最速下降法（梯度法）

寇彩霞

Email: koucx@bupt.edu.cn

北京邮电大学理学院 主楼-816

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0),...$
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程，$x_2, x_3, \cdots$，使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0), ...$
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程，$x_2, x_3, \cdots$，使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程，$x_2, x_3, \cdots$，使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程, $x_2, x_3, \cdots$, 使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程，$x_2, x_3, \cdots$，使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程, $x_2, x_3, \cdots$, 使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程，$x_2, x_3, \cdots$，使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0,
  - 在d_0方向上，搜索下一个迭代点x_1,
  - 重复上述过程, $x_2, x_3, \cdots$, 使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0), \nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0，
  - 在d_0方向上，搜索下一个迭代点x_1，
  - 重复上述过程，$x_2, x_3, \cdots$，使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

优化算法框架

- 算法框架：
  - 初始点x_0开始，计算$f(x_0)$, $\nabla f(x_0)$,...
  - 现有信息基础上，选择从x_0出发的搜索方向d_0,
  - 在d_0方向上，搜索下一个迭代点x_1,
  - 重复上述过程, $x_2, x_3, \cdots$, 使得函数值下降。
- 关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

梯度（GRADIENT）

- $g(x) := \nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}]^T$
- 定义的一个方向——梯度方向
- 为什么选梯度方向？

$$\lim_{\alpha \to 0} \frac{f(x+\alpha d) - f(x)}{\alpha} = \lim_{\alpha \to 0} \frac{\alpha d^T g + o(\alpha)}{\alpha}$$

$$= d^T g = \|d\| \|g\| \cos \bar{\theta}$$

- $\cos \bar{\theta} = -1$，即 d 取负梯度方向——梯度法叫最速下降法。

梯度（Gradient）

- $g(x) := \nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}]^T$
- 定义的一个方向——梯度方向
- 为什么选梯度方向？

$$\lim_{\alpha\to 0}\frac{f(x+\alpha d)-f(x)}{\alpha} = \lim_{\alpha\to 0}\frac{\alpha d^T g + o(\alpha)}{\alpha}$$

$$= d^T g = \|d\|\|g\|\cos\bar{\theta}$$

- $\cos\bar{\theta} = -1$, 即 d 取负梯度方向——梯度法叫最速下降法。

梯度（GRADIENT）

- $g(x) := \nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}]^T$
- 定义的一个方向——梯度方向
- 为什么选梯度方向？

$$\lim_{\alpha\to 0}\frac{f(x+\alpha d)-f(x)}{\alpha} = \lim_{\alpha\to 0}\frac{\alpha d^T g + o(\alpha)}{\alpha}$$

$$= d^T g = \|d\|\|g\|\cos\bar{\theta}$$

- $\cos\bar{\theta} = -1$, 即 d 取负梯度方向——梯度法叫最速下降法。

- $g(x) := \nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}]^T$
- 定义的一个方向——梯度方向
- 为什么选梯度方向？

$$\lim_{\alpha\to 0}\frac{f(x+\alpha d)-f(x)}{\alpha}=\lim_{\alpha\to 0}\frac{\alpha d^T g+o(\alpha)}{\alpha}$$

$$=d^T g=\|d\|\|g\|\cos\bar{\theta}$$

- $\cos\bar{\theta}=-1$, 即 d 取负梯度方向——梯度法叫最速下降法。

梯度（GRADIENT）

- $g(x) := \nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}]^T$
- 定义的一个方向——梯度方向
- 为什么选梯度方向？

$$\lim_{\alpha\to 0}\frac{f(x+\alpha d)-f(x)}{\alpha} = \lim_{\alpha\to 0}\frac{\alpha d^T g + o(\alpha)}{\alpha}$$

$$= d^T g = \|d\|\|g\|\cos\bar{\theta}$$

- $\cos\bar{\theta} = -1$, 即 d 取负梯度方向——梯度法叫最速下降法。

最速下降法

- 设目标函数$f(x)$在x_k附近连续可微，且$g_k = \nabla f(x_k) \neq 0$.
  设$f(x)$在x_k处Taylor展开，

  $$f(x) = f(x_k) + g_k^T (x - x_k) + o(\|x - x_k\|). \tag{4.1.1}$$

  记$x - x_k = \alpha d_k$, 则上式可写为

  $$f(x_k + \alpha d_k) = f(x_k) + \alpha g_k^T d_k + o(\|\alpha d_k\|). \tag{4.1.2}$$

  显然，若d_k满足$g_k^T d_k < 0$, 则d_k是下降方向，使得$f(x_k + \alpha d_k) < f(x_k)$.

最速下降法

- 当α取定后，$d_k^T g_k$值越小，$-d_k^T g_k$值越大，函数$f(x)$在x_k的下降量越大.
- 由Cauchy-Schwartz不等式

$$|d_k^T g_k| \le \|d_k\| \|g_k\|, \tag{4.1.3}$$

当且仅当$d_k = -g_k$时，$d_k^T g_k$最小，$-d_k^T g_k$最大，从而$-g_k$是最速下降方向.

- 以$-g_k$为下降方向的方法叫最速下降法.

最速下降法

- 梯度法(gradient methods)：最简单最基本的多维无约束优化方法
- 回答关键问题：
  - 走哪个方向？
  - 走多长？
  - 算法能找到x^*吗？
  - 多快找到x^*？

- 最速下降法的迭代格式为

$$x_{k+1} = x_k - \alpha_k g_k, \tag{4.1.8}$$

其中步长因子α_k由线性搜索策略确定.

- 算法4.1.1
- 步1. 给出$x_0 \in R^n, 0 \leq \varepsilon \ll 1, k := 0$.
- 步2. 计算$d_k = -g_k$; 如果$\|g_k\| \leq \varepsilon$, 停止.
- 步3. 由线性搜索求步长因子α_k.
- 步4. 计算$x_{k+1} = x_k + \alpha_k d_k$.
- 步5. $k := k + 1$, 转步2.

采用精确步长的梯度法：Cauchy, 1847

例

- 用最速下降法求解无约束优化问题

$$\min f(x) = \frac{1}{2}[x_{(1)}^2 + 2x_{(2)}^2],$$

初始点$x_0 = (4, 4)^T$.

- 只利用局部梯度信息，局部算法，依赖初始点
- 快慢也依赖初始点
- 锯齿(Zigzag)现象

收敛性观察

- 只利用局部梯度信息，局部算法，依赖初始点
- 快慢也依赖初始点
- 锯齿(Zigzag)现象

收敛性观察

- 只利用局部梯度信息，局部算法，依赖初始点
- 快慢也依赖初始点
- 锯齿(Zigzag)现象

用精确线搜索的收敛性：

定理

设基本假设成立，而且$f(x) > -\infty$, 考虑精确线搜索方法，若$\sum_{k\geq 1}\cos^2\theta_k = +\infty$, 则必有$\liminf_{k\to+\infty}\|g_k\| = 0$.

- 梯度法满足 $\cos\theta_k = 1 \Rightarrow \sum_{k\geq 1}\|g_k\|^2 < +\infty$
  梯度法全局收敛

收敛快慢？

- 对于极小化正定二次函数$\min f(x)=\frac{1}{2}x^TGx$, 最速下降法产生的序列满足

$$\frac{f(x_{k+1})-f(x^*)}{f(x_k)-f(x^*)}\le\left(\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}\right)^2=\left(\frac{\kappa-1}{\kappa+1}\right)^2, \tag{4.1.12}$$

$$\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|}\le\sqrt{\frac{\lambda_1}{\lambda_n}}\left(\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}\right)=\sqrt{\kappa}\left(\frac{\kappa-1}{\kappa+1}\right), \tag{4.1.13}$$

其中λ_1和λ_n分别是矩阵G的最大和最小特征值，$\kappa=\lambda_1/\lambda_n$是矩阵$G$的条件数.

- 在非二次情形，如果$f(x)$在x^*附近二次连续可微，$\nabla f(x^*)=0, \nabla^2 f(x^*)$正定，则(4.1.12)也成立.

线性收敛

梯度法的改进——BB方法

- 由于精确线性搜索满足$g_{k+1}^T d_k = 0$, 则

$$g_{k+1}^T g_k = d_{k+1}^T d_k = 0.$$

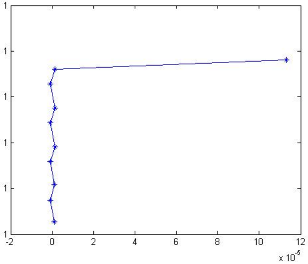
即相邻两次的搜索方向是互相直交的.

- 线性收敛、 开始的几步很快，快到最优点的时候，往往有锯齿现象。

例

$$\min f(x,y) = 10^3 x^2 + 10^{-3} y^2 \tag{1}$$

- 初始点 $[0.01, 1]^T$
  $d_1 = [-20.0000, -0.0020]^T, \quad x_2 = [0.00011316143149, 0.99999901131614]^T$
  $d_2 = [-0.2263, -0.0020]^T, \quad x_3 = [0.00000128055096, 0.99999802263326]^T$
  $d_3 = [-0.0026, -0.0020]^T, \quad x_4 = [-0.00000079556268, 0.99999640137054]^T$
  $d_4 = [0.0016, -0.0020]^T, \quad x_5 = [0.00000126144563, 0.99999381577804]^T$
  $d_5 = [-0.0025, -0.0020]^T, \quad x_6 = [-0.00000078369319, 0.99999219452213]^T$
- 迭代 10 步



梯度法的改进——BB方法

- 步长选取对梯度法的影响非常大.
  用最好的步长搭配最好的方向, 但效果不一定好.
- BB算法:

  参考文献:

  J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal. 8 (1988), pp. 141-148.

BB算法

- 主要思想：用前一步的信息确定当前步的步长。
- $x_{k+1} = x_k - D_k g_k$, 其中 $D_k = \alpha_k I$.
- 为使D_k具有拟牛顿性质，计算

$$\min \|s_{k-1} - D_k y_{k-1}\|, \quad or \quad \min \|D_k^{-1} s_{k-1} - y_{k-1}\|$$

- 解得

$$\alpha_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \quad or \quad \alpha_k^{BB2} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$$

- 图：$f(x, y) = 100x^2 + y^2$ starting at the initial point $(1, 100)^T$

- 主要思想：用前一步的信息确定当前步的步长。
- $x_{k+1} = x_k - D_k g_k$，其中 $D_k = \alpha_k I$.
- 为使D_k具有拟牛顿性质，计算

$$\min \|s_{k-1} - D_k y_{k-1}\|, \quad or \quad \min \|D_k^{-1} s_{k-1} - y_{k-1}\|$$

- 解得

$$\alpha_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \quad or \quad \alpha_k^{BB2} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$$

- 图：$f(x, y) = 100x^2 + y^2$ starting at the initial point $(1, 100)^T$

- 主要思想：用前一步的信息确定当前步的步长。
- $x_{k+1} = x_k - D_k g_k$, 其中 $D_k = \alpha_k I$.
- 为使D_k具有拟牛顿性质，计算

$$\min \|s_{k-1} - D_k y_{k-1}\|, \quad or \quad \min \|D_k^{-1} s_{k-1} - y_{k-1}\|$$

- 解得

$$\alpha_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \quad or \quad \alpha_k^{BB2} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$$

- 图：$f(x, y) = 100x^2 + y^2$ starting at the initial point $(1, 100)^T$

- 主要思想：用前一步的信息确定当前步的步长。
- $x_{k+1} = x_k - D_k g_k$，其中 $D_k = \alpha_k I$.
- 为使D_k具有拟牛顿性质，计算

$$\min \|s_{k-1} - D_k y_{k-1}\|, \quad or \quad \min \|D_k^{-1} s_{k-1} - y_{k-1}\|$$

- 解得

$$\alpha_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \quad or \quad \alpha_k^{BB2} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$$

- 图： $f(x, y) = 100x^2 + y^2$ starting at the initial point $(1, 100)^T$

BB算法

- 主要思想：用前一步的信息确定当前步的步长。
- $x_{k+1} = x_k - D_k g_k$, 其中 $D_k = \alpha_k I$.
- 为使D_k具有拟牛顿性质，计算

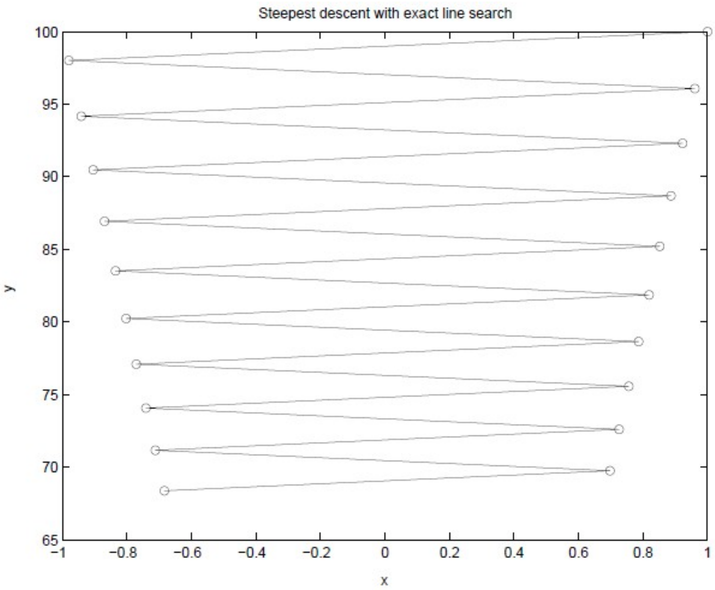
$$\min \|s_{k-1} - D_k y_{k-1}\|, \quad or \quad \min \|D_k^{-1} s_{k-1} - y_{k-1}\|$$

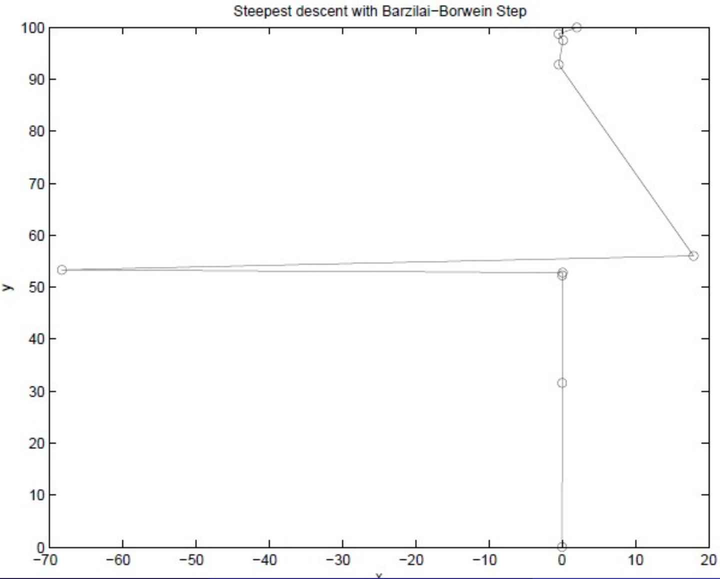
- 解得

$$\alpha_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \quad or \quad \alpha_k^{BB2} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$$

- 图：$f(x, y) = 100x^2 + y^2$ starting at the initial point $(1, 100)^T$

BB算法


Steepest descent with exact line search
100
95
90
85
80
75
70
65
y
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
x


Steepest descent with Barzilai−Borwein Step
100
90
80
70
60
50
40
30
20
10
0
y
−70
−60
−50
−40
−30
−20
−10
0
10
20
x

BB算法收敛性

- 一维时，BB=secant.
- 凸二次目标函数，$n = 2$，超线性收敛（Barzilai, Boruein 1988）.
- 其他情形，收敛性？有待解决.
- BB改进以及变种......

BB算法收敛性

- 一维时，BB=secant.
- 凸二次目标函数，$n = 2$，超线性收敛（Barzilai, Boruein 1988）.
- 其他情形，收敛性？有待解决.
- BB改进以及变种......

BB算法收敛性

- 一维时，BB=secant.
- 凸二次目标函数，$n = 2$，超线性收敛（Barzilai, Boruein 1988）.
- 其他情形，收敛性？有待解决.
- BB改进以及变种......

BB算法收敛性

- 一维时，BB=secant.
- 凸二次目标函数，$n = 2$，超线性收敛（Barzilai, Boruein 1988）.
- 其他情形，收敛性？有待解决.
- BB改进以及变种......

参考文献

- Y.H.Dai, Alternate step gradient method, 2001.
- Y.H.Dai ans R. Fletcher, On the asymptotic behavior of some new gradient methods, 2003.
- Y.H.Dai and X.Q.Yang, A new gradient method with an optimal stepsize property, 2001.
- Y.H.Dai, J.Y.Yuan and Y.Yuan, Modified two-point step-size gradient methods for unconstrained optimization, 2002.
- Y.H.Dai and Y.Yuan, Alternate minimization gradient method, 2003.
- Y.H.Dai and Y.Yuan, Analysis of monotone gradient methods, 2005.
- Y.H.Dai and H. Zhang, An adaptive two-point step-size gradient method, 2001.
- R. Fletcher, On the Barzilar-Borwein method, 2001.
- M. Raydan, On the Barzilar and Borwein choices of steplength for the gradient method, 1993.
- M. Raydan, the Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM J. Optim., 7(1997) 26-33.
- M. Raydan and B.F.Svaiter, Relaxed steepest descent and Cauchy-Barzilai-Borwein method, 2002.
- Y.X.Yuan, Step-sizes for the gradient method, 2007.

作业

- 要求掌握用精确线搜索的最速下降法求解二次函数的最优解.
- 用BB算法和梯度法求解无约束优化问题

$$\min f(x) = \frac{1}{2}[x_{(1)}^2 + 2x_{(2)}^2],$$

初始点$x_0 = (4,4)^T$. 并比较BB算法和最速下降(SD)法。

- $\alpha_k = \frac{g_k^T g_k}{g_k^T Q g_k}$, 考虑连续两步用该步长的最速下降法.