Fuzzy 개념을 갖는 IF-THEN-ELSE 규칙의 휴리스틱 컴파일 기법

김 두현, 김 진형
한국과학기술원 전산학과

임 영환
한국전자통신연구소

요 약

본 논문에서는 fuzzy 개념을 갖는 IF-THEN-ELSE 규칙을 이용한 추론에 대해서 논한다. 기존의 방식들이 fuzzy 개념을 정의하기위해 array를 사용한 반면, 본 논문에서는 기본적으로 standard function을 이용한다. 그리고 IF-THEN-ELSE 규칙의 컴파일을 위해 기존의 방식이 matrix를 사용한 반면, 본 논문에서는 휴리스틱으로써 parameter mapping function 습득의 개념을 제시한다. 또한 습득된 function을 이용한 추론이 기존의 matrix 이용방식보다 효율적이고, 필요한 PROPERTY도 만족시킴을 보인다.

1. 개요

우리는 일상 생활 중에서 다음과 같은 형식의 추론을 많이 한다.

규칙 : 키가 크면 몸무게가 많이 나간다.
정보 : 그 사람은 키가 매우 크다.

결론 : 그 사람은 몸무게가 매우 많이 나간다. (예 1-1)

이런 추론의 특징은 "크다", "매우 크다", "많이 나간다", "매우 많이 나간다" 등의 애매모호한 개념(Fuzzy concept)을 포함하고 있어서 컴퓨터가 기존의 binary logic이나 multi-valued logic을 이용하여 처리하기 힘들다는 것이다.

1970년대에 Zadeh는 이런 애매모호한 개념을 정의하는데 fuzzy set을 이용하고, 이와 같은 추론을 위해 "composition rule of inference"를 제시했다[2]

본 논문에서는 다음과 같이 규칙이 좀더 복잡한 추론에 대해서 논한다.

키가 크면 몸무게가 많이 나가고,
키가 작으면 몸무게가 적게 나가며,
키가 170cm 정도이면 몸무게가 약 65kg 이고,
키가 180cm 정도이면 몸무게가 약 73kg 이다.

이를 형식화하면 다음과 같다.

규칙 : IF x is A0 THEN y is B0 ELSE
IF x is A1 THEN y is B1 ELSE
....
IF x is An THEN y is Bn. (식 1-1)
정보 : x is A' (식 1-2)

결론 : y is B' (식 1-3)

여기서 x, y는 임의 domain U, V 위에 존재하는 변수이고, A,B,A',B'는 U, V 위의 fuzzy 개념이다 즉, 서로 다른 domain에 존재하는 두 변수간의 관계를 좀더 자세히 설명하는 IF-THEN-ELSE(이후부터 ITE라 칭함) 규칙이 지식베이스에 존재하는 경우이다.

이런 추론에서 반드시 만족되어야 할 property는 들어온 정보가 규칙의 조건부 중 어느 하나와 정확히 일치하면 그 조건부와 상응하는 규칙의 결론이 출력되어야 한다는 것이다. 본 논문에서는 이를 existing condition property라 부른다.

정의 : existing condition property

IF $A' = A_i$ THEN $B' = B_i$, $i = 0,1,2,..,n$

일반적인 접근방식은 추론시의 효율성을 위해서 여러개의 규칙을 포함하고있는 ITE 규칙을 하나의 규칙으로 컴파일해놓고 이를 실제 추론에 이용하는 것이다. 이에 대한 연구결과는 Prade[3]와 Martin[4]이 있다. 이들은 fuzzy 개념을 수학적으로 정의하는 fuzzy set의 membership grading function(이후 MGF라 칭함)을 전산화하기위해 array를 사용했다. 예를 들어, U가 [100,260]로써 키의 domain이고 이에 속한 fuzzy set "크다"의 MGF m(u,크다)를 나타내기 위해 [100,260]중 100, 120, 140, 160, 180,200, 220, 240, 260에 대해서만 각 membership grading value를 (0, 0, 0, 0 3, 0.95, 1, 1, 1, 1)와 같이 array로 나타냈다. 또한 ITE 규칙 내의 하나의 규칙을 matrix로 나타내고 , ITE 규칙 전체의 컴파일을 위해서 이 matrix들의 각 항끼리 MAX, MIN 등의 연산을 하여 하나의 matrix를 만들었다.

그런데 문제는 MAX, MIN 등의 연산자들이 각 matrix가 갖고 있던 두 변수간의 관계에 대한 정보를 많이 잃어버리기 때문에 모두 existing condition property 를 만족시키지 못한다는 것이다. 또 실제 추론시 array로 들어오는 정보와 matrix로 컴파일되어 저장된 ITE 규칙을 composition하는 과정에서 Zadeh가 제시한 max-min 연산을 O(n**2)을 걸쳐 행해야 하기 때문에 추론속도면에서 비효율적이다.

MGF를 나타내는 또다른 방법은 본 논문에서 사용한 것처럼 standard function을 이용하는 것이다. 즉 standard function의 parameter에 알맞은 값을 줌으로써 임의 fuzzy set의 MGF를 표현하는 것이다. 본 논문에서는 이의 사용과 아울러 ITE 규칙의 컴파일을 위한 휴리스틱으로써 parameter mapping function 습득의 개념을 제시한다. 또한 습득된 function을 이용한 추론이 기존의 matrix 이용방식보다 효율적이고, existing condition property도 만족시킴을 보인다.

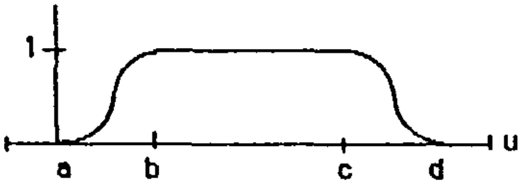
2. 기본적 접근 방법

가. standard function

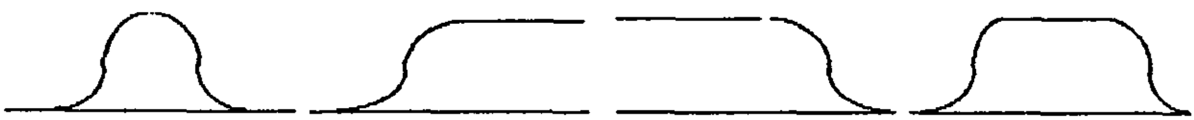
전술한 바, 본 논문에서는 ITE 규칙, 정보(입력), 출력(결론)상의 애매모호한 개념의 수학적으로 정의하기위해 (식 2-1)과 같은 standard function(S)을 이용한다. 여기서 u는 domain (혹, base set)의 원소이며, a, b, c, d는 parameter이다 각 parameter는 (식 2-1)의 정의와 (그림 2-2)에서 잘 나타나듯이, S=1([b,c]), S>0([a,d])인 구간을 지정하는 역할을 한다 S function은 (그림 2-3)에서와 같이 다양한 형태의 개념을 정의할 수 있으며, 이외에 자연

언어로 말할 수 없는 미묘한 지식상태도 표현할 수 있다. 일례로 (그림 2-4)처럼 비대칭형을 들 수 있다.

$$S(u;a,b,c,d) = \begin{cases} 0 & u \le a,\ u \le d \\ 2[(u-a)/(b-a)]^2 & a < u < (a+b)/2 \\ 1-2[(u-b)/(b-a)]^2 & (a+b)/2 < u < b \\ 1 & b \le u \le c,\ u=a=b,\ u=d=c \\ 1-2[(u-c)/(d-c)]^2 & c < u < (c+d)/2 \\ 2[(u-d)/(d-c)]^2 & (c+d)/2 < u < d \end{cases} \qquad \text{(식 2-1)}$$

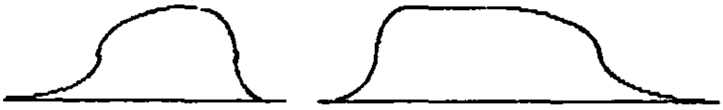


(그림 2-2)



"about 65kg" = S(u; 60, 65, 65, 200)　"heavy" = S(u; 65, 80, 200, 200)　"light" = S(u; 0, 0, 55, 60)　"average" = S(u; 50, 60, 70, 80)

(그림 2-3) domain = [0, 200] (몸무게)



S(u; 60, 75, 75, 80)　S(u; 60, 75, 80, 85)

(그림 2-4) domain = [0, 200] (몸무게)

또, A라는 임의의 개념에 "매우","약간","다소" 등의 조합으로 된 modifier가 붙어 "매우 매우 약간 A" 등의 새로운 개념을 만드는 경우가 있다. 이런 변형된 개념 mA는 A가 S(u;a,b,c,d)일 때 다음처럼 정의된다.

$$S(u;a,b,c,d)^k \qquad \text{(식 2-5)}$$

여기서 k는 modifier를 수치적으로 바꾼 것으로, 0<k 범위 내에서, k=1일 때는 mA=A이고, k가 0에 가까울 수록 mA가 A보다 자신이 없는 상태를, k가 1보다 크면 클 수록 더욱 확신에 찬 상태를 나타내게 된다.

Zadeh의 제안에 따르면 "매우(very)"일 때 k=2, "다소(more or less)" 일 때 k=0.5로 정의하며, 이들의 조합으로된 modifier의 k 값은 조합을 이룬 각 modifier들의 k값의 곱이 된다[2]. 이후부터는 임의 fuzzy concept의 정의를 위해 (식 2-5)를 사용한다.

나. 지식표현

본 논문이 추구하는 시스템은 지식표현을 위해 기호를 사용하지 않고 S function을 사용한다 따라서 ITE규칙은 아래의 (예 2-6)처럼 각 parameter별로 테이블에 저장된다 또 (식 1-2), (식 1-3)과 같은 ITE 규칙의 입출력도 각각 <xa, xb, xc, xd, xk>, <ya, yb, yc, yd, yk>로 나타낸다.

예 2-6:

키가 크면 몸무게가 많이나가고,
키가 약 170cm에서 180cm 정도이면 몸무게는 약 65kg에서 75kg 정도이다.

라는 ITE규칙을 S functiond으로 다음과 같이 입력할 수 있다

IF S(u;165,175,260,260) THEN S(v;60,70,200,200) ELSE
IF S(u,168,170,180,182) THEN S(v;63,65,75,77)

(단 U=[100,260], V=[20,200])
이를 테이블로 나타내면 아래와 같다

table a

x	y
165	70
186	63

table b

x	y
175	70
170	65

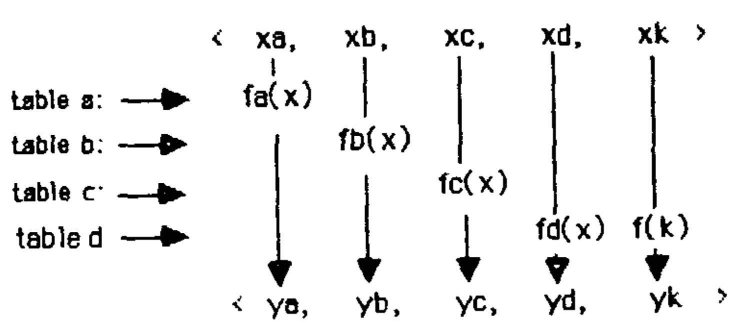
table c

x	y
260	200
180	75

table d

x	y
260	200
182	77

여기서 ITE규칙의 k에 대한 테이블을 고려치 않았다. 이것은 다시말해 ITE 규칙으로 부터는 k에 대해 특별히 얻을 만한 정보가 없다는 것을 뜻하기도 한다. 이에 대해서는 4장에서 설명한다.



(그림 2-7)

다. 추론방법

본 논문의 핵심 아이디어는 (예 2-6)과 같이 저장된 각 테이블로 부터 (그림 2-7)처럼 mapping function fa(x), fb(x), fc(x), fd(x)를 습득하여 이를 추론시 이용한다는 것이다. 이 parameter mapping function을 습득한다는 것은 컴파일을 한다는 것과 같은 개념이다. 결국 ITE 규칙의 컴파일을 하고 그 결과를 추론에 사용하기위해 각 parameter mapping function을 어떻게 얻어내는가가 남은 문제이다. 이에 대해서는 4장에서 논하고 그 보다 먼저 이를 설명하는데 필요한 ITE 규칙의 일관성 조건(consistency condition)을 3장에서 논한다.

3. IF-THEN-ELSE 규칙의 일관성 조건

Parameter mapping function을 얻기위해서는 (예 2-6)의 각 테이블이 one-to-one 이나 many-to-one의 관계를 나타내는 것이어야한다. 이것을 function 습득을 위한 일종의 제약조건이나 가정이라 생각할 수도 있다. 그러나, 본 논문에서는 이런 조건을 ITE 규칙의 일관성 조건이라 해석한다. 즉, 이 조건이 정의될 수 있는 ITE규칙의 범위를 한정한다라기 보다 ITE규칙이 올바른 지식을 포함하도록하는 길잡이의 역할을 한다고 보는 것이다. 이를 본 논문에서는 일관성 조건이라 부른다.

예를 들어 (예 2-6)에서 ITE 규칙을 다음같이 입력될 수도 있다.

IF S(u;165,175,260,260) THEN S(u;60,70,200,200) ELSE
IF S(u;165,170,180,182) THEN S(u;63,70,75,77).

이런 경우 parameter a를 위한 table에서 이미 일관성 조건에 어긋나게 된다.

4. parameter mapping function 습득을 위한 휴리스틱

테이블로 저장된 ITE 규칙이 일관성 조건을 만족한다는 가정하에, 먼저 각 테이블들로 부터 각 parameter를 위한 mapping function fa(x), fb(x), fc(x), fd(x)를 얻어내는 휴리스틱을 설명한다. 이 4개의 function을 얻는 휴리스틱이 동일하므로 설명의 편의를 위해 이들을 대표해서 f(x)라 칭한다. 그 다음 k의 mapping function, f(k)에 대해 설명한다.

(그림 4-1)과 같은 테이블이 주어졌을 때 먼저 (식 4-2)에 따라 (그림 4-3)와 같은 상차표(divided difference table)를 구한다. 그리고, 구해진 상차표의 각열 최하단에 나타난 상차를 이용해서 (식 4-4)와 같은 뉴튼형식(newton form) 의 f(x)를 구한다. 여기서 상차 f[xi,...,xn]이 뉴튼형식의 n-i차항의 계수가 되는 이론적 배경에 대해서는 Conte, Carl de Boor[5]를 참조하기 바란다.

x	f(x)
x0	f(x0)
......	
xi	f(xi)
......	
xn	f(xn)

(그림 4-1)

Divided difference table (n=2)

x	f[ ]=f( )	f[,]	f[,,]
x0	f[x0]		
		f[x0,x1]	
x1	f[x1]		f[x0,x1,x2]
		f[x1,x2]	
x2	f[x2]		

(그림 4-3)

$$f[x0,..,xk] = \frac{f[x1,...,xk] - f[x0,..,x_{k-1}]}{xk - x0} \qquad \text{(식 4-2)}$$

상차표에서 구해진 값들중 실제로 (식 4-4)에서 사용되는 것은 f[xi,...,xn]이다. 따라서 계산의 효율성을 위해 f[xi,...,xn](=di)만을 <알고리즘4-1>을 이용하여 구한다. 또한 주어진 테이블로 부터 습득된 function은 결국 (식 4-4)이라 할 수 있는데, 실제 추론 면에 있어서 n+n(n+1)/2회의 덧뺄셈과 n(n+1)/2회의 곱셈을 해야하므로 효율적이지 못하다 보다 효율적인 추론을 위해서는 다음과 같이 nested newton form을 이용할 수 있다. (식 4-5)은 2n회의 덧뺄셈과 n회의 곱셈만으로 추론할 수 있다.

Algorithm 4-1 :

Given: $f(x_i)$ & x_i $\quad 0 \le i \le n$

Initial: $d_i = f(x_i)$ $\quad 0 \le i \le n$

output: $d_i = f[x_i, \ldots, x_n]$

for k = 1,..n
 for i = 0,...,n-k
  $d_i = (d_{i+1} - d_i)/(x_{i+k} - x_i)$;

$$f(x) = \sum_{i=0}^{n} f[xi, \ldots, xn] \prod_{j=i+1}^{n} (x - xj) \qquad \text{(식 4-4)}$$

$$f(x) = d_n + (x-x_n)\{d_{n-1} + (x-x_{n-1})\{d_{n-2} + (x-x_{n-2})\langle d_{n-2} + \ldots + (x-x_1)(d_1 + (x-x_0)d_0 \cdot \rangle]\} \qquad \text{(식 4-5)}$$

y 변수의 k값, yk는 x 변수에 입력된 parameter a, b, c, d, k값 (각각 xa, xb, xc, xd, xk라 칭함)에의해 결정되어야한다. 다시말해 k parameter의 mapping function은 6차원 공간상에 존재한다는 것이다. 그러나, 이러한 함수는 전문가가 지정한 ITE 규칙으로부터 얻을 수 없다. 예를 들어 다음과 같은 규칙이 ITE 규칙내에 있다하자.

IF $S(u; a, b, c, d)^1$ THEN $S(v; a', b', c', d')^1$ ELSE
IF $S(u; a, b, c, d)^2$ THEN $S(v, a', b', c', d')^2$ ELSE
IF $S(u; a, b, c, d)^{0.5}$ THEN $S(v; a', b', c'. d')^{0.5}$

여기서 물론 k는 linear mapping을 한다는 정보를 얻을 수 있다. 그러나 이것은 단지 xa=a, xb=b, xc=c, xd=d, xk=k라고 주어졌을 때이다. 결국 mapping function 전체를 얻기 위해서는 xa,xb,xc,xd의 4차 공간 내의 모든 점에 대해서 위의 예와같은 규칙이 존재해야 한다는 것이다. 이러한 지식을 전문가로 부터 얻기는 불가능하다.

본 논문에서는 단순히 parameter k에 대해서 domain에 무관하게 인간이 갖고 있는 직관을 고찰하여 이를 이용하고자 한다. 즉 yk = f(xk) 만을 고려한다. 이러한 접근 방식은 Zadeh[2], Mizumoto[6] 등이 "composition rule of inference"를 위한 operator를 개발하는데 취한 방법과 그 맥락을 같이 한다. 그들의 연구를 정리해 보면 f(k)는 positive linear function임을 알 수 있다. 즉, (예 1-1)에서 "그 사람은 무겁다"나 "그 사람은 매우 무겁다" 등의 결론이 추론되지, "그 사람은 약간 무겁다"와 같은 결론은 나오지 않는다는 것이다. 다시말해, 입력된 정보가 확신적이면 확신적일수록 결론도 확신적이 된다는 것이다.

따라서 f(k)를 다음처럼 정의하고자 한다.

$$f(k) = ck, \; c, k > 0$$

여기서 c는 입력정보의 확신정도를 결론에 어느 정도 반영할 것인가를 정해주는 것이다. c=1이면 100% 반영되는 것이 된다. 이 c값은 지식공학자가 시스템의 성능을 테스트하면서 조정할 수 있을 것이다.

5. 결론

본 논문이 제시한 방식의 큰 강점은 existing entity property를 만족시킨다는 것이며, ITE 규칙의 조건부에 나타나지 않았던 fuzzy concept가 입력될 경우 ITE에 나타난 양 변수간의 관계성향을 근거로 합리적인 결론을 출력시킨다는 것이다. 그 이유는 (식 4-4) 자체가 ITE 규칙에 나타난 n+1개의 mapping 정보를 근거로 두 변수간에 최소한의 오차를 갖는 n차 함수를 만들어 주는 것이기 때문이다. 이것은 기존의 방법들이 할 수 없었던 것이다 또한 추론속도 면에서 (식 4-5)가 n회의 곱셈만으로 가능하여 4개의 function에서 4n의 곱셈을 한다해도 O(n)이다. 반면 기존의 martix에의한 방식은 n**2회의 특수한 operation을 행하므로 O(n**2)이라 할 수 있다.

< REFERENCE >

[1] L. A Zadeh, "Fuzzy Sets", Infromation and Control, 8, 1965, pp. 338~353

[2] L A. Zadeh, "A Theory of Approximate Reasoning" in Machine Intelligence 9, Hayes, Michie and Mikulich eds , New York: J. Wiley, 1979, pp. 149~194.

[3] M. Cayrol, H. Farrey and H. Prade, "Fuzzy Reasoning based on Multivalent Logics in the Framework of Production Rule System", Int. Symp. Multi-valued Logic, IEEE, 1980, PP. 143~148.

[4] R. Martin-clouaire, "Efficient Deduction in Fuzzy Logic", Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based System", Paris, 1986, pp. 380~383.

[5] S. D. Conte and Carl de Boor, "Elementary Numerical Analysis: an algorithmic approach", McGraw-Hill, pp. 31~71

[6] S Fukami, M. Mizumoto and K. Tanaka, "Some Considerations on Fuzzy Conditional Inference", Fuzzy Sets and Systems, 4, 1980, pp. 243~273.