知乎 切换

$$
\begin{aligned}
&r_0 = \mathbf{D}^{(0)}\mathbf{q}_0 \\
&for\ i = 0, \dots, L-1 \\
&\{ \quad j = i + \delta\ ; \\
&\quad\ \ \beta_j = \rho_j \mathbf{y}_j^T \mathbf{r}_i\ ; \\
&\quad\ \ \mathbf{r}_{i+1} = \mathbf{r}_i + (\alpha_i - \beta_i)\mathbf{s}_j\ ; \\
&\}
\end{aligned}
$$

牛顿法

大咸鱼

9 人赞同了该文章

牛顿法被称为牛顿-拉夫逊（Newton-Raphson）方法。牛顿在17世纪提出用来求解方程的根。

假设点x*位函数f(x)的根，则f(x*)=0。

将函数f(x)在点

$$x^{(t)}$$

处进行一阶泰勒展开有：

$$f(x) \approx f(x^{(t)}) + (x - x^{(t)}) f'(x^{(t)})$$

假设点

$$x^{(t+1)}$$

为函数f(x)的根，则有：

▲ 赞同 9 ▼ 添加评论 分享 喜欢 收藏 申请转载 ···

知乎 切换

那么可以得到：

$$x^{(t+1)} = x^{(t)} - \frac{f(x^{(t)})}{f'(x^{(t)})}$$

牛顿法通过迭代的方式求解方程f(x)=0的解。

牛顿法求解目标函数极值

对于最优化问题，极值点处函数的一阶导数为0

可以对一阶导数

$$g(x) = f'(x)$$

利用牛顿法通过迭代的方式来求得最优解，即相当于求一阶导数对应函数的根。

$$x^{(t+1)} = x^{(t)} - \frac{f(x^{(t)})}{f'(x^{(t)})} \xRightarrow{g(x)=f'(x)} x^{(t+1)} = x^{(t)} - \frac{g(x^{(t)})}{g'(x^{(t)})} = x^{(t)} - \frac{f'(x^{(t)})}{f''(x^{(t)})}$$

牛顿法是二阶最优化算法。

对多元函数

$$f(x_1, \dots, x_D)$$

，一阶导数换成梯度：

$$\nabla f(x_1, \dots, x_D)$$

，二阶导数换成海森（Hessian）矩阵H，

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial^2 x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_D} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial^2 x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_D} & \frac{\partial^2 f}{\partial x_D \partial x_2} & \cdots & \frac{\partial^2 f}{\partial^2 x_D^2} \end{bmatrix}$$

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \mathbf{H}^{-1}\left(\mathbf{x}^{(t)}\right)\nabla f\left(\mathbf{x}^{(t)}\right)$$

牛顿法求解目标函数极值步骤：

1、从t=0开始，初始化

$$\mathbf{x}^{(0)}$$

为随机值；

2、计算目标函数f(x)在点

$$\mathbf{x}^{(t)}$$

的梯度

$$\mathbf{g}^{(t)} = \nabla f\left(\mathbf{x}^{(t)}\right)$$

和海森矩阵

$$\mathbf{H}^{(t)} = \mathbf{H}\left(\mathbf{x}^{(t)}\right)$$

3、计算移动方向：

$$\boldsymbol{d}^{(t)} = (\mathbf{H}^{(t)})^{-1}\mathbf{g}^{(t)}$$

（一般用线性方程组计算

$$\boldsymbol{d}^{(t)} : \mathbf{H}^{(t)}\boldsymbol{d}^{(t)} = \mathbf{g}^{(t)}$$

。线性方程组求解可用共轭梯度等方法求解）。

4、根据迭代公式，更新x的值：

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \boldsymbol{d}^{(t)}$$

$$\mathbf{x}^{(t+1)}$$

和目标函数最小值

$$f\left(\mathbf{x}^{(t+1)}\right)$$

否则转到第2步。

与一阶梯度法

$$d^{(t)} = -\eta \mathbf{g}^{(t)}$$

移动方向为：

$$d^{(t)} = -(\mathbf{H}^{(t)})^{-1}\mathbf{g}^{(t)}$$

拟牛顿法

牛顿法比一般的梯度下降法收敛速度快。

但在高维情况下，计算目标函数的二阶偏导数的复杂度大，而且有时候目标函数的海森矩阵无法保持正定，不存在逆矩阵，此时牛顿法将不再能使用

因此，人们提出了拟牛顿法（Quasi-Newton Methods）：不用二阶偏导数构造出可以近似Hessian矩阵（或Hessian矩阵的逆矩阵）的正定对称矩阵，进而再逐步优化目标函数。

不同的Hessian矩阵构造方法产生了不同的拟牛顿法：

BFGS/L-BFGS

拟牛顿条件

在t次迭代后，得到

$$\mathbf{x}^{(t+1)}$$

将目标函数f(x)在

$$\mathbf{x}^{(t+1)}$$

$$f(\mathbf{x}) \approx f\left(\mathbf{x}^{(t+1)}\right) + \left(\mathbf{x}-\mathbf{x}^{(t+1)}\right)\nabla f\left(\mathbf{x}^{(t+1)}\right) + \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^{(t+1)}\right)^T \nabla^2 f\left(\mathbf{x}^{(t+1)}\right)\left(\mathbf{x}-\mathbf{x}^{(t+1)}\right)$$

两边同时取梯度运算▿，得到

$$\nabla f(\mathbf{x}) \approx \nabla f\left(\mathbf{x}^{(t+1)}\right) + \nabla^2 f\left(\mathbf{x}^{(t+1)}\right)\left(\mathbf{x}-\mathbf{x}^{(t+1)}\right)$$

取

$$\mathbf{x} = \mathbf{x}^{(t)}$$

，令

$$\mathbf{g}^{(t)} = \nabla f\left(\mathbf{x}^{(t)}\right)$$

知乎 @大咸鱼

，

$$\mathbf{H}^{(t)} = \nabla^2 f\left(\mathbf{x}^{(t)}\right)$$

知乎 @大咸鱼

，则

$$\mathbf{g}^{(t+1)} - \mathbf{g}^{(t)} \approx \mathbf{H}^{(t+1)}\left(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\right)$$

引入记号

$$\mathbf{s}^{(t)} = \mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}$$

，

$$\mathbf{y}^{(t)} = \mathbf{g}^{(t+1)} - \mathbf{g}^{(t)}$$

，则

$$\mathbf{y}^{(t)} \approx \mathbf{H}^{(t+1)}\mathbf{s}^{(t)}$$

知乎 @大咸鱼

令B表示H的近似，D表示

知乎 切换

$$\mathbf{H}^{-1}$$

的近似，根据

$$\mathbf{y}^{(t)} \approx \mathbf{H}^{(t+1)}\mathbf{s}^{(t)}$$

得到拟牛顿条件为：

$$\mathbf{y}^{(t)} = \mathbf{B}^{(t+1)}\mathbf{s}^{(t)}$$

或：

$$\mathbf{s}^{(t)} = \mathbf{D}^{(t+1)}\mathbf{y}^{(t)}$$

BFGS

BFGS算法是Broyden,Fletcher,Goldfarb,Shanno四位研究者发明出来的，被认为是数值效果最好的拟牛顿法，并且具有全局收敛性和超线性收敛速度。

BFGS算法使用迭代法逼近Hessian矩阵：

$$\mathbf{B}^{(t+1)} = \mathbf{B}^{(t)} + \Delta\mathbf{B}^{(t)}$$

初始值

$$\mathbf{B}^{(0)} = \mathbf{I}$$

为单位矩阵，因此关键是如何构造

$$\Delta\mathbf{B}^{(t)}$$

为了保证矩阵B的正定性，令

$$\Delta\mathbf{B}^{(t)} = \alpha\mathbf{u}\mathbf{u}^T + \beta\mathbf{v}\mathbf{v}^T$$

，代入

$$\mathbf{y}^{(t)} = \mathbf{B}^{(t+1)}\mathbf{s}^{(t)} = (\mathbf{B}^{(t)} + \Delta\mathbf{B}^{(t)})\mathbf{s}^{(t)} = \mathbf{B}^{(t)}\mathbf{s}^{(t)} + \Delta\mathbf{B}^{(t)}\mathbf{s}^{(t)}$$

$$= \mathbf{B}^{(t)}\mathbf{s}^{(t)} + \alpha\mathbf{u}\mathbf{u}^T\mathbf{s}^{(t)} + \beta\mathbf{v}\mathbf{v}^T\mathbf{s}^{(t)}$$
$$= \mathbf{B}^{(t)}\mathbf{s}^{(t)} + \mathbf{u}(\alpha\mathbf{u}^T\mathbf{s}^{(t)}) + \mathbf{v}(\beta\mathbf{v}^T\mathbf{s}^{(t)})$$

$$\mathbf{y}^{(t)} = \mathbf{B}^{(t)}\mathbf{s}^{(t)} + \mathbf{u}(\alpha\mathbf{u}^T\mathbf{s}^{(t)}) + \mathbf{v}(\beta\mathbf{v}^T\mathbf{s}^{(t)})$$

令

$$\alpha\mathbf{u}^T\mathbf{s}^{(t)} = 1\ ,\ \beta\mathbf{v}^T\mathbf{s}^{(t)} = -1$$

，得到：

$$\alpha = \frac{1}{\mathbf{u}^T\mathbf{s}^{(t)}}\ ,\ \beta = -\frac{1}{\mathbf{v}^T\mathbf{s}^{(t)}}$$

将

$$\alpha\mathbf{u}^T\mathbf{s}^{(t)} = 1\ ,\ \beta\mathbf{v}^T\mathbf{s}^{(t)} = -1$$

代入

$$\mathbf{y}^{(t)} = \mathbf{B}^{(t)}\mathbf{s}^{(t)} + \mathbf{u}(\alpha\mathbf{u}^T\mathbf{s}^{(t)}) + \mathbf{v}(\beta\mathbf{v}^T\mathbf{s}^{(t)}) = \mathbf{B}^{(t)}\mathbf{s}^{(t)} + \mathbf{u} - \mathbf{v}$$

得到：

$$\mathbf{u} - \mathbf{v} = \mathbf{y}^{(t)} - \mathbf{B}^{(t)}\mathbf{s}^{(t)}$$

不防令

$$\mathbf{u} = \mathbf{y}^{(t)}\ ,\ \mathbf{v} = \mathbf{B}^{(t)}\mathbf{s}^{(t)}$$

，代入

$$\alpha = \frac{1}{\mathbf{u}^T\mathbf{s}^{(t)}} = \frac{1}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}\ ,$$

$$\beta = -\frac{1}{\mathbf{v}^T\mathbf{s}^{(t)}} = -\frac{1}{(\mathbf{B}^{(t)}\mathbf{s}^{(t)})^T\mathbf{s}^{(t)}} = -\frac{1}{(\mathbf{s}^{(t)})^T(\mathbf{B}^{(t)})^T\mathbf{s}^{(t)}}$$

知乎 @大咸鱼

$$= \frac{\mathbf{y}^{(t)}(\mathbf{y}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}} - \frac{\mathbf{B}^{(t)}\mathbf{s}^{(t)}(\mathbf{B}^{(t)}\mathbf{s}^{(t)})^T}{(\mathbf{s}^{(t)})^T(\mathbf{B}^{(t)})^T\mathbf{s}^{(t)}}$$

牛顿法中需要计算Hessian矩阵的逆矩阵。

根据Sherman-Morrison公式，可得到

$$(\mathbf{B}^{(t+1)})^{-1} = \mathbf{D}^{(t+1)} = \left(\mathbf{I} - \frac{\mathbf{s}^{(t)}(\mathbf{y}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}\right)\mathbf{D}^{(t)}\left(\mathbf{I} - \frac{\mathbf{y}^{(t)}(\mathbf{s}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}\right) + \frac{\mathbf{s}^{(t)}(\mathbf{s}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}$$

Sherman-Morrison公式：若A为非奇异方阵，

$$1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u} \neq 0$$

，则

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

BFGS更新参数的流程：

1、从t=0开始，初始化

$$\mathbf{D}^{(0)} = \mathbf{I}$$

2、计算移动方向：

$$\boldsymbol{d}^{(t)} = \mathbf{D}^{(t)}\mathbf{g}^{(t)}$$

3、更新x的值：

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \boldsymbol{d}^{(t)}$$

4、

$$\mathbf{s}^{(t)} = \boldsymbol{d}^{(t)}$$

5、若

，迭代终止；

6、计算：

$$\mathbf{y}^{(t)}=\mathbf{g}^{(t+1)}-\mathbf{g}^{(t)}$$

7、t=t+1，转第2步。

L-BFGS

L-BFGS（limited memory BFGS）不直接存储Hessian矩阵，而是通过存储计算过程中产生的

$$\mathbf{s}^{(t)}$$

和

$$\mathbf{y}^{(t)}$$

来计算Hessian矩阵，从而减少参数存储所需空间。

BFGS中Hessian矩阵更新公式为：

$$\mathbf{D}^{(t+1)}=\left(\mathbf{I}-\frac{\mathbf{s}^{(t)}(\mathbf{y}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}\right)\mathbf{D}^{(t)}\left(\mathbf{I}-\frac{\mathbf{y}^{(t)}(\mathbf{s}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}\right)+\frac{\mathbf{s}^{(t)}(\mathbf{s}^{(t)})^T}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}$$

令

$$\rho^{(t)}=\frac{1}{(\mathbf{y}^{(t)})^T\mathbf{s}^{(t)}}$$

知乎 @大咸鱼

，

$$\mathbf{V}^{(t)} = \mathbf{I} - \rho^{(t)}\mathbf{y}^{(t)}(\mathbf{s}^{(t)})^T$$

则：

$$\mathbf{D}^{(t+1)} = (\mathbf{V}^{(t)})^T\mathbf{D}^{(t)}\mathbf{V}^{(t)} + \rho^{(t)}\mathbf{s}^{(t)}(\mathbf{s}^{(t)})^T$$

展开：

$$\mathbf{D}^{(1)} = (\mathbf{V}^{(0)})^T\mathbf{D}^{(0)}\mathbf{V}^{(0)} + \rho^{(0)}\mathbf{s}^{(0)}(\mathbf{s}^{(0)})^T$$

$$\begin{aligned}\mathbf{D}^{(2)} &= (\mathbf{V}^{(1)})^T\mathbf{D}^{(1)}\mathbf{V}^{(1)} + \rho^{(1)}\mathbf{s}^{(1)}(\mathbf{s}^{(1)})^T \\ &= (\mathbf{V}^{(1)})^T\left((\mathbf{V}^{(0)})^T\mathbf{D}^{(0)}\mathbf{V}^{(0)} + \rho^{(0)}\mathbf{s}^{(0)}(\mathbf{s}^{(0)})^T\right)\mathbf{V}^{(1)} + \rho^{(1)}\mathbf{s}^{(1)}(\mathbf{s}^{(1)})^T \\ &= (\mathbf{V}^{(1)})^T(\mathbf{V}^{(0)})^T\mathbf{D}^{(0)}\mathbf{V}^{(0)}\mathbf{V}^{(1)} + (\mathbf{V}^{(1)})^T\rho^{(0)}\mathbf{s}^{(0)}(\mathbf{s}^{(0)})^T\mathbf{V}^{(1)} + \rho^{(1)}\mathbf{s}^{(1)}(\mathbf{s}^{(1)})^T\end{aligned}$$

知乎 @大咸鱼

一般地：

$$\begin{aligned}\mathbf{D}^{(t+1)} = &\left(\mathbf{V}^{(t)}\right)'\left(\mathbf{V}^{(t-1)}\right)' \dots \left(\mathbf{V}^{(0)}\right)'\mathbf{D}^{(0)}\mathbf{V}^{(0)}\mathbf{V}^{(1)} \dots \mathbf{V}^{(t)} \\ &+\left(\mathbf{V}^{(t)}\right)^T\left(\mathbf{V}^{(t-1)}\right)^T \dots \left(\mathbf{V}^{(1)}\right)^T\rho^{(0)}\mathbf{s}^{(0)}(\mathbf{s}^{(0)})^T\mathbf{V}^{(1)}\mathbf{V}^{(2)} \dots \mathbf{V}^{(t)} \\ &+\left(\mathbf{V}^{(t)}\right)^T\left(\mathbf{V}^{(t-1)}\right)^T \dots \left(\mathbf{V}^{(2)}\right)^T\rho^{(1)}\mathbf{s}^{(1)}(\mathbf{s}^{(1)})^T\mathbf{V}^{(2)} \dots \mathbf{V}^{(t)} \\ &+\cdots \\ &+\left(\mathbf{V}^{(t)}\right)^T\rho^{(t-1)}\mathbf{s}^{(t-1)}(\mathbf{s}^{(t-1)})^T\mathbf{V}^{(t)} \\ &+\rho^{(t)}\mathbf{s}^{(t)}(\mathbf{s}^{(t)})^T\end{aligned}$$

知乎 @大咸鱼
https://blog.csdn.net/qq_40006456

计算将

$$\mathbf{D}^{(t+1)}$$

需要用到

$$\left\{\mathbf{s}^{(k)}, \mathbf{y}^{(k)}\right\}_{k=0}^{t}$$

如果只能存储m组

$$\left\{\mathbf{s}^{(k)}, \mathbf{y}^{(k)}\right\}$$

知乎

切换

$$\mathbf{D}^{(1)},\ldots,\mathbf{D}^{(m)}$$

要丢弃一部分

$$(\mathbf{s}^{(k)},\mathbf{y}^{(k)})$$

的话，丢弃较早生成的那些

$$(\mathbf{s}^{(k)},\mathbf{y}^{(k)})$$

则计算

$$\mathbf{D}^{(m+1)}$$

只存储了

$$\{\mathbf{s}^{(k)},\mathbf{y}^{(k)}\}_{k=1}^{m}$$

丢弃了

$$\{\mathbf{s}^{(k)},\mathbf{y}^{(k)}\}_{k=0}^{1}$$

由于丢弃了部分信息，只能近似计算

$$\mathbf{D}^{(m+1)}$$

当t>m+1时，构造近似公式：

$$\begin{aligned}\mathbf{D}^{(t+1)} = &\left(\mathbf{V}^{(t)}\right)^T\left(\mathbf{V}^{(t-1)}\right)^T\ldots\left(\mathbf{V}^{(t-m+1)}\right)^T\mathbf{D}^{(0)}\mathbf{V}^{(t-m+1)}\ldots\mathbf{V}^{(t)}\\ &+\left(\mathbf{V}^{(t)}\right)^T\left(\mathbf{V}^{(t-1)}\right)^T\ldots\left(\mathbf{V}^{(t-m+2)}\right)^T\rho^{(0)}\mathbf{s}^{(0)}(\mathbf{s}^{(0)})^T\mathbf{V}^{(1)}\mathbf{V}^{(1)}\ldots\mathbf{V}^{(t)}\\ &+\cdots\\ &+\left(\mathbf{V}^{(t)}\right)^T\rho^{(t-1)}\mathbf{s}^{(t-1)}(\mathbf{s}^{(t-1)})^T\mathbf{V}^{(t)}\end{aligned}$$

知乎 @大咸鱼

知乎 切换

$$\mathbf{D}^{(t)}$$

是为了得到搜索方向

$$d^{(t)} = \mathbf{D}^{(t)}\mathbf{g}^{(t)}$$

知乎 @大咸鱼

利用上面的公式，设计快速计算

$$\mathbf{D}^{(t)}\mathbf{g}^{(t)}$$

的方法

1、初始化：

$$\delta = \begin{cases} 0 & \text{if } t \le m \\ t-m & \text{otherwise} \end{cases}, \quad L = \begin{cases} t & \text{if } t \le m \\ m & \text{otherwise} \end{cases}$$

2、向后循环：

$$\begin{aligned} & for\ i = L-1, L-2, \dots, 1, 0 \\ & \{ \quad j = i + \delta\ ; \\ & \qquad \alpha_i = \rho_j \mathbf{s}_j^T \mathbf{q}_{i+1}\ ; \quad \text{//前向循环汇中还要用到} \\ & \qquad \mathbf{q}_i = \mathbf{q}_{i+1} - \alpha_i \mathbf{y}_j\ ; \\ & \} \end{aligned}$$

知乎 @大咸鱼

3、向前循环：

$$\begin{aligned} & r_0 = \mathbf{D}^{(0)}\mathbf{q}_0 \\ & for\ i = 0, \dots, L-1 \\ & \{ \quad j = i + \delta\ ; \\ & \qquad \beta_j = \rho_j \mathbf{y}_j^T \mathbf{r}_i\ ; \\ & \qquad \mathbf{r}_{i+1} = \mathbf{r}_i + (\alpha_i - \beta_i)\mathbf{s}_j\ ; \\ & \} \end{aligned}$$

知乎 @大咸鱼

4、

发布于 2022-05-27 12:18

牛顿法 计算机视觉 人工智能算法

写下你的评论...

还没有评论，发表第一个评论吧

推荐阅读

理解牛顿法

SIGAI

牛顿法和拟牛顿法

牛顿法（Newton method）和拟牛顿法（quasi Newton method）是求解无约束最优化问题的常用方法，有收敛速度快的优点。牛顿法是迭代算法，每一步都需求解目标函数的海塞矩阵（Hessian...

Pikac... 发表于机器学习中...

牛顿法和拟牛顿法

导言牛顿法和拟牛顿法也是求解无约束最优化问题的常用方法【另一常用方法为： 梯度下降法】，有收敛速度快的优点。牛顿法是迭代算法，每一步需要求解目标函数的Hessian 矩阵的逆矩阵，计算...

多鱼

牛顿法进阶

在之前的文章我们介绍过，种，目标函数是每次迭代牛表示为 abla p_k^N = - a

王金戈