Journal of the Korean Institute of Industrial Engineers
Vol. 33, No. 2, pp. 183-190, June 2007.

러프셋 이론과 개체 관계 비교를 통한 의사결정나무 구성

한상욱† · 김재련

한양대학교 산업공학과

A New Decision Tree Algorithm Based on Rough Set and Entity Relationship

Sang-Wook Han · Jae-Yearn Kim

Department of Industrial Engineering, Hanyang University, Seoul 133-791

We present a new decision tree classification algorithm using rough set theory that can induce classification rules, the construction of which is based on core attributes and relationship between objects. Although decision trees have been widely used in machine learning and artificial intelligence, little research has focused on improving classification quality. We propose a new decision tree construction algorithm that can be simplified and provides an improved classification quality. We also compare the new algorithm with the ID3 algorithm in terms of the number of rules.

Keywords: Decision rules, Attribute Core, Discernibility Matrix, Rough Set theory

1. 서 론

의사결정나무(decision tree)는 분류(classification) 분야에서 사용되는 기법으로 데이터 그룹에 속한 개체들의 패턴이나 특징 또는 룰을 추출해 낼 때 사용한다. 의사결정나무 구성 단계에 있어서 의사결정나무의 노드를 구성하는 속성을 선택하는 단계는 중요한 단계이고 속성의 선택에 따라서 룰이 많아질 수도 있고 최소화될 수도 있다. 의사결정 측면에서 볼 때 최소화된 룰을 가진 의사결정나무의 생성이 바람직하다. 앞선 분류 분야의 연구들에서 많이 활용된 의사결정나무 알고리듬인 ID3는 Quinlan(1986)에 의해 소개되었고 엔트로피를 이용하여 의사결정나무를 구성하는 방법을 채택하고 있다. Z. Pawlak (1982)에 의해 소개된 러프셋 이론(rough set theory)은 불확실하거나 모호한 데이터를 다루는 데 유용한 수학적인 이론이다 (Pawlak, 1991). 최근까지 machine learning, data mining, pattern recognition 등을 비롯한 여러 분야에 사용되어 왔다. 러프셋 이론을 응용한 다양한 data mining 기법들이 소개되었는데 이 중 코어(Core)와 리덕트(Reduct), 엔트로피(Entropy)를 이용하여 decision tree를 만드는 연구들이 Yang and Chiam(2000), Bai *et al.*(2003), Yang *et al.*(2003), Karno(2001)등에 의해서 소개가 되었다. 러프셋 이론은 룰 추출시 제거하더라도 원래의 전체 데이터에서 룰을 찾아낼 때와 같은 결과를 유도하는 속성을 제거 가능한(dispensable) 속성으로 보고 이를 제거하는 reduction 과정을 통해 데이터베이스에 있는 지식을 효과적으로 발견해 낼 수 있다. 러프셋 이론에서는 룰 추출시 제거 가능한 (dispensable) 속성과 분류에 있어서 필요한(indispensable) 속성을 구별해 내기 위해 식별가능 행렬(discernibility matrix)을 이용한다. 식별가능 행렬을 이용해서 분류에 필요한 핵심 속성인 코어와 분류에 필요한(indispensable) 속성들의 조합인 리덕트를 구할 수 있다.

앞서 소개한 ID3는 품질이 나쁘거나 빈약한 분류 지식을 가진 의사결정나무를 유도해 내는 등 때로는 불규칙한 작용을 보인다(Tu and Chung, 1992). 또한 ID3는 초기에 열등한 속성을 선택할 수 있을 수도 있고 새로 추가된 속성과 기존에 선택된 속성들의 관계를 고려하지 않고 단순히 단계별 최적 속성에 집중한다(Elashoff *et al.*, 1967; Toussaint, 1971). 본 논문에서는 러프셋 이론을 적용하여 의사결정나무의 루트 노드를 선택할 때 코어 속성과 본 논문에서 제안하는 분류 기여도를 이

† 연락저자 : 한상욱, 133-791 서울특별시 성동구 행당동 17번지 한양대학교 산업공학과, Fax : 02-2296-0471,
E-mail : softhan@hanyang.ac.kr
2006년 10월 접수; 2006년 12월 게재 확정.

용해 루트 노드를 선택하여 의사결정나무를 구성하는 방식을 소개한다. 본 논문에서 제안하는 아이디어는 개체들간의 비교와 속성간의 중요도를 고려하여 이 개체들을 가장 잘 분류해 낼 수 있는 속성을 선택하는 방식이므로ID3가 가질 수 있는 오류를 보완하는 의사결정나무 구성 방법이다

본 논문은 제 6장으로 구성되어있다. 제 2장에서 러프셋 이론의 기본 개념을 설명하고 제 3장에서는 의사결정나무를 만드는 기존 연구 중 엔트로피 방법과 러프셋 이론을 적용해 의사결정나무를 만드는 방법을 소개한다. 제 4장에서는 본 논문에서 제안하는 알고리듬을 소개하고 간단한 예제를 적용했고, 제 5장에서는 문제를 확장해 실험했다. 마지막으로 제 6장에서는 결론과 추후 연구 과제를 제시한다.

2. 러프셋 이론의 개념

러프셋 이론은 모호하고 불명확한 데이터에 대한 새로운 수학적 접근으로 집합 이론에 기반을 두며 응용분야에는 속성의 제거(reduction), 기존 의사결정 시스템에 추가되는 새로운 개체에 대한 클래스 예측 등이 있다. 이번 장에서는 본 논문과 관련된 러프셋의 용어와 개념에 대해 설명한다.

(1) 의사결정 시스템

의사결정 시스템은 러프셋 이론과 본 논문의 개념을 적용하는 시스템으로 개체의 전체 집합 $U=\{x_1, x_2,...,x_n\}$가 있고 조건속성들의 집합인 $C=\{c_1,...,c_n\}$와 결정속성집합 $D=\{d\}$가 있을 때 의사결정 시스템 $S=(U, C\cup D)$라고 정의한다.

(2) 러프셋의 식별 불구분 관계

개체들의 집합 U와 속성집합 C가 있고, V_c를 C의 속성값의 집합이라 하며 $A=C\cup D$, $C\cap D=\phi$이라 하자. 두 개의 개체 x, y가 $P\subseteq A$인 속성 집합 P에 대해 같은 속성값을 갖게 되면 x, y는 속성집합 P에 대해 식별 불구분 관계(indiscernibility relation) 또는 동치 관계(equivalence relation)라 하고 아래 식과 같이 표현할 수 있다.

$$\mathrm{IND}(P)=\{(x, y)\in U\times U: f(x, c)=f(y, c), \forall c\in P\} \quad (1)$$

(3) 러프셋의 근사화

개체들의 집합 U에서 클래스의 부분집합 X를 하나의 속성집합 B로 표현하면, 클래스를 명확하게 구분할 수 있는 개체들의 집합은 하한 근사 집합(lower approximation)이라 하고 이 영역을 긍정영역(positive region)이라 하며 $\underline{B}X$로 표기 한다. 또한 X와의 교집합이 공집합이 아닌 즉, X에 포함되는 개체와 속성값은 같은데 X^c에도 포함되는 동치 관계인 개체들은 상한 근사 집합(upper approximation)이라 하고 $\overline{B}X$로 표기 한다. 또한 상한 근사 집합에서 하한 근사 집합을 뺀 부분을 경계영역(boundary region)이라 한다. $\overline{B}X$와 교집합이 없는 영역은 부정 영역(negative region)이라 한다.

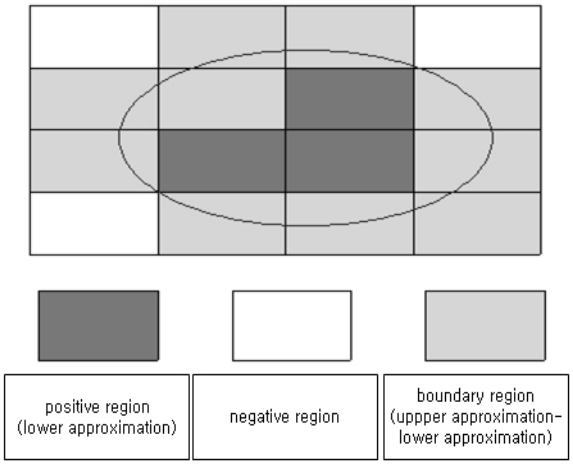


Figure 1. Approximation diagram

<Figure 1>은 타원으로 주어진 지식에 대한 긍정영역(positive region), 부정영역(negative region) 및 경계영역(boundary region)을 나타내고 있다.

$$B\text{-하한근사}(\underline{B}X)=\{x\in U|[x]_B\subseteq X\} \quad (2)$$

$$B\text{-상한근사}(\overline{B}X)=\{x\in U|[x]_B\sqcap X\neq\phi\}\ ; \quad (3)$$

하한근사+Boundary영역

$$\text{Boundary 영역 } BN_R(X)=(\overline{B}X)-(\underline{B}X) \quad (4)$$

(4) 리덕트(Reduct)

속성 집합 C의 하나의 원소 a가 $\mathrm{IND}(C)=\mathrm{IND}(C-\{a\})$를 만족할 경우, 속성 a는 C에서 불필요(dispensable)하고, 그렇지 않으면 a는 C에서 필요(indispensable)이다(Pawlak, 1991). 룰 추출에 있어서 불필요한 속성집합을 뺀 최소속성집합을 C' 라고 할 때 $C'\subset A$이고 $\mathrm{IND}(C)=\mathrm{IND}(C')$일 경우, C'를 C의 리덕트라고 한다.

(5) 코어(Core)

C에서 필요(indispensable)인 모든 속성들의 모임(리덕트들의 교집합)을 C의 코어라 하고, $\mathrm{core}(C)$라 표기한다. 즉, $\mathrm{core}(C)=\cap\mathrm{RED}(C)$이다. 예를 들어 <Table 1>에서의 코어는 리덕트 $\{(c_1, c_4)\}$와$\{(c_1, c_3)\}$의 교집합인 c_1이다. 코어는 리덕트들의 집합에서 교집합인 속성이므로 제거하면 원 데이터의 정보량이 소멸되는, classification에 있어서 없어져서는 안 되는 핵심 속성이다.

(6) 식별가능 행렬(discernibility matrix)

$U=\{x_1, x_2,...,x_n\}$에 대하여 속성집합 C의 부분집합 P에 대하여 각 원소사이의 속성값이 다른 속성들을 행렬 요소로 정의하는 $n\times n$ 행렬을 $M(S)$로 정의하고 다음과 같이 정의

할 수 있다.

$$M_{ij} = \{c \in C : c(x_i) \neq c(x_j)\} \text{ for } i,j = \{1, 2, \cdots, n\} \quad (5)$$

항목 M_{ij}는 x_i와 x_j를 구별할 수 있는 모든 속성들의 집합이다.

[예제 2.1] 식별가능 행렬의 정의에 따라<Table 1>로부터 개체 x_3과 x_4의 속성값이 다른 속성들의 집합인 항목c_{34}은 아래와 같이 구한다.

Table 1. Decision system for c_{34}

U	condition attribute				decision
	c_1	c_2	c_3	c_4	D
x_3	2	0	2	1	0
x_4	0	0	2	2	2

$c_1(x_3) \neq c_1(x_4)$, $c_2(x_3) = c_2(x_4)$, $c_3(x_3) = c_3(x_4)$, $c_4(x_3) \neq c_4(x_4)$ 이므로, $c_{34} = \{c_1, c_4\}$이다.

즉, x_3과 x_4를 구별해 주는 식별가능 속성 집합은$\{c_1, c_4\}$이다. 이와 같은 방법으로 모든 개체에 대해 식별 가능 속성 집합을 구한 것이 식별 가능 행렬이다.

식별가능 행렬에서는 대각선을 기준으로 $M_{ij} = M_{ji}$이므로 한쪽의 값은 생략한다. 또한 클래스를 구별하기 위한 룰을 추출해 내는 목적으로 동일 클래스내의 개체 비교는 하지 않는다.

3. 의사결정나무에 대한 기존연구

3.1 엔트로피 방법

의사결정나무를 만드는 여러 가지 방법 중 잘 알려진 방법은 ID3로 Quinlan(1986)에 의해 소개되었고 이 방법은 정보 엔트로피를 이용해 속성의 중요도를 평가하는 방법이다.ID3에 대한 의사결정나무 구성 방법을 살펴보면 다음과 같다.

m개의 속성들과 m차원 벡터 $x = (f_1, \cdots, f_m)$를 갖는 개체(object)가 있고 각각의 개체가2개의 의사결정속성값w_1 또는 w_2중 한 값에 속할 때 불확도(uncertainty)를 측정하기 위한 엔트로피 함수식은 다음과 같다.

$$H(x) = \sum_x [-P(w_1|x)\log_2 P(w_1|x) - P(w_2|x)\log_2 P(w_2|x)] \quad (6)$$

함수식 (1)에서 $P(w_i|x)$의 w_i는 x의 사후확률(posteriori probabilities)이고 m개의 속성들 은 엔트로피 함수를 평가하여 불확도가 가장 낮은 즉, 가장 작은 엔트로피 함수 값 $H(f_i)$를 갖는 하나의 속성이 선택된다. 속성 f_i가 선택되고 f_i가 가진 속성값에 따라 분지가 된다. 분지된 속성값마다 해당 후보 속성들에 대한 엔트로피 함수 값을 다시 구하고 그 중 가장 작은 엔트로피 함수 값 $H(f_j)$를 갖는 속성 f_j를 서브 노드(sub node)로 선택한다. 이 과정을 반복하여 더 이상 분지될 속성이 없고 의사결정 속성값으로 종결이 되면 의사결정나무에서 하나의 가지가 완성된다. <Table 2>에서 각 속성에 대한 엔트로피는 다음과 같이 구한다.

Table 2. Example table for decision system

attribute / object	A	B	C	D	d
1	1	2	2	1	1
2	1	2	3	2	1
3	1	2	2	3	1
4	2	2	2	1	1
5	2	3	2	2	2
6	1	3	2	1	1
7	1	2	3	1	2
8	2	3	1	2	1
9	1	2	2	2	1
10	1	1	3	2	1
11	2	1	2	2	2
12	1	1	2	3	1

(A, B, C, D are condition attribute, and d is a decision attribute)

$$H(A) = \frac{8}{12}\left(-\frac{7}{8}log_2\frac{7}{8} - \frac{1}{8}log_2\frac{1}{8}\right) + \frac{4}{12}\left(-\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4}\right) = 0.696\ bits$$

속성 A와 같은 방식으로 나머지 속성들에 대한 엔트로피를 구하면 다음과 같다.

$$H(B) = 0.784\ bits,\ H(C) = 0.771\ bits,\ H(D) = 0.729\ bits$$

그러므로, 엔트로피 값이 가장 작은 속성 A를 의사결정나무의 루트로 선택한다. 이 때, 속성 A가 가지는 속성값인 1과 2로 가지가 구성된다. 이러한 과정을 반복하여 생성한 의사결정나무는<Figure 2>와 같다.

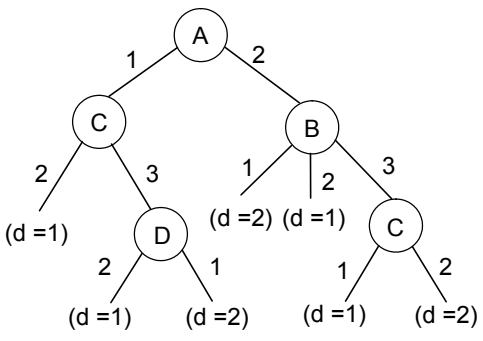


Figure 2. Decision tree for ID3

3.2 러프셋-근사화 방법

Wei *et al.*(2002) 등은 러프셋 이론의 하한근사와 상한근사를 적용하여 불확도가 낮은 속성을 선택하여 의사결정나무를 만드는 방법을 소개했다. <Table 2>에서 결정속성값 중 다수인 1에 대해서 각 속성의 하한근사와 상한근사를 구해보면 다음과 같다.

$\underline{A}(d=1)=\phi$, $\overline{A}(d=1)=\{1,2,\cdots,12\}$
$\underline{B}(d=1)=\phi$, $\overline{B}(d=1)=\{1,2,\cdots,12\}$
$\underline{C}(d=1)=\{8\}$, $\overline{C}(d=1)=\{1,2,\cdots,12\}$
$\underline{D}(d=1)=\{3,12\}$, $\overline{D}(d=1)=\{1,2,\cdots,12\}$

상한근사에서 하한근사를 뺀 경계영역이 가장 작은 D 속성을 의사결정나무의 루트로 선택한다. 즉 식 (4)의 모호한(rough) 부분인 경계영역이 가장 작은 속성을 루트로 선택하는 방법이고 이러한 방법을 더 이상 분지가 되지 않을 때까지 반복하여 <Table 2>에 적용해 구한 의사결정 나무는<figure 3>와 같다.

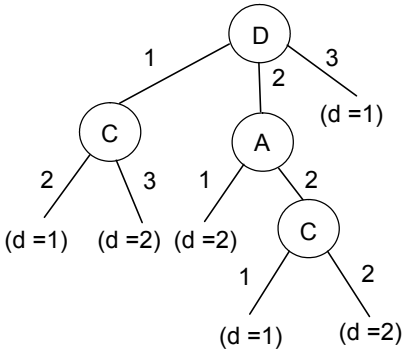


Figure 3. Decision tree for rough set-boundary region

3.3 러프셋-ID3 방법

제 2장에서 코어는 리덕트들의 집합에서 교집합인 속성이므로 제거하면 원 데이터의 정보량이 소멸되는classification에 있어서 없어서는 안 되는 룰을 추출할 때 핵심 속성이라고 언급했었다. Bai *et al.*(2003)등은 러프셋 이론의 코어와 엔트로피 방법 중 ID3를 혼합하여 의사결정 나무를 만드는 방법을 소개했다. 의사결정나무 구성 방법은 다음과 같다.

1 단계) 주어진 의사 결정 시스템에 코어가 있는지 확인한다. 이 때, 코어와 관련된 다음 3가지 경우가 발생할 수 있다.

a-1) 코어가 없는 경우
b-1) 코어가 한 개 있는 경우
c-1) 코어가 두 개 이상 있는 경우

2 단계) 루트 노드로 선택할 속성을 위의 3가지 경우 중 해당 경우에 대해 찾는다.

a-2) 코어가 없으므로 전체 속성에 대한 엔트로피 값을 계산하여 엔트로피 값이 가장 작은 속성을 루트 노드로 선택한다.
b-2) 코어가 한 개 이므로 코어인 속성을 루트 노드로 선택한다.
c-2) 코어 집합에 속하는 원소들인 속성들에 대해 엔트로피 값을 계산하여 값이 가장 작은 속성을 루트 노드로 선택한다.

3 단계) 2단계에서 루트 노드로서의 속성이 결정되면 선택된 속성의 값으로 분지하고 각 속성값에 대해1단계부터 반복한다.

4 단계) 더 이상 분지할 수 없고 결정속성(decision attribute)이 결정되면 분지를 종료한다.

이 방법을 통해 <Table 2>에 적용해 구한 의사결정 나무는 <figure 4>와 같다.

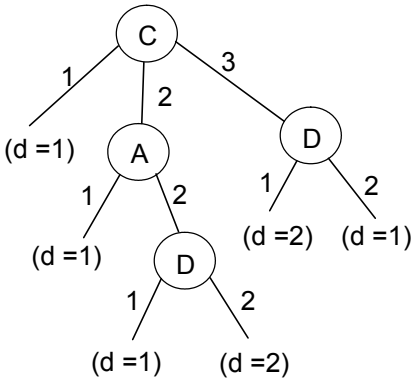


Figure 4. Decision tree for rough set-ID3

4. 제안하는 알고리듬

4.1 알고리듬 소개

ID3 알고리듬은 속성의 엔트로피를 계산해서 의사결정 나무를 확장해 가는 방식으로 속성간의 관계나 개체간의 관계를 고려하지 않는데 반해(Tu and Chung, 1992), 본 논문에서는 러프셋 이론을 적용하고 오브젝트간의 관계와 속성간의 분류 기여도를 고려한 새로운 의사결정나무 생성 방법을 제안한다. <Table 3>은 클래스가 있는 두 개체들을 비교할 때 발생하는4가지 경우를 보여준다. 즉, <case 1>은 두 개체의 조건속성값(condition attribute value)이 같고 결정속성값(decision attribute)이 같은 경우이고, <case 2>는 두 개체의 조건속성값이 같고 결정속성값이 다른 경우이고, <case 3>은 두 개체의 조건속성값이 다르고 결정속성값이 같은 경우이다. 또한, <case 4>는 두 개체의 조건속성값이 다르고 결정속성값도 다른 경우이다. 분류(Classification)기법에서 서로 다른 클래스를 잘 구별해 줄 수 있는 속성이 우선 선택되는 것이 바람직하므로 이 네 가지 경우 중<Table 3>의 <case 4>와 같이 결정속성이 다르고 조건속성값이 다른 속성은 긍정적(positive)으로 볼 수 있다. 또한 <Table 5>의 <case 3>과 같이 결정속성값이 같고 조건속성값이

Table 3. Comparison of two objects

object	case	condition attribute	decision attribute	result
comparison of x_1 and x_2	1	same	same	positive
	2	same	different	negative
	3	different	same	negative
	4	different	different	positive

Table 4. Example table for positive case

No.	A	d	
1	1	1	rule : if A=1 then d =1, support = 5/5 = 1
2	1	1	
3	1	1	
4	1	1	
5	1	1	

Table 5. Example table for negative case than case in Table 6

No.	A	d	
1	1	1	rule : ① if A= 1 then d = 1, support = 2/5, ② if A= 2 then d =1, support = 3/5
2	1	1	
3	2	1	
4	2	1	
5	2	1	

다른 경우는 측정된 개체 수n에 대한 지지도를 고려한 확률적인 측면에서와 룰 수를 증가시키는 면에서 <Table 4>와 같은 <case 4>의 경우보다 부정적(negative)으로 볼 수 있다.

<case 3>과 <case 4>를 클래스가 같을 때와 다를 때의 대표 case로 적용한 결과가 전체 case를 고려한 결과와 동일하므로 본 연구에서는 <Table 3>의 네 가지 case를 모두 적용하지 않고 <case 3>과 <case 4>만 고려한다. 제안하는 방법의 분류 기여도 계산 방법은 다음과 같다. U = $\{x_1, x_2, \cdots, x_n\}$인 지식표현 시스템 $S = (U, A)$에서 $(c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\}$ for $i,j = \{1, 2, \cdots, n\}$인 항목 $c_{i,j}$는 x_i와 x_j를 구별할 수 있는 모든 조건 속성들의 집합이고, 이 때 결정속성 d는 $\{d(x_i) \neq d(x_j)\}$ 또는 $\{d(x_i) = d(x_j)\}$인 경우로 나뉜다. 1부터 k까지 k개의 조건속성이 있고 조건속성 a_k의 분류 기여도 CC(Classification Contribution)는 다음과 같이 구한다. 여기서 CC_p는 긍정적인 경우, CC_n은 부정적인 경우, CC_t는 CC_p와 CC_n의 합인 전체 분류기여도를 의미한다.

① 긍정적인 경우(CC_p)

결정속성 d가 $\{d(x_i) \neq d(x_j)\}$이면서

$(c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\}$인 경우

조건속성 a_k의 긍정적 기여도 $CC_p(a_k) = \sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})}$,

단, $c_{ij} \cap a_k = \phi$ 이면 0 (7)

② 부정적인 경우(CC_n)

결정속성 d가 $\{d(x_i) = d(x_j)\}$이면서

$(c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\}$인 경우

조건속성 a_k의 부정적 기여도 $CC_n(a_k) = -\sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})}$,

단, $c_{ij} \cap a_k = \phi$ 이면 0 (8)

③ 전체 분류 기여도(CC_t)

위 두 가지 경우의 합을 조건속성 a_k가 갖는 분류 기여도라 할 때 다음과 같다.

$$CC_t(a_k) = [\sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})} | d(x_i) \neq d(x_j)] - [\sum_{i,j=1}^{n} \frac{I(c_{ij} \cap a_k)}{n(c_{ij})} | d(x_i) = d(x_j)] \quad (9)$$

(단 I는 0 또는 1의 값을 갖는 index function)

식 (9)에서 나온 결과 값이 가장 큰 조건속성을 분류에 있어서 기여도가 큰 속성으로 판정하고 선택한다. 앞에서 살펴본 개념을 바탕으로 제안하는 의사결정나무 방법은 다음과 같은 절차를 통해 의사결정나무를 전개해 나간다.

1) 노드로서 사용될 속성의 선택은 다음 조건을 만족하는 경우의 순서로 채택한다.
   a) 식별가능 행렬을 구하여 코어(core) 속성이 단일로 존재하면 이 코어 속성을 노드로 선택한다.
   b) 식별가능 행렬에서 코어가 복수이면 코어 속성 중 식(8)에서 살펴본 분류 기여도를 계산하여 분류기여도가 큰 속성을 선택한다.
   c) 코어가 없으면 식별가능 행렬에 원소로 나와 있는 속성들에 대한 분류 기여도를 계산하여 분류기여도가 큰 속성을 선택한다.
2) 의사결정나무의 가지는 선택된 속성이 가지는 속성값들로 구성된다.
3) 노드를 선택하고 가지로 전개해 나가다가 더 이상 분지될 가지와 노드가 없는 경우, 즉 결정속성이 정해지면 그 가지에 대한 전개는 종료된다.

4.2 예제 분석

<Table 2>에 대해 제안하는 방법으로 각 속성의 분류 기여도를 구하기 위해 <Table 6>과 같은 discernibility matrix를 이용한다.

Table 6. Discernibility matrix for Table 2

	1	2	3	4	5	6	7	8	9	10	11	12
1												
2	C, D											
3	D	C, D										
4	A	A, C, D	A, D									
5	A, B, D, d	A, B, C, d	A, B, D, d	B, D, d								
6	B	B, C, D	B, D	A, B	A, D, d							
7	C, d	D, d	C, D, d	A, C, d	A, B, C, D	B, C, d						
8	A, B, C, D	A, B, C	A, B, C, D	B, C, D	C, d	A, C, D	A, B, C, D, d					
9	D	C	D	A, D	A, B, d	B, D	C, D, d	A, B, C				
10	B, C, D	B	B, C, D	A, B, C, D	A, B, C, d	B, C, D	B, D, d	A, B, C	A, C			
11	A, B, D, d	A, B, C, d	A, B, D, d	B, D, d	B	A, B, D, d	A, B, C, D	B, C, d	A, B, d	A, C, d		
12	B, D	B, C, D	B	A, B, D	A, B, D, d	B, D	B, C, D, d	A, B, C, D	B, D	C, D	A, D, d	

<Table 6>에서 코어인 속성은 C와 D이다. 속성 C와 D의 분류 기여도는 다음과 같다.

$$CC_p(C) = (\frac{1}{1}+\frac{1}{3}+\frac{1}{3}+\frac{1}{2}+\frac{1}{2}+\frac{1}{1}+\frac{1}{3}+\frac{1}{2}+\frac{1}{4}+\frac{1}{2}+\frac{1}{3}+\frac{1}{2}+\frac{1}{2}) = 6.58$$

$$CC_n(C) = (\frac{1}{2}+\frac{1}{4}+\frac{1}{3}+\frac{1}{2}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{1}+\frac{1}{3}+\frac{1}{4}+\frac{1}{3}+\frac{1}{3}+\frac{1}{4}+\frac{1}{4}+\frac{1}{3}+\frac{1}{3}+\frac{1}{4}+\frac{1}{3}+\frac{1}{3}+\frac{1}{4}+\frac{1}{2}+\frac{1}{2}) = -8.17$$

$$CC_t(C) = 6.58\text{-}8.17 = -1.59$$

마찬가지로 속성 D에 대한 분류 기여도를 계산해 보면 다음과 같다.

$$CC_p(D) = 0.78$$
$$CC_n(D) = 12.5$$
$$CC_t(D) = 7.08\text{-}12.5 = -5.42$$

위에서 $CC_t(C) = -1.59 > CC_t(D) = -5.42$이므로 속성 C를 노드로 선택한다. 다음 수준은 C의 속성값들로 분지를 해 나간다. 즉 $C=1$일 때 $C=2$일 때 $C=3$일 때로 분지해 나간다. 이와 같은 방법으로 의사결정나무를 구해 보면<Figure 5>와 같은 의사결정나무가 구해진다.

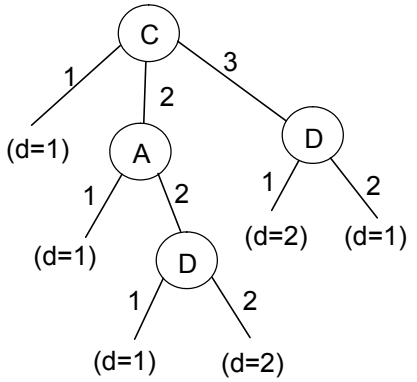
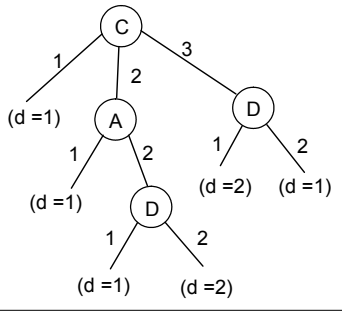
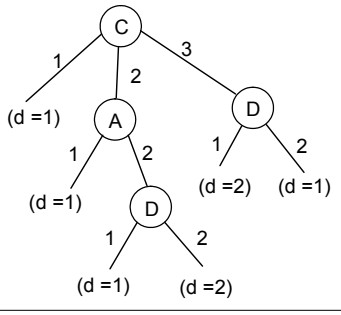


Figure 5. Decision tree for rough set based method

<Table 7>은 본 논문에서 제안하는 알고리듬으로 <table 2>에 대해 룰을 도출한 결과이다. 총 6가지의 룰이 나왔음을 볼 수 있다.

Table 7. Rules for rough set based method

The Solution is below								
Row	0	C	1	-	-	-	-	Decision = 1
Row	1	C	2	A	1	-	-	Decision = 1
Row	2	-	-	A	2	D	1	Decision = 1
Row	3	-	-	-	-	D	2	Decision = 2
Row	4	C	3	D	1	-	-	Decision = 2
Row	5	-	-	D	2	-	-	Decision = 1

ID3	Rough set-approximation
	

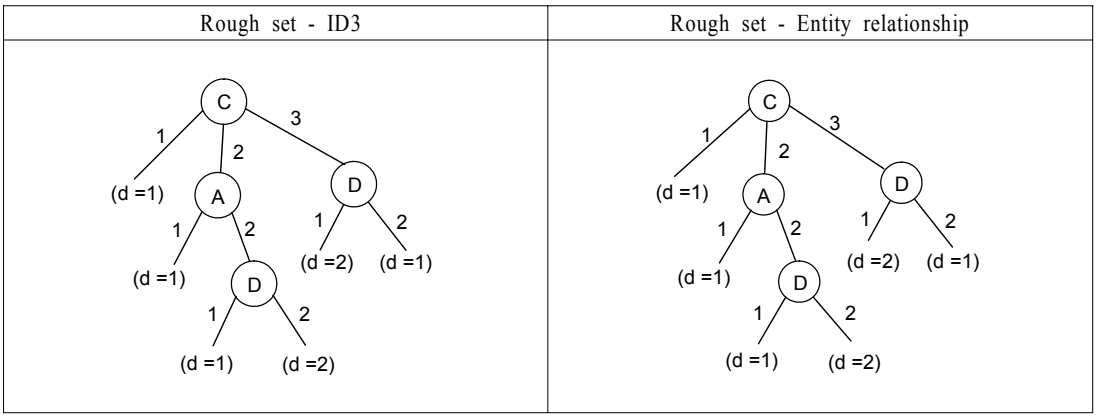


Figure 6. Decision tree for comparison of 4 algorithm

앞서 살펴본 ID3, 러프셋-Boundary region, 러프셋-근사화 방법과 제안하는 의사결정나무(Rough set-Entity relationship tree)를 비교해 보면 <Figure 6>과 같다.

위의 예제 분석 결과 제안하는 알고리듬의 룰 수가ID3의 룰 수보다 적음을 알 수 있다.

5. 실험 결과 분석

러프셋-이웃 탐색법과 엔트로피 기반의 의사결정나무를 비교하기 위해 UCI machine learning database에서 9개의 data set으로 본 논문에서 제안하는 알고리듬과ID3를 비교해 보았다. 실험결과 <Table 8>과 같은 결과를 얻었다. <Figure 7>과 <Figure 8>은 <Table 8>에 대한 요약 도표이다.

Table 8. Comparison of Rough set based method and ID

Data set	Tuples	Attri-bute (C/D)	Number of rules		Accuracy		Number of Cores
			ID3	New method	ID3	New method	
Table3	12	3/2	7	6	42	100	2
Weather	14	3/2	5	5	86	100	2
Contact-lenses	24	3/3	9	9	71	100	4
lymph	148	8/4	55	57	69	100	2
hepatitis	150	2/2	42	37	75	84	11
Breast-cancer	278	6/2	116	88	68	67	7
Heart-c	301	4/2	83	57	78	61	7
Vote	319	2/2	27	24	94	97	7
Primary-tumor	339	3/21	167	135	34	59	15
Credit-a	499	9/2	76	58	85	52	9

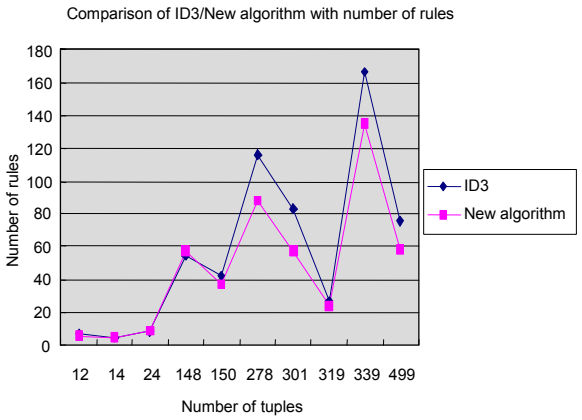


Figure 7. Comparison of ID3/New algorithm with number of rules

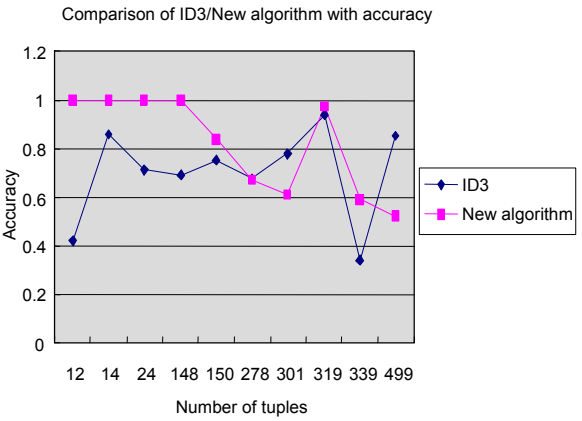


Figure 8. Comparison of ID3/New algorithm with number of rules

<Figure 7>에서 ID3에 비해 제안하는 알고리듬의 룰 수가 작은 것을 알 수 있다.

<Figure 8>에서 ID3에 비해 제안하는 알고리듬의 정확도가 비교적 높음을 볼 수 있다.

6. 결론 및 추후 연구과제

본 연구에서는 러프셋 개념의 적용 및 개체 비교를 통한 의사결정나무 방법을 제시하였다. 제안하는 방법은 엔트로피 기반의 의사결정나무 방법에 비해 간결한 룰을 도출할 수 있었다. 기존의 많은 분류(classification) 방법들이 분류 전에 데이터 마사지(data massage)와 같은 전처리 작업을 해야 되는 번거로움이 있는데 반해 본 논문에서 제안하는 방법은 전처리 작업이 필요 없고 categorical한 data에도 전처리 과정 없이 사용될 수 있다. 또한 개체들간의 관계를 비교해 노드를 선택하기 때문에 ID3보다 data의 의미론 적인 면에서 볼 때 긍정적인 분류 결과를 도출할 수 있다. 향후 연구과제로는 응용 사례에 적용하여 그 효과를 비교해 보고 시간 개념이 추가된 개체들의 의사결정나무 구축 방법을 연구하고자 한다.

참고문헌

Bai, J., Fan, B., and Xue, J. (2003), Knowloedge representation and acquisition approach based on decision tree, *Proc. 2003 Int. Conf. on Natural Language Processing and Knowledge Engineering*, 533-538.

Cover, T. M. (1974), The best two independent measurements are not the two best, *IEEE Transactions on Systems, Man, and Cybernetics*, **4**, 116-117.

Cover, T. M. and Van Campenhout, J. M. (1977), On the possible orderings in the measurement selection problem, *IEEE Trans. on Systems, Man, and Cybernetics*, **7**, 657-661.

Elashoff, J. D., Elashoff, R. M., and Goldman, G. E. (1967), On the choice of variables in classification problems with dichotomous variables, *Biometrika*, **54**, 668-670.

Karno, B. (2001), Core searching on rough sets, 2*3rd Int. Conf. on Technology ITI,* 19-22.

Pawlak, Z. (1982), Rough sets, *International Journal of Computer and Information Science,* **11**(5), 341-356.

Pawlak, Z. (1991), *Rough sets,* Kluwer academic publishers, 33-35.

Quinlan, J. R. (1986), Induction of decision trees, *Machine Learning I*, 81-106.

Quinlan, J. R. (1990), Decision trees and decision making, *IEEE Transactions on Systems, Man and Cybernetics,* **20**, 339-346.

Toussaint, G. T. (1971), Note on Optimal Selection of Independent Binary-Valued Features for Pattern Recognition, *IEEE Transactions on Information Theory*, **IT-17**, 618.

Tu, P.-L. and Chung, J.-Y. (1992), A new decision tree classification algorithm for machine learning, *Proc. 4th Int. Conf. on Tools with Artificial Intelligence(TAI '92),* 370-377.

Wei, J., Huang, D., Wang, S., and Ma, Z. (2002), Rough set Based Decision tree, *Proc. 4th World Congress on Intelligent Control and Automation.* 426-431.

Yang, J., Wang, H., and Hu, X. G. (2003), A new classification algorithm based on rough set and entropy, *Int. Conf. Machine Learning and Cybernetics*, 364-367.

Yang, Y. and Chiam, T. C. (2000), Rule discovery based on rough set theory, *Proc. 3rd Int. Conf. on Information Fusion*, **1**, TUC4/11-TUC4/16.