주성분 분석

학부연구생 이현재

차원 축소 (Dimension Reduction)

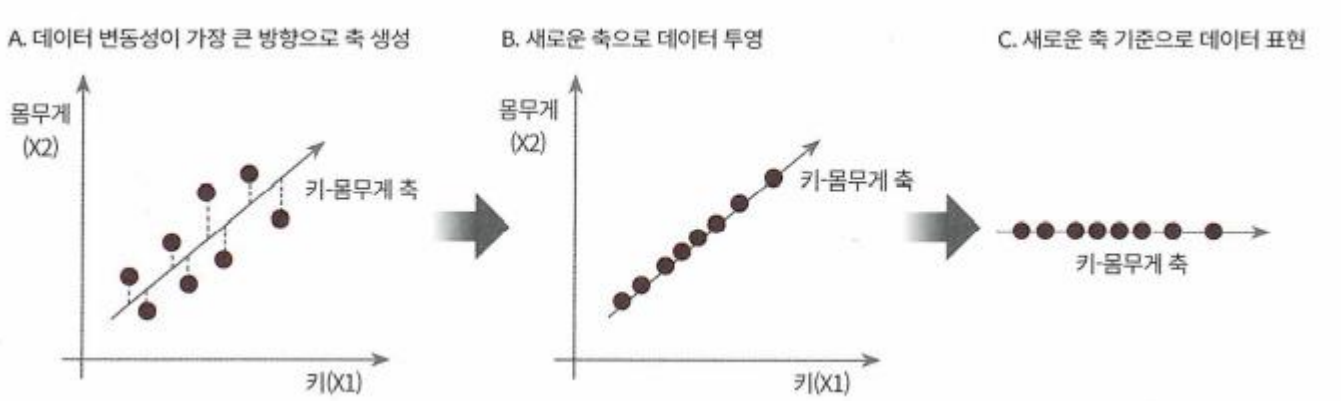
- 정의 : 매우 많은 피처로 구성된 다차원 데이터 세트의 차원을 축소해 새로운 차원의 데이터를 생성하는 것
- 차원 축소를 하는 이유
  - 시각화 : 3차원이 넘어간 시각화는 우리가 볼 수 없기 때문에 차원 축소를 통해 우리가 볼 수 있게 시각화
  - 노이즈 제거 : 쓸모 없는 피처들을 제거해 노이즈 제거
  - 메모리 절약 : 쓸모 없는 피처들을 제거해 메모리 절약
  - 성능 향상 : 차원 축소를 할 경우 학습 데이터의 크기가 줄어들어서 학습에 필요한 처리 능력을 줄일 수 있음
  - **다중공선성 해결** : 상관 관계가 높은 피처들을 제거하여 다중공선성 해결
  - **과적합 해결** : 피처 수가 많으면 과적합 발생 가능성 증가하기 때문에 피처 수 감소
- 차원 축소 접근법
  - 피처 선택(feature selection) : 특정 피처에 종속성이 강한 불필요한 피처는 아예 제거하고, 데이터의 특징을 잘 나타내는 주요 피처만 선택
  - 피처 추출(feature extraction) : 기존 피처를 저차원의 중요 피처로 압축해서 추출, 이렇게 새롭게 추출된 중요 특성은 기존의 피처가 압축된 것이므로 기존의 피처와는 다른 값이 됨
    → 기존 피처가 전혀 인지하기 어려웠던 잠재적인 요소를 추출(예는 슬라이드 설명에서 확인)
  - 피처 생성(feature engineering) : 피처가 부족한 상황일 때 적용하는 기법, 해당 데이터와 만들고자 하는 머신러닝 모델의 기능 활용 목적에 따라 새로운 피처들을 생성

- 차원 축소는 단순히 데이터의 압축을 의미하는 것이 아니라 **차원 축소를 통해 좀 더 데이터를 잘 설명해줄 수 있는 잠재적인 요소를 추출**
- 매우 많은 픽셀로 이뤄진 이미지 데이터에서 잠재된 특성을 피처로 도출해 함축적 형태의 이미지 변환과 압축을 수행 가능(이후에 SVD와 연관 설명)

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

주성분 분석 (PCA, Principal Components Analysis)

- 정의 : 여러 변수 간에 존재하는 상관관계를 이용해 이를 대표하는 주성분을 추출해 차원을 축소하는 기법
- 기존 데이터의 정보 유실을 최소화하기 위해서 **가장 높은 분산을 가지는** 데이터의 축을 찾아 이 축으로 차원을 축소(데이터의 **변동성이 가장 큰** 방향으로 축을 생성하고, 이 축으로 데이터를 **투영**)



- PCA를 이해하기 위해서 알아야 할 선형대수학 개념
  - 공분산 : 두 변수 간의 변동(0보다 크면 a가 증가할 때 b 증가, 공분산 행렬은 정방 행렬 및 대칭 행렬)
  - (A에 대한)고유값 분해 : $A = P\Lambda P^{-1}$ P는 고유 벡터들을 열벡터로 하는 행렬, Λ는 고유값을 대각원소로 가지는 대각 행렬
  - 고유벡터 : 행렬 A를 곱하더라도 방향이 변하지 않고 그 크기만 변하는 벡터 $A\mathbf{x} = \lambda\mathbf{x}$
  - 고유값 : 고유벡터의 크기
  - 대칭 행렬 : 정사각행렬 A와 A의 전치행렬이 같은 경우 $A = A^T$ **(대칭행렬은 모두 고유값 분해가 가능하며 항상 고유벡터를 직교행렬로, 고유값을 정방 행렬로 대각화 가능)**
  - 정방 행렬 : 같은 수의 행과 열을 가지는 행렬 (n*n)
  - 선형 변환 : 행렬 X에 다른 행렬 A를 곱해줌으로써 공간 A에 행렬 X를 맵핑(mapping/ **투영, 사상**) 시켜주는 개념

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근

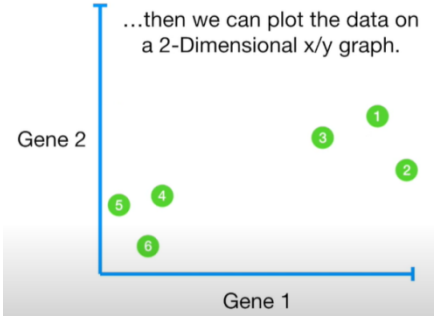
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2
Gene 4	5	7	6	2	4	7

6마리의 서로 다른 쥐에서 유전자 1, 2, 3, 4라는 4개의 유전자를 측정

1 차원

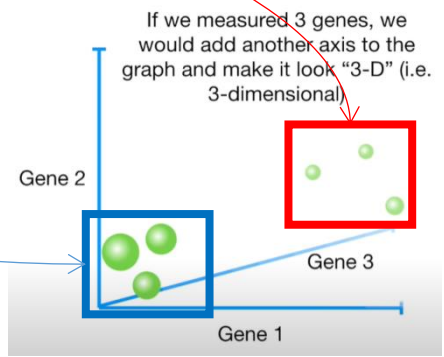
쥐 1, 2, 3은 다른 쥐 4, 5,6 보다 서로 더 비슷

2 차원



쥐 1, 2, 3 군집이 오른쪽 상단에 위치하고
쥐 4, 5, 6 군집은 왼쪽 하단에 위치

3 차원



작은 점들은 유전자 3에 대해 더 큰 값을 가지고 멀리 떨어져 있고, 더 큰 점들은 더 작은 값을 가지고 가까이 있음

4 차원 4차원이기 때문에 데이터를 시각화 할 수 없음

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

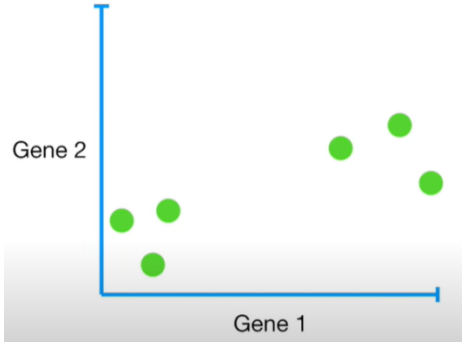
PCA 의 기하학 관점에서의 접근

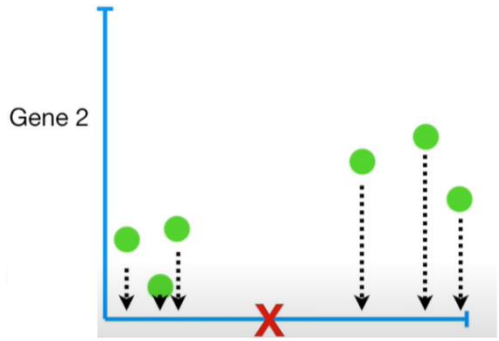
1. 데이터의 중심 구하기

<PCA의 과정을 알아보기 위해서 2개의 유전자만으로 가정>

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

Plot

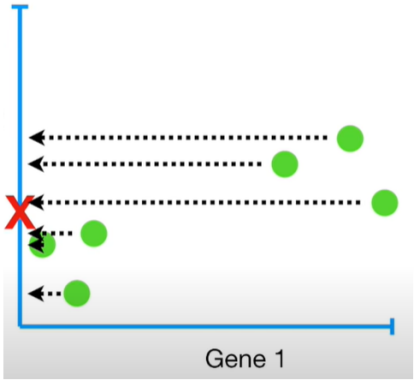




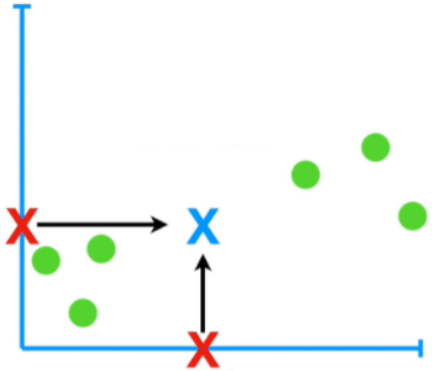
Gene 1

유전자 1에 대한 평균값 측정

Gene 2



유전자 2에 대한 평균값 측정

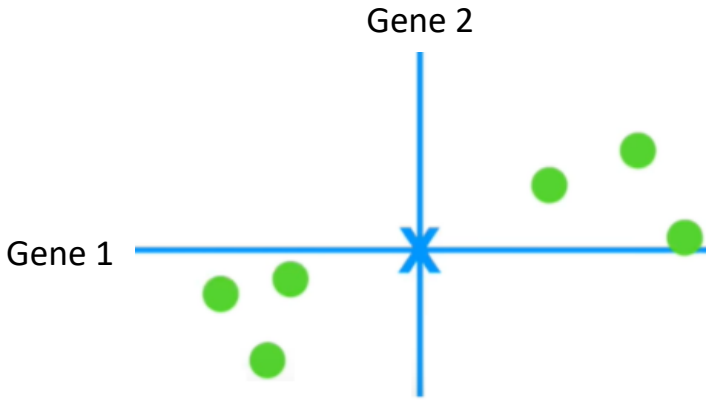


데이터의 중심 구하기

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

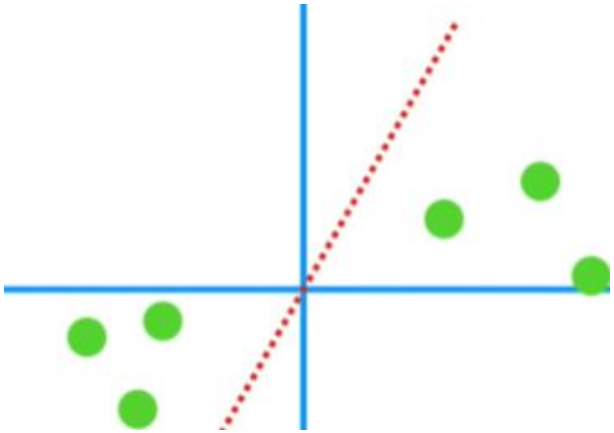
PCA 의 기하학 관점에서의 접근

2. 데이터의 중심을 원점으로 이동하기



데이터의 이동이 데이터 점들의
상대적인 분포를 변화시키지 않음

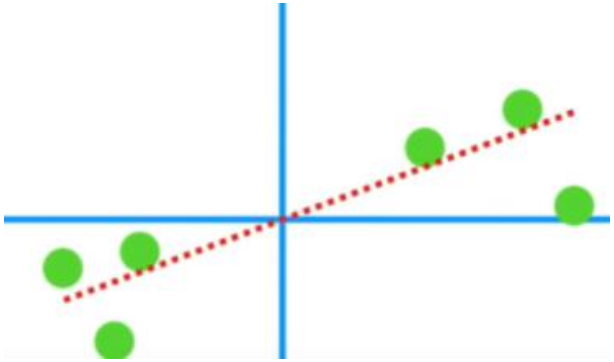
3. 원점을 지나는 직선을 Random하게 그리기



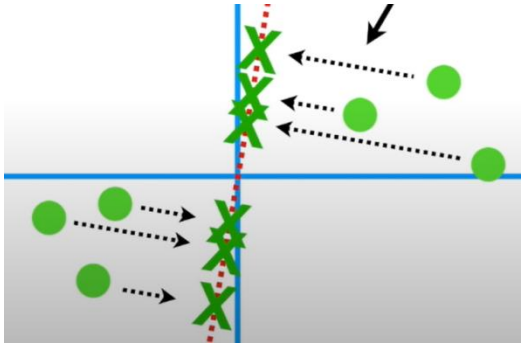
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근

4. 주어진 데이터에 가장 fit하도록 원점을 지나는 직선을 회전

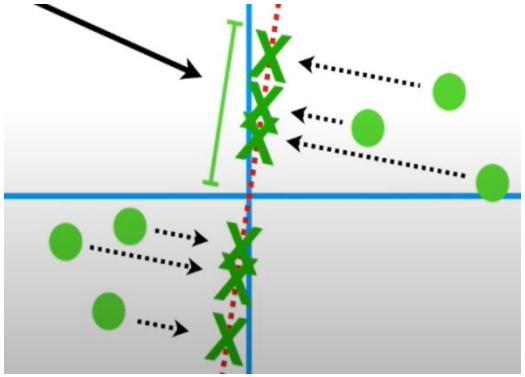


Fit함을 판단하는 방법



선에서 데이터까지의 거리를 측정하고
이 거리를 최소화하는 선 찾기

=

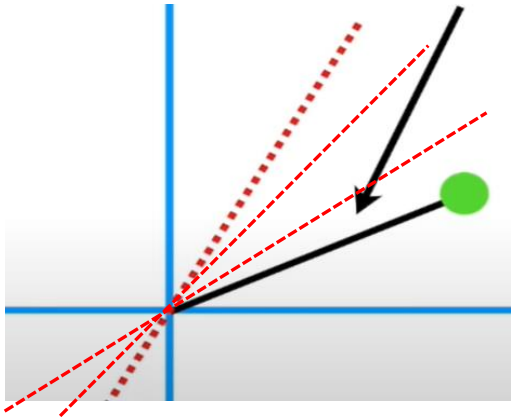


원점에서 투영된 점들까지의 거리를
최대화하는 선 찾기

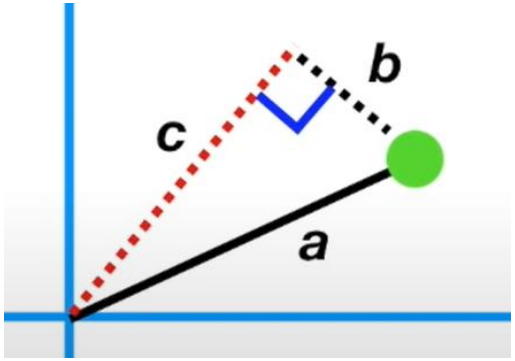
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근

선에서 데이터까지의 거리 최소화 = 투영된 점까지의 거리 최대화의 이유



이 그림에서 초록색 점에서 원점까지의 거리는 빨간색 점선이 아무리 회전해도 변하지 않음을 확인 할 수 있다.(검은 선)



$$a^2 = b^2 + c^2$$

a : 점에서 원점까지의 거리, b : 점과 직선까지의 최소 거리,
c : 직선으로 투영한 점과 원점까지의 거리

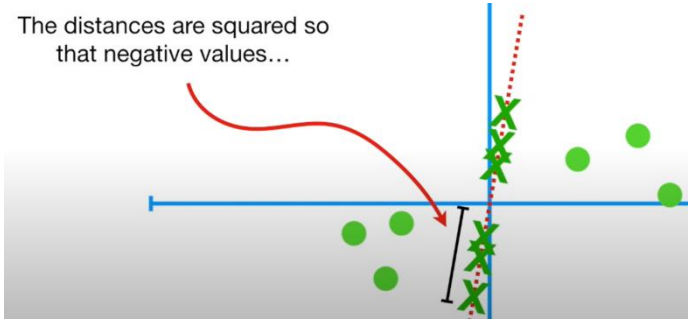
피타고라스 정리에 의해서 b와 c는 서로 트레이드오프인 관계 확인
(FIT해질수록 c는 커지고 b는 작아짐)

원점에서 투영된 점까지의 거리인 c를 계산하는 것이 더 쉽기 때문에 우리는 후자를 사용

⇒ PCA가 가장 적합한 선을 찾는 방법은 **원점에서부터 투영된 점들 간의 거리(c) 제곱의 합을 최대화**하는 것

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근



제곱을 해주는 이유 :
거리가 제곱이 되면서 음수 값은 양수 값을 없애지 못하게 하기 위해

$$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2 = \text{sum of squared distances} = SS(\text{distances})$$

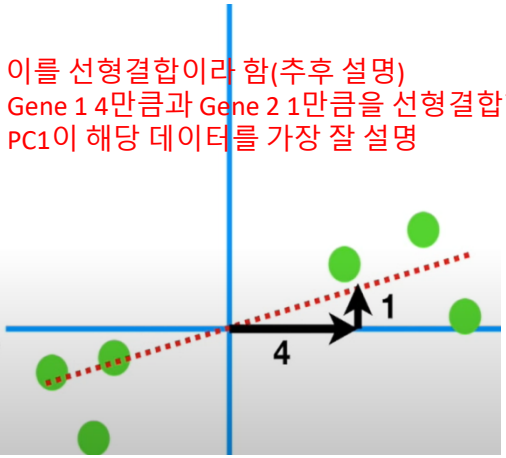
(d는 앞에서 구한 c를 의미)

이 SS가 최대가 되면 직선이 데이터에 Fit => Variance가 최대 => Variance가 최대여야 정보의 손실을 최소화

To make PC1
Mix **4** parts Gene 1
with **1** part Gene 2

⇒ 이를 선형결합이라 함(추후 설명)
Gene 1 4만큼과 Gene 2 1만큼을 선형결합한 PC1이 해당 데이터를 가장 잘 설명

The ratio of Gene 1 to Gene 2 tells you that Gene 1 is more important when it comes to describing how the data are spread out..



이 선을 Principal Component 1(PC1)이라고 함
이 PC1은 0.25라는 기울기를 가짐
(Gene 1으로 4단위, Gene 2로 1단위)

Gene 1 축을 따라서 대부분의 데이터가 분산,
아주 적은 데이터가 Gene 2 축을 따라 분산

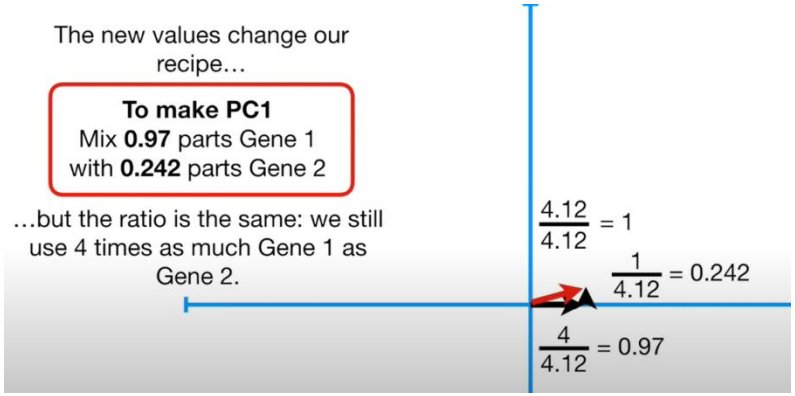
데이터가 어떻게 분산되어 있는지 설명할 때 Gene 1이 더 중요함

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

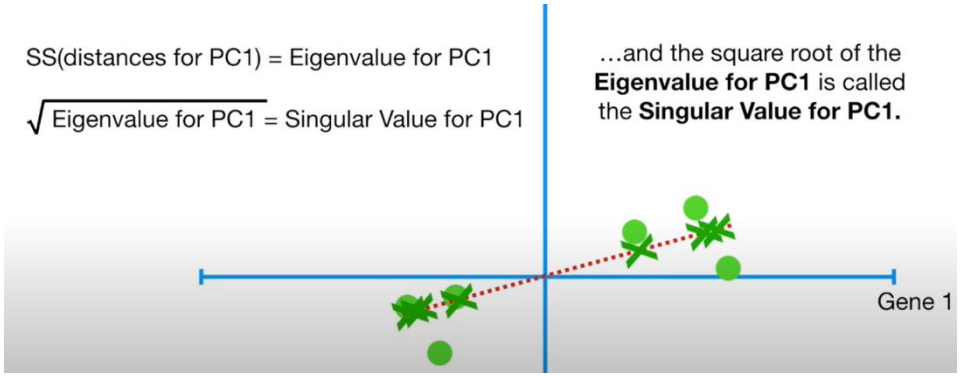
PCA 의 기하학 관점에서의 접근

5. 스케일링을 통해 고유벡터 만들기

피타고라스 정리에 의해 빗변은 4.12가 됨. 이 4.12를 1로 스케일링해줘야 함.



스케일링 결과, 0.97만큼의 Gene 1, 0.242만큼 Gene 2를 믹스하면 1만큼의 단위 벡터가 나온다. 이때 빨간색의 크기 1인 단위 벡터를 PC1의 **단일 벡터** 혹은 **고유 벡터**라고 함.
각각의 유전자 비율(Gene 1 = 0.97)을 Loading score(적재점수)라고 함 => Gene 1이 Gene 2보다 4배 중요



PC1 의 고유값 = SS

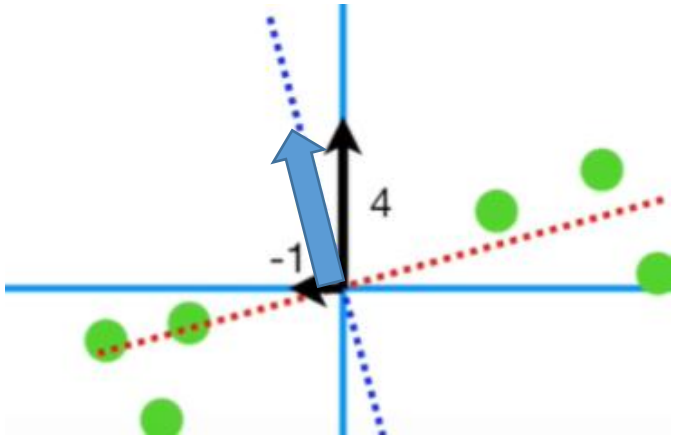
PC1의 특이값= $\sqrt{\text{고유값}}$

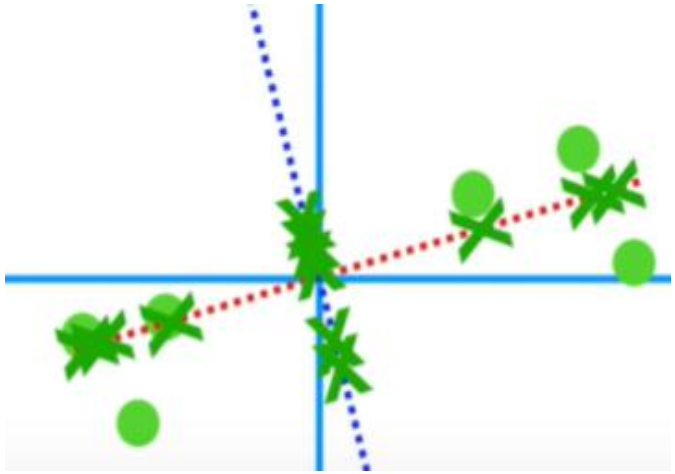
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근

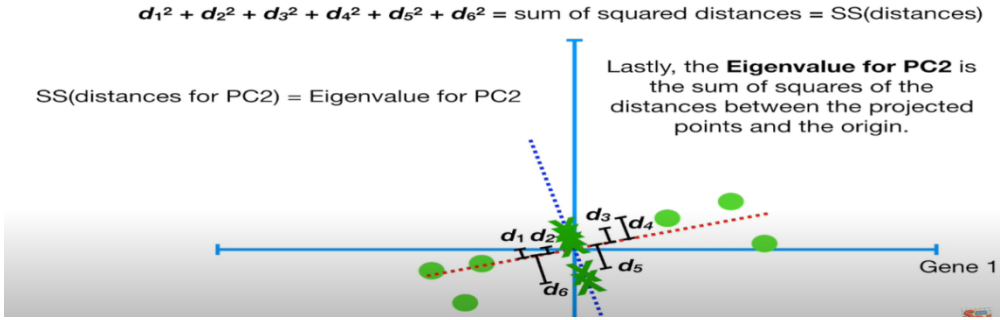
6. PC2도 동일한 방법으로 진행

PC2는 원점을 지나며 PC1에 수직인 직선. PC1의 기울기가 0.25였으므로 PC2의 기울기는 -4이다.(서로 수직인 기울기의 곱은 -1)





PC1과 마찬가지로 PC2도 스케일링
PC2에서의 Loading Score도 계산할 수 있는데, Gene 1은 -0.242, Gene 2는 0.97
=> PC1과는 다르게 Gene 2가 Gene 1보다 4배 중요

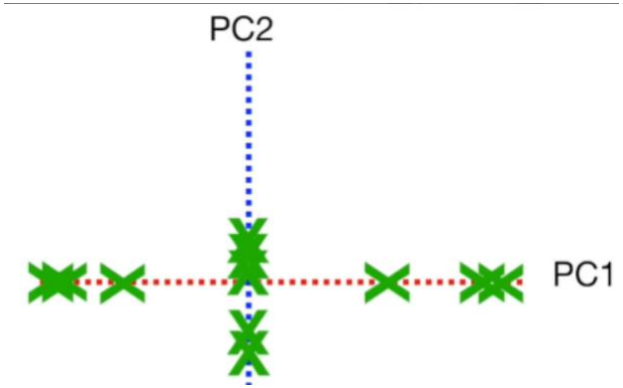


PC1과 PC2에 대해서 사상시킨 점들은 다음과 같다.

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

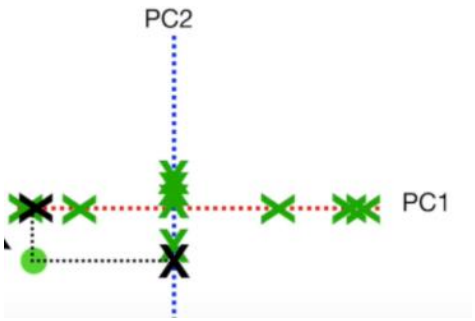
PCA 의 기하학 관점에서의 접근

7. PC축 회전

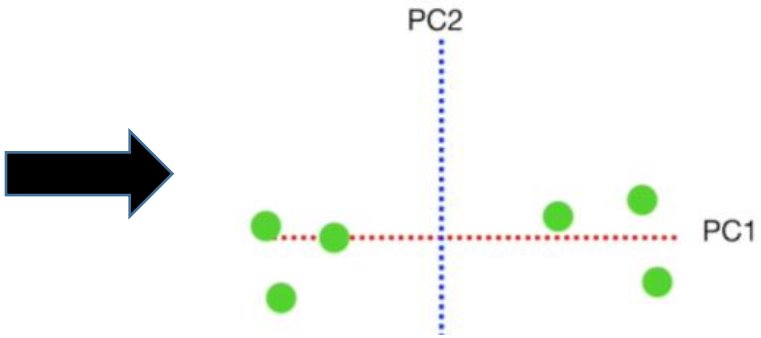


PC1이 x축에 수평이 되게 데이터를 회전

8. 데이터 복원



원 데이터를 구하기 위해서
사상된 점을 활용



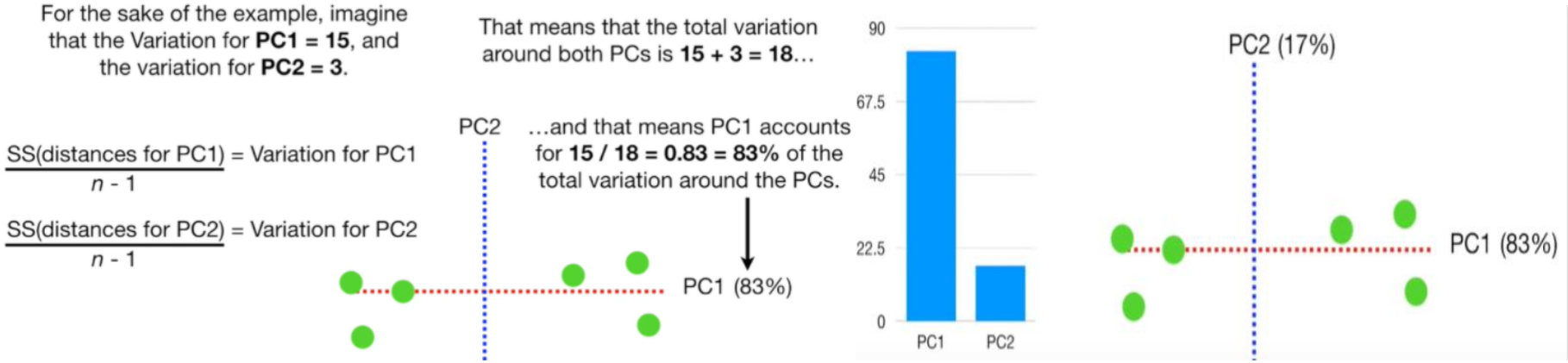
6개의 모든 점에 대해 동일하게 작업

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근

9. Variation

이제 PC1과 PC2가 해당 데이터를 얼마나 잘 설명하는지 알아야 함. 각 SS를 n-1로 나누어 Variation을 구할 수 있음. 식에 의하면 PC1의 Variation은 15, PC2의 Variation은 3이고 전체 Variation은 18이다. PC1은 15/18=0.83=83%만큼 설명 가능하고 PC2의 Variation은 전체의 17%가 된다. 각 PC의 Variation에 대한 그래프를 Scree Plot이라고 한다.



4차원일때 PCA는 이후에 코드에서 확인하겠습니다.

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 기하학 관점에서의 접근

지금까지 한 과정(기하학 관점에서의 접근)을 정리하면,

1) 수많은 직선 중, 데이터들을 직선에 투영했을 때 데이터가 최대한 겹치지 않고 멀리 퍼지는(분산이 큰) 직선 찾기
2) 거기에 데이터들을 투영
3) 만약 또 하나의 직선을 만들 때는 기존 직선에 수직

이 과정을 선형 대수학 관점에서 접근해보자면,

1) 입력데이터의 **공분산 행렬**에서 **고유벡터**와 **고유값**을 구한다
2) 가장 **분산이 큰** 방향을 가진 **고유벡터**(e1)에 입력데이터를 **선형변환**(이 고유벡터가 PCA의 주성분벡터로서 입력 데이터의 분산이 큰 방향을 나타냄, 고유값은 이 고유벡터의 크기이며 입력데이터의 분산을 나타냄)
3) 고유벡터(e1)과 **직교**하며, 다음으로 **분산이 큰 고유벡터**(e2)에 또 **선형변환**

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

PCA 의 선형대수학 관점에서의 접근

입력데이터의 공분산 행렬을 C라고 했을 때, 공분산 행렬의 특성으로 다음과 같이 고유값 분해가 가능
(헷갈리면 P.3으로)

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

이때 P는 n*n 직교행렬, ∑는 n*n 정방행렬, P^T는 행렬 P의 전치행렬

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

공분산 행렬 C는 고유벡터 직교 행렬 * 고유값 정방 행렬 * 고유벡터 직교 행렬의 전치 행렬로 분해
e_i는 i번째 고유벡터를, λ_i는 i번째 고유벡터의 크기, e_1는 가장 분산이 큰 방향을 가진 고유벡터,
e_2는 e_1에 수직이면서 다음으로 분산이 가장 큰 방향을 가진 고유 벡터

선형대수학의 PCA :입력데이터의 공분산 행렬이 고유벡터와 고유값으로 분해될 수 있으며,
이렇게 분해된 고유벡터를 이용해 입력 데이터를 선형 변환하는 방식

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

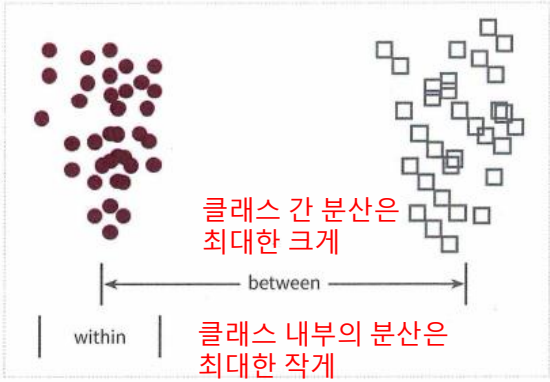
=> 즉, **고유벡터**들이 아까 기하학적 관점에서의 우리가 그토록 찾던 **분산이 가장 큰 직선**이다.

PCA 실습 코드 :

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

LDA (선형 판별 분석법, Linear Discriminant Analysis)

- 정의 : PCA와 유사하게 입력 데이터 세트를 저차원 공간에 투영해 차원을 축소하는 기법
  - PCA 와의 차이점
    - LDA는 지도학습의 분류에서 사용하기 쉽도록 개별 클래스를 분별할 수 있는 기준을 최대한 유지하면서 차원을 축소
    - LDA는 입력 데이터의 결정 값 클래스를 최대한으로 분리할 수 있는 축을 찾음
- 클래스 간 분산은 최대한 크게 가져가고, 클래스 내부의 분산은 최대한 작게 가져가는 방식



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

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

붓꽃 데이터 세트에 LDA 적용하기 :

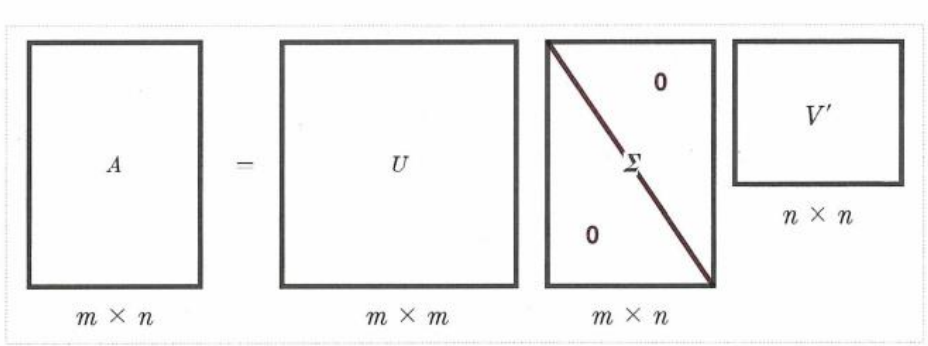
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

SVD (특이값 분해, Singular Value Decomposition)

- 정의 : PCA와 유사한 행렬 분해 기법 → PCA의 경우 정방행렬만을 고유벡터로 분해할 수 있지만,
  SVD는 정방행렬뿐만 아니라 행과 열의 크기가 다른 행렬에도 적용 가능

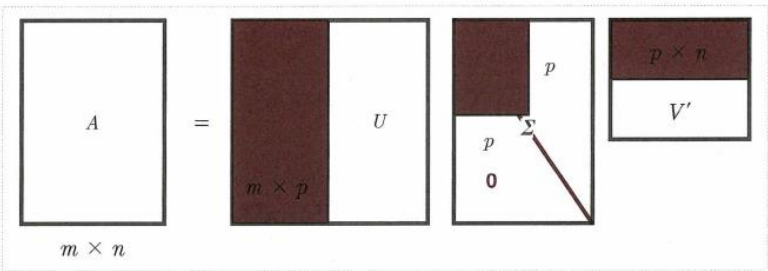
$$A = U \sum V^T$$

- 행렬 U와 V에 속한 벡터는 특이벡터(singular vector)이며, 모든 특이 벡터는 서로 직교하는 성질,
  ∑는 대각행렬이고 ∑이 위치한 0이 아닌 값이 바로 행렬 A의 특이값



U: $m \times m$ 직교행렬 $(AA^T = U(\Sigma\Sigma^T)U^T)$
V: $n \times n$ 직교행렬 $(A^TA = V(\Sigma^T\Sigma)V^T)$
Σ: $m \times n$ 직사각 대각행렬

- Truncated SVD : ∑의 대각원소 중에 상위 몇 개만 추출해서 여기에 대응하는 U와 V의 원소도 함께
  제거해 더욱 차원을 줄인 형태로 분해(연산을 줄이기 위해서 일반적으로 사용)



실습 코드 :

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

참고 자료

- 파이썬 머신러닝 완벽 가이드
- Pytorch로 시작하는 딥러닝 입문
- 귀퉁이 서재 블로그(https://bkshin.tistory.com/)
- 공돌이의 수학노트(https://angeloyeo.github.io/)
- ★ StatQuest: Principal Component Analysis 주성분 분석 (PCA), 스텝 바이 스텝 ★ (https://www.youtube.com/watch?v=FgakZw6K1QQ)