优化方法之二阶优化方法

二阶优化方法主要有，牛顿法和拟牛顿法，和梯度下降一样，用于求解无约束最优化问题。特点是：收敛速度快，牛顿法也是一种迭代方法，每一步需要求解目标函数的海赛矩阵的逆矩阵，计算比较复杂，拟牛顿法通过正定矩阵近似海赛矩阵或者海赛矩阵的逆矩阵，简化计算。

1、牛顿法

考虑无约束最优化问题，是一个二阶问题

$$\min_{x \in R^n} f(x)$$

其中 x^* 为目标函数的极小点。假设 $f(x)$ 具有二阶连续偏导数，若第 k 次迭代值为 $x^{(k)}$，则可以将 $f(x)$ 在 $x^{(k)}$ 附近极小二阶泰勒展开：

$$f(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x - x^{(k)})^2$$

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

其中，$g_k = g(x^{(k)}) = \nabla f(x^{(k)})$ 是 $f(x)$ 的梯度向量在 $x^{(k)}$ 的值，$H(x^{(k)})$ 是 $f(x)$ 的海赛矩阵：

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

在点 $x^{(k)}$ 的值，函数 $f(x)$ 有极值的必要条件是，在极值点处一阶导数为**0**，也就是梯度向量为0，特别是当 $H(x^{(k)})$ 是正定矩阵时，函数 $f(x)$的极值为极小值。

正定矩阵的理解：
半正定矩阵定义为：$X^T MX \geq 0$，若所有特征值均不小于零，则称为半正定。若所有特征值均大于零，则称为正定，X 是特征向量，是任意一个向量，M 变换矩阵。

我们换一个思路看这个问题，矩阵变换中，MX 代表对向量 X 进行变换，我们假设变换后的向量为 Y，记做 $Y = MX$ 。于是半正定矩阵可以写成：

$$X^T Y \geq 0$$

这个是不是很熟悉呢？ 他是两个向量的内积。 同时我们也有公式：

$$cos(\theta) = \frac{X^T Y}{\|X\| \times \|Y\|}$$

$\|X\|, \|Y\|$ 代表向量 X, Y 的长度，θ 是他们之间的夹角。于是半正定矩阵意味着 $cos(\theta) \geq 0$， 这下明白了么？正定、半正定矩阵的直觉，代表一个向量经过它的变化后的向量 Y 与其本身 X^T 的夹角小于等于90度。

牛顿法利用极小值的必要条件：$\nabla f(x) = 0$，每次迭代中，从点 $x^{(k)}$ 开始，求目标函数的极小点，作为第 $k+1$ 次迭代值 $x^{(k+1)}$，假设 $x^{(k+1)}$ 满足：

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

将 $f(x)$ 的二阶泰勒展开式对 x 求导得：

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

其中，$H_k = H(x^{(k)})$，如果 $x = x^{(k+1)}$，那么就有：

$$g_k + H_k(x^{(k+1)} - x^{(k)}) = 0$$

所以有，

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

或者

$$x^{(k+1)} = x^{(k)} + p_k$$

$$H_k p_k = -g_k$$

上面的公式就是牛顿法的基本思想了，下面主要介绍算法：

牛顿法：
输入：目标函数 $f(x)$，梯度 $g(x) = \nabla f(x)$，海赛矩阵 $H(x)$，精度要求 ε
输出：$f(x)$ 的极小点 x^*

- 取初始点 $x^{(0)}$，置 $k = 0$
- 计算 $g_k = g(x^{(k)})$
- 若 $\|g_k\| < \varepsilon$，停止计算，得到近似解 $x^* = x^{(k)}$
- 计算 $H_k = H(x^{(k)})$，并求 p_k

$$H_k p_k = -g_k$$

- 置 $x^{(k+1)} = x^{(k)} + p_k$
- 置 $k = k + 1$

在上面的步骤中，求 p_k，$p_k = -H_k^{-1} g_k$，要求海赛矩阵的逆，计算比较复杂，所以很多人提出了改进算法。

2、拟牛顿法的思想

在牛顿法的迭代中，需要计算海赛矩阵的逆矩阵 $H^{(-1)}$，这一计算比较复杂，考虑用一个 n 阶矩阵 $G_k = G(x^{(k)})$ 来近似替代 $H_k^{-1} = H^{-1}(x^{(k)})$，这就是拟牛顿法的基本想法。

首先看牛顿法迭代中海赛矩阵 H_k 满足的条件，需要满足以下关系，我们假设 $x = x^{(k+1)}$，那么就有：

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

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

为了方便表示，我们假设 $y_k = g_{k+1} - g_k$，$\delta_k = x^{(k+1)} - x^{(k)}$，分别表示梯度差值和 x 的差值，所以有：

$$y_k = H_k \delta_k$$

或者

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

上面公式就表示拟牛顿条件，如果 H_k 是正定的（H_k^{-1} 也是正定的），那么就可以保证牛顿法搜索方向 p_k 是下降方向，因为特征值全部大于**0**，这是因为搜索方向是 $p_k = -H_k^{-1} g_k$，由式中可得：

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

所以 $f(x)$ 在 $x^{(k)}$ 处的一阶泰勒展开式可以近似的表示为：

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

因为 H_k^{-1} 是正定的，所有有 $g_k^T H_k^{-1} g_k > 0$，当 λ 为一个充分小的正数时，总有 $f(x) < f(x^{(k)})$，也就是说 p_k 是下降的方向。

拟牛顿法将 G_k 作为 H_k^{-1} 的近似，要求矩阵 G_k 满足同样的条件。首先，每次迭代矩阵G_k 是正定的，同事，G_k 满足下面的拟牛顿条件，让 $f(x)$ 在 $x = x^{(k+1)}$ 进行二阶泰勒展开：

$$\nabla f(x) = g_{k+1} + H_{k+1}(x - x^{(k+1)})$$

令　$x = x^{(k)}$，那么就有：

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

按照牛顿条件，选择 G_k 作为 H_k^{-1} 的近似，或者选择 B_k 作为 H_k 的近似算法称为拟牛顿法。按照拟牛顿法条件，在每次迭代中可以选择更新矩阵 G_{k+1}：

$$G_{k+1} = G_k + \Delta G_k$$

3、DFP算法

DFP 算法选择 G_{k+1} 的方法是，假设每一个迭代中矩阵 G_{k+1} 是由 G_k 加上两个附加项构成的，即：

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

其中 P_k，Q_k 是待定矩阵。这时，

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

为了使 G_{k+1} 满足牛顿条件，可使 P_k 和 Q_k 满足：

$$P_k y_k = \delta_k$$

$$Q_k y_k = -G_k y_k$$

不难找出 P_k 和 Q_k，例如取，作者随便取的：

$$P_k = \frac{\delta_k \delta_k^T}{\delta_k^T y_k}$$

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

这样就可以得到矩阵 G_{k+1} 的迭代公式了：

$$G_{k+1} = G_k + \frac{\delta_k \delta_k^T}{\delta_k^T y_k} - \frac{G_k y_k y_k^T G_k}{y_k^T G_k y_k}$$

这就是DFP算法，可以证明，如果初始矩阵 G_0 是正定的，则迭代过程中的每个矩阵 G_k 都是正定的。

DFP算法：
输入：目标函数 $f(x)$，梯度 $g(x) = \nabla f(x)$，精度要求 ε
输出：$f(x)$ 的极小点 x^*

- 取初始点 $x^{(0)}$，取 G_0 为正定矩对称矩阵，置 $k = 0$
- 计算 $g_k = g(x^{(k)})$，若 $\|g_k\| < \varepsilon$，停止计算，得到近似解 $x^* = x^{(k)}$
- 置 $p_k = -G_k g_k$
- 一维搜索：求 λ_k 使得：

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

- 置 $x^{(k+1)} = x^{(k)} + \lambda_k p_k$
- 计算 $g_{k+1} = g(x^{(k+1)})$，若 $\|g_{k+1}\| < \varepsilon$，停止计算，得到近似解 $x^* = x^{(k+1)}$，否则计算出 G_{k+1}
- 置 $k = k + 1$

4、BFGS算

BFGS 算法是最流行的拟牛顿法，可以考虑用 G_k 逼近海赛矩阵的逆矩阵 H^{-1}，也可以考虑用 B_k 逼近海赛矩阵 H。这时相应的拟牛顿条件是：

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

可以用同样的方法得到另一种迭代公式，首先令：

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

所以有：

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

考虑使 P_k 和 Q_k 满足：

$$P_k\delta_k = y_k$$

$$Q_k\delta_k = -B_k\delta_k$$

找出适合条件的 P_k 和 Q_k，得到 BFGS 算法矩阵 B_{k+1} 的迭代公式：

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

这就是BFGS算法，可以证明，如果初始矩阵 B_0 是正定的，则迭代过程中的每个矩阵 B_k 都是正定的。

BFGS算法：
输入：目标函数 $f(x)$，梯度 $g(x) = \nabla f(x)$，精度要求 ε
输出：$f(x)$ 的极小点 x^*

- 取初始点 $x^{(0)}$，取 B_0 为正定矩对称矩阵，置 $k = 0$
- 计算 $g_k = g(x^{(k)})$，若 $\|g_k\| < \varepsilon$，停止计算，得到近似解 $x^* = x^{(k)}$
- 置 $B_k p_k = -g_k$
- 一维搜索：求 λ_k 使得：

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

- 置 $x^{(k+1)} = x^{(k)} + \lambda_k p_k$
- 计算 $g_{k+1} = g(x^{(k+1)})$，若 $\|g_{k+1}\| < \varepsilon$，停止计算，得到近似解 $x^* = x^{(k+1)}$，否则计算出 G_{k+1}
- 置 $k = k + 1$