On the Application of Three-Term Conjugate Gradient Method in Regression Analysis

Aliyu Usman Moyi
Department of Mathematics and Statistics
Abdu Gusau Polytechnic,
Talata Mafara, Zamfara State, Nigeria

Wah June Leong
Department of Mathematics
University Putra Malaysia,
Serdang, Selangor, Malaysia

Ibrahim Saidu
Department of Science
College of Agriculture Zuru,
Kebbi State ,Nigeria

ABSTRACT

Conjugate gradient methods have played a useful and powerful role for solving large-scale optimization problems which has become more interesting and essential in many disciplines such as in engineering, statistics, physical sciences, social and behavioral sciences among others. In this paper, we present an application of a proposed three-term conjugate gradient method in regression analysis. Numerical experiments show that the proposed method is promising and superior to many well-known conjugate gradient methods in terms of efficiency and robustness.

Keywords:

Unconstrained Optimization, Three-term conjugate gradient method, symmetric rank-one update, regression analysis

1. INTRODUCTION

This paper concerns the conjugate gradient methods for the numerical solution of the following unconstrained optimization problem

$$\min f(x); x \in R^n \tag{1}$$

where $f : R^n \to R$ is continuously differentiable function and n is the dimension of x, which is assumed to be large. The iterates of the conjugate gradient methods are obtained by

$$x_{k+1} = x_k + \lambda_k d_k, \tag{2}$$

where d_k is the search direction and $\lambda_k > 0$ is the step length. The step length can be calculated by an exact line search :

$$\lambda_k^* = argmin_{\lambda \in R}\{f(x_k + \lambda_k d_k)\} \tag{3}$$

or by some line search strategies such as the Armijo condition;

$$f(x_k + \lambda_k d_k) \le f(x_k) + c_1 \lambda_k g_k^T d_k \tag{4}$$

for some constant $c_1 \in (0, 1)$, where $g_k = \nabla f(x_k)$ denotes the gradient vector of $f(x)$ at the current iterate point x_k. These methods also define the search direction d_k by

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0 \tag{5}$$

for $k \ge 1$, where the parameter $\beta_k \in R$ is a scaler known as conjugate gradient coefficient.
It is well known that regression analysis often arises in economics, finance, trade, meteorology, medicine biology,chemistry physics and so on (see for example [1],[2],[3] and the references therein). The classical regression model is defined by

$$Y = h(X_1, X_2, ..., X_p + \varepsilon) \tag{6}$$

where Y is the response variable, X_i is the predictor variable, $i = 1, 2, , ...p, p > 0$ is an integer constant, and ε is the error term. The function $h(X_1, X_2, ..., X_p)$ explain the type of relationship that exist between Y and $X = (X_1, X_2, ..., X_p)$. Thus we obtain the following linear regression model when h is a linear function

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon \tag{7}$$

which is the simplest regression model, where $\beta_0, \beta_1, ..., \beta_p$ are the regression parameters. The most important task in regression analysis is to estimate the parameter $\beta = (\beta_0, \beta_1, ..., \beta_p)$ and the method of least squares is an important method to determine the parameters which is defined by

$$\min_{\beta \in R^{p+1}} S(\beta) = \sum_{i=1}^{m} (h_i - \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip})^2 \tag{8}$$

where h_i is the data valuation of the ith response variable, $X_{i1}, X_{i2}, ..., X_{ip}$ are p data valuation of ith predictor variable, and m is the number of the data. If the dimension of p and the number m is small, then we can obtain the parameters $\beta = (\beta_0, \beta_1, ..., \beta_p)$ from extreme value of calculus and thus it is not difficult to see that problem (8) is the same as the unconstrained optimization problem (1). The regression parameters of interest are estimated by the least squares method if the dimension of the parameters is small and can be transformed in to unconstrained optimization problems . The numerical optimization techniques to be employed includes the steepest descent method, Newton method, quasi-Newton methods or conjugate gradient methods in finding the solution to such given practical optimization problems.

2. THE DERIVATION PROCESS

The idea of memoryless quasi-Newton method of Perry [7] in which H_{k+1} is updated from $H_k = \theta_k I$ on every iteration is considered. Applying this idea of memoryless scheme to symmetric rank-one (SR1) update, we can obtain the search direction without the computation and storage of matrices which gives immediately the following:

$$H_{k+1} g_{k+1} = \theta_k g_{k+1} + \frac{(s_k - \theta_k y_k)(s_k - \theta_k y_k)^T}{y_k^T (s_k - \theta_k y_k)} g_{k+1},$$

$$= \theta_k g_{k+1} + (s_k - \theta_k y_k)\alpha_k, \quad (9)$$

and

$$d_{k+1} = -H_{k+1}g_{k+1} = -\theta_k g_{k+1} - \alpha_k s_k + \theta_k \alpha_k y_k,$$
$$= -\theta_k g_{k+1} - \alpha_k s_k + \delta_k y_k. \quad (10)$$

where

$$\alpha_k = \frac{(s_k - \theta_k y_k)^T}{y_k^T(s_k - \theta_k y_k)} g_{k+1}, \quad (11)$$

$$\delta_k = \theta_k \alpha_k, \quad (12)$$

$$\theta_k = \frac{\omega_k}{\upsilon_k} - \left\{ \frac{\omega_k^2}{\upsilon_k^2} - \frac{\omega_k}{\eta_k} \right\}^{1/2}. \quad (13)$$

We now present the basic steps of our proposed line search algorithm here represented as TTCG-SR1

Algorithm TTCG-SRI

Step 1 : Given an initial point $x_0 \in R^n$, set $k = 0$ and $d_0 = -g_0$

Step 2 : Test a criterion for stopping the iterations. If the test is satisfied, then stop; otherwise continue with step 3

Step 3 : Compute the search direction d_k by (10)-(12), with θ_k defined in (13)

Step 4 : Find an acceptable steplength λ_k, by using the following line search procedure. Given the constants $\eta \in (0, 1)$ and τ, τ' with $0 < \tau < \tau' < 1$

(i) set $\lambda = 1$

(ii) Test the relation

$$f(x_k + \lambda d_k) \leq f(x_k) + \eta \lambda g_k^T d_k, \quad (14)$$

(iii) If (14) is not satisfied, choose a new λ in $[\tau\lambda, \tau'\lambda]$ and go to (ii). If (14) is satisfied, set $\lambda_k = \lambda$ and $x_{k+1} = x_k + \lambda_k d_k$

Step 5 : Set $k := k + 1$, and go to step 2

3. DESCRIPTION OF THE PROBLEMS

In this section, the detailed description of the problems considered are given below, these problems were obtained from the papers of [8] and [9].

Problem 1. In the table below, there is data of some kind of commodity between year demand and price:

Table 1. Data of demand and price

Price p_i(RM)	1	2	2	2.3	2.5	2.6	2.8
Demand $d_i(500g)$	5	3.5	3	2.7	2.4	2.5	2
Price p_i(RM)	3	3.3	3.5				
Demand $d_i(500g)$	1.5	1.2	1.2				

From the statistical point of view we can infer that there will be possible change in the demand even though the price is inconvenient and the demand will be possible invariably albeit the price changes. In summary, there will be decrease in the demand with the increase in the price and our primary objective is to determine the approximate function between the demand and the price, that is the regression equation of d to p.

From the given data above, one can observe that there exists a linear relationship between the demand and the price, with the regression equation given by $\hat{d} = \beta_0 + \beta_1 p$, with β_0 and β_1 denoting the regression parameters. Solving the above regression equation entails finding the value of β_0 and β_1 by the method of least squares that minimized the problem

$$\min Q = \sum_{i=0}^{n} [d_i - (\beta_0 + \beta_1 p_i)]^2, \quad (15)$$

we can now transform the above least squares problem in to an unconstrained optimization problem as

$$\min_{x \in R^2} f(\beta) = \sum_{i=1}^{n} [d_i - \beta(1, p_i)^T]^2. \quad (16)$$

Problem 2. The table below gives the data of the age x and the average height H of a pine tree:

Table 2. Data of the age and average height

x_i	2	3	4	5	6	7	8
h_i	5.6	8	10.4	12.8	15.3	17.8	19.9
x_i	9	10	11				
h_i	21.4	22.4	23.2				

From the above problem, careful observation reveals that the age x and the average height H have parabolic relations with the regression function defined by $\hat{h} = \beta_0 + \beta_1 x + \beta_2 x^2$, where β_0, β_1 and β_2 are the regression parameters. Similar to the problem 1 above, we can use the method of least squares to solve the problem as

$$\min Q = \sum_{i=0}^{n} [h_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2)]^2. \quad (17)$$

Similar transformation of the above least squares problem in to an unconstrained optimization problem yields the following as

$$\min_{x \in R^3} f(\beta) = \sum_{i=1}^{n} [d_i - \beta(1, x_i, x_i^2)^T]^2. \quad (18)$$

Solving the above problems (16) and (18) using the method of extreme value of calculus yields the solutions $\beta^* = (6.5, -1.6)$, and $\beta^* = (-1.33, 3.46, -0.11)$, respectively. In this context, we employ our proposed three-term conjugate gradient via the symmetric rank-one update method to solve these problems in order to assess the performance of the method in solving regression problems in comparison with the method of extreme value of calculus or other software and some other well-known conjugate gradient methods.

4. NUMERICAL RESULTS AND DISCUSSION

This section is devoted to the application of our proposed sufficient descent three-term conjugate gradient method via the symmetric rank-one method in comparison with:

(i) PR+: Gilbert and Nocedal [4].

(ii) HS: Hestenes and Steifel [6] .

(iii) CG-DESCENT method by Hager and Zhang [5].

(iv) MPRP: A three-term conjugate gradient descent modified Polak-Ribière-Polyak by zhang et al. [10].

(v) " NA 1" standing for Algorithm 1, obtained from Yuan and Wei [8] and the method is implemented with the nonmonotone Wolfe line search rule.

All the experiments are implemented on a PC using MATLAB 7.9.0 (R2009b), with double precision arithmetic. For each test problem, we performed ten numerical experiments with different initial guess in order to evaluate the efficiency of the methods. As regards the stopping criteria used in our experiments, in all the algorithms, convergence is assumed if $\|g_k\| \leq \varepsilon$ where $\varepsilon = 10^{-4}$. We forced the algorithm to stop whenever the number of iterations exceeds 2000, and the symbol "-" is used to represent the failure.

The columns in Table 3 and 4 below has the following meaning:
β^*: the approximate solution from the method of extreme value of calculus or some software.
$\grave{\beta}$: the solution as the programme is terminated.
$\breve{\beta}$: the initial point.
ε^*: the relative error between β^* and $\grave{\beta}$ defined by $\varepsilon^* = \frac{\|\grave{\beta}-\beta^*\|}{\|\beta^*\|}$

The following different initial points are those chosen by [8] and [9] and are adopted here to test the efficiency of our algorithm in Problem 1:
$\breve{\beta}_1$=(1,-0.01), $\breve{\beta}_2$=(-10,0.04), $\breve{\beta}_3$=(-2,-1.0), $\breve{\beta}_4$=(15,15), $\breve{\beta}_5$=(-10,100) $\breve{\beta}_6$=(500,1000), $\breve{\beta}_7$=(-100,-100), $\breve{\beta}_8$=(2,3), $\breve{\beta}_9$=(-2,-3), $\breve{\beta}_{10}$=(-0.001,0.001)
Furthermore, the initial points below are chosen to observe the performance behaviour of our algorithm in solving Problem 2 in comparison with other conjugate gradient methods as follows;
$\breve{\beta}_1$=(1.1,3.0,-0.5), $\breve{\beta}_2$=(-1.2,3.2,-0.3), $\breve{\beta}_3$=(-0.003,7.0,-0.8),;$\breve{\beta}_4$=(-0.001,7.0,-0.5) $\breve{\beta}_5$=(100,100,100), $\breve{\beta}_6$=(0,0,0), $\breve{\beta}_7$=(-10,-100,-1000), $\breve{\beta}_8$=(10,-100,1000) ,$\breve{\beta}_9$=(1,2,3), $\breve{\beta}_{10}$=(0.1,-0.3,0.8)

The numerical results of Tables 3 and 5 show the performance of all the algorithms considered in these applications, the numerical results indicates that our algorithm has made significant performance among these algorithms. The performance from the implementation of problem 1 indicates that irrespective of the different initial points, our proposed method was able to solve the problem, though its performance is competitive with PR+, CG-DESCENT and NA 1, its outperformed MPRP method which failed to solve the problem in four different initial points within 2000 iterations. Similarly, the performance of these algorithms on problem 2 shows that 3TCG-SR1 has potential advantages over the other algorithms since for all the different initial points considered it has successfully solve the problem with more notable results than that of NA1 while PR+, HS, CG-DESCENT and MPRP methods have all failed to solve this problem. Overall, the result of our proposed method has more precision than from the other algorithms and the method of extreme value of calculus or other software.

5. CONCLUSION

In this paper , we applied the three-term conjugate gradient algorithm via the symmetric rank-one method proposed to solve real-life regression problems. Numerical results from the experiments conducted using the TTCG-SR1 algorithm has indicated that our proposed method is competitive and notable in comparison with the other existing algorithms employed in the implementation. Hence, we can conclude that our algorithm is successful for the test problems considered and can further be employed for application in regression analysis especially for large scale problems.

6. REFERENCES

[1] Bates, D. M. and Watts,D. g. (1988). Non linear regeression analysis and its applications. *New York, John Wiley & Sons*

[2] Chatterjee, M. and Machler, M. (1997). Robust regression: A weighted least squares approach *Communications in Statistics-Theory and Methods* 26: 1381–1394.

[3] Christensen, R. (1996). Analysis of variance, design and regression: Applied Statistical Methods *New York, Chapman and Hall*.

[4] Gilbert, J. C. and Nocedal, J. (1992). Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization* 2 (1): 21–42.

[5] Hager, W. W. and Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization* 16 (1): 170–192.

[6] Hestenes, M. R. and Stiefel, E. (1952). *Methods of conjugate gradients for solving linear systems*. Journal of Research of the National Bureau of Standards, 49: 409-436.

[7] Perry, J. M. (1977). A class of conjugate gradient algorithms with a two step variable metric memory. *Center for Mathematical Studies in Economies and Management Science. Evanston Illiois: Northwestern University Press* .

[8] Yuan, G. and Wei, Z. (2009). New line search methods for unconstrained optimization. *Journal of the Korean Statistical Society* 38 (1): 29–39.

[9] Yuan, G. and Wei, Z. (2013). Non Monotone Backtracking Inexact BFGS Method for Regression Analysis. *Communications in Statistics-Theory and Methods* 42 (2): 214–238.

[10] Zhang, L., Zhou, W. and Li, D.(2006) A descent modified Polak-Ribiere-Polyak conjugate gradient method and its global convergence. *IMA Journal of Numerical Analysis* 26 (4): 629–640.

Table 3. Numerical results for problem 1

β^*=(6.5,-1.6)	3TCG-SR1	PR+	CG-DESCENT	MPRP	NA1
Initial points	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$
$\breve{\beta}_1$	(6.438282,-1.575313)	(6.438305,-1.575321)	(6.438261,-1.575305)	(6.438270,-1.575308)	(6.438301,-1.575289)
ε^*	0.009930	0.009926	0.009933	0.009932	0.009929
$\breve{\beta}_2$	(6.438288,-1.575315)	(6.438299,-1.575320)	((6.438344,-1.575336)	-	(6.438280,-1.575313)
ε^*	0.009929	0.009927	0.009920	-	0.009930
$\breve{\beta}_3$	(6.438240,-1.575297)	(6.438289,-1.575315)	(6.4382620,-1.575290)	(6.438212,-1.575287)	(6.438285,-1.575314)
ε^*	0.009937	0.009929	0.009934	0.009941	0.009930
$\breve{\beta}_4$	(6.438325,-1.575329)	(6.438273,-1.575310)	(6.438348,-1.575338)	(6.438283,-1.575313)	(6.438287,-1.575316)
ε^*	0.009923	0.009932	0.009920	0.009930	0.009929
$\breve{\beta}_5$	(6.438290,-1.575316)	(6.438280,-1.575313)	(6.438343,-1.575335)	-	(6.438285,-1.575314)
ε^*	0.009929	0.009930	0.009920	-	0.009930

Table 4. Continuation 1 of Table 3

β^*=(6.5,-1.6)	3TCG-SR1	PR+	CG-DESCENT	MPRP	NA1
Initial points	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$
$\breve{\beta}_6$	(6.438257,-1.575303)	(6.438313,-1.575324)	(6.438277,-1.575311)	(6.438329,-1.575331)	(6.438285,-1.575314)
ε^*	0.009934	0.009925	0.009931	0.009923	0.009930
$\breve{\beta}_7$	(6.438268,-1.575307)	(6.438291,-1.575316)	(6.438339,-1.575334)	(6.438263,-1.575305)	(6.438285,-1.575314)
ε^*	0.009932	0.009929	0.009921	0.009933	0.009930
$\breve{\beta}_8$	(6.438259,-1.575304)	(6.438255,-1.575303)	(6.438267,-1.575308)	(6.438290,-1.575316)	(6.438285,-1.575314)
ε^*	0.009933	0.009934	0.009932	0.009929	0.009930
$\breve{\beta}_9$	(6.438270,-1.575308)	(6.438324,-1.575329)	(6.438337,-1.575333)	-	(6.438285,-1.575314)
ε^*	0.009932	0.009923	0.009921	-	0.009930
$\breve{\beta}_{10}$	(6.438275,-1.575310)	(6.438274,-1.575310)	(6.438330,-1.575331)	-	(6.438285,-1.575314)
ε^*	0.009931	0.009931	0.009922	-	0.009930

Table 5. Numerical results for problem 2

β^*=(-1.33,3.46,-0.11)	3TCG-SR1	PR+	HS	CG-DESCENT	MPRP	NA1
Initial points	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$
$\breve{\beta}_1$	(-1.331261,3.461709,-0.108710)	-	-	-	-	(-1.296574,3.450247,-0.107896)
ε^*	0.000664	-	-	-	-	0.009407
$\breve{\beta}_2$	(-1.331294,3.461721,-0.108712)	-	-	-	-	(-1.328742,3.460876,-0.108650)
ε^*	0.000667	-	-	-	-	0.000551
$\breve{\beta}_3$	(-1.331325,3.461729,-0.108711)	-	-	-	-	(-1.328504,3.460798,-0.108646)
ε^*	0.000682	-	-	-	-	0.000585
$\breve{\beta}_4$	(-1.331291,3.461718,-0.108710)	-	-	-	-	(-1.321726,3.458558,-0.108483)
ε^*	0.000676	-	-	-	-	0.002301
$\breve{\beta}_5$	(-1.331339,3.461735,-0.108712)	-	-	-	-	(-1.331363,3.461742,-0.108712)
ε^*	0.00686	-	-	-	-	0.000690

Table 6. Continuation 1 of Table 5

β^*=(-1.33,3.46,-0.11)	3TCG-SR1	PR+	HS	CG-DESCENT	MPRP	NA1
Initial points	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$	$\grave{\beta}$
$\breve{\beta}_6$	(-1.331374,3.461747,-0.108712)	-	-	-	-	(-1.331363,3.461742,-0.108712)
ε^*	0.000693	-	-	-	-	0.000690
$\breve{\beta}_7$	(-1.331388,3.461751,-0.108713)	-	-	-	-	(-1.331363,3.461742,-0.108712)
ε^*	0.000695	-	-	-	-	0.000690
$\breve{\beta}_8$	(-1.331388,3.461750,-0.108713)	-	-	-	-	(-1.331363,3.461742,-0.108712)
ε^*	0.000695	-	-	-	-	0.000690
$\breve{\beta}_9$	(-1.331372,3.461746,-0.108712)	-	-	-	-	(-1.331363,3.461742,-0.108712)
ε^*	0.000692	-	-	-	-	0.000690
$\breve{\beta}_{10}$	(-1.331442,3.461768,-0.108714)	-	-	-	-	(-1.331363,3.461742,-0.108712)
ε^*	0.000706	-	-	-	-	0.000690