정보학연구 제1권 제1호 1998년 3월(pp. 65~84)

k_NN 분류기의 메모리 사용과 점진적 학습에 대한 연구

이형일 · 윤충화
김포대학 컴퓨터계열 전임강사
명지대학교 컴퓨터공학과 교수

요 약

메모리 기반 추론 기법은 분류시 입력 패턴과 저장된 패턴들 사이의 거리를 이용하는 교사 학습 기법으로써, 거리 기반 학습 알고리즘이라고도 한다. 메모리 기반 추론은 k_NN 분류기에 기반한 것으로, 학습은 추가 처리 없이 단순히 학습 패턴들을 메모리에 저장함으로써 수행된다.
본 논문에서는 기존의 k-NN 분류기보다 효율적인 분류가 가능하고, 점진적 학습 기능을 갖는 새로운 알고리즘을 제안한다. 또한 제안된 기법은 노이즈에 민감하지 않으며, 효율적인 메모리 사용을 보장한다.

Ⅰ. 서론

기계학습 알고리즘 중 메모리 기반 학습 알고리즘은 메모리에 저장된 패턴과 학습 패턴 사이의 거리를 이용하여 학습을 시키는 방법으로 거리기반 학습법이라고도 한다[4]. 이러한 메모리 기반 학습 알고리즘은 k-NN 학습 알고리즘을 기반으로 하고 있으며, k-NN 학습 알고리즘에서 학습은 단순히 모든 학습 패턴을 메모리에 저장하는 것이다. k-NN 학습 알고리즘을 사용한 분류기에서는 입력패턴과 메모리 상에 저장된 모든 패턴 사이의 거리를 계산하여 입력패턴과 가장 가까운 위치에 존재하는 k개의 패턴을 선정, 그 중 가장 많은 패턴이 속하는 클래스로 분류한다[3,7]. k-NN 학습 알고리즘을 사용한 분류기는 주어진 학습패턴에 노이즈 가 없을 경우 효과적으로 동작하며, 학습패턴에 노이즈가 있을 경우 분류기의 성능이 급격히 감소 하게된다[1,5]. 또한 학습패턴 전체를 메모리에 저장하여야 하므로 다른 기계학습 방법에 비하여 상대적으로 많은 저장 공간을 필요로 하게된다[1,5]. 또한 메모리에 저장된 모든 패턴과의 거리를 계산하여야 하므로 상대적으로 많은 분류시간을 요한다는 단점이 있다.

본 논문에서는 메모리 기반 학습 알고리즘 의 기반이 되는 k-NN에 대한 연구를 통하여 보다 높은 인식률을 보장하며 적은 메모리 공간을 필요로 하는 새로운 알고리즘을 제안한다. 또한 제안된 알고리즘에서는 k값을 입력패턴의 분류시에 결정하므로 점진적 학습이 가능하다.

Ⅱ장에서는 기존의 k-NN 학습 알고리즘에 대한 고찰과 k-NN분류기의 특성을 실험을 통하여 분석한다. Ⅲ장에서는 노이즈 적응능력과 상

논문접수일 : 97년 10월 게재확정일 : 98년 2월

대적으로 적은 메모리 사용 및 빠른 분류가 가능한 알고리즘을 제시하고 Ⅳ장에서는 제안된 알고리즘의 성능을 실험적으로 검증한다. 마지막으로 Ⅴ장에서는 결론 및 향후 연구과제에 대한 고찰을 한다.

Ⅱ. k-NN 학습법

k-NN 학습 알고리즘에서의 학습은 전체 학습 패턴을 단순히 메모리에 저장하는 것이며, 입력 패턴의 분류는 현재 메모리에 저장된 패턴 중 입력 패턴과 가장 가까운 k개의 패턴을 선택한다. 이때 입력패턴과 메모리에 저장된 패턴의 거리를 계산하기 위하여 일반적으로 유클리드 거리를 사용하며, 분류기준에 따라 크게 Majority k-NN과 Weight-Vote k-NN으로 나눌 수 있다[3].

2.1 Majority k-NN과 Weight-Vote k-NN

2.1.1 Majority k-NN

k-NN의 가장 기본 적인 알고리즘으로 다음과 같이 동작한다.

1) 주어진 학습패턴을 메모리에 저장한다.
2) 학습패턴 E와 입력패턴 P 사이의 거리 D_{EP} 다음 식1에 의해 결정한다.

$$D_{EP} = \sqrt{\sum_{i}^{n} (E_{fi} - P_{fi})^2} \qquad (식1)$$

이때 n은 패턴을 구성하는 특징의 개수이며, E_{fi}, P_{fi}는 각각 학습패턴과 입력패턴의 i번째 특징 값이다.

3) 입력패턴과 가장 가까운 위치에 있는 k개의 패턴을 메모리 상에서 선택한다.
4) 선택된 k개의 학습 패턴 중 가장 많은 패턴이 소속된 클래스를 출력 클래스로 결정한다.

2.1.2 Weight-Vote k-NN

학습 알고리즘은 다음과 같다.

1) 2.1의 Majority k-NN과 같은 방법으로 k개의 학습 패턴을 선택한다.
2) 선택된 k개의 패턴과 입력패턴 P와의 거리를 고려하여 출력 클래스를 결정한다.

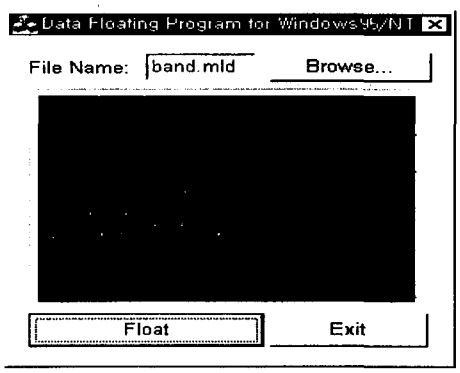
$$O = \max\{\sum_{i=1}^{k}\sum_{c=1}^{m}\frac{(c_{n_i} = c)}{D_{EP}}\} \qquad (식2)$$

이때 k는 입력패턴과 가장 가까운 학습패턴의 개수이며, m은 전체 클래스 수이다. 또한 (C_{ni} = c)는 i번째로 가까운 학습패턴의 클래스가 c일 경우 1을, 아닐 경우 0을 나타낸다.

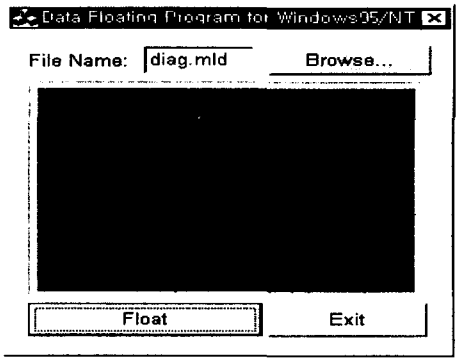
3) 식2 에서 출력클래스 O는 선택된 k개의 학습패턴과 입력패턴사이에 존재하는 거리의 역수를 클래스 별로 합산하여 가장 큰 값을 가지는 클래스를 출력 클래스로 선택하게 된다.

이처럼 $_{wv}$K-NN의 경우 입력 패턴과 멀리 떨어진 패턴일수록 출력 클래스의 결정에 미치는 영향이 작아지는 것을 볼 수 있다. $_{wv}$k-NN의 경우 최적의 k값이 15보다 작은 도메인에서 효과적으로 동작한다[2].

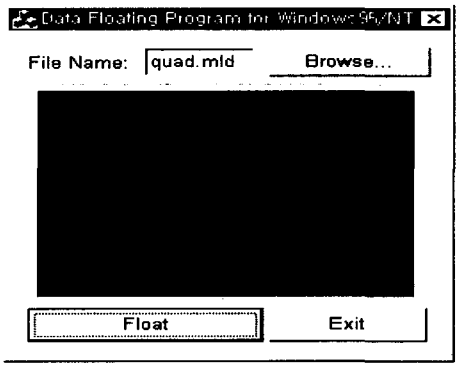
$_{Majority}$k-NN과 $_{wv}$k-NN에서는 k가 분류기의 성능에 영향을 미치는 유일한 파라미터이다[2].



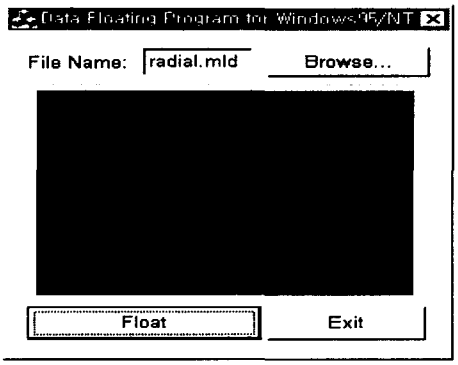
a) Band



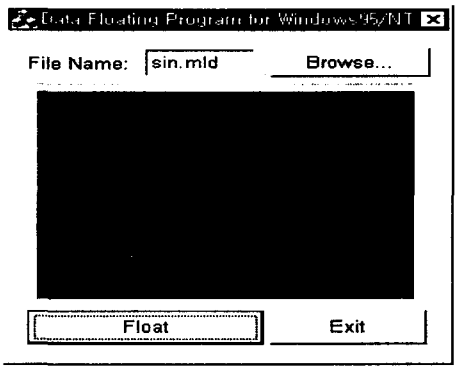
b) Diagonal



c) Quadrant



d) Radial



e) Sinusoidal

그림1 합성 데이터 셋 패턴분포

2.2. k-NN의 문제점 분석

2.2.1 실험 데이터 셋

본 문에서는 실험을 위하여 5개의 합성 데이터 셋과 UCI Repository에서 가져온 3개의 실험 데이터 셋을 선택하여 사용한다.

1) Quadrants, Diagonal, Banded, Radial, Sinusoidal 합성 데이터 셋

합성 데이터 셋의 경우 해당 데이터 셋을 구성하는 패턴에 포함된 노이즈의 정도, 클래스 결정 영역의 모양 등을 실험 이전에 명확히 알 수 있으므로, 패턴의 특성에 따른 분류기의 성능 분석에 유용하게 사용할 수 있다.

이들 5개의 합성데이터 셋은 위의 그림과 같은 패턴 분포를 가지고 있으며, 이중 Quadrants와 Banded는 축에 평행한 결정영역 (Decision Boundary)을 가지며, Diagonal은 대각선 결정영역을 가진다. 또한 Sinusoidal의 경우는 사인커브 모양의 결정영역을 가지며, Radial은 5개의 클래스가 환형을 이루면서 중첩(nested)된 모양을 띈다[5].

<표 34> 합성 데이터 셋의 구성

데이터 셋	패턴 수	특징 수	클래스 수
Band	500	2	10
Diagonal	500	2	2
Quadrant	500	2	2
Radial	500	2	5
Sinusoidal	500	2	2

본 논문에서 사용하는 합성 데이터 셋은 Thomas G. Dietterich가 작성한 프로그램에 의하여 생성하였다[5]. 또한 위의 합성 데이터 셋은 노이즈가 포함되어있지 않은 데이터이다.

2) Glass, Iris, Wind 데이터 셋

본 논문에서는 k-NN분류기의 성능 분석 및 제안된 알고리즘의 성능 평가를 위하여 기계학습의 벤치마크 자료로 사용되는 데이터 셋 중 Glass, Iris, Wine 3개의 데이터 셋을 실험에 사용하였다[6, 7, 8]. 실험 데이터 셋의 특성은 표2와 같다.

<표 33> 실험 데이터 셋의 구성

데이터 셋	패턴 수	특징 수	클래스 수
Glass	214	10	7
Iris	150	4	3
Wine	178	13	3

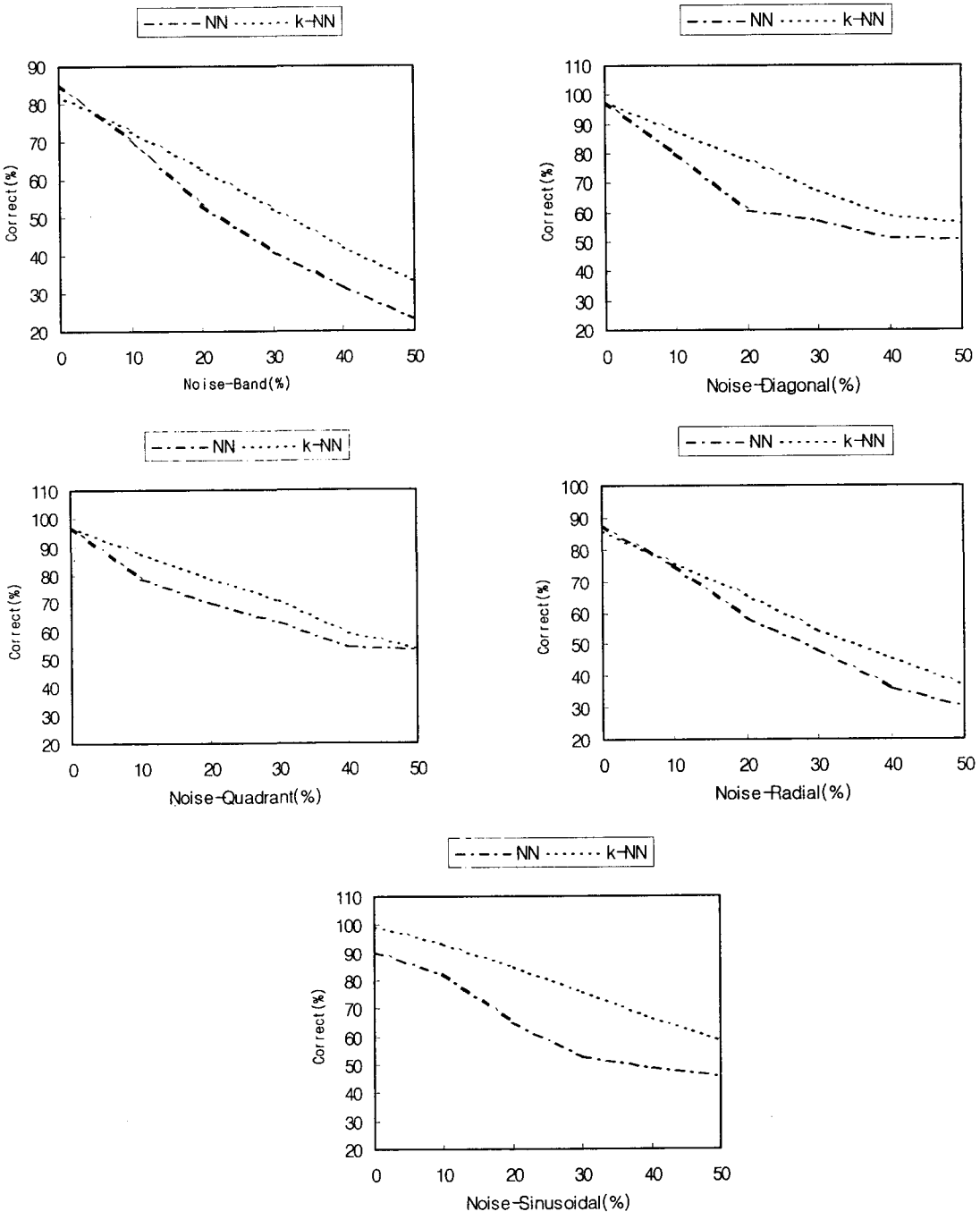
2.2.2 k-NN 학습 알고리즘의 성능 분석

여기에서는 k값에 따른 분류기의 성능변화와 학습패턴에 노이즈가 있을 경우 k-NN분류기의 성능 변화를 실험적으로 보여준다.

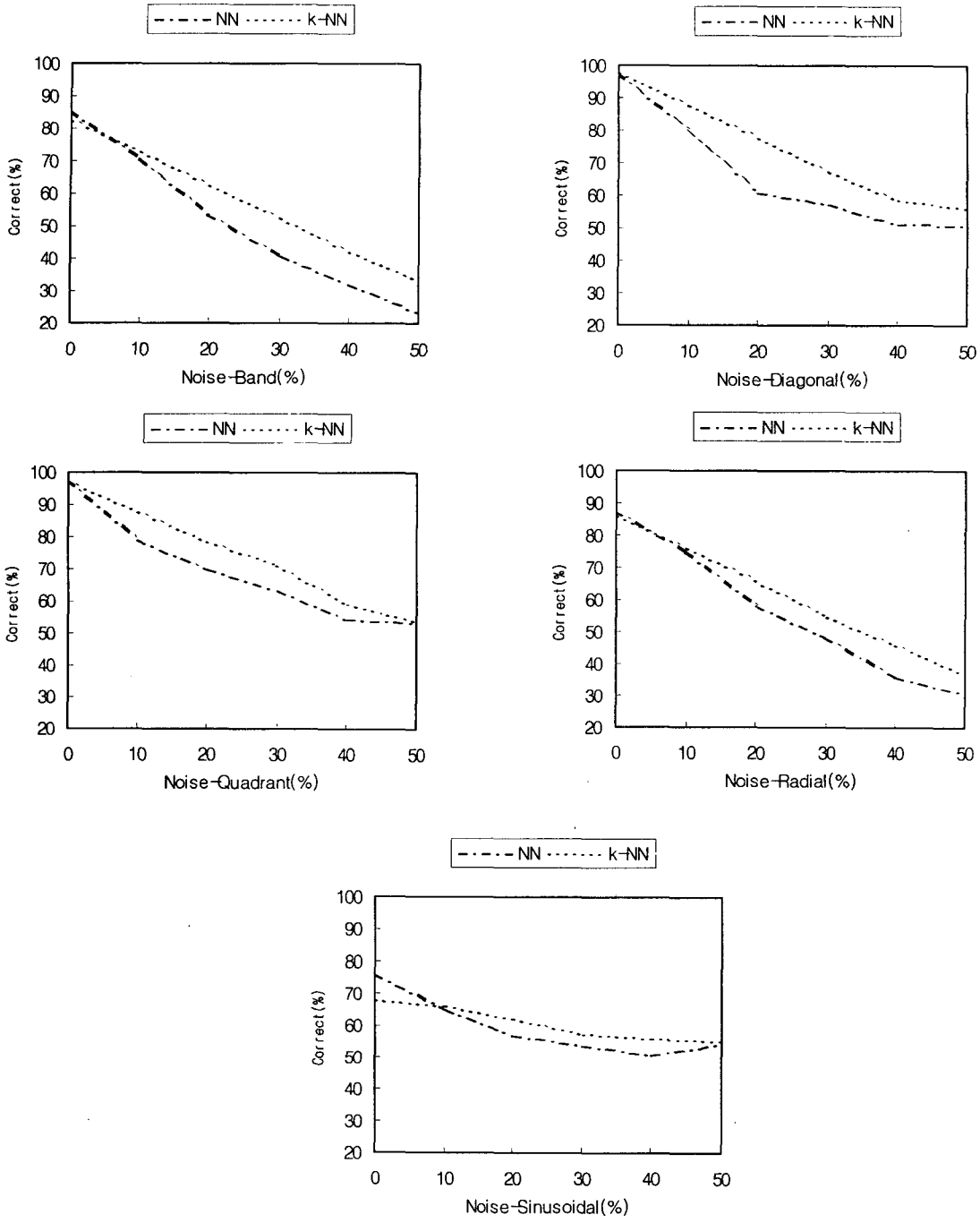
1) k값의 역할

k-NN분류기에서 분류기의 성능에 영향을 미치는 유일한 파라미터는 k값으로, 이 값은 입력패턴의 분류시 몇 개의 가장 근접한 패턴을 대상으로 할 것인가를 결정하는 척도로 사용되며 k-NN분류기에서 k=1인 경우를 NN분류기라고도 한다[8].

다음의 그림2는 5개의 합성 데이터 셋을 이용하여 k값의 변화에 따른 Majority k-NN의 성능 변화를 실험한 것이다. 이 실험에서는 전체 테스트 패턴 중 70%를 학습패턴으로 사용하고 나



(그림 4) 클래스 노이즈와 분류성능



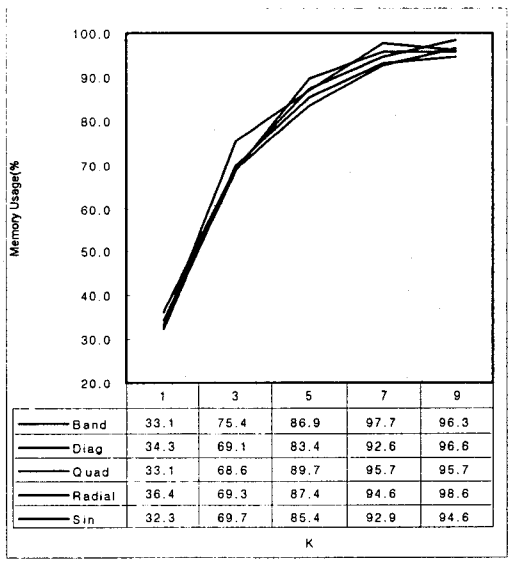
(그림 5) 특징 노이즈와 분류성능

(2) 특징 노이즈

클래스 노이즈의 실험과 동일한 조건에서 특징 노이즈가 포함된 데이터 셋을 대상으로 성능 변화를 측정하였다. 결과는 그림5에서 볼 수 있는 것처럼 NN, k-NN모두 클래스 노이즈가 포함된 경우와 비슷한 결과를 보이고 있다.

2.3 k-NN에서의 메모리 사용

k-NN학습법에서의 학습은 단순히 학습패턴으로 주어진 모든 패턴들을 메모리 상에 저장하는 것이다. 따라서 주어진 학습패턴의 개수만큼의 메모리 공간을 필요로 하게 되며, 이것은 다른 기계학습 방법에 비하여 상대적으로 많은 메모리 공간을 필요로 하는 것이 된다. 또한 모든 학습패턴을 메모리에 저장하고 입력패턴의 분류시, 저장된 모든 패턴과의 거리계산이 수행되어야 하므로, 분류시에도 상대적으로 많은 시간을



	1	3	5	7	9
Band	33.1	75.4	86.9	97.7	96.3
Diag	34.3	69.1	83.4	92.6	96.6
Quad	33.1	68.6	89.7	95.7	95.7
Radial	36.4	69.3	87.4	94.6	98.6
Sin	32.3	69.7	85.4	92.9	94.6

(그림 6) k-NN 분류기의 메모리 사용효율

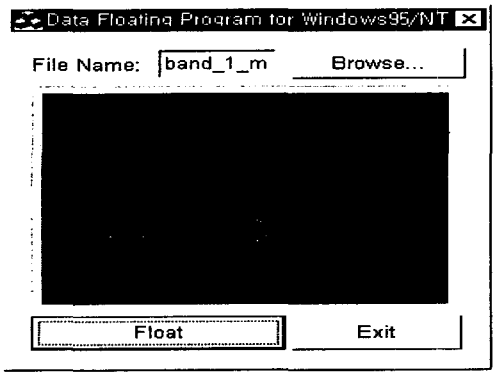
필요로 한다.

2.3.1 k-NN에서의 메모리 사용량 분석

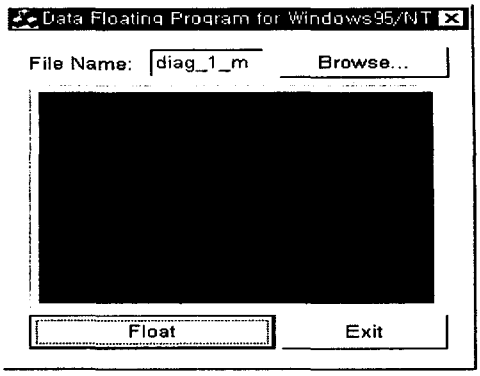
다음 실험은 k-NN분류기에서 메모리 사용량을 효율을 측정하기 위하여, 5개의 합성데이터 셋에 대하여 1회 분류를 시도하여 메모리에 저장된 학습 패턴 중 입력패턴의 분류에 사용된 학습패턴의 비율을 측정한 것이다. 이때 k값은 1에서 9 까지 2씩 증가하면서 사용하였으며, 학습패턴의 비율은 전체 패턴의 70%로 데이터 셋을 구성하는 모든 클래스에서 같은 비율로 학습패턴을 추출하였다.

그림6에서 보는 것처럼 k-NN분류기에서 메모리에 저장된 학습패턴 중 입력패턴의 분류에 사용된 비율은, NN의 경우 약 30%정도만 사용하고 있으며, k값이 증가할수록 대부분의 학습패턴이 분류에 사용되고 있는 것을 볼 수 있다. 즉 NN분류기의 경우 학습패턴 중 직접 분류에 사용되는 패턴만을 저장하여도 분류기의 성능에는 변화가 없다는 것을 알 수 있다. 다음 그림 3.7은 위의 실험에서 각 합성 데이터 셋에서 분류에 사용된 학습 패턴의 분포를 보여주고 있다. 패턴 분포에서 볼 수 있는 것처럼 실제 분류에 사용된 학습 패턴은 전체 패턴 중 상대적으로 가깝게 배치되어 있던 패턴 중 일부만이 학습 패턴으로 사용되는 것을 알 수 있다.

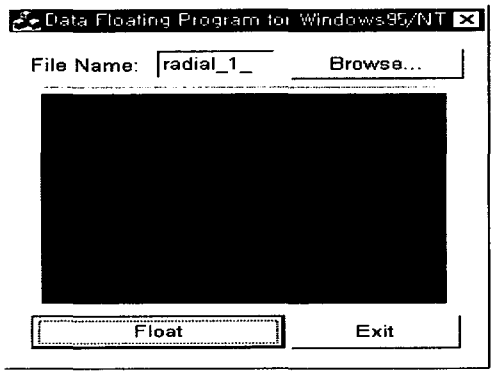
따라서 분류기에서 사용하는 학습패턴 중 다른 학습 패턴에 비하여 상대적으로 가깝게 위치한 패턴의 경우 그들 중 하나 또는 몇 개만을 선택하여 저장하여도 분류기의 성능에 커다란 변화가 없다는 것을 알 수 있다. 이처럼 전체 학습패턴을 저장하지 않고 일정한 개수의 학습패턴들을 그룹화 하여 그들을 대표하는 패턴만을 학습패턴으로 저장하는 기법 중 대표적인 방



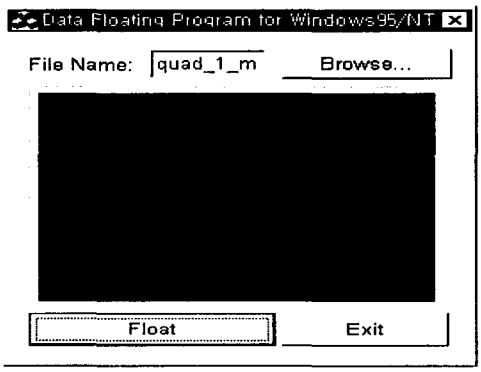
a) Band



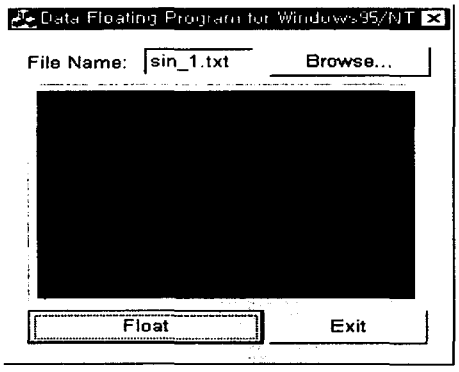
b) Diagonal



d) Radial



c) Quadrant



e) Sinusoidal

(그림 7) NN분류기에서 사용된
학습패턴의 분포

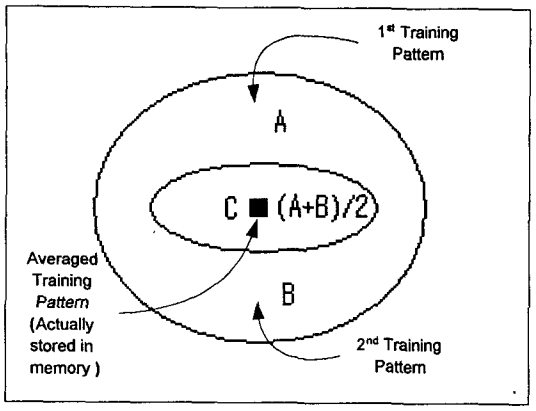


그림8. 패턴평균 기법의 문제점

법은 패턴평균(Instance Averaging)법을 들 수 있다[5]. 이 방법에서는 주어진 학습패턴을 특정한 기준에 의하여 몇 개씩 그룹화 한 후, 그 평균값을 하나의 학습패턴으로 저장한다. 하지만 이 방법의 경우 몇 개의 학습 패턴을 평균하여 대표 패턴으로 저장할 때, 그룹화 방법의 결정 등이 문제가 되며, 주어진 패턴공간에서 클래스가 그림8의 경우처럼 환형의 띠를 이룰 때 A, B의 평균값으로 산출한 대표패턴 C가 클래스의 영역 밖인 띠의 안쪽으로 이동하는 경우가 발생하게 된다. 따라서 패턴평균법을 사용할 경우 클래스가 교차되는 영역에 존재하는 패턴의 경우 평균값을 이용하여 저장 패턴을 결정할 수 없게 된다[2].

Ⅲ. Hybrid-NN (H-NN)

Ⅱ장의 실험에 의하여 k-NN분류기가 가지는 문제점은 다음으로 요약할 수 있다.

첫째, NN분류기의 경우 데이터에 노이즈가 포함되어 있을 경우 분류기의 성능이 급격히 감소한다. 둘째, K-NN분류기의 경우 최적의 분류성능 보장을 위하여 사전에 k값을 결정하여야 하며, 이 방법으로 Cross-Validation법을 사용할 경우 점진적 학습이 불가능하다. 셋째, NN분류기의 경우 전체 학습패턴 중 30%정도만이 분류에 사용된다. 즉, 70%의 불필요한 패턴을 메모리에 저장하고 있다.

본 논문에서는 위에 주어진 3가지 문제점을 보완하기 위하여 다음과 같은 방법을 사용한다.

첫째, k-NN에서의 k값이 가지는 노이즈 필터링 효과를 이용하여 데이터에 포함된 노이즈로 인한 분류기 성능 저하를 방지한다. 이때 k값은 표4의 알고리즘에 의해 입력패턴의 분류시에 결정되며 가변적인 값을 가진다. 둘째, k 값의 결정을 입력패턴의 분류시에 가변적으로 결정함으로써, 추후 학습패턴이 추가되어도 분류기의 성능에 영향을 미치지 않는다. 셋째, 패턴평균법에 기반 하여 주어진 학습패턴을 대표하는 대표패턴만을 메모리에 저장하되, Ⅱ장에서 지적한 문제점을 해결하기 위하여 패턴 공간을 충분히 작은 크기의 초월평면 (Hyperrectangle)으로 분할한 후, 각 초월평면 단위로 패턴평균을 구한다.

위의 방법을 사용하여 노이즈가 포함된 데이터에서도 효율적인 분류가 가능한 증가학습 알고리즘을 제시하고, 이것을 H-NN(Hybrid-NN)이라 칭한다. 제안된 알고리즘은 k-NN에 비해 적은 량의 메모리 공간을 필요로 하며, 패턴 분류의 속도에 있어서도 우월한 성능을 보인다. 또한 점진적 학습이 가능한 알고리즘이다.

다음 표 4는 본 논문에서 제시한 h-NN학습 알고리즘이다.

먼저 학습부분에서는 k-NN에서와 같이 모든

<표 4> 패턴평균 학습 알고리즘

학습

1. 주어진 학습 패턴을 모두 메모리에 저장한다.
2. 패턴공간을 일정한 크기의 초월평면으로 분할한다. 이때 초월평면은 축에 평행한 형태의 초월평면이다.
3. 분할된 각각의 초월평면에 대하여 패턴평균 기법을 적용, 대표패턴을 생성한다. 이때 하나의 초월평면에 각기 다른 클래스에 속하는 학습패턴이 포함되어 있을 경우, 패턴평균기법을 적용하지 않고 모든 학습패턴을 저장한다.

분류

4. 입력패턴과 가장 가까운 학습패턴을 선택(T1).
5. 입력패턴과 두 번째로 가까운 패턴을 선택(T2).
6. Class No(T1)==Class No(T2)일 경우 입력패턴(q)을 Class No(T1)으로 분류.
7. 6의 조건을 만족하지 않을 경우, q에서의 거리를 기준으로 가까운 곳에 위치한 n개의 패턴(Tn)을 선택. 이때 n은 모든 클래스에서 적어도 하나이상의 패턴이 추출될 때까지 선택된 패턴의 개수이다.
8. 선택된 n개의 패턴 중 가장 많은 패턴을 포함하는 클래스로 입력패턴을 분류.

학습패턴을 메모리에 저장하고, 효율적인 메모리 사용을 위하여 패턴평균 기법을 적용한다. 이때 패턴평균법은 주어진 패턴공간을 일정한 크기의 중복되지 않은 초월평면으로 분할한 후 적용되는데, 이때 초월평면의 각 축의 크기는 모두 같으며, 크기는 주어진 학습패턴의 개수에 따라 가변적으로 구해진다.

h-NN에서 패턴공간을 분할하는 초월평면의 축의 길이 및 개수는 다음과 같은 방법으로 구한다.

전체 축의 크기의 1/10으로 하였다. 본 논문에서는 패턴공간을 나타내는 전체 초월평면의 각 축의 길이를 1.0으로 정규화 하여 사용하고 있으므로 h-NN에서 패턴평균을 구하기 위한 하나의 초월평면은 각 축의 길이가 0.1인 초월평면이 된다.

<표 5> 초월평면 분할 알고리즘

1. 초월평면의 개수(H) : 전체 학습패턴 수 * 0.3
2. 초월평면 축의 길이 : 1/R, 여기서 R은 크기 1.0인 축을 R개로 분할하는 것으로 R은 다음 식을 만족하는 값으로 한다.

   $H \fallingdotseq R^n$

   이때 n은 패턴공간의 차원 수이다.

Ⅳ. 실험 및 결과

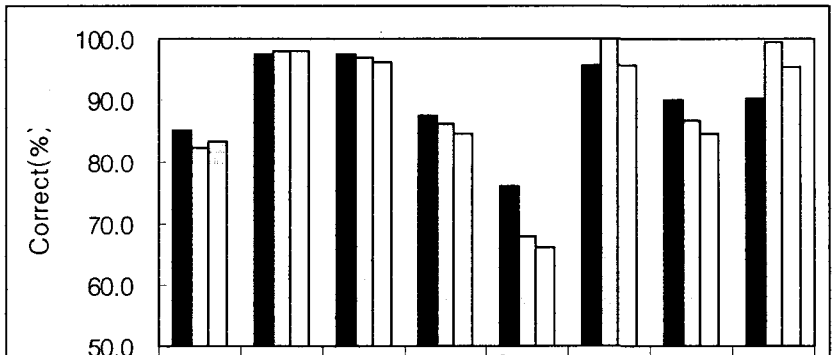
여기에서는 본 논문에서 제안한 h-NN알고리즘의 성능을 측정하기 위에서 제시한 3가지 문제점에 기준 하여 NN, k-NN과 비교실험을 하였다. 또한 h-NN과 다른 두개의 분류기의 분류 속도를 실시간으로 측정하는 비교 실험도 실시하였다. 본 논문에서의 모든 실험에서는 Pentium II-233 with 196Mb RAM을 사용하였으며 운영체제는 WindowsNT 4.0을 사용하였다.

4.1 분류성능 및 노이즈 저항 능력 실험

여기에서는 먼저 8개의 데이터 셋에 대하여 NN, k-NN, h-NN의 분류 성능을 비교한 후 클래스 노이즈와 특징 노이즈를 각각 10-50%까지 추가하면서 이들 세 가지 방법의 노이즈 저항 능력을 측정하였다. 이 실험에서는 각 실험에 대하여 25회 반복 측정한 후 평균값을 결과로 사용하였으며 학습패턴은 데이터 셋을 구성하는 모든 클래스에서 70%씩 추출하여 사용하였다.

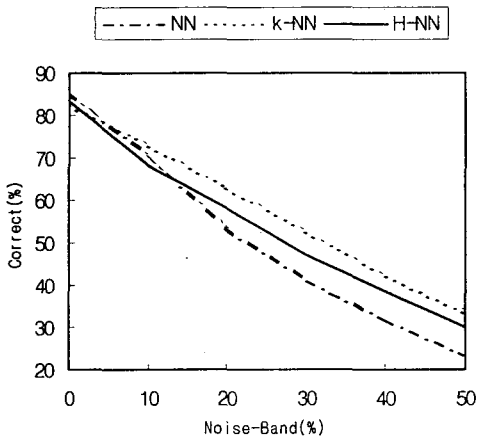
4.1.1 분류성능 비교

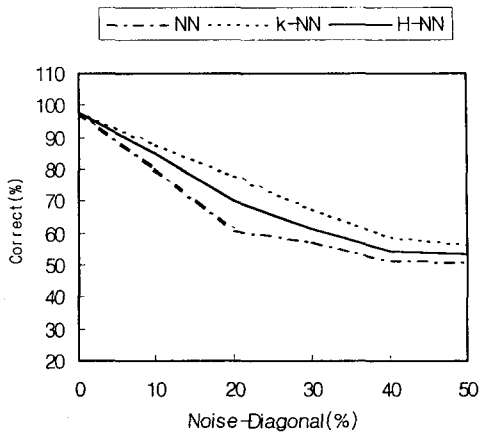
여기에서는 노이즈가 포함되지 않은 8개의 데이터 셋에서 NN, k-NN, H-NN의 분류성능을 비교한다. 그림 9의 결과에서 Band, Diagonal, Quadrant, Radial 데이터 셋의 경우 세 방법 모두가 거의 같은 분류 성능을 보이고 있으며, Sinusoidal, Glass 데이터 셋에서는 NN이, Iris, Wine데이터 셋은 k-NN이 가장 우수한 성능을 보이고 있다. 하지만 본 논문에서 제안한 h-NN 학습 알고리즘의 경우 모든 데이터 셋에서 다른 두 가지 방법에 비하여 크게 뒤지지 않는 분류 성능을 보임을 알 수 있다.

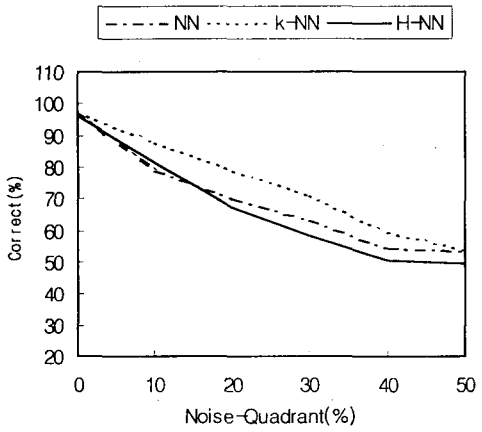


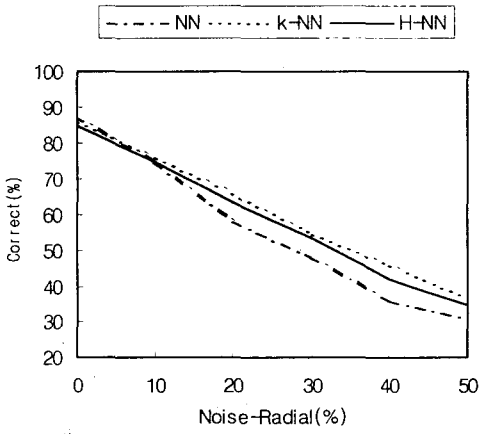
	Band	Diagonal	Quadrant	Radial	Sinusodial	Iris	Glass	Wine
■NN	85.2	97.5	97.4	87.6	76.0	95.6	90.0	90.2
□k-NN	82.5	97.9	97.0	86.1	67.7	100.0	86.7	99.5
□h-NN	83.5	97.8	96.2	84.7	66.1	95.6	84.7	95.5

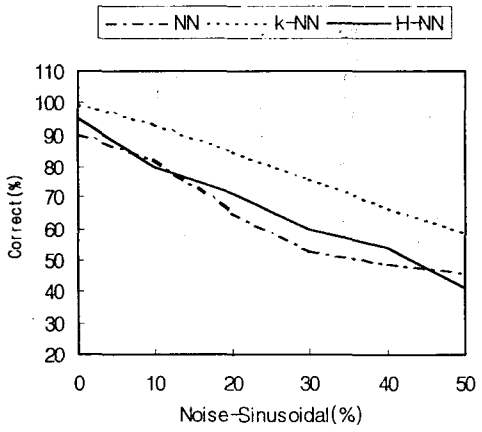
(그림 9) NN, k-NN, h-NN 분류성능 비교 (Noise Free)

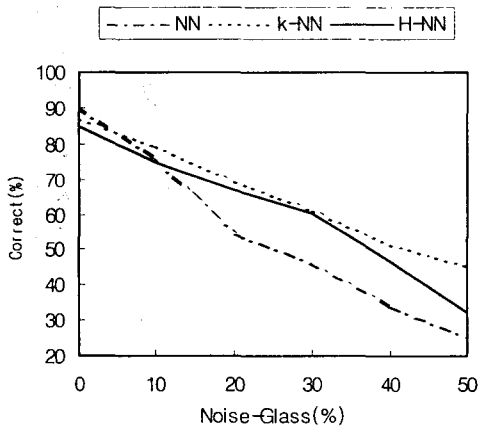

NN
k-NN
H-NN
Correct(%)
Noise-Band(%)

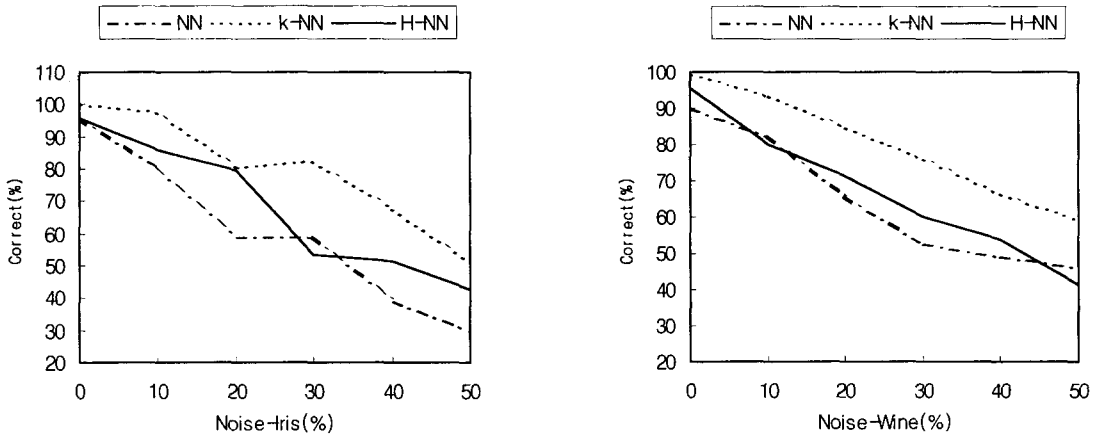

NN
k-NN
H-NN
Correct(%)
Noise-Diagonal(%)


NN
k-NN
H-NN
Correct(%)
Noise-Quadrant(%)


NN
k-NN
H-NN
Correct(%)
Noise-Radial(%)


NN
k-NN
H-NN
Correct(%)
Noise-Sinusoidal(%)


NN
k-NN
H-NN
Correct(%)
Noise-Glass(%)



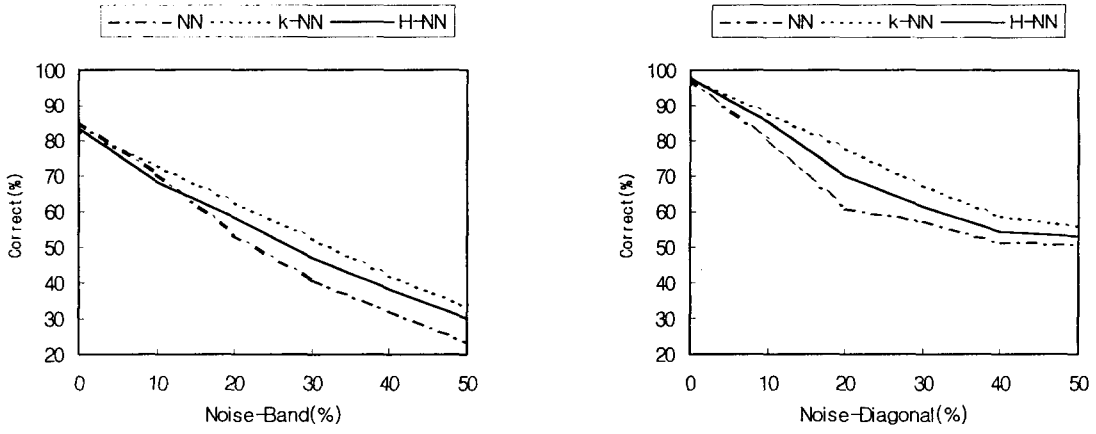
(그림 10) 클래스 노이즈에 따른 NN, k-NN, h-NN 분류성능 변화

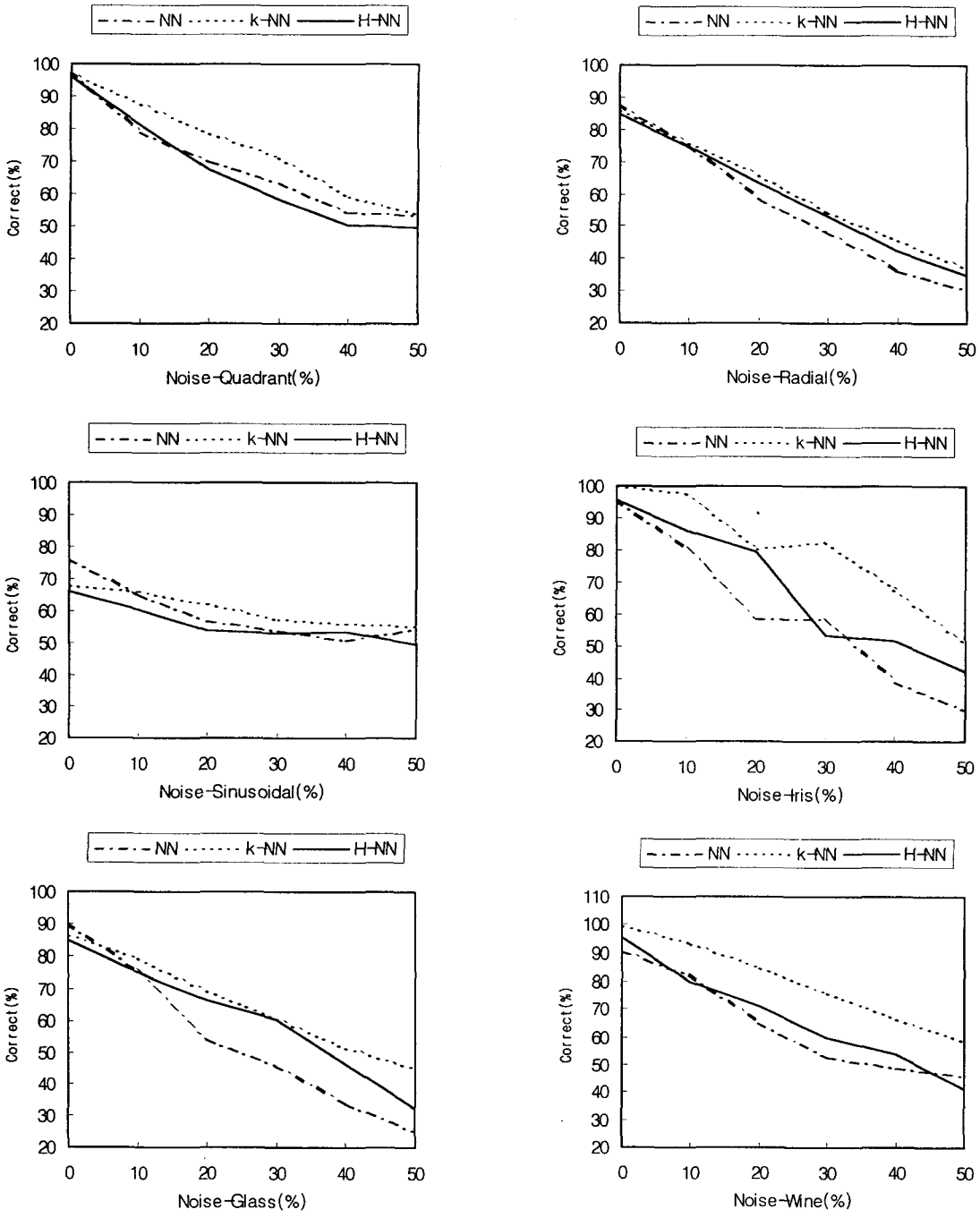
4.1.2 클래스 노이즈 저항 능력

그림10에서 보면 Sinusoidal 데이터 셋을 제외한 4개의 데이터 셋에서 클래스 노이즈가 10%-20%정도 일 경우 H-NN이 NN보다 우수하며, k-NN과는 비슷한 노이즈 저항력이 나타나는 것을 볼 수 있다. 하지만 노이즈의 양이 증가 할 수록 NN보다는 우수하지만 k-NN에 비해서는 성능이 떨어지는 것을 볼 수 있는데, 이것은 H-NN알고리즘에서 노이즈로 판단하는 기준이 가장 가까운 두개의 학습패턴의 클래스만을 비교하기 때문이다. 즉, 학습패턴에 포함된 노이즈들이 1개 이상 밀집되어 있을 경우 오분류를 하게되는 이유 때문이다.

4.1.3 특징노이즈 저항 능력

그림11의 특징노이즈에 대한 실험 결과에서도





(그림 11) 특징 노이즈에 따른 NN, k-NN, h-NN의 분류성능 변화

앞에서 실험한 클래스노이즈의 경우와 비슷한 결과를 보이고 있다. 위의 결과에서 보는 것처럼 본 논문에서 제시한 h-NN알고리즘의 경우 Quadrant 합성데이터 셋을 제외한 7개 데이터 셋에서 NN보다는 우수한 노이즈 저항 능력을 보이고 있다. 특히 Radial, Glass데이터 셋에서는 최적화된 k값을 사용한 k-NN분류기에 와 비교하여 거의 비슷한 노이즈 저항 능력을 보이고 있다.

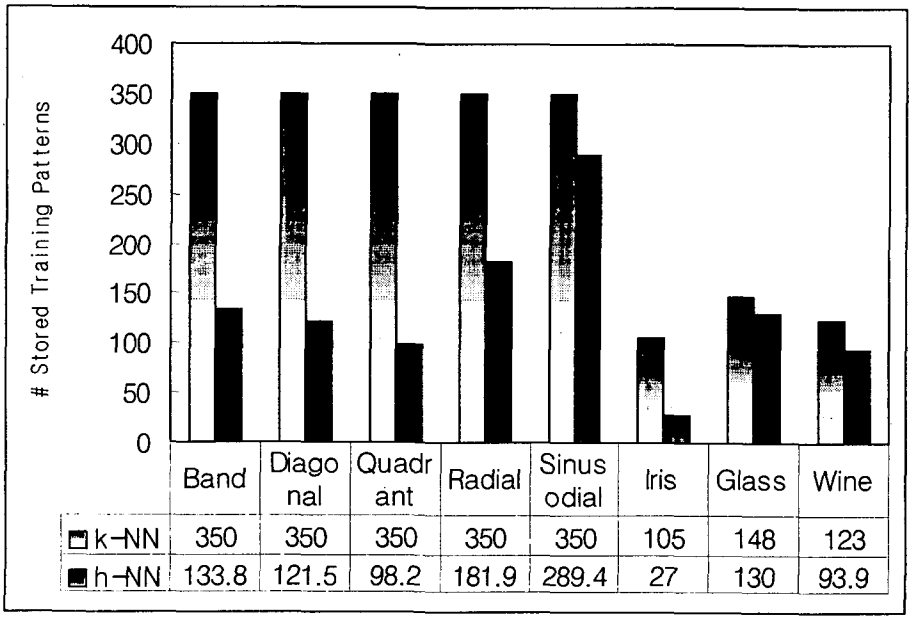
4.2 메모리 사용효율 비교

여기에서는 h-NN의 메모리 사용량을 k-NN 분류기와 비교하여 측정하였으며, h-NN 의 메모리 사용량을 측정하기 위하여 8개의 데이터 셋에 대하여 각각 25회 반복 실험 한 후 평균값을 이용하였다. 학습패턴의 추출은 위의 실험과 동일한 방법을 사용하였으며, 아래의 실험 결과에서처럼 h-NN의 경우 전체 데이터 셋을 모두 저장하는 NN, k-NN분류기에 비하여 충분히 작은 메모리 공간만을 사용하는 것을 볼 수 있다.

그림12의 결과에서 보듯 Iris, Quadrant 데이터의 경우 h-NN알고리즘은 k-NN과 비교하여 약 25%정도의 메모리만을 필요로 하고 있으며, Diagonal, Band의 경우는 35%정도의 공간만을 사용하고 있다. 또한 Radial데이터 셋은 50%의 공간을, 나머지 3개의 데이터 셋에서는 약 80% 정도의 공간만을 사용하고 있다. 그림12에 기록된 메모리 사용량을 나타내는 수치는 메모리에 저장된 학습패턴의 수이다.

4.3 학습을 포함한 전체 소요시간 비교

여기에서는 NN, k-NN, h-NN에서의 학습을



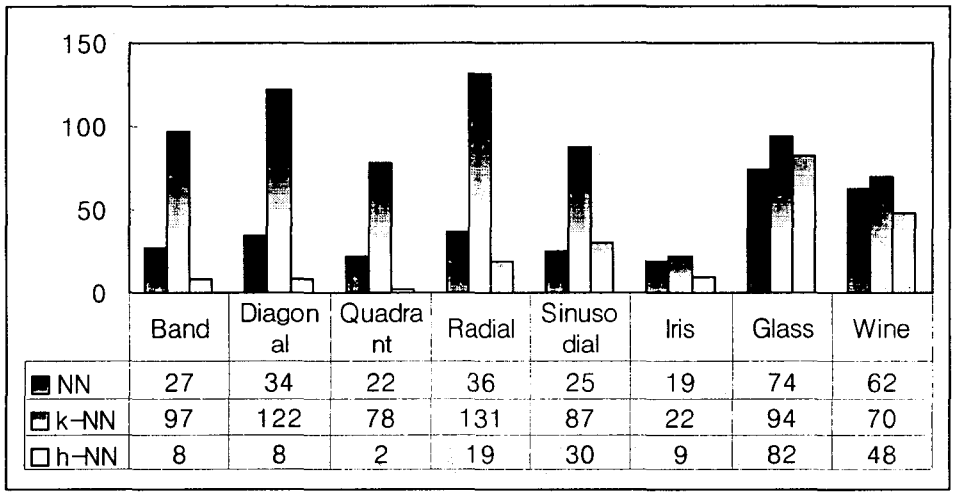
	Band	Diagonal	Quadrant	Radial	Sinusodial	Iris	Glass	Wine
k-NN	350	350	350	350	350	105	148	123
h-NN	133.8	121.5	98.2	181.9	289.4	27	130	93.9

(그림 12) k-NN과 h-NN의 메모리 사용량 비교

포함하여 전체 데이터 셋의 분류에 소요되는 시간을 실시간으로 측정하여 비교하였다. 이 실험에서는 5개의 합성데이터 셋에 대해서는 25회 반복실험에 걸린 시간을 사용하였으며, 3개의 실험데이터 셋에 대해서는 500회 반복실험에 걸린 시간을 사용하였다. 그 이유는 3개의 실험데이터 셋의 경우 합성데이터에 비하여 분류 속도가 현저히 빠르기 때문에 25회 반복으로는 측정이 어렵기 때문이다.

V. 결론

본 논문에서는 메모리기반학습 알고리즘 중 k-NN분류기의 성능 분석을 통하여 k-NN 분류기가 가지는 문제점과 개선점을 제시하였다. 본 논문에서 제안한 h-NN은 기존의 k-NN의 문제점을 개선한 것으로 분류속도와 메모리 사용효율 면에서 만족할 만한 결과를 보이고 있다. 뿐만 아니라 k-NN의 경우 최적의 k값의 결정을 위해서는 점진적 학습이 불가능해진다는 결점을, 패턴 분류시에 가변적으로 변하는 k값을 사용하는 방법으로 점진적 학습이 가능한 알고리즘이다. h-NN학습 알고리즘은 노이즈에 민감히 반응하지 않는다.



	Band	Diagonal	Quadrant	Radial	Sinusodial	Iris	Glass	Wine
NN	27	34	22	36	25	19	74	62
k-NN	97	122	78	131	87	22	94	70
h-NN	8	8	2	19	30	9	82	48

(그림 13) NN, k-NN, h-NN의 분류속도 비교

그림 13이 결과에서 보면 Glass데이터 셋을 제외한 모든 데이터 셋에서 h-NN이 월등한 분류속도를 보이는데 이것은 4.2의 실험에서 보이는 것처럼 적은 수의 학습패턴만을 메모리에 저장하기 때문이다. 하지만 Glass 데이터 셋에서 h-NN이 NN에 비하여 느린 분류속도를 보이는 것은, 전체 학습패턴의 약 85%만을 저장한 후 패턴 분류시 가변 k값을 적용하는 시간으로 인한 것이다. 위의 실험에서 기록된 수치는 초(Second)를 사용한 것이다.

k-NN에서는 메모리 사용량을 최소화 하기 위하여 패턴공간을 같은 크기의 초월평면으로 분할하여 패턴평균법을 적용하였다. 하지만 여

기에서 사용한 패턴평균법은 하나의 초월평면 내부에 모두 같은 클래스의 패턴이 존재할 때만 적용하였고, 그렇지 않은 경우는 모든 학습패턴을 모두 저장하는 방법을 사용하였다. 따라서 클래스 경계면이 복잡한 경우는 NN과 비슷한 수의 학습패턴을 저장하게 된다. 이러한 문제점은 현재 모두 같은 크기의 초월평면으로 분할하는 방법을 개선하는 방법에 대한 연구가 필요하다고 사료된다.

참고 문헌

Dietrich Wettscherck, et al., A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithm, Artificial Intelligence Review Journal, 1996.

David W. 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

Dietrich Wettschereck, Weighted kNN vs. Majority kNN A recommendation, GNRCIT, 1995.

Salzberg S. L, A Nearest hyperrectangle learning method, Machine Learning, no. 1, pp. 251-276, 1991

Thomas G. Dietterich, A study of distance based machine learning, Ph.D. Thesis, Computer Science Dept., OSU, 1994.

김상귀, 이형일, 윤충화, A study on the optimization of binary decision tree, 명지대학교 산업기술 연구소 논문지, vol. 16, pp. 104 112, 1997.

이형일, 정태선, 윤충화, A New weighting method for EACH System, 정보과학회 춘계 학술발표 논문집, vol. 25, no. 1, pp. 288 290, 1998.

심범식, 정태선, 윤충화, Analysis on the Effect of the Number of Seeds in Nearest Hyperrectangle Learning, 정보처리학회 춘계 학술발표 논문집, 1998.

A Study on the Storage Requirement and Incremental Learning of the k-NN Classifier

Hyeong-il Lee*, Chung-Hwa Yoon**

Abstract

The MBR (Memory Based Reasoning) is a supervised learning method that utilizes the distances among the input and trained patterns in its classification, and is also called a distance based learning algorithm. The MBR is based on the k-NN classifier, in which learning is performed by simply storing training patterns in the memory without any further processing.

This paper proposes a new learning algorithm which is more efficient than the traditional k-NN classifier and has incremental learning capability. Furthermore, our proposed algorithm is insensitive to noisy patterns, and guarantees more efficient memory usage.

* Div. of Computer Science, Kimpo College
** Dept. of Computer Science, Myongji University