# [수치해석] Eigen vector / value

## Eigen vector / value

**Spectrum**: Eigen value의 집합

$$Ax = \lambda x$$

- $x$: Eigen vector
- $\lambda$: Eigen value

위 식을 변형하면 아래와 같다.

$$(A - \lambda I)x = 0$$

- **det(**$A - \lambda I$) ≠ 0 → **Trival solution:** $x$ = 0
    - $(A - \lambda I)(A - \lambda I)^{-1}x = 0 \to x = 0$
- **det(**$A - \lambda I$) = 0 → **Non-trival solution:** $x$ ≠ 0
    - Eigen vector는 0이 아닌 벡터이기 때문에 **Non-trival soljtion**을 찾아야 한다.

### Characteristic equation

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

- 이 방정식을 풀면 Eigen value의 집합인 **Specturm을 구할 수 있다.**

### Eigen value의 성질

1. $A = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i$

2. det $A = \sum_{i=1}^{n} \lambda_i$

3. A가 대칭 행렬이라면, **Eigenvector는 서로 Orthogonal**하다.
    - $x_i^T x_j$ = 0 (i ≠ j)
    - $x_i^T x_j$ = $||x_i||^2$ (i = j)

4. A의 Eigen value를 $\lambda_1, \lambda_2, ..., \lambda_n$이라고 할 때, $(A - aI)$의 Eigen value는 $\lambda_1 - a, \lambda_2 - a, ..., \lambda_n - a$이다.
    - a는 임의의 스칼라 값이다.
    - **Exploiting shifting property라고 부른다.**

## Eigen vector의 기하학적 해석

1. $Ax = \lambda x$에서 $A$ 행렬은 **Eigen vector의 방향은 유지한 채, 크기만 변경**한다.

2. A가 대칭 행렬이라면, Eigen vector들을 서로 Orthogonal하다.
    - 대칭 행렬은 원래 Ax에서 x의 차원 축을 뭉개지 않는다.
        - $x_{12} = x_{21}$이라는 것은 1번이 2번에 주는 영향과 2번이 1번에 주는 영향이 동일하기 때문이라고 생각하면 된다.
    - 대칭 행렬 A를 이용해 $Ax = \lambda x$를 통해 Eigen vector를 얻었더니 전부 Orthogonal 했다.

3. A가 Singular matrix라면 det(A) = 0이기 때문에 **최소한 하나의 Eigen value는 0**이다.
    - det $A = \sum_{i=1}^{n} \lambda_i$ = 0

## SVD와 Eigen value decomposition

$$A = U\Sigma V^\top$$

- 이를 변형하면 $Av = \sigma u$이다.
- 이때 Row-Orthogonal space $V$에서 Column-Orthogonal space $U$로 변하도록 만드는 것이라고 생각하면 된다.
- SVD는 정방 행렬이 아니여도 가능하다.

A가 대칭 행렬 (N x N)이라면, $U = V$가 되어 **SVD와 Eigen value decompostion이 동일**해진다.

$$A = Q\Sigma Q^\top$$

- 때문에, 대칭 행렬인 경우 **서로 다른 Eigen vector가 Orthogonal**이다.
- Eigen value decompostion은 정방 행렬인 경우에만 사용 가능하다.

---

# Similar matrix

> **두 개의 N x N 행렬 A, B에 대해 $A = S^{-1}BS$를 만족하는 행렬 $S$가 존재한다면 두 행렬이 Similar하다**.

Eigen vector에 따르면 다음과 같다.

1. $Ax = \lambda x = S^{-1}BSx$
2. $\lambda Sx = BSx$
3. $x$가 0이 아니고 S가 Singular matrix가 아니라면 **(2)의 식이 성립하기 위해선 B의 Eigen value가** $\lambda$여야 한다.
4. 따라서 B$BSx$가 Eigen vector, $\lambda$가 Eigen value이다.

**A와 B가 Similar matrix이면, 두 행렬의 Eigen value가 동일하다.**

- 아래에서 **"좌표계를 변경해도 고유값은 동일하다"**라는 것을 확인할 수 있다.

EX) Rotation matrix

## 좌표 변환 (Coordinate transformation)

우리가 일반적으로 아는 X, Y 좌표계에서의 (3, 1) 벡터를 생각해보자.

- 이때의 Basis는 [1 0], [0 1]이다.

다른 Basis로 변화시켜보자.

- b1 = [1 1], b2 = [-1 1]로 변화시키고자 해보자.

(3, 1) 벡터의 방향은 동일하지만, 숫자로는 다르게 표현되어야 한다.

- 기존 (3, 1)은 **3**[1 0] + **1**[0 1]이다.
- Basis가 바뀌었기 때문에 (c1,c2) = c1[1 1] + c2[-1 1]로 나타나야 한다.
    - 이를 구하면 (2, -1)이다.

이렇게 **좌표계 자체가 변하는 것을 Coordinate transformation**이라고 한다.

- R은 좌표계를 회전 시키는 경우의 Coordinate transformation은 다음과 같이 쓸 수 있다.
- $X' = RX, Y' = RY$

## Similarity transformation

그럼 위 Rotation을 이용한 Coordinate transformation을 Similarity transformation으로 표현해보자.

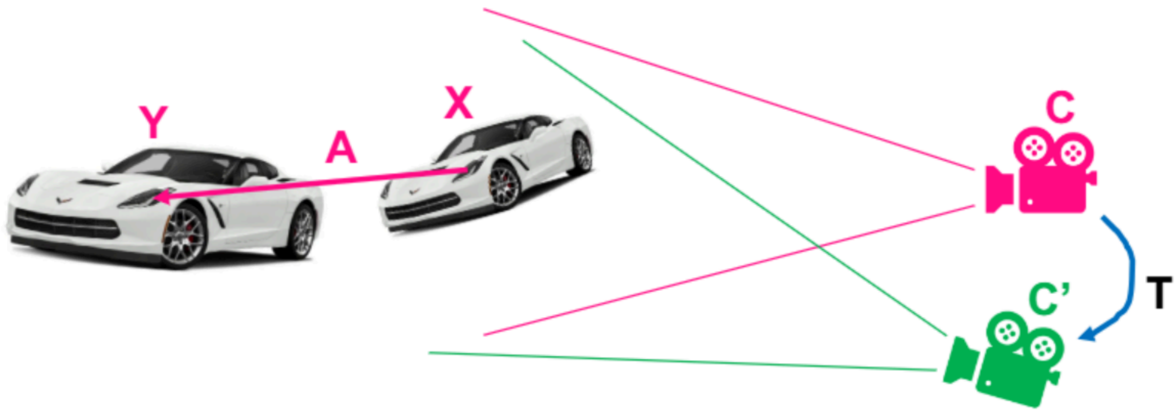
$y = Ax$인 좌표계에서 $y' = Ry$로 회전시키는 상황을 고려하자.

1. $y' = Ry = RAx$ 이다.
2. $x$도 새로운 좌표계에서 표현하면 $x' = Rx$이다.

- 1번 식에 대입하기 위해 $x = R^{-1}x$를 사용하자.

3. (1) 식에 $x = R^{-1}x$를 대입하면 $y' = RAR^{-1}x = Bx$이다.
   - $B = RAR^{-1}$일 때, B와 A는 Similar matrix이다.
   - 원래 좌표계에서 $A$라는 변환이 회전한 좌표계에선 $B$처럼 보이는 것이다.
     - 둘은 한 벡터를 동일하게 변환하는 행렬이고 좌표계가 달라 표현만 다른 것이다.
     - **변환이 동일하기 때문에 Eigen value가 동일**하다.
       - **좌표계가 변경되어도 고유값은 동일**하다.
       - 변화량 자체는 동일하다.

## Example: Computer Graphics




카메라가 원래의 위치 $C$에서 $C'$으로 이동하고 이 이동을 $T$라고 하자.

자동차는 $X$에서 $Y$로 이동하며 그 이동은 $A$라고 한다.

위 상황에서 아래 식을 도출할 수 있다.

- $Y = AX$
- $C' = TC$
- $X' = TX$

- $Y’ = TY$

이때, $C’$ **카메라에서 보이는 차의 이동을 B로 표현**해보자.

- $Y’ = BX’$

$B$를 구해보자.

1. $Y' = TY$에 $Y = AX$을 대입한다,
    - $Y’ = TAX$
2. $Y’ = TAX$에 $X = T^{-1}X’$을 대입한다.
    - $Y’ = TAT^{-1}X’$
3. 따라서 $B = TAT^{-1}$이다.

---

# Numerical methods

> **$det(A - \lambda I) = 0$을 계산하는 방법에는 한계가 있기 때문에 대신 사용하는 Eigen value를 찾는 방법**

## 1. Power method

> **절댓값이 가장 큰 Eigen value를 찾는 방법**

$$Ax^{(k)} = y^{(k+1)} = \lambda^{(k+1)}x^{(k+1)}$$

**과정**

1. $Ax$를 구한다.
2. $Ax$의 결과로 부터 **단위성분**을 만든다.

- $c[vector]$ 형태가 된다.
- Vector의 전체 성분 중 **절댓값이 가장 큰 것으로 나눈다.**

3. (2)에서 $c$가 Eigen value, [vector]가 Eigen vector가 된다.
4. (3)에서의 Eigen vector를 x로 하여 (1) ~ (3) 과정을 반복한다.
5. 수렴했을 때의 Eigen value와 Eigen vector가 가장 큰 Eigen value와 그에 해당하는 Eigen vector이다.

## 2. Inverse method

> **절댓값이 가장 작은 Eigen value를 찾는 방법**

$$A^{-1}x^{(k)} = y^{(k+1)} <=> Ay^{(k+1)} = x <=> Lc = x^{(k)}, Uy^{(k+1)} = c$$

**LU 분해를 이용하여 $A^{-1}x$를 계산하는 것 제외하면 Power method와 동일하다.**

**Exploiting shifting property**를 이용하면, $(A - \lambda I)$의 **Eigen value가 $\lambda - a$가 되는 것을 알기 때문에,**

$(A - \lambda I)$**의 가장 크거나 작은 Eigen value를 찾도록 해서 A의 중간 정도 크기의 Eigen value를 얻을 수도 있다.**

추가로, Power Method는 **"1등 고유값"과 "2등 고유값"의 차이가 클수록 수렴 속도가 빨라지는데, Exploiting shifting property를 통해 차이를 크게 만들어 수렴 속도를 높일 수 있다.**

## 3. Deflation method

> **Power method를 통해 가장 큰 Eigen value를 찾았을 때, 이미 찾은 Eigen value의 영향을 제거해서 다시 Power method를 통해 다음으로 큰 Eigen value를 찾도록 반복하는 방법**

$\lambda_1, \lambda_2, \ldots, \lambda_n$, $v^{(1)}, v^{(2)}, \ldots, v^{(n)}$이 있을 때, $\lambda_1$이 가장 크다는 것을 이미 안다고 가정하자.

**Deflated matrix** $B = A - \lambda_1 v^{(1)} x^T$를 만든다.

- $x^T v^{(1)}$ = 1이어야 한다.
    - 이를 만족하는 $x$를 적절히 선택해야 한다.
- 양변에 $v^{(1)}$을 곱해보면 B에서의 $v^{(1)}$에 대응되는 Eigen value가 0임을 확인할 수 있다.
- Eigen values = $0, \lambda_2, \lambda_3, \ldots, \lambda_n$
    - 고유값은 동일하다.
- Eigen vectors = $v^{(1)}, w^{(2)}, w^{(3)}, \ldots, w^{(n)}$
- **A의 고유 벡터와 B의 고유 벡터의 관계식**: $v^{(i)} = (\lambda_i - \lambda_1) w^{(i)} + \lambda_1 (x^T w^{(i)}) v^{(1)}$

이제 B에서 **Power method를 진행하면 두 번째로 큰 Eigen value를 얻을 수 있다.**

- Power method 또는 직접 Eigen value를 구할 수도 있다.
- B에서 제거한 Eigen value에 해당하는 열과 행은 지우고 Eigen value를 구하면 된다.
- $w^{(2)}$를 얻고, 위 관계식을 통해 $v^{(2)}$를 얻으면 된다.

## Hotelling's Deflation Method

$$A_{i+1} = A_i - \lambda_i v_i v_i^T$$

- 직전 행렬을 통해 구한 가장 큰 고유값 / 고유 벡터를 이용하여 다음 행렬을 계산한다.
- 각 행렬에 대해 Power method를 반복한다.
- Eigen vector가 Orthogonal하면 **Eigen value decomposition**가 깔끔하기 때문에 Symmertric matrix A에 대해서만 가능하다.

**Eigen value decomposition**

$$A = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_n v_n v_n^T$$

- 위 공식을 사용하는 이유이다.

# 4. Jacobi Transformation

> **대칭 행렬에서 대각 원소만 남겨 Eigen value를 얻고자 하는 방법**

**과정**

1. 행렬 A에서 가장 큰 Off-diagonal 원소를 찾아, 해당 원소를 0으로 만들 수 있는 Rotation matrix $P$를 찾는다.
2. $A_{new} = P^T AP$ 를 얻는다.
3. $A_{new}$가 diagonal element만 가질 때까지 위 과정을 반복한다.

**Rotation은 Similarity transformation이기 때문에 Eigen value가 변하지 않아서 가능한 방법이다.**

최종 Diagonal matrix의 각 Element가 해당 방향의 Eigen value이다.