DA_예습과제 7주차_박보영

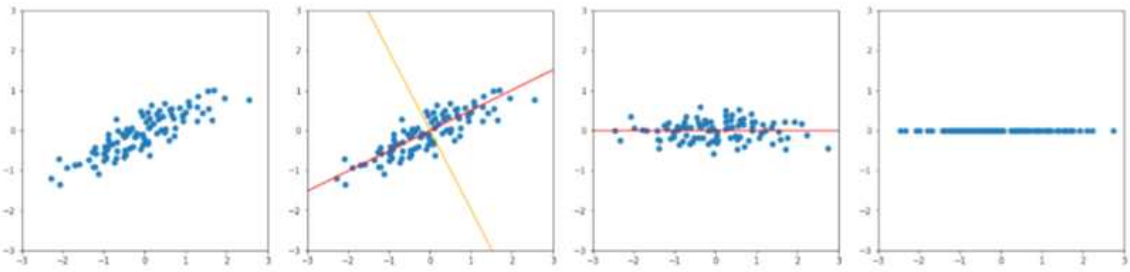
<차원축소>
매우 많은 피처로 구성된 다차원 데이터 세트의 차원을 축소해 새로운 차원의 데이터 세트를 생성하는 것. 차원이 증가할수록 희소한 구조. 학습에 필요한 처리능력도 줄일 수 있음.

차원 축소는 피처 선택, 피처 추출로 나눌 수 있다.

'피처 선택'은 불필요한 피처는 제거하고, 주요 피처만 선택하는 것.
'피처 추출'은 기존 피처를 저차원의 중요 피처로 압축해서 추출하는 것. 기존과는 완전히 다른 값. 피처를 함축적으로 더 잘 설명할 수 있는 또 다른 공간으로 매핑해 (잠재적인 요소)를 추출하는 것. ex. PCA, SVD, NMF

차원 축소 알고리즘은 이미지 데이터에서 잠재된 특성을 피처로 도출해 함축적 형태의 이미지 변화과 압축을 수행할 수 있다. 과적합의 영향력이 작아져서 예측 성능을 올릴 수 있음. 텍스트 문서의 숨겨진 의미를 추출 할 수 있다. 문서 작성자의 의미나 의도를 인지할 수 있다. semantic topic

<PCA>
가장 높은 분산을 가지는 데이터의 축을 찾아 이 축을 차원으로 축소한다. 가장 큰 데이터 변동성을 기반으로 첫 번째 벡터축 생성, 수직이 되는 두 번째 벡터축 생성.



선대 관점에서 입력데이터의 공분산 행력을 고유값 분해하고, 이렇게 구한 고유벡터에 입력데이터를 선형변환하는 것. 고유벡터가 PCA주성분 벡터로서 입력데이터의 분산이 큰 방향 나타냄. 고유값은 고유벡터의 크기이며 입력 데이터의 분산을 나타나냄.m

공분산 행렬은 diagonal, symmetric matrix.

C=PQP^T

순서
1.입력 데이터 세트의 공분산 행렬을 생성
2.공분산 행렬의 P, Q계산
3.고유값이 가장 큰 순으로 k개 e.vector 추출
4.e.value 큰 순으로 추출된 e.vector를 이용해 새롭게 입력 데이터를 변환.

PCA의 가정

1. Linearity
: 데이터가 선형성을 띈다
2. Orthogonality
: 찾은 주축들은 서로 직교한다.
3. 큰 분산을 갖는 방향이 중요한 정보를 담고 있다.

PCA의 장점

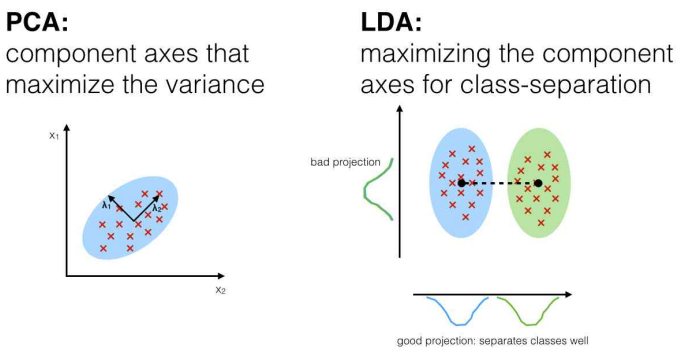
1. 변수 간 상관관계 및 연관성을 이용해 변수를 생성한다.
2. 차원 축소로 인한 차원의 저주를 해결한다.
3. 다중공선성의 문제를 해결한다.

PCA의 단점

1. 데이터가 선형성을 띄지 않으면 적용할 수 없다.

<LDA>
linear discriminant analysis 선형 판별 분석법.
지도학습의 '분류'에서 사용하기 쉽도록 개별 클래스를 분별할 수 있는 기준을 최대한 유지하면서 차원을 축소.

분리를 최대화하는 축을 찾기 위해 클래스 간 분산과 클래스 내부 분산 between/within 비율을 최대화하는 방식 사용.(크게/작게)



순서

1.클래스내부와 클래스간 분산 행렬을 구함. 이 두 개의 행렬은 입력데이터의 결정 값 클래스별로 개별 피처의 평균 벡터를 기반으로 구함.

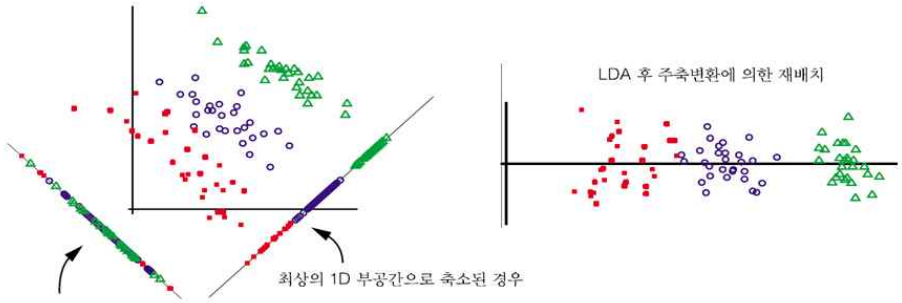
2.클래스내부분산 행렬을 Sw, 간 Sb라고 하면...Sw^tSb=PQP^t

3.고유값이 큰 순으로 k개(LDA변환 차수만큼) 추출

4.고유값이 큰 순으로 추출된 고유벡터를 이용해 새롭게 입력 데이터를 변환

❖ 선형판별분석에 의한 차원 축소

- **주성분 분석법(PCA)은 데이터의 최적 표현의 견지에서 데이터를 축소하는 방법인데 반하여, 선형판별분석법(LDA)은 데이터의 최적 분류의 견지에서 데이터를 축소하는 방법**

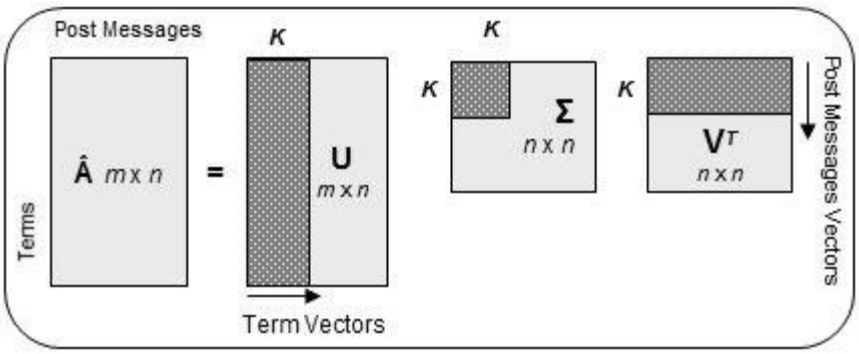


[그림 11-2] LDA에 의한 차원 축소

- **LDA의 목적:가능한 클래스간의 분별 정보를 최대한 유지시키면서 차원을 축소시키는 것**

<SVD>

mXn행렬에서도 가능. $A = U\sum V^T$

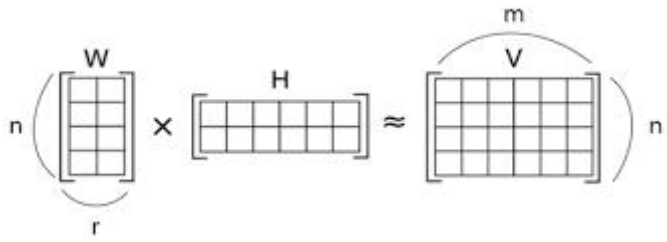


차원을 분해.

TruncatedSVD class를 이용한 변환

from sklearn.decomposition import TruncatedSVD
fit(), transform(), n_components=int 주요 컴포넌트 개수 파라미터,

<NMF>
truncatedSVD와 같이 낮은 랭크를 통한 행렬근사방식의 변형. element>0,



nmf=NMF(n_components=2)
fit(), transform()