선형대수학 클린업 3주차

[REVIEW]

선형변환(일종의 함수로 이해하기로 했다)의 관점에서, 선형방정식 Ax=b를 'x(input)에 A를 선형변환해서 b(output)가 나온 것'이라 해석했습니다. 이를 바탕으로 역행렬, 행렬식, 등의 개념을 이해했는데,,, det(A)=0이면 역행렬이 없다 다들 기억하시죠?? 선형부분공간을 비롯한 공간개념도 배우고, 독립(어느 한 벡터도 다른 벡터들의 일차결합으로 표현될 수 없다), 내적, 직교 등도 다뤘어요! 마지막으로 투영벡터(proj)와 선형회귀분석도 살펴봤었답니다!

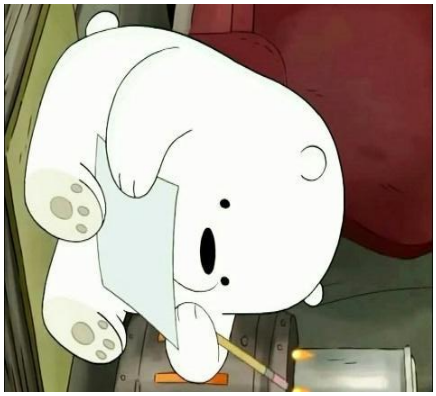
[TODAY's GOAL]

** 오늘은 응용을 뿌십니다 **

1. **차원의 저주와 차원축소**

2. **고유값 분해 (EVD)**
    1) 고유값과 고유벡터
    2) 대각화와 고유값 분해

3. **주성분 분석 (PCA)**
    1) 공분산행렬
    2) 주성분 분석



4. **특이값 분해 (SVD)**

5. **잠재요인분석 (LSA)**

6. **계층화분석법 (AHP)**

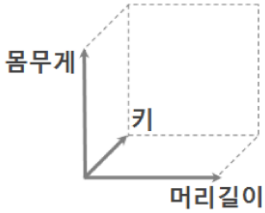
마지막까지 파이팅!

[Contents]

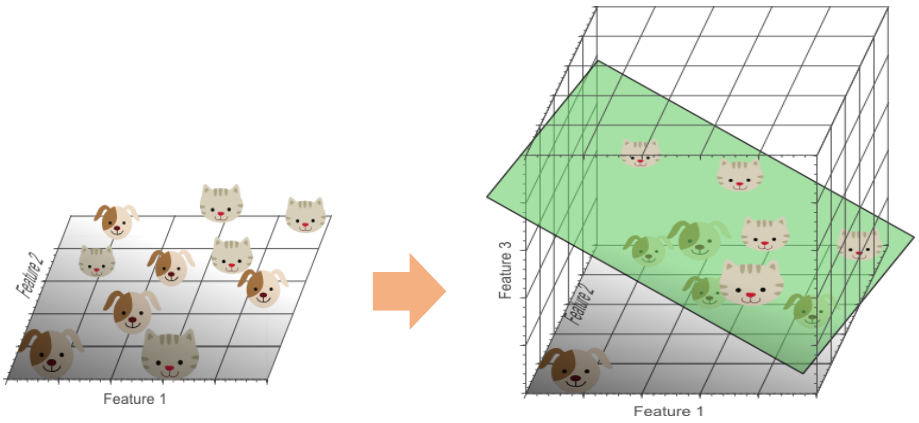
1. 차원의 저주와 차원축소

통계적 관점에서 차원은 변수의 개수로 이해할 수 있다. 변수의 개수가 축의 개수, 곧 차원을 의미하는데, 아래 예시를 보자. 키, 몸무게, 머리길이라는 세개의 변수를 가진 데이터는 원소가 3개인 벡터로, 3차원 공간안에 있다.

키	몸무게	머리길이
168	58	10
162	55	30
159	49	25
165	45	40



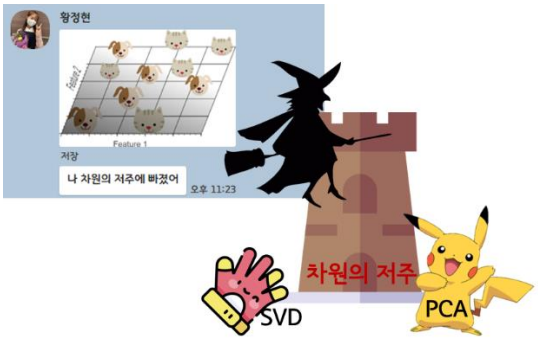
그렇다면! 고차원 데이터는 많은 정보를 가지고 있다는 뜻이므로, 항상 좋은 것일까?? 좋을... 수! 있다!! 아래의 예시는 차원의 증가가 분석에 긍정적으로 작용한 경우이다.



2차원(저차원)에서는 고양이과 강아지의 구분이 쉽지 않았지만,
3차원(고차원)으로 차원이 증가하면서 그 구분이 쉬워졌다!

하지만, 주의해야 할 점도 있다! 변수가 많아질수록 우린 하나하나의 거리에 집착하게 되고, 오버피팅(과적합) 문제가 일어나기 쉬워진다. 또한, 변수간의 관련성이 많은 경우에는 차원의 증가가 공간의 낭비로 이어질 수도 있다. 즉, 비효율적인 분석이 된다는 것!!! 이러한 배경에서 차원의 증가를 '차원의 저주'라고 하는 것이다.

차원의 저주는 주요변수만 일부 사용하는 변수선택(Feature Selection) 혹은 기존 변수를 조합해 새로운 변수를 만드는 변수추출(Feature Extraction)을 통해 해결하는데, 자세한 내용은 데마팀 1주차를 참고하시고~ 이번주 선대는 변수추출에 집중합니다.



👇여기서 잠깐!!

사실 고차원 데이터는 차원의 저주 문제에다가 처리속도도 오래 걸리고, 시각화의 인사이트를 도출하는 것도 힘들어서 가능하다면 차원축소를 고려하시는 것이 좋습니다 😊

2. 고유값 분해 (SVD)

1) 고유값과 고유벡터

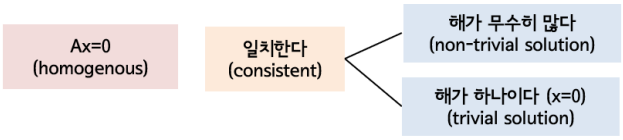
✔ 개념

n x n 행렬 A에 대해, Ax=λx를 만족하는 0이 아닌 해 x가 존재하면, 벡터 x를 행렬 A의 고유벡터(eigenvector), 상수 λ를 행렬A의 고유값(eigenvalue)라 한다.

> 행렬A가 nxn 정방행렬이고 벡터x가 0이 아닐 때,
> 아래의 식을 만족하는 λ를 고유값, 이때 대응하는 x를 고유벡터라 한다
> $Ax=\lambda x$ (x0)
> $Ax=\lambda Ix$ (x0)
> $Ax-\lambda Ix=0$
> $(A-\lambda I)x=0$

(A-λI)x=0를 homogeneous 방정식이라고 봐도 될까요???

당근당근!! 지난주 내용을 완전히 이해하셨군요??!!!!
$x \neq 0$, 즉 trivial solution을 갖지 않는다는 조건을 근거로 (A-λI)x=0의 해가 무수히 많다는 해석도 가능합니다!



여기서 잠깐!!

(A - λI)의 역행렬이 존재한다면?? (A - λI)x =0 의 식 양변에 $(A-\lambda I)^{-1}$곱하게 된다면???
x=0 (trivial solution)이 되어버리죠,,, ??!!! 고유벡터는 0벡터가 아니라는 전제에 어긋납니다ㅠㅠ
그래서!! (A - λI)의 역행렬이 존재하지 않는다! det(A - λI)= 0 이어야 합니다!

✔ 예시 : 계산하기

고유값과 고유벡터를 행렬로 계산할 때는 det(A – λI) =0 임과, Ax – λIx =0 식을 이용한다.

$A=\begin{bmatrix}2&1\\1&2\end{bmatrix}$ 일때 고유값λ와 고유벡터x를 구해보자

*고유값찾기 $\det(A-\lambda I)=0$이용

$$\begin{vmatrix}2-\lambda & 1\\ 1 & 2-\lambda\end{vmatrix}=(2-\lambda)^2-1=\lambda^2-4\lambda+3=0$$

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

$\lambda=3$ or 1

*$\lambda=3$일때고유벡터찾기 $(A-\lambda I)=0$이용

$$\begin{bmatrix}2-3 & 1\\ 1 & 2-3\end{bmatrix}=\begin{bmatrix}-1 & 1\\ 1 & -1\end{bmatrix}\Rightarrow RREF\begin{bmatrix}1&-1\\0&0\end{bmatrix}$$

$$x=\begin{bmatrix}x_1\\x_2\end{bmatrix}=x_2\begin{bmatrix}1\\1\end{bmatrix}$$

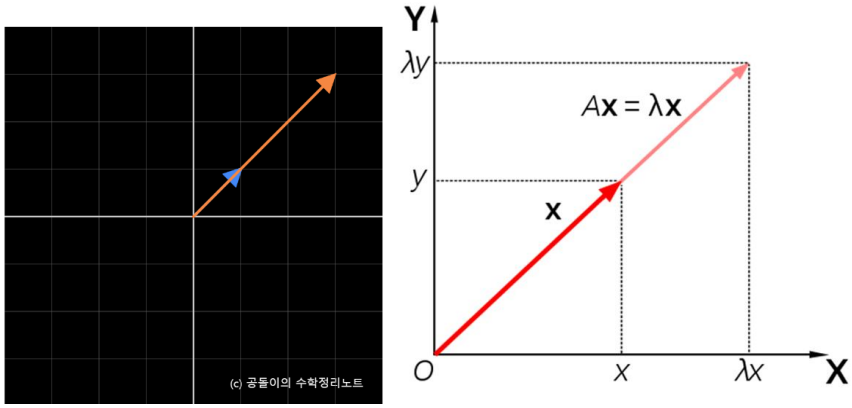
$\lambda=3$일때 고유벡터 $\begin{bmatrix}1\\1\end{bmatrix}$

✔ 고유값과 선형변환

위의 예시를 선형변환의 관점으로 해석해보자.

$$\begin{bmatrix}2&1\\1&2\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}=\begin{bmatrix}3\\3\end{bmatrix} \quad\Rightarrow\quad \begin{bmatrix}2&1\\1&2\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}=3\begin{bmatrix}1\\1\end{bmatrix}$$

지난 2주차 동안 경험한 선형변환은 벡터 x에 A라는 선형변환 이후 새로운 벡터b가 나오는 것이었다. 그런데 이번에는 선형변환 후에 벡터x의 상수배가 나왔다! (Ax=λx (x≠0)라는 식을 떠올리면 이해가 더 쉬울 것이다.) 결국 고유값과 고유벡터는 '선형 변환을 취해주었을 때, 크기만 바뀌고 방향은 바뀌지 않는 경우'라고도 해석할 수 있다.

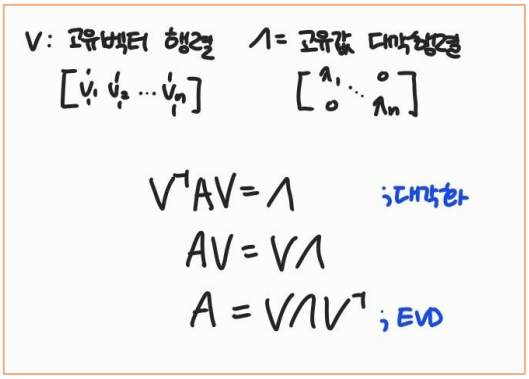


행렬A에 의한 선형변환 전과 후가 평행한 벡터가 고유벡터, 그때의 변화정도가 고유값이다.

2) 대각화와 고유값 분해 (EVD)

✔ 개념

n x n 행렬 A가 n개의 선형독립인 고유벡터가 있다면, 이를 바탕으로 대각행렬 D를 $P^{-1}AP$의 형태로 만들어 주는 것을 대각화라고 한다. 한편 이 식을 A를 기준으로 다시 정리해 $A = PDP^{-1}$ 의 형태로도 만들 수 있는데, 이것을 고유값 분해라고 한다.



✔ 예시

$$A=\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\lambda=3\text{일 때}\begin{bmatrix}1\\1\end{bmatrix},\lambda=1\text{일 때}\begin{bmatrix}1\\-1\end{bmatrix}$$

$$V=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\Lambda=\begin{bmatrix}3 & 0\\ 0 & 1\end{bmatrix}V^{-1}=\begin{bmatrix}0.5 & 0.5\\ 0.5 & -0.5\end{bmatrix}$$

$$A=V\Lambda V^{-1}=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}3 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}0.5 & 0.5\\ 0.5 & -0.5\end{bmatrix}$$

✔ 활용

대각화를 이용하면 det(A), A의 거듭제곱, 역행렬 등을 보다 쉽게 계산할 수 있다. 이래는 대각화를 이용해 행렬A의 거듭제곱($A^k = V\Lambda^k V^{-1}$)을 구하는 예시이다.

$$A^2=(V\Lambda V^{-1})^2=V\Lambda V^{-1}V\Lambda V^{-1}=V\Lambda^2 V^{-1}$$

3. 주성분 분석 (PCA)

1) 공분산행렬

✔ 수식적으로 이해하기

공분산을 행렬로 나타낸 것을 공분산행렬이라 하며, 이때 주각성분은 분산(cov(x,x))을 의미한다.

$$cov(X,Y) = \frac{1}{n-1}(\sum X_i Y_i - \overline{X}\,\overline{Y})$$
$$= \frac{1}{n-1}((X,Y) - \overline{X}\,\overline{Y})$$

$$X^T X = \begin{pmatrix} — & X_1 & — \\ — & X_2 & — \\ & \cdots & \\ — & X_d & — \end{pmatrix} \begin{pmatrix} | & | & & | \\ X_1 & X_2 & \cdots & X_d \\ | & | & & | \end{pmatrix}$$

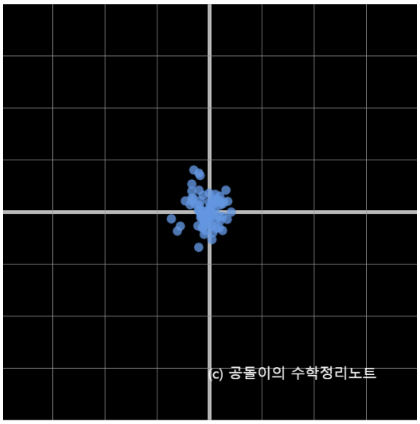
$$= \begin{pmatrix} dot(X_1,X_1) & dot(X_1,X_2) & \cdots & dot(X_1,X_d) \\ dot(X_2,X_1) & dot(X_2,X_2) & \cdots & dot(X_2,X_d) \\ \vdots & \vdots & \ddots & \vdots \\ dot(X_d,X_1) & dot(X_d,X_2) & \cdots & dot(X_d,X_d) \end{pmatrix}$$

✔ 선형변환으로 이해하기

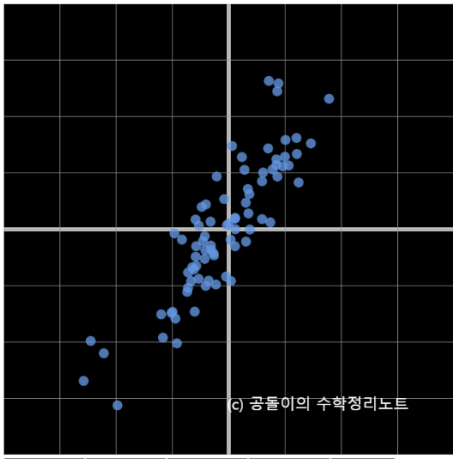
공분산 행렬은 각각의 변수의 퍼져 있는 정도인 분산과 변수들이 어떻게 함께 움직이는지를 설명해주는 공분산으로 이루어져 있다. 이 공분산 행렬을 통해 어떠한 데이터를 맵핑 한다면 분산과 공분산만큼 공간이 변화하게 되고 그 의미는 변수간 어떻게 연관이 되어 퍼져 있는지, 변수들이 어떤 식으로 분포되어 있는지를 나타내게 된다.

2차원 공간에 공분산 행렬을 적용하여 선형변환을 시킨 예시를 통해 이해하자.

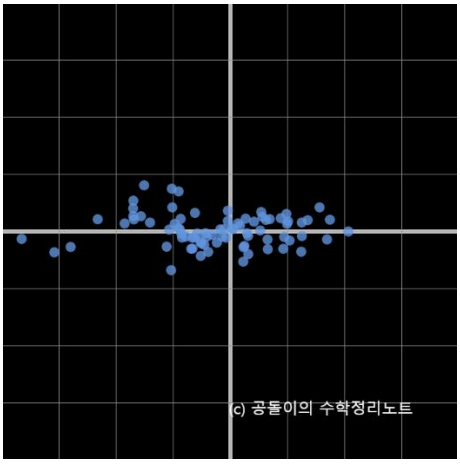
변수 x,y의 공분산 행렬이 아래와 같을 때, 주대각성분은 각각의 원소가 x축, y축으로 퍼진 정도를 의미한다. 축과 수직 또는 수평 방향으로 퍼짐을 표현한다. 한편, 이외의 성분은 x,y로 얼마만큼 함께 퍼지게 할 것인지를 뜻한다.



선형변환 전



$$\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$$



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

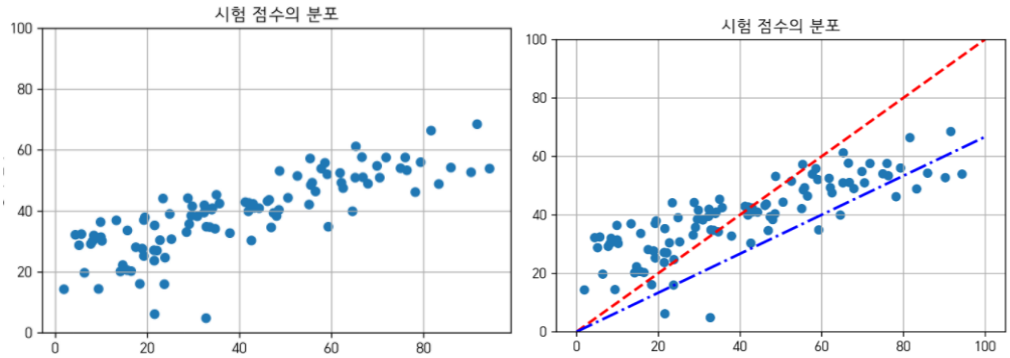
왼쪽 그래프는 x방향으로3, y방향으로는 4, x&y 방향으로는 2만큼 퍼지는 선형변환,
오른쪽 그래프는 x방향으로 5만큼만 퍼지는 선형변환의 결과이다.

2) 주성분분석

PCA는 차원을 축소하는 대표적인이다. n차원의 데이터가 있다면 그 데이터를 가장 잘 설명해주는 주성분(principal component)를 찾아내 그 주성분이 이루는 공간으로 데이터를 정사영시켜 차원을 축소해주는 것이다. 결국 PCA는 데이터의 정사영으로 차원을 낮출 때, 원래의 데이터 구조를 가장 잘 유지하는 주성분(PC)를 찾는 문제가 된다. 예시로 더 이해해보자.

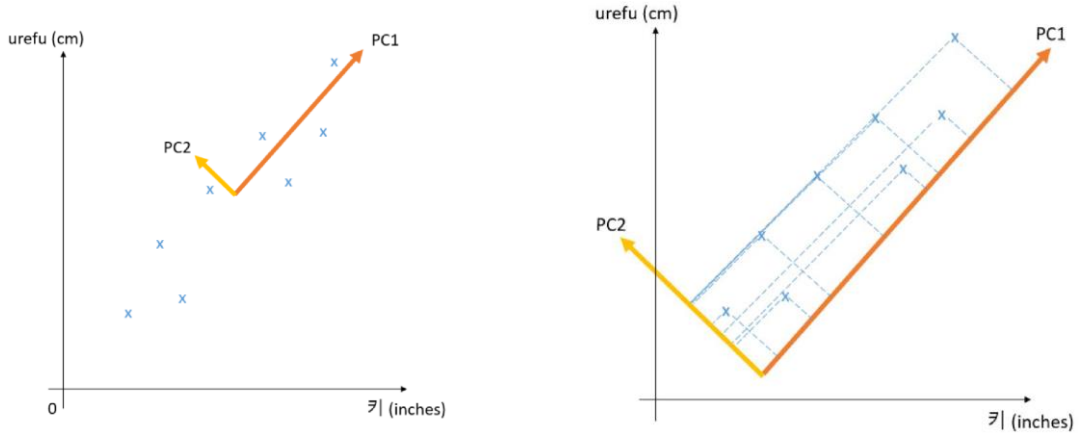
✔ 예시로 원리 이해하기

Sundae학교 학생들의 귀여움과 발표력을 100점 만점으로 평가했다. 귀여움 점수를 x, 발표력점수를 y라고 할 때, 결과를 시각화 하면 왼쪽 그림과 같다. 만약 여기서! 귀여움과 발표력을 고려한 '종합점수'를 만들고 싶다면 어떻게 해야할까? 귀여움과 발표력의 평균(5:5)을 취할까? 귀여운게 최고니까 6:4로 비중을 둘까? 오른쪽 그림의 빨간선은 5:5, 파란선은 6:4 비중을 둔 것이다. 그렇다면 이 중 어떤 방법이 더 효과적일까??



PCA는 데이터의 분포를 잘 반영하고 있는 성분 PC1과 PC2를 찾고(위의 예시의 종합점수),
이 중에서도 데이터를 더 잘 설명하는 PC를 선택(5:5 vs 6:4) 하는 것이다.

왼쪽 그림과 같이 데이터가 분포하고, 이 데이터를 잘 설명할 수 있는 성분으로 PC1과 PC2를 찾았다고 하자. 이들 중 데이터를 더 잘 설명하는 것은 누구일까? 기준이 있을까? 여기서 지난주의 정사영 개념이 등장한다.



각 PC에 대해 proj했을 때 그 분산이 더 큰 것, 즉 데이터에 대해 더 많이 퍼져 있는 pc를 고르는 원리이다.

✔ PC 구하는 방법

데이터를 설명하는 축인 PC(Principal component)는 공분산 행렬을 이용해 구한다. 정확히는 공분산행렬의 고유벡터를 pc로 삼고, 이때 고유값의 크기에 따라 그 중요성을 확인하는 것이다.

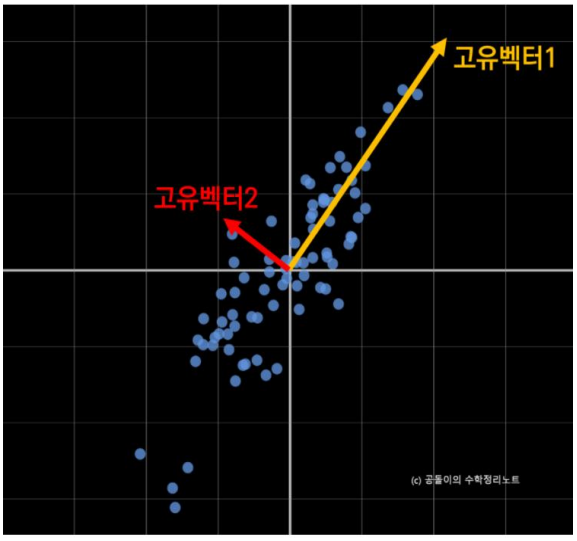
앞서 살펴본 것처럼, 공분산행렬은 데이터의 분포를 설명하는 것으로, 공분산행렬에 의한 맵핑을 통해 그 데이터의 분포 경향을 살펴볼 수 있다.

그렇다면 왜 하필 고유벡터를 이용할까? 고유벡터는 선형변환 후에도 크기만 변할 뿐 기울기가 변하지 않는다 했다. 때문에 고유벡터는 선형변환의 고정된 축(axis)으로도 볼 수 있는 것이다. 한편 고유값은 고유벡터 방향으로 변한 크기를 설명하는데, 이것을 PC가 데이터에 대해 얼마나 퍼져 있는지로 해석하면, 고유값을 기준으로 PC의 중요도를 확인하는 것이 이해될 것이다.

아래 예시는 공분산행렬 $\begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}$ 의 고유값과 고유벡터이다.

```
> eigen(x)
eigen() decomposition
$values
[1] 5.561553 1.438447

$vectors
          [,1]       [,2]
[1,] 0.6154122 -0.7882054
[2,] 0.7882054  0.6154122
```



공분산행렬의 고유값과 고유벡터를 구한 결과는 왼쪽과 같다.

고유벡터 1의 고유값은 5.56으로 고유벡터2(1.44)보다 많은 데이터에 대해 분포한다.

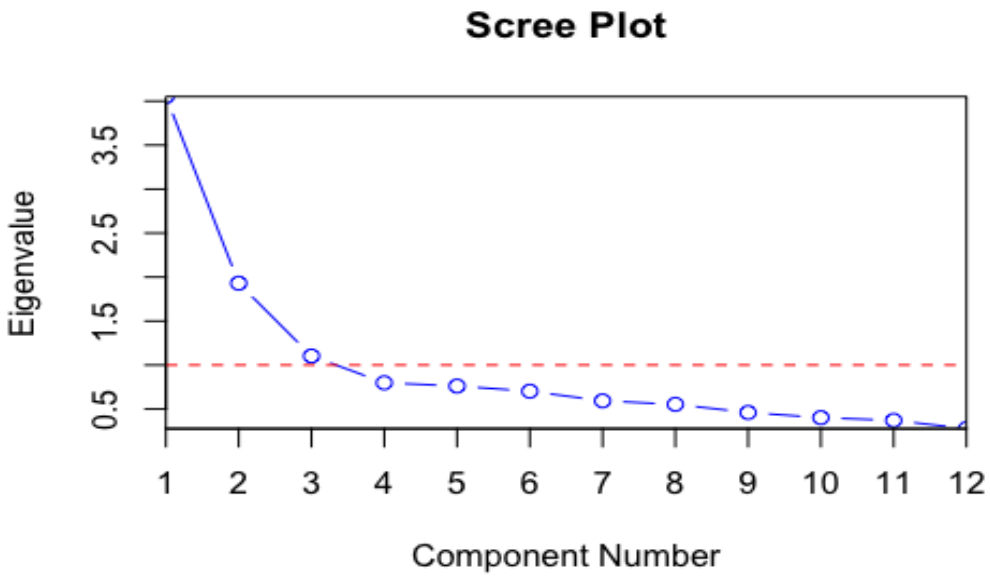
✔ PC 개수 정하는 방법

예시를 통해서는 2차원 데이터를 설명하는 주성분 한 개를 고르는 것을 살펴봤지만, 고차원데이터의 경우는 몇 번째 주성분까지 추출해야 할지 결정해야한다.

논리적으로는 PCA결과의 summary를 근거로 주성분의 개수를 정한다. 결과의 누적비율에 초점을 맞춰, 일반적으로 이것이 90%정도 되는 주성분을 선택한다. 아래 결과를 보면, PC3의 누적비율은 99.5%로, PC3까지만 사용해도 데이터의 99.5%의 변동을 설명할 수 있다. 이런 경우라면 4차원 데이터를 3차원으로 축소할 수 있다.

```
> summary(pca_dt)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

한편, scree plot의 엘보 포인트를 이용하는 방법도 있다. 아래의 경우 세번째 eigenvalue를 기준으로 기울기가 갑자기 변화하는 현상이 나타나는데, 이 지점을 엘보 포인트라고 한다. 이를 근거로 위의 데이터 pc를 3개, 즉 데이터를 3차원까지 축소시킬 것을 생각해 볼 수 있다. Scree Plot은 다른 분석의 기준을 잡는데도 주로 사용되는 방법이므로 이 기회에 딱 눈에 익혀두자!

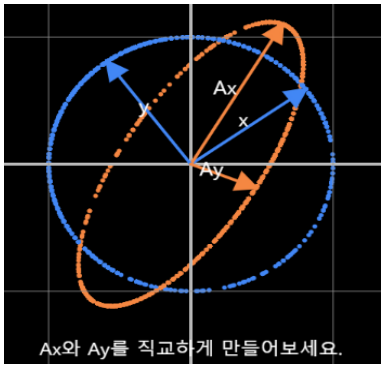


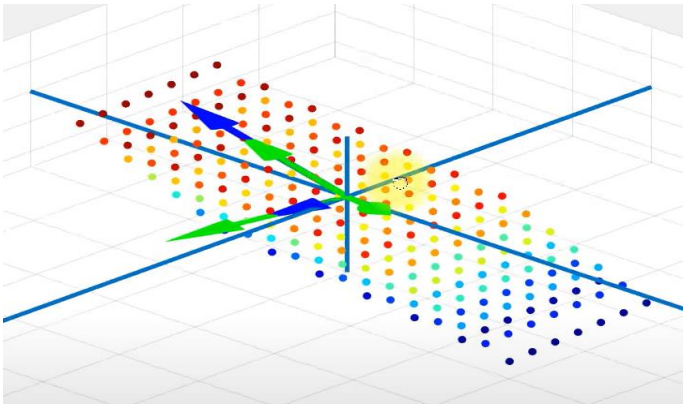
4. 특이값 분해 (SVD)

앞서 살펴본 고유값분해(EVD)는 nxn 정방행렬A에 대해서만 가능했다. 한편, nxm 크기의 일반적인 행렬A에 대해 고유분해 비슷한 처리를 해주는 것도 가능한데, 이것을 특이값 분해(특이분해, svd) 라고 한다.

✔ 기하학적으로 이해하기

직교하는 벡터 집합에 대하여, 선형변환 후에 그 크기는 변하지만 여전히 직교하는 직교집합은 무엇인가? 벡터 x, y가 직교일 때, A라는 선형변환 후의 Ax와 Ay도 직교할까? 이것 질문에 답을 찾는 것이 바로 SVD이다.





직교하는 벡터(v, 녹색)를 선형변형 한 뒤에도 여전히 직교하는 벡터(u, 청색)

✔ 수식적으로 이해하기

직교하는 벡터 집합(V)에 대하여, 선형변환 후에 그 크기(Σ)는 변하지만 여전히 직교하는 직교집합(U)을 식을 세워 정리하고, 이해해보자.

$$V \xrightarrow{T(A)} U$$

(직교) (직교) + 크기변화(Σ)

$$A\begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

$$AV = U\Sigma$$

$$AVV^{-1} = U\Sigma V^{-1}$$

$$A = U\Sigma V^{-1}$$

일단 여기까지!

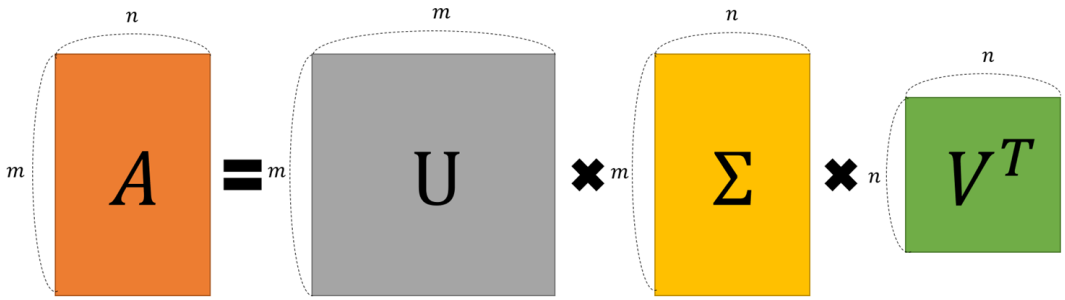
✔ 개념

A : m x n 행렬

Vt : n x n 직교행렬 V의 transpose, 선형변환 전

U : m x m 직교행렬 U, 선형변환 후

Σ : m x n 대각행렬, 크기 변화



💡 여기서 직교행렬 (orthogonal matrix)란?

$VV^T = V^T V = I$를 만족하는 행렬, 즉 $V^T = V^{-1}$인 행렬을 직교행렬이라고 해요!

$$permutation\ \ matrix:$$

$$QQ^T=I \rightarrow \begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \quad \cdots(7)$$

$$Q \qquad Q^T=Q^{-1} \qquad I$$

$A=\begin{bmatrix}\frac{2}{3}&-\frac{2}{3}&\frac{1}{3}\\\frac{2}{3}&\frac{1}{3}&-\frac{2}{3}\\\frac{1}{3}&\frac{2}{3}&\frac{2}{3}\end{bmatrix}$ 라 하면 $A^T=\begin{bmatrix}\frac{2}{3}&\frac{2}{3}&\frac{1}{3}\\-\frac{2}{3}&\frac{1}{3}&\frac{2}{3}\\\frac{1}{3}&-\frac{2}{3}&\frac{2}{3}\end{bmatrix}$ 이고 $A^TA=\begin{bmatrix}\frac{2}{3}&-\frac{2}{3}&\frac{1}{3}\\\frac{2}{3}&\frac{1}{3}&-\frac{2}{3}\\\frac{1}{3}&\frac{2}{3}&\frac{2}{3}\end{bmatrix}\begin{bmatrix}\frac{2}{3}&\frac{2}{3}&\frac{1}{3}\\-\frac{2}{3}&\frac{1}{3}&\frac{2}{3}\\\frac{1}{3}&-\frac{2}{3}&\frac{2}{3}\end{bmatrix}=I_3$

🌞 그러면 앞서 살펴본 $A = U\Sigma V^{-1}$ 이랑 방금 본 $A = U\Sigma V^T$ 가 같은 건가요?

딩동댕동!! 정답입니다~!! 특이값 분해(SVD)는 $A = U\Sigma V^T$ 로 분해하는 것이 정의라는 것을 딱 기억해주세요!

🌞 대각행렬 (diagonal matrix)인데 어떻게 m x n 이죠?

대각행렬은 주대각성분을 제외한 원소가 모두 0인 행렬을 말해요! 이때 대각성분은 큰 수부터 작은 수로 배열되어야 한답니다~ 여러분들이 알고 있는 nxn 대각행렬에 가로 또는 세로로 0벡터를 붙이는 느낌으로 이해하시면 편합니다!

m > n

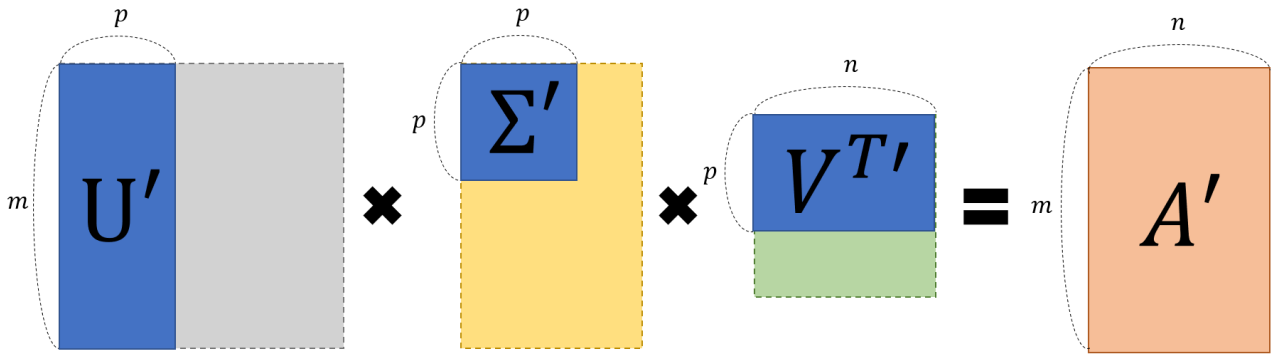
$$\begin{pmatrix}\sigma_1&0&\cdots&0\\0&\sigma_2&\cdots&0\\&&\ddots&\\0&0&\cdots&\sigma_n\\0&0&\cdots&0\\\vdots&\vdots&\vdots&\vdots\\0&0&\cdots&0\end{pmatrix}$$

m<n

$$\begin{pmatrix}\sigma_1&0&\cdots&0&0&\cdots&0\\0&\sigma_2&\cdots&0&0&\cdots&0\\&&\ddots&&&&\\0&0&\cdots&\sigma_m&0&\cdots&0\end{pmatrix}$$

✔ 활용

SVD를 이용해 A라는 여러 개의 행렬로 분해해서 생각할 수 있는데, 사실 분해되는 과정이나 그 결과보다는 분해된 행렬을 다시 조합하는 과정이 주로 이용된다.



✔ 이미지에 활용하기

이미지를 행렬로 저장한 뒤 SVD로 처리하면, 더 적은 정보와 용량을 가진 이미지(저화질)로 표현하는 것이 가능하다.

(1) 정사각 2530 x 2530 사진 화질을 조정해보자!

(2) 직사각 952 x 607 사진의 화질을 조정해보자!

사진 원본, 사진파일의 행렬 표현을 위해서는 흑백 사진으로 설정을 변경한다.
SVD를 통해 각각 100개, 50개, 10개의 singular value를 이용해 복원해보자!

오른쪽으로 갈수록, 더 적은 singular value로 복원할수록 정보량이 적어져 알아보기 어려운 형체가 된다.
이미지 분석에 있어 이러한 기법을 사용하면, 더 적은 정보로, 더 적은 용량으로, 더 빠른 성능으로 분석이 가능하다.

5. 잠재요인분석 (LSA)

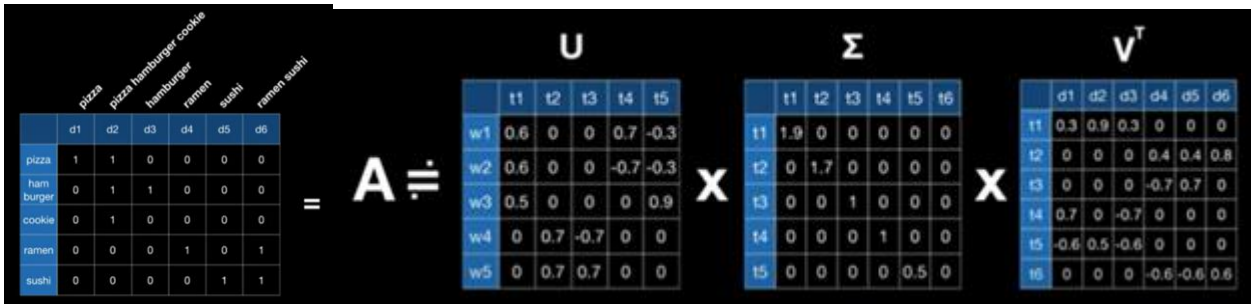
지금까지 특이값분해에서 mxn행렬 A를 $A = U\Sigma V^T$ 로 나타내고, 이후 내림차순으로 정리된 Σ행렬의 대각원소(특이값) 가운데 상위 p개만 골라 새롭게 A'를 표현하는 것까지 이미지파일 예시와 함께 살펴보았다. 이와 비슷한 논리로 진행되는 자연어처리기법인 LSA도 살펴보자.

잠재요인분석이란 자연어 처리에서 문서집합의 추상적인 주제를 발견하기 위해 사용하는 통계적 모델(토픽모델링) 중 하나이다. 단어의 빈도수만을 고려하던 기존의 분석에 의미를 고려하도록 나온 대안이 LSA이다. LSA는 자연어 기반 입력데이터에 특이값 분해를 수행해 데이터의 차원을 줄여 계산 효율성을 높이면서 잠재적인 의미를 이끌어내는 방식이며 예시를 통해 가볍게 이해하고 넘어가자! 자세한 내용은 딥러닝 클린업을 참고~

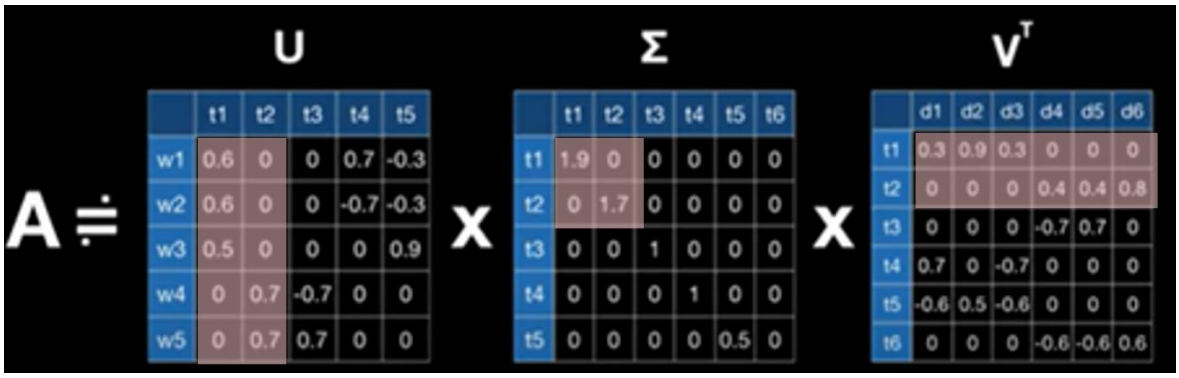
다음과 같은 메뉴판이 있다고 할 때, 이것을 기반으로 LSA를 해보자.

(1) pizza (2) pizza hamburger cookie (3) hamburger
(4) ramen (5)sushi (6) ramen sushi

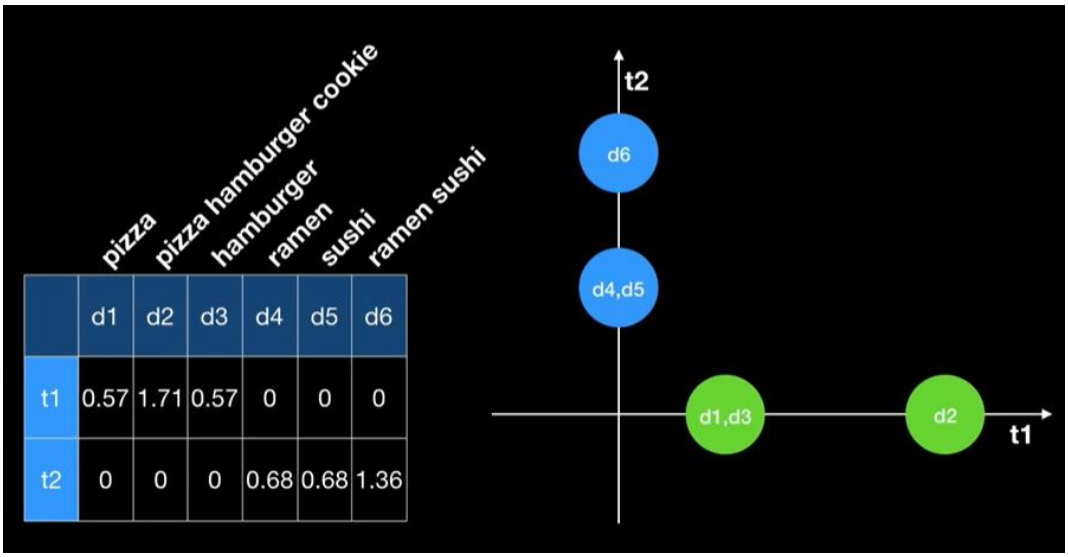
단어를 행(row)로, 메뉴판의 문장을 열(col)로 하는 단어-문서행렬A를 아래와 같이 만들고, 특이값분해(SVD)를 진행했다. 이때, U는 토픽을 위한 '단어'행렬, t(V)는 토픽을 위한 '문장'행렬, 그리고 Σ는 토픽의 강도를 의미한다.



이후 상위 2개의 특이값만 남겨 새로운 단어-문서행렬 (A')를 만들 수 있다.

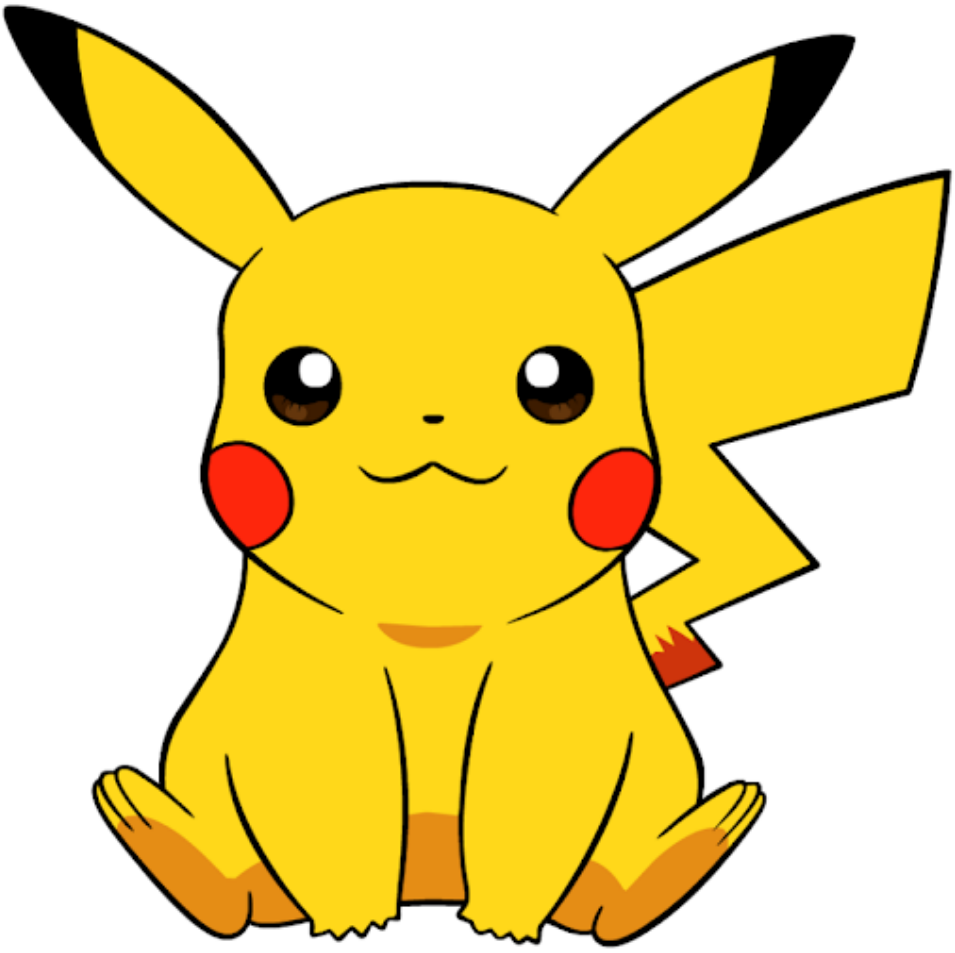


아래 그림은 2개의 특이값만 택해 문장(메뉴)관련 행렬을 새롭게 정리(ΣV^T)하고 시각화 한 결과이다.



d1,d2,d3는 t1으로 묶여 '양식'을, d4,d5,d6는 t2로 묶여 '일식'을 의미함을 해석할 수 있다.

차원의 저주가 끝났습니다!!!



쌍대분석을 기반으로 한 AHP를 다루고 싶다는 욕심 때문에

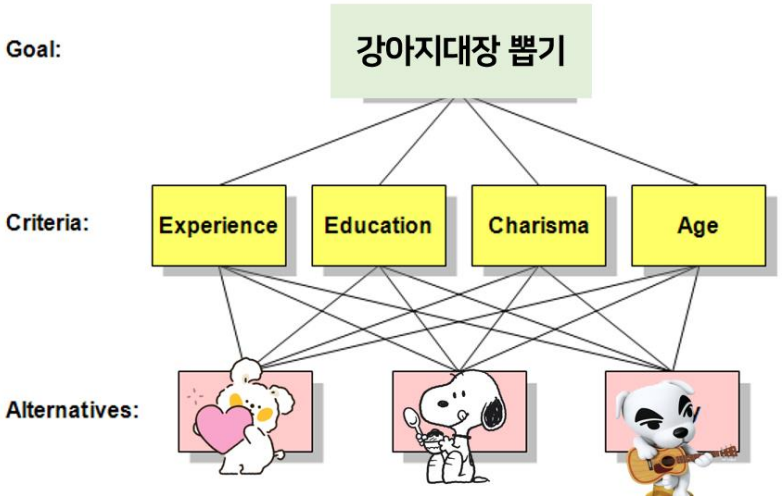
차원축소 파티인 이번 주제와 조금 거리가 있는 내용이 부록처럼이어집니다...ㅎㅎ

---

6. 계층화분석법 (AHP)

AHP는 의사결정문제가 다수의 평가기준으로 이루어져 있을 때, 평가기준을 계층화한 뒤 이에 따라 중요도를 정해가는 다기준 의사결정기법이다. 인간의 의사결정이 계층적이고 상대적인 원칙을 따른다는 관점에서 시작되었으며, 쌍대분석의 반복을 통해 분석이 이뤄진다. 사실, AHP를 '행렬을 이용한 단계적 가중치 산정법' 정도로 쉽게 이해하고 넘어가도 좋다.

본 내용은 쌍대비교 기초 분석을 이해하려는 목적으로 교안에 담았으며, 예시를 바탕으로 '이런 식으로 분석할 수도 있구나~' 정도로만 보기로 하자.



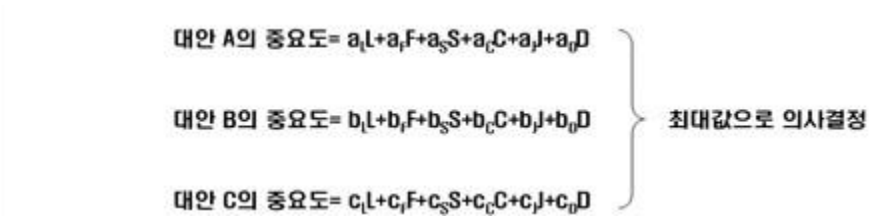
앙꼬, 스누피, K.K. 중에 강아지 대장을 뽑으려고 한다. 누가 대장에 더 어울릴까 고민하다가, 다음의 4가지를 기준으로 삼기로 했다. 세 후보에 대해 4가지 요소를 따져야 한다는 막막함이 밀려올 때, 이런 생각이 들어야한다!! 각 기준별 두 후보씩 비교하면 되는거 아니야? 예를 들어 '앙꼬가 스누피보다 카리스마있나?', '스누피가 kk보다 카리스마있나?','kk가 앙꼬보다 카리스마있나'하는 식으로 말이다. 이런 식으로 factor 두개를 상호 비교하는 것을 쌍대비교라고 한다. 쌍대비교를 하면 결정이 간단해질 뿐만 아니라, A와 B의 비교와 동시에 B와 A의 비교가 가능하다.

카리스마	앙꼬	스누피	KK
앙꼬	1	5	9
스누피	1/5	1	4
KK	1/9	1/4	1

3번의 비교를 통해 위와 같은 행렬을 얻었다면, 이제 셋중에 누가 더 카리스마가 있는지 정리할 차례다. 이때 상호중요도는 (1) 두 행렬을 곱하여, (2) 행간을 더한 행렬을 구한 다음, (3)전체 합에서 각 행의 비율을 계산하는 방식이다.

1	5	9		1	5	9		53.24		0.75
1/5	1	4	*	1/5	1	4	=	13.64	=>	0.19
1/9	1/4	1		1/9	1/4	1		4.31		0.06

이런 방식으로 각 기준(factor)별 비교와 factor에 대한 중요도도 정리하면 최종 의사결정이 가능하다.



Criterion	Priority vs. Goal	Alternative	A		B		C
Experience	0.547	Tom	0.217	x	0.547	=	0.119
		Dick	0.717	x	0.547	=	0.392
		Harry	0.066	x	0.547	=	0.036
			1.000				0.547
Education	0.127	Tom	0.188	x	0.127	=	0.024
		Dick	0.081	x	0.127	=	0.010
		Harry	0.731	x	0.127	=	0.093
			1.000				0.127
Charisma	0.270	Tom	0.743	x	0.270	=	0.201
		Dick	0.194	x	0.270	=	0.052
		Harry	0.063	x	0.270	=	0.017
			1.000				0.270
Age	0.056	Tom	0.265	x	0.056	=	0.015
		Dick	0.672	x	0.056	=	0.038
		Harry	0.063	x	0.056	=	0.004
			1.000				0.056

	Priority with Respect to				
Candidate	Experience	Education	Charisma	Age	Goal
앙꼬	0.119	0.024	0.201	0.015	0.358
스누피	0.392	0.010	0.052	0.038	0.492
KK	0.036	0.093	0.017	0.004	0.149
Totals:	0.547	0.127	0.270	0.056	1.000

이렇게 하면 최종 점수가 제일 높은 스누피로 결정!!

물론, 당연하지만, 과정을 직접 계산하지 않아도 구현해주는 코드! 있습니다! :)

[CODE]

오늘 다룬 내용 관련된 코드실현은 교안에 문제설명을 다루지 않고, html 정리본으로 공유하겠습니다!

[마치며]

누군가에겐 정말 길고 힘들었던, 누군가에게는 아쉬움이 많이 남았던, 또 누군가에게는 새로운 설렘이 시작되었던 3주간의 클린업이 종료되었습니다. 선형대수학팀 클린업에서는 '선형방정식과 선형결합'을 중심으로 소거법, 아핀변환, 역행렬, 공간압축, 정사영, 차원축소 등의 개념을 다뤘습니다. 부족한게 많았을텐데, 정말 감사합니다.

전공 필수로 행렬대수학 강의를 수강하는 내내 '이런걸 도대체 어디에 쓰길래 배우는거지?' 생각했습니다.
행대수 수업을 마무리하면서는 '정말 1도 모르겠는데 그냥 재수강만 안했으면 좋겠다.' 생각했습니다.
회귀분석에서 행렬 개념이 종종 사용되는걸 보면서는 '아,, 이런 느낌으로 쓰이는거라고,,,?' 생각했습니다.

그리고 지난학기 피셋 활동을 하면서는 '이래서 선대가 근본이구나' 느꼈습니다.

이번 학기 클린업을 통해서 여러분들도 부디 저와 비슷한 감동을 받고, 선대의 쓸모를 몸소 느끼길 바랬는데, 어떠셨나요??
저는 저의 빈틈을 제대로 느끼고, 반성하고, 보완하는 3주가 되었네요,, (제가 참 게으르다는 것도 알게 되었답니다^^)

여러분의 피셋 활동은 아직 끝나지 않았습니다!!!!!!

아쉬운건 보완하고, 잘하는건 자랑해가면서 앞으로의 활동도 보람차게! 웃으면서! 이어갑시다!!
이번주 피피티와 클린업도 파이팅 하시고!!! 다가오는 퀴즈와 중간고사도 잘 뿌시자고요!!

제가 준비한 내용은 여기까지 입니다. 고맙습니다.