REPORT

: PCA와 SVD 비교 분석

과목명 지식발견

담당교수 차광호

학과 산업정보시스템전공

학번 17101937

이름 김도현

제출일 20.06.07

우리가 현실에서 사용하는 데이터는 많은 속성(attribute)을 가진 여러 개의 대상(object)로 이루어진 고차원 데이터셋(high-dimensional dataset)이다. 데이터 셋의 차원이 높을수록 관련 없는 속성의 수가 증가하게 되고, 이와 같이 관련 없는 속성은 노이지한 구성요소(noisy component)일 가능성이 크기 때문에 차원을 축소시키는 과정이 중요 해진다. 또한, 모델의 학습을 느리게 하고 정교한 모델을 만들기 어렵게 하기 때문에 차원의 저주(curse of dimensionality)가 일어난다.

지도, 하얀색이(가) 표시된 사진

자동 생성된 설명

(Source: *Curse of Dimensionality of DeepAI*)

• 차원의 저주

* 차원이 증가하면서, 학습데이터 수가 차원의 수 보다 적어져 성능이 저하되는 현상
* 차원이 증가할수록 각각의 차원 내에서 학습할 데이터 수가 적어지는(sparse) 현상

위의 그림에서 1D에서 3D로 차원이 증가함에 따라, 점들 사이에 공간이 많이 비었다는 것을 알 수 있다. 빈 공간은 정보가 없다는 것을 나타냄으로 고차원에서 모델의 성능이 저하되는 현상(차원의 저주)은 당연한 것이다. 이를 해결하기 위해 우리는 PCA 알고리즘을 통해 차원을 축소시킨다.

PCA(주성분 분석, Principal Component Analysis)는 데이터 분산을 최대한 보존하면서 서로 직교(orthogonal)하는 축을 찾아서 초평면에 투영시키는 방법이다. 기존의 속성보다 적은 수의 새로운 변수들을 설정하는 ‘feature extraction’ 방식으로, 지도학습(Supervised learning) 또는 비지도 학습(Unsupervised)을 통해서 진행된다. ‘feature extraction’의 장점은 로딩 매트릭스를 제공하기 때문에, 새로운 데이터가 들어와도 로딩 매트릭스를 통해 낮은 차원으로 바로 바꿔줄 수 있다는 것이다.

PCA는 선형방법(Linear method)로 표현하게 되는데, 이는 새로 생성된 변수들이 기존 변수들의 선형결합으로 표현가능 한 것을 의미한다. 이렇게 표현된 여러 축 위에 데이터를 정사영 시키고, 정사영 된 데이터들의 분산이 가장 커지는 축을 선택하게 된다.

• PCA 방법(Source: *지식발견[PCA]*)

1. Get some data
2. Conduct centering or standardization
3. Calculate the covariance matrix
4. Calculate the eigenvectors & eigenvalues of the covariance matrix
5. Choosing components and forming a feature vector

(centering이 이미 수행되었다는 가정에서 공분산 행렬 구하기)

대각성분이 공분산행렬의 고유 값이고 나머지 성분은 0인 행렬

열벡터가 의 고유벡터로 이루어진 행렬

여기서Λ의 대각성분의 행렬A의 각 변수에 해당하는 분산을 의미한다. 이를 통해, 가장 큰 고유 값을 일부 선택해서, 이에 해당하는 고유벡터를 새로운 기저로 하여 데이터를 선형변환해주면 PCA 작업을 완료하게 된다.

그러나 단순히 각 축의 최대 분산을 구하기 때문에 클래스 간의 분류 용이성(separability)가 고려되지 않는 문제점이 있다. 이는 최대의 분산의 각 축이 클래스를 잘 구별할 수 있는 속성(attribute)을 뽑아 준다는 보장이 없다는 것을 나타낸다. 이런 이유로, 차원 축소에서 PCA 보다 더 일반적으로 쓰이는 방법이 SVD이다.

SVD 원래의 데이터 매트릭스를 재구성하는 것이 목표가 아니라, 속성들을 잘 표현할 수 있는 더 낮은 차원을 찾는 것이 목표이다. 이를 위해서 서로 직교하는 벡터들의 집합에 대하여, 선형 변환한 후에 크기는 변하지만 계속 직교할 수 있는 집합을 찾게 된다.

A= UΣ

=

U: m by m Orthogonal matrix

Σ: m by n Diagonal matrix

V: n by n Orthogonal matrix

위의 관계식에 따라, SVD는 정방행렬 만이 아니라 모든 m by n 행렬에 대해서 적용이 가능하다. SVD의 차원을 감소시키는 과정에서 유사한 항목(similar item)은 거리가 더 가까워지도록, 유사하지 않은 항목(unlike item)은 거리가 멀어지도록 만들어준다.

분산이 큰 것에 대해서만 대각행렬(diagonal matrix) 통해 차원 축소를 시켜서, 가장 의미 있는 상관 관계를 나타내는 요소만 포함하게 되므로 노이즈(noise)를 효과적으로 제거할 수 있게 된다. SVD는 3개의 매트릭스(U, Σ,)로 분해된 A행렬을 상위 특이 값 p개 만으로 복원할 수 있는데, 이는 단 몇 개의 특이 값 몇 개 만으로 데이터의 유용한 정보를 유지할 수 있다는 것을 의미한다.

부분 복원이란 SVD처럼 몇 개의 특이 값 만으로 복원이 가능한 것을 의미한다. 사진 분야에서 부분 복원이 많이 쓰이게 되는데, 부분 복원을 통해 사진의 용량을 줄이면서 기존의 사진이 보여주던 내용을 살릴 수 있게 된다. 이 외에도 원본 행렬의 잠재 요소를 추출하기 때문에 토픽 모델링과 같은 추천 시스템에 활발하게 사용된다.

• PCA와 SVD의 공통된 목적

* 상관된 변수(correlated variables)를 원래의 데이터 항목들 간의 관계를 더 잘 나타내는 상관되지 않는 변수(uncorrelated ones)들의 집합으로 바꾼다.
* 데이터 분산이 큰 순서대로 차원들을 정렬시킨다.
* 원래의 데이터를 가장 잘 표현하는 더 낮은 차원을 이용해서 데이터를 축소시킨다.

차원 축소 방법들 중 하나인 PCA와 SVD를 사용함에 따라, 원래의 데이터를 더 잘 표현하는 부분만 이용하게 되어서 차원이 증가하면서 성능이 저하되었던 문제점을 해결할 수 있게 된다. 또한, 데이터들 사이에 공간이 많이 비어 학습 데이터 수가 적었던 현상에 대해서도, 초평면에 데이터를 투영시킴으로써 spars data 현상을 보완할 수 있게 되었다.

참고자료

* 지식발견[PCA.pdf]
* 지식발견[Singular Value Decomposition.pdf]
* [DeepAI: The front page of A.I.] <https://deepai.org/>