Journal of Korea Multimedia Society Vol. 10, No. 11, November 2007(pp. 1516-1522)

반복적 고정분할 평균기법을 이용한 메모리기반 학습기법

이 형 일†

요 약

FPA(Fixed Partition Averaging) 기법은 기억공간의 효율적인 사용과 분류성능의 향상을 위하여 제안되었던 메모리 기반 추론 기법으로 대상 패턴 공간을 분할 한 후 대표 패턴을 추출하여 분류 기준 패턴으로 사용한다. 이 기법은 메모리 사용 효율과 분류 성능 면에서 우수한 결과를 보인다. 그러나 여러 클래스가 혼합된 분할패턴공간의 경우에 원래의 패턴들을 그대로 저장하여 메모리와 분류성능에 부담으로 작용하는 문제점을 가지고 있다. 본 논문에서는 여러 클래스가 혼합된 분할공간에서 패턴비율을 고려하여 고정분할을 반복적으로 실행하여 초월평면을 생성하고 분류하는 반복적 고정분할평균기법을 제안한다. 본 논문에서 제안한 기법은 기존의 k-NN 기법과 비교하여 현저하게 줄어든 대표패턴을 이용하여 유사한 분류 성능을 보여주며, NGE 이론을 구현한 EACH 시스템과 FPA 기법 등과 비교하여 탁월한 분류 성능을 보여준다.

A Memory-based Learning using Repetitive Fixed Partitioning Averaging

Hyeong-il Yih†

ABSTRACT

We had proposed the FPA(Fixed Partition Averaging) method in order to improve the storage requirement and classification rate of the Memory Based Reasoning. The algorithm worked not bad in many area, but it lead to some overhead for memory usage and lengthy computation in the multi classes area. We propose an Repetitive FPA algorithm which repetitively partitioning pattern space in the multi classes area. Our proposed methods have 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.

Key words: Memory-Based Reasoning(메모리기반 추론), Distance-Based Learning(거리기반학습), Instance-Based Learning(인스턴스 기반 학습)

1. 서 론

메모리 기반 학습은 주어진 학습패턴 자체를 모두 메모리에 저장하고, 입력 패턴과 저장된 패턴들 사이의 거리를 이용하여 패턴을 분류하여 거리기반 학습(Distance Based Learning) 이라고도 한다[1,2].

메모리 기반 학습 방식 중에서 가장 널리 알려진 기법은 k-NN(k-Nearest Neighbors) 분류기를 들 수 있으며, 이 분류기는 메모리에 저장된 학습패턴 중 주어진 입력패턴과 가장 가까운 거리에 있는 k개의 학습패턴을 선택하여 그 중 가장 많은 패턴이 소속된 클래스로 입력패턴을 분류한다[2-4]. 이러한 k-NN 분류기는 그 성능 면에서 만족할 만한 결과를 보이고 있으며, 이미 다양한 분야에 응용되고 있다. 하지만 이 기법의 가장 큰 문제점은 학습 패턴 전체를 메모리에 저장하여야 하므로 다른 기계학습 방법에 비하여 많은 메모리 공간을 필요로 하며, 저장되는 학습 패턴이 증가할수록 분류에 필요한 시간도

※ 교신저자(Corresponding Author) : 이형일, 주소 : 서울시 영등포구 영등포동 8가 91 당산푸르지오아파트 101-304 (150-038), 전화 : 031)999-4173, FAX : 031)999-4775, E-mail : hiyih@paran.com

접수일 : 2007년 6월 4일, 완료일 : 2007년 10월 2일

† 정회원, 김포대학 인터넷정보과 부교수

맡이 소요된다는 단점을 갖는다[4]. 따라서 메모리 기반 학습기법이 갖고 있는 문제점을 해결하기 위한 연구가 지금까지 활발히 진행되어 오고 있으며, 대표적인 연구로 IBL(Instance Based Learning)[4], NGE(Nested Generalized Exemplar) 이론[5-7]과 FPA(Fixed Partition Averaging)[8,9] 등이 있다.

본 논문에서는 메모리 기반 학습 기법에서 보다 효율적인 메모리 사용과 분류 성능을 보장하기 위하여 반복적인 고정 분할 평균기법을 제안한다. RFPA 기법은 주어진 패턴공간을 FPA기법을 이용하여 분할(이하, 공간셀(Space Cell))한 후 같은 클래스가 아닌 패턴들을 가진 공간셀에 대해 패턴비율이 큰 구간들에 대해 반복적으로 고정 분할하여 초월 평면을 생성하고 분류하는 방법이다. 본 논문에서는 UCI Machine Learning Repository에서 벤치마크 데이터를 발췌하여 사용하였으며, 제안한 알고리즘과 k-NN 기법, EACH 시스템, 그리고 FPA 기법의 분류 성능, 메모리 사용 효율을 실험적으로 비교 검증하였다.

2. 관련 연구

2.1 k-NN 기법

k-NN 분류기는 메모리 기반 학습 기법으로 분류되는 대표적인 알고리즘이다. 이 분류기는 학습단계에서는 단순히 학습 패턴을 메모리에 모두 저장하고, 차후 입력패턴의 분류 단계에서 모든 필요한 계산이 수행되어 이를 Lazy learning Algorithm이라고도 부른다[9]. k-NN 분류기의 개략적인 알고리즘은 다음 표 1과 같다.

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

표 1. k-NN 기법

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.

이때, E는 메모리에 저장된 학습패턴을 나타내며, Q는 주어진 입력패턴이다. 또한 n은 패턴을 구성하는 특징의 개수이며, E_i, Q_i는 각각 학습패턴과 입력패턴의 i번째 특징 값을 나타낸다. 이 때 k값은 분류기의 성능을 최적화하기 위하여 일반적으로 Cross Validation기법을 사용하여 결정하며, k=1인 경우를 NN 분류기라 한다[2,3]. 또한 위의 과정 중 4번째 단계에서, 입력패턴과의 거리를 이용하여 가중치를 부여하는 방법을 Weight Vote k-NN이라고 하며, 클래스별로 가중치의 합을 구한 후 합이 가장 큰 클래스로 테스트 패턴을 분류한다[2].

2.2 EACH 시스템

NGE(Nested Generalized Exemplar) 이론에 기반한 학습 기법인 EACH시스템은 학습패턴을 그대로 저장하는 것이 아니라, 인접한 학습패턴들을 포함하는 초월평면 (Hyperrectangle)의 형태로 저장하며, 그 결과 k-NN 기법보다 적은 메모리를 사용한다[5,7]. 다음의 표 2는 EACH 시스템의 알고리즘을 보여준다.

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.

표 2. EACH 시스템

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.

2.3 고정 분할 평균 기법

고정분할평균(Fixed Partition Averaging) 기법은 주어진 패턴공간을 동일한 크기의 초월평면들로 분할한 후 패턴 평균기법을 적용하는 기법이다. 이 기법은 먼저 패턴 공간의 각 특징 축을 다음 식 (2)에 의해 일정한 크기로 분할한다[10].

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

이때, n은 하나의 패턴을 구성하는 특징 개수, |T|는 전체 학습패턴의 개수이다. 또한 전체 학습패턴의 30%에 근사한 초월평면을 형성하도록 선택하였다[10].

FPA 기법에서는 각 축을 같은 크기의 N개로 분할한 후, 분할된 초월평면 단위로 패턴 평균법을 적용한다. 그림 1은 패턴공간을 구성하는 2개의 축을 각각 10개의 영역으로 분할한 경우이다. 그림 1에서 회색으로 표시된 여러 클래스 혼합 부분의 경우에는 패턴 평균법을 적용하지 않고 원래의 패턴들을 그대로 저장하며, 단일 클래스의 부분은 해당 공간셀 내의 모든 패턴을 평균하여 하나의 대표패턴으로 대체하는 방법을 사용한다.

또한 FPA 기법에서는 분류기의 성능향상을 위하여 상호정보를 이용한 특징의 가중치를 사용한다. 특징과 클래스간의 상호정보는 해당 패턴이 클래스의 결정에 미치는 영향력으로 식 (3)과 (4)에 의해 계산된다[9].

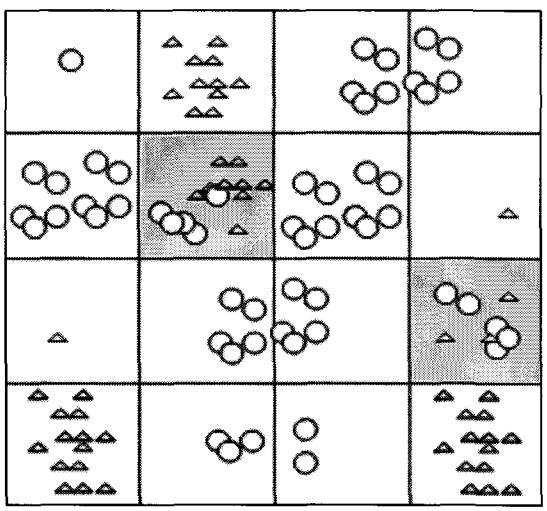


그림 1. 고정 분할 평균법(△: Class 1 o: Class 2)

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

이때 p_i는 전체 학습패턴 중 클래스 i에 소속되는 패턴의 비율, 즉 임의의 패턴이 클래스 i로 분류될 사전확률을 의미하며, C는 전체 학습패턴을 구성하는 클래스의 개수이다. FPA에서 특징 f의 가중치로 사용하는 상호정보이득은 (Mutual Information Gain) 다음의 식 (4)에 의해 계산된다[9].

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

이때 I는 식 (3)에서 정의한 특징 축 분할 이전에 필요한 정보의 양, N은 특징 축 f의 분할 개수이며, 이 값은 식 (2)에 의해 사전에 계산된다. I_i는 특징 f를 기준으로 분류했을 때 분할된 공간에서 필요한 정보의 량이며, 이 값은 식 (3)과 같은 방법을 사용하여 계산한다. 또한 P_i는 전체 학습패턴 중 분할된 초월평면에 할당된 패턴의 비율이다.

FPA 기법의 궁극적인 목표는 전체 학습 패턴을 거리계산에 사용하는 k-NN 기법의 분류성능에 근접하면서, 패턴 평균 기법을 사용하여 k-NN 기법에서 나타나는 메모리 공간의 낭비를 줄이는 것이었다. 그러나 FPA 기법은 특징 축을 고정 개수의 구간으로 일률적으로 분할할 경우, 서로 다른 클래스에 속하는 패턴들이 분할된 초월 평면에 소속될 수 있다. 이러한 결과는 학습후 생성되는 대표패턴을 증가시켜 메모리 사용량과 분류성능에 좋지 않은 영향을 미칠 수 있게 된다.

3. 반복적인 고정분할평균 기법

FPA 기법에서는 그림 1의 회색으로 표시된 서로 다른 클래스에 속하는 패턴들이 혼재되어있는 경우에는 패턴 평균법을 적용하지 않고 원래의 패턴들을 그대로 저장하며, 단일 클래스에 속하는 패턴들이 존재하는 부분은 해당 공간셀 내의 모든 패턴을 평균하여 하나의 대표패턴으로 대체하는 방법을 사용하였다. 보다 효율적인 메모리 사용과 분류 성능을 보장하기 위하여 제안한 반복적인 고정분할 평균(RFPA : Repetitive Fixed Partition Averaging) 기법을 제안한다. 이 기법은 주어진 패턴공간을 FPA 기법을 적용하여 분할한 한 후, 동일 클래스가 아닌 패턴들

이 포함된 공간셀에 대해서 반복적인 고정분할 평균기법을 적용하는 알고리즘으로 공간 분할 시 패턴의 분포를 고려하여 분할점을 선정한다.

3.1 패턴 비율에 기반한 패턴공간의 분할

FPA 기법은 주어진 패턴공간을 동일한 크기의 초월평면들로 분할하기위해 패턴 공간의 각 특징 축을 식 (2)에 의해 일정한 크기의 N개로 분할하였다. 이와 같은 분할은 패턴들의 특성을 고려하지 않아 패턴들이 밀집한 공간을 임의적으로 분할하는 경우가 발생할 수 있다. 따라서 모든 특징에 대해 패턴의 비율를 고려하여 새로운 분할점을 선택함으로써 이러한 문제를 해결할 수 있다. 새로운 분할점을 선정하기 위해서는 먼저, 모든 특징에 대하여 특징 값을 오름차순으로 정렬한 후 특징 값이 변하는 두 수의 중간값을 새로운 분할점의 후보로 선정한다. 선정된 후보값은 식 6의 분포값를 고려하여 d_i값이 높은 N_i개를 선택하며, 이때 N_i은 식 (5)를 이용하여 계산한다.

$$N_i = \lceil \log_n (0.3 \times |T_i|) \rceil \times n \tag{5}$$

$$d_i = \left| \frac{b_j}{N_i} \right| \times 100 \tag{6}$$

식 (5)는 식 (2)를 반복적으로 사용하기위해 재 정의한 식으로 N_i는 i번째 해당 공간셀의 분할 개수이며, n은 하나의 패턴을 구성하는 특징의 개수, 그리고 $|T_i|$는 i번째 해당 공간셀 내에 존재하는 학습패턴의 개수이다. 식 (6)의 N_i는 식 (5)와 같고, b_i는 분할대상 공간셀 내의 j 번째 대상 구간에 속하는 패턴의 개수이다.

그림 2는 특징의 개수가 2개인 패턴공간에서 패턴 비율에 기반한 패턴공간을 분할한 경우(검은색의 실선, 축번호 1, 2, 3)와 일정한 크기로 분할한 경우(적색의 점선, 축번호 1′, 2′, 3′)를 비교한 예이다.

3.2 반복적인 고정분할평균기법

FPA 기법의 학습이 종료되면 대표패턴이 생성된다. 이 중 패턴의 평균값으로 생성된 대표패턴 외에 패턴 그대로 저장된 학습패턴도 존재한다. 이와 같은 패턴 그대로 저장된 학습패턴의 비율이 전체 학습패턴에 대해 약 20%에 달한다. 본 논문에서는 이러한 학습패턴들이 속하는 공간셀에 대해서 고정분할 평균기법(FPA)을 수행하여 원래 패턴이 그대로 저장되는 학습패턴의 비율을 줄이고 그들을 평균하는 대표패턴을 늘이는데 목적이 있다. 즉 반복적 고정평균기법은 서로 다른 클래스가 혼재하는 공간셀에 대해 3.1에서 제안한 방법으로 고정 평균 분할을 계속하는 것으로 전체 분할된 공간셀에 대해 클래스가 혼재된 공간셀의 비율이 10%(실험적으로 수치임) 미만이 될 때까지 분할 과정을 반복한다.

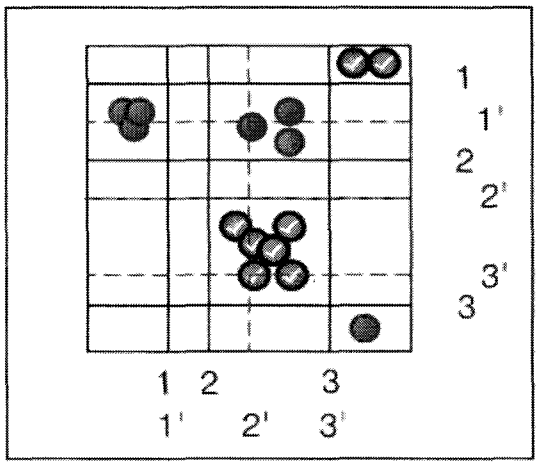


그림 2. 패턴비율에 기반한 패턴공간의 분할

그림 3은 클래스가 혼재된 공간셀을 반복적으로 분할하는 과정을 보여준다. 그림 3의 "1차 RFPA 구간"은 FPA 기법의 알고리즘을 적용한 결과이며, 2차 이후의 결과는 3.1절에서 설명한 패턴의 분포를 고려하여 선정된 분할점을 이용하여 공간셀을 분할한 결과를 보여준다.

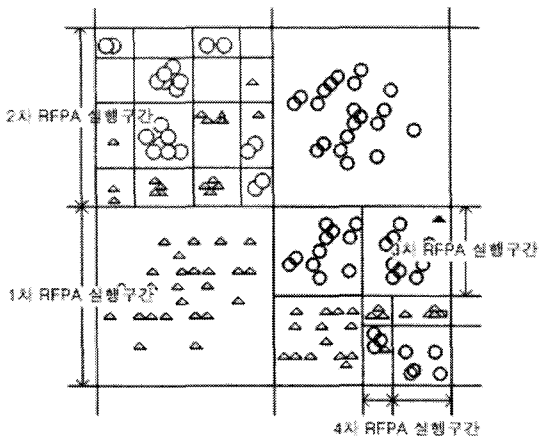


그림 3. RFPA 알고리즘의 패턴공간 분할과정

표 3은 본 논문에서 제안한 RFPA 기법의 알고리즘을 보여준다.

4. RFPA 학습기법 패턴분류

패턴 분류는 메모리 기반 알고리즘에서 사용하는 거리 기반 기법을 사용하여 표 6과 같이 분류한다. 거리의 계산에는 분류성능 향상을 위하여 식 (4)로 주어진 $IG(f)$값을 입력패턴과 메모리에 저장된 학습패턴간의 거리계산에 있어 특징의 가중치로 사용한다. 이때의 거리는 식 (7)에 의해 계산한다.

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

표 3. RFPA 기법

초기화 단계

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

학습 단계

① 현재 분할영역에 포함된 모든 학습패턴의 클래스를 검사한다.
② 만약 모든 학습패턴의 클래스가 동일하면 대표패턴 평균법으로 대표패턴을 추출하고 종료한다.
③ 만약 클래스가 다른 학습패턴이 존재하면, 식 (5)를 이용하여 새로운 분할 개수 Ni을 구하고 모든 특징에 대해 식 (6)의 패턴분포를 고려하여 분포값이 큰 구간을 Ni개를 선택하여 반복적으로 고정 평균 분할을 실시하고, 경계값을 분할점으로 선정한다.
④ 최초 분할 이전과 이후의 상호정보 이득을 이용하여 각 특징의 가중치를 결정한다.
⑤ 단계 ③의 분할영역에 존재하는 서로 다른 클래스의 패턴들의 비율이 10% 미만이 될 때까지 위의 학습 알고리즘을 반복하여 호출한다.

표 4. RFPA 분류알고리즘

분류 단계

① 메모리에 저장된 학습패턴을 입력패턴과의 식 (7)을 계산한다.
② 정렬된 학습패턴의 첫 번째 학습패턴의 클래스로 출력 클래스를 결정한다.

5. 실험 및 분석

본 논문에서 제안한 RFPA 기법의 성능을 k-NN, EACH, 그리고 FPA와 Stratified 10-fold Cross-validation 기법을 사용하여 비교 검증하였다.

5.1 실험 데이터

본 논문에서는 기계 학습의 벤치마크 자료로 많이 사용되는 UCI Machine learning Database Repository에서 6개의 데이터 셋을 발췌하여 사용하였다. 이들 데이터는 모든 특징이 실수 값을 갖는다. 다음의 표 5는 실험 자료의 분포를 보여주고 있다.

Breast-Cancer 데이터 셋은 Wisconsin 대학병원의 William H. Wolberg 박사가 정리한 유방암 진단 자료이며[11,12], 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.

분류 성능 실험에서 k-NN 기법은 Leave-one-out Cross-validation 기법으로 계산한 최적의 k값을 사용하였으며[10], 가중치 변화량 0.2를 초기값으로 설정하여 실험하였다. 다음 표 6은 각 데이터셋에서 사용된 k-NN 기법의 k값과 k값을 계산하기 위하여 사용된 시간을 나타낸다.

표 5. 클래스별 학습패턴의 분포

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	-	-	-

표 6. 분류성능 최적화를 위한 k값 및 계산 시간 (Hour)

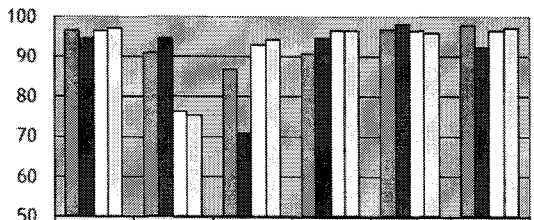
데이터셋	Breast Cancer	Glass	Ionosphere	Iris	New Thyroid	Wine
k값	21	1	1	51	1	19
시간	261	2.26	40.56	0.33	1.61	1.29

5.2 분류성능

그림 4의 결과는 논문에서 제안한 RFPA 기법이 k-NN 기법, EACH 시스템, 그리고 FPA 기법과 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 7은 분류 성능에 대한 표준편차를 보여준다.

5.3 메모리 사용량 비교

그림 5는 각 기법이 사용한 메모리 사용량을 보여주고 있으며, 표에 나타난 수치는 메모리에 저장된

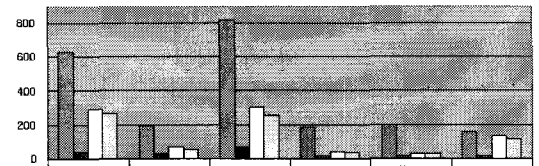


	Breast cancer	Glass	Ionosphere	Iris	New thyroid	Wine
k-NN	96.49	91.1	87.15	90.87	96.9	97.81
EACH	94.84	94.7	70.84	94.64	98.09	92.52
FPA	96.637892	76.419787	93.108737	96.733322	96.642849	96.513859
RFPA	97.25	75.36	94.357	96.53	95.9	97.2

그림 4. 분류성능

표 7. 분류 성능에 대한 표준편차

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
RFPA	2.5	5.33	4.73	5.85	5.24	4.25



	Breast cancer	Glass	Ionosphere	Iris	New thyroid	Wine
k-NN	629.1	192.6	615.9	185	198	160.2
EACH	40.68	27.66	70.84	18.77	20.44	21.1
FPA	294.6	70.7	301.5	41.9	31.2	135.5
RFPA	267.2	55.2	254.6	95.7	27.7	115

그림 5. 메모리 사용량

학습 패턴의 개수를 의미한다. 이때 EACH 시스템의 경우는 메모리에 저장된 초월평면의 수 × 2를 저장된 학습패턴의 수로 사용하였는데, 이는 EACH시스템에서 메모리에 저장되는 초월평면이 평면의 범위를 나타내는 상, 하한의 두 개의 패턴으로 표시되기 때문이다.

FPA와 RFPA 기법은 k-NN 기법보다 메모리 사용이 우수하지만 EACH 시스템과 비교할 때, 대표패턴 개수가 전반적으로 많이 생성된다. 특히 FPA 경우 다른 데이터셋(학습패턴에 대한 대표패턴 비율이 35% 정도, 표준편차 1.5)에 비교하여 메모리 사용량이 큰 Ionosphere와 Wine(학습패턴에 대한 대표패턴 비율이 각각 95%와 88% 정도, 표준편차 0.7, 2.3)는 기법의 특성상 특징의 개수가 많아 과다한 분할이 발생하기 때문이다.

5.4 여러 클래스가 혼재하는 공간셀의 비교

표 8은 본 논문의 주된 목적중의 하나인 같은 공간셀 내에 다른 클래스의 패턴이 존재하는 경우 반복적으로 구간을 분할하는 것이다. 따라서 FPA 실행 후 혼합 셀과 RFPA 실행 후의 혼합셀을 비교하였다. 표에서 보면 FPA의 경우 Ionosphere와 Wine는 7% 대로 획기적으로 그 수를 줄였고, 그 외의 데이터셋의 경우는 22% 정도로 감소하였다. RFPA의 경우는 5% 미만으로 FPA보다 안정적인 결과를 만들었다.

표 8. 대표패턴에 대한 혼합셀의 비율
(괄호안의 숫자:표준편차)

자료명	Breast-cancer	Glass	Ionosphere	Iris	New-thyroid	Wine
FPA	20.9 (1.51)	26.2 (1.57)	5.6 (1.57)	23.9 (4.41)	17.0 (6.03)	7.7 (2.88)
RFPA	0.43 (0.01)	1.0 (0.75)	4.31 (1.54)	0.68 (1.13)	0.69 (1.52)	3.60 (3.32)

4. 결 론

본 논문에서 제안한 RFPA 기법은 FPA 기법을 문제점을 해결하기 위하여 제안되었다. FPA 기법은 클래스가 혼재된 공간셀의 경우에 원본 학습 패턴을 그대로 저장하는 방법을 사용한다. 이는 메모리 사용 효율을 저하시키며, 분류 시간을 증가시키는 결과를 초래한다. RFPA 기법은 FPA 기법을 적용한 후, 혼재된 공간셀이 발견되면 혼재된 공간셀을 반복적으로 분할하는 방법을 사용하며, 전체 공간셀에 대한 혼재된 공간셀의 비율이 10%미만이 될 때까지 학습을 수행한다. 이것은 FPA 기법이 분할점을 선택할 때 패턴의 비율을 고려하지 않기 때문에 클래스가 혼재된 공간셀이 나타날 확률이 높다. 따라서 제안된 RFPA 기법은 패턴의 비율을 고려하여 분할점을 선정함으로써 보다 효율적인 분할이 이루어질 수 있도록 하였다.

본 논문에서 제안한 RFPA 기법은 k-NN, FPA 기법, EACH 시스템과 비교하여 유사하거나 향상된 분류 성능을 보여주며, 메모리 사용량에 있어서 k-NN, FPA 기법보다 줄어드는 것을 확인할 수 있었다.

향후 연구로는 분할된 초월평면의 반복적 고정 분할이 가능한 점진적 학습에 대한 방법을 현재 연구중에 있다.

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*, pp. 323-335, 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*, pp. 184-191, 1994.

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

[8] 정태선, 이형일, 윤충화, "고정 분할 평균알고리즘을 사용하는 새로운 메모리 기반 추론," 한국정보처리학회논문지, 제6권 제6호, pp. 1563-1570, 1999.

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

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

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

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

이 형 일

1985년 2월 명지대학교 전자계산학과 학사
1994년 2월 명지대학교 대학원 전자계산학과 석사
2000년 8월 명지대학교 대학원 컴퓨터공학과 박사
1984년 12월~1989년 11월 (주) 쌍용정보통신
1990년 5월~1994년 8월 (주)시에치노컨설팅
2005년 9월~김포대학 인터넷정보과 부교수
관심분야 : 미디어 영상인식, 패턴인식, 에이전트시스템, 정보검색, 기계학습