선형대수학팀

3팀
윤지영
이지윤
이수린
임지훈
채희지

INDEX

1

공간 (Space)

1 공간 (Space)

벡터공간 (Vector Space)

벡터공간

여러 벡터들이 모여 형성하는 공간

벡터 원소들끼리 서로 더하거나 주어진 배수로 늘이거나 줄일 수 있는 공간

선형부분공간 *(Subspace)*

벡터공간 안의 몇 가지 벡터의 집합으로 이루는 벡터공간 안의 또 다른 부분 벡터공간

선형부분공간 (Subspace)

선형부분공간 *(Subspace)*

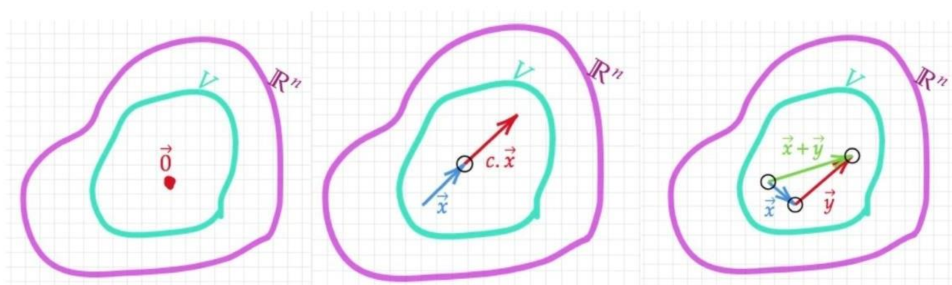
벡터공간 안의 몇 가지 벡터의 집합으로 이루는 벡터공간 안의 또 다른 부분 벡터공간

선형부분공간을 만족하는 조건

① 0벡터(원점)가 존재

② 선형부분공간에 있는 벡터에 다른 벡터를 더했을 때, 그 합도 선형부분공간에 존재

③ 선형부분공간에 있는 벡터에 스칼라를 곱했을 때, 그 곱도 선형부분공간에 존재



선형부분공간을 만족하는 예시

영벡터 *(Zero Vector)*

차원에 상관없이 원소가 영벡터 1개인 벡터공간은 R^n의 가장 작은 subspace

영벡터끼리 더해도 영벡터

영벡터에 어떤 스칼라 값을 곱하든 영벡터

이 집합은 벡터 연산에 닫혀 있음

벡터의 연산 법칙

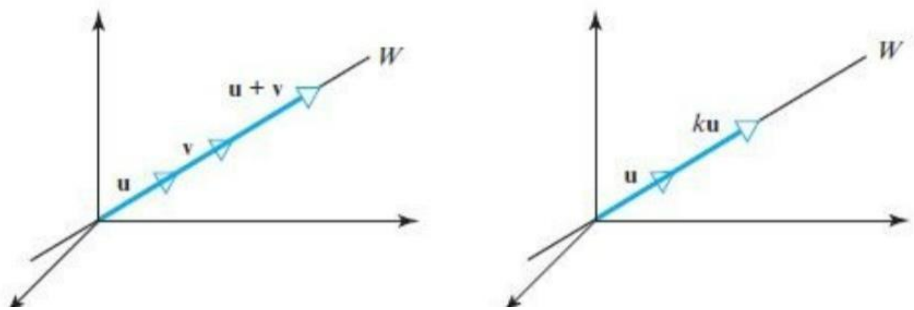
상수배: $cv=\begin{bmatrix} ca \\ cb \end{bmatrix}$ 덧셈: $v+w=\begin{bmatrix} a+c \\ b+d \end{bmatrix}$ 뺄셈: $v-w=\begin{bmatrix} a-c \\ b-d \end{bmatrix}$

벡터의 연산 법칙에 대한 자세한 내용은 선대팀 1주차 참고

1 공간 (Space)

선형부분공간을 만족하는 예시

원점을 지나는 하나의 직선 위에 있는 벡터들



원점을 지나는 하나의 직선 위의 벡터들은 덧셈/곱으로 나머지 성분을 만들 수 있음

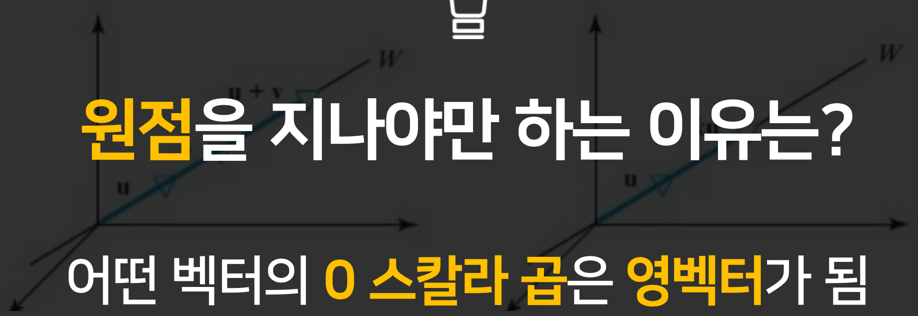
3차원 공간에서 원점을 지나는 평면 (plane) 또한

R^3 의 subspace 평면 안에 있는 벡터끼리 더하든 곱하든 그 평면 내에서만 존재

1 공간 (Space)

선형부분공간 (Subspace)

원점을 지나는 하나의 직선 위에 있는 벡터들



원점을 지나야만 하는 이유는?

어떤 벡터의 **0 스칼라 곱**은 **영벡터**가 됨

∴ 모든 subspace에는 영벡터가 포함되어 있음

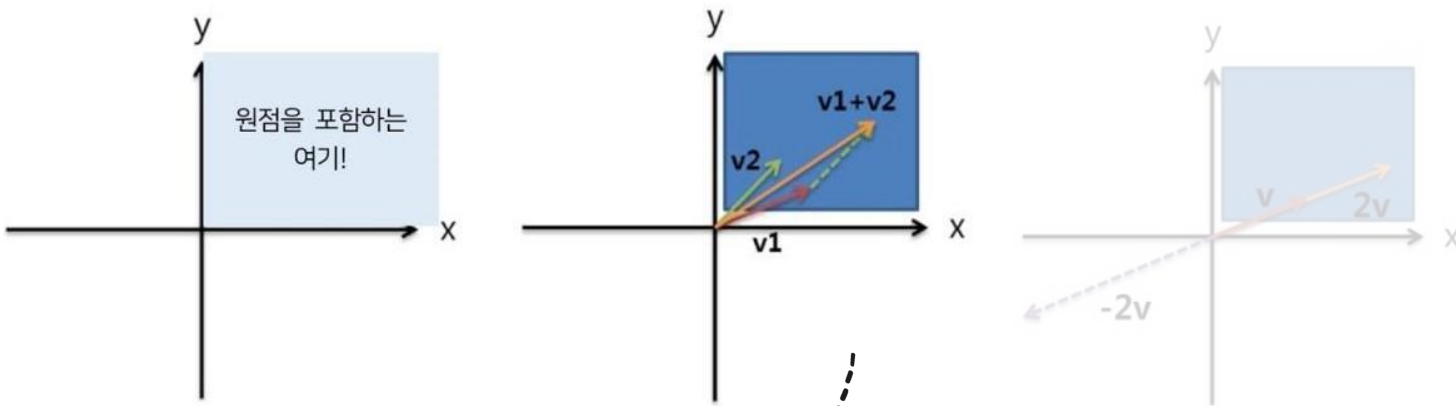
원점을 지나는 하나의 직선 위의 벡터들은 어떤 한 성분의 스칼라 곱이 나머지 성분을 만들 수 있음

3차원 공간에서 원점을 지나는 평면 (plane) 또한

R^3 의 subspace 평면 안에 있는 벡터끼리 더하든 곱하든 그 평면 내에서만 존재

선형부분공간이 아닌 경우

선형부분공간을 만족하는 조건	① 0벡터(원점)가 존재 ② 선형부분공간에 있는 벡터에 다른 벡터를 더했을 때, 그 합도 선형부분공간에 존재 ③ 선형부분공간에 있는 벡터에 스칼라를 곱했을 때, 그 곱도 선형부분공간에 존재



축을 포함한 1사분면은 원점 포함

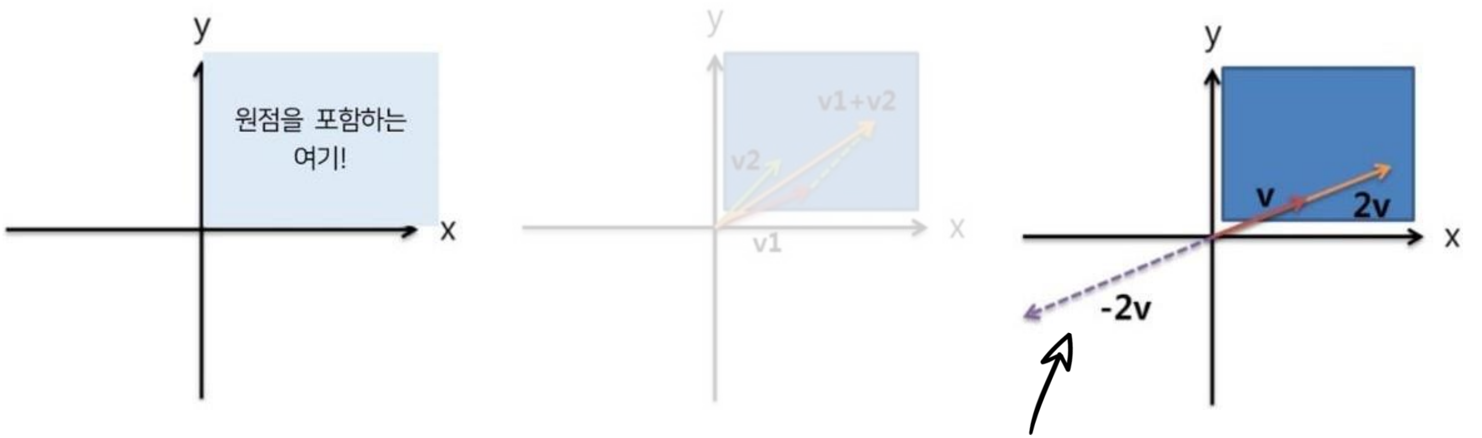
1사분면 내의 벡터끼리 더하면 1사분면에 위치

조건 ①, ② 만족

1 공간 (Space)

선형부분공간이 아닌 경우

선형부분공간을 만족하는 조건	① 0벡터(원점)가 존재 ② 선형부분공간에 있는 벡터에 다른 벡터를 더했을 때, 그 합도 선형부분공간에 존재 ③ 선형부분공간에 있는 벡터에 스칼라를 곱했을 때, 그 곱도 선형부분공간에 존재



1사분면 내의 벡터에 음수 스칼라를 곱하면 1사분면을 벗어남

조건 ③ 만족 X → 곱셈연산에 닫혀있음

열공간 (Column space)

열공간 (Column space)

행렬의 열들의 선형 결합을 통해 만들 수 있는 vector space

EXAMPLE

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \xrightarrow{Ax = b} x\begin{bmatrix} 2 \\ -1 \end{bmatrix} + y\begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

모든 b에 대해서 x solution이 존재

b_1, b_2가 무엇이든 x, y 구할 수 있음

1 공간 (Space)

열공간 (Column space)

열공간 (Column space)

행렬의 열들의 선형 결합을 통해 만들 수 있는 vector space

모든 b에 대해 x 집합 존재

EXAMPLE

$Ax = b$

$\begin{bmatrix}2 & -1\\ 1 & 2\end{bmatrix} \longrightarrow x\begin{bmatrix}2\\ 1\end{bmatrix} + y\begin{bmatrix}-1\\ 2\end{bmatrix} = \begin{bmatrix}b_1\\ b_2\end{bmatrix}$

행렬 A의 두 개의 column의 linear combination으로

모든 R^2 의 공간에 있는 벡터 만들 수 있다는 것을 의미

모든 b에 x solution이 존재

b_1, b_2가 무엇이 오든 x, y 구할 수 있음

열공간 (Column space)

열공간 (Column space)

행렬의 **열들의 선형 결합**을 통해 만들 수 있는 vector space

EXAMPLE

어떤 b에 대해서 solution **구하지 못할** 수 있음

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \xrightarrow{Ax = b} x \boxed{\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}} + y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

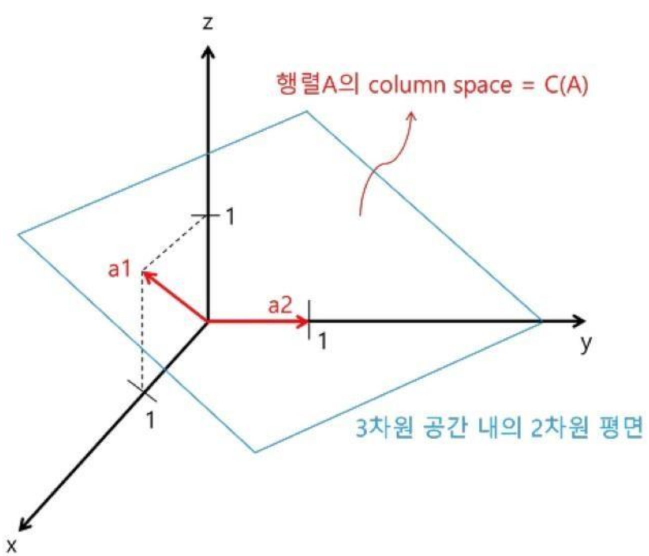
b vector의 첫 번째 성분 = 세 번째 성분

→ x, y 값을 구할 수 있음

열공간 (Column space)

열공간 (Column space)

행렬의 열들의 선형 결합을 통해 만들 수 있는 vector space



우리가 구할 수 있는 b는

linear combination으로 만들 수 있는 것 뿐

1 공간 (Space)

Subspace의 종류

Ax=b가 있다

벡터 b가 col(A)에 있다

모든 b에 대해 해를 갖는다

Col(A) = R^n

모든 b에 대해 해 X

Col(A)

= n보다 작은 차원의 subspace

영공간 (Null Space)

영공간 *(Null space)*

Ax=0을 만족시키는 모든 x solutions의 집합

$$Ax = 0 \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A의 어떤 행렬의 Null space에 영벡터가 무조건 포함

1 공간 (Space)

영공간 (Null Space)

영공간 *(Null space)*

Ax=0을 만족시키는 모든 x solutions의 집합

$$Ax = 0 \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A의 어떤 행렬의 Null space에 영벡터가 무조건 포함

영벡터를 제외한

행렬 A의 column들의 어떤 linear combination이 영벡터를 만들까?

영공간 (Null Space)

두 column의 linear combination으로 나머지 하나의 벡터를 만들 수 있으면

0벡터 말고도 다른 해 [x,y,z] 찾을 수 있음

$$x\begin{bmatrix}1\\0\\1\end{bmatrix}+y\begin{bmatrix}0\\1\\1\end{bmatrix}+z\begin{bmatrix}1\\1\\2\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix} \longrightarrow x\begin{bmatrix}1\\0\\1\end{bmatrix}+y\begin{bmatrix}0\\1\\1\end{bmatrix}=-z\begin{bmatrix}1\\1\\2\end{bmatrix}$$

해 벡터들은 선형부분공간의 조건 만족

(① 해에 0벡터가 포함 ② 상수배나 해벡터끼리의 합도 해의 집합에 포함되기 때문)

Null space도 선형부분공간 중 하나

행공간과 Left Null Space

행공간 *(Row Space)*

행렬의 row의 linear combination 을 통해 만들 수 있는 subspace

A^T의 column space = A의 row space

Left Null Space

A^T의 Null space

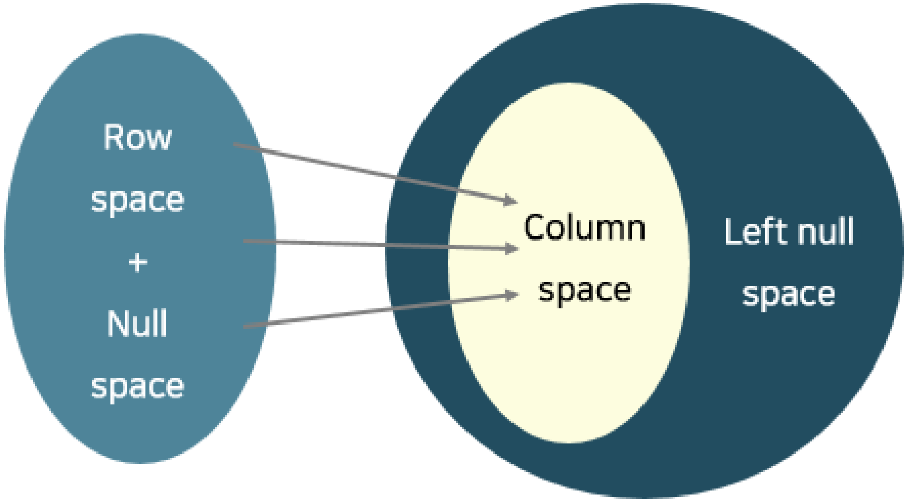
1 공간 (Space)

함수의 개념 적용

선형변환 = 함수

함수: 정의역의 원소들이 함수를 통해 치역에 대응

선형대수: 행공간과 영공간의 원소들이 행렬을 통해 열공간으로 매핑



$$f : X \rightarrow Y$$

선형독립과 선형종속

n개의 벡터에 대한 **선형결합**

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n = 0$$

선형독립 *(Linearly Independent)*

상수 c가 모두 0인 경우를 제외한 어떠한 c_i의 조합으로 식이 성립되지 않는 경우

$$c_1 = c_2 = \cdots = c_n = 0$$

trivial solution만을 가질 때 **A의 각 열벡터**들은 선형독립

선형종속 *(Linearly Dependent)*

0이 아닌 상수 c가 존재

선형독립과 선형종속

n개의 벡터에 대한 **선형결합**

$$c_1 x_1 + c_2 x_2 + \ldots + c_n x_n = 0$$

선형독립 *(Linearly Independent)*

상수 c가 모두 0인 경우를 제외한 어떠한 c_i의 조합으로 식이 성립되지 않는 경우

$$c_1 = c_2 = \cdots = c_n = 0$$

trivial solution만을 가질 때 **A의 각 열벡터**들은 선형독립

선형종속 *(Linearly Dependent)*

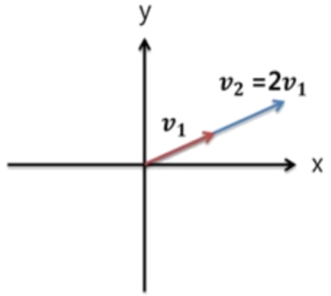
0이 아닌 상수 c가 존재

1 공간 (Space)

선형독립과 선형종속

선형종속인 경우

2차원



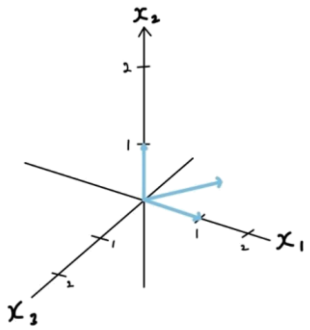
c1v1 + c2v2 = 0는

c1 = 2, c2 =-1일 때 성립

⋮

0벡터가 아닌 벡터c (2,-1)에 의해 식 성립

3차원



(1,2,0) 처럼 (0,0,0) 이외에도 해가 존재

⋮

하나의 벡터를

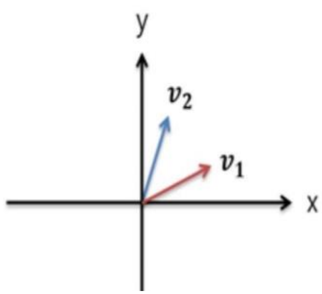
다른 벡터의 조합으로 표현 가능

1 공간 (Space)

선형독립과 선형종속

선형독립인 경우

2차원



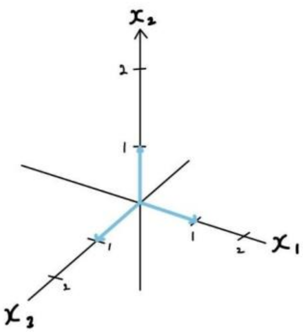
어떤 선형결합을 해도

c가 0인 경우를 제외하고는 0 만들 수 없음

⋮

벡터v1과 v2는 독립

3차원



(1,0,0) (0,1,0) (0,0,1)로

0벡터가 유일한 해

⋮

벡터 x1, x2, x3는 선형독립

Rank

Rank

Col(A)에서 찾을 수 있는 Linearly independent한 벡터의 최대 개수

쉽게 생각하면 Column space의 차원!

EXAMPLE

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

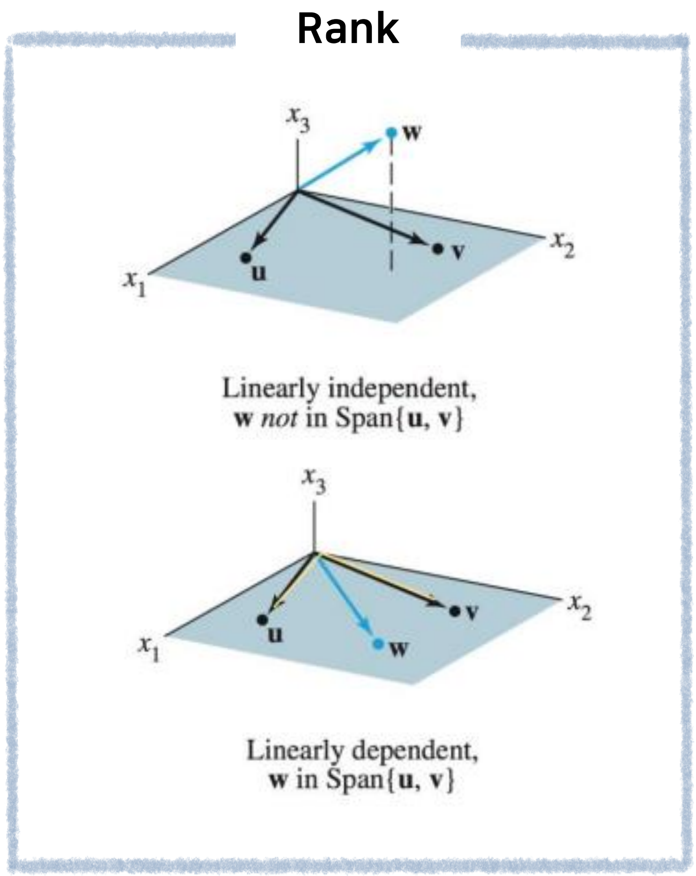
Rank: 2

$$\begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}$$

Rank: 1

Rank

R^n 공간에서 Linearly independent한 벡터의 최대 개수는 n개



u, v 벡터가 linearly independent

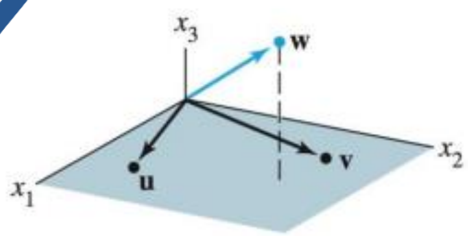
→ 같은 직선 상에 위치 **X**

→ u의 스칼라배로 v를 만듦 **X**

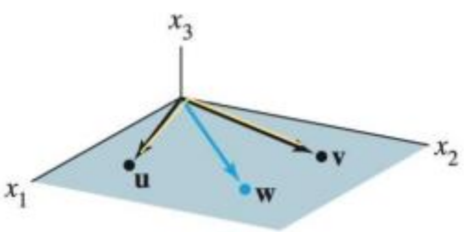
Rank

R^n 공간에서 Linearly independent한 벡터의 최대 개수는 n개

Rank



Linearly independent,
w *not* in Span{**u**, **v**}



Linearly dependent,
w in Span{**u**, **v**}

u, v 벡터가 linearly independent

→ 같은 직선 상에 위치 **X**

→ u의 스칼라배로 v를 만듦 **X**

⋮

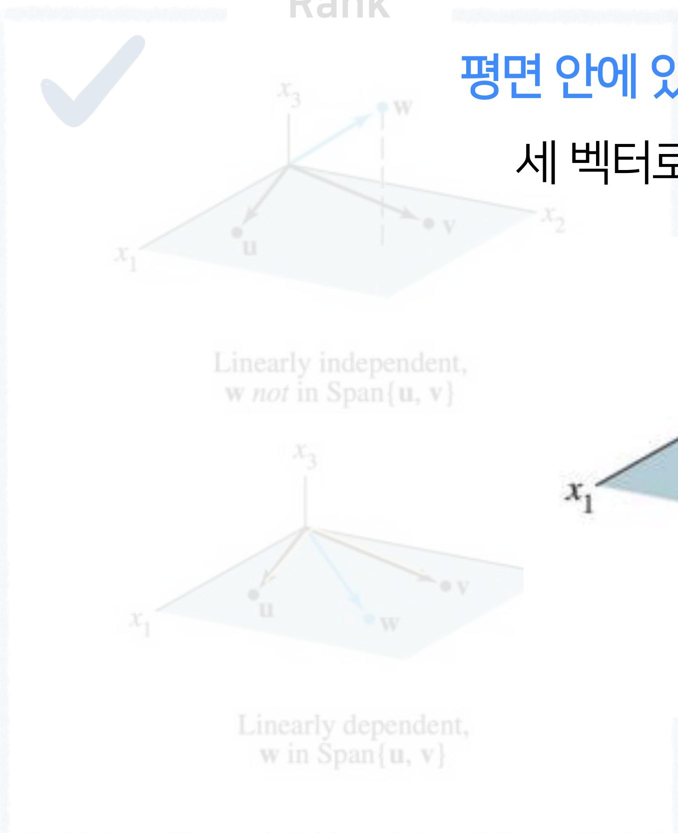
u, v 벡터가 선형독립

u, v 벡터로 파란색 평면 위의 **모든 표현 가능**

Rank

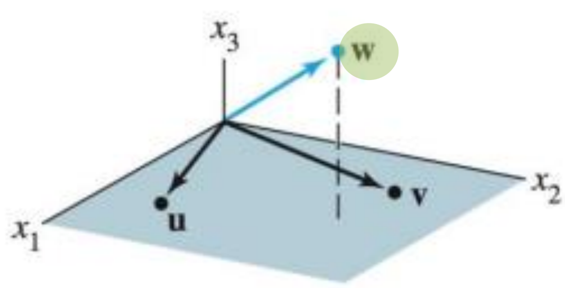
R^n 공간에서 Linearly independent한 벡터의 최대 개수는 n개

Rank



평면 안에 있지 않은(독립) w 벡터를 잡아

세 벡터로 R^3 **공간 전체 표현** 가능 !



Linearly independent,
w *not* in Span{**u**, **v**}

1 공간 (Space)

Rank

따라서…

A 행렬의 rank = A^T 행렬의 rank

Row(A)로 생성되는 벡터 공간의 차원

=

Col(A)으로 생성되는 벡터 공간의 차원

공간 형성 (Span)

공간 형성 *(Span)*

성분 $a_1, a_2, \cdots, a_n$의 모든 선형결합을 조합한 공간

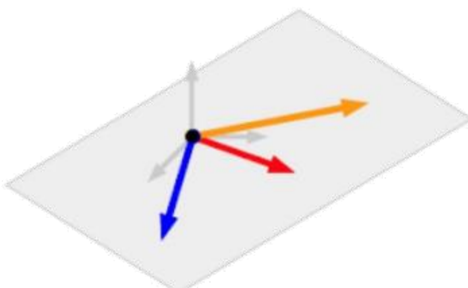
앞서 선형부분공간이란…

벡터공간 안의 몇 가지 벡터의 집합으로 이루는 벡터공간 안의 또 다른 부분 벡터공간

1 공간 (Space)

공간 형성 (Span)

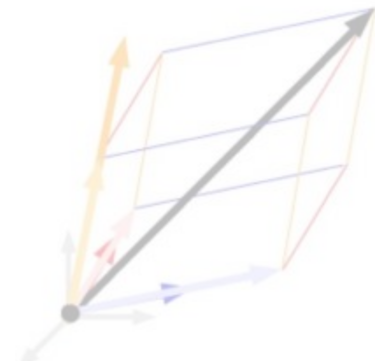
2차원



빨간색, 파란색 벡터 → 평면 공간만을 span

주황색 벡터가 평면에 추가된다면
빨간색 벡터와 파란색 벡터로 표현 가능

3차원



주황색 벡터가 평면 공간 밖에
있어야 R^3 공간 span 가능!

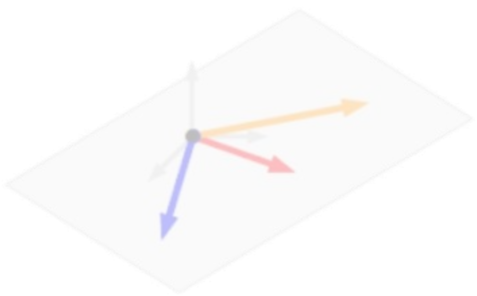
세 벡터로 span할 수 있는 공간은 여전히 평면 뿐

3차원 공간 형성

1 공간 (Space)

공간 형성 (Span)

2차원

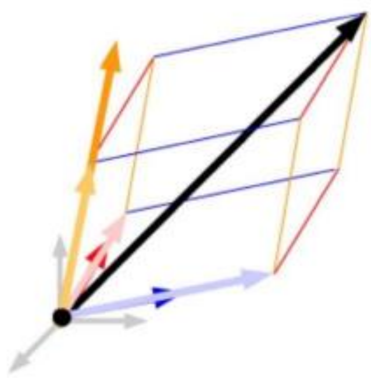


빨간색, 파란색 벡터 → 평면 공간만을 span

만약 주황색 벡터가 추가된다면
빨간색 벡터와 파란색 벡터로 표현 가능

세 벡터로 span할 수 있는 공간은 여전히 평면 뿐

3차원



주황색 벡터가 평면 공간 밖에
있어야 R^3 공간 span 가능!

3차원 공간 형성

공간 형성 (Span)

그래서 이걸 왜 배운…?

$span$과 선형독립의 개념을 이용하여

기저(Basis)에 대해 알아보자!

기저 (Basis)

기저 *(Basis)*

어떤 공간을 구성하는 **span 벡터의 최소 집합**

기존 벡터들의 조합으로 설명되는 벡터를 추가하는 건 의미 X

→ 최소집합 등장

최소집합이란…

벡터 공간을 효율적으로, 최소한으로 표현할 수 있는 벡터들이 기저

기저 (Basis)

기저 *(Basis)*

어떤 공간을 구성하는 **span 벡터의 최소 집합**

기존 벡터들의 조합으로 설명되는 벡터(종속벡터)를 추가하는 건 의미 X

최소 집합: 벡터 공간을 효율적으로, 최소한으로 표현할 수 있는 벡터들이 기저

기저의 조건

① Linearly independent.

② 해당 벡터들이 V 공간을 span

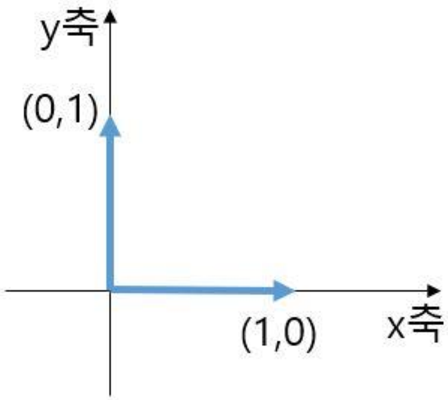
그렇구나…

1 공간 (Space)

기저 (Basis)

기저의 예시

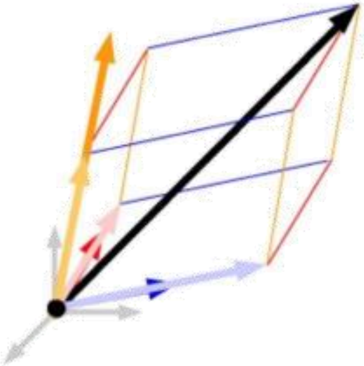
2차원



두 개의 벡터는 **선형독립**

∴ 각 두 벡터가 R^2 공간의 basis가 됨.

3차원



세 개의 벡터는 **선형독립**

∴ 각 세 벡터가 R^3 공간의 basis가 됨.

기저 (Basis)

Non-uniqueness of Basis

기저 예시

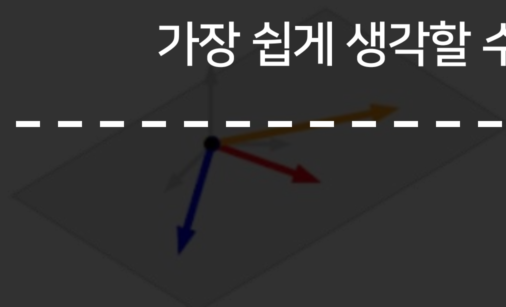
2차원

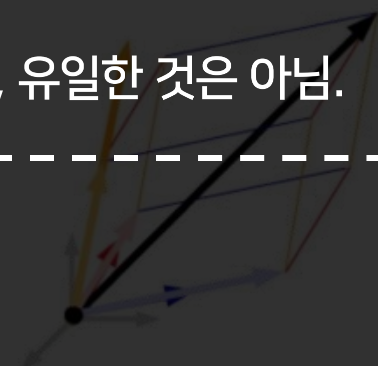
R^2 기저 벡터 : $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

3차원

$\rightarrow R^2$ 공간 span

가장 쉽게 생각할 수 있는 기저 벡터일 뿐, 유일한 것은 아님.





R^n 공간에서의 선형독립인 **어느 n개의 벡터든 R^n span 가능**

주황색 벡터를 추가

선형독립의 가정을 위배

세 개의 벡터는 **선형독립**

어떤 공간을 span하는 기저 벡터는 유일 X

∴ 각 세 벡터가 R^3 공간의 basis가 됨.

1 공간 (Space)

기저 (Basis)와 rank의 관계

Example

$\begin{bmatrix}1\\2\end{bmatrix} \begin{bmatrix}4\\1\end{bmatrix} \begin{bmatrix}3\\1\end{bmatrix}$ $\dashrightarrow$ $\underline{\begin{bmatrix}1\\2\end{bmatrix} \begin{bmatrix}4\\1\end{bmatrix}} \begin{bmatrix}3\\1\end{bmatrix}$

이 공간의 기저 벡터 후보

선형결합으로 마지막 열 생성 가능

∴ 세 벡터는 linearly independent X

$\begin{bmatrix}1\\2\end{bmatrix} \begin{bmatrix}4\\1\end{bmatrix} \begin{bmatrix}3\\1\end{bmatrix}$

1,3번째 열 또한 basis 가능

1 공간 (Space)

기저 (Basis)와 rank의 관계

Example

$\begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}4\\1\end{bmatrix}\begin{bmatrix}3\\1\end{bmatrix}$

이 공간의 기저 벡터 후보

$\begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}4\\1\end{bmatrix}\boxed{\begin{matrix}3\\1\end{matrix}}$

선형결합으로 마지막 열 생성 가능

$\begin{bmatrix}1\\2\end{bmatrix}\boxed{\begin{matrix}4\\1\end{matrix}}\begin{bmatrix}3\\1\end{bmatrix}$

1,3번째 열 또한 basis 가능

어떤 basis를 잡더라도 그 개수는 동일

어떤 공간의 선형독립인 벡터의

최대 개수인 rank와 동일

2

노름과 내적

노름(Norms)

노름 *(Norm)*

벡터의 크기

$$\|\vec{v}\|_p = L_p = \sqrt[p]{|v_1|^p + |v_2|^p + \cdots + |v_n|^p}$$

p = Norm의 차수

선형대수학에서는 위와 같이 정의

일반적으로 p=1인 L1 Norm과 p=2인 L2 Norm이 자주 사용됨

3 노름과 내적, 직교성

노름(Norms)

노름

L_1 Norm

: 맨해튼 노름

각 벡터들의 절댓값을 합한 결과

Lasso Regression에서 사용

$$L_1 = |v_1| + |v_2| + \cdots + |v_n|$$

L_2 Norm

: 유클리드 노름

원점에서 벡터가 연결된 직선 거리

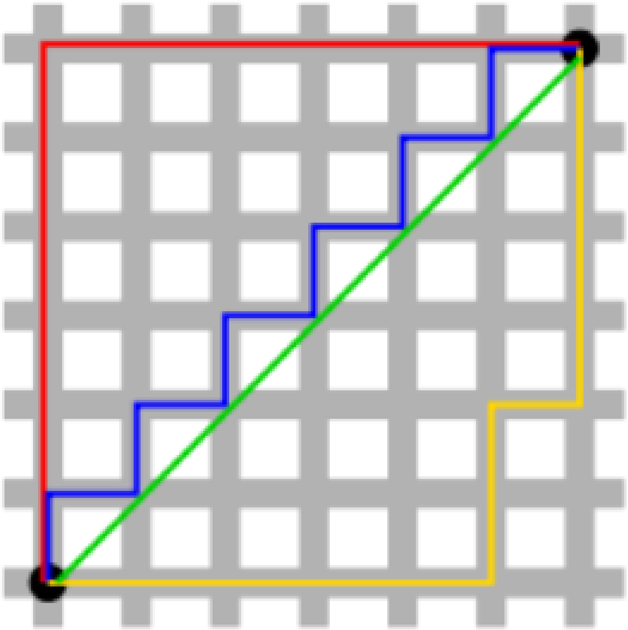
Ridge Regression에서 사용

$$L_2 = \sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}$$

자세한 내용은 회귀분석팀 클린업 참고

3 노름과 내적, 직교성

노름(Norms)



L_1 Norm (맨해튼 노름): — — —

L_2 Norm (유클리드 노름): —

주로 **제약 조건 설정** 시 많이 쓰임

자세한 내용은 회귀분석팀 클린업 참고

내적(Dot product)

내적 *(Dot Product)*

$$① \; x^T y = [x_1 \;\; x_2 \;\; \cdots x_n] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = [x_1 y_1 + x_2 y_2 + \cdots + x_n y_n] = 0$$

$$② \; a \cdot b = \|a\| \|b\| \cos\theta$$

내적의 결과: 스칼라 값

$x^T y = x, y$ 벡터의 각 성분을 곱한 것들의 합

⋮

내적의 기하학 정의를 통해 두 벡터의 각도 θ를 계산 가능

$$\theta = \cos^{-1}\left(\frac{a \cdot b}{\|a\| \|b\|}\right)$$

3

직교성

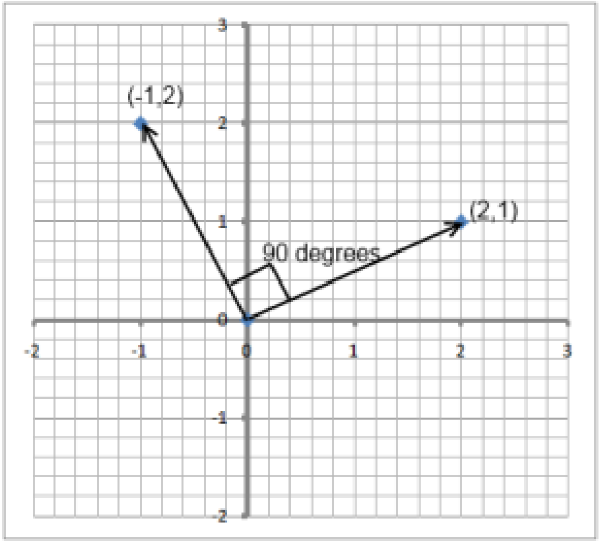
직교성 (Orthogonality)

직교성 *(Orthogonality)*

두 벡터가 이루는 각도 θ가 90°가 되는 경우

두 벡터의 내적 = 0

$$a \cdot b = a^T b = 0$$



$$a \cdot b = \|a\|\|b\| \cos\theta$$

cos 90°거나 영벡터가 아닌 이상

0 될 수 없음

EXAMPLE

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

3 직교성 (Orthogonality)

Space의 Orthogonality

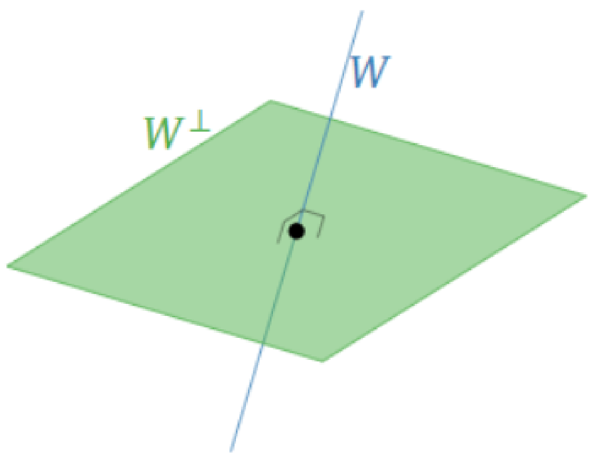
Subspace S $\perp$ Subspace T

=

S 안의 모든 벡터 $\perp$ T 안의 모든 벡터

직교성 (Orthogonality)

1차원 파란색 선 ⊥ 2차원 초록색 평면



1차원 선 ⊥ 2차원 평면

평면과 직선이 **수직**으로 만날 때 뿐!

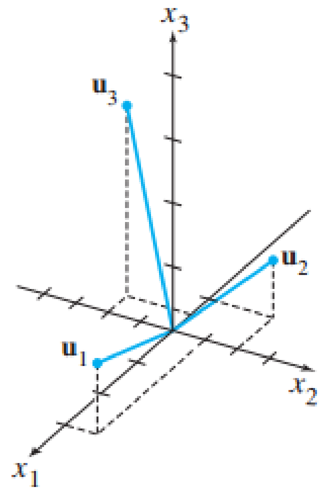
초록색 평면에 어떤 벡터를 그려도

파란색 선 위의 모든 벡터와 수직

직교집합 (Orthogonal set)

직교 집합 *(Orthogonal Set)*

서로 직교하는 벡터들의 집합



직교 집합의 벡터들은 서로 선형 독립

span으로 만들어지는 벡터공간의 basis가 됨.

Ax=b의 해 구하기

배운 것들을 이용하여 Ax=b를 다시 한 번…!

$$\begin{bmatrix} 1 & 1 & 2 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \longrightarrow x_1 \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Ax = b

Linear Combination

3 직교성 (Orthogonality)

Ax=b의 해 구하기

"Ax=b의 해(x)를 찾을 수 있다"

무슨 의미일까?

A 행렬의 Column의 Linear combination으로 b 벡터를 만들 수 있다

A의 Column space에 b가 있다

A 행렬을 통해 b 라는 벡터를 매핑 시킬 수 있는 x벡터를 찾을 수 있다.

1주차 선형변환의 원리!

3 직교성 (Orthogonality)

Ax=b의 해 구하기

“Ax=b의 해(x)를 찾을 수 있다”

무슨 의미일까?

A 행렬의 Column의 Linear combination으로 b 벡터를 만들 수 있다

A의 Column space에 b가 있다

A 행렬을 통해 b 라는 벡터를 매핑 시킬 수 있는 x벡터를 찾을 수 있다.

1주차 선형변환의 원리!

3 직교성 (Orthogonality)

Ax=b의 해 구하기

"Ax=b의 해(x)를 찾을 수 있다"

무슨 의미일까?

A 행렬의 Column의 Linear combination으로 b 벡터를 만들 수 있다

A의 Column space에 b가 있다

A 행렬을 통해 b 라는 벡터를 매핑 시킬 수 있는 x벡터를 찾을 수 있다.

1주차 선형변환의 원리!

Ax=b의 해 구하기

3X3 행렬 A의 Column Space 후보

영공간	3차원 공간	1차원 직선공간	2차원 평면공간

3 직교성 (Orthogonality)

Ax=b의 해 구하기

3X3 행렬 A의 Column Space 후보

영공간	3차원 공간	1차원 직선공간	2차원 평면공간

행렬 A의 Column 모두 선형독립

3개의 벡터로 span 가능한 **최대 공간**

EXAMPLE $\begin{bmatrix}1\\3\\2\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$ $\begin{bmatrix}2\\0\\1\end{bmatrix}$

Ax=b의 해 구하기

3X3 행렬 A의 Column Space 후보

A 행렬의 모든 column이

Linearly independent 하다

영공간	3차원 공간	1차원 직선공간	2차원 평면공간

A의 역행렬이 존재하며

A 행렬의 rank = column 벡터의 차원

행렬 A의 Column 모두 선형독립

$$\begin{bmatrix}1\\3\\2\end{bmatrix} \quad \begin{bmatrix}1\\1\\1\end{bmatrix} \quad \begin{bmatrix}2\\0\\1\end{bmatrix}$$

Ax=b의 해가 제한적인 경우

$$\begin{bmatrix} 1 & 1 & 2 \\ 1 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \dashrightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

1번째 Column + 2번째 Column = 마지막 Column

① Linearly Independent X

② 행렬 A의 Column 개수(3) ≠ Basis의 개수(2)

(2차원 평면 밖의 벡터 표현 불가)

4

투영 (Projection)

회귀식 (Regression Equation)

회귀식 *(Regression Equation)*

선형대수학의 $Ax = b$ 형태를 회귀식의 $X\beta = y$에 적용

$$y = X\beta$$

행렬의 형태로 표시한 경우

y : 실제 예측하고자 하는 종속변수

X : 독립변수들의 집합 X data

β : 각 변수들의 앞에 붙는 베타 계수

Why 투영 (Projection)?

어떤 관계가 있는지 모르는 X로 y를 예측할 때

오차의 발생 불가피

오차를 **최소화**해야 함

$$Y = X\beta + \varepsilon$$

이런 상황에서 Projection

Y data를 최대한 표현할 수 있는 벡터 찾기!

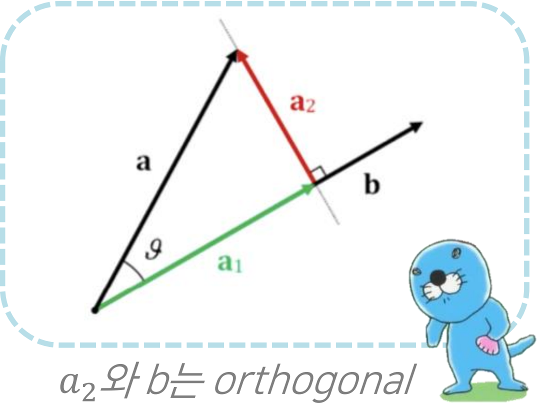
투영 (Projection)

벡터-벡터 Projection

한 벡터의 방향으로 만드는 직선 공간에 다른 벡터를 매핑 시키는 것

벡터 b 방향의 직선공간으로 공간 압축시키는 선형 변환

벡터를 projection 시키면 원래 벡터와의 차이가 생김



a_2와 b는 orthogonal

Error: 원래 벡터(a)와의 차이(a_2)

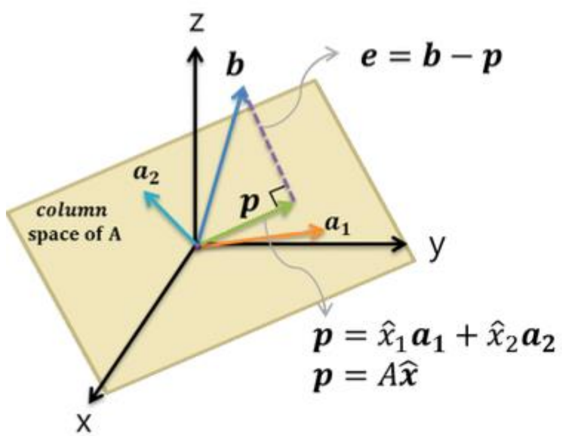
Error가 최소가 되도록 내리는 것이 Projection Vector

투영 (Projection)

벡터-공간 Projection

벡터-벡터 투영에서 확장하여, 벡터를 투영시켰을 때 나오는 벡터와

원래 벡터 간의 차이(error)를 최소화하는 것



① b를 col(A)평면에 투영하여 p 도출

② 벡터 a_1과 a_2의 선형결합으로 p 표현

⋮

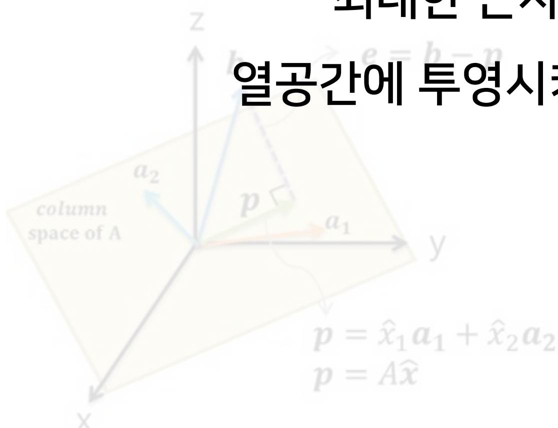
error vector와 평면 orthogonal

4 투영 (Projection)

투영 (Projection)

벡터-공간 Projection

벡터-벡터 투영에서 확장하여, 벡터를 투영시켰을 때 나오는 벡터와 원래 벡터 간의 차이(error)를 최소화하는 것



② 벡터 a_1과 a_2의 선형결합으로 p 표현

⋮

error vector와 평면 orthogonal

최대한 근사한 해를 찾기 위해 최단거리로

열공간에 투영시켜 차원을 낮추는 투영의 원리 적용!

Projection의 회귀 적용

Projection으로 $\hat{Y}$ 구하기

Y 벡터를 X의 Column Space로 Projection 시킨 벡터를 찾아 가장 **근접한 해**로 둠

$$X\beta = y$$

해를 찾을 수 없음 , x Column Space 내에 y가 없는 경우는?

$$X\hat{\beta} = \hat{y}$$

해를 찾을 수 있는 방정식으로 대체

가장 y와 근사한 $\hat{y}$를 기반으로 $\hat{\beta}$도출

Projection의 회귀 적용

Projection으로 $\hat{Y}$ 구하기

Y 벡터를 X의 Column Space로 Projection 시킨 벡터를 찾아 가장 **근접한** 해로 둠

$$\boldsymbol{X\beta} = \boldsymbol{y}$$

해를 찾을 수 없음 즉, x Column Space 내에 y가 없는 경우는?

$$X\hat{\beta} = \hat{y}$$

해를 찾을 수 있는 방정식으로 대체

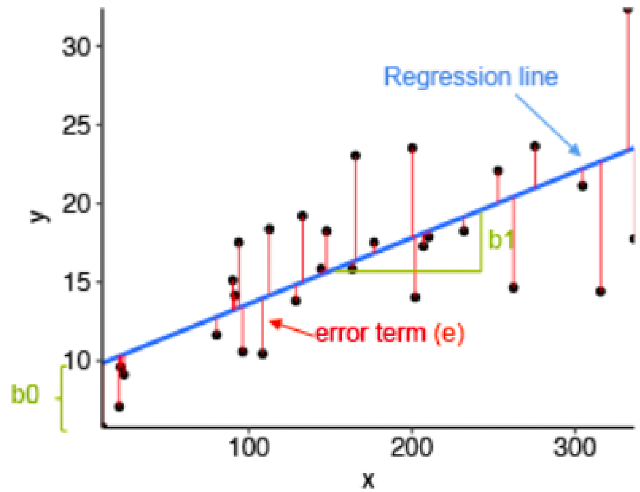
가장 y와 근사한 $\hat{y}$를 기반으로 $\hat{\beta}$도출

Projection의 회귀 적용

모델의 목표

잔차가 평균이 0으로 회귀하는 정확한 모델을 만드는 것

Error 최소화 시키는 추정값 $\hat{\beta}$ 구하기



Error vector와 X의 Column Space의 **Orthogonality** 이용

4 투영 (Projection)

Projection의 회귀 적용

최적의 해를 구할 때 쓰이는 성질

① $e = y - \hat{y} = y - X\hat{\beta}$

② error vector와 X의 Column Space는 Orthogonal

③ x vector의 모든 Column과의 내적값이 모두 0

해를 구하는 과정

$$X^T(y - X\hat{\beta}) = 0$$

$$X^T y = X^T X\hat{\beta}$$

$$(X^T X)^{-1} X^T y = \hat{\beta}$$

해를 도출할 수 있다!

$$\longrightarrow \quad X(X^T X)^{-1} X^T y = X\hat{\beta} = \hat{y}$$

4 투영 (Projection)

Projection의 회귀 적용

최적의 해를 구할 때 쓰이는 성질

① $e = y - \hat{y} = y - X\hat{\beta}$

② error vector와 X의 Column Space는 Orthogonal

③ x vector의 모든 Column과의 내적값이 모두 0

해를 구하는 과정

$$X^T(y - X\hat{\beta}) = 0$$

$$X^T y = X^T X\hat{\beta}$$

$$(X^T X)^{-1} X^T y = \hat{\beta}$$

해를 도출할 수 있다!

$$X(X^T X)^{-1} X^T y = X\hat{\beta} = \hat{y}$$

Projection 변환을 가하는

Projection Matrix

5

회귀적용

잔차제곱을 최소화하는 β 계수 찾기

β값 찾기

① 회귀식 $y = \beta_0 + \beta_1 x$를 $\boldsymbol{X\beta = y}$ **형태**로 바꿈 $(Ax = b)$

② Ax=b에서 해를 찾을 수 있는 방정식 $A\hat{x} = \hat{y}$ 으로 대체

Predicted Y value	Observed Y value
$\beta_0 + \beta_1 x_1$	y_1
$\vdots$	$\vdots$
$\beta_0 + \beta_1 x_n$	y_n

$$X\beta = y, \text{where X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

예측값 $\neq$ 실제값

$\rightarrow X\beta = y$은 해가 존재 안함

5 회귀적용

잔차제곱을 최소화하는 β 계수 찾기

β값 찾기

① 회귀식 $y = \beta_0 + \beta_1 x$를 $\boldsymbol{X\beta = y}$ **형태**로 바꿈 $(Ax = b)$

② Ax=b에서 해를 찾을 수 있는 방정식 $A\hat{x} = \hat{y}$ 으로 대체

Least-Square method를 사용!

$X\beta$와 y의 최소 거리인 간접해를 구함

Predicted Y value	Observed Y value
$\beta_0 + \beta_1 x_1$	[illegible]
$\vdots$	$\vdots$
$\beta_0 + \beta_1 x_n$	y_n

$$X\beta = y, \text{where } \mathrm{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

예측값 $\neq$ 실제값

$\rightarrow X\beta = y$은 해가 존재 안함

잔차제곱을 최소화하는 β 계수 찾기

최소제곱법 *(Least Square Estimator Method)*

y_i와 회귀선 위의 y값의 거리의 제곱합이 **최소가 되도록** 하는 추정치를 찾는 방법

다중선형회귀에서는 행렬을 이용

$$\begin{pmatrix}1 & 1\\1 & 2\\1 & 3\end{pmatrix}\begin{pmatrix}a\\b\end{pmatrix}=\begin{pmatrix}1\\2\\2\end{pmatrix}\rightarrow\begin{pmatrix}1 & 1\\1 & 2\\1 & 3\end{pmatrix}\begin{pmatrix}\hat{a}\\\hat{b}\end{pmatrix}=\begin{pmatrix}\widehat{y_1}\\\widehat{y_2}\\\widehat{y_3}\end{pmatrix}$$

추정치

β값 찾기

① 회귀식 $y=\beta_0+\beta_1 x$를 $X\beta=y$ 형태로 바꿈 $(Ax=b)$

② Ax=b에서 해를 찾을 수 있는 방정식 $A\hat{x}=\hat{y}$ 으로 대체

$$A^T Ax = A^T b$$
$$X^T X\beta = X^T y$$

해 (x,β)를 찾을 수 있음

잔차제곱을 최소화하는 β 계수 찾기

최소제곱법 *(Least Square Estimator Method)*

y_i와 회귀선 위의 y값의 거리의 제곱합이 **최소가 되도록** 하는 추정치를 찾는 방법

다중선형회귀에서는 행렬을 이용

$$\begin{pmatrix}1 & 1\\ 1 & 2\\ 1 & 3\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix} = \begin{pmatrix}1\\ 2\\ 2\end{pmatrix} \rightarrow \begin{pmatrix}1 & 1\\ 1 & 2\\ 1 & 3\end{pmatrix}\begin{pmatrix}\hat{a}\\ \hat{b}\end{pmatrix} = \begin{pmatrix}\widehat{y_1}\\ \widehat{y_2}\\ \widehat{y_3}\end{pmatrix}$$

추정치

β값 찾기

① 회귀식 $y = \beta_0 + \beta_1 x$를 $\boldsymbol{X\beta = y}$ 형태로 바꿈 $(Ax = b)$

② Ax=b에서 해를 찾을 수 있는 방정식 $A\hat{x} = \hat{y}$ 으로 대체

$$A^T A\hat{x} = A^T\hat{b}$$
$$X^T X\hat{\beta} = X^T\hat{y}$$

해 $(\hat{x}, \hat{\beta})$를 찾을 수 있음

가중회귀모델

단순선형회귀식 *(Linear Regression)*

$$y_i = \beta_0 + \beta_1 x_1 + \epsilon_i$$

오차항 (residual)

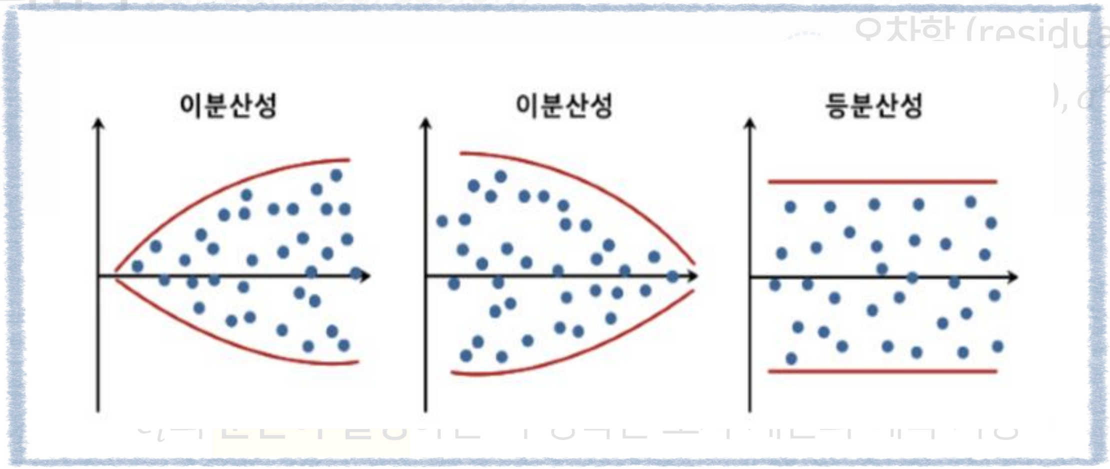
$\epsilon_i \sim N(0, \sigma^2)$

입력 변수와 무관하게 무작위적으로 분포한 등분산성 가정

→ ϵ_i의 **분산이 일정**하면 더 정확한 오차 계산과 예측 가능

하지만 등분산성 가정을 만족하지 않을 때…

가중선형회귀



작은 분산에 가중치를 두어 등분산성을 맞춰 처리

가중회귀모델

가중선형회귀

가중선형회귀 *(Weighted Linear Regression)*

양변에 각각 가중치를 곱한 후 Least-square problem을 적용해 베타 계수 추정

가중치: 해당 관측치의 분산의 역수

$$\begin{pmatrix} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & W_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & W_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

잔차의 분산이 다르기 때문에 **등분산성이 성립 X**

가중치를 곱한다면 분산이 큰 잔차는 값을 작게 만들고, 작은 잔차는 값을 크게 만듦

가중치가 조정 파라미터의 역할을 함

가중선형회귀

가중선형회귀 *(Weighted Linear Regression)*

양변에 각각 가중치를 곱한 후 Least-square problem을 적용해 베타 계수 추정

가중치

$$\begin{pmatrix} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & W_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & W_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

일반 선형회귀 계수 추정

$$X^T X\beta = X^T y$$

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

가중선형회귀 계수 추정

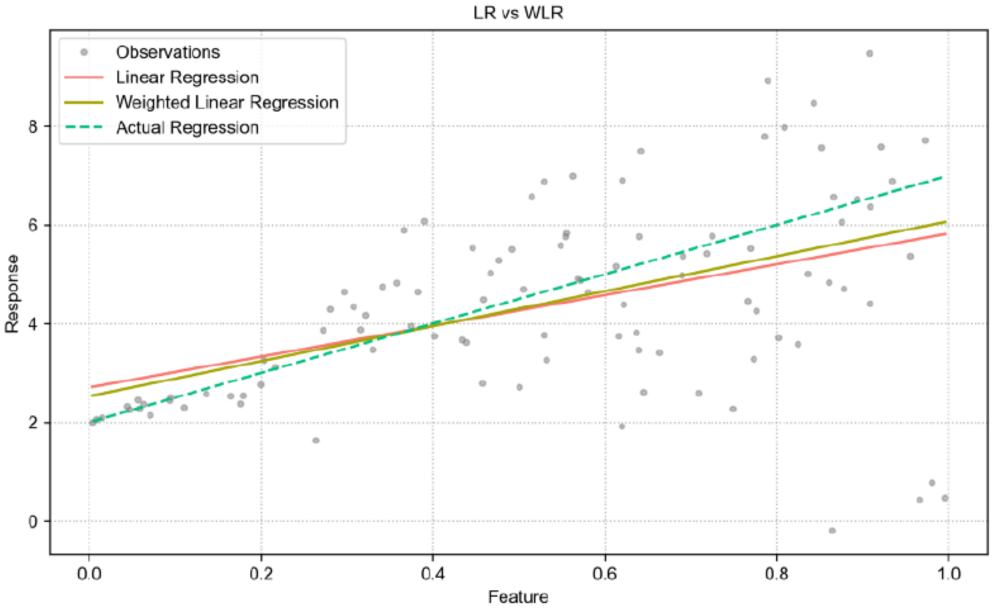
$$(WX)^T WX\beta = (WX^T)Wy$$

$$\hat{\beta} = (WX^T X)^{-1}(WX)^T y$$

구조가 유사함을 확인

가중선형회귀

가중선형회귀선이 단순선형회귀선보다 실제 회귀선에 더 근접

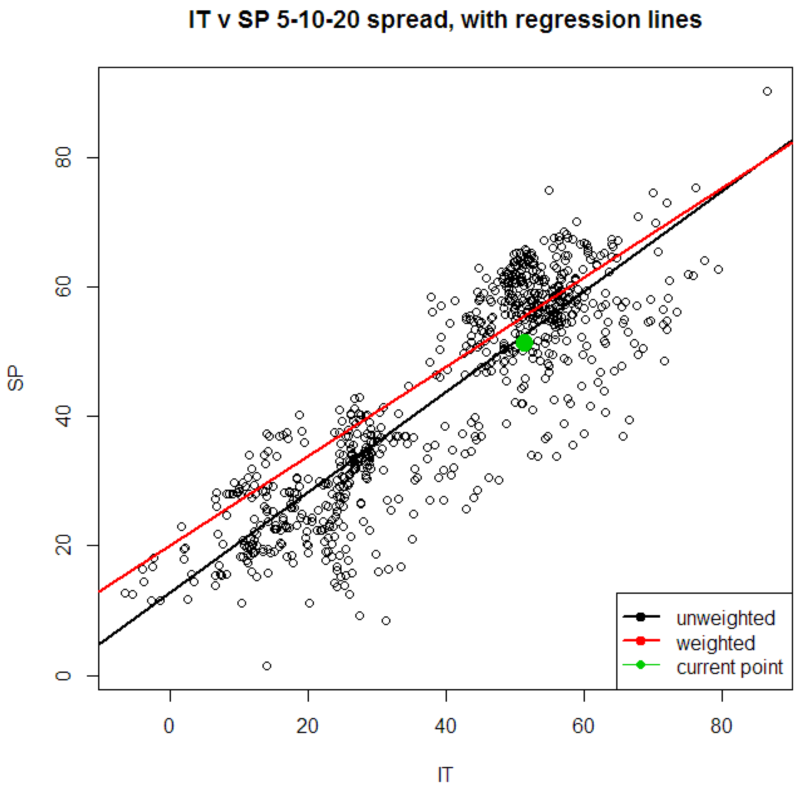


가중선형회귀가 단순선형회귀보다 **유연함**

→ 절대적인 것은 아니다

5 회귀적용

가중선형회귀



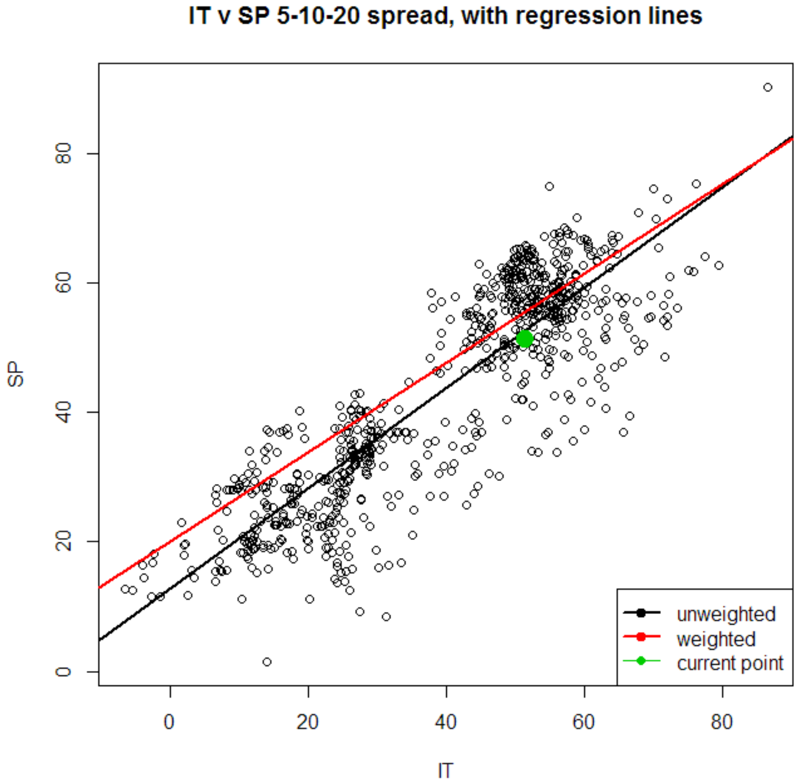
Current point(●): 실제 관측값

단순선형회귀가 예측을 더 정확하게 해냄.

데이터에 따라 성능이 다를 수 있음

5 회귀적용

가중선형회귀



Current point(●): 실제 관측값

단순선형회귀가 예측을 더 정확하게 해냄.

데이터에 따라 성능이 다를 수 있음

각 관측치의 분산을 구하기 어려운 경우가 많음

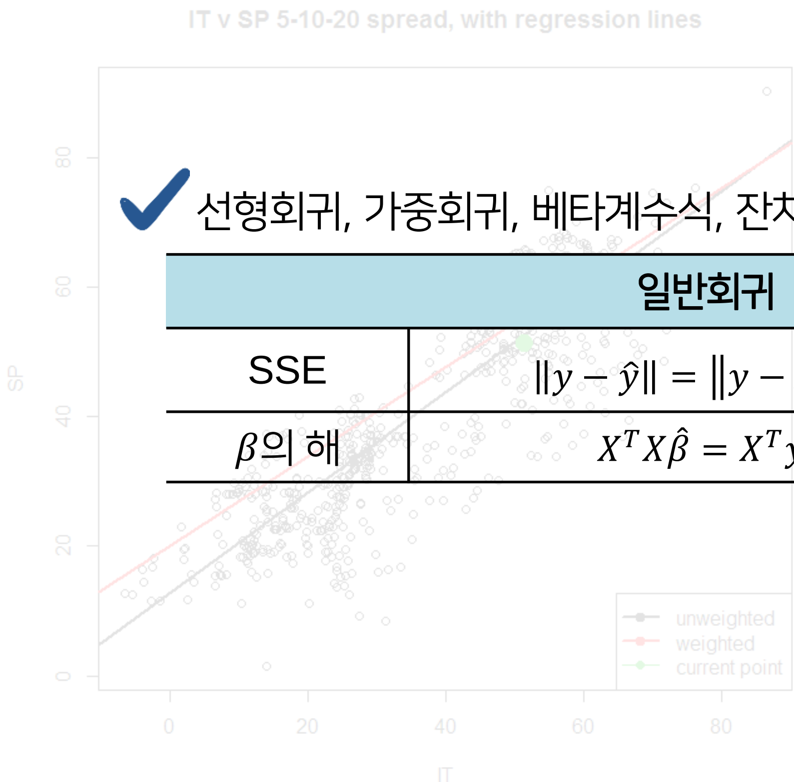
실제로 가중선형회귀를 적용하긴 쉽지 않음

변수변환 같은 해결책 사용

자세한 내용은 회귀분석팀 클린업 참고

5 회귀적용

가중선형회귀



Current point(●): 실제 관측값

다수선형회귀가 예측을 더 정확하게 해냄.

✔ 선형회귀, 가중회귀, 베타계수식, 잔차제곱합 비교

	일반회귀	가중회귀
SSE	$\lVert y-\hat{y}\rVert = \lVert y-X\hat{\beta}\rVert^2$	$\lVert Wy-W\hat{y}\rVert = \lVert Wy-WX\hat{\beta}\rVert^2$
β의 해	$X^TX\hat{\beta} = X^Ty$	$(WX)^TWX\hat{\beta} = (WX)^TWy$

각 관측치의 분산을 구하기 어려운 경우가 많음

실제로 가중선형회귀를 적용하긴 쉽지 않음

변수변환 같은 해결책 사용

다음 주 예고

1. 고유값과 고유벡터
2. 고유값 분해
3. 주성분분석
4. 특이값분해