재귀적 분할 평균에 기반한 점진적 규칙 추출 알고리즘

(An Incremental Rule Extraction Algorithm Based on Recursive Partition Averaging)

한 진 철 † 김 상 귀 † 윤 충 화 ††
(Jin-Chul Han) (Sang-Kwi Kim) (Chung-Hwa Yoon)

요 약 패턴 분류에 많이 사용되는 기법 중의 하나인 메모리 기반 추론 알고리즘은 단순히 메모리에 저장된 학습패턴 또는 초월평면과 테스트 패턴간의 거리를 계산하여 가장 가까운 학습패턴의 클래스로 분류하기 때문에 테스트 패턴을 분류하는 기준을 설명할 수 없다는 문제점을 가지고 있다. 이 문제를 해결하기 위하여, 메모리 기반 학습 기법인 RPA를 기반으로 학습패턴들에 내재된 규칙성을 표현하는 IF-THEN 형태의 규칙을 생성하는 점진적 학습 알고리즘을 제안하였다. 하지만, RPA에 의해 생성된 규칙은 주어진 학습패턴 집합에만 충실히 학습되어 overfitting 현상을 보이게 되며, 또한 패턴 공간의 과도한 분할로 인하여 필요 이상으로 많은 개수의 규칙이 생성된다. 따라서, 본 논문에서는 생성된 규칙으로부터 불필요한 조건을 제거함으로써overfitting 현상을 해결함과 동시에 생성되는 규칙의 개수를 줄일 수 있는 점진적 규칙 추출 알고리즘을 제안하였으며, UCI Machine Learning Repository의 벤치마크 데이타를 이용하여 제안한 알고리즘의 성능을 입증하였다.

키워드 : 규칙 생성, Overfitting 문제, 점진적 학습 알고리즘

Abstract One of the popular methods used for pattern classification is the MBR (Memory-Based Reasoning) algorithm. Since it simply computes distances between a test pattern and training patterns or hyperplanes stored in memory, and then assigns the class of the nearest training pattern, it cannot explain how the classification result is obtained. In order to overcome this problem, we propose an incremental learning algorithm based on RPA (Recursive Partition Averaging) to extract IF-THEN rules that describe regularities inherent in training patterns. But rules generated by RPA eventually show an overfitting phenomenon, because they depend too strongly on the details of given training patterns. Also RPA produces more number of rules than necessary, due to over-partitioning of the pattern space. Consequently, we present the IREA (Incremental Rule Extraction Algorithm) that overcomes overfitting problem by removing useless conditions from rules and reduces the number of rules at the same time. We verify the performance of proposed algorithm using benchmark data sets from UCI Machine Learning Repository.

Key words : Rule Extraction, Overfitting Problem, Incremental Learning Algorithm

1. 서 론

메모리 기반 추론 기법은 단순히 학습패턴이나 초월평면의 형태로 메모리에 저장하며, 테스트 패턴과의 거리 계산을 통하여 분류 결과를 산출하므로 거리 기반 학습(Distance-Based Learning)이라고도 한다[1]. 그러나, 메모리 기반 추론 기법은 분류 결과만을 제공할 뿐이며, 테스트 패턴을 분류하는 기준을 설명할 수 없다는 문제점을 가지고 있다. 반면에, 메모리 기반 추론 범주에 속하는 RPA(Recursive Partition Averaging)는 모든 분할영역에 클래스가 동일한 학습패턴들만 남을 때까지 재귀적으로 분할을 수행하며, 분할영역의 학습패턴들을 이용하여 대표패턴을 구성하는 기법이다. 이때, 각 분할영역은 규칙으로 변환이 가능하며, 본 논문에서는 RPA를 기반으로 규칙을 생성하는 방법을 제안한다. 한

† 학생회원 : 명지대학교 컴퓨터공학과
jchan0415@mju.ac.kr
kimsk98@mju.ac.kr
†† 정 회 원 : 명지대학교 컴퓨터공학과 교수
yoonch@mju.ac.kr
논문접수 : 2006년 2월 1일
심사완료 : 2006년 11월 9일

편, 일반적으로 모든 기계학습 기법들은 주어진 학습패턴 집합에만 충실히 학습되어 overfitting 현상이 나타난다. 이러한 overfitting 현상은 규칙을 prune함으로써 해결할 수 있으며, pruning에는 pre-pruning과 post-pruning 방법이 있다[2]. Pre-pruning은 규칙을 생성하는 도중에 일반화 성능을 떨어뜨리는 조건을 규칙에 추가하지 않는 방법을 의미하며, post-pruning은 학습패턴을 이용하여 규칙을 생성한 다음, 분류 성능을 저하시키는 조건을 규칙으로부터 제거하는 방법이다.

기존의 규칙 생성 기법에는 C4.5[3-5], PRISM[6], RIPPER[7,8]와 PART[9] 등이 있다. 결정 트리 기법인 C4.5는 트리를 완벽하게 생성한 다음 post-pruning 과정을 거쳐 트리의 구조를 단순하게 변환한 후, 루트 노드로부터 각 단말 노드까지의 경로를 규칙으로 생성하는 기법이다. 반면에, PART는 결정 트리 기반이지만 트리를 생성하는 과정에 pre-pruning을 수행하며, 최종적으로 coverage가 큰 단말 노드를 규칙으로 생성한다. 한편, PRISM과 RIPPER는 주어진 학습패턴을 이용하여 조건부를 추가해가며 규칙을 생성하는 방법이며, 생성된 규칙에 대해서 post-pruning과정을 통해 일반화 성능을 저하시키는 조건을 제거한다. 그러나, 본 논문에서는 기존의 규칙 생성 기법들과는 달리, 메모리 기반 학습 기법인 RPA를 이용하여 규칙을 생성한 다음, 규칙의 일반화 성능을 향상시키기 위한 post-pruning을 수행함과 동시에 생성되는 규칙의 개수를 줄일 수 있는 점진적 규칙 추출 알고리즘(IREA: Incremental Rule Extraction Algorithm)을 제안하였다. 그리고, UCI Machine Learning Repository의 벤치마크 데이타를 사용하여 제안한 알고리즘의 성능을 기존 규칙 생성 기법과 비교 검증하였다.

2. 관련연구

기존의 규칙 생성 기법들은 크게 divide-and-conquer와 separate-and-conquer범주로 나눌 수 있다. Divide-and-conquer 는 전체 패턴공간을 작은 공간으로 분할하고 분할된 각 공간을 재귀적으로 해결하는 방식이며, separate-and-conquer 방식은 주어진 학습패턴집합을 이용하여 규칙을 생성하고 생성된 규칙이 처리하는 학습패턴을 학습패턴집합으로부터 제거하며, 남은 학습패턴을 다음 규칙을 생성하기 위하여 사용하는 방식이다.

C4.5는 최초 결정 트리 알고리즘인ID3를 확장한 형태로 실수 처리가 가능하며, post-pruning과정을 거쳐 규칙의 일반화 성능을 향상시킨 divide-and-conquer 형태의 알고리즘이다[3-5]. C4.5는 정보이론을 기반으로 모든 단말 노드가 동일한 클래스의 학습패턴으로만 구성될 때까지 재귀 분할하면서 결정 트리(Decision Tree)를 구성한 다음, 생성된 결정 트리에 subtree replacement와 subtree raising 기법을 적용하여 트리를 prune하고, 루트 노드로부터 단말 노드까지의 경로를 규칙으로 추출하며, 데이타 마이닝 분야에서 가장 널리 사용되고 있는 기법이다.

PRISM은 separate-and-conquer 범주에 속하며, "IF ? THEN class = C" 형태의 초기 규칙에 새로운 조건을 계속 추가해가는 방법이며, 어느 조건을 추가할지를 결정하기 위하여 accuracy와 coverage 개념을 이용한다[6]. Coverage는 규칙의 조건부를 만족하는 학습패턴의 개수이며, accuracy는 규칙의 조건부와 결론부를 모두 만족하는 학습패턴의 개수를 coverage로 나눈 값이다. 그리고, 생성된 규칙으로부터 류율이 더 이상 낮아지지 않을 때까지 조건들을 하나씩 제거하는 방법으로 규칙을 prune한다.

RIPPER(Repeated Incremental Pruning to Produce Error Reduction)는 IREP(Incremental Reduced Error Pruning)의 단점인 실수 데이타와 멀티 클래스 처리를 보완한 알고리즘이며[7,8], reduced error pruning과 separate-and-conquer 개념을 이용하여 일반화된 규칙을 생성한다. RIPPER는 PRISM과 달리 학습패턴 집합을 growing set과pruning set으로 나누어, growing set으로 규칙을 생성한 다음, pruning set을 이용하여 규칙의 일반화 성능을 저해하는 조건들을 제거한다[10]. 이때, 생성된 규칙의 pruning set에 대한 오류율이 50%를 초과하면 규칙 생성 과정을 종료한다.

PART(Partial Decision Tree)는 divide-and-conquer 방식으로 결정트리를 생성하며, separate-and-conquer 방식으로 규칙을 추출하는 하이브리드 기법이다[9]. 그러나, C4.5와는 달리 완전한 형태의 트리를 구성하지 않고 필요한 노드만을 확장하여 부분 트리(partial tree)를 생성하며, 이때 pre-pruning 과정을 거친다. 그리고, coverage가 가장 큰 단말노드를 규칙으로 변환하며, 이 규칙으로 분류 가능한 학습패턴들을 제거하고, 알고리즘을 재귀 호출한다.

3. 점진적 규칙 추출 알고리즘(IREA: Incremental Rule Extraction Algorithm)

본 논문에서는 메모리 기반 추론 기법인 RPA를 이용하여 규칙을 생성하는 알고리즘을 제안하였다. 한편, 일반적으로 모든 기계학습 기법은 주어진 학습패턴 집합만을 대상으로 학습을 수행하므로 overfitting 현상이 발생한다는 문제점을 갖고 있다. 특히, RPA에서는 모든 분할영역에 동일한 클래스에 속하는 학습패턴만이 남을 때까지 재귀적으로 분할하므로, overfitting 현상이 필연적으로 발생하게 되며, 또한 패턴공간의 과도한 분할을

이유로 생성되는 규칙의 개수가 과다한 현상이 발생된다. 따라서, 본 논문에서는 이 문제를 해결하기 위하여 생성된 규칙으로부터 불필요한 조건을 제거하는 규칙 pruning 알고리즘과 생성되는 규칙의 개수를 줄이기 위하여 점진적 학습 알고리즘을 제안하였다.

3.1 RPA 기반 규칙 추출

RPA는 모든 분할영역이 동일한 클래스에 속하는 학습패턴들로 구성될 때까지 재귀적으로 분할한 다음, 각 분할영역에 인스턴스 평균(Instance Averaging)법을 적용하여 대표패턴을 생성하는 알고리즘이며, 특징간의 영향력을 평준화하기 위하여 학습 개시 이전에 모든 특징을 정규화하며, 또한 테스트 패턴에 대한 분류 정확도를 높이기 위하여 특징 가중치 값을 사용한다[11].

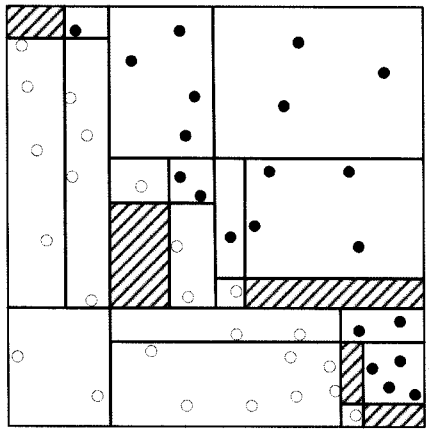
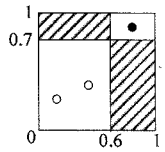


그림 1 RPA의 패턴공간 분할

그림 1은 패턴공간이 RPA 기법에 의해 재귀적으로 분할된 예제이며, 총 17개의 대표패턴이 생성되며, 빗금친 분할영역에는 학습패턴이 없으므로 대표패턴을 생성하지 않는다. 각 분할영역은 특징 별로 하한 값과 상한 값으로 표현되며, 이는 분할영역을 규칙의 형태로 생성할 수 있다는 것을 의미한다. 예를 들어,



○ : Class 1 ● : Class 2

그림 2 패턴 분할 공간

그림 2와 같은 분할이 수행되면, 하나 이상의 학습패턴을 포함하는 모든 분할영역을 규칙으로 변환한다. 이때, 두 개의 규칙이 생성되며 규칙의 형태는 다음과 같다.

IF 0 <= x < 0.6 AND 0
<= y < 0.7 THEN class = 1
IF 0.6 <= x < 1 AND 0.7
<= y < 1 THEN class = 2

한편, 그림 2의 빗금친 분할영역에는 학습패턴이 존재하지 않으므로 규칙으로 변환되지 않으며, 이로 인하여 차후에 규칙만으로 분류가 불가능한 테스트 패턴이 발생할 가능성이 있다. 이러한 테스트 패턴을 분류하기 위하여 하나 이상의 패턴을 포함하는 모든 분할영역에 대한 대표패턴을 구하여, 생성된 규칙과 함께 저장한다.

3.2 규칙 Pruning 알고리즘

규칙을 prune하기 위해서는 어느 조건이 불필요한 조건인지를 결정해야 되며, PRISM, RIPPER의 경우에는 규칙에 새로운 조건을 추가하면서 규칙을 생성하기 때문에 추가된 역순으로 조건을 제거하여 규칙을 pruning한다. 반면에, RPA의 특성상 생성된 규칙의 조건들에 추가된 순서를 정의할 수 없으므로, 규칙을 pruning하기 위해서 어느 조건을 먼저 제거할 지를 결정해야 하며, 본 논문에서는 IG(Information Gain)값을 기준으로 제거할 조건을 선정한다. 이때, 제거할 조건을 선택하기 위한 IG값은 수식 (1), (2)를 이용하여 계산한다.

$$I = -\sum_{i=1}^{C} p_i \log_2 p_i \tag{1}$$

p_i는 학습패턴 집합에서 클래스 i에 소속되는 패턴의 비율이며, C는 클래스의 개수이다.

$$IG(f) = I - \sum_{i=1}^{N} P_i I_i \tag{2}$$

P_i는 분할 이전의 학습패턴 중 분할된 영역에 포함된 학습패턴의 비율이다. I는 분할 이전의 정보량, I_i는 각 분할점들을 기준으로 분할했을 때 각 공간의 정보량이며, 이들은 수식 (1)을 이용하여 계산된다. 또한, N은 특징 f의 분할영역 개수이다.

규칙의 모든 조건에 대해서 IG 값을 계산하여 가장 낮은 IG 값을 가지는 조건-즉, 분류에 미치는 영향력이 가장 작은 조건-을 선택한 다음, 해당 조건을 제거하기 이전의 규칙과 제거한 이후의 규칙에 대한 질(quality)을 평가하기 위해 수식 (3), (4)를 이용하여 PM(Probability Measure) 값을 계산한다. PM 값은 임의로 생성한 규칙의 분류 성능이 특정 규칙의 성능보다 좋을 확률을 의미하며, PM 값이 작을수록 규칙의 질은 좋은 것으로 간주된다[2].

$$\Pr(i) = \frac{\binom{P}{i}\binom{T-P}{t-i}}{\binom{T}{t}} \tag{3}$$

$$PM(R) = \sum_{i=p}^{\min(t,P)} \Pr(i) \quad (4)$$

T는 전체 학습패턴의 개수이며, P는 전체 학습패턴중 규칙 R의 클래스에 속하는 학습패턴의 개수이다. t는 규칙 R의 조건부를 만족하는 학습패턴의 개수이며, p는 규칙 R의 조건부와 결론부를 모두 만족하는 학습패턴의 개수이다.

규칙을 prune하는 알고리즘은 표 1과 같다.

표 1 규칙 pruning 알고리즘

① 규칙 R의 조건부에 포함된 조건 중, IG 값이 가장 작은 조건을 선택한다.
② $PM(R-) < PM(R)$의 관계가 성립되면 그 조건을 제거하고 단계 ①로 간다. ($R-$은 선택된 조건을 제거한 규칙이며, R은 조건을 제거하기 이전 규칙이다.)
③ $PM(R-) >= PM(R)$의 관계가 성립되면 규칙의 pruning을 종료한다.

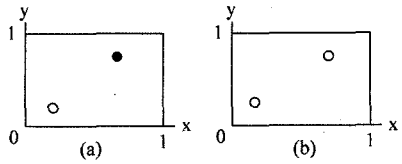
3.3 점진적 학습 알고리즘

RPA를 기반으로 규칙을 생성하면, 3.2절에서 설명한 바와 같이 학습한 패턴 집합에만 국한된 overfitting 현상이 발생할 뿐만 아니라, 패턴 공간의 과도한 분할로 인하여 필요 이상으로 많은 개수의 규칙이 생성되는데, 이 문제를 해결하기 위하여 본 논문에서는 점진적으로 학습패턴 집합을 처리하였다. RPA에 의해 생성된 모든 규칙에 대하여 post-pruning 과정을 수행한 후, 가장 정확도가 높은 규칙을 선정하고, 그 규칙의 조건부를 만족하는 모든 학습 패턴을 제거한 다음, 남은 학습패턴들을 재귀적으로 처리한다. 이와 같은 방법으로 생성되는 규칙의 개수를 줄이기 위한 점진적 규칙 추출 알고리즘(IREA)은 다음의 표 2와 같다.

표 2 점진적 규칙 추출 알고리즘

① RPA 기법을 수행하여 학습패턴 집합을 학습한다.
② 학습패턴을 포함하는 모든 분할영역을 규칙으로 변환하여 저장한다.
③ 생성된 모든 규칙에 대해서 표 1의 규칙 pruning 알고리즘을 적용하고, accuracy가 가장 큰 규칙을 선정한다. (단, accuracy가 동일한 경우, coverage가 가장 큰 규칙을 선정한다.)
④ 선정된 규칙의 조건부를 만족하는 모든 학습패턴을 제거하고, 규칙과 coverage값을 저장한다.
⑤ 더 이상 학습할 패턴이 없을 때까지 단계 ①-④를 반복한다.

제안한 점진적 규칙 추출 알고리즘은 매번 반복 수행될 때마다 학습패턴 집합의 크기가 작아지는 separate-and-conquer범주에 속하며, 특기할 사항으로는 학습이 종료되기 직전에 다음과 같은 문제점이 발견되었다.



○ : Class 1 ● : Class 2

그림 3 분할 영역

그림 3(a)의 경우에는 한 분할영역에 클래스가 다른 패턴들이 존재하므로 재귀적으로 추가 분할되는 반면에, 그림 3(b)의 경우에는 학습 패턴들의 클래스가 동일하므로 더 이상 분할이 일어나지 않고 학습이 종료되며, 다음과 같은 형태의 규칙이 생성된다.

IF $0 < x < 1$ AND $0 < y < 1$ THEN class = 1

이때, 이 규칙의 조건부는 각 특징의 전체 범위를 나타내므로, 모든 테스트 패턴에 적용할 수 있으며, 이는 전체 테스트 패턴 집합에 대한 오분류율을 높이는 결과를 초래할 수도 있으므로, 이러한 규칙은 제거하여 최종 규칙 집합에 포함시키지 않는다.

4. 분류 알고리즘

일반적으로 패턴을 규칙으로 분류하는 알고리즘에는 다음과 같은 현상이 발생할 수도 있다: 특정 테스트 패턴에 두 개 이상의 규칙이 적용되거나, 적용 가능한 규칙을 찾을 수 없는 경우가 발생할 수 있다. 이때, C4.5, PART와 RIPPER의 경우, 알고리즘의 특성상 모든 규칙에 우선순위가 정의된 Decision List의 형태를 갖게 되므로 위와 같은 경우는 발생하지 않는다. 그러나, PRISM과 본 논문에서 제안한 IREA에서는 규칙의 우선 순위가 정의되지 않으므로 위의 현상이 발생할 수도 있다. PRISM에서는 이러한 경우를 처리하기 위하여 테스트 패턴에 적용 가능한 규칙의 개수를 검색하고, 규칙의 개수가 가장 많은 클래스로 분류하며, 만약 규칙의 개수가 동일한 경우, coverage가 큰 규칙의 클래스로 분류한다. 한편, 적용 가능한 규칙이 없는 경우에는 가장 많은 학습패턴이 소속된 클래스(majority class)로 테스트 패턴을 분류한다.

본 논문에서 제안한 IREA의 분류 알고리즘은 다음의 표 3과 같다. 테스트 패턴에 적용 가능한 규칙(들)이 발견되었을 때, 규칙들의 coverage를 합산하여 합이 가장 큰 클래스로 분류하며, 적용 가능한 규칙이 없는 경우에는 각 규칙에 저장된 대표패턴들과 거리를 계산하여 가장 가까운 대표패턴의 클래스로 분류한다. 이때 특징 가중치 값을 거리 계산에 이용하며, 특징 가중치 값은 3.2절의 수식 (1), (2)로 계산된 값을 사용한다[11]. 또한, IREA는 RPA를 여러 번 수행하여 점진적으로 규칙을

추출하지만, 분류에 사용되는 특징 가중치 값은 최초 RPA를 수행할 때 계산된 IG값을 고정적으로 사용하며, 거리 계산은 수식 (5)를 이용한다.

$$D = \sqrt{\sum_{i=1}^{n} W_i (E_i - Q_i)^2} \quad (5)$$

W_i는 특징 i의 가중치 값이며, E_i와 Q_i는 대표패턴과 테스트 패턴의 i번째 특징값이다. n은 패턴의 특징개수를 의미한다.

표 3 분류 알고리즘

① 생성된 규칙 집합에서 테스트 패턴에 적용 가능한 규칙(들)을 찾는다.
② 적용 가능한 규칙이 두 개 이상이고 분류 클래스가 다른 경우, 각 클래스별로 해당 규칙의 coverage를 합산하고, 합이 가장 큰 클래스로 분류한다.
③ 적용 가능한 규칙이 없으면, 각 규칙의 대표패턴과 테스트 패턴간의 거리를 계산하여 가장 가까운 대표패턴의 클래스로 분류한다.

5. 실험 결과

본 논문에서 제안한 IREA와 PRISM, RIPPER, C4.5, PART의 분류 성능 및 규칙 개수를 비교하였으며, 실험 방법은 stratified 10-fold cross-validation 기법을 사용하였다[2].

5.1 실험 데이타

본 논문에서는 기계 학습의 벤치마크 자료로 많이 사용되는 UCI Machine Learning Repository에서 Breast-Cancer-Wisconsin, Glass, Ionosphere, Iris, New-thyroid, Wine 데이타 셋을 사용하였으며, 모든 특징이 실수 값으로 구성되어 있다[12,13]. 다음의 표 4는 실험 데이타 셋의 특성을 나타낸다.

5.2 분류 성능

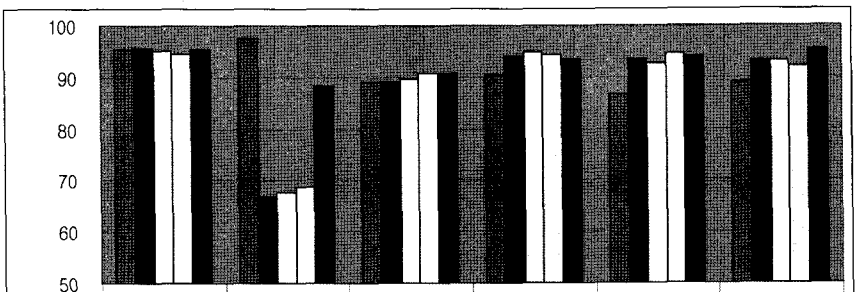
그림 4에 나타난 바와 같이, 본 논문에서 제안한 IREA의 분류 성능은 다른 알고리즘과 유사하거나 높은 것을 볼 수 있다. 그러나, 클래스 1, 2에 많은 패턴들이 편중되어 있는 Glass 데이타 셋에 대한 PRISM의 분류 성능은 IREA 보다 높은 것을 볼 수 있는데, 그 이유는, 규칙으로 분류가 불가능한 테스트 패턴을 majority class(클래스 2)로 처리하는 PRISM의 특성에 기인한 것으로 사료된다.

다음의 표 5는 분류 성능에 대한 표준편차를 나타낸다.

표 5에 나타난 바와 같이, IREA 의 표준편차를 다른 기법들과 비교할 때, 실험을 수행한 6개 데이타 셋에 대

표 4 데이타셋의 패턴 분포

데이타 셋	패턴 개수	특징 개수	클래스별 패턴 수					
			1	2	3	4	5	6
Breast-Cancer Wisconsin	699	10	458	241	-	-	-	-
Glass	214	10	70	76	17	13	9	29
Ionosphere	351	34	225	126	-	-	-	-
Iris	150	4	50	50	50	-	-	-
New-Thyroid	215	5	150	35	30	-	-	-
Wine	178	13	59	71	48	-	-	-

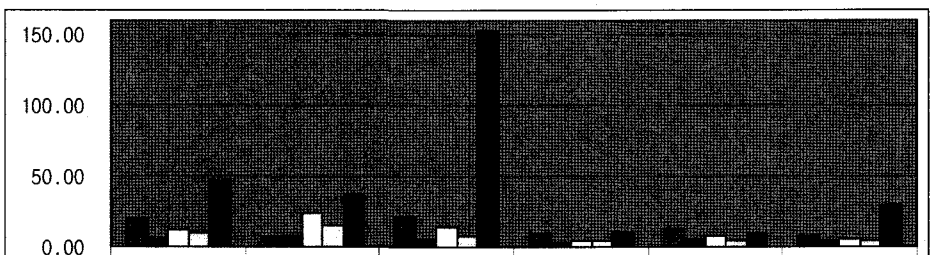


	Breast-cancer	Glass	Ionosphere	Iris	New-thyroid	Wine
■ PRISM	95.47	97.56	89.26	90.40	86.71	89.10
■ RIPPER	95.61	66.78	89.16	93.93	93.55	93.14
□ C4.5	95.01	67.63	89.74	94.73	92.62	93.20
□ PART	94.69	68.75	90.83	94.20	94.52	92.08
■ IREA	95.37	88.44	90.82	93.40	94.09	95.32

그림 4 분류 성능

표 5 분류 성능에 대한 표준 편차

	Breast-cancer	Glass	Ionosphere	Iris	New-thyroid	Wine
PRISM	3.02	3.30	5.28	6.95	5.58	8.21
PART	2.51	10.6	4.66	5.25	4.75	6.28
C4.5	2.73	9.31	4.38	5.30	5.60	5.90
RIPPER	2.22	9.65	4.64	5.77	4.96	6.94
IREA	2.52	6.89	4.37	5.61	4.74	4.92



	Breast-cancer	Glass	Ionosphere	Iris	New-thyroid	Wine
■PRISM	19.91	6.00	20.47	10.21	13.07	7.50
■RIPPER	5.66	7.85	5.16	3.59	5.06	4.02
□C4.5	12.23	23.58	13.87	4.64	7.96	5.30
□PART	9.93	15.25	7.53	3.85	4.32	4.46
■IREA	47.05	36.90	151.95	9.51	10.21	29.33

그림 5 규칙 개수

하여 고른 값을 갖는다는 것을 볼 수 있으며, 이는 IREA가 특정 데이타 셋과 무관한 안정적인 기법이라는 사실을 나타낸다.

5.3 규칙 개수

그림 5에 나타난 대로 IREA 로 생성되는 규칙의 개수가 다른 기법들보다 전반적으로 많은 것을 볼 수 있다. 특히, 특징의 개수가 많은 Ionosphere의 경우, RPA 기법의 특성상 전체 패턴공간의 과도한 분할이 일어나기 때문으로 사료된다.

6. 결 론

메모리 기반 추론 기법은 단순히 학습패턴이나 초월평면의 형태로 메모리에 저장하며, 테스트 패턴과의 거리 계산을 통하여 분류 결과를 산출한다. 결국, 메모리 기반 추론 기법은 분류 결과만을 제공할 뿐이며, 테스트 패턴을 분류하는 기준을 설명할 수 없다는 문제점을 가지고 있다. 본 논문에서는 기존의 규칙 생성 알고리즘들과는 달리 메모리 기반 추론 기법인 RPA를 이용하여 규칙을 생성하는 점진적 규칙 추출 알고리즘(IREA)를 제안하였다. 이때, RPA의 특성상 모든 분할영역이 동일한 클래스에 속하는 패턴들로 구성될 때까지 재귀적으로 분할하므로 많은 개수의 규칙이 생성되는데, IREA는 점진적으로 학습을 수행하여 이 문제에 대한 해결 방안을 모색하였으며, 또한 일반적으로 모든 기계학습 기법에 발생하는 overfitting 현상을 해결하기 위해서 규칙 pruning 작업을 수행하였다.

본 논문에서 제안한 IREA는 제반 다른 기법과 비교하여 우수하거나 유사한 분류 성능을 보여주며, 특히 분류 성능이 고른 분산을 보여준다는 사실은 특정 데이타 셋과 무관한 안정적인 기법이라는 것을 입증하였다. 하지만, 특징 개수가 많은 데이타 셋의 경우에는 분할영역의 과도한 분할로 인하여 생성된 규칙의 개수가 많아지는 단점을 발견하였다. 향후 연구에서는 전처리 과정을 통하여 학습에 사용되는 특징의 개수를 줄이는 방법과 재귀호출과정에서 발생되는 과도한 분할을 방지하는 방법, 그리고 규칙의 질(Quality)을 평가하는 새로운 방법에 대하여 연구를 계속할 예정이다.

참 고 문 헌

[1] T. Dietterich, "A Study of Distance-Based Machine Learning Algorithms," Ph. D. Thesis, computer Science Dept., Oregon State University, 1995.

[2] Ian H. Witten, Eibe Frank, Data Mining, Morgan Kaufmann, 1999.

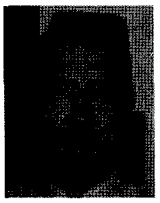
[3] J. R. Quinlan, "Simplifying Decision Trees," Knowledge Acquisition for Knowledge-Based Systems, pp.239-252, Academic Press, 1988.

[4] J. R. Quinlan," Induction of decision trees," Machine Learning, Vol.1, No.1, pp.81-106, 1986.

[5] J. R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[6] J. Cendrowska, "PRISM : An Algorithm for inducing modular rules," International Journal of Man-Machine Studies, 27(4): pp.349-370, 1987.
[7] Johannes Fürnkranz, Gerhard Widmer, "Incremental Reduced Error Pruning," Proceedings of the 11th International Conference on Machine Learning, Morgan Kaufmann, pp.70-77, 1994.
[8] Cohen, W. W., "Fast effective rule induction," In Proceedings of the 12th International Conference on Machine Learning, pp. 115-123, Morgan Kaufmann, 1955.
[9] Eibe Frank, Ian H. Witten, "Generating accurate rule sets without global optimization," Proc. 15th International Conference on Machine Learning, pp. 144-151, Morgan Kaufmann, San Francisco, CA, 1998.
[10] Rissanen, J., "Modelling by shortest data description," Automatica, 14, pp.45-471, 1978.
[11] 이형일, 정태선, 윤충화, 강경식, "재귀분할 평균법을 이용한 새로운 메모리기반 추론 알고리즘," 한국정보처리학회 논문지, Vol.6, No.7, pp.1849-1857, 1999.
[12] D. Aha, "A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and psychological Evaluations," Ph. D. Thesis, Information and Computer Science Dept., University of California, Irvine, 1990.
[13] Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J., "UCI Repository of machine learning databases," Irvine, CA: University of California, Department of Information and Computer Science, 1998 [http://www.ics.uci.edu/~mlearn/MLRepository.html]

한 진 철
1998년 명지대학교 컴퓨터공학과 졸업(공학사). 2000년 명지대학교 일반대학원 컴퓨터공학과 졸업(공학석사). 2004년 명지대학교 일반대학원 컴퓨터공학과 박사수료. 2002년~현재 명지대학교 교양 시간강사. 2006년~현재 명지대학교 산업기술연구소 전임연구원. 관심분야는 인공지능, 지능형 소프트웨어, 데이타마이닝, 시멘틱웹

김 상 귀
1990년 명지대학교 전자계산학과 졸업(학사). 1993년 명지대학교 대학원 전자계산학과 졸업(공학석사). 1995년~1998년 명지대학교 대학원 컴퓨터공학과(박사수료). 1998년~2002년 세경대학 컴퓨터소프트웨어과 전임강사. 2002년~현재 디지털 C&P 정보관리부 차장. 2002년~현재 명지대학교 컴퓨터공학과 겸임교수. 관심분야는 인공지능, 지능형 소프트웨어, 데이타마이닝, 패턴인식



윤 충 화
1979년 9월 서울대학교 자연과학대학 수학과 학사. 1984년 9월 University of Texas at Austin 전산학과 석사. 1989년 7월 Louisiana State University 전산학과 박사. 1990년 3월~현재 명지대학교 공과대학 컴퓨터공학과 교수. 현 데이타마이닝 학회 이사. 관심분야는 인공지능, 지능형 소프트웨어, 데이타마이닝