Chapter 01
패턴인식과 기계학습의 개요

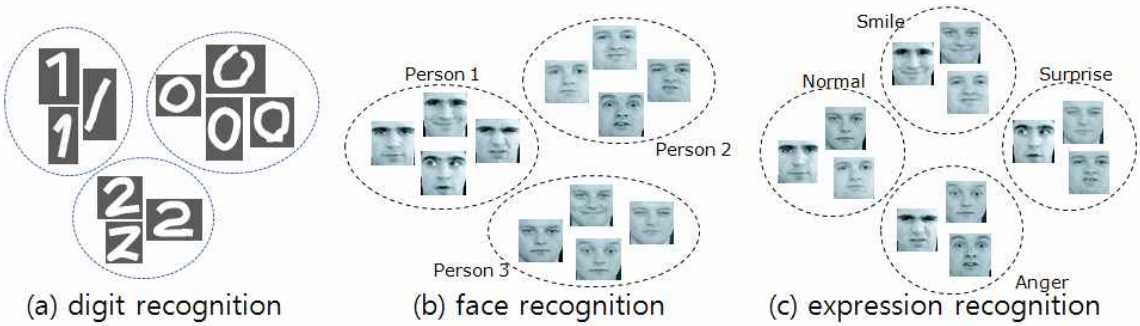
[학습목표]
이 장에서는 패턴인식과 기계학습의 기본 개념을 소개하고, 패턴인식의 처리과정과 패턴인식기의 개발과정을 살펴봄으로써 이 책에서 배우게 될 전체적인 내용을 파악할 수 있도록 준비한다. 또한 패턴인식과 기계학습에서 주로 논의되는 주제와 관련된 용어와 개념을 학습한다.

1.1 패턴인식과 기계학습
1.2 패턴인식의 처리과정
1.3 패턴인식의 기본 요소
1.3.1 데이터와 데이터분포
1.3.2 특징과 특징추출
1.3.3 분류와 결정경계
1.3.4 분류율과 오차
1.4 패턴인식과 관련된 개념들
1.4.1 분류와 군집화
1.4.2 교사학습과 비교사학습
1.4.3 분류기 복잡도와 과다적합
1.5 패턴인식의 응용
연습문제
참고자료

1. 패턴인식과 기계학습의 개요

1.1. 패턴인식과 기계학습

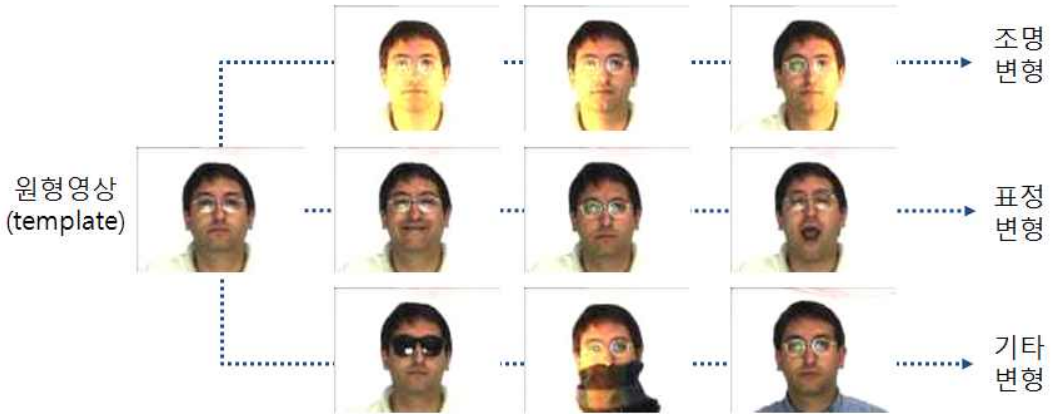
패턴인식이란, 주어진 데이터의 집합에 대해 입력 값을 바탕으로 특정 기준에 따라 여러 개의 그룹으로 분류(인식)하는 것을 말한다. 예를 들어 [그림 1-1a]에서와 같이 숫자 영상들이 주어졌을 때, 각 숫자를 하나의 패턴으로 간주하고, 같은 숫자에 해당하는 영상들의 그룹(클래스)으로 주어진 데이터를 분류할 수 있을 것이다. 이와 같은 문제를 "숫자인식"이라고 부르며, 이는 패턴인식의 하나의 응용 분야에 해당한다. 마찬가지로, [그림 1-1b, c]에서와 같이 얼굴 영상에 대해서도 패턴인식 문제를 생각해 볼 수 있다. 주어진 영상이 누구의 얼굴을 나타내는지, 즉 누구의 얼굴 패턴인지를 인식하는 "얼굴인식"이라는 패턴인식 문제 (b)를 생각할 수 있다. 또한 같은 데이터에 대해서 각 영상이 어떤 표정을 가진 얼굴 영상인지를 인식하는 "표정인식" 문제 (c)도 생각해 볼 수 있다. 이와 같이 주어진 입력 데이터들을 어떤 기준에 따라 몇 개의 그룹으로 나누고, 각 데이터가 어떤 그룹에 해당하는지를 판별하는 것을 패턴인식이라고 한다.



[그림 1-1] 패턴인식의 예: (a) 숫자인식 (b) 얼굴인식 (c) 표정인식
(얼굴영상출처: PICS database (http://pics.psych.stir.ac.uk/))

패턴인식 문제를 해결하기 위해서는 어떤 접근 방법을 취해야 할까? 누구나 쉽게 생각해 볼 수 있는 두 가지 접근 방법에 대하여 간단한 예를 들어 알아보자. [그림 1-1a]와 같은 숫자인식 문제를 생각해 보자. 먼저 각각의 패턴(숫자)이 구조적 관점에서 어떤 공통된 특성을 가지고 있는 지를 생각해보자. 숫자 "1"의 경우는 수직에 가까운 직선 형태를 나타내고, 숫자 "8"의 경우는 작은 원이 아래위로 붙어 있는 형태를 나타낸다. 이와 같이 주어진 패턴의 구조적 특성을 먼저 분석하여 정의해 두고, 각 데이터의 구조적 특성에 따라서 패턴을 분류할 수 있을 것이다. 그러나 이러한 구조적 분석에 의한 패턴인식에는 한계가 있다. 예를 들어 얼굴인식의 경우, 각 사람의 얼굴 특징을 일일이 분석하여 정의하는 것은 매우 어려울 뿐 아니라 특징 자체도 명확하게 정의하기 힘들다.
패턴인식의 또 다른 접근 방법으로서, 구조적 특징을 분석하는 대신 단순히 각 패턴의 원형(template)을 저장해 두고 이것과 주어진 데이터 간의 거리를 계산하여 가까운 패턴의 클래

스로 분류를 하는 방법을 생각해 볼 수 있다. 이러한 접근법을 <템플릿 매칭 (template matching)>이라고 한다. 그러나 이러한 템플릿 매칭 기법도 한계가 있다. 숫자인식의 문제를 템플릿 매칭 기법으로 해결하고자 한다면, 각 숫자별로 그 원형패턴을 저장해 두어야 할 것이다. 그런데, 단순한 "1"이라는 숫자에 대해서도 필체, 선의 굵기, 기울어진 각도 등 다양한 변형이 존재한다는 것은 쉽게 생각할 수 있다. 좀 더 복잡한 "4"나 "9"의 경우에는 그 변형이 훨씬 다양하다. 이러한 다양한 변형들을 하나의 원형패턴으로 대표한다는 것은 무리가 있음을 쉽게 알 수 있다. 얼굴 데이터의 경우에는 표정, 포즈, 조명, 크기, 머리모양, 안경이나 모자의 착용 여부 등 셀 수 없을 정도의 많은 변형이 존재하여 문제는 더욱 어려워진다. [그림 1-2]는 얼굴영상에 대한 다양한 변형의 예를 나타내고 있다. 왼쪽의 원형 영상에 대하여, 오른쪽의 첫 번째 행에는 조명의 영향에 의한 다양한 변형을 나타내었고, 두 번째 행에는 표정에 의한 변형을, 그리고 세 번째 행에는 폐색(occlusion)에 의한 변형을 나타내었다.



[그림 1-2] 얼굴영상 패턴의 원형과 변형들
(출처: AR-face database)

이상에서 살펴본 바와 같이, 구조적 특징에 의한 패턴의 정의 및 인식 방법과 템플릿 매칭 방법은 가장 기본적인 패턴인식 기법이라고 할 수 있다. 그러나 이러한 방법을 실세계 데이터에 적용함에 있어서의 가장 큰 문제점은 패턴들이 가지고 있는 다양한 <변형(variation, transformation)>이다. 사실 패턴인식 문제의 핵심은 이러한 변형을 효과적으로 표현하고 구분하는 보다 정교한 방법을 설계하는데 있다고 볼 수 있다. 이러한 패턴의 변형에 따른 문제를 해결하기 위하여 기계학습 분야의 다양한 방법론들이 적용될 수 있다.

<기계학습 (machine learning)>이란, 인간이 가지고 있는 고유의 지능적 기능 중 하나인 학습 능력을 기계를 통해 구현하는 방법들에 대한 연구이다. 주어진 데이터들을 분석하여 그로부터 일반적인 규칙이나 새로운 지식을 자동적으로 추출해 내는 방법론들을 개발한다. 예를 들어 숫자인식의 경우, 구조적 특징이나 특정한 원형패턴을 정의하는 대신 일련의 데이터 집합을 패턴인식기에 제공하면, 패턴인식기가 자동적으로 데이터를 분석하여 각 숫자 패턴에 대한 정보를 추출하여 저장한다. 이렇게 저장된 정보는 새롭게 주어진 데이터들이 어떤 패턴의 클래스에 해당하는지를 분류하기 위한 기준으로 사용된다. 이때, 숫자 패턴에 대한 정보를 추출하기 위하여 사용된 데이터의 집합을 <학습 데이터(training data)>라고 부른다. 다양한 변형을 포함한 학습 데이터 집합을 인식기에 제공하면 좋은 인식기는 그러한 변형들을 모두 고려한 정보를 추출하여 저장할 수 있을 것이다.

그렇다면, 자동적으로 데이터를 분석하여 주어진 패턴의 대표적 정보를 추출하는 과정은 구체적으로 어떻게 진행되는가? 이에 대한 답이 이 책 전반에 걸쳐 소개되고 있다고 볼 수 있다. 여기서는 가장 간단한 예를 가지고 설명하겠다. [그림 1-3]에는 여러 가지 변형을 가진 숫자 "5"의 데이터가 나타나 있다. 이러한 변형들을 모두 고려하여 숫자 패턴 "5"의 대표적 정보를 추출하는 가장 기초적인 방법으로 평균 영상을 생각해 볼 수 있다. [그림 1-3]의 가장 왼쪽에 나타난 것과 같이, 주어진 5개 영상의 각 픽셀값들의 평균으로 구성된 영상을 생각해 볼 수 있을 것이다. 특정한 원형패턴을 사용하는 대신 이러한 평균 영상을 사용하면 다양한 변형을 어느 정도 흡수할 수 있을 것이다. 이는 기계학습의 가장 기초적인 기법으로, 데이터의 통계적 정보(평균량)를 이용하여 데이터 집합을 표현하는 것이라고 할 수 있다. 이 밖에도 주어진 학습 데이터 집합의 통계적 특성을 분석하는 것은 기계학습에서 매우 중요하며 이를 통해 인식에 필요한 유용한 정보를 제공할 수 있다.

[그림 1-3] 변형을 가진 숫자 패턴과 평균영상

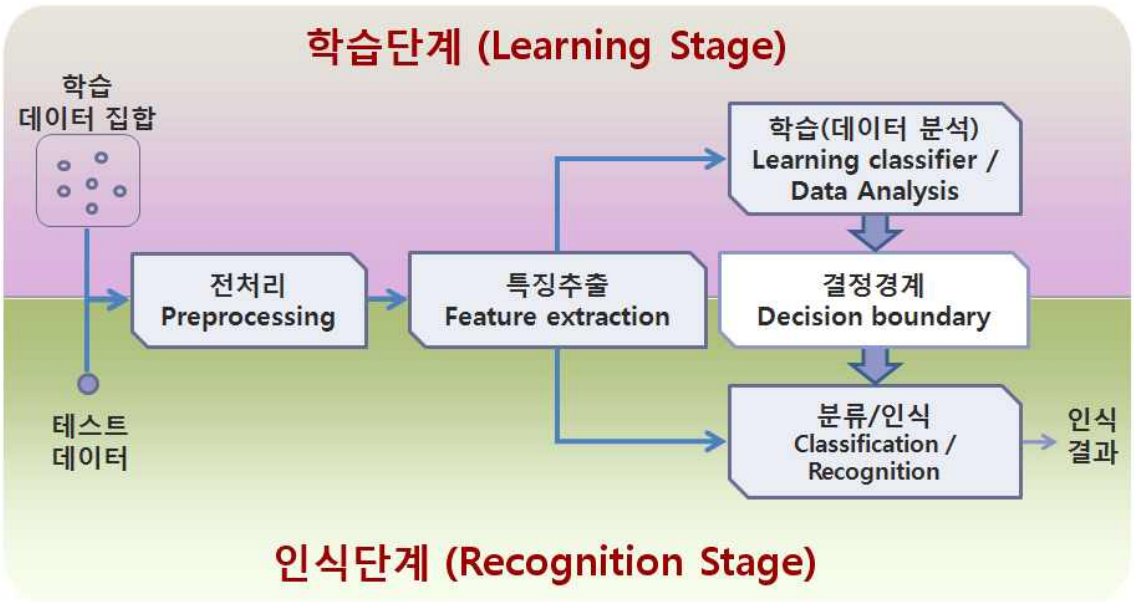
이 책에서는 이러한 통계적 기법을 중심으로 대표적인 패턴인식 기법을 소개할 것이다. 통계적 접근방법은 그 이론적 배경이 탄탄하고, 직관적으로도 이해하기 쉬우며, 구현도 간단하다는 장점이 있다. 또한 통계적 관점에서 데이터를 분석하고 처리하는 과정에 대한 개념이 확립되고 나면, 다른 방법들에 대해서도 같은 관점으로 이해하고 그 특징을 파악할 수 있는 능력을 갖추게 된다. 따라서 먼저 이 책의 전반부(3장~9장)에서는 기본적인 통계적 패턴인식 기법들을 소개하고, 이를 바탕으로 후반부(10장~14장)에서는 최근 많이 사용되고 있는 심화된 기계학습 방법들로 확장함으로써, 패턴인식을 배우는 학습자들로 하여금 여러 패턴인식 기법들에 대한 단편적이고 개별적인 지식이 아니라 패턴인식과 관련된 다양한 방법론들에 대한 전체적인 틀을 형성할 수 있도록 도울 것이다.

1.2 패턴인식의 처리과정

앞 절에서 패턴인식이란 어떤 일을 하는 것이며, 이를 위해서 기계학습이 어떤 도움을 주고 있는지에 대하여 살펴보았다. 이제 패턴인식의 전체적인 처리과정을 살펴봄으로써 좀 더 구체적인 내용으로 들어가 보겠다.

[그림 1-4]에 패턴인식의 전체적인 처리과정을 나타내었다. 앞서 잠깐 언급한 바와 같이, 기계학습 기법을 사용하는 패턴인식에는 크게 두 가지의 처리단계가 존재한다. 먼저 학습단계에서는 주어지는 데이터 집합(학습 데이터)을 이용하여 패턴의 특성을 분석하여 서로 다른 패턴들을 구분하기 위한 핵심정보를 추출한다. 학습이 완료되고 나면 새롭게 주어지는 데이

터(주로 테스트 데이터로 불림)가 어떤 패턴에 해당하는지 분류하고 인식하는 단계가 수행된다. 학습단계는 주로 인식기를 만드는 첫 과정에서 한번만 수행되며, 인식단계는 새로운 데이터가 주어질 때마다 수행된다.



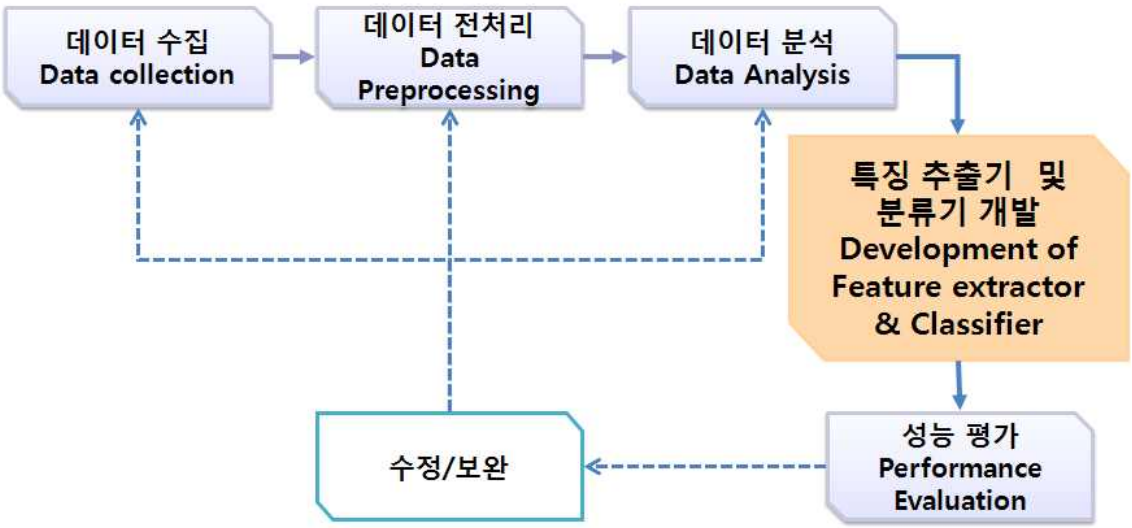
[그림 1-4] 패턴인식의 처리 과정

학습단계를 조금 더 자세히 들여다보면, 먼저 주어진 원래 데이터를 정제하여 인식기가 다루기 쉬운 형태로 변환하는 <전처리 과정 (preprocessing)>이 수행된다. 예를 들어 영상 데이터의 경우에는 크기나 조명을 일정한 범위의 값으로 조정하는 정규화 (normalization)나 한쪽으로 기울어진 영상을 보정하는 각도 보정 등이 전처리 과정에서 이루어진다. 이러한 전처리 과정은 주어진 데이터에 의존하여 그에 최적화된 처리가 수행되어야 하며, 따라서 기계학습의 과정과는 거리가 있다. 예를 들어, 영상 데이터에 대한 전처리와 음성 데이터나 텍스트 데이터에 대한 전처리는 매우 다르다. 따라서 이 책에서는 전처리 과정에 대해서는 깊게 다루지 않을 것이며, 단지 응용문제를 소개할 때 그에 적합한 전처리의 예를 간단히 소개할 것이다. 전처리를 통해 정제된 데이터는 특징추출 단계로 들어간다.

<특징추출(feature extraction)> 단계에서는 데이터의 특성을 분석하여 각 패턴을 표현하는 가장 핵심적인 정보들을 특징으로 찾는다. 데이터를 원래 형태 그대로 사용하지 않고, 핵심적인 특징만을 추출하여 사용함으로써 계산량과 메모리를 절약할 뿐 아니라 데이터에 포함된 불필요한 정보를 제거하는 효과를 얻을 수 있다. 특징추출은 전처리와 마찬가지로 데이터의 특성에 크게 의존하는 단계이기는 하나, 이 책에서는 대부분의 데이터에 공통적으로 적용 가능한 통계적 특징 추출 방법을 소개할 것이다.

특징추출 과정을 거친 학습 데이터 집합은 그 분포 특성을 분석하여 각 패턴들을 구분할 수 있는 기준을 마련하고 이와 관련된 정보를 저장하는 <분류기 학습(learning classifier)> 과정을 거치게 된다. 이 과정에서 학습이 완료되면 패턴들을 분류하는 기준, 즉 결정규칙이 얻어지게 된다. 학습은 패턴인식의 핵심 단계로, 학습 목표와 분류경계에 따라 매우 다양한 방법이 존재한다. 이에 대해서 이 책 전반에 걸쳐서 살펴보게 될 것이다. 이어서 인식단계에서

는 먼저 전처리와 특징추출과정이 학습단계와 동일하게 수행되고, 추출된 특징에 대하여 학습된 분류기를 이용한 인식(분류) 과정을 통해 최종 인식 결과를 얻게 된다.



[그림 1-5] 패턴인식기의 개발 단계

마지막으로 패턴인식기를 개발하는 개발자 입장에서 인식기의 개발 과정에 대해서 간단히 정리해 보겠다. [그림 1-5]에 패턴인식기 개발을 위한 전체적인 과정을 나타내었다. 인식기를 개발하기 위해서는 먼저 학습 데이터를 충분히 수집해야 한다. 다양한 변형을 포함하면서도 불필요한 잡음이 제거된 데이터를 되도록 많이 수집하는 것은 특징추출과 분류기의 성능에도 직접적인 영향을 주게 되므로 매우 중요한 단계라고 할 수 있다. 수집된 데이터에 대해서는 그 특성에 맞는 전처리 알고리즘을 구현하여 정제된 데이터 집합을 생성한다. 이어서 데이터의 분포 특성을 분석하는 단계에서 데이터가 가진 통계적 특성 및 다양한 특성들을 분석하여 적합한 특징추출 기법과 분류기를 선택한 후, 학습을 통하여 분류기준을 결정하게 된다. 이렇게 인식기가 설계되고 나면, 이어서 그 성능을 평가하는 과정을 수행한다. 학습 데이터에 사용된 데이터와는 별도로 새로운 데이터 집합(테스트 데이터)을 구성한 후 인식기를 통해 인식해 봄으로써 개발된 시스템의 성능을 평가하고, 그 결과에 따라서 부족한 부분을 보완하는 과정이 필수적으로 수행되어야 할 것이다.

이상에서 살펴본 바와 같이, 패턴인식기의 설계에 있어서 가장 중요한 부분은 특징추출과 분류기라고 할 수 있다. 이 책에서는 특정한 형태의 데이터에 국한하지 않고, 일반적인 데이터에 대하여 공통적으로 적용할 수 있는 특징추출 및 분류 방법에 대하여 소개할 것이다. 개발자들은 다양한 특징추출 방법과 분류기들의 특성을 고려하여, 주어진 문제에 가장 적합한 방법들을 선별하는 안목을 갖추어야 할 것이다.

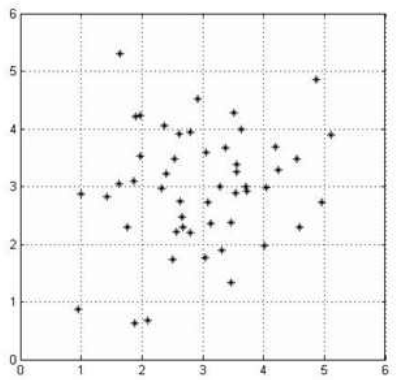
1.3 패턴인식의 기본 요소

이 절에서는 앞에서 살펴본 패턴인식기 설계의 각 단계를 중심으로 패턴인식의 기본 요소에

대하여 소개할 것이다. 특히 이 책은 각 방법들을 소개하면서 동시에 매트랩(Matlab)을 이용한 구현도 함께 살펴보는 형태로 기술되어 있으므로, 이 절에서는 이를 위한 준비 단계로 매트랩을 이용하여 생성된 간단한 2차원 데이터를 이용하여 설명을 진행할 것이다.

1.3.1 데이터와 데이터 분포

실생활에서 다루게 되는 데이터의 형태는 매우 다양하다. 앞 절에서 주로 예를 들어 사용해 온 영상 데이터를 비롯하여 음성 데이터, 최근 들어 그 응용이 활발해진 유전자 정보 데이터, 자연어 문장으로 이루어진 데이터 등 그 형태가 매우 다양하다. 그러나 결과적으로 이러한 데이터들이 컴퓨터에서 다루어지기 위해서는 일련의 수치로 표현되어야 하며, 이를 수학적으로는 벡터로 나타낼 수 있다. 영상 데이터의 경우 n개의 화소로 이루어진 하나의 영상은 n개의 실수값을 가지는 n차원 벡터로 표현할 수 있다. 이 책에서는 n차원의 열벡터 $\boldsymbol{x}=[x_1, x_2, \ldots, x_n]^T$를 이용하여 하나의 데이터를 표현한다.



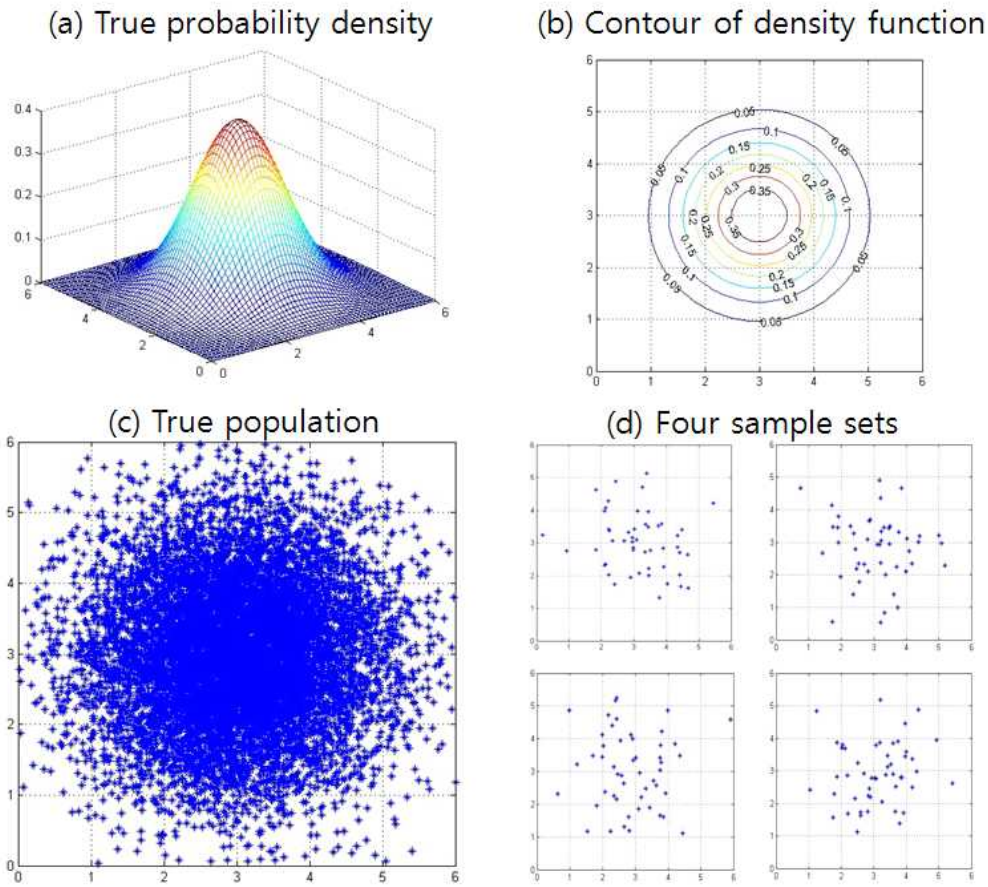
[그림 1-6] 2차원 데이터 집합의 예

데이터가 벡터로 표현되므로, 데이터에 대한 모든 처리도 벡터를 이용한 연산으로 정의된다. 벡터와 관련된 기본적인 내용은 부록 A에 소개해 두었다. 부록 A에서 언급한 바와 같이, n차원 벡터는 n차원의 공간상의 한 점으로 나타낼 수 있다. 이 책에서 다루는 통계적 패턴인식기법은 하나하나의 데이터가 가지는 특성 뿐 아니라, 전체 데이터 집합이 이루는 분포특성을 고려하여 특징을 추출하고 인식을 위한 규칙을 만든다. 이 때 전체 데이터 집합의 분포 특성이란, 해당 공간상에서 점들이 분포되어 있는 모양으로 생각할 수 있다. [그림 1-6]에 50개의 데이터로 구성된 2차원 데이터 집합을 2차원 공간상의 산점도(scatter plot)로 나타내었다. 이 그림으로부터 주어진 데이터가 어떤 범위의 값을 가지며, 가로축(x_1축)의 요소값과 세로축(x_2축)의 요소 값들 사이에는 어떤 관계가 있는지 등의 정보를 얻을 수 있다. 고차원 데이터의 경우에는 시각적으로 표현하는 것이 불가능하나 2차원 데이터에서 적용되는 방법들이 동일하게 적용된다. 따라서 이 책에서는 앞으로 직관적인 이해를 돕기 위하여 2차원 데이터를 활용할 것이다.

2차원 데이터와 같이 시각적으로 그 분포 형태를 확인할 수 있는 데이터를 사용하는 것은 패턴인식 방법을 연구함에 있어서도 매우 중요하다. 특정한 분포 특성을 가진 데이터에 대하여 효율적인 방법을 개발하거나, 혹은 개발된 방법이 여러 분포 특성을 가진 데이터에 대

하여 어떤 성향을 보이는지 등을 연구하기 위해서는 원하는 분포 특성을 가진 2차원 데이터를 인공적으로 만들어서 사용할 필요도 있다. 이를 위해 확률적 난수발생 방법을 사용하는데, 매트랩의 난수 발생 함수를 사용하면 원하는 확률분포를 따르는 원하는 크기의 데이터 집합을 손쉽게 생성할 수 있다. [그림 1-6]의 데이터 집합은 평균이 [3,3]이고 공분산 행렬이 단위행렬인 가우시안 분포(Gaussian distribution)를 따르는 데이터를 확률적으로 생성한 것이다. 이 책에서는 주로 가우시안 분포를 따르는 데이터 집합을 다루게 되는데, 이와 관련된 기본 지식은 부록 B에 소개해 두었다.

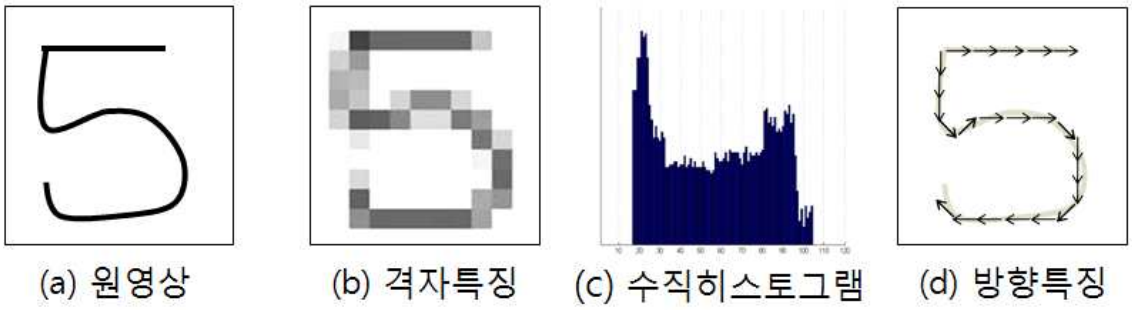
여기서 한 가지 주목할 점은, 인식기를 개발할 때 사용되어지는 데이터는 개발된 그 인식기가 실제로 다루게 될 데이터의 극히 일부에 지나지 않는 다는 것이다. 개발된 인식기가 실제로 사용될 때 주어지게 될 데이터는 분명히 개발 단계에서 주어진 학습 데이터와는 다른 값을 가질 것이다. 그럼에도 불구하고 현재의 학습 데이터를 이용하여 인식기를 개발할 수 있는 것은, 학습 데이터가 앞으로 주어질 데이터들과 동일한 분포 특성을 가진다고 보기 때문이다. 통계적 관점에서 기술하면, 인식의 대상이 되는 전체 데이터 집합을 모집단이라고 할 때, 학습 데이터는 그로부터 추출된 표본 집단이라고 볼 수 있다. 이러한 관점에서 볼 때, 인공적으로 데이터를 생성하는 것은 모집단의 확률분포를 정해 두고 그로부터 표본 집단을 추출하여 사용하는 것으로 해석할 수 있다.



[그림 1-7] 모집단과 표본집합: (a) 모집단의 확률밀도함수 (b) 등고선으로 표시된 밀도함수 (c) 모집단의 데이터 분포 (d) 모집단으로부터 얻어지는 네 개의 표본집합

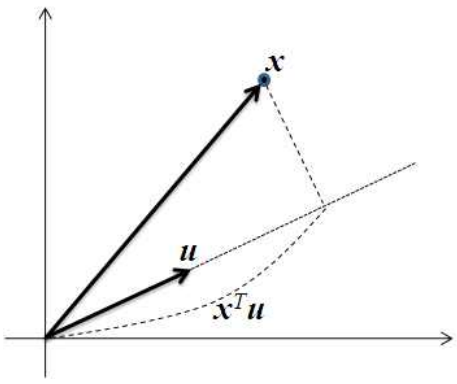
[그림 1-7]에 이 개념을 설명하는 예를 나타내었다. [그림 1-7a]에는 [그림 1-6]에서 데이터 생성을 위해 사용한 가우시안 밀도함수를 나타낸 것이고, [그림 1-7b]는 이를 2차원 평면상에 등고선으로 나타낸 것이다. 확률밀도 값이 높을수록 해당 영역에 많은 수의 데이터가 분포함을 의미한다. 모집단의 분포를 알기 위해서 [그림 1-7c]에는 충분히 많은 수의 데이터(그림에서는 10^4개를 사용하였음)를 추출하여 산점도로 나타내었다. 이어서 [그림 1-7d]는 같은 분포를 따르는 데이터 50개를 임의로 추출하여 만들어진 표본집합 네 개를 각각 산점도로 나타낸 것이다. 그림에서 알 수 있듯이 네 개의 데이터 집합은 비록 같은 모집단으로부터 추출되었으나 확률적 불확실성에 의존하여 각기 다른 집합을 형성하게 된다. 우리가 일반적으로 패턴인식 문제에서 접하게 되는 데이터집합이란 이와 같은 확률적 불확실성을 가지는 표본 집합의 하나로 볼 수 있다. 따라서 하나의 분류기를 설계하고 평가하기 위해서는 단순히 주어진 표본집합에 국한되어서는 안 될 것이며, 항상 모집단의 분포를 함께 고려해 주어야 할 것이다. 이와 관련한 여러 문제에 대하여 이후 1.4절에서 좀 더 논의할 것이며, 통계적 패턴인식 기법을 다루는 이 책에서는 이러한 관점을 바탕으로 하여 여러 가지 개념들을 정의하게 될 것이다.

1.3.2 특징과 특징추출



[그림 1-8] 숫자영상의 특징추출 예

1.2절에서 간단히 언급한 바와 같이, 주어진 데이터를 그대로 사용할 경우 발생하는 계산 비용과 메모리 비용의 증가 및 잡음 등으로 인한 문제를 해결하기 위하여 특징추출 과정은 반드시 필요하다. 특징추출에서는 주어진 패턴을 인식함에 있어서 핵심이 되는 특징을 잘 선택하는 것이 제일 중요하다. [그림 1-8]에서는 숫자영상 데이터에 대해 적용해 볼 수 있는 몇 가지 특징들을 나타내었다. [그림 1-8b]는 숫자영상을 몇 개의 격자 영역으로 나누고 각 영역에 속하는 검은 화소의 수를 그 특징값으로 사용하는 방법이고, [그림 1-8c]는 수직방향으로 영상을 사영하여 검은 화소의 수를 히스토그램으로 나타낸 것이다. 마지막으로 [그림 1-8d]는 영상조각에서 나타나는 선분의 방향성분을 특징으로 나타낸 것이다. 이러한 특징추출 과정을 거치면, 원래 영상 데이터 보다는 훨씬 적은 값을 사용하여 데이터를 표현할 수 있다. 또한 [그림 1-7b]와 같이 격자 특징을 사용하면 원래 영상에서 나타날 수 있는 잡음 등이 어느 정도 완화되는 효과도 얻을 수 있다.



[그림 1-9] 2차원 데이터의 사영에 의한 특징추출

[그림 1-8]과 같은 특징추출은 영상 데이터에 주로 사용될 수 있는 방법으로, 다른 형태의 데이터의 경우에는 적합하지 못하다. 좀 더 일반적으로 임의의 n차원 벡터에 대한 특징추출에 대하여 알아보자. [그림 1-9]에 나타난 2차원 데이터의 경우, 특징추출을 이용하여 차원

을 줄이고자 한다면, 가장 간단한 방법으로 2차원의 두 좌표값을 모두 쓰는 대신 하나만을 사용하는 방법을 생각해 볼 수 있을 것이다. 이는 2차원 벡터 $\boldsymbol{x}$를 가로축 혹은 세로축 방향으로 사영하여 얻어지는 값을 특징값으로 사용하는 것을 의미한다. 나아가, 가로축이나 세로축이 아닌 임의의 2차원 벡터 $\boldsymbol{u}$상으로의 사영값, 즉 $\boldsymbol{x}^T\boldsymbol{u}$를 특징값으로 사용할 수 있을 것이다. 그렇다면 어떤 방향으로 사영하는 것이 좋은가? 이에 대한 답을 얻기 위해 특징추출의 기본적인 목적을 다시 생각해 보자. 특징추출에서는 단순히 차원을 줄이는 것이 아니라 패턴을 인식하기 위한 핵심적인 정보를 추출하는 것이 보다 중요한 목적이 된다. 결국 좋은 사영 방향이란, 주어진 데이터의 분포 특성을 가장 잘 나타낼 수 있는 방향이 될 것이다. 이와 관련된 방법론들은 8장에서 다룰 것이다.

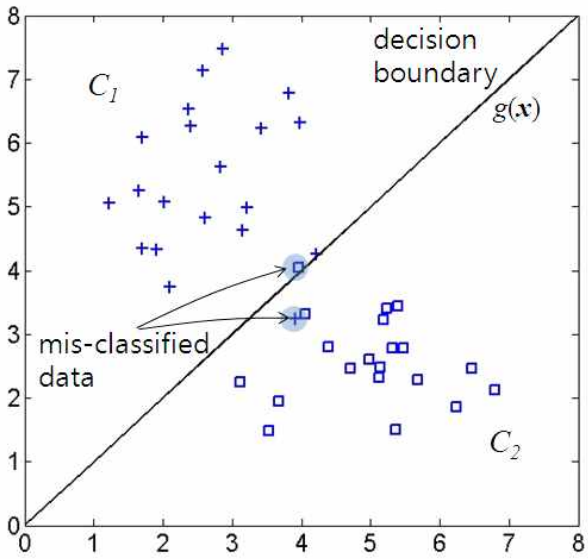
1.3.3 분류와 결정경계

특징추출 과정을 거쳐서 저차원의 특징으로 표현된 데이터 집합을 생각해보자. 이 또한 임의의 차원의 벡터로 나타낼 수 있으므로, 이 책에서는 특별히 구분할 필요가 있는 경우를 제외하고는 원래 데이터나 특징추출 과정을 거쳐서 얻어진 특징값을 모두 벡터 $\boldsymbol{x}$로 나타낸다. N개의 2차원 데이터로 이루어진 학습데이터 집합 $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_N\}$가 주어졌을 때, <분류(classification)>란 각 데이터가 어떤 패턴의 클래스에 속하는 지를 판단하는 과정이므로, 각각의 데이터를 적절한 클래스로 할당하는 것으로 볼 수 있다. 즉, $\boldsymbol{x}_i$에 해당하는 클래스 라벨 $y(\boldsymbol{x}_i)$를 결정해 주는 것이 분류이다.

[그림 1-9]에 2차원 데이터 집합에 대한 분류 예를 나타내었다. 전체 데이터 집합 X는 두 종류의 패턴으로 나누어질 수 있다. 이를 인식하기 위해 데이터 입력값을 바탕으로 가로축의 좌표값이 세로축의 좌표값보다 작은 경우는 클래스 C_1, 그렇지 않는 경우는 클래스 C_2로 분류하였다. 그림에서 보면 대각선을 기준으로 두 클래스가 구분되는데, 이와 같이 클래스를 구분해주는 직선/곡선을 <결정경계(decision boundary)>라고 한다. 즉, 결정경계란 $g(\boldsymbol{x})=0$과 같은 함수식으로 정의되는 입력공간상의 경계면을 말한다. 특히 입력공간이 2차원인 경우의 결정경계는 직선 혹은 곡선이 된다. 또 이 때 결정경계면을 정의하는데 사용되는 함수 $g(\boldsymbol{x})$를 <판별함수(discriminant function)>이라고 한다. 판별함수 $g(\boldsymbol{x})$를 이용하여 주어진 데이터에 대한 클래스 라벨은 다음 [식 1-1]과 같이 결정할 수 있다.

$$y(\boldsymbol{x})=\begin{cases}+1 & \text{if } \; g(\boldsymbol{x})\geq 0 \;\; (\boldsymbol{x}\in C_1)\\ -1 & \text{if } \; g(\boldsymbol{x})<0 \;\; (\boldsymbol{x}\in C_2)\end{cases} \qquad \text{[식 1-1]}$$

이와 같이 결정경계를 바탕으로 최종적으로 클래스를 결정하는 규칙을 <결정규칙(decision rule)>이라고 한다. [그림 1-10]의 예에서 판별함수는 $g(x_1,x_2)=x_2-x_1$가 되고, 결정경계는 $x_1=x_2$ 가 된다. 또 결정규칙은 함수 $\text{sign}(x_2-x_1)$로 나타낼 수 있다. 결국, 학습 단계에서 수행하는 분류기의 학습이란, 주어진 학습 데이터를 제대로 분류해 줄 수 있는 결정경계 $g(\boldsymbol{x})=0$ (혹은 판별함수 $g(\boldsymbol{x})$)를 찾는 것이라고 할 수 있다.



[그림 1-10] 2차원 데이터의 분류와 결정경계

1.3.4 분류율과 오차

분류를 위한 결정경계가 정해지면 성능에 대한 평가가 이루어져야 하는데, 분류 성능을 평가하는 가장 일반적인 기준이 <분류율(classification rate)> 혹은 <분류오차(classification error)>이다. 분류율이란 주어진 결정경계를 이용하여 분류를 수행하였을 때 전체 데이터들 중에 분류에 성공한 데이터의 비율을 의미하며, 분류오차란 반대로 전체 데이터 중에 분류에 실패한 데이터의 비율을 의미한다. [그림1-10]의 경우는 전체 데이터 40개 중 2개가 잘못 분류된 것으로, 분류율이 95%, 분류오차가 5%인 경우이다.

분류율이나 분류오차를 이용하여 인식기의 성능을 평가할 때에는 계산에 사용된 데이터에도 주목할 필요가 있다. 인식기를 학습할 때 사용한 학습데이터 집합 (X_{train})에 대하여 분류오차를 계산한 것을 <학습오차(training error)>라고 하는데, 이 값은 단지 인식기가 학습 과정을 제대로 수행했는지를 판단하는 기준이 될 뿐 큰 의미를 가지지 않는다. 중요한 것은, 앞으로 주어지는 학습할 때 사용되지 않은 새로운 데이터에 대한 분류오차로, 이를 <테스트오차(test error)>라고 한다. 테스트오차를 계산하기 위해서는 학습 데이터 집합 외의 별도의 테스트데이터 집합 (X_{test})이 필요하다. 학습오차(E_{train})와 테스트오차(E_{test})를 각각 수식으로 정의하면 다음과 같다.

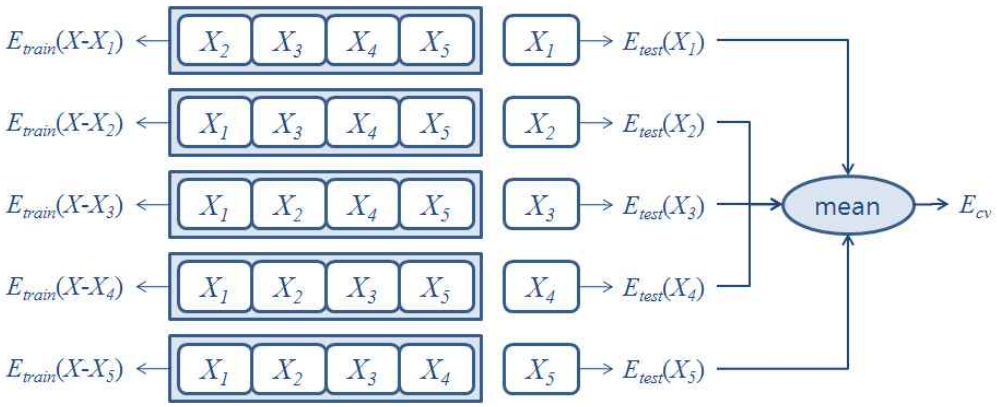
$$E_{train} = \frac{1}{N_{train}} \sum_{\boldsymbol{x} \in X_{train}} \delta[t(\boldsymbol{x}) - y(\boldsymbol{x})] \qquad \text{[식 1-2]}$$

$$E_{test} = \frac{1}{N_{test}} \sum_{\boldsymbol{x} \in X_{test}} \delta[t(\boldsymbol{x}) - y(\boldsymbol{x})] \qquad \text{[식 1-3]}$$

이 식에서 N_{train}과 N_{test}는 각각 학습데이터의 수와 테스트데이터의 수이고, $t(\boldsymbol{x})$는 데이터 $\boldsymbol{x}$가 실제로 속해있는 올바른 클래스라벨을 나타낸다. 또 함수 δ는 [ ]의 값이 0일 때만 0의 값을 가지고 그 이외에는 1을 값을 가진다. 다시 말해 $t(\boldsymbol{x})$와 $y(\boldsymbol{x})$가 같을 때만 0의 값이 된다. 그런데, 학습데이터와 마찬가지로 테스트데이터도 결국 전체 데이터 집합의 일부분에 해당될 뿐이므로 정확한 값이라고는 할 수 없다. 수학적으로 인식기의 성능을 정확히 판단하기 위해서는 결국 데이터의 확률분포함수를 알고 있다는 가정 하에, 그 분포에 따르는 평균오차값을 계산하여야 하는데, 이를 <일반화오차(generalization error)>라고 하고 다음과 같이 정의한다.

$$E_{\text{gen}} = \mathbb{E}[\delta[t(\boldsymbol{x}) - y(\boldsymbol{x})]] = \int_{-\infty}^{\infty} \delta[t(\boldsymbol{x}) - y(\boldsymbol{x})] p(\boldsymbol{x}) d\boldsymbol{x} \qquad \text{[식 1-4]}$$

일반화오차는 인식기의 이론적 성능을 분석하기 위해서는 매우 중요한 값이지만 전체 데이터 집합의 확률밀도함수를 알고 있을 때에만 계산 가능하므로, 실제 응용에서는 측정 불가능한 값이다. 따라서 실제 응용문제에서는 테스트오차를 이용하여 평가하게 되는데, 이는 어디까지나 일반화오차의 경험적 근사에 불과한 것이라는 점에 유념해야 할 것이다. 이런 의미에서 테스트오차를 <경험오차(empirical error)>라고 부르기도 한다.



[그림 1-11] 5-분절 교차검증법의 처리과정

제한된 데이터 집합을 이용하여 보다 일반화오차에 근접한 오차값을 얻어내기 위한 방법으로 <교차검증법(cross validation method)>을 사용할 수 있다. <K-분절 교차검증법(K-fold cross validation)>에서는 주어진 데이터 집합 X를 K개의 부분집합 $X_1, X_2, \ldots, X_K$로 나눈 후, 그 중 첫 번째 집합 X_1을 제외한 나머지 집합들을 사용하여 학습을 수행한 후, 남겨둔 집합 X_1을 이용하여 테스트 오차를 계산한다. 다음으로 두 번째 집합 X_2를 제외한 나머지 집합들로 학습을 수행하고 X_2를 이용한 테스트 오차를 계산한다. 이와 같은 과정을 K번 반복하면 모두 K개의 테스트오차들을 얻게 되는데, 이들의 평균을 취하면, 보다 일반화 오차에 가까운 평가값을 얻게 된다고 알려져 있고, 이를 <교차검증오차(cross validation error)>라고 하여 E_{cv}로 나타낼 수 있겠다. [그림 1-11]에 K=5인 경우의 5-분절 교차검증법을 그림으로 나타내었다. 교차검증법은 단순히 테스트 오차를 계산하기 위해서 뿐 아니라, 이를 이용

하여 보다 적절한 인식기를 얻기 위한 정보로 활용하게 된다. 이에 대해서는 1.4.3절에서 살펴볼 것이다. 일반화 능력을 평가하는 또 다른 방법으로는 주어진 데이터집합으로부터 복원추출과정을 통하여 서로 다른 복수개의 표본집합들을 생성해 내고, 이를 이용하여 오차를 계산하는 방법을 반복적으로 수행하여 그 평균값을 취하는 <붓스트랩 (Bootstrap)> 방법도 존재한다. 이에 대해서는 13장에서 다시 소개할 것이다.

1.4 패턴인식과 관련된 개념들

앞 절에서는 패턴인식기의 기본 요소들을 중심으로 그 개념을 알아보았다. 이 절에서는 패턴인식기 설계에 있어서 고려할 점을 중심으로 관련된 개념들을 소개한다.

1.4.1 분류와 군집화

패턴인식 문제에서 데이터를 분석하는 방법은 크게 <분류(classification)>와 <군집화(clustering)>의 두 가지로 나눌 수 있다. 분류란, 주어진 데이터 집합을 이미 정의되어 있는 몇 개의 클래스로 나누는 문제로서 지금까지 예를 들어온 숫자인식이나 얼굴인식 등이 모두 이에 속한다. 숫자인식의 경우, 데이터가 주어질 때 이미 그 숫자가 "0"부터 "9"까지의 클래스 중 어디에 속하는지의 정보가 함께 주어지게 된다. 즉, 입력 데이터 $X=\{\boldsymbol{x_1},\boldsymbol{x_2},...,\boldsymbol{x_n}\}$와 각 데이터의 클래스 라벨 $\{y(\boldsymbol{x_1}),y(\boldsymbol{x_2}),...,y(\boldsymbol{x_n})\}$도 함께 주어진다. 이 때 인식기는 학습 과정에서 먼저 각 클래스들의 분포 특성을 분석하여 클래스들을 분류하는 기준을 만들게 된다.

대표적인 분류 방법으로는 4장에서 소개하는 <베이지안 분류기(Bayes classifier)>와 5장에서 소개하는 <K-근접이웃 방법(K-Nearest Neighbor method, K-NN)>, 그리고 11장의 <다층신경망(multilayer perceptrons, MLP)>과 12장의 <서포트벡터머신(Support Vector Machine, SVM)> 등이 있다. 이러한 방법들에 대해서는 각 장에서 자세히 소개할 것이다.

이와 달리 군집화란, 주어지는 클래스 정보 없이 단순히 하나의 덩어리로 이루어진 데이터를 받아서, 그 분포 특성을 분석하여 임의로 복수 개의 그룹으로 나누는 것을 말한다. 즉, 미리 정해진 클래스 라벨이 없으므로 단순히 입력값의 유사성에 따라서 비슷한 입력값을 가진 데이터들끼리 같은 군집(cluster)을 이루도록 한다. 예를 들어 홈쇼핑으로부터 얻어진 고객들의 정보 (나이, 성별, 직업, 평균구매금액, 구매빈도, 거주지 등)를 바탕으로 고객을 몇 개의 그룹으로 나누는 경우를 생각해 보자. 만약 고객들을 성별이나 연령대 등 이미 정해진 기준에 따라서 클래스를 나눈다면 분류의 문제가 될 것이다. 그러나 특정 고객층이라는 것은 각 요소들이 서로 복합적으로 작용하여 형성되는 것이므로, 단순히 주어진 분류 기준만으로 고객층을 구분하는 것은 적절하지 못할 것이다. 이런 경우에 입력 정보들을 바탕으로 고객층을 묶어주는 새로운 방법을 모색해 볼 필요가 있고 이를 위해 군집화 방법이 적용될 수 있다. 군집화의 또 다른 예로 <영상분리(image segmentation)>문제가 있다. 영상분리란 하나의 영상을 배경과 전경 등의 몇 개의 영역으로 분리하는 문제이다. 즉, 영상의 각 화소가 하나의 데이터가 되고, 화소들의 정보값(명도, 컬러 등)을 바탕으로 유사한 화소를 같은 그룹으로 군집화함으로써 영상분리가 가능하다.

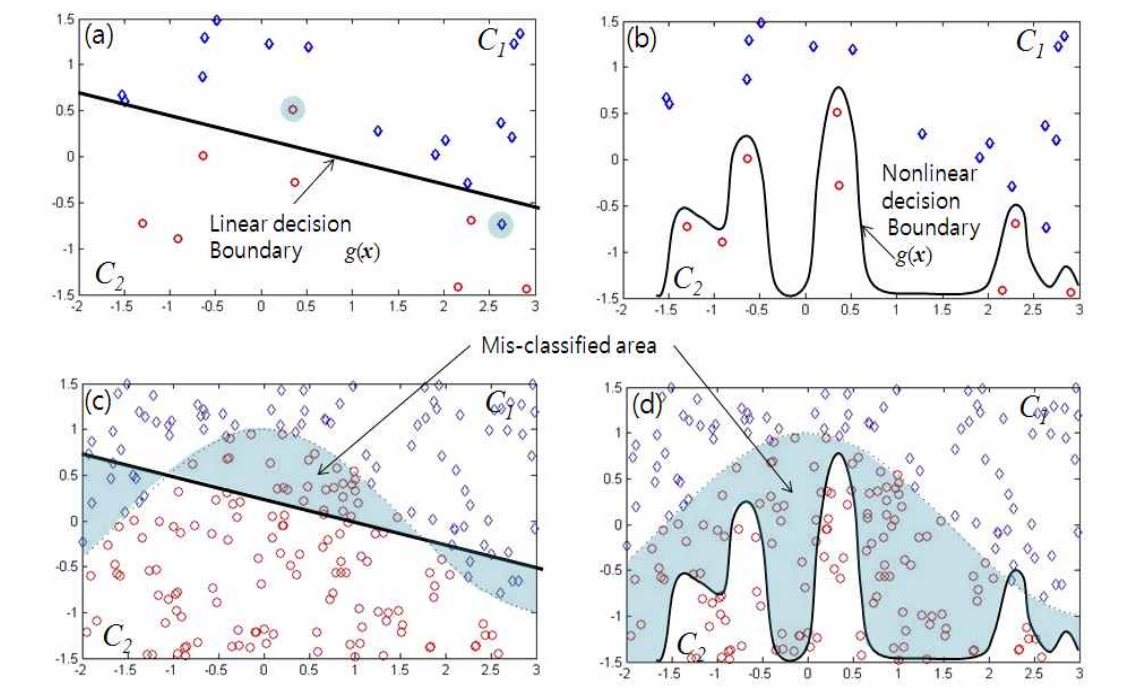
군집화의 대표적인 방법으로는 <K-평균군집화 (K-means clustering)>, <계층적 군집화 (hierarchical clustering)>, 및 <자기조직화특징맵 (self organizing feature map, SOM)>등이 있다. 이들 방법에 대해서는 7절과 11절에서 자세히 소개할 것이다.

1.4.2 교사학습과 비교사학습

앞 절에서 설명한 분류 문제와 군집화 문제의 가장 큰 차이점은 학습에 사용되는 데이터에 원하는 출력정보(클래스 정보)가 함께 주어지는지의 여부로 볼 수 있다. 따라서 패턴인식 분야에서의 분류 문제와 군집화 문제는 기계학습에서의 <교사학습(supervised learning)> 문제와 <비교사학습(unsupervised learning)> 문제로 연결된다. 교사학습이란, 말 그대로 학습시에 그 인식기가 내야할 출력값을 미리 알려주는 "교사(supervisor)"가 존재하는 것을 의미하며, 그렇지 않은 경우를 비교사학습이라고 부른다. 앞 절에서 설명한 대로 교사학습은 분류기를 만들 때 사용되며, 비교사학습은 군집화를 위하여 사용된다고 볼 수 있다. 교사학습의 대표적인 예로는 6장에서 소개하는 최소제곱법과 퍼셉트론학습, 그리고 11장에서 소개하는 다층퍼셉트론 신경망의 오류역전파 학습 알고리즘 등이 있고, 비교사학습의 대표적인 예로는 10장에서 소개하는 가우시안 혼합모델을 위한 EM학습법과, 11장에서 소개하는 자기조직화 특징맵의 경쟁과 협동에 의한 학습법이 등이 있다.
최근 들어서는 이 두 가지 접근법이 결합된 <반교사학습(semi-supervised learning)>에 대한 연구도 이루어지고 있다. 예를 들어 객체인식 문제를 수행하는 경우를 생각해 보자. 기본적으로 객체인식은 각 영상이 어떤 객체(공, 책상, 사람, 인형 등)를 나타내는지가 명확히 정해져 있으므로 분류 문제에 해당한다. 즉, 인식기를 학습시킴에 있어서 교사학습을 이용하려면 모든 데이터에 대하여 어떤 객체 클래스에 속하는지를 알려주는 출력값을 함께 제공해 주어야 한다. 이러한 객체인식기를 학습하기 위한 영상 데이터는 인터넷상에 존재하는 영상 데이터들로부터 쉽게 많은 양을 수집할 수 있으나, 수집된 데이터에 대해 각각 어떤 객체에 속하는지 클래스 라벨을 붙이는 것은 사람이 일일이 수행해 주어야 하는 비용이 많이 드는 작업이 된다. 따라서 기존의 방법에서는 제한된 양의 데이터만을 사용하여 교사학습을 수행하고, 클래스 라벨이 붙어있지 않은 데이터는 활용될 수 없었다. 반교사학습은 이렇게 버려지는 데이터들을 활용하여 분류에 유용한 정보를 추출하고자 하는 시도를 말한다. 인터넷과 정보기기의 발달로 수집 가능한 데이터의 양이 급격히 늘어남에 따라 반교사학습법에 대한 연구도 점점 주목을 받고 있다.

1.4.3 분류기 복잡도와 과다적합

앞 절에서 분류기의 성능을 평가하는 기준으로 학습데이터에 대하여 측정하는 학습오차와 앞으로 새로 주어질 데이터에 대하여 예측하는 일반화오차나 테스트오차가 있음을 배웠다. 일반적으로 학습데이터 집합은 일반화오차의 측정 대상인 모집단으로부터 추출된 표본으로 볼 수 있으므로, 두 오차값은 크게 달라지지 않을 것으로 예상할 수 있다. 그러나 예상과는 달리 이 두 오차에 의미 있는 차이가 생기는 경우가 종종 발생하는데, 이는 학습에 사용된 분류기의 복잡도와 밀접한 관계가 있다.



[그림 1-12] 학습오차(a,b)와 테스트오차(c,d)의 비교
(a,c)는 선형결정경계에 의한 분류 결과이고, (b,d)는 복잡한 비선형 경계에 의한 분류 결과를 나타냄

그 전형적인 예를 [그림 1-12]에 나타내었다. [그림 1-12a, b]는 같은 학습데이터 집합에 대해 서로 다른 복잡도를 가진 판별함수를 이용하여 분류를 수행한 결과이다. [그림 1-12a]에서는 선형함수를 학습하여 학습데이터를 최대한 잘 분류하는 결정경계를 찾은 것으로, 학습오차가 어느 정도 존재함을 확인할 수 있다. [그림 1-12b]는 고차다항식함수를 학습하여 얻어질 수 있는 결정경계로 모든 데이터를 잘 분류하여 학습오차는 0이 된다. 이렇게 얻어진 분류기의 일반화오차를 추정하기 위하여 학습데이터 생성에 사용한 모집단으로부터 충분히 많은 수의 데이터를 추출하여 테스트데이터 집합을 생성하였다. [그림 1-12c]에는 테스트데이터 집합을 선형결정경계로 분류한 예로, 그림에서 어두운 영역이 일반화 오차를 형성하여, 학습오차와 일반화오차는 비슷한 정도의 비율이 됨을 알 수 있다. 반면, [그림 1-12d]는 고차다항식 결정경계를 가진 분류기의 일반화오차를 나타낸 것으로, 학습오차는 0인 반면 일반화오차는 선형분류기보다 더 큰 값을 가지게 된다. 이러한 현상은 복잡한 분류기가 학습을 수행하는 과정에서 학습데이터에 대해서만 지나치게 적합한 형태로 결정경계가 형성되어 결국 전체 데이터 집합이 가지는 특성을 제대로 학습하지 못한 것에 기인하는 것으로, 이를 학습 데이터에의 <과다적합(overfitting)>이라고 부른다.

과다적합이 발생하는 근본적인 원인은 학습데이터가 가지는 확률적 잡음과 학습데이터수의 부족에 있다. [그림 1-12c,d]와 같은 양으로 학습데이터가 충분히 주어진다면 복잡한 결정경계를 가진 분류기로도 [그림 1-12b]와 같이 잘못된 결정경계를 찾는 일은 줄어들 것이다. 그러나 일반적으로 [그림 1-12a,b]와 같이 제한된 수의 데이터로 이루어진 표본집합만으로 학

습을 수행해야 하는 경우가 대부분이다. 따라서 주어진 상황에서 과다적합을 피하고 일반화 오차를 줄이기 위해서는 분류기의 복잡도, 즉 분류기가 정의하는 결정경계의 복잡도를 적절히 조정해 줄 필요가 있다.
결정경계의 복잡도를 조정하는 방법으로는 과다학습이 일어나기 전에 학습을 멈추는 <학습의 조기종료(early stopping)> 방법과, 학습의 목표가 되는 오차함수에 복잡한 결정경계에 대한 페널티 항을 추가하여 학습을 수행하도록 <정규화항을 가진 오차함수(Regularized error function)>를 사용하는 방법, 그리고 여러 종류의 복잡도를 가진 후보 모델들을 학습한 후 최적의 모델을 선택하는 <모델선택(model selection)> 방법 등이 존재한다. 학습의 조기종료 방법에서 종료시점을 결정하는 기준이나 모델 선택 방법에서 최적의 후보 모델을 선택하는 기준으로는 앞 절에서 소개한 교차검증 방법 등으로부터 추정될 수 있는 검증오차를 사용할 수 있다.

1.5 패턴인식의 응용

이상에서 살펴본 바와 같이 패턴인식은 그 분포 특성을 분석하여 몇 개의 그룹으로 구분하는 방법에 관한 연구로, 입력되는 패턴에 대해서 어떠한 제약도 존재하지 않으며, 따라서 응용분야도 매우 다양하다. 아래에 몇 가지 대표적인 응용 예에 대하여 나열해 보았다.

- 문자/문서인식 (character recognition/document analysis): 패턴인식의 가장 초기 연구 대상이 되었던 것 중의 하나로 숫자인식과 문자인식이 있으며, 나아가 특정 형태의 문서를 분석하고 인식하여 자동 처리하는 문제까지 확장되었다. 최근 상용 컴퓨터의 운영체제나 전자사전 등에서 문자인식 기능을 기본적으로 제공하는 경우가 많으며, 은행의 ATM에서 자동 지로납부 기능 등을 제공하는 것이 모두 이 패턴인식 기술이 상용화된 예가 된다.
- 생체인식 (biometrics): 최근 들어 상용화가 활발히 이루어지고 있는 패턴인식 응용분야 중의 하나로, 사람으로부터 얻어지는 다양한 생체정보(지문, 얼굴, 홍채, 망막, 손금 등)를 이용하여 개인의 신원을 확인하는 문제이다.
- 뇌신호처리 (brain signal processing): 뇌과학 연구가 활발해 짐에 따라 관심을 모으고 있는 주제 중의 하나이다. MEG, EEG 등 인간의 뇌로부터 측정되는 다양한 신호를 분석하여 그 의미를 알아내고, 나아가 뇌와 컴퓨터의 인터페이스 수단으로 사용하고자 하는 연구(Brain Computer Interface, BCI)에 활용된다.
- 생물정보학 (bioinformatics): 마이크로어레이나 염기서열 등 다양한 유전 정보를 분석하여 질병을 진단하거나 유전자의 역할을 규명하는 등의 문제도 결국 패턴을 인식하는 문제의 하나로 볼 수 있다. 생물정보학의 초기 단계에서는 주로 통계적 데이터 분석법이 많이 활용되었으나, 점점 기계학습에서 개발된 정교한 방법을 활용한 연구가 주목을 받고 있다.
- 로봇비전 (robot vision): 객체인식 등 영상 데이터로부터 의미 있는 정보를 추출하고 그 패턴을 인식하는 것은 로봇에게 시각정보처리 능력을 부여함에 있어서 필수적인 과제이다. 이 뿐 아니라, 로봇이 받아들이는 다양한 신호(초음파, 레이저, 적외선 등)들을 분석하여 패턴을 분류하는 것도 패턴인식 방법론이 로보틱스 분야에 적용되는 예라고 할 수 있

다.

- 의료정보 (health informatics): 뇌과학, 생물정보학과 함께 최근 들어 융합과학 분야에서 주목 받고 있는 것 중의 하나가 의료와 정보기술의 결합이다. 의료현장에서 얻어지는 다양한 임상 데이터나 최근 개발된 다양한 의료영상기기(MRI, CT, 초음파 등)로부터 얻어지는 데이터들을 분석하여 질병진단 등에 필요한 의미 있는 정보를 추출하는데 있어서 패턴인식 기술은 매우 유용하게 사용될 수 있을 것이다.
- 금융 데이터 분석 (financial data analysis): 앞 절에서 간단한 예로 사용된 것과 같은 홈쇼핑 데이터를 비롯하여 주식데이터, 보험회사의 고객 정보 등 다양한 금융데이터를 분석하여 의미 있는 정보를 추출함에 있어서 패턴인식 기술이 적용되고 있다.

연습 문제

1. 패턴인식기의 처리 과정 중 특징추출이란 어떤 것이며, 패턴인식에서 이 과정이 필요한 이유를 설명하시오.

2. 패턴인식기를 평가하는 기준으로 학습오차, 테스트오차, 일반화오차가 있다. 각각을 비교하여 설명하시오.

3. 과다적합이란 무엇이며, 과다적합이 일어나는 이유와 이를 방지하기 위한 방법을 간단히 설명하시오.

4. 분류와 군집화의 차이를 설명하고, 적용되는 예를 각각 하나씩 들어보시오.

5. 교사학습과 비교사학습에 대하여 비교하여 설명하시오.

6. 좋은 분류기란 어떤 기준을 갖추어야 할지 생각해 보고 평가하는 방법을 생각해 보시오.

7. 좋은 군집화 방법이란 어떤 기준을 갖추어야 할지 생각해 보고, 평가하는 방법을 생각해 보시오.

참고 자료

패턴인식과 기계학습과 관련된 전반적인 내용들을 다루고 있는 대표적인 저서들을 소개한다. [Bishop 96]은 패턴인식과 신경망에 관련된 내용들을 통계이론적 관점에서 체계적으로 잘 소개하고 있으며, 동 저자의 최근 저서 [Bishop 06]은 베이지안 이론의 관점에서 최근의 기계학습 방법론들에 대해서 잘 설명하고 있다. [Fukunaga 90]은 통계적 패턴인식 분야에서 오랫동안 가장 많이 참조되어온 책이며, [Duda, Hart & Stork 01]도 패턴인식의 다양한 방

법론들을 통계 이론을 바탕으로 잘 설명해 주고 있다. 이 밖에 기계학습의 기본적 개념을 알기 쉽게 소개한 책으로 [Alpaydin 04]가 있다.

[Bishop 96] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

[Bishop 06] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[Fukunaga 90] K. Fukunaga. Introduction to Statistical Pattern Recognition (2ed.). Academic Press, 1990.

[Duda, Hart & Stork 01] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification (2ed.). Wiley, 2001.

[Alpaydin 04] E. Alpaydin. Introduction to Machine Learning. MIT Press, 2004.