http://www.zju.edu.cn/jzus; E-mail: jzus@zju.edu.cn
ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.4, No.5, P.578-583, Sep.-Oct., 2003

Application of uncertainty reasoning based on cloud model in time series prediction*

ZHANG Jin-chun(张锦春)†, HU Gu-yu(胡谷雨)
(*Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China*)
†E-mail: jinchunzhang@sina.com.cn

Received Sept.3, 2002; revision accepted Jan.8, 2003

Abstract: Time series prediction has been successfully used in several application areas, such as meteorological forecasting, market prediction, network traffic forecasting, etc., and a number of techniques have been developed for modeling and predicting time series. In the traditional exponential smoothing method, a fixed weight is assigned to data history, and the trend changes of time series are ignored. In this paper, an uncertainty reasoning method, based on cloud model, is employed in time series prediction, which uses cloud logic controller to adjust the smoothing coefficient of the simple exponential smoothing method dynamically to fit the current trend of the time series. The validity of this solution was proved by experiments on various data sets.

Key words: Time series prediction, Cloud model, Simple exponential smoothing method
Document code: A **CLC number:** TP393

INTRODUCTION

Time series prediction has been successfully used in several application areas, such as meteorological forecasting (Andrei, 1972), load prediction in power system (Ou and Li, 1999), market prediction (Giles *et al.*, 1997), network traffic prediction (Edwards *et al.*, 1997), etc., and a number of techniques have been developed for modeling and predicting time series. However, these techniques seldom have ideal prediction effect for forecasting some high-burst and non-stationary time series, such as network traffic, power load, etc. In this paper, an uncertainty reasoning method, based on cloud model, is employed in time series prediction; an advanced solution of which is proposed.

Section 2 briefly discusses time series prediction and several common prediction methods; Section 3 introduces uncertainty reasoning based on cloud model and a solution of time series prediction improved by cloud logic controller; The effectiveness of this solution was proved by experiments on various datasets in Section 4.

TIME SERIES PREDICTION

1. Time series

A time series is a sequence of vectors, $x(t)$, $t = 0, 1, 2, ..., t$, where t represents elapsed time. Theoretically, x may be a value that varies continuously with t, but in practice, for most given physical systems, x will be sampled to give a series of discrete data points (Frank *et al.*, 2001).

The common methods (Gershenfeld and Weigend, 1993) of time series prediction are by: simple average, moving average, simple exponential smoothing.

2. Simple average

In this method, the averages of all the values at the previous time are taken as forecasted values. This method is suitable for forecasting time series without obvious fluctuation. The forecasted value Y_{n+1} is given by

$$Y_{n+1} = \frac{\sum X_i}{n} = \frac{X_1 + X_2 + X_3 + \cdots + X_n}{n} \quad (1)$$

* Project (No.2001AA112030) supported by the National Hi-Tech Development Program (863) of China

here X_i is the ith actual value, and n is the number of observations.

3. Moving average

Moving average is divided into simple moving average and weighted moving average.

Simple moving average forecasts future values based on an average of n past values. The following formula is used in finding the moving average of order d,

$$Y_{n+1} = \frac{1}{d}\sum_{i=n-d+1}^{n} X_i = \frac{X_{n-d+1} + X_{n-d+2} + X_{n-d+3} + \cdots + X_n}{d} \quad (2)$$

where Y_{n+1} is the forecasted value, d is moving period.

However, in weighted moving average method, a weighted average of past values is taken as the forecasted value. As an example, a weighted moving average is:

$$Weighted\ Y_{t+1} = W_1 X_t + W_2 X_{t-1} + W_3 X_{t-2}$$

where the weights are any positive numbers such that: $W_1 + W_2 + W_3 = 1$.

4. Simple exponential smoothing

Simple exponential smoothing, the most common method of time series prediction, was developed on the basis of the moving average technique (Dorffner, 1996). It forecasts the next value based on current actual value and current forecasted value as shown in Eq.(3), which not only reflects the influence of the nearest actual value on the forecasted value, but does not need masses of past values.

$$\begin{aligned} Y_{n+1} &= \alpha X_n + (1-\alpha) Y_n \\ &= Y_n + \alpha(X_n - Y_n)\ (0 \leqslant \alpha \leqslant 1) \end{aligned} \quad (3)$$

Here, α is smoothing coefficient, which ranges from 0 to 1. From Eq. (3), we can see that α can be thought of as the weight given to past history. The larger the value of α, the less weight the past history has in relation to the last actual value of the parameter. The method is called 'exponential', since the forecasted value is the discrete convolution of the observed sequence with an exponential curve with a time constant $1/(1-\alpha)$. Alternately, if the value of X_n becomes fixed, the error $(X_n - Y_n)$ decays exponentially.

The simple exponential smoothing technique is very robust, and has been used in a number of applications. However, a major problem with simple exponential smoothing is the choice of α. While in principle, it can be determined by knowledge of the system and data history, in practice, it is always based on the minimum MAD (Mean Absolute Deviation) of the data history, here $MAD = \frac{1}{n}\sum_{i=1}^{n} | Y_i - X_i |$ (Edwards *et al*., 1997). The value α should be recalculated only if the system behavior changes greatly, but some problems appear, for example, the time of recalculation is hard to determine, the value of α may be changed mistakenly because of a transient spike, etc. Hence, our approach uses cloud logic controller to automatically determine a good value of α.

SOLUTION OF TIME SERIES PREDICTION BASED ON CLOUD MODEL

1. Time Series analysis

A network is composed of many nodes, every one of them has a lot of network applications, and all communications between these applications make up the network traffic. If the traffic between two applications is considered as a random variable, network traffic would be the sum of many random variables so that there is randomness in network traffic (Schwartz, 1998); in the same way, power load can be thought of as a sum of many branch loads. In fact, most time series in the real world are affected by a lot of random factors, or composed of several random variables, so randomness among them must be taken into account during forecasting time series.

Otherwise, we simply divide the time series into two kinds: 'steady' and 'unsteady' time series. For a 'steady' time series, the time series X_n is approximately constant, then α should be low, so that the past history is given more weight, and transient spikes are ignored. In contrast, the data could be from an 'unsteady' time series, so that it varies continuously and considerably over time. In this case, a low α would filter out changes of time series, and cause a large lag during prediction. By choosing a higher value of α, the past history, which only provides obsolete information, is ignored, and the changes of time series are quickly tracked.

However, 'steady' and 'unsteady' are fuzzy linguistic variables, and no well-marked boundary line exists between them, so adjustment of α is also involved in the area of fuzzy control.

From the above discussion, adjustment of α is involved in two kinds of uncertainty: fuzziness and randomness, which can be integrated by cloud model (Li *et al.*, 2000). In next section, cloud model and uncertainty reasoning method based on cloud model are introduced briefly.

2. Uncertainty reasoning based on cloud model

The key problems that should be solved in uncertainty reasoning are the description and transfer of uncertainty. The former is represented by cloud model, and the latter is done by rule constructor based on cloud model.

The cloud model effectively integrates fuzziness and randomness of linguistic terms in a common sense. Based on this model, a method of uncertainty reasoning is developed to bridge the gap between quantitative and qualitative knowledge in a unified way (Li, 1997a).

Let U be the set $U = \{u\}$, as the universe of discourse, and T a term associated with U. The membership degree of u in U to the term T, $C_T(u)$, is a random member with a stable tendency. $C_T(u)$ takes the value in $[0, 1]$. And the distribution of the membership degree on U is called compatibility cloud (Li, 1997a).

It had been proved (Li *et al.*, 2000) that the expected curves of most phenomena in society and science appear Gaussian in their shapes; and that the distribution of the compatibility degrees associated with a certain point of a compatibility cloud is also a Gaussian distribution, the expected value of which is the point on the expected curve of the compatibility cloud. So there are two Gaussian distributions in a compatibility cloud, and a given set $\{Ex, En, He\}$ uniquely defines a particular compatibility cloud, where Ex, En, and He are expected value, entropy and hyper entropy of cloud respectively.

The expected value Ex of a compatibility cloud is the position at U corresponding to the center of gravity of the cloud, and here represents the current status of the network traffic; The entropy En reflects the assembly degree of the samples, and here reflects the fluctuant degree of network performance in a certain time slot; The hyper entropy, He, is a measure of the uncertainty of the entropy En, and here represents the influence of the uncertain factors on the network performance. These three statistical characteristics can reflect the distribution of the samples to a great extent.

The combination of two conditioned cloud generators can represent the rule in terms of "if A then B", as shown in Fig. 1, in which two linguistic terms $A(Ex_A, En_A, He_A)$ and $B(Ex_B, En_B, He_B)$ are given (Li, 1997b). When a particular value comes in as input, the drop of (a, u_a) is produced by the X-conditioned cloud generator CG_A. If the degree of u_a is high, which means the element a highly belongs to the concept A, then u_a triggers the Y-conditioned cloud generator CG_B as the compatibility degree. Thus an element b with (b, u_a) highly belonging to B is generated. This process, through the combination, realizes that the degree of the output $b \in B$ is controlled by the degree of element $a \in A$. That is the rule of "if A then B".

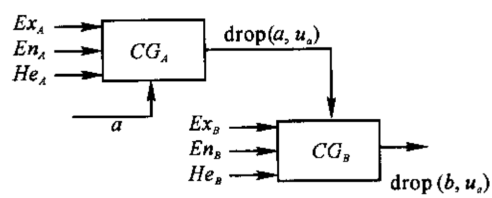

Fig. 1 Rule constructor

Thus, the problem of transfer and update of uncertainty can be solved perfectly by the rule constructor. The detailed algorithm of the rule constructor can be found in the literature (Li, 1997b).

3. Cloud logic controller

In our solution, the smoothing coefficient α is adjusted by cloud logic controller on the basis of rule constructor to fit changeful time series. But some rules should be defined first in terms of linguistic variables. Through analysis of the time series, we know that: when the time series is 'steady', that is, the error between forecasted value and actual value is small, the value of α should be reduced properly; on the contrary, when the time series is 'unsteady', then the error is large, and α should be increased properly.

So, the following logic rules are defined:

If error is very large then α is increased greatly

If error is large then α is increased moderately

If error is medium then α is kept steady

If error is small then α is reduced moderately

If error is very small then α is reduced greatly

The 'error' in the rules is defined as 'proportional error of forecasted value to actual value':

$$\text{proportional error} = \frac{|\text{forecasted value} - \text{actual value}|}{\text{actual value}}$$

The cloud logic controller is generated after the rules above are realized with the cloud model's rule constructor.

4. Flow of time series prediction using cloud logic controller

Fig.2 is the flow of the resultant system; it incorporates exponential smoothing and cloud logic controller:

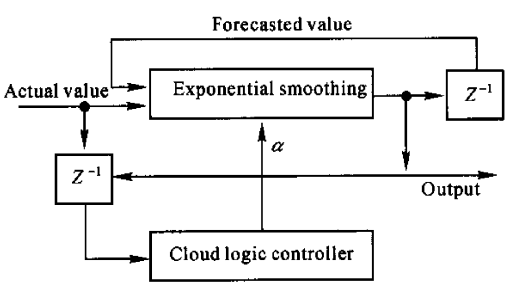

Fig.2 Schematic flow diagram of the prediction system

1) Initialization

2) Find the next value according to the current forecasted value, current actual value and the value of α

3) Adjust the value of α by cloud logic controller, the adjusted α is taken as input of the exponential smoothing method to forecast the next value

4) Repeat Step 2 and Step 3 until enough forecasted values are obtained.

SIMULATION

1. Simulation environment

In order to prove the validity of this solution, we compare exponential smoothing using cloud logic controller (solution 2) with simple exponential smoothing (solution 1) by three data sets. The first data set is sinusoid; the second comes from NLANR (National Lab of Application Network Research), and records the network traffic data in WAN in 122797.83s from 23:46 10/3/1989; and the third comes from China Meteorological Administration, and records the maximum temperature data of 2000 days in Beijing from 12/1/1996. For every kind of data, we sample 2000 data points, an initialization set of 700 points and a test set of 1300 points.

The following are the corresponding parameters of linguistic variables involved in the rules of cloud logic controller in the network traffic prediction:

Table 1 Parameters of linguistic variables

Proportional error (Ex, En, He)	The change of α (Ex, En, He)
Very large (1.05, 0.2, 0.3)	Increased greatly (1.3, 0.15, 0.2)
Large (0.75, 0.18, 0.3)	Increased moderately (1.1, 0.1, 0.3)
Medium (0.45, 0.15, 0.03)	Keep steady (1.0, 0.05, 0.3)
Small (0.15, 0.12, 0.003)	Reduced moderately (0.9, 0.1, 0.03)
Very small (0, 0.1, 0.003)	Reduced greatly (0.7, 0.15, 0.03)

2. Performance measure

There are many performance measures for assessing the prediction method quality. The common two are NMSE (Normal Mean Square Error) and DS (Direction Symmetry).

$NMSE = \frac{1}{N\sigma^2}\sum_{k=1}^{N}[x(k) - \hat{x}(k)]^2$, here $x(k)$ and $\hat{x}(k)$ are the sequence of actual and forecasted values, respectively, N is the number of test points, σ^2 is deviation of the sequence of actual values. The smaller *NMSE*, the better effect the method has, and if *NMSE* = 1, the method corresponds to unconditional mean prediction.

$DS = \frac{1}{N}\sum_{k=1}^{N}\psi(x(k) * \hat{x}(k))$, here ψ is *Heavyside* function, $\psi(x) = \begin{cases} 1, x > 0 \\ 0, x \leqslant 0 \end{cases}$. *DS* is percentage of correct prediction direction. The closer it is to 1, the better effect the method has.

3. Simulation results

In the same simulation environment, the simulation results of both solutions are as follows:

Table 2 The simulation results of both solutions

	Solution 1 (without cloud logic controller)	Solution 2 (with cloud logic controller)
Sinusoid	*NMSE* = 0.0109; *DS* = 0.9831	*NMSE* = 0.0270; *DS* = 0.9800
Network traffic	*NMSE* = 0.5053; *DS* = 0.6975	*NMSE* = 0.2824; *DS* = 0.8732
Temperature	*NMSE* = 0.1428; *DS* = 0.7259	*NMSE* = 0.0974; *DS* = 0.8337

The sinusoidal data, without uncertainty factor, varies regularly, so both solutions have satisfying prediction results(Fig.3).

For network traffic data set, solution 2 has much better prediction results that solution 1 has not only with the comparison curves (Fig.4 and Fig.5, in all figures, actual values and forecasted values are represented by real line and dotted line respectively) but also with *NMSE* and *DS*. Fig.6 is the curve of partial changes of α in solution 2.

For temperature sequence, solution 2 has better prediction results than solution 1 (Fig.7 and Fig.8).

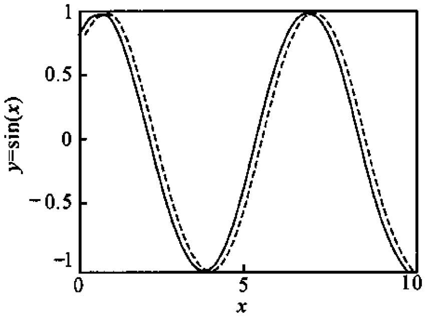

Fig.3 Sinusoid prediction using solution 1

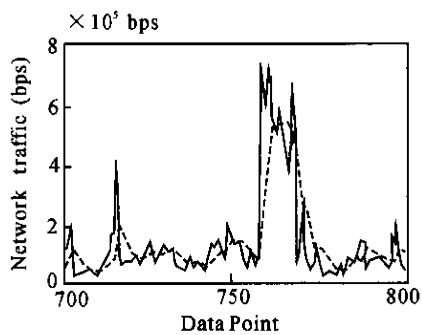

Fig.4 Network traffic prediction using solution 1

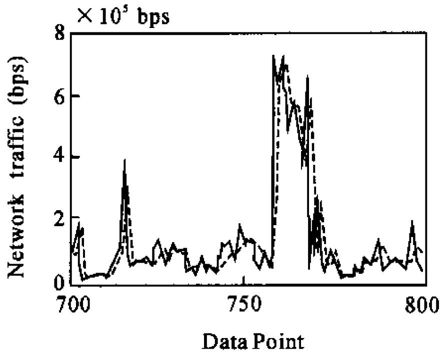

Fig.5 Network traffic prediction using solution 2

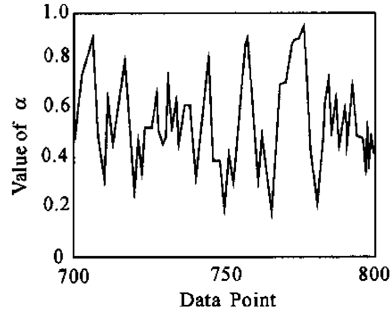

Fig.6 Changes of α in network traffic prediction

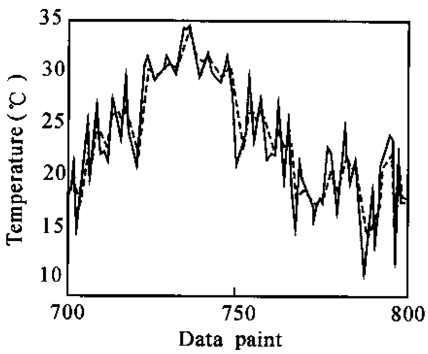

Fig.7 Temperatures prediction using solution 1

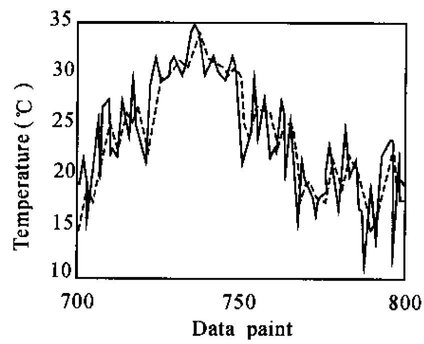

Fig.8 Temperatures prediction using solution 2

4. Conclusions

From the simulation results above, we can conclude that:

1) For temperature sequence and network traffic data set, the prediction performance of the exponential smoothing using cloud logic controller increases greatly, which means that this method is effective.

2) From Fig.6, we can see that the value of α varies considerably, which means that the network traffic is very changeful, and a fixed α is not enough to forecast changeful time series, especially network traffic.

3) For forecasting changeful time series, solution 2 has an advantage over solution 1, and the more changeful the time series is, the better prediction results solution 2 has.

4) In the figures, the forecasted sequence lags behind the actual sequence in both solution 1 and in solution 2, which is a common phenomenon in prediction. But the lag in solution 2 is much slighter than that in solution 1.

SUMMARY

In this paper, several common methods of time series prediction are introduced, and a solution of time series prediction using cloud model is proposed based on the time series analysis. In this solution, the smoothing coefficient of the simple exponential smoothing is adjusted by the cloud logic controller to fit the current trend of the time series. Results of experiments on various data sets proved that the solution was more effective than that of the solation by simple exponential smoothing, especially for forecasting changeful time series like network traffic.

References

Andrei, S. M., 1972. Weather Forecasting as a Problem in Physics. MIT press, Cambridge, MA1972.

Dorffner, G., 1996. Neural networks for time series processing. *Neural Network World*, **4**: 447 – 468.

Edwards, T., Tansley, D. S. W., Frank, R. J. and Davey, N., 1997. Traffic trends analysis using neural networks. *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications*, **3**: 157 – 164.

Frank, R. J., Davey, N. and Hunt, S. P., 2001. Time series prediction and neural networks. *Journal of intelligent and robotic systems*, **31**: 91 – 103.

Gershenfeld, N. A. and Weigend, A. S., 1993. The Future of Time Series. Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley Pub. Co., Santa Fe, NM: 1 – 70.

Giles, C. L., Lawewnce, S. and Tsoi, A. C., 1997. Rule Inference for Financial Prediction Using Recurrent Neural Networks. Proceedings of IEEE/IAFE conference on computational intelligence for financial engineering (CIFEr). Piscataway, NJ: 253 – 259.

Li, D. Y., 1997a. Knowledge Representation and Discovery Based on Linguistic Atoms. Proceedings of the 1st Pacific-Asia Conference, Singapore, p.3 – 20.

Li, D. Y., 1997b. Knowledge representation in KDD based on linguistic atoms. *Journal of Computer Science and Technology*, **12**(6): 1 – 16.

Li, D. Y., Di, K. C., Li, D. R. and Song, Z. L., 2000. Mining association rules with linguistic cloud models. *Journal of Software*, **11**(2): 143 – 158.

Ou, J. P. and Li, L. J., 1999. The application of ANN in short-term load prediction in power system. *Guangdong Electric Power*, **2**: 27 – 31.

Schwartz, M., 1998. Boardband Integrated Networks. Tsinghua University Press, China.