적응형 재귀 분할 평균법을 이용한 메모리기반 추론 알고리즘

(A Memory-based Reasoning Algorithm using Adaptive Recursive Partition Averaging Method)

이 형 일 † 최 학 윤 ††
(Hyeongil Lee) (Hakyoon Choi)

요 약 메모리 기반 추론에서 기억공간의 효율적인 사용과 분류성능의 향상을 위하여 제안되었던 RPA(Recursive Partition Averaging)알고리즘은 대상 패턴 공간을 분할 한 후 대표 패턴을 추출하여 분류 기준 패턴으로 사용한다. 이 기법은 메모리 사용 효율과 분류 성능 면에서 우수한 결과를 보였지만, 분할 종료 조건과 대표패턴의 추출 방법이 분류 성능 저하의 원인이 되는 단점을 가지고 있었다.

여기에서는 기존 RPA의 단점을 보안한 ARPA(Adaptive RPA) 알고리즘을 제안한다. 제안된 알고리즘은 패턴 공간의 분할 종료 조건으로 특징별 최빈 패턴 구간(FPD: Feature-based population densimeter) 추출 알고리즘을 사용하며, 학습 결과 패턴의 생성을 대표패턴 추출기법 대신 최빈 패턴 구간을 이용하여 생성한 최적초월평면(OH: Optimized Hyperrectangle)을 사용한다.

제안된 알고리즘은 k-NN분류기에서 필요로 하는 메모리 공간의 40%정도를 사용하며, 분류에 있어서도 RPA보다 우수한 인식 성능을 보이고 있다. 또한 저장된 패턴의 감소로 인하여, 실제 분류에 소요되는 시간 비교에 있어서도 k-NN보다 월등히 우수한 성능을 보이고 있다.

키워드 : 거리기반학습, 기계학습, 인공지능

Abstract We had proposed the RPA(Recursive Partition Averaging) method in order to improve the storage requirement and classification rate of the Memory Based Reasoning. That algorithm worked not bad in many area, however, the major drawbacks of RPA are it's partitioning condition and the way of extracting major patterns.

We propose an adaptive RPA algorithm which uses the FPD(feature-based population densimeter) to stop the ARPA partitioning process and produce, instead of RPA's averaged major pattern, optimizing resulting hyperrectangles.

The proposed algorithm required only approximately 40% of memory space that is needed in k-NN classifier, and showed a superior classification performance to the RPA. Also, by reducing the number of stored patterns, it showed an excellent results in terms of classification when we compare it to the k-NN.

Key words : Distance Based Learning, Machine Learning, Artificial Intelligence

1. 서 론

메모리 기반 추론 학습은 주어진 학습패턴 그 자체를 모두 메모리에 저장하는 것일 뿐이며, 입력패턴의 분류는 저장된 패턴들과 입력패턴 사이의 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 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 하므

† 정 회 원 : 김포대학 컴퓨터계열 교수
hilee@kimpo.ac.kr
†† 비 회 원 : 김포대학 전자정보계열 교수
hychoi@kimpo.ac.kr
논문접수 : 2003년 1월 29일
심사완료 : 2003년 12월 30일

로 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 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], RPA(Recursive Partition Averaging)[16]를 들 수 있다.

본 논문에서는 패턴 공간의 분할 종료 조건으로 특징별 최빈 패턴 구간(FPD: Feature-based population densimeter) 추출 알고리즘을 사용하며, 학습 결과 패턴의 생성을 대표패턴 추출기법 대신 최빈 패턴 구간을 이용하여 생성한 최적초월평면(OH: Optimized Hyper-rectangle)을 사용하여 효율적인 메모리 사용과 분류성능을 보장하는 알고리즘을 제안하고, UCI Repository의 벤치마크 데이타를 이용하여 성능을 실험적으로 검증하였다.

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 학습 패턴을 메모리에 저장하며, 차후 입력패턴을 분류할 때 모든 계산이 수행되기 때문이다[10].

이러한 k-NN 분류기의 개략적인 알고리즘은 다음과 같다.

① 주어진 학습패턴을 모두 메모리에 저장한다.

② 입력패턴 Q의 분류를 위하여 메모리에 저장된 모든 학습패턴과의 거리를 식 (1)을 이용하여 계산한다.

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

이때 E는 메모리에 저장된 학습패턴을 나타내며, Q는 주어진 입력패턴이다. 또한 n은 패턴을 구성하는 특징의 개수이며, E_i, Q_i는 각각 학습패턴과 입력패턴의 i 번째 특징 값을 나타낸다.

③ 입력패턴 Q와 가장 가까운 k개의 학습패턴을 선정한다.

④ 선택된 k개의 학습패턴 중 가장 많은 개수의 패턴이 소속되는 클래스로 입력패턴 Q를 분류한다.

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

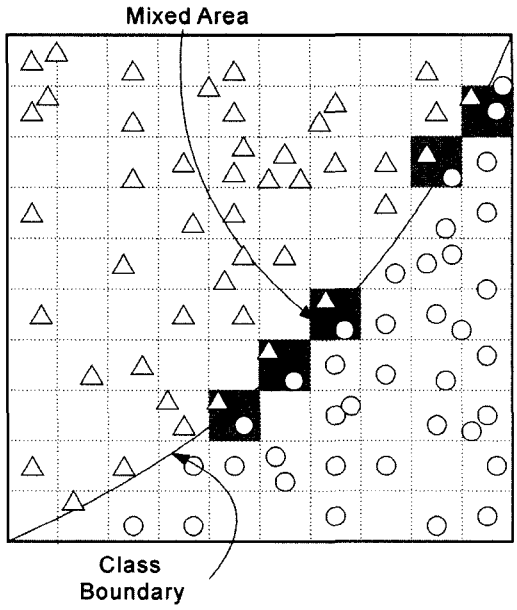
2.2 고정 분할 평균 기법

고정 분할 평균(FPA) 기법은 주어진 패턴공간을 동일한 크기의 초월평면들로 분할한 후 패턴 평균기법을 적용하는 방법이다[9]. 이 방법에서는 먼저 패턴 공간의 각 특징 축을 일정한 크기로 분할한다. 이 때 특징축의 분할 개수는 식 (2)에 의해 결정된다.

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

이때 n은 하나의 패턴을 구성하는 특징 개수, |T|는 전체 학습패턴의 개수이다.

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 클래스 혼합 부분의 경우에는 패턴 평균법을 적용하지 않고 원래의 패턴들을 그대로 저장하며, 클래스가 혼합되지 않은 부분은, 해당 셀 내의 모든 패턴을 평균하여 하나의 대표패턴으로 대체하는 방법을 사용한다.



△: Class 1 ○: Class 2
그림 1 고정 분할 평균법

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_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[12].

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

이 때 I는 식 (3)에서 정의한 특징축 분할 이전에 필요한 정보의 양, N은 특징 축 f의 분할 개수이며 이 값은 식 (2)에 의해 사전에 계산된다. I_c는 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 거리 계산에 사용하는 k-NN 기법의 분류성능에 근접하면서, 패턴 평균 기법을 사용하여 k-NN 기법에서 나타나는 메모리 공간의 낭비를 줄이는 것이다. 그러나 FPA 기법에서와 같이 고정 개수의 분할 구간을 사용할 경우 분할 구간의 크기가 분류성능에 영향을 미치게 된다.

2.3 재귀 분할 평균 기법

재귀 분할 평균(RPA: Recursive Partition Averaging) 기법은 주어진 패턴공간을 재귀적으로 분할해 나가면서 대표패턴을 추출하는 방법이다[16]. 이 방법에서는 메모리기반 학습 기법에서 보다 효율적인 메모리 사용과 분류 성능을 보장하기 위하여 인스턴스 평균 (Instance Averaging) 법을 적용하였으며, 인스턴스 평균법은 여러 개의 학습 패턴의 특징 값을 평균하여 하나의 대표패턴으로 대치하는 방법을 말한다[13,14].

RPA는 주어진 패턴공간의 각 특징 축을 최초 2개의 영역으로 분할한다. 따라서 첫 번째 분할에서는 패턴공간이 2^n개의 공간으로 분할되며, 이때 n은 패턴을 구성하는 특징의 개수 즉, 패턴공간의 차원수가 된다. 따라서 2차원 패턴의 경우 최초 4개의 패턴공간으로 분할되며, 패턴의 분할은 현재 분할된 셀 각각에 대하여 재귀분할 여부를 결정한다. 즉 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. 클래스가 혼합된 부분에 대해서는 점점 세밀하게 분할해 나가게 되므로, 클래스 경계면에 위치한 셀의 경우 많은 분할이 이루어지게 된다.

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.

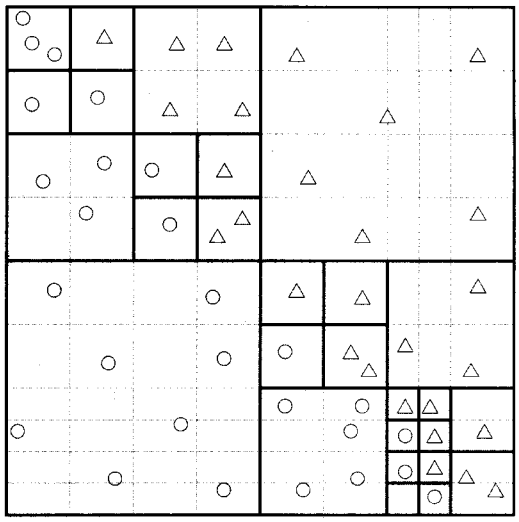


그림 2 재귀분할 평균기법에 의해 분할된 패턴 공간

RPA 기법은 FPA 기법에서 분류성능 저해요인으로 대두된 고정 개수의 분할 구간으로 분할 구간 크기 문제를 해결하기 위해 동적으로 분할 공간의 개수를 결정하는 방법을 사용을 시도하였다. 하지만 RPA 분할 알고리즘에서는 특징별 패턴 분포를 고려하지 않고 단순히 클래스의 순도(purity)만을 만족시키기 위한 분할을 하고 있으며, 이것은 생성된 대표 패턴의 공간 내 위치 형성에 좋지 않은 영향을 미치게 된다.

본 논문에서는 특징 최빈 구간 추출 알고리즘을 이용하여 특징별 패턴 분포를 고려한 분할을 시도하며, 그 결과로 생성된 초월 평면들에 대해서도 같은 알고리즘을 이용하여 공간 내 위치 보정 작업을 병행하는 ARPA(Adaptive RPA) 알고리즘을 제안하였다.

3. ARPA 학습 기법

메모리기반 학습 기법에서 보다 효율적인 메모리 사용과 분류 성능을 보장하기 위하여 적응적 재귀 분할 평균(ARPA: Adaptive Recursive Partition Averaging) 기법을 제안하였다. ARPA 기법은 주어진 패턴공간을 패턴의 빈도를 고려하여 재귀적으로 분할해 나가면서 결과 초월 평면을 생성하는 방법이다. 이 방법에서는 메모리 사용 효율을 증대하기 위하여 학습된 패턴을 NGE 이론에 기반한 EACH시스템에서 제안한 초월 평

면 형태로 저장하는 방법을 사용한다[7]. 이 방법은 인스턴스 평균(Instance Averaging) 기법과는 달리 패턴 분류 시 기준이 되는 패턴을 초월 평면 형태로 메모리 상에 저장하고 있는 방법이다.

3.1 특징의 정규화

메모리 기반 분류기에서 출력 클래스의 결정은 입력 패턴과 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 구성되어 있다. 따라서 두 번째 특징이 조금만 차이가 나더라도 나머지 특징간의 차이에 관련 없이 출력 클래스가 결정된다. 이러한 문제점의 해결을 위하여 MRPA 에서는 다음의 식 (5)를 이용하여 특징 값을 정규화 한다. 이 기법은 식 (5)에 의하여 패턴을 구성하는 모든 특징 값을 0과 1사이의 값으로 정규화 함으로써, 모든 특징의 변화가 패턴의 소속클래스 결정에 미치는 영향력을 동일하게 한다.

$$f_{i_n}=\frac{f_i-f_{i_{min}}}{f_{i_{max}}-f_{i_{min}}} \tag{5}$$

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

3.2 특징별 최빈 패턴 구간 추출

ARPA는 각 특징에 대한 정규화 작업이 완료되면, 공간상에 분포된 패턴들의 특징별 구간 추출 작업(FPD: Feature-Based Population Densimeter)을 실행한다. 이것은 공간에 분포된 패턴의 위상(topology)를 고려하여 각 특징별 유효 구간을 형성하는 과정이다.

FPD 작업은 연속값을 가지는 특징을 고려한 연관규칙의 추출시 사용되는 구간 분할 알고리즘과 유사하다[17]. FDA의 기본 아이디어는 연속(Continuous)된 구간을 이산(Discrete) 구간으로 분할하는 것으로 분할 구간의 크기는 각 구간에 소속된 패턴의 개수에 따라 가변 적이며, 공간상에 분포된 각 클래스 별도로 이루어진다. 다시 말하면, c개의 클래스로 구성된 패턴공간의 경우 각 특징에 대하여 c번의 FPD작업이 수행된다는 것이다.

본 논문에서 제안하는 FPD알고리즘은 연관규칙 추출에 사용된 최빈구간 알고리즘[17]의 이진 분할 뒤 병합과 달리 대신 최소 분할 후 병합(Divide First and Merge) 기법을 사용한다. 즉, 주어진 특징에 대해 일정 크기의 작은 구간으로 전체를 분할 한 후 최소밀도 θ를 만족하는 연속된 구간에 대해서 병합 작업을 수행하는 것이다.

$$N_i=\lceil \log_n(0.3\times|T|)\rceil\times n \tag{6}$$

$$\theta_j=AVE\left(\left|\frac{T_{i_j}}{N_i}\right|\right) \tag{7}$$

식 (6)에서 N_i는 i번째 구간에 존재하는 최소 패턴 개수이며, n은 하나의 패턴을 구성하는 특징 개수, 그리고 |T|는 전체 학습패턴의 개수이다. 또한 전체 학습패턴의 30%에 근사한 초월평면을 형성하도록 선택하였다. 그 이유는 NN 분류기에 있어 실험적으로 전체 패턴의 약 30%만이 실제 분류에 사용되었다는 사실을 기준으로 한 것이다[9]. 식 (7)은 클래스 j에 대한 최소 임계 패턴 밀도 θ_j는 학습 패턴(T) 중에서 구간 $i(T_i)$에 소속되면서 클래스가 $j(T_{i_j})$인 것들의 수를 i 구간에 포함된 최소 패턴 개수로(N_i) 나눈 것의 평균값이 된다.

FPD알고리즘은 식 (6)에 의해 구해진 값을 만족하도록 특징을 구간 분할하고, 각 구간을 검색하면서 식 (7)의 최소 패턴밀도 이상의 패턴이 존재하는가를 검사하는 것이다. 이 때 최소패턴수를 만족하는 연속된 구간에 대해서는 병합 작업이 이루어지며, 만족하지 못하는 공간은 병합대상에서 제외된다. 이러한 병합 작업은 더 이상 병합 공간이 발생하지 않을 때까지 이루어진다.

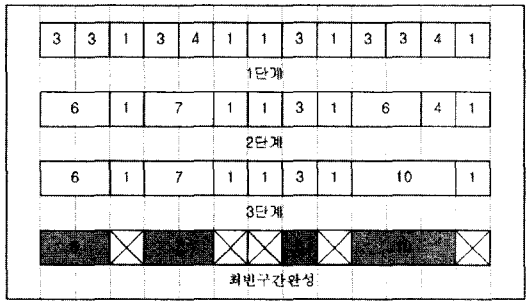


그림 3 FPD를 이용한 유효 최빈 구간 계산

```
For all classes {
  N_i=⌈log_n(0.3×|T|)⌉×n
  θ_j=AVE(|T_ij/N_i|)
  For all features {
    FPD(N_i, θ_j)
    Record valid range R_ij
  }
}
```

그림 4 FPD 알고리즘

그림 3은 임계값 θ_j=2, 구간수 N_{i_j}= 13일 때 그림 4의 FPD 알고리즘을 적용하는 과정이다. 이 때 셀은 구간을 의미하며 셀 내부의 수치는 각 셀에 소속된 패턴의 개수를 나타낸다.

3.3 패턴공간의 분할 주소 부여

ARPA에서 패턴 구간의 분할은 RPA와 유사한 방법을 사용한다. 주어진 패턴공간의 각 특징 축을 최초 2개의 영역으로 분할한다. 본 논문에서는 전체 특징에 대하여 한번씩 분할을 수행하는 것을 라운드(round)라고 표현하며, 1라운드 이후에는 전체 패턴공간이 2n개의 초월평면으로 분할된다. 이때 n은 패턴을 구성하는 특징의 개수 즉, 패턴공간의 차원수가 된다. 따라서 2차원 패턴의 경우 최초 4개의 패턴공간으로 분할된다.

ARPA에서는 현재 분할 하고자하는 셀의 위치를 알아내기 위하여 라벨과 주소를 사용한다. 이 때 각 셀의 주소는 2진수 형태의 값을 가지고 있으며 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 각 축의 위치를 0과 1로 표현이 가능하다. 또한 ARPA에서는 셀의 위치 판별을 위하여 현재 몇 번째 반복 분할을 시도하고 있는가에 대한 정보를 갖고 있으며, 이 값을 레벨이라 한다. 레벨은 주소와 함께 셀의 위치판단에 사용된다. 아래의 그림 5은 RPA에 의해 분할된 패턴공간의 레벨과 주소를 보여 주고 있다.

ARPA에서는 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 재귀분할의 여부와 재귀 분할의 대상이 되는 학습패턴을 결정하기 위해 수행된다.

ARPA의 마지막 단계로 현재 분할된 셀 각각에 대하여 재귀 분할 여부를 결정한다. ARPA에서는 하나의 셀에 소속되는 패턴의 클래스가 모두 같고 식 (7)에서 계산한 구간 임계값(θ_j)보다 적은 수의 패턴이 있을 경우에만 분할을 중지한다.

ARPA 분할이 완료되면 전체 패턴 공간은 H개의 초월 평면으로 구성되며, 이들 초월평면이 1차 학습 결과가 된다.

3.5 초월평면 최적화

ARPA의 다음 작업은 초월평면 최적화(OHC: Optimized Hyperrectangle Carving) 알고리즘의 적용이다. OHC는 3.4에서 얻어진 초월 평면에서 실제 분류에 영향을 미치지 못하는 영역을 제거하는 것이며, 다음 그림으로 설명할 수 있다.

그림 6에서 R_{00}, R_{10}는 그림 3의 FPD알고리즘에 의해 생성된 유효 최빈 구간을 나타낸다. 이 최빈구간은 특징의 범위중 유효한 값을 가지는 영역을 나타내는 것으로 ARPA분할에 의해 형성된 초월 평면의 최적화 작업에 사용된다. 다시말해 형성된 초월평면중 유효구간이 아닌 영역을 포함하는 초월 평면은 유효구간만을 포함하도록 축소된다. 그림 6에서 회색 구간은 초월 평면 중 유효구간에 포함되지 않아 삭제 또는 축소되는 부분을 나타낸다.

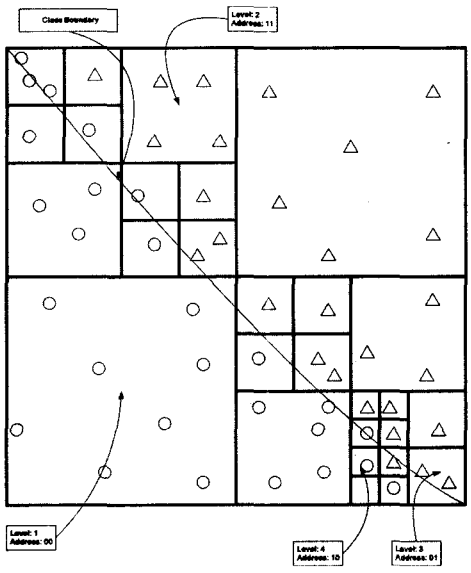


그림 5 패턴공간의 ARPA 분할과 주소부여

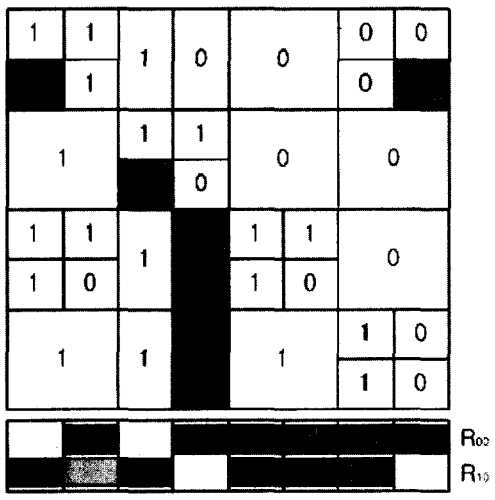


그림 6 OHC를 이용한 초월평면 최적화

3.6 ARPA 학습기법 패턴분류

ARPA의 학습 패턴 분류는 메모리 기반 알고리즘에서 사용하는 거리 기반 기법이며, 거리의 계산에는 분류 성능 향상을 위하여 식 (4)로 주어진 $IG(f)$ 값을 입력패턴과 메모리에 저장된 학습패턴간의 거리계산에 있어 특징의 가중치로 사용한다. 다만 식 (4)에서 고려 대상이 되는 분할 공간과 대상패턴은 전체이지만, ARPA에서는 $IG(f)$는 OCH알고리즘을 적용하여 형성된 최적 초월 평면에 포함되는 영역만을 계산에 사용한다. 이 때의 거리는 식 (8)에 의해 계산한다.

$$D_{EQ}=\sqrt{\sum_{i=0}^{n} IG(i)(E_i-Q_i)^2} \quad (8)$$

ARPA 기법을 이용한 분류기에서는 k-NN 분류기와는 다른 분류 기법을 사용한다. k-NN 분류기의 경우, 분류기의 성능을 최적화하기 위하여 k값을 사전에 결정하고 전체 시스템에서 하나의 고정된 k값을 사용하게 된다. 하지만 이 방법의 경우 k값의 결정을 위해서 주로 사용되는 Cross-Validation법의 특성상 많은 계산시간을 필요하게 되며, 이에는 Leave-1-Out법과 N-Folding법이 있다. Leave-1-Out법은 전체 패턴 중 1개를 제외한 모든 패턴을 학습패턴으로 사용하고, 제외된 1개의 패턴을 테스트 패턴으로 하여 분류를 시도하는 방법으로 모든 패턴이 각각 한번씩 테스트 패턴으로 사용될 때까지 분류를 계속하는 방법이다. 비슷한 방법으로 N-Folding기법은 전체패턴을 N개의 그룹으로 분할하고 각 그룹을 한번씩 돌아가면서 테스트 패턴으로 사용하는 방법이다[15].

ARPA에서는 k값을 학습시에 결정하지 않고 패턴의 분류시에 결정하게 되며, k값은 가변적으로 결정된다. ARPA에서는 패턴의 분류시 가장 인접한 초월평면과 그 다음으로 가까운 패턴의 클래스가 같을 경우 k=1인 NN분류기와 같은 방법으로 분류하게 되며, 만일 가까운 두 패턴의 클래스가 다를 경우, 데이타를 구성하는 모든 클래스에서 적어도 하나의 초월 평면이 추출될 때까지 거리 순서로 패턴을 추출하게되며 이것이 k값이 된다. 따라서 ARPA에서 분류 대상 초월평면이 수는 현재 입력패턴과 가까운 초월 평면의 개수에 따라 가변적이 된다. 그 후 입력패턴의 분류는 가장 많은 초월평면이 소속된 클래스로 분류한다.

4. 실험 및 분석

ARPA 기법을 이용한 분류기의 성능을 k-NN, FPA, RPA 기법과 비교하여 검증하였다. 실험은 기계학습의 벤치마크 자료로 사용되는 7개의 데이타를 이용하였으며, 실험 방법은 70:30법(전체 데이타를 기준으로 70%는 학습패턴으로, 30%는 평가패턴으로 사용하는 방법)을 사용하였다[15]. 이 때 70%의 학습패턴은 전체 패턴의 클래스별 분포를 고려하여 모든 클래스에서 같은 비율로 추출하였다. 실험은 Windows 2000을 적재한 PentiumIII 컴퓨터를 사용하였으며, 모든 실험결과는 25회 반복측정한 후 평균값으로 나타내었다.

4.1 실험 데이타

본 논문에서는 기계 학습의 벤치마크 자료로 사용되는 7개의 데이타를 UCI Machine Learning Database Repository에서 발췌하여 사용하였으며, 이들 7 개의 데이타는 Breast-Cancer Wisconsin, Glass, Ionosphere, Iris, New-Thyroid, Sonar, Wine이며, 이들 데이타는 모든 특징이 실수 값을 갖는다. 다음의 표 1은 실험자료의 특성, 표 2는 7개의 데이타를 70:30법을 이용하여 나누었을 경우, 클래스별 학습패턴의 분포를 보여주고 있다.

표 1 실험 데이타셋의 구성

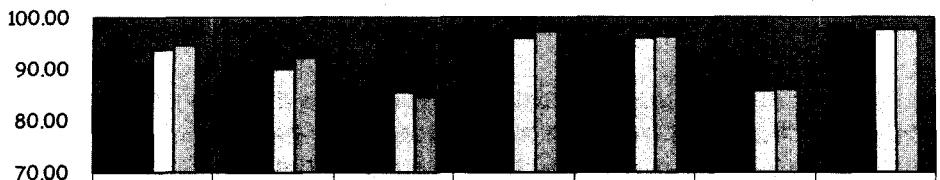
		개수	클래스 개수
	699	10	2
	214	10	6
	351	34	2
	150	4	3
	215	5	3
	208	60	2
Wine	178	13	3

표 2 클래스별 학습패턴 분포

	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 분류성능 실험

그림 7의 k-NN, FPA, RPA, ARPA의 분류성능을 보면, ARPA 기법의 성능이 전체 학습패턴을 고려하는 k-NN과 거의 대등한 분류성능을 보이고 있다. 또한 FPA 기법과의 비교를 보면, Ionosphere 데이타를 제외한 나머지 데이타 모두에서 ARPA가 우수한 분류성능을 보장하고 있으며, RPA의 경우도 Ionosphere에서만 약간 저조한 분류 성능을 보이고 있다.



	Breast Cancer	Glass	Ionosphere	Iris	New Thyroid	Sonar	Wine
■k-NN	96.00	89.70	86.11	98.40	96.86	86.19	98.70
■FPA	90.70	91.00	86.80	95.60	95.90	85.30	97.40
□RPA	93.76	90.00	85.51	95.91	96.06	85.75	97.53
□ARPA	94.61	92.17	84.64	97.17	96.14	86.07	97.70

그림 7 분류성능의 비교

메모리 기반 분류기에서, 저장된 학습 패턴 중 클래스 경계면 근처에 있는 패턴들 보다 경계면에서 멀리 떨어진 패턴일수록 좀 더 정확한 분류를 보장한다는 연구결과가 있는데[5,6], 본 논문에서 제안한 ARPA기법은 클래스 경계면 근처에 접근할수록 좀더 작은 크기의 초월평면으로 재귀 분할하면서 초월 평면을 생성한다. 따라서 클래스 경계면 근처에 분포한 패턴의 수가 줄어들게 되며, 초월평면 최적화 작업시에도 경계면 근처의 초월평면 크기가 줄어드는 효과를 얻을 수 있다. 반면에 FPA기법은 전체 패턴공간을 같은 크기로 분할하므로 클래스 경계면이 복잡한 경우, 경계면 근처에 분포한 대부분의 패턴을 원형 그대로 적용하게 되므로, 오분류되는 패턴의 수가 ARPA기법에 비하여 증가하게 된다.

위 실험에서 k-NN 기법은 사전에 최적의 k값을 미리 결정하여야 한다는 단점이 있으며, FPA 기법에서는 특징축의 분할개수를 미리 결정해야 한다. 반면에 본 논문에서 제안한 ARPA 기법에서는 RPA와 마찬가지로 k값이나 특징축의 분할개수, 즉 외부 파라미터를 전혀 고려치 않아도 된다는 장점을 갖는다.

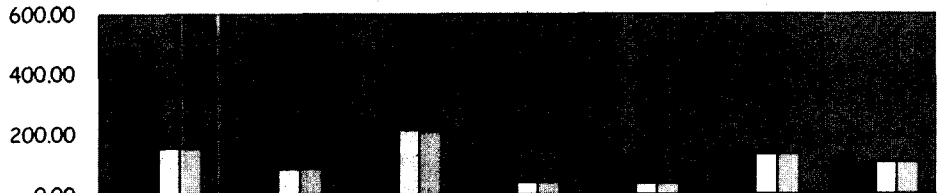
위 실험에서 k-NN 분류기의 성능은 Leave-1-Out Cross Validation기법을 사용하여 계산한 최적의 k값을 사용한 것이며, 다음의 표 3은 각 데이타에서 사용된 k-NN 분류기의 k값을 보여주고 있다.

4.3 메모리 사용량 비교 실험

그림 8의 실험결과에서는 k-NN, FPA, RPA, ARPA 네 가지 방법을 이용한 분류기의 메모리 사용량을 보여주고 있다. 결과에서 보면 k-NN의 경우 모든 학습패턴을 메모리에 저장하고 분류 시 입력패턴을 모든 학습패턴과 비교한다. 하지만 FPA, RPA는 주어진 패턴공간을 초월평면으로 분할하여 각 초월평면을 대표하는 패

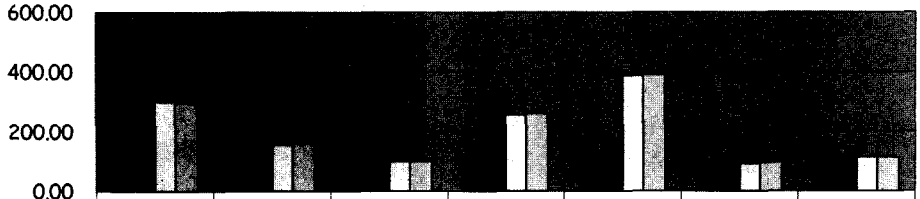
표 3 분류성능 최적화를 위한 k값

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



	Breast Cancer	Glass	Ionosphere	Iris	New Thyroid	Sonar	Wine
■k-NN	488.00	148.00	245.00	105.00	150.00	144.00	123.00
■FPA	414.00	132.80	221.80	30.30	123.70	142.70	98.70
□RPA	152.96	85.92	211.60	39.28	37.50	133.24	103.70
□ARPA	155.40	85.00	209.70	39.60	37.00	132.00	104.60

그림 8 메모리 사용량의 비교



	Breast Cancer	Glass	Ionosphere	Iris	New Thyroid	Sonar	Wine
■k-NN	96.00	89.70	86.11	98.40	96.86	86.19	98.70
■FPA	106.91	101.42	95.88	331.29	116.29	86.08	121.38
□RPA	299.13	155.03	99.01	256.38	384.24	92.67	115.68
□ARPA	297.10	160.48	98.89	257.65	389.76	93.89	114.89

그림 9 분류성능/메모리 사용량

턴을 저장하는 방법을 사용하며, ARPA는 초월 평면 형태로 저장하는 방법을 사용함으로써 우수한 메모리 사용효율을 보장하고 있다.

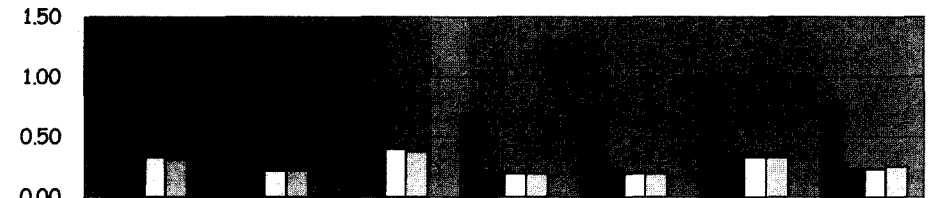
메모리 사용량은 전체 데이타 셋에서 RPA와 비슷한 성능을 보이고 있다. New-Thyroid 데이타의 경우 k-NN대비 약 25% 정도의 메모리만을 사용하고 있으며, Breast-Cancer 데이타의 경우는 30% 정도, Iris 데이타의 경우는 40% 정도의 학습패턴만을 메모리에 저장하는 것을 볼 수 있다. 또한 나머지 데이타에서도 약 60-80% 정도의 학습패턴만을 메모리에 저장한다. FPA 기법과의 비교에서는 Iris, Wine 데이타를 제외한 5개의 데이타 셋에서도 우수한 메모리 사용효율을 보이고 있으며, RPA와의 비교에서는 Glass, Ionosphere, New Thyroid, Sonar 4개의 실험 셋에서 상대적으로 우수한 메모리 사용효율을 보장한다. 이것은 FPA 기법에서 클래스 경계면이 특징 축과 평행하게 분포하는 경우뿐만 아니라 RPA 기법에서 클래스 경계면이 비 평행한 경우에도 재귀 분할을 통한 메모리 절감 효과를 얻을 수 있다. 이들이 가지는 장점에 대해 ARPA의 경우는 초월평면 최적화 단계에서 필요하지 않은 초월 평면을 삭제할 수 있기 때문에 좀더 나은 메모리 사용 효율을 보이게 된다.

실험 4.2와 4.3에서 보는 것처럼 본 논문에서 제안한 ARPA 기법이 메모리 사용효율을 고려한 분류성능에 있어서 기존의 k-NN, FPA, RPA와 비교하여 우수한 성능을 보이고 있는 것을 볼 수 있다.

그림 9의 분류성능/메모리 사용량 비교에서 메모리 사용량은 k-NN분류기에서 사용하는 학습패턴의 개수를 1로 보았을 때, FPA, RPA, ARPA의 메모리 사용량을 적용한 결과이다.

4.4 분류소요시간 비교

메모리 기반 학습 기법을 이용한 분류기의 경우 메모리에 저장된 패턴의 개수와 입력패턴의 분류시간은 직접적인 관계를 가지게 되므로, 본 논문에서 제안한 대표



	Breast Cancer	Glass	Ionosphere	Iris	New Thyroid	Sonar	Wine
■k-NN	1.19	0.79	1.03	0.69	0.69	0.94	0.76
■FPA	0.52	0.24	0.41	0.19	0.21	0.35	0.23
□RPA	0.33	0.21	0.39	0.19	0.19	0.33	0.23
□ARPA	0.31	0.22	0.38	0.20	0.20	0.34	0.25

그림 10 분류소요시간 비교

초월평면을 추출하는 ARPA 기법은 실제 패턴의 분류에 소요되는 시간에 있어서도 k-NN 기법보다 월등히 빠른 분류 속도를 보장한다.

그림 10의 결과에서, ARPA기법은 분류기 성능에 영향을 미치는 k 값을 사전에 결정하지 않고, 저장된 학습 패턴의 수를 줄임으로써 학습과 분류에 소요되는 시간이 모든 데이타 셋에 있어 k-NN기법에 비하여 월등히 적게 소요되는 것을 볼 수 있다. 결과에 표시된 값은 $\log_{10}$(소요시간)을 나타내며, 이때의 소요시간은 각 25회 반복 실험하는데 소요되는 실 소요시간(초)을 나타낸다.

5. 결 론

본 논문에서는, 메모리 기반 추론에 있어 효율적인 메모리 사용과 분류성능을 향상시킬 수 있는 ARPA 알고리즘을 제안하였다. 본 논문에서 제안한 ARPA 기법은 최적 초월평면 형성을 통해 메모리에 저장되는 학습 패턴들을 초월평면으로 대체하는 방법을 채택하였으며, 실험 결과에서 볼 수 있는 것처럼 제안된 ARPA 기법은 k-NN기법과 비교하여 모든 데이타 셋에서 적은 메모리 공간을 필요로 하며, FPA, RPA 기법과의 비교에 있어서는 실험에 사용한 벤치마크 자료 대부분에서 적은 공간을 사용하고 있다. 또한 분류 성능면에서 기존의 k-NN 기법과 거의 비슷한 성능을 보이며, FPA, RPA 기법에 비해서 6개의 데이타 셋에서 우수한 성능을 보이고 있다. 마지막으로 RPA 기법에서는 분류에 영향을 미치는 패턴의 위상을 고려하지 않은 반면 ARPA에서는 특징의 영향력과 패턴의 분포를 초월평면 형성에 반영함으로써 좀더 우수한 분류 성능을 보장 할 수 있었다.

마지막으로 ARPA의 경우도 클래스 경계면에 분포한 패턴을 완전히 제거할 수 없으며, 이 패턴들을 제거할 경우 좀더 나은 분류 성능이 기대된다.

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-NN A 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 평균알고리즘을 사용하는 새로운 메모리 기반 추론, 한국정보처리학회 논문지 제6권 제6호, pp1563-1570, 1999.

[10] D. Wettschereck, 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.

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

[17] 최영희, 장수민, 유재수, 오재철, 수량적 연관규칙탐사를 위한 효율적인 고빈도 항목열 생성기법, 한국정보처리학회 논문지 제6권 제10호, pp. 2597-2607, 1999.

이 형 일

1985년 명지대학교 전자계산학과 졸업(학사). 1994년 명지대학교 대학원 전자계산과(석사). 2000년 명지대학교 대학원 컴퓨터공학과(박사). 1985년~1990년 (주)쌍용정보통신. 1990년~1996년 (주)시에치노컨설팅. 1997년~현재 김포대학 컴퓨터계열 부교수. 관심분야는 기계학습, 에이전트시스템, 정보검색

최 학 윤

1985년 2월 숭실대학교 전자공학과(공학사). 1987년 2월 숭실대학교 대학원 전자공학과(공학석사). 1999년 8월 건국대학교 대학원 전자공학과(공학박사). 1996년 3월~현재 김포대학 전자정보계열 조교수. 관심분야는 신호처리, 안테나 및 전파전파