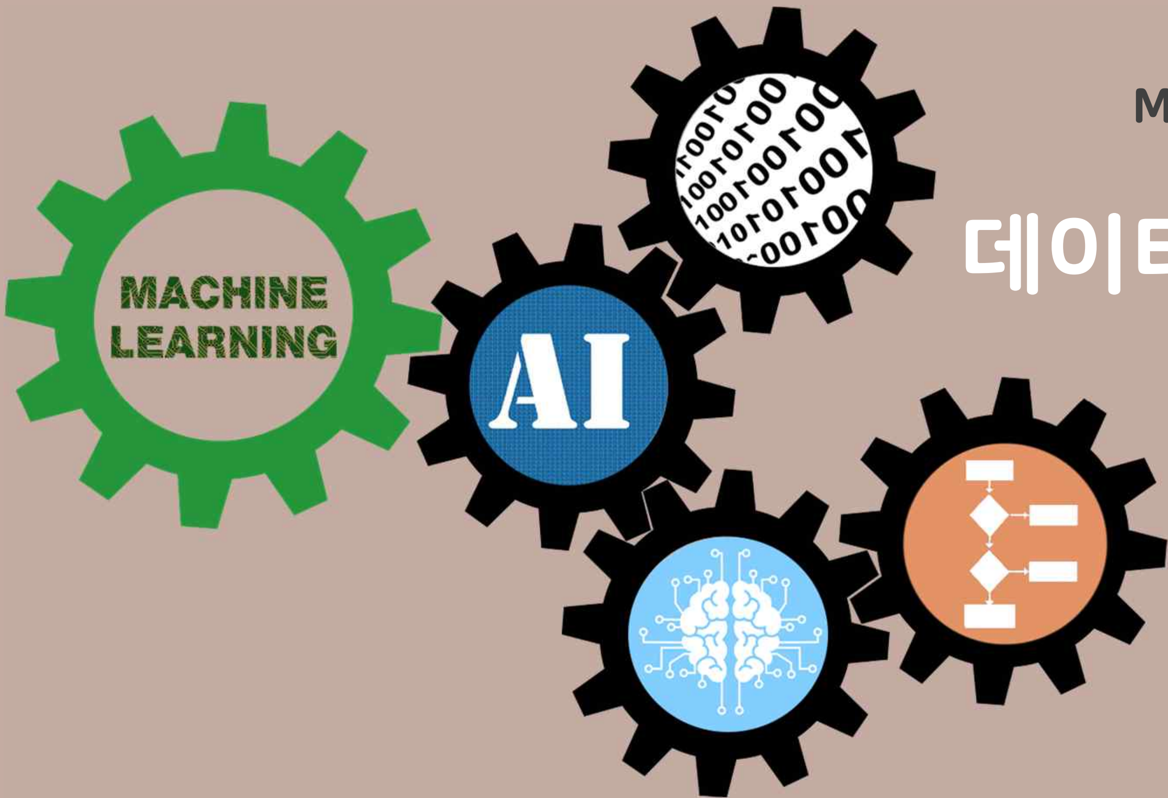


Machine Learning 5강

데이터 표현: 특징추출

컴퓨터과학과 이관용 교수

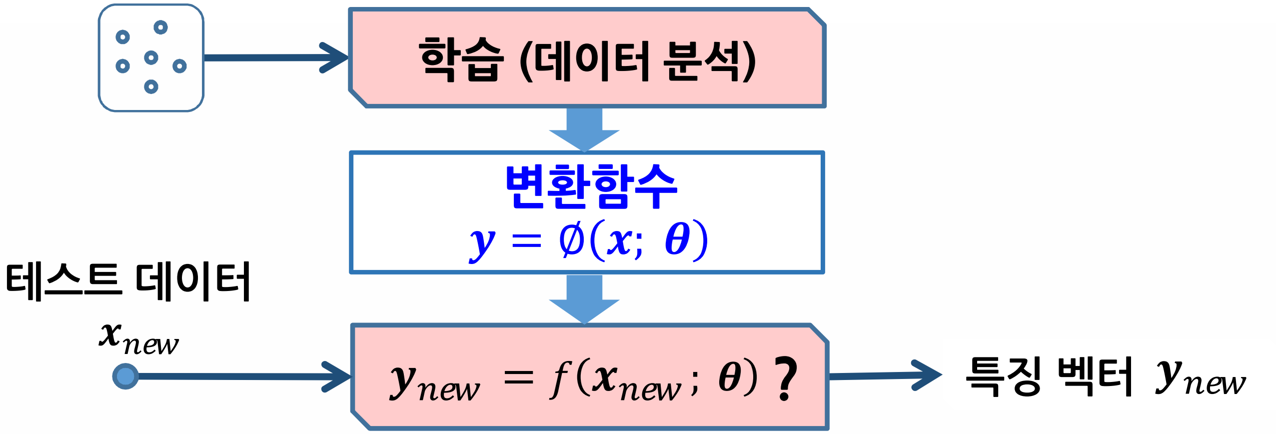
학습목차

1

선형변환에 의한 특징추출

특징추출?

학습 데이터 집합 $(D = \{\boldsymbol{x}_i\}_{i=1,\cdots,N},\ D = \{(\boldsymbol{x}_i, y_i)\}_{i=1,\cdots,N})$



학습의 목적

- 분석에 불필요한 정보를 제거하고 핵심 정보만 추출
- 차원 축소를 통한 분석 시스템의 효율 향상

변환함수

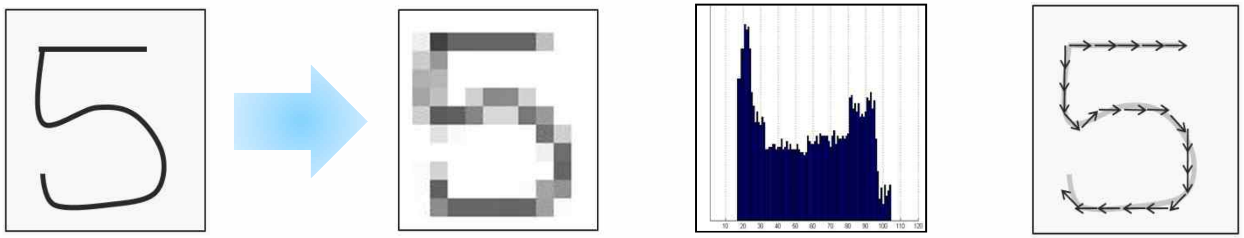
변환함수 embedding function, transformation function의 종류

- 선형변환 linear transformation $\boldsymbol{y} = \emptyset(\boldsymbol{x}) = \mathbf{W}^T\boldsymbol{x}$
  - ✓ n차원 열벡터 $\boldsymbol{x}$에 변환행렬 $\mathbf{W}$ $(n \times m)$을 곱하여 m차원 특징을 획득
  - ✓ 통계적 방법으로 특징벡터 $\boldsymbol{y}$가 원하는 분포가 되도록 하는 $\mathbf{W}$를 찾음
- 비선형변환
  - ✓ 복잡한 비선형함수 $\emptyset(\boldsymbol{x})$를 이용하여 n차원 벡터를 m차원 벡터로 매핑
  - ✓ 수작업 handcrafted에 의한 특징추출
  - ✓ 표현학습 representation learning

특징추출을 위한 접근법

수작업에 의한 특징추출

- 입력 데이터의 특성과 분석 목적에 맞는 특징을 개발자가 설계함
  - ✓ 숫자인식을 위한 특징



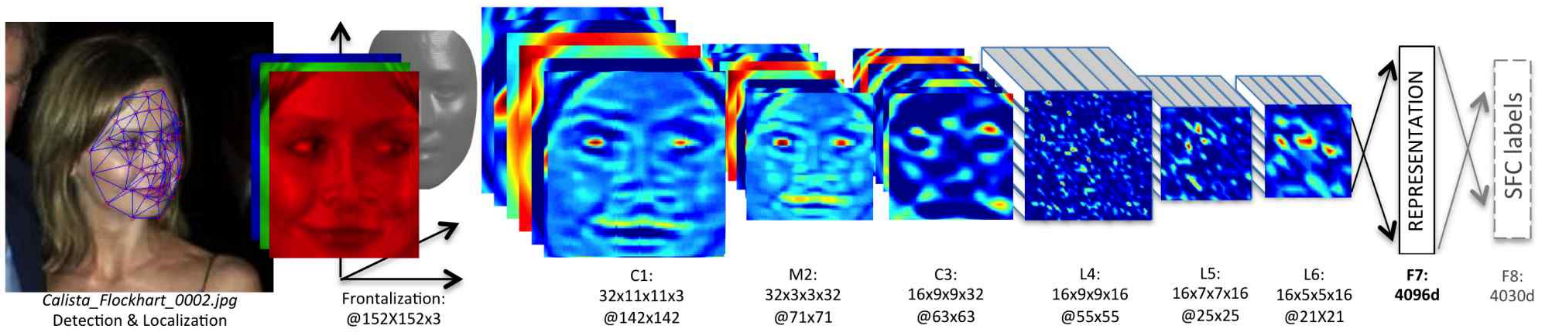
원영상(120×120) 격자특징(12×12) 수직히스토그램 방향특징

  - ✓ 영상 분석을 위한 특징 → 에지, 가로/세로 방향 성분 등
  - ✓ 문서 분석을 위한 특징 → 단어의 발생 빈도 등

특징추출을 위한 접근법

표현학습

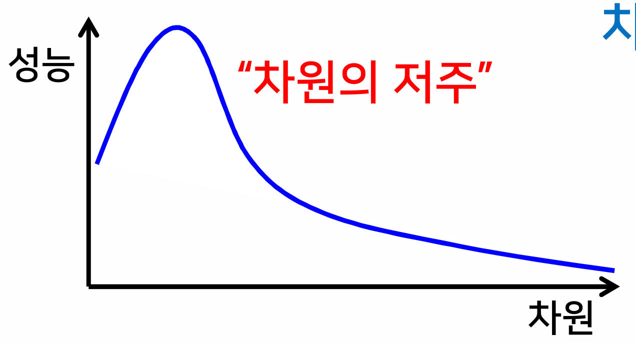
- 특징추출을 위한 비선형 변환함수를 신경망 등의 머신러닝 모델로 표현
- 학습을 통해 분석이 잘 되도록 변환함수의 최적화가 가능
- 예: 딥러닝 모델을 이용한 얼굴인식용 특징추출



Y. Taigman 등, "DeepFace: Closing the gap to Human-Level Performance in Face Verification", CVPR2014

선형변환에 의한 특징추출

차원 축소 관점에서의 특징추출



차원축소?

n차원 입력 데이터 $\boldsymbol{x}$

↓

선형변환행렬 $\boldsymbol{W}$ $(n \times m)$
$(n > m)$

↓

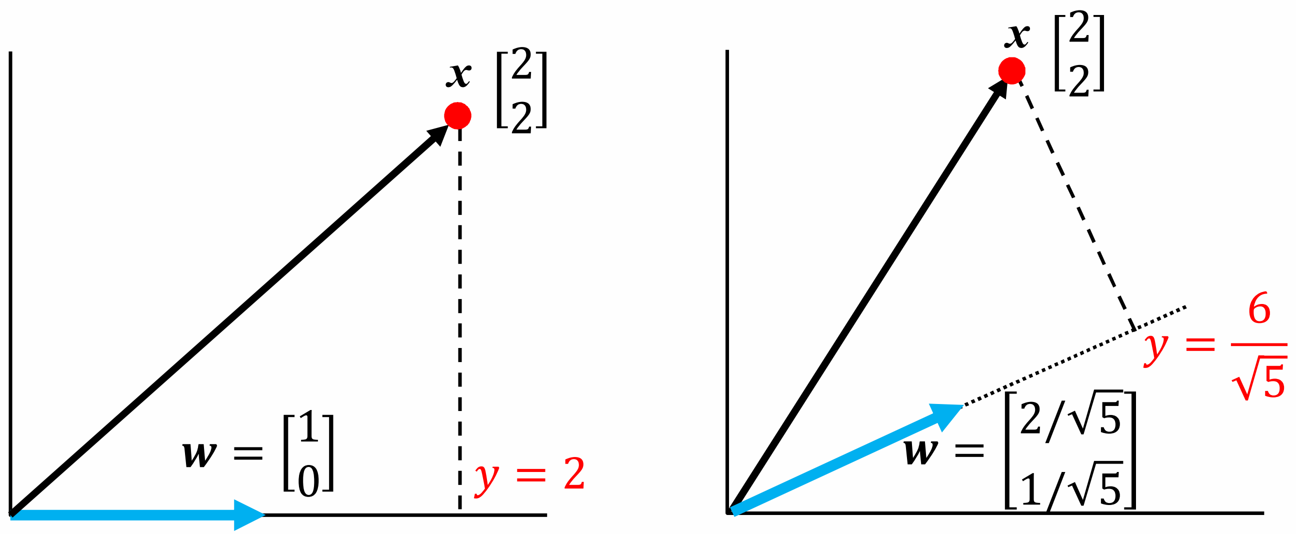
m 차원 특징 벡터 $\boldsymbol{y}$

$$\boldsymbol{y} = \mathbf{W}^T \boldsymbol{x} \rightarrow \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_1{}^T \boldsymbol{x} \\ \boldsymbol{w}_2{}^T \boldsymbol{x} \\ \vdots \\ \boldsymbol{w}_m{}^T \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_1{}^T \\ \boldsymbol{w}_2{}^T \\ \vdots \\ \boldsymbol{w}_m{}^T \end{bmatrix} \boldsymbol{x} = [\boldsymbol{w}_1, \ \boldsymbol{w}_2, \dots, \ \boldsymbol{w}_m]^T \boldsymbol{x} = \mathbf{W}^T \boldsymbol{x}$$

- 특징값 y_i
  - ✓ $\boldsymbol{x}$를 $\mathbf{W}$의 열벡터 $\boldsymbol{w}_i$위로 사영한 값 (단, $\boldsymbol{w}_i$는 단위벡터)

선형변환에 의한 특징추출

- 2차원 데이터 $\boldsymbol{x}$를 1차원 특징 y로 변환
  - 벡터 $\boldsymbol{x}$를 $\boldsymbol{w}$ 위로 사영 → $y = \boldsymbol{w}^T\boldsymbol{x}$ (단, $\boldsymbol{w}$는 단위벡터)

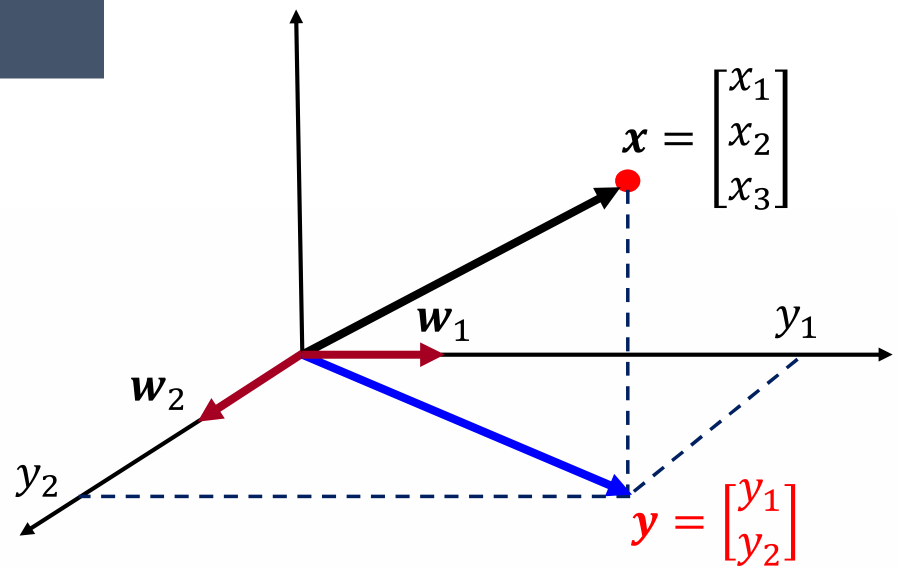


선형변환에 의한 특징추출

- 3차원 데이터 → 2차원 특징추출

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{y} = \mathbf{W}^T\boldsymbol{x} = \begin{bmatrix} {\boldsymbol{w}_1}^T\boldsymbol{x} \\ {\boldsymbol{w}_2}^T\boldsymbol{x} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$



$\boldsymbol{y}$는 $\boldsymbol{x}$를 열벡터 $\boldsymbol{w}_1, \boldsymbol{w}_2$가 이루는 2차원 평면 위로의 사영으로 얻어지는 2차원 벡터

- n차원 데이터 → m차원 특징추출

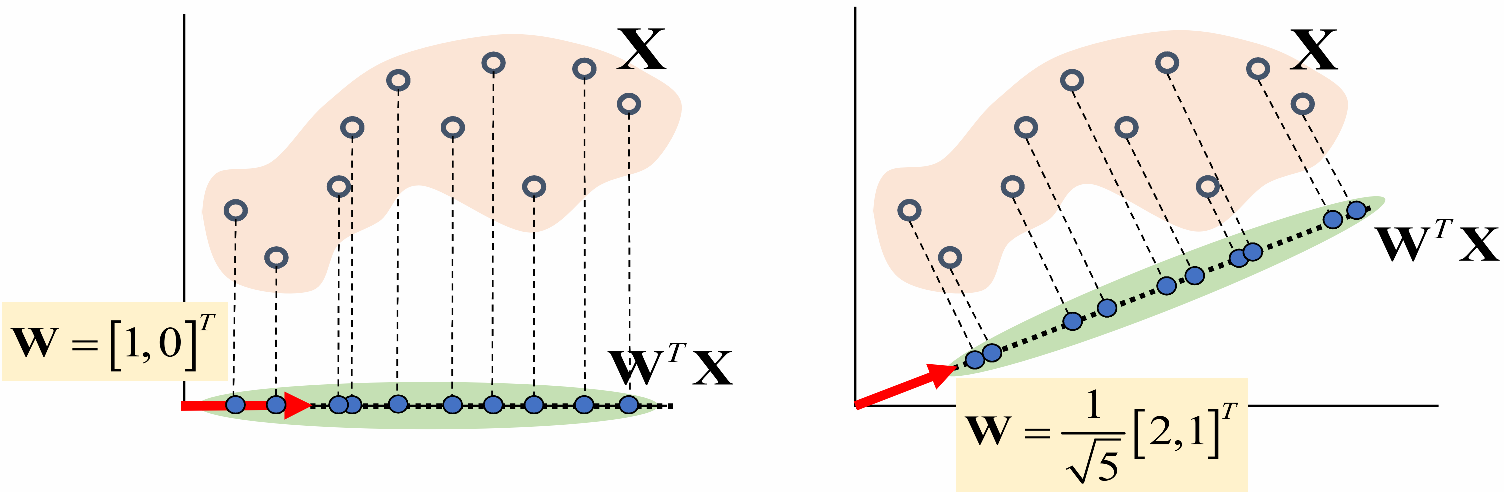
$$\boldsymbol{y} = [\boldsymbol{w}_1,\ \boldsymbol{w}_2, \dots,\ \boldsymbol{w}_m]^T\boldsymbol{x} = \begin{bmatrix} {\boldsymbol{w}_1}^T \\ {\boldsymbol{w}_2}^T \\ \vdots \\ {\boldsymbol{w}_m}^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} {\boldsymbol{w}_1}^T\boldsymbol{x} \\ {\boldsymbol{w}_2}^T\boldsymbol{x} \\ \vdots \\ {\boldsymbol{w}_m}^T\boldsymbol{x} \end{bmatrix} = \mathbf{W}^T\boldsymbol{x}$$

선형변환에 의한 특징추출

○ 전체 데이터 집합 X에 대한 특징추출

- $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N]$ ($n \times N$ 행렬)
- $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N]$ ($m \times N$ 행렬)

$$Y = W^T X = \left[W^T x_1, W^T x_2, \dots, W^T x_N \right]$$



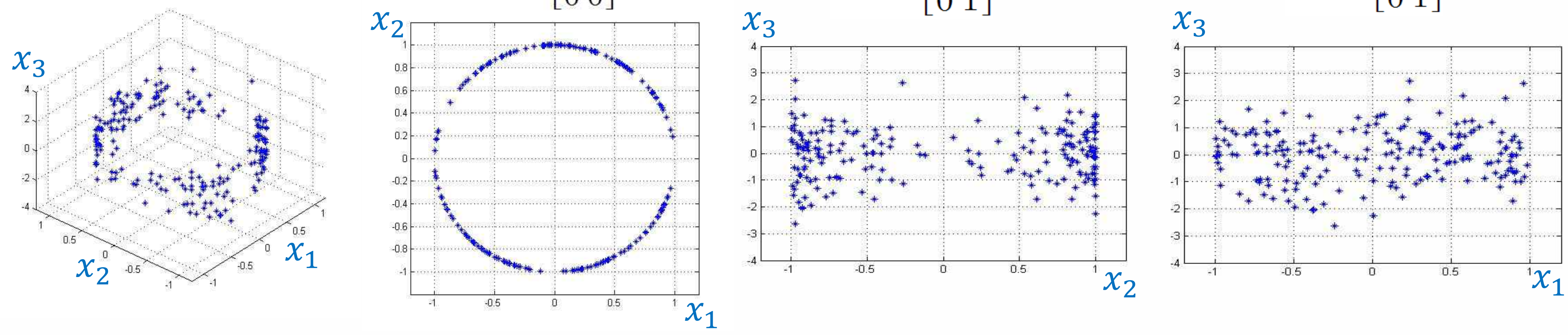
선형변환에 의한 특징추출? 주어진 데이터를 변환행렬 **W**에 의해 정해지는 방향으로 사영함으로 저차원의 특징값을 얻는 것

선형변환에 의한 특징추출

변환행렬에 따른 특징의 분포

원래 데이터

$$W_a = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad W_b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad W_c = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

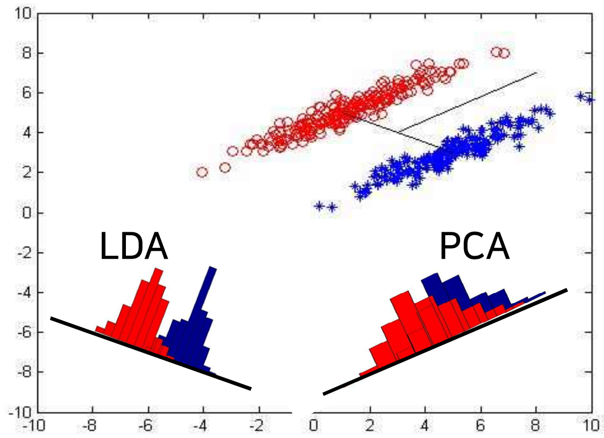


- 좋은 특징추출이란?
  - ✓ 변환행렬 **W**를 적절히 조절해서 분석 목적에 맞는 특징 분포를 만드는 것

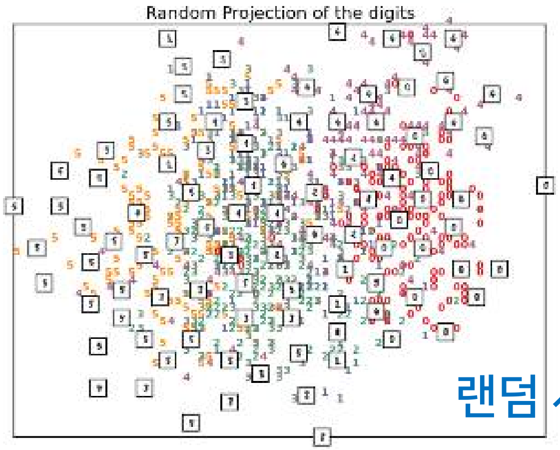
→ 통계적 특징추출

통계적 특징추출

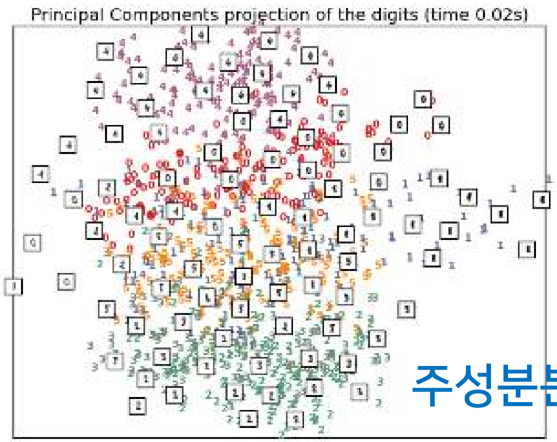
- 선형변환을 사용하는 대표적인 통계적 특징추출 방법
  - 주성분분석법 Principal Component Analysis: PCA
    - ✓ 클래스 정보 미사용 → 비지도 학습
  - 선형판별분석법 Linear Discriminant Analysis: LDA
    - ✓ 클래스 정보 사용 → 지도 학습



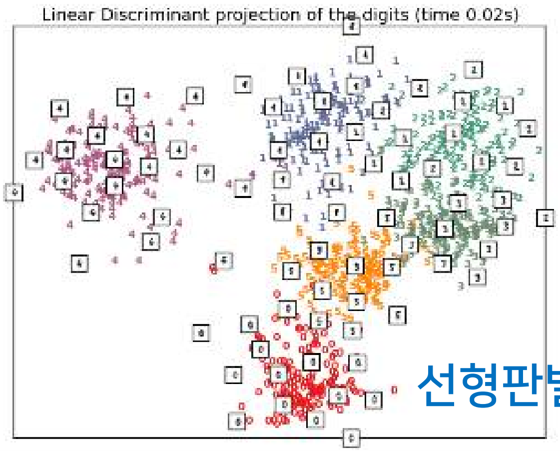
통계적 특징추출의 예



랜덤 사영



주성분분석법

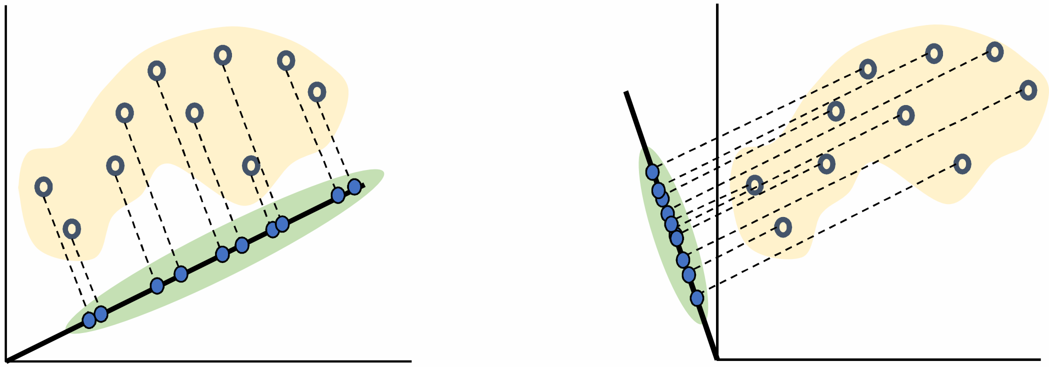


선형판별분석법

2 주성분분석법

주성분분석법

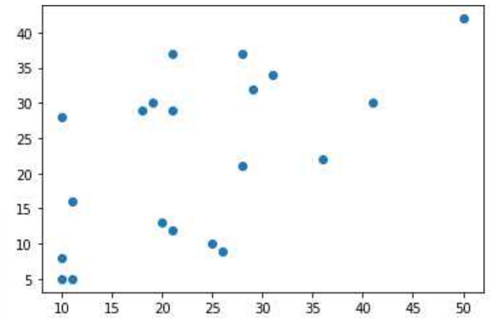
- 목적 → 변환 전의 데이터 X가 가지고 있는 정보를 차원 축소 후에도 최대한 유지



- 데이터 집합이 가능한 넓게 퍼질 수 있는 방향으로 사영을 수행
- 데이터의 분산이 가장 큰 방향으로의 선형변환을 수행
- 가장 큰 분산과 그 방향 = 공분산행렬의 최대 고유치와 고유벡터

→ 데이터의 공분산행렬의 고유치와 고유벡터를 찾아,
고유치가 가장 큰 값부터 순서대로 이에 대응하는 m개의 고유벡터를 찾아서 행렬 $\mathbf{W}$를 구성

PCA 알고리즘의 수행 단계



① 입력 데이터 집합 X의 평균 $\boldsymbol{\mu}_x$와 공분산 $\boldsymbol{\Sigma}_x$를 계산

$$\boldsymbol{\mu}_x = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i \qquad \boldsymbol{\Sigma}_x = \frac{1}{N}(X-\boldsymbol{M}_x)(X-\boldsymbol{M}_x)^T \qquad M_x = \boldsymbol{\mu}_x\mathbf{1}^T$$

(1은 모든 원소의 값이 1인 N차원 열벡터)

② 고유치 분석을 통해 공분산 $\boldsymbol{\Sigma}_x$의 고유치행렬 $\boldsymbol{\Lambda}$과 고유벡터행렬 $\boldsymbol{U}$를 계산

고유치 $\lambda_1, \cdots, \lambda_n$을 대각 성분으로 가지는 대각행렬

고유벡터 $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n$을 열벡터로 가지는 행렬

$$\boldsymbol{\Sigma}_x = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_n\end{bmatrix}[\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]^T$$

```
[ 55.34980455 202.39493229]
```

```
[[-0.75801271 -0.65223979]
 [ 0.65223979 -0.75801271]]
```

PCA 알고리즘의 수행 단계

③ 고유치가 큰 것부터 순서대로 m개의 고유치 $\{\lambda_1, \lambda_2, \cdots, \lambda_m\}$를 선택

```
[ 55.34980455 202.39493229]
```

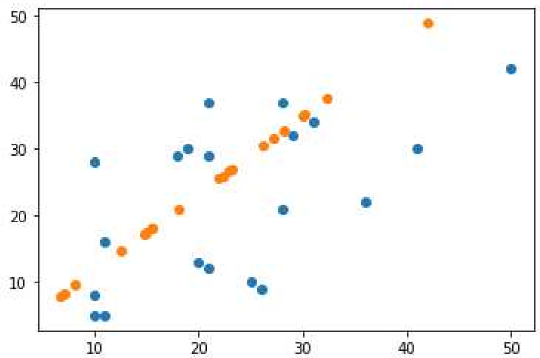
④ 선택한 고유치에 대응되는 고유벡터를 열벡터로 가지는 변환행렬 $\mathbf{W}$를 생성

$$\mathbf{W} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m]$$

```
[[-0.75801271 -0.65223979]
 [ 0.65223979 -0.75801271]]
```

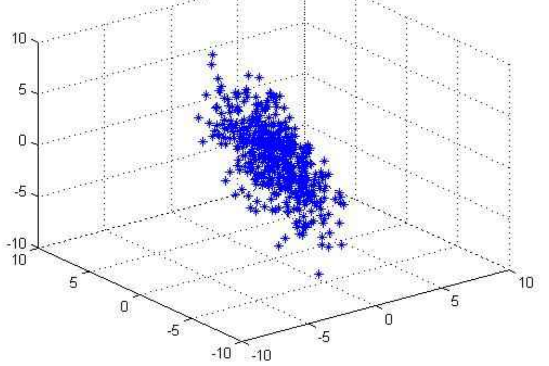
⑤ $\mathbf{W}$에 의한 선형변환으로 특징 데이터 Y를 얻음

$$\mathrm{Y} = \mathbf{W}^T \mathrm{X}$$

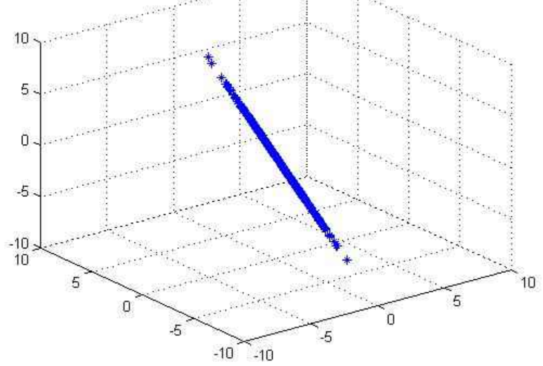


PCA 적용의 예

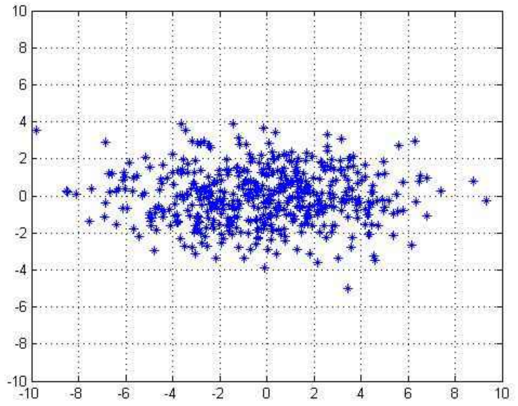
입력 데이터



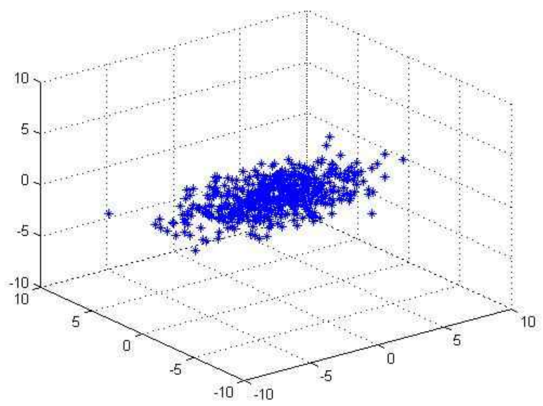
1차 주성분 벡터에 의한 특징추출



1,2차 주성분 벡터에 의한 특징추출



1,2,3차 주성분 벡터에 의한 특징추출



PCA의 수학적 유도

축소되는 차원 m을 선택하는 기준

손실되는 정보량의 비중
(m개의 특징으로 표현 가능한 정보의 비율)

$$r(n, m) = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

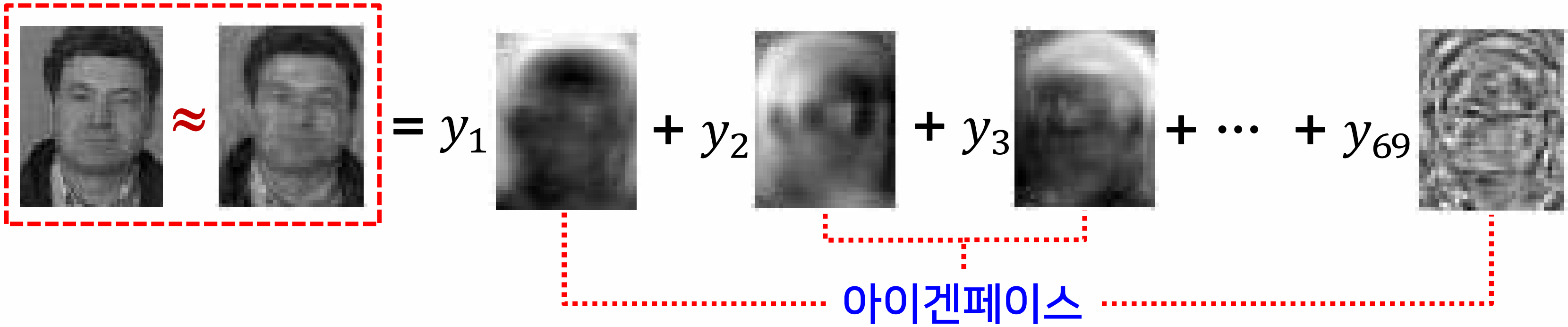
선택할 고유벡터의 수(특징의 차수) m

$$\sum_{i=1}^{m} \lambda_i \Big/ \sum_{i=1}^{n} \lambda_i > \theta$$

[예] 얼굴 영상의 표현

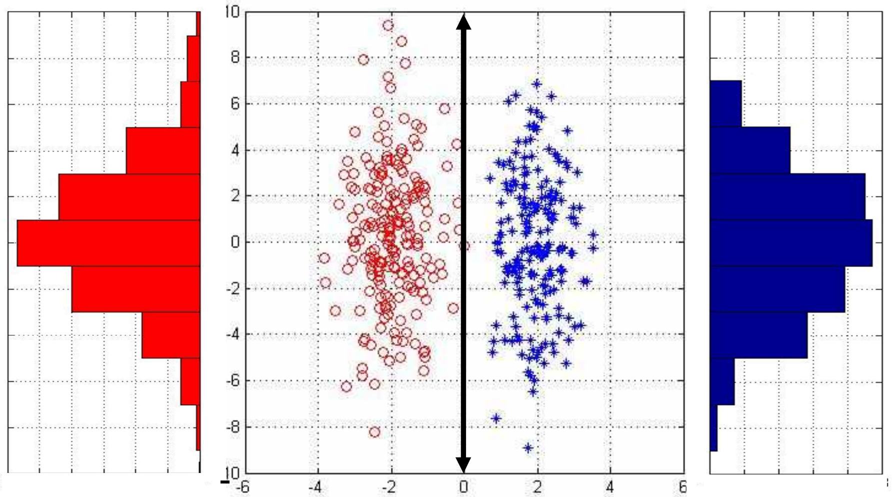
- n차원의 얼굴 영상을 m개의 기저벡터를 사용해서 표현
  - n차원을 m차원으로 차원 축소 (n≫m)
  - 좋은 기저벡터 → “Eigenface”
    - ✓ Eigenface → 얼굴 영상에 PCA를 적용하여 찾아진 고유벡터를 영상으로 표현한 것

$$\boldsymbol{x} = \mathbf{W}\boldsymbol{y} = y_1\boldsymbol{w}_1 + y_2\boldsymbol{w}_2 + \cdots + y_m\boldsymbol{w}_m$$



주성분분석법의 특성과 문제점

- 데이터 분석에 대한 특별한 목적이 없는 경우에 가장 합리적인 차원 축소의 기준
- 비지도학습
  - 클래스 레이블 정보를 활용하지 않음 → 분류의 핵심 정보의 손실 초래



분류에 적합한 특징추출?

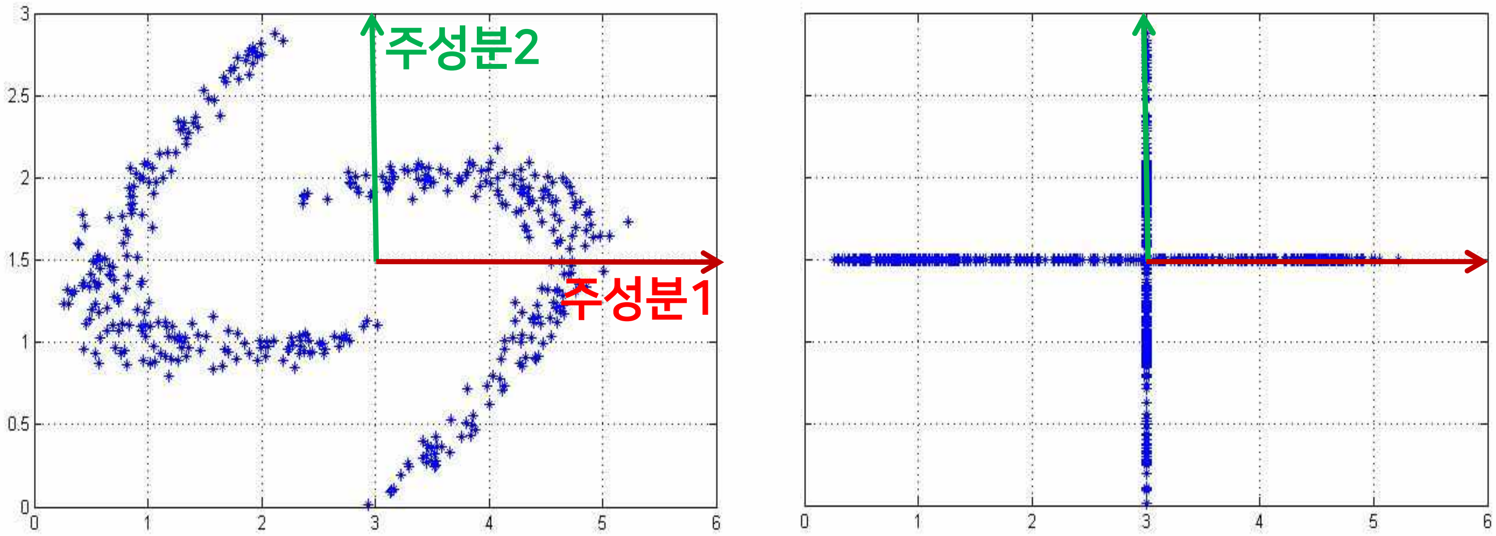
→ 데이터의 클래스 정보를 활용한 지도학습이 필요

→ "선형판별분석법(LDA)"

주성분분석법의 특성과 문제점

선형변환의 한계

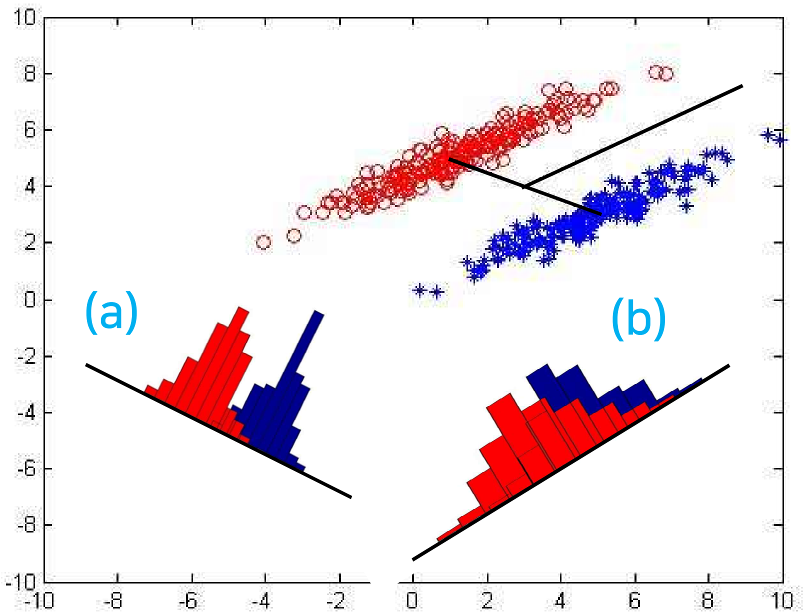
- 데이터의 비선형 구조를 반영하지 못함



3 선형판별분석법

선형판별분석법

- 목적 → 클래스 레이블 정보를 적극 활용
  → 클래스 간 판별이 잘 되는 방향으로 차원 축소



분류에 적합한 특징의 방향이란?

→ 각 클래스가 가능한 서로 멀리 떨어질 수 있도록 거리를 유지

LDA: 이진 분류 문제

클래스 간 거리에 대한 목적함수

$$J = \frac{(m_2 - m_1)^2}{{s_1}^2 + {s_2}^2}$$

$$m_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{w}^T \boldsymbol{x}_i = \boldsymbol{w}^T \boldsymbol{m}_k$$

$$s_k = \sum_{\boldsymbol{x}_i \in C_k} (\boldsymbol{w}^T \boldsymbol{x}_i - m_k)^2 = \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{w}^T (\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T \boldsymbol{w}$$

클래스 간 산점행렬
between-scatter matrix

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w}}{\boldsymbol{w}^T \sum_{k=1}^{2} \sum_{\boldsymbol{x}_i \in C_k} (\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T \boldsymbol{w}} = \frac{\boldsymbol{w}^T \boxed{S_B} \boldsymbol{w}}{\boldsymbol{w}^T \boxed{S_W} \boldsymbol{w}}$$

클래스 내 산점행렬
within-scatter matrix

목적함수를 최대화하는 기저벡터 $\boldsymbol{w}$

$$\boldsymbol{w} \propto {S_W}^{-1} (\boldsymbol{m}_1 - \boldsymbol{m}_2)$$

LDA: 다중 클래스 분류

목적함수 → 각 클래스 내의 산점도는 작게, 클래스 간의 산점도는 크게

$$J(\mathbf{W}) = \mathrm{Trace}\left\{(\mathbf{W}S_W\mathbf{W}^T)^{-1}(\mathbf{W}S_B\,\mathbf{W}^T)\right\}$$

Trace → 정방행렬의 대각원소의 합을 계산하는 연산

$$S_W = \sum_{k=1}^{M} S_k = \sum_{k=1}^{M}\sum_{\boldsymbol{x}_i \in C_k}(\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T$$

$$S_B = \sum_{k=1}^{M} N_k(\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T$$

N_k → 클래스 C_k에 속한 데이터의 수

$\boldsymbol{m}$ → 전체 데이터 집합에 대한 평균

목적함수를 최대로 하는 변환행렬 $\mathbf{W}$

${S_W}^{-1}S_B$의 고유벡터들을 열벡터로 가지는 행렬

$${S_W}^{-1}S_B = \boldsymbol{U\Lambda U}^T \longrightarrow \mathbf{W} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m]$$

알고리즘의 수행 단계

① 입력데이터 X를 각 클래스 레이블에 따라 M개의 클래스로 나누어 각각 평균 $\boldsymbol{m}_k$와 클래스 간 산점행렬 S_B, 그리고 클래스 내 산점행렬 S_W를 계산

$$\boldsymbol{m}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i \quad (k = 1,2,\cdots,M)$$

$$S_W = \sum_{k=1}^{M} S_k = \sum_{k=1}^{M} \sum_{\boldsymbol{x}_i \in C_k} (\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T$$

$$S_B = \sum_{k=1}^{M} N_k (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T$$

N_k → 클래스 C_k에 속한 데이터의 수

$\boldsymbol{m}$ → 전체 데이터 집합에 대한 평균

알고리즘의 수행 단계

② 고유치 분석을 통해 행렬 ${S_W}^{-1}S_B$의 고유치행렬 $\boldsymbol{\Lambda}$와 고유행렬벡터 $\boldsymbol{U}$를 계산

$$ {S_W}^{-1}S_B = \boldsymbol{U\Lambda U}^T = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]^T $$

③ 고유치가 큰 것부터 순서대로 m개의 고유치 $\{\lambda_1, \lambda_2, \cdots, \lambda_m\}$를 선택

④ 선택한 고유치에 대응되는 고유벡터를 열벡터로 가지는 변환행렬 $\mathbf{W}$를 생성

$$ \mathbf{W} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m] $$

⑤ $\mathbf{W}$에 의한 선형변환으로 특징 데이터 Y를 얻음

$$ \mathrm{Y} = \mathbf{W}^T \mathrm{X} $$

선형판별분석법의 특성과 문제점

- 지도학습 능력
- 선형변환의 한계
  - 복잡한 비선형 구조를 가진 경우에는 적절한 변환이 불가
    → 커널법, 비선형 매니폴드 학습법 등을 이용한 접근 필요
- 선택하는 고유벡터의 개수(축소된 특징 차원) m의 결정
  - 직접 분류를 통해 얻어지는 데이터에 대한 분류율을 기준으로 결정
  - 행렬 $S_W{}^{-1}S_B$에 의해 찾아지는 고유벡터의 개수가 제한
    - ✓ 클래스의 개수가 M이면 특징 벡터는 최대 $M-1$차원으로 제한

선형판별분석법의 특성과 문제점

작은 표본집합의 문제 small sample set problem

- 입력 데이터 수가 입력 차원보다 작은 경우

  → 클래스 내 산점행렬(S_W)의 역행렬이 존재하지 않음

  → PCA로 먼저 차원 축소한 후, 이에 대해 LDA 적용

4 거리 기반 차원 축소 방법

거리 기반 차원 축소 방법

○ 기본 목적

- □ 두 데이터 쌍 간의 거리를 최대한 유지하는 방향으로 차원 축소
  - ✓ 원래 데이터 → $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$
  - ✓ 추출된 저차원의 특징 → $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n\}$ $\quad (\boldsymbol{y}_i = f(\boldsymbol{x}_i))$
  - ✓ 원래 데이터의 거리행렬 → $\mathrm{D} = \{d_{ij}\}$ $\quad d_{ij} = \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$
  - ✓ 특징 벡터의 거리행렬 → $\Delta = \{\delta_{ij}\}$ $\quad \delta_{ij} = \mathrm{dist}(\boldsymbol{y}_i, \boldsymbol{y}_j)$
  - ✓ 목적 → $\sum_{i<j}(d_{ij} - \delta_{ij})^2$의 최소화

거리 기반 차원 축소 방법

- 거리의 정의에 따라 다양한 방법이 존재
  - 다차원 척도법 Multi-Dimensional Scaling: MDS
    - ✓ 유클리디안 거리 사용
  - t-SNE
    - ✓ 확률밀도함수를 활용하여 거리를 정의
  - Isomap
    - ✓ 측지 거리 geodesic distance 사용

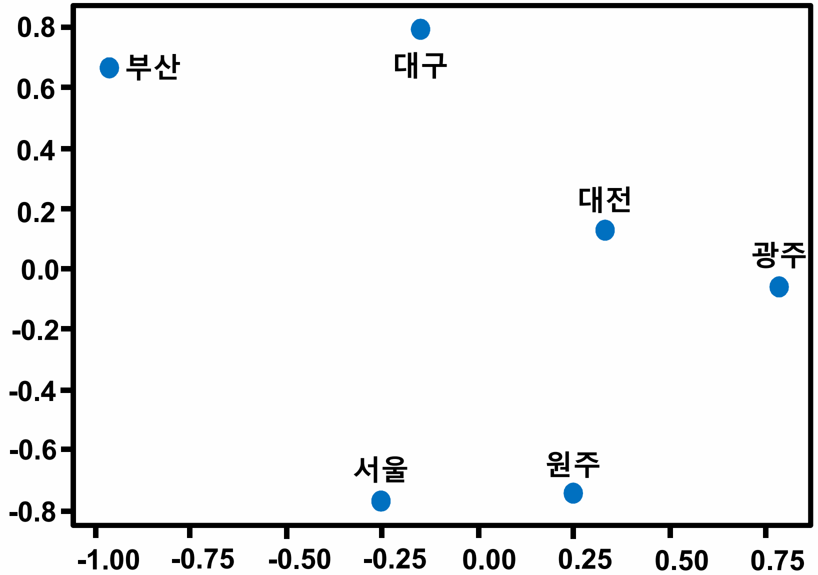
다차원 척도법

- 거리행렬 D가 값으로 정의로 되거나 유클리디안 거리 사용

$$d_{ij} = \mathrm{dist}(x_i, x_j) = \|x_i - x_j\|^2$$

- 목적 → $\sum_{i,j}(d_{ij} - \delta_{ij})^2$를 최소화하는 특징 $\{y_1, y_2, \cdots, y_n\}$를 찾는 것

D	서울	대구	대전	광주	원주	부산
서울	0	350	239	330	187	480
대구	350	0	123	283	379	110
대전	239	123	0	186	135	390
광주	330	283	186	0	274	400
원주	187	379	135	274	0	450
부산	480	110	390	400	450	0



t-SNE

t-통계적 이웃 임베딩 t-Stochastic Neighbor Embedding

- 데이터 간의 거리와 특징 간의 거리를 조건부확률을 이용한 유사도로 정의

$$d_{ij} = \text{similarity}(\boldsymbol{x}_i, \boldsymbol{x}_j) = p_{j|i} = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2 / 2\sigma_i^2)}$$

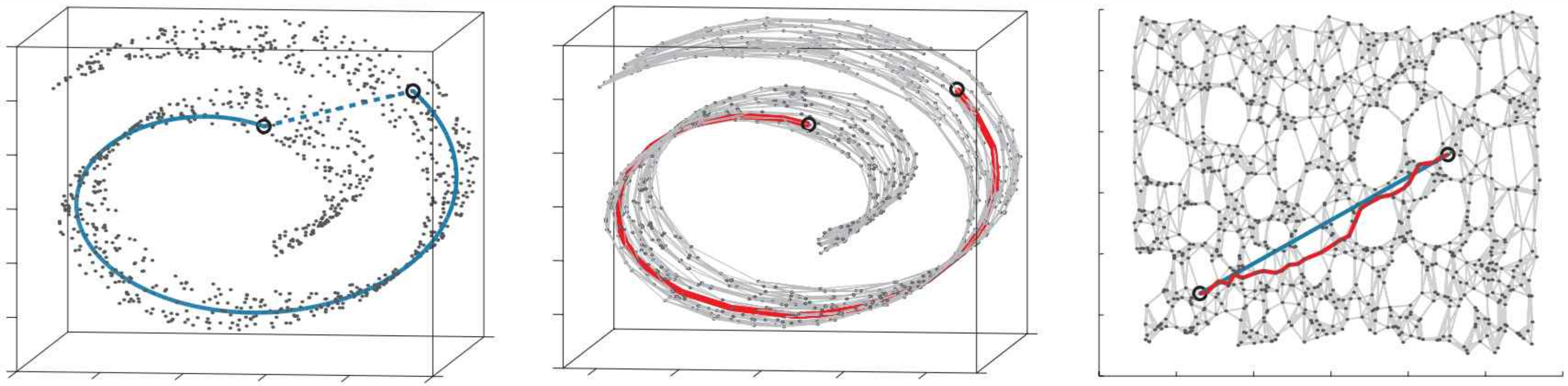
$$\delta_{ij} = \text{similarity}(\boldsymbol{y}_i, \boldsymbol{y}_j) = q_{j|i} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\boldsymbol{y}_i - \boldsymbol{y}_k\|^2)^{-1}}$$

- 거리가 멀리 떨어진 데이터 사이의 관계를 더 잘 반영

Isomap

측지 거리 사용

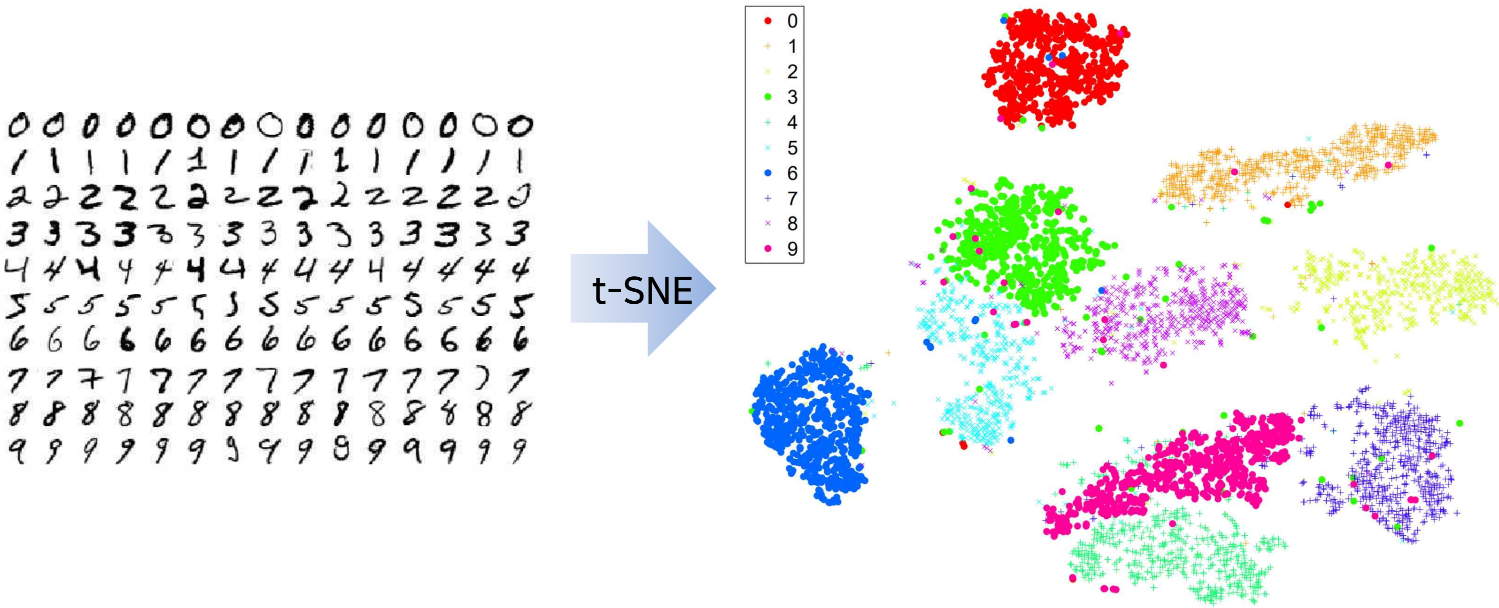
- 데이터들을 정점으로 가지는 그래프 간의 경로를 데이크스트라 Dijkstra알고리즘으로 계산

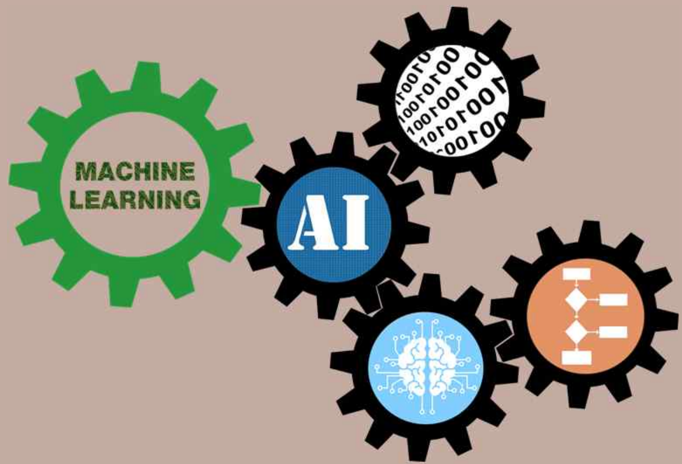


DOI: 10.1126/science.290.5500.2319

거리 기반 차원 축소 방법의 특징

- 입력 데이터와 특징 데이터 간의 매핑 함수를 정의하지 않음
  - 새로운 데이터에 대해서는 그에 대응하는 특징값을 찾을 수 없음
  - 데이터 시각화의 용도로 주로 사용





다음시간안내

제6강

앙상블 학습