3. 고윳값

1. 고윳값과 고유벡터

> 💡 $Av = \lambda v$ (단, v는 0이 아님)
>
> 선형 변환 시 공간이 바뀌지 않는 벡터.

- 특성다항식

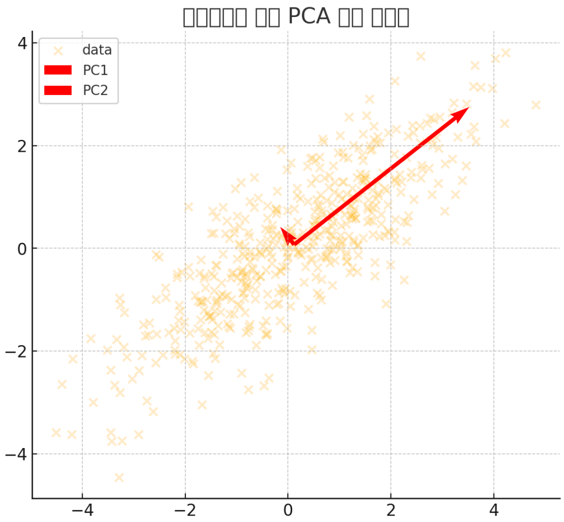
  $$f(\lambda) = \det(\lambda I - A)$$

- 특성방정식

  $$f(\lambda) = \det(\lambda I - A) = 0$$

🖥️ 활용

- PCA, 추천 시스템, 스펙트럼 기반 알고리즘
    - 알고리즘에서 “고유벡터 = 방향성”, “고윳값 = 중요도”로 작용함.
    - 고유 벡터 = 선형변환을 통해 방향이 변하지 않는 벡터 = 데이터의 중요한 성질을 담고 있다.
    - 고윳값 = 해당 고유 벡터의 방향의 가중치로, 중요한 정보를 많이 담고 있을 수록 커진다.



2. 행렬의 대각화

$P^{-1}AP = D$ (D는 대각행렬) 를 만족하면, 행렬 A는 대각화가 가능하다고 한다.

- 행렬 연산량을 줄이고 직관성을 높이기 위함
- P는 고유벡터, D는 고윳값과 같다
- 특성방정식의 해가 중근인 경우, 대각화가 불가능한 경우도 있다

활용

- 반복되는 행렬 연산을 간단히 정리.
- PCA에서 고유벡터와 고윳값을 찾는데 활용.

3. 대칭행렬의 대각화

대칭행렬 A는 정규직교행렬 Q가 존재하여 항상 대각화가 가능하다.

$A = Q^T DQ$

💡 **유사행렬** : 임의의 정칙행렬 P에 대하여 행렬 A와 PAP^{-1}, 또는 $P^{-1}AP$를 유사행렬이라고 한다. (PAP^{-1} 또는 $P^{-1}AP$는 대각행렬이 아닐 수도 있음)

💡 유사행렬을 사용하면,

1. 유사행렬은 **원래 행렬과 유사한 성질**을 갖는다. (유사행렬은 서로 같은 고유값을 갖는다.)
2. 유사행렬로 변환하면 **계산이 쉬운 형태로 변환**할 수 있다.
3. **대칭행렬은 유사 행렬로 변환하지 않아도 계산을 편리**하게 할 수 있다.

▼ 고유값, 고유벡터, 유사 행렬 계산 코드

```
import numpy as np

A = np.array([[4, 1],
              [0, 2]])

eigvals, P = np.linalg.eig(A) # eigvals : 고유값, P : 고유벡터
D = np.diag(eigvals) # D: 고유값 대각행렬
P_inv = np.linalg.inv(P)

# 유사행렬 관계 확인
Lambda = P_inv @ A @ P
print("유사행렬 :", Lambda)
```

4. 양의 정부호 행렬

모든 n차원 벡터에 대해,

양의 정부호 행렬 (PDM) : $x^TAx > 0$ 을 만족하는 A

양의 준정부호 행렬 (PSDM) : $x^TAx \geq 0$ 을 만족하는 A

- 양의 정부호 행렬이나 준정부호 행렬은 계산이 굉장히 안정적임.

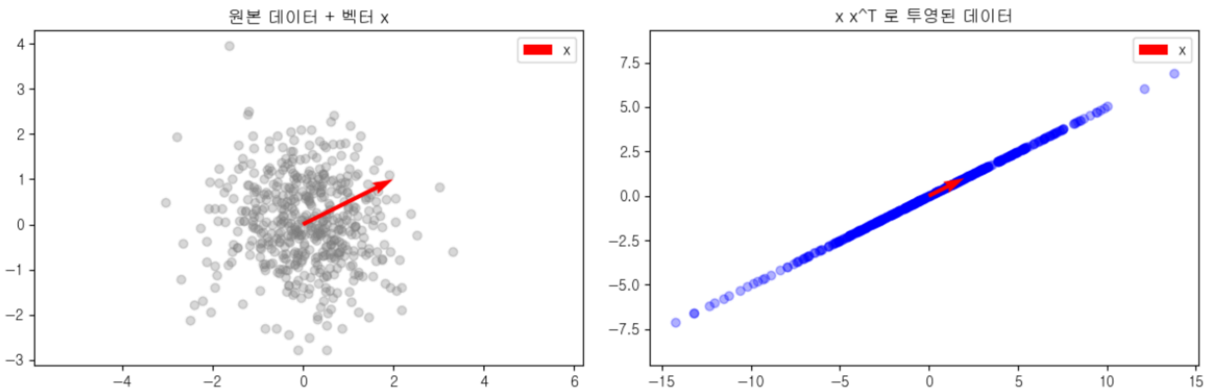
🖥️ 활용

- 공분산 계산, PCA, cholesky 분해 등에서 편리한 계산을 위해 활용.
- 실제로는 PSDM 형태를 더 많이 보게 됨.

5. 외적과 행렬의 제곱근

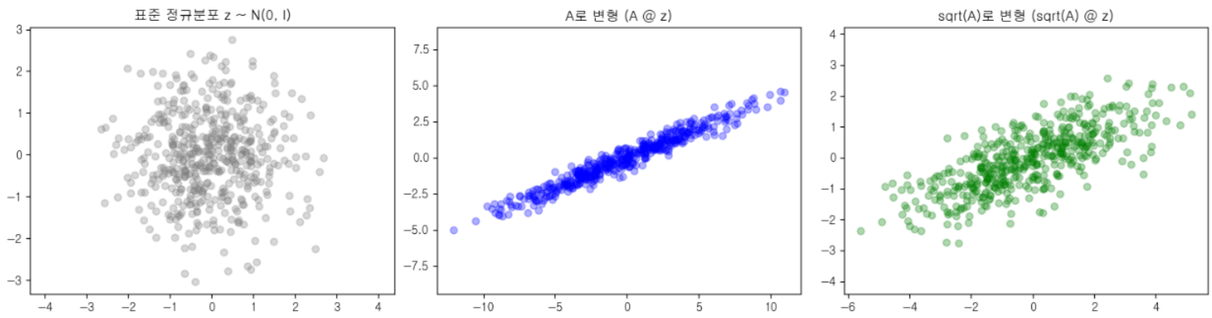
벡터의 외적 : Rank가 증가하는 벡터의 곱 계산.

- x^Tx는 대칭행렬이면서 양의 준정부호 행렬이다. → 안정적인 행렬을 만든다.
- x^Tx는 벡터 x의 방향으로 크게 반응하는 선형변환 행렬을 만들어낸다.



💡 행렬의 제곱근 : $A = B^2$을 만족하는 행렬 B

- A가 양의 정부호 행렬이고 대칭 행렬일 때 활용.
- 데이터를 어떤 방향으로 얼마나 퍼뜨릴 지 결정.



🖥️ 활용

- 외적 : PCA의 공분산 행렬 계산에 활용.
- 제곱근 : 정규분포 샘플링 등.

6. 스펙트럴 클러스터링

💡 데이터 클러스터링의 한 방법

- W (인접행렬) : 각 노드들 사이의 거리에 따른 가중치를 담은 행렬. (대칭행렬)
- D (차수행렬) : W의 각 노드 별 가중치 합을 담은 행렬. (대각행렬)
- L (라플라시안 행렬) : D - W로, 노드 사이의 정보를 종합적으로 담은 행렬. (대칭행렬)
- 라플라시안 행렬의 고유벡터가 데이터 분할의 기준이 된다.

  $Lf = \lambda v$

- 고유값 활용의 예시!

🖥️ 활용

- 클러스터링