Advances in Engineering Research, volume 113
International Forum on Mechanical, Control and Automation (IFMCA 2016)

An accurate comparison of type-reduction algorithms for interval type-2 fuzzy sets using simulated data

XING Haihua[1,a], LIN Hongyan[1,b], SONG Chunhui[1,c]

[1]Hainan Normal University, College of Information Science and Technology,Haikou, China

[a] hhxing@hainnu.edu.cn ; [b] Corresponding author: lhyhaoyunqi@126.com;[c] song@hainnu.edu.cn

Keywords: Interval type-2 fuzzy sets(IT2FSs), Type-reduction; Karnik–Mendel (KM) algorithms, Enhanced Karnik–Mendel (EKM) algorithms, Iterative Algorithm with Stop Condition (IASC), Enhanced Iterative Algorithm with Stop Condition(EIASC)

Abstract: In order to find the best type-reduction algorithm of interval type-2 fuzzy sets with different characteristics, this paper makes a comparative analysis of KM, EKM, IASC and EIASC. Experiments are carried out on three type-2 fuzzy sets to compare the four algorithms.The results show that the four algorithms can accurately find the switching points, and the EIASC algorithm is the most efficient. This study provides an accurate and reliable comparative analysis for evaluating the applicability of the algorithm on different data characteristics.

Introduction

Type-reduction is used to calculate the centroid of type-2 fuzzy set, which is the main operation of the type-2 fuzzy inference[1]. Type-2 fuzzy rules of type-2 fuzzy systems are represented by type-2 fuzzy sets, and the output of type-2 fuzzy systems is a type-2 fuzzy set. Therefore, it is necessary to reduce it to type-1 fuzzy set, and then the fuzzy set is transformed to the exact value by defuzzifing[2]. At present, the classical method of type-reduction are as follows: the Karnik-Mendel (KM) algorithm is the most used method in Type-2 fuzzy logic applications[3]; based on the KM algorithm, Wu et al. proposed EKM algorithm with higher efficiency[4]; Melgarejo based on the EKM algorithm, proposed IASC(Iterative Algorithm with Stop Condition) algorithm for a new stop condition[5]; Wu made two improvements to IASC, and proposed EIASC(Enhanced IASC) algorithm[6]. Type reduction is much more time-consuming than defuzzification. Therefore, making type reduction more efficient can do good to the growing interest in using type-2 fuzzy systems [7-10].

In this paper, KM, EKM, IASC and EIASC are compared and analyzed, trying to find the type-reduction algorithm suitable for different data characteristics of the interval type-2 fuzzy set. This paper using four kinds of type-reduction algorithm for different types of type-2 fuzzy sets for experiments, aiming to provide more accurate and reliable evaluation of type-reduction algorithm and to provide effective reference for type-reduction method improvement.

Interval type-2 fuzzy sets (IT2-FS)

Definition 1. A T2 FS, denoted $\tilde{A}$, is characterized by a (3D) T2 MF $m_{\tilde{A}}(x,u)$, where $x \in X$

and $u \in J_x \subseteq [0,1]$

$$\tilde{A} = \{((x,u), m_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \quad (1)$$

In which $0 \le m_{\tilde{A}}(x,u) \le 1$, $\tilde{A}$ is also expressed as

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} m_{\tilde{A}}(x,u)/(x,u) \qquad J_x \subseteq [0,1] \quad (2)$$

Where $\iint$ denotes union over all admissible x and u. For discrete universes of discourse $\int$ is replaced by $\sum$. J_x is called the primary membership of x, and $m_{\tilde{A}}(x,u)$ is called the secondary grade, and for an interval T2 FS all $m_{\tilde{A}}(x,u) = 1$. An interval T2 FS is denoted as

$$\tilde{A} = \{((x,u),1) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} = \int_{x \in X} \int_{u \in J_x} 1/(x,u) \quad (3)$$

KM and EKM algorithm

Computing the generalized centroid of an IT2-FS is an important step in the operation of an Interval Type-2 Fuzzy Logic System (IT2-FLS). Although several algorithms have been proposed to compute it, the Karnik-Mendel (KM) algorithm is the most used method in Type-2 fuzzy logic applications. The centroid of IT2-FS can be expressed as an interval number$[c_l, c_r]$, and can be computed from the lower and upper membership functions of the FOU($\tilde{A}$) in the following way:

$$c_l = \frac{\sum_{i=1}^{L} x_i \overline{f}(x_i) + \sum_{i=L+1}^{N} x_i \underline{f}(x_i)}{\sum_{i=1}^{L} \overline{f}(x_i) + \sum_{i=L+1}^{N} \underline{f}(x_i)} \quad (4)$$

$$c_r = \frac{\sum_{i=1}^{R} x_i \underline{f}(x_i) + \sum_{i=R+1}^{N} x_i \overline{f}(x_i)}{\sum_{i=1}^{R} \underline{f}(x_i) + \sum_{i=R+1}^{N} \overline{f}(x_i)} \quad (5)$$

Where N is the number of sampling points, $\underline{f}(x_i)$ and $\overline{f}(x_i)$ is the membership value of UMF and LMF, L and R is the left and right switching points.

A. KM for Computing c_l

Step 1: Sort $x_i (i = 1,..., N)$ in increasing order, and match the membership $\underline{f}(x_i)$ and $\overline{f}(x_i)$ with their respective x_i, and renumber them so that index corresponds to the numbered x_i.

Step 2:Initialize $f(x_i)$ by setting $f(x_i)=\frac{\underline{f}(x_i)+\overline{f}(x_i)}{2}$ $(i=1,...,N)$

and then compute $y=\frac{\sum_{i=1}^{N}x_i f(x_i)}{\sum_{i=1}^{N}f(x_i)}$

Step 3:Find switch point L($1\le L\le N-1$) such that $x_L\le y\le x_{L+1}$

Step 4:Set $f(x_i)=\begin{cases}\overline{f}(x_i), i\le L\\ \underline{f}(x_i), i>L\end{cases}$

and compute $y'=\frac{\sum_{i=1}^{L}x_i\overline{f}(x_i)+\sum_{i=L+1}^{N}x_i\underline{f}(x_i)}{\sum_{i=1}^{L}\overline{f}(x_i)+\sum_{i=L+1}^{N}\underline{f}(x_i)}$

Step 5:If $y=y'$,stop and set $c_l=y'$,the switch point is L; otherwise, set $y=y'$ and got Step 3.

B. KM for Computing c_r

Step 1:The same as Step 1 of KM for computing c_l.

Step 2:The same as Step 2 of KM for computing c_l.

Step 3:Find switch point R($1\le R\le N-1$) such that $x_R\le y\le x_{R+1}$

Step 4:Set $f(x_i)=\begin{cases}\underline{f}(x_i), i\le R\\ \overline{f}(x_i), i>R\end{cases}$

and compute $y'=\frac{\sum_{i=1}^{R}x_i\underline{f}(x_i)+\sum_{i=R+1}^{N}x_i\overline{f}(x_i)}{\sum_{i=1}^{R}\underline{f}(x_i)+\sum_{i=R+1}^{N}\overline{f}(x_i)}$

Step 5:If $y=y'$,stop and set $c_r=y'$,the switch point is R; otherwise, set $y=y'$ and got Step 3.

C. EKM for Computing c_l

Step 1: The same as Step 1 of KM for computing c_l.

Step 2: Set k=[N/2.4](the nearest integer to N/2.4)

Step 3:Compute $a=\sum_{i=1}^{k}x_i\overline{f}(x_i)+\sum_{i=k+1}^{N}x_i\underline{f}(x_i)$, $b=\sum_{i=1}^{k}\overline{f}(x_i)+\sum_{i=k+1}^{N}\underline{f}(x_i)$, $y=a/b$

Step 4:Find k' ($1\le k\le N-1$) such that $x_{k'}\le y\le x_{k'+1}$.

Step 5:If $k'=k$,stop and return $c_l=y$,L=k; otherwise,continue

Step 6:Compute $s = sign(k' - k)$

$$a' = a + s\sum_{i=\min(k',k)+1}^{\max(k',k)} x_i(\overline{f}(x_i) - \underline{f}(x_i)) \quad , \quad b' = b + s\sum_{i=\min(k',k)+1}^{\max(k',k)} (\overline{f}(x_i) - \underline{f}(x_i)), \quad y' = a'/b'$$

Step 7:Set $y = y', a = a', b = b'$ and $k = k'$. Go to Step 4.

D. EKM for Computing c_r

Step 1: The same as Step 1 of KM for computing c_r.

Step 2: Set k=[N/1.7](the nearest integer to N/1.7)

Step 3:Compute $a = \sum_{i=1}^{k} x_i \overline{f}(x_i) + \sum_{i=k+1}^{N} x_i \underline{f}(x_i)$, $b = \sum_{i=1}^{k} \overline{f}(x_i) + \sum_{i=k+1}^{N} \underline{f}(x_i)$, $y = a/b$

Step 4:Find k' $(1 \le k \le N-1)$ such that $x_{k'} \le y \le x_{k'+1}$.

Step 5:If $k' = k$,stop and return $c_r = y$,R=k; otherwise,continue

Step 6:Compute $s = sign(k' - k)$

$$a' = a - s\sum_{i=\min(k',k)+1}^{\max(k',k)} x_i(\overline{f}(x_i) - \underline{f}(x_i)), \quad b' = b - s\sum_{i=\min(k',k)+1}^{\max(k',k)} (\overline{f}(x_i) - \underline{f}(x_i)), \quad y' = a'/b'$$

Step 7:Set $y = y', a = a', b = b'$ and $k = k'$. Go to Step 4.

IASC and EIASC algorithm

A. IASC for Computing c_l

Step 1: The same as Step 1 of KM for computing c_l.

Step 2:Initialize $a = \sum_{i=1}^{N} x_i \underline{f}(x_i)$, $b = \sum_{i=1}^{N} \underline{f}(x_i)$, $c_l = x_N$, L=0

Step 3:Compute L=L+1 , $a = a + x_L(\overline{f}(x_L) - \underline{f}(x_L))$, $b = b + (\overline{f}(x_L) - \underline{f}(x_L)), c = a/b$

Step 4:If $c > c_l$,stop and return L=L-1; otherwise, set $c_l = c$,go to Step 3.

B. IASC for Computing c_r

Step 1:The same as Step 1 of KM for computing c_l.

Step 2:Initialize

$a = \sum_{i=1}^{N} x_i \overline{f}(x_i)$, $b = \sum_{i=1}^{N} \overline{f}(x_i)$, $c_r = x_1$, R=0

Step 3:Compute R=R+1

$a = a + x_R(\overline{f}(x_R) - \underline{f}(x_R))$, $b = b + (\overline{f}(x_R) - \underline{f}(x_R))$, $c = a / b$

Step 4:If $c < c_r$,stop and return R=R-1; otherwise, set $c_r = c$,go to Step 3.

C. EIASC for Computing c_l

Step 1,Step 2 and Step 3 : The same as IASC for computing c_l.

Step 4:If $c_l \le x_{L+1}$,stop and return L=L-1; otherwise, set $c_l = c$,go to Step 3.

D. EIASC for Computing c_r

Step 1: The same as Step 1 of IASC for computing c_r.

Step 2:Initialize

$a = \sum_{i=1}^{N} x_i \overline{f}(x_i)$, $b = \sum_{i=1}^{N} \overline{f}(x_i)$, $c_r = x_1$, R=N

Step 3:Compute

$a = a + x_R(\overline{f}(x_R) - \underline{f}(x_R))$, $b = b + (\overline{f}(x_R) - \underline{f}(x_R))$, $c_r = a / b$, R=R-1

Step 4:If $c_r > x_R$,stop and return R; otherwise, go to Step 3.

Experimental comparison

In this section we compare the performance of the four type-reduction algorithms (KM, EKM, IASC, and EIASC) using three simulations. The cycle times of various algorithms are compared in order to evaluate the efficiency of each algorithm, and then compare the centroid $[c_l, c_r]$ of T2 FS and the switch points $[x_l, x_r]$ obtained by the different algorithms in order to evaluate convergence of each algorithm.

A. Experiment 1

Principal membership function of type-2 fuzzy set is a symmetric Gauss function, and the mean fixed, the variable of variance. It is expressed as follows:

$m(x) = \exp[-0.5((x-5)/s^2)]$

Where $m \in [0.25,1.75]$, $x \in [0,10]$. We shown the membership function in Fig.1. The cycle times of each algorithm in Table 1, and the centroid and switch points in Table 2.

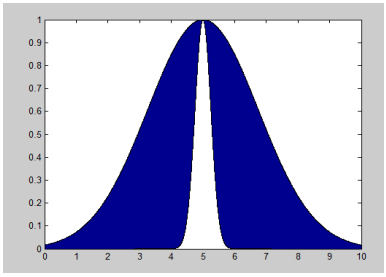

Fig.1 Principal membership function

Table 1 cycle times of each algorithm

N	100	500	1000	5000
KM	1010	4008	8008	40008
EKM	606	3006	6006	35007
IASC	103	503	1003	5003
EIASC	72	360	720	3596

B. Experiment 2

Principal membership function of type-2 fuzzy set is a Gauss function, and the variance fixed, the variable of mean. It is expressed as follows:

$$m(x) = \exp[-0.5((x-m)^2)]$$

Where $m \in [0.25,1.75]$, $x \in [0,10]$. We shown the membership function in Fig.2.The cycle times of each algorithm in Table 3, and the centroid and switch points in Table 4.

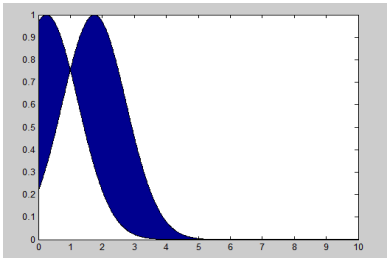

Fig.2 Principal membership function

Table 2 The centroid and switch points of each algorithm

Algorithms	N	100	500	1000	5000
KM IASC EIASC	cl	3.5939	3.5951	3.5953	3.5955
	L	36	180	360	1798
	xl	3.500	3.580	3.5900	3.5940
	cr	6.4061	6.4049	6.4047	6.4045
	R	65	321	641	3203
	xr	6.400	6.400	6.400	6.400
EKM	cl	3.5983	3.5949	3.5951	3.5954
	cr	6.4524	6.4119	6.4081	6.4052

Table 3 cycle times of each algorithm

N	100	500	1000	5000
KM	606	3006	7007	30006
EKM	606	4008	8008	45009
IASC	30	142	281	1396
EIASC	91	451	902	4503

Table 4 The centroid and switch points of each algorithm

Algorithm	N	100	500	1000	5000
KM IASC EIASC	cl	0.8672	0.8899	0.8928	0.8951
	L	9	45	90	448
	xl	0.800	0.8800	0.8900	0.8940
	cr	1.880	1.8882	1.8892	1.8900
	R	19	95	189	945
	xr	1.800	1.8800	1.8800	1.8880
EKM	cl	0.8461	0.8849	0.8902	0.8946
	cr	1.880	1.8886	1.8894	1.8901

In Experiment 1 and Experiment 2, the centroid $[c_l, c_r]$ and the switch points $[x_l, x_r]$calculated by KM,IASC and EIASC are the same, but the centroid calculated by the EKM are slightly different from the other methods. The reason is related to the iterative formula of the EKM algorithm. In

Experiment 1, the main membership is symmetric Gauss's function. The cycle times of EIASC is the least, and the efficiency of EIASC is the highest, followed by IASC. The efficiency of EKM is significantly higher than KM. In Experiment 2, the main membership is non symmetric Gauss's function. The cycle times of IASC is the least, followed by EIASC.The efficiency of KM is significantly higher than EKM. The reason is that the IASC for computing cr start from switch point 1 and increase it gradually to find the correct switch points,but EIASC computing cr start from switch point N and decrease. When the right switch points in the left half of N, the efficiency of the EIASC is low.

Conclusions

Type-reduction algorithms are very important for type-2 fuzzy sets and systems. In this paper we have reviewed several more efficient type-reduction algorithms, and also made an accurate comparison and practical implementation of them for interval type-2 fuzzy sets using simulated data. Experimental results showed that the centroid calculated by EKM is not as accurate as KM, IASC and EIASC. The reason is related to the iterative formula of the EKM algorithm. When the left and right switching points are in the middle position, the EIASC and EKM algorithms are more efficient than the IASC and KM algorithms. Otherwise, the efficiency of EKM and EIASC are not as good as the KM and IASC algorithms. This study will be very helpful in improving type-reduction algorithms and promoting the popularity of type-2 fuzzy sets and systems.

Acknowledgement

This research was financially supported by the Hainan Natural Science Foundation (Grant No.20156227 and Grant No.20156242). Corresponding author: LIN Hongyan.

References

[1] Qilian L, Jerry M M. Interval Type-2 Fuzzy Logic Systems: Theory and Design [J]. IEEE Transactions on Fuzzy Systems (S1063-6076) 2000, 8(5): 535-550.

[2] J. Mendel, Uncertain rule-based fuzzy logic systems, Prentice Hall PTR, NJ, 2001.

[3] Karnik N N, Mendel J M. Centroid of a type-2 fuzzy set[J]. Information Sciences, 2001, 131(1):195-220.

[4] Dongrui Wu, Jerry M M. Enhanced Karnik - Mendel algorithms for interval type-2 fuzzy sets and systems[C]. Proc of North American Fuzzy Information Processing Society. San Diego, 2007: 184-189.

[5] Melgarejo M. A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set[C]. Proc of Annual Conference of the North American Fuzzy Information Processing Society. 2007:190-194.

[6] Wu D, Nie M. Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems[C]. Proc of IEEE International Conference on Fuzzy Systems. 2011:2131-2138.

[7] O. Castillo and P. Melin. A new approach for plant monitoring using type-2 fuzzy logic and fractal theory[J]. Int. J. General Syst., 2004,vol. 33, no. 2/3, 305-319.

[8] P. Melin and O. Castillo. An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory[J]. Inform. Sci., 2007,vol. 177, no. 7, 1543-1557.

[9] N. R. Cazarez-Castro, L. T. Aguilar, and O. Castillo. Hybrid genetic-fuzzy optimization of a type-2 fuzzy logic controller. Proc. 8th Int. Conf. Hybrid Intell. Syst., Sep. 2009, pp. 718-725.

[10] N.R.Cazares-Castro, L.T.Aguilar, O. Castillo. Designing type-2 fuzzy logic systemcontrollers via fuzzy Lyapunov synthesis for the output regulator of a servomechanism with nonlinear backlash. in Proc. IEEE Int. Conf. Syst., Man Cybern., Oct. 2009, pp. 268-273.