퍼지 데이타에 대한 퍼지 결정트리 기반 분류규칙 마이닝

(Classification Rule Mining from Fuzzy Data based on Fuzzy Decision Tree)

이 건 명 †

(Keon-Myung Lee)

요 약 결정트리 생성은 일련의 특징값으로 기술된 사례들로부터 분류 지식을 추출하는 학습 방법중의 하나이다. 현장에서 수집되는 사례들은 관측 오류, 주관적인 판단, 불확실성 등으로 인해서 애매하게 주어지는 경우가 많다. 퍼지숫자나 구간값을 사용함으로써 이러한 애매한 데이터의 수치 속성은 쉽게 표현될 수 있다. 이 논문에서는 수치 속성은 보통값 뿐만아니라 퍼지숫자나 구간값을 갖을 수 있고, 비수치 속성은 보통값을 가지며, 데이터의 클래스는 확신도를 가지는 학습 데이터들로부터, 분류 규칙을 마이닝하기 위한 퍼지 결정트리 생성 방법을 제안한다. 또한 제안한 방법에 의해 생성된 퍼지 결정트리를 사용하여, 새로운 데이터에 대한 클래스를 결정하는 추론 방법을 소개한다. 한편, 제안된 방법의 유용성을 보이기 위해 수행한 실험의 결과를 보인다.

Abstract Decision tree induction is one of useful methods to extract classification knowledge from a set of instances described in feature values. Due to observation error, subjective judgement, inherent uncertainty, and so on, real-world data are often obtained in a fuzzy way. Fuzzy numbers and interval values are useful to represent fuzzy numeric attributes. This paper presents a method inducing a fuzzy decision tree that mines classification rules from a set of fuzzy training data. In our method, training data may have various types of attribute values including non-numeric values, fuzzy numbers, and intervals. Training data also may have uncertain classification in terms of certainty degrees of classes. In addition, it introduces an inference procedure to determine the class of new data based on the fuzzy decision trees created by the proposed induction method. It also shows some experimental results to show the applicability of the proposed method.

1. 서 론

결정트리(decision tree)는 분류(classification) 또는 의사결정(decision making)을 위해 특징 속성값으로 표현된 사례들로부터 분류 지식을 추출하기 위해 널리 사용되어 온 방법이다[1]. 사례들로부터 결정트리를 생성하기 위한 방법으로 ID3, C4.5, CART등의 단변수(univariate) 결정트리 알고리즘, OC1, LMDT 등의 다변수(multivariate) 결정트리 알고리즘 등 여러 가지 방법이 제안되어 왔다[1].

실제 현장에서 얻어지는 많은 데이타는 관측오류, 불확실성, 주관적인 판단 등으로 인해서 애매한 형태로 주어진다. 대부분의 기존 결정트리 생성 방법은 데이타에 내포된 애매함에 대해 충분히 고려를 하지 못하고 있지만, 퍼지 데이타에 대한 몇몇 결정트리 생성 방법도 시도되고 있다[2-6]. Yuan[2] 등은 각 속성에 대한 퍼지 언어항[7]을 정의한 다음, 속성값의 퍼지 언어항에 대한 소속정도 정보를 이용하여 퍼지 결정트리를 생성하는 방법을 제안하였다. 이 방법에서는 속성값이 퍼지 언어항들에 대한 소속정도로 주어지는 것을 전제하고 있으며, 속성값 자체가 퍼지값인 경우는 고려하지 않고 있다. Ittner[3-4] 등은 연속 수치영역 속성값으로 표현되

· 이 연구는 한국학술진흥재단 1998년도 신진교수과제 지원으로 수행된 것임.

† 비 회 원 : 충북대학교 컴퓨터과학과 교수
kmlee@cbucc.chungbuk.ac.kr

논문접수 : 2000년 2월 1일
심사완료 : 2000년 11월 9일

는 데이타에 대한 퍼지 결정트리 생성 방법을 제안하였는데, 대상으로 하는 데이타가 퍼지값이 아닌 보통값(crisp value)으로 표현되는 것이다. Janikow[5]는 수치 속성만을 갖는 데이타에 대한 퍼지 결정트리 생성 방법을 제안하였는데, 속성값으로 보통값만이 허용되고, 속성영역을 분할하는 퍼지 언어항이 미리 주어진다고 가정하였다. Kim[14] 등은 히스토그램 분석을 통해서 퍼지 언어항을 생성한 다음, 이를 이용하여 퍼지 결정트리 생성 방법을 제안하였는데, 학습 데이타가 보통값을 갖는 수치 속성만을 포함하는 데이타이다.

이들 퍼지 결정트리 생성 방법은 사용되는 학습 데이타의 형태에 따라 다음 두가지로 나눌 수 있다. 하나는 일반 데이타들에 대한 퍼지 결정트리를 생성하는 것으로, 생성되는 퍼지 결정트리는 일반화(generalization) 특성을 향상시키기 위해서 소속함수에 의해 정의되는 퍼지 언어항을 사용한다. 퍼지 언어항을 사용하게 되면 데이타의 속성값에 대한 약간의 변화에 의해 데이타 클래스가 갑자기 부적절하게 변하는 것을 피할 수 있게 된다. 다른 하나는 퍼지값을 포함하는 학습 데이타들로부터 퍼지 결정트리를 생성하는 것이다. 퍼지 데이타에 대한 기존의 퍼지 결정트리 생성 방법들 대부분은 수치 속성만을 갖는 퍼지 데이타만을 학습 데이타로 하는 등 제한적인 형태의 퍼지 데이타만을 대상으로 하고 있다.

이 논문에서 대상으로 하는 학습 데이타는 수치 속성 및 비수치 속성으로 기술되는 것으로, 연속 수치영역 속성값은 보통값뿐만 아니라 퍼지숫자[7] 및 구간값에 의해 표현되고, 비연속 영역 수치 속성은 보통값에 의해 표현되고, 비수치 속성값은 보통값에 의해 표현되며, 데이타가 속하는 클래스에 대한 확신도가 있는 것이다. 논문에서는 이러한 퍼지 데이타들로부터 분류규칙을 마이닝하는데 적용할 수 있는 새로운 퍼지 결정트리 생성 방법을 제안한다

이 논문은 다음과 같이 구성된다. 2절에서는 데이타의 표현방법에 대해서 기술하고, 3절에서는 퍼지 결정트리의 형태에 대해서 살펴본다. 4절에서는 제안한 퍼지 결정트리 생성 방법을 소개하고, 5절에서 제안된 방법에 대한 실험결과를 보이고, 끝으로 6절에서 결론을 맺는다.

2. 데이타의 표현

이 논문에서 고려하는 객체 또는 사건에 대한 데이타 D_i는 다음과 같이 객체 또는 사건의 특징을 나타내는 속성값들로 구성된 것이다.

$$D_i = (A_1^i, A_2^i, \ldots, A_n^i) \quad (1)$$

여기에서 A_j^i는 j 번째 속성값으로서, 연속 수치영역 속성인 경우에는 보통값뿐만 아니라 퍼지숫자 및 구간값에 의해서 표현될 수 있고, 이산영역 수치 속성인 경우에는 보통값으로 표현되며, 비수치 속성인 경우에는 보통값에 의해 표현되는 것이다. 한편 퍼지 결정트리를 생성하기 위해 사용되는 학습 데이타인 경우에는 다음과 같이 데이타의 클래스와 이에 대한 확신도를 포함된다. 여기에서 C_i는 데이타 D_i의 클래스를 나타내고, cf_i는 데이타 D_i가 클래스 C_i에 속한다는 것에 대한 확신도로서 구간 [0,1]사이의 값을 갖는다.

$$D_i = (A_1^i, A_2^i, \ldots, A_n^i, C_i ; cf_i) \quad (2)$$

퍼지숫자로는 표현 및 처리가 간단하여 널리 사용되는 사다리꼴 퍼지숫자 $Trap(\alpha,\beta,\gamma,\delta)$만을 허용한다. 다음의 (식 3)은 사다리꼴 퍼지숫자에 대한 소속함수를 나타낸 것이고, (그림 1)은 사다리꼴 퍼지숫자의 형태를 본인 것이다.

$$Trap(\alpha,\beta,\gamma,\delta) = \begin{cases} 0 & \text{if } x < \alpha \\ (x-\alpha)/(\beta-\alpha) & \text{if } \alpha \le x \le \beta \\ 1 & \text{if } \beta \le x \le \gamma \\ (x-\delta)/(\gamma-\delta) & \text{if } \gamma \le x \le \delta \\ 0 & \text{if } x > \delta \end{cases} \quad (3)$$

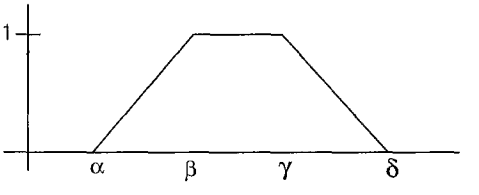


그림 1 사다리꼴 퍼지숫자 Trap($\alpha,\beta,\gamma,\delta$)

퍼지숫자로서 사다리꼴 퍼지숫자만을 사용한다면, 수치 속성값을 나타내기 위해 사용되는 보통값, 퍼지숫자 및 구간값이 모두 다음과 같이 사다리꼴 퍼지숫자로 표현될 수 있다.

보통값 $a = Trap(a, a, a, a)$ (4)

퍼지숫자 $F = Trap(a, b, c, d)$ (5)

구간값 $[a, b] = Trap(a, a, b, b)$ (6)

3. 퍼지 결정트리

이 논문에서 대상으로 하는 퍼지 결정트리의 형태는 (그림 2)와 같은 것이다. 트리에서 각 비단말(non-terminal) 노드는 하나의 속성(예, *Salary, Age* 등)을 나타내고, 부모노드를 자식노드로 연결하는 링크에는 부모노드의 속성 영역을 분할하는 값이 부여된다. (그림 2)에서 근(root) 노드의 *Job*과 같은 비수치 속성

의 링크에는 {*A,D*}, {*B*}과 {*A,C*}와 같이 속성값의 집합이 부여된다. *Salary*와 같은 수치 속성에 대한 노드의 링크에는 트리 생성과정에 소속함수의 형태가 정의되는 '*low*' 또는 '*high*' 등과 같은 퍼지 언어항이 부여된다. 단말노드는 근노드에서 해당 노드로의 경로상에 있는 모든 조건을 만족하는 데이타가 속하는 클래스를 나타내는데, 단말노드가 나타내는 클래스는 퍼지집합 형태(예, {(C1, 0.4), (C3, 1.0)})로 표현된다. 예를 들어, 어떤 단말노드의 클래스가 {(C1, 0.4), (C3, 1.0)}와 같이 부여되어 있다면, 이 단말노드에 도달하는 데이타는 클래스 C1에는 0.4의 가능성 정도로 속하고, 클래스 C3에는 1.0의 가능성 정도로 속한다는 것을 의미한다. 일반 결정트리와는 달리, 퍼지 결정트리에서는 근노드에 입력된 데이타는 여러 단말노드에 도달할 수 있다. 따라서 입력된 데이타에 대한 클래스는 데이타가 도달한 모든 단말노드의 클래스 정보를 이용하여 결정되어야 한다. 퍼지 데이타로부터 이러한 퍼지 결정트리를 생성하기 위해 제안한 방법과, 이러한 퍼지 결정트리를 이용해서 임의의 데이타가 속할 클래스를 결정하는 방법은 다음 4절에서 자세히 설명한다.

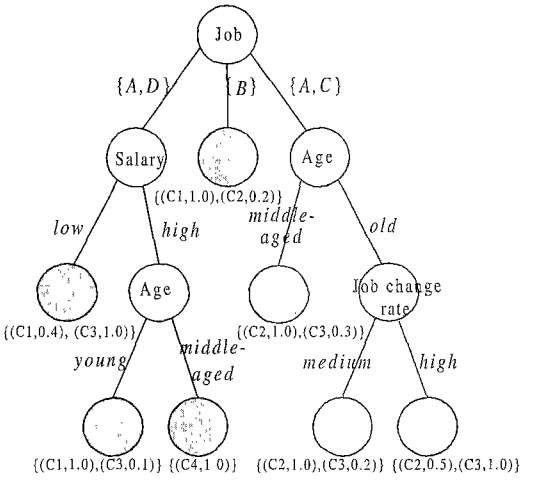


그림 2 퍼지 결정트리의 예

4. 제안한 퍼지 결정트리 생성 방법

제안한 퍼지 결정트리 생성 방법은 노드를 반복적으로 분할해 가는 트리를 확장하는 ID3 알고리즘[1]과 비슷한 접근방법을 따른다. ID3와 같이 반복적인 분할 방법으로 퍼지 결정트리를 생성하기 위해서는 비수치 속성 및 수치 속성의 영역분할 방법, 분기 속성 선택방법, 노드의 링크를 따라 내려가는 데이타의 링크값에 대한 만족정도를 계산하는 방법, 단말 노드에 대한 클래스 부여 방법이 필요하다. 또한 결정트리를 사용하여 새로운 데이타를 분류하는 추론 방법이 필요하다. 이 절에서는 제안한 방법에서의 이들 사항에 대해서 기술한다.

4.1 속성 영역 분할

제안한 결정트리 생성 방법에서는 결정트리의 크기가 지나치게 커지는 것을 피하기 위해서 트리의 분기 크기를 제어할 수 있도록, 비수치 속성 영역 및 수치 속성 영역을 원하는 개수로 분할할 수 있도록 한다. 비수치 속성 영역은 속성값의 집합들로 분할하고, 수치 속성 영역은 사다리꼴 퍼지숫자를 사용하여 분할한다. 먼저 비수치 속성에 대한 영역 분할을 위해 제안한 방법을 설명한 다음, 수치 속성 영역 분할에 대해서 설명한다.

4.1.1 비수치 속성 영역 분할

비수치 속성 분할을 위해 주어진 속성 A에 대하여 다음과 같이 정의되는 클래스별 원소집합(classwise element set, CWS)라는 개념을 도입한다.

$$CWS_i^k = \{x \mid D_j.A = x,\ x \in \Delta_A^k,\ D_j.class = i,\ x_k(D_j) > 0\},\ i \in K \quad (7)$$

여기에서 CWS_i^k는 노드 N_k에서의 클래스 i에 대한 클래스별 원소집합을 나타낸다. $D_j.A$는 데이타 D_j의 속성 A의 값을 나타내고, Δ_A^k는 노드 N_k에 나타나는 데이타의 속성 A의 값들의 집합이고, $D_j.class$는 데이타 D_j가 속하는 클래스를 나타내며, $x_k(D_j)$는 데이타 D_j의 노드 N_k에 대한 소속정도이고, K는 클래스 이름의 집합이다. 예를 들어, $K=\{1,2,3,4\}$, $\Delta_A^k=\{a,b,c,d,e\}$이고, 노드 N_k에 도달한 데이타들의 속성값 $x.A$와 클래스 $x.class$의 쌍으로 구성된 집합이 $\{(a,1),(b,1),(b,2),(c,1),(c,2),(c,3),(d,2),(d,4),(e,3),(e,4)\}$와 같다고 하자. 이때 노드 N_k에 대한 클래스별 원소집합은 다음과 같다.

$$CWS_1^k=\{a,b,c\}, \qquad CWS_2^k=\{b,c,d\},$$
$$CWS_3^k=\{c,e\}, \qquad CWS_4^k=\{d,e\}$$

클래스별 원소집합 CWS_i^k는 '노드 N_k에서 CWS_i^k에 속하는 원소를 속성값으로 갖는 데이타는 클래스 i에 속할 가능성이 있다'는 것을 의미한다. 이러한 관점에서 볼 때, CWS_i^k를 비수치 속성 분할을 위한 기본 분할로 사용하는 것이 타당하다고 생각할 수 있다. 제안한 방법에서는 클래스별 원소집합들을 분기 개수(branching factor) 만큼으로 조합(combination)하는 방법으로 비수치 속성 영역을 분할한다. 위의 클래스별 원소집합에 대해서 $\{CWS_1^k, CWS_2^k\}$와 $\{CWS_3^k, CWS_4^k\}$를 분할 조합으로 한다

면, 속성 A의 영역은 $\{a,b,c,d\}$과 $\{c,d,e\}$ 두 개의 속성값의 집합으로 분할하는 것이 된다. 여기에서 보는 바(예, 원소 d)와 같이 제안한 방법에서는 하나의 속성값이 여러 분할에 포함되는 것도 허용한다. 분기 개수가 클래스 개수보다 크다면, 클래스별 원소집합을 다시 분할하는 별도의 처리가 필요하다. 다음은 비수치 속성 공간을 분할하는 알고리즘이다.

procedure 비수치_속성_공간_분할

입력 : 비수치 속성에 대한 속성값의 집합, 분기 개수

출력 : 속성값의 집합들로 구성된 분할 조합과 이에 대한 정보이득(information gain) 값

begin

현재 노드에 대해서 클래스별 원소집합 *CWS*을 구한다.

if (클래스 크기가 분기 개수보다 크거나 같다) **then**

*CWS*를 사용하여 가능한 모든 분기 조합을 생성한다.

각 조합에 대하여 정보이득을 구한다.

가장 큰 정보이득을 갖는 분할 조합을 반환한다.

else

*CWS*의 원소들로 구성된 초기 분할 조합을 만든다.

각 분할에 대하여 엔트로피를 계산한다.

do

가장 큰 엔트로피값을 갖는 분할 *MC*을 찾는다.

분할 *MC*을 두 개로 분할하는 모든 가능한 조합을 만든다.

기존 분할과 동일한 분할을 포함하는 조합은 제거한다.

각 조합에 대해서 정보이득을 계산한다.

정보이득이 가장 좋은 조합을 선택한다.

*MC*를 선택된 조합의 두 개 분할로 대체한다.

until 분기 개수 만큼의 분할 생성

정보이득과 함께 최종 분할조합을 반환한다.

endif

end.

위의 절차에서 사용된 분할 조합에 대한 정보이득 및 엔트로피를 구하는 방법에 대해서는 4.2절에서 설명한다.

4.1.2 수치 속성 영역 분할

제안한 방법은 수치 속성 영역을 사다리꼴 퍼지숫자를 사용하여 퍼지 분할한다. 이산영역 수치 속성값도 연속 수치영역 속성값의 특별한 경우로 간주하여 연속 수치영역 속성의 경우와 같은 방법으로 처리한다.

제안한 수치 영역의 분할 방법을 다음과 같다. 제안한 방법에서 고려하는 수치영역의 속성값인 보통값, 퍼지숫자, 구간값은 (식 4, 5, 6)에 보인 바와 같이 사다리꼴 퍼지숫자로 표현될 수 있다. 따라서 모든 수치 속성값을 퍼지 사다리꼴 퍼지숫자로 간주하여, 이들 퍼지숫자의 지지 집합(support)[7]에 대한 경계값(즉, 양 끝점)들을 찾아서 오름차순으로 정렬한다. 정렬된 끝점들의 배열에서 인접한 두 끝점이 서로 다른 클래스에 속한다면 이 두 끝점 사이를 영역 분할 경계의 후보로 고려한다. (그림 3-(a))는 속성값을 소속함수 형태로 표현한 것으로, 소속함수 위의 *A, B, C*등의 기호는 해당 속성값을 갖는 데이타가 속하는 클래스를 나타낸다. (그림 3-(b))는 이들 소속함수에 대한 경계값과 이들이 속하는 클래스 이름을 써놓은 것이다. (그림 3-(b))에서 P_1, P_2, P_3와 같이 모두 동일한 클래스에 속하는 경계값 사이에는 분할 경계를 설정하는 것이 바람직하지 않고, 서로 다른 클래스에 속하는 경계값 사이에 분할 경계를 설정하는 것이 바람직하다. (그림 3-(c))는 P_3와 P_4, P_5와 P_6, P_9과 P_{10} 사이에 분할 경계를 설정하여 퍼지 분할한 예를 보인 것이다. (그림 3-(c))에서 보는 바와 같이 사다리꼴 퍼지숫자를 이용하여 퍼지 분할을 할 때, 분할 경계로 선택된 점(예, P_3와 P_4의 사이)을 중심으로 두 개의 사다리꼴 퍼지숫자(예, L_1와 L_2)를 정의한다. 이 때 경계로 선택된 점(예, m)의 양쪽 사다리꼴 퍼지숫자에 대한 소속정도가 0.5가 되고, 경계영역(즉, P_3와 P_4의 사이) 내에서만 소속함수가 중첩되도록 한다. 영역분할을 할 때는, 가능한 영역분할 방법 중에서 가장 좋은 분류 특성을 갖는 것을 선택한다. 영역분할 방법에 대한 분류 성능은 다음 절에서 설명하는 엔트로피 척도에 기반한 정보이득을 이용하여 평가한다.

다음은 수치 속성 영역을 분할하는 알고리즘이다.

procedure 수치_속성_영역분할

입력 : 수치 속성의 속성값의 집합, 분기 개수

출력 : 퍼지분할과 이에 대한 정보이득

begin

현재 속성에 대한 모든 속성값들의 지지 집합에 대한 경계값들을 구한다.

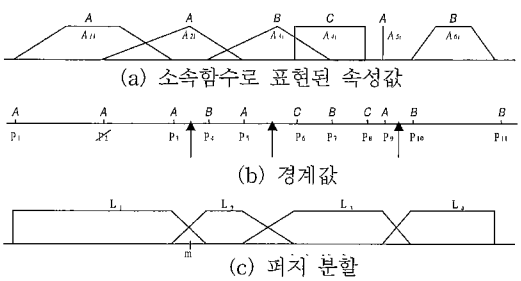
경계값을 크기순으로 리스트에 정렬한다.

동일한 클래스에 속하는 3개 이상의 연속된 경계값으로 구성된 리스트 구역(subsequence)에서, 내부에 있는 경계값들을 제거한다.

리스트에서 분기 개수보다 1개 적은 수의 점(실제, 인접한 두점에 의한 경계)을 선택하는 모든 가능한 조합을 만든다.

각 조합에 대하여 선택된 점을 사용하여 분기 개수 만큼의 사다리꼴 퍼지숫자를 통해 정의된 퍼지분할을 생성한다.

생성된 퍼지 분할들에 대해서 정보이득을 계산한다.
가장 큰 정보이득을 갖는 퍼지 분할을 반환한다.
end.



(a) 소속함수로 표현된 속성값

(b) 경계값

(c) 퍼지 분할

그림 3 수치 속성 영역 분할

4.2 분기 속성 선택

속성의 영역을 결정트리의 분기 개수 만큼으로 분할하는 방법이 4.1절에서 살펴본 바와 같이 여러가지 있을 수 있으므로, 그 중에서 가장 효과적인 것을 선택하는 것이 필요하다. 또한, 분기 속성으로 사용될 수 있는 속성이 여러 개 일 때는, 그중 하나를 선택하는 것이 필요하다. 따라서 제안한 방법에서는 분기 속성이 될 수 있는 각 속성에 대해서 가장 효과적인 속성 영역분할 방법을 선택한 다음, 이들 중에서 가장 효과적인 영역으로 영역분할을 하는 속성을 분기 속성으로 선택한다. 속성 영역 분할 방법을 결정하기 위해, 엔트로피 척도[1]에 기반한 정보이득(infomation gain)[1]을 이용하여 영역분할 방법을 평가한다. 여기에서 사용하는 엔트로피 척도 및 정보이득 척도는 데이타의 소속정도값을 반영할 수 있도록 기존 척도를 확장한 것이다. S^k를 현재 노드 N_k에 도달한 데이타의 집합, 즉 $S^k=\{(x,\mu_{S^k}(x))\,|\,x\in U\}$ 이라고 하자. 여기에서 x는 데이타를 나타내고, $\mu_{S^k}(x)$는 x의 S^k에 대한 소속정도를 나타내며, $Supp(S^k)$는 S^k의 지지 집합을 나타낸다. 이때 S^k에 대한 엔트로피 $Entropy(S^k)$는 다음과 같이 정의된다.

$$C^i_{S^k} = \sum_{\substack{x,\, class=i \\ x\in Supp(S^k)}} \mu_{S^k}(x) \quad C_{S^k} = \sum_i C^i_{S^k} \quad P_i^{S^k} = \frac{C^i_{S^k}}{C_{S^k}}$$

$$Entropy(S^k) = -\sum_i P_i^{S^k} \log_2 P_i^{S^k} \tag{8}$$

엔트로피 척도는 해당 집합의 순수도(purity)를 평가하는 척도로서 집합이 동일한 클래스에 속하는 원소(데이타)로 구성될수록 작은 값을 갖게 된다.

$S^k_{v|A}$가 데이타 집합 S^k를 가지고 있는 현재 노드 N^k를, 속성 A를 분기 속성으로 하여 분할할 때, v값을 갖는 링크에 연결된 자식노드에 전달되는 데이타의 집합이라 하자. $S^k_{v|A}$는 다음과 같이 정의되는 S^k의 퍼지 부분집합이다.

$$S^k_{v|A}=\{(x,\mu_{S^k_{v|A}}(x))\,|\,\mu_{S^k_{v|A}}(x) = f(\mu_{S^k}(x), M(x.A,v)),\ x\in Supp(S^k)\} \tag{9}$$

여기에서 A는 분기 속성을, v는 링크값을 나타내고, $M(x.A,v)$는 링크값 v에 대한 데이타 x의 속성 A값의 만족정도를 나타내며, $f(\cdot,\cdot)$는 min, 곱셈 등의 T-norm 연산자[7]이다. 이때 데이타 집합 S^k에 대한 속성 A의 분할방법에 대한 정보이득은 다음과 같이 계산된다. 여기에서 $Iterm(A)$은 속성 A을 영역분할할 때 각 분할을 나타내는 값들의 집합, 즉 현재 노드에서 자식노드로 연결되는 링크에 부여된 값들의 집합을 나타낸다.

$$Gain(S^k,A) = Entropy(S^k) - \sum_{v\in Iterm(A)} \frac{C_{S^k_{v|A}}}{C_{S^k}} Entropy(S^k_{v|A}) \tag{10}$$

$$Entropy(S^k_{v|A}) = -\sum_i P_i^{S^k_{v|A}} \log_2 P_i^{S^k_{v|A}} \tag{11}$$

정보이득이 큰 분할 방법일수록 생성되는 결정트리의 복잡도를 줄이는 경향이 있으므로, 정보이득을 계산하여 정보이득이 가장 큰 속성분할 방법을 선택하여, 이를 이용하여 결정트리를 확장한다.

4.3 트리 링크에서의 만족도 검사

이 논문에서 대상으로 하는 퍼지 결정트리의 링크에 부여될 수 있는 링크값은, 퍼지숫자 또는 속성값의 집합이다. 이러한 퍼지 결정트리를 이용하여 분류할 데이타는 속성값으로 보통값, 퍼지숫자, 구간값 등을 갖는다. 일반 결정트리를 이용하여 데이타를 분류할 때, 주어진 데이타는 근노드에서 시작하여 자신의 속성값과 일치하는 값을 갖는 링크를 따라 단말노드로 내려가게 되며, 단말노드에 부여된 클래스를 자신의 클래스로 한다. 퍼지 결정트리를 이용하여 데이타를 분류할 때는 데이타의 속성값이 링크의 값과 완전히 일치하는 경우보다는 부분적으로 일치하는 경우가 많다. 따라서 퍼지 결정트리를 이용하여 데이타를 분류할 때는, 링크값에 대한 데이타 속성값의 만족정도를 계산하여, 이 값에 따라 (식 9)와 같은 방법으로 데이타를 자식노드로 내려보낸다. 제안한 방법에서는 링크값(링크에 부여된 값)에 대한 데이타 속성값의 만족정도 $M(\cdot,\cdot)$를 다음과 같은 방법으로 계산한다.

비수치 속성인 경우에는, 속성값 n의 링크값인 속성값의 집합 L에 대한 소속여부에 따라 다음과 같이 만족정도를 결정한다.

$$M(n,L) = \begin{cases} 1 & \text{if } n\in L \\ 0 & otherwise \end{cases} \tag{12}$$

속성값 F가 구간값이나 퍼지숫자인 경우에는, 링크값인 퍼지 언어항 L에 대한 만족정도를 다음과 같이 계산한다.

$$M(F,L) = \frac{\int \mu_{F\cap L}(x)dx}{\int \mu_F(x)dx} \quad (13)$$

속성값 c가 보통값인 경우에는, c의 링크값인 퍼지 언어항 L에 대한 소속정도를 만족정도로 사용한다.

$$M(c,L) = \mu_L(c) \quad (14)$$

4.4 단말노드 클래스 부여

제안한 방법에 의해서 생성되는 퍼지 결정트리의 단말 노드는 서로 다른 소속정도로 복수 개의 클래스를 나타낼 수 있다. 제안한 퍼지 결정트리 생성 방법에서는 단말 노드가 나타내는 클래스를 다음과 같은 방법으로 결정한다. T_i를 단말 노드 N_i에 도달한 학습 데이타 집합이라고 하고, $\mu_{T_i}(x)$를 데이타 x가 노드 N_i에 속하는 정도라고 하자. T_i에 대해서 각 클래스 j에 대한 학습 데이타들의 소속정도를 다음과 같이 누적한다.

$$\sigma_i^j = \sum_{\substack{x\in T_i \\ x.class=j}} \mu_{T_i}(x) \quad (15)$$

단말 노드 N_i의 클래스 k에 대한 소속정도 $x_{N_i}(k)$를 다음과 같이 정규화된 가능성 분포로 설정한다. $CL(N_i)$는 단말 노드 N_i가 표현하는 클래스를 나타내는 퍼지집합이다.

$$CL(N_i) = \{(k, x_{N_i}(k)) \mid k\in C,\ x_{N_i}(k) = \sigma_i^k / \max_j \sigma_i^j\} \quad (16)$$

4.5 퍼지 결정트리 생성 절차

제안한 퍼지 결정트리 생성 절차를 정리하면 다음과 같다.

단계 1. 노드의 최대 분기 개수, 노드의 최대 허용 엔트로피를 설정하고, 근노드를 생성한다.

단계 2. 모든 학습 데이타를 근노드에 위치시키고, 각 데이타의 근노드에 대한 소속정도를 해당 데이타의 클래스에 대한 확신도로 설정한다.

단계 3. 현재 노드까지 사용되지 않은 속성들 중에서, 4.1절의 속성영역 분할방법에 따라 해당 속성의 영역을 최대 분기 개수 이내로 분할하면서, 4.2절에서 설명한 정보이득을 최대로 하는 분기 속성 및 속성 영역 분할 방법을 선택한다.

단계 4. 선택된 속성 및 영역분할 방법에 따라 자식노드들을 만들고, 링크에 속성 분할값(즉, 퍼지숫자 또는 속성값의 집합)을 부여한다. 4.3절의 링크 만족도 검사 방법을 이용하여, 링크에 부여된 속성 분할값에 대한 만족정도를 반영시켜 현재 노드의 각 데이타를 자식노드로 내려 보낸다.

단계 5. 새로 만들어진 노드들에 대해서, 근노드로부터 해당 노드까지 사용되지 않은 속성이 있으면서 현재 노드의 엔트로피가 허용 엔트로피보다 크다면, 단계 3-4를 반복 수행한다.

단계 6. 4.4절의 클래스 부여 방법을 이용하여 생성된 퍼지 결정트리의 단말 노드에 클래스를 부여한다.

4.6 클래스 결정을 위한 추론

결정트리는 결정트리 생성에 사용된 적이 없는 새로운 데이타에 대한 클래스를 결정하기 위해 사용된다. 제안한 방법에 의해 생성된 퍼지 결정트리에서는 새로운 데이타에 대한 클래스를 결정하기 위해 다음과 같은 추론방법을 사용한다. 분류될 데이타는 근노드에 초기 소속정도 1로 하여 주어지고, 자식노드로 링크를 따라 내려간다. 임의의 노드 N_i는 데이타 x를 받을 때 마다, 노드에서 나가는 링크에 부여된 링크값에 대한 속성값의 만족정도를 계산한다. 데이타 x의 속성값 v를 갖는 링크에 연결된 자식노드에 대한 소속정도는 $f(\mu_{N_i}(x), M(x.A, v))$에 의해 계산된다. 여기에서 $\mu_{N_i}(x)$는 데이타 x의 현재 노드 N_i에 대한 소속정도이고, $M(x.A, v)$는 현재 노드 N_i의 분류속성 A에 대한 데이타의 속성값 $x.A$의 링크값 v에 대한 만족정도이고, $f(\cdot,\cdot)$는 해당 응용분야에서 선택한 임의의 T-norm 연산자이다. 데이타의 자식노드에 대한 소속정도가 미리 지정된 임계값 이상이면, 데이타를 자식노드에 전달한다. 현재 노드가 나타내는 속성에 대해 데이타의 속성값이 없는 경우에는, 해당 속성값이 현재 노드의 모든 링크값에 완전히 만족되는 것으로 가정하여 데이타를 모든 자식노드에 전달한다. 퍼지 결정트리에서는 하나의 데이타가 여러 개의 단말 노드에 도달할 수 있다. 따라서 새로운 데이타에 대한 클래스를 부여하기 위해 다음과 같은 방법을 사용한다.

$$c_k(x) = \sum_i x_{N_i}(k) \cdot K_{N_i}(x) \quad (17)$$

$$cf_k(x) = \frac{c_k(x)}{\max_i c_i(x)} \quad (18)$$

여기에서 $x_{N_i}(k)$는 단말 노드 N_i가 클래스 k를 나타내는 정도이고, $K_{N_i}(x)$는 데이타 x의 단말 노드 N_i에 대한 소속정도이고, $cf_k(x)$는 데이타 x를 클래스 k로 분류하는 것에 대한 확신도이다.

5. 실 험

제안한 퍼지 결정트리 생성 방법의 유용성을 보이기

위해 몇가지 실험을 수행하였다. 실험은 수치 속성만을 갖는 퍼지 데이타와, 수치 속성과 비수치 속성을 함께 갖는 퍼지 데이타, 일반 데이타로서 대표적인 벤치마크 문제인 Iris 데이타 등 세가지 데이타에 대해서 수행하였다. 실험에서 퍼지 결정트리의 분류 정확도 *Correctness*는 분류결과 가장 큰 확신도를 갖는 클래스와 학습 데이타에 지정된 클래스가 일치하는 비율로 결정하였다.

$$Correctness = \frac{\sum_{C_i=k} 1}{\frac{cf_k(x_i)=\max_j(cf_j(x_i))}{|X|}, x_i \in X} \quad (19)$$

여기에서 C_i는 학습 데이타 x_i에 지정된 클래스를, k는 x_i에 분류결과 확신도가 가장 큰 클래스를 나타내고, X는 전체 학습데이타 집합을 나타낸다.

표 1 학습 데이타

ID	attribute 1	attribute 2	class
1	0	[0.0,0.5]	2
2	0	2	2
3	*Trap*(0.0,0.2,0.4,0.5)	4	1
4	*Trap*(1.2,1.5,2.0 ,2.7)	[1.0,2.0]	2
5	1	[1.7,2.3]	2
6	[1.0,1.2]	[3.0,3.3]	1
7	3	*Trap*(1.8,2.0,2.4,2.6)	4
8	4	2	4
9	5	[1.0,1.7]	3
10	*Trap*(4.8,4.9,5.3,5.4)	*Trap*(1.6,1.7,2.0,2.3)	4
11	5	[2.2,3.2]	2
12	[5.7,6.2]	*Trap*(0,0.2,0.5,0.6)	3
13	[6.0,6.6]	[1.8,2.3]	4
14	6	4	2
15	*Trap*(1.2,1.3,1.4,1.4)	*Trap*(1.9,2.0,2.1,2.2)	2
16	2	2	2

(표 1)은 실험에서 사용된 수치 속성만을 갖는 퍼지 데이타로서, 각 데이타의 클래스에 대한 소속정도는 1로 가정한 것이다. 표에서 $[a,b]$는 구간값을 나타내고, *Trap*(a,b,c,d)는 사다리꼴 퍼지숫자를 나타낸다. (그림 4)은 (표 1)의 데이타에 대해서 최대 분기 개수를 3으로 하여 제안한 퍼지 결정트리 생성 방법이 생성한 퍼지 결정트리를 보인 것이다. 결정트리에서 단말 노드에 할당된 퍼지집합은 근노드에서 해당 단말 노드 사이의 경로상의 소속함수들에 의해 표현된 모든 조건을 만족하는 데이타가 속하는 클래스를 나타낸 것이다. 예를 들면, (그림 4)의 퍼지 결정트리에서 가장 왼쪽의 단말 노드의 퍼지집합은 해당 노드에 도달하는 데이타가 확신도 1로 클래스 2에 속한다는 것을 나타낸다.

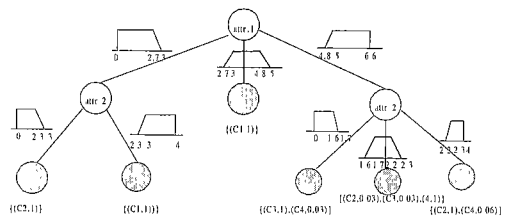


그림 4 (표1)의 퍼지 데이타에 대해 생성된 퍼지 결정트리

(그림 4)의 퍼지 결정트리는 (표 1)의 데이타를 100% 정확히 분류를 할 수 있는 트리이다. 다음은 생성된 퍼지 결정트리를 사용하여 데이타를 분류한 예를 보인 것이다.

데이타 : (4.9, 2.25) → 분류결과(클래스) : {(2,0.5), (3,0.2), (4,1.0)}

데이타 : (*Trap*(1.0,1.2,1.4,1.6), *Trap*(1.0,1.2,1.4,1.6)) → 분류결과(클래스) : {(2,1.0)}

데이타 : (*Trap*(4.2,4.6,5.3,6.0), *Trap*(3.0,3.2,4.4,4.6)) → 분류결과(클래스) : {(2,1.0), (4,0.73)}

데이타 : (Trap(3.6,3.9,4.2,6.0), Trap(2.0,2.2,3.6,3.8)) → 분류결과(클래스) : {(2,0.31), (4,1.0)}

위의 예에서 보이는 바와 같이 퍼지 결정트리는 분명한 속성값으로 기술된 데이타뿐만 아니라 퍼지숫자로 기술된 속성값을 갖는 데이타도 분류할 수 있다. 또한 분류결과로서 클래스별로 확신도를 제공함으로써 사용자가 최종 의사결정을 할 때 이들 부가 정보를 참고할 수 있도록 한다.

(표 2)는 보통값, 구간값, 퍼지숫자값을 갖는 두 개의 수치 속성과 보통값만을 갖는 하나의 비수치 속성값을 갖는 퍼지 데이타로서, 클래스 정보에 대한 확신도는 1로 가정한 것이다. (그림 5)는 (표 2)의 퍼지 데이타에 대해 제안된 방법을 이용하여 얻어진 퍼지 결정트리를 보인 것이다. 이 실험에서는 결정트리의 최대 분기 개수를 3으로 하였기 때문에, (그림 5)의 결정트리에서 4가지 속성값을 갖는 비수치 속성인 attribute-3도 영역 분할을 3개로 하고 있다. 비수치 속성 attribute-3에 대한 분기 링크에 부여된 속성값의 집합(예, {yellow, dark})은 데이타의 해당 속성값이 이 집합의 원소이면, 데이타가 링크를 따라 내려간다는 것을 나타낸다. (그림 5)에서 attribute-3 속성 영역이 {yellow, dark}, {blue, cyan}, {dark}로 분할되는 것처럼, 비수치 속성의 속성 영역도 배타적으로 분할되지 않고 중복되게 분할될 수 있다. 따라서 분류과정에서 동일한 데이타가 동시에 여러 개의 분기 링크를 따라 내려갈 수 있다.

표 2 학습 데이타

	attribute 1	attribute 2	attribute 3	class
1	2	[1, 1.2]	yellow	1
2	Trap(1, 1.1, 1.2, 1.2)	3	yellow	1
3	4	2	yellow	2
4	4.5	[2.8, 3.3]	yellow	2
5	[6, 6.3]	2	blue	3
6	7	Trap(1.1, 1.2, 1.2, 1.3)	blue	3
7	3	5	yellow	1
8	Trap(3.7, 3.9, 4.0, 4.1)	4.5	yellow	1
9	6	3.5	dark	1
10	8	[8.9, 4.1]	dark	1
11	[1.8, 2.1]	8	cyan	3
12	3.5	Trap(8.3, 8.5, 8.9, 9.0)	cyan	3
13	6	5	dark	1
14	7	[8.5, 9]	dark	1
15	Trap(7.8, 8.0, 8.1, 8.2)	7	dark	4
16	[8.6, 9.4]	9	dark	4

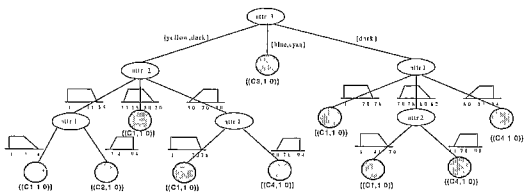


그림 5 (표 2)의 학습 데이타에 대한 퍼지 결정트리

제안한 퍼지 결정트리 생성 방법의 퍼지 데이타가 아닌 일반 데이타에 대한 적용가능성을 보이기 위해서, 분류규칙 마이닝을 위한 대표적인 벤치마크 문제인 Iris 데이타 집합에 대한 실험을 하였다. Iris 데이타 집합은 4개의 수치 속성값으로 기술된 150개의 붓꽃(iris) 데이타를 3개의 클래스로 분류해 놓은 것이다. 실험에서는 150개의 데이타를 75개씩 2개로 나누어, 하나는 결정트리 생성에 이용하고 다른 하나는 생성된 결정트리를 검증하는데 이용하는 실험을 한 다음, 다시 두 개의 역할을 바꾸어서 실험하는 2-fold cross validation을 20회 수행하였다. 실험결과 생성된 퍼지 결정트리는 학습 데이타에 대해서 평균적으로 97.5%의 정확도로 데이타를 분류하고, 검증 데이타에 대해서는 90.6%의 정확도로 데이타를 분류하였다. (그림 6)은 Iris 데이타에 대해서 제안된 방법에 의해 생성된 학습 데이타 및 검증 데이타에 대해서 100% 정확도로 분류하는 퍼지 결정트리를 보인 것이다. 그림에서 보는 바와 같이 퍼지 결정트리는 트리의 크기가 다소 크기는 하지만, 제안된 퍼지 결정트리 생성 방법이 퍼지 데이타뿐만 아니라 분명한 값만을 갖는 일반 데이타에 대해서도 적용될 수 있음을 알 수 있다.

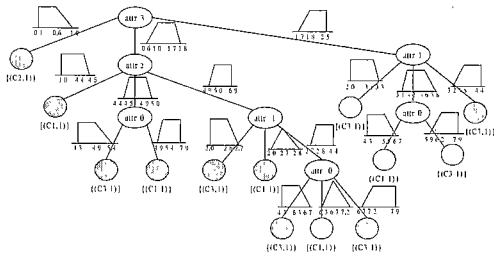


그림 6 Iris 데이타에 대한 퍼지 결정트리

6. 결 론

본 논문에서는 퍼지 학습 데이타로부터 분류규칙을 마이닝하기 위한 새로운 퍼지 결정트리 생성 방법을 제안하였다. 제안한 방법은 다음과 같은 특징을 갖는다. 첫째, 학습 데이타로 사용될 수 있는 데이타의 형태가 수치 속성값과 비수치 속성값으로 기술되면서, 수치 속성은 보통값 뿐만아니라 퍼지숫자, 구간값을 갖을 수 있고, 비수치 속성은 보통값을 가지며, 데이타의 클래스는 구간 [0,1] 사이의 확신도를 가지는 것이다. 둘째, 수치 속성 및 비수치 속성에 대한 영역분할이 결정트리 생성 과정에서 동적으로 이루어진다. 셋째, 생성되는 결정트리의 최대 분기 개수를 임의로 조정할 수 있다. 이를 위해 수치 속성 영역을 원하는 개수의 사다리꼴 퍼지숫자로 분할하고, 비수치 속성의 영역을 속성값의 집합으로 분할하는 방법을 도입하였다.

또한 생성된 결정트리를 사용하여 새로운 퍼지 데이타를 분류하는 추론 방법을 소개하였다. 제안한 방법에 의해서 생성되는 퍼지 결정트리는 주어진 데이타에 대한 클래스를 결정할 때 확신도값을 함께 제공하기 때문에 사용자가 최종 의사결정을 할 때 이러한 정보를 참고할 수 있다. 한편 몇 가지 실험을 통해 제안된 방법의 유용성을 보였다.

이 연구에서는 생성된 퍼지 결정트리의 가지치기(prunning)를 고려하지 않았으므로 앞으로 이에 대한 연구가 필요하다. 또한 비연속 수치 속성값에 퍼지집합이 허용되고, 비수치 속성값에 퍼지집합이 허용되는 데이타 집합에 대한 퍼지 결정트리 생성 방법에 대한 연구가 필요하다. 한편, 대규모 퍼지 데이타에 대해서 분류 규칙을 마이닝하기 위해 제안한 알고리즘을 적용하기 위해서는 샘플링 방법 등과 같은 속도 향상을 위한 개선 방법에 대한 연구가 앞으로 필요하다. 또한 제안한 퍼지 결정트리 생성 방법을 실제 현장에서 수집된 퍼지 데이타에 적용하여 성능을 평가하는 작업이 앞으로 필요하다.

참 고 문 헌

[1] T. M. Mitchell, *Machine Learning,* The McGraw-Hill Co.,414p, 1997.

[2] Y. Yuan, M. J. Shaw, Induction of fuzzy decision trees, *Fuzzy Sets and Systems,* Vol.69, pp.125-139, 1995.

[3] A. Ittner, J. Zeidler, R. Rossius, W. Dilger, M. Schlosser, Feature space partitioning by non-linear and fuzzy decision trees, *Proc. of the 7-th IFSA World Congress*(Prague, Czech), pp.394-398, 1997.

[4] J. Zeidler, M. Schlosser, Continuous-valued attributes in fuzzy decision trees, *Proc. of the 6-th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (Granada, Spain), pp.395-400, 1996.

[5] C. Z. Janikow, Fuzzy Decision Trees: Issues and Methods, *IEEE Trans. on Systems, Man, and Cybernetics - Part B,* Vol.28, No.1, pp.1-14, 1998.

[6] R.L.P. Chang, T. Pavlidis, Fuzzy Decision Tree Algorithms, *IEEE Trans. on Systems, Man, and Cybernetics,* Vol.7, No.1, pp.28-35, 1977.

[7] H.-J. Zimmermann, *Fuzzy Set Theory and its Applications,* Kluwer Academic Pub., 399p, 1991.

[8] D. Rasmussen, R. R. Yager, SummarySQL: A fuzzy tool for data mining, *Intelligent Data Analysis,* Vol.1, No.1, 1997.

[9] S. A. Maelainin, A. Bensaid, Fuzzy data mining query language, *Proc. of the Second Int. Conf. on Knowledge-Based Systems*(Adelaide, Australia), pp.335-340, 1998,

[10] K. Nozaki, H. Ishibuch, H. Tanaka, Adaptive Fuzzy Rule-based Classification Systems, *IEEE Trans. on Fuzzy Systems,* Vol.4, No.3, pp.238-250, 1996.

[11] T. Tani, M. Sakoda, Fuzzy modeling by ID3 algorithm and its application to prediction of heater outlet temperature, *Proc. of IEEE Int. Conf. on Fuzzy Systems* (San Diego), pp.923-930, 1992.

[12] R. Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, *Proc. of 2nd Int. Conf. on Fuzzy Logic and Neural Networks* (Iizuka), pp.265-268, 1992.

[13] J. Jang, Structure determination in fuzzy modeling: A fuzzy CART approach, *Proc. IEEE Conf. on Fuzzy Systems,* pp.480-485, 1994.

[14] M. W. Kim, J. G. Lee, C. Min, Efficient fuzzy rule generation based on fuzzy decision tree for data mining, *Proc. IEEE Int. Fuzzy Systems Conference Proceedings,* pp.1223-1228, 1999.

이 건 명

1990년 한국과학기술원 학사. 1992년 한국과학기술원 석사. 1995년 한국과학기술원 박사. 1995년 ~ 1996년 프랑스 INSA de Lyon 연구원. 1996년 미국 PSI사 연구원. 1996년 ~ 현재 충북대학교 컴퓨터과학과 조교수. 관심분야는 소프트컴퓨팅, 데이타마이닝, 에이전트 시스템 등.