# 재귀분할 평균법을 이용한 새로운 메모리기반 추론 알고리즘

이 형 일[†] · 정 태 선[††] · 윤 충 화[†††] · 강 경 식[††††]


## 요 약

메모리 기반 추론에서 기억공간의 효율적인 사용과 분류성능의 향상을 위하여, 재귀 분할 평균 기법을 제안하였다. 이 알고리즘은 패턴공간을 구성하는 각 초월 평면이 동일한 클래스 소속의 패턴으로 구성될 때까지 재귀적으로 분할한 후, 초월 평면별로 소속된 패턴들의 평균값을 계산하여 대표패턴을 추출한다. 또한 각 특징과 클래스간의 상호정보를 특징의 가중치로 사용하여 분류 성능의 향상을 시도하였다. 제안된 알고리즘은 k-NN(k-Nearest Neighbors) 분류기에서 필요로 하는 메모리 공간의 30~90%만을 사용하며, 분류에 있어서도 k-NN과 유사한 인식 성능을 보이고 있다. 또한 저장된 패턴 개수의 감소로 인하여, 실제 분류에 소요되는 시간에 있어서도 k-NN보다 월등히 우수한 성능을 보이고 있다.


# A New Memory-Based Reasoning Algorithm using the Recursive Partition Averaging

Hyeong-Il Lee[†] · Tae-Sun Cheong[††] · Chung-Hwa Yoon[†††] · Kyung-Sik Kang[††††]


## ABSTRACT

We proposed the RPA (Recursive Partition Averaging) method in order to improve the storage requirement and classification rate of the Memory Based Reasoning. This algorithm recursively partitions the pattern space until each hyperrectangle contains only those patterns of the same class, then it computes the average values of patterns in each hyperrectangle to extract a representative. Also we have used the mutual information between the features and classes as weights for features to improve the classification performance. The proposed algorithm used 30~90% of memory space that is needed in the k-NN (k-Nearest Neighbors) classifier, and showed a comparable classification performance to the k-NN. Also, by reducing the number of stored patterns, it showed an excellent result in terms of classification time when we compare it to the k-NN.


## 1. 서 론

메모리 기반 추론의 학습은 주어진 학습패턴 그 자체를 모두 메모리에 저장하는 것일 뿐이며, 테스트패턴의 분류는 저장된 학습패턴들과 테스트패턴간의 거리를 이용하므로 거리기반 학습(Distance Based Learning)이라고도 한다[1,2].

메모리기반 학습 알고리즘에 기반을 둔 분류기로는 k-NN(k-Nearest Neighbors) 분류기를 들 수 있으며 k-NN 분류기는 메모리에 저장된 학습패턴 중 주어진 테스트패턴과 가장 가까운 거리에 있는 k개의 학습패

※ 본 연구는 1996년도 한국학술진흥재단 대학부설연구소과제 연구비에 의하여 연구되었음.
† 종신회원 : 김포대학 전자계산과 교수
†† 준 회 원 : 명지대학교 대학원 컴퓨터공학과
††† 정 회 원 : 명지대학교 컴퓨터공학과 교수
†††† 정 회 원 : 명지대학교 산업공학과 교수
논문접수 : 1999년 1월 29일, 심사완료 : 1999년 5월 25일

턴을 선택하여 그중 가장 많은 패턴이 소속된 클래스로 테스트패턴을 분류하는 방법을 사용한다[2,3,4]. 이러한 k-NN 분류기는 그 성능 면에서 만족할 만한 결과를 보이고 있으며, 이미 다양한 분야에 응용되고 있다. 하지만 이 기법의 가장 큰 문제점은 학습패턴 전체를 메모리에 저장하여야 하므로 다른 기계학습 방법에 비하여 많은 메모리 공간을 필요로 하며, 저장되는 학습패턴이 증가할수록 분류에 필요한 시간도 많이 소요된다는 단점을 갖는다[5,6]. 따라서 메모리 기반 학습기법이 갖고 있는 문제점을 해결하기 위한 연구가 지금까지 활발히 진행되어 오고 있으며, 대표적인 연구로 IBL (Instance Based Learning)[6], NGE(Nested Generalized Exemplar)[7,8] 이론과 FPA(Fixed Partition Averaging) [9]를 들 수 있다.

본 논문에서는 주어진 패턴 공간을 재귀적으로 분할해 나가면서 각 분할된 초월 평면을 대표하는 패턴을 추출하여 효율적인 메모리 사용과 분류성능을 보장하는 새로운 알고리즘을 제안하고, UCI Repository의 벤치마크 데이터를 이용하여 성능을 실험적으로 검증하였다.

## 2. 관련 연구

### 2.1 k-NN 기법

k-NN 분류기는 메모리기반 학습기법을 사용한 최초의 분류기로써 이 방법은 Lazy Learning Algorithm 이라고도 하는데, 그 이유는 학습 시에는 단순히 학습패턴들을 메모리에 저장하며, 차후 테스트패턴을 분류할 때 모든 계산이 수행되기 때문이다[10].

이러한 k-NN 분류기의 개략적인 알고리즘은 다음과 같다.

① 주어진 학습패턴을 모두 메모리에 저장한다.

② 테스트패턴 $Q$의 분류를 위하여 메모리에 저장된 모든 학습패턴과의 거리를 식 (1)을 이용하여 계산한다.

$$D_{EQ}=\sqrt{\sum_{i=1}^{n}(E_i-Q_i)^2} \qquad (1)$$

이때 E는 메모리에 저장된 학습패턴을 나타내며, Q는 주어진 테스트패턴이다. 또한 n은 패턴을 구성하는 특징의 개수이며, $E_i, Q_i$는 각각 학습패턴과 테스트패턴의 $i$번째 특징 값을 나타낸다.

③ 테스트패턴 $Q$와 가장 가까운 k개의 학습패턴을 선정한다.

④ 선택된 k개의 학습패턴 중 가장 많은 개수의 패턴이 소속되는 클래스로 테스트패턴 $Q$를 분류한다.

위에서 보이는 것처럼 k-NN 분류기에서의 학습은 학습패턴을 저장하는 것 이외에 아무런 조치를 취하지 않는다. 이때 k값은 분류기의 성능을 최적화하기 위하여 일반적으로 Cross Validation 기법을 사용하여 결정하며, k=1인 경우를 NN 분류기라 한다[2,3,4]. 또한 위의 과정 중 4번째 단계에서, 테스트패턴과의 거리를 이용하여 가중치를 부여하는 방법을 WeightVote k-NN이라고 한다[3,4].

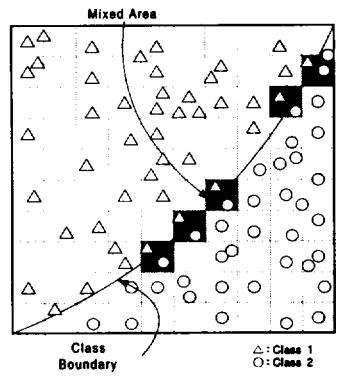
### 2.2 고정 분할 평균 기법

고정 분할 평균(FPA) 기법은 주어진 패턴공간을 동일한 크기의 초월평면들로 분할한 후 패턴 평균법을 적용하는 방법이다. 이 방법에서는 먼저 패턴 공간의 각 특징 축을 일정한 크기로 분할한다. 이때 특징축의 분할 개수는 식 (2)에 의해 결정된다.

$$N=\lceil \log_n(0.3\times T) \rceil \qquad (2)$$

이때 n은 하나의 패턴을 구성하는 특징 개수, T는 전체 학습패턴의 개수이다.

FPA 기법에서는 각 축을 같은 크기의 N개로 분할한 후, 분할된 초월평면 단위로 패턴 평균법을 적용한다. (그림 1)은 패턴공간을 구성하는 2개의 축을 각각 10개의 영역으로 분할한 경우이다. (그림 1)에서 회색으로 표시된 클래스 혼합 부분의 경우에는 패턴 평균법을 적용하지 않고 원래의 패턴들을 그대로 저장하며, 클래스가 혼합되지 않은 부분은, 해당 셀 내의 모든




(그림 1) 고정 분할 평균법

패턴을 평균하여 하나의 대표패턴으로 대체하는 방법을 사용한다.

또한 FPA 기법에서는 분류기의 성능 향상을 위하여 상호정보를 이용한 특징의 가중치를 사용한다. 특징과 클래스간의 상호정보는 해당 패턴이 클래스의 결정에 미치는 영향력으로 식 (3)과 (4)에 의해 계산된다[11].

$$I = -\sum_{i=1}^{C} p_i \log_2 p_i \quad (3)$$

이때 $p_i$는 전체 학습패턴 중 클래스 $i$에 소속되는 패턴의 비율, 즉 임의의 패턴이 클래스 i로 분류될 사전 확률을 의미하며, C는 전체 학습패턴을 구성하는 클래스의 개수이다.

FPA에서 특징 $f$의 가중치로 사용하는 상호정보이득(Mutual Information Gain)은 다음의 식 (4)에 의해 계산된다[12].

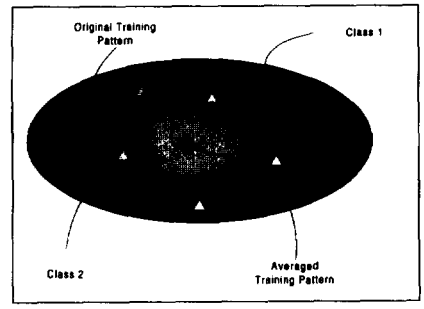
$$IG(f) = I - \sum_{i=1}^{N} P_i I_c \quad (4)$$

이때 $I$는 식 (3)에서 정의한 특징축 분할 이전에 필요한 정보의 양, N은 특징 축 $f$의 분할 개수이며 이 값은 식 (2)에 의해 사전에 계산된다. $Ic$는 특징 $f$를 기준으로 분류했을 때 분할된 공간에서 필요한 정보의 량이며, 이 값은 식 (3)과 같은 방법을 사용하여 계산한다. 또한 $P_i$는 전체 학습패턴 중 분할된 초월평면에 할당된 패턴의 비율이다.

FPA 기법의 목표는 전체 학습 패턴을 거리 계산에 사용하는 k-NN 기법의 분류성능에 근접하면서, 패턴 평균법을 사용하여 k-NN 기법에서 나타나는 메모리 공간의 낭비를 줄이고자 하는 것이다. 그러나, FPA 기법에서는 식 (2)에 의해 정의되는 고정 개수의 분할 구간을 사용하며, 이 수치는 전체 학습패턴의 개수에 의해 사전에 결정된다. 또한 이 분할 구간의 크기는 분류성능에 영향을 미치게 된다. 따라서, 본 논문에서는 이러한 문제점을 해결하기 위하여 학습패턴의 분포 특성에 따라 동적으로 분할 공간의 개수를 결정하는 RPA 기법을 제안하였다.

## 3. RPA 학습 기법

본 논문에서는 메모리기반 학습 기법에서 보다 효율적인 메모리사용과 분류 성능을 보장하기 위하여 재귀분할 평균(RPA : Recursive Partition Averaging) 기법을 제안하였다. RPA 기법은 주어진 패턴공간을 재귀적으로 분할해 나가면서 대표패턴을 추출하는 방법이다. RPA에서는 메모리 사용 효율을 증대하기 위하여 인스턴스 평균(Instance Averaging) 법을 적용하였으며, 인스턴스 평균법이란 여러 개의 학습패턴의 특징값을 평균하여 하나의 대표패턴으로 대치하는 방법을 의미한다[13,14]. 하지만 단순히 인스턴스 평균법을 적용하는 경우, 클래스가 다음의 (그림 2)와 같이 환형을 이루고 있을 경우 문제가 발생하게 된다. (그림 2)에 나타난 것처럼 단순히 패턴 평균법을 적용할 경우 클래스 1에 소속된 5개 패턴의 평균으로 구한 대표패턴이 원래의 클래스와는 다른 클래스 2에 소속되며, 이 경우 분류기는 오인식을 하게 된다.




(그림 2) 인스턴스 평균법의 문제점

하지만, RPA에서는 분할된 모든 초월평면이 하나의 클래스에 속한 패턴들로 구성될 때까지 계속해서 재귀적으로 분할해 나가므로, (그림 2)의 문제점은 발생하지 않는다.

### 3.1 특징의 정규화

메모리 기반 분류기에서 출력 클래스의 결정은 테스트패턴과 메모리에 저장된 학습패턴 사이의 거리를 이용하게된다. 이 기법에서는 패턴을 구성하는 특징들이 갖는 값의 범위가 판이하게 다를 경우 문제가 발생하게 된다. 예를 들어 (0.9, 400, 0.0004), (0.8, 410, 0.02)와 같은 특징으로 구성된 패턴에서, 두 번째 특징은 다른 두 개의 특징에 비하여 상대적으로 큰 값으로 구성되어있다. 따라서 두 번째 특징이 조금만 차이가 나더라도 나머지 특징간의 차이에 관련 없이 출력 클래스가 결정된다. 이러한 문제점의 해결을 위하여 RPA

에서는 다음의 식 (5)를 이용하여 특징값을 정규화한다. 이 기법은 식 (5)에 의하여 패턴을 구성하는 모든 특징 값을 0과 1사이의 값으로 정규화함으로써, 모든 특징의 변화가 패턴의 소속클래스 결정에 미치는 영향력을 동일하게 한다.

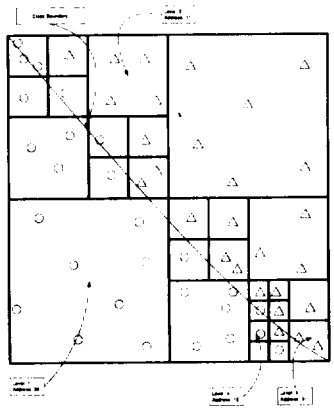
$$f_{i*} \frac{f_i - f_{i_{min}}}{f_{i_{max}} - f_{i_{min}}} \quad (5)$$

이때 $f_i$는 $i$번째 특징 값, $f_{i_{max}}$, $f_{i_{min}}$는 $f_i$가 가질 수 있는 최대, 최소 값을 나타낸다.

### 3.2 패턴공간의 분할과 주소 부여

정규화 작업을 수행한 후, RPA는 주어진 패턴공간의 각 특징 축을 최초 2개의 영역으로 분할한다. 따라서 첫 번째 분할에서는 패턴공간이 $2^n$개의 공간으로 분할되며, 이때 n은 패턴을 구성하는 특징의 개수 즉, 패턴공간의 차원수가 된다. 따라서 2차원 패턴의 경우 최초 4개의 패턴공간으로 분할되며, 패턴공간 또는 셀의 분할은 3.3절에서 설명하는 조건에 따라 재귀적으로 이루어진다.

RPA에서는 현재 분할하고자 하는 셀의 위치를 알아내기 위하여 레벨과 주소를 사용한다. 이때 각 셀의 주소는 2진수 형태의 값을 갖게 되며, LSB(Least Significant Bit)가 패턴의 첫 번째 특징을, MSB(Most Significant Bit)가 패턴의 마지막 특징을 나타내는 주소를 가지도록 구성된다. RPA에서는 매 분할시 각 축을 2개의 영역으로 구분하므로 셀을 구성하는 각축의 위치를 0과 1로 표현이 가능하다. 또한 RPA에서는 셀의 위치 판별을 위하여 현재 몇 번째 분할을 시도하고 있는가에 대한 정보를 갖고 있으며, 이 값을 레벨(Level) 이라 한다. 레벨은 주소와 함께 셀의 위치판단에 사용된다. 다음의 (그림 3)은 RPA에 의해 분할된 패턴공간의 레벨과 주소를 보여 주고 있다.



(그림 3) 패턴공간의 RPA 분할

### 3.3 대표 패턴의 추출

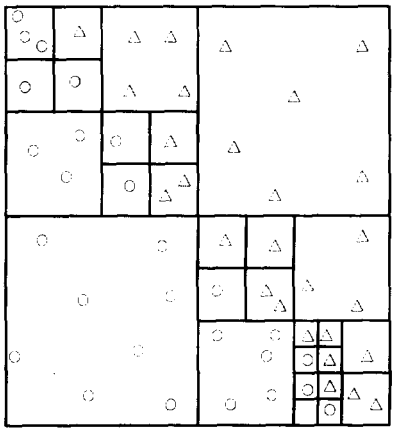
RPA에서는 학습패턴이 소속되는 셀의 판별과 재귀분할 여부의 결정을 위하여 전체 학습패턴에 대하여 주소를 부여한다. 주소 부여방법은 해당 패턴이 소속되는 셀의 주소를 패턴의 주소로 받게 되며, 주소 부여가 완료되면 학습패턴을 주소별로 정렬한다. 이 과정은 재귀 분할의 여부와 재귀 분할의 대상이 되는 학습패턴을 결정하기 위해 수행된다.

RPA의 마지막 단계로 현재 분할된 각 셀에 대하여 재귀 분할 여부를 결정한다. RPA에서는 하나의 셀에 소속되는 패턴의 클래스가 모두 같을 경우, 해당 셀의 패턴들에 대하여 패턴평균법을 적용하여 대표 패턴을 추출한다. 반대로 셀에 여러 개의 클래스에 소속되는 패턴이 혼합되어 있을 경우, 해당 셀을 다시 분할한다. 특정 셀의 분할이 결정되면 3.2와 3.3절의 작업을 반복한다. RPA의 경우, 클래스가 혼합된 부분에 대해서는 점점 세밀하게 분할해 나가게 되므로, 클래스 경계면에 근접한 셀의 경우 상대적으로 많은 분할이 이루어지게 된다.

### 3.4 특징 가중치의 계산

전체 학습패턴에 대한 분할 및 대표 패턴 추출 작업이 완료되면, 2.2절의 FPA기법과 동일한 방식으로 식 (3)과 (4)를 이용하여 각 특징의 가중치를 계산한다. 이때 RPA에서는 주어진 패턴공간을 재귀적으로 분할하므로, 각 특징축의 분할개수를 결정하는 방법이 필요하게 되는데, (그림 4)에서 보는 것과 같이 실제 분할된 패턴공간에 가상 분할선을 사용하여 각 특징의 분할 공간 개수를 정의하였다.

다음의 (그림 4)에서 굵은 실선으로 표시된 부분은 실제 RPA에 의해 형성된 초월평면을 나타내는 것이며, 가는 점선으로 표시된 부분은 특징 가중치 계산을 위하여 패턴공간을 가상으로 분할한 선을 나타낸다. 이 경우 가로 특징 축은 9개, 세로 특징 축은 10개로 분할된 것을 볼 수 있다. RPA에서는 식 (4)로 주어진 $IG(f)$ 값을 테스트패턴과 메모리에 저장된 학습패턴간의 거리계산에 있어 특징의 가중치로 사용하며, 이때



(그림 4) 특징가중치 계산을 위한 분할

의 거리는 식 (6)에 의해 계산한다.

$$D_{EQ}=\sqrt{\sum_{i=0}^{n} IG(i)(E_i-Q_i)^2} \quad (6)$$

### 3.5 RPA 기법의 패턴 분류

RPA 기법을 이용한 분류기에서는 k-NN 분류기와는 다른 분류 기법을 사용한다. k-NN 분류기의 경우, 분류기의 성능을 최적화 하기 위하여 k값을 사전에 결정하고 전체 시스템에서 하나의 고정된 k값을 사용하게된다. 하지만 이 방법의 경우 k값의 결정을 위해서 주로 사용되는 Cross-Validation법의 특성상 많은 계산 시간을 요하게되며, 이에는 Leave-1-Out법과 N-Folding 법이 있다. Leave-1-Out법은 전체 패턴 중 1개를 제외한 모든 패턴을 학습패턴으로 사용하고, 제외된 1개의 패턴을 테스트패턴으로 하여 분류를 시도하는 방법으로 모든 패턴이 각각 한번씩 테스트패턴으로 사용될 때까지 분류를 계속하는 방법이다. 비슷한 방법으로 N-Folding 기법은 전체 패턴을 N개의 그룹으로 분할하고 각 그룹을 한번씩 돌아가면서 테스트패턴으로 사용하는 방법이다[15].

반면에 RPA에서는 k값을 학습시에 결정하지 않고 테스트패턴의 분류시에 결정하게 되며, k값은 가변적으로 결정된다. RPA의 패턴 분류시, 가장 인접한 패턴과 그 다음으로 가까운 패턴의 클래스가 같을 경우 k=1인 NN 분류기와 같은 방법으로 분류하게 되며, 만일 가까운 두 패턴의 클래스가 다를 경우, 데이터를 구성하는 모든 클래스에서 적어도 하나의 패턴이 추출될 때까지 거리 순서로 패턴을 추출하게 된다. 이때 k값이 되는 패턴의 개수는 현재 테스트패턴과 가까운 패턴에 따라 변하게 된다. 그 후 테스트패턴의 분류는 k-NN 분류기와 동일하게 가장 많은 학습패턴들이 소속된 클래스로 분류한다.

## 4. 실험 및 분석

RPA 기법을 이용한 분류기의 성능을 k-NN, FPA 기법과 비교하여 검증하였다. 실험은 기계학습의 벤치마크 자료로 사용되는 7개의 데이터를 이용하였으며, 실험 방법은 70 : 30법(전체 데이터를 기준으로 70%는 학습패턴으로, 30%는 테스트패턴으로 사용하는 방법)을 사용하였다[15]. 이때 70%의 학습패턴은 전체 패턴의 클래스별 분포를 고려하여 모든 클래스에서 같은 비율로 추출하였다. 실험은 Windows NT를 적재한 PentiumII-300 컴퓨터를 사용하였으며, 모든 실험결과는 25회 반복 측정한 후 평균값으로 나타내었다.

### 4.1 실험 데이터

본 논문에서는 기계 학습의 벤치마크 자료로 사용되는 7개의 데이터를 Univ. of California at Irbine의 Machine Learning Database Repository에서 발췌하여 사용하였으며, 이들 7개의 데이터는 Breast-Cancer Wisconsin, Glass, Ionosphere, Iris, New-Thyroid, Sonar, Wine이며, 이들 데이터는 모든 특징이 실수값을 취한다. 다음의 <표 1>은 실험자료의 특성, <표 2>는 7개의 데이터를 70 : 30법을 이용하여 나누었을 경우, 클래스별 학습패턴의 분포를 보여주고 있다.

〈표 1〉 실험 데이터의 특성

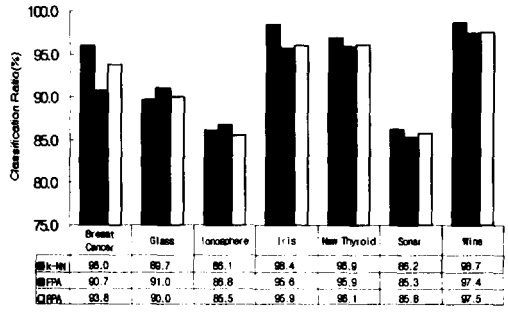
| 데 이 터 | 패 턴 개 수 | 특 징 개 수 | 클래스 개 수 |
|---|---|---|---|
| Breast-Cancer Wisconsin | 699 | 10 | 2 |
| Glass | 214 | 10 | 6 |
| Ionosphere | 351 | 34 | 2 |
| Iris | 150 | 4 | 3 |
| New-Thyroid | 215 | 5 | 3 |
| Sonar | 208 | 60 | 2 |
| Wine | 178 | 13 | 3 |

〈표 2〉 클래스별 학습패턴의 분포

| 데 이 터 | 학습패턴 개 수 | 클래스별 학습패턴 개수 | | | | | |
|---|---|---|---|---|---|---|---|
| | | C1 | C2 | C3 | C4 | C5 | C6 |
| Breast-Cancer Wisconsin | 488 | 320 | 168 | × | × | × | × |
| Glass | 148 | 53 | 11 | 0 | 9 | 6 | 20 |
| Ionosphere | 245 | 157 | 88 | × | × | × | × |
| Iris | 105 | 35 | 35 | 35 | × | × | × |
| New-Thyroid | 150 | 105 | 24 | 21 | × | × | × |
| Sonar | 144 | 67 | 77 | × | × | × | × |
| Wine | 123 | 41 | 49 | 33 | × | × | × |

### 4.2 분류성능 실험

(그림 5)의 k-NN, FPA, RPA의 분류성능을 보면, RPA 기법의 성능이 전체 학습패턴을 고려하는 k-NN 기법보다는 떨어지나 거의 대등한 분류성능을 보이고 있다. 또한 FPA 기법과의 비교를 보면, Glass와 Ionosphere 데이터를 제외한 나머지 데이터에서 RPA가 우수한 분류성능을 보장하고 있다.




(그림 5) 분류성능의 비교

메모리 기반 분류기에서는, 저장된 학습패턴 중 클래스 경계면 근처에 있는 패턴들보다 경계면에서 멀리 떨어진 패턴들이 좀더 정확한 분류를 보장한다는 연구결과가 있는데[5,6], 본 논문에서 제안한 RPA 기법은 클래스 경계면에 근접할수록 좀더 작은 크기의 초월평면으로 재귀 분할하면서 패턴평균법을 적용한다. 따라서 클래스 경계면 근처에 분포한 패턴의 수가 줄어들게 되는데 반하여, FPA 기법은 전체 패턴공간을 같은 크기로 분할하므로 클래스 경계면이 복잡한 경우, 경계면 근처에 분포한 대부분의 패턴을 원형 그대로 적용하게 되므로, 오분류되는 패턴의 수가 RPA 기법에 비하여 증가하게 된다. 또한 k-NN 기법은 사전에 최적의 k값을 미리 결정하여야 한다는 단점이 있으며, FPA 기법에서는 특징축의 분할개수를 미리 결정해야 한다. 반면에 본 논문에서 제안한 RPA 기법에서는 k값이나 특징축의 분할개수, 즉 외부 파라미터를 전혀 사용하지 않는다는 장점을 갖는다.

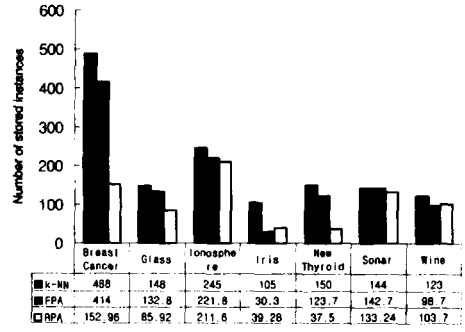
(그림 5)의 실험에서 k-NN 분류기의 성능은 Leave-1-Out Cross Validation기법을 사용하여 계산한 최적의 k값을 사용한 것이며, 다음의 <표 3>은 각 데이터에서 사용된 k-NN 분류기의 k값을 보여주고 있다.

〈표 3〉 k-NN 분류기의 분류성능 최적화를 위한 k값

| Breast Cancer | Glass | Ionosphere | Iris | New Thyroid | Sonar | Wine |
|---|---|---|---|---|---|---|
| 21 | 1 | 1 | 51 | 1 | 1 | 19 |

### 4.3 메모리 사용량 비교 실험

(그림 6)의 실험결과에서는 k-NN, FPA, RPA 세 가지 방법을 이용한 분류기의, 메모리 사용량을 보여주고 있다. k-NN의 경우 모든 학습패턴을 메모리에 저장하고 분류시 테스트패턴을 모든 학습패턴과 비교한다. 하지만 FPA, RPA 기법의 경우, 주어진 패턴공간을 초월평면으로 분할하여 각 초월평면을 대표하는 패턴을 저장하는 방법을 사용함으로써 우수한 메모리 사용효율을 보장하게 된다.

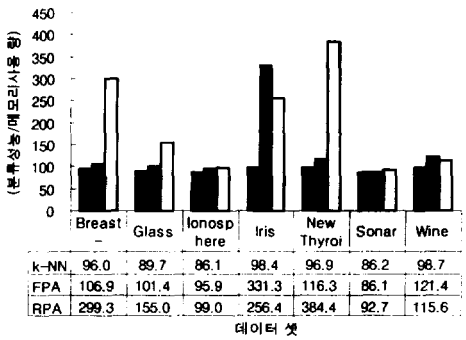



(그림 6) 메모리 사용량의 비교

New-Thyroid 데이터의 경우, k-NN 대비 약 25%의 메모리만을 사용하고 있으며, Breast-Cancer 데이터의 경우는 30%, Iris 데이터의 경우는 40%의 학습패턴만을 메모리에 저장하는 것을 볼 수 있다. 또한 나머지 데이터에서도 약 60-90%의 학습패턴만을 메모리

에 저장한다. FPA 기법과의 비교에 있어서도 Iris, Wine 데이터를 제외한 5개의 데이터 셋에서도 우수한 메모리 사용효율을 보이고 있는 것을 볼 수 있다. 이것은 FPA 기법의 경우, 기법의 특성상 클래스 경계면이 특징 축과 평행을 이룰 때 우수한 메모리 사용효율을 보장하는 반면, RPA 기법은 특징축과 클래스 경계면이 평행하지 않은 데이터 셋에 있어서도 재귀 분할을 통한 패턴평균법을 적용하여 우수한 메모리 사용효율을 얻을 수 있기 때문이다.

(그림 7)의 실험결과를 보면, RPA 기법이 메모리 사용효율을 고려한 분류성능에 있어서 기존의 k-NN 분류기 및 FPA 기법에 비하여 우수한 성능을 보이고 있는 것을 볼 수 있다. (그림 7)의 분류성능/메모리 사용량 비교 실험에서 메모리 사용량은 k-NN 분류기에서 사용하는 전체 학습패턴의 개수를 1로 보았을 때, FPA, RPA의 메모리 사용량을 적용한 결과이다.

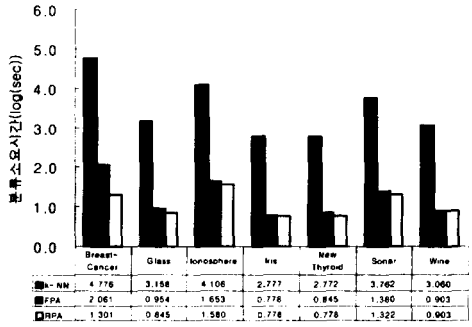



| | Breast- | Glass | Ionosphere | Iris | New Thyroi | Sonar | Wine |
|---|---|---|---|---|---|---|---|
| k-NN | 96.0 | 89.7 | 86.1 | 98.4 | 96.9 | 86.2 | 98.7 |
| FPA | 106.9 | 101.4 | 95.9 | 331.3 | 116.3 | 86.1 | 121.4 |
| RPA | 299.3 | 155.0 | 99.0 | 256.4 | 384.4 | 92.7 | 115.6 |

(그림 7) 분류성능/메모리 사용량

### 4.4 분류 소요시간 비교

메모리기반 학습기법을 이용한 분류기의 경우, 메모리에 저장된 학습패턴의 개수와 테스트패턴의 분류시간은 직접적인 관계를 가지게 되며, 초월평면별로 대표패턴을 추출하는 RPA 알고리즘은 테스트패턴의 분류에 소요되는 시간에 있어서도 k-NN 기법보다 빠른 분류 속도를 보장한다. (그림 8)의 결과에서, RPA 기법은 분류기 성능에 영향을 미치는 k값을 사전에 결정하지 않고, 저장된 학습패턴의 개수를 줄임으로써 학습과 분류에 소요되는 시간이 모든 데이터 셋에 있어 k-NN 기법에 비하여 월등히 적게 소요되는 것을 볼 수 있다. 그림에 표시된 값은 $\log_{10}$(소요시간)을 나타내며, 이때의 소요시간은 25회 반복 실험하는데 소요되는 총 소요시간을 나타낸다.




(그림 8) 분류소요시간 비교

## 5. 결 론

본 논문에서는, 메모리 기반 추론에 있어 효율적인 메모리 사용과 분류성능의 향상 기법을 제안하였다. 본 논문에서 제안한 RPA 기법은 패턴평균법을 사용하여 메모리에 저장되는 학습패턴들을 대표패턴으로 대체하는 방법을 채택하였으며, 실험 결과에서 볼 수 있는 것처럼 제안된 RPA 기법은 k-NN 기법과 비교하여 모든 데이터 셋에서 적은 메모리 공간을 필요로 하며, FPA 기법과의 비교에 있어서는 실험에 사용한 벤치마크 자료 7개중에서 5개의 데이터 셋에서 적은 공간을 사용하고 있다. 또한 분류 성능면에서 기존의 k-NN 기법과 거의 비슷한 성능을 보이며, FPA 기법에 비해서는 5개의 데이터 셋에서 우수한 성능을 보이고 있다. 마지막으로 FPA 기법에서는 패턴을 구성하는 특징축의 분할개수에 따른 분류기의 성능변화가 발생하지만, RPA 기법에서는 패턴공간을 재귀적으로 분할하게 되므로 이러한 외부 파라미터가 필요 없다는 장점을 갖는다.

기존의 메모리 기반 추론 기법에서는 규칙의 추출이 불가능하였으며, 규칙의 추출이 필요한 경우, 결정 트리 기법이나 유전자 알고리즘을 사용하여 왔다. 그러나 RPA에서는 분할된 초월평면에 소속되는 패턴들이 모두 하나의 클래스로 구성되므로, 규칙의 추출이 가능한 반면, FPA에서는 클래스가 혼합된 셀이 존재하게 되므로 규칙 추출에 어려움이 있다. 그러므로 본

논문의 저자들은 이러한 RPA의 규칙 추출가능성에 대해 연구하고 있는 중이다.

## 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] D. Wettschereck and T. Dietterich, Locally Adaptive Nearest Neighbor Algorithms, Advances in Neural Information Processing Systems 6, pp.184-191, Morgan Kaufmann, San Mateo, CA. 1994.

[3] D. Wettschereck, Weighted k-NN versus Majority k-NNA Recommendation. German National Research Center for Information Technology, 1995.

[4] S. Cost and S. Salzberg, A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features, Machine Learning, Vol.10, No.1, pp.57-78, 1993.

[5] 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.

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

[7] D. Wettschereck and T. Dietterich, An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms, Machine Learning, Vol.19, No.1, pp.1-25, 1995.

[8] S. Salzberg, A Nearest hyperrectangle learning method, Machine Learning, No.1, pp.251-276, 1991.

[9] 정태선, 이형일, 윤충화, 고정 분할 평균기법을 사용하는 향상된 메모리 기반 추론, 명지대학교 산업기술연구소 논문지, Vol.17, 1998.

[10] D. Wettschereck, et al., A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms, Artificial Intelligence Review Journal, 1996.

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

[12] 김상귀, 이형일, 윤충화, A study on the optimization of binary decision tree, 명지대학교 산업기술연구소 논문지, Vol.16, pp.104-112, 1997.

[13] G. Bradshaw, Learning about speech sounds : The NEXUS project. In Proceedings of the Fourth International Workshop on Machine Leaning, pp.1-11, Irvine, CA : Morgan Kaufmann, 1987.

[14] T. Kohonen, Learning vector quantization for pattern recognition (Technical Report TKK-F-A601). Espoo, Finland : Helsinki University of Technology, Department of Technical Physics, 1986.

[15] S. Salzberg, On Comparing Classifiers : Pitfalls to Avoid and a Recommended Approach, Data Mining and Knowledge Discovery, Vol.1, pp.1-11, 1997.



**이 형 일**

e-mail : hilee@kimpo.ac.kr

1985년 명지대학교 전자계산학과 (학사)

1985~1989년 (주) 쌍용컴퓨터 근무

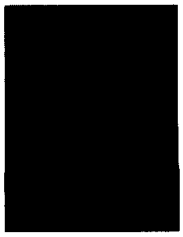
1990~1995년 CHNO System Consulting Co. 근무

1994년 명지대학교 대학원 전자계산학과(석사)

1997년 명지대학교 대학원 컴퓨터공학과 박사과정 수료

1997년~현재 김포전문대 전자계산과 전임강사

관심분야 : 신경회로망, 기계학습, 지능형 소프트웨어 에이전트


**정 태 선**

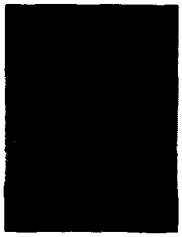
e-mail : Jupiter@ce.myoungji.ac.kr

1995년 명지대학교 컴퓨터공학과 (학사)

1998년 명지대학교 대학원 컴퓨터공학과 (석사)

1998년~현재 명지대학교 대학원 컴퓨터공학과 박사과정 재학

관심분야 : 신경회로망, 기계학습, 지능형 소프트웨어 에이전트




## 윤 충 화

e-mail : yoonch@wh.myongji.ac.kr

1979년 서울대학교 자연과학대학 수학과(학사)

1984년 미국 텍사스 주립대 전자계산학과(석사)

1989년 미국 루이지아나 주립대 전자계산학과(박사)

1990년~현재 명지대학교 컴퓨터공학과 부교수

관심분야 : 신경회로망, 전문가시스템, 지능형 소프트웨어 에이전트, 기계학습



## 강 경 식

e-mail : kangks@wh.myongji.ac.kr

1974년 인하대학교 산업공학과(학사)

1976년 연세대학교 공업경영학과(석사)

1977년 인하대학교 산업공학과(석사)

1987년 경희대학교 산업공학과(박사)

1976년~현재 명지대학교 산업공학과 교수

관심분야 : 전문가시스템, 생산관리