선형대수학 클린업 1주차

[인사말]

P-SAT 27기, 28기 여러분 환영합니다.
저희는 3팀으로 선형대수학 팀이며, 선형대수학(Linear Algebra)을 클린업 때 공부하게 됩니다.
다들 행렬대수학을 이미 수강하셨으므로 일부 내용은 복습 차원에서 알고 있던 내용을 다루겠지만,
다소 새로운 내용도 포함되어 있을 것으로 생각됩니다.

저희 팀은 선형대수 개념을 수식적, 공간적으로 이해하는 것은 물론, 나아가서 통계학 및 머신러닝에서 다루게 되는 수식의 배경지식을 정복하는 것이 이번 학기 클린업의 목표입니다.
또한, 매주 응용편도 함께 다루어 선형대수학이 통계에서 어떻게 적용되는지를 알아보는 시간을 가지겠습니다.

한 학기 동안 잘 부탁드리겠습니다 ~!!

[Week1 - Contents]

5. Linear Algebra in Machine Learning(응용편)

1) 아핀 공간(Affine Spaces)와 아핀 변환(Affine Transfomration)

2) 파이썬 특강

3) Vectors and Matrix in Python

1. 선형대수학의 시작

1) 선형대수학 소개

선형대수학은 대수학의 한 분야로, 선형공간 및 그 1차 변환에 관한 이론을 연구하는 수학의 한 부문이다.

선형대수학은 행렬이나 벡터를 활용한 이론 등으로 선형방정식의 해를 구하는 것을 핵심으로 다룬다. 즉, '행렬'과 '벡터'를 바탕으로 문제를 공간적으로 이해 및 처리한다.

통계학에서도 선형대수는 문제의 이해와 처리, 그 시작점에 위치한다.
데이터분석에서 수십만 차원의 고차원 데이터를 다루는 것은 매우 흔한 일이다. 이 처리를 보다 효율적으로 할 수 있게 만들어주는 것이 선형대수학이다. 우리는 선형대수학의 기본적인 개념부터 시작해, 통계 및 머신러닝에서 사용하는 기법에서 어떻게 선형대수학이 효율적인 연산을 도와주는지 알아보는 시간을 가질 것이다.

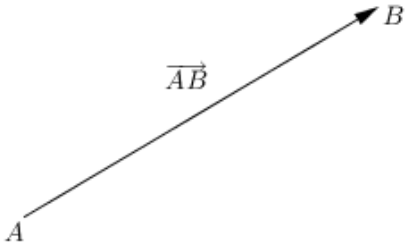
2) 벡터와 행렬

아마도, 여러분은 벡터와 행렬에 대해서 이미 익숙하실 겁니다.

따라서, 본 교안에서는 벡터와 행렬의 개념에 대해서 간단하게만 소개하고 연산 등에 대해서는 따로 소개하지 않을 예정이니 그 점 참고해주시기 바랍니다.

■ 벡터(Vector)

벡터의 가장 중요한 특징은 바로 "크기"와 "방향"을 가진 단위라는 것이다("크기"만 가진 것을 스칼라라고 한다).



공간에서 벡터를 표현할 때 화살표로 표기하게 되는데, 시작점에서 끝점으로 화살표를 표시한다.
일반적으로 "방향"은 시작지점에서 끝 부분을 가리키는 방향,
"크기"는 시작 지점과 끝 지점까지의 거리로 나타내게 된다.

■ 행렬(Matrix)

행렬이란 수나 식을 직사각형 모양의 행과 열로 배열한 것을 행렬이라고 한다.
배열 안의 수 각각을 행렬의 원소(element) 또는 성분(entry)이라고

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ \vdots & \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn} \end{bmatrix}$$

하며, 행(m)과 열(n)의 개수를 이용해 크기(m x n)를 표현한다.

특히, m과 n의 크기가 같아 n x n의 행렬이 되는 경우, 이때의 행렬을 **정사각행렬(square matrix)**이라 한다.

2. 선형 방정식과 선형 변환

1) 선형 방정식(Linear Equation)

선형 방정식은 쉽게 말해서 1차 방정식을 일컫는다.
즉, 최고 차수의 항의 차수가 1을 넘지 않는 다항 방정식이다.
이를 식으로 표현하면 $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$ 형태로 나타나며, 이 때 n은 양의 정수이다.

2) 선형 시스템(Linear System)

■ 선형 시스템의 개념

앞에서 언급했듯이, 고차원 선형방정식의 해를 찾는 것은 선형대수학에서 핵심적인 부분을 차지한다.
위에서 소개한 선형 방정식들의 집합을 선형시스템(Linear System; 혹은 연립선형방정식)이라고 부른다.

[연립선형방정식]

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \qquad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

왼쪽의 식이 선형시스템을 나열해 놓은 것이고, 오른쪽 식은 이를 행렬로 표현한 것이다.

이 때, a_{ij}을 모아놓은 m x n 행렬을 A, x_i를 모아놓은 n x 1 열 벡터를 **x**, b_i를 모아놓은 m x 1 열 벡터를 b라고 하며,

이를 A**x** = **b**로 표현할 수 있다.

■ 기본 행 연산(Elementary Row Operations; 혹은 Elementary Transformation)

위의 예시처럼 미지수의 개수가 적은 연립방정식의 경우에는 우리가 일반적으로 행하는 소거법이나 대입법 등의 방법으로 방정식을 풀어도 쉽게 해를 구할 수 있다.

하지만, n개의 미지수를 가진 연립방정식은 그렇지 못하다.
우리는 미지수의 개수가 많더라도 쉽게 해를 구할 수 있는 방법을 찾고 싶다.

이 때 등장하게 되는 것이 Elementary Row operations이다.

Elementary Row Operations는 행 간의 연산을 통하여 연립방정식의 해를 쉽게 구할 수 있게 해준다. 다음과 같은 3가지의 기본 연산으로 이루어진다.

(1) 두 행을 교환 $R_i \leftrightarrow R_j$

(2) 한 행에 0이 아닌 실수를 곱한다. $R_i \leftarrow \lambda R_i$

(3) 한 행에 0이 아닌 실수배를 하여 다른 행에 더한다. $R_i \leftarrow R_i + \lambda R_j$

■ Row Echelon Form(REF)와 Reduced Row Echelon Form(RREF)

다음과 같은 모양의 행렬을 생각해보자.

(1) 모든 수가 0으로 이루어진 행이 아니라면, 그 행에서 첫 번째로 나타나게 되는 0이 아닌 수는 1이다. 이때 이 1을 'leading 1(선행성분, pivot)'이라고 한다.

(2) 모든 수가 0으로 이루어진 행은 가장 아래 행으로 위치시킨다.

(3) i 행과 (i+1)행에 모두 leading1이 존재하면, (i+1)행의 선행성분은 i 행의 선행성분보다 오른쪽에 오도록 행을 위치시킨다.

(4) 모든 열에는 많아도 한개의 leading 1만이 존재하며, leading 1을 제외한 수는 모두 0이다.

이 때, (1) ~ (3)을 만족하는 행렬을 **Row Echelon Form(REF)**, (1) ~ (4)을 모두 만족하는 행렬을 **Reduced Row Echelon Form(RREF)**이라고 한다.

■ Gauss-Jordan 소거법(Gauss-Jordan Elimination)

Gauss-Jordan 소거법이란 위에서 소개한 기본 연산을 이용해 연립방정식의 해를 구하는 방법이다.
이는 다음과 같은 방식으로 이루어진다.

(1) 성분이 모두 0인 행이 존재하면 그 행은 행렬의 맨 아래에 위치시킨다.

(2) 첫 번째 열의 수가 0이 아닌 행을 제일 위에 오도록 위치시킨다.

(3) 첫 번째 행이 1이 되도록 나눠준다(기본연산을 통해)

(4) 다른 행의 첫 번째 열의 수가 0이 되도록 첫 번째 행을 실수배하여 다른 행에 더해준다.

(5) 첫 번째 행과 첫 번째 열을 제외한 나머지 Submatrix를 가지고 위의 과정을 반복한다.

(6) 한 개의 열에는 많아야 한 개의 leading 1만 존재하도록 가장 아래에 있는 Non-zero행부터 다른 행에 적절한

실수배를 하여 더하는 방식으로 기본연산을 진행한다.

이 때, (1) ~ (5)의 과정을 **가우스 소거법(Gaussian elimination)**이라고 하며 이를 통해 구한 행렬은 **Row Echelon Form(REF)**이 된다.

(1) ~ (6)의 모든 과정을 거쳤을 때, 이를 **가우스-조던 소거법(Gauss-Jordan Elimination)**이라고 하며, 이를 통해 구한 행렬은 **Reduced Row Echelon Form(RREF)**이 된다.

예를 들어보자.

다음 연립방정식의 해를 구해보자.

x + y + 3z = 0

2x + 3y − z = 9

3x − 2y + 3z = -4

물론, 이 경우 미지수가 고작 3개이기 때문에 직접 해를 구해도 되지만, 가우스-조던 소거법으로 구해보겠다.

먼저, Augmented Matrix Form으로 바꾸면 다음과 같다.

$$\begin{bmatrix} 1 & 1 & 3 & 0 \\ 2 & 3 & -1 & 9 \\ 3 & -2 & 3 & -4 \end{bmatrix}$$

이제, Gauss Elimination 과정을 통해 각 열에 leading 1이 나타나도록 변형한다.
결과적으로 REF 형태의 행렬을 얻게 된다.

$$\begin{bmatrix} 1 & 1 & 3 & 0 \\ 0 & 1 & -7 & 9 \\ 3 & -2 & 3 & -4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 3 & 0 \\ 0 & 1 & -7 & 9 \\ 0 & -5 & -6 & -4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 3 & 0 \\ 0 & 1 & -7 & 9 \\ 0 & 0 & -41 & 41 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1 & 1 & 3 & 0 \\ 0 & 1 & -7 & 9 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

이제, Gauss-Jordan Elimination을 통해 RREF 형태의 행렬을 얻자.

$$\begin{bmatrix}1 & 1 & 3 & 0\\ 0 & 1 & -7 & 9\\ 0 & 0 & 1 & -1\end{bmatrix} \rightarrow \begin{bmatrix}1 & 0 & 10 & -9\\ 0 & 1 & -7 & 9\\ 0 & 0 & 1 & -1\end{bmatrix} \rightarrow \begin{bmatrix}1 & 0 & 0 & 1\\ 0 & 1 & 0 & 2\\ 0 & 0 & 1 & -1\end{bmatrix}$$

즉, 결과적으로 우린 $\begin{bmatrix}1 & 0 & 0 & 1\\ 0 & 1 & 0 & 2\\ 0 & 0 & 1 & -1\end{bmatrix}$ 이라는 행렬을 얻었고,

이는 x = 1, y = 2, z = -1을 의미한다.

이를 일반화해서, 만약 Augmented Matrix의 RREF 모양의 행렬이 $\begin{bmatrix}1 & 0 & 0 & a\\ 0 & 1 & 0 & b\\ 0 & 0 & 1 & c\end{bmatrix}$ 라면,

연립방정식의 해는 x = a, y = b, z = c가 되는 것이다.

그렇다면, $\begin{bmatrix}1 & 0 & 0 & a\\ 0 & 1 & 0 & b\\ 0 & 0 & 0 & c\end{bmatrix}$ 과 같이 일부 열에 leading 1(즉, 피벗)이 없을 때는 어떻게 될까?

이에 대해서 살펴보겠다.

■ **선형 시스템에서 해집합**

선형시스템에서 해를 구할 때 해는 총 3가지의 경우로 나뉜다.

(i) 유일한 해가 존재하는 경우

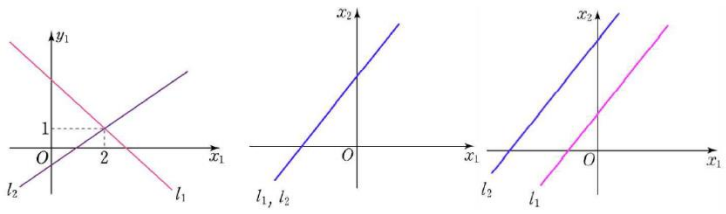
(ii) 해가 무수히 많은 경우

(iii) 해가 존재하지 않는 경우

이 때, 해가 존재하는 경우인 (i), (ii)를 Consistent하다, 해가 존재하지 않는 경우인 (iii)을 Inconsistent하다고 한다.

3가지 경우를 간단히 미지수가 2개인 연립방정식에 대해 생각하면, 기하학적으로는 아래와 같이 나타낼 수 있다.

(i) (ii) (iii)



즉,

(i) 두 직선이 한 점에서 만나는 경우

(ii) 두 직선이 일치하는 경우

(iii) 두 직선이 평행한 경우

로 나타낼 수 있다.

우리가 궁금한 것은 이러한 해의 종류를 판별하는 방법이다.

해의 판별법은 다음과 같다.

Ax = b 라는 연립방정식이 있을 때, 이를 Augmented Matrix 형태로 바꾼다.

그 후, 가우스-조던 소거법을 통해서 RREF 형태의 행렬로 변형한다.

그 결과를 [H | k]라 하자.

그렇다면,

(i) 유일한 해가 존재하는 경우: H의 모든 열에 피벗이 존재

(ii) 해가 무수히 많은 경우: H의 일부 열에 피벗이 없고 그 행에 해당하는 k의 수가 0일 때

(iii) 해가 존재하지 않는 경우: H의 일부 열에 피벗이 없고 그 행에 해당하는 k의 수가 0이 아닐 때

즉, 예를 들면,

$\begin{bmatrix}1&0&0&1\\0&1&0&1\\0&0&0&0\end{bmatrix}$ → 해가 무수히 많다

$\begin{bmatrix}1&0&0&1\\0&1&0&1\\0&0&0&1\end{bmatrix}$ → 해가 존재하지 않는다

3) 선형 변환(Linear Transformation 혹은 Linear Mapping)

■ **선형 변환의 정의**

식 A**x** = b를 살펴보면, 열 벡터 **x**가 행렬 A를 곱해서 열 벡터 b로 변형되었다고 할 수 있다.
이를 대수적인 측면에서 살펴보면, **x**가 b로 mapping 되었다고 볼 수도 있을 것이다.

이 때, mapping T: $\mathbb{R}^n \to \mathbb{R}^m$ 이 임의의 벡터 u, v 및 임의의 스칼라 a, b에 대해

T(au + bv) = aT(u) + bT(v) 를 만족하는 경우 mapping T를 선형 변환(Linear Transformation or Linear Mapping)이라 한다.

■ **기저 변환(Basis Change)**

기저는 바로 다음 챕터에서 설명할 예정이지만, 선형 변환의 중요한 성질 중 하나가 기저를 변환하는 것이므로 간단하게 언급하고 넘어가겠다.

앞서 말했듯이, 선형 변환은 mapping의 일종이므로 특정 벡터 x가 벡터 b로 변환된다.

이 때, x에서 b로의 변환을 x를 이루고 있는 기저 벡터공간 B에서 다른 기저 벡터공간 $\tilde{B}$ 로의 변환으로 생각해볼 수 있다는 것이다.

즉, 쉽게 설명하자면 좌표축을 변환시킨다고 생각하는 것이다.

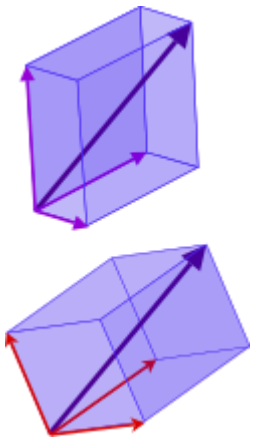
예를 들어보자.

$\begin{bmatrix}2 & 5\\1 & 3\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}7\\4\end{bmatrix}$ 를 보면, 벡터 $\begin{bmatrix}1\\1\end{bmatrix}$이 행렬 A에 의해 벡터 $\begin{bmatrix}7\\4\end{bmatrix}$로 변환되었음을 알 수 있다.

이 때, $\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}$이며 $\begin{bmatrix}2 & 5\\1 & 3\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}2 & 5\\1 & 3\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}2 & 5\\1 & 3\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = 1\begin{bmatrix}2\\1\end{bmatrix} + 1\begin{bmatrix}5\\3\end{bmatrix} = \begin{bmatrix}7\\4\end{bmatrix}$이므로

선형변환 T =$\begin{bmatrix}2 & 5\\1 & 3\end{bmatrix}$에 의해 기저 B = { $\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$ }에서 기저 $\tilde{B} = \{\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}5\\3\end{bmatrix}\}$로 변환되었음을 알 수 있다.

<그림을 통한 예시>



선형변환에 의해 기저를 변환시킬 수 있고, 그림과 같이 위와 아래의 좌표축이 달라지게 된다.

3. 선형 독립(Linear Independence)

1) 선형 결합(Linear Combination)

선형 결합이란, 스칼라 a_1, a_2, …, a_n과 벡터 x_1, x_2, …, x_n에 대해 덧셈과 곱셈의 결과물로 $a_1x_1 + a_2x_2 + \cdots + a_nx_n$ 의 꼴을 일컫는다.

추가적으로, $a_1x_1 + a_2x_2 + \cdots + a_nx_n = \begin{bmatrix} a_1 a_2 \cdots a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ 로 나타낼 수 있다.

2) 선형 독립(Linearly Independent)

$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$ 에서 b = 0일때를 생각해보자.

$a_1x_1 + a_2x_2 + \cdots + a_nx_n$ = 0일 때, 해당 방정식의 해가 trivial solution만 존재할 때(즉, 모든 a_i가 0인 경우만이 해가 될 때), 벡터 $x_1, x_2, \cdots, x_n$ 들을 **선형 독립(Linear Independent)**이라고 한다.

벡터 $x_1, x_2, \cdots, x_n$ 가 독립이 아닐 때(즉, 하나 이상의 a_i가 0이 아니어도 방정식이 성립할 때)를 **선형 종속(Linear Dependent)**이라고 한다.

정의는 위와 같은데, 이는 추상적으로 느껴질 수 있다. 따라서, 이를 좀 더 자세히 살펴보겠다.

벡터 $x_1, x_2, \cdots, x_n$ 가 **선형 종속**이라 하자. 그렇다면, 정의에 의해서 어떤 a_i는 0이 아니다.
a_1을 0이 아닌 상수라고 하자.

그렇다면, $a_1x_1 = -(a_2x_2 + a_3x_3 + \cdots + a_nx_n)$ 이 되므로 $x_1 = -\frac{1}{a_1}(a_2x_2 + a_3x_3 + \cdots + a_nx_n)$ 을 얻게 된다.

이 말은 즉, 벡터 $x_1, x_2, \cdots, x_n$ 가 선형 종속일 때는
어떤 벡터(계수가 0이 아닌 벡터)는 그 벡터를 제외한 다른 벡터들의 선형 결합으로 표현된다는 것이다.

이는 굉장히 유의미한 특징이다.

바로 뒤에 나올 벡터 공간과 Span 챕터의 표현을 미리 빌리면,
이는 **어떤 벡터가 다른 벡터들의 Span으로 형성되는 벡터 공간 속에 속한다**는 것이다.

이를 다시 해석하면,
벡터 $x_1, x_2, \cdots, x_n$ 가 선형 독립일때는 어떤 벡터도 다른 벡터들의 선형결합으로 표현할 수 없다는 것이다.

4. 공간에서의 선형대수학(Linear Algebra in Spaces)

1) 벡터 공간(Vector Spaces)

■ **벡터 공간(Vector Spaces)**

벡터 공간(Vector Spaces)이란 다음과 같이 정의된 집합 V를 의미한다.

- 집합 V는 벡터들의 집합이다(즉, V의 모든 원소는 벡터이다)
- 집합 V는 두 개의 연산(Operation)을 가진다
  $+ : V \times V \rightarrow V$
  $\cdot : \mathbb{R} \times V \rightarrow V$

참고로, 이 조건을 만족하면 **"집합 V는 연산 +, ·에 닫혀 있다(closed)"**라고 말한다.
또한, 위에 언급된 · 연산은 내적(inner product)이 아니라 Scalar Multiplication임에 주의해야 한다.

- 집합 V는 + 연산에 대해 교환법칙이 성립한다
- 집합 V는 + 연산에 대해 결합법칙이 성립한다
- 집합 V는 + 연산에 대해 항등원이 존재하고, 그 항등원은 0(영벡터)이다
- 집합 V 임의의 원소 u에 대해 + 연산에 대한 역원이 존재하고, 그 역원은 -u이다
- 집합 V는 · 연산에 대해 분배법칙이 성립한다
- 집합 V는 · 연산에 대해 결합법칙이 성립한다
- 집합 V는 · 연산에 대해 항등원이 존재하고, 그 항등원은 1이다

벡터 공간에 대한 엄밀한 정의는 위와 같지만, 이는 매우 추상적으로 느껴질 수 있다

위의 개념을 살펴보면 느끼겠지만, 어떻게 보면 벡터들의 연산에 관한 당연한 개념들을 나열한 것에 불과하다. 물론 엄밀하게 정의를 다루면 좋겠지만, 우리는 벡터 공간이란 단순히 (i) 벡터들의 연산이 정의된, (ii) 벡터들을 모아 놓은 집합으로 이해해도 충분하다.

■ **벡터 부분공간(Vector Subspaces)**

벡터 부분공간(Vector Subspaces)란 직관적으로 벡터 공간의 부분집합 중 벡터 공간의 조건을 만족하는 집합을 말한다.
즉, 집합 S가 (i) 벡터 공간 V의 부분집합이며, (ii) S가 벡터 공간의 조건을 만족한다면 집합 S를 V의 벡터 부분공간이라고 한다.

2) Span과 기저(Basis)

■ Span

Span이란 **어떤 벡터 공간 V가 있을 때 V의 원소들의 선형결합들에 의해 만들어지는 공간을** 의미한다.
즉, $V = \{v_1, v_2, \cdots, v_n\}$일 때 임의의 실수 $a_1, a_2, \cdots, a_n$에 대해 $a_1v_1 + a_2v_2 + \cdots + a_nv_n$이 만들어내는 공간을 $\text{span}\{v_1, v_2, \cdots, v_n\}$이라고 한다.

정리하자면, **Span{V}는 벡터 공간 V의 원소들의 모든 가능한 선형결합들에 의해 형성되는 벡터공간**이다.

참고로, u, v를 각각 벡터라고 했을 때,

벡터 한 개만을 Span시킨 Span{u}는 원점을 지나고u의 스칼라곱으로 표현될 수 있는 직선으로,

2개의 벡터를 Span시킨 Span{u,v}는 원점을 지나고 u와 v로
형성되는 평면이 된다.

■ 기저(Basis)

기저(Basis)란 어떤 벡터 공간 V에 대해 그 V를 span하는 **가장 작은(minimal) 집합**을 말한다.
이러한 집합이 되려면 다음과 같은 조건을 만족해야 한다.

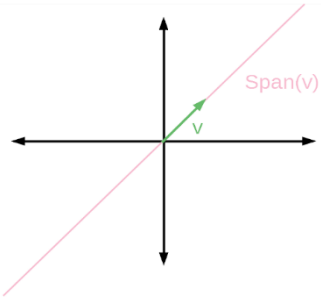
(i) 기저에 속한 벡터들이 선형 독립

(ii) 기저에 속한 벡터들이 벡터 공간 V를 형성(span)

참고로, **기저는 유일하지 않다**.

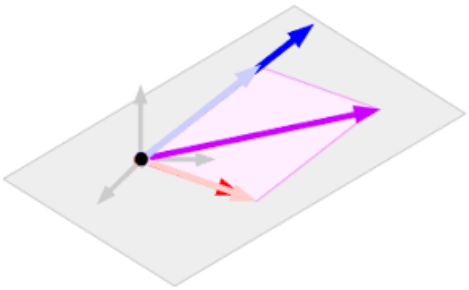
즉, 어떤 벡터 공간 V에 대해 V를 span하는 기저는 여러 개가 될 수 있다.

예를 들어, xyz 공간에서 xy평면을 하나의 벡터 공간 V라고 할 때,
벡터 (1,0,0), (0,1,0)도 V의 기저가 될 수 있지만, 벡터 (2,1,0), (1,1,0)도 V의 기저가 될 수 있다.



■ 표준 기저(Standard Basis)

표준 기저(Standard Basis)는 기저의 일종인데, (1,0,⋯,0), (0,1,⋯,0), ⋯, (0,0,⋯,1) 등으로 표현되는 벡터들로 이루어진 기저를 말한다. 쉽게 알 수 있듯이 위의 벡터들은 **서로 직교하며 크기가 1인 벡터들**이다.



3) 열공간(Column Spaces)과 영공간(Null Spaces)

■ 열공간(Column Spaces)

열공간(column spaces)이란 어떤 행렬 A에서 A의 열벡터들의 선형결합으로 만들어지는 벡터 공간이다.
즉, 행렬의 열벡터들을 span해서 만들어지는 벡터 공간으로 이해할 수 있다. 이를 col(A)라고 표기한다.

예를 들어, $\begin{bmatrix}1 & 2\\ 4 & 5\end{bmatrix}$라는 행렬의 열공간은 열벡터 $\begin{bmatrix}1\\4\end{bmatrix}$, $\begin{bmatrix}2\\5\end{bmatrix}$로 만들어지는 벡터 공간이다.

■ 선형 시스템에서 열공간의 의미

Ax = b 라는 선형 시스템을 생각해보자. 이 때, A를 m x n 행렬, x를 n x 1 벡터, b를 m x 1 벡터라고 하자

그렇다면 A = $[v_1 \quad v_2 \quad \cdots \quad v_n]$, x = $\begin{bmatrix}x_1\\x_2\\\cdots\\x_n\end{bmatrix}$, b = $\begin{bmatrix}b_1\\b_2\\\cdots\\b_m\end{bmatrix}$ 이라 쓸 수 있고, A에 있는 $v_1, v_2, \cdots, v_n$ 는 열벡터이다.

$Ax = [v_1 \quad v_2 \quad \cdots \quad v_n]\begin{bmatrix}x_1\\x_2\\\cdots\\x_n\end{bmatrix} = x_1v_1 + x_2v_2 + \cdots + x_nv_n = b$ 가 됨을 알 수 있다.

식 $x_1v_1 + x_2v_2 + \cdots + x_nv_n = b$와 x_i는 스칼라, v_i는 벡터임을 주목하자.

이는 곧 **벡터 b가 $v_1, v_2, \cdots, v_n$의 선형 결합으로 표현됨**을 알 수 있다.

위에서 언급했듯이 선형시스템에는 해가 존재할 수도, 존재하지 않을 수도 있다.

방금 유도한 식에서 알 수 있는 것은

(i) 해가 존재한다면 : 벡터 b가 $v_1, v_2, \cdots, v_n$의 선형 결합으로 표현됨

(ii) 해가 존재하지 않는다면 : 벡터 b가 $v_1, v_2, \cdots, v_n$의 선형 결합으로 표현되지 않음

이는 다시 말하면 다음과 같다.

(i) 해가 존재한다면 : 벡터 b가 span{$v_1, v_2, \cdots, v_n$}에 의해 만들어지는 벡터 공간에 속함

(ii) 해가 존재하지 않는다면 : 벡터 b가 span{$v_1, v_2, \cdots, v_n$}에 의해 만들어지는 벡터 공간에 속하지 않음

참고로, Ax = b가 모든 벡터 b에 대해서 해를 가진다는 것은 무엇을 의미할까?

b는 span{$v_1, v_2, \cdots, v_n$}에 속해야 하므로 span{$v_1, v_2, \cdots, v_n$}가 모든 벡터를 표현할 수 있어야 한다.

이는 곧 span{$v_1, v_2, \cdots, v_n$}가 $\mathbb{R}^n$을 형성함을 뜻한다.

■ 열공간의 기저

일반적으로, 어떤 벡터공간의 기저를 구하는 일은 귀찮고, 때로는 만만치 않게 보일 수 있다.

열공간은 쉽게 기저를 구할 수 있는 방법이 알려져 있다.
→ **열 공간에서의 기저는 피벗 컬럼에 해당하는 열의 집합이다.**

피벗 컬럼을 구하기 위해서는 Elementary Row Operation을 진행해야 한다.

즉, 열공간의 기저를 다음과 같은 순서로 구할 수 있다.

(i) 주어진 행렬을 REF Form으로 만든다

(ii) Pivot Column을 찾는다

(iii) 그 Column들을 모아 놓은 집합이 열공간의 기저가 된다

여기서 주의할 점은 REF 형태에서의 열벡터가 아니라, 기존 행렬에서의 열벡터가 열공간의 기저가 되는 것이다.

예를 들어보자.

행렬 $\begin{bmatrix}1 & 3 & 2\\2 & 7 & 4\\1 & 5 & 2\end{bmatrix}$ 의 열공간의 기저를 구해보겠다.

(i) $\begin{bmatrix}1 & 3 & 2\\2 & 7 & 4\\1 & 5 & 2\end{bmatrix} \sim \begin{bmatrix}1 & 3 & 2\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix}$: REF

(ii) Pivot Column = $\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}3\\1\\0\end{bmatrix}$: 첫 번째와 두 번째 열

(iii) 열공간의 기저 ={ $\begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}3\\7\\5\end{bmatrix}$ } 이 된다({ $\begin{bmatrix}\mathbf{1}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{3}\\\mathbf{1}\\\mathbf{0}\end{bmatrix}$ }**이 기저가 아닌 것에 주의**)

그렇다면, 일반적인 벡터 공간의 기저를 구하는 방법으로 다음과 같은 방법을 생각해볼 수 있다.

벡터 공간을 이루는 모든 벡터들을 열벡터로 하는 행렬을 정의한 후, 그 행렬의 열공간의 기저를 구하는 것이다.

예를 들어, 벡터 공간 { $\begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}3\\7\\5\end{bmatrix}, \begin{bmatrix}2\\4\\2\end{bmatrix}$ }의 기저를 구하는 방법으로, 행렬 $\begin{bmatrix}1 & 3 & 2\\2 & 7 & 4\\1 & 5 & 2\end{bmatrix}$을 정의한 후, 이 행렬의 열공간의 기저를 구하면 되는 것이다.

■ 영공간(Null Space)

영공간(Null Space)이란 Ax = 0의 해공간(Solution Spaces)을 의미한다.

즉, Ax = 0을 만족하는 벡터 x들에 의해 만들어지는 벡터 공간이다.
당연한 얘기지만, x가 0일 때(영벡터일 때) Ax는 0이 되므로 영벡터는 항상 영공간에 포함되어 있다.

Null Space 대신에 kernel이라는 표현도 많이 사용한다.

■ 영공간의 기저

영공간의 기저는 자유변수를 계수로 하는 선형결합으로 표현될 때 존재하는 벡터들의 집합이다.

문장이 굉장히 이해하기 힘들게 느껴질 수 있는데, 예를 들면 쉽게 이해할 수 있다.

행렬 $\begin{bmatrix} 1 & 1 & 0 & 2 \\ -2 & -2 & 1 & -5 \\ 1 & 1 & -1 & 3 \\ 4 & 4 & -1 & 9 \end{bmatrix}$ 의 영공간의 기저를 구해보자

행렬 A = $\begin{bmatrix} 1 & 1 & 0 & 2 \\ -2 & -2 & 1 & -5 \\ 1 & 1 & -1 & 3 \\ 4 & 4 & -1 & 9 \end{bmatrix}$, x = $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ 이라 하면 영공간은 $\begin{bmatrix} 1 & 1 & 0 & 2 \\ -2 & -2 & 1 & -5 \\ 1 & 1 & -1 & 3 \\ 4 & 4 & -1 & 9 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ 을 만족하는 벡터 x들의 집합이다. Augmented Matrix로 표현한 후 이를 RREF로 정리하면 다음과 같다.

$\begin{bmatrix} 1 & 1 & 0 & 2 & 0 \\ -2 & -2 & 1 & -5 & 0 \\ 1 & 1 & -1 & 3 & 0 \\ 4 & 4 & -1 & 9 & 0 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ 즉, $x_1+x_2+2x_4 = 0$, $x_3-x_4 = 0$ 을 얻는다.

이를 정리하면 $x_1 = -x_2-2x_4$, $x_3 = x_4$ 이 되는데, 이를 벡터로 표현하면 다음과 같다.

x = $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -x_2 - 2x_4 \\ x_2 \\ x_4 \\ x_4 \end{bmatrix}$ = s $\begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + t \begin{bmatrix} -2 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ where $s = x_2$, $t = x_4$ and $s,t \in \mathbb{R}$

따라서, 영공간의 기저는 $\left\{ \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right\}$ 이 됨을 알 수 있다.

4) 차원(Dimensions)과 행렬의 계수(Rank)

■ 차원(Dimensions)

차원(Dimensions)이란 어떤 벡터 공간에 대해서 기저를 이루는 벡터의 개수이다.

■ 열공간의 차원

열공간의 차원은 Pivot의 개수와 같다.

■ 영공간의 차원

영공간의 차원은 행렬에서 열의 개수에서 pivot의 개수를 뺀 것과 같다.

또한, 이는 자유변수의 개수와 같다.

참고로, 열공간의 차원과 영공간의 차원을 더하면 열의 개수가 된다.

■ 행렬의 계수(Rank)

행렬의 계수(Rank)는 열공간의 차원을 말한다.

예를 들어, 위에서 구한 행렬 $\begin{bmatrix}1 & 3 & 2\\2 & 7 & 4\\1 & 5 & 2\end{bmatrix}$의 rank는 2가 된다(∵ $\begin{bmatrix}1 & 3 & 2\\2 & 7 & 4\\1 & 5 & 2\end{bmatrix}$의 기저는 ={ $\begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}3\\7\\5\end{bmatrix}$ })

참고로, Rank의 정의는 열공간의 차원을 의미하지만, 실제로는 행공간(Row Space)의 차원과도 일치한다.

5. Linear Algebra in Machine Learning(응용편)

1) 아핀 공간(Affine Spaces)과 아핀 변환(Affine Transformation)

■ 아핀 공간(Affine Spaces)

아핀 공간(Affine Spaces)의 정의는 다음과 같다.

어떤 벡터 공간 V가 있다고 했을 때, 부분공간 $U \subseteq V$와 벡터 $x_0 \in V$에 대해
$L = x_0 + U$로 나타나는 L을 V의 아핀(부분)공간이라고 말한다.

■ 아핀 변환(Affine Transformation)

우리는 위에서 선형 변환을 살펴보았는데, 이를 간단히 행렬 A에 대해 Ax로 표현했음을 기억할 것이다.

아핀 변환은 Ax + b로 표현할 수 있는데, 이는 선형 변환 후 b만큼의 평행이동을 한 모양과 같다.

아핀 변환 자체는 선형 변환이 아닌데, 그 이유는 아핀변환 T(x) = Ax + k 라고 했을 때

T(au + bv) = (A(au) + k) + (A(bv) + k) = aA(u) + k + bA(v) + k
= a(T(u) − k) + b(T(v) − k) + 2k

= aT(u) +bT(v) + (2 – a – b)k 가 되므로 선형 변환의 조건을 만족시키지 못하기 때문이다.

■ Affine과 Deep Learning

딥러닝의 모든 것을 지금 다룰 수는 없으므로, 딥러닝의 구조를 아주 간단하게만 소개하고 넘어가겠다.

딥러닝은 머신러닝의 일종으로, 인공신경망이라고 불리는 개념을 사용한다.

이 때, 딥러닝의 학습은 input layer와 hidden layer를 거치면서 이루어진다.

각 단계를 거칠 때, 가중치 행렬이라는 것에 의해 값들이 조정되는데, 이 과정이 **Affine 변환**과 유사하다.

즉, A를 가중치 행렬, x를 다음 Layer에 들어갈 input, b를 bias로 정의하며 이 때 output은 **Ax + b**가 되는 것이다.