高斯-牛顿法(Guass-Newton Algorithm)与莱文贝格-马夸特方法(Levenberg–Marquardt algorithm)求解非线性最小二乘问题

发表于**2018年8月11日**

众所周知，最小二乘法通过最小化误差平方和获得最佳函数。有时候你可能产生疑问，为什么不能通过其他方式获得最优函数，比如说最小化误差的绝对值的和？本文中，我将会从概率的角度解释最小二乘法的依据（参考自andrew ng 《机器学习课程》 第三讲）。最小二乘问题可以分为线性最小二乘和非线性最小二乘两类，本文的目标是介绍两种经典的最小二乘问题解法：高斯牛顿法与莱文贝格-马夸特方法。实际上，后者是对前者以及梯度下降法的综合。

最小二乘法的概率解释(probabilistic interpretation)

以线性回归为例，假设最佳函数为 $y = \theta^T \mathbf{x}$ (θ, x为向量), 对于每对观测结果$(x^{(i)}, y^{(i)})$，都有

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

其中 ϵ为误差，基于一种合理的假设（中心极限定理），我们可以认为误差的分布服从正态分布(又称高斯分布)，即 $\epsilon \sim N(0, \sigma^2)$ ，那么，我们可以认为$y^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$,根据正态分布的概率公式

$$P(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2})$$

在统计学中，将所有的$P(y|x)$累乘作为θ的似然函数,用以衡量θ的或然性(likelihood)

$$L(\theta) = P(y|x;\theta) = \prod_{i=0}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2})$$

最佳的参数θ应该是使得所有数据出现的概率最大的那个，这个过程称之为最大似然估计(maximum likelihood estimation)。为了数学计算上的便利，采用单调函数:log函数$l(\theta) = log(L(\theta))$，称为对数似然函数代表$L(\theta)$:

$$
\begin{aligned}
l(\theta) &= log\prod_{i=0}^{m}\frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}) \\
&= \sum_{i=0}^{m} log\frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}) \\
&= mlog\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{i=0}^{m}(y^{(i)}-\theta^T x^{(i)})^2
\end{aligned}
$$

需要指出的是，在Andrew ng的讲解认为σ不影响θ的决定，因此，最大化$l(\theta)$等同于最小化$\sum_{i=0}^{m}(y^{(i)}-\theta^T x^{(i)})^2$。最小二乘法通过最小化误差平方和得到最佳函数的方法存在概率论方面的基础。

高斯-牛顿法(Guass-Newton Algorithm)

高斯-牛顿法是在牛顿法基础上进行修改得到的，用来(仅用于)解决非线性最小二乘问题。高斯-牛顿法相较牛顿法的最大优点是不需要计算二阶导数矩阵(Hessian矩阵)，当然，这项好处的代价是其仅适用于最小二乘问题。如下是其推导过程：

最小二乘方法的目标是令残差的平方和最小：

$$
f(\theta) = \frac{1}{2}\sum_{i=0}^{m} r(\mathbf{x_i})^2
$$

采用牛顿法求解$f(\theta)$的最小值，需要计算其梯度向量与Hessian矩阵。

$$
\nabla_\theta f = \frac{\partial f}{\partial \theta} = \sum r_i \frac{\partial r_i}{\partial \theta} = \begin{bmatrix} r(x_1) & r(x_2) & \dots & r(r_m) \end{bmatrix} \begin{bmatrix} \nabla_\theta r(x_1)^T \\ \nabla_\theta r(x_2)^T \\ \vdots \\ \nabla_\theta r(x_m)^T \end{bmatrix}
$$

其中

$$
J_r(\theta) = \left[\frac{\partial r_j}{\partial \theta_i}\right]_{j=1,\dots,m;i=1,\dots,n} = \begin{bmatrix} \nabla_\theta r(x_1)^T \\ \nabla_\theta r(x_2)^T \\ \vdots \\ \nabla_\theta r(x_m)^T \\ \end{bmatrix}
$$

称为r的雅各比(Jacobian)矩阵。因此上式可以写作

$$
\nabla_\theta f = r^T J_r = J_r^T r
$$

其中$r = \begin{bmatrix} r(x_1) & r(x_2) & \dots & r_m \end{bmatrix}^T$。再看Hessian矩阵的计算：

$$
H = \left[\frac{\partial^2 f}{\partial \theta^2}\right] = \sum \left[r_i \frac{\partial^2 r_i}{\partial \theta^2} + (\frac{\partial r_i}{\partial \theta})(\frac{\partial r_i}{\partial \theta})^T \right]
$$

观察二阶导数项$r_i \frac{\partial^2 r_i}{\partial \theta^2}$，因为残差 $r_i \approx 0$,因此我们可以认为此项接近于0而舍去。所以Hessian矩阵可以近似写成：

$$
H \approx \sum \left[(\frac{\partial r_i}{\partial \theta})(\frac{\partial r_i}{\partial \theta})^T \right] = J_r^T J_r
$$

这里我们可以看到高斯-牛顿法相对于牛顿法的不同就是在于采用了近似的Hessian矩阵降低了计算的难度，但是同时，舍去项仅适用于最小二乘问题中残差较小的情形。

将梯度向量，Hessian矩阵(近似)带入牛顿法求根公式，得到高斯-牛顿法的迭代式：

$$\theta_i = \theta_{i-1} - (J_r^T J_r)^{-1} J_r^T r$$

只需要计算出$m \times n$的Jacobian矩阵便可以进行高斯-牛顿法的迭代，计算已经算是非常简便的了。

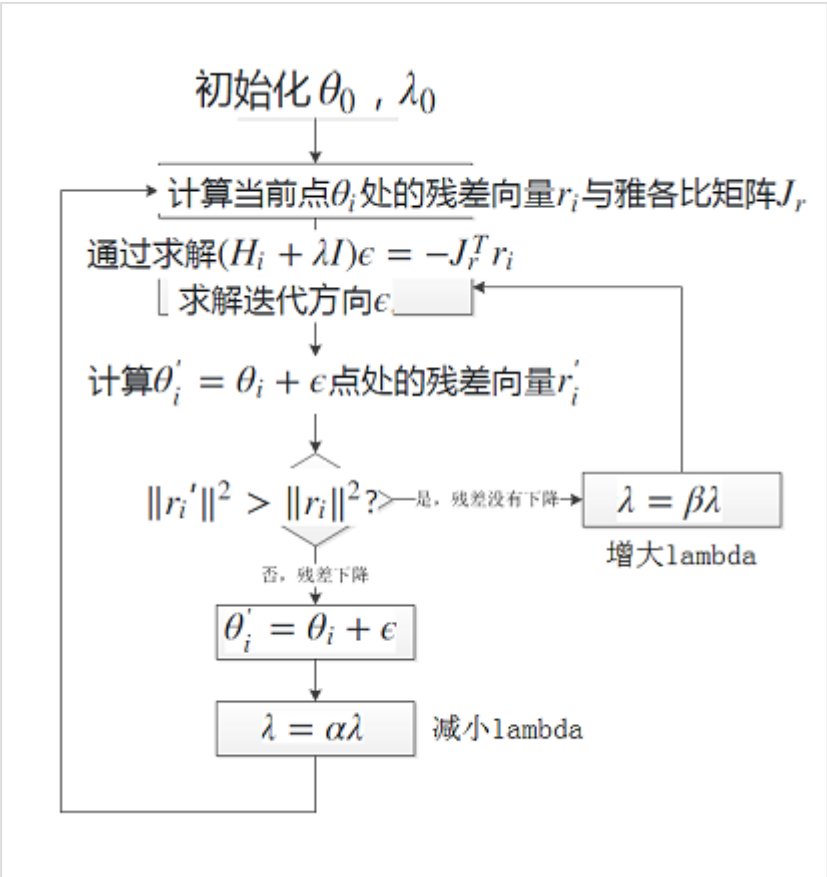
Levenberg-Marquart 算法

与牛顿法一样，当初始值距离最小值较远时，高斯-牛顿法的并不能保证收敛。并且当$J_r^T J_r$近似奇异的时候，高斯牛顿法也不能正确收敛。Levenberg-Marquart 算法是对上述缺点的改进。L-M方法是对梯度下降法与高斯-牛顿法进行线性组合以充分利用两种算法的优势。通过在Hessian矩阵中加入阻尼系数λ来控制每一步迭代的步长以及方向：

$$(H + \lambda I)\epsilon = -J_r^T r$$

- 当λ增大时，$H + \lambda I$趋向于λI，因此ϵ趋向于 $-\lambda J_r^T r$，也就是梯度下降法给出的迭代方向；
- 当λ减小时，$H + \lambda I$趋向于H，ϵ趋向于$-H^{-1} J_r^T r$，也就是高斯-牛顿法给出的方向。

λ的大小通过如下规则调节，也就是L-M算法的流程：



1. 初始化 θ_0，λ_0。
2. 计算当前点θ_i处的残差向量r_i与雅各比矩阵J_r。
3. 通过求解$(H_i + \lambda I)\epsilon = -J_r^T r_i$ 求解迭代方向ϵ。
4. 计算$\theta_i' = \theta_i + \epsilon$ 点处的残差向量r_i'。
5. 如果$\|r_i\prime\|^2 > \|r_i\|^2$?,即残差没有下降，则更新$\lambda = \beta\lambda$，增大$\lambda$重新回到第三步重新求解新的$\epsilon$。如果残差下降，则更新$\theta_{i+1} = \theta_i + \epsilon$ ，到第二步，并且降低$\lambda = \alpha\lambda$，增大迭代步长。

在曲线拟合实践中，α通常选取 0.1，β选取10。

相比于高斯-牛顿法，L-M算法的优势在于非常的鲁棒，很多情况下即使初始值距离(局部)最优解非常远，仍然可以保证求解成功。作为一种阻尼最小二乘解法，LMA(Levenberg-Marquart Algorithm)的收敛速度要稍微低于GNA(Guass-Newton Algorithm)。L-M算法作为求解非线性最小二乘问题最流行的算法广泛被各类软件包实现，例如google用于求解优化问题的库 Ceres Solver。后续，我会通过最小二乘圆拟合的案例给出L-M算法的实现细节。

参考资料

1.maximum likelihood regression-university of manitoba

2.过拟合与欠拟合-网易公开课：斯坦福大学机器学习课程

3.Using Gradient Descent for Optimization and Learning

4.Numerical Optimization using the Levenberg-Marquardt Algorithm-Los Alamos National Laboratory

5.Circular and Linear Regression Fitting Circles and Lines by Least Squares – Nikolai Chernov – UAB