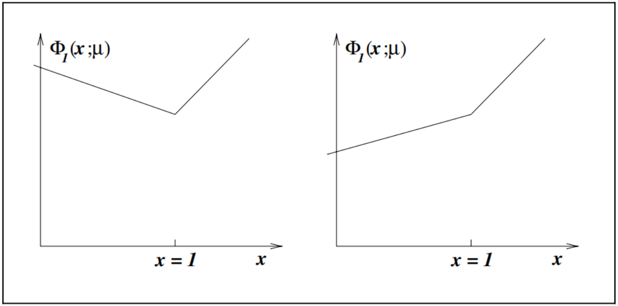


Figure: $\mu > 1$(left), $\mu < 1$(right).

非光滑罚函数

定义：平衡点

非光滑罚函数

定义：平衡点

- $\hat{x} \in R^n$被称为平衡点，如果对任意的$p \in R^n$

$$D(\phi_1(\hat{x};\mu);p) \geq 0.$$

非光滑罚函数

定义：平衡点

- $\hat{x} \in R^n$被称为平衡点，如果对任意的$p \in R^n$

$$D(\phi_1(\hat{x};\mu);p) \geq 0.$$

- 同样地，$\hat{x}$也如下"不可行测度"的一个平衡点

$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} [c_i(x)]^-$$

如果$D(h(\hat{x});p) \geq 0$, 对任意的 $p \in R^n$。

非光滑罚函数

定义：平衡点

- $\hat{x} \in R^n$被称为平衡点，如果对任意的$p \in R^n$

$$D(\phi_1(\hat{x};\mu);p) \geq 0.$$

- 同样地，$\hat{x}$也如下"不可行测度"的一个平衡点

$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} [c_i(x)]^-$$

  如果$D(h(\hat{x});p) \geq 0$, 对任意的 $p \in R^n$。

- 如果$\hat{x}$是不可行的，但仍旧是$h(x)$的平衡点，称之为不可行平衡点。

非光滑罚函数

定义：平衡点

- $\hat{x} \in R^n$被称为平衡点，如果对任意的$p \in R^n$

$$D(\phi_1(\hat{x};\mu);p) \geq 0.$$

- 同样地，$\hat{x}$也如下"不可行测度"的一个平衡点

$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} [c_i(x)]^-$$

如果$D(h(\hat{x});p) \geq 0$, 对任意的 $p \in R^n$。

- 如果$\hat{x}$是不可行的，但仍旧是$h(x)$的平衡点，称之为不可行平衡点。
- 上述例子里

$$D(\phi_1(x^*;\mu);p) = \begin{cases} p & \text{if } p \geq 0, \\ (1-\mu)p & \text{if } p < 0; \end{cases}$$

显然，当$\mu > 1$时，我们有$D(\phi_1(x^*;\mu);p) \geq 0$。

非光滑罚函数

定理4：

假设$\hat{x}$是罚函数$\phi_1(x;\mu)$的稳定点对于$\forall\ \mu > \hat{\mu}$. 那么以下结论成立。

1. 如果$\hat{x}$是可行的，则$\hat{x}$满足KKT条件。
2. 如果$\hat{x}$不是可行点，则它是不可行的稳定点。

非光滑罚函数

- 假设$\hat{x}$是可行的，则根据方向导数的定义有

$$D(\phi_1(\hat{x},\mu);p) = \nabla f(\hat{x})^T p + \mu \sum_{i\in\mathcal{E}} |\nabla c_i(\hat{x})^T p| + \mu \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^-$$

非光滑罚函数

- 假设$\hat{x}$是可行的，则根据方向导数的定义有

$$D(\phi_1(\hat{x},\mu);p) = \nabla f(\hat{x})^T p + \mu \sum_{i\in\mathcal{E}} |\nabla c_i(\hat{x})^T p| + \mu \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^-$$

- 在线性化后的可行方向集$\mathcal{F}(\hat{x})$中考虑任何方向p，我们可以得到

$$|\nabla c_i(\hat{x})^T p| + \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^- = 0$$

非光滑罚函数

- 假设$\hat{x}$是可行的，则根据方向导数的定义有

$$D(\phi_1(\hat{x},\mu);p) = \nabla f(\hat{x})^T p + \mu \sum_{i\in\mathcal{E}} |\nabla c_i(\hat{x})^T p| + \mu \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^-$$

- 在线性化后的可行方向集$\mathcal{F}(\hat{x})$中考虑任何方向p，我们可以得到

$$|\nabla c_i(\hat{x})^T p| + \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^- = 0$$

- 再根据平衡点的定义可知

$$0 \le D(\phi_1(\hat{x};\mu);p) = \nabla f(\hat{x})^T p, \quad \forall p \in \mathcal{F}(\hat{x}).$$

非光滑罚函数

- 假设$\hat{x}$是可行的，则根据方向导数的定义有

$$D(\phi_1(\hat{x},\mu);p) = \nabla f(\hat{x})^T p + \mu \sum_{i\in\mathcal{E}} |\nabla c_i(\hat{x})^T p| + \mu \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^-$$

- 在线性化后的可行方向集$\mathcal{F}(\hat{x})$中考虑任何方向p，我们可以得到

$$|\nabla c_i(\hat{x})^T p| + \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^- = 0$$

- 再根据平衡点的定义可知

$$0 \le D(\phi_1(\hat{x};\mu);p) = \nabla f(\hat{x})^T p, \quad \forall p \in \mathcal{F}(\hat{x}).$$

- 最后使用Farkas引理可得

$$\nabla f(\hat{x}) = \sum_{i\in\mathcal{A}(\hat{x})} \hat{\lambda}_i \nabla c_i(\hat{x}), \text{ and } \hat{\lambda}_i \ge 0 \text{ for all } i \in \mathcal{I}\cap\mathcal{A}(\hat{x}).$$

非光滑罚函数

- 假设$\hat{x}$是可行的，则根据方向导数的定义有

$$D(\phi_1(\hat{x},\mu);p) = \nabla f(\hat{x})^T p + \mu \sum_{i\in\mathcal{E}} |\nabla c_i(\hat{x})^T p| + \mu \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^-$$

- 在线性化后的可行方向集$\mathcal{F}(\hat{x})$中考虑任何方向p，我们可以得到

$$|\nabla c_i(\hat{x})^T p| + \sum_{i\in\mathcal{I}\cap\mathcal{A}(\hat{x})} [\nabla c_i(\hat{x})^T p]^- = 0$$

- 再根据平衡点的定义可知

$$0 \le D(\phi_1(\hat{x};\mu);p) = \nabla f(\hat{x})^T p, \quad \forall p \in \mathcal{F}(\hat{x}).$$

- 最后使用Farkas引理可得

$$\nabla f(\hat{x}) = \sum_{i\in\mathcal{A}(\hat{x})} \hat{\lambda}_i \nabla c_i(\hat{x}), \text{ and } \hat{\lambda}_i \ge 0 \text{ for all } i \in \mathcal{I}\cap\mathcal{A}(\hat{x}).$$

- 再由一阶最优性条件定理(即KKT条件)，上述等式可以推出KKT条件满足。

非光滑函数

重新考虑第一个例子，其对应的ℓ_1罚函数应该是

$$\phi_1(x;\mu) = x_1 + x_2 + \mu|x_1^2 + x_2^2 - 2|.$$

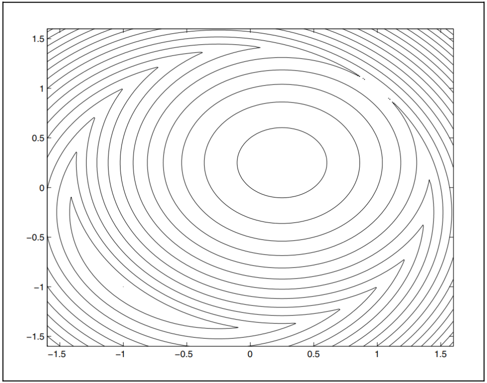


Figure: contours of $\phi_1(x;2)$

非光滑函数

重新考虑第一个例子，其对应的ℓ_1罚函数应该是

$$\phi_1(x;\mu) = x_1 + x_2 + \mu|x_1^2 + x_2^2 - 2|.$$

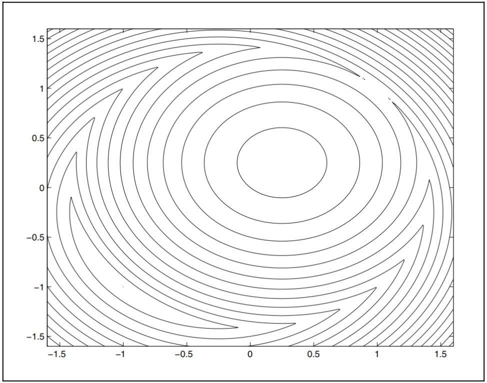


Figure: contours of $\phi_1(x;2)$

- 根据定理3，当$\mu > |\lambda^*| = 0.5$时，$\phi_1(x;\mu)$的极小值与$(-1,-1)^T$重合。
- 等高线上的尖点表示，沿着边界$x_1^2 + x_2^2 = 2$，$\phi(x;\mu)$是不光滑的。

非光滑函数算法

Algorithm 2: 经典ℓ_1罚函数法

给定$\mu_0 > 0$，容忍误差$\tau > 0$，起始点x_k^s;
for $k = 0, 1, 2, \cdots$
  从初始值x_k^s出发，寻找$\phi_1(x; \mu_k)$的极小子x_k;
  if $h(x_k) \leq \tau$
    stop with x_k
  end(if)
  选择新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择新的初始点x_{k+1}^s;
end(for)

非光滑函数算法

Algorithm 2: 经典ℓ_1罚函数法

给定$\mu_0 > 0$，容忍误差$\tau > 0$，起始点x_k^s;
for $k = 0, 1, 2, \cdots$
  从初始值x_k^s出发，寻找$\phi_1(x; \mu_k)$的极小子x_k;
  if $h(x_k) \leq \tau$
    stop with x_k
  end(if)
  选择新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择新的初始点x_{k+1}^s;
end(for)

- 由于$\phi_1(x; \mu)$是不光滑的，求解有一定困难。可以用SQP(sequential quadratic programming)求解。

非光滑函数算法

Algorithm 2: 经典ℓ_1罚函数法

给定$\mu_0 > 0$，容忍误差$\tau > 0$，起始点x_k^s;
for $k = 0, 1, 2, \cdots$
  从初始值x_k^s出发，寻找$\phi_1(x; \mu_k)$的极小子x_k;
  if $h(x_k) \leq \tau$
    stop with x_k
  end(if)
  选择新的惩罚参数$\mu_{k+1} > \mu_k$;
  选择新的初始点x_{k+1}^s;
end(for)

- 由于$\phi_1(x; \mu)$是不光滑的，求解有一定困难。可以用SQP(sequential quadratic programming)求解。
- 如何更新μ_k? 可以用简单的策略，倍数增加。当μ_k比较大时，更难求解，需要更多计算量。

非光滑函数实用算法

- 定义

$$q(p;\mu)=f(x)+\nabla f(x)^Tp+\frac{1}{2}p^TWp+\mu\sum_{i\in\mathcal{E}}|c_i(x)+\nabla c_i(x)^Tp|$$
$$+\mu\sum_{i\in\mathcal{I}}[c_i(x)+\nabla c_i(x)^Tp]^-$$

其中W是对称矩阵，包含了f以及$c_i(x)$的二阶导数信息。

非光滑函数实用算法

- 定义

$$q(p;\mu) = f(x) + \nabla f(x)^T p + \frac{1}{2}p^T W p + \mu \sum_{i\in\mathcal{E}} |c_i(x) + \nabla c_i(x)^T p|$$
$$+ \mu \sum_{i\in\mathcal{I}} [c_i(x) + \nabla c_i(x)^T p]^-$$

其中W是对称矩阵，包含了f以及$c_i(x)$的二阶导数信息。

- 显然$q(p;\mu)$也不是光滑的，但是我们将其转化为光滑的二次规划问题

$$\left\{\begin{array}{rl} \min\limits_{p,r,s,t} & f(x) + \frac{1}{2}p^T W p + \mu \sum_{i\in\mathcal{E}}(r_i + s_i) + \mu \sum_{i\in\mathcal{I}} t_i \\ \text{subject to} & \nabla c_i(x)^T p + c_i(x) = r_i - s_i, \quad i \in \mathcal{E} \\ & \nabla c_i(x)^T p + c_i(x) \geq -t_i, \quad i \in \mathcal{I} \\ & r, s, t \geq 0. \end{array}\right.$$

非光滑函数实用算法

- 定义

$$q(p;\mu) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T W p + \mu \sum_{i\in\mathcal{E}} |c_i(x) + \nabla c_i(x)^T p| + \mu \sum_{i\in\mathcal{I}} [c_i(x) + \nabla c_i(x)^T p]^-$$

其中W是对称矩阵，包含了f以及$c_i(x)$的二阶导数信息。

- 显然$q(p;\mu)$也不是光滑的，但是我们将其转化为光滑的二次规划问题

$$\begin{cases} \min\limits_{p,r,s,t} & f(x) + \frac{1}{2} p^T W p + \mu \sum_{i\in\mathcal{E}} (r_i + s_i) + \mu \sum_{i\in\mathcal{I}} t_i \\ \text{subject to} & \nabla c_i(x)^T p + c_i(x) = r_i - s_i, \quad i \in \mathcal{E} \\ & \nabla c_i(x)^T p + c_i(x) \geq -t_i, \quad i \in \mathcal{I} \\ & r, s, t \geq 0. \end{cases}$$

- 上述子问题可以由标准的二次罚函数方法求解。

拉格朗日增广函数

- 该方法可以看作是拉格朗日乘子函数和二次罚函数的组合

$$\mathcal{L}_A(x,\lambda;\mu) = f(x) - \sum_{i\in\mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i^2(x)$$

拉格朗日增广函数

- 该方法可以看作是拉格朗日乘子函数和二次罚函数的组合

$$\mathcal{L}_A(x,\lambda;\mu)=f(x)-\sum_{i\in\mathcal{E}}\lambda_i c_i(x)+\frac{\mu}{2}\sum_{i\in\mathcal{E}}c_i^2(x)$$

- 首先固定μ_k，然后固定当前估计λ^k，寻找最优解x_k。其满足的最优性条件是

$$0\approx\nabla_x\mathcal{L}_A(x_k,\lambda^k;\mu_k)=\nabla f(x_k)-\sum_{i\in\mathcal{E}}[\lambda_i^k-\mu_k c_i(x_k)]\nabla c_i(x_k).$$

拉格朗日增广函数

- 该方法可以看作是拉格朗日乘子函数和二次罚函数的组合

$$\mathcal{L}_A(x,\lambda;\mu) = f(x) - \sum_{i\in\mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i^2(x)$$

- 首先固定μ_k，然后固定当前估计λ^k，寻找最优解x_k。其满足的最优性条件是

$$0 \approx \nabla_x \mathcal{L}_A(x_k,\lambda^k;\mu_k) = \nabla f(x_k) - \sum_{i\in\mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)]\nabla c_i(x_k).$$

- 根据二次罚函数的定理2，我们可以得到

$$\lambda_i^* \approx \lambda_i^k - \mu_k c_i(x_k), \quad \forall i \in \mathcal{E}.$$

即： $$c_i(x_k) \approx -\frac{1}{\mu_k}(\lambda_i^* - \lambda_i^k), \quad \forall i \in \mathcal{E}.$$

拉格朗日增广函数

- 该方法可以看作是拉格朗日乘子函数和二次罚函数的组合

$$\mathcal{L}_A(x,\lambda;\mu) = f(x) - \sum_{i\in\mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2}\sum_{i\in\mathcal{E}} c_i^2(x)$$

- 首先固定μ_k，然后固定当前估计λ^k，寻找最优解x_k。其满足的最优性条件是

$$0 \approx \nabla_x \mathcal{L}_A(x_k,\lambda^k;\mu_k) = \nabla f(x_k) - \sum_{i\in\mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)]\nabla c_i(x_k).$$

- 根据二次罚函数的定理2，我们可以得到

$$\lambda_i^* \approx \lambda_i^k - \mu_k c_i(x_k), \quad \forall i \in \mathcal{E}.$$

$$\text{即：} \quad c_i(x_k) \approx -\frac{1}{\mu_k}(\lambda_i^* - \lambda_i^k), \quad \forall i \in \mathcal{E}.$$

- 由于λ^*很接近λ^k，这说明x_k的不可行性远小于$\frac{1}{\mu_k}$，因此我们可以设置

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k), \quad \forall i \in \mathcal{E}.$$

拉格朗日增广函数算法

Algorithm 3: 等式约束下的增广拉格朗日方法

给定$\mu_0 > 0$, 容忍误差τ_0, 初始出发点x_0^s和λ^0;
for $k = 0, 1, 2, \cdots$
  从x_k^s出发找到$\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$的近似极小子$x_k$ 直到$\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \le \tau_k$
  if a convergence test is satisfied
    stop with x_k
  end(if)
  更新拉格朗日乘子$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$;
  选择新的惩罚参数$\mu_{k+1} \ge \mu_k$;
  选择新的迭代出发点$x_{k+1}^s = x_k$;
  选择新的容忍误差τ_{k+1};
end(for)

拉格朗日增广函数算法

Algorithm 3: 等式约束下的增广拉格朗日方法

给定$\mu_0 > 0$, 容忍误差τ_0, 初始出发点x_0^s和λ^0;
for $k = 0, 1, 2, \cdots$
    从x_k^s出发找到$\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$的近似极小子$x_k$ 直到$\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \le \tau_k$
    if a convergence test is satisfied
        stop with x_k
    end(if)
    更新拉格朗日乘子$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$;
    选择新的惩罚参数$\mu_{k+1} \ge \mu_k$;
    选择新的迭代出发点$x_{k+1}^s = x_k$;
    选择新的容忍误差τ_{k+1};
end(for)

- 这里不需要$\mu_k \to \infty$来保证收敛性

拉格朗日增广函数算法

Algorithm 3: 等式约束下的增广拉格朗日方法

给定$\mu_0 > 0$, 容忍误差τ_0, 初始出发点x_0^s和λ^0;
for $k = 0, 1, 2, \cdots$
  从x_k^s出发找到$\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$的近似极小子$x_k$ 直到$\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \leq \tau_k$
  if a convergence test is satisfied
    stop with x_k
  end(if)
  更新拉格朗日乘子$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$;
  选择新的惩罚参数$\mu_{k+1} \geq \mu_k$;
  选择新的迭代出发点$x_{k+1}^s = x_k$;
  选择新的容忍误差τ_{k+1};
end(for)

- 这里不需要$\mu_k \to \infty$来保证收敛性
- ill-condition也不是一个问题, x_{k+1}^s的选取也不是关键的.

拉格朗日增广函数算法

Algorithm 3: 等式约束下的增广拉格朗日方法

给定$\mu_0 > 0$, 容忍误差τ_0, 初始出发点x_0^s和λ^0;
for $k = 0, 1, 2, \cdots$
  从x_k^s出发找到$\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$的近似极小子$x_k$ 直到$\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \le \tau_k$
  if a convergence test is satisfied
    stop with x_k
  end(if)
  更新拉格朗日乘子$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$;
  选择新的惩罚参数$\mu_{k+1} \ge \mu_k$;
  选择新的迭代出发点$x_{k+1}^s = x_k$;
  选择新的容忍误差τ_{k+1};
end(for)

- 这里不需要$\mu_k \to \infty$来保证收敛性
- ill-condition也不是一个问题, x_{k+1}^s的选取也不是关键的.
- τ_k 可以根据$\sum_{i \in \mathcal{E}} |c(x_k)|$选取

拉格朗日增广函数算法

ALGORITHM 3: 等式约束下的增广拉格朗日方法

给定$\mu_0 > 0$, 容忍误差τ_0, 初始出发点x_0^s和λ^0;
for $k = 0, 1, 2, \cdots$
  从x_k^s出发找到$\mathcal{L}_A(\cdot, \lambda^k; \mu_k)$的近似极小子$x_k$ 直到$\|\nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)\| \leq \tau_k$
  if a convergence test is satisfied
    stop with x_k
  end(if)
  更新拉格朗日乘子$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k)$;
  选择新的惩罚参数$\mu_{k+1} \geq \mu_k$;
  选择新的迭代出发点$x_{k+1}^s = x_k$;
  选择新的容忍误差τ_{k+1};
end(for)

- 这里不需要$\mu_k \to \infty$来保证收敛性
- ill-condition也不是一个问题, x_{k+1}^s的选取也不是关键的.
- τ_k 可以根据$\sum_{i\in\mathcal{E}} |c(x_k)|$选取
- 如果不可行测度下降不充分，可以增加μ.

拉格朗日增广函数

仍然考虑前面的例子，其对应的拉格朗日增广函数是

$$\mathcal{L}_A(x, \lambda; \mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2).$$

拉格朗日增广函数

仍然考虑前面的例子，其对应的拉格朗日增广函数是

$$\mathcal{L}_A(x,\lambda;\mu)=x_1+x_2-\lambda(x_1^2+x_2^2-2)+\frac{\mu}{2}(x_1^2+x_2^2-2).$$

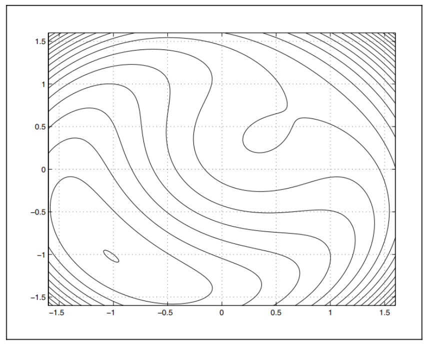


Figure: $\lambda=-0.4$ and $\mu=1$

拉格朗日增广函数

仍然考虑前面的例子，其对应的拉格朗日增广函数是

$$\mathcal{L}_A(x,\lambda;\mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2).$$

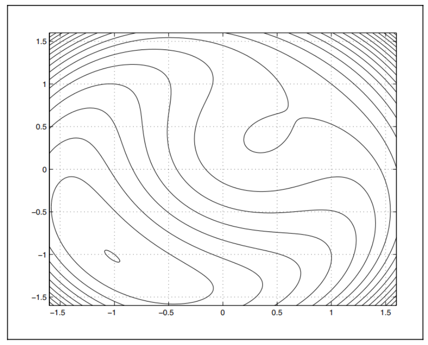


Figure: $\lambda = -0.4$ and $\mu = 1$

拉格朗日增广函数

仍然考虑前面的例子，其对应的拉格朗日增广函数是

$$\mathcal{L}_A(x, \lambda; \mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2).$$

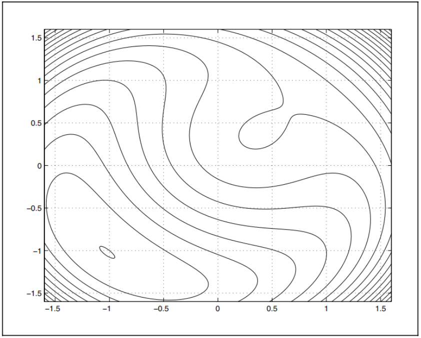


Figure: $\lambda = -0.4$ and $\mu = 1$

- 精确解是$x^* = (-1, -1)^T$, $\lambda^* = -0.5$.

拉格朗日增广函数

仍然考虑前面的例子，其对应的拉格朗日增广函数是

$$\mathcal{L}_A(x,\lambda;\mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2).$$

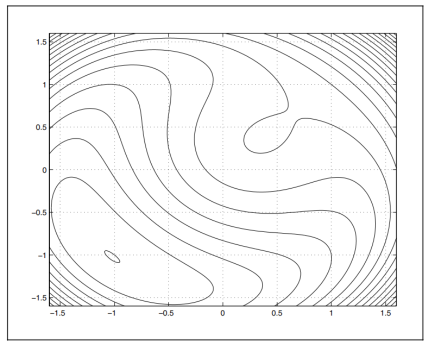


Figure: $\lambda = -0.4$ and $\mu = 1$

- 精确解是$x^* = (-1,-1)^T$, $\lambda^* = -0.5$.
- 从等高线密度来看与使用二次罚函数相似（$\mu = 1$）,说明条件数还比较好。

拉格朗日增广函数

仍然考虑前面的例子，其对应的拉格朗日增广函数是

$$\mathcal{L}_A(x,\lambda;\mu) = x_1 + x_2 - \lambda(x_1^2 + x_2^2 - 2) + \frac{\mu}{2}(x_1^2 + x_2^2 - 2).$$

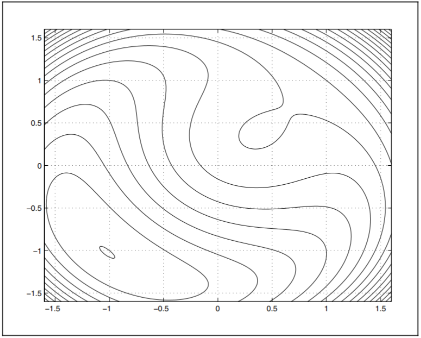


Figure: $\lambda = -0.4$ and $\mu = 1$

- 精确解是$x^* = (-1,-1)^T$, $\lambda^* = -0.5$.
- 从等高线密度来看与使用二次罚函数相似（$\mu = 1$）,说明条件数还比较好。
- 但是解$(-1.02,-1.02)^T$要远好于使用二次罚函数情形。

拉格朗日增广函数性质

定理5：

如果x^*是原问题的解，并且在x^*处LICQ成立，二阶充分性条件在$\lambda = \lambda^*$处也成立。那么，存在一个阈值$\bar{\mu}$，当$\mu \geq \bar{\mu}$时，x^*是$\mathcal{L}_A(x, \lambda^*; \mu)$的一个严格局部极小子。

拉格朗日增广函数性质

定理5：

如果x^*是原问题的解，并且在x^*处LICQ成立，二阶充分性条件在$\lambda = \lambda^*$处也成立。那么，存在一个阈值$\bar{\mu}$，当$\mu \geq \bar{\mu}$时，x^*是$\mathcal{L}_A(x, \lambda^*; \mu)$的一个严格局部极小子。

定理6：

如果定理5中条件在x^*和λ^*处成立，$\bar{\mu}$也如定理5中取值。那么存在正的δ, ϵ和M满足 对于所有的λ^k和μ_k满足

$$\|\lambda^k - \lambda^*\| \leq \delta\mu_k,\ \mu_k \geq \bar{\mu}$$

拉格朗日增广函数性质

定理5：

如果x^*是原问题的解，并且在x^*处LICQ成立，二阶充分性条件在$\lambda=\lambda^*$处也成立。那么，存在一个阈值$\bar{\mu}$，当$\mu \geq \bar{\mu}$时，x^*是$\mathcal{L}_A(x,\lambda^*;\mu)$的一个严格局部极小子。

定理6：

如果定理5中条件在x^*和λ^*处成立，$\bar{\mu}$也如定理5中取值。那么存在正的δ, ϵ和M满足 对于所有的λ^k和μ_k满足

$$\|\lambda^k - \lambda^*\| \leq \delta\mu_k,\ \mu_k \geq \bar{\mu}$$

1. 问题 $\min_x \mathcal{L}_A(x,\lambda^k;\mu_k)$ subject to $\|x - x^*\| \leq \epsilon$ 存在唯一的解x_k，并且满足 $\|x_k - x^*\| \leq M\|\lambda^k - \lambda^*\|\mu_k^{-1}$。

拉格朗日增广函数性质

定理5：

如果x^*是原问题的解，并且在x^*处LICQ成立，二阶充分性条件在$\lambda=\lambda^*$处也成立。那么，存在一个阈值$\bar{\mu}$，当$\mu\geq\bar{\mu}$时，x^*是$\mathcal{L}_A(x,\lambda^*;\mu)$的一个严格局部极小子。

定理6：

如果定理5中条件在x^*和λ^*处成立，$\bar{\mu}$也如定理5中取值。那么存在正的δ, ϵ和M满足 对于所有的λ^k和μ_k满足

$$\|\lambda^k-\lambda^*\|\leq\delta\mu_k,\ \mu_k\geq\bar{\mu}$$

1. 问题 $\min_x \mathcal{L}_A(x,\lambda^k;\mu_k)$ subject to $\|x-x^*\|\leq\epsilon$ 存在唯一的解x_k, 并且满足 $\|x_k-x^*\|\leq M\|\lambda^k-\lambda^*\|\mu_k^{-1}$。
2. 如果$\lambda^{k+1}=\lambda^k-\mu_k c_i(x_k)$, 则有 $\|\lambda^{k+1}-\lambda^*\|\leq M\|\lambda^k-\lambda^*\|\mu_k^{-1}$.

拉格朗日增广函数性质

定理5：

如果x^*是原问题的解，并且在x^*处LICQ成立，二阶充分性条件在$\lambda=\lambda^*$处也成立。那么，存在一个阈值$\bar{\mu}$，当$\mu\geq\bar{\mu}$时，x^*是$\mathcal{L}_A(x,\lambda^*;\mu)$的一个严格局部极小子。

定理6：

如果定理5中条件在x^*和λ^*处成立，$\bar{\mu}$也如定理5中取值。那么存在正的δ, ϵ和M满足 对于所有的λ^k和μ_k满足

$$\|\lambda^k-\lambda^*\|\leq\delta\mu_k,\ \mu_k\geq\bar{\mu}$$

1. 问题 $\min_x \mathcal{L}_A(x,\lambda^k;\mu_k)$ subject to $\|x-x^*\|\leq\epsilon$ 存在唯一的解x_k, 并且满足 $\|x_k-x^*\|\leq M\|\lambda^k-\lambda^*\|\mu_k^{-1}$。
2. 如果$\lambda^{k+1}=\lambda^k-\mu_k c_i(x_k)$, 则有 $\|\lambda^{k+1}-\lambda^*\|\leq M\|\lambda^k-\lambda^*\|\mu_k^{-1}$.
3. 对应的Hessian阵$\nabla^2_{xx}\mathcal{L}_A(x_k,\lambda^k;\mu_k)$是正定的，并且所有约束的梯度$\nabla c_i(x_k)$是线性无关的。

拉格朗日增广函数实用方法：有界约束

- 如果出现不等式约束，可以转化为有界约束。引入变量s_i，把所有不等式约束$c_i(x) \geq 0$改为 $c_i(x) - s_i = 0$ 和 $s_i \geq 0$.

拉格朗日增广函数实用方法：有界约束

- 如果出现不等式约束，可以转化为有界约束。引入变量s_i，把所有不等式约束$c_i(x) \geq 0$改为 $c_i(x) - s_i = 0$ 和 $s_i \geq 0$.
- 原非线性规划的一般形式可以写为

$$\min_x f(x) \text{ subject to } c_i(x) = 0,\ i = 1, \cdots, m,\ l \leq x \leq u.$$

拉格朗日增广函数实用方法：有界约束

- 如果出现不等式约束，可以转化为有界约束。引入变量s_i，把所有不等式约束$c_i(x) \geq 0$改为 $c_i(x) - s_i = 0$ 和 $s_i \geq 0$.
- 原非线性规划的一般形式可以写为

$$\min_x f(x) \text{ subject to } c_i(x) = 0,\ i = 1, \cdots, m,\ l \leq x \leq u.$$

- 其对应的增广拉格朗日函数为

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} c_i^2(x).$$

拉格朗日增广函数实用方法：有界约束

- 如果出现不等式约束，可以转化为有界约束。引入变量s_i，把所有不等式约束$c_i(x) \geq 0$改为 $c_i(x) - s_i = 0$ 和 $s_i \geq 0$.
- 原非线性规划的一般形式可以写为

$$\min_x f(x) \text{ subject to } c_i(x) = 0,\ i = 1, \cdots, m,\ l \leq x \leq u.$$

- 其对应的增广拉格朗日函数为

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} c_i^2(x).$$

- 有界约束直接显示地在子问题中强制成立

$$\min_x\ \mathcal{L}_A(x, \lambda; \mu) \text{ subject to } l \leq x \leq u.$$

拉格朗日增广函数实用方法：有界约束

- 求解有界约束的方法主要是梯度投影法。

拉格朗日增广函数实用方法：有界约束

- 求解有界约束的方法主要是梯度投影法。
- 上述问题的一阶最优性条件为

$$x - P\big(x - \nabla_x \mathcal{L}_A(x, \lambda; \mu), l, u\big) = 0$$

其中$P(g, l, u)$是把向量g投影到有l, u确定的矩形区域算子

$$P(g, l, u)_i = \begin{cases} l_i & \text{if } g_i \leq l_i, \\ g_i & \text{if } g_i \in (l_i, u_i), \\ u_i & \text{if } g_i \geq u_i, \end{cases}$$

拉格朗日增广函数实用算法

Algorithm 4: 带有界约束的增广拉格朗日函数法

选择初始点x_0和初始乘子λ^0;选择收敛容忍误差η_*和ω_*;
设$\mu_0 = 10$, $\omega_0 = \mu_0^{-1}$ 以及$\eta_0 = \mu_0^{-0.1}$;
for $k = 0, 1, \cdots$
    找到上述子问题的一个解x_k满足
    $\|x_k - P\big(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u\big)\| \le \omega_k$;
    if $\|c(x_k)\| \le \eta_k$
        (% 测试收敛性条件);
        if $\|c(x_k)\| \le \eta_*$ 并且 $\|x_k - P\big(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u\big)\| \le \omega_*$
            stop with 近似解x_k;
        end(if)
        (% 更新拉格朗日乘子，收紧容忍误差)
        $\lambda^{k+1} = \lambda^k - \mu_k c(x_k)$, $\mu_{k+1} = \mu_k$, $\eta_{k+1} = \eta_k \mu_{k+1}^{-0.9}$, $\omega_{k+1} = \omega_k \mu_{k+1}^{-1}$;
    else
        (% 增加惩罚参数，收紧容忍误差)
        $\lambda^{k+1} = \lambda^k$, $\mu_{k+1} = 100\mu_k$, $\eta_{k+1} = \mu_{k+1}^{-0.1}$, $\omega_{k+1} = \mu_{k+1}^{-1}$
    end(if)
end(for)

THANKS FOR YOUR ATTENTION