Chained DLS-ICBP Neural Networks with Multiple Steps Time Series Prediction

Dai Qun Chen Songcan
Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016

Abstract—Based on the circular back-propagation (CBP) network, the improved circular back-propagation(ICBP) neural network was previously put forward and exhibits more general architecture than the former. It has a favorable characteristic that ICBP is better than CBP in generalization and adaptation though the number of its adaptable weights is generally less than that of CBP. The forecasting experiments on chaotic time series, multiple-input multiple-output (MIMO) systems and the data sets of daily life water consumed quantity have proved that ICBP has better capabilities of prediction and approximation than CBP. But in the above predicting process, ICBP neglects inherent structural changes and time correlation in time series themselves. In other words, they do not take into account the influence of different distances between observations and the predicting point on forecasting performance. The principle of discounted least square (DLS) formulates this influence exactly. In this paper, the DLS principle is borrowed to construct the learning algorithm of DLS-ICBP. On this basis we construct chained DLS-ICBP neural networks by combining a new kind of chain structure to DLS-ICBP and investigate multiple steps time series prediction. We prove that DLS-ICBP has better single and multiple step predictive capabilities than ICBP through experiments on the data sets of Benchmarks and water consumed quantity.

Keywords—Neural networks, Chain structure, Discounted least square, Improved circular back-propagation, Multiple steps time series prediction

Ⅰ. Introduction

As a generalization of multilayer perceptron (MLP) and a more general network model analogous to BP, circular backpropagation (CBP) adds to its input layer an extra node having an input of the sum of the squared input components. It possesses favorable capabilities in generalization and adaptability. Under its frame, the vector quantization (VQ) and the radial basis function (RBF) networks are constructed, which shows great flexibility. Retaining the original structure of CBP, we obtain a more general network model–ICBP through a special construction to the extra node in CBP input layer and special evaluations to the weights between the node and the hidden layer[5]. Firstly, in addition to CBP characteristic of constructive equivalence to VQ and RBF[6,7], ICBP can simulate the famous Bayesian classifier in a constructive way. Secondly, although ICBP has less adaptable weights than CBP, it is better in generalization and adaptation than CBP. Thirdly, it still adopts the BP learning algorithm with learning complexity equal to CBP. Testing results show that ICBP possesses better predictive and approximative capabilities than CBP does.

However, during predicting process ICBP neglects inherent structural changes and time correlation in time series itself. Intuitively, predicting point has stronger correlation to observations closer to it and weaker one to those far away from it. Therefore in training process samples in time window impose different influences on network weights: the nearer is the observation from the predicting point, the greater is the influence. Moreover the idea of discounted

least square formulates exactly this influence [4]. To make ICBP embody the above characteristic, we bring forward a DLS-ICBP, based upon DLS and oriented to time series prediction, introducing DLS to its cost function. DLS cost function biases learning towards most recent observations in a time series but without ignoring long term effects[4]. The experiments of non-stationary covariance and certain city's water consumed quantity time series prediction indicate that DLS improves ICBP performance.

When neural networks are trained to predict signals p steps ahead, the quality of the prediction typically decreases for large values of p . One of the reasons for this is the fact that the information in the inputs does not contain much information about the output, if this output lies far ahead in the future[1]. Therefore Duhoux and Suykens used a new kind of neural network chain[2,3] in their experiments and concluded that this network chain leads to an improved prediction of the temperature. In the paper this kind of chain is adopted to constitute chained DLS-ICBP neural networks.

This paper is organized as follows. In Section Ⅱ, we will explain how DLS-ICBP is formed. In Section Ⅲ, we will illuminate how chained DLS-ICBP is built. In Section Ⅳ, we will give results on experimental data.

Ⅱ. DLS-ICBP Network

A. ICBP network

Fig.1 shows a three-layer ICBP network with N_O output nodes, N_h hidden nodes, d input nodes with respect to d dimensional input pattern or vector and an extra input node with input being $x_{d+1} = \sum_{i=1}^{d} a_i^2 x_i^2$. In particular, when all a_i are taken equally, ICBP reduces to the CBP. And at the same time, ICBP weights connecting the extra node to hidden layer differ from CBP ones: $v_{j(d+1)}(j = 1 \ldots N_h)$ for ICBP take a common constant directly while the counterparts are adaptable parameters. Consequently, the discrepancy of the number of adaptable parameters for these two models is $|N_h - d|$. In general, the number of hidden nodes is larger than that of input nodes due to the proven result that the forward multi-layer networks with sufficient hidden node number can approximate any continuous function to arbitrary precision. Therefore, the adjustable parameters of ICBP are often less than that of CBP. Now let the network expected outputs be $o_i(i = 1 \ldots\ldots N_O)$, and a corresponding sum-of-squares error function is defined as:

$$E = \frac{1}{2}\sum_i (o_i - y_i)^2 \qquad (1)$$

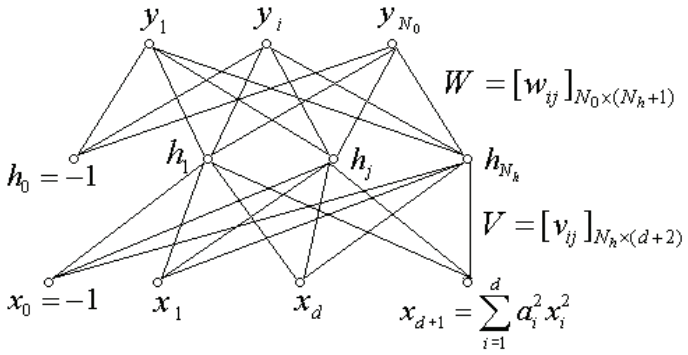

Fig. 1 ICBP three-layer network model

Adopting the known-as error back-propogation learning algorithm (in fact, any other improved algorithms can be applied), the weight adjustments between output and hidden layer are easily derived as follows:

$$\Delta w_{ij}(t) = -\eta \frac{\partial E_p}{\partial w_{ij}} = \eta[o_i(t) - y_i(t)] \cdot h_j(t), \quad i = 1.....N_o, j = 0.....N_h \tag{2}$$

The weight adjustments between hidden layer and input layer are:

$$\Delta v_{jk}(t) = \eta[\sum_{l=1}^{N_o}(o_l(t) - y_l(t))w_{lj}] \cdot h_j(t) \cdot (1 - h_j(t))x_k(t)$$

$$j = 1.....N_h, k = 0.....d \tag{3}$$

Adjusting formula for $a_k, (k = 1...d)$ is:

$$\Delta a_k = 2\eta \sum_{j=1}^{N_h}\left\{\sum_{i=1}^{N_o}\left[(o_i - y_i)w_{ij}\right] \cdot h_j(1-h_j)v_{j,d+1}\right\} \cdot a_k x_k^2, \quad k = 1.....d \tag{4}$$

B. DLS error back-propogation algorithm

Learning by error backpropagation is an error minimization procedure which uses gradient descent into weight error space to minimize a quadratic measure of total error. The most commonly used error measure is the Ordinary Least-Square criterion (OLS) as shown in (5).

$$E_{LS} = \frac{1}{2N}\sum_{p=1}^{N}(\vec{o}_p - \vec{y}_p)^2 \tag{5}$$

where N is the total number of observations in the sample, $\vec{o}_p$ is the desired response and $\vec{y}_p$ is the observed response, with p=N being the most recent observation. Least Squares measures give equal weight to all observations in the sample (training set). In time series analysis with structural changes it is often desirable to overweight more recent observations for the reasons discussed above[4]. The idea of Discounted Least Squares arises from here. The cumulative error calculated by the DLS procedure is given by

$$E_{DLS} = \frac{1}{2N}\sum_{p=1}^{N} w(p)(\vec{o}_p - \vec{y}_p)^2 \tag{6}$$

N, $\vec{o}_p$, $\vec{y}_p$ are defined as above, and $w(p)$ is an adjustment of the contribution of observation p to the overall error[4] or discounted rate. In general, there are many different ways of biasing the cost function through $w(p)$ (such as linear, exponential, etc.) to differentially weight the contribution of each observation towards the total error. In the paper we examine a simple sigmoidal decay as shown in (7)

$$w(p) = \frac{1}{1+e^{(a-bp)}} \qquad \text{where} \quad b = \frac{2a}{N} \tag{7}$$

The parameters a and b are used to scale and offset the sigmoid. The DLS cost function is asymptotically invariant with respect to the sample size (N). Since b in (7) is derived from a and N, the only control parameter is the discount rate a.. The learning rule is derived in the usual way by repeatedly changing the weights by an amount proportional to

$$\frac{\partial E_{DLS}}{\partial W} = \frac{1}{1+e^{(a-bp)}} \frac{\partial E_{LS}}{\partial W} \tag{8}$$

C. DLS-ICBP neural network

In this paper DLS cost function is adopted into ICBP network. A group of biased weights modifying formulae are derived out accordingly from (8),(2),(3) and (4). The weight adjustments between output and hidden layer are:

$$\Delta w_{ij}(t) = -\eta \frac{\partial E_p}{\partial w_{ij}} = \frac{\eta}{1+e^{(a-bp)}} [o_i(t) - y_i(t)] \cdot h_j(t), \quad i = 1.....N_o, j = 0.....N_h \tag{9}$$

Weights adjustments between hidden layer and input layer are:

$$\Delta v_{jk}(t) = \frac{\eta}{1+e^{(a-bp)}} [\sum_{l=1}^{N_o} (o_l(t) - y_l(t)) w_{lj}] \cdot h_j(t) \cdot (1 - h_j(t)) x_k(t)$$

$$j = 1.....N_h, k = 0.....d \tag{10}$$

Adjusting formula for $a_k, (k = 1...d)$ is:

$$\Delta a_k = \frac{2\eta}{1+e^{(a-bp)}} \sum_{j=1}^{N_h} \left\{ \sum_{i=1}^{N_o} \left[(o_i - y_i) w_{ij} \right] \cdot h_j (1 - h_j) v_{j,d+1} \right\} \cdot a_k x_k^2, \quad k = 1.....d \tag{11}$$

In DLS-ICBP, learning is biased toward more recent observations with long term effects experiencing exponential decay through time. This is particularly important in systems in which the structural relationship between input and response vectors changes gradually over time but certain elements of long-term memory are still retained. Experiments results show that DLS-ICBP achieves better predictive effects than ICBP in both single step and multiple steps prediction.

Ⅲ. Chained Neural Networks

A. Classical chain structure neural networks[3]

For the classical chain networks, we train one single one-step ahead predictor and use this network iteratively p times while shifting the inputs appropriately each time a prediction of the previous network is available.(Fig. 2) When using this sliding input system, one trains one neural

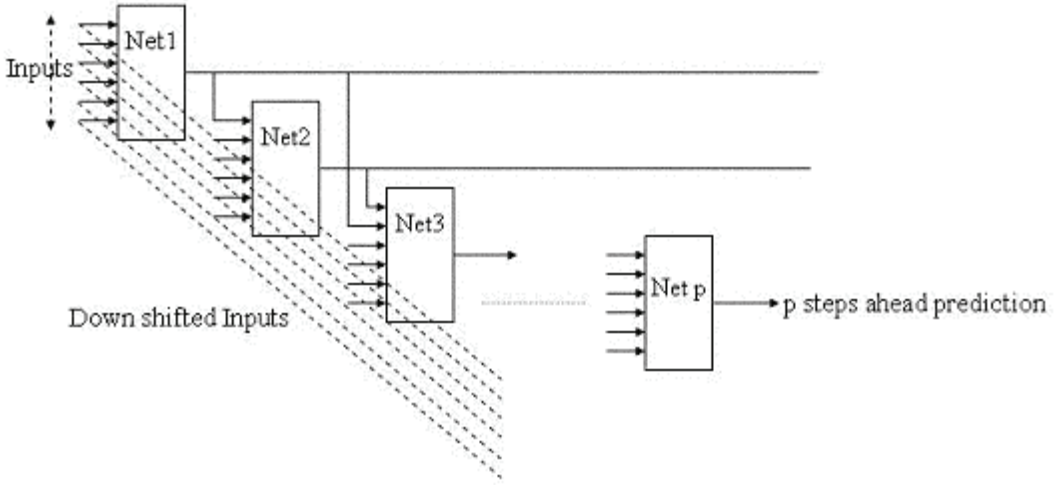

Fig. 2 Classical chain structure neural networks

network f as:

$$\hat{y}_{k+1} = f(y_k, y_{k-1}, ..., y_{k-q+1}, u_k, u_{k-1}, ..., u_{k-r+1}) \tag{12}$$

with q being the number of predictive variables we wish to include from the past, and r the same for other inputs. To calculate the p-step ahead prediction, we use:

$$\begin{aligned} \hat{y}_{k+p} &= f(\hat{y}_{k+p-1}, \hat{y}_{k+p-2}, \ldots, \hat{y}_{k+p-i}, y_{k+p-i-1}, \ldots, y_{k+p-q}, \\ &\quad u_k, u_{k-1}, \ldots, u_{k-r+1}) \quad if\ i < q, \\ \hat{y}_{k+p} &= f(\hat{y}_{k+p-1}, \hat{y}_{k+p-2}, \ldots, \hat{y}_{k+p-q}, \\ &\quad u_k, u_{k-1}, \ldots, u_{k-r+1}) \quad if\ i \geq q. \end{aligned} \tag{13}$$

Where i depends on how many estimates have been calculated in a previous run: beginning with i=1, one gradually has to include more previous estimates for the output $\hat{y}$, until one finally arrives at the p[th] sample prediction, $\hat{y}_{k+p}$. Notice that the variable to be predicted uses a constant number of previous values, namely q. That is the reason why the network f can be used in all the steps iteratively. The classical chained networks have a smaller learning complexity than the new kind of chain structure. Yet certain input data are abandoned gradually along the chain, so parts of the historical information are left out. This makes the chained networks go short of needed predictive information and the quality of the prediction decrease fast as p increases.

B. New kind of chain structure neural networks[2]

Fig. 3 shows a new kind of chain structure we adopted in this paper. For this chain, one trains p different networks, each using the normal inputs and targets, but now with the prediction of the previous steps as extra inputs. Separately trained predictors f_j can also be used to calculate individual intermediate predictions. Therefore, it is necessary to train p networks $f_j, j = 1 \cdots p$, augmenting the vector $\vec{y}_k$ with estimates $\hat{y}_{k+j-1}$ from previous networks $f_{j-1\ldots1}$. It is necessary to train exactly p different networks, and that the usage of the structure is therefore not

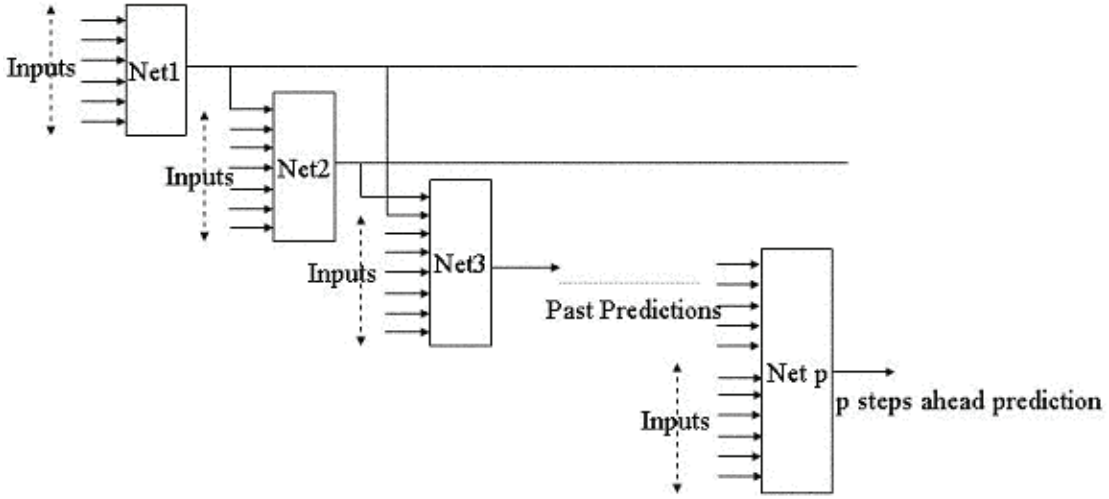

Fig. 3 A new kind of chain structure neural networks

uses in an iterative way for predicting. In fact, one has a chain of one-step ahead predictors, where each k-th model is retrained, eventually with the final weights and biases in the parameter vector θ_{k-1} of its predecessor as a part of its own initial guess θ_{k0}. The learning complexity for this new chain is much higher than that for the classical one. However, there doesn't exist informational loss in this kind of new structure. The network inputs for each step include not only previous predictive results but also all historical information. Therefore it processes multiple steps prediction much better than the former one.

Ⅳ. Experimental results

A. Multiple steps non-stationary variance prediction[4]

To compare the performance characteristics of DLS-ICBP and ICBP, we carry out an controlled multiple steps predictive experiment using a simple sinusoid function. In the experiment the variance of the time series is changing through time. The non-stationary variance time series is built from:

$$y = a(n+x) * \sin(2\pi x) + b, \quad x \in [0,1], \quad n \in \{0,\ldots\ldots,7\}$$

With the parameters set to a=3, b=[2a]/N. The new kind of chained neural networks discussed in section Ⅲ is adopted here. Fig. 4, 5, 6 show ICBP-1 and DLS-ICBP single step, double and four steps predictive results separately. The item in table 1 is MSE (the mean squared differences between the 100 predictive results and targets of ICBP-1 and DLS-ICBP). It can be easily seen from these figures and tables that DLS-ICBP possesses much more excellent performance than ICBP-1.

B. Multiple steps prediction of city daily life water consumed quantity

To predict city daily life water consumed quantity, especially multiple steps, can help to lay a productive course, economize energy sources and boost production benefit. It is practically valuable for civil life and manufacture. In terms of the historical data provided, we predict the water consumed quantity one month to one quarter ahead for the use of water supply department. In this paper, single step, double and four steps forecasting experiments are carried out on the data sets of the most recent twelve months. Experimental results for ICBP-1 and DLS-ICBP are listed in contrast in Table 2. The items in this table are MPE:

$$MPE = \frac{1}{NPN_o}\sum_{i=1}^{N}\sum_{j=1}^{P}\sum_{k=1}^{N_o}\left|\frac{\Pr ediction_k^j - Actual_k^j}{Actual_k^j}\right|$$

Where P represents the P^{th} predictive result, N represents the repeated experimental times and N_o represents the number of output nodes. Here N=10, P=12, N_o=1. Fig. 7,8 and 9 show respectively the single step, double steps and four steps forecasting outputs in comparison with practical water consumed quantity. It can be concluded from experiments that single step and multiple steps predictive effects for DLS-ICBP are better than those for ICBP-1.

Ⅴ. Conclusion

Since DLS-ICBP still reserves the simple structure of ICBP, it inherits most merits of ICBP, such as having less adaptable parameters and better prediction capability than CBP. By introducing discounted least square error function to BP algorithm, the learning of DLS-ICBP is biased toward most recent observations and the long term effects in time series are taken into account at the same time. It has been proved by experiments that DLS-ICBP represents a obviously better performance than ICBP for both single step and multiple steps time series prediction. In daily life water consumed quantity prediction, the MPE of DLS-ICBP is lower than 3%, improving that of ICBP by 1-2 percents. So DLS-ICBP meets the demands of this kind of applications better.

Problems that result from the proposed approach, are first that if introducing a certain kind of initialization algorithm to DLS-ICBP, can we boost the network convergent speed? And second that if function $w(p)$ is adopted aiming at certain applications, can we further advance the predicting effects? We are currently undertaking the topics.

References

[1] C.M.Bishop. Neural networks for pattern recognition. Oxford University Press, 1995.

[2] M.Duhoux, J.Suykens, B.De Moor and J.Vandewalle. Improved long-term temperature prediction by chaining of neural networks. International Journal of Neural Systems, Vol.11, No.1: 1-10, 2001.

[3] J.Suykens and J.Vandewalle. Nonlinear modeling: advanced black box techniques. Kluwer Academic Publishers, Boston, 1998.

[4] Apostolos-Paul Refenes, Yves Bentz and Derek W. Bunn. Financial time series modelling with discounted least squares backpropagation. Neurocomputing 14: 123-138, 1997.

[5] Dai Qun，Chen Song Can，Zhang Ben Zhu，Improved CBP neural networks with its applications in time series prediction，Submitted to Chinese Journal of Computer.

[6] Zhang Ben Zhu，Chen Song Can, Equivalence between vector quantization and ICBP networks，Journal of Data Acquisition and Processing，2001,16(3):291-294

[7] Zhang Ben Zhu，The research on the performance and applications of improved BP neural networks, Thesis of Master degree，Nanjing University of Aeronautics and Astronautics，2001,2.

[8] Zhang Ben Zhu，Chen Song Can，The equivalence between ICBP and the Bayesian classifier，Manuscript.

Table 1. Single step and multiple steps experimental results on non-stationary covariance time series. The items in table are MSE for 100 network outputs and targets

Prediction steps / Network model	Single step	Double steps	Four steps
ICBP-1	5.6420	13.1875	22.6163
DLS-ICBP-1	2.8835	1.5998	18.6701

Table 2. Water consumed quantity single step and multiple steps prediction results
The items in table are MPE(%) for the network outputs and practical quantity of twelve months

Perdiction steps / Network model	Single step	Double steps	Four steps
ICBP-1	3.8382	3.1094	2.3707
DLS-ICBP-1	1.9166	2.3621	1.5947

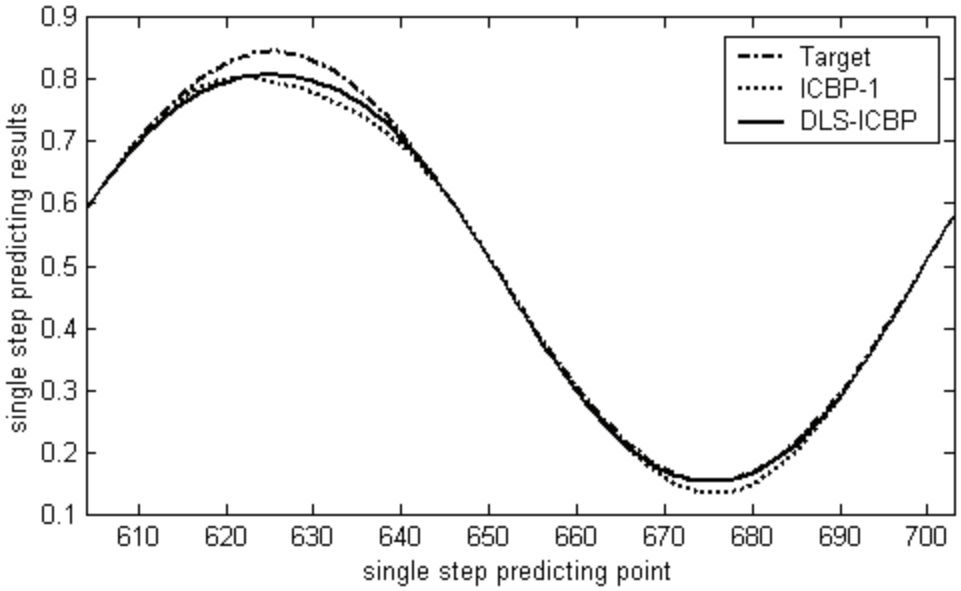

Fig. 4 ICBP-1 and DLS-ICBP single step predicting results on the 100 predicting points in [604, 703] for non-stationary covariance time series prediction

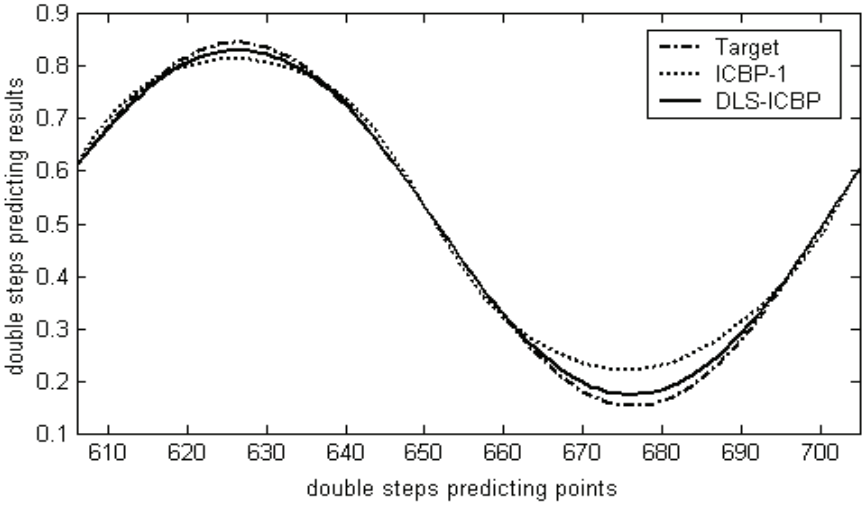

Fig. 5 ICBP-1 and DLS-ICBP double steps predicting results on the 100 predicting points in [606, 705] for non-stationary covariance time series prediction

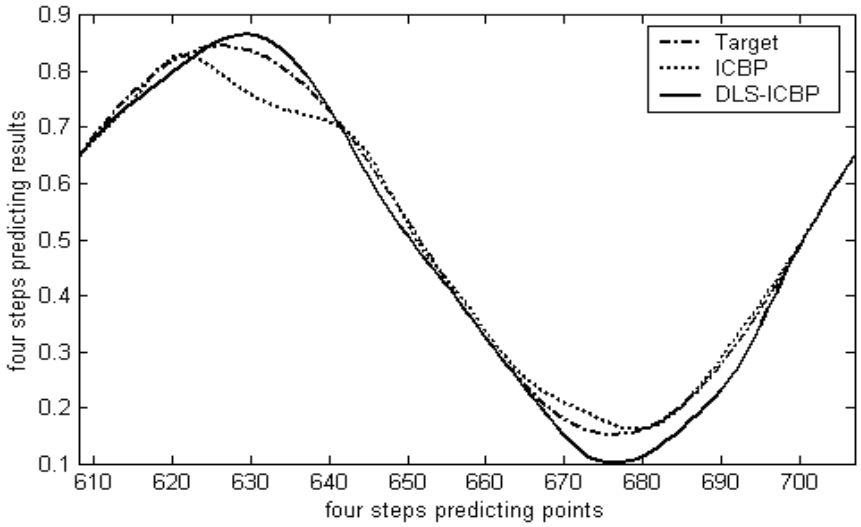

Fig. 6 ICBP-1 and DLS-ICBP double steps predicting results on the 100 predicting points in [606, 705]

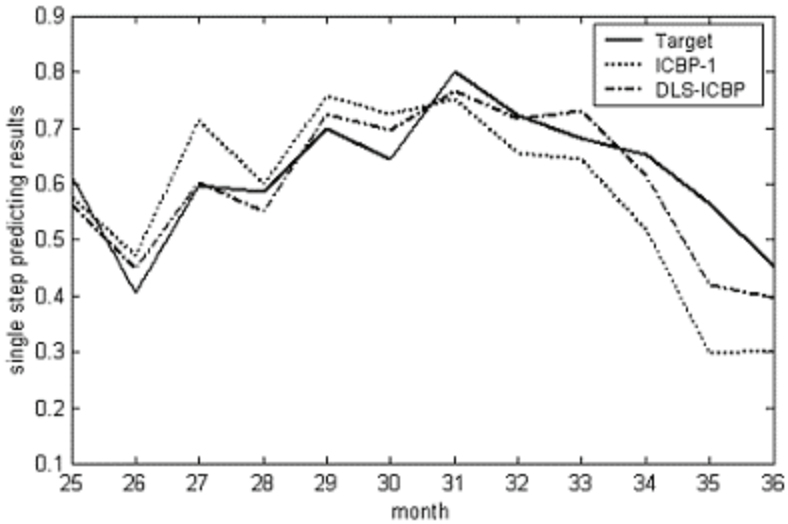

Fig. 7 ICBP-1 and DLS-ICBP single step predicting results for 12 months water consumed

quantity in comparison with practical one

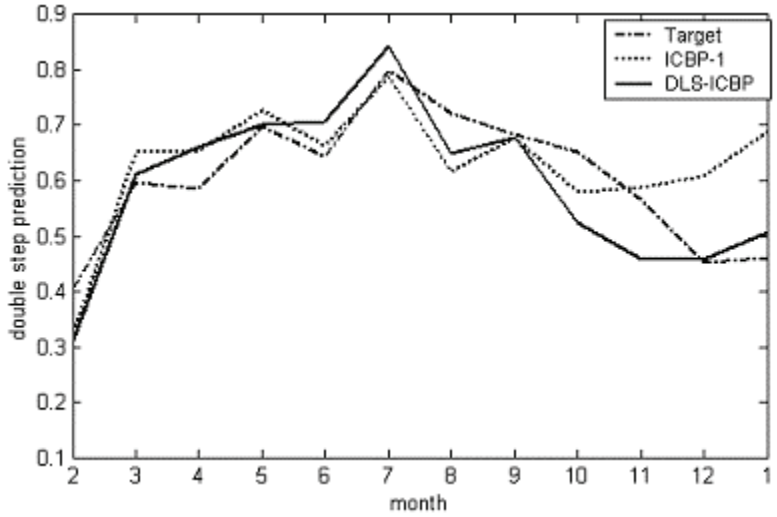

Fig. 8 ICBP-1 and DLS-ICBP double steps predicting results for 12 months water consumed quantity

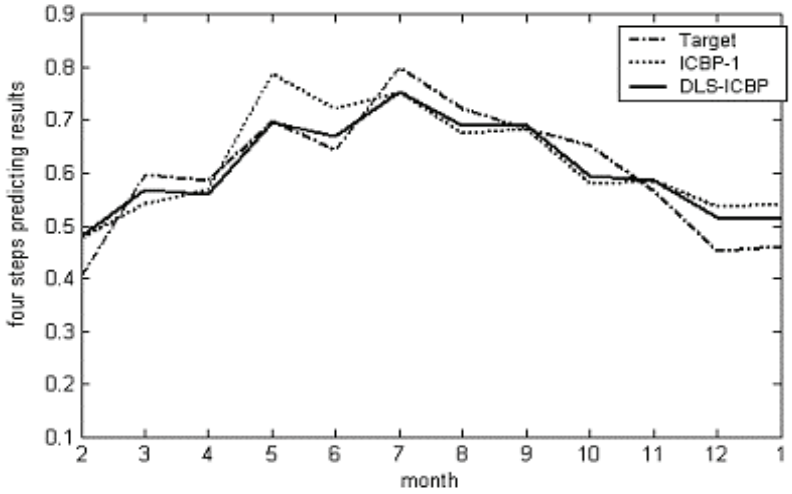

Fig. 9 ICBP-1 and DLS-ICBP four steps predicting results for 12 months water consumed quantity