〈데이터분석과기계학습 6주차〉
차원 축소를 사용한 데이터 압축

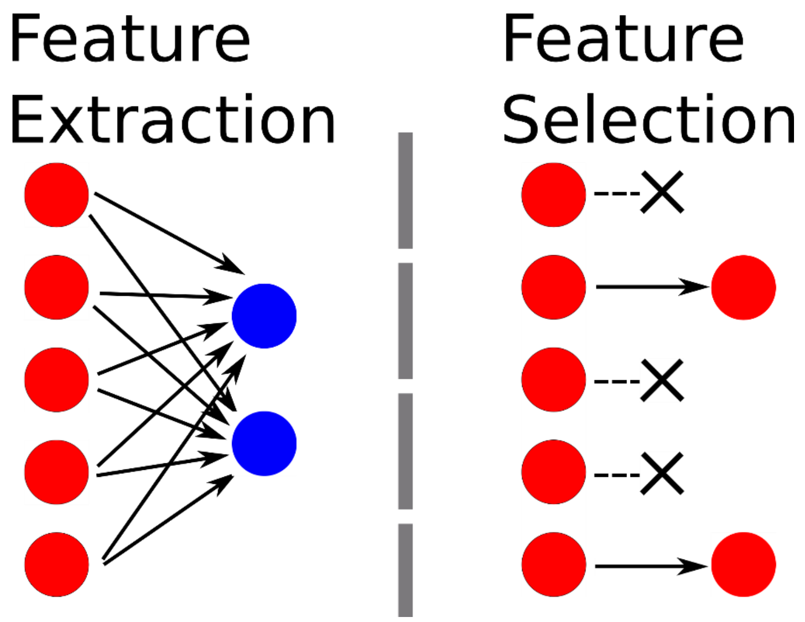
인공지능융합공학부 데이터사이언스전공
곽찬희

차원 축소를 사용한 데이터 압축

5.1. 주성분 분석을 통한 비지도 차원 축소

- 특성 추출은 데이터셋의 특성 개수를 줄이는 방법 중 하나 (다른 하나는 특성 선택)
  - ✓ 특성 선택: 원본 특성 유지
  - ✓ 특성 추출: 새로운 특성 생성 (원본의 특성을 근거로 한)

- 특성 추출의 특징
  - ✓ 저장 공간 절약
  - ✓ 알고리즘 계산 효율성 향상
  - ✓ 차원의 저주 문제 해결



5.1.1. 주성분 분석의 주요 단계

- 주성분 분석(Principal Component Analysis, PCA)
  - ✓ 고차원 데이터에서 분산이 가장 큰 방향을 찾고, 좀더 작거나 같은 수의 차원으로 투영
  - ✓ 새롭게 생긴 직교 좌표(Principal Component)는 주어진 조건에서 최대 분산!
  - ✓ 모든 주성분은 다른 주성분과 상관관계가 0(직교)을 가정
  - ✓ 특성의 scale에 민감 -> 전처리(표준화) 필요

- 수학적으로는,

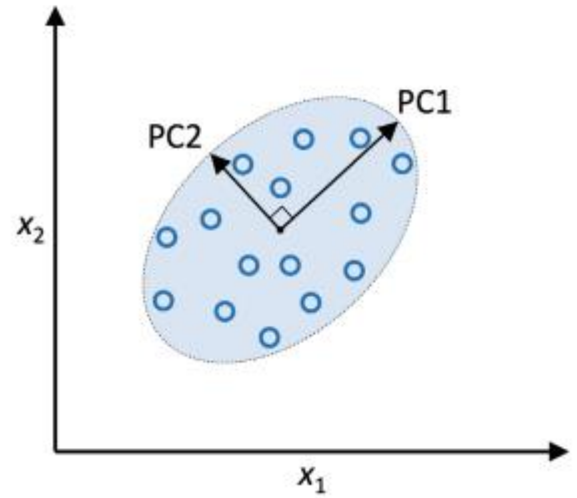
$x = [x_1, x_2, \ldots, x_d], x \in \mathbb{R}^d$

를 $xW, W \in \mathbb{R}^{d \times k}$ 로 변환

$z = [z_1, z_2, \ldots, z_k], z \in \mathbb{R}^k$

  - ✓ 원본 d 차원의 데이터를 k 차원으로 변환 (대개 k << d)

▼ 그림 5-1 원본 특성에서 찾은 주성분



5.1.1. 주성분 분석의 주요 단계

- PCA 알고리즘의 주요 단계
  1. d 차원 데이터셋을 표준화(전처리)
  2. 공분산 행렬(covariance matrix) 생성
  3. 공분산 행렬을 고유 벡터(eigenvector)와 고윳값(eigenvalue)로 분해
  4. 고윳값 내림차순으로 정렬 후 순위 매김
  5. 고윳값이 가장 큰 k 개의 벡터 선택 (k는 새로운 차원의 개수, k<=d)
  6. 최상위 k개의 고유 벡터로 투영 행렬(projection matrix) W 생성
  7. W를 활용해 d차원 데이터셋 X를 k 차원으로 변환

5.1.2. 주성분 추출 단계

- PCA의 네 단계 계산
  - ✓ 데이터 표준화 (StandardScaler)
  - ✓ 공분산 행렬 (numpy 이용)
    - 공분산은 다음과 같이 계산

$$\sigma_{jk} = \frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$$

- 평균에서 뺀 두 값이 각각 평균적으로 어떻게 움직이나?
  - 양의 공분산: 특성이 같은 방향으로 움직임(양의 상관관계)
  - 음의 공분산: 특성이 반대 방향으로 움직임(음의 상관관계)

- 최종 공분산 행렬 (3개일 때의 예)

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix}$$

5.1.2. 주성분 추출 단계

- PCA의 네 단계 계산
  - ✓ 공분산 행렬의 고윳값과 고유 벡터 (numpy 이용)
    - 고유벡터 ν 는 다음을 만족 (λ는 고윳값)

$$\Sigma\nu = \lambda\nu$$

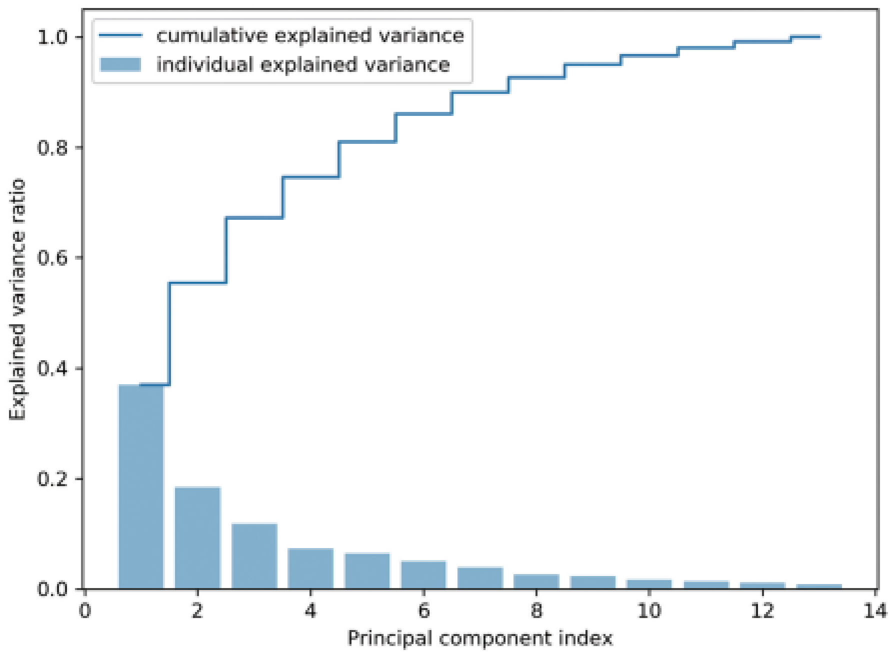
  - ✓ 고윳값을 내림차순으로 정렬한 후 순위 매김

5.1.3. 총분산과 설명된 분산

- PCA: 차원의 압축을 위해 가장 많은 정보(분산)을 가진 고유 벡터(주성분) 일부만 선택

$$\frac{\lambda_j}{\sum \lambda_j} = \frac{\text{해당 주성분의 분산(정보량)}}{\text{총 분산의 합 (정보량)}}$$

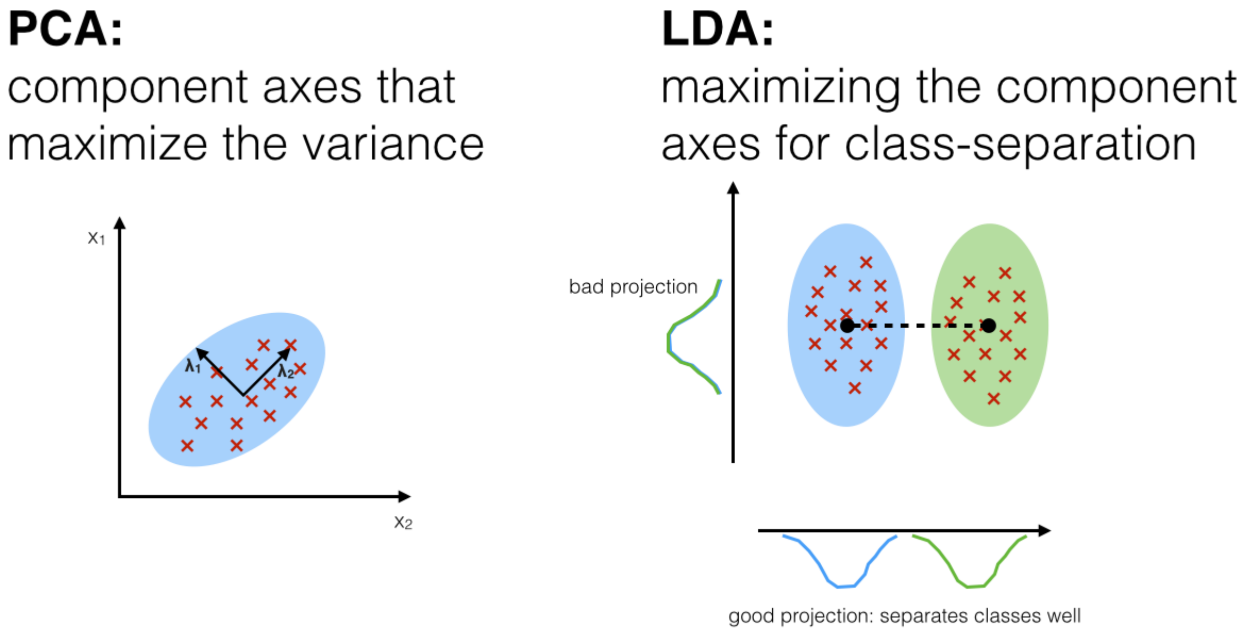
▼ 그림 5-2 주성분의 설명된 분산



강남대학교 KANGNAM UNIVERSITY

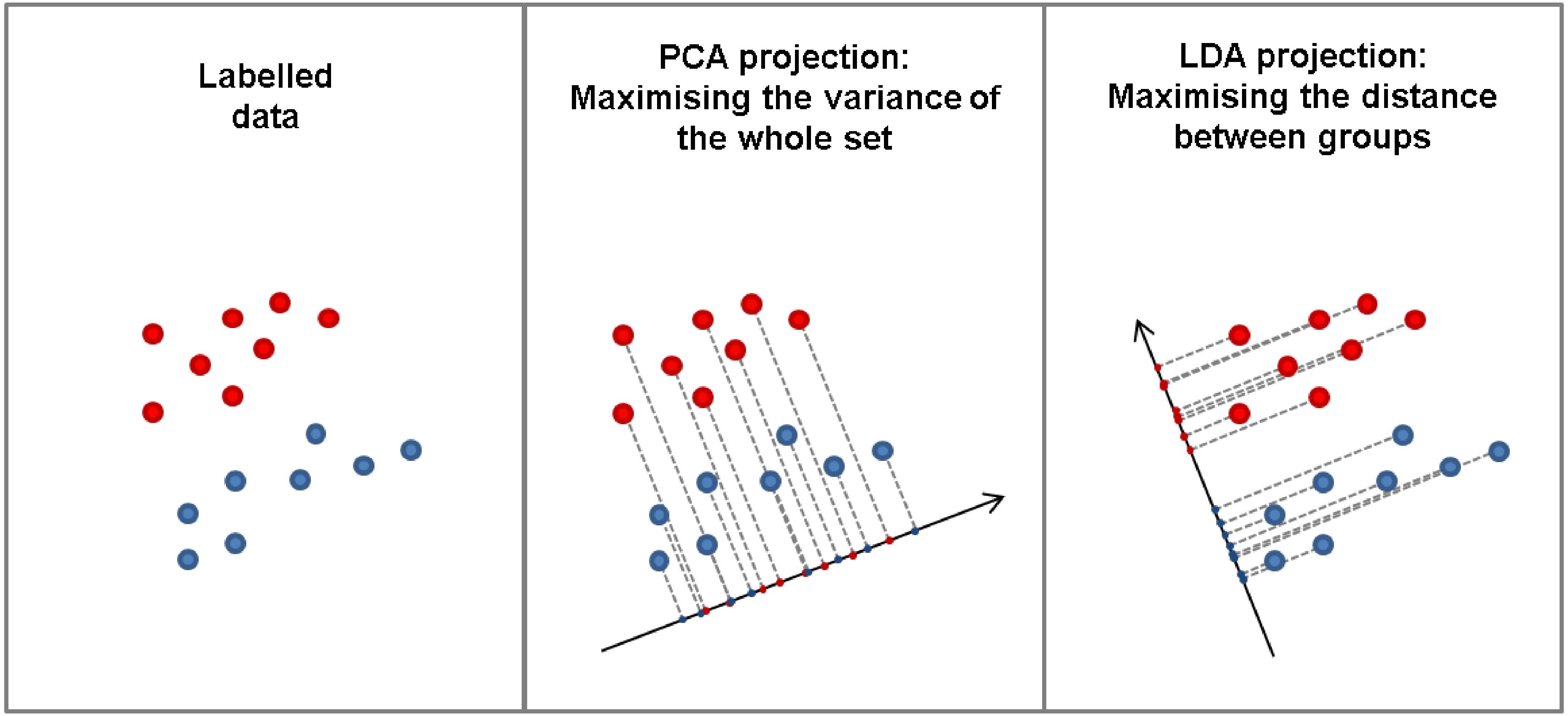
5.2. 선형 판별 분석을 통한 지도 방식의 데이터 압축

- 선형 판별 분석 (Linear Discriminant Analysis, LDA)
  - ✓ 규제가 없는 모델에서 과적합 정도를 줄이고 계산 효율성을 높이기 위한 특성 추출 기법
  - ✓ PCA는 분산이 최대인 직교 성분 축을 찾으려 하지만, LDA는 최적의 클래스 분류가 가능한 특성 부분 공간을 찾는 것이 목표



5.2. 선형 판별 분석을 통한 지도 방식의 데이터 압축

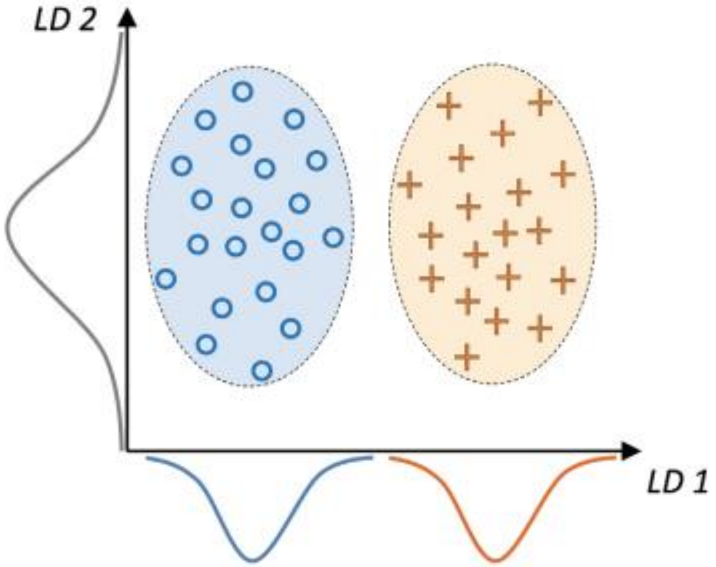
- 직관적인 이해 시도



5.2. 선형 판별 분석을 통한 지도 방식의 데이터 압축

- LDA의 특징
  - ✓ 데이터가 정규분포라 가정
  - ✓ 클래스가 동일한 공분산 행렬을 가짐
  - ✓ 샘플은 서로 독립
  - ✓ 하지만 위의 가정이 위배되더라도 잘 작동하는 편

그림 5-6 선형 판별 분석



5.2.2. LDA 내부 동작 방식

1. d 차원의 데이터셋을 표준화 전처리
2. 각 클래스에 대해 d차원의 평균 벡터 계산
3. 클래스 간 산포 행렬(scatter matrix) S_B 와 클래스 내 산포 행렬 S_W 구성
4. $S_W^{-1} S_B$ 의 고유 벡터와 고윳값 계산
5. 고윳값을 내림차순 정렬 후 순서를 매김
6. 고윳값이 가장 큰 k 개의 고유 벡터를 선택해 d x k차원의 변환 행렬 W 구성
7. W를 이용해 샘플을 새로운 특성 부분 공간으로 투영

5.2.3. 산포 행렬 계산

- 평균벡터 m_i 는 클래스 I 의 샘플에 대한 특성의 평균값 μ_m 를 저장

$$m_i = \frac{1}{n}\sum_{x \in D_i}^{c} x_m$$

- 클래스 내 산포도 S_W 는 개별 클래스 i의 산포 행렬을 더해서 만듦

$$S_W = \sum_{i=1}^{c} S_i$$

$$\mathrm{S_i} = \sum_{x \in D_i}^{c} (x - m_i)(x - m_i)^T$$

5.2.3. 산포 행렬 계산

- S_W 로 더하기 전, 스케일 조정 필요. 각 클래스 샘플 개수로 나눠줌
  - ✓ 산포 행렬의 정규화 버전 -> 공분산 행렬

$$\Sigma_i = \frac{1}{n_i} S_i = \frac{1}{n_i} \sum_{x \in D_i}^{c} (x - m_i)(x - m_i)^T$$

  - ✓ 다만 항상 샘플 개수로 나누는 것은 아니며, 산포 행렬을 계산하는 방식에는 여러 가지가 있음

5.2.3. 산포 행렬 계산

- **산포 행렬 S_B**

$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T$$

  - ✓ 여기서 m은 모든 클래스의 샘플을 포함하여 계산된 전체 평균

- **위 정보를 이용하여 $S_W^{-1} S_B$의 고윳값을 계산**
  - ✓ Why?

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

$$S_B = (m_1 - m_2)(m_1 - m_2)^T$$

$$S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T$$

$$(w^T S_B w) S_W w = (w^T S_W w) S_B w$$

$$S_W w = \lambda S_B w$$

$$S_B^{-1} S_W w = \lambda w$$

5.3. 커널 PCA를 사용하여 비선형 매핑

- SVM에서 나왔던 커널 트릭을 PCA에 적용
  - ✓ 최대분산 찾기를 커널 함수를 이용
  - ✓ Polynomial, hyperbolic tangent, radial basis function

▼ 그림 5-11 선형 문제와 비선형 문제