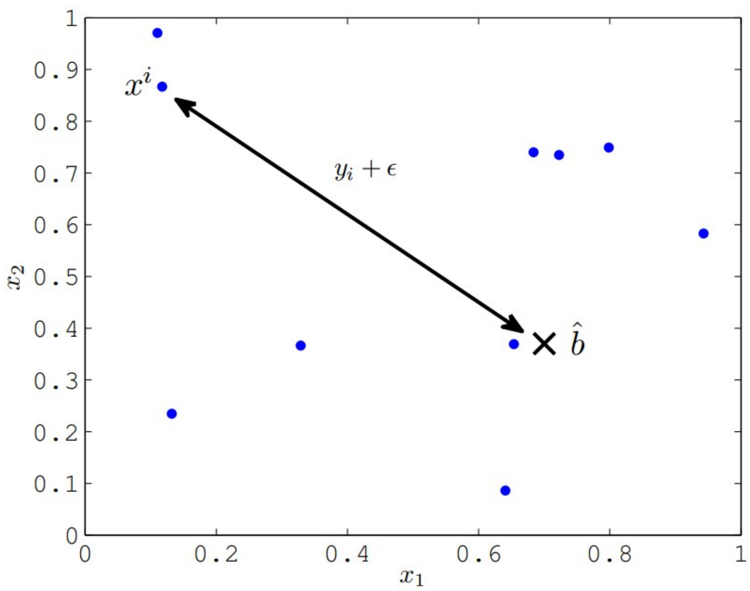
Final Project

1. 问题

在一片区域内$[0,1]^2$,有着10个接收器,分别位于$(x_1^1, x_1^2), (x_2^1, x_2^2), ..., (x_{10}^1, x_{10}^2)$, 并且区域内有一个无线发射器,位于$(b_1, b_2)$(未知). 这些接收器可以测量发射器的距离,但存在一些误差, 请根据测量数据求出无线发射器的位置.
接收器位置数据如下表所示

x_1	**0.1104**	**0.1175**	**0.6407**	**0.3288**	**0.6538**
x_2	0.9706	0.8669	0.0862	0.3664	0.3692
x_1	0.7984	0.9430	0.6837	0.1321	0.7227
x_2	0.7491	0.5832	0.7400	0.2348	0.7350

2. 题目分析



由图像可知,接收器接收信号后测出距离的误差为
$r_i(b) = \sqrt{(b_1 - x_i^1)^2 + (b_2 - x_i^2)^2} - y_i$

为了准确求得发射器的位置,因此要求得使误差最小的(b_1, b_2),即使得

$\phi(b) = \frac{1}{2}\sum_{i=1}^{10} r_i^2(b)$ 最小的(b_1, b_2).

因此这是一个最小二乘问题. 又$r_i(\alpha + \beta) \neq r_i(\alpha) + r_i(\beta)$, 因此, 这是一个非线性最小二乘问题.

求解非线性最小二乘问题的方法有很多,可以使用高斯牛顿法, L-M方法等. 但高斯牛顿法可能会存在Hessian矩阵不可逆的问题, 因此可采用Levenberg–Marquardt法求解.

Levenberg–Marquardt算法流程为

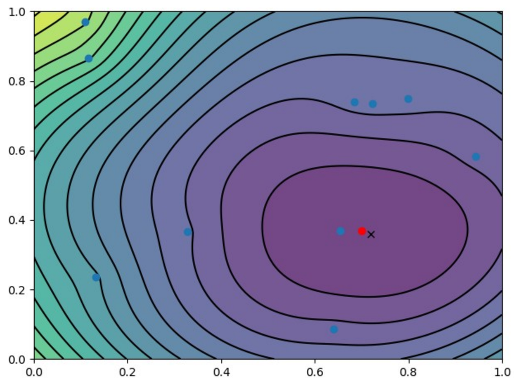
Algorithm 1 Levenberg-Marquardt

begin
$k := 0; v := 2; x := x_0$
$H := J^T J; g := J^T f$
found $= (||g||_\infty < \epsilon_1); \mu := \gamma \cdot \max a_{ii}$
while (not found and $k < k_{max}$) **do**
$k := k + 1; Solve(H + \mu I)h = -g$
if $(||h|| < \epsilon_2(||x|| + \epsilon_2))$ **then**
found = true;
else {}

$x_{new} := x + h$
$\delta = \frac{F(x)-f(x+h)}{L(0)-L(h)}$
end if
if $(\delta > 0)$ **then**
$x := x_{new}$
$H = J^T J; g := J^T f$
found $= (||g||_\infty < \epsilon_1)$
$\mu := \mu \cdot \max\{\frac{1}{3}, 1-(2\delta-1)^3\}; v := 2$
else {}
$\mu := \mu \cdot v; v := 2 \cdot v$
end if
end while

3. 结果与分析

实验结果如下图所示



其中红色点为拟合出来的(b_1, b_2), 叉为真实点的位置. 等高线为$\phi(b)$的等高线, 目的是找到使得$\nabla\phi=0$
最终的计算结果为
Predicted location: [0.71941894 0.35915887]
grad(phi): [-9.71445147e-17 -2.08166817e-17]
phi: 0.005975188570976857

可以看到误差的平方和仅有0.005975188570976857, 可以接受.

4. 结论

Levenberg-Marquardt方法可以有效地克服Gauss-Newton法所遇到的困难,并且可以很好的解决非线性最小二乘问题.

5. 参考文献

[1] 吴福朝. 计算机视觉中的数学方法[M]. 北京：科学出版社, 2008.03.
[2] 王景恒. 最优化理论与方法[M]. 北京：北京理工大学出版社, 2018.08
[3] 李春明. 优化方法[M]. 南京：东南大学出版社, 2009.10.
[4] 陈卫东，蔡荫林，于诗源. 工程优化方法[M]. 哈尔滨：哈尔滨工程大学出版社, 2006.02..
[5] 许国根，赵后随，黄智勇. 最优化方法及其MATLAB实现[M]. 北京：北京航空航天大学出版社, 2018.07.
[6] 万仲平，费浦生. 优化理论与方法[M]. 武汉：武汉大学出版社, 2004.06.
[7] 姚恩瑜，何勇等. 数学规划与组合优化[M]. 杭州：浙江大学出版社, 2001.10.
[8] 卢险峰. 最优化方法应用基础[M]. 上海：同济大学出版社, 2003.08..
[9] 席少霖. 非线性最优化方法[M]. 北京：高等教育出版社, 1992.06.

6. 附录

python代码如下

```python
from math import *
import numpy as np
import matplotlib.pyplot as plt
from random import random
from scipy.optimize import root

# Define the transmitter's true location
bx_t = 0.7
by_t = 0.37

x_beac = [0.7984, 0.9430, 0.6837, 0.1321, 0.7227, 0.1104, 0.1175, 0.6407, 0.3288, 0.6538]
y_beac = [0.7491, 0.5832, 0.7400, 0.2348, 0.7350, 0.9706, 0.8669, 0.0862, 0.3664, 0.3692]

# Generate the (noisy) data y, and set initial guess
noise_level = 0.05
y = np.zeros(10)
for i in range(10):
    dx = bx_t - x_beac[i]
    dy = by_t - y_beac[i]
    y[i] = sqrt(dx * dx + dy * dy) + noise_level * random()
b_init = np.array([0.4, 0.9])

# Function to be minimized
def phi(x):
    s = 0
    for i in range(10):
        dx = x[0] - x_beac[i]
        dy = x[1] - y_beac[i]
        ss = sqrt(dx * dx + dy * dy) - y[i]
        s += ss * ss
    return s

# Gradient
def grad_phi(x):
    f0 = 0
    f1 = 0
    for i in range(10):
        dx = x[0] - x_beac[i]
        dy = x[1] - y_beac[i]
        d = 1 / sqrt(dx * dx + dy * dy)
        f0 += 2 * dx - 2 * y[i] * dx * d
        f1 += 2 * dy - 2 * y[i] * dy * d
    return np.array([f0, f1])

# Do Levenberg-Marquardt algorithm
sol = root(grad_phi, b_init, jac=False, method='lm')
print("Predicted location:", sol.x)
print("grad(phi):", grad_phi(sol.x))
print("phi:", phi(sol.x))

# Plot results
n = 100
xx = np.linspace(0, 1, n)
yy = np.linspace(0, 1, n)
X, Y = np.meshgrid(xx, yy)
pxy = np.zeros((n, n))
for i in range(n):
    for j in range(n):
        pxy[i, j] = phi([X[i, j], Y[i, j]])
plt.contourf(X, Y, pxy, 16, alpha=.75)
C = plt.contour(X, Y, pxy, 16, colors='black')
plt.plot(x_beac, y_beac, 'o')
plt.plot(sol.x[0], sol.x[1], 'x', color='black')
plt.plot(bx_t, by_t, 'o', color='red')
plt.show()
```