Proceedings of KFIS Autumn Conference 2005 Volume 15, Number 2

퍼지 시스템을 이용한 함수표현의 장점; A REVIEW

Approximation of the smooth functions by using fuzzy systems; A review of the advantages

문병수, 이장수, 이동영, 권기춘
한국원자력연구소

B. S. Moon, J. S. Lee, D. Y. Lee, K. C. Kwon
Dept. of Instrumentation & Control - Human Factors, Korea Atomic Energy Research Institute
E-mail : bsmoon@kaeri.re.kr

Abstract

A review of how the functions of two or more independent variables can be approximated by using fuzzy systems is provided in this paper. We start with an exact represention of a linear interpolation function of two independent variables by using a fuzzy system. Next, we describe how this function can be approximated by another fuzzy system with a lesser number or with a desired number of output fuzzy sets. Thus, a reduction of the storage needed is achieved by storing the fuzzy rules or equivalently the output fuzzy set numbers instead of storing the whole discrete function values. A description on how the cubic spline interpolation function can be represented exactly by using the fuzzy system method is provided, along with a few examples where fuzzy rule tables with a size of 7×7 provide a representation of the functions with relative errors of the order of 10^{-2} or less.

1. Introduction

It has been proven a long time ago that the fuzzy systems are universal approximators. Castro and Delgado [1] showed that for a *continuous function* f(x) *on a compact set and* for a given $\varepsilon>0$, there exists a fuzzy system that approximates f(x) within ε. B. Kosko[2] proposed a fuzzy system with two levels, one of which is being used to approximate and tune the fuzzy rules. Wang et al. [4] provided a constructive method for building a fuzzy system to approximate a function within a prescribed accuracy.

As these papers use linear approximations, they require a very large number of fuzzy rules and a large number of fuzzy sets. By *using the cubic splines,* we proved earlier [5] that a continuously differentiable function can be represented very accurately by using a small number of fuzzy rules In fact, we showed that the cubic spline interpolation functions for the functions of two variables can be exactly represented by fuzzy systems. In the following, we first describe how to generate a fuzzy system for an exact representation of the linear interpolation functions of two independent variables.

We then proceed to show how the number of output fuzzy sets can be reduced without losing the accuracy. Using the fact that the linear interpolation has the approximation accuracy of O(h) while the accuracy is O(h^2) for the cubic spline interpolation functions, we repeat the same process to transform the cubic spline interpolation function to a fuzzy system form.

2. A fuzzy system to represent linear interpolation functions

Let $z=f(x,y)$ be an arbitrary function of two independent variables defined on the interval $[-1,1]\times[-1,1]$ and let $x_i=-1+ih$, $y_j=-1+jh$, where $h=\frac{1}{N}$ for a positive even integer N. Let $g(x,y)$ be the linear interpolation of $f(x,y)$ at the grid points (x_i,y_j), $i,j=0,1,\ldots,N$. Then $g(x,y)$ is defined by a linear interpolation of the function values at the four nearest grid points $f_{k,l}$, $f_{k,l+1}$, $f_{k+1,l}$, $f_{k+1,l+1}$, where x_k is the nearest grid point such that $x_k \le x$ and y_l is the nearest with $y_l \le y$.

Let $T_i(x)$ be the triangular function or equivalently the spike function defined on the support interval $[x_{i-1},x_{i+1}]$ for $i=0,1,\ldots,N$ and similarly for $T_j(y)$'s be the triangular functions defined on the grids $y_0, y_1, \cdots, y_N$, then we can write g(x,y)= $\sum_{i,j=0}^{N} f_{ij}T_i(x)T_j(y)$.

(1) Fuzzification of the input variables x,y

We take $T_i(x), i=0,1,2,\ldots N$ be the fuzzy sets for the input variable x and $T_j(y), j=0,1,2,\ldots N$ be the fuzzy sets for the input variable y. Given an arbitrary point (x_0,y_0) in the interval $[-1,1]\times[-1,1]$, we fuzzify x to find the fuzzy sets $T_k(x)$, $T_{k+1}(x)$ with the nonzero membership values λ, $1-\lambda$ respectively. Similarly, we fuzzify y to find the fuzzy sets $T_l(y)$, $T_{l+1}(y)$ with the nonzero membership values μ, $1-\mu$ respectively.

Thus, we have four fuzzy sets $T_k(x)\times T_l(y)$,$T_k(x)\times T_{l+1}(y)$, $T_{k+1}(x)\times T_l(y)$, $T_{k+1}(x)\times T_{l+1}(y)$ with membership values $\lambda\mu$, $\lambda(1-\mu)$, $(1-\lambda)\mu$, $(1-\lambda)(1-\mu)$.

(2) Generation of the fuzzy rule table

First, we form a $(N+1)\times(N+1)$ matrix whose (i,j) entry is $f_{i,j}$. Next, we sort the array of $(N+1)^2$ entries into an increasing order and delete the duplicate ones. Let the ordered list be $\{t_k | k=1,2,\ldots M\}$ and assign ordinal numbers to $f_{i,j}$ as '1' to those corresponding to t_1, '2' to those corresponding to t_2, and so forth. Note that the largest will get an ordinal number which will be less than or equal to $(N+1)^2$. We now form a rule table by using the ordinal number corresponding to $f_{k,l}$ as the entry of (k,l), we obtain a fuzzy rule table.

(3) Output fuzzy sets

Using the array $\{t_k | k=1,2,\ldots M\}$, we form a set of triangular fuzzy sets $\{T_k | k=1,2,\ldots M\}$, i.e. T_k is supported on the interval $[t_{k-1},t_{k+1}]$ and is centered at t_k.

(4) Defuzzification

For the defuzzification of the output, we use the center area defuzzification method. When we have four fuzzy sets T_{k_i,l_i}'s with nonzero weights ν_{k_i,l_i}, for i=1,2,3,4 as the output from the fuzzy inferences, we compute $\sum_{i=1}^{4}\nu_{k_i,l_i}t_{k_i,l_i}$ as the value of the fuzzy system, where t_{k_i,l_i} is the center of the support for the triangular fuzzy set T_{k_i,l_i}.

It is fairly easy to see that the above fuzzy system is an exact representation of the linear interpolation of the function. A simple routine proof is omitted.

To reduce the number of output fuzzy sets and to adjust the support intervals for the output fuzzy sets, we define $t_m = \min\{f_{i,j} | i,j = 1,2,...N\}$. Let $t_0 = t_m - h$, $t_1 = t_0 + h$, ..., $t_k = t_{k-1} + h$. We sort $f_{i,j}$ in an increasing order, assign $g_{k,l} = t_j$ to all $f_{k,l} \in (t_{j-1}, t_{j+1}]$ for $j = 1,2,...$ so that a discrete function $g_{k,l}$ is defined on the interval $[-1,1] \times [-1,1]$ as an approximation of $f(x,y)$

Now, we revise the fuzzy rules and the output fuzzy sets properly to make the necessary changes. Then we have the following.

Theorem 1. Let $F(x,y)$ be the function defined by the fuzzy system described above, then we have $|F(x,y) - G(x,y)| \le h$ for all $(x,y) \in [-1,1] \times [-1,1]$ where $G(x,y)$ is the function defined by the earlier fuzzy system, and hence $F(x,y) - f(x,y) = O(h)$.

Finally, consider the case where we want to fix the number of output fuzzy sets, say as L. Divide the interval $[t_m, t_M]$ into L subintervals of equal length. Let $t_j, j = 0,1,2,...L$ be the nodal points and let $I_k = [t_{k-1}, t_{k+1}]$, using $t_{-1} = t_0 - h$, $t_{L+1} = t_L + h$ with $h = \frac{t_M - t_m}{L}$. Now, if we define the output fuzzy sets to be triangular sets with I_k as the support intervals and t_k be the center of the supports.

3. A fuzzy system to represent the cubic spline interpolation function

Let x_i, y_j, $i,j = -1,0,1,2,...N,N+1$ be the nodal points in each axis and define the cubic B-splines $B_i(t)$'s as $\frac{1}{6h^3} \times$

$$\begin{cases} (t - t_{i-2})^3 \\ h^3 + 3h^2(t - t_{i-1}) + 3h(t - t_{i-1})^2 - 3(t - t_{i-1})^3 \\ h^3 + 3h^2(t_{i+1} - t) + 3h(t_{i+1} - t)^2 - 3(t_{i+1} - t)^3 \\ (t_{i+2} - t)^3 \\ 0 \end{cases} \qquad (1)$$

on each of the intervals $[t_{i-2}, t_{i-1}]$, $[t_{i-1}, t_i]$, $[t_i, t_{i+1}]$, and $[t_{i+1}, t_{i+2}]$ respectively. Then the function $f(x,y)$ can be approximated by a cubic spline function

$$g(x,y) = \sum_{i,j}^{N} f_{i,j} B_i(x) B_j(y) \qquad \text{---} (2)$$

where $f_{i,j} = f(x_i, y_j)$.

We use $B_i(x) \times B_j(y)$ as input fuzzy sets instead of $T_i(x) \times T_j(y)$ for the fuzzification of (x,y), then we have the following.

Theorem 2. Let $G(x,y)$ be the function defined by the fuzzy system described above, then we have $G(x,y) - f(x,y) = O(h^2)$

4. Examples

In this section, we describe two examples [5] of the fuzzy system designed to represent the cubic spline interpolation function for the polynomials in x and y. We compare the evaluation results with those of a Lagrangian interpolation. Double precision calculations are used for all the cases so that the calculation error is relatively minor when compared to the intrinsic system error. For all of the cases in the following examples, we evaluated the fuzzy systems at 10,000 points and computed the average of the absolute errors.

Table 1. Fuzzy rules to compute $f(x,y) = x^3 - x^2 y$

	1	2	3	4	5	6	7
1	23	9	6	4	3	2	1
2	26	22	12	8	7	5	3
3	23	22	20	18	16	14	13
4	15	16	17	18	19	20	21
5	23	22	20	18	16	14	13
6	31	29	28	26	24	14	11
7	36	34	33	32	30	27	13

Example 1. A Fuzzy system for $f(x,y) = x^3 - x^2 y$

The fuzzy rules obtained for this function on $[-1,1]$ with n=4 divisions are shown in Table 1. Note that we need 7 fuzzy sets for the input fuzzificatin in this case. A summary of the evaluation results by the fuzzy system at 10,000 points, along with those by a fuzzy system using triangular input fuzzy sets for n=4,10,20 are shown in Table 2.

Table 2. Comparison of the Evaluation Errors (Average; Spline vs Spike)

n	*Spline*	*Spike*	*No. of Rules*
4	0.15×10^{-7}	0.2034	49(25)
10	0.18×10^{-7}	0.0370	169(121)
20	0.19×10^{-7}	0.0093	529(441)

Example 2. A Fuzzy system for $f(x,y)=x^4+y^4$

The fuzzy rules obtained for this function on $[-1,1]$ with n=4 are shown in Table 3 and the centers of the support for the output fuzzy sets are at -0.083333, -0.020833, 0.041666, 0.479166, 0.541666, 1.041666, 3.91666, 3.979166, 4.479166, 7.916666. Table 4 shows a summary of the evaluation errors at 10,000 points, including those by a fuzzy system using triangular fuzzy sets for n=4,10,20.

Table3. Fuzzy rules to compute $f(x,y)=xy$

	1	2	3	4	5	6	7
1	13	12	10	7	4	2	1
2	12	11	9	7	5	3	2
3	10	9	8	7	6	5	4
4	7	7	7	7	7	7	7
5	4	5	6	7	8	9	10
6	2	3	5	7	9	11	12
7	1	2	4	7	10	12	13

Table 4. Comparison of evaluation errors

n	*Spline*	*Spike*	*No. of Rules*
4	0.007812	0.434699	49(25)
10	0.000196	0.097400	169(121)
20	0.000012	0.026187	529(441)

5. Conclusion

We reviewed how to setup a fuzzy system to represent a linear interpolation function or a cubic spline interpolation function for an arbitrary smooth function, i.e. twice or three times continuously differentiable function. In our earlier papers, we showed how polynomials can be represented by a fuzzy system [5] and how a twice continuously differentiable function can be represented [6] by a fuzzy system with an emphasis on the gray scale image presentation of the fuzzy rules.

We showed in this paper that the fuzzy system representation not only provides an efficient tool for evaluating the cubic spline interpolation function but also it provides a mechanism to approximate the dependent variable so that the number of output fuzzy sets can be reduced to a great deal in some cases without loosing the accuracy.

6. References

[1] J. L. Castro and M. Delagado, 'Fuzzy systems with defuzzification are universal approximators', IEEE Trans. Systems, Man, Cybernet. Vol.26, No.1 (1996) 149-152.

[2] B. Kosko, 'Fuzzy systems are universal approximators', IEEE Trans. Comput. Vol.43, No. 11 (1994) 1329-1333.

[3] A. Shmilovici and O. Maimon, 'The fuzzy rule-base solution of differential equations', Information Sciences, Vol. 92 (1996) 233-254.

[4] L. X. Wang, 'Fuzzy systems are universal approximators', Proc. IEEE Int. Conf. on Fuzzy Systems, San Diego (1992) 1163-1170.

[5] Byung Soo Moon, 'A practical algorithm for representing polynomials of two variables by fuzzy systems with accuracy $O(h^4)$', Fuzzy Sets and Systems Vol. 119 (2001) 321-327.

[6] Byung Soo Moon, et al., 'A fuzzy system representation of functions of two variables and its application to gray scale images', J. Korea Fuzzy Logic and Intelligent Systems Vol. 11, No. 9 (2001) 569-573.