6장. 차원 축소

01 차원 축소(Dimension Reduction) 개요

- 정의: 많은 피처로 구성된 다차원 데이터 세트의 차원을 축소해 새로운 차원의 데이터 세트를 생성하는 것
- 직관적인 데이터 해석을 가능하게 함
- 피처 선택과 피처 추출로 나뉨

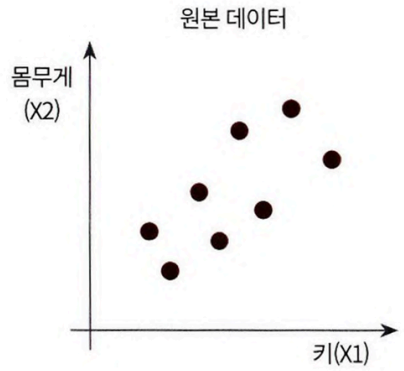
피처 선택 **(특성 선택)**	특정 피처에 종속성이 강한 불필요한 피처를 아예 제거하고, 데이터 특징을 잘 나타내는 주요 피처만 선택
피처 추출 **(특성 추출)**	기존 피처를 저차원의 중요 피처로 압축해 추출 피처를 함축적으로 더 잘 설명할 수 있는 다른 공간으로 매핑해 추출 데이터를 잘 설명할 수 있는 잠재적인 요소를 추출함

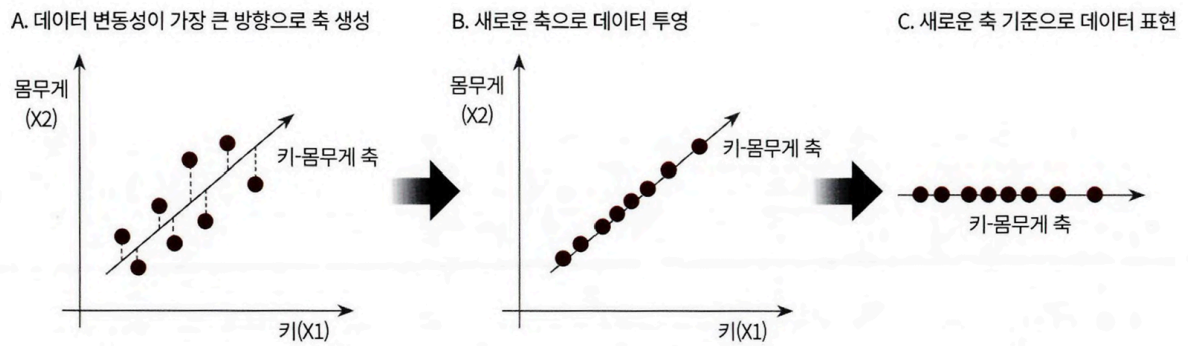
- 잠재적인 요소를 찾는 대표적인 차원 축소 알고리즘으로 PCA, SVD, NMF 등을 들 수 있음
- 이미지 데이터와 텍스트 문서를 다룰 때 주로 사용됨

02 PCA(Principal Component Analysis)

PCA 개요

- 대표적인 차원 축소 기법, 주성분 분석법이라고도 불림
- 여러 변수 간 존재하는 상관관계를 이용해 이를 대표하는 주성분을 추출해 차원 축소
- 기존 데이터 정보 유실의 최소화
- 가장 높은 분산을 가지는 데이터의 축을 찾아 차원을 축소, 이것을 PCA의 주성분으로 사용





- 가장 큰 데이터 변동성을 가진 방향으로 첫 번째 벡터 축을 생성하고, 이에 직각이 되는 벡터를 두 번째 벡터 축으로 함. 두 번째 축과 직각이 되는 벡터를 다시 세 번째 축으로 설정해 생성
- 생성된 벡터 축에 원본 데이터를 투영 → 벡터 축의 개수만큼의 차원으로 원본 데이터 차원 축소

  ⇒ 원본 데이터의 피처 개수에 비해 매우 작은 주성분으로 원본 데이터의 총 변동성을 설명 가능

- 선형대수 관점

  입력 데이터의 공분산 행렬을 고유값 분해하고, 구한 고유벡터에 입력 데이터를 선형 변환

고유 벡터	PCA 주성분, 입력 데이터의 분산이 큰 방향
고윳값	고유벡터의 크기, 입력 데이터의 분산

  다음의 단계로 수행됨

1. 입력 데이터 세트의 공분산 행렬을 생성합니다.
2. 공분산 행렬의 고유벡터와 고유값을 계산합니다.
3. 고유값이 가장 큰 순으로 K개(PCA 변환 차수만큼)만큼 고유벡터를 추출합니다.
4. 고유값이 가장 큰 순으로 추출된 고유벡터를 이용해 새롭게 입력 데이터를 변환합니다.

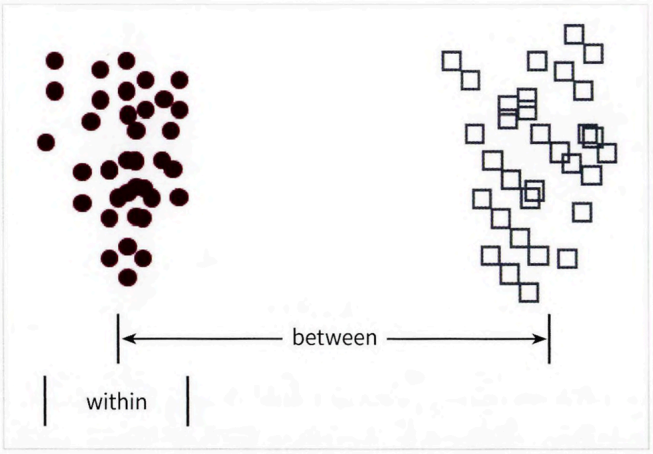
03 LDA(Linear Discriminant Analysis)

LDA 개요

- 선형 판별 분석법
- PDA와 매우 유사하지만, 분류에서 사용하기 쉽도록 개별 클래스를 분별할 수 있는 기준을 최대한 유지하면서 차원을 축소한다는 차이점을 가짐
- 비교

PCA	입력 데이터 변동성의 가장 큰 축을 찾음
LDA	입력 데이터의 결정 값 클래스를 최대한으로 분리할 수 있는 축을 찾음

- 클래스 간 분산은 크게, 클래스 내부 분산은 작게 하는 방식으로 비율을 최대화하여 차원을 축소함



- 클래스 간 분산과 클래스 내부 분산 행렬을 생성 → 이에 기반해 고유벡터를 구하고 입력 데이터를 투영

  다음의 단계로 수행됨

  1. 클래스 내부와 클래스 간 분산 행렬을 구합니다. 이 두 개의 행렬은 입력 데이터의 결정 값 클래스별로 개별 피처의 평균 벡터(mean vector)를 기반으로 구합니다.
  2. 클래스 내부 분산 행렬을 S_w, 클래스 간 분산 행렬을 S_B라고 하면 다음 식으로 두 행렬을 고유벡터로 분해할 수 있습니다.

$$S_W^T S_B = \begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} e_1^T \\ \cdots \\ e_n^T \end{bmatrix}$$

  3. 고유값이 가장 큰 순으로 K개(LDA변환 차수만큼) 추출합니다.
  4. 고유값이 가장 큰 순으로 추출된 고유벡터를 이용해 새롭게 입력 데이터를 변환합니다.

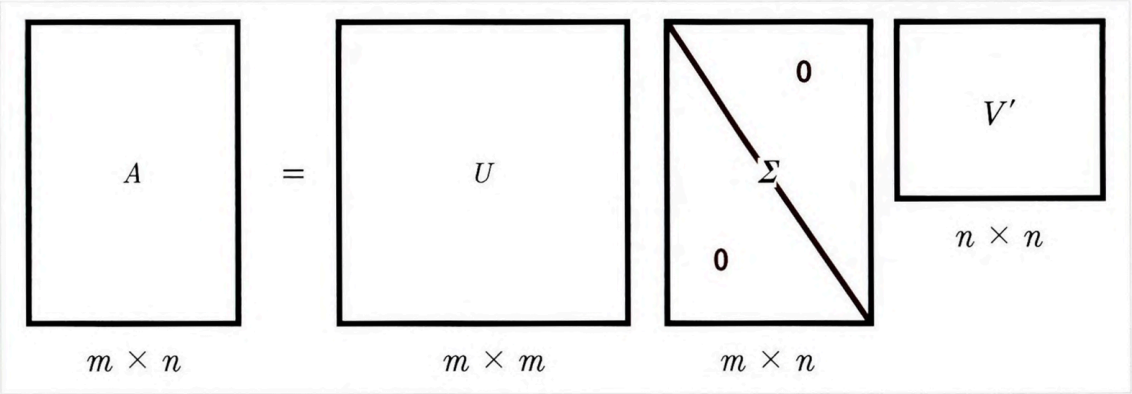
04 SVD(Singular Value Decomposition)

SVD 개요

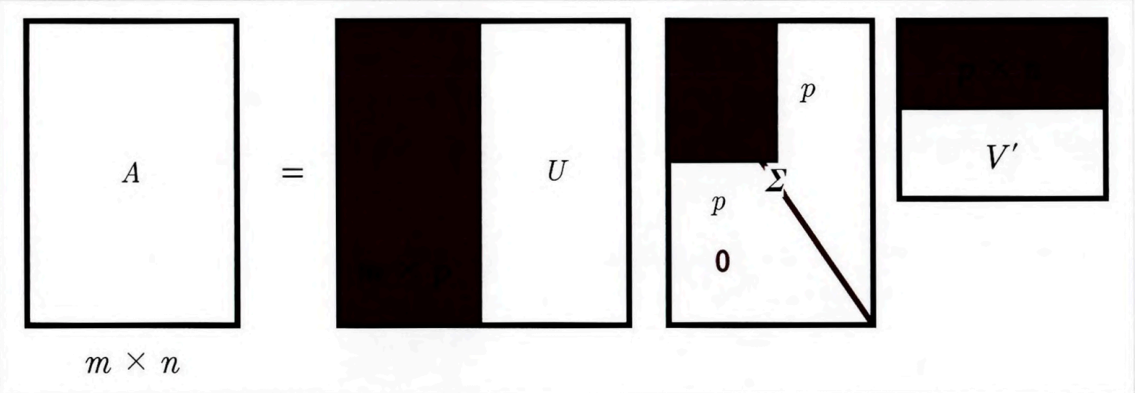
- 특이값 분해
- PDA와 유사한 행렬 분해 기법을 이용하지만, 행과 열의 크기가 다른 행렬에도 적용할 수 있다는 차이!
- 아래의 수식에서 행렬 U와 V에 속한 벡터가 특이벡터임

$$A = U \sum V^T$$

- 다음과 같이 행렬 A를 분해



특이값이 0인 부분 제거



사이킷런 TruncatedSVD 클래스를 이용한 변환

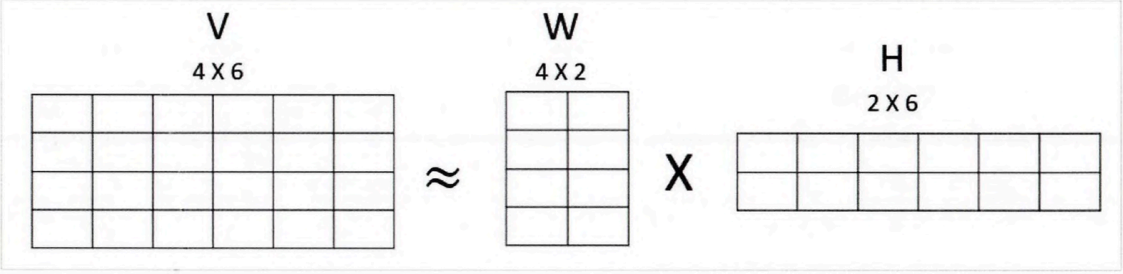
- 사이파이의 svds와 달리 U, sigma, Vt 행렬을 반환하지 않음
- fit()과 transform()을 호출 → 원본 데이터를 몇 개의 주요 컴포넌트로 차원 축소해 변형

05 NMF(Non-Negative Matrix Factorization)

NMF 개요

- Truncated SVD와 같이 낮은 랭크를 통한 행렬 근사 방식
- 원본 행렬 내 모든 원소 값이 모두 양수라는 보장 필요

- 4×6 원본 행렬V를 4x2 행렬W와 2x6 행렬H로 근사해 분해



일반적으로

행렬W은 원본 행렬과 행의 크기가 같고, 열의 크기가 작은

행렬 H은 원본 행렬보다 행의 크기가 작고, 열의 크기가 같은 형태로 분해됨

- 차원 축소를 통한 잠재 요소 도출로 이미지 변환 및 압축, 텍스트 토픽 도출 등의 영역에서 사용됨

06 정리

PCA	1. 입력 데이터의 변동성이 가장 큰 축을 구함 2. 구한 축에 직각인 축을 축소하려는 차원의 개수만큼 반복적으로 구함 3. 입력 데이터들을 이 축들에 투영해 차원 축소 입력 데이터의 공분산 행렬을 기반으로 고유 벡터를 생성하고, 이를 기반으로 입력 데이터를 선형 변환
LDA	PDA와 유사, 입력 데이터의 결정 값 클래스를 최대한으로 분리할 수 있는 축을 찾는 방식으로 차원 축소
SVD	많은 피처 데이터를 가진 고차원 행렬을 두 개의 저차원 행렬로 분리
NMF	많은 피처 데이터를 가진 고차원 행렬을 두 개의 저차원 행렬로 분리