An Integrated Intelligent Fuzzy System for Data Mining

Do-Thanh Sang and Dong-Min Woo
Department of Electronics Engineering, Myongji University,
San 38-2 Namdong
Yongin, Korea 449-728
dmwoo@mju.ac.kr

ABSTRACT

This paper presents an advanced model based on fuzzy inference system, namely Standard Additive Model (SAM) for forecasting the output of any record given the input variables only from the database, the age of abalone in particular. The modeling and learning power of the SAM have been beneficial for the building of a model that is capable of prediction functionalities. In order to intensify the processing speed and accuracy of the SAM as well as to allow it to be more concise, we propose a learning process incorporate a genetic algorithm. In addition, advanced algorithms for unsupervised and supervised learning are employed. Experimental results have demonstrated that the system has the ability necessary for application to financial data mining and prediction in comparison with multilayer perceptron neural networks with the same context. In particular, the proposed method has shown a robustness against noise when the genetic algorithm is employed with a reasonable generation number.

Keywords

fuzzy inference, Standard Additive Model, financial data, prediction, neural network

1. INTRODUCTION

Data mining is the term used to describe the process of extracting hidden information from large relational databases. Even though such data is available, very few companies have been able to realize the actual value of the information stored in it. Because information technology has developed rapidly, many data owners demand predictive information from their data warehouses. Many technologies available for data mining have been mentioned in the literature and these methods have achieved some results with reasonable accuracy [13][10]. Zhang [12] used logistic regression in childhood obesity prediction. Regression analysis is the statistical method of defining an algorithm that describes a set of data. The advantage of regression is that it can find a pattern or trend in the data. The disadvantage of regression analysis is that the pattern may not be useful or valid.

Khan [5] suggests the use of decision trees to continuously extract information for clinical reasoning, in the form of medical expert's actions, which are inherent in a large number of EMRs (Electronic Medical Records). Such information comes in a tree-shaped structure that represents a set of decisions. These decisions generate rules for the classification of the data set. The data mining application analyzes the data, and then builds a tree of these rules to represent the trends. The trends can be used to identify different classes or continuous variables. The primary disadvantage of decision tree forests is that the model is complex and cannot be visualized like a single tree. Harchegani [3] presented a solution for prediction of information sent to mobile systems that uses a cache operation by means of Genetic Algorithms (GAs). One of the differences between GA and other algorithms is that in GA, we deal with a population or set of points at once while in the older methods we operated with just one special point. This improvement means that GA can cover a lot of designs at once. The key disadvantage of the GA is that we have to determine a fitness function for each certain problem.

In this paper, we propose a technique of incorporating Standard Additive Model (SAM) [7], [6] and GA to prediction problem. A fuzzy system based on fuzzy logic which has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design. On the other hand, fuzzy logic solutions are easy to verify and optimize. Performance of SAM is verified by forecasting the age of abalone via its other attributes in database. For ecological and commercial fish farming purposes, the age composition of abalone populations are pertinent. It is often determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope, which is a boring, time-consuming and expensive task. The age could be forecasted by computational intelligent techniques like SAM with reasonable results. The substantial intention of this paper is to elucidate that SAM could be widely employed in the field of data mining.

The paper is organized as follows. The next section describes briefly about the SAM fuzzy system. Section 3 and 4 present the applying SAM to forecasting problem and learning process. Designing a SAM for prediction involving database and experimental results are shown in section 5. Finally, our conclusion is summarized in the last section.

2. STANDARD ADDITIVE MODEL

A fuzzy inference system is constructed by the parallel associative structure of m fuzzy rules (Fig. 1). Each fuzzy rule is defined as a conditional IF-THEN proposition of the form R_j: IF $x = A_j$ THEN $y = B_j$; $j = \overline{1,m}$; where $x \in R_n$, $y \in R_p$, are numerical multi-dimensional vectors, A_j and B_j are fuzzy sets on the input space X and the output space Y

respectively.

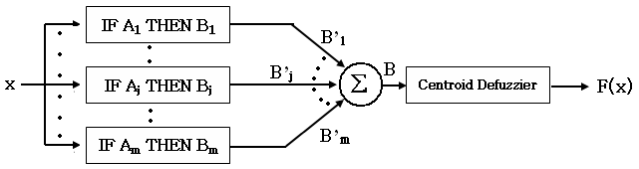

Figure 1: SAM's parallel combination structure

Because of SUM (summary) combination of fuzzy rules, this fuzzy system is named Standard Additive Model (SAM). In SAM, each input x fires j^{th} fuzzy rule and results in fuzzy set B'_j determined by the PRODUCT operation between membership degree of if-part $a_j(x)$ in [0,1] and then-part fuzzy set $B_j : B'_j = a_j(x)B_j$. The system can use joint set function a_j or some factored form such as $a_j(x) = a_j^1(x_1)...a_j^n(x_n)$ or $a_j(x) = min(a_j^1(x_1), ..., a_j^n(x_n))$ or any other conjunctive form for input vector $x \in R_n$. Then-part fuzzy set $B_j \subset R_p$ has set function $b_j : R_p \to [0, 1]$ and volume V_j and centroid (center of gravity) c_j of fuzzy set.

$$F(x) = Centroid\left(\sum_{j=1}^{m} w_j a_j(x) B_j\right) = \frac{\sum_{j=1}^{m} w_j a_j(x) V_j c_j}{\sum_{j=1}^{m} w_j a_j(x) V_j} = \sum_{j=1}^{m} p_j(x) c_j \quad (1)$$

The above formula shows that $F(x)$ is a convex sum of then-part set centroids where the convex weight:

$$p_j(x) = \frac{w_j a_j(x) V_j}{\sum_{k=1}^{m} w_j a_k(x) V_k} \quad (2)$$

A SAM inferential fuzzy system can uniformly approximate continuous and bounded measurable functions on compact domains. If $y = f(x)$ is not analytically known, we cannot write an equation in explicit form. However, we can use the relationship between the input space X and the output space Y which is given by $Y = F(x)$, a relationship that links subsets of the input space X to subsets of the output space Y. Fig. 2 illustrates the Cartesian fuzzy rule patches in the input-output space and how these patches cover the graph of $f(x)$.

The approximation capability of SAM is outlined as follows:

- Each fuzzy rule R_j is formed by the combination of if-part fuzzy set A_j (its space defined by the X axis) and then-part fuzzy set B_j (space defined by the Y axis). The fuzzy patch $R_j = A_j \times B_j$ is determined in the product space $X \times Y$. This leads to exponential rule explosion in high dimensions. Optimal rules can help deal with this drawback. Fuzzy sets A_j and B_j in above figure have shape of Cauchy set function.

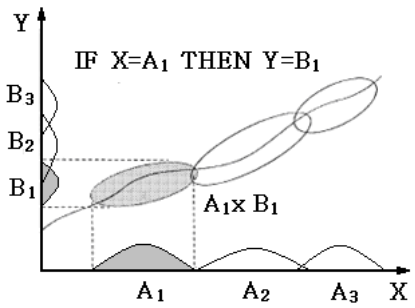

Figure 2: Mapping of input space into output space

- By defuzzifying the combination of fuzzy patches R_j, SAM can cover the graph of the unknown function $f(x)$.

2.1 Shape of fuzzy sets

The shape - membership function - of if-part fuzzy sets affects how well fuzzy systems approximate nonlinear continuous functions [9]. There is no function as best selection, it depends on each specific application. In trying to achieve accuracy and stability in application, all membership functions are examined in this research. Some membership functions (e.g. trapezoid, difference hyperbolic tangent, metrical difference logistic) often take much time in processing and therefore they are less effective.

Gaussian function gives richer fuzzy systems with simple learning laws that tune the bell-curve means and variances. But this popular choice comes with a special cost: it converts fuzzy systems to radial-basis-function neural networks or to other well-known systems that predate fuzzy systems.

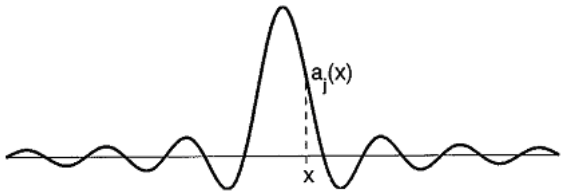

Figure 3: *Sinc* set function in 1-D case

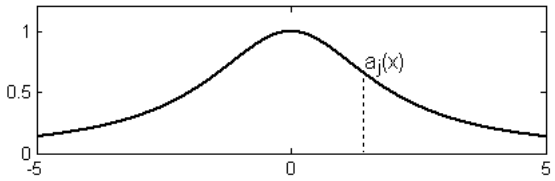

Figure 4: *Cauchy* set function in 1-D case

The *sinc* set function often does well but its undulating side lobes may have no linguistic meaning (Fig. 3). The *Cauchy* function is a bell curve with thicker tails than the Gaussian bell curve and with infinite variance and higher order moments. Thus the *Cauchy* set function is chosen for experiments in this research. The j^{th} *Cauchy* set function (Fig.

4) centered at m_j and width $d_j > 0$ is defined as

$$a_j(x) = \frac{1}{1 + \left(\frac{x - m_j}{d_j}\right)^2} \quad (3)$$

2.2 Constructing fuzzy rules

Based on the clustering results, the fuzzy rules system is constituted. The centers of fuzzy rules c_j are identified via centroid vectors. Identifying the parameters of membership functions demands the width of fuzzy sets. The width of j^{th} fuzzy set, for example, is set simply via its neighbors by the following formula

$$V_j = \frac{|c_j - c_c|}{r} \quad (4)$$

where c_j is the center of j^{th} fuzzy set, c_c is the center of the closest fuzzy set, and r is the overlap coefficient. Besides, fuzzy set width could be tuned afterwards in the parameter learning process.

Generally, a prediction problem is considered as one of nonlinear function approximations wherein the function's approximated value will be used as the prediction. In order to improve approximating performance, SAM needs a strong learning process aiming at obtain a set of robust fuzzy rules. Through modifying volume and centroid of fuzzy rules, SAM relocates automatically fuzzy patches' position and size hereby the approximation can be expected more accurate. Regarding SAM, a learning method is evaluated whether well or not based on the way it adjusts size and position as well as ensures to maintain fuzzy patches at curve points in graphical presentation of the function $f(x)$.

3. LEARNING PROCESS

The learning process of SAM (or fuzzy system, generally) usually encompasses two main stages those are structure learning and parameter learning as shown in Fig. 5. Also, regarding the database applications, the data preprocessing stage is required. Hence, the process includes:

- **Data preprocessing**: Data mining often deals with data that have not been looked at for years, so that much of the data contain field values that have expired, are no longer relevant, or are simply missing. In order to achieve reliable results, the databases need to undergo preprocessing, in the form of data cleaning and data transformation [8].

- **Structure learning**: This stage is a self-study technique - unsupervised learning. By clustering input data using Centroid Neural Network, SAM will detect fuzzy rules needed for approximating specific nonlinear function.

- **Optimal learning**: This optional stage enables SAM to eliminate unnecessary fuzzy rules using the genetic algorithm. SAM will be more concise in order to enhance the processing speed and accuracy.

- **Parameter learning**: This stage employs the supervised learning. By gradient descent method incorporated with momentum, SAM will tune its parameters such as rule weights, volumes and centroids as long as error between system's outcomes and expected results reach the desired criterion.

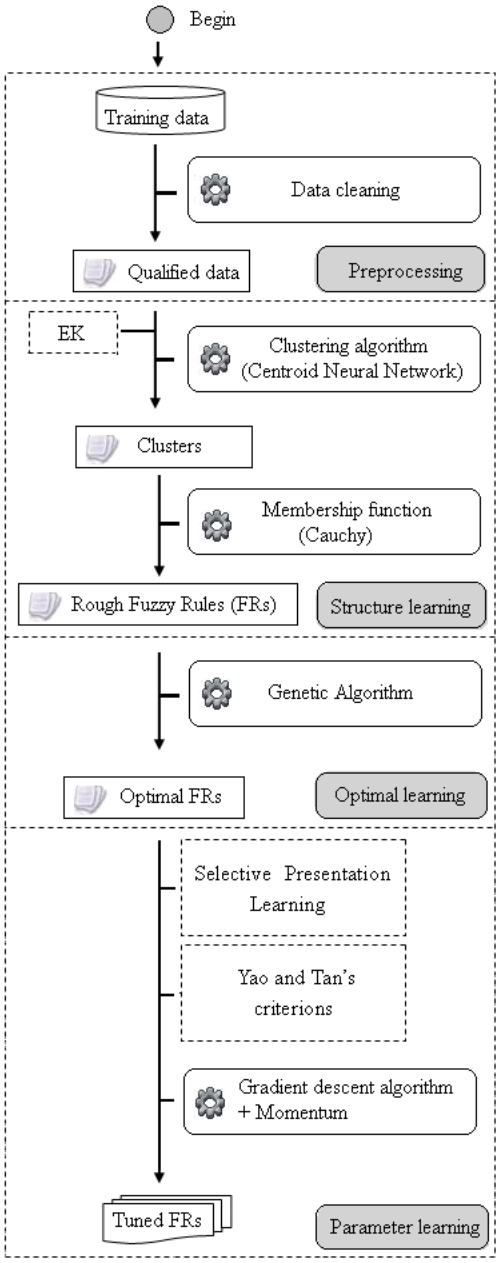

Figure 5: The general learning process of SAM

3.1 Structure learning

Unsupervised learning begins with categorizing data patterns into clusters. The purpose is that from a limited set of learning data patterns, we establish a set of clusters in which data patterns in a cluster are as same as possible and data patterns in distinct clusters are as different as possible.

Park's Centroid Neural Network (CNN) algorithm [11] has been applied in this research. The CNN algorithm is originated from the conventional k-means algorithm and finds the centroid of data in corresponding clusters at each presentation of data vector. In lieu of calculating the centroids of the clusters while every data is being presented, the CNN algorithm updates their weights only when the status of the output neuron for the presenting data has changed: that is, the weights of the winner neuron in the current epoch for the data change only when the winner neuron did not win the data in the previous presentation and the weights of the winner neuron in the previous epoch for the data change only when the neuron does not win the data in the current epoch. We call the former one a "winner neuron" and the latter one a "loser neuron". When an input vector x is applied to the network at time n, the weight update equations for winner neuron j and loser neuron i in CNN can be written as follows:

$$\begin{aligned} w_j(n+1) &= \frac{1}{N_j+1}\left[N_j w_j(n) + x(n)\right] \\ &= w_j(n) + \frac{1}{N_j+1}\left[x(n) - w_j(n)\right] \end{aligned} \tag{5}$$

$$\begin{aligned} w_i(n+1) &= \frac{1}{N_i-1}\left[N_i w_i(n) - x(n)\right] \\ &= w_i(n) - \frac{1}{N_i-1}\left[x(n) - w_i(n)\right] \end{aligned} \tag{6}$$

where $w_j(n)$ and $w_i(n)$ represent the weight vectors of the winner neuron and the loser neuron, respectively while N_i and N_j denote the number of data vectors in cluster i^{th} and j^{th} at the time of iteration, respectively.

The learning rule for CNN is based on the following theorem and condition for minimum energy clustering:

- **Theorem 1**: The centroid of data in a cluster is the solution which gives minimum energy in L_2 norm.
- **Minimum energy condition**: The weights for a given output neuron should be chosen in a way to minimize the total distance in L_2 norm from the vectors in its class such as

$$w_j = \min_{w} \sum_{i=1}^{N_j} \|x_j(i) - w\|^2 \tag{7}$$

or by Theorem 1

$$w_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_j(i) \tag{8}$$

where N_j is the number of members in cluster j.

When CNN is compared with conventional competitive learning algorithms, the CNN produces very comparable results with less computational effort. That is, the CNN requires neither a predetermined schedule for learning gain nor a total number of iterations for clustering and it converges stably to suboptimal solutions while conventional algorithms including Self Organizing Map (SOM) may give unstable results depending on the initial learning gain and the total number of iterations.

After clustering data patterns, the next task is to build up fuzzy rules from their centroid vectors using fuzzy sets. Constructing fuzzy rules is stated in previous section.

3.2 Optimal learning

Theoretically, regarding the fuzzy system in general or the SAM in particular, the more number of fuzzy rules the more accuracy in approximation process. Nevertheless, if a system has too many fuzzy rules, it would take a long time in learning process. An optimal system will only keep necessary fuzzy rules. One of solutions for the above problem is using Genetic Algorithm (GA) as follows:

- **Step 1**: Initialize ten chromosomes. Each chromosome is a chain of binary values describing status of corresponding rules in SAM. Value "*0*" means the rule is omitted while value "*1*" means the rule is selected. Every generation only uses ten chromosomes. One of individuals in the first generation contains all rules (all gene values of chromosome are equal to "*1*").
- **Step 2**: Create new chromosomes by crossover (probability 0.5) and mutation (probability 0.01).
- **Step 3**: Use roulette wheel with adaptive function to select ten best chromosomes which have the minimal Fit(.) value.

$$Fit(m) = ln\left(\bar{\sigma}_\varepsilon^2\right) + \frac{log_n(m)}{n} \tag{9}$$

where:

$$\bar{\sigma}_\varepsilon = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - F(x_j)\right)^2 \tag{10}$$

  - m: number of used rules.
  - n: number of training data samples.

- **Step 4**: If the stopping condition (i.e. expected error) is not satisfied, return step 2.
- **Step 5**: Choose the best one in ten chromosomes at the final population.

The found binary chain of the best chromosome will be used for eliminating unnecessary rules.

3.3 Parameter learning

The supervised learning helps decrease error between system's outcomes and expected results. The problem is stated as follows:

- Given input training data set $\{x_j\}$ and expected result set $\{y_j\}, j = \overline{1, ntd}$, where ntd is number of training data.
- Given SAM's fuzzy rules and parameters.

- Adjust parameters of if-part fuzzy set, then-part and weights as long as error between system's outcomes and expected results reaches the stopping criteria.

There are a number of training methods such as the conjugate gradient or GA [4]. The above methods could converge to the minimum with fewer steps than gradient descent method; however, the number of calculations in each step is more. Regarding the SAM often has many parameters, they are inefficient. Thus, the gradient descent algorithm has been deployed in this step. The aim of parameter adjustment learning phase is to minimize the square of error:

$$E(x) = \frac{1}{2}(f(x) - F(x))^2 \tag{11}$$

The learning rule applies for variable ξ in SAM has following form:

$$\xi(t+1) = \xi(t) - \mu_t \frac{\partial E}{\partial \xi} \tag{12}$$

where μ_t is the learning rate. The learning rule for each parameter is expanded in detail as follows.

$$\frac{\partial E}{\partial F} = -(f(x) - F(x)) = -\varepsilon(x) \tag{13}$$

$$c_j(t+1) = c_j(t) + \mu_t.\varepsilon(x).p_j(x) \tag{14}$$

$$V_j(t+1) = V_j(t) + \mu_t.\varepsilon(x).\left[c_j - F(x)\right].\frac{p_j(x)}{V_j} \tag{15}$$

$$w_j(t+1) = w_j(t) + \mu_t.\varepsilon(x).\left[c_j - F(x)\right].\frac{p_j(x)}{w_j} \tag{16}$$

Formulas for tuning parameters of *Cauchy* membership function are as follows:

$$m_j(t+1) = m_j(t) + 2\mu_t\varepsilon(x)p_j(x)[c_j - F(x)]\frac{x - m_j}{d_j^2}a_j(x) \tag{17}$$

$$d_j(t+1) = d_j(t) + 2\mu_t\varepsilon(x)p_j(x)[c_j - F(x)]\frac{(x - m_j)^2}{d_j^3}a_j(x) \tag{18}$$

All initial configuration parameters are chosen based on clusters after unsupervised learning. The momentum technique is also integrated in the parameter tuning process. This helps the supervised learning to avoid local minimum cases and to reduce the learning time. The learning formula with momentum is as follows

$$\xi(t+1) = \xi(t) - \mu_t \frac{\partial E}{\partial \xi} + \gamma\Delta\xi(t) \tag{19}$$

where γ is the momentum coefficient.

4. EXTRACTING EVENT-KNOWLEDGE

Knowledge of the events is divided into two kinds: negative-event-knowledge for events which tend to reduce stock prices and positive-event-knowledge for events which tend to raise them. Event-knowledge (EK) is non-deterministic knowledge showing only tendencies for stock price prediction. Actual stock price movement is influenced by a wide range of factors including numerical economic indicators and the EK. Therefore, the direction predicted using only EK does not always correspond to actual direction of next-day's stock price movement. In Kohara's experiments, the directions indicated by extracted EK corresponded to the actual next-day's direction approximately 60% of the time. The EK is stated abstractly by IF-THEN rules as follows:

Rule 1: IF (domestic politics = good) THEN (event = positive-event)

Rule 2: IF (domestic politics = bad) THEN (event = negative-event)

Rule 3: IF (business prospects = good) THEN (event = positive-event)

Rule 4: IF (business prospects = bad) THEN (event = negative-event)

A set of above rules could be expanded with more factors to constitute prior knowledge of system. The detailed extracting EK from time series algorithm includes following steps:

- **Step 1**: Make smooth series by single exponential technique [1].
- **Step 2**: Compute the difference values between smoothed and original values.
- **Step 3**: Normalize difference values into [0, 1].
- **Step 4**: Assign normalized value at (t+1) to value EK(t).

Smoothing allows the model series-data to be represented as a really nonlinear function that is assumed to present the trend of time series in normal condition. The normalized difference value at (t + 1) shows the influence-degree of summarized events at (t) onto time series at (t + 1). The near 1 (one) value presents influence-degree of positive-EK whereas the value 0 (zero) represents the opposite case. An EK value of 0.5 presents the normal condition or the balance of influence degree between positive-EK and negative-EK. Application of EK into SAM fuzzy system is depicted in Fig 6.

5. EXPERIMENTS

The first Abalone data set [2] aims at the task of trying to predict the number of rings in the shells of abalone (which is related to their age) based on a series of biometric measures of these animals (1 nominal and 7 continuous attributes as shown in Fig. 7). The ring value adds 1.5 gives the age in years.

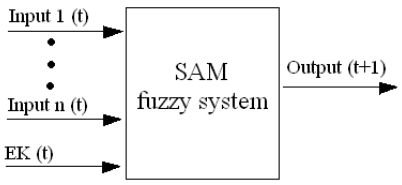

Figure 6: EK(t) is fed to the system to predict value (t+1)

Table_Abalone : Table

	ID	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
1	M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15
2	M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7
3	F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9
4	M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10
5	I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7
6	I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8
7	F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20

Record: 8 of 8

Figure 7: The database shows all fields of Abalone data

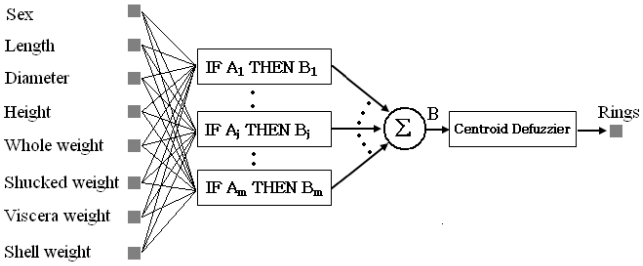

Figure 8: SAM's structure for forecasting the ring of Abalone

Fig. 8 describes the structure of SAM for predicting the ring. The squares on the left represent the input nodes in first layer. There are eight fields of Abalone data, the input vector thus is 8-dimensional. The last node on the right is ring as well as predicted value - the output of system. The similar structure could be applied for forecasting any field based on other available fields from database. Regarding to nominal variables, they must be reflected into normalized numeric value in [0,1], to standardize the scale of effect each variable.

The M (male), F (female) and I (infant) value of field sex are mirrored of 1, 0, and 0.5 respectively. The Abalone dataset is divided into training set which includes 3777 samples and test set with 400 last samples. The performance of SAM and Multi-Layer Perceptron Neural Network (MLPNN) in terms of MAE is evaluated on this test set. We do not use validation set because we want to have a fair evaluation between SAM and MLPNN.

We have used Fuzzy logic Toolbox of MATLAB to employ MLPNN model with number of nodes in hidden layer as shown in Table 1. The configuration of training process of SAM and MLPNN is identical. In detail, the learning rate is 0.0001; the number of epochs changes linearly with number of clusters; momentum coefficient is 0.9. The more of clusters, the larger number of epochs should be. In parameter learning, SAM uses conventional gradient descent algorithm whereas MLPNN uses Levenberg-Marquardt algorithm.

No.	Weight - wj	Volume - Vj	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6	Center 7	Center 8	...
1	0.9698787...	0.9698787...	0.4861856...	0.7616634...	0.4233826...	0.1478056...	0.8534082...	0.3292734...	0.1753195...	0.260263479520905...	...
2	1.4225815...	1.4225815...	0.4348625...	0.0797126...	-0.007823...	0.0188296...	-0.207360...	0.7583138...	0.4743455...	0.013068558796045...	...
3	0.8362397...	0.8362397...	0.5926872...	0.6956970...	0.4756587...	0.1800221...	1.0349788...	0.0925382...	0.5678863...	0.579004102012372	...
4	1.0095548...	1.0095548...	0.5354582...	0.5188797...	0.3961916...	0.0374774...	0.6901840...	0.3191644...	0.1578594...	0.170058636089172 6	...
5	1.0698845...	1.0698845...	0.6893875...	0.6799026...	0.5368590...	0.2785239...	1.5576919...	0.3696598...	0.2648838...	0.523959548434320 9	...
6	0.8990264...	0.8990264...	0.7387505...	0.5422856...	0.5411803...	0.4216669...	1.5490875...	1.0412521...	0.3989522...	0.299676314872929	...

Figure 9: If-part fuzzy rules after training

Fuzzy rules - clusters - have various widths and centers according to tuning them in parameter learning as shown in Fig. 9. Fig. 11 and 12 illustrate four first *Cauchy* membership functions after training process in case of 60 clusters. Conventional fuzzy systems often define membership functions as well as linguistic terms in advance. This task requires much experiences and the system becomes inflexible unintentionally. The SAM with strong learning process could determine better parameters of function. Thus, prior setting is not needed and causes less efficient.

Table 1: The results of comparison between MLPNN and SAM in terms of predicting error (MAE) on test set

Clusters (hidden nodes)	Epochs	MLPNN	SAM
50	800	2.1898	1.42
60	1000	2.0272	1.3901
70	1200	2.4919	1.415
80	1400	2.4312	1.4332
90	1600	3.1206	1.4214

Table 1 exhibits the predicting errors on test set of normal SAM (without Optimal learning) are always smaller than one of MLPNN. Choosing 60 clusters is the best choice with smallest error: 1.3901. After SAM's structure learning stage, parameters are determined directly from results whereas there is no method to initialize them in MLPNN except randomization. Each fuzzy rule weight w_j is normalized in [0:1] at the beginning, it gets largest value if the fuzzy rule - fuzzy patch - has most data vectors. The SAM's error point hence is nearer optimum point in performance surface. This assertion is proven by Table 2, MEA values before performing parameter learning on training set of SAM outperforms definitely one of MLPNN. The illustration of gradient descent algorithm in training process is shown in Fig. 10 with 42 fuzzy rules obtained by GA in case of initial 60 fuzzy rules. The number of generations in GA is 5. It is chosen by experience with an assumption is sufficient to eliminate redundant fuzzy rules and not cause over-fitting. Table 2 also shows the optimal number of fuzzy rules after applying GA.

In each epoch, SAM tunes parameters of every fuzzy rule which included a set of certain data vectors, whereas MLPNN tunes them on each vector. The performance function of SAM thus is simple and has less error valleys than MLPNN, this makes the convergence easier and seldom fall into local minimum. From this characteristic, we could select properly which parameter is appropriate for training, training

Table 2: Initial MAE values on training set

Clusters	MLPNN	SAM	Optimal SAM	rule no.
50	49.6257	4.0815	2.9848	33
60	16.1679	3.3071	2.7152	42
70	39.8172	3.9346	2.8406	51
80	54.4544	3.7585	2.7881	55
90	52.7509	2.3252	1.9949	56

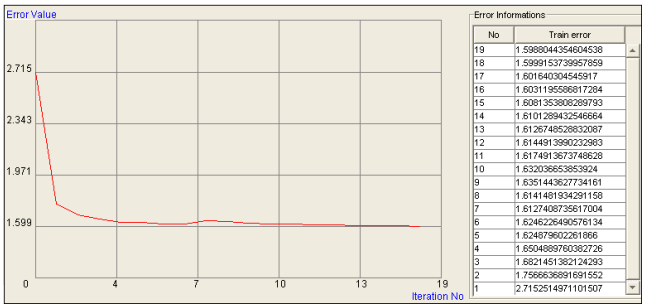

Figure 10: MAE information while training by gradient descent algorithm with 42 fuzzy rules

of all parameters is not needed. There are some outlooks that the more clusters we use, the more accuracy the results get. This point is not always exact while we predict for a long term in financial time series. Using more clusters, the system could be turned into over-fitting status which only suitable for predict a short term.

The configuration of SAM fuzzy system in five datasets is identical. In detail, SAM uses 20 fuzzy clusters, learning rate is 0.0001, momentum coefficient is 0.9, number of generation in GA is 5 and training cycle is 4000 epochs. The dimension of an input pattern is 4, it means four historical values in time series produce one predict value. This number is chosen by knowledge that stock price of current day is affected by last four days and not cause over-fitting while using large number. Note that with models which use EK, the dimension of an input pattern is also 4, but the last element is EK value.

6. CONCLUSION

In lieu of employing the traditional models for forecasting data series, this paper presented an application of a newly popularized technique: the SAM fuzzy inference system. We obtained the optimal SAM with least number of fuzzy rules by using GA. The results of experiment on prediction the age of Abalone showed that SAM offers an preferable solution and can be definitely an alternative approach for conventional models such as MLPNN. This comparison does not mean that SAM is constantly better than neural networks on all applications. As regards financial time series, we have used event-knowledge and new training criteria to improve the ability to predict large changes. The results of several experiments on stock market prediction showed that model EK_SEL outperforms conventional approaches. We understand that success in several kind of stock prices does not necessarily mean the same for others and the financial market often comprises many complicated factors. Nevertheless, our approach could be seen as new research direction involves demands of economic forecasting. A general drawback of the SAM model is time-consuming in processing but when the information technology has been advanced rapidly, SAM in particular and machine learning techniques in general show their absolute primacy.

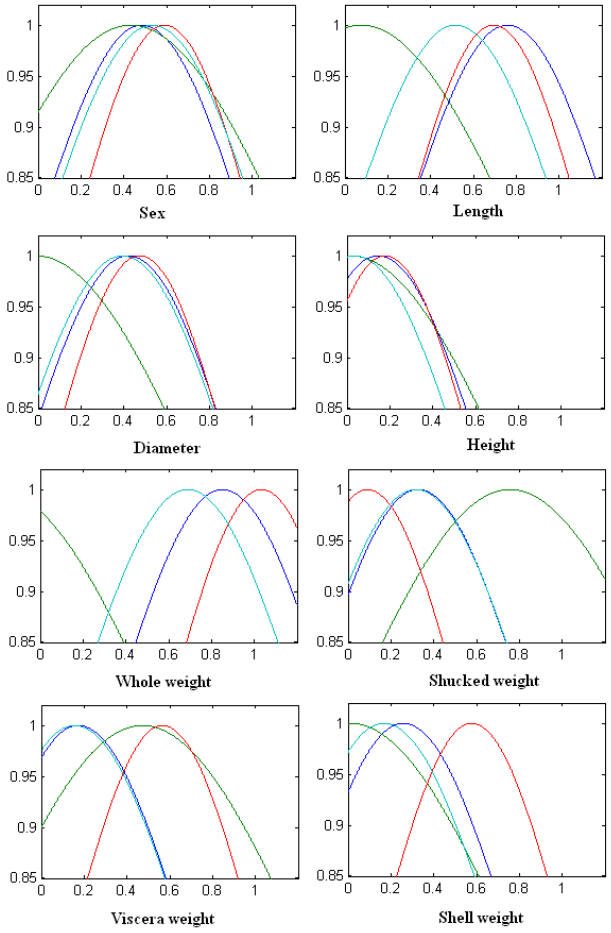

Figure 11: Membership functions of four first fuzzy rules on each input variable

7. REFERENCES

[1] T. Cipra, J. McKenzie, and E. McKenzie. Exponential smoothing for irregular data. *Journal of Applications of Mathematics*, 51(6):597–604, 2006.

[2] M. J. Gacto, R. Alcala, and F. Herrera. Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 13(5):419–436, 2008.

[3] H. Harchegani. Using genetic algorithm for the prediction of information for cache operation in mobile databases. *MASAUM Journal of Computing*, 1:314–317, 2009.

[4] R. K. Jong and U. J. Do. Optimizing the fuzzy classification system through genetic algorithm. In *Proc. Int. Conference on Convergence and Hybrid Information Technology*, pages 903–908, 2008.

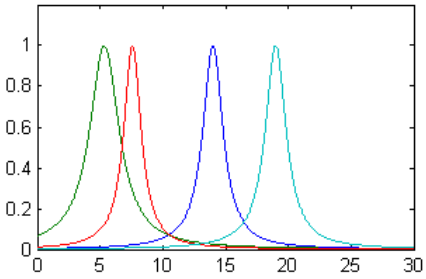

Figure 12: Membership functions of four first fuzzy rules on output variable

[5] F. Khan, R. Anwer, O. Torgersson, and G. Falkman. Data mining in oral medicine using decision trees. *World Academy of Science, Engineering and Technology*, 37:225–230, 2008.

[6] B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall PTR, 1991.

[7] B. Kosko. *Fuzzy Engineering*. Prentice Hall PTR, 1996.

[8] D. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons, Inc., 2004.

[9] S. Mitaim and B. Kosko. The shape of fuzzy sets in adaptive function approximation. *IEEE Trans. on Fuzzy Systems*, 9:637–656, 2001.

[10] X. Ni. Research of data mining based on neural networks. *World Academy of Science, Engineering and Technology*, 39:381–384, 2008.

[11] D. Park. Centroid neural network for unsupervised competitive learning. *IEEE Trans. Neural Networks*, 11:520–528, 2000.

[12] S. Zhang, C. Tjortjis, X. Zeng, H. Qiao, L. Buchan, and J. Keane. Comparing data mining methods with logistic regression in childhood obesity prediction. *Information Systems Frontiers*, 11(4):449–460, 2009.

[13] Y. Zhang, M. Fraser, R. Gagliano, and A. Kandel. Granular neural networks for numerical-linguistic data fusion and knowledge discovery. *IEEE Trans. on Neural Networks*, 11:658–667, 2000.