数值优化（Numerical Optimization）学习系列-最小二乘问题（Least-Squares）

下一步 于 2015-12-27 18:52:09 发布

概述

最小二乘问题在实际应用中非常广泛，也是无约束最优化问题的重要应用之一，但是对于该问题还有一些特殊的求解思路，供参考。该小结主要介绍：

> 1. 问题定义
> 2. 线性最小二乘问题以及求解
> 3. 非线性最小二乘问题以及求解
>   总结

最小二乘问题

描述

最小二乘问题泛指具有如下形式的问题

$$minf(x) = \frac{1}{2}\sum_{j=1...m} r_j^2(x)$$

其中m一般指实例的个数，r_j指残差，即目标值和预估值的差，根据模型的不同有线性模型和非线性模型，分别线性最小二乘和非线性最小二乘。

向量表示

$r(x) = (r_1(x), r_2(x) \ldots r_m(x))^T$，则目标函数表示为$f(x) = \frac{1}{2}||r(x)||^2$
J(x)表示Jacobian矩阵，即r(x)对每一个参数的导数矩阵，即$J(x) = \frac{\partial r_j}{\partial x_i}$，$j = 1..m, i = 1..n$
则原始问题的一阶梯度和Hessian句分别为

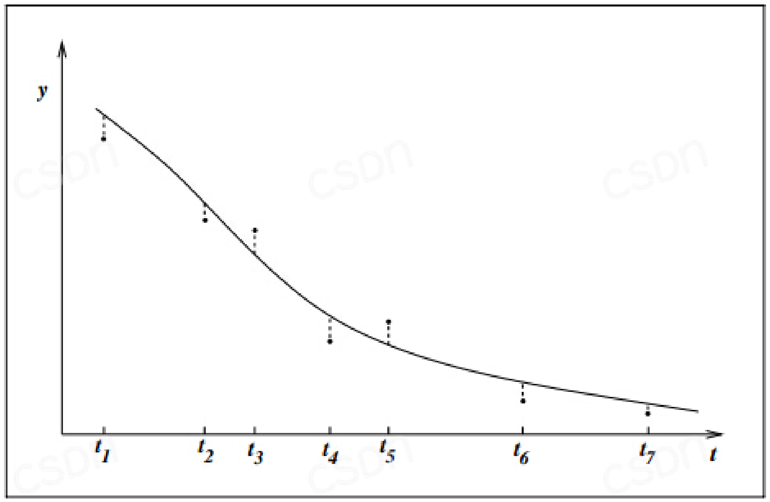
$$\nabla f(x) = \sum_{j=1...m} r_j(x)\nabla r_j(x) = J(x)^T r(x)$$

$$\nabla^2 f(x) = \sum_{j=1...m} \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1...m} r_j(x) \nabla^2 r_j(x)$$
$$= J(x)^T J(x) + \sum_{j=1...m} r_j(x) \nabla^2 r_j(x)$$

> 如果能够得到梯度和Hessian矩阵，则可以使用基于梯度的相关算法进行优化。该问题有一点好处是，如果知道了一阶梯度则可以近似得到Hessian矩阵，即忽略Hessian矩阵的第二项，因为对于某些问题二阶导数比较小

概率解释

对于某个实际问题，数据可能存在一定误差，需要寻找一条最优的曲线去拟合给定数据。即



则观察值$y_j = f(x_j) + \epsilon_j$，如果假设误差分布为正态分布N(0,$\sigma^2$)。
从概率论进行分析，需要使得模型最大程度拟合数据，释然值最大。
$maxN(f(x_j) - y_j, \sigma^2)$，一般会转换为log释然。推导可发现最大释然等价于 最小二乘法。

> 因此最小二乘问题就是误差满足正态分布的数据拟合问题。

线性最小二乘问题

采用线性模型去拟合数据，即y(x)=Jx，则目标函数为$f(x) = \frac{1}{2}||Jx - y||^2$，同时

$$\nabla f(x) = J^T(Jx - y),\ \nabla^2 f(x) = J^T J$$

根据最优化定理，该问题的最优解满足如下条件： $\nabla f(x)=0$，即 $J^TJx=J^Ty$
求解该方程，有如下方法
1. Cholesky 分解：当m>>n时，并且J比较稀疏时比较实用，但是其精度和条件数的平方相关
2. QR分解：精度和条件数相关
3. SVD分解：相对比较鲁棒，同时也可以添加正则项避免J是个病态矩阵。
4. 数值迭代算法：对于大规模数据比较实用。

非线性最小二乘问题

采用非线性模型拟合数据，模型本身可能梯度或者Hessian矩阵不容易得到。

高斯牛顿方法（Gauss-Newton方法）

根据标准牛顿方程，$\nabla^2 f(x_k)p=-\nabla f(x_k)$可以得到该问题的需要满足的条件，即

$$J^TJP^{GN}=-J^Tr_k$$

此时Hessian矩阵用第一项代替，即 $\nabla^2(fx)\approx J^TJ$
得到搜索方向后，可以使用线搜索的方法得到步长，重复进行可以得到最优解。

> 1. 很多情况下，非线性问题的Hessian矩阵，可以由第一项主导，即（J^TJ）
> 2. 当J满秩并且$\nabla f(x)$非零时，搜索方向p^{GN}必定是一个下降方向。因为$(p^{GN})^T\nabla f(x_k)=(p^{GN})^TJ^Tr_k=-(p^{GN})^TJ^TJ(p^{GN})\leq=0$
> 3. 对比该牛顿方程和线性最小二乘问题，会发现有类似的结构，即该搜索方向是$min\frac{1}{2}||Jp+r_k||^2$的最优解，可以应用线性相关算法进行求解。即在某次迭代过程中，残差由线性模型近似。$r(x_k+p)\approx r_k+J_kp$
> 4. 根据之前的理论可以知道该算法能够保证收敛，同时收敛速度为二次收敛。

Levenberg-Marquardt 方法

类似于高斯牛顿方法，Hessian矩阵通过J^TJ近似，但是将线搜索替换为信赖域方法。
该问题子问题为 $min\frac{1}{2}||Jp+r_k||^2\ \ ||p||\leq\Delta_k$

和高斯牛顿方法有同样的问题是，可能会有局部最优解

大规模最小二乘问题

以上算法当问题规模比较大时，收敛速度回接近线性，不如采用牛顿或者拟牛顿方法效率高（超线性）。
如果问题规模不好确定时，可以采用混合方法进行求解。

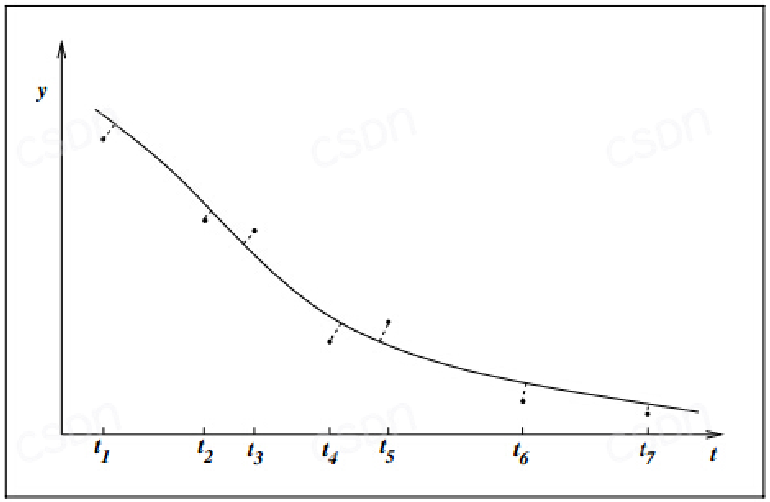
1. 混合思路一：根据搜索方向带来的函数值的减小量决定采用哪种算法。
2. 混合思路二：将问题建模为$B_k = J^TJ + S_k$，其中S_k为Hessian矩阵的第二项。

正交距离回归

该类算法解决的是，以上问题仅仅考虑了纵坐标轴上误差，没有考虑横向坐标也可能有误差，如果将其也考虑进来，进行建模，可以得到。
$y_j = \phi(x; t_j + \delta_j) + \epsilon_j$，其中$\delta_j$表示横向误差。
同时定义最优化问题

$$min \frac{1}{2} \sum_{j=1...m} (w_j^2 \epsilon_j^2 + d_j^2 \delta_j^2)$$

其中权重w和d可以根据模型人工确定或者其他方法得到。
上述最短距离可以看做点(t_j, y_j)到曲线$\phi(x;t)$之间的最短距离，即正交距离，图示如下



可以将该问题转换为无约束最小二乘问题，思路为
$minF(x, \delta) = \frac{1}{2} \sum_{j=1...m} [w_j^2 (y_j - \phi(x; t_j + \delta_j))^2] + d_j^2 \delta_j^2)$
$= \frac{1}{2} \sum_{j=1...2m} r_j^2(x, \delta)$
其中

$$r_j(x, \delta) = \begin{cases} w_j(y_j - \phi(x; t_j + \delta_j)), & \text{j=1,2...m} \\ d_{j-m}\delta_{j-m}, & \text{j=m+1,...,2m} \end{cases}$$

问题变为具有m+n个未知变量的最小二乘问题，可以通过相关算法进行求解。同时该问题的雅克比矩阵是一个非常稀疏的矩阵，可以利用稀疏性进行算法优化。

总结

通过该小结的学习，可以了解到

1. 最小二乘问题的一般形式以及概率解释
2. 线性和非线性最小二乘法求解算法
3. 大规模最小二乘问题如何求解