상태변화에 따른 자율적 의사결정을 위한 퍼지 에이전트

이태경*
동국대학교 컴퓨터멀티미디어학과
e-mail:tklee@mail.dongguk.ac.kr

A Fuzzy Agents for an Autonomous Decision Making on State Transition

TaeKyung Lee*
Dept. of Computer Science and Multimedia

요 약

본 논문에서는 상태변화에 대한 자율적 의사결정을 하는 퍼지논리를 이용한 에이전트를 구현하는 것을 연구의 목적으로 한다. 이를 위하여 제한적인 조건을 설정하여 부분적인 실험을 하였다. 에이전트를 구성하기 위한 추론방식으로는 max-product 기법을 사용하였으며, n개 퍼지 규칙들 또는 연관들 (A_1, B_1), ... , (A_n, B_n)을 가지는 상황을 고려하여 비퍼지화 작업을 수행하여 중심값을 추출하여 추론 작업을 실행하였다.

1. 서론

현대는 교통량의 증가로 인하여 차량의 통제가 어려운 경우가 많이 생긴다. 교통 법규 준수를 위한 여러 가지 방법과 기계들이 늘어나고 있지만 증가하는 차량에 발 맞추지 못하고 항상 부족한 상태이다. 이러한 상태를 해결하기 위하여 사람이 아닌 인공지능 로봇으로 인하여 사람을 대신할 수 있다면 해결 방안의 하나가 될 것이다.

지능형 에이전트에는 분산 인공지능, PDAs (personal digital assistants), 사용자 인터페이스, 전자우편, 그룹웨어, CSCW(computer support for co-operative work), 작업 관리 시스템 등의 기술이 영향을 미치고 있다. 즉, 이들의 개발로 지능적 에이전트의 성능이 점점 향상되고 있다[1,2].

그러므로 본 논문에서는 퍼지규칙에 의하여 외부의 환경의 상태변화에 따른 즉각적인 응답을 하는 반응형 퍼지 에이전트를 구현하고 실험을 한다. 본 논문에서는 에이전트의 추론을 구성하기 위하여 퍼지논리를 적용하였으며, 이에 따른 제한적인 조건을 설정하여 부분적인 실험을 하였다. 특히 에이전트를 구성하기 위한 추론 방식으로는 max-product 기법을 사용하였으며, n개 퍼지 규칙들을 가지는 상황을 고려하여 비퍼지화 작업을 수행 하여 중심값을 추출하여 추론 작업을 실행하였다.

2. 이론적배경

2.1 퍼지 논리와 퍼지 집합

퍼지시스템에서 퍼지규칙들에 언어 변수를 사용한다. 퍼지규칙은 정보의 전제에 포함된 다른 변수에 관한 정보로부터 정보의 결론에 포함된 언어 변수에 관한 정보를 추론한다. 예를 들어, "Speed is slow" 문장에서 변수 speed는 0과 100km/h 사이의 범위을 가진다. 여기서 변수의 논의영역(universe of discourse)은 언어적 변수의 가능한 값의 범위이다.

퍼지 집합은 집합에 연관된 항을 더욱 자연스럽게 반영하는 0과 1사이의 소속 값을 할당한다[4].

정의 : X의 원소(x)들을 가지는 X를 논의영역이라고 하자. X의 퍼지집합 A는 소속함수 $\mu_A(x)$(A에 소속값의 정도를 가지는 각 원소 x에 연관된다.)

에 의해 결정된다. 퍼지논리는 소속함수를 기초로 하여 사건에 값을 대입한다.

소속함수 정의 : $\mu_A(x) : X \rightarrow [0, 1]$

$\mu_A(x)$ = Degree($x \in A$)이고, $0 \leq \mu_A(x) \leq 1$ 이다.

2.2 퍼지 관계와 규칙베이스

퍼지관계는 구성원소들의 귀속도들이 구성된 형태에 따라 각기 다른 이름을 갖는다.

정의 : 퍼지관계 R은 2개의 퍼지 집합 A, B 사이의 Cartesian Product이다.

$R : X \rightarrow Y$

IF (X is A) THEN (Y is B)

퍼지 관계 R은 X에서 Y로의 관계이고, Cartesian Product X×Y의 퍼지 subset이며 맴버쉽 함수 $\mu_R(x, y)$로 나타낸다.

IF X is A THEN Y is B

이 퍼지규칙은 두 명제들 사이의 관계 또는 연관을 확립한다.

퍼지규칙에서 IF A THEN B를 적용했던 이 연산을 고려하자. 여기서 A는 X에 정의된 퍼지 집합이고 B는 Y에 정의된 퍼지 집합이다. 행 fit vectors A와 B는 아래와 같이 표현한다.

$$A = (a_1, a_2, \cdots, a_n); \quad a_i = \mu_A(x_i)$$

$$B = (b_1, b_2, \cdots, b_p); \quad b_i = \mu_B(y_i)$$

여기서 p 행렬 M에 의해 퍼지 n을 정의할 수 있다.

$$A \circ M = B$$

그리고 아래 식에 의해 b_j를 계산한다.

$$b_j = \max_{1 \leq i \leq n} \{\min(a_i, m_{ij})\}$$

2.3 Max-Product 퍼지 추론

퍼지추론은 Max-min와 Max-product 추론을 사용하며, Max-product 추론에서 M의 구성요소를 만들 때 사상 연산자로 표준 곱을 사용한다.

$$m_{ij} = a_i b_j$$

fit vector값은 보통 값을 포함하며, B′의 계산을 쉽게 하기 위하여 다시 새로운 변수를 만들 수 있는데 A의 측정은 x_k이다.

$$B' = \mu_A(x_k) \cdot \mu_B(y)$$

Max-product 추론 기법에 의한 계산결과는 삼각형모양의 형태 값을 갖는다. 이런 의미에서 Max-product 추론은 Max-min 추론보다 더 많이 정보를 보존한다.

2.4 비퍼지화와 다중 퍼지규칙

비퍼지화 기법은 퍼지중심값(fuzzy centroid) 기법으로서 B′로부터 단일 값 y_i를 제공한다[5].

$$y_i = \frac{\sum_{j=1}^{p} y_j \, m_{B'}(y_j)}{\sum_{j=1}^{p} m_{B'}(y_j)}$$

일반적으로 비퍼지화는 다중 규칙들이 같은 이벤트를 결론지을 때 중요하다. 또한, n개의 다중 퍼지 규칙들 또는 연관들(A_1, B_1), ... ,(A_n, B_n)을 가지는 상황을 고려한다.

이 집합에서 퍼지규칙들의 촛점은 단일 측정 A′가 주어진 B에서 전체 신뢰값의 결과이다. 우리는 규칙들의 은행에 병렬적으로 A′를 적용하는 것, 각 규칙에 대해 유도된 퍼지 집합 B'_i를 생성하는 것이다. 그리고 나서 유도된 퍼지집합 B′의 결과에 대한 합성을 만들기 위해 B'_i의 모두를 합친다. 여기서 B는 도메인 X에서 정의한다.

$$B' = B'_1 \cup B'_2 \cup \ldots B'_{n-1} \cup B'_n$$

$$= \max(B'_1(x), B'_2(x), \ldots, B'_{n-1}(x), B'_n(x)) \quad \text{for all } x \in X$$

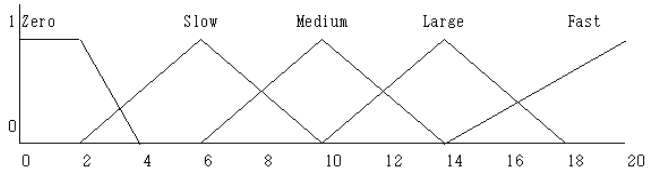
이 합집합 연산 다음에 중심값 방법을 사용하여 결과 B′를 비퍼지화할 수 있다.

3. 구현및실험결과

3.1 구현을 위한 조건

구현을 위한 전제조건은 1) 주행구간내 신호등은 2개(400m, 820m)가 있으며, 100m 전에 인식한다고 가정한다. 2) 갑자기 뛰어드는 사람과 앞의 차는 임의로 발생시키며, 차는 50%, 사람은 25%의 빈도로 발생한다. 또한, 언어변수 정의는 1) 입력 언어변수는 속도: 0 - 20m/s, 거리: 0 - 1000m이고, 2) 출력 언어변수 엑셀레이션 : -10 - 10m/s/s이다. 여기서 3) 택시의 구동제어를 위한 퍼지추론은 다중입력, 단일 출력 방식을 선택한다. 전방에 신호등, 차, 사람에 관한 변수는 행렬(5×5)로 처리를 하였다.

(1) 속도에 대한 퍼지집합

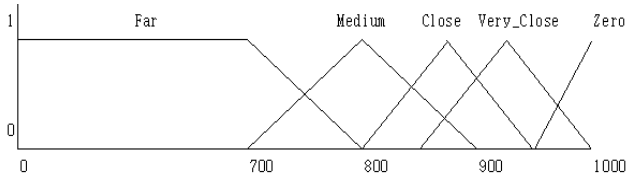


[그림 1] 속도의 소속함수

[표 1] 속도의 소속값들

Zero (ZS)		Slow (SS)		Medium (MS)		Slight_Fast (LS)		Fast (FS)	
X	Y	X	Y	X	Y	X	Y	X	Y
0	1	2	0	6	0	10	0	14	0
2	1	6	1	10	1	14	1	20	1
4	0	10	0	14	0	18	0		

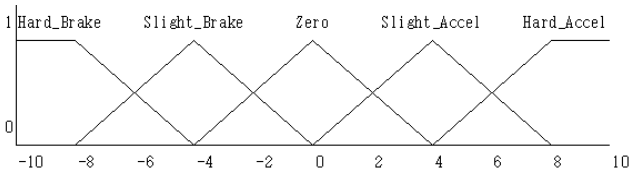
(2) 거리에 대한 퍼지집합



[그림 2] 거리의 소속함수[표 2] 거리의 소속값들

Far (FD)		Medium (MD)		Close (CD)		Very_ Close(VD)		Zero (ZD)	
X	Y	X	Y	X	Y	X	Y	X	Y
0	1	700	0	800	0	850	0	950	0
700	1	800	1	875	1	925	1	1000	1
800	0	900	0	950	0	100	0		

(3) 엑셀레이션에 대한 퍼지집합



[그림 3] 엑셀레이션의 소속함수

[표 3] 엑셀레이션의 소속값들

Hard_Brake(HB)		Slight_ Brake(SB)		Zero (ZA)		Slight_ Accel(SA)		Hard_ Accel(HA)	
X	Y	X	Y	X	Y	X	Y	X	Y
-10	1	-8	0	-4	0	0	0	4	0
-8	1	-4	1	0	1	4	1	8	1
-4	0	0	0	4	0	8	0	10	1

3.2 주요 알고리즘

(1) 소속 값 계산

거리의 경우 입력된 거리 값(xo)과 거리의 논의영역(행렬 X)과 퍼지집합(행렬 A)을 통해 소속 값을 계산한다. 속도의 경우도 거리와 마찬가지로 입력된 속도 값과 속도의 논의영역과 퍼지집합을 통해 소속 값을 계산한다.

```
float MemberShip(float xo, float X[], float A[])
{
  // 논의영역 밖이면 가장 가까운 소속 값 계산
  if (xo < X[0]) {
     return(A[0]);
  } else if (xo > X[N_ELEM -1]) {
    return(A[N_ELEM -1]);
  }

  // 소속 값 계산
  for (si = 0; si < N_ELEM-1; si++) {
    if ((xo >= X[si])&&(xo <= X[si+1])) {
      if (X[si] == X[si+1]) {
         소속 값은 가장 큰 값으로
      } else {
         소속 값은 선형 보간법으로
      }
    }
  }
  return(ao);
}
```

(2) 퍼지 추론

가. max-product 추론

max-product 추론에 의해 엑셀레이션의 소속값을 계산한다. 예를 들어 「A_HA = max(S_SS*D_FD, A_HA)」의 경우는 「IF 속도 = SS AND 거리 = FD THEN 엑셀레이션 = HA」를 계산하는 것이다. A_HA에는 엑셀레이션의 소속 값이 저장된다.

```
A_SA = max(S_ZS, A_SA);
A_HA = max(S_SS*D_FD, A_HA);
        . . . .   . . . .
A_SB = max(S_FS*D_VD, A_SB);
A_HB = max(S_FS*D_ZD, A_HB);
```

나. 매칭되는 규칙들의 합집합

max-product 추론에서 계산된 엑셀레이션의 소속값과 엑셀레이션의 퍼지집합을 통해 엑셀레이션의 퍼지집합을 추론할 수 있다. 예를 들면 「InferEngine (A_HB, HB, R)」은 소속 값 A_HB와 퍼지집합 HB의 곱을 통해 엑셀레이션의 집합(행렬 R)을 추론한다.매칭되는 엑셀레이션 퍼지집합은 「Maximum(R, O, O)」에 의해 행렬 O에 더해진다.

```
InferEngine(A_HB,  HB, R);
Maximum(R, ZERO, O);
InferEngine(A_SB,  SB, R);
Maximum(R, O, O);
InferEngine(A_ZA,  ZA, R);
Maximum(R, O, O);
InferEngine(A_SA,  SA, R);
Maximum(R, O, O);
InferEngine(A_HA,  HA, R);
Maximum(R, O, O);
```

(3) 비퍼지화

매칭되는 규칙들의 합집합에서 합해진 엑셀레이션의 퍼지집합(행렬 B)과 엑셀레이션의 퍼지집합(행렬 Y)은 퍼지중심값 공식에 의해 Crisp value(Yo)로 계산된다. yB_sum은 $\sum_{j=1}^{p} y_j \, m_{B}(y_j)$, B_sum은 $\sum_{j=1}^{p} m_{B}(y_j)$ 을 나타낸다. yB_sum / B_sum을 통해 비퍼지화 값이 계산된다.

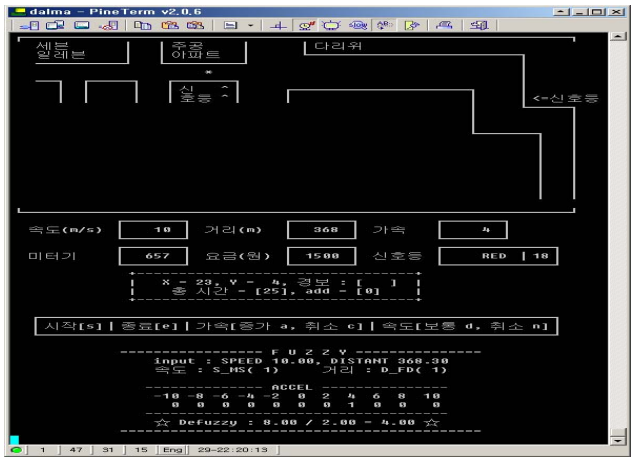
```
void DeFuzzyier(float Y[N_ELEM], float
                B[N_ELEM], float *Yo)
{
  yB_sum = 0.0;  B_sum  = 0.0;

  for (si = 0; si < (N_ELEM -1); si++) {
     yB_sum += ((Y[si] + Y[si+1])/2)*
               ((B[si] + B[si+1])/2);
     B_sum += (B[si] + B[si+1])/2;
  }

  if (B_sum != 0.0) {
    *Yo = yB_sum/B_sum;
}
```

3.3 실험 결과

(1) 전방에 신호등이 있는 조건

비퍼지화된 엑셀레이션 값과 Signal_distance[5][5]에 정의된 값에 의해 택시의 속도가 결정된다.

아래 [그림 4]와 같이 FUZZY input에서 속도가 10이고 거리가 368일 때 엑셀레이션은 4이므로 속도는 14가 되어야 하지만 속도는 10이다.



[그림 4] 신호등 앞에서의 상황

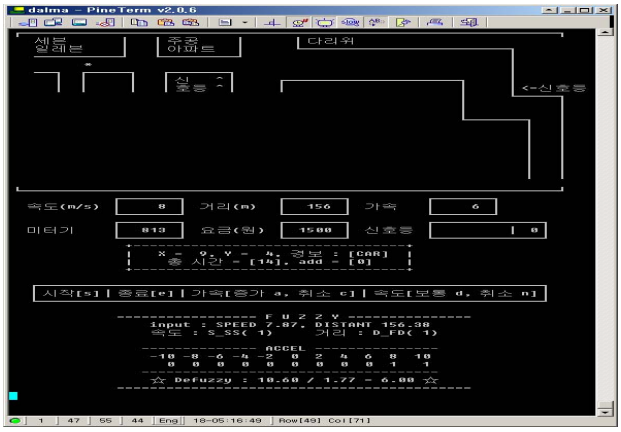
(2) 전방에 차, 사람이 있는 조건

가. 전방에 차가 있는 경우

비퍼지화된 엑셀레이션 값과 Car_distance[5][5]에 정의된 값에 의해 택시의 속도가 결정된다.

나. 전방에 사람이 있는 경우

비퍼지화된 엑셀레이션 값과 Man_distance[5][5]에 정의된 값에 의해 택시의 속도가 결정된다. 아래 [그림 5]와 같이 FUZZY input에서 속도가 8이고 거리가 156일 때 엑셀레이션은 6이므로 속도는 14가 되어야 하지만 속도는 8이다.



[그림 5] 차가 나타난 상황

4. 결론 및 향후과제

본 논문에서는 반응형 에이전트를 구성하기 위하여 퍼지논리를 적용하였으며, 이에 따른 제한적인 조건을 설정하여 부분적인 실험을 하였다. 추론방식으로는 max-product 기법을 사용하였으며, n개 퍼지 규칙들을 고려하여 비퍼지화 작업을 수행 하여 중심값을 추출하여 추론 작업을 실행하였다. 차후의 과제로는 퍼지지식을 학습하는 과정이 포함되는 지식베이스를 구성하여야 한다.

참고문헌

[1] 조영임, 최신 인공지능, 학문사, 1999

[2] Nils J. Nilsson, 최중민외 3인 공저, 인공지능 -지능형 에이전트를 중심으로-, (주)사이텍미디어, 2000

[3] Stuart Russell, Peter Novig, Artificial Intelligence -A Modern Approach-, Prentice Hall, 1995

[4] Timothy J, Ross, Fuzzy Logic with Engineering Applications, Mc Graw Hill, 1995

[5] John Durkin, Expert System -Design and Development-, Macmillan,1994

[6] Walter Brenner, Rudiger Zarnekow, Hartmut Wittig, Intelligent Software Agents Springer, 1998