Linguistic granular model: design and realization*

YUE Shihong[1,2]**, LI Ping[2] and SONG Zhihuan[2]

(1. Institute of Industrial Process Control, Zhejiang University, Hangzhou 310027, China; 2. School of Electric Engineering and Atomization, Tianjin University, Tianjin 300072, China)

Received November 2, 2004; revised December 28, 2004

Abstract A new linguistic granular model is proposed and the effect of its parameters on the output is analyzed. The design of the model consists of two stages: using conditional fuzzy clustering for information granular, and integrating all information granules to final output. The integrating tool is fuzzy integral based on fuzzy measure, and the generalization of fuzzy integral increases flexibility of the linguistic granular model greatly. A heuristic algorithm to determine the parameters in the fuzzy integral is used to realize the linguistic model. Two experiments verify the feasibility of the proposed model.

Keywords: **linguistic model, granular, fuzzy clustering, fuzzy integral.**

Linguistic granular model[1−5] has emerged as an interesting, attractive and powerful modeling environment. In contrast to those numerically linguistic models, linguistic granular model focuses on creating the instructive linguistic granular in data space rather than the numeric formation. Most numerically linguistic models emphasize their accuracy rather than transparency. However, some researchers utilized conditional fuzzy clustering (CFC)[6−8] to construct a user-driven linguistic model, in which the information and requirements of user are represented by a group of context aspects (CAs) from the input to output of the model. The constructed linguistic model could discover the inherent structure of input space, reflect the interaction level and requirements of the user, and provide a generalized assessment for the existing wide linguistic models. Likewise it emerged apparent advantages such as high accuracy, flexibility and sound transparency.

Nevertheless, the constructed linguistic model is far from applicable since it is only a conceptual framework and lacks necessary realized procedure for parameter estimation. To overcome these problems, this study first analyzes these parameters involved in the linguistic model, and then constructs a Neural Network (NN) with a single hidden layer to realize the linguistic model. This constructed NN is created by two steps: first, forming an information granular from input to the hidden layer by CFC; second, enhancing the generalization ability of the linguistic model by using the choquet fuzzy integral (CFI)[9] for integrating the information resources in the hidden layer to the output layer. According to optimizing index and Heuristic Algorithm (HA)[10] for CFI, we adjust these parameters in the NN to their optimal values.

CFI has greatly succeeded in pattern recognition, image process, and etc. Regarding the issue to integrate a number of information resources to their synthetic evaluation, the traditional way is weighted average. The way assumes that there are independency and no interaction in these information resources, but this is not true in real circumstance. Currently, CFI is a most effective way to solve the problem. We shall contrast the equivalency between the CFI and feedforward NN, and emphasize that the CFI is helpful to arise the generalization of linguistic model.

1 Related work

1.1 Linguistic model based on CFC

The linguistic model based on CFC depends on two concepts[8]: CA and CFC. CFC is performed by user-driven CA. If the sum of all membership degrees for any specified data point in CFC is always equal to 1, CFC reduces the classical Fuzzy c Means (FCM) algorithm[7]. Considering that fuzzy set is a popular way to represent linguistic granule, thus we always use it as the specified form of linguistic model. As far as the Fuzzy Set for CA (FSCA) is given, the CFC

* Supported by the Hi-Tech Research and Development Program of China (Grant No. 2002AA214010)
** To whom correspondence should be addressed. E-mail: xlqz@vip.sina.com.cn

can find these clusters or granules in the corresponding input space (see Fig.1 (a)).

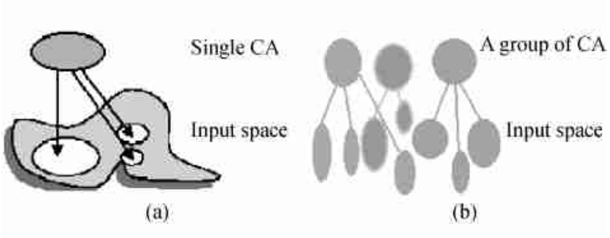


Fig. 1. Framework of linguistic model. (a) Partition input space by a CA; (b) partition input space by a group of CAs.

The system of FSCA must create a fuzzy partition of output space. Using the membership degree of CFC, any FSCA corresponds to a number of clusters in input space, and all FSCA in output space discover the distribution structure of input space (see Fig. 1 (b)). As usual, the simulation based fuzzy rule is confined by the well-known dimensionality curse. The linguistic model can avoid the crux efficiently since the clustering in linguistic model is not directly related to the dimensionality of space. These illustrations can be formulated as follows:

Let z_{ij} be the output decided by j-th cluster for i-th FSCA denoted as A_i. According to different membership degrees given by a system of FSCA, the output of the model is

$$\begin{aligned} y = &\{z_{11} \otimes A_1 \oplus \cdots \oplus z_{1n_1} \otimes A_1\} \\ &\oplus \{z_{21} \otimes A_2 \oplus \cdots \oplus z_{2n_2} \otimes A_2\} \\ &\oplus \cdots \oplus \{z_{c1} \otimes A_c \oplus \cdots \oplus z_{cn_1} \otimes A_c\}. \\ &c_j = 1, 2, \cdots, c, \qquad (1) \end{aligned}$$

where $\otimes$, $\oplus$ are standard fuzzy operators, n_k is the number of granular for k-th FSCA, $k = 1, 2, \cdots, c$. Using triangular fuzzy set $A_i = (a_{i-}, a_i, a_{i+})$ for FSCA, Eq.(1) can be rewritten as

Lower bound:

$$\begin{aligned} y = &\{z_{11} \otimes a_{1-} \oplus z_{12} \otimes a_{1-} \oplus \cdots \oplus z_{1n_1} \otimes a_{1-}\} \\ &\oplus \cdots \oplus \{z_{c1} \otimes a_{c-} \oplus z_{c2} \otimes a_{c-} \\ &\oplus \cdots \oplus z_{cn_1} \otimes a_{c-}\}. \end{aligned}$$

Middle value:

$$\begin{aligned} y = &\{z_{11} \otimes a_1 \oplus z_{12} \otimes a_1 \oplus \cdots \oplus z_{1n_1} \otimes a_1\} \\ &\oplus \cdots \oplus \{z_{c1} \otimes a_c \oplus z_{c2} \otimes a_c \\ &\oplus \cdots \oplus z_{cn_1} \otimes a_c\}. \end{aligned}$$

Upper bound:

$$\begin{aligned} y = &\{z_{11} \otimes a_{1+} \oplus z_{12} \otimes a_{1+} \oplus \cdots \oplus z_{1n_1} \otimes a_{1+}\} \\ &\oplus \cdots \oplus \{z_{c1} \otimes a_{c+} \oplus z_{c2} \otimes a_{c+} \\ &\oplus \cdots \oplus z_{cn_1} \otimes a_{c+}\}. \end{aligned}$$

Thus both the input and output in the linguistic model are linguistic granules. In a 2-D input-output plane, the output curves of linguistic model are band-shaped area enclosed by its lower and upper bounds. For different input values the bandwidth of the linguistic model is different.[10]. The main problems in the linguistic model are:

(i) It is only a conceptual framework and lacks a sound realized algorithm. In fact, CFC can result in the linguistic granules or clusters in input space, but no guideline in it tells how to partition the output space and how to integrate the obtained granules, and etc.

(ii) Its results are determined by parameters used. These parameters include the position of prototypes, fuzzy exponents and number of clusters in CFC. Without the analysis of these parameters, the model cannot be performed well under any circumstances.

(iii) The fuzzy operators "$\otimes$, $\oplus$" are difficult to be optimized for their piecewise representative and limited applicable conditions, despite that they can act as the way to integrate the linguistic granules.

(iv) In terms of the position of approximation, Eq.(1) is a fuzzy approximation system and has been proven to have advantages over the typical linear fitness system. However, what is the difference between Eq.(1) and fuzzy or other approximation system? Furthermore, how can these information and requirements of user lead the model to the expected results? Here our work will face these problems.

1.2 Choquet fuzzy integral

Let $X = \{x_1, x_2, \cdots, x_n\}$ be a limited set with n input resources, $P(X)$ the power set of X, i.e. the subset of X with arbitrary elements. The fuzzy measure defined on X is a function[9] $: g : P(X) \mapsto [0, 1]$, such that

$$g(\Phi) = 0, \quad g(X) = 1;$$
$$g(A) \leqslant g(B), \text{ if } A, B \subset P(X) \text{ and } A \subseteq B.$$

Denoting CFI as $E_g = \int_X (\circ) \circ g(\circ)$, the discrete form of CFI is defined by[10]

$$\begin{aligned} E_g &= \sum_{i=1}^{n} (x_i^* - x_{i-1}^*) g(A_i^*) \\ &= \sum_{i=1}^{n} (g(A_i^*) - g(A_{i+1}^*)) x, \end{aligned}$$

$$\text{s.t. } x_1^* \leqslant \cdots \leqslant x_{n+1}^*, \ x_0^* = 0,$$
$$g(A_{n+1}^*) = 1, \tag{2}$$

where x_1^*, $\cdots$, x_n^* satisfy an increasing order of x_1, $\cdots$, x_n after rearranging them, $A_i^* = \{x_i^*, x_{i+1}^*, \cdots, x_n^*\}$, $i = 1, 2, \cdots, n$.

A single CFI is equivalent to a singe output NN without any hidden layer. The key using CFI is to determine its parameters by a realized algorithm. An inaccurate algorithm will make it invalid[8,9]. Currently HA is the most effective way for this purpose[10]. Its basic idea is that the most effective way is arithmetic average in the absence of sufficient information for the parameter estimation of CFI, i.e. these parameters in CFI are taken as equadistributed ones. In the process of determining these parameters, HA is performed in each ordinal subspace of input space in which all data vectors have a unique order.

2 Structure and parameter estimation of linguistic model

2.1 Topological structure and realized way of linguistic model

Based on the above analysis, we propose an NN-based linguistic model, and Fig. 2 illustrates its topological structure.

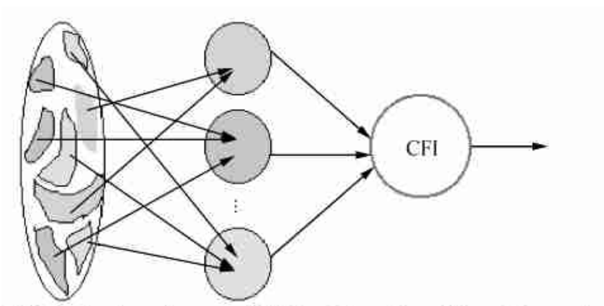


Fig. 2. Topological structure of the new linguistic model.

The NN consists of three layers: input layer, hidden layer with FSCA units, and output layer. The structure of new linguistic model and running procedure are as follows:

(i) Perform FCM clustering algorithm[3] in output space where the fuzzy exponent is taken as 2; in light of the obtained membership degrees of all vectors, we can find fuzzy partition of output space in an automatic way and thereby form a group of FSCA. In case of many output variables—where the partition of output by artificial forms becomes very slow and infeasible, this operation is essentially attractive. Likewise we use linguistic variables such as "large", "very small", etc. to translate the users' requirement into the system of FSCA.

(ii) Perform CFC to obtain the linguistic granular for any FSCA. Here, FSCA acts as the connected weights from input to output.

(iii) Integrate the information granules in the hidden layer by CFI. Modified HA algorithm can identify the weight values between them (see Appendix 1). According to the above step, the initial weight values of the linguistic model are determined.

For any input vector x, we denote $u_1^q(x)$, $u_2^q(x)$, $\cdots$, $u_{n_q}^q(x)$ as the membership degrees of the granules from qth FSCA, h_1^q, h_2^q, $\cdots$, $h_{n_q}^q$ as prototypes in output space, and one-to-one corresponding to the prototypes in input space: v_1^q, v_2^q, $\cdots$, $v_{n_q}^q$, $q = 1, 2, \cdots, c$. Thus the output $\hat{y}$ of the linguistic model can be represented by the following weighted sum:

$$\hat{y} = \sum_{q=1}^{c} \sum_{i=1}^{n_q} h_i^q u_i^q(x),$$
$$\text{s.t. } u_i^q(x, v_i^q) = A_q(x) \Big\backslash \sum_{j=1}^{n_q} (\| x - v_i^q \| / \| x - v_j^q \|)^{2/(m-1)}, \tag{3}$$

where $A_q(x)$ is the membership degree of qth FSCA associated with x; $\hat{y}$ is the linguistic granules determined by the triangular fuzzy set $A_q(x)$.

Without the fuzzy operators such as "$\otimes$, $\oplus$", the obtained linguistic model can be easily optimized, and the optimization aims at adjusting the prototype position and number of cluster in CFC. Optimizing these parameters is necessary for the application since they play a great role in the output of linguistic model. Noting that the output of the linguistic model is a granular form, thus we must change the standard Mean-Square-Error (MSE) criterion to an acceptable numeric form. For a given group of sample vectors (x_i, y_i), $i = 1, 2, \cdots, N$, we let y_i^M be the middle value of ith output fuzzy set associated with ith input. Thus the MSE is

$$\text{Min} E = \left(\sum_{i=1}^{N} (y_i - y_i^M)^2 / N \right)^{1/2},$$
$$i = 1, 2, \cdots, N, \tag{4}$$

where N is the total number of sample vectors. In this optimizing process the AM as well as the gradient descended algorithm is performed alternatively till the least error is encountered. This procedure for optimization is shown in the Appendix. We call this linguistic model (LMCFC) based on CFC. Simultaneously, the number of clusters is added from a smaller initial value to its upper bound to prevent the infinite recycle of procedure. LMCFC sets an upper bound of number of clusters, $c_{\max}$, to force the procedure stop, which satisfies the following equation[11]: $c_{\max} \leqslant \sqrt{n}$, where n is the number of samples.

2.2 Parameter sensitivity of LMCFC

In LMCFC there exist three parameters: fuzzy exponent m, prototype position, and number of clusters in CFC. When these prototype positions are fixed, the output of LMCFC makes rather large differences as m is varied (see Fig. 3 (a)).

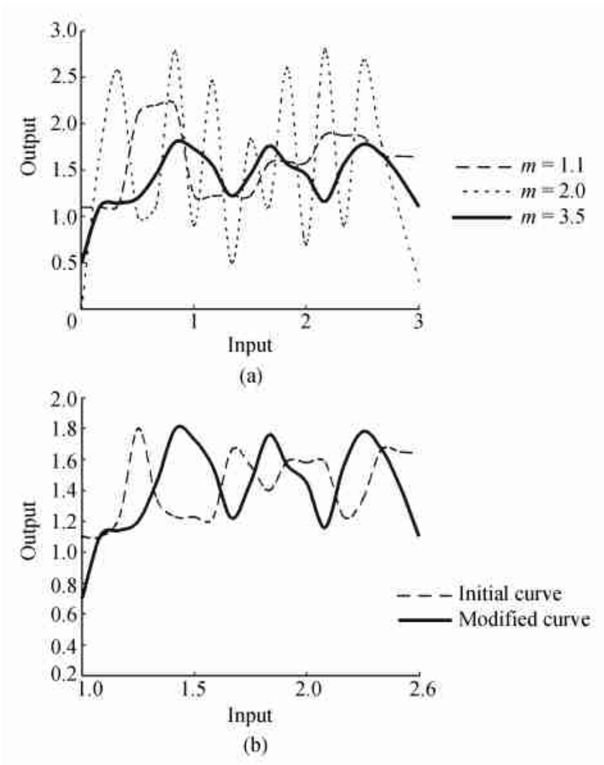


Fig. 3. Effect of different parameters. (a) Effect of fuzziness exponent m, (b) effect of prototype positions.

The simulation demonstrates that the LMCFC is a nonlinear mapping from input to output as m is varied. When m is close to 1, LMCFC gradually reduces a piecewise linear function. When m is larger than 3, the model looks like close to a high-frequency varied function, and does not make against the fine approximation for arbitrary function. Nevertheless, when m is equal to 2, LMCFC provides a preferable smooth curve whereby to help approximate most of functions. Therefore, one takes $m=2$ for granted. In fact, that $m=2$ is beneficial to better approximation had been proven theoretically to a great extent[11].

Likewise the output of LMCFC greatly depends on the prototype positions in these FSCA. A very small variance of them can lead to large difference of output values of LMCFC. Fig. 3 (b) shows that the approximation of LMCFC for any function can be greatly affected by adjusting the number of clusters associated with FSCA. However, when sufficient samples are available, the larger the number of clusters is, the more details are captured.

In the meaning of approximation, LMCFC has either a similarity or difference compared with the existing fuzzy system and NN:

(i) The structure is similar with the one of additive fuzzy system. Note that the optimal fuzzy rule "pitch" must cover the extreme and inflection points of the approximated function[24—26], where the pitch means the nonzero area of fuzzy membership degree. In particular, more pitches should cover these highly nonlinear positions while fewer pitches do these positions with slow variance to save the resources. However, fuzzy system realizes these requirements by global optimization throughout all sample vectors. Such approximation is often inaccurate regardless of the structure of space while LMCFC finds the data distribution consisting of information granules in its own.

(ii) The former performs a global approximation while the latter is a local approximation. Fuzzy approximation partitions each axis to construct fuzzy rules, but these rules are local since each rule is limited to a pair of input-output fuzzy set by specified positions. Therefore some inherent relations in data vectors are cut down or cannot be resumed. As opposed, LMCFC performs a global search in the whole input space to form linguistic granules[6]. Furthermore, fuzzy system has to face dimension curse for its number of rules is directly related to the dimension, whereas the LMCFC can avoid this problem to a great extent for the clustering in it is not related to dimension when it creates fuzzy rules.

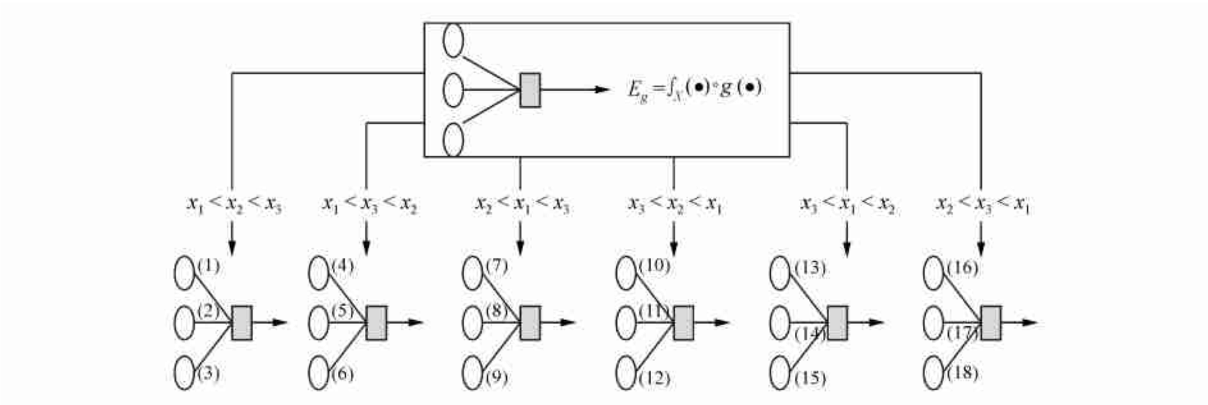
(iii) Both input and output are linguistic granules while the latter can act only for numeric information. Hence the former can model any information that can be granulated, such as multimedia data, speech signals, and etc.

(iv) FSCA has the action of "filter" since it emphasizes user-driven requirements by large membership degrees of points and confines the useless data point by small ones.

(v) The familiar redial basis function (RBF)[13, 14] NN is a bit similar to LMCFC. The nonzero area of membership degree in CFC is very similar to the acceptable field in RBF. The action that CFC solves a group of fuzzy membership functions is equivalent to that RBF optimizes the acceptable field parameters. However, there exist substantial differences between them. RBF NN can act for only the numeric area, and lacks reasonable transparency and is difficult to determine its size.

2.3 Equivalency between CFI and feedforward NN

According to the computational procedure of CFI, it must divide input space into different subspaces so that all data points in any subspace have a unique order of components. In total there exist $n!$ orders for n input resources. For a CFI with n inputs and single output, it in fact is equivalent to $n!$ feedforward NNs associated with $n!$ ordinal subspaces. For instance, an input vector in a 3-D space is equivalent to $3! = 6$ feedforward NN, and the weights of each feedforward NN consist of different permutation of three same input resources, x_1, x_2, x_3 (see Fig. 4).



(1) $g_{123} - g_{23}$; (2) $g_{23} - g_3$; (3) $g_3 - g_4$; (4) $g_{132} - g_{32}$; (5) $g_{32} - g_2$; (6) $g_3 - g_4$; (7) $g_{213} - g_{13}$; (8) $g_{13} - g_3$; (9) $g_3 - g_4$; (10) $g_{321} - g_{21}$; (11) $g_{21} - g_1$; (12) $g_1 - g_4$; (13) $g_{312} - g_{12}$; (14) $g_{12} - g_2$; (15) $g_2 - g_4$; (16) $g_{231} - g_{23}$; (17) $g_{31} - g_1$; (18) $g_1 - g_4$.

Fig. 4. Equivalency between CFI and feedforward NN.

The above analysis demonstrates the high generalization of CFI. $n!$ tends to infinity as n increases, thus this makes a great number of feedforward NN. For instance, if $n = 4$, $n! = 24$, and if $n = 8$, $n! = 40320$. It is the objective of user to use the smallest size of NN to attain extreme of the objective function associated with all trained sample vectors, and this is the reason why we choose the CFI to integrate these linguistic granules. The user does not need to worry about how to determine the accurate weight of these $n!$ NN in which there exist $2n$ parameters in total, because this problem has been solved by AM well. The flow chart of the parameter estimation for LMCFC is shown in Fig. 5.

3 Experiments

3.1 Comparison between RBF NN and LMCFC

An NN named GRBF[13] based on RBF had performed the classification for the standard test set—Iris dataset. We redo the experiment to compare the LMCFC with the GRBF, and also additive fuzzy system.

3.1.1 Data description The Iris dataset in a 4-D space consists of three clusters. One of the three clusters is well separated from the other two, which are not easily separable due to the overlapping of their convex hulls. The 150 samples contained in the Iris data set are randomly split to form the training and testing sets, each containing 75 samples.

3.1.2 Algorithm description The center of RBF in GRBF is determined by the clustering algorithm named ELEANNE[7], by which the input space is partitioned in a hierarchical way to find the best number of RBF (this equals the total number of clusters in LMCFC). The width of RBF is heuristically computed by averaging inputs. According to the feedback

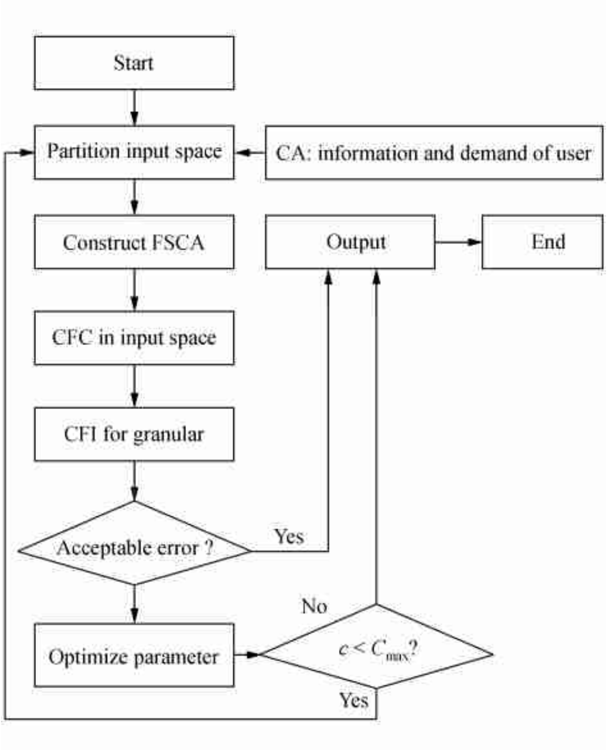


Fig. 5. Flaw chart of parameter estimation of LMCFC.

and stop criteria the prototypes in the training process are partitioned hierarchically. Since there exist only three samples in the output space (i.e. three labels of clusters), LMCFC directly partitions the output space by 1/2 overlapping triangular fuzzy sets to form FSCA, and the obtained fuzzy sets are used to simulate these requirements and interaction levels of user. After performing the CFC and the identification procedure of CFI, the initial weight values are obtained. The refinements of these weights by gradient descend make these prototype positions move on to a great extent. Here the fuzzy exponent is taken as 2. In addition, the used fuzzy system is TS model[13] in which the final output is a weighted average of all outputs of all fuzzy rules.

3.1.3 Results The missing sample number in three algorithms is shown in Table 1. It indicates that the missing number of samples in the training set varies from 0 to 6 when performing GRBF NN, and the corresponding testing set has only three missing samples. The LMCFC has less missing samples than 4 in both training and testing sets. Hence, they have nearly the same accuracy. However TS makes 3 to 16 missing samples, which is attributed to the locality of fuzzy rules in TS model. Both training time and space complexity of the LMCFC are far less than the GRBF NN in spite of the performance in the small size of training set with 75 samples (see Table 1). In general, the fuzzy approximation for numeric area has less time cost and space complexity, but it comes out rather large errors. Note that the initial setting of LMCFC parameters is usually coarse. Interesting enough, no matter how we take different groups of FSCA, the LMCFC always tends to the same final parameters. This demonstrates that LMCFC is very robust. Furthermore, the obtained parameters are very close to the optimal ones reported in some literature where few missing samples in iris dataset were made[16,17]. Thus it is optimization that can automatically find the best parameter and natural relation in data vectors. As seen, the LMCFC has high generalization since it uses only few neurons to attain the expected error criterion while the GRBF must use more neurons (see Table 1).

Table 1. Comparison among three approaches

		E_{training}	E_{Testing}	Time cost (s)	Space cost (m)
GRBFNN		6	3	15	6.3
Fuzzy approx.		7	4	9	1.2
LMCFC	$c=3$	6	7	7	1.0
	$c=5$	5	5	10	1.7
	$c=7$	3	3	11	2.2
	$c=9$	4	2	12	2.6

3.2 Test on speech signal

This experiment is performed for the comparison between the linguistic models with and without optimization procedure (the former refers to the model in Ref. [8]), and as well as GRBF NN. The testing data come from a signal trace problem in MPG database[15]. To identify the speaker, one must trace his/her speech signal in advance. The obtained results can server as the objective or control signals. The initial signal of speaker can be simulated by Eq.(5) below. In the experiment, we divide the signal into time series. Tracing the signal needs to separate it from noise circumstance and mixed outliers. To assure the accuracy, all values of four parameters in the signal are recorded, including frequency, position, width, and time delay. These four systems of values are synthesized to the output of LMCFC, and compared with two other approaches as well.

3.2.1 Data description The signal is simulated by a group of Guessian functions G_b with fixed radius and center. The center is limited in interval [−0.2, 1.2]. G_b can be formulated as

$$G_b = b \circ G,$$

$$\text{s.t. } G := \left\{ \frac{e^{-50(x-t)^2}}{\sqrt[4]{\pi/100}} \mid x, t \in [-0.2, 1.2]; \forall t, \| g(t) \| = b \right\}. \quad (5)$$

The used control strategy is the typical PID control and $b=22$.

3.2.2 Algorithm description Here LMCFC uses trapezoid-shaped fuzzy set. The trapezoid is isosceles in which there exist 1/2 overlapping positions with each other, and the obtained fuzzy sets are used to simulate these requirements and interaction levels of user. After creating the initial LMCFC, the optimization procedure makes final prototype positions greatly move on (see Fig. 6 (a)). LMCFC uses the following time series to trace the initial sign:

$$x(t) = F(x(t-1), x(t-2), x(t-3), x(t-4)). \quad (6)$$

Namely LMCFC predicts the current output by top four of past records. $F(\circ)$ is an arbitrary function and here taken as arithmetic average of input variables. These parameters in PID are adjusted in light of the current obtained feedback.

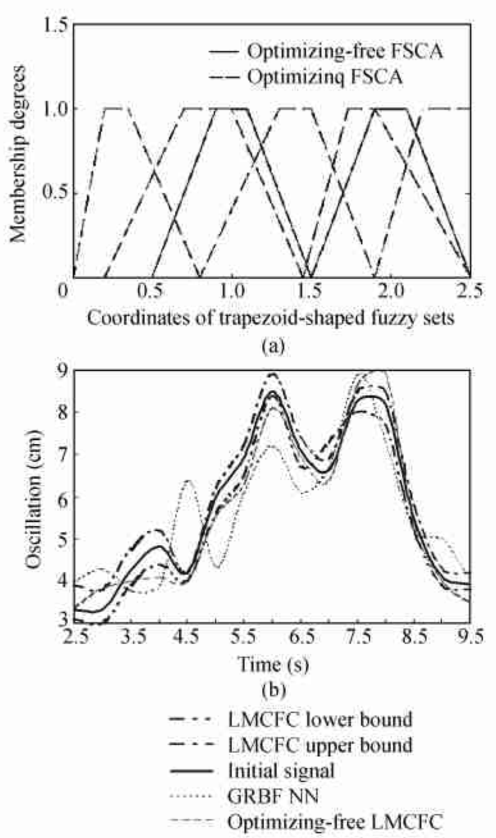


Fig. 6. Outcome of LMCFC with and without optimization procedure and GRBF. (a) Adjustment of FSCA positions, (b) output of LMCFC.

3.2.3 Results The experiment shows that when there exist few samples all three models result in large errors, but in meaning of average LMCFC they have the smallest errors (see Fig. 6(b)). In most circumstances, the real signal falls within the range of the lower and upper bounds of output granules in LMCFC. As choosing more clusters, the individual error of three models decreases to its least to a great extent, yet the run times have large differences. The optimizing-free LMCFC has the least time cost (12s), whereas the optimizing LMCFC and GRBF NN are approximately equal (21 s vs 28 s). However, using only 18 neurons the optimizing LMCFC satisfies the error criterion while the optimizing-free LMCFC needs 64 at least. More importantly, LMCFC can automatically arrange their rule pitches toward the extreme points of the signal curve, and put more rule pitches in high frequency of signals and fewer in low frequency signals. It verifies that LMCFC has the function similar to the automatic Parzen window[16]. Without doubt, these are attributed to the optimization in LMCFC. Without optimization, these initial FSCA would be coarse ones. Also from Fig.6 (b) we may see that the signal filtered by FSCA is clearer than the filtered-free one. Thus LMCFC can effectively confine noises and outliers.

4 Conclusion

The proposed LMCFC is a user-driven model, and a user can design these FSCA independently. One does not need to worry about the problem of coarse initial parameter setting since LMCFC can optimize them to its best. Without optimization, these initial FSCA would be coarse ones. It is optimization that finds these natural laws hidden in given data. These laws might fail to be found by the user. LMCFC model can attain higher accuracy, and better transparency as well. Its design helps models get most information that can be granulized. The CFI in LMCFC has a great effect on the final result, and enhances the generalization of LMCFC. Nevertheless, the mechanism of HA is not clear, thus one could expect to discover more knowledge about it.

References

1 Gonzalez J., Rojas I., Pomares H. et al. A new clustering technique for function approximation. IEEE Trans. Neural Networks, 2002, 13(1): 132—142.

2 Botteldooren D. and Verkeyn A. Linguistic fuzzy modeling for industrial applications. J. Sci. & Industrial Research, 2001, 60(11): 851—859.

3 Leephakpreeda T. Resolving scaling problem of wide-range data in neural network training with linguistic model. Advances in Engineering Software, 2001, 32(8): 641—646.

4 Botteldooren D. and Verkeyn A. Fuzzy models for accumulation of reported community noise annoyance from combined sources. J. Acoustical Society of Amer. 2002, 112(4): 1496—1508.

5 Moreno-Garcia J., Linares L. J., Castro-Schez J.J. et al. A direct linguistic induction method for systems. Fuzzy Sets Syst., 2004, 146(1): 79—96.

6 Pedrycz W. Conditional fuzzy C-means. Patt. Reco. Lett., 1996, 17, 625—632.

7 Pedrycz W. Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Trans. Neural Networks, 1998, 9(4): 601—612.

8 Pedrycz W. and Vasilakos A. V. Linguistic models and linguistic modeling. IEEE Trans. SMC-B, 1999, 29(6): 745—757.

9 Wang Z. and Klir G. J. Fuzzy Measure Theory. New York: Plenum, 1992.

10 Grabisch M. A new algorithm for identifying fuzzy measure and its application to pattern recognition. Fuzzy Set. Syst., 1995, 69: 279—298.

11 Yu J., Cheng Q. and Huang H. Analysis of the weighting exponent in the FCM. IEEE Trans. SMC-B, 2003, 33(3): 1—5.

12 Kosko B. Optimal fuzzy rules cover extrema. Int. J. Intell. System, 1995, 10: 249—255.

13 Karyannis N. B. and Mi G. W. Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. IEEE Trans. Neural Networks, 1997, 8(6): 1492—1506.

14 Joo M., Wu S., Lu J. et al. Face recognition with radial basis function (RBF) neural networks. IEEE Trans. Neural Networks, 2002, 13(3): 697—710.

15 http://www.ics.uci.edu/~mlearn/MLRepository.html.

16 Joachims T. Estimating the generalization performance of an SVM efficiently, University Dortmund, Germany, 1999. Available in URL: http://www.kernel-machine.org/.

17 Bezdek J. C. and Hathway R. J. Numerical comparison of the RFCM and AP algorithms for clustering relational data. Pattern Recognition, 1991, 24(8): 783—791.

18 Yue S. and Li P. Choquet fuzzy integral: comparison and performance I, Information Fusion (To appear).

Appendix 1 Parameter estimation for LMCFC

1. Start: Let all weight value be arithmetic average of equivoques; input samples; acceptable error.
2. partition the output space;
3. construct the FSCA based on the user information;
4. perform CFC to form the partition of input space;
5. check the acceptable error. If satisfied, the algorithm stops and output the result; or go to the next step;
6. call the gradient descend procedure;
7. if $E>0$, repeat the above operation from $u(1)$, $u(2)$, $\cdots$, $u(n-1)$ each,
   if $E<0$, repeat the above operation from $u(n-1)$, $u(n-2)$, $\cdots$, $u(1)$ each,
   continue the above operation for all data;
8. check the monotone demand. If not, repeat the above step;
9. for all unmodified $u(i)$, perform the following operation:
   If $\overline{m}(i)+\widetilde{m}(i)-2u(i)>0$,
   $u^{\text{new}}=u^{\text{old}}(i)-\beta(\overline{m}(i)+\widetilde{m}(i)-2u(i))\overline{d}_{\min}(i)(2(\overline{m}(i)-\widetilde{m}(i)))^{-1}$,
   If $\overline{m}(i)+\widetilde{m}(i)-2u(i)<0$,
   $u^{\text{new}}=u^{\text{old}}(i)-\beta(\overline{m}(i)+\widetilde{m}(i)-2u(i))\widetilde{d}_{\min}(i)(2(\overline{m}(i)-\widetilde{m}(i)))^{-1}$,
   where $\overline{m}(i)=(n-i)^{-1}\sum$ (parameter with index less than i in last layer),
   $\widetilde{m}(i)=(n-i)^{-1}\sum$ (parameter with index larger index i in next layer);
10. check the acceptable error. If satisfied, stop;
    if not call gradient descend procedure to adjust parameter (the number of clusters and FSCAs);
11. check the acceptable error. If satisfied, stop;
    if not, go to step 2.

Gradient descend procedure

1. For any input sample, compute its error E; denote the corresponding parameter of the sample as: $u(0)$, $u(1)$, $\cdots$, $u(n)$, where $u(0)=0$, and $u(n)=1$.
2. Update $u(i)$ as follows:

$$u^{\text{new}}=u^{\text{old}}(i)-\alpha E(p_{n-i}-p_{n-i-1})/E_{\max}$$

α is a constant for controlling the convergence rate, $E_{\max}$, maximal error. $x_{(i)}$, i-th components, and P_i refers to the parameters in LMCFC, i.e. input, number of clusters and FSCA.