网络优先数字出版时间:2016-11-21 【DOI】10.13656/j.cnki.gxkx.20161121.003
网络优先数字出版地址:http://www.cnki.net/kcms/detail/45.1206.G3.20161121.1520.006.html

一种广义 BFGS Levenberg-Marquardt 算法*
A Generalized BFGS Levenberg-Marquardt Algorithm

冀祥麟,韦增欣**
JI Xianglin, WEI Zengxin

(广西大学数学与信息科学学院,广西南宁 530004)
(College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:提出一种基于 BFGS 更新的 Levenberg-Marquardt 算法,该算法不仅具有全局收敛性和二次收敛速度,而且可以更有效地求解大规模优化问题.数值实验表明,该算法在求解大规模绝对值方程问题方面也是有效的.

关键词:广义 Levenberg-Marquardt 算法 BFGS 更新 全局收敛性 绝对值方程
中图分类号:O224 **文献标识码**:A **文章编号**:1005-9164(2016)05-0428-04

Abstract: This paper proposes a modified Levenberg-Marquardt algorithm with BFGS update formula. Our algorithm converges globally to an optimal solution and the convergence rate is quadratic. Moreover, it has better efficiency for solving large-scale problems. Numerical results show that this algorithm is promising for solving large-scale absolute value equations problems.

Key words: generalized Levenberg-Marquardt algorithm, BFGS update, global convergence, absolute value equations

0 引言

【研究意义】考虑具有如下形式的非线性方程组:

$$g(x)=0, \tag{1}$$

其中,$g(x):\Re^n \to \Re^n$ 为连续可微函数.问题(1)在工程中有广泛应用,例如函数逼近,非线性拟合等.到目前为止,有多种算法可以用来解决问题(1),例如:Gauss-Newton 方法[1-4],信赖域方法[5],拟牛顿法[6]等.其中 Gauss-Newton 方法要求在算法迭代过程中 Jacobian 矩阵列满秩,这无疑限制了算法的应用范围.为解决此类问题,Levenberg[4] 和 Marquardt[1] 提出 Levenberg-Marquardt 方法(L-M 方法).令 $f(x)=\frac{1}{2}\|g(x)\|^2$,假设 $g(x)$ 有零点,那么非线性方程(1)等价于如下优化问题:

$$\min f(x), x\in\Re^n.$$

传统 L-M 方法通过求解下述模型获取搜索方向

$$s_k=\arg\min_{s\in\Re^n}\|J_k s+g_k\|^2+\mu_k\|s\|^2. \tag{2}$$

搜索方向为

$$s_k=-(J_k^T J_k+\mu_k I)-1J_k^T g_k, \tag{3}$$

式中,$g_k:=g(x_k)$,s_k 是 $g(x)$ 在 x_k 处的下降方向,$J_k:=J(x_k)$ 是 $g(x)$ 在 x_k 处的 Jacobian 矩阵,$\mu_k>0$ 为迭代参数.不难看出,传统 L-M 方法的每一次迭代都需要计算 Jacobian 矩阵,这对于大规模问题,会增加算法的计算存储量和时间.**【前人研究进展】**绝对值方程(AVE)是由 Rohn[7] 首次提出,是一类不可微的 NP-hard 问题[8].文献[9]证明该方程解的存在性

收稿日期:2016-07-25

作者简介:冀祥麟(1993-),男,硕士研究生,主要从事最优化理论研究。

*国家自然科学基金资助项目(11161003)和广西杰出青年科学基金项目(2015GXNSFGA139001)资助。

**通信作者:韦增欣(1962-),男,教授,博士生导师,主要从事最优化理论研究,E-mail:zxwei@gxu.edu.cn。

和唯一性.文献[10]也给出一类绝对值方程与线性互补问题的等价关系.在AVE求解方面,Caccetta等[11]提出用光滑牛顿方法求解绝对值方程,并且证明其具有全局收敛性质.Mangasarian[12]提出用广义牛顿方法求解此类问题.Ketabchi等[13]提出一种范数方法求解绝对值方程.更多绝对值方程的求解方法见参考文献[14-15].**【本研究切入点】**为有效求解大规模问题,本文在适当的条件下提出一种在不需要非退化假设[16]即可具备全局收敛性,并且在一定条件下具有二次收敛性的算法.最重要的是,可以不必在每次迭代时都计算Jacobian矩阵,而且对于Jacobian矩阵 J_k,通过BFGS推广得到.令 $y_k=g(x_{k+1})-g(x_k)$,B_k 为广义BFGS,有近似关系:

$$y_k=g(x_{k+1})-g(x_k)\approx\nabla g(x_{k+1})s_k. \tag{4}$$

此时,B_{k+1} 满足正割方程 $B_{k+1}s_k=y_k$,又有逼近关系:

$$B_{k+1}s_k\approx\nabla g(x_{k+1})s_k.$$

这意味着 B_{k+1} 沿方向 s_k 逼近 $\nabla g(x_{k+1})$,即

$$B_{k+1}=B_k-\frac{B_ks_ks_k^TB_k}{s_k^TB_ks_k}+\frac{y_ky_k^T}{y_k^Ts_k}, \tag{5}$$

其中,$s_k=x_{k+1}-x_k$,$y_k=g(x_{k+1})-g(x_k)$,x_{k+1} 为下一次迭代.**【拟解决的关键问题】**提出一种基于BFGS更新的L-M算法,并分析算法在如下形式的绝对值方程中的应用:

$$g(x):=Ax-|x|-b=0, \tag{6}$$

其中,A,b 为给定的已知常量矩阵或向量,$x,b\in\Re^n$,$A\in\Re^{n\times n}$,$x=(x_1,x_2,\cdots,x_n)^T$.文中,$\|\ \|$ 表示欧几里得范数.

1 绝对值方程的基本结论

Mangasarian[8]证明求解AVE是一个不可微的NP-hard的优化问题.又由文献[17-18]可知基于 $|x|$ 的次梯度的广义Jacobian矩阵 $\partial|x|$ 为一对角矩阵,即

$$h(x)=\mathrm{diag}(\mathrm{sign}(x)), \tag{7}$$

其中,

$$\mathrm{sign}(x)=\begin{cases}1,\text{when } x>0,\\0,\text{when } x=0,\\-1,\text{when } x<0.\end{cases}$$

$\mathrm{diag}(\cdot)$ 表示一对角阵,即有 $\partial g(x):=J(x)=A-h(x)$.

根据文献[10]中的性质3和4,给出绝对值方程解的存在性和唯一性基本结论.

引理1 i)若 $A\in\Re^n\times\Re^n$,且 A 的所有奇异值都大于1,则对任意的 $b\in\Re^n$,绝对值方程存在唯一解;

ii)若 $A\in\Re^n\times\Re^n$,且 $\|A^{-1}\|<1$,则对任意的 $b\in\Re^n$,绝对值方程存在唯一解.

2 算法及其收敛性分析

基于BFGS更新,提出一种BFGS Levenberg-Marquardt算法.

算法1

Step 0 选取参数 $\beta,\sigma\in(0,1)$,$\mu_0>0$,初始迭代点 $x_0\in\Re^n$,$0\leqslant\varepsilon\leqslant1$.令 $k:=0$.

Step 1 若 $f(x)\leqslant\varepsilon$,算法终止;否则,进入Step 2.

Step 2 求解方程组 $(J_k^TJ_k+\mu_kI)s=-J_k^Tg_k$,得到 s_k.

Step 3 Armijo线搜索.令 λ_k 是满足下面不等式的最小非负整数 λ:

$$f(x_k+\beta^\lambda s_k)\leqslant f_k+\sigma\beta^\lambda\nabla f_k^Ts_k, \tag{8}$$

令 $\alpha_k:=\beta^{\lambda_k}$,$x_{k+1}:=x_k+\alpha_ks_k$.

Step 4 $\mu_k=\|g(x_k)\|^{1+\tau}$,$\tau\in[0,1]$,令 $y_k=g_{k+1}-g_k$,如果 $y_ks_k>0$,按照(5)式更新 J_{k+1};否则,令 $J_{k+1}=J_k$.

Step 5 令 $k:=k+1$,转Step 1.

引理2 式(2)中,s_k 是 $f(x)$ 在 x_k 处的下降方向.

证明 由最优性条件,知 s_k 满足

$$\nabla(\|J_ks+g_k\|^2+\mu_k\|s\|^2)=2[(J_k^TJ_k+\mu_kI)s+J_k^Tg_k]=0,$$

求得 $s_k=-(J_k^TJ_k+\mu_kI)^{-1}J_k^Tg_k$.若 $J_k^Tg_k\neq0$,则对 $\forall\mu_k>0$,有

$$(J_k^Tg_k)^Ts_k=-(J_k^Tg_k)^T(J_k^TJ_k+\mu_kI)^{-1}(J_k^Tg_k)<0.$$

所以,s_k 是 $f(x)$ 在 x_k 处的下降方向,则引理2得证.

定理1(全局收敛性) 假设 $\{x_k\}$ 由算法1迭代产生得到,且步长满足Armijo线搜索准则,如果存在一个子序列 $x_{k_j}\to x^*$,而且满足相应的子序列 $\{J(x_{k_i})^TJ(x_{k_i})+\mu_{k_i}I\}$ 收敛于正定矩阵 $J(x^*)^TJ(x^*)+\mu^*I$,那么,$\nabla f(x^*)=0$.

证明 若 $\nabla f(x_k)\neq0$,则 $s_k\neq0$.由引理2,$\mu_k>0$ 及 $x_{k_j}\to x^*$ 得到

$$J_{k_j}^TJ_{k_j}\to J(x^*)^TJ(x^*),u_{k_j}\to u^*.$$

由于 $s_{k_j}=-[J(x_{k_j})^TJ(x_{k_j})+\mu_{k_j}I]^{-1}J(x_{k_j})^Tg(x_{k_j})$,则 $s_{k_j}\to s^*=-[J(x^*)^TJ(x^*)+\mu^*I]^{-1}J(x^*)^Tg(x^*)$.

因此对于 $\beta\in(0,1)$,存在非负整数 λ^* 使得

$$f(x^*+\beta^{\lambda^*}s^*)<f(x^*)+\sigma\beta^{\lambda^*}\nabla f(x^*)^Ts^*.$$

对于子列 $x_{k_j} \to x^*$，当 j 充分大时，有

$$f(x_{k_j} + \beta^{\lambda^*} s_{k_j}) < f(x_{k_j}) + \sigma\beta^{\lambda^*} \nabla f(x_{k_j})^T s_{k_j}.$$

由 Armijo 步长准则知 $\lambda^* \geqslant \lambda_{k_j}$，所以

$$f(x_{k_{j+1}}) = f(x_{k_j} + \beta^{\lambda_{k_j}} s_{k_j}) \leqslant f(x_{k_j}) +$$
$$\sigma\beta^{\lambda_{k_j}} \nabla f(x_{k_j})^T s_{k_j} \leqslant f(x_{k_j}) + \sigma\beta^{m^*} \nabla f(x_{k_j})^T s_{k_j},$$

即对充分大的 j，有

$$f(x_{k_{j+1}}) \leqslant f(x_{k_j}) + \sigma\beta^{\lambda^*} \nabla f(x_{k_j})^T s_{k_j}. \qquad (9)$$

又

$$\lim_{j\to\infty} f(x_{k_{j+1}}) = \lim_{j\to\infty} f(x_{k_j}) = f(x^*),$$

从而对(9)式两边求极限，得

$$f(x^*) \leqslant f(x^*) + \sigma\beta^{\lambda^*} \nabla f(x^*)^T s^*.$$

这与引理 2 矛盾，所以 $\nabla f(x^*) = 0$. 由反证法知定理 1 得证.

又根据文[19]中定理 7.4 和注 7.2，可以得到算法的收敛速度是二次的. 这里省去二次收敛性质的证明过程，详细证明见文献[19]. 文献[20]也给出了类似的证明.

定理 2(二次收敛性质) 假设 $\{x_k\}$ 是由算法迭代得到的，且收敛到 $f(x)$ 的一个局部最小值点 x^*，$\mu_k = \| g(x_k) \|^{1+\tau}$，$\tau \in [0,1]$，那么，算法的收敛速度是二次收敛的.

3 数值实验

以求解绝对值方程问题为例，验证算法 1 对于求解大规模优化问题的有效性. 实验的所有代码都在 Matlab R2010b 中运行，其中，电脑的配置：CPU 为 Intel Pentium (R) Dual-Core E5800 3.20 GHz，SDRAM 为 2.00 GB 的 Windows7 操作系统.

在引理 1 的条件下，对随机生成不同维数的绝对值方程进行计算，维数 Dim 分别设置为 500，1000，1500，2000，2500，3000. 对于不同维数的绝对值方程，分别随机生成 10 次进行计算并记录数据. 在每个随机问题的求解中，算法的参数选择为 $\beta=0.5$，$\sigma=0.3$，$\tau = \frac{1}{2}$，$\varepsilon = 10^{-8}$，初始点 $x_0 = \mathrm{rand}(n,1)$.

数值实验结果见表 1，其中，Dim 表示问题维数；$\mathrm{Iter}_{\mathrm{total}}$ 表示算法 1 求解 10 次问题所需的总迭代次数；$\mathrm{Cputime}_{\mathrm{total}}$ 表示算法求解 10 次问题所需总时间；$\mathrm{Opt}_{\mathrm{average}}$ 表示算法求解 10 次问题的平均误差，即

$$\mathrm{Opt}_{\mathrm{average}} = \frac{\sum_{i=1}^{10} (f^i_{\mathrm{final}} - \mathrm{ops})}{10}.$$

这里，f^i_{final} 为算法 1 求解第 i 个随机算例的终值，ops 为对应问题的最优值，在本文中为 0. 对于随机生成的不同维数算例中，每种维数选取其中一次做 Opt(Iter)值迭代随迭代次数 Iter 的变化图(图 1). 结果显示算法 1 是有效的.

表 1 数值结果

Table 1 Numerical results

Dim	$\mathrm{Iter}_{\mathrm{total}}$	$\mathrm{Cputime}_{\mathrm{total}}$	$\mathrm{Opt}_{\mathrm{average}}$
500	59	1.456425e+02	2.573539e−10
1000	59	6.337541e+02	3.963551e−10
1500	63	1.556172e+03	1.288582e−09
2000	55	2.260922e+03	1.869581e−09
2500	64	4.888463e+03	2.653575e−09
3000	64	7.682067e+03	1.955933e−09

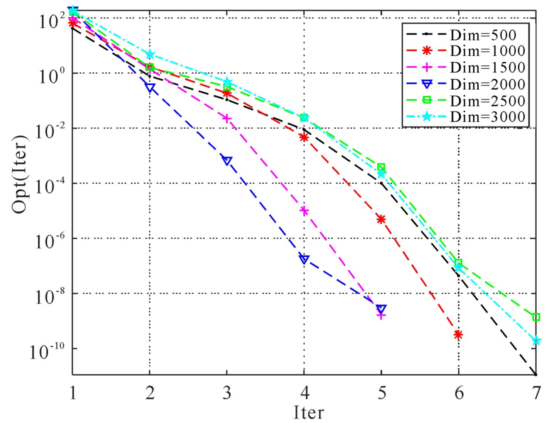

图 1 Opt(Iter)的值随迭代次数 Iter 的变化

Fig. 1 The values of Opt(Iter) with the number of iterations Iter

4 结论

本文提出一种基于 BFGS 更新的 L-M 算法，证明算法具有全局收敛性和二次收敛性质. 该算法在每次迭代中，可以避免计算 Jacobian 矩阵，这在处理大规模问题中可以节约 CPU 耗费时间. 在数值实验中，我们用算法测试了随机生成不同维数绝对值方程问题，结果表明算法是有效的.

参考文献：

[1] MARQUARDT D W. An algorithm for least-squares estimation of nonlinear parameters[J]. Journal of the Society for Industrial and Applied Mathematics, 1963, 11 (2): 431-441.

[2] NOCEDAL J, WRIGHT S J. Numerical Optimization [M]. New York: Springer-Verlag, 2006.

[3] BERTSEKAS D P. Nonlinear Programming[M]. 2nd edition. Belmount, Mass, USA: Athena Scientific, 1999.

[4] LEVENBERG K. A method for the solution of certain non-linear problems in least squares[J]. Quarterly of Applied Mathematics, 1944, 2(2): 164-168.

[5] YUAN Y X. Trust Region Algorithms for Nonlinear E-

quations[M]. Hong Kong:Department of Mathematics, Hong Kong Baptist University,1994.
[6] ZHU D T. Nonmonotone backtracking inexact quasi Newton algorithms for solving smooth nonlinear equations [J]. Applied Mathematics and Computation, 2005, 161(3):875-895.
[7] ROHN J. Systems of linear interval equations[J]. Linear Algebra and its Applications,1989,126:39-78.
[8] MANGASARIAN O L. Absolute value programming [J]. Computational Optimization and Applications, 2007,36(1):43-53.
[9] ROHN J. A theorem of the alternatives for the equation [J]. Linear and Multilinear Algebra, 2004, 52(6): 421-426.
[10] MANGASARIAN O L, MEYER R R. Absolute value equations[J]. Linear Algebra and Its Applications, 2006,419(2/3):359-367.
[11] CACCETTA L, QU B, ZHOU G L. A globally and quadratically convergent method for absolute value equations[J]. Computational Optimization and Applications,2011,48(1):45-58.
[12] MANGASARIAN O L. A generalized newton method for absolute value equations[J]. Optimization Letters, 2009,3(1):101-108.
[13] KETABCHI S, MOOSAEI H. An efficient method for optimal correcting of absolute value equations by minimal changes in the right hand side[J]. Computers & Mathematics with Applications, 2012, 64(6): 1882-1885.
[14] YONG Q L. An iterative method for absolute value equations problem[J]. Information,2013,16(1):7-12.
[15] HU S L, HUANG Z H. A note on absolute value equations[J]. Optimization Letters,2010,4(3):417-424.
[16] YUAN G L, WEI Z X, LU X W. A BFGS trust-region method for nonlinear equations[J]. Computing, 2011, 92(4):317-333.
[17] POLYAK B T. Introduction to Optimization[M]. New York:Optimization Software Inc,1987.
[18] ROCKAFELLAR R T. New Applications of Duality in Convex Programming[C]//Proceedings of the Fourth Conference on Probability. Brasov,1971.
[19] 马昌凤.最优化方法及其 Matlab 程序设计[M].北京:科学出版社,2010.
MA C F. Optimization Methods and Matlab Programming[M]. Beijing:Science Press,2010.
[20] 袁亚湘,孙文瑜.最优化理论与方法[M].北京:科学出版社,1997.
YUAN Y X, SUN W Y. Optimization Theory and Method[M]. Beijing:Science Press,1997.

(责任编辑:尹　闯)