最优化方法

授课教师：徐海涛

2024年秋

第三章 无约束优化问题

3.7 牛顿法

- 最速下降法和共轭梯度法用到了一阶导数或梯度信息
- 牛顿法用到了函数的二阶导数或黑塞矩阵
- 因此，牛顿法如果收敛，那么其收敛速度也要比一阶方法更快

3.7 牛顿法

在迭代点$\boldsymbol{x}_k$处，构造二次型近似函数

$$q(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^{\mathrm{T}}\nabla^2 f(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)$$

如果函数(f) 在点(x_k) 处的Hessian矩阵($\nabla^2 f(x_k)$) 是正定的，那么可以通过求解($\nabla q = 0$)，来找到函数$q(x)$的极小点

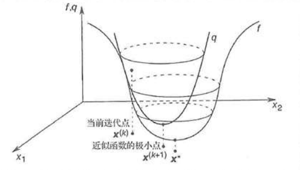

图 9.1 目标函数的二次型近似函数(用一阶和二阶导数构造而来)

$$[\nabla^2 f(\boldsymbol{x}_k)]\boldsymbol{d}_k = -\nabla f(\boldsymbol{x}_k) \quad \Rightarrow \quad \boldsymbol{d}_k = -[\nabla^2 f(\boldsymbol{x}_k)]^{-1}\nabla f(\boldsymbol{x}_k)$$

3.7 牛顿法

如果函数(f) 在点(x_k) 处的Hessian矩阵$(\nabla^2 f(x_k))$ 是正定的，那么可以通过求解$(\nabla q = 0)$，来找到函数$q(x)$的极小点

$$[\nabla^2 f(\boldsymbol{x}_k)]\boldsymbol{d}_k = -\nabla f(\boldsymbol{x}_k) \quad \Rightarrow \quad \boldsymbol{d}_k = -[\nabla^2 f(\boldsymbol{x}_k)]^{-1}\nabla f(\boldsymbol{x}_k)$$

其中，$\mathbf{d}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k$。求解式（3.37），可得 $\mathbf{d}_k$，并由此构造一个新点：

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$$

计算新点处的函数梯度和黑塞矩阵，再次利用式（3.37）和式（3.38），对迭代点进行更新。以此类推，给定一个初始点 $\mathbf{x}_0$ 就可以产生一个序列$\{\mathbf{x}_k\}$，其中的元素为各迭代点。序列的极限为极小点 $\mathbf{x}^*$，满足$\nabla f(\mathbf{x}^*) = \mathbf{0}$。当然，如果$f$本身就是二次型函数，且黑塞矩阵正定，则牛顿法将只需要一次迭代就收敛到极小点。但是，对于一般形式的函数，牛顿法并不一定收敛，除非其初始点与最优点特别接近。这可以从以下两个方面进行解释。

3.7 牛顿法（问题1）

- 如果函数是高度非线性的，那么二次型近似式对函数的近似程度可能非常差，图3.8给出了这样一种情形，其中的函数为一元函数
- 从图3.8中可以看出，新点所对应的函数值甚至比当前点的函数值更大

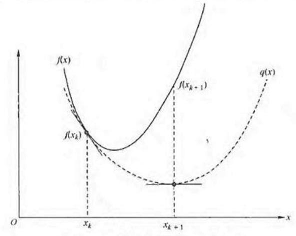

图 3.8 高度非线性函数的二次型近似

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$$

对牛顿方向的理解 $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$

3.7 牛顿法（问题2）

黑塞矩阵正定性的问题（一般形式函数）

其次，$\mathbf{x}^*$是极小点的充分条件要求黑塞矩阵$\nabla^2 f(\mathbf{x}^*)$正定，但是，在迭代过程中，并不能保证点 $\mathbf{x}_k$ 处黑塞矩阵的正定性。因此，如果$\nabla^2 f(\mathbf{x}_k)$不正定或者是奇异的，$q(\mathbf{x})$就不存在极小点。比如，可能存在鞍点甚至干脆是平坦的。式（3.37）可能不可解，或者即使可解，得到的也是比当前点更差的点。当然，如果f是严格凸的，那么黑塞矩阵是正定的，牛顿法结合线性搜索，即通过极小化$f(\mathbf{x}_k + \alpha \mathbf{d}_k)$来求取步长的工作方式是非常高效的。

3.7 牛顿法（例题）

例 3.10 给定函数$f(x) = x_1 - x_2 + 2x_1x_2 + 2x_1^2 + x_2^2$，初始点 $\mathbf{x}_0 = [1, 5]^{\mathrm{T}}$，试求点 $\mathbf{x}_0$ 处的牛顿方向。

有

$$\nabla f = [1 + 2x_2 + 4x_1, -1 + 2x_1 + 2x_2]^{\mathrm{T}}$$

$$\nabla f(\mathbf{x}_0) = [15, 11]^{\mathrm{T}}$$

$$\nabla^2 f = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

故牛顿方向为

$$\mathbf{d} = [\nabla^2 f]^{-1}(-\nabla f) = \begin{bmatrix} -2 \\ -3.5 \end{bmatrix}$$

3.7 牛顿法

围绕“创新性”论述与说明方面，通常考虑“灵魂五问”

1. 问题是什么？问题对象是什么？科学问题还是工程问题？
2. 前人是怎么分析这个问题的？他们采用了什么样的方案？还有什么是不明确的？
3. 如果给你来做，如何做得更好？从分析前人解决方案中，找到各个要素去改进
4. 你所提出新方法，难点在哪里？
5. 你将如何应对难点的挑战？技术方案

3.7 牛顿法（算法）

纯牛顿法的算法步骤

1. 选定初始点 $\mathbf{x}_0$ 以及精度参数 $\varepsilon_G, \varepsilon_A, \varepsilon_R$，令迭代次数 $k=0$。

2. 计算梯度$\nabla f(\mathbf{x}_k)$和黑塞矩阵$\nabla^2 f(\mathbf{x}_k)$。如果 $\|\nabla f(\mathbf{x}_k)\| \leqslant \varepsilon_G$，则算法停止；否则，按照式（3.37）确定下降方向。注意：不要把该方向归一化处理。

3. 更新迭代点 $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$。

4. 计算函数值 $f(\mathbf{x}_{k+1})$，判断是否满足式（3.18），如果是，算法停止；否则，令 $k=k+1$，$\mathbf{x}_k = \mathbf{x}_{k+1}$转第 2 步。

3.7 修正牛顿法（问题1）

为了避免出现图3.8 所示的情况，在迭代公式中引入步长α_k，即

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k \quad \Rightarrow \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$$

问题1

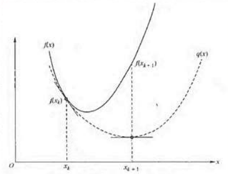

图 3.8 高度非线性函数的二次型近似

其中，α_k为利用线性搜索的方式求出的$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k)$极小点

方向$\boldsymbol{d}_k$ 可以按照修正黑塞矩阵的方式确定，也可以依据“充分下降＂的策略确定

3.7 修正牛顿法（问题2）

其次，另外一个方面的改进是保证方向 $\mathbf{d}_k$ 是函数f在点 $\mathbf{x}_k$ 处的下降方向。如果 $\mathbf{d}_k$ 是下降方向，那么利用线性搜索就可能产生一个新点 $\mathbf{x}_{k+1}$，满足$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$。为此，必须保证

$$\nabla f(\mathbf{x}_k)^{\mathrm{T}} \mathbf{d}_k < 0$$

结合式（3.37）可知，这相当于保证

$$-\nabla f(\mathbf{x}_k)^{\mathrm{T}} [\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k) < 0 \tag{3.39}$$

如果$\nabla^2 f(\mathbf{x}_k)$是正定矩阵，那么其逆矩阵同样也是正定的，可以满足式（3.39）。受此启发，一种修正策略就是利用一个对称正定矩阵 $\mathbf{F}_k$ 取代黑塞矩阵。矩阵 $\mathbf{F}_k$ 最简单的定义方式为

$$\mathbf{F}_k = \nabla^2 f(\mathbf{x}_k) + \gamma \mathbf{I} \tag{3.40}$$

3.7 修正牛顿法（问题2）

如果$\nabla^2 f(\mathbf{x}_k)$是正定矩阵，那么其逆矩阵同样也是正定的，可以满足式（3.39）。受此启发，一种修正策略就是利用一个对称正定矩阵 $\mathbf{F}_k$ 取代黑塞矩阵。矩阵 $\mathbf{F}_k$ 最简单的定义方式为

$$\mathbf{F}_k = \nabla^2 f(\mathbf{x}_k) + \gamma \mathbf{I} \tag{3.40}$$

$\mathbf{I}$ 为单位矩阵，通过选择合适的 γ，可以使得 $\mathbf{F}_k$ 的所有特征值都是正数，即全都大于某个值 $\delta>0$。因此，下降方向 $\mathbf{d}_k$ 就可以按照如下方式定义：

$$[\mathbf{F}_k]\mathbf{d}_k = -\nabla f(\mathbf{x}_k) \tag{3.41}$$

3.7 修正牛顿法（问题2）

需要注意的是，如果 δ 太接近于 0，矩阵 $\mathbf{F}_k$ 将是病态的，在求解式（3.41）时可能会产生舍入误差。参数 γ 可以依据$\nabla^2 f(\mathbf{x}_k)$的最小特征值确定，这可以通过逆幂法或对黑塞矩阵进行 $\mathbf{LDL}^{\mathrm{T}}$ 分解完成。在后者，$\mathbf{D}$ 为对角矩阵，如果所有对角元素 $D_{ii}>0$，则黑塞矩阵是正定的，否则，修改负的特征值，使其增加为正数，以保证 $\mathbf{F}_k$ 是正定的。随着 $\gamma\to\infty$，由式(3.40)可知，$\mathbf{F}_k\to\gamma\mathbf{I}$；此时，由式(3.41)可知，$\mathbf{d}_k\to-\frac{1}{\gamma}\nabla f(\mathbf{x}_k)$，这就是最速下降方向。$\gamma=1$ 意味着这是纯牛顿法。由此可见，γ 决定了下降方向和步长。最速下降方向自然是下降方向。因此，在牛顿法的迭代过程中，如果函数值 f 不再下降，应该增加 γ 的值。需要指出的是，对于一般形式的函数，修正牛顿法并没有特殊的吸引力。大家更倾向于前面讨论过的共轭梯度法和后面将要提到的拟牛顿法。但是，对于一些特殊结构的函数，其黑塞矩阵可以被非常准确地近似或者求解非常简单，修正牛顿法具备明显优势。

3.8 拟牛顿法

拟牛顿法利用一个近似黑塞矩阵 $\mathbf{F}_k$ 来替代黑塞矩阵，$\mathbf{F}_k$ 在前面已经提到过了。拟牛顿法不需要计算函数的二阶导数，即只利用了梯度信息。修正牛顿法的迭代公式为

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k[\mathbf{F}_k]^{-1}\nabla f(\mathbf{x}_k)$$

其中，$\mathbf{F}_k$ 是一个正定矩阵，是对黑塞矩阵$\nabla^2 f(\mathbf{x}_k)$的近似。上式涉及矩阵 $\mathbf{F}_k$ 的求逆，令 $\mathbf{H}_k$ 表示 $\mathbf{F}_k$ 的逆矩阵，可将迭代公式修改为

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k \nabla f(\mathbf{x}_k) \tag{3.43}$$

步长 α_k 通过求 $f(\mathbf{x}_k+\alpha\mathbf{d}_k)$的极小点确定，$\alpha>0$，$\mathbf{d}_k=-\mathbf{H}_k\nabla F(\mathbf{x}_k)$。

3.8 拟牛顿法

拟牛顿法的基本理念是从一个初始的对称正定矩阵 $\mathbf{H}$ 出发，可令 $\mathbf{H}_0=\mathbf{I}$，逐步更新矩阵 $\mathbf{H}$，使其越来越接近黑塞矩阵。第 1 章中已经提到过，前向差分公式可用于近似一阶导数，即利用两个不同的函数值来近似一阶导数。类似地，可以利用两个不同点处的梯度来近似二阶导数。具体而言，根据泰勒级数，可得

$$\nabla f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_k) + [\nabla^2 f(\mathbf{x}_k)]\boldsymbol{\delta}_k + \cdots \tag{3.44}$$

其中，$\boldsymbol{\delta}_k=\mathbf{x}_{k+1}-\mathbf{x}_k$。对于二次型函数，式（3.44）中未列出的高阶项都为零，因此，如果直接忽略掉它们，可得

$$[\nabla^2 f(\mathbf{x}_k)]\boldsymbol{\delta}_k = \boldsymbol{\gamma}_k \tag{3.45}$$

其中，$\boldsymbol{\gamma}_k=\nabla f(\mathbf{x}_{k+1})-\nabla f(\mathbf{x}_k)$。如果用矩阵 $\mathbf{H}$ 对黑塞矩阵逆矩阵进行近似，则可按照式（3.45）进行更新。这就意味着在迭代过程中，$\mathbf{H}_{k+1}$必须满足所谓的"拟牛顿条件"：

$$\mathbf{H}_{k+1}\boldsymbol{\gamma}_k = \boldsymbol{\delta}_k \tag{3.46}$$

3.8 拟牛顿法

DFP（Davidon-Fletcher-Powell）方法是一种拟牛顿法，顾名思义，由 Davidon、Fletcher 和 Powell 首次提出［Davidon，1959，Fletcher and Powell，1963］。该方法中，$\mathbf{H}$ 的更新方式为

$$\mathbf{H}_{k+1}=\mathbf{H}_k+a\mathbf{u}\mathbf{u}^{\mathrm{T}}+b\mathbf{v}\mathbf{v}^{\mathrm{T}}$$

代入式（3.46），可得

$$H_k\gamma_k+a\boldsymbol{u}\boldsymbol{u}^T\gamma_k+b\boldsymbol{v}\boldsymbol{v}^T\gamma_k=\boldsymbol{\delta}_k$$

令 $\mathbf{u}=\boldsymbol{\delta}_k$，$\mathbf{v}=\mathbf{H}_k\boldsymbol{\gamma}_k$，可得 $a\mathbf{u}^{\mathrm{T}}\boldsymbol{\gamma}_k=1$ 和 $b\mathbf{v}^{\mathrm{T}}\boldsymbol{\gamma}_k=-1$，由此确定 a 和 b，故 DFP 方法的更新方程为

$$\mathbf{H}_{k+1}^{\mathrm{DFP}}=\mathbf{H}_k-\frac{\mathbf{H}_k\boldsymbol{\gamma}_k\boldsymbol{\gamma}_k^{\mathrm{T}}\mathbf{H}_k}{\boldsymbol{\gamma}_k^{\mathrm{T}}\mathbf{H}_k\boldsymbol{\gamma}_k}+\frac{\boldsymbol{\delta}_k\boldsymbol{\delta}_k^{\mathrm{T}}}{\boldsymbol{\delta}_k^{\mathrm{T}}\boldsymbol{\gamma}_k} \tag{3.47}$$

容易看出，矩阵 $\mathbf{H}$ 在迭代过程中始终保持正定。而且，可以证明，$\mathbf{H}$ 始终保持正定（只要 $\mathbf{H}_0$ 是正定矩阵）。因此，$\mathbf{d}_k=-\mathbf{H}_k\nabla f(\mathbf{x}_k)$ 可以保证是下降方向。如果目标函数是二次型函数 $q(\mathbf{x})=\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}+\mathbf{c}^{\mathrm{T}}\mathbf{x}$ 则有 $\mathbf{H}_n=\mathbf{A}^{-1}$。这意味着对于二次型函数，$n$ 次迭代之后，$\mathbf{H}$ 就是黑塞矩阵的逆矩阵，即算法收敛。

3.8 拟牛顿法

BFGS 算法是拟牛顿法的另外一种算法，由 Broyden，Fletcher，Goldfarb 和 Shanno 共同提出［Fletcher，1987］。在该方法中，$\mathbf{H}$ 的更新公式为

$$\mathbf{H}_{k+1}^{\mathrm{BFGS}}=\mathbf{H}_k-\left(\frac{\boldsymbol{\delta}_k\boldsymbol{\gamma}_k^{\mathrm{T}}\mathbf{H}_k+\mathbf{H}_k\boldsymbol{\gamma}_k\boldsymbol{\delta}_k^{\mathrm{T}}}{\boldsymbol{\delta}_k^{\mathrm{T}}\boldsymbol{\gamma}_k}\right)+\left(1+\frac{\boldsymbol{\gamma}_k^{\mathrm{T}}\mathbf{H}_k\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_k^{\mathrm{T}}\boldsymbol{\gamma}_k}\right)\frac{\boldsymbol{\delta}_k\boldsymbol{\delta}_k^{\mathrm{T}}}{\boldsymbol{\delta}_k^{\mathrm{T}}\boldsymbol{\gamma}_k} \tag{3.48}$$

3.8 拟牛顿法

例 3.11 考虑二次型函数 $f=x_1^2+10x_2^2$，利用 DFP 方法开展两次迭代，初始点为 $[1,1]$。初始的 $\mathbf{H}_0$ 为单位矩阵，故下降方向为最速下降方向，即 $\mathbf{d}_0=[-2,-20]^{\mathrm{T}}$，计算得到步长为 $\alpha_0=0.05045$，构建新点为 $\mathbf{x}_1=[0.899,-0.0089]^{\mathrm{T}}$。在本书附带的程序 DFP 中，对下降方法进行了归一化处理，但此处并没有执行这一步。新迭代点处的梯度为 $\nabla f(\mathbf{x}_1)=[1.798,-0.180]$。因此，$\gamma_0=\nabla f(\mathbf{x}_1)-\nabla f(\mathbf{x}_0)=[-0.2018,-20.180]^{\mathrm{T}}$，$\delta_0=\mathbf{x}_1-\mathbf{x}_0=[-0.101,-1.1099]^{\mathrm{T}}$，由式(3.47)可得 $\mathbf{H}_1=\begin{bmatrix}1.0 & -0.005\\ -0.005 & 0.05\end{bmatrix}$，得到方向 $\mathbf{d}_1=[-1.80,0.018]^{\mathrm{T}}$，步长 $\alpha_1=0.5$ 构造新点 $\mathbf{x}_2=(0.0,0.0)$。梯度向量的范数为 0，算法停止，这就是最优解。如果继续更新矩阵 $\mathbf{H}$，可得 $\mathbf{H}_2=\begin{bmatrix}0.5 & 0\\ 0 & 0.05\end{bmatrix}$ 这恰好就是目标函数的黑塞矩阵。

3.10 使用MATLAB求解无约束优化问题

在 MATLAB 中，n 维无约束优化问题的求解函数为 fminunc。在调用该函数之前，需要自行构造一个子程序 GETFUN，用来描述目标函数 f。比如，可创建一个文件 getfun.m，代码如下：

```
function [f] = getfun(X)
f = 100 * (X(1) * X(1) - X(2)) ^ 2 + (1 - X(1)) ^ 2;
```

在命令行输入如下代码，调用程序 getfun：

```
[Xopt,fopt,iflag,output] = fminunc('getfun', X);
```

可得到最优解。通过命令

```
optimset('fminunc')
```

可看到函数的默认参数值。

本章小结

Summary
主要探讨了无约束优化问题的各种方法

1. 介绍了无约束优化问题的基本概念和重要性。
2. 最优性的必要条件和充分条件讨论了判断一个点是否为最优点的条件，包括一阶导数（梯度）和二阶导数（Hessian矩阵）的条件。
3. 介绍了凸性，解释了凸函数的性质，以及凸性在优化问题中的应用，特别是在确定最优点时的作用。
4. 基本概念：包括初始化、搜索方向和步长。
5. 介绍了最速下降法，利用目标函数梯度信息实现迭代优化
6. 讨论了共轭梯度法，可用于解决大规模优化问题的迭代方法，特别适合于处理稀疏矩阵。
7. 介绍了牛顿法，其使用二阶导数信息的优化方法，通常收敛速度快，但在计算二阶导数时可能代价较高。
8. 介绍了拟牛顿法，其是牛顿法的改进，它不直接计算Hessian矩阵，而是通过迭代更新一个近似的逆Hessian矩阵。

习题

作业题： P3.1 P3.3 P3.4 P3.10

思考题：P3.5 P3.6

人物拓展

牛顿法是牛顿在17世纪提出，用于求解非线性方程组。牛顿法利用函数的泰勒级数展开，通过迭代寻找方程的根。

20世纪中叶，由于牛顿法在计算Hessian矩阵及其逆时的复杂性，拟牛顿法被提出以简化计算。这种方法通过近似Hessian矩阵的逆来更新搜索方向。

关键科学家的贡献：
袁亚湘：中国科学院院士，与M.J.D.Powell共同研究，推动了优化领域的发展。
Davidon、Fletcher和Powell：共同提出了DFP方法，这是拟牛顿法的重要分支

本节结束，谢谢

Thank you for listening

课程名称：最优化方法