Threshold를 이용한 의사결정나무의 생성
-Induction of Decision Tress Using the Threshold Concept-

이 후 석
Lee, Hu Seok
김 재 련
Kim, Jae Yearn

Abstract

This paper addresses the data classification using the induction of decision trees. A weakness of other techniques of induction of decision trees is that decision trees are too large because they construct decision trees until leaf nodes have a single class. Our study include both overcoming this weakness and constructing decision trees which is small and accurate. First, we construct the decision trees using classification threshold and exception threshold in construction stage. Next, we present two stage pruning method using classification threshold and reduced error pruning in pruning stage.

Empirical results show that our method obtain the decision trees which is accurate and small.

1. 서 론

기업의 업무 환경이 전산화됨에 따라 모든 데이터가 데이터베이스로 관리되기 시작했고 그 결과로 대용량의 데이타베이스가 출현하게 되었다. 이러한 대용량의 데이터베이스는 우리 주위에서도 쉽게 찾아 볼 수 있는데 예를 들면 신용카드, 회사, 병원, 패스트 푸드점 등은 무수히 많은 정보들을 매일 생산해내고 있다. 그러나 데이타를 수집하고 저장하는 능력에 비해 데이타를 분석하고 이해하는 능력이 뒤떨어지게 되었다. 그리하여 대용량의 데이타베이스로 부터 의사결정에 유용한 지식을 추론해 내는 새로운 기술이 필요하게 되었고 그 결과로 데이터 마이닝이 출현하게 되었다.

데이터베이스로 부터 지식을 탐사할 때 찾고자 하는 지식의 종류에 따라 마이닝 방법도 달라지게 된다. 가장 많이 사용되고 있는 마이닝 기법은 분류(Classification), 연관(Association), Clustering 등의 방법들이 있다. 본 연구에서는 데이터 마이닝의 여러 기법 중 의사결정나무를 이용한 데이터의 분류에 대해 논의한다. 의사결정나무를 이용한 데이터의 분류는 이미 오래전 부터 여러 분야에서 연구되어 왔고 그 결과로 많은 분류 알고리즘이 여러 문헌에서 제안되었다. 가장 잘 알려진 데이터 분류 방법은 Quinlan[1], Kamber, et al.[4], SLIQ[6], Weiss[9]

*한양대학교 산업공학과

등이다. 그 중 Quinlan[1]는 데이터를 분류하는 방법 중 가장 널리 알려진 방법으로써 본 연구에서도 Quinlan[1]에서 사용되었던 Gain Ratio방법을 채택하여 사용하고 있다. 그러나 Quinlan[1]는 속성 선택에 있어 숫자 속성을 주로 선택하는 단점과 크기가 큰 데이터를 처리할 수 없다는 단점을 가지고 있어 대용량의 데이터베이스에 적용하기에는 적합치 않다. 또한 이러한 단점은 기존 연구들의 단점이기도 한데 최근에 와서 이러한 기존 연구의 단점을 보완한 방법들이 많이 제안되었다. SLIQ[6]와 SPRINT[7]는 기존 방법들의 한계를 극복하기 위한 방법으로 Pre-Sorting, Breadth-First Growth 방법을 제안하였다. 또한 Kamber, et al.[4]은 Attribute Oriented Induction 방법과 Threshold를 이용하여 대용량의 데이터 베이스에 적용할 수 있는 분류 방법을 제안하였다.

본 연구에서는 의사결정나무의 생성 단계에서 Classification Threshold와 Exception Threshold를 사용하여 각각의 노드가 같은 등급(Class)으로만 구성되지 않아도 말단 노드로 함으로써 의사결정나무의 노드 수를 줄이는 방법을 사용하였고 또한 Pruning 단계에서 의사결정나무의 생성 단계에서 사용하였던 Classification Threshold를 이용하여 Pruning 방법의 효율을 개선하는 2 단계 Pruning방법을 제안하였다.

2. 분류 알고리즘

본 장에서는 기존 연구의 문제점에 대해서 살펴보고 본 연구가 제안하는 Threshold를 이용한 의사결정나무의 생성과 Pruning 방법을 제시한다.

2.1 기존 연구의 문제점

본 연구의 방법이 적용될 데이터는 대용량의 데이터베이스이다. 대용량의 데이터 베이스는 그 특성상 기존의 방법들이 사용했던 정제된 데이터들과는 달리 데이터의 통계적 변동이 크다는 특징을 가지고 있다. 이러한 데이터의 통계적 변동은 여러 가지 원인에 기인할 수 있겠지만 가장 중요한 원인은 데이터의 크기이다. 즉 데이터의 개수가 많아질 수록 그 안에 포함되어 있는 데이터의 다양성이 증가하여 각 데이터의 변동의 범위가 커지는 것이다. 이러한 데이터의 통계적 변동으로 나타날 수 있는 문제들 중 가장 큰 문제는 [Table 2.1]에서 보여지듯이 대용량의 데이터베이스에서는 모든 속성값이 같은 데도 불구하고 다른 등급에 속하는 데이터가 존재할 수 있다는 것이다.

기존의 방법들은 데이터의 수가 작기 때문에 데이터의 통계적 변동의 범위가 작고, 속성값이 같은 데도 다른 등급으로 분류되는 문제를 가지지 않는다. 그러나 분류 방법이 적용될 데이터의 개수가 10^6-10^7가 되는 경우에는 기존의 방법들이 제안한 각각의 말단 노드가 같은 등급으로 분류되어야만 한다는 가정은 타당하지 않다.

[Table 2.1] 데이터의 통계적 변동의 예

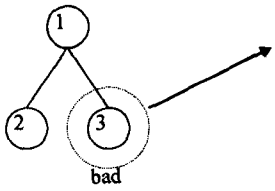
outlook	Temperature	Humidity	Windy	Class
rain	70	80	False	Play
rain	70	80	False	Don't Play
sunny	69	70	False	Don't Play
sunny	69	70	False	Play

오히려 의사결정나무의 생성단계에서 데이터의 통계적 변동을 고려하여 말단노드의 등급을 결정하는 것이 새로운 데이터에 대한 의사결정나무의 분류의 정확도를 높이는 방법이 될 수도 있다[4],[5].

본 연구에서는 의사결정나무의 생성단계 뿐 아니라 Pruning 단계에서도 이러한 데이터의 통계적 변동을 고려한 방법을 제안하였다.

2.2 의사결정나무의 생성

본 연구의 의사결정나무의 생성 방법은 Kamber, et al.[4]에서 제안되었던 Classification Threshold와 Exception Threshold를 사용한 방법이다. 기존의 방법들의 문제점 중 한가지는 [Fig. 2.1]에서 보여지듯이 반복적인 분지의 결과로 인해 생성된 학습자료집합이 너무 작아서 더 이상의 분지가 통계적으로 무의미한 경우가 발생한다는 것이다. 즉 이러한 학습자료집합을 분지함으로 인해 학습자료집합에서 발생확률이 낮은 데이터에 대한 분지가 이루어짐으로써 이러한 방법으로 생성된 의사결정나무를 새로운 데이터를 분류하기 위해 적용하였을 때 오히려 분류의 정확도를 떨어뜨리는 결과를 초래할 수도 있는 것이다. [Fig. 2.1]에서 Table은 노드 3에 속하는 학습자료집합을 나타내고 있고 이와 같이.학습자료집합의 개수가 적은 경우에도 기존 방법을 적용하면 노드3 이 세 가지의 등급을 가짐으로 인해 분지가 발생하게 된다. 이러한 문제를 해결하기 위해 도입된 방법이 Exception Threshold를 사용하는 것이다. Exception Threshold는 이러한 통계적으로 의미가 없는 데이터의 최대 개수이다. 예를 들어 Exception Threshold를 5로 했을 경우 분지를 고려할 학습자료집합의 데이터의 개수가 5이하인 경우 분지를 멈추고 최다 빈도수를 가지는 등급을 그 노드의 등급으로 한다. 본 논문3장의 실험에 사용된 Exception Threshold의 값은 10이다.

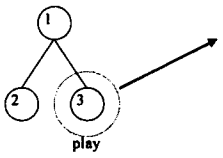


Humidity	Windy	Class
65	True	good
75	False	bad
78	False	bad
90	True	very good

[Fig. 2.1] Exception Threshold에 의한 말단노드의 생성

Classification Threshold는 의사결정나무의 어느 한 노드가 모두 같은 등급으로 구성되지 않더라도 어느 한 등급에 속하는 데이터의 비율이 Classification Threshold보다 큰 경우 분지를 멈추고 그 노드를 말단노드로 만들기 위해 사용된다. 이 경우 그 노드의 등급은 최대 비율을 가지는 등급이 된다. 이러한 Classification Threshold의 사용으로 기존 연구가 가지는 각각의 말단노드가 단 하나의 등급만을 가져야 한다는 제약을 극복할 수 있다. [Fig. 2.2]는 노드 3에서 분지를 고려할 때 90% Classification Threshold에 의해 분지가 멈춰지고 노드 3이 말단노드로 되는 것을 보여주고 있다. [Fig. 2.2]에서 보여지듯이 노드3에 속하는 데이터의 등급들 중 play의 비율이 Classification Threshold 90%보다 크기 때문에 더 이상의 분지를 고려하지 않고 노드 3을 말단노드로 하고 노드 3의 등급은 play가 된다. [Fig. 2.2]에서 Table은 노드 3에 속하는 학습자료집합을 나타내고 있다.

[Fig. 2.2] Classification Threshold에 의한 말단노드의 생성



Humidity	Windy	Class
65	True	Play
75	False	Play
78	False	Play
90	True	Play
70	True	Play
80	False	Play
80	True	Don't Play
80	False	Play
96	False	Play
55	True	Play
70	False	Play

2.3 의사결정나무의 Pruning

본 연구에서 제안한 Pruning방법은 Threshold를 이용한 Pruning이다. 본 연구에서 Threshold를 이용한 Pruning 방법을 제안하게 된 것은 2.1절에서 언급한 것처럼 대용량의 데이터베이스의 데이터의 다양성으로 인해 학습자료집합에 대해 생성된 의사결정나무를 시험자료집합에 대해 적용할 때 의사결정나무의 생성시 분류되었던 등급으로 분류된다는 보장이 없다는 것이다. 오히려 의사결정나무의 생성시 에러로 분류되었던 데이터가 시험자료집합에서는 참으로 분류될 수도 있는 것이다. 물론 반대의 경우도 발생할 수도 있지만 이러한 데이터의 통계적 변동은 본 연구가 제안한 Pruning방법의 타당성의 근거를 제시해 주고 있다.

본 연구의 Pruning방법은 두 단계로 구성된다. 1 단계에서는 Classification Threshold를 이용한 Pruning방법을 사용하고 2 단계에서는 1 단계 Pruning방법에 의해 생성된 의사결정나무에 대해 Reduced Error Pruning방법을 사용한다.

2.3.1 Threshold Pruning

본 연구가 제안하는 Threshold Pruning은 학습자료집합에 대해 수행된다. 의사결정나무 Pruning에 앞서 모든 말단 노드를 검토한다. 말단 노드에는 여러 등급의 학습자료가 존재할 수 있으나 이들을 모두 최다 빈도수 등급의 자료로 간주한다. 이러한 작업을 모든 말단 노드에 대해 개별적으로 수행한 후 다음의 pruning을 수행한다.

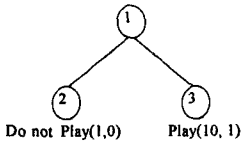
의사결정나무의 노드를 Ni라 하자. N1은 근노드를 나타내고 Nn은 너비우선 탐색 방식으로 검토할 때 마지막 노드를 나타낸다. 다음의 Threshold Pruning방법은 노드 N1부터 시작하여 너비우선 탐색 방식으로 수행되며 모든 노드 Ni에 대해 수행된다.

단계 1 노드 Ni와 자식노드 관계에 있는 모든 말단 노드의 최다 빈도수 등급의 비율을 계산(i =1 ,·, n)

단계 2 Classification Threshold와 비교하여 단계 1에서 구한 비율이 크면 Ni와 자식노드 관계에 있는 모든 노드를 Prune하여 하나의 말단노드로 만든다.

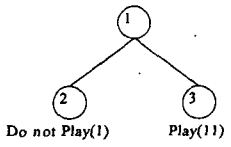
단계 3 단계 2에서 Prune된 노드들을 제외한 나머지 노드들에 대해 i를 증가시켜서 단계 1 과정부터 되풀이 한다. Ni+1노드가 마지막 노드Nn이 되면 루프를 종료한다.

원래 말단노드의 등급은 의사결정나무의 생성시Classification Threshold나 Exception Threshold에 의해 분지가 멈춰져서 생긴 최다 빈도수 등급이다. 따라서 말단노드에는 최다 빈도수 등급에 속하는 데이터 뿐만 아니라 최다 빈도수 등급 이외의 등급에 속하는 데이터가 같이 섞여 있을 수 있다. 그러나 말단 노드에 속하는 모든 학습자료를 최다 빈도수 등급에 속하는 것으로 간주한다는 것은 의사결정나무의 생성시에는 최다 빈도수 등급에 속하지 않았던 데이터도 Threshold Pruning 단계에서는 최다 빈도수 등급의 데이터의 개수에 포함시키겠다는 것을 의미한다. 이러한 작업을 수행하는 것은 의사결정나무가 만들어질 때 최다 빈도수 등급에 속하는 데이터의 비율이 Classification Threshold값보다 약간 적어서 분지가 수행된 경우를 Pruning하기 위함이다. [Fig. 2.3]는 Threshold Pruning이 수행되기 전의 90% Classification Threshold를 사용하여 생성된 의사결정나무를 나타내고 있다.



[Fig. 2.3] Threshold를 사용하여 생성된 의사결정나무

[Fig. 2.3]의 가로안의 숫자는 앞의 숫자가 최다 빈도수 등급의 데이터의 개수를 나타내고 뒤의 숫자가 최다 빈도수 이외의 등급의 데이터의 개수를 나타낸다. 노드 3의 경우 최다 빈도수 등급은 play이고 등급 play의 데이터의 개수는 10이고 나머지 등급의 개수가 1임을 알 수 있다. 따라서 노드 3에 속하는 학습자료집합의 개수는 11이다. 노드1이 말단노드가 되지 않고 노드 2와 노드 3으로 분지된 것은 노드 1의 12개의 학습자료 중 등급 play의 개수가 10개이므로 등급 play의 비율이 Classification Threshold(90%)값보다 적어서 분지되었다는 것을 의미하고 있다. 그러나 노드 3의 등급 Play의 개수를 노드 3에 속하는 모든 학습자료의 개수(즉 11)로 할 때는 경우가 달라지게 된다. 다음은 말단 노드에 속하는 모든 데이터의 개수를 최다 빈도수 등급의 데이터의 개수로 가정했을 때의 의사결정나무를 나타낸다.



[Fig. 2.4] 선처리 작업 후의 의사결정나무

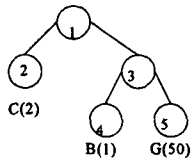
[Fig. 2.4]의 말단노드 2와 3의 데이터의 개수를 살펴보자. [Fig. 2.3]에서 10이었던 등급 play의 데이터의 개수가 [Fig. 2.4]에서는11로 바뀌었고 등급 Don't play의 데이터의 개수가 1이다. 따라서 이제는 등급 Play의 비율이 90% 이상이 되기 때문에 노드 2와 노드 3이 Prune되고 노드 1이 말단노드가 되고 등급은 play가 된다. 즉 말단노드의 최다 빈도수 등급의 개수를 그 말단노드에 속하는 모든 학습자료집합의 데이터의 개수로 함으로써 의사결정나무의 생성시 분지되었던 노드를 Prune할 수 있게 되었다. 이러한 Pruning 방법의 적용으로 최다 빈도수 등급의 비율이 Classification Threshold보다 조금 작아서 분지가 발생한 부노드를 Prune할 확률이 높아지게 되는 것이다. 다음은 본 연구가 제안한 Threshold Pruning을 적용한 후의 의사결정나무이다.

[Fig. 2.5] Threshold Pruning이 적용된 후의 의사결정나무

2.3.2 Reduced Error Pruning

위에서 제시한 방법은 Classfication Threshold값에 조금 못 미쳐 분지된 노드를 Prune할 확률을 높일 수 있지만 최다 빈도수 등급의 비율이 Classification Threshold와 많은 차이가 나는 경우는 prune이 되지 않는다. 이러한 경우를 보완하기 위해 수행되는 방법이 Reduced Error Pruning[4]이다. Reduced Error Pruning은 1 단계에서 Prune된 의사결정나무를 사용하여 시험자료집합에 대해 수행된다. Pruning방법은 다음과 같다.

생성된 의사결정나무를 T라 하자. T의 모든 부나무 S에 대해서 S가 말단노드로 대치되었을 때의 시험자료집합에 대해서 새로운 의사결정나무의 에러의 개수를 측정한다. 새로운 의사결정나무의 에러의 개수가 본래의 의사결정나무 T의 에러의 개수보다 작거나 같으면 부나무 S는 Prune되고 S는 말단노드로 된다. 이러한 과정은 모든 부나무에 대해 수행되고 에러의 개수가 증가할 때까지 계속된다. 그러나 Reduced Error Pruning은 [Fig. 2.6]과 같이 부모노드의 형제노드의 등급이Prune된 노드의 등급과 다른 경우는 Prune이 발생하지 않는 단점이 있다. [Fig. 2.6]과 같은 의사결정나무는 노드 4, 5를 prune하여 등급 G가 된다고 가정했을 때 노드 2의 등급이 C이기 때문에 노드 2, 3, 4, 5를 prune하면 에러가 증가함으로 인해 prune이 발생하지 않게 된다.



[Fig. 2.6] Reduced Error Pruning의 예

그러나 [Fig. 2.6]에서 등급들의 개수를 살펴보면. 등급 G의 비율이 Classification Threshold의 값보다 크다는 것을 알 수 있다. 따라서 위와 같은 의사결정나무는 본 연구가 제안하는 Threshold Pruning에 의해 노드 2, 3, 4, 5가 Prune됨을 알 수 있다. 이와 같이 본 연구가 제안하는 2 단계 Pruning방법은 Reduced Error Pruning 방법으로 Prune되지 않는 의사결정나무가 Threshold Pruning 방법을 통하여 Prune되는 상호 보완적인 작용을 하면서 본 연구의 목적인 정확하면서도 노드수가 작은 의사결정나무를 생성하는 데 효율적인 방법이 됨을 알 수 있다.

3. 실험 및 결과 분석

3.1 실험 방법

본 연구의 의사결정나무의 생성 및 Pruning 이 적용될 데이터는 의사결정나무의 생성을 위한 학습자료집합과 검사 및 Pruning을 위한 시험자료집합으로 분리된다. 먼저 학습자료집합에 대해 본 연구가 제안하는 방법을 통하여 의사결정나무를 생성하고 완성된 의사결정나무에 대

해서 학습자료집합을 사용하여 Threshold Pruning방법을 적용하고 Threshold Pruning이 적용된 의사결정나무에 대해 시험자료집합을 사용하여 Reduced Error Pruning방법을 적용하여 그 결과로 생성된 의사결정나무의 정확도 및 노드수를 측정한다. 측정된 정확도와 노드수는 Quinlan[1]의 정확도 및 노드수와 비교하여 본 연구의 타당성을 검증한다.

본 연구는 실험의 신뢰성을 높이기 위해U.C Irvine repository에 있는 개인의 신용도를 평가하는 Crx 데이터베이스와 자동차의 구매성향을 평가하기 위한 Car 데이터베이스와 의학 관련 내용의 Nurse 데이터베이스에 대해 실험을 수행하였다. 다음은 각각의 데이터베이스의 속성 개수 및 등급의 개수 등에 대한 정보를 표로 요약한 것이다.

[Table 3.1] 실험 데이터의 정보 요약

	Crx	Car	Nurse
데이터 개수	465	1470	11000
속성 개수	15	6	8
등급 개수	2	4	5

위의 실험에 추가하여 본 연구는 각각의 데이터베이스에 대해 최적의 Classification Threshold를 결정하기 위한 실험을 수행하였다. 기존의 Threshold를 이용한 방법들은 Threshold를 결정하는 데 있어 관리자의 경험적 판단에 근거한 경우가 대부분이었다. 그러나 본 연구에서는 여러 경우의 Threshold에 대해 실험을 수행하여 그 결과를 제시함으로써 사용자가 업부 수행에 적합한 Threshold값을 선택할 수 있도록 하였다.

3.2 실험 결과

먼저 각각의 데이터베이스에 대해서 본 연구가 제안하는 방법(Classification Threshold 80%, 85%, 90%, 95%)을 사용한 경우 및 Quinlan[1]에 의해 생성된 의사결정나무의 노드수에 대한 정보를 요약하면 다음 표와 같다.

[Table 3.2] 의사결정나무의 노드수

Classification Threshold	Crx	Car	Nurse
80	3	145	378
85	3	145	392
90	31	145	410
95	62	145	426
C4.5	44	182	498

Quinlan[1]에서 사용된 Pruning 방법은 Pessimistic Pruning방법이다.

[Table 3.2]의 결과로 부터 CRX데이터베이스에 대한 95% Threshold를 제외하고는 모든 경우에 있어서 노드수가 Quinlan[1]에 의해 생성된 의사결정나무 보다 감소한 것을 알 수 있다. Threshold 95%에서 노드수가 감소되지 않은 이유는 Threshold가 높아질 수록 말단노드가 순

수하게 단 하나의 등급으로만 구성될 확률이 높아지기 때문에 본 연구가 제안하는 방법에 의한 Prune이 일어나지 않기 때문이다. 다음은 본 연구의 방법 및 Quinlan[1]를 사용하여 생성된 의사결정나무의 정확도에 대한 정보를 표로 요약한 것이다.

[Table 3.3] 의사결정나무의 정확도

Classification Threshold	Crx	Car	Nurse
80	83	93	96.2
85	83	93	97.6
90	74	93	98.05
95	85.5	93	98.05
C4.5	91.3	96.7	98.2

위의 결과들을 종합해 보면 본 연구가 제안하는 방법이 Quinlan[1]에 비해 정확도가 크게 떨어지지 않으면서 노드수를 줄이는 방법임을 알 수 있다. 또한 80-95%의 Classification Threshold에 대해 실험을 수행한 결과 CRX 데이터베이스에 대해서는 80, 90% Threshold일 때 또한 Nurse데이터베이스에 대해서는 80%일 때 최적의 의사결정나무을 생성하고 CAR데이터베이스에 대해서는 고려된 모든 Classification Threshold가 같은 결과를 산출하는 것으로 나타났다. 여기서 80-95% Threshold를 사용한 것은 기존의 방법들로 생성된 의사결정나무를 새로운 시험 데이터에 적용했을 때의 정확도가 대략 80-95% 사이에 분포하고 있기 때문이다. 따라서 본 연구에서는 이러한 80-95% Threshold를 사용하여 각각의 데이터베이스에 대해 가장 좋은 결과를 산출하는 Classification Threshold값을 제시하였다.

4. 결 론

데이터의 분류는 데이터 마이닝에서 중요한 부분을 차지하고 있다. 그러나 기존의 데이터 분류에 대한 많은 연구에도 불구하고 대용량의 데이터베이스를 대상으로 한 연구는 많지 않은 실정이다. 이러한 이유로 인해 본 연구는 대용량의 데이터베이스에 적용할 수 있는 의사결정나무를 이용한 데이터 분류 방법을 제안하였다. 본 연구가 사용한 방법은 Threshold를 이용한 의사결정나무의 생성과 Pruning이다. 즉 Classification Threshold와 Exception Threshold를 사용하여 의사결정나무를 생성하고 생성된 의사결정나무에 대하여 2 단계 Pruning방법을 사용하여 최종 의사결정나무를 구하는 방법을 제안하였다. 실험 결과는 본 연구가 제안하는 데이터의 분류방법이 기존 연구에 비해 노드수가 적고 정확도가 떨어지지 않는 의사결정나무를 구하는 데 효율적인 방법임을 보여주고 있다.

참고 문헌

1. Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, LA, California, 1993.
2. Quinlan, J. R., Improved Use of Continuous Attributes in C4.5, *Journal of Artificial Intelligence Research*, Vol. 4, PP. 77-90, 1996.
3. Quinlan, J. R., Simplifying Decision Trees, *Int. J. Man-Machine Studies*, Vol. 27, PP. 221-234. 1987.

4. Kamber, M., L. Winstone, W. Gong, S. Cheng, and J. Han, Generalization and Decision Tree Induction: Efficient Classification in Data Ming, *Proc. of 1997 Int'l Workshop on Research Issues on Data Engineering (RIDE'97)*, PP. 111-120, Birmingham, England, April 1997.
5. Agrawal, R., et al, An Interval Classifier for Database Mining Application, *Proceedings of the 18th VLDB Conference*, Vancouver, British Columbia, Canada, 1992.
6. Mehta, M., R. Agrawal, and J. Rissanen, SLIQ : A Fast Scalable Classifier for Data Mining, *In Proc. 1996 Intl. Conf. On Ectending Database Technology(EDBT96)*, Avigon, France , March, 1996.
7. Shafer, J., R. Agrawal, and M. Mehta, SPRINT: A Scalable Parallel Classifier for Data Mining, *In Proc. 22nd Intl. Conf. Very Large Data Bases(VLDB)*, PP. 544-555, Mumbai, India, 1996.
8. Dougherty, J., R. Kohavi, and M. Sahami, Supervised and unsupervised discretization of continuous features, *Proc. European Working Session on Learning*, PP. 194-202, San Francisco, Morgan Kaufmann, 1995.
9. Weiss, M. S. and A. C. Kulikowski, *Computer Systems That Learn*, Morgan Kaufmann, 1991.
10. Brieman, et. al, *Classification and Regression Trees*, Wadsorth, 1984.