이 장의 목표

계산이 아닌 직관에 집중:

- 역행렬, 열공간, 계수, 영공간의 **기하학적 의미**
- 선형 시스템을 **기하학적으로** 이해하기
- 계산 방법(가우스 소거법 등)은 다루지 않음

왜 이 접근인가?

- 실제로는 컴퓨터가 계산
- 우리는 **개념적 이해**가 중요
- 직관이 있으면 추후 학습이 수월

선형 시스템 (Linear System of Equations)

일반적인 형태:

```
2x + 5y + 3z = -3
4x + 0y + 8z = 0
1x + 3y + 0z = 2
```

행렬 형태:

```
[2  5  3] [x]    [-3]
[4  0  8] [y] = [0 ]
[1  3  0] [z]    [2 ]

Ax = v
```

여기서:

- A: 계수 행렬
- x: 미지수 벡터
- v: 결과 벡터

기하학적 해석

질문의 재구성:

```
Ax = v
```

의미: "어떤 벡터 x를 변환 A로 보냈을 때 v에 도달하는가?"

시각화:

- A = 선형변환
- x = 입력 (찾고자 하는 것)

- v = 출력 (주어진 것)
- 변환 A를 적용했을 때 v로 가는 벡터 x를 찾기

역행렬 (Inverse Matrix)

det(A) ≠ 0인 경우:

역변환의 개념:

- A를 "되돌리는" 변환 A^{-1}
- A^{-1}는 A의 **역변환(inverse transformation)**

해법:

```
Ax = v
A⁻¹Ax = A⁻¹v
Ix = A⁻¹v
x = A⁻¹v
```

기하학적 의미:

1. v에서 시작
2. 역변환 A^{-1} 적용
3. 원래 벡터 x로 돌아감

항등 변환 (Identity Transformation)

항등 행렬 I:

```
I = [1  0  0]  (3D의 경우)
    [0  1  0]
    [0  0  1]
```

의미:

- "아무것도 하지 않는" 변환
- 모든 벡터를 그대로 유지
- Ix = x (모든 x에 대해)

역행렬의 정의:

```
AA⁻¹ = A⁻¹A = I
```

연주 비유:

- A = 앞으로 재생
- A^{-1} = 되감기
- AA^{-1} = 재생 후 되감기 = 제자리

역행렬이 존재하는 경우

조건:

```
det(A) ≠ 0
```

의미:

- 변환 A가 공간을 압축하지 않음
- Full rank (차원 유지)
- 정보 손실 없음

2D 예:

```
A = [3   0]  det(A) = 6 ≠ 0
    [0   2]

A⁻¹ = [1/3   0 ]
      [0    1/2]
```

확인:

```
AA⁻¹ = [3·(1/3)   0    ] = [1  0] = I ✓
       [0         2·(1/2)]   [0  1]
```

역행렬이 없는 경우

조건:

```
det(A) = 0
```

의미:

- 공간이 낮은 차원으로 압축됨
- 정보 손실
- 되돌릴 수 없음

왜 되돌릴 수 없는가?

2D → 1D 압축 예:

```
[2D 평면] --A--> [직선]
```

되돌리려면:

```
[직선] --A⁻¹--> [2D 평면] ???
```

- 직선 위의 한 점이 원래 어디였는지 알 수 없음
- 무수히 많은 2D 점들이 같은 1D 점으로 매핑됨
- 일대일 대응 불가

함수로 비유:

```
f(x) = x²

f(2) = 4
f(-2) = 4

역함수 f⁻¹(4) = ??? (2? -2?)
```

일대일이 아니면 역함수 없음!

계수 (Rank)

정의: 변환 후 공간의 **차원**

2D 예:

- Rank 2: 2D 전체 (full rank)
- Rank 1: 직선으로 압축
- Rank 0: 점으로 압축 (모든 것이 원점으로)

3D 예:

- Rank 3: 3D 전체 (full rank)
- Rank 2: 평면으로 압축
- Rank 1: 직선으로 압축
- Rank 0: 점으로 압축

Full Rank:

```
rank(A) = 열의 개수
```

- 차원 손실 없음
- det(A) ≠ 0
- 역행렬 존재

Rank Deficient:

```
rank(A) < 열의 개수
```

- 차원 축소
- det(A) = 0
- 역행렬 없음

열공간 (Column Space)

정의: 행렬의 **열벡터들의 span**

행렬 A:

```
A = [a₁  a₂  a₃]
```

여기서 a_1, a_2, a_3는 열벡터

열공간:

```
Col(A) = span(a₁, a₂, a₃)
```

기하학적 의미:

변환 관점:

- A를 적용했을 때 **도달 가능한 모든 벡터**의 집합
- Col(A) = {Ax | x는 모든 가능한 벡터}

기저벡터 관점:

- 변환된 기저벡터들이 span하는 공간
- 기저벡터들이 "착지"하는 공간

예시 (2D):

Full rank:

```
A = [3  1]
    [0  2]

열벡터: [3, 0], [1, 2]
```

- 선형독립
- 열공간 = 2D 전체 평면

Rank 1:

```
A = [2  4]
    [1  2]

열벡터: [2, 1], [4, 2]
```

- 선형종속 ([4, 2] = 2[2, 1])
- 열공간 = [2, 1] 방향의 직선

열공간의 중요성

선형 시스템 Ax = v의 해:

해가 존재하려면:

```
v ∈ Col(A)
```

즉, v가 열공간 안에 있어야 함!

이유:

- Ax는 A의 열벡터들의 선형결합
- 따라서 Ax는 항상 열공간 안에
- v가 열공간 밖이면 도달 불가

예:

```
A가 2D를 직선으로 압축

v가 그 직선 위에 있으면: 해 존재
v가 직선 밖에 있으면: 해 없음
```

영공간 (Null Space / Kernel)

정의: 원점으로 매핑되는 모든 벡터의 집합

```
Null(A) = {x | Ax = 0}
```

항상 포함:

```
A·0 = 0
```

영벡터는 항상 영공간에 포함

Full rank인 경우:

```
Null(A) = {0}
```

원점만 원점으로 (당연한 경우만)

Rank deficient인 경우: 영공간이 더 큼!

영공간의 기하학적 의미

2D → 1D 압축:

```
A = [2  4]
    [1  2]
```

공간 변환:

- 2D 평면이 [2, 1] 방향 직선으로

영공간:

- 원점으로 압축되는 벡터들
- [-2, 1] 방향의 직선 전체

- 왜? 압축되는 직선에 수직인 방향

시각화:

```
변환 전:            변환 후:
       ↑ [수직]
       |                ────→ [직선]
──────────→ [압축 방향]
```

수직 방향 벡터들 → 모두 원점으로

3D → 2D 압축:

```
3D 공간 → xy평면으로
```

영공간:

- z축 방향 직선 전체
- z축 상의 모든 벡터가 원점으로

3D → 1D 압축:

```
3D 공간 → 어떤 직선으로
```

영공간:

- 그 직선에 수직인 평면 전체
- 평면 상의 모든 벡터가 원점으로

계수와 영공간의 관계

계수-영성 정리 (Rank-Nullity Theorem):

```
rank(A) + dim(Null(A)) = n
```

여기서 n = 열의 개수

의미:

- "차원 보존"
- 압축된 만큼 영공간이 커짐

예 (2×2 행렬):

```
rank = 2: nullity = 0 (영벡터만)
rank = 1: nullity = 1 (직선)
rank = 0: nullity = 2 (전체 평면)
```

예 (3×3 행렬):

```
rank = 3: nullity = 0 (영벡터만)
rank = 2: nullity = 1 (직선)
rank = 1: nullity = 2 (평면)
rank = 0: nullity = 3 (전체 공간)
```

선형 시스템의 해

Ax = v의 해 분석:

경우 1: det(A) ≠ 0 (Full rank)

```
x = A⁻¹v  (유일해)
```

조건:

- v는 어디든 상관없음 (모든 v 도달 가능)
- 정확히 하나의 해

경우 2: det(A) = 0, v ∈ Col(A)

```
무수히 많은 해
```

구조:

```
x = x₀ + (영공간의 임의의 벡터)
```

여기서 x_0는 특수해 (particular solution)

이유:

- Ax_0 = v (한 해를 찾음)
- Ax_null = 0 (영공간 벡터)
- A(x_0 + x_null) = Ax_0 + Ax_null = v + 0 = v ✓

예:

```
2D → 1D 압축
v가 직선 위에 있으면:
- 직선에 수직인 방향으로 무한히 많은 원본 존재
- 해의 집합 = 한 점 + 영공간 = 직선
```

경우 3: det(A) = 0, v ∉ Col(A)

```
해 없음
```

이유:

- v가 도달 불가능한 곳에 있음

- Ax는 절대 v에 도달 못함

예:

```
2D → 1D 압축
v가 직선 밖에 있으면:
- 어떤 벡터도 v로 변환 안 됨
```

해의 구조 요약

det(A)	v 위치	해의 개수	해의 구조
$\neq 0$	어디든	1개	$x = A^{-1}v$
$= 0$	Col(A) 안	무한	x_0 + Null(A)
$= 0$	Col(A) 밖	0개	없음

핵심 직관 정리

1. 역행렬 = 되돌리기

- $AA^{-1} = I$
- det(A) ≠ 0일 때만 존재
- 변환을 "되감기"

2. 열공간 = 도달 가능한 곳

- 변환 후 도달 가능한 모든 벡터
- 열벡터들의 span
- rank = 열공간의 차원

3. 영공간 = 원점으로 가는 곳

- Ax = 0인 모든 x
- "소멸"하는 방향
- 차원 축소의 "희생자"

4. rank + nullity = n

- 차원 보존
- 압축된 만큼 영공간 커짐

5. 해의 존재:

- $v \in$ Col(A) $\iff$ 해 존재
- det(A) ≠ 0 $\iff$ 유일해
- det(A) = 0 & $v \in$ Col(A) $\iff$ 무한해

실전 예제

예제 1: Full rank (역행렬 존재)

```
A = [3  1]  det(A) = 6 ≠ 0
    [0  2]

Ax = [5, 4]

해: x = A⁻¹[5, 4]
```

- 유일해
- 어떤 v든 도달 가능

예제 2: Rank 1 (역행렬 없음)

```
A = [2  4]  det(A) = 0
    [1  2]

Ax = [6, 3]  (직선 위)
```

- 무한해
- 예: x = [3, 0], [1, 1], [-1, 2], ...
- 해의 집합 = 특수해 + 영공간

예제 3: 해 없음

```
A = [2  4]  det(A) = 0
    [1  2]

Ax = [1, 0]  (직선 밖)
```

- 해 없음
- [1, 0]이 열공간 밖

---

2. 퀴즈

문제 1

선형 시스템 Ax = v를 기하학적으로 해석하시오. 특히 "어떤 벡터 x를 변환 A로 보냈을 때 v에 도달하는가?"라는 관점에서 역행렬 A^{-1}의 역할을 설명하시오.

문제 2

다음 두 행렬에 대해:

```
(a) A = [2  0]
        [0  3]

(b) B = [1  2]
        [2  4]
```

각각의 행렬식, 계수(rank), 열공간(column space), 영공간(null space)을 구하고, 역행렬의 존재 여부를 판단하시오. 기하학적 의미와 함께 설명하시오.

문제 3

열공간(column space)의 정의를 두 가지 관점에서 설명하시오: (a) 열벡터들의 span으로서 (b) 변환의 출력 집합으로서 그리고 이 두 정의가 왜 동등한지 논리적으로 서술하시오.

문제 4

영공간(null space)의 기하학적 의미를 설명하시오. 특히: (a) det(A) ≠ 0일 때와 det(A) = 0일 때 영공간이 어떻게 다른가? (b) 2D → 1D 압축의 경우, 영공간이 무엇인지 설명하시오. (c) 계수-영성 정리(rank + nullity = n)의 기하학적 의미를 서술하시오.

문제 5

선형 시스템 Ax = v의 해가: (a) 유일하게 존재하는 경우 (b) 무수히 많이 존재하는 경우 (c) 존재하지 않는 경우 각각의 조건을 det(A), 열공간, 영공간의 관점에서 설명하고, (b)의 경우 해의 구조를 서술하시오.

---

3. 퀴즈 답

문제 1 답안

선형 시스템 Ax = v의 기하학적 해석:

전통적 관점 (대수적):

```
2x + 5y = -3
4x + 0y = 0
```

연립방정식의 해를 찾기

기하학적 관점 (변환):

```
Ax = v
```

재구성된 질문: "어떤 벡터 x에 변환 A를 적용하면 벡터 v가 되는가?"

시각화:

```
입력 공간              출력 공간
     x        --A-->        v
     ?                    (주어짐)
```

우리가 찾는 것: x (입력) 우리가 아는 것: v (출력), A (변환)

역행렬 A^{-1}의 역할:

det(A) ≠ 0인 경우:

1. 역변환의 존재:

- A^{-1}는 A를 "되돌리는" 변환
- A가 공간을 어떻게 변형했든, A^{-1}는 원상복구

2. 해법:

```
Ax = v

양변에 A⁻¹ 적용:
A⁻¹(Ax) = A⁻¹v
(A⁻¹A)x = A⁻¹v
Ix = A⁻¹v
x = A⁻¹v
```

3. 기하학적 과정:

```
Step 1: v에서 시작 (출력 공간)
Step 2: 역변환 A⁻¹ 적용 (되감기)
Step 3: x에 도달 (입력 공간)
```

시각화:

```
x --A--> v  (순방향: 주어진 변환)
x <--A⁻¹-- v  (역방향: 해를 찾는 과정)
```

비유:

- A = 녹음 과정 (원본 → 파일)
- A^{-1} = 재생 과정 (파일 → 원본)
- x = 원본 소리
- v = 녹음 파일

v(파일)가 주어졌을 때, A^{-1}(재생)으로 x(원본)를 복원

항등 변환의 역할:

```
AA⁻¹ = I (항등 변환)
```

- A 적용 후 A^{-1} 적용 = 아무 일도 안 일어남
- 제자리로 돌아옴
- 이것이 "역"의 정의

결론: 역행렬은 변환을 되돌려서 출력(v)에서 입력(x)을 복원하는 도구입니다. 이것이 선형 시스템을 푸는 기하학적 의미입니다.

문제 2 답안

(a) A = [2 0] [0 3]

행렬식:

```
det(A) = 2 × 3 - 0 × 0 = 6 ≠ 0
```

계수(Rank):

```
rank(A) = 2 (full rank)
```

이유:

- 열벡터 [2, 0], [0, 3]이 선형독립
- 2D 공간 전체를 span
- 차원 손실 없음

열공간(Column Space):

```
Col(A) = R² (2차원 전체 공간)
```

기하학적 의미:

- 변환 A는 각 축 방향 독립적 스케일링
- x축 2배, y축 3배
- 2D 평면의 모든 점에 도달 가능

영공간(Null Space):

```
Null(A) = {[0, 0]} (영벡터만)
```

확인:

```
[2  0] [x]   [0]
[0  3] [y] = [0]

2x = 0 → x = 0
3y = 0 → y = 0
```

원점만 원점으로

역행렬 존재:

```
det(A) ≠ 0이므로 A⁻¹ 존재

A⁻¹ = [1/2   0 ]
      [0    1/3]
```

기하학적 의미:

- A는 차원 축소 안 함
- 모든 변환을 되돌릴 수 있음

- 일대일 대응

(b) B = [1 2] [2 4]

행렬식:

```
det(B) = 1 × 4 - 2 × 2 = 4 - 4 = 0
```

계수(Rank):

```
rank(B) = 1
```

이유:

- 열벡터 [1, 2], [2, 4]가 선형종속
- [2, 4] = 2 × [1, 2]
- 2D → 1D 압축

열공간(Column Space):

```
Col(B) = span([1, 2]) = {t[1, 2] | t ∈ R}
```

기하학적 의미:

- [1, 2] 방향의 직선
- 기울기 2인 직선
- 2D 평면이 이 직선으로 압축됨

시각화:

```
      ↑
     2|  Col(B)
     1|  ↗
   ──┼──→
      1
```

영공간(Null Space):

```
Null(B) = span([-2, 1]) = {s[-2, 1] | s ∈ R}
```

유도:

```
[1  2] [x]    [0]
[2  4] [y] = [0]

x + 2y = 0
2x + 4y = 0 (같은 방정식)
```

```
x = -2y

y = t라 하면:
x = -2t
[x, y] = [-2t, t] = t[-2, 1]
```

기하학적 의미:

- [-2, 1] 방향의 직선
- 열공간에 **수직**인 방향
- 이 방향의 모든 벡터가 원점으로 압축

시각화:

```
   Null(B)
      ↖
       \
        \  Col(B)
         ＼ ↗
```

역행렬 존재:

```
det(B) = 0이므로 B⁻¹ 존재 안 함
```

이유:

- 정보 손실 (2D → 1D)
- 직선을 평면으로 되돌릴 수 없음
- 일대일 대응 불가

계수-영성 정리 확인:

```
rank(B) + dim(Null(B)) = 1 + 1 = 2 ✓
```

문제 3 답안

열공간의 두 가지 정의:

(a) 열벡터들의 span으로서:

정의 1:

```
A = [a₁  a₂  ...  aₙ]

Col(A) = span(a₁, a₂, ..., aₙ)
```

의미:

- 열벡터들의 **모든 선형결합**

- $\{c_1a_1 + c_2a_2 + ... + c_na_n \mid c_1, c_2, ..., c_n \in R\}$

예:

```
A = [1  3]
    [2  6]

열벡터: a₁ = [1, 2], a₂ = [3, 6]

Col(A) = span([1, 2], [3, 6])
       = span([1, 2])  (a₂ = 3a₁이므로)
       = [1, 2] 방향의 직선
```

(b) 변환의 출력 집합으로서:

정의 2:

```
Col(A) = {Ax | x는 모든 가능한 벡터}
```

의미:

- 변환 A를 적용했을 때 **도달 가능한 모든 벡터**
- "변환의 치역(range)"

예:

```
A = [1  3]
    [2  6]

임의의 x = [x₁, x₂]:

Ax = [1  3] [x₁]   [x₁ + 3x₂]      [1]       [3]
     [2  6] [x₂] = [2x₁ + 6x₂] = x₁[2] + x₂[6]

Col(A) = 모든 Ax의 집합
       = [1, 2] 방향의 직선
```

두 정의의 동등성 증명:

정의 1 → 정의 2:

$span(a_1, ..., a_n)$의 임의의 벡터 v:

```
v = c₁a₁ + c₂a₂ + ... + cₙaₙ

x = [c₁, c₂, ..., cₙ]ᵀ라 하면:
```

```
Ax = A[c₁]   = c₁(A의 1열) + c₂(A의 2열) + ...
      [c₂]
      [⋮ ]
      [cₙ]

    = c₁a₁ + c₂a₂ + ... + cₙaₙ = v
```

따라서 span(a_1, ..., a_n) ⊆ {Ax}

정의 2 → 정의 1:

{Ax}의 임의의 벡터 w:

```
w = Ax (어떤 x에 대해)

x = [x₁]
    [x₂]
    [⋮ ]
    [xₙ]

Ax = x₁(A의 1열) + x₂(A의 2열) + ... + xₙ(A의 n열)
   = x₁a₁ + x₂a₂ + ... + xₙaₙ
```

이것은 열벡터들의 선형결합!

따라서 {Ax} ⊆ span(a_1, ..., a_n)

결론:

```
span(a₁, ..., aₙ) = {Ax}
```

기하학적 직관:

행렬-벡터 곱셈의 의미:

```
Ax = x₁a₁ + x₂a₂ + ... + xₙaₙ
```

- x의 각 성분 = 해당 열벡터의 스케일
- Ax = 열벡터들의 가중합
- 따라서 Ax는 **항상** 열벡터들의 span 안에

역으로:

- 열벡터들의 임의의 선형결합
- = 적절한 x를 선택하여 Ax로 표현 가능

이것이 두 정의가 같은 이유입니다!

문제 4 답안

영공간(Null Space)의 기하학적 의미:

정의:

```
Null(A) = {x | Ax = 0}
```

변환 A를 적용했을 때 **원점으로 가는 모든 벡터**

(a) det(A) ≠ 0 vs det(A) = 0:

det(A) ≠ 0인 경우 (Full Rank):

```
Null(A) = {0} (영벡터만)
```

이유:

- 변환 A가 공간을 압축하지 않음
- 일대일 대응
- 원점만 원점으로 (당연한 경우만)

기하학적:

- 모든 비영 벡터가 비영 벡터로
- "소멸"하는 벡터 없음

det(A) = 0인 경우 (Rank Deficient):

```
Null(A)가 부분공간 (차원 ≥ 1)
```

이유:

- 변환 A가 공간을 압축
- 정보 손실
- 압축되는 방향의 벡터들이 모두 원점으로

기하학적:

- 여러 벡터가 같은 곳(원점)으로
- "소멸"하는 방향이 존재

(b) 2D → 1D 압축의 영공간:

예:

```
A = [2  4]  (det = 0)
    [1  2]
```

```
열공간: [2, 1] 방향의 직선
```

변환 과정:

```
2D 평면 --A--> [2, 1] 방향 직선
```

영공간 유도:

```
[2  4] [x]   [0]
[1  2] [y] = [0]

2x + 4y = 0
x + 2y = 0

x = -2y

y = t라 하면:
[x, y] = [-2t, t] = t[-2, 1]

Null(A) = span([-2, 1])
```

기하학적 의미:

1. 수직성:

```
열공간: [2, 1] 방향
영공간: [-2, 1] 방향

내적: [2, 1] · [-2, 1] = -4 + 1 = -3 ≠ 0
```

(완전히 수직은 아니지만 직교하는 관계)

실제로는:

- 열공간: 압축되는 목표 직선
- 영공간: 압축에 "희생"되는 방향

2. 시각화:

```
변환 전:                변환 후:
      ↑ 영공간
      |                   ────→ 열공간
─────────→ 열공간       (원점)
```

영공간 상의 모든 벡터 → 원점으로

3. 물리적 비유:

- 3D → 2D 투영 (그림자)
- 빛의 방향 = 영공간

- 그림자 평면 = 열공간

(c) 계수-영성 정리의 기하학적 의미:

정리:

```
rank(A) + dim(Null(A)) = n
```

여기서 n = 입력 공간의 차원 (열의 개수)

기하학적 해석:

"차원 보존의 법칙"

총 n차원의 입력 공간:

- rank 차원: 출력 공간 (열공간)으로 "살아남음"
- nullity 차원: 원점으로 "소멸"

예 (3×3 행렬):

경우 1: rank = 3, nullity = 0

```
3D --A--> 3D (full rank)
```

- 모든 차원 보존
- 아무것도 소멸 안 함

경우 2: rank = 2, nullity = 1

```
3D --A--> 2D 평면
```

- 2차원은 평면으로
- 1차원은 소멸 (직선 → 원점)

경우 3: rank = 1, nullity = 2

```
3D --A--> 1D 직선
```

- 1차원은 직선으로
- 2차원은 소멸 (평면 → 원점)

직관:

- 압축된 만큼 영공간이 커짐
- 총 차원은 항상 n
- "에너지 보존 법칙" 같은 것

물리적 비유:

- n명이 탄 배
- rank명은 목적지 도착 (열공간)
- nullity명은 바다에 빠짐 (영공간)
- rank + nullity = n (총원)

결론: 계수-영성 정리는 변환이 차원을 재분배하는 방식을 설명하며, 압축(정보 손실)이 영공간(소멸)으로 나타남을 보여줍니다.

문제 5 답안

선형 시스템 Ax = v의 해 분석:

(a) 유일해가 존재하는 경우:

조건:

```
det(A) ≠ 0
```

의미:

1. **Full rank:**
   - rank(A) = n (열의 개수)
   - 차원 손실 없음
2. **역행렬 존재:**
   - A^{-1}이 존재
3. **열공간:**
   - Col(A) = 전체 출력 공간
   - 모든 v에 도달 가능
4. **영공간:**
   - Null(A) = {0}
   - 원점만 원점으로

해:

```
x = A⁻¹v  (유일해)
```

기하학적:

- v가 어디에 있든 정확히 하나의 x만 v로 매핑
- 일대일 대응
- 정보 손실 없음

예:

```
A = [2  1]  det = 3 ≠ 0
    [1  2]

Ax = [5, 4]
```

```
x = A⁻¹[5, 4] (유일)
```

(b) 무수히 많은 해가 존재하는 경우:

조건:

```
det(A) = 0  AND  v ∈ Col(A)
```

의미:

1. **Rank deficient:**
   - rank(A) < n
   - 차원 축소
2. **v가 도달 가능:**
   - v가 열공간 안에
   - 적어도 하나의 해 존재
3. **영공간이 비자명:**
   - dim(Null(A)) ≥ 1
   - 원점으로 가는 벡터들이 많음

해의 구조:

```
x = x₀ + x_null
```

여기서:

- x_0: **특수해 (particular solution)**
  - Ax_0 = v를 만족하는 아무 해
- x_null: **영공간의 임의의 벡터**
  - Ax_null = 0

증명:

```
A(x₀ + x_null) = Ax₀ + Ax_null
               = v + 0
               = v ✓
```

기하학적 의미:

2D → 1D 예:

```
A = [1  2]  det = 0
    [2  4]

Ax = [3, 6]  (열공간 위)
```

분석:

```
열공간: [1, 2] 방향 직선
영공간: [-2, 1] 방향 직선

[3, 6] = 3[1, 2] ∈ Col(A) ✓
```

해의 집합:

```
특수해: x₀ = [3, 0]  (Ax₀ = [3, 6])

일반해: x = [3, 0] + t[-2, 1]
          = [3 - 2t, t]  (t ∈ R)
```

시각화:

```
입력 공간:           출력 공간:
     ↑
x₀   ●  ───→ (해들)       ● v
     |                 ───→ (직선)
  ───┼──→
```

해들이 직선을 이룸 (영공간 방향)

직관:

- 압축된 방향으로는 "자유도"
- 그 방향으로 얼마든지 이동 가능
- 결과는 같음 (모두 v로)

(c) 해가 존재하지 않는 경우:

조건:

```
det(A) = 0   AND   v ∉ Col(A)
```

의미:

1. **Rank deficient:**
    - 차원 축소
2. **v가 도달 불가:**
    - v가 열공간 밖에
    - 어떤 x도 v에 도달 못함

해:

```
없음 (no solution)
```

기하학적:

- 출력이 낮은 차원에만 존재
- v가 그 부분공간 밖에
- Ax는 절대 v에 도달 안 함

예:

```
A = [1  2]  det = 0
    [2  4]

Ax = [1, 0]  (열공간 밖)
```

분석:

```
열공간: [1, 2] 방향 직선

[1, 0]은 이 직선 위에 없음
(기울기가 다름: 0 ≠ 2)

따라서 해 없음
```

시각화:

```
입력 공간:              출력 공간:
                            ● v (도달 불가)
    ● x
    |                     ───→ Col(A)
  ──┼──→
```

v가 열공간 밖에

해의 존재 요약:

조건	열공간	영공간	해
det ≠ 0	전체	{0}	유일
det = 0, v ∈ Col	부분공간	부분공간	무한
det = 0, v ∉ Col	부분공간	부분공간	없음

핵심:

- **det ≠ 0:** 항상 유일해
- **det = 0:** v의 위치에 따라 무한해 또는 해 없음
- **무한해의 구조:** 특수해 + 영공간