

模式识别第三次实验

邹永浩 2019211168

1. 公式

根据实验指导和 [Distributed Convex Optimization with Many Non-Linear Constraints](#) , 本次实验的相关公式如下 :

$$\min \sum (z_i^+ + z_i^-)$$

$$\text{s.t. } \max \{0, g_1(z_i^+)\}^2 = 0$$

$$\max \{0, g_2(z_i^-)\}^2 = 0$$

$$\max \{0, -z_i^+\}^2 = 0$$

$$\max \{0, -z_i^-\}^2 = 0$$

$$\text{其中, } g_1(z_i^+) = -z_i^+ - R + x_i^2 - 2ax_i + A + y_i^2 - 2by_i + B$$

$$g_2(z_i^-) = -z_i^- + R - x_i^2 + 2ax_i - A - y_i^2 + 2by_i - B$$

则拉氏式为

$$L(z_i^+, z_i^-, a, b, r, \overset{u_1, u_2, u_3, u_4}{\cancel{u_1, u_2, u_3, u_4}})$$

$$\begin{aligned} = & \sum (z_i^+ + z_i^-) + \frac{\rho}{2} \|\max \{0, g_1(z_i^+)\}^2\|^2 + u_1^* \max \{0, g_1(z_i^+)\}^2 \\ & + \frac{\rho}{2} \|\max \{0, g_2(z_i^-)\}^2\|^2 + u_2^* \max \{0, g_2(z_i^-)\}^2 \\ & + \frac{\rho}{2} \|\max \{0, -z_i^+\}^2\|^2 + u_3^* \max \{0, -z_i^+\}^2 \\ & + \frac{\rho}{2} \|\max \{0, -z_i^-\}^2\|^2 + u_4^* \max \{0, -z_i^-\}^2 \end{aligned}$$

则参数更新为

$$\xi_i^{+k+1} = \operatorname{argmin}_{\xi_i} L(\xi_i^+, \xi_i^-, a, b, r)$$

$$u_i^{k+1} = u_i^k + p \max\{0, g(\xi_i^{k+1}, y^2)$$

2. 代码

主要约束 g_1 和 g_2

```
# g1 和 g2
def g1(a, b, r, ksi1, x, y):
    return -ksi1 - r + x * x - 2 * a * x + a * a + y * y - 2 * b * y + b *
b

def g2(a, b, r, ksi2, x, y):
    return -ksi2 + r - x * x + 2 * a * x - a * a - y * y + 2 * b * y - b *
b
```

参数迭代

```
# 求梯度
grad_ksi1 = 0.1 + p * tmp_g1 * tmp_g1 * 2 * tmp_g1 * tmp_g1_ksi1 \
    + p * 2 * u1[k] * tmp_g1 * tmp_g1_ksi1 \
    + 2 * p * (max(0, -ksi1[k]) ** 3) * -1 \
    + 2 * u3[k] * (max(0, -ksi1[k])) * -1

grad_ksi2 = 0.1 + p * tmp_g2 * tmp_g2 * 2 * tmp_g2 * tmp_g2_ksi2 \
    + p * 2 * u2[k] * tmp_g2 * tmp_g2_ksi2 \
    + 2 * p * (max(0, -ksi2[k]) ** 3) * -1 \
    + 2 * u4[k] * (max(0, -ksi2[k])) * -1

grad_r = p * tmp_g1 * tmp_g1 * 2 * tmp_g1 * tmp_g1_r \
```

```

+ p * 2 * u1[k] * tmp_g1 * tmp_g1_r \
+ p * tmp_g2 * tmp_g2 * 2 * tmp_g2 * tmp_g2_r \
+ p * 2 * u2[k] * tmp_g2 * tmp_g2_r

grad_a = p * tmp_g1 * tmp_g1 * 2 * tmp_g1 * tmp_g1_a \
+ p * 2 * u1[k] * tmp_g1 * tmp_g1_a \
+ p * tmp_g2 * tmp_g2 * 2 * tmp_g2 * tmp_g2_a \
+ p * 2 * u2[k] * tmp_g2 * tmp_g2_a

grad_b = p * tmp_g1 * tmp_g1 * 2 * tmp_g1 * tmp_g1_b \
+ p * 2 * u1[k] * tmp_g1 * tmp_g1_b \
+ p * tmp_g2 * tmp_g2 * 2 * tmp_g2 * tmp_g2_b \
+ p * 2 * u2[k] * tmp_g2 * tmp_g2_b

# 更新
ksi1[k] -= step * grad_ksi1
ksi1[k] = ksi1[k] if ksi1[k] > 0 else 0

ksi2[k] -= step * grad_ksi2
ksi2[k] = ksi2[k] if ksi2[k] > 0 else 0

r -= step * grad_r
a -= step * grad_a
b -= step * grad_b

```

拉格朗日参数迭代

```

tmp_g1 = max(0, g1(a, b, r, ksi1[k], x, y))
tmp_g2 = max(0, g2(a, b, r, ksi2[k], x, y))

# update u1, u2, u3, u4

u1[k] += p * tmp_g1 * tmp_g1

u2[k] += p * tmp_g2 * tmp_g2

u3[k] += p * max(0, -ksi1[k])

u4[k] += p * max(0, -ksi2[k])

```

3. 结果

data index	a	b	r
0	0.9745374281009183	0.6235492104547817	0.08489968590589783
1	0.7284045964651864	0.004988300763539417	0.08712502493068193
2	0.1326842217350493	0.6000539207299932	0.11622189887728795

data index	a	b	r
3	0.7923196983268246	0.31831212759932553	0.2148619069738617
4	0.5451907892810237	0.974882989021801	0.5112548818541877