A Class of LBFGS-Type Algorithms for Large-Scale Unconstrained Optimization*

QIAN Xiaoyan[1†] SHI Qingsheng[1] LIU Hao[1] SHI Kuiran[2]

Abstract In this paper, value information of objective function is exploited in limited memory BFGS-type algorithms. We first construct a new quadratic function satisfying some interpolation conditions to approximate the objective function, and get a new weak secant equation. Combining the new weak secant equation with that obtained by Yuan[1], a class of limited memory BFGS-type algorithms including the classic LBFGS algorithm based on a new weak secant equation is proposed. The convergence of this class limited memory BFGS-type algorithms is proved. Numerical results for standard test problems from CUTE are reported, which indicate that all the algorithms in the proposed class perform quite well.

Keywords unconstrained optimization, weak secant equation, BFGS algorithm, convergence analysis, limited memory

Chinese Library Classification O221

2010 Mathematics Subject Classification 90C30, 65K05

大规模无约束优化的一族 LBFGS 类算法

钱小燕[1†] 施庆生[1] 刘 浩[1] 石岿然[2]

摘要 尝试在有限存储类算法中利用目标函数值所提供的信息. 首先利用插值条件构造了一个新的二次函数逼近目标函数, 得到了一个新的弱割线方程, 然后将此弱割线方程与袁[1] 的弱割线方程相结合, 给出了一族包括标准 LBFGS 的有限存储 BFGS 类算法, 证明了这族算法的收敛性. 从标准试验函数库 CUTE 中选择试验函数进行了数值试验, 试验结果表明这族算法的数值表现都与标准 LBFGS 类似.

关键词 无约束优化, 弱割线方程, BFGS 算法, 收敛性分析, 有限存储

中图分类号 O223

数学分类号 90C30, 65K05

收稿日期: 2011 年 1 月 23 日.

* This work is supported by the National Natural Science Foundation of China (71071075) and the Natural Science Project of Nanjing University of Technology (39704017).

1. College of Sciences, Nanjing University of Technology, Nanjing 210009, China; 南京工业大学理学院, 南京 210009

2. College of Economics and Management Science, Nanjing University of Technology, Nanjing 210009, China; 南京工业大学经济与管理学院, 南京 21009

† 通讯作者 Corresponding author

0 Introduction

We consider the following nonlinear unconstrained optimization

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

where the objective function $f(x)$ is assumed to be twice continuously differentiable in R^n and n is sufficient large.

For nonlinear unconstrained optimization problems Quasi-Newton iterative algorithms are widely used. On the kth iteration, an approximation point x_k and a $n\times n$ matrix B_k are available and a search direction $d_k = -B_k^{-1}\nabla f(x_k)$ is calculated. The next iterative point x_{k+1} is set to be $x_k + \alpha_k d_k$, where the step-length α_k is calculated to satisfy certain line search conditions. One of the important features of the algorithms is the choice of matrices B_k. Quasi-Newton algorithms require B_k symmetric positive definite and satisfying the quasi-Newton equation

$$B_{k+1}s_k = y_k, \tag{2}$$

where $s_k = x_{k+1} - x_k = \alpha_k d_k, y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

Quasi-Newton iterative algorithms have some good properties, such as quadratic termination, fast local convergence rate and less computational efforts than Newton method, especially the numerical performance of the famous BFGS method in Broyden class is little affected by inexact line searches. In the past decades, BFGS quasi-Newton algorithm is widely applied for nonlinear minimization, and its local and global convergence is proved. It is also well known that the quasi-Newton equation can be derived from the following quadratic function

$$m_{k+1}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^{\mathrm{T}}(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^{\mathrm{T}}B_{k+1}(x - x_{k+1}) \tag{3}$$

to approximate the objective function at x_{k+1}, which satisfies the following three interpolation conditions

$$m_{k+1}(x_{k+1}) = f(x_{k+1}), \tag{4}$$

$$\nabla m_{k+1}(x_{k+1}) = \nabla f(x_{k+1}), \tag{5}$$

$$\nabla m_{k+1}(x_k) = \nabla f(x_k). \tag{6}$$

However, there is no function value information in the quasi-Newton equation (2). Recently there are some papers try to include function value information in Quasi-Newton equation. In 1991, Yuan[1] proposed a modified BFGS quasi-Newton algorithm, which requires the approximation quadratic function (3) satisfying the interpolation conditions (4)-(5)and

$$m_{k+1}(x_k) = f(x_k) \tag{7}$$

instead of (6). It is easy to find from Yuan's approximate quadratic function that the weak secant equation

$$s_k^{\mathrm{T}}B_{k+1}s_k = 2[f(x_k) - f(x_{k+1}) + s_k^{\mathrm{T}}\nabla f(x_{k+1})] \tag{8}$$

is satisfied. Considering the Hermite interpolation model satisfying the conditions(4)-(7), one can get a new weak equation

$$s_k^{\mathrm{T}} B_{k+1} s_k = 4 s_k^{\mathrm{T}} \nabla f(x_{k+1}) + 2 s_k^{\mathrm{T}} \nabla f(x_k) - 6(f(x_{k+1}) - f(x_k)). \tag{9}$$

Yuan and Byrd[2] used this equation to derive some non-quasi-Newton updates. Zhang and Xu[3] derived a similar form of this equation in a different way and a new modified quasi-Newton equation

$$B_{k+1} s_k = y_k + \frac{\theta_k}{s_k^{\mathrm{T}} u} u \text{ with } s_k^{\mathrm{T}} u \neq 0 \tag{10}$$

where $\theta_k = 6(f(x_k) - f(x_{k+1})) + 3(g_k + g_{k+1})^{\mathrm{T}} s_k$, is given. Davidon[4] introduced 'conic models' where a non-quadratic approximate function is constructed satisfying interpolation conditions (4)-(7). we refer [5] for recent development of conic model methods. Wang and Ni[6] introduced a new moving asymptotes model for nonlinear unconstrained optimization, where a nonquadratic approximate function is constructed satisfying interpolation conditions (4)-(6). Wei, Li and Qi[7] presented some other modified quasi-Newton algorithms which preserve the local and global convergence properties.

When n is sufficient large, it is impossible for us to store and update an n order square matrix. Hence, conjugate gradient algorithms or limited memory BFGS [8−10] algorithms are perfered for large-scale unconstrained optimization. Generally speaking, conjugate gradient algorithms only use several vectors with linear convergence rate. For more details about conjugate gradient algorithms please see [11] and reference therein. Limited memory algorithms need not store an n order square matrix, only compute square matrices multiply vectors with R-linear convergence rate. Although there are a lot of modified quasi-Newton equations incorporating function value information, only a few papers discuss limited memory BFGS-type algorithms[12−13] with function value information. Hence, it may be interesting to discuss on limited memory BFGS-type algorithms with objective function value information.

In this paper, we propose a class of limited memory BFGS-type algorithms for large-scale unconstrained optimization, which has one parameter γ ranging from zero to one and includes the standard LBFGS. If the parameter γ equals to $\frac{1}{2}$, the proposed algorithm turns out to be the standard LBFGS of [8].

1 Algorithms

In this section, we derive a new weak secant equation by quadratic interpolation and propose a new class of limited memory BFGS-type algorithms for large-scale unconstrained optimization.

Assume the following quadratic function

$$m_{k+1}(x) = \hat{f}_{k+1} + \hat{g}_{k+1}^{\mathrm{T}}(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^{\mathrm{T}} B_{k+1}(x - x_{k+1}) \tag{11}$$

satisfying the interpolation conditions (4), (6) and (7), one can see that

$$\begin{aligned}&\hat{f}_{k+1} = f(x_{k+1}),\\ &\hat{g}_{k+1} - B_{k+1}s_k = \nabla f(x_k),\\ &\hat{f}_{k+1} - \hat{g}_{k+1}^{\mathrm{T}}s_k + \frac{1}{2}s_k^{\mathrm{T}}B_{k+1}s_k = f(x_k).\end{aligned}$$

Combining the above three equations, we obtain a new weak secant equation

$$s_k^{\mathrm{T}}B_{k+1}s_k = 2[f(x_{k+1}) - f(x_k) - \nabla f(x_k)^{\mathrm{T}}s_k]. \tag{12}$$

Note that the new weak secant equation is always satisfied when the objective function $f(x)$ is quadratic, and when the objective function $f(x)$ is strictly convex, $2[f(x_{k+1}) - f(x_k) - g_k^{\mathrm{T}}s_k] > 0$ is always true. Since the weak secant equation (8) and the scalar $2[f(x_k) - f(x_{k+1}) + g_{k+1}^{\mathrm{T}}s_k]$ have the same above properties, we propose a new class of modified BFGS updates as Yuan[1] in the following

$$B_{k+1} = B_k - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + t_k \frac{y_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}, \tag{13}$$

where

$$\begin{aligned}&t_k = \gamma\mu_k + (1-\gamma)\nu_k, \quad 0 \leqslant \gamma \leqslant 1,\\ &\mu_k = \frac{2}{s_k^{\mathrm{T}}y_k}[f(x_k) - f(x_{k+1}) + \nabla f(x_{k+1})^{\mathrm{T}}s_k],\\ &\nu_k = \frac{2}{s_k^{\mathrm{T}}y_k}[f(x_{k+1}) - f(x_k) - \nabla f(x_k)^{\mathrm{T}}s_k].\end{aligned}$$

From the update formula (13), one can easily see the following weak secant equation

$$s_k^{\mathrm{T}}B_{k+1}s_k = \gamma[f(x_k)-f(x_{k+1})+\nabla f(x_{k+1})^{\mathrm{T}}s_k]+(1-\gamma)[f(x_{k+1})-f(x_k)-\nabla f(x_k)^{\mathrm{T}}s_k] \tag{14}$$

is satisfied. It is easy to see that $t_k \geqslant 0$, if $f(x)$ is convex. Furthermore, $t_k \equiv 1$, if $f(x)$ is quadratic on the line segment between x_k and x_{k+1}. Moreover, when $\gamma = 1/2$, $t_k = \frac{1}{2}\mu_k + \frac{1}{2}\nu_k \equiv 1$, i.e. the update (13) turns out to be the standard BFGS update ignoring whether the objective function is quadratic or not, while for other cases the function value information is exploited.

Denote $\hat{y}_k = t_k y_k$, $H_k = B_k^{-1}$, we rewrite (13) in BFGS-like form

$$B_{k+1} = B_k - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + \frac{\hat{y}_k \hat{y}_k^{\mathrm{T}}}{s_k^{\mathrm{T}} \hat{y}_k}.$$

Apply Sherman-Morrision formula two times or the duality between DFP and BFGS updates, we obtain

$$\begin{aligned}H_{k+1} &= H_k + \frac{(s_k - H_k\hat{y}_k)s_k^{\mathrm{T}} + s_k(s_k - H_k\hat{y}_k)^{\mathrm{T}}}{s_k^{\mathrm{T}}\hat{y}_k} - \frac{(s_k - H_k\hat{y}_k)^{\mathrm{T}}\hat{y}_k}{(s_k^{\mathrm{T}}\hat{y}_k)^2}s_k s_k^{\mathrm{T}}\\ &= \left(I - \frac{s_k\hat{y}_k^{\mathrm{T}}}{s_k^{\mathrm{T}}\hat{y}_k}\right)H_k\left(I - \frac{\hat{y}_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}}\hat{y}_k}\right) + \frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}}\hat{y}_k}\\ &= \left(I - \frac{s_k y_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}\right)H_k\left(I - \frac{y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}\right) + t_k^{-1}\frac{s_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}.\end{aligned}$$

Compared with the inverse form BFGS update, one can find the only difference is that $t_k^{-1} \equiv 1$ in BFGS update formula. Since the inverse updating formula has the same form of inverse BFGS update, we can describe the proposed algorithms in the following.

Algorithm 1.1 (A class of LBFGS-type algorithms for large-scale unconstrained optimization)

Step1 Given small constant $\epsilon_1 > 0, \epsilon_2 > 0$ and $0 < c_1 < \frac{1}{2}, c_1 < c_2 < 1$, choose initial point x_0, integer m and initial symmetric positive matrix H_0, set $k := 0$.

Step2 Compute $\nabla f(x)$, if $\|\nabla f(x_k)\| < \epsilon_1 \max\{1, \|x_k\|\}$, where here, and for the rest of the paper, $\|\cdot\|$ denotes Euclidean vector or matrix norm , or $f(x_k) - f(x_{k+1}) < (1 + |f(x_k)|)\epsilon_2$, stop; else go to Step 3.

Step3 Compute $d_k = -H_k \nabla f(x_k)$, and set $x_{k+1} = x_k + \alpha_k d_k$, where α_k satisfies Wolfe-Powell inexact line search conditions:

$$f(x_k + \alpha_k d_k) \leqslant f(x_k) + c_1 \alpha_k \nabla f(x_k)^{\mathrm{T}} d_k, \tag{15}$$

$$\nabla f(x_k + \alpha_k d_k) \geqslant c_2 \nabla f(x_k)^{\mathrm{T}} d_k. \tag{16}$$

Step4 Let $\hat{m} = \min\{k, m-1\}$. Update H_0 $\hat{m}+1$ times using the triplets $\{s_j, y_j, t_j\}_{j=k-\hat{m}}^{k}$, i.e. let

$$\begin{aligned} H_{k+1} = {} & (V_k^{\mathrm{T}} \cdots V_{k-\hat{m}}^{\mathrm{T}}) H_0 (V_{k-\hat{m}} \cdots V_k) \\ & + \rho_{k-\hat{m}} (V_k^{\mathrm{T}} \cdots V_{k-\hat{m}+1}^{\mathrm{T}}) s_{k-\hat{m}} s_{k-\hat{m}}^{\mathrm{T}} (V_{k-\hat{m}+1} \cdots V_k) \\ & + \rho_{k-\hat{m}+1} (V_k^{\mathrm{T}} \cdots V_{k-\hat{m}+2}^{\mathrm{T}}) s_{k-\hat{m}+1} s_{k-\hat{m}+1}^{\mathrm{T}} (V_{k-\hat{m}+2} \cdots V_k) \\ & \quad \vdots \\ & + \rho_k s_k s_k^{\mathrm{T}}. \end{aligned} \tag{17}$$

where $V_k = I - \frac{y_k s_k^{\mathrm{T}}}{s_k^{\mathrm{T}} y_k}$, $\rho_k = (t_k s_k^{\mathrm{T}} y_k)^{-1}$. Increase k by one, go to Step 2.

Remarks (1) The matrices H_k are not formed explicitly, but the $\hat{m}+1$ previous triplets of (s_j, y_j, t_j) are stored separately. (2) In order to ensure the search directions $d_k, k = 1, 2, \cdots$, are descent defections, the matrices H_k should be positive definite. If the objective function is strong uniformly convex, t_k are bounded and $t_k > 0$ are always true. However, when the objective is not uniformly convex, t_k may be unbounded or $t_k > 0$ may not be maintained. Hence, t_k must be truncated in an interval $[a, b]$, where $b > a > 0$. (3) As for the initial symmetric positive matrix H_0, we generally choose H_0 to be an identity matrix or a scaled diagonal positive matrix $\gamma_0 I$, where $\gamma_0 = s_0^{\mathrm{T}} y_0 / t_0 \|y_0\|^2$ providing $t_0 > 0$.

In the next section, we will prove all the algorithms in the proposed class are globally convergent on uniformly convex problems and that their rates of convergence are all R-linear.

2 Convergence analysis

In order to prove the global convergence of the proposed class algorithms, we give an equivalent form of Algorithm 1.1 where $B_k = H_k^{-1}$.

Algorithm 2.1 (General form limited memory BFGS-type algorithms)

Step1 Given small constant $\epsilon_1 > 0, \epsilon_2 > 0$ and $0 < c_1 < \frac{1}{2}, c_1 < c_2 < 1$, choose initial point x_0, integer m and initial symmetric positive matrix B_0, set $k := 0$.

Step2 Compute $\nabla f(x)$, if $\|\nabla f(x_k)\| < \epsilon_1 \max\{1, \|x_k\|\}$, or $f(x_k) - f(x_{k+1}) < (1 + |f(x_k)|)\epsilon_2$, stop; else go to Step 3.

Step3 Compute $d_k = -B_k^{-1}\nabla f(x_k)$, and set $x_{k+1} = x_k + \alpha_k d_k$, where α_k satisfies Wolfe-Powell inexact line search conditions:

$$f(x_k + \alpha_k d_k) \leqslant f(x_k) + c_1\alpha_k \nabla f(x_k)^{\mathrm{T}} d_k,$$
$$\nabla f(x_k + \alpha_k d_k) \geqslant c_2 \nabla f(x_k)^{\mathrm{T}} d_k.$$

Step4 Let $\hat{m} = \min\{k, m-1\}$, and define a symmetric and positive definite matrix $B_k^{(0)}$. Update $B_k^{(0)}$ $\hat{m}+1$ times using the triplets $\{s_j, y_j, t_j\}_{j=k-\hat{m}}^{k}$, i.e. for $j = k-\hat{m}, \cdots, k$ compute

$$B_k^{(j)} = B_k^{(j)} - \frac{B_k^{(j)} s_j s_j^{\mathrm{T}} B_k^{(j)}}{s_j^{\mathrm{T}} B_k^{(j)} s_j} + t_k^{(j)} \frac{y_j y_j^{\mathrm{T}}}{s_j^{\mathrm{T}} y_j},$$

set $B_{k+1} = B_k^{(k)}$, increase k by one, go to Step 2.

Before proven the global convergence of Algorithm 2.1, we give the following assumption.

Assumption 2.2

(1)The objective function $f(x)$is twice continuously differentiable in R^n.

(2)The level set $D_k = \{x \in R^n : \ f(x) \leqslant f(x_0)\}$ is convex.

(3) There exist positive constants M_1 and M_2 such that

$$M_1\|z\|^2 \leqslant z^{\mathrm{T}} G(x) z \leqslant M_2\|z\|^2, \tag{18}$$

where $G(x)$ denotes the second derivatives of the objective function $f(x)$, for all $z \in R^n$ and $x \in D$. Note that this implies that $f(x)$ has a unique minimizer x^* in D.

Now under Assumption 2.2, we state a lemma to estimate the bounds of t_k.

Lemma 2.3 If Assumption 2.2 is satisfied. Then

$$\frac{M_1}{M_2} \leqslant t_k \leqslant \frac{M_2}{M_1}.$$

Proof Since Assumption 2.2 is satisfied, we derive that

$$\begin{aligned}
2[f(x_k) - f(x_{k+1}) + \nabla f(x_{k+1})^{\mathrm{T}} s_k] &= 2\Big[- s_k^{\mathrm{T}} \int_0^1 \nabla f(x_{k+1} + \tau(x_k - x_{k+1}))\mathrm{d}\tau \\
&\qquad + \nabla f(x_{k+1})^{\mathrm{T}} s_k \Big] \\
&= 2 s_k^{\mathrm{T}} \Big[\int_0^1 [\nabla f(x_{k+1}) - \nabla f(x_{k+1} - \tau s_k)]\mathrm{d}\tau \Big] \\
&= 2 s_k^{\mathrm{T}} \int_0^1 \int_0^\tau \tau G(x_{k+1} - (1-\beta)\tau s_k)\mathrm{d}\beta\mathrm{d}\tau s_k \\
&= s_k^{\mathrm{T}} G(\hat{x}) s_k
\end{aligned}$$

holds. Therefore, we obtain

$$M_1\|s_k\|^2 \leqslant 2[f(x_k) - f(x_{k+1}) + \nabla f(x_{k+1})^{\mathrm{T}} s_k] \leqslant M_2\|s_k\|^2.$$

Similarly we can prove that

$$M_1\|s_k\|^2 \leqslant 2[f(x_{k+1}) - f(x_k) - \nabla f(x_k)^{\mathrm{T}} s_k] \leqslant M_2\|s_k\|^2.$$

Define $\bar{G}_k = \int_0^1 G(x_k + \tau s_k)\mathrm{d}\tau$, then

$$s_k^{\mathrm{T}} y_k = s_k^{\mathrm{T}} \bar{G}_k s_k.$$

Hence, it is easy to see that

$$M_1\|s_k\|^2 \leqslant s_k^{\mathrm{T}} y_k \leqslant M_2\|s_k\|^2. \tag{19}$$

From the definition of t_k, we obtain

$$\frac{M_1}{M_2} \leqslant t_k \leqslant \frac{M_2}{M_1}.$$

This completes the proof of the lemma.

Denote $\cos\theta_k = \frac{s_k^{\mathrm{T}} B_k s_k}{\|s_k\|\|B_k s_k\|}$, we cite Lemma 2.1 of [14] in the following for use.

Lemma 2.4 If the step length α_k satisfies the Wolfe-Powell inexact line search conditions (15)-(16) and Assumption 2.2 holds, then there exists a positive constant $c > 0$ such that

$$f(x_{k+1}) - f(x^*) \leqslant (1 - c\cos^2\theta_k)(f(x_k) - f(x^*)).$$

For the sake of simplicity, we assume Algorithm 2.1 always generates infinitive iterative points. Now we are ready to present the main convergence theorem of LBFGS-type algorithms as follows.

Theorem 2.5 Let x_0 be a starting point for which $f(x)$ satisfies Assumption 2.2, and assume that the matrices chosen such that $\{\|B_k^{(0)}\|\}$ and the norms of their inverse matrices are bounded. Then for any positive definite B_0, Algorithm 2.1 generates a sequence $\{x_k\}$ which converges to the unique minimizer x^*. Moreover there is a constant $0 \leqslant r < 1$ such that

$$f(x_k) - f(x^*) \leqslant r^k[f(x_0) - f(x^*)], \tag{20}$$

which implies that $\{x_k\}$ converges $R-$linearly.

Proof From Assumption 2.2 we have

$$\frac{\|y_k\|^2}{s_k^{\mathrm{T}} y_k} = \frac{s_k^{\mathrm{T}} \bar{G}_k^2 s_k}{s_k^{\mathrm{T}} \bar{G}_k s_k} \leqslant M_2. \tag{21}$$

We proceed by analyzing the trace and determinant of B_{k+1}. From Lemma 2.3, (21) and the bound of $\{\|B_k^{(0)}\|\}$, we have

$$\begin{aligned}
\mathrm{tr}(B_{k+1}) &\leqslant \mathrm{tr}(B_k^{(0)}) + \sum_{j=k-\hat{m}}^{k} t_k^{(j)} \frac{\|y_j\|^2}{s_j^{\mathrm{T}} y_j} \\
&\leqslant \mathrm{tr}(B_k^{(0)}) + \hat{m}\frac{M_2^2}{M_1} \\
&\leqslant M_3,
\end{aligned} \tag{22}$$

for some positive constant M_3. It is easy to find that

$$\det(B_{k+1}) = \det(B_k^{(0)}) \prod_{j=k-\hat{m}}^{k} t_k^{(j)} \frac{s_j^{\mathrm{T}} y_j}{s_j^{\mathrm{T}} B_k^{(j)} s_j} = \det(B_k^{(0)}) \prod_{j=k-\hat{m}}^{k} t_k^{(j)} \frac{s_j^{\mathrm{T}} y_j}{s_j^{\mathrm{T}} s_j} \frac{s_j^{\mathrm{T}} s_j}{s_j^{\mathrm{T}} B_k^{(j)} s_j}.$$

Since by (22) the largest eigenvalue of $B_k^{(l)}$ is also less than M_3. From (19), Lemma 2.3 and the bound of $\{\|B_k^{(0)}\|\}$, we have

$$\det(B_{k+1}) \geqslant \det(B_k^{(0)}) \prod_{j=k-\hat{m}}^{k} \frac{M_1^2}{M_2 M_3} = \det(B_k^{(0)}) \left(\frac{M_1^2}{M_2 M_3}\right)^{\hat{m}} \geqslant M_4, \tag{23}$$

for some positive constant M_4. Therefore from (22) and (23) we conclude that there is a constant $\delta > 0$ such that $\cos\theta_k \geqslant \delta$. From Lemma 2.4 we obtain (20). Since the objective function is uniformly convex, we have

$$\frac{1}{2} M_1 \|x_k - x^*\|^2 \leqslant f(x_k) - f(x^*),$$

which together with (20) implies

$$\|x_k - x^*\| \leqslant r^{k/2} [2(f(x_0) - f(x^*))/M_1]^{1/2},$$

where $r = 1 - c\delta^2$. Hence the sequence $\{x_k\}$ is $R-$ linearly convergent. The proof of this theorem is complete.

In the above, we have proved the convergence of the proposed algorithms. The classic LBFGS algorithm can be viewed as a special case in our class. In the next section, we will pay attention to the numerical performance of the proposed algorithms (Algorithm 1.1).

3 Numerical experiments

In this section, we examine the numerical performance of Algorithm 1.1 by choosing different γ in (17). We choose some standard test problems from CUTE[16] and code Algorithm 1.1 by slightly modifying Nocedal's LBFGS subroutine with double precision.

By setting $\gamma = 0, 1/2, 1$ in (17) respectively, we obtain the LBFGS-type algorithm derived by our approximate function(see the first part of Section 1), the classic LBFGS algorithm and the LBFGS-type algorithm which limited memorizes corresponding modified BFGS algorithm of Yuan[1]. If $\gamma = 1/4, 3/4$, one can give other LBFGS -type algorithms with function value information in the proposed class. Note when $\gamma = 2$, Algorithm 1.1 turns out to be that of [13]. Although it is not included in our proposed class, one can find that its convergence properties are still preserved (see [13]).

Since we can not ensure that t_k is positive and bounded for general nonlinear functions, it is necessary to truncate $t_k \in [a, b]$. Here we try the same strategy of Yuan[1], i.e. if $t_k < 0.01$, set $t_k = 0.01$; if $t_k > 100$, then $t_k = 100$.

Table 3.1 The numerical results of Algorithm 1.1 with $m = 3$

Name	Dim	$\gamma = 0$	$\gamma = 1/4$	$\gamma = 1/2$	$\gamma = 3/4$	$\gamma = 1$	$\gamma = 2$
arwhead	1000	11/22/13	11/22/14	11/22/13	11/21/13	10/21/12	12/26/14
beale	1000	13/20/16	13/20/16	14/21/16	11/18/13	11/18/14	9/17/11
brownbs	1000	14/43/19	13/42/18	12/37/17	14/43/19	14/45/19	12/38/18
cragglvy	1000	29/140/103	20/26/22	33/39/36	18/24/20	15/22/18	39/48/41
dixmaana	1000	15/20/17	13/17/15	15/19/17	14/18/16	13/18/16	14/19/16
dixmaanb	1000	28/59/32	27/53/32	18/19/23	26/45/33	17/24/20	15/23/18
dixmaanc	1000	31/63/39	28/55/35	25/44/29	19/29/22	17/29/21	11/22/14
dixmaane	1000	256/261/258	192/197/194	244/249/246	199/203/201	240/243/242	244/248/246
dqdrtic	2000	13/37/15	13/37/15	13/37/15	13/37/15	13/37/15	13/37/15
eg2	2000	12/74/20	7/48/34	10/64/23	11/58/21	5/45/27	9/57/27
freuroth	2000	8/16/11	10/17/12	9/18/11	10/20/12	10/18/12	9/17/11
himmelblau	2000	6/18/10	5/15/10	9/19/13	10/21/14	9/19/13	11/22/15
nondia	2000	52/132/68	57/137/67	63/161/85	61/153/77	58/127/69	42/90/49
nondquar	2000	239/360/251	291/407/300	214/306/221	270/378/280	289/432/295	203/296/211
penalty1	2000	108/223/115	136/333/148	46/81/54	139/419/151	71/170/78	147/446/160
powellsg	2000	47/100/51	48/103/52	49/104/53	35/74/38	36/74/39	41/84/44
quartc	3000	15/18/17	15/18/17	14/17/16	14/17/16	13/16/15	9/12/11
rosen	3000	34/62/38	34/66/39	34/64/39	31/56/33	33/62/40	26/52/28
tridia	1000	342/1372/344	342/1372/344	342/1372/344	342/1372/344	342/1372/344	342/1372/344
whiteholst	4000	26/54/30	24/51/28	22/43/24	13/27/18	24/45/27	28/52/32
woods	10000	117/360/123	109/359/128	92/291/107	152/481/174	164/518/186	82/264/96

As for the termination and line search parameters, we set $\epsilon_1 = 1.0D-5$, $\epsilon_2 = 1.0D-16$, while $c_1 = 1.0D-4, c_2 = 0.9D0$. The initial positive symmetric matrix is $H_0 = I$. All calculations are carried out on a portable computer(AMD,1.61GHZ, 480M memory). The numerical results of $m = 3$ are listed in Table 3.1 with iteration times/ function value calculation times/ gradient calculation times in order.

Since the proposed m of the standard LBFGS is $3-7$(see Nocedal's LBFGS subroutine), here we only give numerical results of $m = 3$ without scaling techniques. From Table 3.1 we can see that there is little difference between the numerical results as γ changes from zero to one, hence we do not give CPU time. In fact, the difference between the CPU time is so little that we nearly can not find it for nearly all the problems(except dixmaane, nondquar and woods).

4 Conclusion

In this paper, we first construct a quadratic function by interpolation to approximate the objective function at x_{k+1}, get a new weak secant equation and then combining the secant equation gotten by Yuan[1] present a class of limited memory BFGS-type algorithms including the standard LBFGS. All the algorithms in the proposed class have global $R-$linear convergent rate for uniformly convex functions. We also note that just recently Li, Qi and Roshchina[17] propose a new class of quasi-Newton updates, which do not satisfy some interpolation conditions. Hence, it is possible to limited memorize quasi-Newton type updates which do not satisfy interpolation conditions. Cheng and Li[18] propose a spectral scaling BFGS algorithm which has the the following update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + \gamma_k \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}, \tag{24}$$

where $\gamma_k = \frac{y_k^{\mathrm{T}} s_k}{\|y_k\|^2}$, and report that the spectral scaling BFGS algorithm performs better than the standard BFGS algorithm. Note that the formula (24) has the same form as (13), we think limited memory spectral scaling BFGS algorithm may perform better than the standard limited memory BFGS algorithm.

References

[1] Yuan Y X. A modified BFGS algorithm for unconstrained optimization [J]. *IMA J Numer Anal,* 1991, **11**: 325-332.

[2] Yuan Y X, Byrd R H. Non-quasi-Newton updates for unconstrained optimization[J]. *J Comput Math*, 1995,**13**(2): 95-107.

[3] Zhang J Z, Xu C X. Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations [J]. *J Comput Appl Math*, 2001,**137**: 269-278.

[4] Davidon W C. Conic approximations and collinear scalings for optimizers [J]. *SIAM J Numer Anal,* 1980, **17**: 268-281.

[5] Ni Q. Optimality conditions for trust-Region subproblems involving a conic model [J]. *SIAM J Optim*, 2005, **15**: 826-837.

[6] Wang H J, Ni Q. A new method of moving asymptotes for large scale unconstrained optimization [J]. *Appl Math Compt*, 2008, **203**: 62-71.

[7] Wei Z X, Li G Y, Qi L Q. New quasi-Newton methods for unconstrained optimization problems [J]. *Appl Math Comput*, 2006,**175**: 1156-1188.

[8] Nocedal J. Updating quasi-Newton matrices with limited storage [J]. *Math Comp*, 1980, **35**: 773-782.

[9] Liu D C, Nocedal J. On the limited memory BFGS method for large scale optimization [J]. *Math Program*, 1989, **45**: 503-528.

[10] Byrd R H, Nocedal J, Schnabel R B. Representations of quasi-Newton matrices and their use in limited memory methods [J]. *Math Program*, 1994, **63**: 129-156.

[11] Dai Y H, Yuan Y X. Nonliner Conjugate Gradient Methods [M]. Shanghai: Shanghai Science and Technology Press, 2000(in chinese).

[12] Xiao Y H, Wei Z X, Wang Z G. A limited memory BFGS-type method for large-scale unconstrained optimization [J]. *Comput Math Appl*, 2008, **56**: 1001-1009.

[13] Yang Y T, Xu C X. A compact limited memory method for large-scale unconstrained optimization [J]. *European Journal of Operational Research*, 2007, **180**: 48-56.

[14] Byrd R H, Nocedal J, Yuan Y X. Global convergence of a class of quasi-Newton methods on convex problems [J]. *SIAM J Numer Anal*, 1987, **24**: 1171-1190.

[15] Bongartz I, Conn A R, Gould N, et al. CUTE: Constrained and unconstrained testing environment [J]. *ACM Trans Math Softw*, 1995, **20**(1): 123-160.

[16] Li D H, Qi L Q, Roshchina V. A new class of quasi-Newton updating formulas [J]. *Optim Meth Softw*, 2008, **23**: 237-249.

[17] Cheng W Y, Li D H. Spectral scaling BFGS method [J]. *J Optim Theory Appl*, 2010, **146**: 305-319.