Eigenvalue Decomposition Based Modified Newton Algorithm

Wenjun Wang , Jubo Zhu , Xiaojun Duan
National University of Defense Technology
Department of Mathematics and System
Science College, Changsha, 410073, China
e-mail: cammer@tom.com

***Abstract*—When the Hessian matrix is not positive, the Newton direction maybe not the descending direction. A new method named eigenvalue decomposition based modified Newton algorithm is presented, which first takes eigenvalue decomposition on the Hessian matrix, then replaces the negative eigenvalues with their absolutely values, finally reconstruct Hessian matrix and modify searching direction. The new searching direction is always the descending direction, and the convergence of the algorithm is proved and conclusion on convergence rate is presented qualitatively. At last, a numerical experiment is given for comparing the convergence domains of modified algorithm and classical algorithm.**

Keywords-Newton algorithm; eigenvalue decomposition; convergence; convergence domain

I. INTRODUCTION

NEWTON algorithm (also known as Newton's method) is commonly used in numerical analysis, especially in nonlinear optimization[1]. But, in the definition domain of the object function, the method requires: 1) object function $f(\mathbf{x})$ is twice differentiable; 2) the Hessian matrix must be positive; 3) the initial solution $\mathbf{x}^{\mathbf{0}}$ should be near the extreme point $\mathbf{x}^{\mathbf{k}}$ [2]. If there is a point in the definition domain whose Hessian matrix is not positive, then the Newton direction of this point is not the descending direction. So the classical algorithm may not be convergence in the definition domain.

There are several improvements forms of the classical Newton algorithm towards the above problem: 1) Combining of Newton direction with the steepest descent direction[3]; 2) Combining of Newton direction with the negative curvature direction[4]; 3) adding a diagonal matrix $v_k\mathbf{I}$ to the Hessian matrix so that the sum matrix is positive, where v_k is a positive number bigger than the absolutely value of the smallest eigenvalue[5]. This paper will present a new algorithm to handle the "non-positive problem".

II. NEWTON ALGORITHM

This section we discuss the extreme value problems of multi-function. Suppose object function $f(\mathbf{x})$ is a twice differentiable function, and the extreme value problem of $f(\mathbf{x})$ is

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

If the Hessian matrix of $f(\mathbf{x})$ is positive at point $\mathbf{x}^k$, the Taylor expansion of $f(\mathbf{x})$ is:

$$\begin{aligned} f(\mathbf{x}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) \\ + \frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T \nabla^2 f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) \end{aligned} \tag{2}$$

As we know, when $\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{0}$, function $f(\mathbf{x})$ can reach its minimum. So the derivative form of (2) is

$$\begin{aligned} \nabla_{\mathbf{x}} f(\mathbf{x}) = 0 + \nabla f(\mathbf{x}^k) + (\mathbf{x}-\mathbf{x}^k)^T \nabla^2 f(\mathbf{x}^k) \\ = 0 \end{aligned} \tag{3}$$

Equation (3) is usually called Newton's equation. When matrix $\nabla^2 f(\mathbf{x}^k)$ is positive, and the inverse matrix $\left[\nabla^2 f(\mathbf{x}^k)\right]^{-1}$ exists, we can get an iterative equation as follow:

$$\mathbf{x} = \mathbf{x}^k + \left[\nabla^2 f(\mathbf{x}^k)\right]^{-1}\left[-\nabla f(\mathbf{x}^k)\right] \tag{4}$$

Equation (4) is usually called Newton iterative algorithm format, where $-\nabla f(\mathbf{x}^k)$ is called negative gradient direction, $\left[\nabla^2 f(\mathbf{x}^k)\right]^{-1}\left[-\nabla f(\mathbf{x}^k)\right]$ is called Newton direction.

Next, we must give a proper step length to ensure the value of function declining. A well-known algorithm is "Damped Newton Algorithm", which get the step length by solving the following optimization problem:

$$\alpha = \arg\min_{\alpha}\left\{\mathbf{x}^k + \alpha\left[\nabla^2 f(\mathbf{x}^k)\right]^{-1}\left[-\nabla f(\mathbf{x}^k)\right]\right\} \tag{5}$$

III. MODIFIED NEWTON ALGORITHM BASED ON EIGENVALUE DECOMPOSITION

A. Modified algorithm

First, take eigenvalue decomposition on the Hessian matrix $\left[\nabla^2 f(\mathbf{x}^k)\right]^{-1}$:

$$\left[\nabla^2 f(\mathbf{x}^k)\right]^{-1} = \mathbf{U}\Sigma\mathbf{U}^H \tag{6}$$

Where $\mathbf{U}$ is a unitary matrix; Σ is a diagonal matrix. Because matrix $\left[\nabla^2 f(\mathbf{x}^k)\right]^{-1}$ is not positive, so there must be a few negative eigenvalues in Σ. We replace the negative eigenvalues with their absolutely values, so the modified searching direction is:

$$\mathbf{d}^{\mathbf{k}} = \mathbf{U}\left|\Sigma\right|\mathbf{U}^H\left[-\nabla f(\mathbf{x}^k)\right] \tag{7}$$

The inner product of the negative gradient direction with $\mathbf{d}^\mathbf{k}$ is always positive, because the right part of the next equation is a quadratic form[6]:

$$\begin{aligned}&\left\langle \left[-\nabla f(\mathbf{x}^k)\right], \mathbf{U}\left|\Sigma\right|\mathbf{U}^H\left[-\nabla f(\mathbf{x}^k)\right]\right\rangle \\ &=\left[-\nabla f(\mathbf{x}^k)\right]^T \mathbf{U}\left|\Sigma\right|\mathbf{U}^H\left[-\nabla f(\mathbf{x}^k)\right] > 0\end{aligned} \quad (8)$$

Equation (8) shows that the angle between the negative gradient direction and $\mathbf{d}^\mathbf{k}$ is less than 90°, so the new direction $\mathbf{d}^\mathbf{k}$ is always a descent direction. Then we can get the similar iterative equation by replacing the Newton direction with $\mathbf{d}^\mathbf{k}$, that is:

$$\mathbf{x} = \mathbf{x}^k + \alpha\mathbf{d}^\mathbf{k} \quad (9)$$

Where, α is the step length obtained by solving optimization problem (5).

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. If you are using A4-sized paper, please close this template and download the file for A4 paper format called “CPS_A4_format”.

B. Convergence conclusion

We will prove that the modified Newton algorithm based on eigenvalue decomposition is convergent. First, we talk about Wolfe-Powell condition[7].

If step length $\alpha_k > 0$ satisfies the next two equations:

$$f(\mathbf{x}^k + \alpha_k\mathbf{d}^k) - f(\mathbf{x}^k) \le \rho\alpha_k\nabla f(\mathbf{x}^k)^T\mathbf{d}^k \quad (10)$$

$$\nabla f(\mathbf{x}^k + \alpha_k\mathbf{d}^k)^T\mathbf{d}^k \ge \sigma\nabla f(\mathbf{x}^k)^T\mathbf{d}^k \quad (11)$$

Equations (10) and (11) are named Wolfe-Powell condition, where $\rho \in (0,1)$ and $\sigma \in (\rho,1)$.

Theorem 1: Let f be a continuously differentiable function, and let $\hat{\alpha}$ be defined by

$$\begin{aligned}\hat{\alpha} &= \min\{\alpha \ge 0 \mid f(\mathbf{x}^k + \alpha\mathbf{d}^k) \\ &= f(\mathbf{x}^k) + \alpha\rho\nabla f(\mathbf{x}^k)^T\mathbf{d}^k\}\end{aligned}$$

$\hat{\alpha}$ has a bound. Then there must be a finite interval $(a,b) \subseteq [0,\hat{\alpha}]$, $\forall\ \alpha_k \in (a,b)$ the Wolfe-Powell conditions are right.

Proof: let $\phi(t) \triangleq f(\mathbf{x}^k + t\mathbf{d}^k)$, as $\hat{\alpha} > 0$ and satisfies the next equation:

$$\phi(\alpha) \triangleq \phi(0) + \hat{\alpha}\rho\phi'(0) \quad (12)$$

Then

$$\phi(\alpha) < \phi(0) + \hat{\alpha}\rho\phi'(0)\ \ \forall\alpha \in (0,\hat{\alpha}) \quad (13)$$

According to the Mean Value Theorem, exist $\bar{\alpha} \in (0,\hat{\alpha})$ satisfies the following equation:

$$\frac{\phi(\hat{\alpha}) - \phi(0)}{\hat{\alpha}} < \phi'(\bar{\alpha}) < 0 \quad (14)$$

Notice $\rho\phi'(0) = \phi'(\bar{\alpha})$, and $\rho < \sigma$, so we can get:

$$\phi'(\bar{\alpha}) > \sigma\phi'(0) \quad (15)$$

Notice ϕ is a continuously differentiable operator, so there is a open set (a,b) satisfies $(a,b) \subset (0,\hat{\alpha})$, where $\bar{\alpha} \in (a,b)$. For any α_k which $\alpha_k \in (a,b)$, the following equations are right.

$$\phi(\alpha_k) \le \phi(0) + \alpha_k\phi'(0) \quad (16)$$

$$\phi'(\alpha_k) \ge \sigma\phi'(0) \quad (17)$$

Finally, we can get the Wolfe-Powell condition. QED.

Theorem 2: let $f : \mathbf{R}^\mathbf{n} \to \mathbf{R}$ be a continuously differentiable function, so f has lower bounded in $\mathbf{R}^\mathbf{n}$; ∇f is uniformly continuous on O, which contains a horizontal set $Lev_{\alpha_0} := \{\mathbf{x} \in \mathbf{R}^\mathbf{n} \mid f(\mathbf{x}) \le \alpha_0\}$, where $\alpha_0 = f(\mathbf{x}^\mathbf{0})$ and $\mathbf{x}^\mathbf{0} \in \mathbf{R}^\mathbf{n}$. If $\mathbf{d}^k$ satisfies the following conditions: $\theta^k \le \pi/2 - \mu$, where θ^k stands for the angle between $-\nabla f(\mathbf{x}^k)$ and $\mathbf{d}^k$, and $\mu \in (0,\pi/2)$ is a constant, then there must be a k satisfies $\nabla f(\mathbf{x}^k) = \mathbf{0}$ or $\nabla f(\mathbf{x}^k) \to \mathbf{0}$ by Wolfe-Powell conditions.

Proof: We will proof the theorem by contradiction. Suppose $\nabla f(\mathbf{x}^k) \ne \mathbf{0}$ or $\nabla f(\mathbf{x}^k) \not\to \mathbf{0}$ for all k, and there is a set $\{k_i\}$ and a constant $\varepsilon > 0$ satisfy $\left\|\nabla f(\mathbf{x}^{k_i})\right\| \ge \varepsilon$ for all i. Because $\mathbf{d}^k$ satisfies the angle condition $\theta^k \le \pi/2 - \mu$, so we can get:

$$\begin{aligned}\nabla f(\mathbf{x}^k)^T\mathbf{d}^k &= -\left\|\nabla f(\mathbf{x}^k)\right\|\left\|\mathbf{d}^k\right\|\cos\theta_k \\ &\le -\left\|\nabla f(\mathbf{x}^k)\right\|\left\|\mathbf{d}^k\right\|\sin\mu < 0\end{aligned} \quad (18)$$

Equation (18) shows that $\mathbf{d}^k$ is a descending direction, so $\{f(\mathbf{x}^\mathbf{k})\}$ is a decreasing sequence. Suppose the limit of $\{f(\mathbf{x}^\mathbf{k})\}$ is ξ, and $\xi > -\infty$. That is $f(\mathbf{x}^\mathbf{k}) \to \xi$, so $f(\mathbf{x}^\mathbf{k}) - f(\mathbf{x}^{\mathbf{k}+1}) \to 0$, take the first equation of Wolfe-Powell condition on index k_i, then

$$\begin{aligned}f(\mathbf{x}^{k_i}) - f(\mathbf{x}^{k_i+1}) &\ge -\rho\alpha_{k_i}\nabla f(\mathbf{x}^{k_i})^T\mathbf{d}^{k_i} \\ &\ge \left\|\nabla f(\mathbf{x}^{k_i})\right\|\left\|\alpha_{k_i}\mathbf{d}^{k_i}\right\|\cos\theta_{k_i} \\ &\ge \varepsilon\sin\mu\left\|\alpha_{k_i}\mathbf{d}^{k_i}\right\|\end{aligned} \quad (19)$$

So it is easy to get $\left\|\alpha_{k_i}\mathbf{d}^{k_i}\right\| \to 0$, again by the second equation of Wolfe-Powell condition, we can get

$$\left[\nabla f(\mathbf{x}^{k_i+1}) - \nabla f(\mathbf{x}^{k_i})\right]^T\mathbf{d}^{k_i} \ge -(1-\sigma)\nabla f(\mathbf{x}^{k_i})^T\mathbf{d}^{k_i} \quad (20)$$

Further

$$\left\|\nabla f(\mathbf{x}^{k_i+1}) - \nabla f(\mathbf{x}^{k_i})\right\| \ge \varepsilon(1-\sigma)\sin\mu \quad (21)$$

Because of $\left\|\mathbf{x}^{k_i+1} - \mathbf{x}^{k_i}\right\| = \left\|\alpha_{k_i}\mathbf{d}^{k_i}\right\| \to 0$, equation (21) is contradiction with the assumption that ∇f is uniformly continuous on Lev_{α_0}. The contradiction suggest that $\nabla f(\mathbf{x}^k) \to \mathbf{0}$. QED.

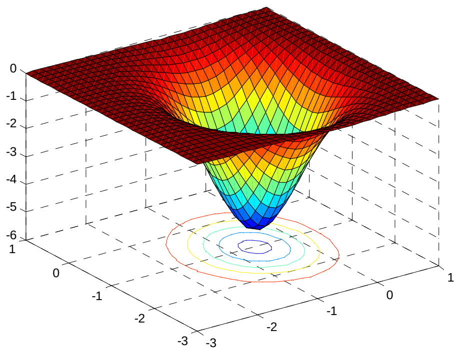
IV. SIMULATIONS ON CONVERGENCE

As the new searching direction $\mathbf{d}^k$ satisfies the following equation:

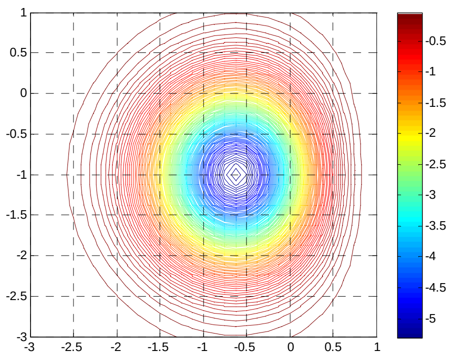
$$\mathbf{d}^{\mathbf{k}} = \begin{cases} \mathbf{U}\Sigma\mathbf{U}^H\left[-\nabla f(\mathbf{x}^k)\right] & if\ \ \nabla^2 f(\mathbf{x}^k) > 0 \\ \mathbf{U}\left|\Sigma\right|\mathbf{U}^H\left[-\nabla f(\mathbf{x}^k)\right] & else \end{cases} \quad (22)$$

So, if the Hessian matrix of all points in definition domain is positive, then the convergence rate of modified algorithm equals to the classical Newton algorithm. The other side, if there are any points with its Hessian matrix negative, then the convergence rate at most equals to the one of classical Newton algorithm.

In addition, in order to verify the improved algorithm has a wider convergence domain than the classical Newton algorithm, we employ a convex function: $f(x,y) = -3(1-x)^2 e^{-x^2-(y+1)^2}$ as the objective function. Let the definition domain is $(x,y) \in \{[-3,1],[-3,1]\}$, and we choose the initial point by 0.1 interval, then use both classical Newton's algorithm and eigenvalue decomposition based algorithm to find the minimum value. The results is drawn in the following figures.

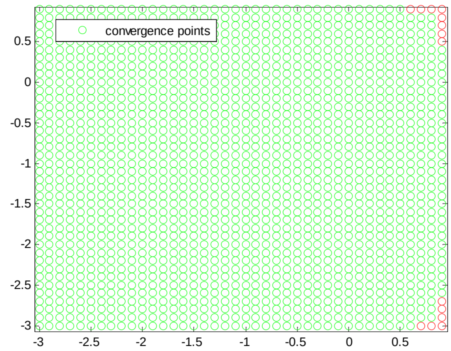


(a) Objective function $f(x,y) = -3(1-x)^2 e^{-x^2-(y+1)^2}$;



(b) Contour map;

(c) Convergence domain of Newton's algorithm;



(d) Convergence domain of EVD based algorithm.

Fig. 1 Results of Numerical Experiment

In fig 1-(c), the green point means that the algorithm will find the minimum value using this point as the initial solution; and the red point means that the algorithm will not converge or will not find the minimum value. Obviously, the convergence domain of eigenvalue decomposition algorithm is bigger than the one of the classical Newton algorithm.

V. CONCLUSION

A new algorithm is put forward base on eigenvalue decomposition and classical Newton algorithm. The eigenvalue decomposition based algorithm has a wider convergence domain than the classical Newton algorithm.

ACKNOWLEDGMENT

First, I should give my best regard to Professor Ju-bo Zhu who usually tell me a big success always exist in a little problem. Also I appreciate all the teachers in the department of mathematics and systems science and the members in group 4202, who gave us many useful suggestions. Thank you!

This work is supported by the Natural Science Foundation of China (60974124), the Program for New Century Excellent Talents in University and the Project-sponsored by SRF for ROCS, SEM in China, the key lab open foundation for Space Flight Dynamics technique (SFDLXZ-2010-004).

REFERENCES

[1] D. P. Bertsekas. Nonlinear Programming. Athena Scientific, Belmont, Massachussetts, 1999.

[2] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge University Press, Cambridge, UK, 2004.

[3] A. Goldstein and J. F. Price, On Descent from Local Minima, Mathematics of Computation, Volume 25, No. 115, July, 1971,pp:569-574.

[4] AV. Fiacco, GP. McCormick, The sequential unconstrained minimization technique for nonlinear--a primal-dual method, Management Science, Vol 10, No.2, 1964, pp:360-365.

[5] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York-London, 1970.

[6] R. Horn and C. R. Johnson, Matrix analysis, Cambridge University, Press, New York, 1985.

[7] M. J. D. Powell, An e_cient method for _nding the minimum of a function of several variables without calculating derivatives, The Computer Journal, 7, pp.155-162, 1964.