2장. 기계학습과 수학

강원대학교 빅데이터메디컬융합학과
: 헬스케어 분야에서의 딥러닝

본 강의자료는 한빛아카데미에서 제공하는 강의자료를 바탕으로 작성되었음

PREVIEW

기계 학습에서 수학의 역할

- 수학은 **목적함수를 정의**하고, **목적함수가 최저가 되는 점을 찾아주는 최적화 이론 제공**
- 최적화 이론에 규제, 모멘텀, 학습률, 멈춤조건과 같은 제어를 추가하여 알고리즘 구축
- 사람은 알고리즘을 설계하고 데이터를 수집함

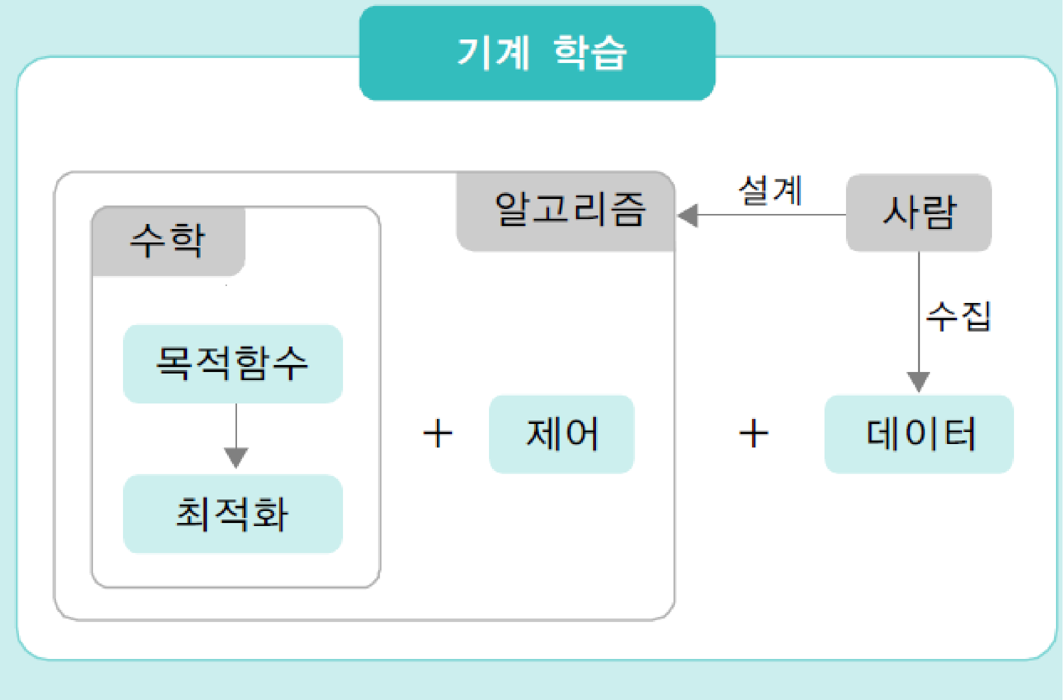


그림 2-1 기계 학습의 개념도

각 절에서 다루는 내용

- 2.1절: 선형대수를 다룬다.
- 2.2절: 확률과 통계를 다룬다.
- 2.3절: 최적화 이론을 다룬다.

- 선형대수: 이 분야의 개념을 이용하면 학습 모델의 매개변수집합, 데이터, 선형연산의 결합 등을 행렬 또는 텐서로 간결하게 표현할 수 있다. 데이터를 분석하여 유용한 정보를 알아내거나 특징 공간을 변환하는 등의 과업을 수행하는 데 핵심 역할을 한다.
- 확률과 통계: 데이터에 포함된 불확실성을 표현하고 처리하는 데 활용한다. 베이즈 이론과 최대 우도 기법을 이용하여 확률 추론을 수행한다.
- 최적화: 목적함수를 최소화하는 최적해를 찾는 데 활용하며, 주로 미분을 활용한 방법을 사용한다. 수학자들이 개발한 최적화 방법을 기계 학습이라는 도메인에 어떻게 효율적으로 적용할지가 주요 관심사이다.

2.1 선형대수

- 2.1.1 벡터와 행렬
- 2.1.2 놈과 유사도
- 2.1.3 퍼셉트론의 해석
- 2.1.4 선형결합과 벡터공간
- 2.1.5 역행렬
- 2.1.6 행렬 분해

2.1.1 벡터와 행렬

벡터

- 샘플을 특징 벡터로feature vector 표현
- 예) Iris 데이터에서 꽃받침의 길이, 꽃받침의 너비, 꽃잎의 길이, 꽃잎의 너비라는 4개의 특징이 각각 5.1, 3.5, 1.4, 0.2인 샘플

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$$

- 여러 개의 특징 벡터를 첨자로 구분

$$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 4.7 \\ 3.2 \\ 1.3 \\ 0.2 \end{pmatrix}, \quad \cdots, \mathbf{x}_{150} = \begin{pmatrix} 5.9 \\ 3.0 \\ 5.1 \\ 1.8 \end{pmatrix}$$

2.1.1 벡터와 행렬

- 행렬
  - **여러 개의 벡터를 담음**
  - 훈련집합을 담은 행렬을 설계행렬이라 부름
  - 예) Iris 데이터에 있는 150개의 샘플을 설계 행렬 $\mathbf{X}$로 표현

$$\mathbf{X} = \begin{pmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ \vdots & \vdots & \vdots & \vdots \\ 6.2 & 3.4 & 5.4 & 2.3 \\ 5.9 & 3.0 & 5.1 & 1.8 \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ \vdots & \vdots & \vdots & \vdots \\ x_{149,1} & x_{149,2} & x_{149,3} & x_{149,4} \\ x_{150,1} & x_{150,2} & x_{150,3} & x_{150,4} \end{pmatrix}$$

행 row

열 column

2.1.1 벡터와 행렬

- 행렬 $\mathbf{A}$의 전치행렬 $\mathbf{A}^{\mathrm{T}}$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}, \quad \mathbf{A}^{\mathrm{T}} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \ldots & a_{nm} \end{pmatrix}$$

예를 들어, $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$라면 $\mathbf{A}^{\mathrm{T}} = \begin{pmatrix} 3 & 0 \\ 4 & 5 \\ 1 & 2 \end{pmatrix}$

- Iris의 설계 행렬을 **전치행렬** 표기에 따라 표현하면,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{150}^{\mathrm{T}} \end{pmatrix} \longleftarrow \boxed{\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}}$$

2.1.1 벡터와 행렬

- **행렬을 이용하면 수학을 간결하게 표현할 수 있음**
  - 예) 다항식의 행렬 표현

$$
\begin{aligned}
f(\mathbf{x}) &= f(x_1, x_2, x_3) \\
&= 2x_1x_1 - 4x_1x_2 + 3x_1x_3 + x_2x_1 + 2x_2x_2 + 6x_2x_3 - 2x_3x_1 + 3x_3x_2 + 2x_3x_3 + 2x_1 + 3x_2 - 4x_3 + 5 \\
&= \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 2 & -4 & 3 \\ 1 & 2 & 6 \\ -2 & 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 2 & 3 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + 5 \\
&= \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c
\end{aligned}
$$

- 특수한 행렬들

정사각행렬 $\begin{pmatrix} 2 & 0 & 1 \\ 1 & 21 & 5 \\ 4 & 5 & 12 \end{pmatrix}$, 대각행렬 $\begin{pmatrix} 50 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix}$,

단위행렬 $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, 대칭행렬 $\begin{pmatrix} 1 & 2 & 11 \\ 2 & 21 & 5 \\ 11 & 5 & 1 \end{pmatrix}$

2.1.1 벡터와 행렬

행렬 연산

- **행렬 곱셈** $\mathbf{C} = \mathbf{AB}$, 이때 $c_{ij} = \sum_{k=1,s} a_{ik} b_{kj}$ (2.1)

2*3 행렬 $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 0 & 5 & 2 \end{pmatrix}$와 3*3행렬 $\mathbf{B} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 0 & 5 \\ 4 & 5 & 1 \end{pmatrix}$을 곱하면 2*3 행렬 $\mathbf{C} = \mathbf{AB} = \begin{pmatrix} 14 & 5 & 24 \\ 13 & 10 & 27 \end{pmatrix}$

  - **교환법칙 성립하지 않음: $\mathbf{AB} \neq \mathbf{BA}$**
  - **분배법칙과 결합법칙 성립: $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$이고 $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$**

- **벡터의 내적**

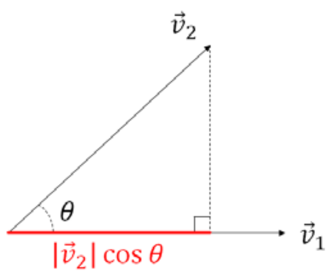
벡터의 내적 $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_{k=1,d} a_k b_k$ (2.2)

$\mathbf{x}_1 = \begin{pmatrix} 5.1 \\ 3.5 \\ 1.4 \\ 0.2 \end{pmatrix}$와 $\mathbf{x}_2 = \begin{pmatrix} 4.9 \\ 3.0 \\ 1.4 \\ 0.2 \end{pmatrix}$의 내적 $\mathbf{x}_1 \cdot \mathbf{x}_2$는 37.49

$\vec{v}_1 \cdot \vec{v}_2 = [a \quad b] \begin{bmatrix} x \\ y \end{bmatrix} = ax + by$

why?

=



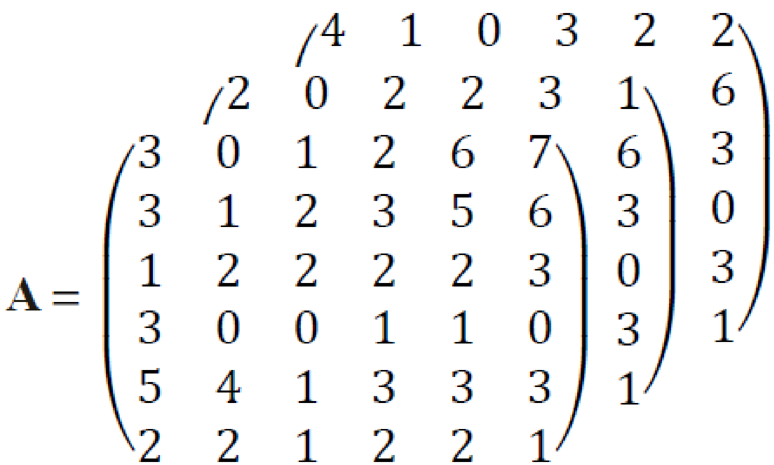
$\vec{v}_1 \cdot \vec{v}_2 = |\vec{v}_1||\vec{v}_2| \cos\theta$

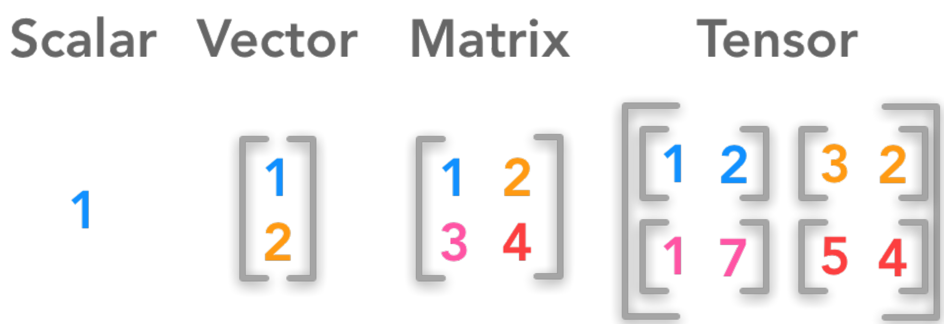
내적: 벡터에는 방향이 있으므로,
방향이 일치하는 만큼만 곱

2.1.1 벡터와 행렬

텐서

- **3차원 이상의 구조를 가진 숫자 배열**
- 예) 3차원 구조의 RGB 컬러 영상





행렬은 2차원 숫자 배열을 표현할 수 있는데, 기계학습에서는 3차원 이상의 숫자 배열이 필요한 경우가 종종 있음 (RGB 영상 데이터 → 2차원 행렬이 세 장 있음)

스칼라: 0차원 텐서
벡터: 1차원 텐서
행렬: 2차원 텐서

2.1.2 놈과 유사도

Norm이 측정한 벡터의 크기는 원점에서 벡터 좌표까지의 거리 혹은 magnitude라고 합

p=2일 때는 유클리드 공간에서의 일반적으로 이야기하는 거리에 해당한다.

벡터와 행렬의 크기를 놈으로 측정

- 벡터의 p차 놈

$$p\text{차 놈: } \|\mathbf{x}\|_p = \left(\sum_{i=1,d} |x_i|^p\right)^{\frac{1}{p}} \tag{2.3}$$

$$\text{최대 놈: } \|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \cdots, |x_d|) \tag{2.4}$$

- 예) $\mathbf{x} = (3 \;\; -4 \;\; 1)$ 일 때, 2차 놈은 $\|\mathbf{x}\|_2 = (3^2 + (-4)^2 + 1^2)^{1/2} = 5.099$

- 행렬의 프로베니우스 놈

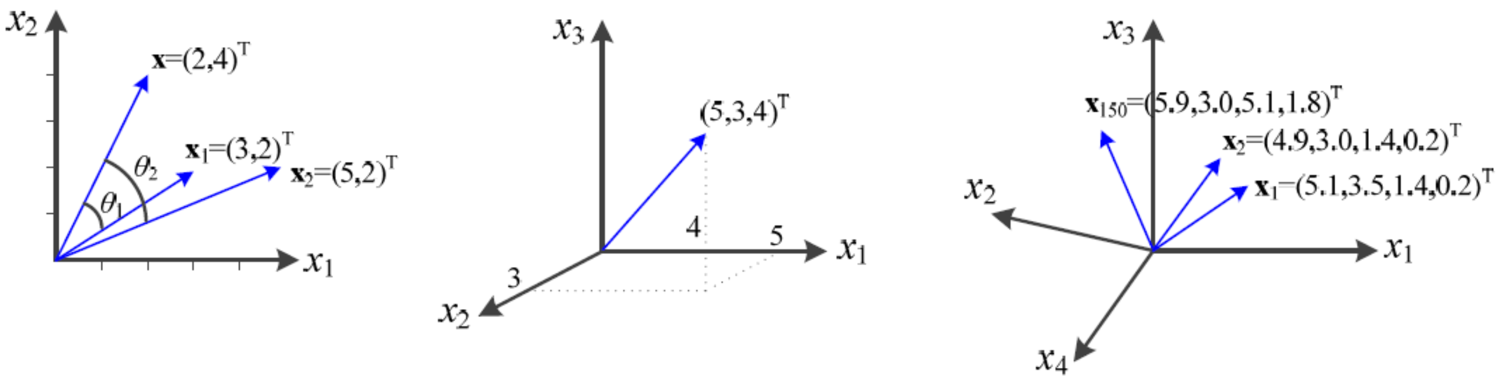
$$\text{프로베니우스 놈: } \|\mathbf{A}\|_F = \left(\sum_{i=1,n}\sum_{j=1,m} a_{ij}^2\right)^{\frac{1}{2}} \tag{2.6}$$

예를 들어, $\left\|\begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix}\right\|_F = \sqrt{2^2 + 1^2 + 6^2 + 4^2} = 7.550$

2.1.2 놈과 유사도

■ 유사도와 거리

- 벡터를 기하학적으로 해석



(a) 2차원 벡터 (b) 3차원 벡터 (c) 4차원 벡터(Iris 데이터)

그림 2-2 벡터를 기하학적으로 해석

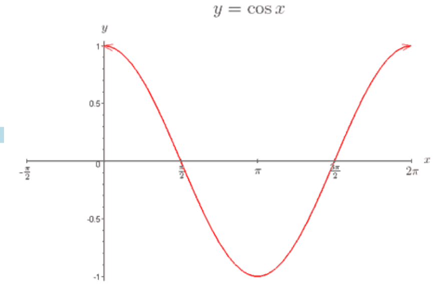
두 벡터 사이의 거리 → 유클리디안 거리로 측정할 수 있고, 이는 벡터를 이용하여 2차 놈으로도 표현 가능함
→ 내적을 통해서 계산 가능

x=(x_1, x_2), y=(y_1, y_2) → d(x, y) = $\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$ = $\|x-y\|_2$

d(x, y) = $\sqrt{x_1^2 + y_1^2 + x_2^2 + y_2^2 - 2(x_1 y_1 + x_2 y_2)}$
= $\sqrt{x \cdot x + y \cdot y - 2x \cdot y}$

만약 벡터 x와 y가 같으면 0이 된다.

2.1.2 놈과 유사도



유사도와 거리

- 코사인 유사도

$$cosine\_similarity(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}}{||\mathbf{a}||} \cdot \frac{\mathbf{b}}{||\mathbf{b}||} = cos(\theta) \qquad (2.7)$$

유사도는 −1에서 1까지의 값을 가짐
: −1은 서로 완전히 반대되는 경우, 0은 서로 독립적인 경우, 1은 서로 완전히 같은 경우를 의미

주어진 두 벡터에 대해서 어떻게 cosine 값을 구할 것인가?

$$\mathrm{a} \cdot b = \|a\|\|b\| \cos\theta$$

a, b가 직교하면, θ가 90이 되고 cos(90)은 0이기 때문에 내적은 0이 된다.

1. 코사인 유사도(Cosine Similarity)

[참고] https://wikidocs.net/24603

코사인 유사도는 두 벡터 간의 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도를 의미합니다. 두 벡터의 방향이 완전히 동일한 경우에는 1의 값을 가지며, 90°의 각을 이루면 0, 180°로 반대의 방향을 가지면 -1의 값을 갖게 됩니다. 즉, 결국 코사인 유사도는 -1 이상 1 이하의 값을 가지며 값이 1에 가까울수록 유사도가 높다고 판단할 수 있습니다. 이를 직관적으로 이해하면 두 벡터가 가리키는 방향이 얼마나 유사한가를 의미합니다.

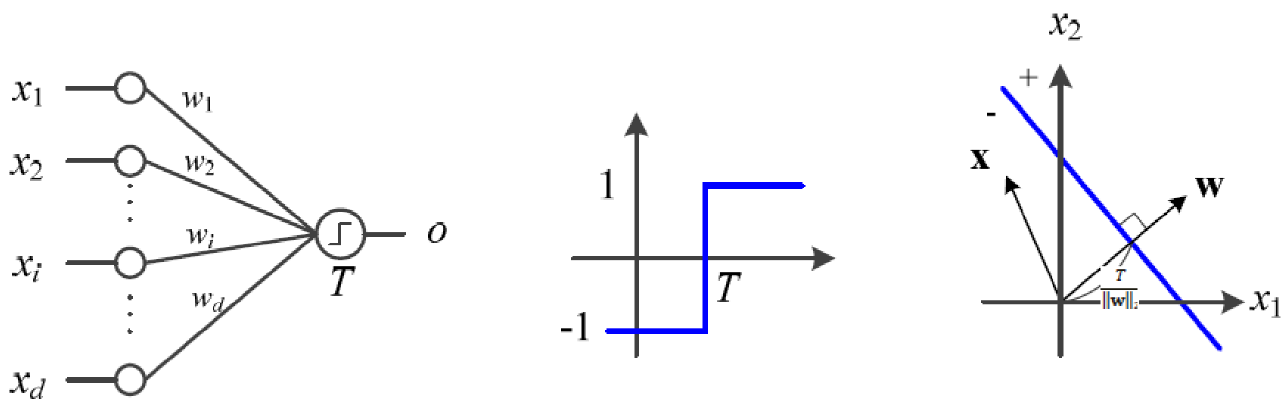
두 벡터 A, B에 대해서 코사인 유사도는 식으로 표현하면 다음과 같습니다.

$$similarity = cos(\Theta) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

2.1.3 퍼셉트론의 해석

퍼셉트론

- 1958년 로젠블렛이 고안한 분류기 모델



(a) 퍼셉트론 구조 (b) 계단형 활성함수(비선형) (c) 퍼셉트론의 공간 분할

그림 2-3 퍼셉트론의 구조와 동작

파란색 선: 두 개의 부분 공간을 나누는 결정 직선 (decision line)

벡터 w에 수직이고, 원점으로부터 $T/\|w\|_2$만큼 떨어져 있고, 결정 직선에 위치한 점 x는 w·x = T이다.

w·x < T 이면 -1, w·x > T 이면 +1의 공간으로 나누는 것

- 퍼셉트론의 동작을 수식으로 표현하면,

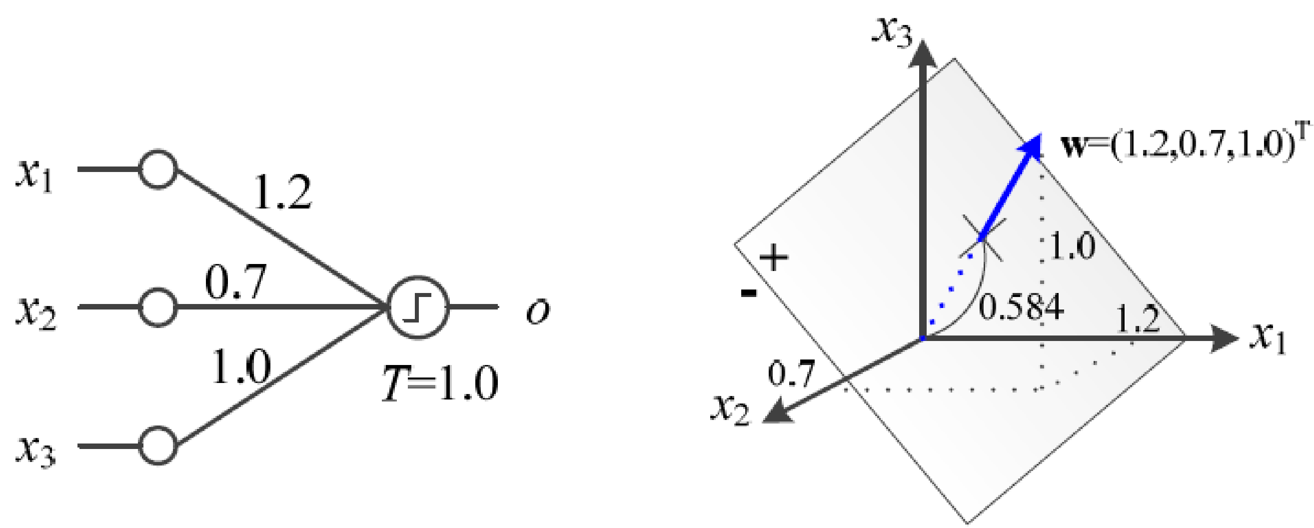
$$o = \tau(\mathbf{w} \cdot \mathbf{x}), \quad \text{이때} \quad \tau(a) = \begin{cases} 1, & a \geq T \\ -1, & a < T \end{cases} \qquad (2.8)$$

  - 활성 함수 τ로는 계단함수 사용

2.1.3 퍼셉트론의 해석

퍼셉트론

- [그림 2-3(c)]의 파란 직선은 두 개의 부분공간을 나누는 결정직선decision line
  - $\mathbf{w}$에 수직이고 $\frac{T}{\|\mathbf{w}\|_2}$만큼 떨어져 있음
- 3차원 특징공간은 결정평면decision plane, 4차원 이상은 결정 초평면decision hyperplane
- 예) 3차원 특징공간을 위한 퍼셉트론



(a) 퍼셉트론　　(b) 공간 분할(2부류 분류)

그림 2-4 퍼셉트론의 예(3차원)

2.1.3 퍼셉트론의 해석

- 출력이 여러 개인 퍼셉트론

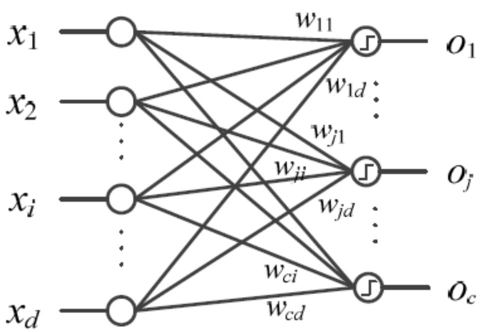


그림 2-5 출력이 여러 개인 퍼셉트론

출력은 벡터 $\mathbf{o} = (o_1, o_2, \cdots, o_c)^{\mathrm{T}}$로 표기

j번째 퍼셉트론의 가중치 벡터를
$\mathbf{w}_j = (w_{j1}, w_{j2}, \cdots, w_{jd})^{\mathrm{T}}$와 같이 표기

- 동작을 수식으로 표현하면,

$$\mathbf{o} = \tau \begin{pmatrix} \mathbf{w}_1 \cdot \mathbf{x} \\ \mathbf{w}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{w}_c \cdot \mathbf{x} \end{pmatrix} \longrightarrow$$

행렬로 간결하게 쓰면 $\mathbf{o} = \boldsymbol{\tau}(\mathbf{W}\mathbf{x})$

이때 $\mathbf{W} = \begin{pmatrix} \mathbf{w}_1^{\mathrm{T}} \\ \mathbf{w}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{w}_c^{\mathrm{T}} \end{pmatrix}$

- 가중치 벡터를 각 부류의 기준 벡터로 간주하면, c개 부류의 유사도를 계산하는 셈

2.1.3 퍼셉트론의 해석

학습의 정의

- 식 (2.10)은 학습을 마친 프로그램을 현장에 설치했을 때 일어나는 과정

$$\text{분류라는 과업:}\quad \overset{?}{\mathbf{o}} = \tau(\overset{\text{앎}}{\mathbf{W}}\,\overset{\text{앎}}{\mathbf{x}}) \qquad (2.10)$$

- 식 (2.11)은 학습 과정
  - 학습은 훈련집합의 샘플에 대해 식 (2.11)을 가장 잘 만족하는 **w**를 찾아내는 작업

$$\text{학습이라는 과업:}\quad \overset{\text{앎}}{\mathbf{o}} = \tau(\overset{?}{\mathbf{W}}\,\overset{\text{앎}}{\mathbf{x}}) \qquad (2.11)$$

현대 기계 학습에서 퍼셉트론의 중요성

- **딥러닝은 퍼셉트론을 여러 층으로 확장하여 만듦**

2.1.4 선형결합과 벡터공간

벡터

- 공간상의 한 점으로 화살표 끝이 벡터의 좌표에 해당

벡터 a와 b에 각각 상수를 곱한 다음 결과를 더하는 연산을 선형 결합(linear combination)이라고 한다.
→ 퍼셉트론에서 w·x 에 해당한다.

그림 2-7에서 α1와 α2를 변형시키면 2차원 공간의 모든 점을 만들 수 있다. 즉 a와 b의 선형 결합으로 2차원 공간 전체를 덮을 수 있다.

2-6에서는 α1와 α2를 변형시켜도 결코 직선을 벗어나지 못한다. 이런 경우(즉, 기저벡터가 선형독립이 아닌 경우) 벡터 공간을 만들 수 없다.

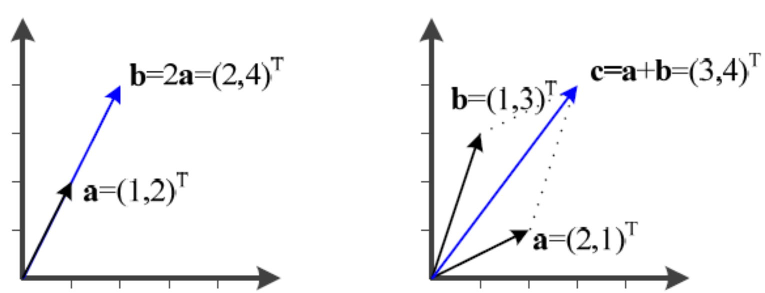
기계 학습에서는 행렬의 공간 변환 능력을 이용하여 분류모델을 만든다. 딥러닝은 공간 변환을 잘하기 때문에 성능이 높은 것이고, 벡터공간이 만들어지지 않는다면 학습이 잘 안될 수 있다.

선형결합이 만드는 벡터공간

- 기저벡터 **a**와 **b**의 선형결합

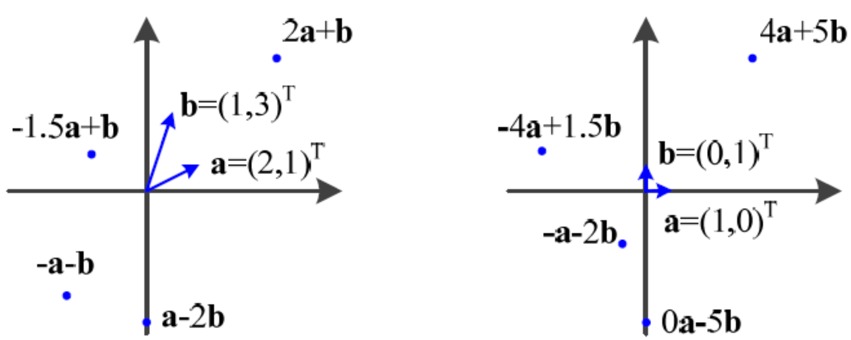
$$\mathbf{c} = \alpha_1 \mathbf{a} + \alpha_2 \mathbf{b}$$

- 선형결합으로 만들어지는 공간을 벡터공간이라 부름



(a) 벡터에 스칼라 곱 (b) 두 벡터의 덧셈

그림 2-6 벡터의 연산

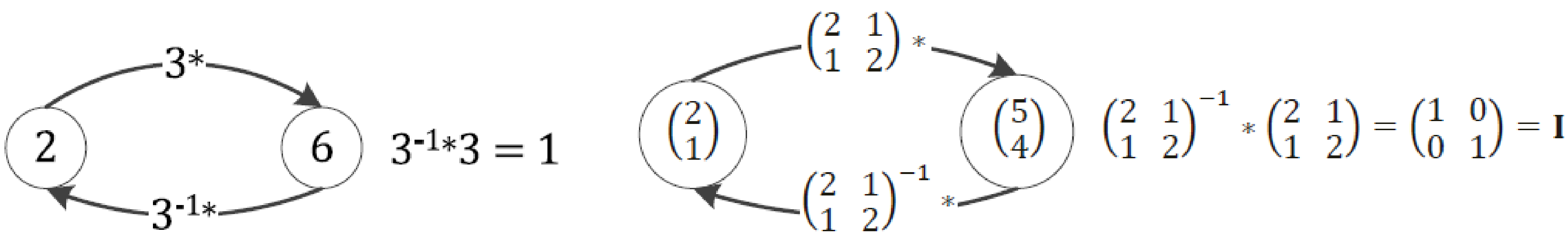


(a) 기저 벡터와 벡터공간 (b) 정규직교 기저 벡터

그림 2-7 벡터공간

2.1.5 역행렬

역행렬의 원리



(a) 역수의 원리 (b) 역행렬의 원리

그림 2-9 역행렬

- 정사각행렬 $\mathbf{A}$의 역행렬 $\mathbf{A}^{-1}$ (모든 행렬이 역행렬을 갖는 것은 아니다)

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

- 예를 들어,

$$\begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix}\text{의 역행렬은 }\begin{pmatrix} 2 & -0.5 \\ -3 & 1 \end{pmatrix}$$

예를 들어 2에 3을 곱하면 6으로 변환됨.
변환된 6을 원래 값으로 역변환 하려면 3의 역수 즉 3^{-1}=1/3을 곱하면 됨.
데이터 분석 과정에서 역행렬은 많이 활용됨 (뒤 고유값 분해)

2.1.5 역행렬

- 행렬 **A**의 행렬식 *det*(**A**)

$$\left.\begin{array}{l} \det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc \\ \det\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - ceg - bdi - afh \end{array}\right\} \quad (2.15)$$

예를 들어 $\begin{pmatrix} 2 & 1 \\ 6 & 4 \end{pmatrix}$의 행렬식은 2*4−1*6=2

- 기하학적 의미
  - 2차원에서는 2개의 행 벡터가 이루는 평행사변형의 넓이
  - 3차원에서는 3개의 행 벡터가 이루는 평행사각기둥의 부피

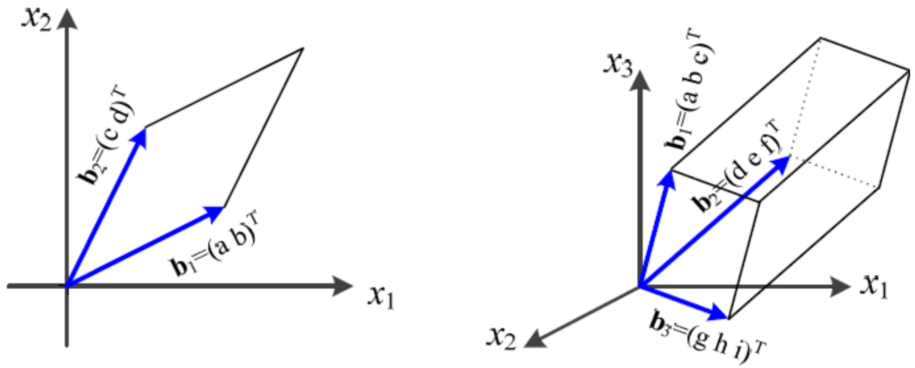


그림 2-10 행렬식의 기하학적 해석

행렬식이 0이라는 것은 행렬을 구성하는 벡터가 서로 동일선상(collinear)에 있다는 것을 의미함

이는 행렬을 구성하는 벡터가 해당 공간의 기저가 아님을 의미한다.

2.1.6 행렬 분해

분해란?

- 정수 3717은 특성이 보이지 않지만, 3*3*7*59로 소인수 분해를 하면 특성이 보이듯이, 행렬도 분해하면 여러모로 유용함

고유값과 고유 벡터

- 고유 벡터 $\mathbf{v}$와 고유값 λ

$$\mathbf{Av} = \lambda\mathbf{v}$$

- 예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3\begin{pmatrix} 1 \\ 1 \end{pmatrix}$이고 $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1\begin{pmatrix} 1 \\ -1 \end{pmatrix}$이므로,

$\lambda_1 = 3, \lambda_2 = 1$이고

$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

w·x
c = α_1a + α_2b
x에 해당하는 것이 (a, b)
(물론 x는 행렬, a, b는 벡터)
x에 α에 해당하는 w를 선형결합하면
벡터공간 만들어짐

➔ Ax = λx
➔ λ 는 원본 벡터 x를 A로 선형변환 했을 때
얼마나 축소하고 확대했는지에 대한 값

2.1.6 행렬 분해

■ 고윳값과 고유 벡터의 기하학적 해석

예제 2-5

[그림 2-12]의 반지름이 1인 원 위에 있는 4개의 벡터 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$가 $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$에 의해 어떻게 변환되는지 살펴보자. 변환 후의 벡터를 각각 $\mathbf{x}_1', \mathbf{x}_2', \mathbf{x}_3', \mathbf{x}_4'$로 표기한다.

$$\mathbf{x}_1' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \end{pmatrix}$$
$$\mathbf{x}_2' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$
$$\mathbf{x}_3' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$
$$\mathbf{x}_4' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$$

눈 여겨 볼 점은 $\mathbf{A}$의 고유 벡터 $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}$과 방향이 같은 $\mathbf{x}_1$과 $\mathbf{x}_3$이다. 이들은 변환 때문에 길이가 달라지더라도 방향은 그대로 유지한다. 식 (2.20)을 충실히 따르고 있다. 이때 길이의 변화는 고윳값 λ에 따른다. 즉, $\mathbf{x}_1$은 3배만큼, $\mathbf{x}_3$은 1배만큼 길이가 변한다. 나머지 $\mathbf{x}_2$와 $\mathbf{x}_4$는 길이와 방향이 모두 변한다. 파란 원 위에 있는 모든 점을 변환하면 빨간색의 타원이 된다. 파란 원 위에 존재하는 무수히 많은 점(벡터) 중에 방향이 바뀌지 않는 것은 고유 벡터에 해당하는 $\mathbf{x}_1$과 $\mathbf{x}_3$뿐이다.

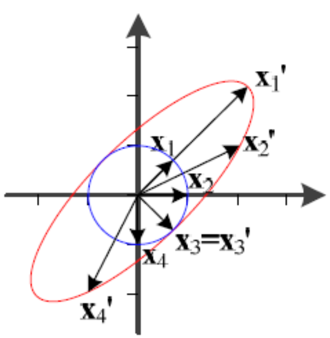


그림 2-12 고유 벡터의 공간 변환

- 예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3\begin{pmatrix} 1 \\ 1 \end{pmatrix}$이고 $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1\begin{pmatrix} 1 \\ -1 \end{pmatrix}$이므로,

  $\lambda_1 = 3, \lambda_2 = 1$이고

  $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ \mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

2.1.6 행렬 분해

- 고윳값 분해eigen value decomposition

$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1} \qquad (2.21)$$

  - $\mathbf{Q}$는 $\mathbf{A}$의 고유 벡터를 열에 배치한 행렬이고 $\boldsymbol{\Lambda}$(람다의 대문자)는 고윳값을 대각선에 배치한 대각 행렬
  - 예를 들어, $\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} \boxed{1} & \boxed{1} \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}$
  - **고윳값 분해는 정사각행렬에만 적용 가능한데, 기계 학습에서는 정사각행렬이 아닌 경우의 분해도 필요하므로 고윳값 분해는 한계를 가짐**

고윳값 분해eigen value decomposition

고유값 분해(eigen decomposition)

정방 행렬(nxn) A를 선형 변환으로 봤을 때(즉 열벡터 v에 행렬 A를 곱하는 것), **선형 변환 A에 의한 변환 결과가 자기 자신(열벡터 v)의 상수 배가 되는 0이 아닌 벡터를 고유벡터(eigenvector, v)라고 하고, 이 상수배 값을 고유값(eigenvalue, λ)**이라고 함

➔ A**v** = λ**v**

A**v** = λ**v**이 만족한다는 것은 **벡터 v에 대해 선형 변환 A를 해주었을 때, 벡터 v의 방향은 변하지 않고 크기만 변했다**는 뜻

보통 어떤 벡터에 선형 변환을 하면 방향이 바뀜. 하지만 선형 변환을 했음에도 어떤 벡터 **v**의 방향이 바뀌지 않고 크기만 변했다면, 그리고 변한 크기가 원래 벡터 크기의 λ배라면 그 λ를 고유값이라고 하고 **v**는 고유 벡터라고 함

행렬 A의 **고유 벡터들을 열 벡터로 하는 행렬을 Q(v에서 도출)**, 고유값을 대각원소로 가지는 **대각 행렬을 Λ**라 하면 다음 식이 성립

➔ AQ = QΛ
➔ A = QΛQ^{-1}

이것을 행렬 A에 대한 고유값 분해라고 함

Q

$$A[\ \boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots\ \boldsymbol{v}_n\ ] = [\ \lambda_1\boldsymbol{v}_1\ \lambda_2\boldsymbol{v}_2\ \cdots\ \lambda_n\boldsymbol{v}_n\ ]$$

$$= [\ \boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots\ \boldsymbol{v}_n\ ]\begin{bmatrix}\lambda_1 & & & 0\\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n\end{bmatrix}$$

$$\begin{pmatrix} v_{11} & v_{12} & v_{13}\\ v_{21} & v_{22} & v_{23}\\ v_{31} & v_{32} & v_{33}\end{pmatrix}\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{pmatrix} = \begin{pmatrix}\lambda_1 v_{11} & \lambda_2 v_{12} & \lambda_3 v_{13}\\ \lambda_1 v_{21} & \lambda_2 v_{22} & \lambda_3 v_{23}\\ \lambda_1 v_{31} & \lambda_2 v_{32} & \lambda_3 v_{33}\end{pmatrix}$$

대각행렬은 위 그림과 같이 대각 원소의 크기 만큼 상수배가 될 수 있음

2.1.6 행렬 분해

A = QΛQ^{-1} → 정방행렬 n*n에만 가능

- ***m*n* 행렬 A의 특이값 분해**SVD(singular value decomposition)

$$\mathbf{A} = \mathbf{U\Sigma V^T} \tag{2.22}$$

  - 왼쪽 특이행렬 **U**는 **AA**$^{\mathrm{T}}$의 고유 벡터를 열에 배치한 *m**m* 행렬
  - 오른쪽 특이행렬 **V**는 **A**$^{\mathrm{T}}$**A**의 고유 벡터를 열에 배치한 *n**n* 행렬
  - **Σ**는 **AA**$^{\mathrm{T}}$의 고윳값의 제곱근을 대각선에 배치한 *m**n* 대각행렬

$A = U\Sigma V^T$

$AA^T = U\Sigma V^T (U\Sigma V^T)^T$

$= U\Sigma V^T V \Sigma^T U^T$

$= U\Sigma\Sigma^T U^T$

$= U(\Sigma\Sigma^T)U^{-1}$ {← $U^T = U^{-1}$ 이므로}

∴ AA^T를 고유값 분해하면

$U(\Sigma\Sigma^T)U^{-1}$ 가 됨.

⇒ U는 AA^T의 고유벡터들로 구성된 행렬

예를 들어, **A**를 4*3 행렬이라고 했을 때 다음과 같이 특잇값 분해가 된다.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 3 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix} = \begin{pmatrix} -0.1914 & -0.2412 & 0.1195 & -0.9439 \\ -0.5144 & 0.6990 & -0.4781 & -0.1348 \\ -0.6946 & -0.6226 & -0.2390 & 0.2697 \\ -0.4651 & 0.2560 & 0.8367 & 0.1348 \end{pmatrix}$$

$$\begin{pmatrix} 3.7837 & 0 & 0 \\ 0 & 2.7719 & 0 \\ 0 & 0 & 1.4142 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -0.7242 & -0.4555 & -0.5177 \\ -0.6685 & 0.2797 & 0.6891 \\ 0.1690 & -0.8452 & 0.5071 \end{pmatrix}$$

[참고] https://bkshin.tistory.com/entry/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-20-%ED%8A%B9%EC%9D%B4%EA%B0%92-%EB%B6%84%ED%95%B4Singular-Value-Decomposition

특이값 분해의 활용

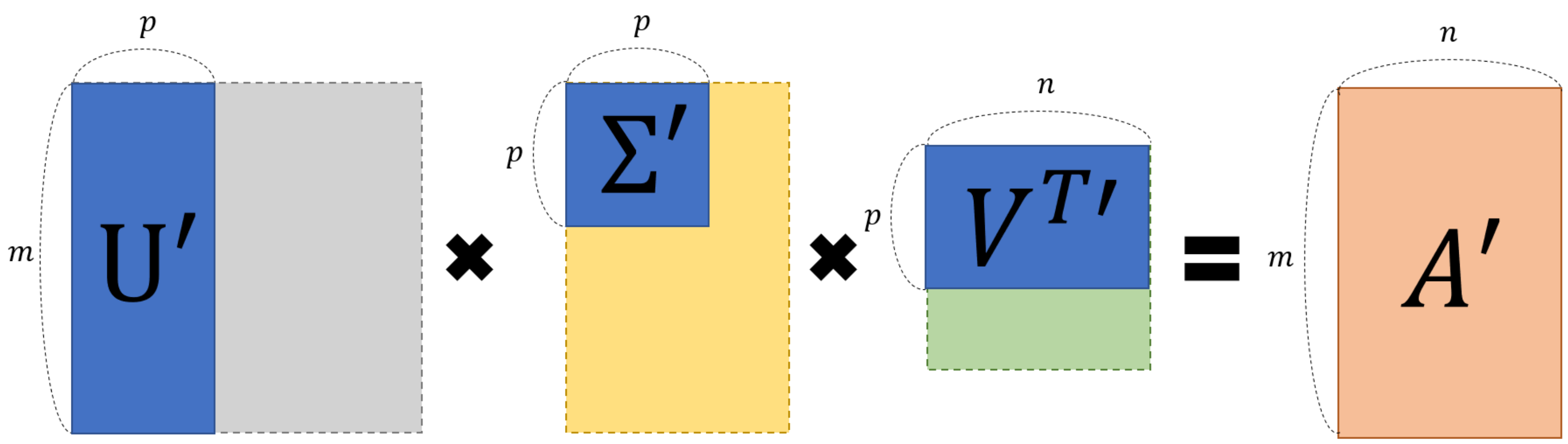


그림. 특이값 분해를 통해 얻어진 U, Sigma, V 행렬에서 일부만을 이용해 적당한 A'를 부분복원 하는 과정

특이값 분해는 분해되는 과정보다는 분해된 행렬을 다시 조합하는 과정이 더 많이 응용이 됨

기존의 U,Σ,V^T 로 분해되어 있던 A행렬을 특이값 p개만을 이용해 A'라는 행렬로 '부분 복원' 시킬 수 있다. (A'는 A와 완전히 같지는 않지만 거의 근사하다)

특이값의 크기에 따라 A의 정보량이 결정되기 때문에 값이 큰 몇 개의 특이값들을 가지고도 충분히 유용한 정보를 유지할 수 있다.

[참고] https://angeloyeo.github.io/2019/08/01/SVD.html