

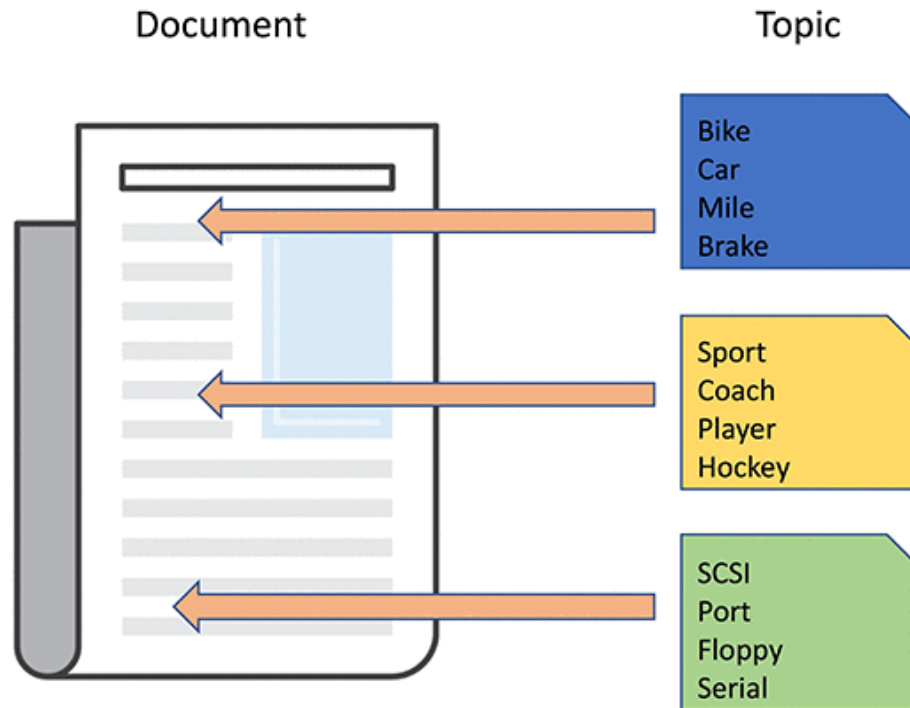
# Topic Modeling

토픽 모델링

권예진(응용통계학과)



# *What is Topic Modeling?*



## 토픽 모델링(Topic Modeling)이란?

기계 학습 및 자연어 처리 분야에서 토픽이라는 문서 집합의 추상적인 주제를 발견하기 위한 통계적 모델 중 하나로, 텍스트 본문의 숨겨진 의미 구조를 발견하기 위해 사용되는 텍스트 마이닝 기법

---

# 잠재 의미 분석 (Latent Semantic Analysis, LSA)

! co-occurrence 정보를 이용

☞ 단어의 '형태(morphology)'가 아닌 의미(semantic)'를 이용

⚙ 절단된 특이값 분해(Singular Value Decomposition, SVD)를 이용

---

# !! 특이값 분해(Singular Value Decomposition, SVD)

A가  $m \times n$  행렬일 때,  
다음과 같이 3개의 행렬의 곱으로 분해(decomposition)!

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

여기서 각 3개의 행렬은 다음과 같은 조건을 만족합니다.

$U$  :  $m \times m$  직교행렬 ( $AA^T = U(\Sigma\Sigma^T)U^T$ )

$V$  :  $n \times n$  직교행렬 ( $A^T A = V(\Sigma^T \Sigma)V^T$ )

$\Sigma$  :  $m \times n$  직사각 대각행렬

	문서1	문서2	문서3	문서4	문서5	문서6
cosmonaut	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

After the SVD ▼

T

	차원1	차원2	차원3	차원4	차원5
cosmonaut	-0.44	-0.30	0.57	0.58	0.25
astronaut	-0.13	-0.33	-0.59	0.00	0.73
moon	-0.48	-0.51	-0.37	0.00	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-0.26	0.65	-0.41	0.58	-0.09

S

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

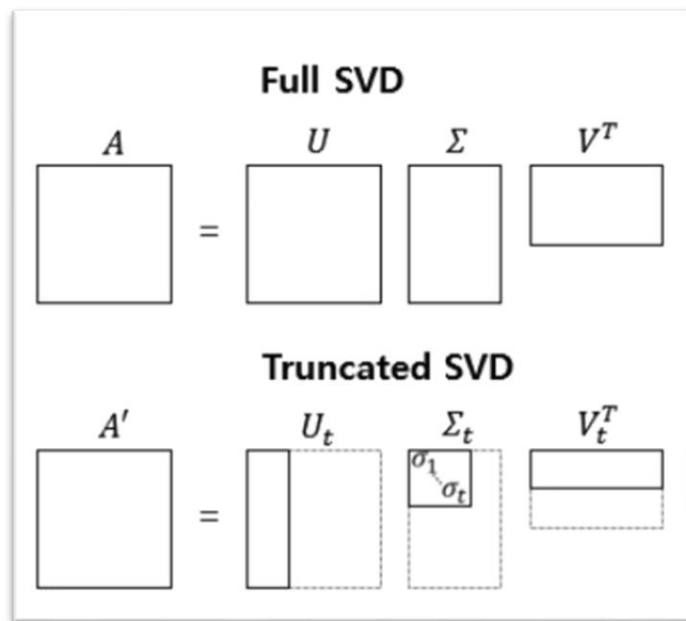
D<sup>T</sup>

	문서1	문서2	문서3	문서4	문서5	문서6
차원1	-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
차원2	-0.29	-0.53	-0.19	0.63	0.22	0.41
차원3	0.28	-0.75	0.45	-0.20	0.12	-0.33
차원4	0.00	0.00	0.58	0.00	-0.58	0.58
차원5	-0.53	0.29	0.63	0.19	0.41	-0.22

# 절단된 SVD(Truncated SVD)를 이용하는 LSA

앞의 SVD를 풀 SVD(full SVD)라고 하는데,

LSA의 경우 풀 SVD에서 나온 3개의 행렬에서 일부 벡터들을 삭제시킨 **절단된 SVD(truncated SVD)**를 사용



$$A_{t \times d} = U_{t \times n} \Sigma_{n \times n} V^T_{n \times d}$$

t(단어 개수), d(문서 개수), n(토픽의 수를 반영한 하이퍼 파라미터 값)

◀ 절단된 SVD는 대각 행렬  $\Sigma$ 의 대각 원소의 값 중에서 상위값 t개만 남긴다.

If  $n \uparrow$ ,  
기존의 행렬 A로부터 다양한 의미를 가져갈 수 있다.

If  $n \downarrow$ ,  
노이즈를 제거하는 효과(설명력이 낮은 정보를 삭제하고 설명력이 높은 정보를 남기는 효과),  
계산 비용이 낮아짐.

LSA는 기본적으로 DTM이나 TF-IDF 행렬에 절단된 SVD(truncated SVD)를 사용하여 차원을 축소시키고, 단어들의 잠재적인 의미를 끌어낸다는 아이디어를 갖고 있다.

# ✂ LSA의 장단점(Pros and Cons of LSA)

👍 LSA는 쉽고 빠르게 구현이 가능

👍 단어의 잠재적인 의미를 이끌어낼 수 있어 문서의 유사도 계산 등에서 좋은 성능을 보여줌

🔧 SVD의 특성상 이미 계산된 LSA에 새로운 데이터를 추가하여 계산하려고 하면 보통 처음부터 다시 계산해야 함. -> 즉, 새로운 정보에 대해 업데이트가 어렵다.



*LSA 실습*

---

# 잠재 디리클레 할당 (Latent Dirichlet Allocation, LDA)

! LDA는 각 문서의 토픽 분포와 각 토픽 내의 단어 분포를 추정한다.

---



# ✓ LDA의 가정

1) 문서에 사용할 단어의 개수  $N$ 을 정한다.

ex) 5개의 단어를 정했다.

2) 문서에 사용할 토픽의 혼합을 결정한다.

ex) 토픽이 2개라고 하였을 때 강아지 토픽을 60%, 과일 토픽을 40%와 같이 선택할 수 있다.

3) 문서에 사용할 각 단어를 (아래와 같이) 정한다.

3-1) 토픽 분포에서 토픽  $T$ 를 확률적으로 고른다.

ex) 60% 확률로 강아지 토픽을 선택하고, 40% 확률로 과일 토픽을 선택할 수 있다.

3-2) 선택한 토픽  $T$ 에서 단어의 출현 확률 분포에 기반해 문서에 사용할 단어를 고른다.

ex) 강아지 토픽을 선택했다면, 33% 확률로 강아지란 단어를 선택할 수 있다. 이제 3) 을 반복하면서 문서를 완성한다.

이러한 과정을 통해 문서가 작성되었다는 가정 하에 LDA는 토픽을 뽑아내기 위하여 위 과정을 역으로 추적하는 **역공학(reverse engineering)**을 수행한다.

# !LDA의 수행 과정

1) 사용자는 알고리즘에게 토픽의 개수  $k$ 를 알려준다.

2) 모든 단어를  $k$ 개 중 하나의 토픽에 할당한다.

이제 각 문서는 토픽을 가지며, 토픽은 단어 분포를 가지는 상태이다. 물론 랜덤으로 할당하였기 때문에 사실 전부 틀린 상태이다. 만약 한 단어가 한 문서에서 2회 이상 등장하였다면, 각 단어는 서로 다른 토픽에 할당되었을 수도 있다.

3) 이제 모든 문서의 모든 단어에 대해서 아래의 사항을 반복 진행한다.(iterative)

어떤 문서의 각 단어  $w$ 는 자신은 잘못된 토픽에 할당되어 있지만, 다른 단어들은 전부 올바른 토픽에 할당되어 있는 상태라고 가정한다. 이에 따라 단어  $w$ 는 아래의 두 가지 기준에 따라서 토픽이 재할당된다. 이를 반복하면, 모든 할당이 완료된 수렴 상태가 된다.

- $p(\text{topic } t \mid \text{document } d)$  : 문서  $d$ 의 단어들 중 토픽  $t$ 에 해당하는 단어의 비율
- $p(\text{word } w \mid \text{topic } t)$  : 단어  $w$ 를 갖고 있는 모든 문서들 중 토픽  $t$ 가 할당된 비율

doc1

word	apple	banana	apple	dog	dog
topic	B	B	???	A	A

doc2

word	cute	book	king	apple	apple
topic	B	B	B	B	B



doc1

word	apple	banana	apple	dog	dog
topic	B	B	???	A	A

doc2

word	cute	book	king	apple	apple
topic	B	B	B	B	B



doc1


word	apple	banana	apple	dog	dog
topic	B	B	???	A	A

doc2

word	cute	book	king	apple	apple
topic	B	B	B	B	B

# LDA와 LSA의 차이

 LSA : DTM을 차원 축소 하여 축소 차원에서 근접 단어들을 토픽으로 묶는다.

 LDA : 단어가 특정 토픽에 존재할 확률과 문서에 특정 토픽이 존재할 확률을 결합확률로 추정하여 토픽을 추출한다.



*LDA 실습*