한국퍼지 및 지능 시스템학회 논문지 1998, Vol. 8, No. 6. (1998) 27~36

지식 발견을 위한 라프셋 중심의 통합 방법 연구

Integrated Method Based on Rough Sets for Knowledge Discovery

정 홍 · 정환묵*

Hong Chung and Hwan Mook Chung*

계명대학교 컴퓨터전자공학부, *대구효성가톨릭대학교 전자정보공학부

요 약

본 논문은 대규모 데이터베이스에서 유용한 지식을 발견하기 위해 라프셋을 중심으로 한 통합적 방법을 제시한다. 본 방법에서는 데이터베이스에 있는 실제 데이터에서 일반화된 데이터를 추출하기 위해 속성중심의 개념계층 상승기법을 사용하고, 획득 정보량을 측정하기 위해 결정 트리에 의한 귀납법을 사용한다. 그리고 불필요한 속성 및 속성값을 제거하기 위해 라프셋 이론의 지식감축 방법을 적용한다. 통합 알고리즘은 먼저, 개념의 일반화에 의해 데이터베이스의 크기를 줄이고, 다음으로 결정속성에 영향을 적게 미치는 조건속성을 제거함으로써 속성의 수를 줄인다. 마지막으로 속성간의 종속관계를 분석함으로써 불필요한 속성값을 제거하여 간략화된 결정규칙을 유도한다.

ABSTRACT

This paper suggests an integrated method based on rough sets for discovering useful knowledge from a large database. Our approach applies attribute-oriented concept hierarchy ascension technique to extract generalized data from actual data in databases, induction of decision trees to measure the information gain, and knowledge reduction method of rough set theory to remove superfluous attributes and attribute values. The integrated algorithm first reduces the size of database through the concept generalization, reduces the number of attributes by means of eliminating condition attributes which have little influence on decision attribute, and finally induces simplified decision rules by removing the superfluous attribute values by analyzing the dependency relationships among the attributes.

1. 서 론

최근 대규모의 데이터베이스로부터 유용한 정보를 추출하고자 하는 연구가 많이 진행되고 있다. 이를 KDD(Knowledge Discovery in Databases), Data Mining, 패턴 처리, 혹은 정보 획득이라고 하는데, 이들은 데이터로부터 함축적이고 유용한 정보를 추출한다는 공통적인 의미를 갖고 있다[2]. KDD에 관한 이론은 데이터베이스에서의 지식발견과 관련된 최근의 여러 워크샵[7,8,12,13]과 논문지[1,9,10,14]에서 보듯이 기계 학습, 지능 데이터베이스, 지식 획득 등에 있어서 학자들의 관심을 끌고 있다.

데이터베이스로부터 지식을 발견하는 방법에는 여러 가지 기법들이 많이 있으나, Han[3]의 속성중심(Attribute-Oriented) 귀납법, Quinlan[11]의 결정트리(Decision Tree)에 의한 귀납법, Pawlak[5]의 라프셋(Rough Set)에 의한 지식감축 방법 등이 많이 거론되고 있다. 그런데 속성중심 귀납법은 속성간에 데이터 종속관계를 분석하지 않아 생성된 규칙이 약간의 중복 정보와 불필요한 제약을 포함하고 있어 간략성과 강력성이 없다[4]. 따라서 규칙을 일반화하기 전에 일반화된 데이터의 특성을 심도있게 분석할 필요가 있다. 그리고 결정트리에 의한 귀납법은 속성에 의한 종속관계를 트리라는 그래프로 나타냄으로써 간단하기는 하나 속성과 튜플의 수가 많은 대규모 데이터베이스에는 부적합하다. 또 라프셋 이론에 의한 방법은 속성집합을 전반적으로 분석할 수 있는 도구를 제공하나 NP-hard적인 계산 복잡성 때문에 직접 대형 데이터베이스에 적용하기 어렵다[4]. 따라서 라프셋 이론을 적용하기 위해서는 다른 기법들을 사용하여 튜플과 속성의 수를 줄여야 한다.

데이터베이스에서의 지식 발견은 데이터베이스로부터 관심있는 지식을 발견하고 고수준 언어로 지식을 표현하는 학습 형태이다. 대부분의 학습 알고리즘은 매우 큰 데이터집합을 효율적으로 다루지 못하기 때문에 학습이 수행되는 데이터의 크기를 축소할 필요가 있다.

본 논문에서는 데이터베이스의 일반화를 위한 속

성중심 귀납법에서 사용하는 개념계층 상승기법, 결정트리에 의한 귀납법에서 사용하는 획득 정보량의 측정에 의한 속성 감축, 그리고 라프셋에 의한 지식감축 방법을 복합하여 저수준의 데이터를 고수준 정보로 일반화하고, 불필요한 속성 및 속성값들을 최대한 감축하여 간략화된 결정규칙을 도출하는 통합방법을 제시하고자 한다.

본 논문에서 제시하는 방법의 적용에 있어서는 다음과 같은 전제를 필요로 한다.

- 데이터베이스는 대량의 비교적 신뢰할 만한 데이터를 저장하고 있다.
- 지식발견 과정에서 개념 계층이라는 배경지식이 사용된다.
- 지식발견 과정은 사용자의 학습 요구수준에 따라 약간씩 달라질 수 있다.

2. 데이터베이스의 일반화

대규모 데이터베이스는 보통 속성값의 거대한 집합을 포함하고 있다. 이를 간략화하여 각 클래스에 대한 결정규칙을 유도하기 위해서는 기초 데이터 인스턴스를 고수준으로 일반화해야 한다. 이 작업은 업무에 적합한 관계에 대한 속성중심 일반화에 의해 실현된다.

개념계층은 데이터베이스의 속성에 있어서 일반화 관계의 집합이다[3]. 일반화 관계는 속성값의 전체집합과 고수준으로 일반화된 단일값간의 관계이다. 일반화 관계는 $\{A_1, A_2, \cdots, A_k\} \subset B$로 표현되는데, 이때 B는 각 $A_i(1 \le i \le k)$의 일반화이다. 데이터베이스에 대한 모든 개념계층은 영역 전문가가 작성하고 개념계층 테이블에 저장한다.

표 1.

No	Model	Size	Type	Trans-mitter	Year	Odo-meter
1	Sonata	Medium	DOHC	Auto	90	156000
2	Lanos	Small	DOHC	Auto	94	100000
3	Concord	Medium	SOHC	Manual	88	256000
4	Grandeur	Large	SOHC	Auto	88	200000
5	Sonata	Medium	SOHC	Manual	89	163000
6	Leganza	Medium	DOHC	Auto	95	111000
7	Credos	Medium	SOHC	Manual	89	180000
8	Avante	Small	DOHC	Auto	92	120000
9	Prince	Medium	SOHC	Auto	90	85000
10	Concord	Medium	SOHC	Manual	90	130000
11	Nubira	Small	DOHC	Auto	94	89000
12	Pride	Small	DOHC	Manual	91	175000
13	Sonata	Medium	DOHC	Auto	90	160000
14	Potentia	Large	SOHC	Manual	90	211000
15	Credos	Medium	SOHC	Auto	93	195000
16	Avante	Small	DOHC	Auto	93	175000

예를 들어 표 1과 같은 중고자동차 데이터베이스가 있다고 하자.

이 저수준의 데이터베이스를 고수준의 데이터베이스로 일반화하기 위해 사용할 자동차 관계의 개념 계층을 다음과 같이 정한다.

Model:
{Sonata, Grandeur, Avante}⊂Hyundai
{Pride, Concord, Potentia, Credos}⊂Kia
{Leganza, Lanos, Prince, Nubira}⊂Daewoo
{Hyundai, Kia, Daewoo}⊂Car_maker
Year
{..90}⊂Old
{91..93}⊂Medium
{94..}⊂New
Odometer
{..120000}⊂Short
{121001..190000}⊂Medium
{191001..}⊂Long

일반화는 각 속성에 대한 정의역의 집합이 있고 고수준의 개념계층이 있으면 개념계층을 상승(각 튜플의 속성값에 대응하는 고수준 개념으로 대치)시킴으로써 수행된다. 관계의 키(key)가 되는 속성은 개념계층에서 그와 같은 속성에 제공된 고수준의 개념이 없으므로 일반화가 될 수 없다.

표 1에서 Model은 메이커로 일반화시키고, Year, Odometer는 등급으로 일반화시키며, Size, Type, Transmitter는 개념계층에 없으므로 그대로 둔다. 여기서 어느 수준가지 일반화시킬 것인가는 응용별 개념계층에 따라 달라진다. 고수준으로 일반화시킨 데이터베이스는 표 2와 같다.

편의상 각 속성의 값을 다음과 같은 숫자로 표기한다.

Model={Kia, Hyundai, Daewoo}={1, 2, 3}
Size={Large, Medium, Small}={1, 2, 3}
Type=(SOHC, DOHC}={1, 2}
Transmitter={Auto, Manual}={1, 2}
Year={Old, Medium, New}={1, 2, 3}
Odometer={Long, Medium, Short}={1, 2, 3}

따라서 일반화된 데이터베이스를 표 3과 같은 지식

표 2.

No	Model	Size	Type	Trans-mitter	Year	Odo-meter
1	Hyundai	Medium	DOHC	Auto	Old	Medium
2	Daewoo	Small	DOHC	Auto	New	Short
3	Kia	Medium	SOHC	Manual	Old	Long
4	Hyundai	Large	SOHC	Auto	Old	Long
5	Hyundai	Medium	SOHC	Manual	Old	Medium
6	Daewoo	Medium	DOHC	Auto	New	Short
7	Kia	Medium	SOHC	Manual	Old	Medium
8	Hyundai	Small	DOHC	Auto	Medium	Short
9	Daewoo	Medium	SOHC	Auto	Old	Short
10	Kia	Medium	SOHC	Manual	Old	Medium
11	Daewoo	Small	DOHC	Auto	New	Short
12	Kia	Small	DOHC	Manual	Medium	Medium
13	Hyundai	Medium	DOHC	Auto	Old	Medium
14	Kia	Large	SOHC	Manual	Old	Long
15	Kia	Medium	SOHC	Auto	Medium	Long
16	Hyundai	Small	DOHC	Auto	Medium	Medium

표현 형식의 사례 테이블로 표시할 수 있다. 여기서 규칙 생성을 위해 Model, Size, Type, Transmitter, Year(*a*, *b*, *c*, *d*, *e*로 표기)를 조건속성, Odometer(*f*로 표기)는 결정속성으로 한다. 그리고 앞으로 튜플은 사례라는 용어로 표기할 것이다.

저수준의 데이터를 일반화하기 위해 개념계층을 상승시키면 중복 사례가 발생하거나 일관성이 없는

표 3.

	a	*b*	*c*	*d*	*e*	*f*
1	2	2	2	1	1	2
2	3	3	2	1	3	3
3	1	2	1	2	1	1
4	2	1	1	1	1	1
5	2	2	1	2	1	2
6	3	2	2	1	3	3
7	1	2	1	2	1	2
8	2	3	2	1	2	3
9	3	2	1	1	1	3
10	1	2	1	2	1	2
11	3	3	2	1	3	3
12	1	3	2	2	2	2
13	2	2	2	1	1	2
14	1	1	1	2	1	1
15	1	2	1	1	2	1
16	2	3	2	1	2	2

사례가 발생할 수가 있다. 이를 라프셋 이론을 적용하여 분석한다.

라프셋은 1982년 Pawlak[6]에 의해 제안되었는데, 이는 부정확하고 불완전한 정보를 분류하는 분제와 데이터베이스에서 원인, 결과를 데이터베이스 학습 형태로 인식하고자 하는 것이다.

U를 전체집합, R을 U에 있는 동치관계라 할 때, $A=(U, R)$을 근사공간이라 한다. $x,y\in U$, $(x,y)\in R$일 때 x와 y를 A에서 불분간이라고 한다. 관계 R에 있는 각 동치 클래스를 A에서 기본집합이라 하며, A에 있는 기본집합의 유한 합집합을 A의 복합집합이라 한다.

X를 U의 부분집합이라 할 때 A에서 X를 포함한 최소 복합집합을 A에서 X의 상한근사라 하며 R_UX로 표기하고, A에서 X를 포함한 최대 복합집합을 A에서 X의 하한근사라 하며 R_LX로 표기하고, 다음과 같이 정의한다.

A에서 X의 하한근사: $R_LX=\{x\in U|[x]_R\subseteq X\}$
A에서 X의 상한근사: $R_UX=\{x\in U|[x]_R\cap X\neq\phi\}$

여기서 $[x]_R$은 U의 원소 x에 대해 X를 포함하는 R의 동치 클래스이다. R_LX는 X에 포함된 모든 기본집합의 합집합이며, R_UX는 X와 non-empty 교집합인 기본집합의 합집합이다. 다시 말하면, $x\in R_LX$는 X에 확실히 포함되는 것이고, $x\in R_UX$는 X에 포함될 가능성이 있는 것이다. 예를 들어, 동치관계 $R=\{\{x_2, x_4, x_5, x_8\}, \{x_1, x_3\}, \{x_6, x_7, x_9\}\}$가 있을 때, $Y=\{x_1, x_3, x_5\}$에 대한 하한근사는 $R_LY=\{x_1, x_3\}$이며, 상한근사는 $R_UY=\{x_1, x_2, x_3, x_4, x_5, x_8\}$이다. $Z=\{x_1, x_7\}$에 대한 하한근사는 $R_LZ=\{\ \}$이며, 상한근사는 $R_UZ=\{x_1, x_3, x_6, x_7, x_9\}$이다.

$R_LX=R_UX$이면 R-definable이라 하며, $R_LX\neq R_UX$이면 X는 R에 대해 rough라 한다. $P\subseteq R$일 때 $\cup P$ 또한 동치관계로서 $IND(P)$로 표기하며, P에 대한 불분간 관계라 한다.

라프셋을 영역으로 정의하면 다음과 같다.

X의 R-양영역, $POS_R(X)=R_LX$
X의 R-음영역, $NEG_R(X)=U-R_UX$
X의 R-경계영역, $BND_R(X)=R_UX-R_LX$

분류의 품질은 지식 R을 적용하여 결정 클래스로 정확히 분류될 수 있는 사례의 백분율을 표시한다. 즉, $A=(U, R)$을 지식베이스라 하고 $P, Q\subset R$일 때, 지식 Q는 지식 P에 종속도 $k(0\leq k\leq 1)$만큼 종속된다.

$$k=\gamma_P(Q)=card\ POS_P(Q)/card\ U$$

여기서 A를 사례집합, P를 조건속성, Q를 결정속성이라 할 때, k는 $P\Rightarrow_k Q$인 결정 테이블의 품질 척도

라 할 수 있다.

표 3에서 결정속성값이 1인 클래스의 사례는 3, 4, 14, 15인데, 이를 다음과 같이 표시하면

$X_1 = |f1| = \{3, 4, 14, 15\}$

이와같이, 결정속성값이 2, 3인 경우는

$X_2 = |f2| = \{1, 5, 7, 10, 12, 13, 16\}$
$X_3 = |f3| = \{2, 6, 8, 9, 11\}$

또 조건속성값이 $a2\ b2\ c2\ d1\ e1$인 사례는 1, 13인데, 이를 다음과 같이 표시하면

$Y_1 = |a2\ b2\ c2\ d1\ e1| = \{1, 13\}$

이와같이 나머지에 대해서는

$Y_2 = |a3\ b3\ c2\ d1\ e3| = \{2, 11\}$
$Y_3 = |a1\ b2\ c1\ d2\ e1| = \{3, 7, 10\}$
$Y_4 = |a2\ b1\ c1\ d1\ e1| = \{4\}$
$Y_5 = |a2\ b2\ c1\ d2\ e1| = \{5\}$
$Y_6 = |a3\ b2\ c2\ d1\ e3| = \{6\}$
$Y_7 = |a2\ b3\ c2\ d1\ e2| = \{8, 16\}$
$Y_8 = |a3\ b2\ c1\ d1\ e1| = \{9\}$
$Y_9 = |a1\ b3\ c2\ d2\ e2| = \{12\}$
$Y_{10} = |a1\ b1\ c1\ d2\ e1| = \{14\}$
$Y_{11} = |a1\ b2\ c1\ d1\ e2| = \{15\}$

조건속성을 $C=\{a, b, c, d, e\}$라 하면

$C_L X_1 = Y_4 \cup Y_{10} \cup Y_{11} = \{4, 14, 15\}$
$C_U X_1 = Y_3 \cup Y_4 \cup Y_{10} \cup Y_{11} = \{3, 4, 7, 10, 14, 15\}$
$C_L X_2 = Y_1 \cup Y_5 \cup Y_9 = \{1, 5, 12, 13\}$
$C_U X_2 = Y_1 \cup Y_3 \cup Y_5 \cup Y_7 \cup Y_9$
$= \{1, 3, 5, 7, 8, 10, 12, 13, 16\}$
$C_L X_3 = Y_2 \cup Y_6 \cup Y_8 = \{2, 6, 9, 11\}$
$C_U X_3 = Y_2 \cup Y_6 \cup Y_7 \cup Y_8 = \{2, 6, 8, 9, 11, 16\}$

$X=\{X_1, X_2, X_3\}$이라 할 때 표 3에 대한 양영역의 사례 집합은

$POSc(X) = C_L X_1 \cup C_L X_2 \cup C_L X_3$
$= \{1, 2, 4, 5, 6, 9, 11, 12, 13, 14, 15\}$

따라서 표 3의 품질은

$\gamma_C(X) = card\ POSc(X)/card\ U = 11/16 = 0.69$

품질 척도가 0.6 이상이면 타당성있는 규칙을 발견할 수 있다고 가정하면 표 3에서 타당성있는 규칙을 발견할 수 있을 것이다.

표 4.

	a	*b*	*c*	*d*	*e*	*f*
1	1	2	1	2	1	1
2	2	1	1	1	1	1
3	1	1	1	2	1	1
4	1	2	1	1	2	1
5	2	2	1	2	1	2
6	2	2	2	1	1	2
7	1	2	1	2	1	2
8	1	3	2	2	2	2
9	2	3	2	1	2	2
10	3	3	2	1	3	3
11	3	2	2	1	3	3
12	2	3	2	1	2	3
13	3	2	1	1	1	3

표 3에서 사례 1과 13, 2와 11, 7과 10이 중복되므로 각각 1개씩 삭제 한다. 삭제후 결정속성값 순으로 나열하고 사례 번호를 다시 부여하면 표 4와 같이 된다.

불필요한 속성을 제거하기 위해 라프셋 이론[5]의 지식감축 방법을 이용한다. 지식의 감축은 불필요한 부분(동치관계)을 제거하고, 지식의 불필요한 기본 범주를 제거함으로써 지식이 필수적인 범주의 집합만으로 구성되도록 하는 것이다. 이는 불필요한 지식을 제거함으로써 실제 유용한 지식만 유지하도록 하는 것이다.

다음과 같은 지식 시스템이 있을 때

$S=\{U, A, V\}$ $U=\{x_1, x_2, \cdots, x_n\}$인 사례의 유한집합
$A=\{C, D\}$: 속성의 유한집합,
C는 조건속성, D는 결정속성
$V=\cup_{p\in A} V_p$, V_p: 속성 p의 정의역

$A=C\cup D$이고 $B\subset C$일 때, $IND(D)$에서 양영역 B는 다음과 같이 정의한다.

$POS_B(D) = \cup \{B_L X \mid X \in IND(D)\}$

즉, $POS_B(D)$는 $IND(B)$에 있는 분류 정보에 근거하여 $IND(D)$의 클래스로 분류될 수 있는 모든 사례을 포함한다. $POS_B(D)=POS_{B-\{p\}}(D)$라면 D에 대해 속성 $p \in B$는 B에서 불필요(dispensable) 속성이며, 그렇지 않으면 필수(indispensable) 속성이다.

표 4에서 삭제할 수 있는 속성이 존재하는지 라프셋 이론을 적용하여 조사한다. 조건속성을 $C=\{a, b, c, d, e\}$, 결정속성을 $D=\{f\}$라 할 때

$POS_C(D)=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$

$POS_{C-\{a\}}(D)$={2, 3, 4, 6, 8, 10, 11, 13}
$POS_{C-\{b\}}(D)$={2, 4, 5, 6, 8, 10, 11, 13}
$POS_{C-\{c\}}(D)$={2, 3, 4, 5, 6, 8, 10, 11, 13}
$POS_{C-\{d\}}(D)$={2, 3, 4, 5, 6, 8, 10, 11, 13}
$POS_{C-\{e\}}(D)$={2, 3, 4, 5, 6, 8, 10, 11, 13}

a, b, c, d, e 모두 *D*-indispensable이므로 삭제할 조건속성이 없다.

결과적으로 표 1과 같은 저수준의 데이터베이스가 개념계층의 상승에 의해 표 4와 같이 일반화된 사례 집합으로 간략화된다. 표 4를 정규형의 결정규칙[5]으로 표시하면 다음과 같다.

$a1b2c1d2e1 \vee a2b1c1d1e1 \vee a1b1c1d2e1$
$\vee a1b2c1d1e2$ ---> $f1$
$a2b2c1d2e1 \vee a2b2c2d1e1 \vee a1b2c1d2e1$
$\vee a1b3c2d2e2 \vee a2b3c2d1e2$ ---> $f2$
$a3b3c2d1e3 \vee a3b2c2d1e3 \vee a2b3c2d1e2$
$\vee a3b2c1d1e1$ ---> $f3$

이 규칙은 속성간의 관계가 분석되지 않아 약간의 중복 정보와 불필요한 제약을 포함하고 있을 가능성이 있으며, 규칙이 너무 복잡하므로 일부 조건속성을 제거하여 간략화해야 할 필요가 있다.

3. 조건 속성의 감축

결정속성에 영향을 적게 미치는 조건속성을 감축하기 위해 결정트리에 의한 귀납법중 획득 정보량의 측정방법을 이용한다. 결정트리에 의한 귀납법은 사례가 주어졌을 때 이로부터 개념을 구별할 수 있는 결정트리 형태의 결정규칙을 생성시킨다[15]. 여기서 분류하고자 하는 개념들을 클래스라 하고, 이 클래스는 해당 클래스를 몇개의 속성으로 기술한다. 사례는 속성으로 구성되어 있고, 각 속성은 취할 수 있는 값들을 가지고 있다. 규칙생성은 사례집합 K에 속한 사례들이 모두 하나의 클래스에 속한 것이 아니라면, 어느 한 속성을 선택하여 그 속성이 취하는 값에 따라 사례집합 K를 $K_1, K_2, \cdots, K_n$으로 나눈다. 여기서 K_i는 해당 속성이 i번째 속성값을 취하는 부분집합이다. 이때 선택된 속성은 루트노드를 형성한다. 속성을 선택할 때는 가급적 트리의 크기가 작아질수 있도록, 즉 사례들의 분별력이 가장 큰 속성을 선택하는 것이 바람직하다. 각 속성들의 분별력의 정도를 측정하기 위해 Quinlan[11]은 정보의 복잡성 및 단순성을 측정하는 정보 측정치(Information theoretic measure)를 사용하였다.

사례집합 K가 가지고 있는 정보값은 다음과 같은 엔트로피(Entropy)로 나타낼 수 있다.

$$M(K) = \sum_{i=1}^{m} P_i \log_m (1/P_i) = -\sum_{i=1}^{m} P_i \log_m P_i$$

여기서 P_i는 클래스 K_i가 사례집합 K에서 차지하는 비율이다.

속성 X_j, $j=1, \cdots, m$가 $|Xj|$가지의 속성값을 가지고, 클래스는 K_i, $i=1, \cdots, m$일 때, 속성 Xj를 사용하여 집합 K를 나누었을 경우 정보값 $B(Xj)$는 다음과 같다.

$$B(Xj) = \sum_{i=1}^{|Xj|} Wi * M(Si)$$

여기서 $M(Si)$는 Xj 속성의 i번째 클래스의 값을 가지는 경우 하위 사례집합 Si의 정보값이고, Wi는 가중치로서 다음과 같다.

$$Wi = \frac{Si\text{에서의 사례의 수}}{K\text{에서의 사례의 수}}$$

정보값 $M(K)$를 가지고 있는 사례집합을 속성 x를 선택하여 하위 사례집합으로 나누었을 경우의 정보값 $B(x)$가 원래 $M(K)$보다 작다면 속성 x로 인해 획득한 정보값 $gain(x)$는 다음과 같다.

$$gain(x) = M(K) - B(x)$$

그리고 속성별 획득 정보량을 상대적으로 평가하기 위해 $gain(x)$를 정규화한 속성의 중요도 $S(x)$를 다음과 같이 정의한다.

$$S(x) = gain(x)/M(K)$$

여기서 $S(x)$는 0에서 1까지의 값을 가지게 되는데, 1에 가까울수록 속성의 중요도가 크며 결정속성에 영향을 많이 미친다고 볼 수 있다.

표 4의 사례 테이블에서 결정트리를 생성하는 데는 어느 조건속성을 루트로 하느냐에 따라 5가지 경우가 있다. e를 결정속성이라 할 때 이 사례집합의 정보값은

$$M(K) = -(4/13)\log_3(4/13) - (5/13)\log_3(5/13) - (4/13)\log_3(4/13) = 0.995$$

• 속성 a를 선택하여 하위 사례집합으로 나눈 경우의 결정트리는

a					
1		2		3	
12121	1	21111	1	33213	3
11121	1	22121	2	32213	3
12112	1	22211	2	32111	3
12121	2	23212	2		
13222	2	23212	3		

1의 하위집합 정보값:

$M(S_1) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$

2의 하위집합 정보값:

$M(S_2) = -(1/5)\log_3(1/5) - (3/5)\log_3(3/5) - (1/5)\log_3(1/5) = 0.861$

3의 하위집합 정보값:

$M(S_3) = 0$

$B(a) = 0.971*(5/13) + 0.861*(5/13) + 0*(3/13) = 0.704$

$gain(a) = 0.995 - 0.704 = 0.291$

$S(a) = 0.291/0.995 = 0.292$

• 속성 b를 선택하여 하위 사례집합으로 나눈 경우의 결정트리는

b		
1	2	3
21111 1	12121 1	13222 2
11121 1	12112 1	23212 2
	22121 2	33213 3
	22211 2	23212 3
	12121 2	
	32213 3	
	32111 3	

1의 하위집합 정보값:

$M(S_1) = 0$

2의 하위집합 정보값:

$M(S_2) = -(2/7)\log_3(2/7) - (3/7)\log_3(3/7) - (2/7)\log_3(2/7) = 0.983$

3의 하위집합 정보값:

$M(S_3) = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$

$B(b) = 0*(2/13) + 0.983*(7/13) + 1*(4/13) = 0.837$

$gain(b) = 0.995 - 0.837 = 0.158$

$S(b) = 0.1580.995 = 0.159$

이와같이 c, d, e에 대해서도 하위 사례집합을 만들어 중요도를 계산해 보면 각각 0.065, 0.040, 0.213이 나온다. 만약 중요도가 0.1미만은 결정속성에 별 영향을 미치지 않아 버릴 수 있다고 가정하면, 속성 c, d는 제거되어 표 4는 표 5와 같이 간략화된다. 즉, Type, Transmitter는 Odometer에 가장 영향을 적게 미치는 속성임을 알 수 있다.

사례 5는 사례 6과 같으므로 사례 6을 제거하고 사례 번호를 다시 부여하면 표 6과 같이 된다.

표 6을 정규형의 결정규칙으로 표현하면 다음과 같다.

$a1b2e1 \vee a2b1e1 \vee a1b1e1$ ---> $f1$

$a2b2e2 \vee a2b2e1 \vee a1b2e1 \vee a1b3e2 \vee a2b3e2$ ---> $f2$

표 5.

	a	b	e	f
1	1	2	1	1
2	2	1	1	1
3	1	1	1	1
4	1	2	2	1
5	2	2	1	2
6	2	2	1	2
7	1	2	1	2
8	1	3	2	2
9	2	3	2	2
10	3	3	3	3
11	3	2	3	3
12	2	3	2	3
13	3	2	1	3

표 6.

	a	b	e	f
1	1	2	1	1
2	2	1	1	1
3	1	1	1	1
4	1	2	2	1
5	2	2	1	2
6	1	2	1	2
7	1	3	2	2
8	2	3	2	2
9	3	3	3	3
10	3	2	3	3
11	2	3	2	3
12	3	2	1	3

$a3b3e3 \vee a3b2e3 \vee a2b3e2 \vee a3b2e1$ ---> $f3$

그런데, 이 결정규칙은 속성값간에 중복정보를 포함하고 있을 가능성이 있으므로 속성값간의 종속관계를 분석하여 중복 속성값을 제거하도록 한다.

4. 속성값의 제거

라프셋 이론[5]에 의한 속성값의 제거에 있어서 감축(reduct)과 core라는 두가지 기본 개념이 매우 중요한 역할을 한다. 지식의 감축은 특정 지식에 필요한 기본 개념을 정의하는데 충분한 필수 부분이며, core는 그중 가장 중요한 부분이다. 결정속성 D에 대해 조건속성 C에 있는 필수 속성집합을 C의 core라 하며 다음과 같이 정의한다.

$CORE(C,D)=\{p\in C \mid POS_C(D)\neq POS_{C-\{p\}}(D)\}$

그리고 D에 대한 C의 감축 $RED(C, D)$와의 관계는 다음과 같다.

$CORE(C,D)=\cap RED(C,D)$

$B\subset C$일 때 $POS_C(D)=POS_B(D)$이면 B는 지식 시스템의 감축이다. 즉, 감축은 지식 시스템에 의해 결정규칙을 분간할 수 있는 필수 부분이다.

집합군 $F=\{X_1,\cdots,X_n\}$에서 $\cap(F-\{Xi\})=\cap F$이면 Xi는 dispensable고, 그렇지 않으면 indispensable이다. 예를 들어 $F=\{X,Y,Z\}$에서 $X=\{x_1,x_3,x_8\}$, $Y=\{x_1,x_3,x_4,x_5,x_6\}$, $Z=\{x_1,x_3,x_4,x_6,x_7\}$이라면, $\cap F=X\cap Y\cap Z=\{x_1,x_3\}$, $\cap\{F-\{X\}\}=Y\cap Z=\{x_1,x_3,x_4,x_6\}$, $\cap(F-\{Y\})=X\cap Z=\{x_1,x_3\}$, $\cap(F-\{Z\})=X\cap Y=\{x_1,x_3\}$이므로, Y, Z는 dispensable이다. 따라서 F의 core는 X이고 감축은 $\{X, Y\}$, $\{X, Z\}$이다.

표 6을 라프셋 이론에 의하여 분류하려면, 조건 속성의 남아있는 속성 값을 감축하기 위해 조건 속성의 core 값을 계산해야 한다. 사례 1에 있어서 집합군의 core는 다음과 같이 계산한다.

$$F=[1]a,[1]b,[1]e=\{\{1,3,6,7\},\{1,4,5,6,10,12\},\{1,2,3,5,6,12\}\}$$

dispensible 속성을 찾기 위해 한번에 한 속성씩 제거하고 남은 속성의 교집합이 결정 속성 $[1]f=\{1, 2, 3\}$에 포함되어 있는지를 조사해야 한다. 즉,

$[1]b\cap[1]e=\{1,4,5,6,10,12\}\cap\{1,2,3,5,6,12\}=\{1,5,6,12\}$

$[1]a\cap[1]e=\{1,3,6,7\}\cap\{1,2,3,5,6,12\}=\{1,3,6\}$

$[1]a\cap[1]b=\{1,3,6,7\}\cap\{1,4,5,6,10,12\}=\{1,6\}$

모두 $[1]f=\{1,2,3\}$에 포함되지 않으므로 a, b, e 모두 core이며, core 값은 $a(1)=1$, $b(1)=2$, $e(1)=1$이다.

이와같이 모든 사례에서 조건속성의 core 값을 계산하면, 사례 2에서는 $[2]b\cap[2]e=\{2,3\}\cap\{1,2,3,5,6,12\}=\{2,3\}$, $[2]a\cap[2]e=\{2,4,5,8,11\}\cap\{1,2,3,5,6,12\}=\{2,5\}$, $[2]a\cap[2]b=\{2,4,5,8,11\}\cap\{2,3\}=\{2\}$이므로, $[2]a\cap[2]e$는 $[2]f=\{1,2,3\}$에 포함되지 않아 core는 b이며 core 값은 $b(2)=1$이다.

사례 3에서는 $[3]b\cap[3]e=\{2,3\}\cap\{1,2,3,5,6,12\}=\{2,3\}$, $[3]a\cap[3]e=\{1,3,6,7\}\cap\{1,2,3,5,6,12\}=\{1,3,6\}$, $[3]a\cap[3]b=\{1,3,6,7\}\cap\{2,3\}=\{3\}$이므로, $[3]a\cap[3]e$는 $[3]f=\{1,2,3\}$에 포함되지 않아 core는 b이며, core 값은 $b(3)=1$이다.

사례 4에서는 $[4]b\cap[4]e=\{1,4,5,6,10,12\}\cap\{4,7,8,11\}=\{4\}$, $[4]a\cap[4]e=\{2,4,5,8,11\}\cap\{4,7,8,11\}=\{4,8,11\}$, $[4]a\cap[4]b=\{2,4,5,8,11\}\cap\{1,4,5,6,10,12\}=\{4,5\}$이므로, $[4]a\cap[4]e$는 $[4]f=\{4,5,6,7,8\}$에 포함되지 않아 core는 b이며, core 값은 $b(4)=2$이다.

사례 5에서 $[5]b\cap[5]e=\{1,4,5,6,10,12\}\cap\{1,2,3,5,6,12\}=\{1,5,6,12\}$, $[5]a\cap[5]e=\{2,4,5,8,11\}\cap\{1,2,3,5,6,12\}=\{2,5\}$, $[5]a\cap[5]b=\{2,4,5,8,11\}\cap\{1,4,5,6,10,12\}=\{4,5\}$이므로, $[5]b\cap[5]e$, $[5]a\cap[5]e$는 $[5]f=\{4,5,6,7,8\}$에 포함되지 않아 core는 a, b이며, core 값은 $a(5)=2$, $b(5)=2$이다.

사례 6에서 $[6]b\cap[6]e=\{1,4,5,6,10,12\}\cap\{1,2,3,5,6,12\}=\{1,5,6,12\}$, $[6]a\cap[6]e=\{1,3,6,7\}\cap\{1,2,3,5,6,12\}=\{1,3,6\}$, $[6]a\cap[6]b=\{1,3,6,7\}\cap\{1,4,5,6,10,12\}=\{1,6\}$이므로, 모두 $[6]f=\{4,5,6,7,8\}$에 포함되지 않아 a, b, e 모두 core이다.

사례 7에서 $[7]b\cap[7]e=\{7,8,9,11\}\cap\{4,7,8,11\}=\{7,8,11\}$, $[7]a\cap[7]e=\{1,3,6,7\}\cap\{4,7,8,11\}=\{7\}$, $[7]a\cap[7]b=\{1,3,6,7\}\cap\{7,8,9,11\}=\{7\}$이므로, $[7]b\cap[7]e$는 $[7]f=\{4,5,6,7,8\}$에 포함되지 않아 core는 a이며, core 값은 $a(7)=1$이다.

사례 8에서 $[8]b\cap[8]e=\{7,8,9,11\}\cap\{4,7,8,11\}=\{7,8,11\}$, $[8]a\cap[8]e=\{2,4,5,8,11\}\cap\{4,7,8,11\}=\{4,8,11\}$, $[8]a\cap[8]b=\{2,4,5,8,11\}\cap\{7,8,9,11\}=\{8,11\}$이므로, 모두 $[8]f=\{4,5,6,7,8\}$에 포함되지 않아 a, b, e 모두 core이다.

사례 9에서 $[9]b\cap[9]e=\{7,8,9,11\}\cap\{9,10\}=\{9\}$, $[9]a\cap[9]e=\{9,10,12\}\cap\{9,10\}=\{9,10\}$, $[9]a\cap[9]b=\{9,10,12\}\cap\{7,8,9,11\}=\{9\}$이므로, 모두 $[9]f=\{9,10,11,12\}$에 포함되어 core는 없다.

사례 10에서 $[10]b\cap[10]e=\{1,4,5,6,10,12\}\cap\{9,10\}=\{10\}$, $[10]a\cap[10]e=\{9,10,12\}\cap\{9,10\}=\{9,10\}$, $[10]a\cap[10]b=\{9,10,12\}\cap\{1,4,5,6,10,12\}=\{10\}$이므로, 모두 $[10]f=\{9,10,11,12\}$에 포함되어 core는 없다.

사례 11에서 $[11]b\cap[11]e=\{7,8,9,11\}\cap\{4,7,8,11\}=\{7,8,11\}$, $[11]a\cap[11]e=\{2,4,5,8,11\}\cap\{4,7,8,11\}=\{4,8,11\}$, $[11]a\cap[11]b=\{2,4,5,8,11\}\cap\{7,8,9,11\}=\{8,11\}$이므로, 모두 $[11]f=\{9,10,11,12\}$에 포함되지 않아 a, b, e 모두 core이다.

사례 12에서 $[12]b\cap[12]e=\{1,4,5,6,10,12\}\cap\{1,2,3,5,6,12\}=\{1,5,6,12\}$, $[12]a\cap[12]e=\{9,10,12\}\cap\{1,2,3,5,6,12\}=\{12\}$, $[12]a\cap[12]b=\{9,10,12\}\cap\{1,4,5,6,10,12\}=\{10,12\}$이므로, $[12]b\cap[12]e$는 $[12]f=\{9,10,11,12\}$에 포함되지 않아 core는 a이며, core 값은 $a(12)=3$이다.

이상과 같이 각 사례에 있어서 조건 속성의 core 값을 테이블로 표시하면 표 7과 같다.

표 7.

	a	*b*	*e*	*f*
1	1	2	1	1
2	–	1	–	1
3	–	1	–	1
4	–	2	–	2
5	2	2	–	2
6	1	2	1	2
7	1	–	–	2
8	2	3	2	2
9	–	–	–	3
10	–	–	–	3
11	2	3	2	3
12	3	–	–	3

조건 속성의 core 값을 알면, 속성값의 감축을 계산할 수 있다. 이때 core 값만으로 감축이 될 수 있는지 먼저 조사한다.

사례 1은 모두 core이므로 감축이 없다.

사례 2는 [2]*b*-->[2]*f*가 성립하므로 이 자체가 감축이다.

사례 3은 [3]*b*-->[3]*f*가 성립하므로 이 자체가 감축이다.

사례 4는 [4]*b*-->[4]*f*가 성립하지 못하므로, F={[4]*a*, [4]*b*, [4]*e*}={{2, 4, 5, 8, 11}, {1, 4, 5, 6, 10, 12}, {4, 7, 8, 11}}에서 core를 포함하는 서브군 $G \subseteq F$ such that $\cap G \subseteq$[4]*f*={4, 5, 6, 7, 8}을 찾으면

[4]*b* ∩ [4]*e*={1, 4, 5, 6, 10, 12} ∩ {4, 7, 8, 11}={4}
[4]*a* ∩ [4]*b*={2, 4, 5, 8, 11} ∩ {1, 4, 5, 6, 10, 12}={4, 5}

둘다 [4]*f*에 포함되므로 *a*(4)=2 and *b*(4)=2와 *b*(4)=2 and *e*(4)=2가 속성값의 감축이다.

사례 5는 [5]*a* ∩ [5]*b*-->[5]*f*가 성립하므로 이 자체가 감축이다.

사례 6은 모두 core이므로 감축이 없다.

사례 7은 [7]*a*-->[7]*f*가 성립하지 못하므로, F={[7]*a*, [7]*b*, [7]*e*}={{1, 3, 6, 7}, {7, 8, 9, 11}, {4, 7, 8, 11}}에서 core를 포함하는 서브군 $G \subseteq F$ such that $\cap G \subseteq$[7]*f*={4, 5, 6, 7, 8}을 찾으면

[7]*a* ∩ [7]*e*={1, 3, 6, 7} ∩ {4, 7, 8, 11}={7}
[7]*a* ∩ [7]*b*={1, 3, 6, 7} ∩ {7, 8, 9, 11}={7}

둘다 [7]*f*에 포함되므로 *a*(7)=1 and *b*(7)=3과 *a*(7)=1 and *e*(7)=2가 속성값의 감축이다.

사례 8은 모두 core이므로 감축이 없다.

사례 9는 core가 없으므로, F={[9]*a*, [9]*b*, [9]*e*}={{9, 10, 12}, {7, 8, 9, 11}, {9, 10}}에서 서브군 $G \subseteq F$ such that $\cap G \subseteq$[9]*f*={9, 10, 11, 12}를 찾으면 [9]*a*와 [9]*e*이므로 *a*(9)=3과 *e*(9)=3이 속성값의 감축이다.

사례 10은 core가 없으므로, F={[10]*a*, [10]*b*, [10]*e*}={{9, 10, 12}, {1, 4, 5, 6, 10, 12}, {9, 10}}에서 서브군 $G \subseteq F$ such that $\cap G \subseteq$[10]*f*={9, 10, 11, 12}를 찾으면 [10]*a*와 [10]*e*이므로 *a*(10)=3과 *e*(10)=3이 속성값의 감축이다.

사례 11은 모두 core이므로 감축이 없다.

사례 12는 [12]*a*-->[12]*f*가 성립하므로 이 자체가 감축이다.

따라서 표 8을 얻을 수 있다.

여기서 중복되는 사례들중 하나씩만 남기고 제거하

표 8.

	a	*b*	*e*	*f*
1	1	2	1	1
2	×	1	×	1
3	×	1	×	1
4	2	2	×	2
4'	×	2	2	2
5	2	2	×	2
6	1	2	1	2
7	1	3	×	2
7'	1	×	2	2
8	2	3	2	2
9	3	×	×	3
9'	×	×	3	3
10	3	×	×	3
10'	×	×	3	3
11	2	3	2	3
12	3	×	×	3

표 9.

	a	*b*	*e*	*f*
1	1	2	1	1
2	×	1	×	1
4	2	2	×	2
4'	×	2	2	2
6	1	2	1	2
7	1	3	×	2
7'	1	×	2	2
8	2	3	2	2
9	3	×	×	3
9'	×	×	3	3
11	2	3	2	3

면 표 9와 같이 된다.

표 9를 결정규칙으로 표현하면 다음과 같다.

$a1b2e1 \vee b1$ --> $f1$
$a2b2 \vee b2e2 \vee a1b2e1 \vee a1b3 \vee a1e2 \vee a2b3e2$ --> $f2$
$a3 \vee e3 \vee a2b3e2$ --> $f3$

이를 중고자동차 데이터베이스에서 발견한 지식으로 표현한다면 다음과 같다.

1) Model=Kia and Size=Medium and Year=Old or Size=Large --> Odometer=Long
2) Model=Hyundai and Size=Medium or Size=Medium and Year=Medium or Model=Kia and Size=Medium and Year=Old or Model=Kia and Size=Small or Model=Kia and Year=Medium or Model=Hyundai and Size=Small and Year=Medium --> Odometer=Medium
3) Model=Daewoo or Year=Short or Model=Hyundai and Size=Small and Year=Medium --> Odometer=Short

5. 평가 및 결론

속성중심 귀납법은 표 1과 같은 저수준의 데이터베이스를 개념계층에 의한 개념상승을 시켜 표 4와 같은 일반화된 지식표현 시스템으로 유도했으나 이를 발견된 규칙으로 보기에는 너무 복잡하다. 예를 들어 클래스 1에 대해 언어변수를 사용한 지식으로 표현한다면 다음과 같이 매우 복잡하다.

Model=Kia and Size=Medium and Type=SOHC and Transmitter=Manual and Year=Old or Model=Hyundai and Size=Large and Type=SOHC and Transmitter=Auto and Year=Old or Model=Kia and Size=Large and Type=SOHC and Transmitter=Manual and Year=Medium or Model=Kia and Size=Medium and Type=SOHC and Transmitter=Auto and Year=Medium --> Odometer=Long

표 1을 속성중심 귀납법에 의해 일반화를 시키지 않고 바로 결정트리에 의한 귀납법에 적용해 본다면 결정트리의 크기가 매우 크게되어 다루기가 힘들 뿐만 아니라 획득 정보량 계산에도 많은 시간을 필요로 한다. 그리고 실제 결정규칙을 도출했다 하더라도 너무 저수준의 표현이라 지식으로서의 추상성이 없다. 예를 들어 표 1에 있는 첫 번째 튜플을 다음과 같이 지식으로 표현했다 하더라도 아무런 의미가 없는 사실에 지나지 않는다.

Model=Sonata and Size=Medium and Type=DOHC and Transmitter=Auto and Year=90 --> Odometer=156000

또, 표 1에 바로 라프셋의 지식감축 방법을 적용시켜 본다면 속성의 감축이나 속성값의 감축에 있어서 속성수가 늘어남에 따라 계산시간이 $O(2^n)$의 시간 복잡도를 가진다. 또한 저수준의 결정규칙이 도출되므로 별 의미가 없는 지식이 된다. 따라서 이 세가지 기법을 결합한 통합 방법은 먼저 데이터베이스를 일반화시켜 추상성을 높이고 또한 튜플 수를 감소시키며, 둘째 의미가 적은 속성을 제거함으로써 조건속성 수를 줄여 최소화된 지식표현 시스템으로 변환한다. 그리고 속성값간의 관계를 조사하여 불필요한 속성값을 제거함으로써 최소화 결정규칙을 도출한다. 이 결정규칙은 4장 끝부분에서 본바와 같이 매우 간결하며 지식이 고수준의 추상으로 표현되어 그 의미를 이해하기 쉽다.

데이터베이스에서의 지식 발견은 중요한 규칙을 도출할 수 있을 뿐만 아니라 고수준 개념의 질의처리에도 유용하게 이용될수 있으며, 발견된 지식으로 지식베이스를 구축하는데 도움을 줄 수 있다. 본 연구의 결과는 의사결정, 데이터 분류, 데이터 요약, 분류규칙의 학습 등 여러 분야에 적용될수 있을 것이다.

그런데 서론에서 데이터베이스는 대량의 비교적 신뢰할 만한 데이터를 저장하고 있다고 전제로 제시한 바와 같이 데이터가 신뢰할만 하다면 도출된 지식은 모순이 없이 신뢰할 수 있으나, 데이터가 부정확하다면 도출된 지식도 신뢰할 수 없다. 이를 해결하는 방향은 도출된 지식을 훈련 데이터를 사용하여 계속 정련함으로서 신뢰도를 높이는 방법이 있으며, 또 지식의 유도 과정에서 서로 모순이 되는 규칙을 찾아 제거하는 방법이 있다. 전자는 동적 데이터베이스에서의 점진적 지식발견 방법을 개발해야 할 것이며, 후자의 방법은 모순된 데이터를 제거함으로써 생기는 정보의 손실을 정량적으로 분석해서 제거의 유무를 판단할 수 있는 근거를 설명할 수 있는 알고리즘을 개발해야 할 것이다.

데이터베이스를 일반화시켜 만든 지식표현 시스템에서 얼마나 타당성 있는 규칙을 발견할 수 있는가는 품질 척도에 의해 사용자가 결정해야 하는데, 이는 서론에서 제시한 전제의 하나인 지식발견 과정은 사용자의 학습 요구수준에 따라 약간씩 달라질 수 있다는 것을 의미한다. 그런데 이 품질 척도의 기준을 사용자가 임의로 정한다면 유도된 규칙에 보편적인 신뢰성을 부여하기가 어려울 것이다. 따라서 품질척도의 기준에 대해서도 상기 후자의 방법에서 같이 연구되어야 할 과제이다.

참고문헌

[1] N. Cercone and M. Tsuchiya, "Special Issue on Learning and Discovery in Databases", *IEEE Transactions on Knowledge and Data Engineering* 5, 1993.

[2] W. Frawley, G. Piatetsky-Shapiro and C. Matheus, "Knowledge Discovery in Databases: An Overview", *AI Magazine*, Fall, 1992.

[3] J. Han, Y. Cai and N. Cercone, "Knowledge Discovery in Databases: An Attribute-Oriented Approach," *Proceeding of the 18th Conference on Very Large Data Bases*, Vancouver, Canada, pp. 340-355, 1992.

[4] X. Hu, N. Cercone and J. Han, "An Attribute-Oriented Rough Set Approach for Knowledge in Databases," *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, Alberta, Canada, 12-15, October 1993.

[5] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer, 1991.

[6] Z. Pawlak, "Rough Sets," *International Journal of Computer and Information Science*, **11**, pp. 341-356, 1982.

[7] G. Piatetsky-Shaprio, *Report on AAA-91 Workshop on Knowledge Discovery in Databases*, IEEE Expert, October, 1991.

[8] G. Piatetsky-Shaprio, *KDD-93: Proceedings of AAA-93 Workshop on Knowledge Discovery in Databases*, AAAI Press, 1993.

[9] G. Piatetsky-Shaprio and C. Matheus, *Knowledge Discovery Workbench for Exploring Business Databases*, Internal J. of Intelligent Systems, September, 1992.

[10] G. Piatetsky-Shaprio, "Special Issue on Knowledge Discovery in Databases", *J. of Intelligent Information Systems* 3, December, 1994.

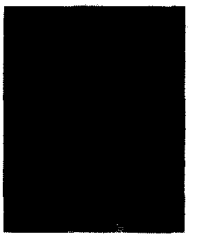
[11] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning* **1**, pp. 81-106, 1986.

[12] W. Ziarko, *Proceedings of the Int. Workshop on Rough Sets and Knowledge Discovery*, Banff, Canada, 1993.

[13] J, Zytkow, *Proceedings of the Machine Discovery Workshop*, Aberdeen, Scotland, July, 1992.

[14] J. Zytkow, "Special Issue on Machine Discovery", *Machine Learning* 12, 1993.

[15] 이재규 등, *전문가 시스템*, 법영사, 1995.



정 홍(Hong Chung)

1972년 : 한양대학교 원자력공학과(공학사)

1976년 : 고려대학교 경영대학원(경영학석사)

1996년 : 대구효성가톨릭대학교 전산통계학과(이학석사)

1996년~현재 : 대구효성가톨릭대학교 박사과정

1972년~1981년 : 한국과학기술연구원 선임연구원

1981년~현재 : 계명대학교 컴퓨터공학과 부교수

주관심분야 : 지능정보시스템, 소프트웨어공학

정환묵(Hwan Mook Chung) 종신회원

대구효성가톨릭대학교 공과대학 전자공학부 교수

퍼지 및 지능 시스템학회 논문지 7권 2호 참조