고정 분할 평균 알고리즘을 사용하는 향상된 메모리 기반 추론

정 태 선† · 이 형 일†† · 윤 충 화†††

요 약

본 논문에서는 메모리 기반 추론(MBR : Memory Based Reasoning) 기법에서 사용하는 기억공간과 분류시간의 향상을 위하여 고정 분할 평균(FPA : Fixed Partition Averaging) 알고리즘을 제안하였다. 제안된 방법은 전체 학습패턴들을 대표하는 패턴을 추출하여 효과적인 메모리 사용을 가능하게 하는 방법으로써, 패턴 공간을 일정 개수의 초월평면으로 분할한 후, 초월평면별로 소속된 패턴들의 평균값을 계산하여 대표패턴을 추출한다. 또한 분류성능의 향상을 위하여, 특징과 클래스간의 상호정보(Mutual Information)를 특징의 가중치로 사용하였다.

An Improved Memory Based Reasoning using the Fixed Partition Averaging Algorithm

Tae-Sun Cheong† · Hyeong-Il Lee†† · Chung-Hwa Yoon†††

ABSTRACT

In this paper, we proposed the FPA(Fixed Partition Averaging) algorithm in order to improve the storage requirement and classification time of Memory Based Reasoning method. The proposed method enables us to use the storage more efficiently by extracting representatives out of training patterns. After partitioning the pattern space into a fixed number of equally sized hyperrectangles, it averages 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.

1. 서 론

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

메모리기반 학습 알고리즘에 기반을 둔 분류기로는 k-NN(k-Nearest Neighbors) 분류기를 들 수 있으며 k-NN 분류기는 메모리에 저장된 학습패턴 중 주어진 입력패턴과 가장 가까운 거리에 있는 k개의 학습패턴을 선택하여 그중 가장 많은 패턴이 소속된 클래스로 입력패턴을 분류하는 방법을 사용한다[1,2,3,4,7,9,10]. 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

※ 본 연구는 1996년도 한국학술진흥재단 대학부설연구소과제 연구비에 의하여 연구되었음.
† 준 회 원 : 명지대학교 대학원 컴퓨터공학과
†† 준 회 원 : 김포전문대학 컴퓨터계열 교수
††† 정 회 원 : 명지대학교 컴퓨터학부 교수
논문접수 : 1998년 12월 18일, 심사완료 : 1999년 3월 13일

전체를 메모리에 저장하여야 하므로 다른 기계학습 방법에 비하여 많은 메모리 공간을 필요로 하며, 저장되는 학습 패턴이 증가할수록 분류에 필요한 시간도 많이 소요된다는 단점을 갖는다[5,6]. 따라서 메모리 기반 학습기법이 갖고 있는 문제점을 해결하기 연구가 지금까지 활발히 진행되어 오고 있으며, 대표적인 연구로 IBL(Instance Based Learning)과 NGE(Nested Generalized Exemplar)이론을 들 수 있다[1,3,7,8].

2. 관련 연구

2.1 k-NN 기법

k-NN 분류기는 메모리기반 학습기법을 사용한 최초의 분류기로 이 방법은 Lazy Learning Algorithm이라고도 하는데, 그 이유는 학습 시에는 단순히 학습패턴을 메모리에 저장하며, 차후 입력패턴을 분류할 때 모든 계산이 수행되기 때문이다[4].

이러한 k-NN 분류기의 개략적인 알고리즘은 다음과 같다.

① 주어진 학습패턴을 모두 메모리에 저장한다.

② 입력패턴 $\boldsymbol{Q}$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 입력패턴이다. 또한 n은 패턴을 구성하는 특징의 개수이며, E_i, Q_i는 각각 학습패턴과 입력패턴의 i번째 특징 값을 나타낸다.

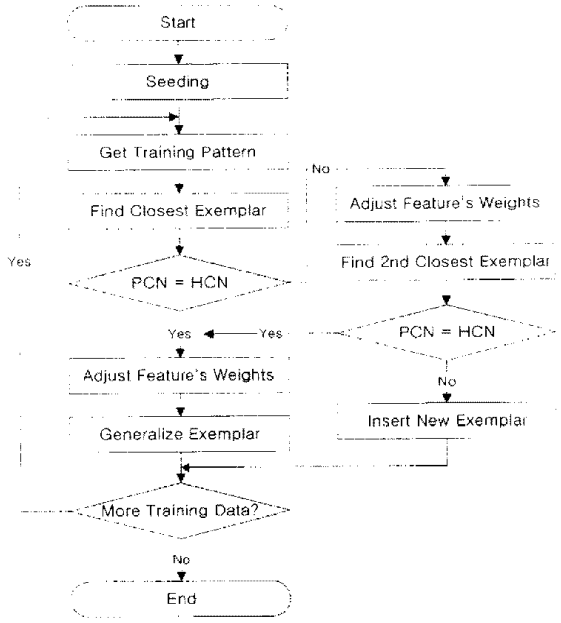
③ 입력패턴 $\boldsymbol{Q}$와 가장 가까운 k개의 학습패턴을 선정한다.

④ 선택된 k개의 학습패턴 중 가장 많은 개수의 패턴이 소속되는 클래스로 입력패턴 $\boldsymbol{Q}$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 Validation기법을 사용하여 결정하며, k=1인 경우를 NN 분류기라 한다[2,9,10]. 또한 위의 과정중 4번째 단계에서, 입력패턴과의 거리를 이용하여 가중치를 부여하는 방법을 WeightVote k-NN이라고 한다[2,10].

2.2 NGE 이론

Steven Salzberg는 1990년에 NGE(Nested Generalized Exemplar) 이론을 발표하고, 이를 이용하여 EACH 시스템이라는 분류기를 구현하였으며, 이 시스템에서는 주어진 학습패턴을 메모리공간에 초월평면(Hyperrectangle)의 형태로 저장한다[3,8]. EACH 시스템에서는, 모든 학습패턴을 그대로 저장하는 것이 아니고, 학습패턴을 일정 크기로 그룹화 하여 하나의 인스턴스로 표현을 하므로 k-NN과 같은 분류기에 비하여 상대적으로 높은 메모리 효율을 보장한다[6,13,14]. EACH시스템에서의 학습은 다음의 (그림 1)과 같이 이루어진다.



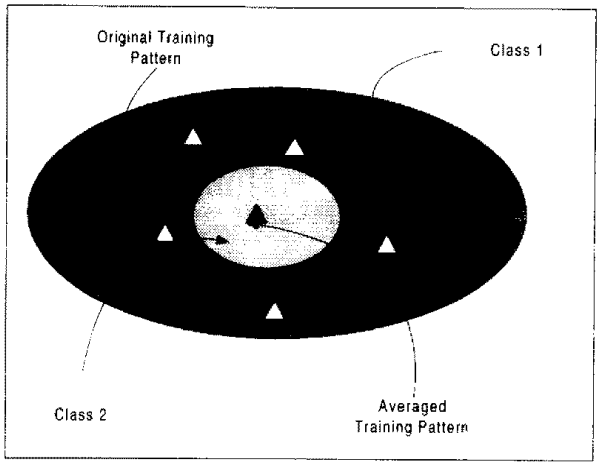
HCN: Hyperrectangle's Class Number
PCN: Pattern's Class Number

(그림 1) EACH 학습 알고리즘

3. FPA 학습 기법

본 논문에서는 메모리기반 학습 기법에서, 보다 효율적인 메모리사용과 빠른 분류 속도를 보장하기 위하여 FPA(Fixed Partition Averaging) 기법을 제안하였다. FPA 기법은 주어진 패턴공간을 일정 크기의 영역으로 분할 한 후 패턴 평균기법을 적용하는 방법이다. 이 기법은, 메모리 기반 학습기법에서 메모리 사용효율의 증대를 위한 방법으로 사용되는 인스턴스 평균(Instance Averaging) 법을 적용한 것으로, 인스턴스 평균법은 여러 개의 학습패턴의 특징 값들을 평균하여

하나의 학습패턴으로 대치하는 방법을 말한다[5]. 하지만 단순히 인스턴스 평균법을 적용하는 경우, 클래스가 다음 (그림 2)와 같이 환형을 이루고 있을 경우 문제가 발생하게 된다. (그림 2)에 나타난 것처럼 단순히 패턴 평균법을 적용할 경우 클래스 1에 소속된 5개의 패턴의 평균으로 구한 패턴이 원래의 클래스와는 다른 클래스 2에 소속되며, 이 경우 분류기가 오인식을 하게 된다.



(그림 2) 인스턴스 평균법의 문제점

3.1 특징 정규화

메모리 기반 분류기에서 출력 클래스의 결정은 입력 패턴과 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 FPA에서는 다음의 식 (2)를 이용하여 특징 값을 정규화한다. 이 기법은 식 (2)에 의하여 패턴을 구성하는 모든 특징 값을 0과 1사이의 값으로 정규화 함으로써, 모든 특징의 변화가 패턴의 소속클래스 결정에 미치는 영향력을 동일하게 한다.

$$f_{i_n} = \frac{f_i - f_{i_{min}}}{f_{i_{max}} - f_{i_{min}}} \qquad (2)$$

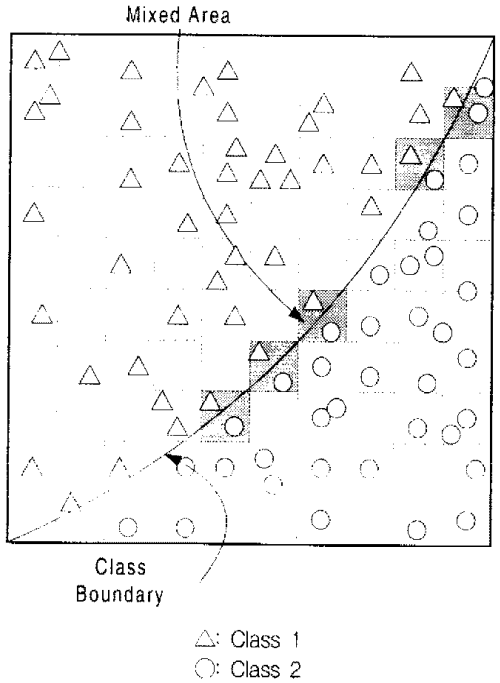
이때 f_i는 i번째 특징 값, $f_{i_{max}}, f_{i_{min}}$는 f_i가 가질 수 있는 최대, 최소 값을 나타낸다.

3.2 고정 분할 평균

본 논문에서는 패턴평균법의 문제점을 해결하기 위하여 특징공간을 일정한 크기를 가지는 N개의 초월평면으로 분할하고, 각 초월평면별로 패턴 평균법을 적용하는 방법을 제안하였다.

이 방법에서는 먼저 패턴 공간의 각 축을 일정한 크기로 분할한다. 예를 들어, 2차원 패턴공간의 경우 이 방법에 의해 분할하면 (그림 3)과 같이 패턴 공간이 격자모양으로 분할된다. (그림 3)은 패턴공간을 구성하는 2개의 축을 각각 10개의 영역으로 분할한 경우이다.

고정 분할평균 기법에서는 각 축을 같은 크기의 N개로 분할한 후, 분할된 초월평면 단위로 패턴 평균법을 적용하게 된다. 이때 (그림 3)에서 회색으로 표시된 클래스 혼합부분 (Mixed Area)의 경우에는 패턴 평균법을 적용하지 않고 원래의 패턴을 그대로 저장하는 방법을 사용하여 (그림 2)에서 기술한 문제점을 해결하였으며, 이때 각 축은 식 (3)을 만족하는 N개의 영역으로 분할된다. 이때 n은 하나의 패턴을 구성하는 특징 개수, $|T|$는 전체 학습패턴 개수이다.



(그림 3) 고정 분할 평균법

$$N = \lceil \log_n(0.3 \times |T|) \rceil \qquad (3)$$

따라서 식 (3)에 의해 계산된 값은 전체 학습패턴

개수의 30%보다 큰 정수 중 가장 작은 값을 선택하는 것이 된다. 여기에서 전체 학습패턴의 30%에 근사한 초월평면을 형성하도록 선택한 이유는, NN 분류기에 있어 실험적으로 전체 패턴의 약 30%만이 실제 분류에 사용되었다는 사실을 기준으로 한 것이다[13].

또한 패턴이 소속되는 셀의 위치를 판별하기 위하여 주소를 사용하는데, FPA의 주소는 n의 길이를 가지는 N진법의 숫자로 표현된다. 이때 n과 N은 구성하는 특징 수와 축의 분할 개수이다. 셀 주소는 오른쪽에서 왼쪽으로 해석되며 각 숫자는 분할 공간의 위치를 나타내며, 0에서부터 시작한다. 예를 들어 3개의 특징을 가지는 패턴공간에서 각축이 10개로 분할된 경우 셀 주소 123은 첫 번째 특징이 4번째 분할영역에, 두 번째 특징이 3번째 분할영역에, 그리고 마지막 특징이 2번째 분할영역에 속하는 것을 의미한다

3.3 상호정보를 이용한 특징 가중치의 적용

본 논문에서 제안한 FPA 기법에서는 분류기의 성능 향상을 위하여 특징과 클래스간의 상호정보 이득(Mutual Information Gain)을 사용하며, 이 방법은 결정 트리 분류기에서 각 노드의 특징과 해당 임계치를 선정하기 위하여 주로 사용되는 기법이다[12,15].

3.3.1 상호 정보

분류기를 정보이론의 측면에서 보면 "패턴 p는 클래스 i에 소속된다"라는 메시지를 표시하는 메시지의 근원이라고 볼 수 있으며, 이러한 메시지를 생성하기 위한 전체 필요 정보의 양은 식 (4)에 의해 계산 할 수 있다[12,15].

$$I = -\sum_{i=1}^{C} p_i \log_2 p_i \qquad (4)$$

이때 p_i는 전체 패턴 중 클래스 i에 소속되는 패턴의 비율, 즉 임의의 패턴 p가 클래스 i로 분류될 사전확률을 의미하며, C는 학습패턴을 구성하는 클래스의 개수이다.

3.3.2 평균 상호정보 이득

이 값은 FPA 기법에서 특징의 가중치로 사용되며, FPA 에서는 실제 메모리에 저장된 패턴들 모두가 분류에 사용되는 것이 아니고, 고정 크기로 분할된 초월평면 별로 패턴평균법을 적용하므로, 특징공간의 분할 이전과 이후에 필요한 정보의 양 사이에 차이가 발생하게 된다. 이때 분할 이전과 이후에 발생하는 정보량의 차이를 상호정보 이득 (Mutual Information Gain) 이라 표현하며, 이 값은 각 특징이 클래스의 결정에 미치는 영향력으로 해석 할 수 있다.

FPA에서 사용하는 상호정보 이득은 다음의 식 (5)로 나타내어진다.

$$IG(f) = I - \sum_{i=1}^{N} P_i I_c \qquad (5)$$

이때 I는 식 (4)에서 정의한 분할 이전에 필요한 정보의 량이며, N은 특징 축 f의 분할 개수이며 이 값은 식 (3)에 의해 계산된다. Ic는 특징 f를 기준으로 분류했을 때 분할된 공간에서 필요한 정보의 량이며, 이 값은 식 (4)와 같은 방법을 사용하여 계산한다. 또한 P_i는 전체 학습패턴 중 분할된 초월평면에 할당된 패턴의 비율이다.

식 (5)에서 얻어진 값은 특징공간의 분할 이전에 필요한 정보의 양과 분할된 각 초월평면에서 필요한 평균 정보양의 합간의 차이가 된다. 이 값은 패턴공간을 하나의 특징을 기준으로 분할함으로써 얻어지는 정보의 이득이 되며, 어떤 특징을 기준으로 패턴공간을 분할해 나가는 것이 가장 효율적인 가를 판단하는 기준으로 사용 할 수 있다. 따라서 본 논문에서 제안한 FPA 기법에서는 식 (5)로 주어진 $IG(f)$ 값을 입력패턴과 메모리에 저장된 학습패턴간의 거리계산에 사용하며, 이때의 거리는 식 (6)에 의해 계산한다.

$$D_{EQ} = \sqrt{\sum_{i=0}^{n} IG(i)(E_{f_i} - Q_{f_i})^2} \qquad (6)$$

3.4 FPA 기법의 패턴 분류

본 논문에서 제안한 FPA 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-Validation 법의 특성상 많은 계산시간을 요하게되며, 이에는 Leave-One-Out법과 N-Folding법이 있다. Leave-One-Out법은 전체 패턴 중 1개를 제외한 모든 패턴을 학습패턴으로 사용하고, 제외된 1개의 패턴을 테스트 패턴으로 하여 분류를 시도하는 방법으로 모든 패턴이 각각 한 번씩 테스트 패턴으로 사용될 때까지 분류를 계속하는

방법이다. 비슷한 방법으로 N Folding 기법은 전체패턴을 N개의 그룹으로 분할하고 각 그룹을 한번씩 놓아가면서 테스트패턴으로 사용하는 방법이다[1, 5].

본 논문에서 제안한 방법에서는 k값을 학습 시에 결정하지 않고 패턴의 분류 시에 결정하게 되며, k값은 가변적으로 결정된다. FPA에서는 패턴의 분류시 가장 인접한 패턴과 그 다음으로 가까운 패턴의 클래스가 같을 경우 k=1인 NN분류기와 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 입력패턴과 가까운 패턴에 따라 변하게 된다. 그 후 입력패턴의 분류는 k-NN 분류기와 동일하게 가장 많은 패턴들이 소속된 클래스로 분류한다.

4. 실험 및 분석

FPA 기법을 이용한 분류기의 성능을 k-NN, EACH 기법과 비교하여 검증하였다. 실험은 기계학습의 벤치마크 자료로 사용되는 7개의 데이터를 이용하였으며, 실험 방법은 70:30 법(전체 데이터를 기준으로 70%는 학습패턴으로, 30%는 평가패턴으로 사용하는 방법)을 사용하였다[11]. 이때 70%의 학습패턴은 전체 패턴의 클래스별 분포를 고려하여 모든 클래스에서 같은 비율로 추출하였다. 실험은 Windows NT를 적재한 PentiumII-233 컴퓨터를 사용하였으며, 모든 실험결과는 25회 반복측정 한 후 평균값으로 나타내었다.

4.1 실험 데이터

본 논문에서는 기계 학습의 벤치마크 자료로 사용되는 7개의 데이터를 UCI Machine Learning Database Repository에서 발췌하여 사용하였으며, 이들 7개의 데이터는 Breast Cancer Wisconsin, Glass, Ionosphere, Iris, New Thyroid, Sonar, Wine이며, 이들 데이터는 모든 특징이 실수 값을 가진다[5].

〈표 1〉 실험 데이터의 특성

데이터	전체 패턴 개수	특징 개수	클래스 개수
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

다음의 〈표 2〉는 7개의 데이터를 70:30법을 이용하여 분할하였을 경우, 클래스별 학습패턴의 분포를 보여주고 있다.

〈표 2〉 클래스별 학습패턴의 분포

데이터	전체 학습패턴 개수	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	×	×	×

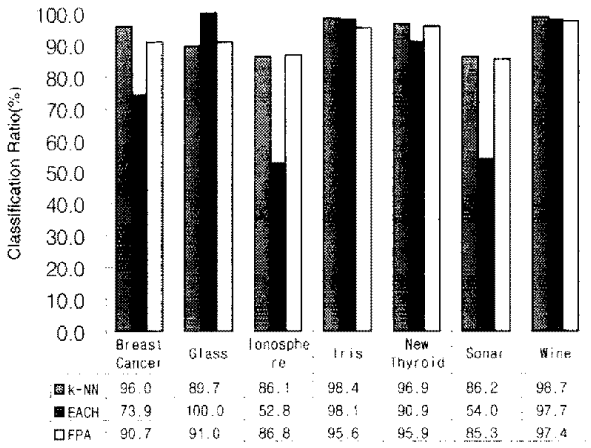
4.2 분류 성능 실험

(그림 4)의 k-NN, EACH, FPA의 분류성능을 보면, 본 논문에서 제안한 FPA 기법이 Glass, Ionosphere 두 개의 데이터에서는 k-NN에 비하여 우수한 분류성능을 보이고 있는 반면, Breast Cancer 및 Iris 두 개의 데이터에서는 다소 저조한 분류 성능을 보이고 있으며, 나머지 3개의 데이터에서는 두 분류기가 비슷한 성능을 나타내고 있음을 볼 수 있다. EACH 시스템의 경우, Glass 데이터에서 다른 기법에 비하여 우수한 분류 성능을 보이기는하지만, Breast-Cancer, Ionosphere, Sonar 3개의 데이터에서는 아주 저조한 분류성능을 보이고 있다. 이처럼 EACH 시스템이 저조한 분류성능을 보이는 것은, EACH 시스템에서 사용하고 있는 초기 시드 개수의 영향에 의한 것으로 볼 수 있다[14].

본 논문에서 제안한 FPA 기법과 Salzberg의 EACH 시스템은 학습패턴으로 주어진 패턴 중 일부만을 저장하여 입력 패턴의 분류에 사용한다. 위의 결과에서 보면 EACH 시스템의 경우 데이터에 따라서 분류기의 성능변화가 심한 것에 반하여, FPA기법은 안정된 분류성능을 보이고 있다.

위 실험에서 k-NN 분류기의 성능은 Leave-One-Out Cross Validation기법을 사용하여 계산한 k값을

사용한 것이며, EACH 시스템의 분류성능은 초기 시드 개수 5, 가중치 증가량 0.2를 사용하여 측정한 결과이다.



(그림 4) k-NN, EACH, FPA 분류성능 비교

다음의 <표 3>은 각 데이터에서 사용된 k-NN 분류기의 k값을 보여주고 있다.

〈표 3〉 분류성능 최적화를 위한 k값

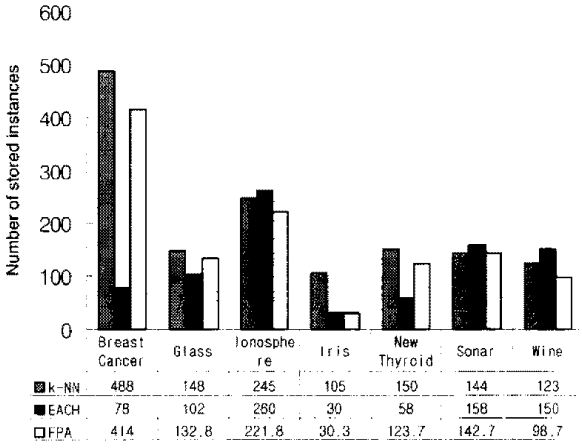
데이터	Breast Cancer	Glass	Ionosphere	Iris	New Thyroid	Sonar	Wine
k값	21	1	1	51	1	1	19

4.3 메모리 사용량 비교 실험

(그림 5)의 실험결과에서는 k-NN, EACH, FPA 세 가지 방법을 이용한 분류기의, 메모리 사용량을 보여주고 있으며, (그림 5)에 나타난 수치는 메모리에 저장된 학습 패턴의 개수를 의미한다. 이때 EACH 시스템의 경우, 메모리에 저장된 초월평면의 수×2를 저장된 학습패턴의 수로 사용하였는데, 이는 EACH 시스템에서 메모리에 저장되는 초월평면이 평면의 범위를 나타내는 상, 하한의 두 개의 패턴으로 표시되기 때문이다.

결과에서 보면 k-NN의 경우 모든 학습패턴을 메모리에 저장하고 분류시 입력패턴을 모든 학습패턴과 비교한다. 하지만 FPA 기법의 경우, 주어진 패턴공간을 초월평면으로 분할하여 각 초월평면을 대표하는 패턴을 저장하는 방법을 사용함으로써 우수한 메모리 사용효율을 보장하고 있다.

본 논문에서 제안한 FPA 기법에서는 Iris 데이터의 경우 약 30%정도의 메모리만을 사용하고 있는 것을 볼 수 있으며, 나머지 6개의 데이터에서도 k-NN 대비 약 60-80%정도의 학습패턴만을 메모리에 저장하는 것을 볼 수 있으며, EACH 시스템과의 비교에 있어서도 3개의 데이터 셋에 있어 우수한 메모리 사용 효율을 보이고 있다. 여기에서 주목할 만한 것은, EACH 시스템의 경우, Ionosphere, Sonar, Wine 3개의 데이터에서 전체 학습패턴으로 주어진 패턴 수 보다 많은 패턴을 메모리에 저장하는 결과를 보이는데 이것은 앞에서 언급한 바와 같이 EACH 시스템에서는 메모리에 저장되는 패턴을 초월평면의 형태로 표현하며, 이때의 초월평면은 상, 하한을 나타내는 2개의 패턴으로 구성이 되기 때문이다.



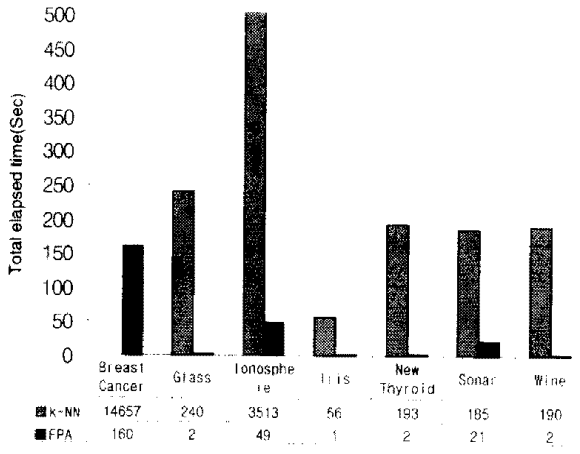
(그림 5) k-NN, EACH, FPA 메모리 사용량 비교

실험 4.2와 4.3에서 보는 것처럼 본 논문에서 제안한 FPA 기법이 분류성능 대비 메모리 사용효율의 측면에서 기존의 k-NN 분류기 및 EACH 시스템에 비하여 우수한 성능을 보이고 있는 것을 볼 수 있다.

4.4 분류 소요시간 비교

여기에서는 k-NN과 FPA를 사용한 분류기의 실제 데이터 분류에 소요되는 시간을 측정하였다. 실제 데이터 분류에 소요되는 시간의 측정은 본 논문에서 제안한 FPA 기법을 사용하는 분류기의 경우 k값을 입력패턴의 분류 시에 결정되게 되므로, Cross-Validation 기법을 사용하는 k-NN 분류기에 비하여 월등히 빠른 속도의 학습과 분류를 보장하며, 메모리 상에 저장되는 학습패턴의 수를 줄임으로써 실제 데이터 분류에 소요되는 시간은 (그림 6)에서 보는 것처럼, 본 논문에

서 제안한 FPA 기법이 k-NN 분류기가 필요로 하는 시간의 약 0.5%-1.5%만을 사용한다.



(그림 6) k-NN, FPA 분류시간 비교

위의 실험 결과처럼 k-NN 분류기와 본 논문에서 제안한 FPA 기법간에 월등한 분류시간 의 차이가 발생하는 이유는, k-NN 분류기의 경우 실험 4.2에서처럼 분류기 성능의 최적화를 위하여 Cross Validation기법을 k값을 결정하는데 사용하는 것에 반하여, FPA 에서는 k값이 3.4절에서 설명 한대로 입력 패턴의 분류시 가변적으로 결정되기 때문이다.

5. 결론 및 향후 연구과제

본 논문에서는, 메모리 기반 추론에 있어 효율적인 메모리 사용과 분류 속도의 향상 기법을 제안하였다. 본 논문에서 제안한 FPA 기법은 패턴 평균법을 사용하여 메모리에 저장되는 학습 패턴들을 대표 패턴으로 대체하는 방법을 채택하였으며, 실험 결과에서 볼 수 있는 것처럼 분류 성능 및 분류 시간에 있어 기존의 k-NN 기법에 비하여 우수한 성능을 보였다.

하지만 FPA기법의 경우, 전체 패턴공간을 일정한 개수의 초월평면으로 분할하여 패턴 평균법을 적용하고 있는데, 최악의 경우에는 분할된 패턴공간의 개수만큼 패턴이 저장되게 된다. 따라서 데이터의 특성에 따라 특정 축을 가변적으로 분할할 경우, 좀더 효율적인 분류와 메모리 사용이 기대되며, 본 논문에서 제안한 FPA 기법에서 사용하는 축의 분할 개수는 실험적으로 결정된 값으로, 이 값을 최적화할 경우 분류기의 성능 향상이 기대된다.

기존의 메모리 기반 추론 기법은 규칙의 추출이 불가능하였으며, 규칙의 추출이 필요한 경우, 결정 트리 기법이나 유전자 알고리즘을 사용하여 왔다. 그러나 FPA 기법에서는 분할된 초월평면을 이용한 규칙의 추출이 가능하며, 또한 학습 패턴이 점차로 증가하는 경우, 초월평면의 통계자료만을 이용한 점진적 학습(Incremental Learning)도 가능하다고 사료된다.

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 k-NN versus Majority k-NN 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. Wettschereck, et al., A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms, Artificial Intelligence Review Journal, 1996.

[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, Vol.6, No.3, pp.251-276, 1991.

[9] 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.

[10] 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.

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

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

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

[14] 이형일, 정태선, 윤충화, EACH시스템에서의 새로운 가중치 적용법, 한국 정보과학회 '98 추계학술대회, 1998.

[15] 김상귀, 이형일, 윤충화, A study on the optimization of binary decision tree, 명지대학교 산업기술연구소 논문지, Vol.16, pp.104-112, 1997. 2.

정 태 선

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

1995년 명지대학교 컴퓨터공학과(학사)

1998년 명지대학교 대학원 컴퓨터공학과(석사)

1998년~현재 명지대학교 대학원 컴퓨터공학과 박사과정 재학 중

관심분야 : 신경회로망, 기계학습, 지능형 소프트웨어 에이전트

이 형 일

e-mail : hilee@kimpo.ac.kr

1985년 명지대학교 전자계산학과(학사)

1985년~1989년 (주)쌍용컴퓨터 근무

1990년~1995년 CHNO System Consulting Co. 근무

1994년 명지대학교 대학원 전자계산학과(석사)

1997년 명지대학교 대학원 컴퓨터공학과 박사과정 수료

1997년~현재 김포전문대 컴퓨터계열 전임강사

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 공대 컴퓨터학부 부교수

관심분야 : 신경회로망, 전문가시스템, 지능형 소프트웨어 에이전트, 기계학습