3.4 특잇값 분해

정방행렬은 고유분해로 고윳값과 고유벡터를 찾을 수 있었다.
정방행렬이 아닌 행렬은 고유분해가 불가능하다. 하지만 대신 고유분해와 비슷한 특이분해를 할 수 있다.

(특잇값과 특이벡터)

$N \times M$ 크기의 행렬 A를 다음과 같은 3개의 행렬의 곱으로 나타내는 것을 특이분해 (singular-decomposition) 또는 특잇값 분해 (singular value decomposition)라고 한다.

$$\boxed{A = U \Sigma V^T}$$

(# 고유분해 = $A = V \Lambda V^T$)

여기에서 U, Σ, V는 다음 조건을 만족해야 한다.

- 대각성분이 양수인 대각행렬이어야 한다. (큰 수부터 작은 수 순서로 배열한다.)

$$\Sigma \in \mathbb{R}^{N \times M}$$

- U는 N차원 정방행렬로, 모든 열벡터가 단위벡터이고 서로 직교해야 한다.

$$U \in \mathbb{R}^{N \times N}$$

- V는 M차원 정방행렬로, 모든 열벡터가 단위벡터이고 서로 직교해야 한다.

$$V \in \mathbb{R}^{M \times M}$$

위 조건을 만족하는 행렬 Σ의 대각성분들을 특잇값 (singular value), 행렬 U의 열벡터들을 왼쪽 특이벡터 (left singular vector), 행렬 V의 행벡터들을 오른쪽 특이벡터 (right singular vector)라고 부른다.

[정리]

특이분해는 모든 행렬에 대해 가능하다.

즉, 어떤 행렬이 주어지더라도 위와 같이 특이분해 할 수 있다.

i) 특이분해, $N > M$일 때 → M개의 특잇값 (대각성분)

$$\underbrace{A}_{N\times M} = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_M & \cdots & u_N \end{bmatrix}}_{N\times N} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_M \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}}_{N\times M} \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_M^T \end{bmatrix}}_{M\times M}$$

ii) 특이분해, $N < M$일 때 → N개의 특잇값 (대각성분)

$$\underbrace{A}_{N\times M} = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}}_{N\times N} \underbrace{\begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & & & & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_N & 0 & \cdots & 0 \end{bmatrix}}_{N\times M} \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_N^T \\ \vdots \\ v_M^T \end{bmatrix}}_{M\times M}$$

"추천시스템, PCA 등에서 활용"

특잇값 분해의 축소형 reduced form

특잇값 대각행렬에서 0인 부분은 사실상 아무런 의미도 없기 때문에 대각행렬의 0 원소 부분과 이에 대응하는 왼쪽 (혹은 오른쪽) 특이벡터들을 없애고 다음처럼 축소된 형태로 해도 마찬가지로 원래 행렬이 나온다.

N이 M보다 큰 경우에는 왼쪽 특이벡터 중에서 $u_{M+1}, \cdots u_N$을 없앤다.

$$\underset{N\times M}{A} = \overbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_M \end{bmatrix}}^{M} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_M \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_M^T \end{bmatrix}$$

$U = N \times M$ $\Sigma = M \times M$ $V^T = M \times M$

N이 M보다 작은 경우에는 오른쪽 특이벡터 중에서 $v_{N+1}, \cdots v_M$을 없앤다.

$$\underset{N\times M}{A} = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_N \end{bmatrix} \overbrace{\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_N^T \end{bmatrix}}^{M}$$

$U = N \times N$ $\Sigma = N \times N$ $V^T = N \times M$

(파이썬을 사용한 특이분해)

transpose 된 상태.

```
from numpy.linalg import svd    // U, S, V_T = svd(array)
from scipy.linalg import svd
```

singular 값만 나옴, 대각행렬로 만드려면 np.diag(S, 1)[:, 1:]

* 축소형 parameter svd(A, full_matrices=False)

특이값과 특이벡터의 관계

행렬 V는 정규직교(orthonormal) 행렬이므로,
전치행렬이 역행렬이다.

$$V^T = V^{-1}$$

특이분해된 등식의 양변에 V를 곱하면,

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

$$A\begin{bmatrix} v_1 & v_2 & \cdots & v_M \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}\begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

행렬 A를 곱하여 정리하면, M이 N보다 클 때는

$$\begin{bmatrix} Av_1 & Av_2 & \cdots & Av_N \end{bmatrix} = \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \cdots & \sigma_N u_N \end{bmatrix}$$

이 되고, N이 M보다 클 때는

$$\begin{bmatrix} Av_1 & Av_2 & \cdots & Av_M \end{bmatrix} = \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \cdots & \sigma_M u_M \end{bmatrix}$$

이 된다.

즉, i번째 특잇값 σ_i와 특이벡터 u_i, v_i는 다음과 같은 관계가 있다.

$$\boxed{Av_i = \sigma_i u_i \quad (i = 1, \ldots, \min(M, N))}$$

고유 $AV = V\Lambda$

특이 $AV = U\Sigma$

고유 $Av = \lambda v$

특이 $Av = \sigma u$

특이분해와 고유분해의 관계 ✓

행렬 A의 분산행렬 A^TA는

$$A^TA = (V\Sigma^T U^T)(U\Sigma V^T) = V\Lambda V^T$$

가 되어, 행렬 A의 특잇값의 제곱(과 0)이 분산행렬 A^TA의 고윳값, 행렬 A의 오른쪽 특이벡터가 분산행렬 A^TA의 고유벡터가 된다.

위 식에서 Λ은 $N > M$일 때,

$$\Lambda = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_M^2 \end{bmatrix}$$

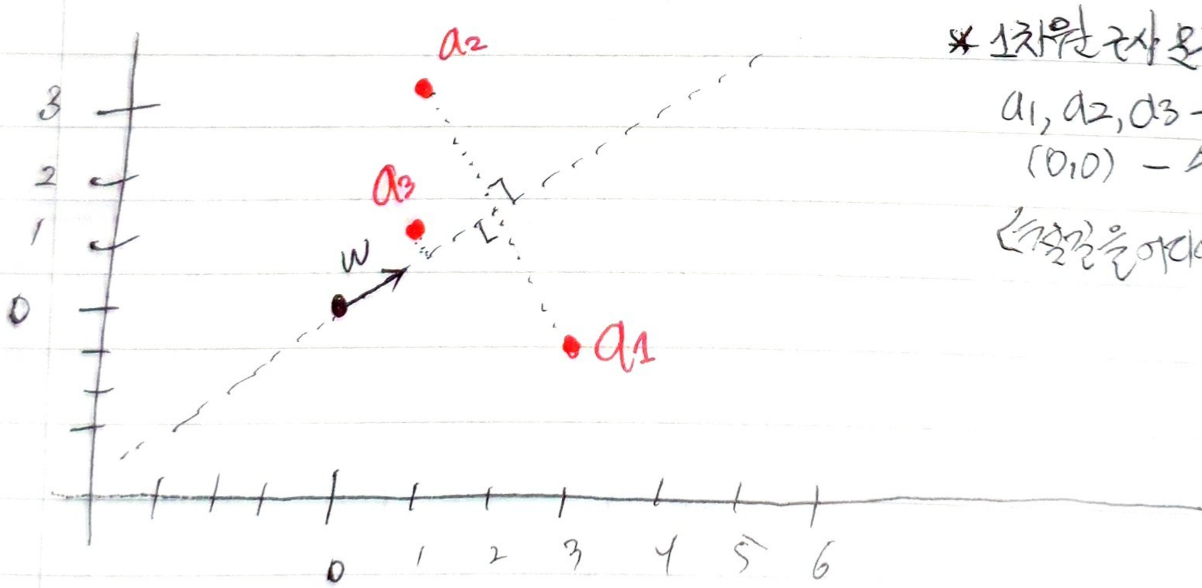
또, $M > N$일 때,

$$\Lambda = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

$$= \mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots \sigma_N^2, 0, \cdots 0)$$

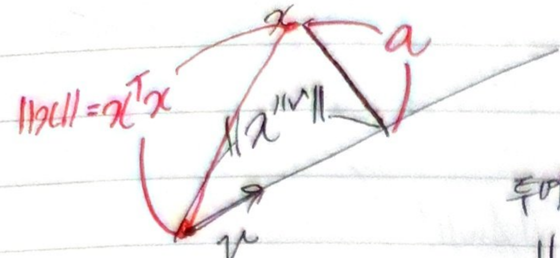
1차원 근사

* 특이분해를 배우는 이유? 근사(approximate) 문제해결 위해!
(이후 추천시스템, PCA 등 응용).



* 1차원 근사 문제.
a_1, a_2, a_3 — 마을
$(0,0)$ — 수도
(직선길을 어디에 놓아야 할까?)

① 투영문제 — 피타고라스정리 사용.



투영벡터

$$\|x^{\parallel v}\| = x^T v$$

$$\left(x^T \frac{v}{\|v\|}\right)$$

$$a^2 = \|x\|^2 - (x^T v)^2$$

위의 투영문제와 동일하게 계산하면,

벡터 w와 점 a_i의 거리의 제곱은 다음처럼 계산할 수 있다.

$$\|a_i^{\perp w}\|^2 = \|a_i\|^2 - \|a_i^{\parallel w}\|^2 = \|a_i\|^2 - (a_i^T w)^2$$

벡터 a_1, a_2, a_3를 행벡터로 가지는 행렬 A를 가정하면,

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$$

행벡터의 norm의 제곱합은 행렬의 norm이므로, 모든 점들간의 거리의 제곱의 합은 행렬의 norm으로 계산된다.

$$\sum_{i=1}^{3} \|a_i^{\perp w}\|^2 = \sum_{i=1}^{3} \|a_i\|^2 - \sum_{i=1}^{3} (a_i^T w)^2$$

$$= \|A\|^2 - \|Aw\|^2$$

점 a_i의 위치가 고정되어 있으므로, 행렬 A의 norm 값은 고정되어 있다. 따라서 이 값이 가장 작아지려면 $\|Aw\|^2$의 값이 가장 크게 만드는 w를 찾아야 한다. 이 문제는 다음처럼 수식으로 풀 수 있다.

$$“\arg\max_w \|Aw\|^2”$$

1차원 근사

2차원 평면 위에 3개의 2차원 벡터 a_1, a_2, a_3가 있다. 원점을 지나면서 모든 점들과 가능한 한 가까이 있는 직선을 만들고 싶다면 직선의 방향을 어떻게 해야 할까?

직선 방향을 나타내는 단위벡터를 w라고 하자.

고정지점 방향?

$$w^* = \arg\max_w \left\| \boxed{A}\,\boxed{w} \right\|$$

단, $\|w\| = 1$

1차원 근사의 풀이

예로 들었던 행렬 $A \in R^{3\times2}$를 특이분해하면 2개의 특잇값, 왼쪽/오른쪽 특이벡터를 가진다. 이를 각각 다음처럼 이름붙인다.

- 첫번째 특잇값: σ_1, 첫 번째 왼쪽 특이벡터 $u_1 \in R^3$,
  첫번째 오른쪽 특이벡터 $v_1 \in R^2$
- 두번째 특잇값: σ_2, 두 번째 왼쪽 특이벡터 $u_2 \in R^3$,
  두 번째 오른쪽 특이벡터 $v_2 \in R^2$

- 첫번째 특잇값 σ_1은 σ_2보다 크거나 같다.

$$\sigma_1 \ge \sigma_2$$

* A에 오른쪽 특이벡터를 곱하면, 왼쪽 특이벡터 방향이 된다.

$$\begin{aligned} A v_1 &= \sigma_1 u_1 \\ A v_2 &= \sigma_2 u_2 \end{aligned}$$

스칼라

길이는 다르지만 방향은 같음!

오른쪽 특이벡터 v_1, v_2는 직교하므로 (같은 방향 아니라서) 선형독립이고, 2차원 평면의 기저벡터가 될 수 있다.

** 선형 독립인 2차원벡터 2개는 2차원 plane의 기저벡터가 될 수 있다 → 선형조합으로 모든 2차원 공간의 벡터를 표현할 수 있다.

우리는 $\|Aw\|$의 값이 최대가 되는 w를 찾아야 하는데, w는 2차원 벡터이므로 2차원 평면 기저벡터인 v_1, v_2의 선형조합으로 표현할 수 있다.

$$\boxed{w = w_1 v_1 + w_2 v_2}$$

w는 단위벡터이므로 w_1, w_2는 다음 조건을 만족해야 한다.

$$\|w\|^2 = w_1^2 + w_2^2 = 1^2$$

이 때 $\|Aw\|$의 값은,

$$\begin{aligned} \|Aw\|^2 &= \| A(w_1 v_1 + w_2 v_2) \|^2 \\ &= \| w_1 A v_1 + w_2 A v_2 \|^2 \\ &= \| \underbrace{w_1 \sigma_1}_{\text{스칼라}} u_1 + \underbrace{w_2 \sigma_2}_{\text{스칼라}} u_2 \|^2 \rightarrow u_1, u_2 \text{ 선형조합} \\ &= w_1^2 \sigma_1^2 \|u_1\|^2 + w_2^2 \sigma_2^2 \|u_2\|^2 \quad (\text{orthogonal}) \end{aligned}$$

(피타고라스 정리 빗변$^2 = a^2 + b^2$)

* u_1, u_2는 단위벡터이므로

$$= w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 \quad (\text{unit vector})$$

$(\sigma_1 > \sigma_2 > 0)$

그러므로 $w_1^2 + w_2^2 = 1$ 이라는 조건을 만족하면서

$w_1^2\sigma_1^2 + w_2^2\sigma_2^2$를 가장 크게 하는 w_1, w_2를 구하는 문제.

$$w_1 = 1, \quad w_2 = 0$$

즉, 첫번째 오른쪽 특이벡터 방향으로 가치관을 놓으면 된다.

$$w = v_1$$

이때 $\|Aw\|$는 첫번째 특이값이 된다.

$$\|Aw\| = \|Av_1\| = \|\sigma_1 u_1\| = \sigma_1\|u_1\| = \sigma_1$$

$\boxed{\text{일반적인 풀이}}$

만약 $N=3$이 아니라 일반적인 경우에는 다음처럼 풀 수 있다.

$$\|Aw\|^2 = \sum_{i=1}^{N} (a_i^T w)^2$$

$$= \sum_{i=1}^{N} (a_i^T w)^T (a_i^T w)$$

$$= \sum_{i=1}^{N} w^T a_i a_i^T w$$

$$= w^T \left(\sum_{i=1}^{N} a_i a_i^T \right) w$$

$$= w^T A^T A w$$

분산행렬의 고유분해 공식을 이용하면,

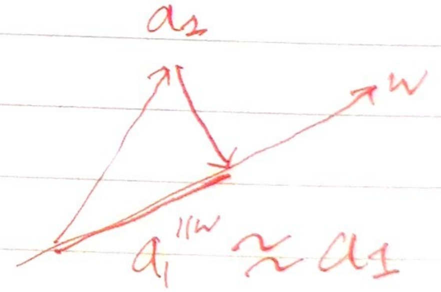
$$
\begin{aligned}
w^T A^T A w &= w^T V \Lambda V^T w \\
&= w^T \left(\sum_{i=1}^{M} \sigma_i^2 v_i v_i^T \right) w \quad \text{rank-1 행렬의 합} \\
&= \sum_{i=1}^{M} \sigma_i^2 (w^T v_i)(v_i^T w) \\
&= \sum_{i=1}^{M} \sigma_i^2 \| v_i^T w \|^2
\end{aligned}
$$

이 식에서 M은 0이 아닌 특잇값 갯수다.
즉, 우리가 풀어야 할 문제는 다음과 같다.

$$\boxed{\arg\max_w \|Aw\|^2 = \arg\max_w \sum_{i=1}^{M} \sigma_i^2 \|v_i^T w\|^2}$$

이 값을 가장 크게 하려면, w를 가장 큰 특잇값에 대응하는 오른쪽 고유벡터 v_1로 해야 한다.

Rank-1 근사 문제



투영벡터를 원본벡터처럼!

또, a_i를 w에 투영한 벡터는

$$a_i^{\parallel w} = (a_i^T w) w$$

이므로, w벡터를 이용하면 N개의 M차원 벡터 $a_1, a_2, \cdots, a_N$ ($a_i \in \mathbb{R}^M$)을 1차원으로 투영(projection)하여 가장 비슷한 N개의 1차원 벡터 $a_1^{\parallel w}, a_2^{\parallel w}, \cdots a_N^{\parallel w}$ ($a_i^{\parallel w} \in \mathbb{R}^1$)을 만들 수 있다.

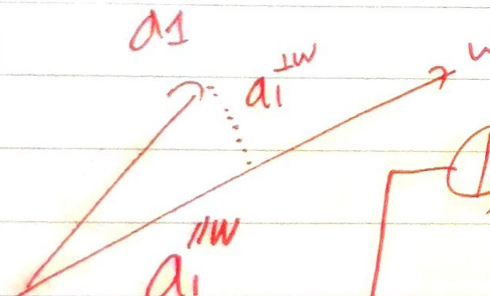
$$A' = \begin{bmatrix} (a_1^{/\!/w})^T \\ (a_2^{/\!/w})^T \\ \vdots \\ (a_N^{/\!/w})^T \end{bmatrix} = \begin{bmatrix} a_1^T ww^T \\ a_2^T ww^T \\ \vdots \\ a_N^T ww^T \end{bmatrix} = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_N^T \end{bmatrix} ww^T = Aww^T$$

이 답은 원래 행렬 A에 rank-1 행렬 ww^T를 곱해서 원래의 행렬 A와 가장 비슷한 행렬 A'을 만드는 문제와 같다.

$$\arg\min_w \|A - A'\| = \arg\min_w \|A - A\underbrace{ww^T}_{\text{rank-1 matrix}}\|$$

이러한 문제를 rank-1 행렬 근사 문제 (rank-1 approximation problem)라고 한다.

☆관점의 차이



① $\arg\min_w \|a_1^{\perp w}\|$

→ $\|Aw\|$ 최대화

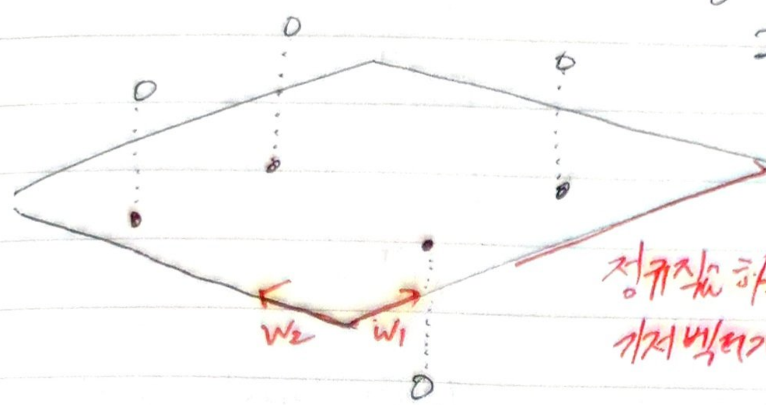
② a_1과 $a_1^{/\!/w}$ 근사

→ $\|A - Aww^T\|$ 최소화

"사실상 같은 해를 갖는 같은 문제"

K차원 근사

3차원 공간의 점들을 가장 가까운 2차원 평면 위로 놓자
→ 2차원 근사문제



정규직교 하는 기저벡터가 만드는 공간

N개의 M차원 벡터 $a_1, a_2, \cdots, a_N$ ($a_i \in R^M$)를 1차원이 아니라 정규직교인 기저벡터 $w_1, w_2, \cdots, w_K$로 이루어진 K차원 벡터공간으로 투영하여, 가장 비슷한 N개의 K차원 벡터 $a_1^{\Vert w}, a_2^{\Vert w}, \cdots, a_N^{\Vert w}$를 만들기 위한 정규직교 기저벡터 $w_1, w_2, \cdots w_K$를 찾는 문제를 생각하자. 이 문제를 rank-K 근사문제라고 한다.

기저벡터행렬을 W라고 하자. (단위벡터, 서로직교) → 여러개를 찾아야함.

$$W = [w_1 \quad w_2 \quad \cdots \quad w_K]$$

기저벡터들의 구성을 어떻게 하면 → 어떤각으로 잘 평면을 만들면 물체들과 가장 가깝게 볼 수 있나?

일반적으로는 쓰지않는다

* 정규직교 기저벡터에 대한 벡터 a_i의 투영 $a_i^{\Vert w}$는 각 기저벡터에 대한 내적의 합으로 만들 수 있다.

$$a_i^{\Vert w} = (a_i^T w_1) w_1 + (a_i^T w_2) w_2 + \cdots + (a_i^T w_K) w_K$$
$$= \sum_{k=1}^{K} (a_i^T w_k) w_k$$

벡터 $a_1, a_2, \dots a_N$을 행벡터로 가지는 행렬 A를 가정하면

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_N^T \end{bmatrix}$$

모든 점들과의 거리의 제곱의 합은 다음처럼 행렬의 놈으로 계산할 수 있다.

$$\sum_{i=1}^{N} \|a_i^{\perp w}\|^2 = \sum_{i=1}^{N} \|a_i\|^2 - \sum_{i=1}^{N} \|a_i^{\parallel w}\|^2$$

$$= \|A\|^2 - \sum_{i=1}^{N} \|a_i^{\parallel w}\|^2$$

행렬 A는 이미 주어져 있으므로, 이 값을 가장 작게 하려면 두번째 항의 값을 가장 크게 하면 된다. 두번째 항은 $K=1$일 때와 같은 방법으로 분산행렬 형태로 바꿀 수 있다.

$$\sum_{i=1}^{N} \|a_i^{\parallel w}\|^2 = \sum_{i=1}^{N} \sum_{k=1}^{K} \|(a_i^T w_k) w_k\|^2$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \|a_i^T w_k\|^2$$

$$= \sum_{k=1}^{K} w_k^T A^T A w_k$$

분산행렬의 고유분해를 사용하면

$$\sum_{k=1}^{K} w_k^T A^T A w_k = \sum_{k=1}^{K} w_k^T V \Lambda V^T w_k$$

$$= \sum_{k=1}^{K} w_k^T \left(\sum_{i=1}^{M} \sigma_i^2 v_i v_i^T \right) w_k$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{M} \sigma_i^2 \|v_i^T w_k\|^2$$

이 문제도 1차원 근사문제처럼 풀면 다음과 같은 답을 얻을수 있다.

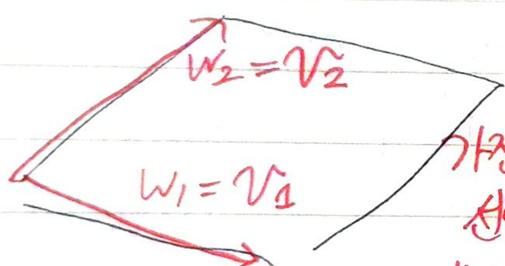
* 가장 큰 k개의 특잇값에 대응하는 오른쪽 특이벡터가 기저벡터일 때 가장 값이 커진다.

Ex 3차원 → 2차원 근사

$$\sigma_1^2 \| v_1 w_1 \| + \sigma_2^2 \| v_2 w_2 \|$$

앞 항과 똑같이 v_1을 선택하고 싶으나, 선형독립을 침범하게 됨. v_2를 쓸 수밖에 없음.

$v_1 + v_2$ 선형조합으로 표현가능 가정이면 $v_1 > v_2$ 이므로 v_1 선택 (가장 큰 σ_1)



가장 큰 (좋은) v를 선택해서 기저를 형성하면 "가장 가까운" 평면공간을 얻을수있음.

랭크-k 근사문제

우리가 찾아야 하는 것은 이 값을 가장 크게하는 k개의 열벡터가 아닌 직교하는 단위벡터 w_k이다. 고유분해의 성질로부터 오른쪽 기저벡터 중 가장 큰 k개의 특잇값에 대응하는 오른쪽 특이벡터가 우리가 찾는 기저벡터가 된다.

이 문제는 다음처럼 rank-k 근사문제의 형태로 만들수도 있다.

$$a_i^{\parallel w} = (a_i^T w_1) w_1 + (a_i^T w_2) w_2 + \cdots + (a_i^T w_k) w_k$$

$$= \begin{bmatrix} w_1 & w_2 & \cdots & w_k \end{bmatrix} \begin{bmatrix} a_i^T w_1 \\ a_i^T w_2 \\ \vdots \\ a_i^T w_k \end{bmatrix}$$

$$= \begin{bmatrix} w_1 & w_2 & \cdots & w_k \end{bmatrix} \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_k^T \end{bmatrix} a_i$$

$$= WW^T a_i$$

이러한 투영벡터를 모아놓은 행렬 A'는

$$A' = \begin{bmatrix} (a_1^{\parallel w})^T \\ (a_2^{\parallel w})^T \\ \vdots \\ (a_N^{\parallel w})^T \end{bmatrix} = \begin{bmatrix} a_1^T WW^T \\ a_2^T WW^T \\ \vdots \\ a_N^T WW^T \end{bmatrix} = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_N^T \end{bmatrix} WW^T$$

$$= AWW^T$$

따라서 이 문제는 원래 행렬 A에 랭크-k 행렬 WW^T를 곱해서 원래의 행렬 A와 가장 비슷한 행렬 A'를 만드는 문제와 같다.

$$\arg\min_{w_1, \cdots, w_k} \| A - AWW^T \|$$

Recap :

고유분해 $\boxed{A} = \boxed{V}\boxed{\Lambda}\boxed{V^T}$ (only 대칭행렬)

특이분해 $\boxed{A} = \boxed{U}\boxed{\Sigma}\boxed{V^T}$

$= (축소형) \boxed{U}\boxed{\Sigma}\boxed{V^T}$

양의 정부호 ⟺ 고유값 모두 양수

양의 준정부호 ⟺ 고유값 모두 0 이상

정방행렬 $A \in R^{N \times N}$

- λ는 복소수
- λ는 N개
- $AV = V\Lambda$
- $\sum_{i=1}^{N} \lambda_i = trA$
- $\prod_{i=1}^{N} \lambda_i = det A$

대칭행렬 $A = A^T$

- λ는 실수
- $V^T = V^{-1}$
- $A = V\Lambda V^T$
- $A = \sum_{i=1}^{N} \lambda_i v_i v_i^T$ rank-1

(음수인 고유값 포함)

공분산행렬 $A = X^T X$

- 모든 고유값이 0 이상 ($\lambda_i \geq 0$)

양의 정부호 (PD)

- X = full rank
- X^{-1} 존재
- $\lambda_i > 0$

고유분해 $Av = \lambda v$

특이분해 $Av = \sigma u$

"Rank-k 근사문제에 활용"

$$\boxed{\arg\min_{w_1, \cdots w_k} \| A - Aww^T \|}$$