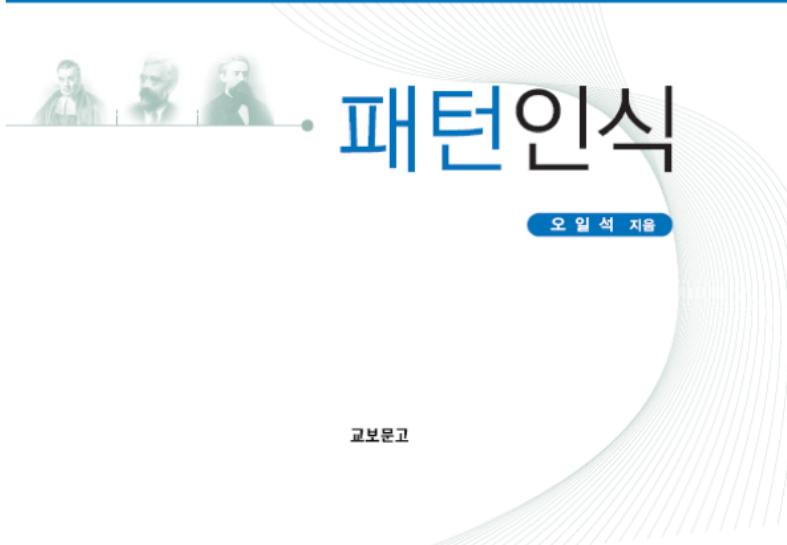


<출처>



# 1. 소개

## Recognition

사람과 동물의 인식 기능 - 매우 강건



인식 인식 인식	
인식 인식 인식	(기), 몸무게, 높은 혈압, 낮은 혈압,
인식 인식 인식	혈당, SGOT, SGPT)
인식 인식 인식	= (169, 71, 130, 80, 94, 18, 26)

(a) 누구인가?

(b) 무슨 글자인가?

(c) 정상인가?

그림 1.1 인식 해보자.

컴퓨터라는 기계도 인식할 수 있나?

### 1.1 왜 패턴 인식인가?

- 사람과 기계의 차이가 연구자들을 끌어들이는 매력으로 작용 (인식은 사람에게는 극히 쉬워나, 기계에서는 극히 어려움)
- 과학적 접근  
뇌의 정보처리 과정에 대한 이해를 바탕으로 뇌를 모방한 인식 컴퓨터에 대한 욕구  
ex) 신경망 연구
- 공학적 접근  
정도로봇, 고속 단속기 ...

## 1.2 어떻게 인식하나?

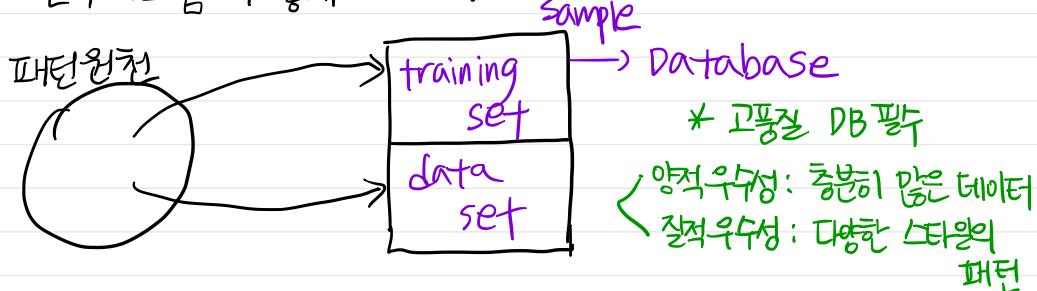
Pattern → feature → classification → class

패턴 → 특징 → 분류 → 부류

패턴	→	특징	→	분류	→	부류	( 허성설 )
$\begin{pmatrix} \text{눈의 크기 } x_1 \\ \text{얼굴 크기 } x_2 \\ \text{코의 모양 } x_3 \end{pmatrix}$			$\begin{pmatrix} x_1 = \text{크다} \\ x_2 = \text{작다} \\ x_3 = \text{뾰족하다} \\ \downarrow \\ \text{허성설의 가능성} \uparrow \end{pmatrix}$			$\begin{pmatrix} \text{김현희} \\ \text{채수현} \\ \vdots \end{pmatrix}$	

### 1.2.1 데이터베이스 수집

## 인식 시스템 어떻게 만드나?



0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

01224 E / 7 89

0 1 2 3 4 5 6 7 8 9

## 1.2.2. 특징 : 샘플 분류에 필요한 특징 추출

- 개별적인 화소를 특징으로 하면 64개의 특징,  
64차원 특징 벡터 (feature vector)  $x = (x_1, \dots, x_{64})$
- 검은 화소의 비율을 특징으로 한다면?

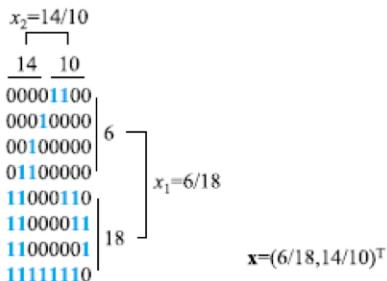


그림 1.5 필기 숫자에서 특징 추출 예

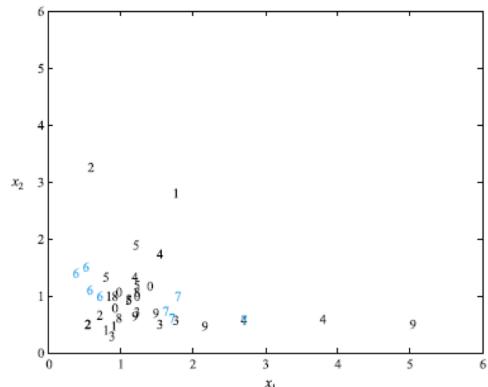


그림 1.6 특징 공간에서 샘플의 분포

- 얼마나 잘 구별해 줄까가 중요  
분별력 (discriminating power)
  - feature의 개수는 중요치 X
  - 벡터의 차원이 커질수록 계산량 또는 메모리 요구량이  
폭발적으로 증가 (curse of dimensionality)  
차원의 저주

### 1.2.3 분류

- Classification: 어떤 패턴이 들어왔을 때, M개의 부류 중의 하나로 할당해 주는 작업
- Classifier : 이 작업을 담당하는 프로그램

분류를 위해서는 1. 분류기 모델 선택 2. 분류기 학습의 과정이 필요.

#### ① classifier model selection

분류기를 표현하는 수학적 모델로 무엇을 쓸 것인지?

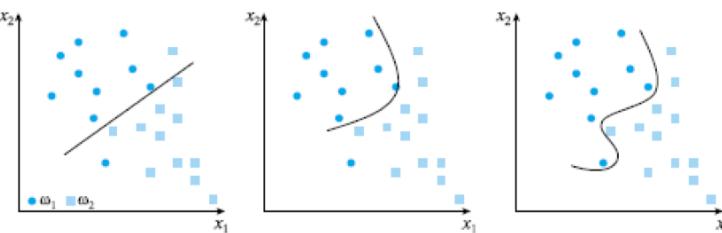
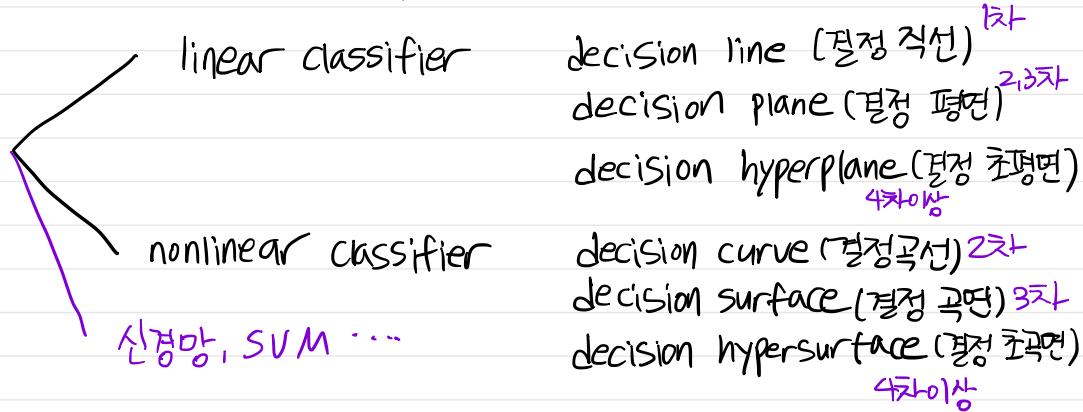


그림 1.7 분류기 설계 (모델 선택과 학습의 두 단계를 거침)

## ② classifier learning / training

- 분류기 학습, 그림 1.7(a)의 경우 직선의 식이 분류기가 되고, 매개변수인 기울기와 절편을 구하는 과정이 학습
- 훈련 집합, training set을 사용한다.

training set에 라벨의 유무 (샘플의 부류 정보 유무)에 따라 supervised learning / unsupervised learning 으로 나뉨.  
비슷한 샘플들을 같은 집단으로 모으는 것 → clustering

### 1.2.4 성능평가

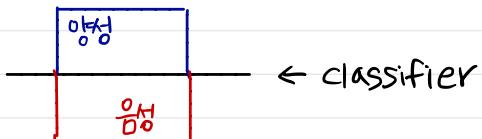
- ① 맞추는 sample, 틀리는 sample, 기각하는 sample의 수를 세어 그것을 이용
- ② 틀리게 분류하였을 때 발생할 위험을 고려하여 위험 정도를 수치화

①  $\begin{cases} \text{정인식율 (correct recognition rate)} = c/N \\ \text{기각율 (rejection rate)} = r/N \\ \text{오류율 (error rate)} = e/N \\ N = c + r + e \end{cases}$

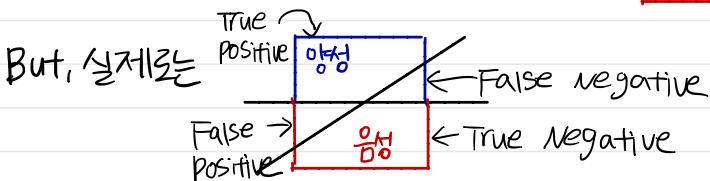
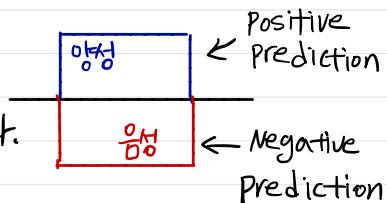
오류 경향을 더 세밀하게 분석할 때는 Confusion matrix, 흔히 행렬 사용

분류 결과		$W_1$	$W_2$
참 부류	$W_1$	$\pi_{11}$	$\pi_{12}$
	$W_2$	$\pi_{21}$	$\pi_{22}$

이진 분류 문제에서 양의 양성과 음성 데이터를 가지고 있다고 하자.



만약 모델의 정확도가 100% 이면,  
양성과 음성 데이터를 100% 잘 구분할 것이다.



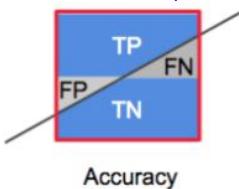
양성인데, 양성으로 제대로 검출된 것은 TP  
 음성인데, 음성으로 제대로 검출된 것은 TN  
 양성인데, 음성으로 잘못 검출된 것은 FN  
 음성인데, 양성으로 잘못 검출된 것은 FP



보통 표로 다음과 같이 표현 된다.

		Predicted		
		Positive	Negative	
Observed	Positive	TP	FN	P
	Negative	FP	TN	N

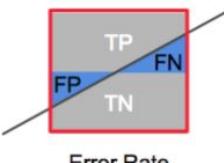
\* Accuracy



$$\frac{TP+TN}{P+N}$$

Accuracy

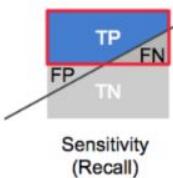
\* Error Rate



$$\frac{FP+FN}{P+N}$$

Error Rate

\* Sensitivity (=Recall)

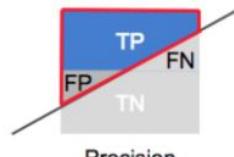


$$\frac{TP}{TP+FN}$$

모델이 얼마나 정확하게  
positive값을 찾는지

Sensitivity  
(Recall)

\* Precision



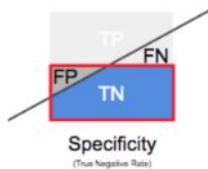
$$\frac{TP}{TP+FP}$$

Positive로  
예측한 내용 중에,  
실제 positive의 비율

Ex) 원본 데이터에 암 양성이 100개 있었는데, 모델에 있어서 90개가 양성으로 분류되었으면 Sensitive Rate = 0.9

## \* specificity (True Negative Rate)

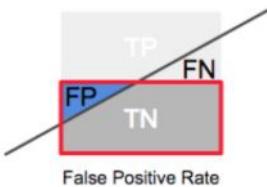
$$SP = TN / (TN + FP)$$



$$\frac{TN}{TN+FP(N)}$$

Negative로 판단한 것 중  
실제 Negative 값의 비율

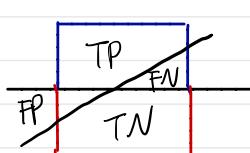
## \* False Positive Rate



$$\frac{FP}{TN+FP(N)}$$

Negative로 판단한 것 중,  
실제 Positive 값의 비율

## \* False Negative Rate



$$\frac{FN}{TP+FN(P)}$$

참고 문서:

<http://bcho.tistory.com/1206>

<https://classeval.wordpress.com/introduction/basic-evaluation-measures/>

②

위험 정도를 떠지는 두 번째 기준을 보자.

암환자를 정상인이라 판별하는 것과, 정상인을 암환자로 판별하는 것보다 위험도가 크다.

risk(위험)은 항상 그에 따른 손실을 발생시키므로 loss(손실)이라고도 부른다

⇒ 결국은 시스템이 훈련집합이 아닌 다른샘플집합, 분류기가 한 번도 보지 않은 샘플집합에 대해 보여주는 generalization(일반화) 능력이 중요하다.

너무 복잡한 모델을 사용하여 overfitting(과적합)이 되는 일은 피하는게 좋다.

모델

훈련 데이터로부터 하나의 함수가 유추되고 나면 해당 함수에 대한 평가를 통해 파라미터를 최적화한다. 이러한 평가를 위해 validation set이 사용된다. validation set은 training set에서 정합을 분할하여 사용하거나, training set의 양이 많지 않을 경우, resampling (cross validation, bootstrap...)이 사용된다.

## 1.3 시스템 설계

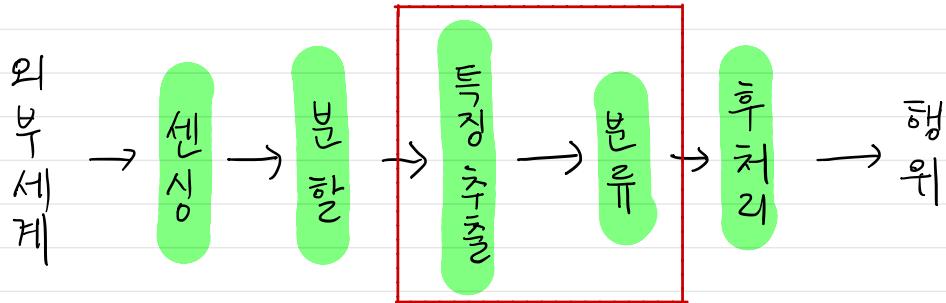
시스템 설계할 때 중요하게 고려하는 사항에 대해 구성요소 중심으로 살펴보자

\* 패턴 인식 시스템의 개발 사이클



- ① 개발초기에 고품질의 데이터베이스 확보
- ② 수단계 여러번 피드백
- ③ 특징과 분류기의 밀접한 관련성  
서로 다른 부류가 특징 공간에서  
겹침 없이 분포하는 특징을 개발  
했다면, 단순한 분류기로 고성능
- ④ 시스템이 작동할 범위를 규정하는 일반성  
인식기 개발 가능  
얼굴 인식기를 만들면, 얼굴 정면만 할지, 얼굴 앞뒤 다 할지..

## \* 패턴인식 시스템의 처리 과정



— 특징, 분류가 개발 이후 추가로 개발할 모듈이 있다

- 센싱모듈로 영상획득
  - 분할(Segmentation) 모듈 → 전체 영상에서 얼굴영역
- 인식 성능을 위해 multiple classifier combination  
선택 가능
- 후처리(Pre-processing)  
/ 패(파)/ ? / 인(임) / 시(식) → 패턴인식  
Context를 적절히 이용해야 함

최근 플랫폼시장↑ : 플랫폼에 모듈 장착하는 기술이 중요해짐

- ↖ 스탠드어론 : 프로세서와 메모리의 자원이 한계 → 알고리즘 선택 중요
- 클라이언트 서버 : 모바일 단말이 센싱, 행위 담당, 계산은 서버가