파머완 6장: 차원 축소

상태	진행 중
담당자	 시현 이
마감일	@11/10/2025
작업 유형	개념정리 및 필사
설명	파이썬 머신러닝 완벽 가이드_개정2판_제6장 개념 정리+ 필사 링크
업데이트 시간	@November 8, 2025 1:04 AM

01. 차원 축소(Dimension Reduction)개요

차원축소

: 다차원 데이터세트의 차원을 축소해 새로운 차원의 데이터 세트를 생성하는 것

- 차원 증가할 수록, 데이터 포인트 간의 거리가 기하급수적으로 멀어짐 + 희소한 구조 가짐
  - 수백 개 이상의 피처로 구성된 데이터 세트의 경우, 상대적으로 적은 차원에서 학습된 모델보다 예측 신뢰도가 떨어짐
  - 피처 多 → 개별 피처 간의 상관관계 높음 ⇒ 선형 모델에서는 '다중공선성' 문제로 모델의 예측 성능이 저하됨
- 피처 수를 줄이면 더 직관적으로 데이터 해석 가능 + 학습에 필요한 처리 능력도 줄일 수 있음
- 피처 선택(feature selection)과 피처 추출(feature extraction)로 나눌 수 있음

1. **피처 선택**
   a. 특정 피처에 종속성이 강한 불필요한 피처는 아예 제거하고, 데이터의 특징을 잘 나타내는 주요 피처만 선택
2. **피처 추출**

a. 기존 피처를 저차원의 중요 피처로 압축해서 추출(기존 피처와는 완전 다른 값)
b. 피처를 함축적으로 더 잘 설명할 수 있는 또 다른 공간으로 매핑해 추출
c. 기존 피처가 전혀 인지하지 어려웠던 잠재적인 요소(latent factor)를 추출하는 것

- **차원 축소 알고리즘: PCA, LDA, SVD, NMF**
  - 매우 많은 픽셀로 이뤄진 이미지 데이터에서 잠재된 특성을 피처로 도출해 함축적 형태의 이미지 변환과 압축을 수행할 수 있음
    - 변환된 이미지는 적은 차원으로 이뤄져 이미지 분류 등의 분류 수행 시에 과적합 영향력이 작아져서 예측성능이 끌어올려짐→차원 수가 너무 크면 적은 픽셀의 차이가 잘못된 예측으로 이어질 수 있기 때문
  - 텍스트 문서의 숨겨진 의미를 추출할 때도 쓰임
    - 문서 내 단어들의 구성에서 숨겨져 있는 시맨틱(Semantic) 의미나 Topic을 잠재요소로 간주하고 찾아낼 수 있음
    - SVD와 NMF는 이러한 Semantic Topic 모델링을 위한 기반 알고리즘으로 사용됨

02. PCA(Principal Component Analysis)

PCA 개요

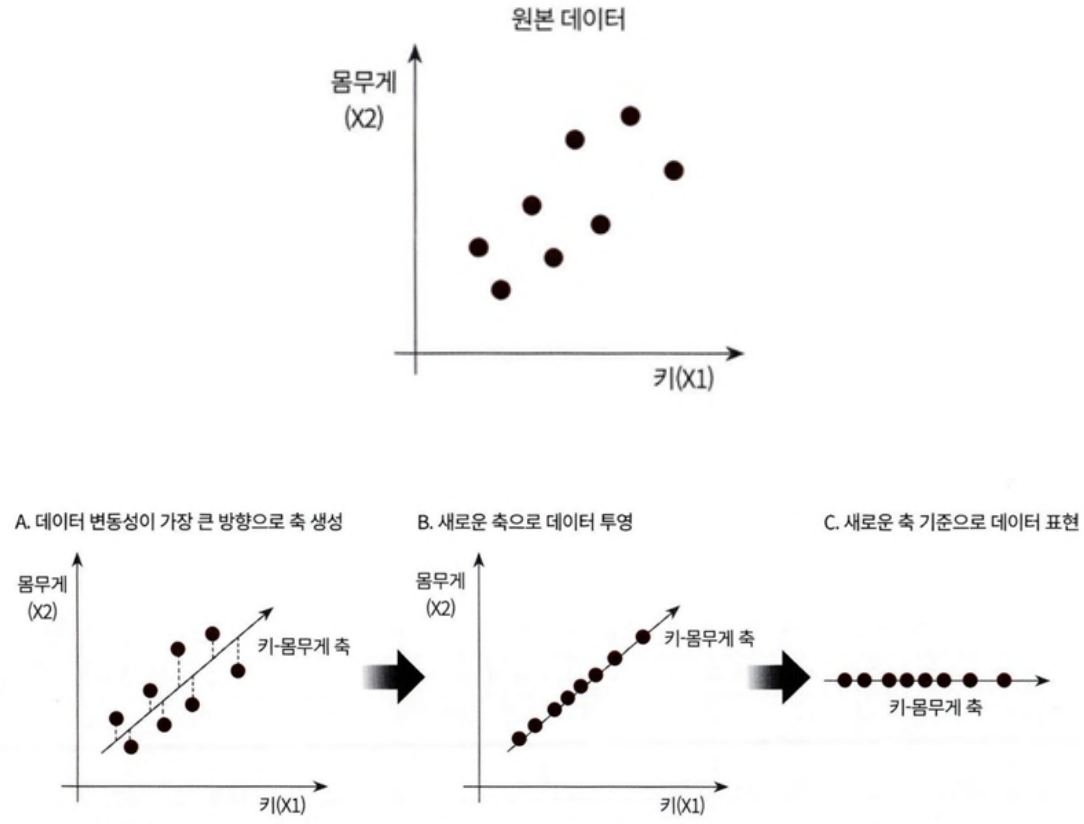
PCA(Principal Component Analysis)

:여러 변수 간에 존재하는 상관관계를 이용해 이를 대표하는 주성분(Principal Componene)을 추출해 차원을 축소하는 기법

:원본 데이터의 피처 개수에 비해 매우 작은 주성분으로 원본 데이터의 총 변동성을 대부분 설명할 수 있는 분석법

- 기존 데이터 정보 유실이 최소화
  - 가장 높은 분산을 가지는 데이터의 축을 찾아 이 축으로 차원을 축소⇒PCA의 주성분

ex) 키와 몸무게 feature 데이터 셋

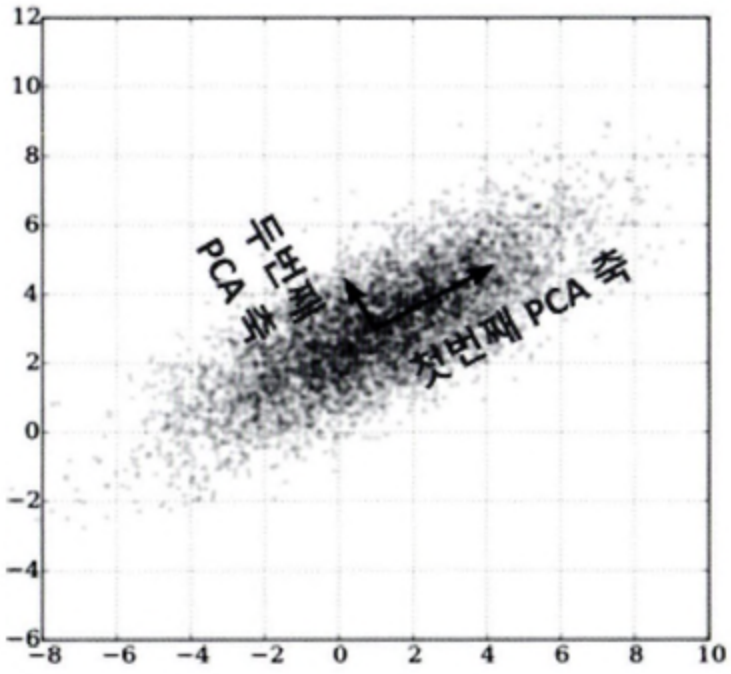


2개의 피처를 한 개의 주성분을 가진 데이터 세트로 차원 축소

[차원 축소 방법]

1. 가장 큰 데이터 변동성(Variance)을 기반으로 첫 번째 벡터 축 생성
2. 두 번째 축은 첫 번째 벡터 축에 직각이 되는 벡터를 축으로 함
3. 세 번째 축은 다시 두 번째 축과 직각이 되는 벡터를 설정

⇒생성된 벡터 축에 원본 데이터를 투영하면 벡터 축의 개수만큼의 차원으로 원본 데이터가 차원 축소됨



[선형대수 관점 해석]

입력 데이터의 공분산 행렬(Convariance Matrix)을 고유값 분해하고, 이렇게 구한 고유벡터에 입력 데이터를 선형 변환하는 것

→고유벡터가 PCA의 주성분 벡터로서 입력 데이터의 분산이 큰 방향을 나타냄

→고윳값은 이 고유벡터의 크기를 나타냄과 동시에 입력 데이터의 분산을 나타냄

▼ 선형대수 개념 정리

▼ 분산: 한개의 특정한 변수의 데이터 변동//공분산: 두 변수 간의 변동

▼ Cov(X,Y) >0 ⇒ X가 증가할 때, Y도 증가한다는 의미

▼ 공분산 행렬: 여러 변수와 관련된 공분산을 포함하는 정방형 행렬

	X	Y	Z
X	**3.0**	−0.71	−0.24
Y	−0.71	**4.5**	0.28
Z	−0.24	0.28	**0.91**

▼ 대각선 원소는 각 변수의 분산, 대각선 이외의 원소는 모든 변수 쌍 간의 공분산을 의미

▼ 고유벡터: 행렬 A를 곱하더라도 방향이 변하지 않고 그 크기만 변하는 벡터. 행렬을 분해하는데 쓰임

▼ 공분산 행렬 = 정방 행렬(n*n)이자 대칭 행렬(A(Transpose) = A). 개별 분산값을 대각 원소로하는 대칭 행렬

▼ 대칭 행렬: 항상 고유벡터를 직교 행렬로, 고유값을 정방행렬로 대각화할 수 있음

▼ 공분산 행렬 C라고 하면, 다음과 같이 분해

$$C = P \sum P^T$$

▼ P = n*n 직교행렬, Sigma = n*n 정방행렬, P(Transpose)

▼ C는 고유벡터 직교행렬*고유값 정방 행렬*고유벡터 직교 행렬의 전치행렬로 분해됨

$$C = [e_1 \cdots e_n] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} e_1^t \\ \cdots \\ e_n^t \end{bmatrix}$$

⇒ 입력 데이터의 공분산 행렬이 고유벡터와 고유값으로 분해될 수 있으며, 이렇게 분해된 고유벡터를 이용해 입력 데이터를 선형 변환하는 방식이 PCA

[PCA 수행 과정]

1. 입력 데이터 세트의 공분산 행렬을 생성
2. 공분산 행렬의 고유벡터와 고유값을 계산
3. 고유값이 가장 큰 순으로 K개(PCA 변환 차수)만큼 고유벡터를 추출
4. 고유값이 가장 큰 순으로 추출된 고유벡터를 이용해 새롭게 입력 데이터를 변환

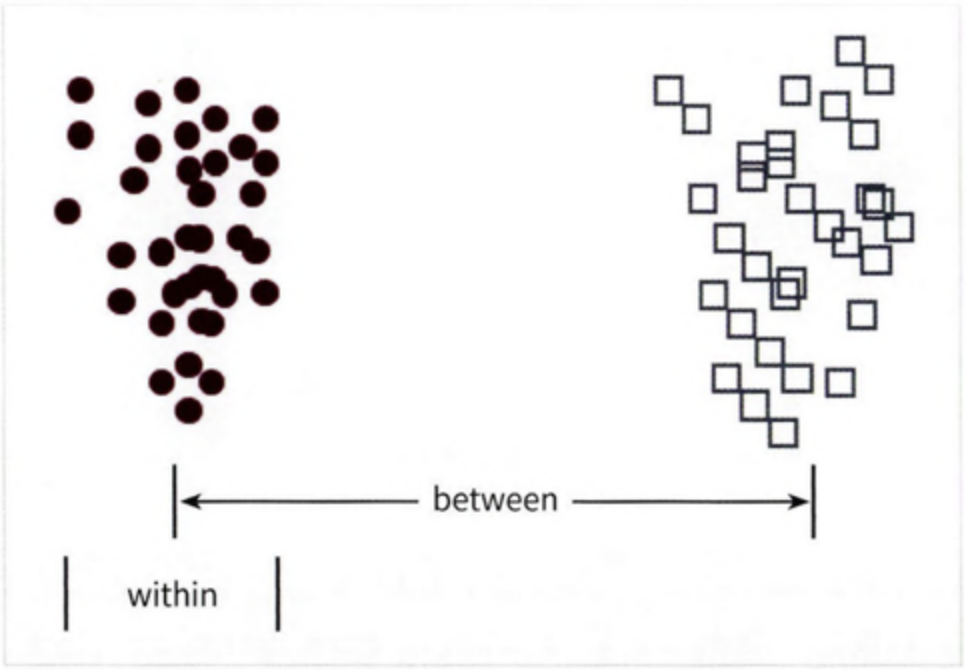
03.LDA(Linear Discriminant Analysis)

LDA 개요

LDA(Linear Discriminant Analysis) = 선형 판별 분석법

:PCA와 유사하게 입력 데이터 세트를 저차원 공간에 투영해 차원을 축소하는 기법. LDA는 지도학습의 분류(Classification)에서 사용하기 쉽도록 개별 클래스를 분별할 수 있는 기준을 최대한 유지하면서 차원을 축소함

- 입력 데이터의 결정 값 클래스를 최대한으로 분리할 수 있는 축을 찾음
- 특정 공간상에서 클래스 분리를 최대화하는 축을 찾기 위해 클래스 간 분산(between-class scatter)과 클래스 내부 분산(within-class scatter)의 비율을 최대화하는 방식으로 차원을 축소
    - 클래스 간 분산은 최대한 크게 가져가고, 클래스 내부 분산은 최대한 작게 가져가는 방식



- 클래스 간 분산과 클래스 내부 분산 행렬을 생성한 뒤, 이 행렬에 기반해 고유벡터를 구하고 입력 데이터를 투영함

[LDA 구하는 과정]

1. 클래스 내부와 클래스 간 분산 행렬을 구함. 이 두 개의 행렬은 입력 데이터의 결정 값 클래스별로 개별 피처의 평균 벡터를 기반으로 구함
2. 클래스 내부 분산 행렬을 Sw, 클래스 간 분산행렬을 S_B라고 하면 다음 식으로 두 행렬을 고유벡터로 분해

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

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

04.SVD(Singular Value Decomposition)

SVD 개요

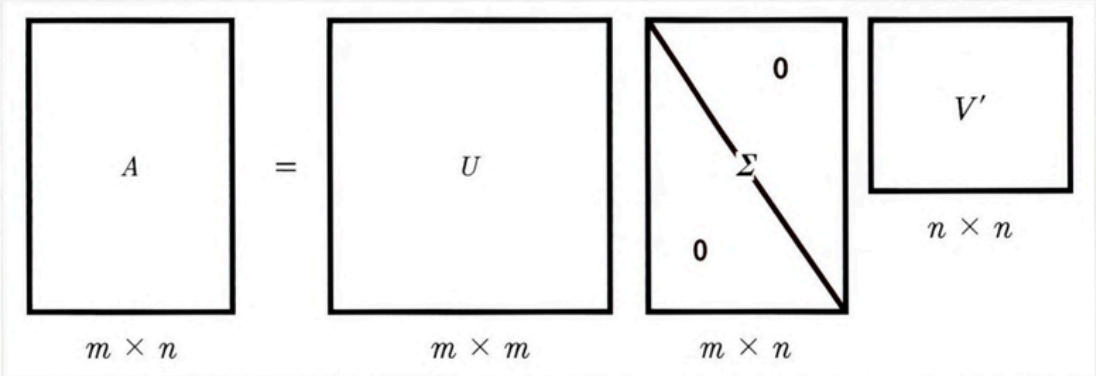
SVD(Singular Value Decomposition) = 특이값 분해

정방행렬뿐만 아니라 행과 열의 크기가 다른 행렬에도 적용 가능

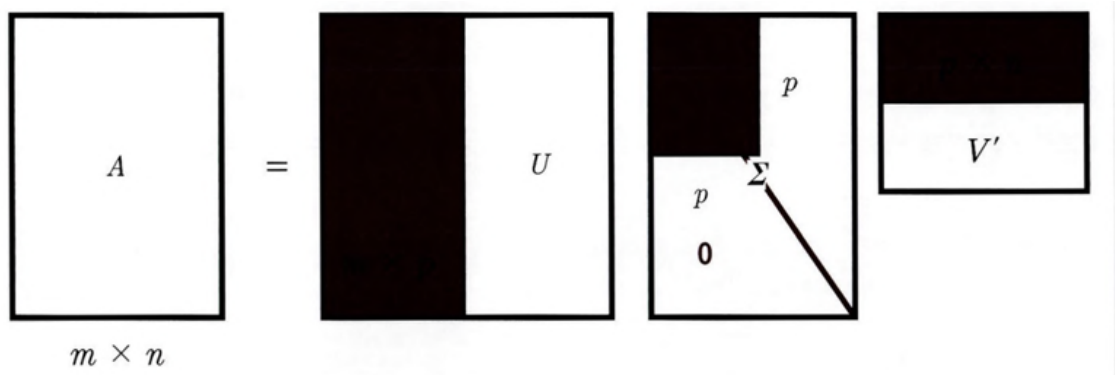
일반적으로 m*n 크기의 행렬 A를 다음과 같이 분해

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

- 행렬 U(m*m)와 V(n*n)에 속한 벡터는 특이벡터(singular vector)(서로 직교하는 성질 지님)
- Σ(m*n)는 대각 행렬. 0이 아닌 값이 행렬 A의 특이값



- 일반적으로 다음과 같이 Σ의 비대각인 부분과 대각 원소 중에 특이값이 0인 부분도 모두 제거하고 Σ에 대응되는 U와 V원소도 함께 제거해 차원을 줄인 형태로 SVD를 적용



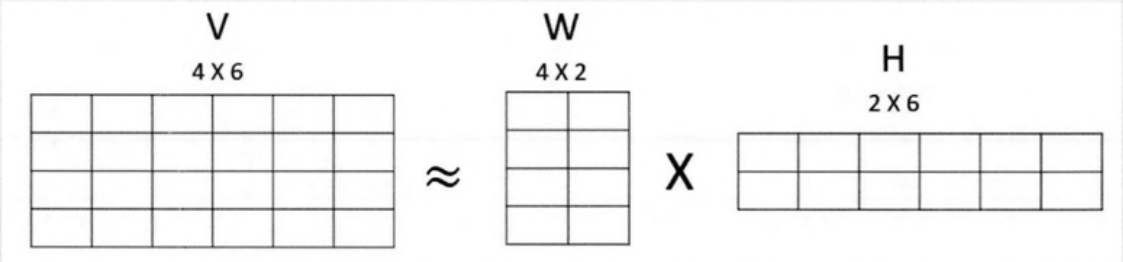
- 이렇게 컴팩트하게 SVD 적용 시, A 차원이 m*n일 때, U의 차원 m*p, Σ의 차원 p*p, V^T의 차원 p*n으로 분해됨
- Truncated SVD
  - Σ의 대각 원소 중에 상위 몇개만 추출해서 여기에 대응하는 U와 V의 원소도 함께 제거해 더욱 차원을 줄인 형태로 분해하는 것

05.NMF(Non-Negative Matrix Factorization)

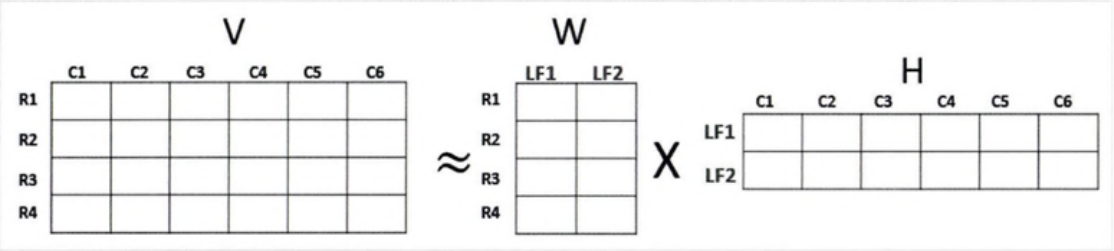
NMF 개요

NMF는 Truncated SVD와 같이 낮은 랭크를 통한 행렬 근사 방식의 변형

:원본 행렬 내의 모든 원소 값이 모두 양수라는게 보장되면 다음과 같이 좀 더 간단하게 두개의 기반 양수 행렬로 분해될 수 있는 기법을 지칭



- 분해된 행렬은 잠재요소를 특성으로 가지게 됨.
- W는 원본 행에 대해 이 잠재 요소 값이 얼마나 되는지에 대응하며, H는 이 잠재요소가 원본 열(원본 속성)로 어떻게 구성됐는지 나타냄



- NMF도 SVD와 유사하게 이미지 압축을 통한 패턴인식, 텍스트의 토픽 모델링 기법, 문법 유사도 및 클러스터링에 잘 사용됨
- 영화 추천과 같은 추천 영역에서도 활발하게 적용됨
    - 사용자의 상품 평가 데이터세트인 사용자-평가 순위 데이터 세트를 행렬 분해하면서 사용자가 평가하지 않은 상품에 대한 잠재적인 요소를 추출해 이를 통해 평가 순위를 예측하고, 높은 순위로 예측된 상품을 추천해주는 방식(=잠재요소 기반의 추천방식)

06.정리

PCA는 입력 데이터의 변동성이 가장 큰 축을 구하고, 다시 이 축에 직각인 축을 반복적으로 축소하려는 차원 개수만큼 구한 뒤 입력 데이터를 이 축들에 투영해 차원을 축소하는 방식.

LDA는 PCA와 유사하며, 입력 데이터의 결정 값 클래스를 최대한으로 분리할 수 있는 축을 찾는 방식으로 차원을 축소

SVD와 **NMF**는 매우 많은 피처 데이터를 가진 고차원 행렬을 두 개의 저차원 행렬로 분리하는 행렬 분해 기법.

지원 파일