Available online at www.elixirpublishers.com (Elixir International Journal)

Applied Mathematics

Elixir Appl. Math. 38 (2011) 4205-4207

ISSN: 2229-712X

Artificial neural network with trust region strategy for parameter estimation

Shan Rui, Gao Donglian, Gao Jinghui, Wang Shuhua and Li Lingling

College of Science, University of Yanshan, Qinhuangdao, 066004, China.

ARTICLE INFO

Article history:
Received: 3 July 2011;
Received in revised form:
23 August 2011;
Accepted: 27 August 2011;

Keywords
Time Series,
ARMA model,
Parameter estimation,
Neural network,
Trust region algorithm,
MSC: 90C30; 65K05 Chinese
Library Classification: O224; TP183.

ABSTRACT

In this paper, we present a novel algorithm for time series ARMA parameter estimation, namely, artificial neural network with trust region strategy. It combines the merit of neural network which has the capacity of highly parallel computing with the global convergence of trust region algorithm. The convergence of the algorithm is proved under certain conditions. It offers high accuracy for the parameter value of the ARMA model and makes the model is more superior. Numerical experiment shows that the new method is effective and attractive.

© 2011 Elixir All rights reserved.

Introduction

It is very important for parameter estimation after time series ARMA model is determined. Parameter estimation plays a key role in establishing proper model. It effects not only the adaptability but also predicting results. Parameter estimate has an important position and has attracted many scholars' attention to studying ARMA parameter estimation [1]. The method of parameter estimation can be roughly classified into three kinds [2].

One is developed from timing theory itself, called timing theory parameters estimation method of ARMA model. Another kind is iterative algorithm of optimization theory, called optimization theory parameters estimation method of ARMA model. The third type is the difference model of control theory, called control theory parameters estimation method of ARMA mode. This paper discusses the optimization theory parameters estimation method and presents a novel algorithm for time series ARMA parameter estimation, denoted by neural networks with trust region strategy. The speed is accelerated and convergence properties of the method are proved under certain conditions. Hence, it improves the prediction performance of ARMA model.

ARMA model

Suppose the date $\{x_t\}$ $(t=1,2,\Lambda,n)$ are observations from the causal invertible ARMA (p, q) process. Generally, ARMA (p,q) can be described as follows [3].

$$x_t=\varphi_1 x_{t-1}+\varphi_2 x_{t-2}+\Lambda+\varphi_p x_{t-p}+\varepsilon_t-\theta_1\varepsilon_{t-1}-\theta_2\varepsilon_{t-2}-\Lambda-\theta_q\varepsilon_{t-q} \quad (1)$$

where $\varphi_p,\theta_q\neq 0$, $\{\varepsilon_t\}$ is white noise with zero-mean. The roots of $\Phi(B)=1-\varphi_1 B-\varphi_2 B^2-\varphi_p B^p$ and $\Theta(B)=1-\theta_1 B-\theta_2 B^2-\Lambda-\theta_q B^q$ are out of the unit circle and have no common factor, which ensure the Sequence is stable and reversible. Set

$$\beta=[\beta_1,\beta_2,\Lambda,\beta_n]^T=[\varphi_1,\varphi_2,\Lambda,\varphi_p,\theta_1,\Lambda,\theta_q]^T,$$

$$X_t=[x_{t-1},x_{t-2},\Lambda,x_{t-p},\varepsilon_{t-1},\Lambda,\varepsilon_{t-q}]^T.$$

Then formula (1) can be denoted by $x_t=X_t^T\beta+\varepsilon_t$. $\varepsilon_{t-1},\varepsilon_{t-2},\Lambda,\varepsilon_{t-q}$ is the function of $x_{t-1},x_{t-2},\Lambda,x_{t-p}$, so β is nonlinear. Let f expresses the relation between X_t and β, then $x_t=\varepsilon_t+f(X_t,\beta)$.

Residual sum of squares of ARMA model is defined by

$$S(\beta)=\sum_{t=p+1}^{n}\varepsilon_t^2=\sum_{t=p+1}^{n}\lfloor x_t-f(X_t,\beta)\rfloor^2 \qquad (2)$$

Parameter estimation is to minimize $S(\beta)$ and obtains the solution of $S(\beta)$, that is, $\min S(\beta)$.

Neural Networks with Trust Region Strategy for Parameter Estimation

Algorithm Process

The trust region method is a kind of iterative method; a trial step is usually executed by solving the following quadratic model, called the trust region subproblem [4],

$$\min q^k(\beta)=S(\beta_k)+g_k^T d+\frac{1}{2}d^T B_k d \qquad (3)$$
$$s.t. \quad \|d\|\le\Delta_k$$

Where $d=\beta-\beta_k$, $g_k=\nabla S(\beta_k)$, B_k is an $n\times n$ symmetric matrix which is the Hessian or its approximation of $S(\beta)$ at the current iterate point, $\Delta_k>0$ is called the trust region radius, and $\|\bullet\|$ refers to the Euclidean norm. Solving (3) is a key work in the trust region method. Many authors have studied the problem and they have proposed a lot of methods [5-6]. Generally, it is costly, especially when B_k is large scale and

Tele:
E-mail addresses: 583047360@qq.com

© 2011 Elixir All rights reserved

dense.

Here, we present a new method, namely, neural networks with trust region strategy to solve the trust region subproblem. Artificial neural networks [7-8] have the merits of their rapidness and accuracy. Using relatively simple neural network architecture which has ability of high parallel computing, even relatively complex optimization problems can also be real-time solved.

In formula (3), using l_2 norm, the constrained condition equals to $d^T d \ -\Delta_k^2 \leq 0$, then (3) is

$$\begin{cases} \min q^k(d) = S(\beta_k) + g_k^T d + \dfrac{1}{2} d^T B_k d \\ s.t. \qquad d^T d - \Delta_k^2 \leq 0 \end{cases} \tag{4}$$

Based on the augmented Lagrange multiplier method, Lagrange function of (4) is as follows.

$$L(d,\lambda) = S(\beta_k) + g_k^T d + \frac{1}{2} d^T B_k d + \lambda \max\{0, d^T d - \Delta_k^2\} + \frac{K}{2} \max\{0, d^T d - \Delta_k^2\}^2 \tag{5}$$

where λ is the Lagrange multiplier, K is punishment parameter.

(5) can be written to a more compact form

$$L(d,\lambda) = S(\beta_k) + g_k^T d + \frac{1}{2} d^T B_k d + S\left[\lambda(d^T d - \Delta_k^2) + \frac{K}{2}(d^T d - \Delta_k^2)^2\right] \tag{6}$$

where

$$S = \begin{cases} 0 & d^T d - \Delta_k^2 \leq 0 \\ 1 & d^T d - \Delta_k^2 > 0 \end{cases}$$

The renewal equation of neural networks is as follows.

$$d(k+1) = d(k) - \mu_d \frac{\partial L(d,\lambda)}{\partial d} \tag{7}$$

$$\lambda(k+1) = \lambda(k) + \mu_\lambda \frac{\partial L(d,\lambda)}{\partial \lambda} \tag{8}$$

where $\mu_d, \mu_\lambda > 0$ denote learning rate parameters.

$$\frac{\partial L(d,\lambda)}{\partial d} = g_k + B_k d + S(\lambda + K(d^T d - \Delta_k^2))d \tag{9}$$

$$\frac{\partial L(d,\lambda)}{\partial \lambda} = S(d^T d - \Delta_k^2) \tag{10}$$

Hence, the motion equation of neural networks is

$$d(k+1) = d(k) - \mu_d \left\{g_k + B_k d(k) + S\left[\lambda + K(d(k)^T d(k) - \Delta_k^2)\right] d(k)\right\} \tag{11}$$

$$\lambda(k+1) = \lambda(k) + \mu_\lambda S(d(k)^T d(k) - \Delta_k^2) \tag{12}$$

When $\|d_k\| < \varepsilon$ then stop, $d(k+1)$ is the solution of (11), it is also the solution of (3). Or the motion equation of neural networks will be updated until it satisfies the condition.

At point β_k, let d_k be the solution of (3). The actual reduction of the objection function is defined by

$$Ared_k = S(\beta_k) - S(\beta_k + d_k) \tag{13}$$

The predictive reduction is defined by

$$\Pr ed_k = q^k(0) - q^k(d_k) \tag{14}$$

The ratio between these two reductions is defined by

$$r_k = \frac{Are_k}{\Pr ed_k} = \frac{S(\beta_k) - S(\beta_k + d_k)}{q^k(0) - q^k(d_k)} \tag{15}$$

As is known, the value of r_k plays a key role in the trust region method to decide whether the trial step d_k is accepted and to adjust the trust region radius. If d_k is successful, one accepts the trial step and enlarges the trust region radius; otherwise, one rejects the trial step, reduces the trust region radius and resolves the subproblem.

Algorithm Model

In this section, we will present the model of the new method.

Algorithm 3.2.1

Step 1

Given $\beta_0, \Delta_o \in (0, \Delta)$, $\varepsilon > 0$, $0 < \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$, $k = 0$;

Step 2 If $\|g_k\| \leq \varepsilon$,then stop; else go to step 3;

Step 3 Solve (11) giving d_k.

Step 4 Compute r_k by (13)-(15).

Step 5 Update β_k, Δ_k by r_k .Set

$$\beta_{k+1} = \begin{cases} \beta_k + d_k, & r_k \geq \eta_1 \\ \beta_k \quad , & r_k < \eta_1 \end{cases}$$

$$\Delta_{k+1} = \begin{cases} \gamma_2 \Delta_k, & r_k \geq \eta_2 \\ \Delta_k \quad , & \eta_1 \leq r_k < \eta_2 \\ \gamma_1 \Delta_k, & r_k < \eta_2 \end{cases}$$

Step 6 Compute $y_k = g_{k+1} - g_k$,if $y_k = 0$,then stop and output optimal solution β_{k+1} ; else update B_k by BFGS

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T d_k} - \frac{B_k d_k d_k^T B_k}{d_k^T B_k d_k}$$

Step 7 Compute $S(\beta_{k+1})$,if $|S(\beta_{k+1})| < \varepsilon$,then stop; else set $k = k+1$, go to step 2.

Convergences

Before addressing some theoretical issues, we would like to make the following hypotheses.

H1: Hessian or its approximation B_k is bounded, that is, there exists a positive constant M, such that $\|B_k\| \leq M$.

H2: The level set $L(\beta) = \{\beta | S(\beta) \leq S(\beta_0)\}$ is bounded for any given β_0 and $S(\beta)$ is continuously differentiable in $L(\beta)$ for any given β_0 and bounded.

H3: $\|d_k\| \leq \delta \Delta_k$ for a constant δ .

Lemma 4.1 *For the solution of (3), we have*

$$q^k(0) - q^k(d_k) \geq \alpha_1 \|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}$$

where $\alpha_1 \in (0,1]$.

Lemma 4.2 *Suppose that H1-H3 hold and* $g_k \neq 0$, $\Delta_k \leq \Delta'$, Δ' *is a small finite, then* $\Delta_{k+1} \geq \Delta_k$.

Theorem 4.3 *Assume that H1-H3 hold, then sequence* g_k generated by Algorithm 3.2.1 satisfies

$$\lim_{k\to\infty}\inf\|g_k\| = 0 \qquad (16)$$

Proof: We prove by contradiction. If (16) is not true, there exists $\varepsilon > 0$ and a positive index K, such shat

$$\|g_k\| \geq \varepsilon, \qquad \forall k \geq K$$

First, we assume that there exists infinite successful iteration, then, by Algorithm 3.2.1 and Lemma 4.1, we have

$$\begin{aligned} S(\beta_k) - S(\beta_k + d_k) &\geq \eta_1[q^k(0) - q^k(d_k) \\ &\geq \eta_1\,\alpha_1\|g_k\|\min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\} \\ &\geq \eta_1\alpha_1\varepsilon\min\left\{\Delta_k, \frac{\varepsilon}{M}\right\} \end{aligned}$$

Because $S(\beta)$ is bounded, which implies

$$\lim_{k\to\infty}\Delta_k = 0$$

which gives a contradiction to Lemma 4.2.

Then we assume that there only exists limited successful iteration, then we reduce the trust region radius, i.e. $\Delta_k \to 0$, which also gives a contradiction to Lemma 4.2. The proof is completed.

Numerical Experiments

In this section, we implement the new Algorithm 3.2.1 and compare it with Gauss-Newton algorithm for time series ARMA parameter estimation. The test problem comes from [1] and is written in MATLAB 7.0.

We generate the ARMA（2,2）as follows with Monte Carlo method.

$$x_t = -1.45x_{t-1} - 0.6x_{t-2} + \varepsilon_t + 0.5\varepsilon_{t-1} - 0.3\varepsilon_{t-2}$$

We generate the dates using the above model and estimate the parameters with the new method proposed in this paper compared it with Gauss-Newton algorithm.

For all algorithms, the initial value is 0, the error value is 0.01. For Algorithm 3.2.1, we use $\eta_1 = 0.2, \eta_2 = 0.8, \gamma_1 = 0.5, \gamma_2 = 2, \Delta_0 = \|\nabla S(\beta_0)\|$.

The numerical results are summarized in table 1 and table 2.

It is seen from Table 1 that Algorithm 3.2.1 performs better than the Gauss-Newton algorithm. Therefore, Algorithm 3.2.1 is effective and improves the prediction performance of ARMA model.

Conclusions

Artificial neural networks posses inherent massively parallel processing and fast convergence. Much attention has been paid for to its application to optimization. This paper combines it with the trust region method and proposes a new method to estimate the parameters of ARMA model. Theoretical analysis and simulation both demonstrate the efficiency of the model.

Acknowledgment

This work is supported by Scientific Research Program of Hebei Province Education Bureau (No.A1447). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Huang Yanyong, Wang Qin, Li Yuqi. An Evolution Algorithm for ARMA Parameter Estimation [J]. *Statistics and Decision*, 2009,(16):7- 9.

[2] Shan Rui, Zheng Caiping. An Optimum Algorithm for Nonlinear Times Series ARMA Parameter Estimation [J]. *Statistics and Decision*, 2008, (7): 169-171.

[3] Wang Yan. Applied Time Series Analysis [M]. *Beijing: China Renmin University Press,* 2005.

[4] Sun Wenyu, Xu Chengxian, Zhu Detong. Optimization Method [M]. *Beijing: Higher Education Press,* 2004.

[5] Zhaoyang Sang, Qingying Sun. A self-adaptive trust region method with line search based on a simple subproblem model [J]. *Journal of Computational and Applied Mathematics* 232 (2009) 514-522.

[6] Zhen-Jun Shi, Jin-Hua Guo. A new trust region method for unconstrained optimization [J]. *Journal of Computational and Applied Mathematics* 213 (2008) 509 – 520.

[7] Fredric M Ham, Ivica Kostanic .Principles of Neurocomputing for Science & Engineering[M].*Beijing: China Machine Press,*2007.

[8] Jiao Licheng. Neural Network Computing [M]. *Xian: Xian Electronic Sience &Technology University Press,* 1993.

Table 1 Parameter estimate of ARMA（2,2）

parameters value of real model	Gauss-Newton algorithm parameters evaluations	absolute value between Gauss-Newton algorithm and real value	Artificial neural network with trust region strategy algorithm parameters evaluations	absolute value between artificial neural network with trust region strategy and real value
-1.45	-0.5824	0.8676	-1.2156	0.2344
-0.6	-0.3669	0.2331	-0.4267	0.1733
-0.5	0.4178	0.9178	-0.4387	0.0613
0.3	-0.6152	0.9152	-0.0984	0.2016

Table 2 Residual sum of squares of ARMA（2,2）

Gauss-Newton algorithm	neural network with trust region strategy algorithm
31.4750	17.8043