Homework 04 Levenberg-Marquardt Algorithm

王晓捷 (11521053)

May 18, 2016

1 引言

Levenberg-Marquardt算法（简称LM算法）通常用于非线性最小二乘法的目标函数极小化，是最优化方法中的置信域方法的一种。最优化就是寻找使得函数值最小或最大的参数向量。它的应用领域非常广泛，如：经济学、管理优化、网络分析等。根据使用的模型不同，可以分为无约束最优化、约束最优化以及最小二乘最优化。

在最优化算法中，都可以看做是求一个函数的极小值，每一步迭代中，都要求目标函数值是下降。最优化方法的基本结构如下：

给定初始点x_0

- 确定搜索方向d_k，即按照一定规则构造f在x_k点处的下降方向为搜索方向
- 确定步长因子α_k，使目标函数值有某种意义下降
- 令$x_{k+1} = x_k + \alpha_k d_k$
  - 若x_{k+1}满足某种终止条件，则停止迭代，得到近似最优解
  - 否则，重复以上步骤

Lenvenberg-Marquardt是使用最广泛地非线性最小二乘算法，属于一种信赖域法。信赖域方法的主要思想是：

- 首先选择一个步长r，使得$\nabla x - x_k \nabla < r$范围内（信赖域）
- 目标函数用n维二次模型来逼近，并依次选择一个搜索方向s_k，取$x_{k+1} = x_k + s_k$

Lenvenberg-Marquardt算法利用梯度求最大值或最小值的算法。同时具有梯度法和牛顿法的优点。当λ 很小时，步长等于牛顿法步长；当λ 很大时，步长约等于梯度下降法的步长[1]。

2 方法概述

假设最优化问题的目标函数为$f(x)$，使用信赖域方法，采用l_2范数，原模型等效于

$$\min q^{(k)} = f_k + g_k^T s + \frac{1}{2} s^T G_k s \qquad s.t. \|s\|_2 \le h_k \tag{2.1}$$

引入*Lagrange*函数有：

$$L(s,u) = q^{(k)}(s) + \frac{1}{2}\mu(s^T s - h_k^2) \tag{2.2}$$

根据约束最优化的最优性条件知：$\nabla_s L = 0$, $\mu = 0$。从而推出：

$$L(s,u) = q^{(k)}(s_k) + \frac{1}{2}(s - s_k)^T (G_k + u_k I)(s - s_k) \tag{2.3}$$

为求得总体最优解，LM方法要确定一个$\mu_k \ge 0$，使得$(G_k + u_k I)$正定，并用$g_k + (G_k + u_k I)s = 0$（即$\nabla_s L = 0$）求解s_k[2]。

Levenberg-Marquardt算法伪代码如下[3]：

(a) 给定初始点x_0，$\mu_0 > 0, k = 0$

(b) 计算g_k, G_k

(c) 若$\nabla g_k \nabla < \varepsilon$, return x_k

(d) $k++$

(e) 求解$G_k + u_k I s_k = g_k$得s_k

- 若$\nabla s_k \nabla < \nabla x_{k-1} \nabla$，return x_{k-1}

(f) 计算$r_k = \frac{\triangle f_k}{\triangle q^{(k)}}$

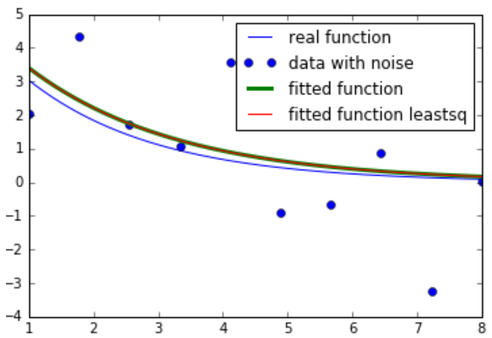
(g) 如果$r_k > 0$，更新$x_k = x_{k-1} + s_k$，并以一定比例更新u_k

(h) 否则，$x_k = x_{k-1}$，更新$\mu_k + 1 = v * \mu_k$，转至(b)

3 实验结果

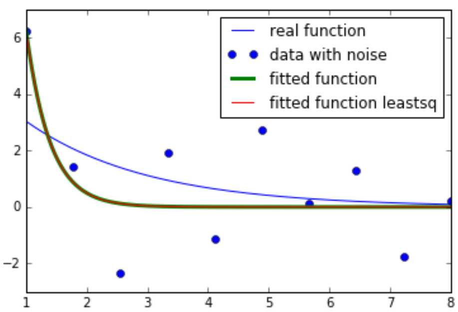
为了判断LM算法的可靠性，实验中使用了Python的scopy.optimize包中的leastsq方法进行验证。图1，图2和图3左半部分为函数的绘制，蓝色线条为真实函数，红色细线条为由LM算法求解得到的函数，绿色粗线为由leastsq方法求解得到的函数，蓝色圆点为添加高斯噪声后的真实数据；右半部分反映了实验的迭代过程，最后一行是由leastsq方法计算出的参数值，前面每一行表示“迭代次数：[参数] 当前函数值”。

图1和图2对应的真实函数均为$f(x) = ae^{-bxa}$，其中，$a = 5, b = 0.5$，在$[1, 8]$ 之间抽取了10个点，添加了高斯噪声后，由LM算法计算最小二乘来求解参数a 和b，实验结果如下两图所示。




```
1 : [ 10.03058201   1.21487195] 6.18132047056
2 : [ 9.59084618  0.76042816] 5.72848678058
3 : [ 7.40297894  0.58578658] 5.47966247669
4 : [ 4.68955799  0.37929441] 5.36922144967
5 : [ 5.41519443  0.45986929] 5.35866525158
6 : [ 5.05727643  0.40869907] 5.35639268051
7 : [ 5.31997998  0.44312977] 5.35458469962
8 : [ 5.16521859  0.42204015] 5.35405816667
9 : [ 5.26262775  0.43524474] 5.35380177744
10 : [ 5.20484866  0.42726755] 5.35371548389
11 : [ 5.23972658  0.43210069] 5.35368082799
12 : [ 5.21921575  0.42922695] 5.3536687817
13 : [ 5.23129173  0.43092883] 5.35366437409
14 : [ 5.22427816  0.42993255] 5.35366285689
15 : [ 5.22833531  0.43051233] 5.35366233062
16 : [ 5.22600855  0.4301777 ] 5.35366215347
17 : [ 5.22733602  0.43036968] 5.35366209421
18 : [ 5.22658353  0.43026026] 5.35366207474
19 : [ 5.22658353  0.43026026] 5.35366207474
leastsq method: [ 5.22681868  0.43029603]
```

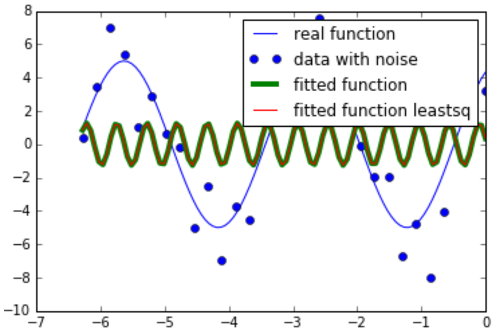
图 1: $f(x) = ae^{-bxa}, a = 5, b = 0.5$




```
1 : [ 10.10921226   0.92025695] 5.60855884051
2 : [ 10.40757006   0.90527119] 5.58170125816
3 : [ 11.25691134   0.96344316] 5.52058119333
4 : [ 13.31874177   1.08177278] 5.39716762436
5 : [ 17.25306507   1.27633611] 5.23208912928
6 : [ 21.56595036   1.42688184] 5.1063592831
7 : [ 27.38190942   1.61921825] 5.00697750857
8 : [ 32.71466499   1.74974819] 4.94660991441
9 : [ 39.72725895   1.92177882] 4.90343319177
10 : [ 45.61544813   2.03301816] 4.87959644204
11 : [ 52.97913002   2.17189983] 4.86369742196
12 : [ 59.01280593   2.26570008] 4.85551914368
13 : [ 65.55266748   2.3645285 ] 4.85088593378
14 : [ 70.66639865   2.43296969] 4.84886932803
15 : [ 74.92161498   2.48781682] 4.8480882378
16 : [ 77.46726519   2.5186987 ] 4.84789408555
17 : [ 78.6210706    2.53243287] 4.84786749053
18 : [ 78.89716366   2.53565941] 4.84786627662
19 : [ 78.9235204    2.53595909] 4.84786626581
[ 78.93664804   2.53612116]
```

图 2: $f(x) = ae^{-bxa}, a = 5, b = 0.5$

图3对应的真实函数为$f(x) = a\sin(2b\pi x + c)$，其中，$a = 5, b = 0.34, c = \frac{\pi}{3}$，在$[0, -2\pi]$之间抽取了30个点，添加了高斯噪声后，由LM算法计算最小二乘来求解参数a，b和c，实验结果如下图所示。




```
1 : [ 0.99833195 -2.00748595  1.00052338] 23.9656422065
2 : [ 0.99391415 -2.02157581  1.00156715] 23.867338429
3 : [ 0.98405018 -2.0401643   1.00306424] 23.7272638682
4 : [ 0.96505947 -2.05955738  1.00294597] 23.5969170789
5 : [ 0.94183345 -2.08213375  0.9833479 ] 23.4788753495
6 : [ 0.9842605  -2.10050503  0.88298392] 23.4285681934
7 : [ 1.09857655 -2.10932941  0.64239927] 23.3876793931
8 : [ 1.21325246 -2.11915812  0.38134263] 23.3631837659
9 : [ 1.25807286 -2.1223841   0.29266243] 23.3607453593
10 : [ 1.26255308 -2.12292064  0.27795777] 23.3606944389
11 : [ 1.26262928 -2.12298848  0.27611296] 23.3606937817
12 : [ 1.26262928 -2.12298848  0.27611296] 23.3606937817
leastsq method: [ 1.26262219 -2.12299569  0.27592327]
```

图 3: $f(x) = a\sin(2b\pi x + c), a = 5, b = 0.34, c = \frac{\pi}{3}$

4 小结与讨论

从图1、图2以及图3的右半部分都可以看到，实现的LM算法与leastsq方法返回的参数结果十分接近，可以确定LM算法的实现是正确的。

图1中显示出拟合效果很好，求解得到的参数与真实参数十分接近，曲线接近重合。然而，在图2与图1的函数模型、真实参数、取样点以及参数初始值均相同的情况下，图2的拟合效果却比图1差很多，而且求解得到的参数也相差甚远。我认为原因是因为在原始真实数据上所添加的高斯噪声不同的原因。这是因为LM算法是用来求解非线性最小二乘，而最小二乘方法会受到噪声的影响，噪声越大，则影响越大，拟合效果就会越差。

图3的效果可以看出来也很差。我认为这可能是因为sin函数是一个周期函数的影响，还需要具体考虑一下。这也说明LM算法并不是对所有的函数都有效，需要根据函数模型等现实因素去选择合适的最优化方法。

参考文献

[1] Andrew Ng. Cs229 lecture notes. CS229 Lecture notes, 1(1):1–3, 2000.

[2] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In Numerical analysis, pages 105–116. Springer, 1978.

[3] Manolis IA Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar. Foundation of Research and Technology, 4:1–6, 2005.