梯度下降法(gradient descent)与牛顿法(newton’s method)求解最小值

发表于**2018年7月29日**

梯度下降法与牛顿法是求解最小值/优化问题的两种经典算法。本文的目标是介绍两种算法的推导思路与流程，并且从初学者的角度就一些容易混淆的话题如 梯度下降法(gradient descent)与最速下降法(steepest descent)的联系与区别、牛顿求根迭代方法(Newton–Raphson method) 与牛顿法求解最小值算法的联系(来自 Andrew Ng 机器学习课程第四讲)进行说明。本文的内容将对高斯牛顿法(Gauss–Newton algorithm) ,Levenberg-Marquardt算法(LM算法)等非线性最小二乘问题解法起到引出作用。

1.梯度下降法

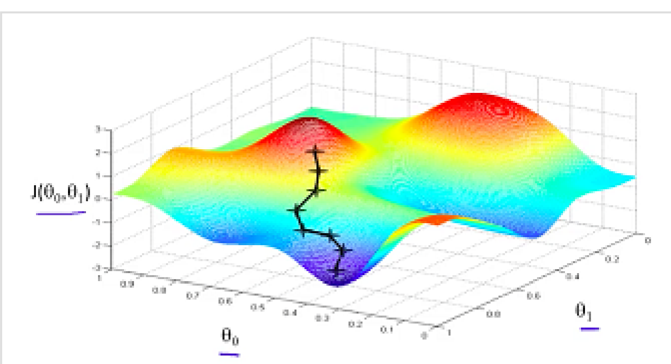
已知多元函数 $f(x_1,x_2,\ldots,x_n)$ 在定义域上可微，如果将$f(\mathbf{x})$在$\mathbf{x}$处一阶泰勒展开(tayler expansion),可得到：(说明：为了编辑方便下文中统一以 $x=[x_1,x_2,\ldots,x_n]^T$ 代替 $\mathbf{x}$)。

$$f(x+\epsilon)=f(x)+\epsilon^T\nabla_x f+O(||\epsilon||)\approx f(x)+\epsilon^T\nabla_x f$$

其中$\nabla_x f=\left[\frac{\partial f}{\partial x_1},\ldots,\frac{\partial f}{\partial x_n}\right]^T$为 f在x处的梯度向量。

这个式子我们可以解读为当 x增加 ϵ时，$f(x)$增加$\epsilon^T\nabla_x f$，即ϵ与梯度$\nabla_x f$的内积。如果我们限定ϵ的模长为定值，其方向怎样才能获得$f(x+\epsilon)$的最小值呢？答案当然是与$\nabla_x f$方向相反的时候，此时$\epsilon^T\nabla_x f$获得最小值。

我们也可以利用直觉上较好理解的爬山的例子来解释梯度下降法。假设你位于山上某一点坐标为$\theta(\theta_1,\theta_2)$，那么在此处(注意，是在这一点)下山最快的方向当然是沿着此处的梯度方向。

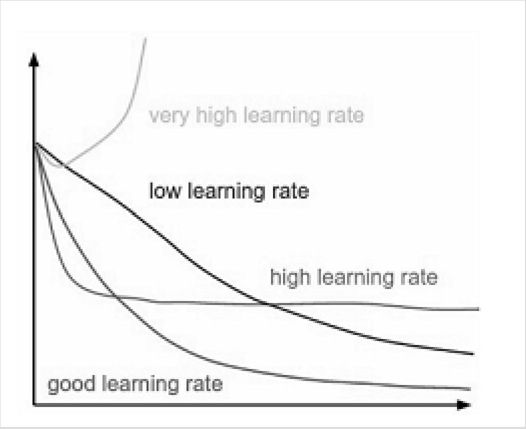


所以说，将整个故事串起来，梯度下降法的思路可以总结如下：欲求多元函数$f(x)$ 的最小值，可以采用如下步骤：

1. 给定初始值x_0。
2. 按照如下方式“下山”：$x_{i+1} = x_i - \eta \nabla_{xi} f$ 。其中$\eta > 0$，在机器学习领域，η也被称之为学习率(learning rate)。
3. 直到x 满足收敛条件为止。如$||f(x_{i+1}) - f(x_i)|| < \epsilon$或$||\nabla_{xi} f|| \approx 0$。

学习率的重要性：

学习率作为控制下降步长的参数，影响函数下降的速度。学习率是我们根据经验确定的一个参数，因此在机器学习领域中这样的参数也被成为超参数(hyperparameter)。学习率的选取不能过大或者过小，如下图，不同的学习率导致函数不同的收敛速度，甚至可能导致函数不收敛。



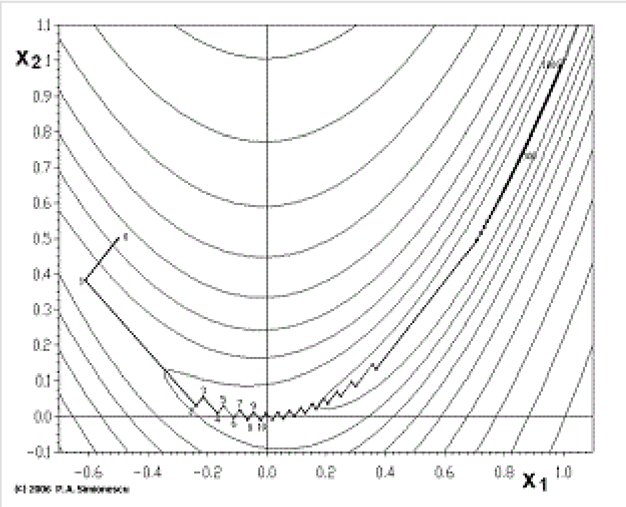
1.1 梯度下降法的优势

1.时间复杂度低，在每一个迭代中，只需要计算梯度，不需要对二阶导数矩阵（即海森矩阵(Hessian Matrix)）进行计算。

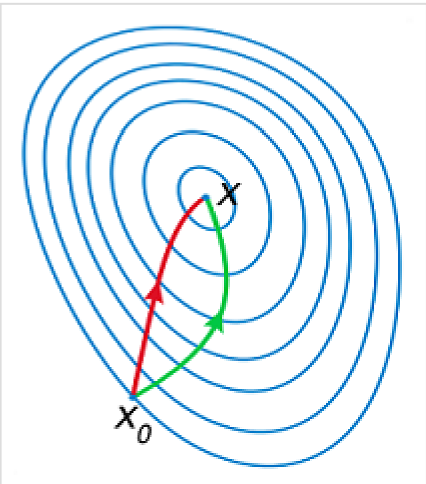
2.空间复杂度低，因为梯度向量为一个$n \times 1$的向量，比起Hessian Matrix来，占用存储空间小n倍。在实际应用中，$\mathbf{x}$的维度可能非常高。

1.2梯度下降法的局限

对于部分求解函数，梯度下降法可能会出现下降非常缓慢的情形。其收敛速度也较其他方法低（其他文献分析其收敛速度为线性，本文不作推导）。如下图，梯度下降法的路径出现了z字型。



究其原因，我认为，某一点的梯度只能作为这一点的一个极小的领域处的最快下降方向，一旦梯度变化较快，梯度下降法会出现因为学习率不合适而出现”zigzag”现象。而且，如果我们将梯度下降法与下文的牛顿法做对比，你会发现，一直沿着梯度方向下降的速度不一定是最快的。如下图：



1.3 最速下降法(steepest decent) 与 梯度下降法(gradient descent)的联系

总结一下就是 梯度下降法是最速下降法的一种特例。在最速下降法中，对于某一范数下$epsilon$的取值根据以下原则：

$$\triangle \epsilon_{nsd} = argmin_v(\nabla f(x)^T \epsilon \mid \|\epsilon\| = 1)$$

当我们指定的范数为欧几里得范数时，最速下降法给出的下降方向就是梯度的负方向，即梯度下降法给出的方向。

在wikipedia中说明，梯度下降法也被称为最速下降法(Gradient descent is also known as steepest descent)。

2.牛顿法

如同根据一阶泰勒展开推导出梯度下降法一样，根据二阶泰勒展开可以推导出牛顿优化法(newton’s method in optimization)。将$f(\mathbf{x})$在$\mathbf{x}$处一阶泰勒展开(tayler expansion),可得到：

$$f(x+\epsilon)=f(x)+\epsilon^T\nabla_x f+\frac{1}{2}\epsilon^T H\epsilon+O(||\epsilon||^2)\approx f(x)+\epsilon^T\nabla_x f+\frac{1}{2}\epsilon^T H\epsilon$$

如果我们将x看做固定的已知量，将f看做关于ϵ的函数，那么欲求$f(\epsilon|x)$的最小值，必要条件(注意：不是冲要条件)是$\frac{\partial f}{\partial \epsilon}=0$其中

$$H=\begin{bmatrix}\frac{\partial^2 f}{\partial x1\partial x1} & \cdots & \frac{\partial^2 f}{\partial x1\partial xn}\\ \vdots & \ddots & \vdots\\ \frac{\partial^2 f}{\partial x_n\partial x1} & \cdots & \frac{\partial^2 f}{\partial x_n\partial x_n}\end{bmatrix}$$

称之为f的$Hessian$矩阵。因为二阶连续混合编导数具备性质

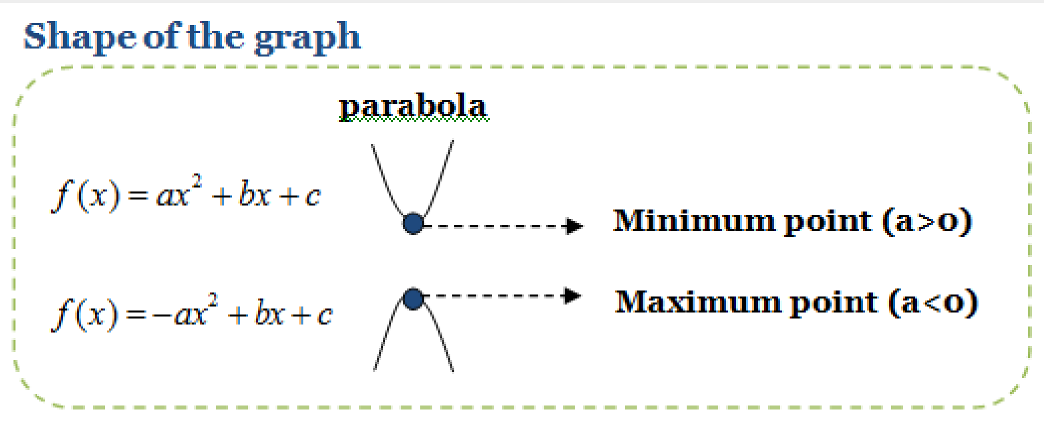
$$\frac{\partial^2 f}{\partial x_i\partial x_j}=\frac{\partial^2 f}{\partial x_j\partial x_i}$$

因此$Hessian$矩阵为对称矩阵。根据矩阵求导法则，可以得到

$$\frac{\partial f}{\partial \epsilon}=\nabla_x f^T+\epsilon^T H=0$$

$$\epsilon=-H^{-1}\nabla_x f$$

可见，牛顿法的思路是将函数f在x处展开为多元二次函数，再通过求解二次函数最小值的方法得到本次迭代的下降方向ϵ。那么问题来了，多元二次函数在梯度为0的地方一定存在最小值么？直觉告诉我们是不一定的。以 一元二次函数 $g(x)=ax^2+bx+c$ 为例，我们知道当$a>0$时，$g(x)$可以取得最小值，否则$g(x)$不存在最小值。



— https://perfectmaths.wordpress.com/2011/08/20/chapter-3-%E2%80%93-quadratic-functions/

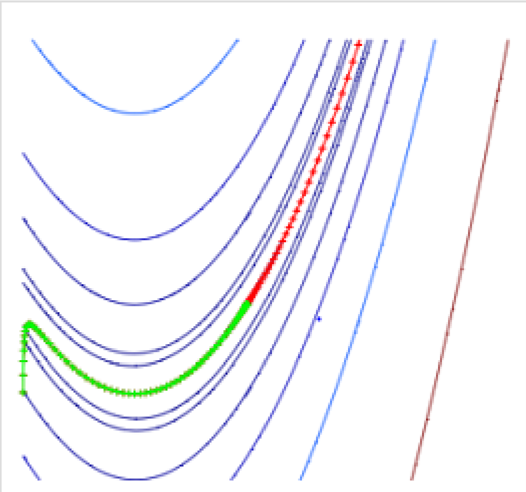
推广到多元的情况，可以得出二次项矩阵必须是正定(positive definite)的，对应上式即$Hessian$为正定矩阵时，函数$f(\epsilon|x)$的最小值才存在。

因此，牛顿法首先需要计算$Hessian$矩阵并且判断其正定性，当$Hessian$矩阵正定，此时其所有特征值均>0,当然$Hessian$矩阵也是可逆的，最小值存在。

需要指出的是，当多元函数 f 本身就是二次函数并且存在最小值时，牛顿法可以一步解出最小值。

2.1 牛顿法的优点：

因为目标函数在接近极小值点附近接近二次函数，因此在极小值点附近，牛顿法的收敛速度较梯度下降法快的多。其他文献分析其收敛速度为2次收敛，本文不给出推导。下图是牛顿法应用在Rosenbrock函数上的效果：



2.2 牛顿法的缺点：

1.$Hessian$矩阵的计算难度非常的大。因此在高维度应用案例中，通常不会计算$Hessian$矩阵。因此牛顿法也产生了很多变种，主要的思想就是采用其他矩阵近似$Hessian$矩阵，降低计算复杂度。

2.牛顿法当$Hessian$矩阵为正定矩阵时，最小值才存在。牛顿法经常会因为$Hessian$矩阵不正定而发散(diverage)。因此 牛顿法并不是非常的稳定。

2.3 牛顿法求根公式与牛顿优化法之间的联系

在说道牛顿优化方法的时候，上过《计算方法》这门课的同学经常会说，牛顿法不是用来求根的么？实际上，牛顿优化法还真可以用牛顿求根法推导得出。我看到的材料是 Andrew Ng在《机器学习课程》中给出的一种推导。在牛顿求根公式中，$f(x)=0$的解由迭代式

$$x_{i+1}=x_i-\frac{f(x_i)}{f\prime(x_i)}$$

给出。在牛顿优化法中，我们欲求得梯度$g(x)=f'(x)=0$对应的\(x)。

因此 x可以根据求根公式

$$x_{i+1}=x_i-\frac{f\prime(x_i)}{f\prime\prime(x_i)}$$

求出。推广到多元函数上，$1/f\prime\prime(x_i)$演变为H^{-1}，$f\prime(x_i)$演变为$\nabla_x f(x_i)$因此

$$x_{i+1}=x_i-H^{-1}\nabla_x f(x_i)$$

与根据二阶泰勒展开并求$f(\epsilon|x)$的最小值得到的结论一致。

3.参考文献：

1.gradient descent in a nutshell – towardsdatascience.com

2.Newton's method in Optimization-wikipedia

3.Gradient Descent Method – Rochester Institute of Technology

4.Using Gradient Descent in Optimization and Learning – University Collage London

5.Difference between Gradient Descent method and Steepest Descent – stack exchange

6.In optimization, why is Newton's method much faster than gradient descent?