# 새로운 규칙 생성 알고리즘

김상귀° 윤충화
명지대학교 컴퓨터공학과
{kimsk98°, yoonch}@mju.ac.kr

# A New Rule-Generation Algorithm

Sang-kwi Kim° Chung-hwa Yoon
Dept. of Computer Engineering, Myongji University

요 약

패턴 분류에 많이 사용되는 MBR(Memory Based Reasoning) 기법은 메모리에 저장된 학습패턴과 테스트 패턴간의 거리를 계산하여 가장 가까운 학습패턴의 클래스로 분류하기 때문에 테스트 패턴을 분류하는 기준을 설명할 수 없다는 문제점을 가지고 있다. 본 논문에서는 RPA(Recursive Partition Averaging) 기법을 이용하여 분류 기준을 설명할 수 있는 IF-THEN 형태의 규칙을 생성하고 생성된 규칙의 일반화 성능을 향상시키기 위하여 불필요한 조건을 제거하는 규칙 pruning 알고리즘과 생성되는 규칙의 개수를 줄일 수 있는 점진적 규칙 추출 알고리즘을 제안한다.

## 1. 서론

메모리 기반 추론 기법은 학습패턴 전체를 단순히 메모리에 저장한 다음, 테스트 패턴과의 거리를 계산하여 분류하므로 거리 기반 학습(Distance-Based Learning) 이라고도 한다[1]. 그러나 메모리 기반 추론 기법은 테스트 패턴을 분류하는 기준을 설명할 수 없는 문제점을 가지고 있다.

본 논문에서는 RPA(Recursive Partition Averaging) 기법을 이용하여 분류 기준을 설명할 수 있는 IF-THEN 형식의 규칙 생성 알고리즘을 제안한다. 하지만, RPA 기법은 모든 분할영역에 동일 클래스에 속하는 학습패턴들만 남을 때까지 재귀적으로 분할을 수행하므로 학습패턴에만 충실한 학습이 수행되어 overfitting 현상이 발생한다. 이러한 overfitting 현상을 해결하기 위하여 규칙에서 불필요한 조건을 제거하여 일반화 성능을 향상시키기 위한 규칙 pruning 알고리즘과 생성되는 규칙의 개수를 줄이기 위한 점진적 규칙 추출 알고리즘을 제안한다. 제안한 알고리즘의 성능은 UCI Machine Learning Repository의 벤치마크 데이터를 이용하여 대표적인 규칙 생성 알고리즘인 PRISM 기법과 실험적으로 비교 검증하였다.

## 2. PRISM 기법

PRISM 기법은 "IF ? THEN class = C" 형태의 초기 규칙에 새로운 조건을 계속적으로 추가해 가는 방법이며, 어느 조건을 추가할지를 결정하기 위하여 accuracy와 coverage개념을 이용한다[2]. Coverage는 규칙의 IF절을 만족하는 학습패턴의 개수이며, accuracy는 규칙의 IF와 THEN절을 모두 만족하는 학습패턴의 개수를 coverage로 나눈 값이다.

PRISM 기법은 학습패턴의 특징이 실수인 경우, 먼저 특징 값들을 오름차순으로 정렬하고 특징 값이 변화하는 곳에서 경계 값을 정의한다. 만약 특징 x의 값이 68, 70인 경우, 경계 값 69가 선정되며 조건 'x > 69', 'x <= 69' 이 만들어 진다. 학습패턴의 모든 특징에 대해서 동일한 과정으로 조건들을 선정하고 그 중 가장 accuracy가 높은 조건을 규칙에 추가한다. 만약 조건 'x > 69' 을 선택했을 때 accuracy가 가장 높다면, "IF x > 69 THEN class = C" 라는 규칙이 생성된다.

PRISM 기법의 분류 과정은 각 클래스 별로 적용 가능한 규칙의 개수를 검색하고, 규칙의 개수가 가장 많은 클래스로 분류한다. 만약 규칙의 개수가 동일하면, coverage가 가장 큰 규칙의 클래스로 분류한다. 또한, 적용 가능한 규칙이 없는 경우에는 학습패턴 집합에서 가장 많은 학습패턴이 소속된 클래스(majority class)로 테스트 패턴을 분류한다.

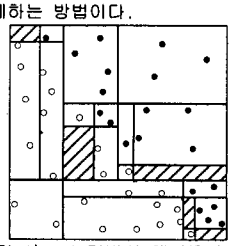
다음 <표 1>은 PRISM 기법을 설명한다.

<표 1> PRISM 기법

① 각 클래스(C)별로 단계 ②-⑧을 반복 수행한다.
② temp에 전체 학습패턴 집합을 저장한다.
③ IF 절이 비어있는 규칙(IF ? THEN class = C)을 생성한다.
④ 규칙의 IF절에 포함되지 않은 특징을 이용하여 가능한 모든 조건을 생성한다.
⑤ 생성된 조건들 중에서 규칙에 포함되었을 때 규칙의 accuracy를 가장 높게 하는 조건을 추가한다. (단, accuracy가 동일한 경우, coverage가 가장 큰 조건을 선택한다.)
⑥ 더 이상 추가할 특징이 없거나 accuracy가 100%가 될 때까지 단계 ④-⑤를 반복한다.
⑦ 생성된 규칙의 IF절을 만족하는 모든 학습패턴을 temp로부터 제거한다.
⑧ temp에 클래스가 C인 학습패턴들이 모두 제거될 때까지 단계 ③-⑦을 반복한다.

## 3. RPA 기법

RPA 기법은 모든 분할영역이 동일한 클래스에 속하는 학습패턴들로 구성될 때까지 재귀적으로 분할하고, 인스턴스 평균(Instance Averaging)법을 이용하여 대표패턴을 생성한다[3]. 인스턴스 평균법은 분할영역에 포함된 학습패턴들의 특징별 평균값을 계산하여 하나의 대표패턴으로 대체하는 방법이다.



(그림 1) RPA 기법의 패턴공간 분할

(그림 1)은 패턴공간이 RPA 기법에 의해 재귀적으로 분할된 예제이며, 총 17개의 대표패턴이 생성되고, 빗금친 분할영역은 학습패턴이

존재하지 않으므로 대표패턴이 생성되지 않는다. 또한, RPA 기법은 특징간의 영향력을 평준화하기 위하여 학습 개시 이전에 모든 특징 값들을 0과1사이의 값으로 정규화하며, 테스트 패턴에 대한 분류 정확도를 높이기 위하여 특징 가중치 값을 사용한다.

### 3.1 패턴공간의 분할

RPA 기법에서 분할이 발생할 때 모든 특징에 대한 분할점을 결정해야 하며, 특징의 분할점을 선택하기 위하여 특징 값을 오름차순으로 정렬하고 특징 값이 변화하는 위치를 경계 값으로 선정한다. 예를 들어, 70과 72 사이의 경계 값은 두 특징 값의 평균인 71이 된다.

구한 경계 값들 중에서 결정트리 알고리즘에서 특징의 분할점을 선정할 때 사용하는 *IG*(Information Gain) 값을 이용하여 가장 변별력이 좋은 경계 값을 분할점으로 선택한다[4]. *IG*값은 수식 (1), (2)를 이용하여 계산한다.

$$I = -\sum_{i=1}^{C} p_i \log_2 p_i \qquad (1)$$

$p_i$는 학습패턴 집합에서 클래스 $i$ 에 소속되는 패턴의 비율이며, $C$는 클래스의 개수이다.

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

$I$는 분할 이전의 정보량이며, $P_i$는 분할 이전의 학습패턴 중에서 분할된 영역에 포함된 학습패턴의 비율이다. $I_i$는 특정 경계 값 $f$를 기준으로 분할했을 때 분할된 각 공간의 정보량을 의미하며, 수식 (1)을 이용하여 계산된다.

여기에서 $I$값이 크다는 사실은 올바르게 분류하기 위하여 많은 양의 정보가 필요하다는 것을 의미하며, *IG*값은 분할 이전의 정보량과 경계 값을 기준으로 분할했을 경우 정보량의 차이를 의미한다. 즉, *IG*값은 분할 이후의 정보량이 작아질 경우에 큰 값을 가지게 되며, *IG*값이 큰 경계 값을 분할점으로 선택할 때 효율적인 분할이 가능하다.

다음 <표 2>는 RPA 기법의 알고리즘을 보여준다.

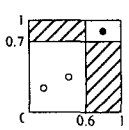
<표 2> RPA 기법

초기화 단계
① 전체 패턴 집합을 정규화한다.
② 패턴 집합을 학습패턴과 테스트 패턴 집합으로 분리한다.
③ 전체 학습패턴 집합을 포함하는 영역을 초기 분할영역으로 정의한 후, 다음의 학습 알고리즘을 적용한다.

학습 알고리즘
① 현재 분할영역에 포함된 모든 학습패턴의 클래스를 검사한다.
② 만약 모든 학습패턴의 클래스가 동일하면, 인스턴스 평균법으로 대표패턴을 추출하고 종료한다.
③ 만약 클래스가 다른 학습패턴이 존재하면, 현재 분할영역의 특징별로 새로운 경계 값을 결정하고, 이 중에서 가장 효율적인 경계 값을 분할점으로 선정한다.
④ 단계 ③에서 선정된 분할점을 이용하여 새로운 분할영역들을 구성한다.
⑤ 단계 ④에서 구성된 분할영역 중에서 하나 이상의 학습패턴을 포함하는 각 분할영역에 대하여 위의 학습 알고리즘을 재귀 호출한다.

### 3.2 규칙 추출과 대표패턴 생성

RPA 기법의 학습이 종료되면 하나 이상의 학습패턴이 존재하는 모든 분할영역을 규칙으로 생성하고 ORS(Original Rule Set)에 저장한다.




(그림 2) 패턴 공간 분할

(그림2)와 같은 분할이 수행되면, 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 2)의 빗금친 분할영역은 학습패턴이 존재하지 않으므로 규칙으로 생성되지 않으며, 이로 인하여 차후에 규칙만으로 분류가 불가능한 테스트 패턴이 발생할 가능성이 있다. 이러한 테스트 패턴을 분류하기 위하여 패턴을 포함하는 각 분할영역에 대하여 대표패턴을 구하여 생성된 규칙과 함께 저장한다.

### 3.3 규칙 pruning 알고리즘

RPA 기법은 모든 분할영역에 동일한 클래스에 속하는 학습패턴만 남을 때까지 재귀적으로 분할하므로 overfitting 현상이 발생하며, 본 논문에서는 이 문제를 해결하기 위하여 3.2절에서 생성된 규칙으로부터 불필요한 조건을 제거하는 규칙 pruning 알고리즘을 제안한다.

규칙에서 불필요한 조건을 제거하기 위해서는 어느 조건을 먼저 제거할 지를 결정해야 하며, 3.1절에서 설명한 *IG*값을 기준으로 제거할 조건의 순서를 결정한다. 규칙의 모든 조건에 대해서 *IG*값을 계산하여 가장 낮은 *IG*값을 가지는 조건-즉, 분류에 미치는 영향력이 가장 작은 조건- 순으로 <표 3>의 규칙 pruning 알고리즘을 적용한다. 제거할 조건을 선택하기 위한 *IG*값은 수식 (3)으로 계산한다.

$$IG(f) = I - \sum_{i=1}^{N} P_i I_i \qquad (3)$$

$P_i$는 분할 이전의 학습패턴 중 분할된 영역에 포함된 학습패턴의 비율이다. $I$는 분할 이전의 정보량, $I_i$는 각 분할점들을 기준으로 분할했을 때 각 공간의 정보량이며, 이들은 수식 (1)을 이용하여 계산된다. 또한, $N$은 특징 $f$의 분할영역 개수이다.

또한, 규칙의 질(quality)을 평가하기 위해서 수식 (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 질이 좋은 것으로 간주된다[4].

$$\Pr(i) = \frac{\binom{P}{i}\binom{T-P}{t-i}}{\binom{T}{t}}, \qquad PM(R) = \sum_{i=p}^{\min(t,P)} \Pr(i) \qquad (4)$$

$T$는 전체 학습패턴의 개수이며, $P$는 전체 학습패턴중 규칙 $R$의 클래스에 속하는 학습패턴의 개수이다. $t$는 규칙 $R$의 IF절을 만족하는 학습패턴의 개수이며, $p$는 규칙 $R$의 IF절과 THEN절을 모두 만족하는 학습패턴의 개수이다.

규칙을 pruning하는 알고리즘은 <표 3>과 같다.

<표 3> 규칙 pruning 알고리즘

① 규칙 $R$의IF절에 포함된 조건중에서, *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, $R$은 조건을 제거하기 이전의 규칙이다.)
③ $PM(R-) >= PM(R)$의 관계가 성립되면 종료한다.

### 3.4 점진적 규칙 추출 알고리즘

RPA 기법으로 규칙을 생성하는 경우, 학습패턴 집합에만 충실하게 학습되어 overfitting 현상이 발생할 뿐만 아니라, 과도한 분할로 인하여 필요 이상 많은 개수의 규칙이 생성된다. 그러므로 본 논문에서는 생성되는 규칙의 개수를 줄이기 위하여 <표 4>의 점진적 규칙 추출 알고리즘을 제안하였다.

<표 4> 는 점진적 규칙 추출 알고리즘을 보여준다.

<표 4> 점진적 규칙 추출 알고리즘

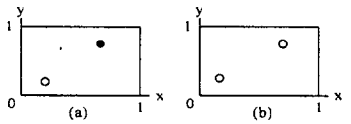
① RPA 기법으로 학습패턴 집합을 학습한다.
② 학습패턴을 포함하는 모든 분할영역을 규칙으로 변환하여 ORS (Original Rule Set)에 저장한다.
③ ORS의 모든 규칙에 대해서 <표 3>의 규칙 pruning 알고리즘을 적용하고, accuracy가 가장 높은 규칙을 선정한다. (단, accuracy가 동일한 경우, coverage가 가장 큰 규칙을 선정한다.)
④ 선정된 규칙의 IF절을 만족하는 모든 학습패턴을 제거하고, 규칙과 coverage값을 PRS(Pruned Rule Set)에 저장한다.
⑤ 더 이상 학습할 패턴이 없을 때까지 단계 ①-④를 반복한다.

한편, <표 4>의 점진적 규칙 추출 과정의 마지막 단계에서 (그림3, a)의 경우에는 재귀적으로 분할되는 반면에, (그림 3, b)의 경우에는

학습 패턴들의 클래스가 동일하므로 더 이상 분할이 일어나지 않고 학습이 종료되며, 다음과 같은 규칙이 생성된다.

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

이때, 이 규칙의 IF절은 각 특징의 전체 범위를 나타내며, 모든 테스트 패턴에 적용할 수 있기 때문에 테스트 패턴 집합에 대한 오분류율을 높게 하므로 최종 규칙 집합에 포함시키지 않는다.




(그림 3) 분할 영역

## 4. 분류 알고리즘

생성된 규칙 집합(PRS)에서 테스트 패턴에 적용 가능한 규칙이 없는 경우, 각 규칙에 저장된 대표패턴들과 거리를 계산하여 가장 가까운 대표패턴의 클래스로 분류한다. 이때 특징 가중치 값을 거리 계산에 이용하며, 특징 가중치 값은 3.3절에서 수식(3)으로 계산된 값을 사용한다[3]. 또한, 점진적 규칙 추출 알고리즘은 RPA 기법을 여러 번 수행하여 규칙을 추출하므로, 분류에 사용되는 특징 가중치 값은 최초 분할시 계산된 $IG$값을 사용하며, 거리 계산은 수식 (5)를 이용한다.

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

$W_i$는 $i$번째 특징의 가중치 값이며, $E_i$와 $Q_i$는 대표패턴과 테스트 패턴의 $i$번째 특징 값이다. $n$은 패턴의 특징 개수를 의미한다.

분류 알고리즘은 <표 5>와 같다.

<표 5> 분류 알고리즘

| |
|---|
| ① 규칙 집합에서 테스트 패턴에 적용 가능한 규칙(들)을 검색한다. |
| ② 적용 가능한 규칙이 없으면, 모든 규칙의 대표패턴과 거리를 계산하여 가장 가까운 클래스로 분류한다. |
| ③ 적용 가능한 규칙이 두 개 이상이고, 클래스가 다른 경우에는 각 클래스 별로 해당 규칙의 coverage를 합산하고 합이 가장 큰 클래스로 분류한다. |

## 5. 실험 결과

본 논문에서 제안한 점진적 규칙 추출 알고리즘과 ORS, PRISM의 분류 성능 및 생성된 규칙 개수를 비교 실험하였다. 실험 방법은 stratified 10-fold cross-validation 기법을 사용하였으며[5], 기계학습 분야의 벤치마크 자료로 많이 사용되는 UCI Machine Learning Repository에서 Breast-Cancer-Wisconsin, Glass, Ionosphere, Iris, New-thyroid, Wine 데이터를 발췌하여 사용하였다. 이들 데이터는 모든 특징이 실수 값으로 구성되어 있다[6].

### 5.1 규칙 분류 비중에 대한 실험

<표 6>과 같이 RPA 기법을 이용하여 생성된 ORS의 경우, 규칙으로 분류가 가능한 테스트 패턴의 비율이 그리 높지 않은 것을 볼 수 있다. 다시 말해, 규칙만으로 분류가 불가능한 경우, 거리 계산으로 테스트 패턴을 분류해야 하며, 이는 분류에 소요되는 계산시간을 증가시키는 결과를 초래한다. 하지만, 본 논문에서 제안한 점진적 규칙 추출 알고리즘을 사용할 경우에 보다 많은 테스트 패턴을 규칙으로 분류할 수 있게 된다.

<표 6> 규칙으로 분류 가능한 테스트 패턴 비율(%)

| | Breast-cancer | Glass | Ionosphere | Iris | New-thyroid | Wine |
|---|---|---|---|---|---|---|
| PRISM | 97.86 | 81.66 | 89.86 | 94 | 94.5 | 87.54 |
| ORS | 79.09 | 65.46 | 27.29 | 88.06 | 91.20 | 43.59 |
| PRS | 97.20 | 84.06 | 51.80 | 97.46 | 97.62 | 71.23 |

### 5.2 분류 성능 실험 결과

<표 7>의 분류 성능 실험 결과에서 Glass, Ionosphere의 경우에는 PRISM 기법이 PRS보다 높은 성능을 보여주고 있으며, Breast-Cancer, Iris, New-Thyroid, Wine에서는 PRS가 높은 성능을 보여주고 있다. Glass, Ionosphere에서 PRS보다 PRISM의 분류 성능이 높은 이유는 PRISM 기법의 경우, 테스트 패턴에 적용 가능한 규칙이 없으면 majority class로 분류하기 때문이며, Glass, Ionosphere의 패턴들이 특정 클래스에 편중되어 있는 데이터셋이다. 다시 말해서, PRISM 기법은 학습패턴 집합의 분포에 따라서 분류 성능이 영향을 받지만, 본 논문에서 제안한 알고리즘은 규칙만으로 분류가 불가능한 경우, 대표패턴을 이용하여 분류하기 때문에 안정적인 분류 성능을 보여준다.

<표 7> 분류 성능

| | Breast-cancer | Glass | Ionosphere | Iris | New-thyroid | Wine |
|---|---|---|---|---|---|---|
| PRISM | 95.55 | 92.22 | 94.20 | 87.80 | 92.15 | 87.22 |
| ORS | 95.85 | 86.85 | 90.02 | 97.07 | 94.46 | 93.67 |
| PRS | 95.37 | 88.44 | 90.82 | 93.40 | 94.09 | 95.32 |

### 5.3 규칙 개수

<표 8> 규칙 개수

| | Breast-cancer | Glass | Ionosphere | Iris | New-thyroid | Wine |
|---|---|---|---|---|---|---|
| PRISM | 43.59 | 39.08 | 35.53 | 16.87 | 22.12 | 18.24 |
| ORS | 204.79 | 112.40 | 260.86 | 44.15 | 46.69 | 121.43 |
| PRS | 47.05 | 36.90 | 151.95 | 9.51 | 10.21 | 29.33 |

<표 8>에서 Breast-cancer, Glass는 PRISM과 PRS가 유사하며, Iris, New-thyroid는 PRS가 좀더 적은 것을 볼 수 있다. Ionosphere의 경우에 오히려 규칙의 개수가 증가하는 현상을 보여주는데, 이는 특징의 개수가 많을 경우에 RPA 기법의 특성상 과도한 분할이 발생하기 때문으로 사료된다. ORS에 비해서는 Breast-cancer 77%, Glass 68%, Ionosphere 42%, Iris 80%, New-thyroid 79%, Wine 76% 정도 줄어드는 것을 볼 수 있다.

## 6. 결론

본 논문에서는 메모리 기반 추론 기법을 이용하여 분류 기준을 설명할 수 있는 규칙을 생성하고 일반화 성능을 향상시킬 수 있는 규칙 pruning 알고리즘과 생성되는 규칙의 수를 줄이기 위한 점진적 규칙 추출 알고리즘을 제안하였다. PRS가 PRISM보다 모든 데이터 셋에 대해서 안정적인 분류 성능을 보여주며, 불필요한 조건을 제거한 규칙을 점진적으로 추출함으로써, ORS에 비해서 많은 개수의 규칙을 줄일 수 있었다.

## 7. 향후 연구

본 논문에서 제안한 방법은 데이터 셋의 특징 수가 많은 경우에 과도한 분할이 발생하여 생성되는 규칙의 수가 많아지는 현상을 보여주고 있다. 향후 연구에서는 학습에 사용되는 특징의 수를 줄 일 수 있는 방법과 불필요한 분할을 방지하는 방법에 대한 연구를 진행할 예정이다. 또한, 규칙의 질(Quality)를 평가하는 새로운 방법에 대해서도 연구를 진행할 예정이다.

## 참고문헌

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

[2] J. Cendrowska, PRISM : "An Algorithm for inducing modular rules", International Journal of Man-Machine Studies 27 (4): 349-370, 1987

[3] 이형일, 정태선, 윤충화, 강경식, 재귀분할 평균법을 이용한 새로운 메모리기반 추론 알고리즘, 한국정보처리학회 논문지, Vol.6 No7, pp.1849-1857, 1999

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

[5] D. Aha, Instance-Based Learning Algorithms, Machine Learning, Vol. 6, No. 1, pp. 37-66, 1991.

[6] 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.