机器学习笔记 (5): 牛顿法和拟牛顿法

https://blog.csdn.net/gaocui883

2021 年 4 月 8 日

1 牛顿法

牛顿法和拟牛顿法是求解无约束最优化问题的常用方法，收敛速度快。牛顿法需要求解黑塞矩阵的逆矩阵，计算比较复杂。

拟牛顿法通过正定矩阵近似黑塞矩阵的逆矩阵或者黑塞矩阵本身，简化了计算过程。

无约束优化：$\min_{x\in\mathbf{R}^n} f(x)$

二阶泰勒展开：

$f(x)=f\left(x^{(k)}\right)+g_k^{\mathrm{T}}\left(x-x^{(k)}\right)+\frac{1}{2}\left(x-x^{(k)}\right)^{\mathrm{T}}H\left(x^{(k)}\right)\left(x-x^{(k)}\right)$

其中：$g_k=g\left(x^{(k)}\right)=\nabla f\left(x^{(k)}\right)$

Hesse matrix :

$$H(x)=\left[\frac{\partial^2 f}{\partial x_i\partial x_j}\right]_{n\times n}$$

极小值的必要条件： $\nabla f(x)=0$

每次从 x^k 迭代，假设 x^{k+1} 满足：

$$\nabla f\left(x^{(k+1)}\right)=0$$

利用泰勒展开的结果：

$$\nabla f(x)=g_k+H_k\left(x-x^{(k)}\right)=0$$

$$x^{(k+1)}=x^{(k)}-H_k^{-1}g_k$$
$$x^{(k+1)}=x^{(k)}+p_k$$
$$H_kp_k=-g_k$$

Algorithm 1 牛顿法：

Input:

目标函数 $f(x)$, 梯度 $g(x) = \nabla f(x)$, Hesse 矩阵 $H(x)$, 精度要求 ϵ

Output:

$f(x)$ 的极小值点 x^*。

1: 取初始值 x^0, $k = 0$

2: 计算 $g_k = \nabla f(x^k)$

3: 如果 $||g_k|| < \epsilon$, 则停止计算，当前的 x^k 就是 x^*

4: 计算黑塞矩阵 $H(x)$, 求 p_k

$$H_k p_k = -g_k$$

5: $x^{k+1} = x^k + p_k$

6: $k = k + 1$ 重复到第二部，求 $g_k = \nabla f(x^k)$

用 x^k 与 x^{k+1} 不停的迭代，知道 g_k 逐渐真正的接近于 0.

牛顿法计算复杂之处在于求黑塞矩阵和黑塞矩阵的逆矩阵，而拟牛顿法的思路就是在这上边进行改进。

2 拟牛顿法

拟牛顿法的思路是用一个 n 阶矩阵 $G_k = G(x^k)$ 来近似相应的黑塞矩阵的逆矩阵 $H^{-1}(x^k)$; 或者用一个 n 阶矩阵 $B_k = B(x^k)$ 来近似黑塞矩阵本身 $H(x^k)$

$$\nabla f(x) = g_k + H_k \left(x - x^{(k)}\right) = 0$$

得到：

$$g_{(k+1)} = \nabla f(x+1) = g_k + H_k \left(x^{k+1} - x^{(k)}\right) = 0$$

$$g_{(k+1)} - g_k = H_k \left(x^{k+1} - x^{(k)}\right)$$

设：$y_k = g_{(k+1)} - g_k, \delta_k = \left(x^{k+1} - x^{(k)}\right)$

有：

$$y_k = H_k \delta_k$$

$$H_k^{-1} y_k = \delta_k$$

当 H_k 为正定的，那么可以保证 $f(x)$ 是下降方向的，也就是会不停的减少的。

寻找一个 G_k 满足：$G_{k+1}y_k = \delta_k$ 并且正定即可。

2.1 DFP 算法

构造：$G_{k+1} = G_k + P_k + Q_k$

$$G_{k+1}y_k = G_k y_k + P_k y_k + Q_k y_k$$

让其中的

$$P_k y_k = \delta_k$$
$$Q_k y_k = -G_k y_k$$

即可得到最终的 $G_{k+1}y_k = \delta_k$

构造 Pk, Qk:

$$P_k = \frac{\delta_k \delta_k^{\mathrm{T}}}{\delta_k^{\mathrm{T}} y_k}$$
$$Q_k = -\frac{G_k y_k y_k^{\mathrm{T}} G_k}{y_k^{\mathrm{T}} G_k y_k}$$

可以满足上述的要求。

最终得到了迭代公式：$G_{k+1} = G_k + \frac{\delta_k \delta_k^{\mathrm{T}}}{\delta_k^{\mathrm{T}} y_k} - \frac{G_k y_k y_k^{\mathrm{T}} G_k}{y_k^{\mathrm{T}} G_k y_k}$

2.2 DFGS 算法

DFP 算法使用直接逼近 $H^{-}1_k$ 矩阵的，如果改成逼近 H_k 矩阵的 B_k 矩阵，那么就是 DFGS 算法。

$$B_{k+1}\delta_k = y_k$$

$$B_{k+1} = B_k + P_k + Q_k$$

$$B_{k+1}\delta_k = B_k\delta_k + P_k\delta_k + Q_k\delta_k$$

$$P_k\delta_k = y_k$$
$$Q_k\delta_k = -B_k\delta_k$$
$$B_{k+1} = B_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}}\delta_k} - \frac{B_k\delta_k\delta_k^{\mathrm{T}}B_k}{\delta_k^{\mathrm{T}}B_k\delta_k}$$

现在问题是，虽然 B_k 可以逼近 H_k 但是其实求解迭代时候还是需要用 H_k^{-1} 或者 G_k.

Algorithm 2 拟牛顿法 DFP 算法：

Input:

目标函数 $f(x)$, 梯度 $g(x)=\nabla f(x)$, Hesse 矩阵 $H(x)$, 精度要求 ϵ

Output:

$f(x)$ 的极小值点 x^*。

1: 取初始值 x^0, $k=0$，取正定对称矩阵 $G_0=I$.

2: 计算 $g_k=\nabla f(x^k)$, 如果 $||g_k||<\epsilon$, 则停止计算，当前的 x^k 就是 x^*.

3: 计算 $p_k=-G_k g_k$, 其中的 p_k 的计算用 GK 代替了牛顿法中的黑塞矩阵。

$$x=x^{(k)}+\lambda p_k=x^{(k)}-\lambda H_k^{-1}g_k$$

$$f(x)=f\left(x^{(k)}\right)-\lambda g_k^{\mathrm{T}}H_k^{-1}g_k$$

第二个公式是从开始的泰勒展开推出来的，表示正定矩阵条件下，f（x）必定会越来越小。

4: 一维搜索：求 λ_k

$$f\left(x^{(k)}+\lambda_k p_k\right)=\min_{\lambda\geqslant 0} f\left(x^{(k)}+\lambda p_k\right)$$

此时的 $x^{(k+1)}=x^{(k)}+\lambda_k p_k$

5: 计算此时的 g_{k+1}，如果达到要求，则停止，当前的 x^{k+1} 就是 x^*，否则求 $G_{k+1}=G_k+\frac{\delta_k\delta_k^{\mathrm{T}}}{\delta_k^{\mathrm{T}}y_k}-\frac{G_k y_k y_k^{\mathrm{T}}G_k}{y_k^{\mathrm{T}}G_k y_k}$ ，设置 k=k+1. 继续到 $p_k=-G_k g_k$，继续迭代，直到收敛。

我们可以得知，B_k 可以可以迭代得到，对于这种矩阵的逆矩阵可以使用 Sherman-Morrision 公式：

$$\left(A+uv^{\mathrm{T}}\right)^{-1}=A^{-1}-\frac{A^{-1}uv^{\mathrm{T}}A^{-1}}{1+v^{\mathrm{T}}A^{-1}u}$$

得到 G 的迭代公式：

$$G_{k+1}=\left(I-\frac{\delta_k y_k^{\mathrm{T}}}{\delta_k^{\mathrm{T}}y_k}\right)G_k\left(I-\frac{\delta_k y_k^{\mathrm{T}}}{\delta_k^{\mathrm{T}}y_k}\right)^{\mathrm{T}}+\frac{\delta_k\delta_k^{\mathrm{T}}}{\delta_k^{\mathrm{T}}y_k}$$

虽然形式与 DFP 方法所得到的形式不同，但是它是满足拟牛顿条件的。

Algorithm 3 拟牛顿法 DFGS 算法：

Input:

目标函数 $f(x)$, 梯度 $g(x) = \nabla f(x)$, Hesse 矩阵 $H(x)$, 精度要求 ϵ

Output:

$f(x)$ 的极小值点 x^*。

1: 取初始值 x^0, $k = 0$，取正定对称矩阵 $B_0 = I$.

2: 计算 $g_k = \nabla f(x^k)$, 如果 $||g_k|| < \epsilon$, 则停止计算，当前的 x^k 就是 x^*.

3: 计算 $B_k p_k = g_k$, 其中的 p_k 的计算用 bK 代替了牛顿法中的黑塞矩阵。

4: 一维搜索：求 λ_k

$$f\left(x^{(k)} + \lambda_k p_k\right) = \min_{\lambda \geqslant 0} f\left(x^{(k)} + \lambda p_k\right)$$

此时的 $x^{(k+1)} = x^{(k)} + \lambda_k p_k$

5: 计算此时的 g_{k+1}，如果达到要求，则停止，当前的 x^{k+1} 就是 x^*，否则求 $B_{k+1} = B_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \delta_k} - \frac{B_k \delta_k \delta_k^{\mathrm{T}} B_k}{\delta_k^{\mathrm{T}} B_k \delta_k}$ ，设置 k=k+1. 继续到 $p_k = -G_k g_k$，继续迭代，直到收敛。