Jacobian矩阵和Hessian矩阵

jacoxu 机器学习算法那些事 2019-03-20

> 作者：Jacobian
>
> 链接：
>
> http://jacoxu.com/jacobian%E7%9F%A9%E9%98%B5%E5%92%8Chessian%E7%9F%A9%E9%98%B5/
>
> 编辑：石头

前言

还记得被Jacobian矩阵和Hessian矩阵统治的恐惧吗？本文清晰易懂的介绍了Jacobian矩阵和Hessian矩阵的概念，并循序渐进的推导了牛顿法的最优化算法。希望看过此文后，你对这两类矩阵有一个更深刻的理解。

在向量分析中，雅可比矩阵是一阶偏导数以一定方式排列成的矩阵，其行列式称为雅可比行列式. 还有， 在代数几何中，代数曲线的雅可比量表示雅可比簇：伴随该曲线的一个代数群，曲线可以嵌入其中。它们全部都以数学家卡尔·雅可比(Carl Jacob, 1804年10月4日 - 1851年2月18日)命名；英文雅可比量”Jacobian”可以发音为[ja ˈko bi ən]或者[ʤə ˈko bi ən]。

雅可比矩阵

雅可比矩阵的重要性在于它体现了一个可微方程与给出点的最优线性逼近. 因此, 雅可比矩阵类似于多元函数的导数 。

假设 $F : R_n \to R_m$ 是一个从欧式n维空间转换到欧式m维空间的函数。这个函数由m个实函数组成： y1(x1,...xn),...,ym(x1,...,xn)。这些函数的偏导数(如果存在)可以组成一个m行n列的矩阵, 这就是所谓的雅可比矩阵：

$$\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

此矩阵表示为： $J_F(x_1,...,x_n)$ ，或者为 $\frac{\partial(y_1,...,y_m)}{\partial(x_1,...,x_n)}$ 。

这个矩阵的第i行是由梯度函数的转置yi(i=1,…,m)表示的。

如果p是Rn中的一点，F在p点可微分，那么在这一点的导数由$J_F(p)$给出(这是求该点导数最简便的方法). 在此情况下，由F(p)描述的线性算子即接近点p的F的最优线性逼近，x逼近于p:

$$F(\mathbf{x}) \approx F(\mathbf{p}) + J_F(\mathbf{p}) \cdot (\mathbf{x} - \mathbf{p})$$

雅可比行列式

如果m=n，那么F是从n维空间到n维空间的函数，且它的雅可比矩阵是一个方块矩阵。于是我们可以取它的行列式，称为雅可比行列式。

在某个给定点的雅可比行列式提供了在接近该点时的表现的重要信息。 例如， 如果连续可微函数F在p点的雅可比行列式不是零，那么它在该点附近具有反函数。 这称为反函数定理。 更进一步， 如果p点的雅可比行列式是正数，则F在p点的取向不变；如果是负数，则F的取向相反。而从雅可比行列式的绝对值，就可以知道函数F在p点的缩放因子；这就是为什么它出现在换元积分法中。

对于取向问题可以这么理解，例如一个物体在平面上匀速运动，如果施加一个正方向的力F，即取向相同，则加速运动， 类比于速度的导数加速度为正；如果施加一个反方向的力F，即取向相反，则减速运动，类比于速度的导数加速度为负。

2. 海森Hessian矩阵

在数学中，海森矩阵(Hessian matrix或Hessian)是一个自变量为向量的实值函数的二阶偏导数组成的方块矩阵，此函数如下：

$$f(x_1, x_2 \ldots, x_n)$$

如果f的所有二阶导数都存在，那么f的海森矩阵即：

$$H(f)_{ij}(x) = D_i D_j f(x)$$

其中x=(x1,x2,...,xn)，即H(f)为：

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

（也有人把海森定义为以上矩阵的行列式）海森矩阵被应用于牛顿法解决的大规模优化问题。

海森矩阵在牛顿法中的应用

一般来说，牛顿法主要应用在两个方面，1，求方程的根；2，最优化。

1) 求解方程

并不是所有的方程都有求根公式，或者求根公式很复杂，导致求解困难。利用牛顿法，可以迭代求解。

原理是利用泰勒公式， 在x0处展开，且展开到一阶，即：

$$f(x) = f(x_0) + (x-x_0)f'(x_0)$$

求解方程f(x)=0，即

$$f(x_0)+(x-x_0)f'(x_0)=0$$

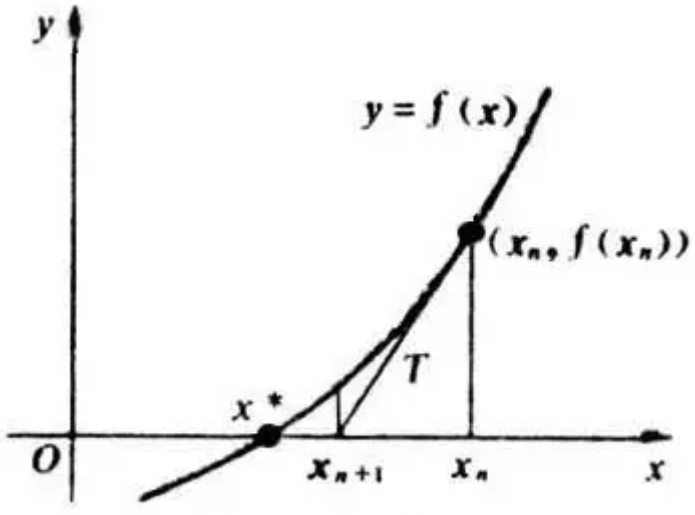
上式求解得：

$$x = x_1 = x_0-f(x_0)/f'(x_0)$$

因为这是利用泰勒公式的一阶展开，$f(x) = f(x_0) + (x-x_0)f'(x_0)$处并不是完全相等，而是近似相等，这里求得x1并能让f(x)=0，只能说$f(x_1)$的值比$f(x_0)$更接近f(x)=0，于是乎，迭代求解的想法就很自然了，可以进而推出：

$$x_{n+1} = x_n-f(x_n)/f'(x_n)$$

通过迭代，这个式子必然在$f(x^*)=0$的时候收敛，整个过程如下图：



牛顿法求实根图示

2), 最优化

在最优化的问题中，线性最优化至少可以使用单纯形法(或称不动点算法)求解，但对于非线性优化问题，牛顿法提供了一种求解的办法。假设任务是优化一个目标函数f，求函数f的极大极小问题，可以转化为求解函数f的导数$f^{'}=0$的问题，这样求可以把优化问题看成方程求解问题（$f^{'}=0$）。剩下的问题就和第一部分提到的牛顿法求解很相似了。

这次为了求解$f'=0$的根，首先把f(x)在探索点x_n处泰勒展开，展开到2阶形式进行近似：

$$f(x)=f(x_n)+f'(x_n)(x-x_n)+\frac{f''(x_n)}{2}(x-x_n)^2$$

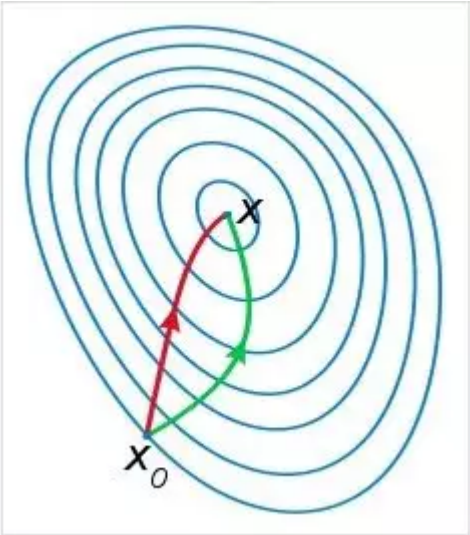
然后用f(x)的最小点做为新的探索点x_{n+1}，据此，令：

$$f'(x)=f'(x_n)+f''(x_n)(x-x_n)=0$$

求得出迭代公式：

$$x_{n+1}=x_n-\frac{f'(x_n)}{f''(x_n)},n=0,1,\ldots$$

一般认为牛顿法可以利用到曲线本身的信息，比梯度下降法更容易收敛（迭代更少次数），如下图是一个最小化一个目标方程的例子，红色曲线是利用牛顿法迭代求解，绿色曲线是利用梯度下降法求解.



在上面讨论的是2维情况，高维情况的牛顿迭代公式是：

$$x_{n+1}=x_n-[Hf(x_n)]^{-1}\nabla f(x_n),n\geq 0$$

其中H是hessian矩阵，定义见上.

高维情况依然可以用牛顿迭代求解，但是问题是Hessian矩阵引入的复杂性，使得牛顿迭代求解的难度大大增加，但是已经有了解决这个问题的办法就是Quasi-Newton method，不再直接计算hessian矩阵，而是每一步的时候使用梯度向量更新hessian矩阵的近似。

参考

1.Hessian参考：Wikipedia

2. Newton优化参考：运筹学第三版 刁在筠著

推荐阅读

常见的几种优化算法

主成分分析原理总结