**고급소프트웨어실습1 - 7주차 과제**

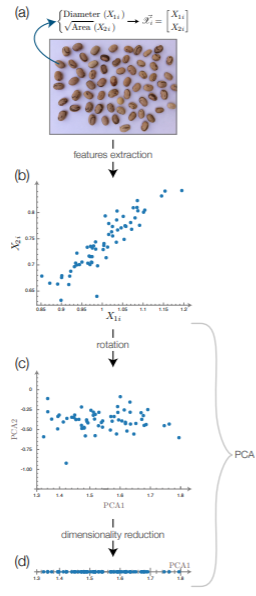
Principal Component Analysis

컴퓨터공학과

20181593

계인혜

1. **Eigenvector의 개수는 원래 데이터의 차원(dimension)과 같으므로 모든 eigenvector를 이용하여 원래 데이터를 투영하면 데이터 압축의 효과를 볼 수 없다. 또한 데이터의 분산 특성을 고려하지 않고 특정 몇 개의 eigenvector를 이용하여 투영시키면 데이터의 유용한 정보가 손실될 수 있다. 유용한 정보를 손실하지 않으면서 최대로 압축 효과를 얻을 수 있도록 eigenvector의 개수를 설정할 수 있는 방법이 있는지 설명해 보자.**

[[1]](#endnote-1)

PCA(Principal Component Analysis)란 데이터 집합을 분석하는 기법 중 하나로, 데이터의 좌표 축을 변환시켜 각 좌표축에 대하여 원래 데이터를 사상시켰을 때 분산 값이 커지는 순으로 서로 독립된 좌표계를 구하여 원래의 데이터를 변환시키는 방법이다.

PCA는 데이터의 차원을 낮춰 데이터의 압축 효과를 가져온다. 그러나, 이 과정에서 데이터 손실이 발생한다. 이를 해결하기 위해 압축에 사용할 Eigenvector의 수를 늘리게 되면, 압축된 데이터와 원 데이터와의 유사도가 높아지고 정보 손실 가능성은 줄어들지만 압축 효과는 감소한다. 또한 더 많은 데이터를 저장해야 하고 작업량도 늘어나게 된다. 한편, Eigenvector 수를 줄이면 데이터 압축률은 커지지만 데이터 손상 정도가 크고, 정보 손실 가능성이 커진다.

일반적으로 각 샘플 차원의 분산과 Eigenvalue는 비례하기 때문에 Eigenvector를 정렬할 때, Eigenvalue를 기준으로 내림차순으로 정렬하게 된다. 즉, Eigenvalue값이 큰 Eigenvector는 분산이 크고, 데이터 사이에 변화량이 크다는 것이다. 따라서 Eigenvalue값이 큰 Eigenvector가 데이터 복구에 유용하게 사용될 수 있다. 반면에 Eigenvalue값이 작은 Eigenvector는 작은 노이즈 정도로 취급되기 때문에 압축 시 무시할 수 있다. 이렇게 하면 유용한 정보를 최대한 손실시키지 않고 압축 효과를 얻을 수 있다.

Eigenvalue값이 작은 Eigenvector를 제외하는 방법으로는 다음과 같은 것들이 있다.

1. Eigenvalue 값의 평균보다 작은 Eigenvalue 값을 갖는 Eigenvector을 제외한다.
2. 원 데이터 차원수의 상위 몇 퍼센트 만큼의 Eigenvector를 선택한다.
3. 원 데이터 차원수의 제곱근 만큼의 Eigenvector를 선택한다.

다음 식은 optimal한 eigenvalue의 개수를 찾는데 이용되는 식이다.

(K : 선택한 Eigenvalue의 개수, N : Eigenvalue의 전체 개수)

1 - {Sum(lambda(i)) | i from 1 to K} / {Sum(lambda(i))| i from 1 to N}

이 값의 기준은 절대적으로 정해지는 것이 아니라 프로그램을 다루는 사람이 선택하는 것이다.

1. Felipe L. GewersGustavo R. Ferreira,2 Henrique F. de Arruda,3 Filipi N. Silva,1, 4 Cesar H. Comin,5 Diego R. Amancio,3, 4 and Luciano da F. Costa11. (2018). Principal Component Analysis: A Natural Approach to Data Exploration . [↑](#endnote-ref-1)