

3.10 拉格朗日算法基础

#3.10拉格朗日乘子法 张满满

##讲义和规范

•<https://www.matongxue.com/madocs/939.html>

一、拉格朗日乘数法

拉格朗日乘数法是一种优化算法，主要运用于解决优化问题，它的基本思想就是用过拉格朗日乘子来把含有 m 个变量和 l 个约束条件的约束优化问题转换成含有 $(m+l)$ 个变量的无约束优化问题。

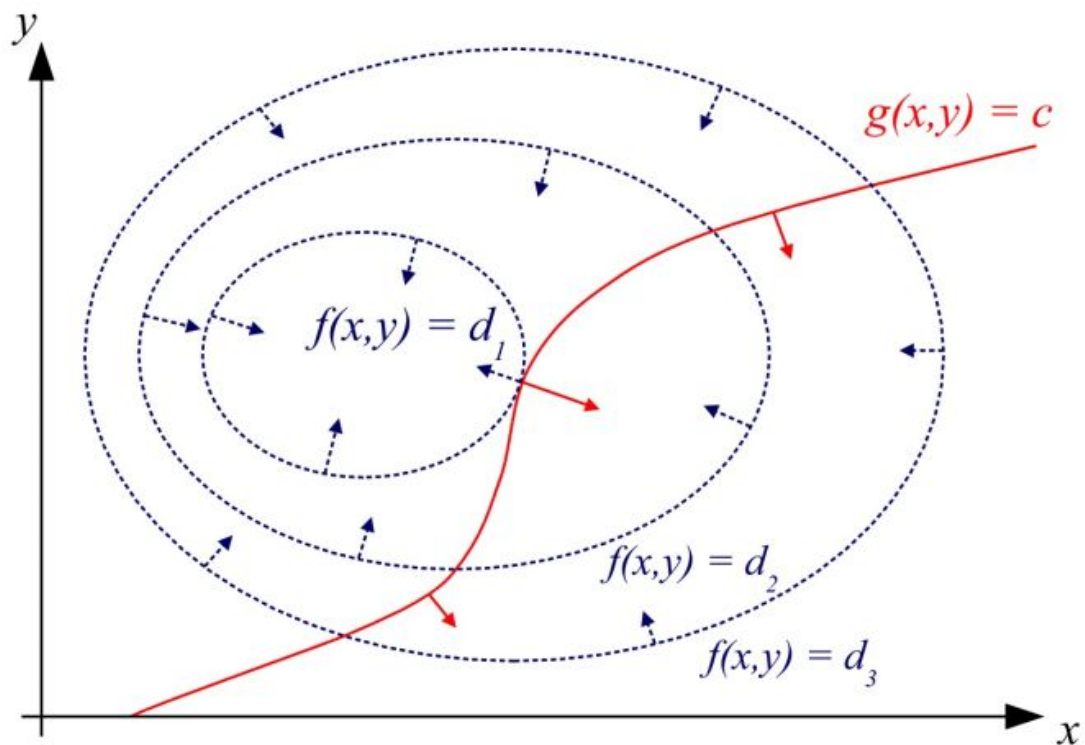
定义：

假设有自变量 x 和 y ，给定约束条件 $g(x,y)=c$ ，要求 $f(x,y)$ 在约束 g 下的极值。

我们可以画出 f 的等高线图，如下图。此时，约束 $g=c$ 由于只有一个自由度，因此也是图中的一条曲线（红色曲线所示）。显然地，当约束曲线 $g=c$ 与某一条等高线 $f=d_1$ 相切时，函数 f 取得极值。

两曲线相切等价于两曲线在切点处拥有共线的法向量。因此可得函数 $f(x,y)$ 与 $g(x,y)$ 在切点处的梯度（gradient）成正比。

于是我们便可以列出方程组求解切点的坐标 (x,y) ，进而得到函数 f 的极值



二、约束条件的集中形式**

- 等式条件下求最优解

$$\begin{aligned} & \min f(x, y) \\ \text{s.t.} \quad & h_i(x, y) = c, i = 1, 2, 3 \dots l \end{aligned}$$

- 不等式条件下求最优解

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & h_j(x) = 0, j = 1, 2, 3 \dots l \\ & g_i(x) \leq 0, i = 1, 2, 3 \dots l' \end{aligned}$$

三、求解不同约束条件问题的最优方法

(1) 等式条件下求最优解：

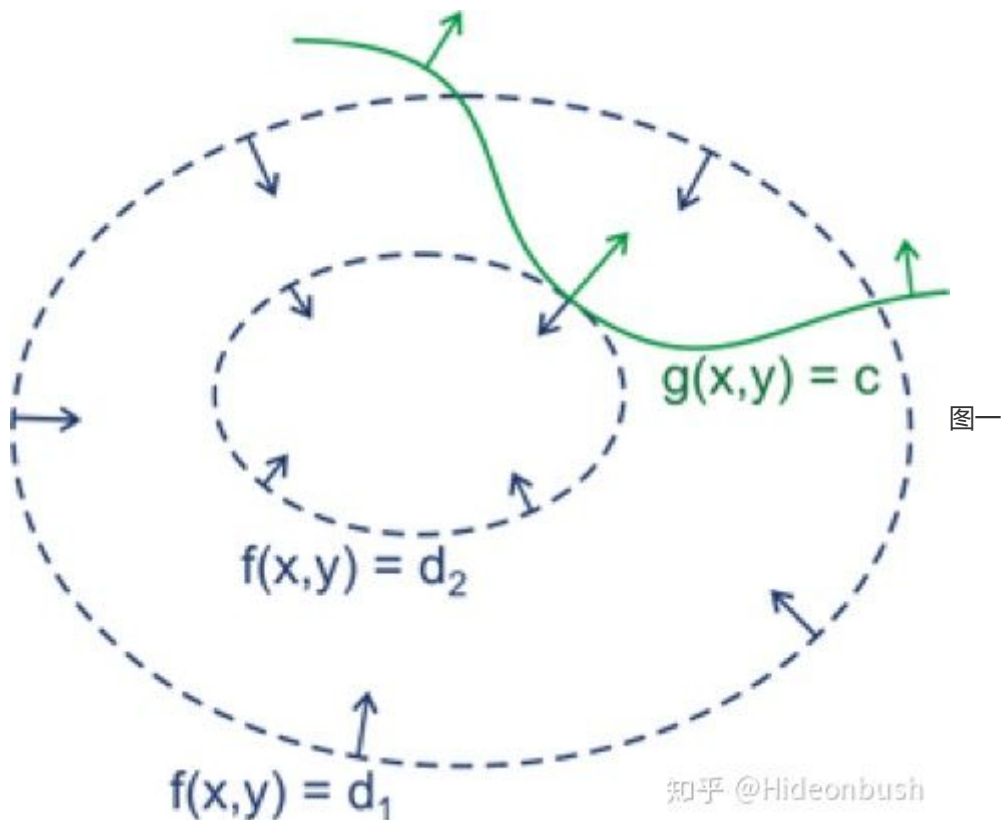
其实，高中的时候已经接触过等式条件机制求最优解的问题，都知道用拉格朗日乘数法可以求得最优解。首先构建一个Lagrange multiplier：

$$L(x, y, \lambda) = f(x, y) + \lambda(h(x) - c)b$$

构建出 $L(x, y, \lambda)$ 后，跟原来的问题对比一下，构建出的Lagrange multiplier把原来的等数条件约束情况变成含三个参数 (x, y, λ) 无条件极值的问题，最后可以通过求导数，令极值为零可以求出可行解 x 。

$$\nabla L(x, y, \lambda) = \nabla f(x) + \lambda \nabla(g(x) - c)$$

（拉格朗日乘子求的解不一定是最优解，其实只是局部最优解，这里称作可行解，只有在凸函数中才能保证最优解）至于为什么可以把原始的等式条件极值问题转化成求 $L(x, y, \lambda)$ 的极值问题？可以观察一下下面的二维图：



图一

知乎 @Hideonbush

蓝色虚线是目标函数 $f(x, y)$ 的等高线，绿色实现 $g(x, y) = c$ 是约束条件。图中，目标函数和条件函数有三种情况：

1、相离

对于相离的情况，我们以前学过，两个函数有交点才说明是两个函数的解，所以相离明显不行。

2、相交

两个函数相交的才是两个函数的解，但是相交得到的一定不是最优值，因为相交意味着肯定还存在其它的等高线在该条等高线的内部或者外部，使得新的等高线与目标函数的交点的值更大或者更小

3、相切

图中，等高线与条件函数相切时候，只有一个交点，是可行解。

当等高线与条件函数在可行解处（相切时候）的梯度平行有下式：

$$\nabla f(x) = -\lambda \nabla(g(x) - c)$$

对上式子移项后，可得到：

$$\nabla L(x, y, \lambda) = \nabla f(x) + \lambda \nabla(g(x) - c) = 0$$

该式子恰好与令Lagrange multiplier的导数为零是式子相同。

举一个小栗子：

$$\begin{aligned} \min f(x, y) &= x^2 + y^2 \\ \text{s.t.} \quad xy &= 3 \end{aligned}$$

根据上式子可知是一个典型的约束优化问题，约束条件是等式，可以用拉格朗日乘子。

$$L(x, y, \lambda) = x^2 + y^2 + \lambda(xy - 3)$$

原来的条件约束问题，转化为无约束方程组问题。

$$2x + \lambda y = 0$$

$$2y + \lambda x = 0$$

$$xy - 3 = 0$$

求得 $\lambda = \pm 2$ ，当 $\lambda = 2$ 时， $x = \sqrt{3}$ ， $y = \sqrt{3}$ ；当 $\lambda = -2$ ， $x = -\sqrt{3}$ ， $y = -\sqrt{3}$ 。

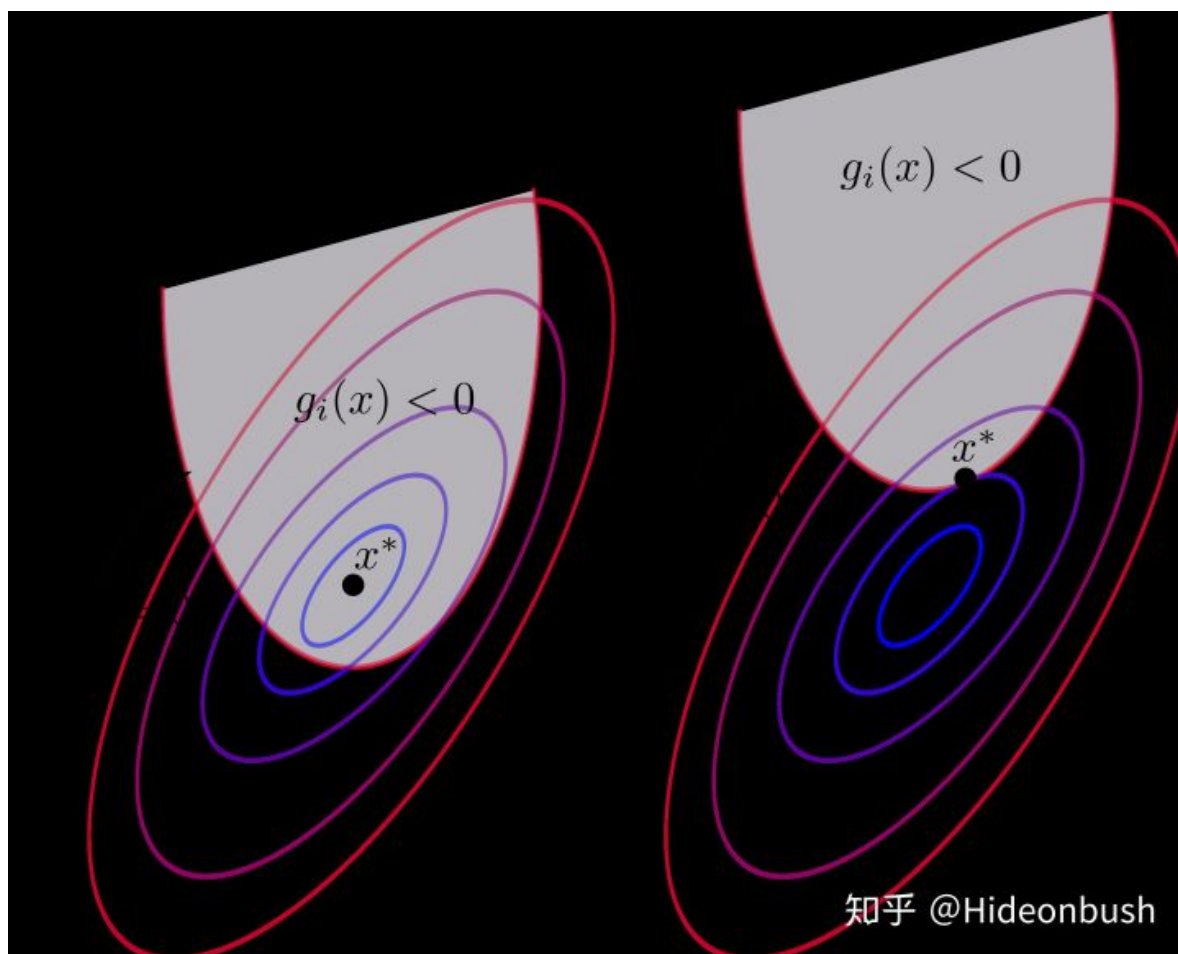
(2) 不等式条件下求最优解

上述都是等式条件约束的优化问题，但事实上，很多时候等式约束很难覆盖我们显示的问题，计算成本时候，通常说不能超过多少资金，不能超多少时间，都是一个范围内，通常面对更多的是不等式条件约束的情况，面对这种情况，可以通过增加KKT条件后，通过拉格朗日乘子求解不等式约束的优化问题。

可以把目标函数和所有的约束条件写成一个式子：

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{l'} \lambda_i g_i(x) + \sum_{j=1}^l \mu_j h_j(x)$$

对于不等式条件的情况有两种：即可行解在 $g(x) < 0$ 或者 $g(x) = 0$ 区域内取得（如下图）



图二

- 在 $g(x) < 0$ 内，就是没有限制条件下的最优解，正好满足限制条件，换句话说数约束函数不起作用（如上图左侧）这情况可以直接最小化目标函数，找处可行解 x^* ，所以有：

$$\nabla_{x^*} f(x^*) = 0, \quad \lambda = 0, \quad g(x^*) < 0$$

- 在 $g(x) = 0$ 上，换句话说，就是没有限制条件下的最优解，不能满足限制条件，需要条件起作用，此时 $\lambda \neq 0$ ，因此可行解在 $g(x) = 0$ 上，就变成等式条件约束的情况，经之前分析，可以证明可行解 x^* 发生在 $\nabla f(x) = -\lambda \nabla g(x)$ ，即是说，存在一个 λ 使得 $\nabla f(x^*) = -\lambda \nabla g(x^*)$ 。为确定 λ 范围，回看图一，梯度 $\nabla f(x)$ 方向与梯度 $\nabla g(x)$ 的方相反且梯度平行，所以，若要使得 $\nabla f(x) = -\lambda \nabla g(x)$ 成立，则 $\lambda > 0$ ，所以有：

$$g(x^*) = 0, \quad \lambda > 0$$

把两个情况综合考虑得到：

$$\begin{aligned} \lambda &> 0 \\ \lambda g(x^*) &= 0 \\ g(x^*) &\leq 0 \end{aligned}$$

以上就是KKT条件。

考虑不等式条件下求最优解时：

$$\begin{aligned} &\min f(x) \\ \text{s.t.} \quad &h_j(x) = 0, j = 1, 2, 3 \dots l \\ &g_i(x) \leq 0, i = 1, 2, 3 \dots l' \end{aligned}$$

定义拉格朗日函数：

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{l'} \lambda_i g_i(x) + \sum_{j=1}^l \mu_j h_j(x)$$

其中 λ_i 是 $g_i(x) \leq 0$ 的格朗日乘子， μ_j 是 $h_j(x) = 0$ 的拉格朗日乘子。

经过之前的分析观察，得知加上不等式约束后可行解 x 需要满足的就是以下的 KKT 条：

$$\nabla_x L(x, \lambda, \mu) = 0 \quad ①$$

$$h_j(x) = 0, \quad j = 1, 2, 3 \dots l \quad ②$$

$$g_i(x) \leq 0, \quad i = 1, 2, 3 \dots l' \quad ③$$

$$\lambda_i > 0, \quad i = 1, 2, 3 \dots l' \quad ④$$

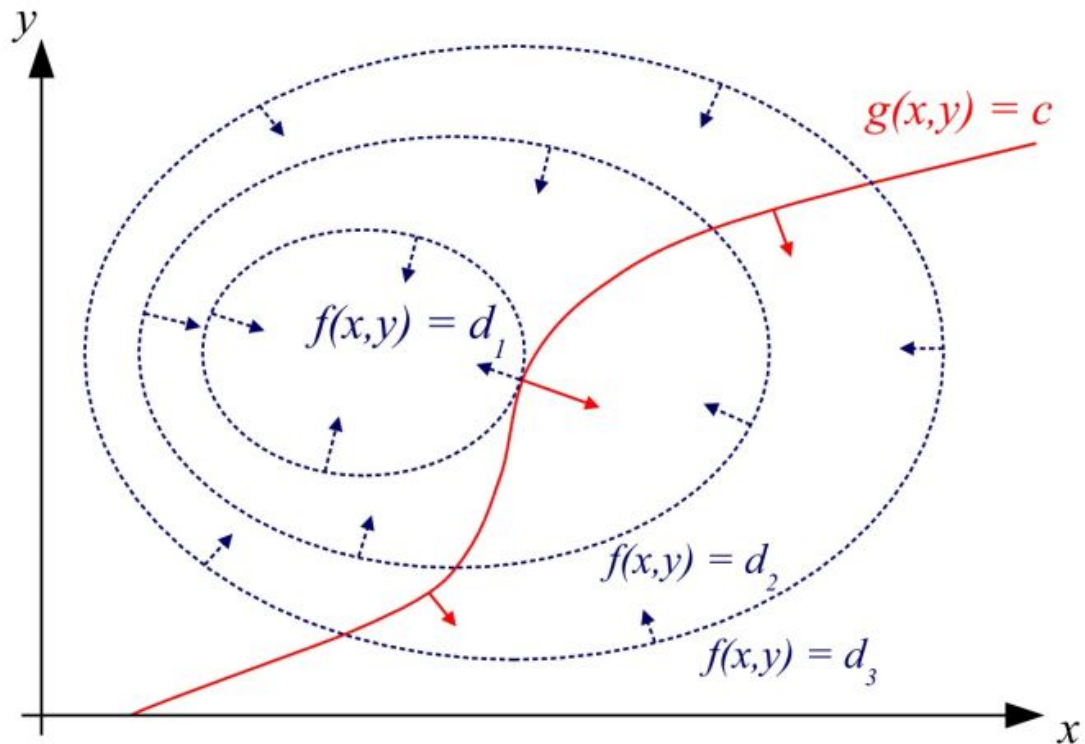
$$\lambda_i g_i(x) = 0, \quad i = 1, 2, 3 \dots l' \quad ⑤$$

定义：

假设有自变量x和y，给定约束条件g(x,y)=c，要求f(x,y)在约束g下的极值。

我们可以画出 f 的等高线图，如下图。此时，约束 $g=c$ 由于只有一个自由度，因此也是图中的一条曲线（红色曲线所示）。显然地，当约束曲线 $g=c$ 与某一条等高线 $f=d_1$ 相切时，函数 f 取得极值。两曲线相切等价于两曲线在切点处拥有共线的法向量。因此可得函数 $f(x,y)$ 与 $g(x,y)$ 在切点处的梯度（gradient）成正比。

于是我们便可以列出方程组求解切点的坐标 (x,y) ，进而得到函数 f 的极值



下面的例子是一道约束非线性优化问题，摘自高等数学规划作业题，原题为：

```
from scipy.optimize import fsolve
import sympy
esp1=0.0001
a0=-1
def check(a,b,w):
    s = a**2-b-3
    print "a**2-b-3:",s
    if s<float(w)/float(2):
        return True
    else:
        return False
def modify(w,v,x1,x2):
    w = max(0,w-2*((x1-2)**2-x2))
    v = v-2*(2*x1-x2-1)
    return (w,v)
def iterf(i,w,v,a):
    b=(-w-v+2*a**2-6+4*a)/6
    a=afs(w,v,b)
    x1=a+2
    x2=b+3
```

```

print "第%s次迭代: a=%s,b=%s,x1=%s,x2=%s" %(i,a,b,x1,x2)
wv = modify(w,v,x1,x2)
w = wv[0]
v = wv[1]
print "w=%s,v=%s" %(w,v)
#a = asolve(a,w,v,b)
return (w,v,a)
def asolve(a,w,v,b):
    #af = afs(a,w,v,b)
    return fsolve(afs(a,w,v,b),a0)
def afs(w,v,b):
    res = fsolve(lambda a: 2*a**3+(1-w-2*b-2)*a-v-2*b, 0)
    #res = sympy.solve('a**3+((1-w)/2-b-1)*a-v/2-b')
    flag = check(res,b,w)
    print "是否满足: ",flag
    return res

```