

# 머신 러닝을 위한 수학

미적분, 선형대수

# 목 차

- 1 머신 러닝을 위한 미적분
- 2 머신 러닝을 위한 선형대수

# 머신 러닝을 위한 미적분

미분의 기초

## 미분

- 아주 짧은 순간의 함수에 대한 기울기를 구하는 것
- 도함수 :  $f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(a+\Delta x) - f(a)}{\Delta x}$
- 극한값이 실수로 존재하면  $f$ 는  $x=a$ 에서 미분 가능하며  $f'(a)$ 로 표시

# 머신 러닝을 위한 미적분

미분의 기초

## 미분

- $\frac{d}{dx} x^n = nx^{n-1}$
- $\frac{d}{dx} \ln x = \frac{1}{x}$
- $\frac{d}{dx} e^x = e^x$
- $\frac{d}{dx} (af + bg) = a \frac{df}{dx} + b \frac{dg}{dx}$
- $\frac{d}{dx} (fg) = f \frac{dg}{dx} + \frac{df}{dx} g$
- $\frac{d}{dx} \frac{1}{f} = -\frac{1}{f^2} \frac{df}{dx}$

# 머신 러닝을 위한 미적분

미분의 기초

## 머신 러닝에서의 활용

- 손실 함수의 값을 최소로 만들기 위해 다양한 기법을 사용
- 손실 함수를 미분하면 어떤 특정 지점에서 어느 정도의 기울기가 나오는 지 알 수 있다.

### 상미분(Ordinary derivative)

- 변수가 하나만 있는 함수의 미분
- $y = x^r$  일 때  $\frac{dy}{dx} = rx^{r-1}$
- $\frac{d}{dx}\{f(x) + g(x)\} = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$
- $\frac{d}{dx}\{kf(x)\} = k \frac{df(x)}{dx}$

# 머신 러닝을 위한 미적분

상미분과 편미분

## 전미분(Total differentiation)

- $f(x, y)$ 의 미분
- $\Delta z = f(x + \Delta x, y + \Delta y) - f(x, y)$

# 머신 러닝을 위한 미적분

상미분과 편미분

## 편미분(Partial derivative)

- $y$ 를 상수라 생각하고 고정시킨 다음,  $x$ 에 대해 편미분 :  $\frac{\partial f(x,y)}{\partial x}, f_x(x, y)$
- $x$ 를 상수라 생각하고 고정시킨 다음,  $y$ 에 대해 편미분 :  $\frac{\partial f(x,y)}{\partial y}, f_y(x, y)$

$$f(x, y) = 3x^2 + 5xy + 3y^3$$

$$\frac{\partial f(x,y)}{\partial x} = 6x + 5y$$

$$\frac{\partial f(x,y)}{\partial y} = 5x + 9y^2$$



### 벡터(Vector)란?

- 크기와 방향을 갖는 직선
- 벡터공간(Vector space)을 이루는 원소
- 열벡터(종벡터) : 벡터의 원소를 세로로 표현한 경우
- 행벡터(횡벡터) : 벡터의 원소를 가로로 표현한 경우

$$\mathbf{a} = [a_1, a_2, \dots, a_n]$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

덧셈과 뺄셈, 그리고 스칼라배

## 덧셈과 뺄셈

- 서로 대응하는 성분끼리 덧셈 또는 뺄셈

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 + 4 \\ 2 + 5 \\ 3 + 6 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 9 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 - 4 \\ 2 - 5 \\ 3 - 6 \end{bmatrix} = \begin{bmatrix} -3 \\ -3 \\ -3 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

덧셈과 뺄셈, 그리고 스칼라배

## 덧셈과 뺄셈

- 서로 다른 차원의 벡터끼리는 덧셈과 뺄셈을 할 수 없다.

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ 5 \\ 6 \\ 7 \end{bmatrix} = \text{계산불가}$$

# 머신 러닝을 위한 선형대수

덧셈과 뺄셈, 그리고 스칼라배

## 스칼라배(Scalar multiple)

- 벡터의 모든 성분에 같은 수를 곱하는 것

$$2 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \times 1 \\ 2 \times 2 \\ 2 \times 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

덧셈과 뺄셈, 그리고 스칼라배

## 머신 러닝에서의 활용

- 언어를 처리하기 위해 단어들을 벡터처럼 취급
- 일반적인 벡터 연산처럼 단어들도 덧셈과 뺄셈이 가능
- '왕' - '남성' + '여성' = '여왕'
- '동경' - '일본' + '한국' = '서울'

### 내적(Inner product)이란?

- 벡터에서 서로 대응하는 성분끼리 곱한 다음 그것들을 모두 더한 값
- 벡터와 벡터의 내적은 스칼라가 된다.
- 서로 다른 차원의 벡터끼리는 계산을 할 수 없다.

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \langle \mathbf{a}, \mathbf{b} \rangle = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{i=1}^n a_i b_i$$

### 내적(Inner product)이란?

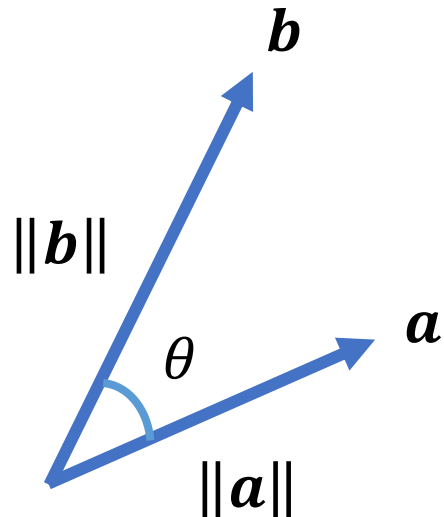
- 벡터에서 서로 대응하는 성분끼리 곱한 다음 그것들을 모두 더한 값
- 벡터와 벡터의 내적은 스칼라가 된다.
- 서로 다른 차원의 벡터끼리는 계산을 할 수 없다.

$$\mathbf{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \quad \langle \mathbf{a}, \mathbf{b} \rangle = 1 \times 4 + 2 \times 5 + 3 \times 6 = 32$$

### 기하학적 정의

- 벡터  $a$ 와  $b$ 가 이루는 각이  $\theta$ 일 때  $\langle a, b \rangle$ 는 다음과 같이 정의

$$\langle a, b \rangle = \|a\| \|b\| \cos \theta$$





### 기하학적 정의

- 벡터  $\mathbf{a}$ 와  $\mathbf{b}$ 가 이루는 각이  $\theta$ 일 때  $\langle \mathbf{a}, \mathbf{b} \rangle$ 는 다음과 같이 정의

$$\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

$$\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos 45^\circ = \sqrt{5} \times \sqrt{10} \times \frac{\sqrt{2}}{2} = 5$$

# 머신 러닝을 위한 선형대수

## 벡터의 노름

### 노름(Norm)이란?

- 벡터를 이루는 방향과 이동 거리 중 이동 거리를 의미
- $\|x\| = 0 \leftrightarrow x=0$
- $\|\alpha x\| = |\alpha| \|x\|$
- $\|x + y\| \leq \|x\| + \|y\|$

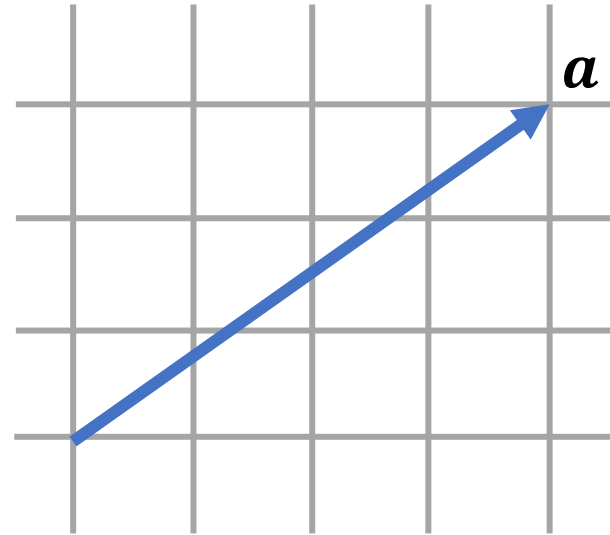
# 머신 러닝을 위한 선형대수

벡터의 노름

## L1 노름

- 벡터 성분의 절대값을 모두 더한 값
- 맨해튼 거리

$$\|a\|_1 = |a_1| + |a_2| + \dots + |a_n| = \sum_{i=1}^n |a_i|$$



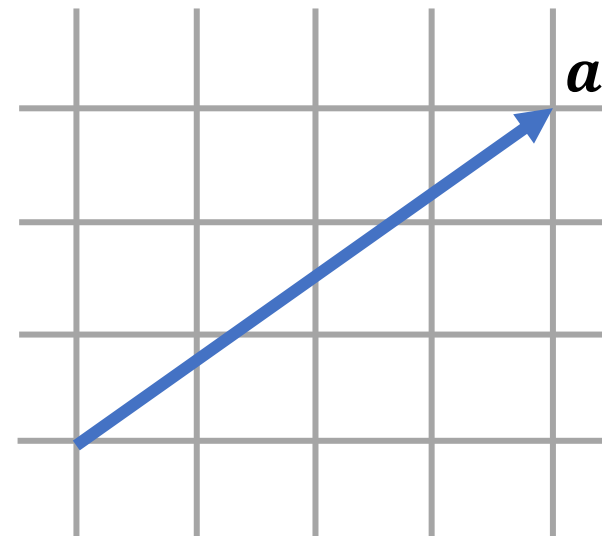
# 머신 러닝을 위한 선형대수

벡터의 노름

## L2 노름

- 시작점에서 목적지까지 직선으로 움직인 거리
- 유클리드 거리

$$\begin{aligned}\|a\|_2 &= \sqrt{\sum_{i=1}^n a_i^2} = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} \\ &= \sqrt{\langle a, a \rangle}\end{aligned}$$



# 머신 러닝을 위한 선형대수

벡터의 노름

## 머신 러닝에서의 활용

- 선형회귀 모델의 정규화 항
- 데이터셋을 학습 데이터와 테스트 데이터로 분류하는 과정에서 과적합
- 정규화 항을 덧붙여 계수의 절대값이나 제공한 값이 너무 커지지 않도록 만들어줘야 한다.

# 머신 러닝을 위한 선형대수

코사인 유사도

## 코사인 유사도

$$\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

$$\cos \theta = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

$$\langle \mathbf{a}, \mathbf{b} \rangle = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{i=1}^n a_i b_i$$

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^n a_i^2} = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$$

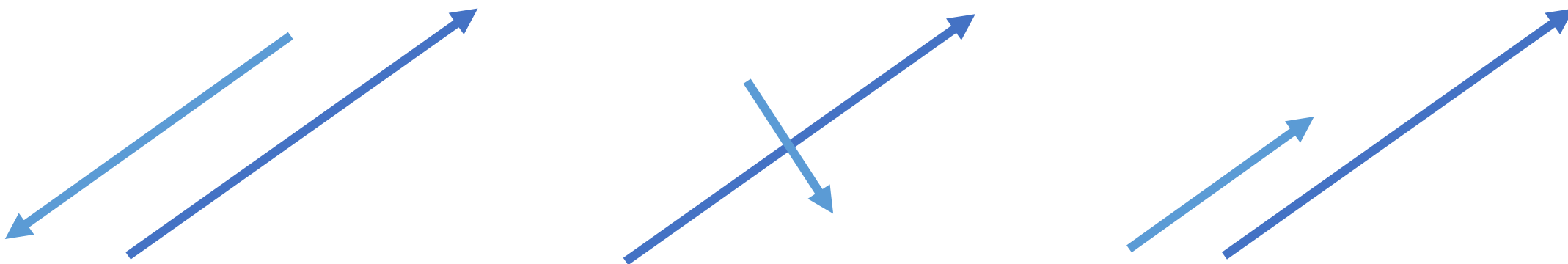
$$\cos \theta = \cos(\mathbf{a}, \mathbf{b}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}}$$

# 머신 러닝을 위한 선형대수

코사인 유사도

## 코사인 유사도

- -1 이상 1 이하의 값을 가진다.
- 코사인 유사도가 높다는 말은 벡터가 더 비슷하다는 의미



# 머신 러닝을 위한 선형대수

코사인 유사도

## 머신 러닝에서의 활용

- 텍스트를 분석할 때 벡터로 만들어진 단어나 문장들의 관계성을 파악할때 활용
- 코사인 유사도가 높을수록 단어나 문장들은 더 가까운 관계라는 것을 의미



# 머신 러닝을 위한 선형대수

행렬의 덧셈과 뺄셈

## 행렬이란?

- 같은 차원의 벡터를 모아 나열한 것
- 행렬의 크기 :  $m \times n$ ,  $m$ 행  $n$ 열

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

# 머신 러닝을 위한 선형대수

행렬의 덧셈과 뺄셈

## 행렬의 덧셈과 뺄셈

- 서로 대응하는 성분끼리 덧셈 또는 뺄셈

$$\begin{bmatrix} 0 & 7 & 2 & 2 \\ 1 & 2 & 6 & 1 \\ 5 & 3 & 3 & 4 \end{bmatrix} + \begin{bmatrix} 2 & 6 & 7 & -1 \\ 1 & 8 & 3 & 5 \\ 0 & -1 & 6 & 11 \end{bmatrix} = \begin{bmatrix} 0+2 & 7+6 & 2+7 & 2-1 \\ 1+1 & 2+8 & 6+3 & 1+5 \\ 5+0 & 3-1 & 3+6 & 4+11 \end{bmatrix} = \begin{bmatrix} 2 & 13 & 9 & 1 \\ 2 & 10 & 9 & 6 \\ 5 & 2 & 9 & 15 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 7 & 2 & 2 \\ 1 & 2 & 6 & 1 \\ 5 & 3 & 3 & 4 \end{bmatrix} - \begin{bmatrix} 2 & 6 & 7 & -1 \\ 1 & 8 & 3 & 5 \\ 0 & -1 & 6 & 11 \end{bmatrix} = \begin{bmatrix} 0-2 & 7-6 & 2-7 & 2+1 \\ 1-1 & 2-8 & 6-3 & 1-5 \\ 5-0 & 3+1 & 3-6 & 4-11 \end{bmatrix} = \begin{bmatrix} -2 & 1 & -5 & 3 \\ 0 & -6 & 3 & -4 \\ 5 & 4 & -3 & -7 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 곱셈

$$A = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$A\mathbf{b} = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \times 3 + 2 \times 2 \\ 1 \times 3 + 1 \times 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$A = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix}$$

$$A\mathbf{b} = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} -1 \times 3 + 2 \times 2 & -1 \times 1 + 2 \times 3 \\ 1 \times 3 + 1 \times 2 & 1 \times 1 + 1 \times 3 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 5 & 4 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 곱셈

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\begin{aligned} A\mathbf{b} &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \mathbf{b} = \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{b} \rangle \\ \langle \mathbf{a}_2, \mathbf{b} \rangle \\ \vdots \\ \langle \mathbf{a}_n, \mathbf{b} \rangle \end{bmatrix} \\ &= \begin{bmatrix} a_{11}b_1 + a_{12}b_2 + \cdots + a_{1n}b_n \\ a_{21}b_1 + a_{22}b_2 + \cdots + a_{2n}b_n \\ \vdots \\ a_{m1}b_1 + a_{m2}b_2 + \cdots + a_{mn}b_n \end{bmatrix} \end{aligned}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 곱셈

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

$$\begin{aligned} AB &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} (\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_l) \\ &= \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{b}_1 \rangle & \langle \mathbf{a}_1, \mathbf{b}_2 \rangle & \cdots & \langle \mathbf{a}_1, \mathbf{b}_l \rangle \\ \langle \mathbf{a}_2, \mathbf{b}_1 \rangle & \langle \mathbf{a}_2, \mathbf{b}_2 \rangle & \cdots & \langle \mathbf{a}_2, \mathbf{b}_l \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{a}_m, \mathbf{b}_1 \rangle & \langle \mathbf{a}_m, \mathbf{b}_2 \rangle & \cdots & \langle \mathbf{a}_m, \mathbf{b}_l \rangle \end{bmatrix} \end{aligned}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 곱셈

- 첫번째 행렬의 행과 두번째 행렬의 열에 대하여 원소의 위치가 대응하는 값끼리 곱한 다음 더해준다.
- 행렬 간의 곱에서는 앞 행렬의 행과 뒤 행렬의 열의 수가 같아야 한다.

$$A\mathbf{b} = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} -1 \times 3 + 2 \times 2 & -1 \times 1 + 2 \times 3 \\ 1 \times 3 + 1 \times 2 & 1 \times 1 + 1 \times 3 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 5 & 4 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 곱셈

- $A(BC) = (AB)C$  : 결합 법칙
- $A(B+C) = AB+AC, (A+B)C = AC+BC$  : 분배 법칙
- $AB \neq BA$  : 교환 법칙은 성립하지 않는다.

$$\begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 5 & 4 \end{bmatrix} \neq \begin{bmatrix} -2 & 7 \\ 1 & 7 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 스칼라배

- 행렬의 모든 성분에 같은 수를 곱하는 것

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



# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 영행렬(Zero matrix) : 행렬의 성분 전체가 0인 행렬

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 정방행렬(Square matrix) : 행과 열의 크기가 같은 행렬

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 대각행렬(Diagonal matrix) : 정방행렬에서 대각선에 해당되는 대각 성분만 0이 아니고, 나머지 비대각 성분의 원소 값이 0인 행렬

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 단위행렬(Identity matrix) : 대각행렬 중 대각 성분이 1로만 구성된 행렬
- 어떤 행렬이나 벡터에 단위행렬을 곱하면 결과가 달라지지 않는다  
: 항등사상

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

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 상삼각행렬(Upper triangular matrix) : 대각선을 기준으로 위쪽에만 원소 값이 있는 행렬

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 하삼각행렬(Lower triangular matrix) : 대각선을 기준으로 아래쪽에만 원소 값이 있는 행렬

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ 3 & 5 & 6 \end{bmatrix}$$

## 행렬의 종류

- 대칭행렬(Symmetric matrix) : 대각 원소에 대하여 대칭인 원소가 같은 행렬
- $A = A^T$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

# 머신 러닝을 위한 선형대수

행렬의 곱셈

## 행렬의 종류

- 직교행렬(Orthogonal matrix) : 열벡터들이 서로 직교하고 길이가 1인  
행렬
- $PP^T = P^T P = I$



# 머신 러닝을 위한 선형대수

역행렬

## 행렬식(Determinant)

- $|A| = \det A = ad-bc$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- $|A| = \det A = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$

### 역행렬(Inverse matrix)

- $AA^{-1} = A^{-1}A = I$ 를 만족하는 행렬  $A$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

### 역행렬(Inverse matrix)

- 정방행렬이어야한다.
- 행렬식이 0인 경우 역행렬이 존재하지 않는다. (nonsingular matrix)
- $2 \times 2$  행렬보다 큰 정방행렬의 역행렬은 가우스 소거법이나 여인자 전개와 같은 방법으로 계산

### 역행렬(Inverse matrix)의 특징

- 항등행렬 :  $AA^{-1} = A^{-1}A = I$
- 행렬 곱셈 전체에 대한 역행렬이 존재한다면, 곱한 행렬도 각각 역행렬이 존재한다. :  $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$
- A의 역행렬이 존재한다면, A의 역행렬도 역행렬이 존재한다.  
:  $A^{-1}=B, B^{-1} = A$
- A의 전치행렬(Transpose matrix)의 역행렬은 A의 역행렬의 전치행렬과 같다. :  $(A^T)^{-1} = (A^{-1})^T$

# 머신 러닝을 위한 선형대수

고윳값과 고유벡터

## 고윳값(Eigenvalue)과 고유벡터(Eigenvector)

- 다음의 식을 만족하는 열벡터  $\mathbf{x}(\mathbf{x} \neq \mathbf{0})$ 가 존재할 때,  $\lambda$ 를 행렬  $A$ 의 고윳값이라 하고,  $\mathbf{x}$ 를 고유벡터라고 한다.

$$A\mathbf{x} = \lambda\mathbf{x}$$

$$(A - \lambda I)\mathbf{x} = \mathbf{0}$$

$$(A - \lambda I)^{-1}(A - \lambda I)\mathbf{x} = (A - \lambda I)^{-1} \mathbf{0}$$

$$\mathbf{x} = (A - \lambda I)^{-1} \mathbf{0}$$

$$\det(A - \lambda I) = 0 \rightarrow \text{고유방정식(Eigen Equation)}$$

# 머신 러닝을 위한 선형대수

고윳값과 고유벡터

## 고윳값과 고유벡터 계산

$$A = \begin{bmatrix} 2 & 4 \\ -1 & -3 \end{bmatrix}$$

$$\det\left(\begin{pmatrix} 2 & 4 \\ -1 & -3 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) = 0$$

$$\det\begin{pmatrix} 2-\lambda & 4 \\ -1 & -3-\lambda \end{pmatrix} = 0$$

$$(2-\lambda)(-3-\lambda)-4(-1) = 0$$

$$\lambda^2 + \lambda - 2 = 0$$

$$(\lambda+2)(\lambda-1) = 0$$

# 머신 러닝을 위한 선형대수

고윳값과 고유벡터

## 고윳값과 고유벡터 계산

$\lambda = -2$  일때,

$$(A - (-2)I)\mathbf{x} = \begin{pmatrix} 2 - (-2) & 4 \\ -1 & -3 - (-2) \end{pmatrix} \mathbf{x} = \begin{pmatrix} 4 & 4 \\ -1 & -1 \end{pmatrix} \mathbf{x} = \mathbf{0}$$

$\mathbf{x} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ 이라 가정

$$\begin{pmatrix} 4 & 4 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\alpha + \beta = 0$$

$$\mathbf{x} = t \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

# 머신 러닝을 위한 선형대수

고윳값과 고유벡터

## 고윳값과 고유벡터 계산

$\lambda = 1$  일때,

$$(A-I)\mathbf{x} = \begin{pmatrix} 2-1 & 4 \\ -1 & -3-1 \end{pmatrix}\mathbf{x} = \begin{pmatrix} 1 & 4 \\ -1 & -4 \end{pmatrix}\mathbf{x} = \mathbf{0}$$

$\mathbf{x} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ 이라 가정

$$\begin{pmatrix} 1 & 4 \\ -1 & -4 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\alpha + 4\beta = 0$$

$$\mathbf{x} = t \begin{pmatrix} 1 \\ -\frac{1}{4} \end{pmatrix} = t \frac{1}{4} \begin{pmatrix} 4 \\ -1 \end{pmatrix}$$



# 머신 러닝을 위한 선형대수

고윳값과 고유벡터

## 머신 러닝에서의 활용

- 데이터가 많이 흩어져 있는 분포 상황에서는 주어진 문제를 해결하기 위해 고윳값과 고유벡터의 도움을 받는다.
- 고윳값은 어떤 데이터가 가진 특징을 얼마나 잘 설명할 수 있는지 가늠할 때 사용된다.
- 기여율은 각 고유벡터에 대응하는 고윳값을 전체 고윳값들의 총합으로 나눈 것으로, 우리가 가진 데이터를 얼마나 잘 설명할 수 있는지 평가하는 척도로 사용된다.

감사합니다 :)