

# Chapter 05 K-근접이웃 분류기

## [학습목표]

이 장에서는 확률밀도함수를 추정하는 대신 학습 데이터를 바로 활용하여 분류를 수행하는 비모수적 접근법의 대표적인 예로 K-근접이웃 분류기에 대하여 알아본다. 또한 4장에서 소개한 베이지안 분류기와 비교를 통해서 결정경계의 특성 및 문제점에 대하여 알아본다. 마지막으로 매트랩을 이용하여 간단한 K-근접이웃 분류기를 설계해본다.

## 5.1 K-근접이웃 분류기

### 5.1.1 비모수적 확률밀도 추정과 최근접이웃 분류기

### 5.1.2 K-근접이웃 (K-NN) 분류기

### 5.1.3 K-NN 분류기와 비모수적 밀도추정

## 5.2 K-근접이웃 분류기의 특성

### 5.2.1 K-NN 분류기의 결정경계

### 5.2.2 K-NN 분류기의 설계 고려사항

## 5.3 매트랩을 이용한 K-NN 분류기 실험

## 연습문제

## 5 K-근접이웃 분류기

### 5.1 K-근접이웃 분류기

4장에서 알아본 베이지안 분류기를 개발할 때에는 데이터의 확률분포함수를 미리 가정하고 이를 추정하여 분류에 활용하는 모수적 접근 방법을 취하는 것이 일반적이다. 그러나 앞서 살펴본 바와 같이 미리 가정된 확률모델이 주어진 데이터 분포에 적합하지 않은 경우에는 좋은 성능을 기대하기 힘들다 (그 예로 4.4장의 실험에 사용된 데이터를 들 수 있겠다). 이러한 문제에 대한 대안으로 비모수적 밀도추정에 기반을 둔 K-근접이웃 분류기를 생각해 볼 수 있다. 이절에서는 먼저 가장 간단하면서 직관적인 최근접이웃 분류기에 대하여 알아보고, 이를 일반화 한 K-근접이웃 분류기에 대하여 알아본다.

#### 5.1.1 비모수적 확률밀도 추정과 최근접이웃 분류기

3장에서 살펴본 비모수적 밀도 추정 방법에 대한 일반식 [식 3-17]을 바탕으로 여기서는 부피  $V$ 를 고정하는 대신 데이터의 수를  $K$ 로 고정하고 부피를  $\mathbf{x}$ 의 함수로 생각하였다. 이에 따라 각 클래스  $C_i$ 에 대한 확률밀도함수를 정의하면 다음과 같이 나타낼 수 있다.

$$p(\mathbf{x} | C_i) = \frac{1}{V_i(\mathbf{x})} \frac{K}{N} \quad [\text{식 5-1}]$$

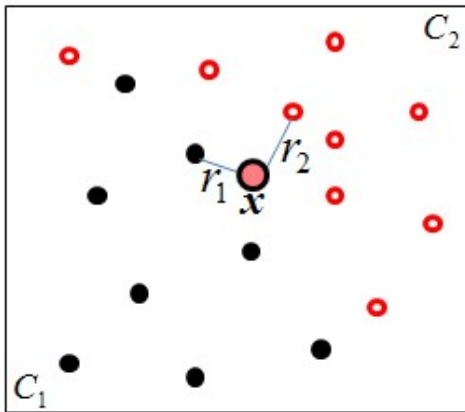
이 때 부피  $V_i(\mathbf{x})$ 는 클래스  $C_i$ 에 속하는 데이터들 중에서  $\mathbf{x}$ 에서  $K$ 번째로 가까운 데이터  $\mathbf{x}_K^i$ 까지의 거리  $d(\mathbf{x}, \mathbf{x}_K^i)$ 를 반경  $r_i(\mathbf{x})$ 으로 하는 초구로 정의한다. 이 정의에 따라 부피는 다음과 같이 고쳐 쓸 수 있다.

$$p(\mathbf{x} | C_i) = \frac{1}{v_n} \frac{1}{(r_i(\mathbf{x}))^n} \frac{K}{N} \quad [\text{식 5-2}]$$

여기서  $v_n$ 은  $n$ 차원 입력공간상의 단위구(unit hypersphere)의 체적이다.

이 확률밀도함수를 이용하여 베이즈 분류기의 결정규칙 [식 4-5]에 따라 분류를 수행한다면, 각 클래스에 대한 선형확률이 모두 동일하다고 가정하고 각 클래스에 무관한 값들을 제외하면 다음과 같은 결정 규칙을 얻을 수 있다.

$$y(\mathbf{x}) = \operatorname{argmin}_i \{r_i(\mathbf{x})\} \quad [\text{식 5-3}]$$



$$y(x) = \arg \min \{r_1, r_2\} = 1$$

[그림 5-1] 최근접이웃 분류기의 할당 규칙

특히  $K=1$ 인 경우에는 주어진 데이터  $x$ 에 대해 각 클래스별로 가장 가까운 데이터까지의 거리를 계산하여, 그 거리값이 가장 작은 클래스에 할당하는 분류 방법을 생각해 볼 수 있다 ([그림 5-1] 참조). 그런데 이 과정을 좀 더 간단히 하면 클래스와 상관없이 모든 데이터들 중 가장 작은 거리값을 가지는 데이터를 찾아 그 데이터가 속하는 클래스로 할당하는 방법도 [식 5-3]과 동일한 결과를 가져올 수 있다. [그림 5-1]에서 가장 작은 거리값  $r_1$ 을 가지는 데이터는  $C_1$ 에 속하므로  $x$ 도  $C_1$ 으로 분류하게 된다. 이러한 결정규칙에 의한 분류 방법을 <최근접이웃 분류기 (Nearest Neighbor Classifier)>라고 한다. 최근접이웃 분류기의 결정규칙을 식으로 다시 표현하면 다음과 같이 정리할 수 있다.

$$\begin{aligned} \mathbf{x}_{\min} &= \operatorname{argmin}_{\mathbf{x}_i \in X} \{d(\mathbf{x}, \mathbf{x}_i)\} \\ y(\mathbf{x}) &= y(\mathbf{x}_{\min}) \end{aligned} \quad [\text{식 5-4}]$$

아래에 최근접이웃 분류기의 수행단계를 간단히 정리하였다.

#### [최근접이웃 분류기의 수행 단계]

① 주어진 데이터  $x$ 와 모든 학습 데이터  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 과의 거리를 계산한다.

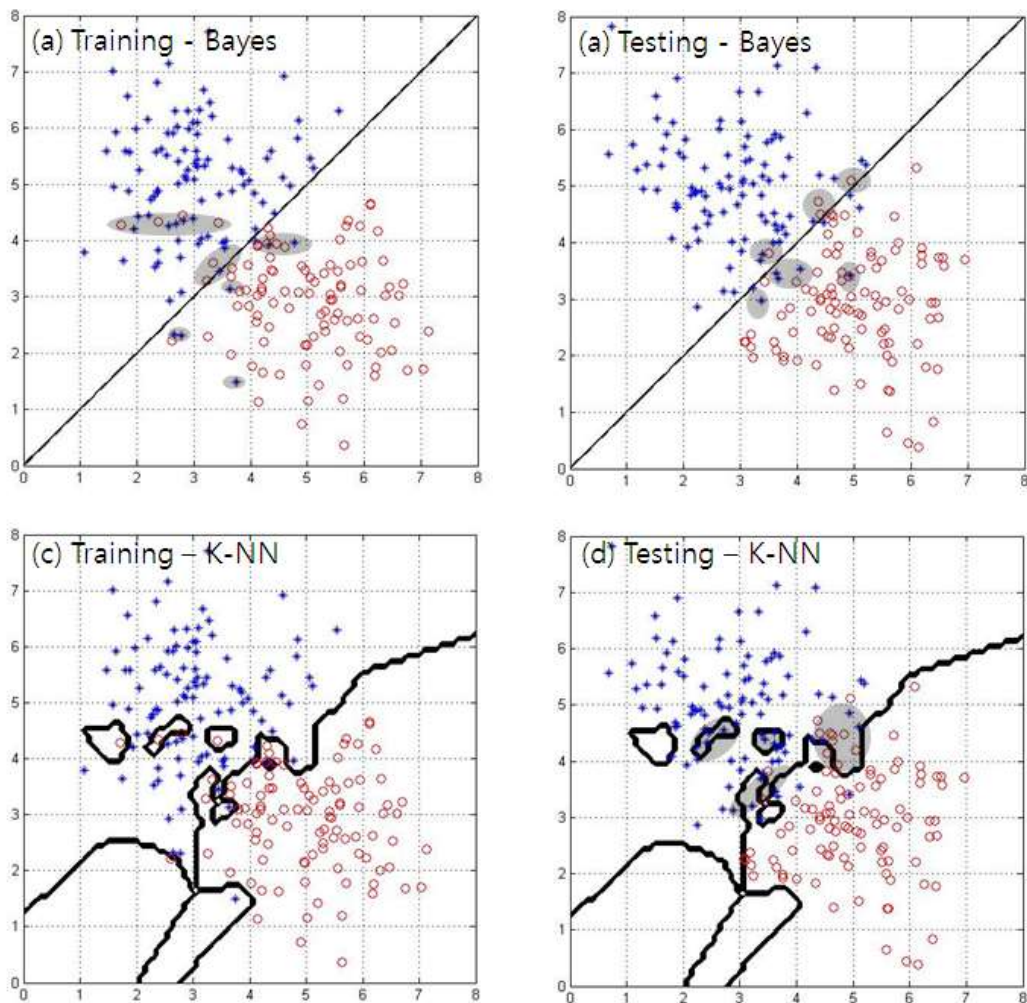
② 거리가 가장 가까운 데이터를 찾아  $\mathbf{x}_{\min}$  으로 둔다.

$$\mathbf{x}_{\min} = \operatorname{argmin}_{\mathbf{x}_i \in X} \{d(\mathbf{x}, \mathbf{x}_i)\}$$

③  $\mathbf{x}_{\min}$ 이 속하는 클래스에 할당한다. 즉  $y(\mathbf{x}_{\min})$ 과 같은 값을 가지도록  $y(\mathbf{x})$ 를 결정한다.

## 5.1.2 K-근접이웃 분류기

최근접이웃 분류기의 결정규칙은 결국 새롭게 주어진 데이터의 클래스 라벨이 가장 가까운 데이터의 클래스 라벨과 동일하다고 생각하여 그것에 할당하는 것이다. 그러나 이는 단 하나의 데이터만으로 분류를 결정짓게 되는 것으로, 데이터에 포함된 잡음에 매우 민감한 문제를 가지게 된다. 특히 결정경계 부근에 있는 데이터들의 경우에는 오분류율이 매우 높아진다.



[그림 5-2] 최근접이웃 분류기와 베이즈 분류기의 결정경계

[그림 5-2]에 최근접이웃 분류기의 결정경계를 베이즈 분류기와 비교하여 나타내었다. 그림에서 사용된 데이터의 두 클래스는 평균만 다르고 공분산은 같은 가우시안 분포로 정의되었다. 따라서 최소오류확률 결정경계는 가우시안을 가정한 베이즈 분류기로부터 [그림 5-2a]에 나타난 것처럼 직선 형태를 가진다. 학습데이터로부터 얻은 결정경계는 학습데이터에 대해서 어느 정도 오류를 포함하나, [그림 5-2b]와같이 테스트 데이터집합에 적용해도 학습오류

와 유사한 정도의 오류를 나타낸다. 그러나 최근접이웃 분류기의 경우는 [그림 5-2c]와 같이 복잡한 비선형 결정경계를 찾아, 학습데이터를 모두 완벽하게 분류 가능하다. 그러나 이렇게 학습데이터에 과다적합된 결정경계를 테스트 데이터에 대하여 적용하면 그 분류성능은 가우시안 베이스 분류기에 비해 저조한 결과를 보이게 된다. (그림에 오분류된 데이터가 존재하는 영역을 음영으로 표시해 두었다.) 이 데이터의 경우, 베이스 분류기의 테스트오차는 6.5%이고, 최근접이웃 분류기의 테스트오차는 13.5%로, 베이스 분류기의 성능이 월등히 좋았다. 이와 같은 최근접이웃 분류기의 문제를 해결하기 위하여 [식 5-4]를 확장한 K-근접이웃 분류기를 생각해 볼 수 있다. <K-근접이웃(K-Nearest Neighbor, K-NN) 분류기>는 주어진 데이터로부터 거리가 가까운 순서대로  $K$ 개의 데이터를 찾은 후, 그 중 가장 많은 수의 데이터가 속한 클래스로 할당하는 방법이다. K-NN 분류기의 결정규칙을 단계별로 나누어 기술하면 다음과 같다.

#### [K-근접이웃 분류기의 수행 단계]

- ① 주어진 데이터  $\mathbf{x}$ 와 모든 학습 데이터  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 과의 거리를 계산한다.
- ② 거리가 가장 가까운 것부터 순서대로  $K$ 개의 데이터를 찾아 후보 집합  $N(\mathbf{x})$ 를 만든다.  

$$N(\mathbf{x}) = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^K\}$$
- ③ 후보 집합의 각 원소가 어떤 클래스에 속하는지 그 라벨값  $y(\mathbf{x}^1), y(\mathbf{x}^2), \dots, y(\mathbf{x}^K)$ 을 찾는다.
- ④ 찾아진 라벨 값 중 가장 많은 빈도수를 차지하는 클래스를 찾아  $\mathbf{x}$ 를 그 클래스에 할당한다.

### 5.1.3 K-NN 분류기와 비모수적 밀도추정

앞 절에서 살펴본 K-NN 분류기의 분류 방식은, 기본적으로  $K=1$ 인 경우에는 최근접이웃 분류기와 동일한 처리를 수행함을 쉽게 알 수 있다. 그러나 그 유도과정에 있어서는 앞서 최근접이웃 분류기를 유도하기 위하여 사용된 밀도함수 추정식 [식 5-2]를 그대로 사용한 것과는 차이가 있음에 유의해야 한다. 여기서는 K-근접이웃 분류기가 비모수적 확률밀도함수 추정과 어떤 관계가 있는지에 대하여 설명한다.

데이터  $\mathbf{x}$ 가 주어지면, 먼저 클래스와 상관없이 전체 데이터 집합을 대상으로 데이터의 수  $K$ 를 고정하여 그 수만큼의 데이터를 포함하는 영역의 부피  $V_K(\mathbf{x})$ 를 찾는다.

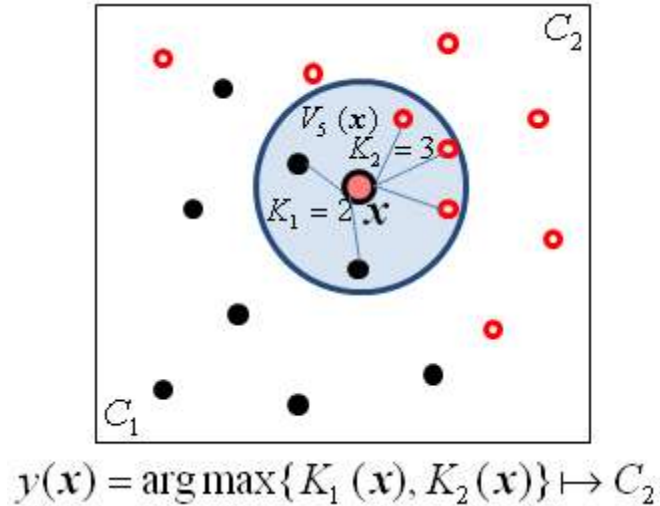
이제 각 클래스별 데이터 분포함수를 추정하기 위해서는 고정된 부피  $V_K(\mathbf{x})$ 내에서 각 클래스별로 몇 개의 데이터가 포함되어 있는지를 찾아 그 값을 각각  $K_1, K_2, \dots, K_M$ 로 놓는다. 이를 이용하여 각 클래스별 분포함수의 추정식을 쓰면 다음과 같다.

$$p(\mathbf{x} | C_i) = \frac{1}{V_K(\mathbf{x})} \frac{K_i(\mathbf{x})}{N} \quad [\text{식 5-5}]$$

이 확률밀도함수를 이용하여 베이즈 분류기의 결정규칙 [식 4-5]에 따라 분류를 수행한다면,  $V_K(\mathbf{x})$ 와  $N$ 은 클래스  $C_i$ 에 무관하므로 제외할 수 있다. 또 각 클래스에 대한 선형확률  $p(C_i)$ 가 모두 동일하다고 가정하여 제외하면 다음과 같은 간단한 결정규칙을 얻을 수 있다.

$$y(\mathbf{x}) = \operatorname{argmax}_i \{K_i(\mathbf{x})\} \quad [\text{식 5-6}]$$

이 결정규칙은 앞서 제시한 K-근접이웃 분류기의 분류 방법과 동일한 결과를 나타낼 수 있게 알 수 있다. [그림 5-3]에서는  $K=5$ 일 때 K-근접이웃 분류기에 의한 분류 방법을 그림으로 나타내었다.



[그림 5-3] K-근접이웃 분류기의 할당규칙 ( $K=5$ 인 경우)

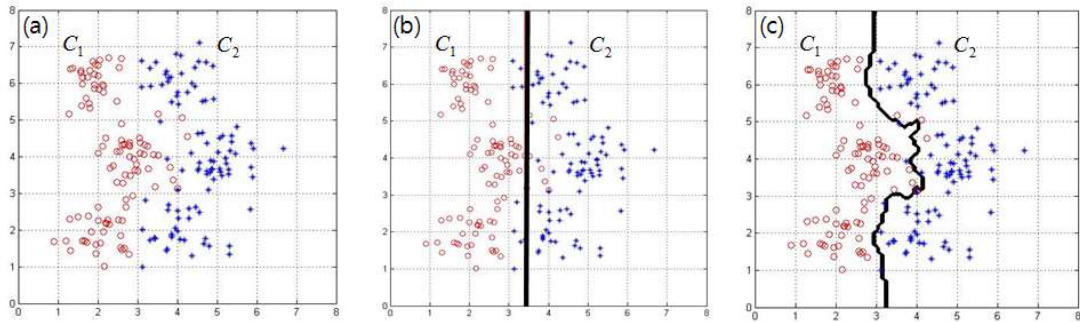
## 5.2 K-근접이웃 분류기의 특성

### 5.2.1 K-NN 분류기의 결정경계

K-근접이웃 분류기는 4장에서 살펴본 최소거리 분류기에 비해 그 성능이 일반적으로 더 좋다고 알려져 있으며, 따라서 보다 널리 사용되고 있다. 이 절에서는 K-근접이웃 분류기의 결정경계를 살펴봄으로써 그 이유를 알아보고 분류기의 특성에 대한 이해를 높이하고자 한다.

[그림 5-2]에서 살펴본 바와 같이 최근접이웃 분류기나 K-근접이웃 분류기는 가우시안 베이즈 분류기에 비하여 매우 비선형적인 결정경계를 가진다. 또한 비모수적 방법에 기반을 두므로, 데이터의 분포 형태에 따라 성능이 크게 좌우되지 않는다. 예를 들어, [그림 5-4a]의

경우와 같이 데이터 분포가 복잡한 비선형 구조를 가지는 경우에 가우시안 분포를 가정한 베이지 분류기를 사용하게 되면 [그림 5-4b]와 같이 적절한 결정경계를 찾는데 실패하여 성능의 저하를 초래하지만, K-근접이웃 분류기의 경우 [그림 5-4c]와 같이 어느 정도 정확한 결정경계를 찾아주고 있음을 알 수 있다.



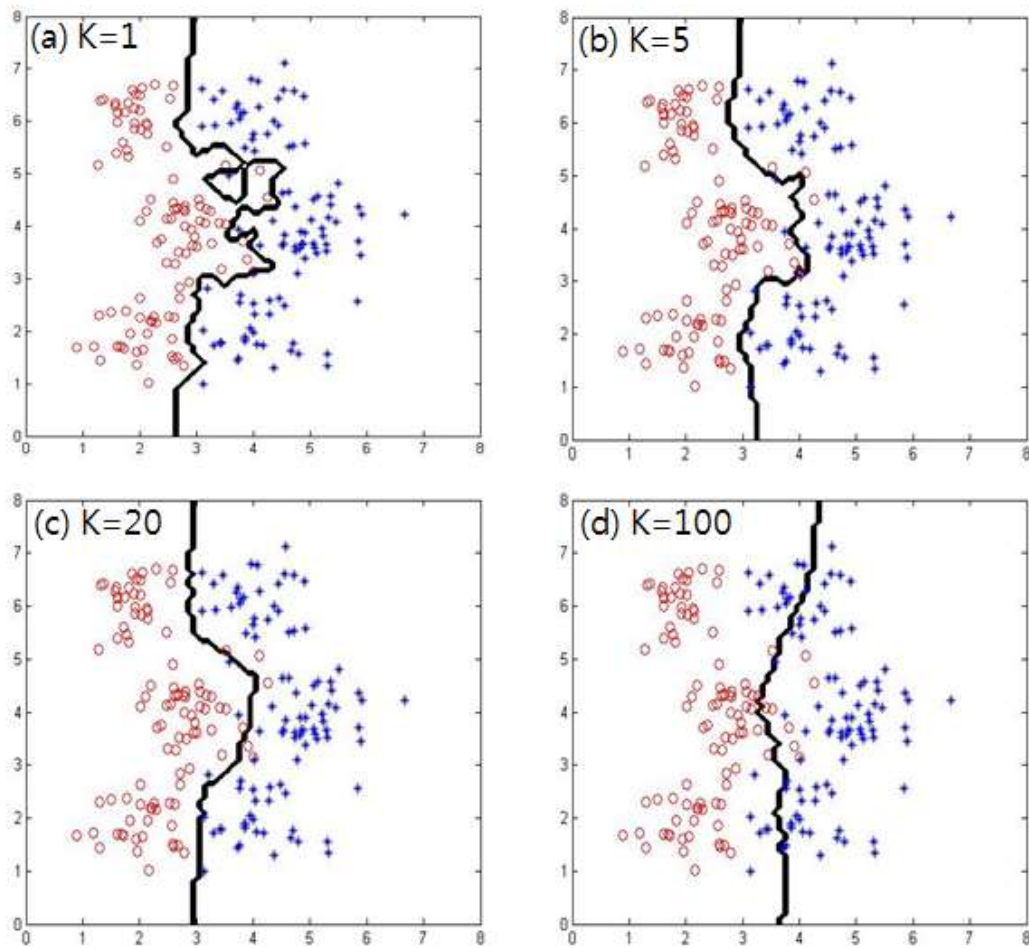
[그림 5-4] 가우시안 베이지 분류기(b)와 K-근접이웃 분류기(c)의 비교.

그럼에도 불구하고, 모수적 확률밀도함수 추정에 의한 베이지 분류기가 K-근접이웃 분류기에 비하여 선호되는 경우도 존재한다. 가우시안 베이지 분류기의 경우, 확률밀도함수는 평균과 표준편차에만 의존하므로 학습 데이터를 이용하여 일단 파라미터를 추정하고 나면, 이후 분류 과정에서는 더 이상 학습 데이터는 필요로 하지 않고 단지 평균과 표준편차만을 이용한 결정규칙을 사용하게 된다. 그러나 K-근접이웃 분류기의 경우 새로운 데이터가 주어질 때 마다 학습 데이터 집합 전체와의 거리 계산을 통하여 K개의 이웃 데이터를 선정해 주어야 한다. 따라서 계산 시간이 오래 걸릴 뿐 아니라, 항상 학습 데이터를 저장하고 있어야만 분류가 가능하다. 특히 데이터의 수가 늘어나면 그만큼 분류기의 정확도는 높아질 수 있겠지만 그에 비례하여 계산량과 메모리도 함께 증가하는 문제점을 가진다. 특히 최근 들어 많이 연구되고 있는 임베디드 시스템에 분류기를 탑재하고자 할 때에는 K-근접이웃 분류기는 많은 제약을 받게 된다. 따라서 응용시스템 설계자들은 각 분류 방법들이 가지는 특성을 잘 파악하고 목적에 맞는 분류기를 선택할 수 있는 안목을 갖는 것이 매우 중요하다.

## 5.2.2 K-NN 분류기의 설계 고려사항

K-근접이웃 분류기를 실제 문제에 응용하고자 할 때에는 먼저  $K$ 값을 어떻게 정해야 할 지 생각해 보아야 한다.  $K=1$ 인 경우는 바로 이웃한 데이터에만 의존하여 클래스가 결정되므로 앞서 살펴본 바와 같이 노이즈에 민감한 결과를 초래한다. 반면  $K$ 값이 지나치게 커지면 주어진 데이터 주변의 영역만을 중심으로 분류가 수행되는 것이 아니라, 전체 데이터 영역에서 각 클래스가 차지하는 비율, 즉 선형확률에 의존하여 분류를 수행하는 결과를 초래하게 된다.





[그림 5-5] K값에 따른 결정경계의 변화

[그림 5-5]에  $K$ 값에 따른 결정경계의 변화를 나타내었다. 그림에서  $K=1$ 인 경우에는 노이즈에 과다적합되어 지나치게 복잡한 결정경계를 이루게 됨을 볼 수 있다. 이 결정경계를 사용하는 경우, 학습데이터에 대해서는 좋은 결과를 보일 것이나 새롭게 주어지는 테스트데이터에 대해서는 그 성능이 떨어질 것이다.  $K=100$ 인 경우는 데이터 집합이 가지는 결정경계에서 멀어지면서 [그림 5-4b]에서 찾은 선형결정경계와 그 형태가 유사해 짐을 알 수 있다. 전체적으로 볼 때  $K=20$ 인 경우에 가장 자연스러운 결정경계를 찾고 있다.

이와 같이  $K$ 값을 선택하는 것은 분류기의 성능에 직접적인 영향을 주게 되므로, 적절한  $K$ 값을 선택할 수 있는 방법이 필요하다. 한 가지 방법으로는 3장에서 설명한 방법과 같이 전체 데이터의 수에 의존하여 결정하는 방법을 생각해 볼 수 있을 것이다. 앞 절에서 살펴본 바와 같이  $K$ 값은 전체 데이터의 확률분포  $p(\mathbf{x})$ 를 추정하기 위해 고정하는 값으로 볼 수 있으므로, 전체 데이터 수가  $N$ 이라고 할 때  $K=\sqrt{N}$ 을 사용할 수 있다. 그러나 이것은 어디까지나 여러 가지 가능한 선택의 한 예이며, 최적의  $K$ 값은 데이터의 수 뿐 아니라 데이터의 분포 특성에도 의존하므로 주어진 문제(혹은 학습 데이터 집합)에 의존적이라고 할 수 있다. 데이터를 활용하여 적절한  $K$ 값을 선택하기 위해서는 학습 데이터, 혹은 검증 데이터를 활용하여 분류를 수행해 봄으로써 가장 좋은 성능을 주는 값을 찾는 방법도 생각해 볼



수 있다.

이밖에 K-근접이웃 분류기의 설계 시 고려해야할 사항으로 거리함수가 있다. K-근접이웃 분류기는 주어진 데이터와 학습 데이터들 간의 거리를 바탕으로 이웃 데이터를 찾는 것에서 시작한다. 따라서 어떤 거리함수를 이용하느냐에 따라 선택되는 이웃이 달라질 수 있고, 결과적으로 분류 성능에 직접적인 영향을 주게 된다. 선택 가능한 거리함수로는, 가장 기본적인 유클리디안 거리함수를 비롯하여 4절에서 소개한 정규화된 유클리디안 거리나 마할라노비스 거리와 같이 데이터 집합의 분산을 고려한 함수를 생각해 볼 수 있다. 이 밖에도 벡터 연산으로부터 정의된 내적함수나 벡터간의 각도차를 사용하는 코사인 거리함수, 벡터의 1차 노름 등도 널리 사용되고 있다.

[표 5-1] 자주 쓰이는 거리함수

함수이름	함수식
2차 노름 (유클리디안 거리)	$d_E(\mathbf{x}, \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ _2 = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
1차 노름	$d_1(\mathbf{x}, \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ _1 = \sum_{i=1}^n  x_i - y_i $
p차 노름	$d_p(\mathbf{x}, \mathbf{y}) = \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^p}$
내적	$d_{IN}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^n x_i y_i$
코사인거리	$d_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\ \mathbf{x}\  \ \mathbf{y}\ }$
정규화된 유클리디안 거리	$d_{NE}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n \frac{(x_i - y_i)^2}{\sigma_i^2}} \quad (\sigma_i^2 \text{는 데이터의 분산})$
마할라노비스 거리	$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})} \quad (\Sigma \text{는 데이터의 공분산행렬})$

이상과 같이 자주 쓰이는 거리함수들에 대한 정의를 [표 5-1]에 정리하였다. 2차노름은 가장 일반적으로 사용되는 유클리디안 거리로, 두 벡터가 나타내는 공간상의 두 점을 잇는 직선 거리의 길이에 해당한다. 1차노름의 경우 각 좌표값들의 차이를 모두 더한 값이며, 1차, 2차 노름의 일반화된 형태가 p차 노름으로, p의 값을 적절히 조정하여 다양한 거리함수를 만들 수 있다. 이밖에 주로 사용되는 거리함수로 내적과 코사인 거리를 들 수 있는데, 내적의 경우 두 벡터의 방향이 비슷할수록 그 값이 커진다. 따라서 이는 엄밀히 말해 거리함수라기보다 유사한 정도를 나타내는 유사도 함수라고 보는 것이 적합하다. 또한, 내적값은 각 벡터들의 크기가 클수록 그 값이 커지게 되어, 두 벡터가 유사하지 않더라도 각각의 크기가 크면 큰 값을 가지게 된다. 따라서 내적을 거리함수(혹은 유사도함수)로 사용하기 위해서는 각 벡터들이 정규화되어 있다는 가정이 먼저 만족되어야 할 것이다. 이러한 문제를 해결하여, 각 벡터간의 각도차이만으로 거리를 평가하는 함수가 코사인함수이다. 식에서 알 수 있듯이, 두 벡터의 크기를 분모에 나누어 주어 정규화시킨 값을 사용하고 있다. 정규화된 유클리디안 거리와 마할라노비스 거리는 4장에서 살펴본 바와 같이, 각 좌표축 방향으로의 분산

의 차이를 고려하여 반영해 준 거리함수이다.

4장에서도 잠깐 살펴본 바와 같이, 이들 거리함수 중 어떤 것이 가장 적절할 것인지는 주어진 데이터의 특성에 의존하는 것이므로 일반적인 상황에서 평가하는 것은 불가능하다. 따라서  $K$ 값을 선택할 때와 마찬가지로 학습 데이터나 검증 데이터를 이용하여 직접 비교해보는 방법 등을 통해 주어진 목적에 맞는 함수를 선택해야 할 것이다.

### 5.3 매트랩을 이용한 K-NN 분류기 실험

지금까지 살펴본 내용을 바탕으로 간단한 2차원 데이터 분류를 위한 K-근접이웃 분류기를 만들고 성능을 평가해 보겠다. 사용한 데이터는 4.4절의 베이스 분류기 설계 시에 사용한 것과 동일하다. 즉, 모두 3개의 클래스로 이루어져 있고, 학습 데이터로는 각 클래스별 100개씩 모두 300개가 사용되었다([그림 4-7] 참조).

3장의 베이스 방법과는 달리 K-근접이웃 방법에서는 각 클래스의 확률밀도함수를 추정하는 별도의 단계 없이 바로 5.1.2절에 제시한 결정규칙에 따라 분류를 수행할 수 있다.  $K$ 값에 따른 성능을 비교하기 위하여  $K=1, 5, 10, 50$ 인 경우에 각각 분류를 수행하였다.

[프로그램 5-1]에 매트랩으로 구현된 분류기 코드를 나타내었다. 먼저 학습 데이터를 불러들여 현재 주어진 데이터와의 거리를 각각 계산한 후, 거리값들을 정렬하여 가까운 이웃  $K$ 개를 찾는다. 찾아진 이웃 데이터들의 클래스 라벨들을 이용한 투표 과정을 거쳐서 가장 많은 표를 얻은 클래스에 주어진 데이터를 할당하게 된다. 이렇게 분류 결과가 결정되면 그 결과와 원래 데이터를 생산한 클래스가 일치하는지 점검하여 오분류율을 계산한다.

프로그램 5-1 K-Nearest Neighbor Classifier	
K-근접이웃 분류기를 이용하여 데이터를 분류하고 학습 오차를 출력	
001	load dataCh4_7 %학습 데이터 로드
002	X=[X1:X2:X3];
003	Etrain=0;
004	N = size(X,1);
005	for i=1:N
006	x=X(i,:); % 각 데이터에 대해 분류시작
007	for j=1:N % 모든 데이터와의 거리 계산
008	d(j,1)=norm(x-X(j,:));
009	end
010	[sx,si]=sort(d); % 거리 순으로 정렬
011	K=5; c=zeros(3,1); % K=5로 정함
012	for j=1:K % 이웃한 K개 데이터의 라벨을
013	if (si(j)<=100) c(1)=c(1)+1; end % 점검하여 투표수행
014	if (si(j)>200) c(3)=c(3)+1; end
015	if ((si(j)>100) & (si(j)<=200)) c(2)=c(2)+1; end
016	end
017	[maxv, maxi]=max(c); % 최대 투표수를 받은 클래스로 할당
018	if (maxi~=floor((i-1)/100)+1)) % 원래 클래스 라벨과 다른면
019	Etrain(1,1) = Etrain(1,1)+1; % 오류데이터의 개수를 증가
020	end
021	end
022	Error_rate = Etrain/N %오분류율 출력

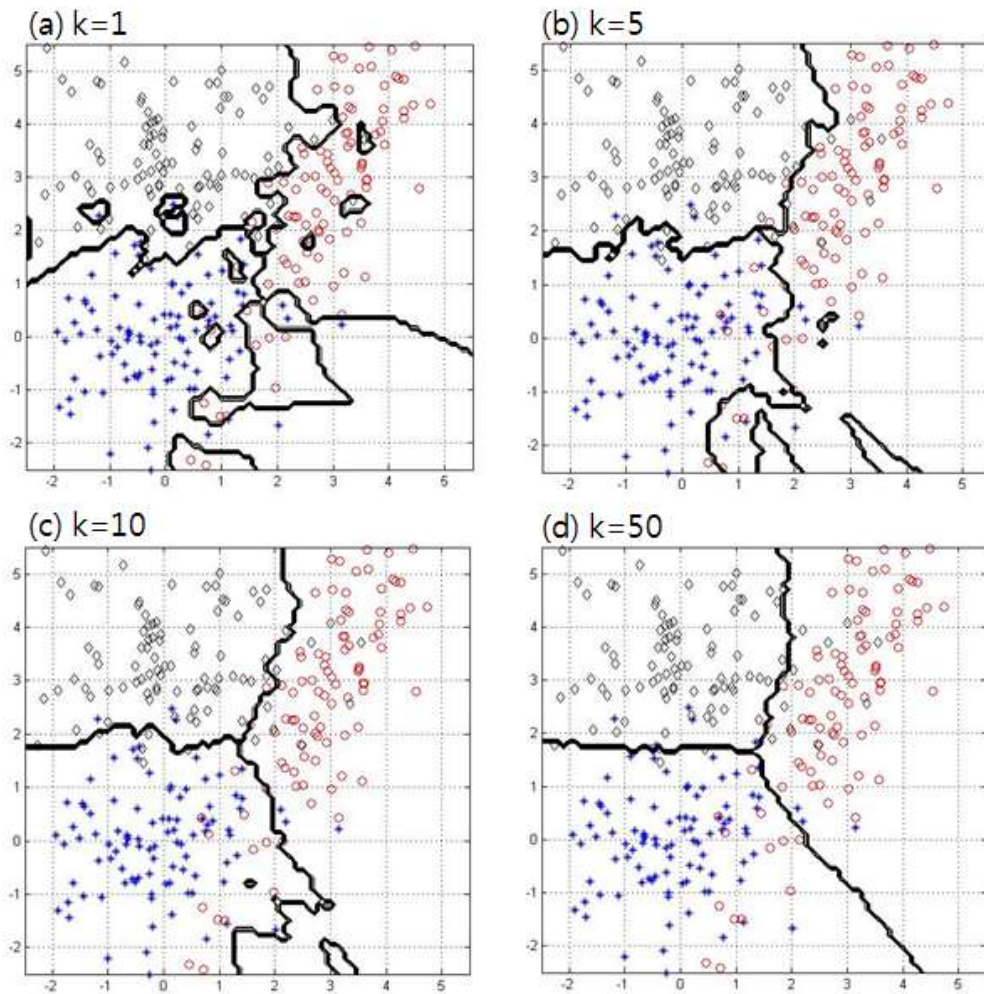
학습 데이터에 대하여 [프로그램 5-1]을 수행한 결과와 테스트 데이터에 대해서 동일한 처리 과정을 수행한 결과를 [표 5-2]에 정리하였다. 이 문제의 경우 데이터 분포가 비교적 단순하므로  $K$ 값에 따른 오차의 차이는 크지 않았다. 그러나 전체적으로 학습데이터에 대한 변화양상과 테스트 데이터에 대한 변화 양상이 다를 수 있다. 학습데이터의 경우,  $K$ 값이 커짐에 따라 오분류율이 커짐에 반해, 테스트 데이터의 경우, 작은  $K$ 값에서는 오분류율이 높아지고, 또 반대로  $K$ 값이 지나치게 커져도 다시 오분류율이 높아져서,  $K=10$ 일 때 가장 작은 오분류율을 보였다.

[표 5-2] K-근접이웃 분류기의 분류결과

	$K=1$	$K=5$	$K=10$	$K=50$
오분류율(학습오차)	0%	9.67%	11.67%	11.67%
오분류율(테스트오차)	14.74%	11.11%	10.46%	10.52%

$K$ 값에 따른 차이를 보다 확실히 알아보기 위하여 [그림 5-6]에  $K$ 값에 따른 결정경계를 나타내었다. [프로그램 5-2]에는 결정경계를 그리는 매트랩 코드를 나타내었다. 결정경계는, 관심대상이 되는 입력공간에 있는 모든 점들에 대하여 분류를 수행하고, 그 결과 얻어진 클래스 라벨을 이용하여 contour함수를 이용하여 등고선을 그리는 방식을 사용하였다. [그림 6-5]에서 확인할 수 있듯이, 테스트 오차로부터 예상할 수 있는 것과 유사하게  $K=1$ 인 경우는 학습데이터에 과다적합된 복잡한 결정경계를 찾았으며,  $K=50$ 인 경우에는 반대로 직선에 가까운 형태의 결정경계를 찾았다. 결정경계와 테스트오차의 결과로부터  $K=10$ 이 가장 적당할 것으로 판단할 수 있다.

프로그램 5-2 Decision boundary of K-NN Classifier	
K-근접이웃 분류기의 결정경계 그리기	
001	load dataCh4_7 %학습 데이터 로드
002	X=[X1;X2;X3];
003	[x,y]=meshgrid([-2.5:0.1:5.5],[-2.5:0.1:5.5]); %입력공간전체의 데이터준비
004	XY=[x(:), y(:)];
005	figure(1); hold on
006	plot(X1(:,1), X1(:,2), '*'); %학습데이터 그리기
007	plot(X2(:,1), X2(:,2), 'ro');
008	plot(X3(:,1), X3(:,2), 'kd');
009	for i=1:size(XY,1) %전체 입력공간의 데이터에 대해
010	xt=XY(i,:); %클래스 라벨을 결정
011	for j=1:size(X,1)
012	d(j,1)=norm(xt-X(j,:));
013	end
014	[sx,si]=sort(d);
015	K=1; c=zeros(3,1);
016	for j=1:K
017	if (si(j)<=100) c(1)=c(1)+1; end
018	if (si(j)>200) c(3)=c(3)+1; end
019	if ((si(j)>100) & (si(j)<=200)) c(2)=c(2)+1; end
020	end
021	[maxv, maxi]=max(c);
022	rx1(i,1)=maxi;
023	end
024	rx1=reshape(rx1,size(x));
025	contour(x, y,rx1); %클래스 라벨에 따른 등고선 그리기
026	axis([-2.5 5.5 -2.5 5.5]); grid on



[그림 5-6] 여러 가지 K값에 대한 결정경계

마지막으로, [그림 5-6]의 결과와 [그림 4-8]의 결과를 비교하면 베이지 분류기와 K-NN분류기의 차이점을 잘 알 수 있다. 베이지 분류기의 경우 좋은 성능을 얻기 위해서는 주어진 데이터 집합에 적합한 확률모델을 세우는 것이 중요하며, K-NN 분류기의 경우는 적절한 K의 값을 찾아주는 것이 중요하다. K-NN 분류기의 경우, K값에 따라 다양한 비선형 결정경계를 찾을 수 있어 간단하면서도 분류 성능이 비교적 좋은 분류기로 알려져 있다. 그러나 앞서도 잠깐 언급하였듯이 분류를 위해서는 학습데이터를 모두 기억하고 있어야 하는 문제점을 안고 있다. 베이지 분류기의 경우, 특히 일반적으로 많이 사용하는 모든 클래스가 동일한 공분산 행렬을 가지는 가우시안 분포를 모델로 사용한 경우에는 그 계산이 매우 간단하다는 장점을 가지고 있으나, 결정경계가 선형이 되므로 복잡한 결정경계를 가지는 문제에서는 적합하지 못하다. 그러나 최근 들어 많이 연구되고 있는 고차원 영상데이터의 경우, 입력 공간의 차원에 비해 분류 대상이 되는 데이터의 수가 적어 데이터 집합이 입력공간상에 희박(sparse)하게 분포하는 양상을 나타내므로, 선형결정경계 만으로도 어느 정도 좋은 성능을 기대할 수 있는 경우도 많다. 따라서 패턴인식 설계자들은 두 방법의 장단점과 특성을 충분히 이해하여 주어진 문제에 적절한 방법을 선택하는 능력을 갖추는 것이 필요할 것이다.

## 연습 문제

1. 연습문제 4.3에서 사용한 데이터에 대해 K-근접이웃 방법을 이용하여 분류를 수행하고자 한다. 다음 절차에 따라 수행하시오.
  - (1) 연습문제 4.3(1)과 같이 데이터 집합을 생성한다.
  - (2) (1)의 데이터 집합과는 별도로 각 그룹별로 10개씩의 데이터를 따로 생성한다.
  - (3) (2)에서 생성된 데이터를 K-근접이웃 방법으로 분류해 본다. 이때 K의 값을 3, 5, 10으로 변화시키면서 수행하고, 그 결과를 비교하시오.
  - (4) (3)에서 사용한 K 값에 대해, 찾아지는 결정경계를 그래프로 그려보시오.
2. 1번에서 사용된 데이터에 대해, [표 5-1]에서 제시된 여러 가지 거리함수들을 적용하여 분류를 수행해 보고 그 결과를 비교해 보시오.
3. 베이지 분류기의 경우 클래스의 선형확률이나 베이지 위험 등을 고려하여 그 결정경계를 조정하는 것이 가능하였다. K-근접이웃 분류기에도 같은 사항이 고려될 수 있는지 생각해 보시오.
4. 베이지 분류기와 K-NN분류기의 장단점을 비교하시오.
5. 베이지 분류기는 기본적으로 각 클래스에 대한 확률밀도 함수를 추정하여, 이를 바탕으로 데이터의 후험확률을 계산하고, 이를 분류의 판단 기준으로 사용하는 방법이다. 따라서 확률밀도 함수를 반드시 모수적 방법으로 추정할 필요는 없으며, 이 장에서 배운 비모수적 방법으로 확률밀도 함수를 추정한 후, 이를 이용하여 베이지 분류를 수행하는 것도 생각해 볼 수 있다. 이 경우 하나의 데이터를 분류하기 위한 구체적인 처리 단계가 어떻게 될 지 생각해 보시오.
6. K-근접이웃 분류기에서 적절한 K값을 선택하는 방법에 대하여 생각해 보시오.

## 참고 자료

이 장에서 배운 K-NN분류기는 패턴인식 분야에서 가장 기본적인 방법론으로, 모든 패턴인식 교과서에서는 빠짐없이 소개되어 있다. 그러나 많은 경우에는 단순한 처리 단계를 소개하는데 그치고 있는 반면, 통계적 접근법을 기본으로 하는 교과서들에서는 이 책에서 소개한 바와 같이 비모수적 밀도함수 추정이론을 바탕으로 K-NN을 설명하고 있다. 이러한 접근법을 취하는 대표적 교재로는 1장에서 소개한 [Fukunaga 90]과 [Duda, Hart & Stork 01]을 들 수 있겠다.

또한 5.2절에서 잠깐 언급한 거리함수도 K-NN 분류기의 성능을 좌우하는 중요한 요소가 된다. 거리함수와 관련된 연구는 오래전부터 여러 분야에서 수행되어 왔다. [표 5-1]에 소개된 거리함수들 중,  $p$ 차 노름을 비롯한 벡터들의 노름에 관한 내용은 선형대수과목의 교과서와 아래의 [Horn & Johnson, 1990] 등에서 참조할 수 있다. 또한 정규화된 유클리디안 거리와 마할라노비스 거리와 같이 데이터의 분포를 고려한 통계적 거리와 관련하여서는 [McLachlan 1992]에 소개되어 있으며, 보다 깊은 내용에 대하여서는 [Banerjee et al. 2005]를 참조하기 바란다. 이밖에 실제 패턴인식 문제에서 다양한 거리함수들의 성능을 비교한 연구들은 다수 존재하는데, 한 예로 얼굴인식에서 거리함수들의 성능을 비교한 논문으로 [Vytutas 2004] 등이 있다.

[Horn & Johnson, 1990] R. A. Horn and C. R. Johnson. "Norms for Vectors and Matrices," Ch. 5 in Matrix Analysis. Cambridge, England: Cambridge University Press, 1990.

[McLachlan 1992] G. J. McLachlan. Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience, 1992

[Banerjee et al. 2005] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh. "Clustering with Bregman divergences," Journal of Machine Learning Research, 6, 1705–1749, 2005.

[Vytutas 2004] P. Vytutas. "Distance measures for PCA-based face recognition," Pattern Recognition Letters, 25(6), 711-724, 2004.