차원축소

6.1 개요

- 매우 많은 피처로 구성된 다차원 데이터 세트의 차원을 축소해 새로운 차원의 데이터 세트를 생성하는 것

  👉 **차원이 증가할수록**

  데이터 포인트 간의 거리가 멀어지고 희소 (sparse) 한 거리 가지게 된다.

  적은 차원에서 학습된 모델이 예측 신뢰도 높다.

  👉 **피처가 많을수록**

  개별 피처 간에 상관관계가 높을 가능성이 크다.

  선형 모델에서는 입력 변수 간의 상관관계가 높으면 '다중 공선성 문제'로 모델의 예측 성능↓

6.1.1 차원을 축소하면 좋은 점?

- 직관적으로 데이터 해석 가능
- 학습 데이터의 크기가 줄어들어서 학습에 필요한 처리 능력도 줄일 수 있음

6.1.2 차원 축소 방법

1️⃣ 피처 선택 (feature selection)

- 특정 피처에 종속성이 강한 불필요한 피처는 제거하고 데이터의 특징을 잘 나타내는 주요 피처만 선택

2️⃣ 피처 추출 (feature extraction)

- 기존 피처를 저차원의 중요 피처로 압축해서 추출
- 새롭게 추출된 중요 특성은 기존의 피처와는 완전히 다른 값

- 단순 압축 x
- 함축적으로 더 잘 설명할 수 있는 또 다른 공간으로 매핑해 추출
- 기존 피처가 인지하기 어려웠던 잠재적인 요소를 추출
- e.g. 내신 성적, 수능 성적, 봉사활동, 대외활동, 수상경력 → 학업 성취도, 커뮤니케이션 능력, 문제해결력
- PCA, SVD, NMF 알고리즘

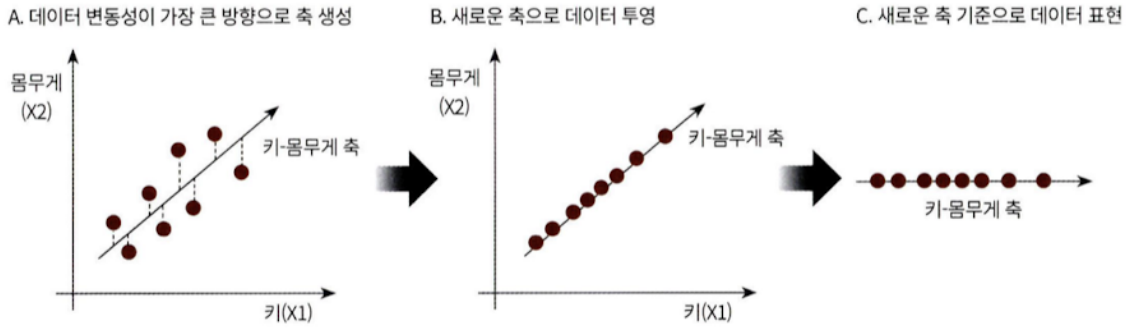
6.1.3 활용

- 이미지 변환과 압축
  - 이미지 분류 수행 시 과적합 영향력이 작아져서 예측성능 up
  - 텍스트 문서의 숨겨진 의미 추출 → 단어들의 구성에서 숨겨져 있는 semantic 의미나 topic을 잠재요소로 간주하고 찾아낼 수 있음 by. SVD, NMF

6.2 PCA (Principal Component Analysis)

6.2.1 개요

- 여러 변수 간에 존재하는 상관관계를 이요해 이를 대표하는 주성분을 추출해 차원을 축소하는 기법
- 기존 데이터의 정보 유실이 최소화
- 가장 높은 분산을 가지는 데이터의 축을 찾아 이 축으로 차원을 축소



- 가장 큰 데이터 변동성을 기반으로 첫 번째 벡터 축 생성

- 두 번째 축은 이 벡터 축에 직각이 되는 직교 벡터
- 세 번째 축은 두 번째 축과 직각이 되는 벡터

⇒ 원본 데이터의 피처 개수에 비해 작은 주성분으로 원본 데이터의 총 변동성 설명 가능

6.2.2 붓꽃 데이터 실습

1. 개별 속성 스케일링 (각 속성값을 동일한 스케일로 변환)
    - 사이킷런의 StandardScaler 이용
2. 4차원의 데이터를 2차원 PCA 데이터로 변환
    - PCA 클래스
    - fit, transform

6.3 LDA (Linear Discriminant Analysis)

6.3.1 개요

- 선형 판별 분석법
- 입력 데이터 세트를 저차원 공간에 투영해 차원을 축소 (PCA 와 유사)
- 분류(Classification) 에서 사용하기 쉽도록 개별 클래스를 분별할 수 있는 기준을 최대한 유지하면서 차원 축소
    - PCA : 데이터의 변동성의 가장 큰 축
    - LDA : 입력 데이터의 결정 클래스를 최대한 분리할 수 있는 축

6.3.2 클래스 분리 최대화

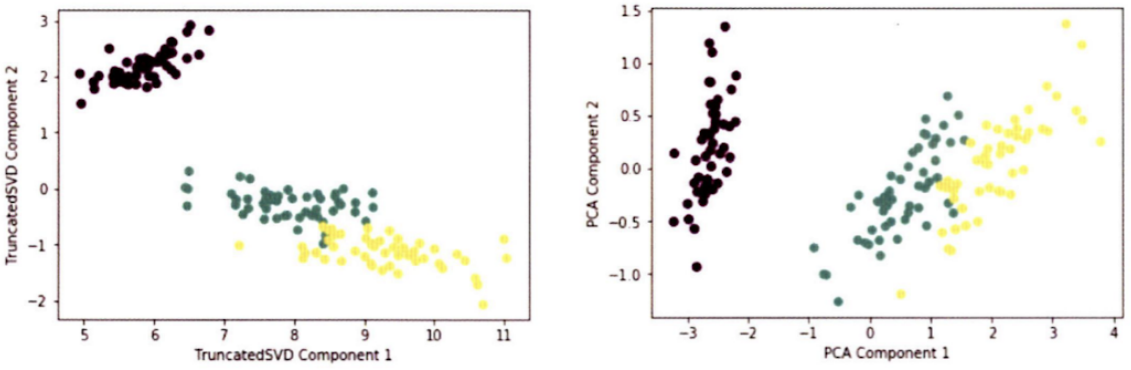
- 클래스 간 분산 (between-class scatter) 과 클래스 내부 분산(within-class scatter)의 비율을 최대화하는 방식으로 차원 축소
- 클래스 간 분산은 최대한 크게!
- 클래스 내부 분산은 최대한 작게!

6.4 SVD (Singular Value Decomposition)

6.4.1 개요

- 특이값 분해
- 정방행렬 뿐만 아니라 행과 열의 크기가 다른 행렬에도 적용 가능
- Truncated SVD 는 넘파이가 아닌 사이파이에서만 지원

6.4.2 Truncated SVD 와 PCA 비교



왼: TruncatedSVD 로 변환된 붓꽃 데이터 세트

오: PCA 로 변환된 붓꽃 데이터 세트

- PCA 와 유사하게 변환 후에 품종별로 클러스터링이 가능할 정도로 각 변호나 속성으로 뛰어난 고유성을 가지고 있음
- SVD 이용해 행렬 분해

⇒ 2개의 변환이 서로 동일

- PCA가 SVD 알고리즘으로 구현됨
- PCA는 밀집 행렬에 대한 변환만 가능
- SVD 는 희소 행렬에 대한 변환도 가능
- 컴퓨터 비전 영역에서 이미지 압출을 통한 패턴 인식과 신호 처리 분야에서 사용됨

6.5 NMF (Non-Negative Matrix Factorization)

6.5.1 개요

- Truncated SVD와 같이 낮은 랭크를 통한 행렬 근사 방식의 변형
- 원본 행렬 내의 모든 원소 값이 모두 양수라는 게 보장되면 좀 더 간단하게 두 개의 기반 양수 행렬로 분해 가능
- 사이킷런에서 NFM 클래스 이용해 지원됨

6.5.2 SVD와 비교

- SVD와 유사하게 차원 축소를 통한 잠재 요소 도출로 이미지 변환, 압축, 텍스트의 토픽 도출 등의 영역에서 사용됨

6.5.3 활용

- 이미지 압축을 통한 패턴 인식, 텍스트의 토픽 모델링 기법, 문서 유사도 및 클러스터링에 사용됨
- 영화 추천과 같은 영역에 적용 [Latent Factoring 기반 추천 방식]
- 사용자의 상품 평가 데이터 세트인 사용자-평가 순위 데이터 세트를 행렬 분해 기법을 통해 분해
- 사용자가 평가하지 않은 상품에 대한 잠재적인 요소를 추출
- 이를 통해 평가 순위 (Rating) 예측
- 높은 순위로 예측된 상품을 추천해주는 방식