인공지능수학

(선형대수학2)

개 요

벡터, 행렬		
특징	**연산**	**특수형태**
노름, 거리, 차원, 내적, 행렬식	벡터합, 스칼라곱, 행렬합, 행렬곱	영행렬, 음행렬, 단위행렬, 역행렬, 대각행렬, 전치행렬 등

선형문제
- 선형결합
- 기본행변환
- 가우스-조르단 소거법

알고리즘
- LU분해
- QR분해
- 최소제곱법
- 그람-슈미트 방법

행렬의 특성값
- 고윳값&고유벡터
- 특잇값
- 특잇값분해 (SVD)

Gram-Schmidt 방법

Gram-Schmidt 방법

Definition

Gram-Schmidt: 내적 공간 (inner product space)에서 유한 개의 선형 독립 벡터 집합을 정규 직교 기저 (orthonormal basis)로 변환하는 방법입니다.

때문에 그람 슈미트 과정 (Gram-Schmidt Process) 또는 그람 슈미트 단위 직교화 (Gran-Schmidt orthonormalization)이라고 부릅니다.

Gram-Schmidt 방법

Why?

그람 슈미트 과정의 필요성은 다음과 같습니다. 흔히 우리가 다루는 3차원 공간에는 x, y, z 축이 있으며 각 축에 대한 기저가 존재합니다. 이것들을 우리는 standard basis라고도 부르며 해당 기저들은 서로 독립이기 때문에 다른 벡터를 표현할 때 basis의 조합으로 표현이 가능합니다.

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

3차원에서의 standard basis

standard basis와 같이 직교성을 갖는 기저는 많은 연산과정에서 편리하다는 장점이 있습니다. 즉 직교 기저에 의해서 생성되는 공간의 벡터들을 표현하기 위해서 기저에 특정한 가중치들을 가진 선형 결합으로서 쉽게 표현이 가능한 장점이 생깁니다.

Gram-Schmidt 방법

Gram-Schmidt과정 또한 어떤 벡터들이 span하는 공간에 대해 공간을 대표하는 basis를 찾는 과정입니다.

Gram-Schmidt 과정을 통해서 나온 basis의 특성은 다음과 같습니다. basis끼리 직교 (Orthogonal)이며 정규화된 (Normalized) 값입니다. 즉 정규 직교 (Orthnormal) basis를 얻게 됩니다.

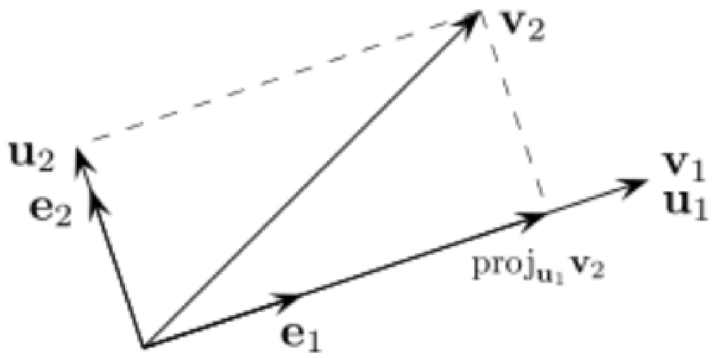
**직교 (Orthogonality): 두 벡터의 내적이 0값을 가지는 것 (또는 선형 독립일 때)*

**정규화 (Normalization): 벡터를 벡터의 norm으로 나눈 것 (때문에 정규화된 벡터는 최대 크기가 1 이하로 됨)*

Gram-Schmidt 방법

2D example

아래 그림을 보겠습니다. 우선 내적 공간 V의 기저 {v1, v2}가 주어져 있습니다.



[1] Gram-Schmidt 과정의 기본 원리

v1과 v2는 선형 독립이며 이때 v1과 v2가 만들어내는 벡터 공간의 직교 벡터는 u1과 u2입니다. 그리고 u1과 u2를 단위 벡터 (또는 정규화) 생성 과정을 통해서 정규 직교 기저 (Orthonormal basis)은 e1과 e2가 되며 해당 e1과 e2를 찾는 과정이 Gram-Schmidt입니다.

Gram-Schmidt 방법

순서는 다음과 같습니다.

1. 먼저 하나의 기준 벡터를 선정합니다.

임의의 벡터 u1은 제시된 벡터와 동일하게 둡니다 (u1 = v1)

$$\boxed{u_1 = v_1}$$

2. u1에 수직 (Orthogonal)한 벡터인 u2를 찾기 위해서 먼저 v2를 v1 (또는 u1)에 정사영 (Projection)합니다.

정사영된 벡터를 v2에 빼줌으로써 u2를 찾을 수 있습니다.

$$\boxed{u_2 = v_2 - proj_{u1}v_2 = v_2 - \frac{v_2 \cdot u_1}{u_1 \cdot u_1}u_1}$$

Gram-Schmidt 방법

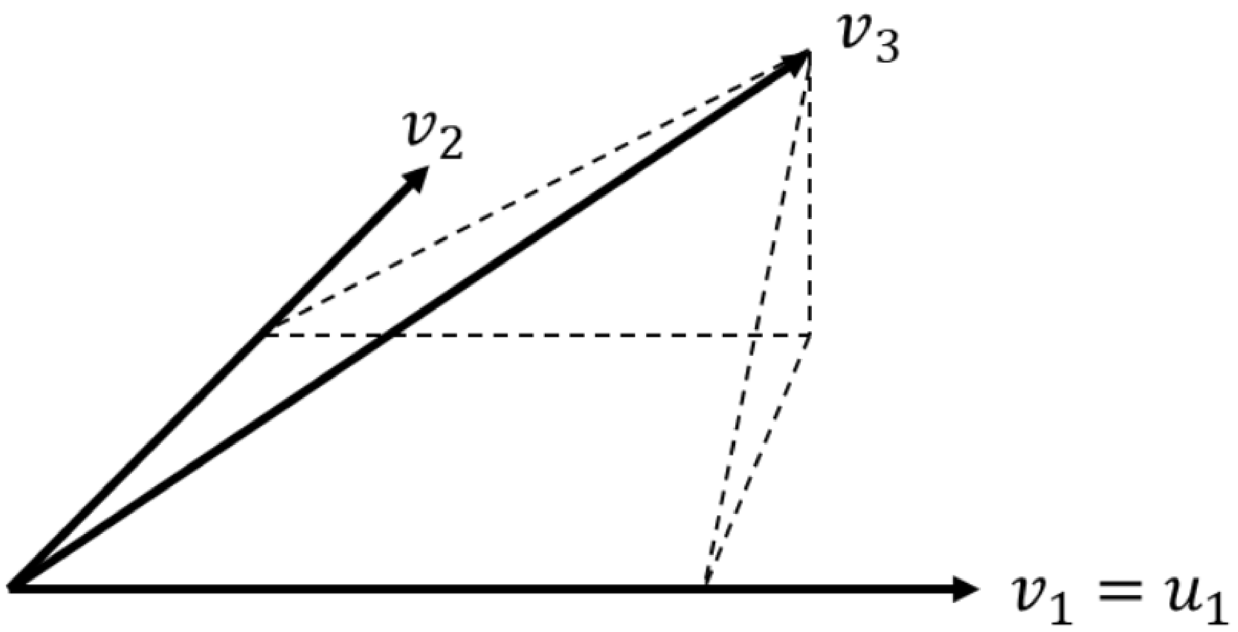
3. u1과 u2를 모두 찾았으니 정규화를 합니다.

$$e_1 = u_1 / \| u_1 \|$$
$$e_2 = u_2 / \| u_2 \|$$

**1번 과정에서 u2 = v2로 시작을 해도 가능하지만, 반드시 v1과 v2를 사용해야 합니다. 아무런 벡터를 잡고 Gram-Schmidt를 진행할 수는 없으며 u는 v 벡터들을 생성하는 직교 기저이어야만 v- proju(v)값이 v공간에 속할 수 있기 때문입니다*
(부분 공간에 포함).

Gram-Schmidt 방법

3D example



v1, v2, v3의 벡터가 있을 때 Gram-Schmidt 과정을 적용해보겠습니다.

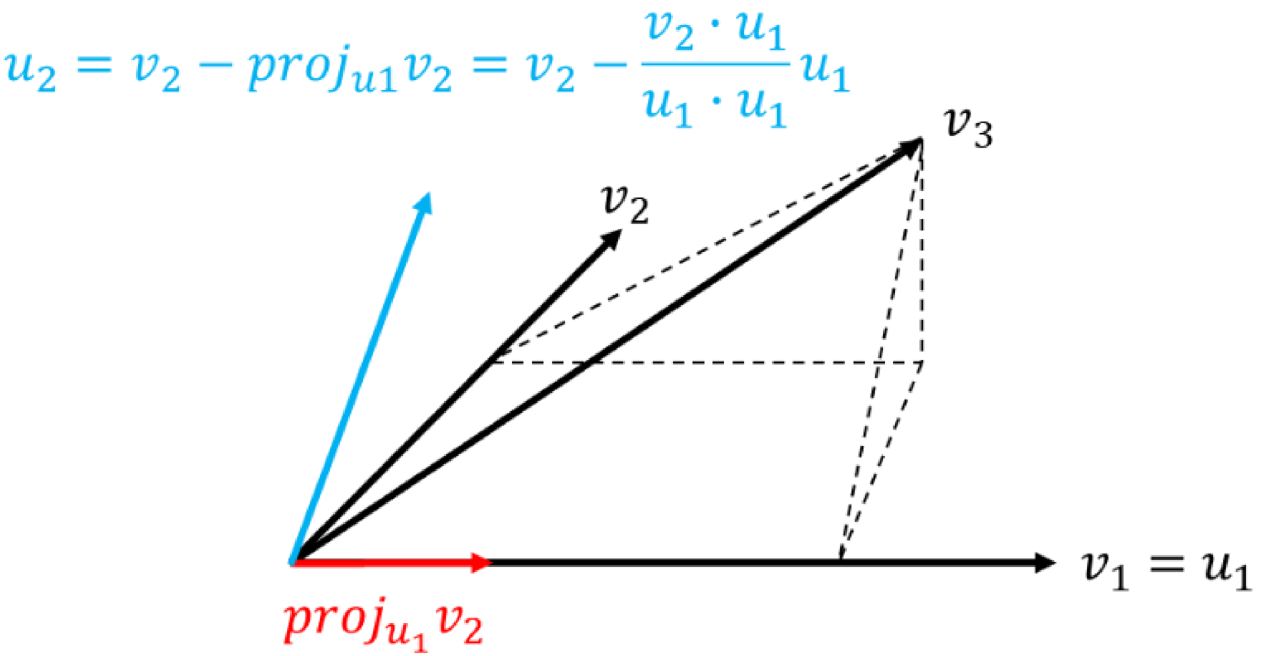
Gram-Schmidt 방법

아래 그림과 같이 v2와 정사영된 벡터를 빼줌으로써 u2를 찾을 수 있습니다.

1. 먼저 임의의 벡터 하

2. u2를 찾기 위해서 2I

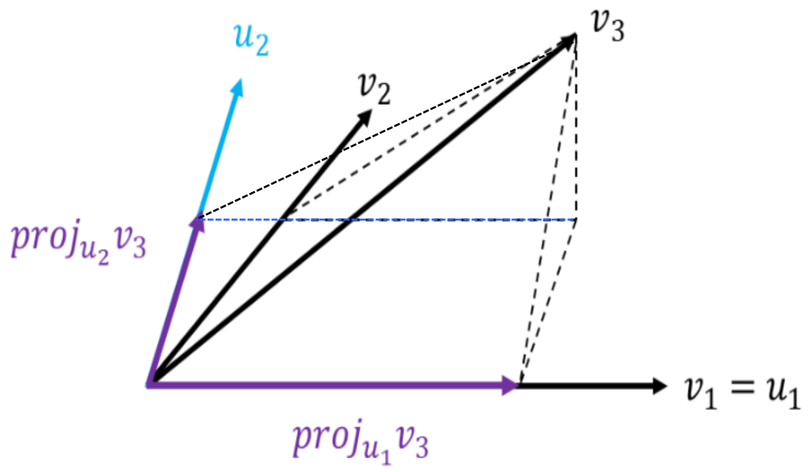
찾고



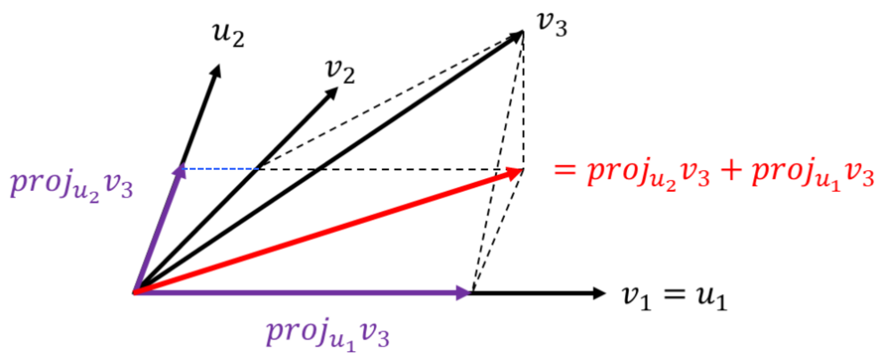
$$u_2 = v_2 - proj_{u1}v_2 = v_2 - \frac{v_2 \cdot u_1}{u_1 \cdot u_1}u_1$$

Gram-Schmidt 방법

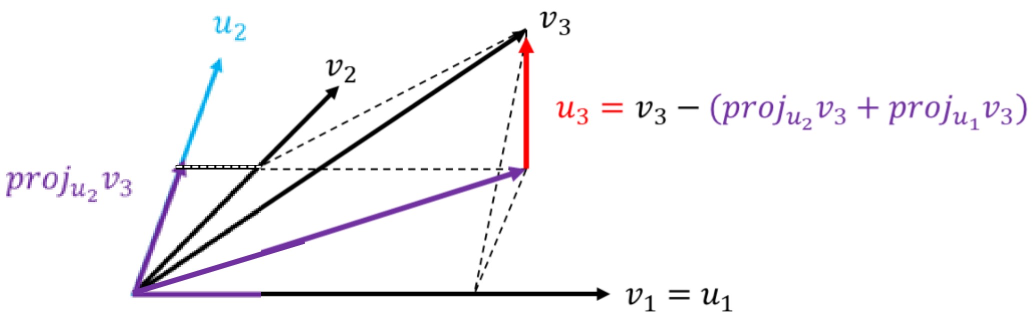
3. u3를 찾기 위해서 위의 과정에서 구한 u1과 u2 벡터에 v3를 정사영을 하여 proju2(v3)와 proju1(v3)를 구합니다.



계산된 2개의 정사영 벡터의 합 (빨간색)은 아래 그림과 같이 v3를 u1, u2 평면에 정사영한 벡터가 됩니다.



u3를 구하기 위해서 v3에 정사영 벡터 합을 빼줌으로서 u3를 얻을 수 있습니다.



$$u_3 = v_3 - proj_{u_1}v_3 - proj_{u_2}v_3 = v_3 - \frac{v_3 \cdot u_1}{u_1 \cdot u_1}u_1 - \frac{v_3 \cdot u_2}{u_2 \cdot u_2}u_2$$

Gram-Schmidt 방법

4. 마지막으로 정규화 과정을 수행하면 정규 직교 기저를 얻을 수 있습니다.

$$e_1 = \frac{u_1}{\|u_1\|}$$

$$e_2 = \frac{u_2}{\|u_2\|}$$

$$e_3 = \frac{u_3}{\|u_3\|}$$

Gram-Schmidt 방법

Generalization

일반화된 과정은 아래의 사진과 같습니다. 1번에서 basis가 주어졌을 때 2번 과정을 통해서 직교 벡터들 (w)를 구했으며, 3번에서 정규화하여 orthonormal basis를 얻어낼 수 있었습니다.

THEOREM 5.12 Gram-Schmidt Orthonormalization Process

1. Let $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ be a basis for an inner product space V.
2. Let $B' = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$, where

$$\mathbf{w}_1 = \mathbf{v}_1$$

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{\langle \mathbf{v}_2, \mathbf{w}_1 \rangle}{\langle \mathbf{w}_1, \mathbf{w}_1 \rangle}\mathbf{w}_1$$

$$\mathbf{w}_3 = \mathbf{v}_3 - \frac{\langle \mathbf{v}_3, \mathbf{w}_1 \rangle}{\langle \mathbf{w}_1, \mathbf{w}_1 \rangle}\mathbf{w}_1 - \frac{\langle \mathbf{v}_3, \mathbf{w}_2 \rangle}{\langle \mathbf{w}_2, \mathbf{w}_2 \rangle}\mathbf{w}_2$$

$$\vdots$$

$$\mathbf{w}_n = \mathbf{v}_n - \frac{\langle \mathbf{v}_n, \mathbf{w}_1 \rangle}{\langle \mathbf{w}_1, \mathbf{w}_1 \rangle}\mathbf{w}_1 - \frac{\langle \mathbf{v}_n, \mathbf{w}_2 \rangle}{\langle \mathbf{w}_2, \mathbf{w}_2 \rangle}\mathbf{w}_2 - \cdots - \frac{\langle \mathbf{v}_n, \mathbf{w}_{n-1} \rangle}{\langle \mathbf{w}_{n-1}, \mathbf{w}_{n-1} \rangle}\mathbf{w}_{n-1}.$$

   Then B' is an *orthogonal* basis for V.

3. Let $\mathbf{u}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}$. Then $B'' = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ is an *orthonormal* basis for V.
   Also, $\text{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\} = \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ for $k = 1, 2, \ldots, n$.

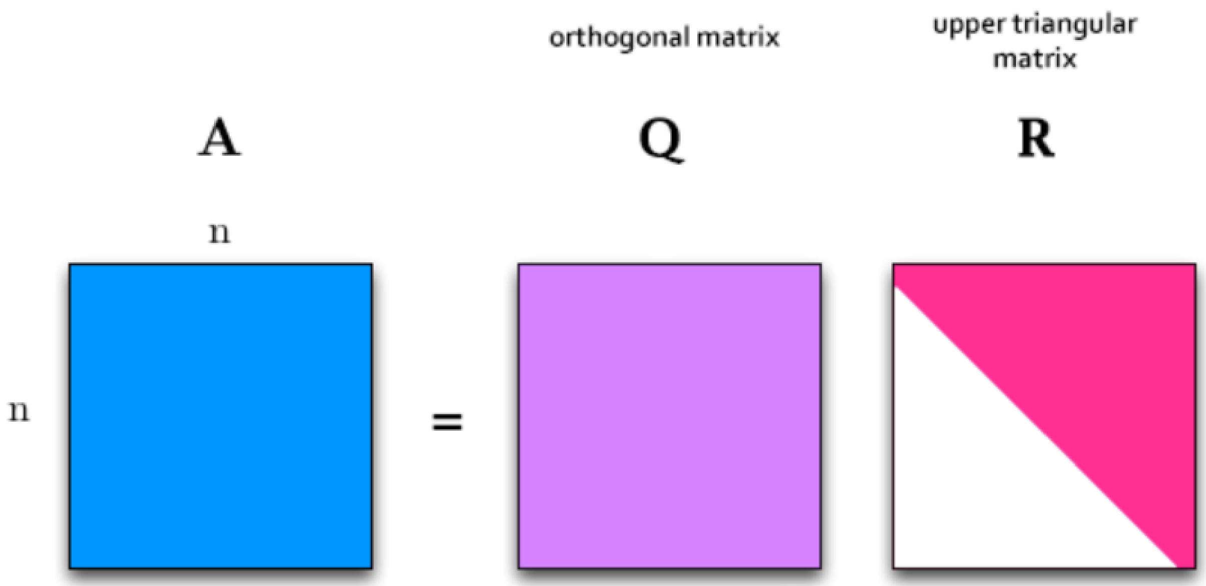
[2] Generalization of Gram Schmidt Process

QR 분해법

QR 분해법

Definition

QR 분해는 실수 행렬을 직교 행렬 (Q, Normal orthogonal matrix)과 상삼각 행렬 (R, upper triangular matrix)의 곱으로 나타내는 행렬 분해 방법입니다.



[1] Overview of QR Decomposition

QR 분해법

Why

Ax = b 문제는 상당히 실생활에 많이 존재합니다. b라는 결과를 얻기 위해서 시스템 A에 어떠한 x 인풋을 넣어야 얻을 것인지에 생각할 때 분야를 막론하고 다양한 예제들이 존재할 것입니다.

$$Ax = b$$

Ax = b를 만족하는 x를 구하고 싶다고 가정할 때 일반적인 방법은 x = A^(-1)b 또는 pesudo inverse와 같이 A의 역행렬을 계산하여 b에 곱하고 x를 얻을 수가 있습니다.

$$\boxed{x = A^{-1}b \quad or \quad x = A^{+}b}$$

하지만 역행렬을 구하는 것이 전공책에 제시된 문제 수준에서 더 나아가 차원이 커질수록 시간의 복잡도가 기하급수적으로 커지게 됩니다. 때문에 우리는 역행렬을 직접 구하지 않고 시간 복잡도가 선형이며, 수치적으로 용이한 LU 분해 또는 QR 분해를 사용합니다.

QR 분해법

(1) Characteristics of `Q` (Q 행렬 특성)

QR 분해에서 Q는 Gram-Schmidt Process를 통해서 얻어진 $\boxed{\text{정규 직교 기저로}}$ 이뤄진 행렬입니다.

$$Q = [q_1 \quad q_2 \quad \cdots \quad q_n]$$

Q는 직교 행렬이기 때문에 symmetric matrix이면서 다음의 성질을 가지고 있습니다.

$$(1) Q^T Q = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}, \qquad i.e. Q^T Q = QQ^T = I \qquad \therefore Q^{-1} = Q^T$$

$$(2) \|Q\underline{x}\| = \|\underline{x}\| \ \ for \, \forall \, \underline{x}, \qquad < Q\underline{x}, Q\underline{y} > = < \underline{x}, \underline{y} > for \ \underline{x}, \underline{y}$$

QR 분해법

(2) Characteristics of `R` (R 행렬 특성)

R은 상삼각 행렬입니다. 행렬의 주대 각선을 기준으로 대각 항의 모든 아래쪽 원소 값이 0인 행렬이 상 삼각 행렬입니다.

$$\begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

상삼각 행렬은 덧셈, 곱셈, 역행렬에 대해 닫혀있습니다. 때문에 상삼각 행렬끼리의 덧셈, 곱셈, 역행렬 연산을 통해 나오는 행렬 (또는 결과) 또한 상삼각 행렬 형태를 지닙니다 (마찬가지로 하 삼각 행렬 또한 동일합니다).

**다만 순 삼각 행렬은 주 대각선 성분들이 0 값을 가지기에 역행렬에 대해 닫혀있기 위해서는 삼각 행렬이 가역 행렬 이어야 하는 추가 조건이 반드시 존재해야 합니다.*

QR 분해법

이러한 성질을 통해서 결과적으로 Ax=b문제는 다음과 같이 풀이될 수 있습니다. A=QR일 때 아래의 순서와 동일하게 진행을 하면 x를 도출할 수 있습니다.

$$\boldsymbol{A = QR}$$

$$(1)\ \mathrm{R^T Q^T Q R x = R^T Q^T b}$$

$$(2)\ R^T R x = R^T Q^T b$$

$$(3)(R^T)^{-1} R^T R x = (R^T)^{-1} R^T Q^T b$$

$$(4)\ Rx = Q^T b$$

$$(5)\ if\ \boxed{y = Rx = Q^T b}\ we\ can\ obtain\ y$$
$$and\ can\ solve\ Rx = y\ with\ back\ subtituition$$

QR 분해법

QR Decomposition advantages (QR 분해 장점들)

(1) QR 분해는 A가 정방 행렬이 아닐 때 (행과 열 개수가 다른 행렬)도 적용이 가능합니다.

(2) 컴퓨터에 알고리즘을 적용 시 구현이 용이합니다.

QR 분해법

다음 과정은 R 행렬을 얻는 과정입니다.

A = QR 일 때, 열 벡터 a_n을 가진 행렬 A는 다음과 같습니다.

$$A = [a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_n]$$

선형 독립인 열 벡터 a_n으로 구성된 A에 Gram-Schmidt과정을 적용하면 다음과 같이 Q 행렬을 얻습니다.

$$Q = [q_1 \quad q_2 \quad \cdots \quad q_n]$$

이때 Q 행렬의 q_n 열 벡터는 A에 대해서 정규 직교 기저이기 때문에 a_n 을 q_n의 선형 결합으로 표현 가능합니다.

$$\begin{aligned}
\mathbf{a}_1 &= (\mathbf{a}_1^T\mathbf{q}_1)\mathbf{q}_1 \\
\mathbf{a}_2 &= (\mathbf{a}_2^T\mathbf{q}_1)\mathbf{q}_1 + (\mathbf{a}_2^T\mathbf{q}_2)\mathbf{q}_2 \\
&\vdots \\
\mathbf{a}_n &= \sum_{i=1}^{n}(\mathbf{a}_n^T\mathbf{q}_i)\mathbf{q}_i
\end{aligned}$$

QR 분해법

정리하면 A = QR 형태는 아래와 같습니다.

$$\begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} a_1 \cdot q_1 & a_2 \cdot q_1 & \cdots & a_n \cdot q_1 \\ a_1 \cdot q_2 & a_2 \cdot q_2 & \cdots & a_n \cdot q_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1 \cdot q_n & a_2 \cdot q_n & \cdots & a_n \cdot q_n \end{bmatrix}$$

이때 a1·q2 을 생각해볼 때 q2는 a1 및 q1 성분이 모두 제거된 성분이기에 값이 0이 됩니다.

QR 분해법

마찬가지로 a_i · q_j 에 대해 생각할 때 i < j인 경우 a_i · q_j = 0이 되어버리기 때문에 아래와 같은 식으로 변경되어서 R행렬이 곧 상 삼각 행렬로 됩니다.

$$\begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^T\mathbf{q}_1 & \mathbf{a}_2^T\mathbf{q}_1 & \cdots & \mathbf{a}_n^T\mathbf{q}_1 \\ 0 & \mathbf{a}_2^T\mathbf{q}_2 & \cdots & \mathbf{a}_n^T\mathbf{q}_2 \\ 0 & 0 & \cdots & \mathbf{a}_n^T\mathbf{q}_3 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{a}_n^T\mathbf{q}_n \end{bmatrix}$$

정리하면 R 행렬은 a와 q로 이뤄진 행렬이며 a와 q정보는 이미 다 알고 있으므로 R을 도출해낼 수가 있습니다.

QR 분해법

Example

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

A가 위와 같은 행렬일 때 QR 분해를 해보겠습니다. 순서는 다음과 같습니다.

(1) Gram-Schmidt 과정을 통해 Q 행렬 계산

(2) Q 행렬과 A 행렬 연산을 통해 R 행렬 계산

QR 분해법

먼저 (1) Gram-Schmidt 과정을 수행하기 위해서 행렬 A의 열 벡터를 다음과 같이 정의합니다.

$$a_1 = \begin{bmatrix}1\\1\\0\end{bmatrix}, a_2 = \begin{bmatrix}1\\0\\1\end{bmatrix}, \ a_3 = \begin{bmatrix}0\\1\\1\end{bmatrix}$$

직교 벡터인 u1, u2, u3에 대해서 계산을 위해서 먼저 하나의 a_n 벡터를 u_n 벡터와 동일하게 정합니다.

$$a_1 = \begin{bmatrix}1\\1\\0\end{bmatrix} = u_1$$

u2 벡터를 구하기 위해서 u1에 a2 벡터를 정사 영하여서 계산합니다.

$$u_2 = a_2 - proj_{u_1} a_2 = a_2 - \frac{u_1 \cdot a_2}{u_1 \cdot u_1} u_1 = (\frac{1}{2}, -\frac{1}{2}, 1)$$

QR 분해법

u3 벡터를 구하기 위해서 u1과 u2에 먼저 정사영을 각각 하여 정 사영된 벡터를 각각 구하고, u3에 정사영된 벡터들을 빼줍니다.

$$u_3 = a_3 - proj_{u_1}a_3 - proj_{u_2}a_3 = a_3 - \frac{u_1 \cdot a_3}{u_1 \cdot u_1}u_1 - \frac{u_2 \cdot a_3}{u_2 \cdot u_2}u_2 = (-\frac{2}{3}, \frac{2}{3}, \frac{2}{3})$$

마지막으로 정규화를 합니다.

$$e_1 = \frac{u_1}{\|u_1\|} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right)$$
$$e_2 = \frac{u_2}{\|u_2\|} = (\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}})$$
$$e_3 = \frac{u_3}{\|u_3\|} = (-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$$

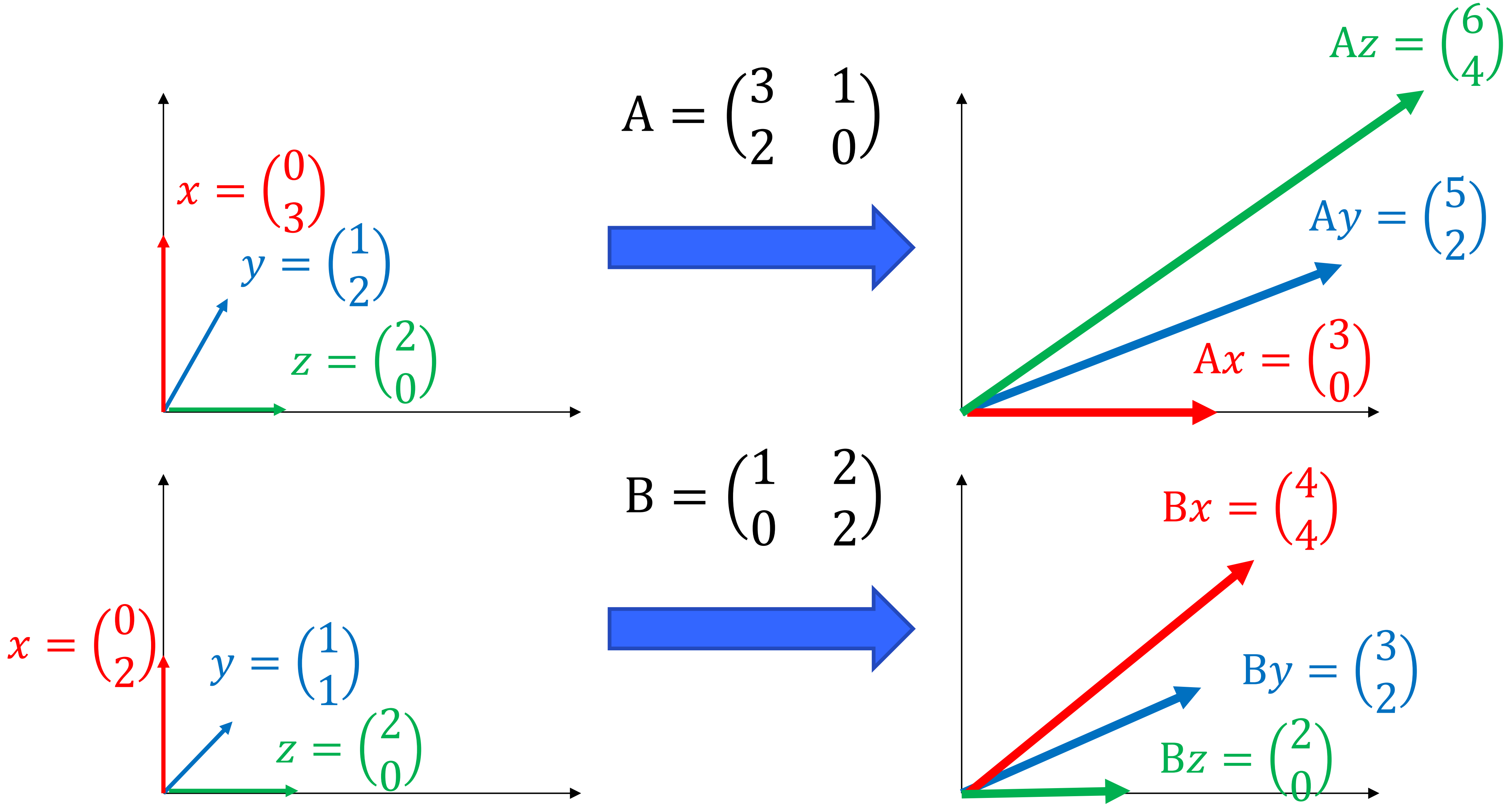
QR 분해법

(2) 과정인 R 행렬을 계산하면 최종적으로 아래와 같이 A 행렬을 구할 수 있습니다.

$$A = QR = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 0 & 2/\sqrt{6} & 1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 2/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 3/\sqrt{6} & 1/\sqrt{6} \\ 0 & 0 & 2/\sqrt{3} \end{bmatrix}$$

고유값 과 고유벡터
고유분해 (eigen – decomposition)

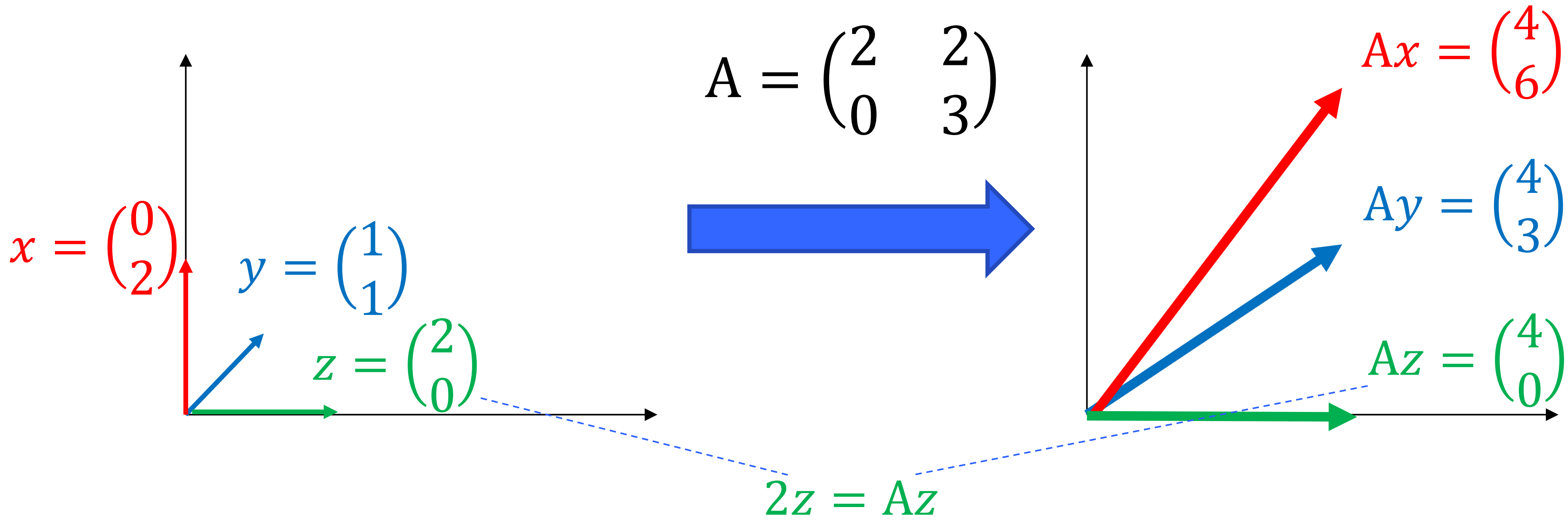
행렬-벡터 곱셈



$x = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$
$y = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$
$z = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$
$A = \begin{pmatrix} 3 & 1 \\ 2 & 0 \end{pmatrix}$
$Az = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$
$Ay = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$
$Ax = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$
$x = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$
$y = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
$z = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$
$B = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$
$Bx = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$
$By = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$
$Bz = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$

행렬-벡터 곱셈

$$A = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix}$$



- 어떤 벡터들은 벡터의 크기만 변화 시키기도 한다.

고유값 & 고유벡터

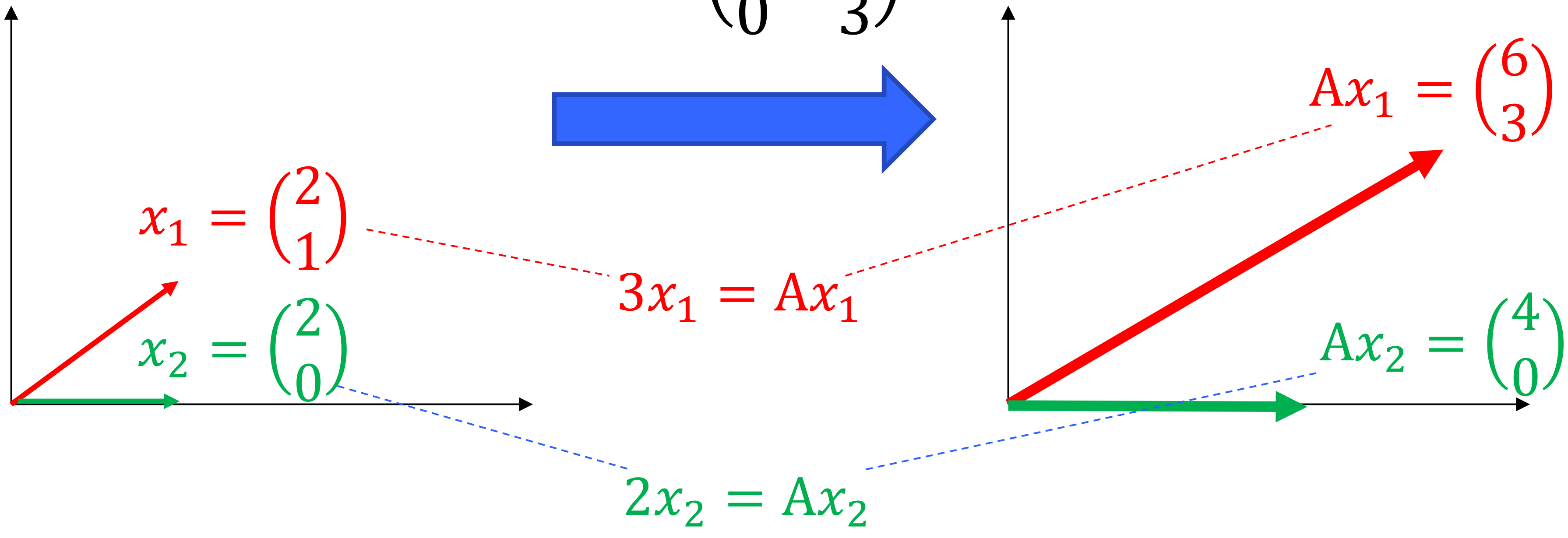
고유치(값) – 고유벡터

벡터 x 에 대해 $Ax = \lambda x$ 를 만족시키는 값 λ

x : A 의 고유벡터

λ : A 의 고유값(치)

$$\mathrm{A} = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix}$$



고유치(값) - 고유벡터

- 고유벡터 λ 는 다음 조건 $Ax = \lambda x$ 를 만족시킨다.

$$Ax = \lambda x$$

$$\Rightarrow \quad Ax - \lambda x = (A - \lambda I)x = 0$$

$$\Rightarrow \quad \underline{\det(A - \lambda I) = 0}$$

A 의 특성 다항식

예제)

$$\mathrm{A} = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix} \quad \Rightarrow \quad \det(A - \lambda I) = \det\begin{pmatrix} 2-\lambda & 2 \\ 0 & 3-\lambda \end{pmatrix}$$

$$= (2-\lambda)(3-\lambda)$$

그러므로, $\lambda = 2, 3$

고유치(값) - 고유벡터

- 고유벡터 x 는 각 고유값 λ 에 대해 $Ax = \lambda x$ 를 만족한다.

- 하나의 고유값 $\lambda = 2$ 에 대한 고유벡터를 찾아라

$$\mathrm{A} = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix} \Rightarrow \text{하나의 고유값 } \lambda = 2 \text{ 라 하자}$$

$$\Rightarrow \begin{pmatrix} 2-2 & 2 \\ 0 & 3-2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} 0 \times x_1 + 2 \times x_2 = 0 \\ 0 \times x_1 + 1 \times x_2 = 0 \end{cases} \Rightarrow x_1 : \text{임의의 값}, \ x_2 = 0$$

$$\Rightarrow \quad x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

(여기서 $x_1 = 1$ (기본 값)을 택한다.)

고유치(값) - 고유벡터

- 고유벡터 x 는 고유값 λ 에 대해 $Ax = \lambda x$ 를 만족한다.

- 하나의 고유값 $\lambda = 3$ 에 대한 고유벡터를 찾아라

$$\mathrm{A} = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix} \Rightarrow \text{ 고유값 } \lambda = 3 \text{ 이라고 하자}$$

$$\Rightarrow \begin{pmatrix} 2-3 & 2 \\ 0 & 3-3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -1 & 2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} -1 \times x_1 + 2 \times x_2 = 0 \\ 0 \times x_1 + 0 \times x_2 = 0 \end{cases} \Rightarrow x_2 : \text{임의의값}, \ x_1 = 2$$

$$\Rightarrow \ x = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad (x_2 = 1 \text{ 값을 택한다.})$$

고유치(값) - 고유벡터

예제 2) $\mathrm{B} = \begin{pmatrix} 4 & 2 \\ -1 & 1 \end{pmatrix} \Rightarrow \det(B - \lambda I) = \det\begin{pmatrix} 4-\lambda & 2 \\ -1 & 1-\lambda \end{pmatrix}$

$= (4-\lambda)(1-\lambda)+2$

$= \lambda^2 - 5\lambda + 6$

$= (\lambda-2)(\lambda-3)=0$

그러므로, $\lambda = 2, 3$

$\Rightarrow$ 고유값 $\lambda = 3$에 대해

$$\Rightarrow \begin{pmatrix} 4-3 & 2 \\ -1 & 1-3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ -1 & -2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} 1 \times x_1 + 2 \times x_2 = 0 \\ -1 \times x_1 - 2 \times x_2 = 0 \end{cases}$$ $\Rightarrow x_1 = 2$ 를 택하면 $x_2 = -1$

$$\Rightarrow \quad x = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

고유값 $\lambda = 2$에 대해서도 위와 같은 방법으로 진행하면 고유벡터를 얻을 수 있다.
따라서

$$x = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad (x_2 = 1)$$

고유치(값) - 고유벡터

➲ 고유치가 / 중근인 경우 $(\lambda_1 = \lambda_2 = \ldots = \lambda_l \neq \ldots \neq \lambda_n)$

행렬 $\mathbf{A}$의 고유치가 중근이면, 앞의 방법으로는 n개의 선형 독립인 고유벡터를 구할 수 없으며 대각화도 할 수 없다. 그러나 고유치가 l 중근인 경우, 즉 $\lambda_1 = \lambda_2 = \cdots = \lambda_l$이면, 대각행렬과 유사한 다음과 같은 방법으로 표현할 수 있다.

$$\Lambda = \begin{bmatrix} \lambda_1 & 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \lambda_2 & 1 & 0 & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \lambda_{l-1} & 1 & \cdot & 0 \\ 0 & 0 & 0 & \cdot & 0 & \lambda_l & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \lambda_n \end{bmatrix} \qquad (2.81)$$

한편, 고유치가 l 중근인 경우, $\lambda_1 = \lambda_2 = \cdots = \lambda_l$에 대한 고유벡터는 단근인 경우의 식 (2.79)와 같이

$$\mathbf{AX}_1 = \lambda_1 \mathbf{X}_1 \qquad (2.82)$$

로부터 $\mathbf{X}_1$을 먼저 구한 다음,

고유치(값) - 고유벡터

➲ 고유치가 / 중근인 경우

$$(\lambda_1 = \lambda_2 = \dots = \lambda_l \neq \dots \neq \lambda_n)$$

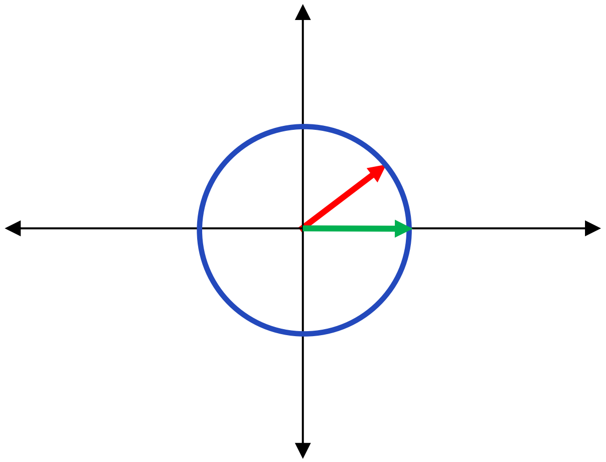
$$
\begin{aligned}
\mathbf{AX}_2 &= \mathbf{X}_1 + \lambda_1 \mathbf{X}_2 \\
\mathbf{AX}_3 &= \mathbf{X}_2 + \lambda_1 \mathbf{X}_3 \\
&\vdots \\
&\vdots \\
\mathbf{AX}_l &= \mathbf{X}_{l-1} + \lambda_1 \mathbf{X}_l
\end{aligned}
\tag{2.83}
$$

또는

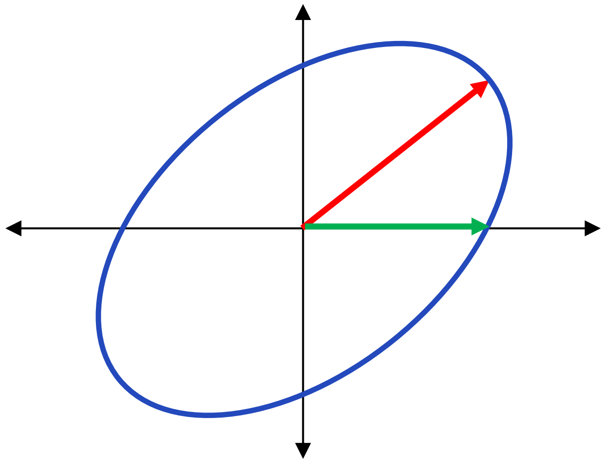
$$
\begin{aligned}
(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{X}_2 &= \mathbf{X}_1 \\
(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{X}_3 &= \mathbf{X}_2 \\
&\vdots \\
&\vdots \\
(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{X}_l &= \mathbf{X}_{l-1}
\end{aligned}
\tag{2.84}
$$

에서 차례대로 구한다. 나머지 $(n-l)$개의 고유치는 단일 고유치를 구하는 방법과 같다.

그래프에서의 행렬 곱셈



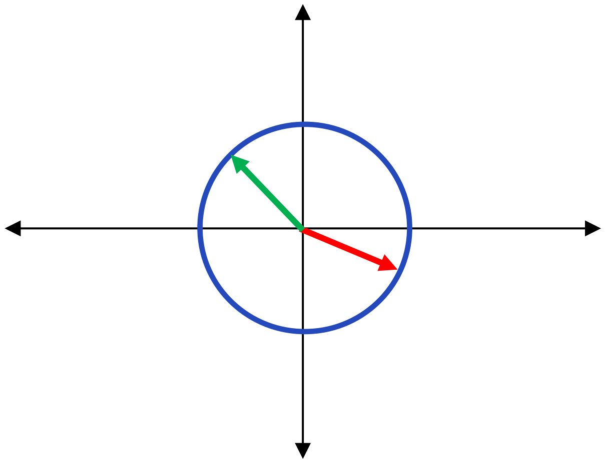
$$A = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix}$$



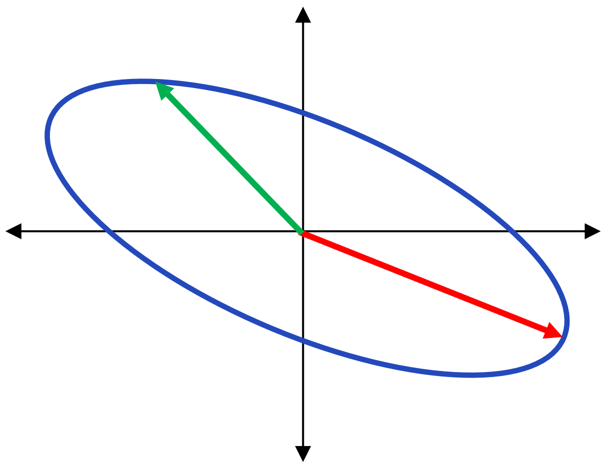
고유치,
고유벡터

$$x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\lambda = 3, 2$$



$$B = \begin{pmatrix} 4 & 2 \\ -1 & 1 \end{pmatrix}$$



고유치,
고유벡터

$$x = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\lambda = 3, 2$$

고유-분해 (eigen – decomposition)

$\lambda_1, \lambda_2, \cdots, \lambda_n$ 이 $n \times n$ 정방행렬 A, 의 *고유값이라면*

$x_1, x_2, \cdots, x_n$ 는 각 고유값 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 에 대응하는 고유벡터이다

$$\mathrm{P} = (x_1 \ \ x_2 \ \ \cdots \ \ x_n) \text{ 이고 } \Sigma = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix},$$

그때 $\mathbf{AP} = \mathbf{P}\,\boldsymbol{\Sigma}$, 그리하여 $\mathbf{A} = \mathbf{P}\,\boldsymbol{\Sigma}\,\mathbf{P}^{-1}$.

즉. 고유벡터 행렬 P 는 행렬 A를 이용하여 대각행렬 $\boldsymbol{\Sigma}$ 를 만들 수 있다.

$$\mathbf{P}^{-1}\mathbf{AP} = \boldsymbol{\Sigma}$$

고유-분해 (eigen – decomposition)

$$\mathbf{A = P\,\Sigma\,P^{-1}}$$

예제)

행렬	고유값 고유벡터	고유-분해
$A = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix}$	$x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $\lambda = 3, 2$	$\begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^{-1}$
$B = \begin{pmatrix} 4 & 2 \\ -1 & 1 \end{pmatrix}$	$x = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ $\lambda = 3, 2$	$\begin{pmatrix} 4 & 2 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}^{-1}$

고유-분해 (eigen – decomposition)

$$\mathbf{A} = \mathbf{P}\,\boldsymbol{\Sigma}\,\mathbf{P}^{-1}$$

고유벡터 $x_1, x_2, \cdots, x_n$에 대해

단위벡터

$$\mathrm{P}e_i = x_i\,,\quad \mathrm{P}^{-1}x_i = e_i,$$

예제)

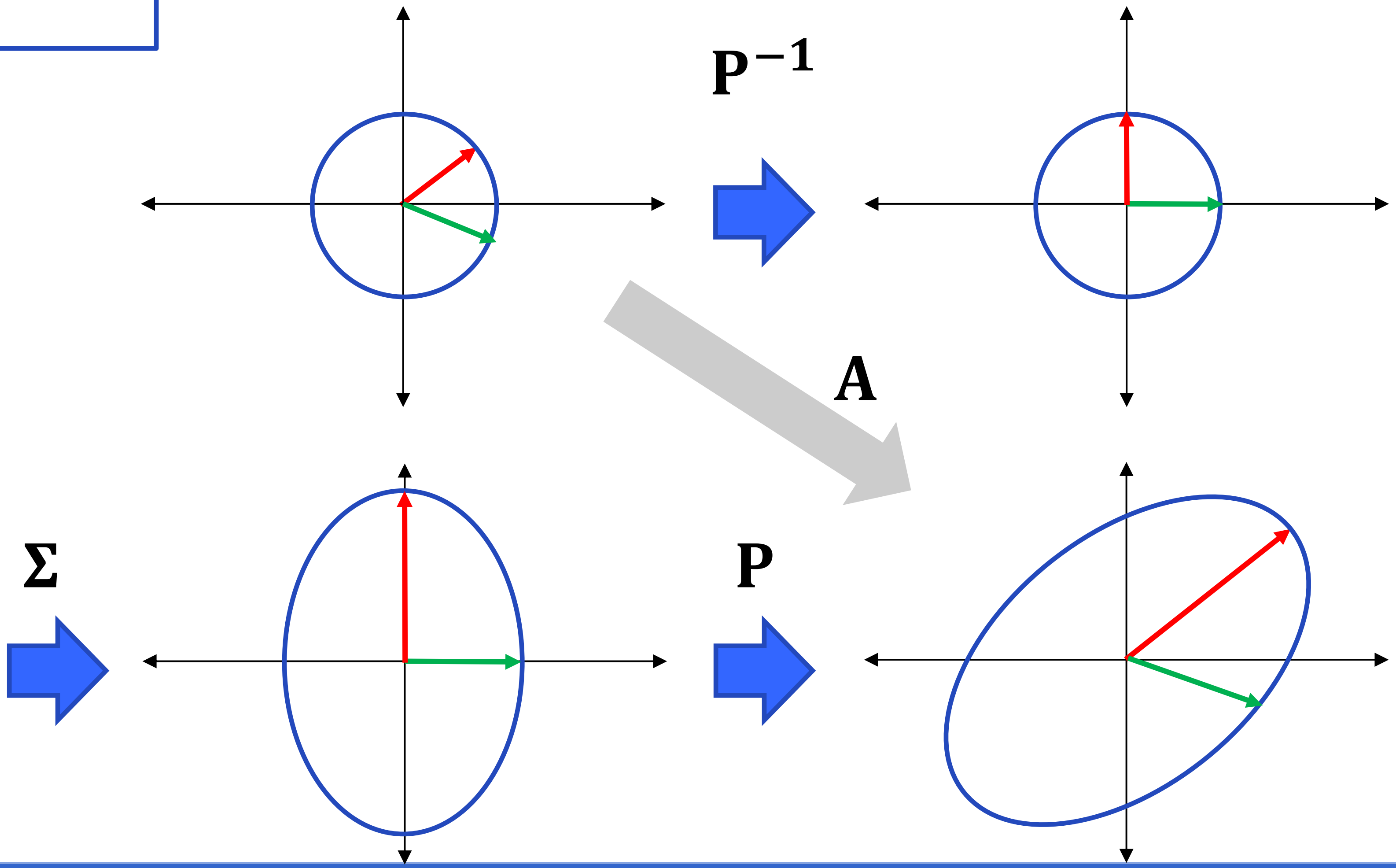
A 의 고유벡터

$$\mathrm{A} = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix} \quad x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \Rightarrow \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

그래프 상에서의 고유-분해의 의미

$$\mathbf{A} = \mathbf{P}\,\boldsymbol{\Sigma}\,\mathbf{P}^{-1}$$



고유-분해의 의미

- $\mathrm{P} = (x_1 \;\; x_2 \;\; \cdots \;\; x_n), \Sigma = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$, $\mathrm{P}^{-1} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$ *일때*

  행렬 $\mathrm{A} = \mathrm{P}\,\Sigma\,\mathrm{P}^{-1}$,

  따라서 $\mathrm{P}\,\Sigma\,\mathrm{P}^{-1} = \boldsymbol{\lambda_1 x_1 y_1 + \lambda_2 x_2 y_2 + \cdots + \lambda_n x_n y_n}$

따라서 고유-분해는 때때로 스펙트럼분해(*the spectral decomposition*.) 라고도 불린다.

예제)

$$\mathrm{A} = \begin{pmatrix} 2 & 2 \\ 0 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} = 3\begin{pmatrix} 2 \\ 1 \end{pmatrix}(0 \;\; 1) + 2\begin{pmatrix} 1 \\ 0 \end{pmatrix}(1 \;\; -2)$$

$$= 3\begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix} + 2\begin{pmatrix} 1 & -2 \\ 0 & 0 \end{pmatrix}$$

고유-분해의 대수적 의미

$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ 일때

$$A = \lambda_1 x_1 y_1 + \lambda_2 x_2 y_2 + \cdots + \lambda_{n-1} x_{n-1} y_{n-1} + \lambda_n x_n y_n$$

중요 ⟷ 무시

특징들

- A 를 분석하기 위해 중요한 특징만 사용한다 → 특징추출(Feature extraction)
- 무시가능 특징들을 버릴 수 있다. → 잡음제거(Denoising)

고유-분해 과정

고유-분해의 과정

입력	고유값	고유벡터	분해
$n \times n$ 정방행렬 A	$\det(A-\lambda I)=0$ 를 만족하는 λ	$(A-\lambda I)x=0$ 를 만족하는 x	P, Σ 행렬 만들기 $A=\mathrm{P}\,\Sigma\,\mathrm{P}^{-1}$ 구성

고유벡터를 찾기 위해, 특성 다항식의 해를 구해야한다.

그러나 3차 이상의 다항식을 푸는 것은 불가능하다