한국지능시스템학회 논문지 2008, Vol. 18, No. 4, pp. 456-462

# 동적 분할 평균을 이용한 새로운 메모리 기반 학습기법

# A New Memory-based Learning using Dynamic Partition Averaging

이형일
Hyeong-il, Yih

김포대학 인터넷정보과

## 요 약

분류란 새로운 자료를 주어진 클래스 중의 하나로 구분하는 것으로 가장 일반적으로 사용되는 데이터마이닝 기법 중의 하나이다. 그중 메모리기반 추론(MBR : Memory-Based Reasoning)은 추론 규칙 없이 특징들의 최초의 벡터 형태에 의해 표현된 학습패턴을 단순히 저장한다. 그리고 분류 시에 새로운 자료가 메모리에 저장된 학습패턴들과의 거리를 계산하여 가장 가까운 거리에 있는 학습패턴의 클래스로 분류하는 기법이다. MBR 기법에서 학습패턴이 커지면 저장에 필요한 메모리의 크기도 커질 뿐만 아니라 추론을 위한 계산도 많아지는 문제점을 가지고 있다. 이러한 문제를 해결하기 위한 대표적인 방법으로 초월평면을 이용하는 NGE이론과 대표패턴을 추출하여 학습하는 FPA기법과 RPA 기법 등을 들을 수 있다. 본 논문에서는 학습패턴 공간을 GINI-Index값을 이용하여 일련의 최적 분할점을 찾아 가변크기로 분할하는 동적분할평균(DPA : Dynamic Partition Averaging)기법을 제안하였다.
제안한 기법의 성능을 검증하기위하여 MBR기법 중 널리 사용되는 k-NN 기법과 비교하였다. 제안한 기법이 k-NN기법에 비해 대표패턴 개수는 줄이고 분류성능은 유사하게 유지시킨 것을 보여주었다. 또한, 제안한 기법은 NGE 이론을 구현한 EACH 시스템과 대표패턴 기법인 FPA기법과 RPA기법 등과 비교하여 탁월한 분류 성능을 보여주었다.

## Abstract

The classification is that a new data is classified into one of given classes and is one of the most generally used data mining techniques. Memory-Based Reasoning(MBR) is a reasoning method for classification problem. MBR simply keeps many patterns which are represented by original vector form of features in memory without rules for reasoning, and uses a distance function to classify a test pattern. If training patterns grows in MBR, as well as size of memory great the calculation amount for reasoning much have. NGE, FPA, and RPA methods are well-known MBR algorithms, which are proven to show satisfactory performance, but those have serious problems for memory usage and lengthy computation.
In this paper, we propose DPA (Dynamic Partition Averaging) algorithm. it chooses partition points by calculating GINI-Index in the entire pattern space, and partitions the entire pattern space dynamically. If classes that are included to a partition are unique, it generates a representative pattern from partition, unless partitions relevant partitions repeatedly by same method. The proposed method has been successfully shown to exhibit comparable performance to k-NN with a lot less number of patterns and better result than EACH system which implements the NGE theory and FPA, and RPA.

Key Words: Memory-Based Learning(메모리 기반 학습), Distance-Based Learning(거리기반학습), GINI Index

## 1. 서 론

분류란 새로운 자료를 주어진 클래스 중의 하나로 구분하는 것으로 가장 일반적으로 사용되는 데이터마이닝 기법 중의 하나이다. 그중 메모리기반 추론(MBR : Memory-Based Reasoning)은 추론 규칙 없이 특징들의 최초의 벡터 형태에 의해 표현된 학습패턴을 단순히 저장한다. 그리고 분류 시에 새로운 자료가 메모리에 저장된 학습패턴들과의 거리를 계산하여 가장 가까운 거리에 있는 학습패턴의 클래스로 분류하는 기법으로 거리기반 학습(Distance Based Learning) 이라고도 한다[1][2].

MBR 방법에서 널리 사용되는 대표적인 분류기는 k-NN(k-Nearest Neighbors) 분류기이다. 이것은 특징들의 최초의 벡터 형태에 의해 표현된 학습패턴들을 메모리에 모두 저장한다. 그리고 저장된 학습패턴들 중 주어진 입력패턴과 가장 가까운 거리에 있는 k개의 학습패턴들을 선택하여 그 중 가장 많은 패턴들이 소속된 클래스로 입력패턴을 분류하는 기법을 사용한다[2][3][4]. 성능 면에서는 이 기법은 만족할 만한 결과를 보여 이미 다양한 분야에 응용되고 있다. 그러나 메모리와 추론 시간에서는 학습 패턴 전체를 메모리에 저장하여야 하므로 저장되는 학습패턴이 커지면 저장에 필요한 메모리의 크기도 커질 뿐만 아니라 추론을 위한 계산도 많아지는 문제점을 가지고 있다[4].

접수일자 : 2007년 8월 27일
완료일자 : 2008년 5월 10일
이 논문은 2008학년도 김포대학의 연구비 지원에 의하여 연구되었음

MBR 방법이 갖고 있는 문제점을 해결하기 위한 연구가 지금까지 활발히 진행되어 오고 있으며, 대표적인 연구로 학습패턴을 그대로 저장하는 것이 아니라, 인접한 학습패턴들을 포함하는 초월평면 (Hyperrectangle)의 형태로 저장하여 이용하는 NGE(Nested Generalized Exemplar) 이론과 대표패턴을 추출하여 학습하는 FPA기법과 RPA 기법 등을 들을 수 있다[5][6][7][10][11].

본 논문은 동적 분할 평균(DPA : Dynamic Partition Averaging)을 이용한 새로운 메모리기반 학습(Memory-Based Learning) 기법을 제안한다. 이 기법은 패턴 공간을 분할할 때, GINI-Index 값을 이용하여 패턴의 분포를 산출한다. 이를 이용하여 패턴공간의 축에 대해 여러 개의 분할점을 선정하고 특징축을 분할한다. 그리고 생성된 분할영역 각각에 존재하는 패턴들이 하나의 클래스에 소속하는 경우는 평균기법을 이용해 대표 패턴을 추출하여 분류 기준 패턴으로 사용하며, 반면 분할영역에 서로 다른 클래스들의 패턴이 존재하는 경우는 하나의 클래스의 패턴이 존재할 때까지 패턴의 분포를 고려하여 분할영역을 가변크기 재귀이분할하여 분류하는 방법이다. 제안기법의 분류 성능을 확인하기 위하여 UCI Machine Learning Repository에서 벤치마크 데이터를 발췌하여 사용하였으며, 제안한 알고리즘과 k-NN 기법, EACH 시스템, 그리고 FPA 기법, RPA기법 등의 분류 성능, 메모리 사용 효율을 실험적으로 비교 검증하였다.

## 2. 관련 연구

### 2.1 k-NN 기법

k-Nearest Neighbor(k-NN) 분류기는 특징 공간에서 가장 가까운 학습 패턴들에 의거하는 입력패턴들을 분류하는 방법으로 메모리 기반 학습 기법으로 분류되는 대표적인 알고리즘이다. 또한 이 분류기는 단순히 학습 패턴을 메모리에 그대로 저장하고 모든 계산은 분류 시까지 연기되어 Lazy learning Algorithm이라고 불리 운다[8]. 이 분류기는 모든 기계학습 알고리즘들 중에서 가장 단순한 알고리즘으로 k개의 가장 가까운 서로 이웃하는 패턴들 중에서 가장 공통의 클래스로 분류되는 대다수 표결에 의해 분류한다. 이때, k는 일반적으로 작은 양의 정수이다. 만약 k가 1이면, 객체는 단지 그 가장 가까운 서로 이웃하는 것의 클래스에 할당하게 된다. 그리고 이진수의 분류 문제에 있어서는, 동수 투표들을 피하기 때문에 k를 기수이도록 뽑는 것은 도움이 된다.

k-NN 분류기의 개략적인 알고리즘은 다음 그림 1과 같다[2][3][10][11].

① 전체 학습패턴을 메모리에 저장한다.
② 테스트 패턴과 학습패턴들과의 거리를 식 (1)을 이용하여 계산한다.
③ 위에서 계산한 거리를 기준으로 테스트 패턴과 근접한 k개의 학습패턴을 선정한다.
④ 이 k개 중에서 가장 많은 개수의 학습패턴을 포함하는 클래스로 테스트 패턴을 분류한다.

그림 1. k-NN 기법
Figure 1. k-NN Method

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

이때, $E$는 메모리에 저장된 학습패턴을 나타내며, $Q$는 주어진 입력패턴이다. 또한 $n$은 패턴을 구성하는 특징의 개수이며, $E_i, Q_i$는 각각 학습패턴과 입력패턴의 $i$번째 특징값을 나타낸다. 이 때 k값은 분류기의 성능을 최적화하기 위하여 일반적으로 Cross Validation기법을 사용하여 결정하며, k=1인 경우를 NN 분류기라 한다. 또한 위의 과정 중 4번째 단계에서, 입력패턴과의 거리를 이용하여 가중치를 부여하는 방법을 WeightVote k-NN이라고 하며, 클래스별로 가중치의 합을 구한 후 합이 가장 큰 클래스로 테스트 패턴을 분류한다.

### 2.2 EACH 시스템

Nested Generalized Exemplar (NGE) 이론은 예들로부터의 귀납적 학습이 증가하는 형태로 Nearest Neighbor 분류 방법을 확장한 것이다[5][7]. NGE는 클래스 표본들에 기반하는 유도된 가정이 n 차원 유클리드의 공간에서 일련의 초월평면들의 그래픽의 모양을 가지는 학습 패러다임이다. 클래스들의 표본들은 초월평면들이거나 점으로 표현되는 하나의 학습 예제가 된다. EACH 시스템은 NGE 이론을 구현한 시스템으로 학습패턴을 그대로 저장하는 것이 아니라, 인접한 학습패턴들을 포함하는 초월평면(Hyperrectangle)의 형태로 저장하며, 그 결과 표본으로 저장되어 k-NN 기법보다 적은 메모리를 사용한다[5][7][10][11].

다음의 그림 2는 EACH 시스템의 알고리즘을 보여준다.

① 무작위로 몇 개의 학습패턴을 시드(seed)로 선택하여 예제(Exemplar)로 저장한다.
② 학습패턴을 선택하고, 가장 가까운 예제를 검색한다.
③ 학습패턴의 클래스와 가장 가까운 예제의 클래스가 동일하면, 학습패턴을 이용하여 그 예제를 확장하고 예제의 가중치를 수정한 한 다음, 단계 ⑥을 수행한다.
④ 클래스가 다를 경우, 가중치를 수정하고 두 번째로 가까운 예제를 선택한다.
⑤ 학습패턴의 클래스와 두 번째로 가까운 예제의 클래스가 동일하면, 예제를 확장하고 가중치를 수정하며, 다를 경우, 학습패턴을 별도의 새로운 예제로 저장한다.
⑥ 학습패턴 집합이 공집합이 될 때까지 단계 ②~⑤를 반복한다.

그림 2. EACH 시스템
Figure 2. EACH system

알고리즘에서 EACH 시스템의 학습이 종료되면, 학습패턴들은 예제의 집합으로 표현된다. 예제는 점 또는 초월평면의 형태를 취하게 되며 테스트 패턴은 가장 가까운 예제의 클래스로 분류한다. 예제가 점(point)일 경우에는 점과의 거리를 계산하며, 초월평면일 경우에는 가까운 면과의 거리를 계산한다.

### 2.3 고정분할평균(FPA) 기법과 재귀분할평균(RPA) 기법

고정분할평균(FPA: Fixed Partition Averaging) 기법은 주어진 패턴공간을 동일한 크기의 분할영역들로 분할한 후 패턴 평균기법을 적용하는 방법이다. 즉, 각 축을 식 (2)에

서와 같이 같은 크기의 N개로 분할한 후, 분할영역 단위로 패턴 평균법을 적용한다. 이때, 여러 클래스의 패턴이 존재하는 분할영역의 경우에는 패턴 평균법을 적용하지 않고 원래의 패턴들을 그대로 저장하며, 단일 클래스의 경우는 해당 분할영역 내의 모든 패턴을 평균하여 하나의 대표패턴으로 대체하는 방법을 사용한다. 이때, n은 하나의 패턴을 구성하는 특징 개수, |T|는 전체 학습패턴의 개수이다[10].

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

이와 같이 FPA 기법은 패턴공간을 동일한 크기로 분할하기 때문에 패턴의 분포를 고려할 수 없어 대표패턴의 개수는 증가하고 성능은 저하되는 문제점을 가지고 있다.

재귀분할평균(RPA: Recursive Partition Averaging) 기법은 주어진 패턴공간을 재귀적으로 분할해 나가면서 대표패턴을 추출하는 방법이다[11]. 즉, 주어진 패턴공간의 각 특징 축을 최초 2개의 영역으로 분할한다. 따라서 첫 번째 분할에서는 패턴공간이 $2^n$개의 공간으로 분할되며, 이때 $n$은 패턴을 구성하는 특징의 개수 즉, 패턴공간의 차원수가 된다. 따라서 2차원 패턴의 경우, 최초 4개의 분할영역으로 분할되며, 현재 분할영역 각각에 대하여 재귀 분할 여부를 결정한다. RPA에서는 하나의 분할영역에 소속되는 패턴의 클래스가 모두 같을 경우, 해당 분할영역의 패턴들에 대하여 패턴평균법을 적용하여 대표 패턴을 추출한다. 반면에 분할영역에 소속된 패턴들의 클래스가 여러 개로 혼합되어 있을 경우, 해당 분할영역을 다시 분할하는 방법이다[11].

RPA기법[10]와 FPA기법[11] 모두 패턴의 분포를 고려하지 않아 클래스가 혼합된 영역에 대해서는 점점 세밀하게 동일한 간격으로 분할해 나가게 되므로, 클래스 경계면에 위치한 분할영역의 경우 많은 분할이 이루어지게 되어 대표패턴의 개수가 많게 된다.

## 3. DPA(Dynamic Partition Averaging) 기법

본 논문에서 제안하는 동적분할평균(DPA : Dynamic Partition Averaging) 기법은 그림 3과 같이 두 단계로 구성되어 있다. 첫 번째 단계는 학습패턴공간을 다수의 분할영역들로 분할하는 동적분할단계이다. 즉 분할 시에 특징축을 모든 특징에 대해 2.3의 FPA에서 사용한 것같이 일정한 크기의 여러 개로 분할하지 않고 현재의 패턴들의 분포를 고려하여 동적인 크기의 여러 개로 분할한다. 그리고 각 분할영역에 포함된 모든 학습패턴의 클래스를 검사하여 학습패턴의 클래스가 동일한 경우는 대표패턴을 생성하고 종료하며, 서로 다른 클래스에 속하는 패턴들이 혼재되어있는 경우는 두 번째 단계를 실시한다. 즉 분할영역에 속한 패턴들의 분포를 고려하여 가장 효율적인 하나의 분할점을 정해 가변크기 재귀이분할해가면서 대표패턴을 생성한다. 이 때 대표패턴은 패턴평균기법을 이용하여 계산한다.

그림 4는 DPA 기법에 의해 분할된 특징의 개수가 2개인 2차원 학습패턴공간의 예제를 나타낸다. 첫 번째 단계에서 특징 축을 3개의 분할점을 선정하여 가변크기 분할하여 16개의 분할영역을 생성하였다. 그 중 13개의 영역은 단일 클래스의 영역으로 대표 패턴을 생성하고, 나머지 3개의 회색으로 표시된 분할영역은 서로 다른 클래스에 속하는 패턴들이 혼재되어있는 분할영역으로 각 특징 축을 최적의 하나의 분할점을 찾아 가변크기 이분할하여 2개의 분할영역을 생성한다. 생성된 분할영역을 다시 조사하여 혼재된 클래스가 더 이상 생성되지 않을 때까지 재귀적으로 가변크기 이분할을 실시한다. 테스트 패턴 분류 시 특징별 최종분할공간에 대한 GINI-Index값을 특징 가중치로 이용하여 분류정확성을 높였다.

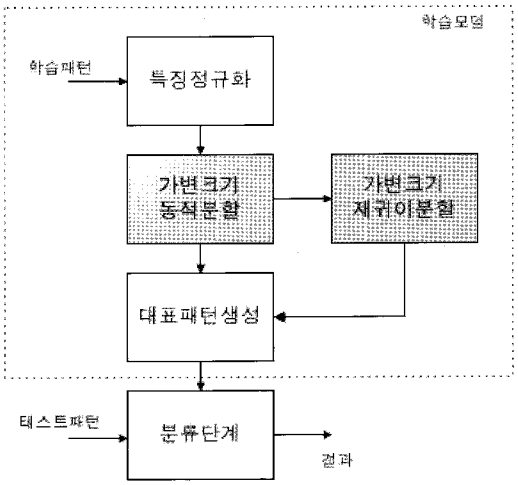


그림 3. DPA 학습모델
Figure 3. Model of DPA Algorithm.

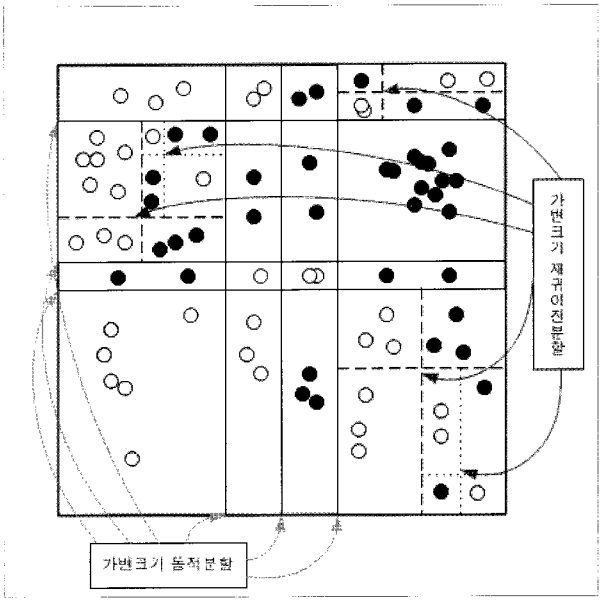


그림 4. DPA 알고리즘의 2차원 패턴공간 분할
Figure 4. Partition of 2 Dimensions Pattern Space in the DPA Algorithm.

### 3.1 특징의 정규화

인스턴스 기반 추론에서 출력 클래스의 결정은 입력패턴과 메모리에 저장된 학습패턴 사이의 거리를 이용하게 된다. 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 다음의 식 (2)를 이용하여 특징 값을 정규화 한다. 이 기법은 패턴을 구성하는 모든 특징 값을 0과 1사이의 값으로 정규화 함으로써, 모든 특징의 변화가 패턴의 소속 클래스 결정에 미치는 영향력을 동일하게 한다[10][11].

$$f_{i_n} = \frac{f_i - f_{i_{\min}}}{f_{i_{\max}} - f_{i_{\min}}} \tag{3}$$

이 때 $f_i$는 $i$번째 특징 값, $f_{i_{\max}}, f_{i_{\min}}$는 각각 $f_i$가 가질 수 있는 최대값과 최소값을 나타낸다.

### 3.2 동적분할

동적분할은 모든 특징에 대해 동일한 크기의 일정 개수로 특징축을 분할하는 것이 아니라 각 특징에 나타나는 패턴들을 고려하여 여러 개로 동적분할을 한다. 즉 각 특징축에 대해 가장 많은 패턴을 포함한 적절한 경계값을 구하여 분할점을 선정하여 분할하는 것이다. 따라서 특징에 존재하는 특징값의 분포를 구하고, 특징값과 특징값 사이의 값을 경계값으로 정한다.

다음 표 1은 breast-cancer-wisconsin 학습 자료의 첫 번째 특징에 대해 3.1의 정규화 과정을 실시한 후 특징값, 패턴의 개수, 그리고 경계값을 구한 예이다. 표 3에서 특징값 0.1111에 대한 패턴의 개수가 32개이고 특징값 0.2222에 대한 패턴의 개수가 37개 이므로 0.1111과 0.2222를 분할하는 경계값은 두 특징값을 더한 후 2로 나눈 0.1667이 된다.

표 2. 경계값의 계산
Talbe 2. Boundary Values

| 특징값 | 0 | 0.1111 | 0.2222 | 0.3333 | 0.4444 | 0.5556 | 0.6667 | 0.7778 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| 패턴개수 | 509 | 32 | 37 | 13 | 5 | 3 | 8 | 8 | 14 |
| 경계값 | 0.0556 | 0.1667 | 0.2778 | 0.3889 | 0.5 | 0.6111 | 0.7222 | 0.8889 | 1 |

구한 경계값들 $GINI_{split}$ 값을 이용하여 가장 변별력이 좋은 일련의 경계값을 분할점으로 선택한다[8]. $GINI_{split}$ 값은 식 (4), (5)을 이용하여 계산한다.

$$GINI(t) = 1 - \sum_{j=1}^{C} [p(j|t)]^2 \tag{4}$$

$p(j|t)$는 학습패턴 집합에서 분할점 t에서 클래스 j의 상대빈도이며, $C$는 클래스의 개수를 의미한다.

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i) \tag{5}$$

$n_i$는 분할된 영역에 포함된 학습패턴의 개수이고, $n$은 분할되기 전의 영역에 포함된 학습패턴의 개수이다. 식 (4)에서 GINI(t) 값은 $1 - \frac{1}{n_c}$인 경우, 즉 모든 클래스의 패턴이 골고루 분포되었을 경우이다. 반대로 0.0인 경우는 구간에 포함된 패턴들이 하나의 클래스에 소속될 때를 의미한다. 결국 $GINI_{split}$값이 큰 경계값을 분할점으로 선택할 때 효율적인 분할이 가능하다.

표 2. 경계값에 따른 $GINI_{split}$ 값
Table 2. $GINI_{split}$ Values in Boundary Values

| 경계값 | 0.0556 | 0.1667 | 0.2778 | 0.3889 | 0.5 | 0.6111 | 0.7222 | 0.8889 |
|---|---|---|---|---|---|---|---|---|
| $GINI_{split}$ | 0.2206 | 0.2237 | 0.2677 | 0.2691 | 0.3137 | 0.3487 | 0.3684 | 0.3753 |

표 2는 표 1에서 구해진 각각의 경계값을 기준으로 식 (5)를 이용하여 계산한 $GINI_{split}$값이다. 따라서 구해진 $GINI_{split}$값 중 최적의 값을 식 (2)에서 특징축의 분할 개수 (N) 만큼 경계값을 선택하여 분할점으로 선정하고 분할을 실시한다. 표 3은 최종적으로 최적의 $GINI_{split}$값으로 선택된 N=3인 경우의 특징축 분할점이다.

표 3. 최적의 $GINI_{split}$값에 의해 N=3로 선택된 분할점
Table 3. Partition Points by Best 3 $GINI_{split}$ Values

| 분할점 | 0.6111 | 0.7222 | 0.8889 |
|---|---|---|---|
| $GINI_{split}$ | 0.3487 | 0.3684 | 0.3753 |

모든 특징별 특징축의 분할점을 이용하여 주어진 학습패턴공간을 분할하고 각 분할영역에 포함된 현재의 학습패턴이 속한 클래스를 검사하여 학습패턴의 클래스가 동일한 경우는 대표패턴을 생성하고 종료하며, 서로 다른 클래스에 속하는 패턴들이 혼재되어있는 경우는 3.3의 두 번째 단계를 실시한다.

표 4는 본 논문에서 제안한 DPA 기법의 첫 번째 단계의 동적분할의 알고리즘을 보여준다.

표 4. 동적분할 알고리즘
Table 4. Dynamic Partition Algorithm

**초기화**

① 전체 패턴 집합을 정규화한다.
② 패턴 집합을 학습패턴과 테스트 패턴 집합으로 분리한다.
③ 전체 학습패턴 집합을 포함하는 영역을 식 (5)의 패턴공간을 구성하는 특징축의 분할 개수 N을 결정한다.

**학습 알고리즘**

① 모든 특징축에 대해
 (ㄱ) 표 3과 같이 특징값 분포를 구한다.
 (ㄴ) 특징값 분포를 이용해 표 3과 같이 구간값을 구한다.
 (ㄷ) 표 3의 각 구간값을 기준으로 표 4의 $GINI_{split}$ 값을 구한다.
 (ㄹ) $GINI_{split}$ 값을 크기 역순으로 초기화 ③의 N개를 표 5와 같이 선택하여 각각의 구간값을 선택한다.
② 단계 ①에서 선택된 N개의 구간값을 기준으로 패턴공간을 N분할을 실시한다.
③ 모든 분할영역에 대해 서로 다른 클래스의 학습패턴이 같은 분할 영역에 존재하는지 검사한다.
④ 포함된 학습패턴의 클래스가 동일하면 패턴평균법으로 대표패턴을 생성하고 종료한다.
⑤ 만약 클래스가 다른 학습패턴이 존재하면, 표 7의 재귀이분할 알고리즘을 재귀 호출한다.

### 3.3 재귀이분할

첫 번째 단계에서는 분할된 각각의 분할영역에 포함된

모든 학습패턴의 클래스를 검사하여 학습패턴의 클래스가 동일한 경우는 대표패턴을 생성하고 종료하였다. 반면 분할영역에 서로 다른 클래스에 속하는 패턴들이 혼재되어있는 경우는 패턴들의 분포를 고려하여 가장 효율적인 분할점을 정해 가변크기 재귀이분할해 가면서 대표패턴을 생성한다. 다음 표 5는 3.2에서와 같은 breast-cancer-wisconsin 학습자료의 첫 번째 특징에 대해 DPA의 첫 번째 단계를 실행한 후 생성된 다중 클래스 분할영역에 포함된 패턴들의 특징값, 패턴의 개수, 그리고 경계값을 구한 예이다. 그리고 표 6은 표 5에서 구해진 각각의 경계값을 기준으로 식 (5)를 이용하여 계산한 $GINI_{split}$값이다. 따라서 구해진 $GINI_{split}$값 중 최적의 값을 하나 선택하여 그 경계값을 분할점으로 분할을 실시한다. 현재의 분할 영역에 포함된 패턴들의 모든 특징별 특징축의 분할점을 이용하여 패턴공간을 분할하고 생성된 각 분할영역에 포함된 학습패턴이 속한 클래스를 검사하여 학습패턴의 클래스가 동일한 경우는 대표패턴을 생성하고 종료하며, 서로 다른 클래스에 속하는 패턴들이 혼재되어있는 경우 분할된 분할영역에 더 이상의 서로 다른 클래스의 패턴이 소속되지 않을 때까지 계속한다.

표 5. 분할영역의 경계값 계산
Talbe 5. Boundary Values

| 특징값 | 0.2222 | 0.3333 | 0.4444 | 0.5556 | 0.7778 | 0.8889 |
|---|---|---|---|---|---|---|
| 패턴개수 | 15 | 13 | 8 | 1 | 1 | 2 |
| 경계값 | 0.2778 | 0.3889 | 0.5 | 0.6667 | 0.8333 | 1 |

표 6. 경계값에 따른 $GINI_{split}$값
Table 6. $GINI_{split}$ Values in Boundary Values

| 경계값 | 0.2778 | 0.3889 | 0.5 | 0.6667 | 0.8333 |
|---|---|---|---|---|---|
| $GINI_{split}$ | 0.132 | 0.1125 | 0.0986 | 0.082 | 0.0487 |

표 7. 재귀이분할 알고리즘
Table 7. Recursive Binary Partition Algorithm

① 현재 분할영역에 포함된 모든 학습패턴의 클래스를 검사한다.
② 만약 모든 학습패턴의 클래스가 동일하면, 패턴평균법으로 대표패턴을 생성하고 종료한다.
③ 만약 클래스가 다른 학습패턴이 존재하면, 현재 분할영역의 특징별로 새로운 경계값을 구하고, 이 중에서 가장 효율적인 경계값을 분할점으로 선정한다.
④ 단계 ③에서 선정된 분할점을 이용하여 새로운 영역들로 분할한다.
⑤ 단계 ④의 분할영역 중 한 개 이상의 학습패턴을 포함하는 모든 분할영역에 대하여 위의 학습 알고리즘을 재귀호출한다.

표 7은 본 논문에서 제안한 DPA 기법의 두 번째 단계의 재귀이분할 알고리즘을 보여준다.

### 3.4 대표패턴의 생성

대표패턴의 생성은 3.2의 동적분할의 표 6의 학습알고리즘 단계 ④와 3.3의 재귀이분할의 표 9의 단계②의 패턴평균법은 같은 클래스의 학습패턴들을 평균하여 하나의 대표패턴을 만들어 대체하는 방법으로 각각의 특징값들에 대해 평균한다.

### 3.5 DPA 학습기법 패턴분류

테스트 패턴을 분류하기 위하여 대표패턴들과 수식 (7)로 거리 계산을 하며, 가장 가까운 대표패턴의 클래스를 출력으로 결정한다. 거리의 계산에는 분류성능 향상을 위하여 학습패턴의 최종적으로 생성된 분할영역에 대응하는 식 (4)의 $GINI_{split}(i)$값 구해 입력패턴과 메모리에 저장된 학습패턴간의 거리계산에 있어 특징의 가중치로 사용한다.

$$D_{EQ}=\sqrt{\sum_{i=0}^{n} GINI_{split}(i)(E_{f_i}-Q_{f_i})^2} \qquad (7)$$

## 4. 실험 및 분석

본 논문에서 제안한 DPA 기법의 성능을 Stratified 10-fold Cross-validation 기법을 사용하여 k-NN, EACH, FPA와 RPA 등의 알고리즘에 대해 비교 검증하였다.

### 4.1 실험 데이터

본 논문에서는 기계 학습의 벤치마크 자료로 많이 사용되는 UCI Machine learning Database Repository에서 6개의 데이터 셋을 발췌하여 사용하였다[10][11][14]. 이들 데이터는 모든 특징이 실수 값을 갖는다. 다음의 표 8은 실험자료의 분포를 보여주고 있다.

표 8. 클래스별 학습패턴의 분포
Table 8. Training Patterns in Classes

| 데이터 셋 | 패턴 개수 | 특징 개수 | 클래스 별 패턴 개수 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 |
| Breast-Cancer | 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 데이터 셋은 Wisconsin 대학병원의 William H. Wolberg 박사가 정리한 유방암 진단 자료이며[13], Glass 데이터 셋은 범죄 수사 연구에 사용하기 위해서 유리를 분석한 자료이다. Ionosphere 데이터 셋은 Goose Bay에서 수집된 레이더 데이터이며, Iris 데이터 셋은 패턴인식 분야에서 가장 많이 사용되는 꽃잎과 꽃받침의 길이와 너비 수치를 기반으로 식물의 종류를 판별하는 데이터 셋이다. New-Thyroid 데이터 셋은 갑상선 진단 자료이며, Wine 데이터 셋은 이탈리아의 동일 지역에서 세 가지 다른 품종으로 재배된 와인의 화학적 분석 결과이다.

### 4.2 분류성능

분류 성능 실험에서 k-NN 기법은 Leave-one-out Cross-validation 기법으로 계산한 최적의 k값을 사용하였

으며[9], 가중치 변화량 0.2를 초기값으로 설정하여 실험하였다. 다음 표 9는 각 데이터 셋에서 사용된 k-NN 기법의 k값과 k값을 계산하기 위하여 사용된 시간을 나타낸다.

표 9. 분류성능 최적화를 위한 k값 및 계산 시간 (Hour)
Table 9. k Value and Hour for kNN Method

| 데이터셋 | Breast Cancer | Glass | Ionosphere | Iris | New Thyroid | Wine |
|---|---|---|---|---|---|---|
| k값 | 21 | 1 | 1 | 51 | 1 | 19 |
| 시간 | 261 | 2.26 | 40.56 | 0.33 | 1.61 | 1.29 |

그림 5의 결과는 논문에서 제안한 DPA 기법이 k-NN, EACH 시스템, FPA 기법, 그리고 RPA기법과 비교하여 유사하거나 향상된 분류 성능을 보여주고 있다. EACH 시스템의 Ionosphere에서 저조한 성능을 보이는 것은 무작위(Random)로 설정한 초기 시드(seed)의 영향으로 분석되며, 본 논문에서 제안한 기법이 EACH 시스템보다 모든 데이터 셋에서 안정적인 성능을 보여준다. 표 10은 분류 성능에 대한 표준편차를 보여준다.

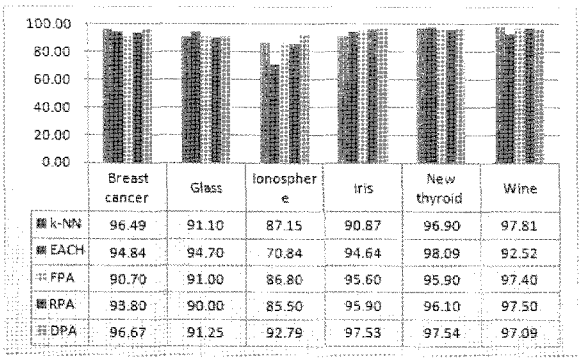


그림 5. 분류성능
Figure 5. Performances

표 10. 분류 성능에 대한 표준편차
Table 10. Standard Deviations of the Performances

| 자료명 | Breast-cancer | Glass | Ionosphere | Iris | New-thyroid | Wine |
|---|---|---|---|---|---|---|
| k-NN | 2.24 | 5.37 | 5.08 | 7.16 | 3.66 | 3.57 |
| EACH | 3.66 | 5.19 | 18.13 | 5.58 | 4.84 | 6.29 |
| FPA | 2.2 | 4.94 | 4.65 | 4.45 | 5.03 | 4.32 |
| RPA | 2.5 | 5.33 | 4.73 | 5.85 | 5.24 | 4.25 |
| DPA | 2.27 | 8.76 | 4.07 | 3.98 | 3.40 | 3.84 |

### 4.3 메모리 사용량 비교

그림 6은 각 기법이 사용한 메모리 사용량을 보여주고 있으며, 표에 나타난 수치는 메모리에 저장된 학습 패턴의 개수를 의미한다. 이때 EACH 시스템의 경우는 메모리에 저장된 분할영역의 수 × 2를 저장된 학습패턴의 수로 사용하였는데, 이는 EACH시스템에서 메모리에 저장되는 분할영역이 평면의 범위를 나타내는 상, 하한의 두 개의 패턴으로 표시되기 때문이다.

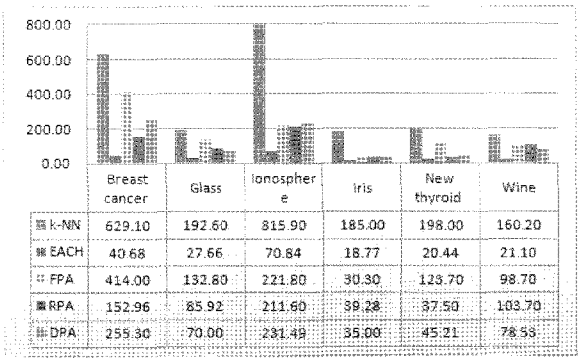


그림 6. 메모리 사용량
Figure 6. Memory Usages

FPA와 RPA, 그리고 DPA 기법은 k-NN 기법보다 메모리 사용이 우수하지만 EACH 시스템과 비교할 때, 대표패턴 개수가 전반적으로 많이 생성된다. FPA는 클래스 경계면이 특징 축과 평행하게 분포할 경우에 대해, 그리고 RPA는 특징축과 클래스 경계면이 평행하지 않은 데이터 셋에 있어서도 재귀이분할을 통한 패턴평균법을 적용하여 메모리 사용효율을 얻을 수 있다. 그러나 DPA는 GINI-Index 수치를 이용한 패턴의 분포를 고려하여 특징축을 분할하여 우수한 메모리 사용효율을 보여주었다.

### 4.4 여러 클래스가 혼재하는 분할영역의 비교

표 11은 학습에 의해 분할된 영역에 서로 다른 클래스의 패턴이 존재하는 경우(이하 혼합셀, multiclass cell)를 분석한 것이다. FPA와 RPA, 그리고 DPA는 학습패턴 공간을 분할하여 학습하고 분류하는 개념으로 학습 후 분할된 공간들에 서로 다른 혼합셀을 비교할 수 있다. 표에서 보면 DPA의 경우 각 실험 자료에 대해 FPA나 RPA에 비해 전반적으로 안정적인 결과를 만들었다.

표 11. 혼합셀에 포함된 패턴 현황(괄호안의 숫자 : 표준편차)
Table 11. the number of Patterns in the Partition Spaces of the multi class cell(Standard Deviation)

| 자료명 | Breast-cancer | Glass | Ionosphere | Iris | New-thyroid | Wine |
|---|---|---|---|---|---|---|
| FPA | 250 (14.03) | 65.3 (15.14) | 76.8 (4.86) | 16.9 (4.21) | 23.2 (31.4) | 55.1 (5.75) |
| RPA | 155.8 (7.03) | 56.7 (8.1) | 65.2 (5.4) | 17.2 (4.0) | 34.5 (3.2) | 68.4 (2.3) |
| DPA | 39.1 (7.93) | 10.79 (9.8) | 0 (0) | 10.38 (4.15) | 0.1 (0.43) | 1.44 (1.99) |

## 5. 결 론

본 논문에서 제안한 DPA 기법은 FPA와 RPA기법을 문제점을 해결하기 위하여 제안되었다. FPA 기법은 클래스가 혼재된 혼합셀의 경우에 원본 학습 패턴을 그대로 저장하는 방법을 사용하며, RPA 기법은 재귀이분할을 통한 방법을 사용한다. 이는 공간의 과다분할 등 메모리 사용 효율을 저하시키며, 분류시간이 증가되는 결과를 초래한다. 또한

FPA 기법 및 RPA 기법 모두 분할점을 선택할 때 패턴의 분포를 고려하지 않기 때문에 클래스가 혼재된 분할영역이 많이 나타난다.

그러나 DPA 기법은 분할 시 패턴의 분포를 고려하여 분할점을 선정하는 가변크기 분할을 실시한 후, 클래스가 혼재된 분할영역이 발견되면 혼재된 분할영역을 재귀적으로 가변크기 이분할하는 방법을 사용한다. 따라서 성능과 메모리 사용에 있어 안정적이고 효율적인 분할이 이루어질 수 있도록 하였다.

본 논문에서 제안한 DPA 기법은 k-NN, FPA, RPA 기법, 그리고 EACH 시스템 등과 비교하여 유사하거나 향상된 분류 성능을 보여주었다. 그리고 메모리 사용량에 있어서도 k-NN, FPA 기법, 그리고 RPA 기법 보다 줄어드는 것을 확인할 수 있었다.

## 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, "Weighted kNN versus Majority kNN :A Recommendation," *German National Research Center for Information Technology*, 1995.

[3] D. Wettschereck, "A Hybrid Nearest-Neighbor and Nearest-Hyperrectangle Algorithm," *Proceedings of the 7th European Conference on Machine Learning*, 1995.

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

[5] 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.

[6] D. Wettschereck and T. Dietterich, "Locally Adaptive Nearest Neighbor Algorithms," *Advances in Neural Information Processing Systems 6:* 184-191, 1994.

[7] S. Salzberg, "A Nearest Hyperrectangle Learning Method," *Machine Learning*, Vol. 6. No. 3. pp. 251-276, 1991.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees.* Wadsworth International Group, 1984.

[9] 심범식, 정태선, 윤충화, "최근접 초월평면 학습법에서 시드개수의 영향에 대한 분석", *한국정보처리학회*, '98 춘계학술대회, 1998.

[10] 정태선, 이형일, 윤충화, 고정 분할 평균알고리즘을 사용하는 새로운 메모리 기반 추론, *한국정보처리학회 논문지* 제6권 제6호, pp. 1563-1570, 1999.

[11] 이형일, 정태선, 윤충화, 강경식, 재귀 분할 평균기법을 이용한 새로운 메모리 기반 추론 알고리즘, *한국정보처리학회 논문지* 제6권 제7호, pp. 1849-1857, 1999.

[12] 이형일, 초월평면 최적화를 이용한 최근접 초월평면 학습법의 성능 향상 방법, *한국퍼지및지능시스템학회논문지*, 2003, 13(3), pp.328-333

[13] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming", *SIAM News*, Volume 23, Number 5, September 1990, pp 1 & 18.

[14] http://www.ics.uci.edu/~mlearn


## 저 자 소 개


**이 형 일**(Hyeong-il, Yih)

2000. 8. : 명지대학교 대학원 컴퓨터공학과 박사

1984.12.~1989. 11. : (주)쌍용정보통신

1990. 5.~1994. 8. : (주)시에치노컨설팅

1997. 3.~현재 : 김포대학 인터넷정보과 부교수

관심분야 : 에이전트시스템, 기계학습, 미디어 영상인식, 패턴인식

Phone : 031-999-4173

E-mail : hilee@kimpo.ac.kr