牛顿法和拟牛顿法

> - **牛顿法**（Newton method）和**拟牛顿法**（Quasi-Newton method）也是求解*无约束最优化问题*常用方法，优点是*收敛速度快*；
> - 牛顿法迭代的每一步需要求解目标函数的海森矩阵的逆矩阵，复杂度高；
> - 拟牛顿法通过正定矩阵近似海森矩阵的逆矩阵或海森矩阵，简化计算；

1. 牛顿法

- 考虑无约束最优化问题（x^*表示极小值点）：

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1}$$

假设$f(x)$具有二阶连续偏导数，在第k次迭代值为$x^{(k)}$，可将$f(x)$在$x^{(k)}$附近进行*二阶泰勒展开*:

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

其中$g_k = g(x^{(k)}) = \nabla f(x^{(k)})$表示$f(x)$在$x^{(k)}$的梯度向量的值，$H_k = H(x^{(k)})$是$f(x)$的**海森矩阵**（Hessian matrix）$H(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]_{n \times m}$在点$x^{(k)}$的值；

- **算法思路**：函数$f(x)$有极值的必要条件是在极值点处一阶导为0，即梯度向量为0，特别是当$H(x^{(k)})$是正定矩阵时，该极值点是极小值点；牛顿法利用极小值的

必要条件（$\nabla f(x)=0$），迭代时从点$x^{(k)}$出发，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$\nabla f(x^{(k+1)})=0$，根据泰勒展开式有：

$$\nabla f(x)=g_k+H_k(x-x^{(k)}) \tag{3}$$

代入可得：

$$g_k+H_k(x^{(k+1)}-x^{(k)})=0 \tag{4}$$

即

$$x^{(k+1)}=x^{(k)}+p_k \tag{5}$$

其中$p_k=-H_k^{-1}g_k$；

- **牛顿法算法流程**：
  1. **输入**目标函数$f(x)$，梯度$g(x)=\nabla f(x)$，海森矩阵$H(x)$，精度要求ϵ；
  2. 取初始点$x^{(0)}$，$k=0$；
  3. 计算梯度$g_k=g(x^{(k)})=\nabla f(x^{(k)})$，当$\|g_k\|<\epsilon$时停止迭代，令$x^*=x^{(k)}$；否则进入下一步；
  4. 计算$H_k=H(x^{(k)})$，求$p_k=-H_k^{-1}g_k$；
  5. 更新$x^{(k+1)}=x^{(k)}+p_k$，$k=k+1$，重复步骤（3）；
  6. **输出**极小值点x^*；
- 上述流程中步骤（4）需要求H_k^{-1}，计算比较复杂；

2. 拟牛顿法思路

- **算法思路**：考虑用一个n阶矩阵$G_k=G(x^{(k)})$来近似替代$H_k^{-1}=H^{-1}(x^{(k)})$；
- 在牛顿法中，海森矩阵H_k满足以下条件（根据泰勒展开式）：

$$g_{k+1}-g_k=H_k(x^{(k+1)}-x^{(k)}) \tag{6}$$

记$y_k=g_{k+1}-g_k$，$\delta_k=x^{(k+1)}-x^{(k)}$，那么$y_k=H_k\delta_k$或$H_k^{-1}y_k=\delta_k$称为**拟牛顿条件**；

- 如果H_k是正定的（H_k^{-1}也是正定），那么可以保证牛顿法*搜索方向*是下降的，因为搜索方向是$p_k=-H_k^{-1}g_k$，有

$$x=x^{(k)}+\lambda p_k=x^{(k)}-\lambda H_k^{-1}g_k \tag{7}$$

而$f(x)$在$x^{(k)}$的泰勒展开式近似写成：

$$f(x)=f(x^{(k)})-\lambda g_k^T H_k^{-1}g_k \tag{8}$$

因为H_k^{-1}正定，$g_k^T H_k^{-1}g_k>0$；当λ是一个充分小的正数时，总有$f(x)<f(x^{(k)})$，也即证明了p_k是下降方向；

- 拟牛顿法将G_k作为H_k^{-1}的近似（或者B_k逼近H_k），要求矩阵G_k满足同样条件：1）每次迭代G_k是正定的；2）满足拟牛顿条件$G_k y_k=\delta_k$；
- 按照拟牛顿条件，每次迭代中选择更新矩阵：$G_{k+1}=G_k+\Delta G_k$，根据不同具体实现产生多种算法

3. DFP算法

- **DFP算法**（Davidon-Fletcher-Powell algorithm）选择G_{k+1}思路：假设每次迭代矩阵G_{k+1}由G_k加上两个附加项（待定矩阵）构成

$$G_{k+1}=G_k+P_k+Q_k \tag{9}$$

- 因此有：

$$G_{k+1}y_k=G_k y_k+P_k y_k+Q_k y_k \tag{10}$$

为了使G_{k+1}满足拟牛顿条件，可使待定矩阵满足

$$\begin{aligned}P_k y_k&=\delta_k\\ Q_k y_k&=-G_k y_k\end{aligned}$$

可以容易地找出这样的P_k和Q_k：

$$\begin{aligned}P_k&=\frac{\delta_k\delta_k^T}{\delta_k^T y_k}\\ Q_k&=-\frac{G_k y_k y_k^T G_k}{y_k^T G_k y_k}\end{aligned}$$

- 所以G_{k+1}的迭代更新公式为：

$$G_{k+1}=G_k+\frac{\delta_k\delta_k^T}{\delta_k^Ty_k}-\frac{G_ky_ky_k^TG_k}{y_k^TG_ky_k} \tag{11}$$

- **可证明**：如果初始矩阵G_0是*正定的*，则迭代时G_k都是*正定的*；

- **DFP算法流程**：
  1. 输入**目标函数$f(x)$，梯度$g(x)=\nabla f(x)$，精度要求ϵ；
  2. 取初始点$x^{(0)}$，取G_0为正定对称矩阵，$k=0$；
  3. 计算梯度$g_k=g(x^{(k)})=\nabla f(x^{(k)})$，当$\|g_k\|<\epsilon$时停止迭代，令$x^*=x^{(k)}$，否则进入下一步；
  4. 求$p_k=-G_kg_k$；
  5. 一维搜索，求λ_k使其满足$f(x^{(k)}+\lambda_kp_k)=\min\limits_{\lambda\geq 0}f(x^{(k)}+\lambda p_k)$；
  6. 更新$x^{(k+1)}=x^{(k)}+\lambda_kp_k$；
  7. 计算梯度$g_{k+1}=g(x^{(k+1)})=\nabla f(x^{(k+1)})$，当$\|g_{k+1}\|<\epsilon$时停止迭代，令$x^*=x^{(k+1)}$，否则计算$G_{k+1}$：

  $$G_{k+1}=G_k+\frac{\delta_k\delta_k^T}{\delta_k^Ty_k}-\frac{G_ky_ky_k^TG_k}{y_k^TG_ky_k} \tag{12}$$

  8. $k=k+1$，重复步骤（4）；
  9. **输出**极小值点x^*；

4. BFGS算法

- **BFGS算法**（Broyden-Fletcher-Goldfarb-Shanno algorithm）：是最流行的拟牛顿算法，考虑用B_k逼近海森矩阵H_k；
- 相应的拟牛顿条件：

$$B_{k+1}\delta_k=y_k \tag{13}$$

同样地假设迭代矩阵B_{k+1}满足：

$$
\begin{aligned}
B_{k+1} &= B_k + P_k + Q_k \\
B_{k+1}\delta_k &= B_k\delta_k + P_k\delta_k + Q_k\delta_k
\end{aligned}
$$

可以考虑这样的P_k和Q_k：

$$
\begin{aligned}
P_k\delta_k &= y_k \\
Q_k\delta_k &= -B_k\delta_k
\end{aligned}
$$

- 找到B_{k+1}的迭代更新公式为：

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} \tag{14}$$

- **可证明**：如果初始矩阵B_0是*正定的*，则迭代时B_k都是*正定的*；

- **BFGS算法流程**：
  1. **输入**目标函数$f(x)$，梯度$g(x) = \nabla f(x)$，精度要求ϵ；
  2. 取初始点$x^{(0)}$，取B_0为正定对称矩阵，$k = 0$；
  3. 计算梯度$g_k = g(x^{(k)}) = \nabla f(x^{(k)})$，当$\|g_k\| < \epsilon$时停止迭代，令$x^* = x^{(k)}$，否则进入下一步；
  4. 由$B_k p_k = -g_k$求p_k；
  5. 一维搜索，求λ_k使其满足$f(x^{(k)} + \lambda_k p_k) = \min_{\lambda \geq 0} f(x^{(k)} + \lambda p_k)$；
  6. 更新$x^{(k+1)} = x^{(k)} + \lambda_k p_k$；
  7. 计算梯度$g_{k+1} = g(x^{(k+1)}) = \nabla f(x^{(k+1)})$，当$\|g_{k+1}\| < \epsilon$时停止迭代，令$x^* = x^{(k+1)}$，否则计算$B_{k+1}$：

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} \tag{15}$$

  8. $k = k + 1$，重复步骤（4）；
  9. **输出**极小值点x^*；

5. Broyden类算法

- 在BFGS算法中B_k的迭代公式中，如果记$G_k = B_k^{-1}, G_{k+1} = B_{k+1}^{-1}$，那么对迭代式（14）应用两次**Sherman-Morrison公式**可得：

$$G_{k+1}=\left(I-\frac{\delta_k y_k^T}{\delta_k^T y_k}\right)G_k\left(I-\frac{\delta_k y_k^T}{\delta_k^T y_k}\right)^T+\frac{\delta_k\delta_k^T}{\delta_k^T y_k} \tag{16}$$

称为BFGS算法关于G_k的迭代公式；

- 由DFP算法得到的G_{k+1}记作G^{DFP}，由BFGS算法得到的G_{k+1}记作G^{BFGS}，都满足拟牛顿条件，所以他们的线性组合 $G_{k+1}=\alpha G^{\mathrm{DFP}}+(1-\alpha)G^{\mathrm{BFGS}},0\le\alpha\le 1$也*满足拟牛顿条件*，而且*正定*；
- 于是就得到了一类拟牛顿法，称为Broyden类算法；

> **Sherman-Morrison公式**：假设A是n阶可逆矩阵，u,v是n维向量，且$A+uv^T$也是可逆矩阵，则有：
>
> $$(A+uv^T)^{-1}=A^{-1}-\frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u} \tag{17}$$