

# 토픽모델링 (Topic modeling)

LSA, pLSA, LDA

Fininsight

데이터 분석가 김현진

# 토픽모델링

# 토픽 모델링(Topic Modeling)



- 구조화되지 않은 방대한 문헌집단에서 주제를 찾아내기 위한 알고리즘
- 맥락과 관련된 단서들을 이용하여 의미를 가진 단어들을 클러스터링하여 주제를 추론함

기계 학습 및 자연언어 처리 분야에서 토픽 모델(Topic model)이란 문서 집합의 추상적인 "주제"를 발견하기 위한 통계적 모델 중 하나로, 텍스트 본문의 숨겨진 의미구조를 발견하기 위해 사용되는 텍스트 마이닝 기법 중 하나이다.

-wikipedia-

- 토픽모델링은 ‘문서는 여러 주제로 구성되어 있고, 각 주제는 단어 집합으로 구성된다.’는 가정에서 시작한다.

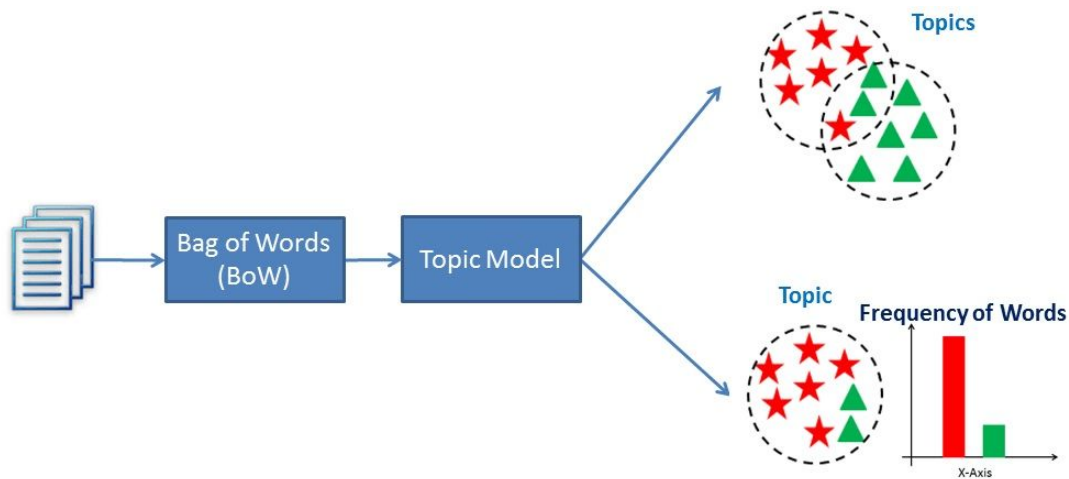
ex) 문서에 ‘멍멍’, ‘뽀다귀’, ‘야옹’, ‘생선’ 이라는 단어가 자주 등장했다면?

강아지

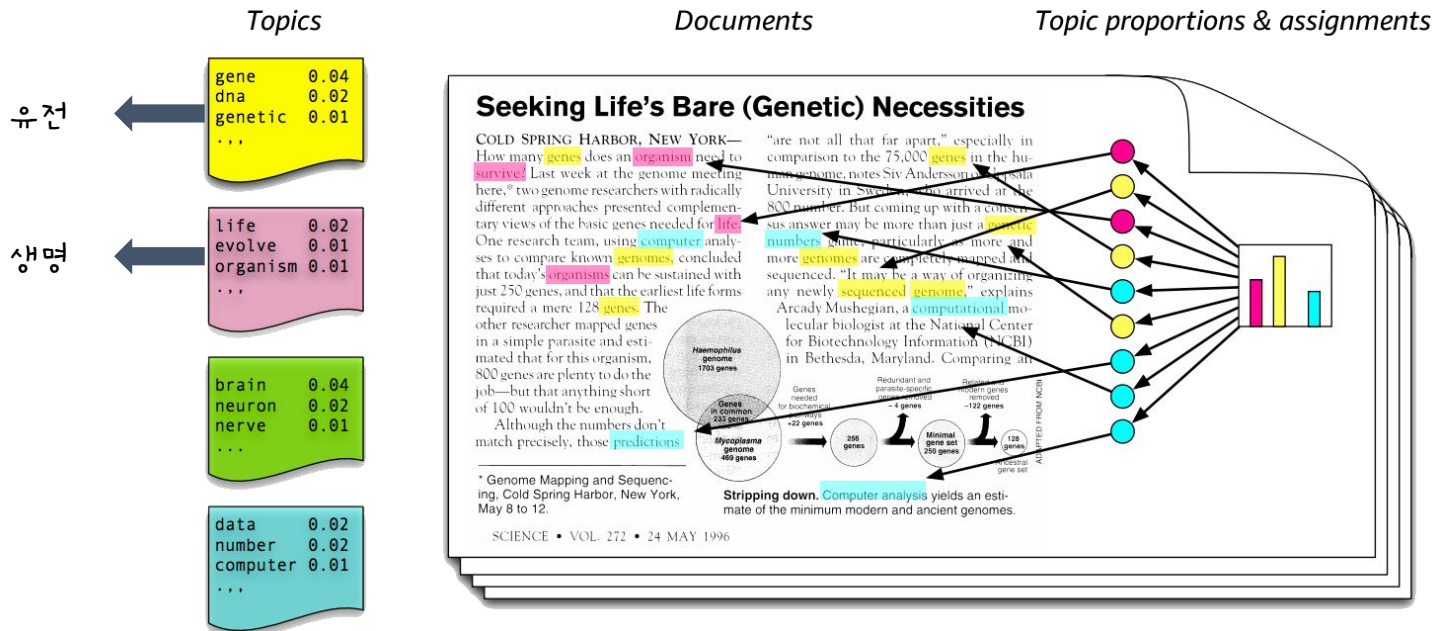
고양이

# 토픽 모델링(Topic Modeling)

- 텍스트에 숨겨져 있는 주제들을 찾아내기 위한 통계추론에 기반한 분석 기법.
- 개별 문서는 다수의 주제, 혹은 토픽으로 구성된 혼합체로 간주하고,  
각 토픽을 추출된 키워드의 분포로 나타냄으로써 텍스트 내의 구조를 파악할 수 있다.



# 토픽 모델링(Topic Modeling)



# 토픽 모델링 활용(1) - 이슈분석



- 뉴스 이슈분석

사회문제를 다루는 대용량 뉴스 기사를  
토픽모델링을 통해 주제를 찾아 사회적  
이슈에 관한 키워드를 찾는 시스템에 활용

- SNS 이슈 트래킹

트위터 데이터로 SNS상의 주요 이슈를  
추출하는 이슈 트래킹에 활용

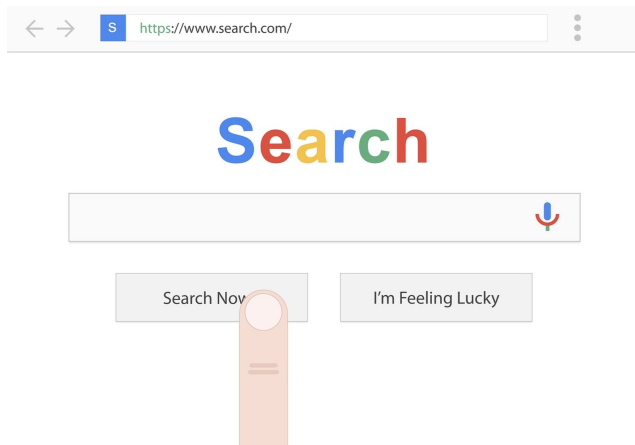
## 토픽 모델링 활용(2) - 트렌드 분석



- 산업 트렌드 분석  
산업분야에 토픽모델링을 적용하여 이슈를 발견하고 트렌드를 분석하여 전략수립에 활용할 수 있다.



## 토픽 모델링 활용(3) - 검색엔진



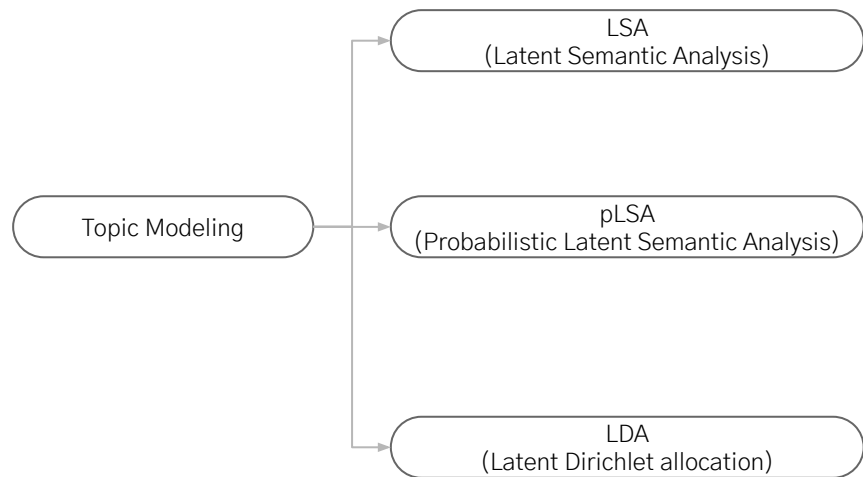
- 검색엔진 최적화(SEO)

주제와 관련 키워드를 파악하여 온라인 기사, 블로그, 문서에 태그를 지정할수 있다.  
지정된 태그로 검색 결과 최적화 개선에 활용한다.



이제, 토픽모델링을 하는 방법을 알아보까요?

# 토픽모델링 종류



# LSA(Latent Semantic Analysis)

# LSA(Latent Semantic Analysis)



- 잠재의미분석
- 대량의 텍스트 문서에서 발생하는 단어들간의 연관관계를 분석함으로써 잠재적인 의미구조를 도출
- 문서 집합 내의 연관성, 즉 동시출현(co-occurrence)빈도가 높은 단어들을 기준으로 유사한 문서를 추출함





“동시출현(co-occurrence)”의 의미

동시출현 정보를 이용한다는 것은 형태(morphology)가 아닌 의미(semantic)를 이용한다는 뜻이다.

예를 들어, '배'라는 단어는 같은 문장에 동시출현하는 동사가 '타다'인지 '먹다'인지에 따라 의미가 달라지게 된다.

# LSA(Latent Semantic Analysis)

- ‘동일한 의미를 공유하는 단어들은 같은 텍스트 안에서 발생한다.’는 가정에서 시작 
- TDM(단어-문서행렬)을 바탕으로 문서의 잠재된(Latent) 의미를 이끌어 내는 방법으로 기존의 BOW에 기반한 TDM과 TF-IDF가 단어 빈도수로 중요도를 판단하는 단점을 보완한 모델
- 문서집합을 TDM(단어-문서행렬)으로 표현하고, 이것을 SVD 분해를 통해 차원수를 줄여 계산의 효율성을 높이고, 잠재적(Latent)의 의미를 찾아낸다. 

$$\begin{array}{c} \text{단어} \\ \text{문서} \end{array} \begin{array}{c} \text{문서} \\ \text{TDM 행렬} \end{array} = \begin{array}{c} \text{주제} \\ \text{단어} \end{array} \times \begin{array}{c} \text{주제} \\ \text{문서} \end{array} \times \begin{array}{c} \text{문서} \\ \text{단어} \end{array}$$

\* 주제 = Latent

# SVD 분해

- SVD(Singular Value Decomposition, 특이값 분해)

실수 공간, 행렬  $A = m \times n$ 에 대하여 다음과 같이 행렬분해(decomposition)을 할 수 있다.

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

$U$ :  $m \times m$  orthogonal matrix

$\Sigma$ :  $m \times n$  diagonal matrix

$V$ :  $n \times n$  orthogonal matrix

$$\begin{pmatrix} A \\ m \times n \end{pmatrix} = \begin{pmatrix} U \\ m \times m \end{pmatrix} \begin{pmatrix} \Sigma \\ m \times n \end{pmatrix} \begin{pmatrix} V^T \\ n \times m \end{pmatrix}$$

# SVD 분해 - 참고

- SVD(Singular Value Decomposition, 특이값 분해)

❖ orthogonal matrix(직교행렬)

$$U^T U = U U^T = I$$

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

❖ diagonal matrix(대각행렬)

대각성분을 제외한 나머지는 모두 '0'인 행렬

$$\text{diag}(d_1, \dots, d_n) = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}$$

왜, SVD 분해를 사용할까요?



# Reduce SVD

$$A = U \times \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_s \\ \hline & & & 0 \end{bmatrix} \times V^T$$



full SVD

$$A = U_s \times \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_s \\ \hline & & & 0 \end{bmatrix} \times V^T$$



thin SVD

$$A = U_r \times \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \\ \hline & & & 0 \end{bmatrix} \times V_r^T$$



compact SVD

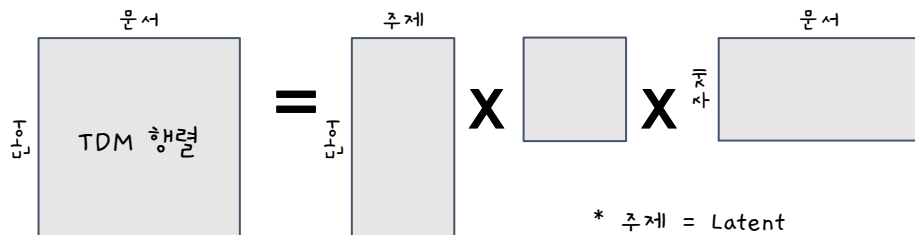
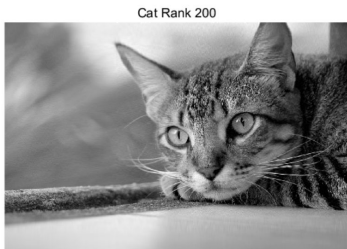
$$A' = U_t \times \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_t \\ \hline & & & 0 \end{bmatrix} \times V_t^T$$



truncated SVD

데이터 압축과 노이즈제거에 효과적으로 사용될 수 있다!

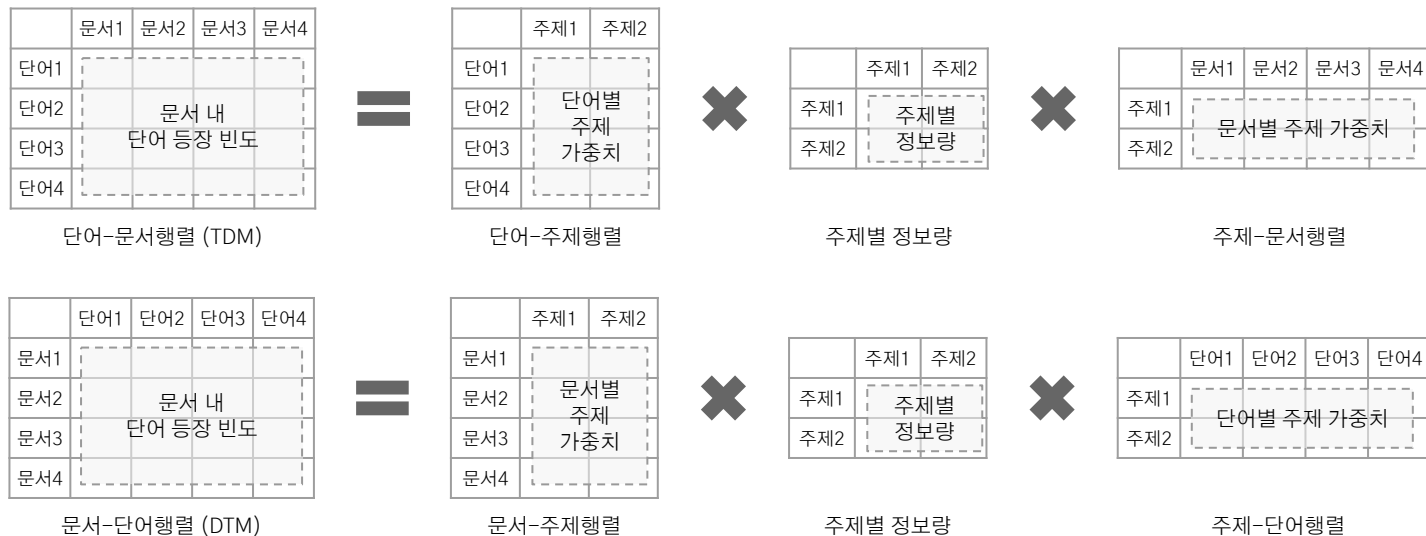
\_\_\_\_\_



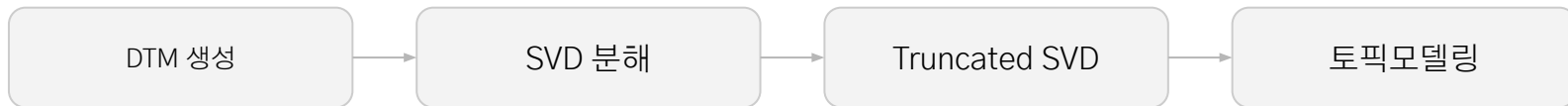
다시 LSA로 되돌아가서,

# LSA(Latent Semantic Analysis)

## - LSA의 행렬분해



# LSA 과정

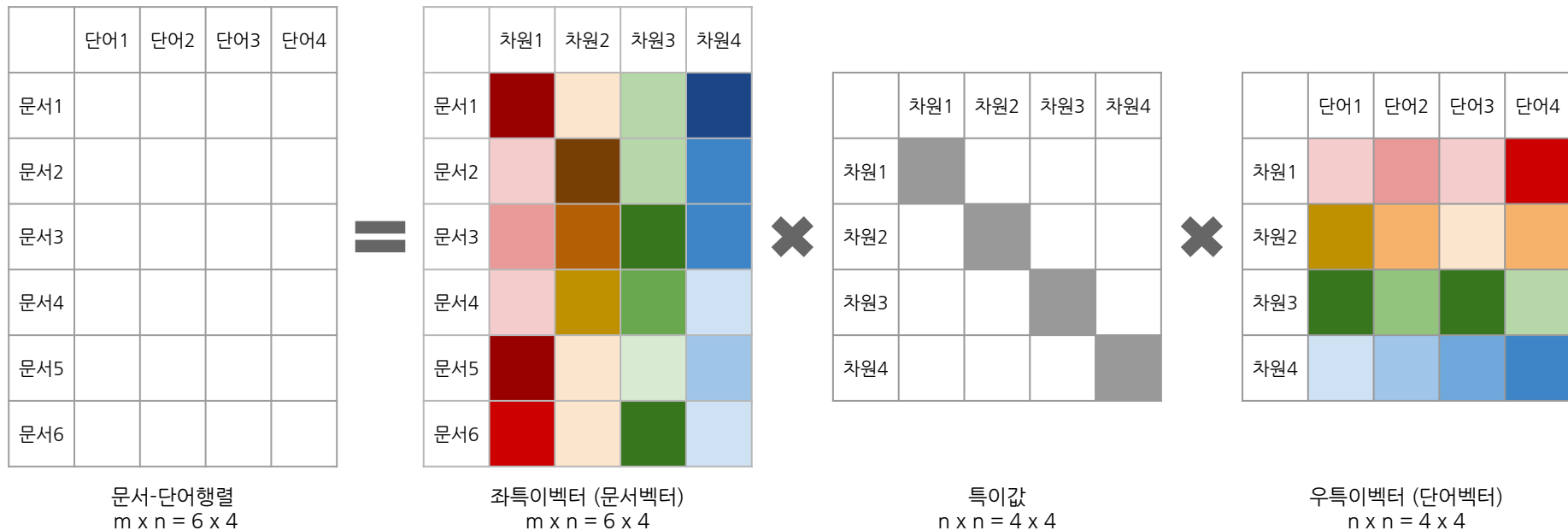


DTM 외에도 TDM이나  
TF-IDF도 사용될 수 있다

사용자가 설정한 '주제 수'  
에 따라 결정된다.

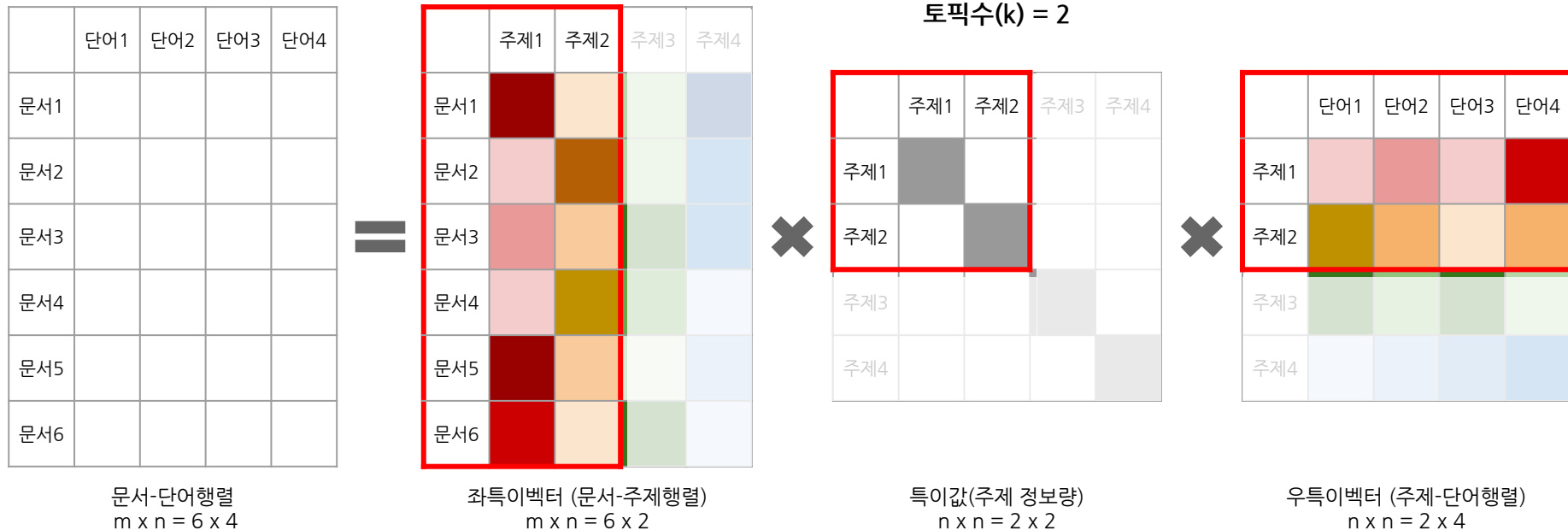
# LSA 과정 (1)

- DTM 생성 후, SVD 분해



# LSA 과정 (2)

- Truncated SVD -> 주제 2개로 가정

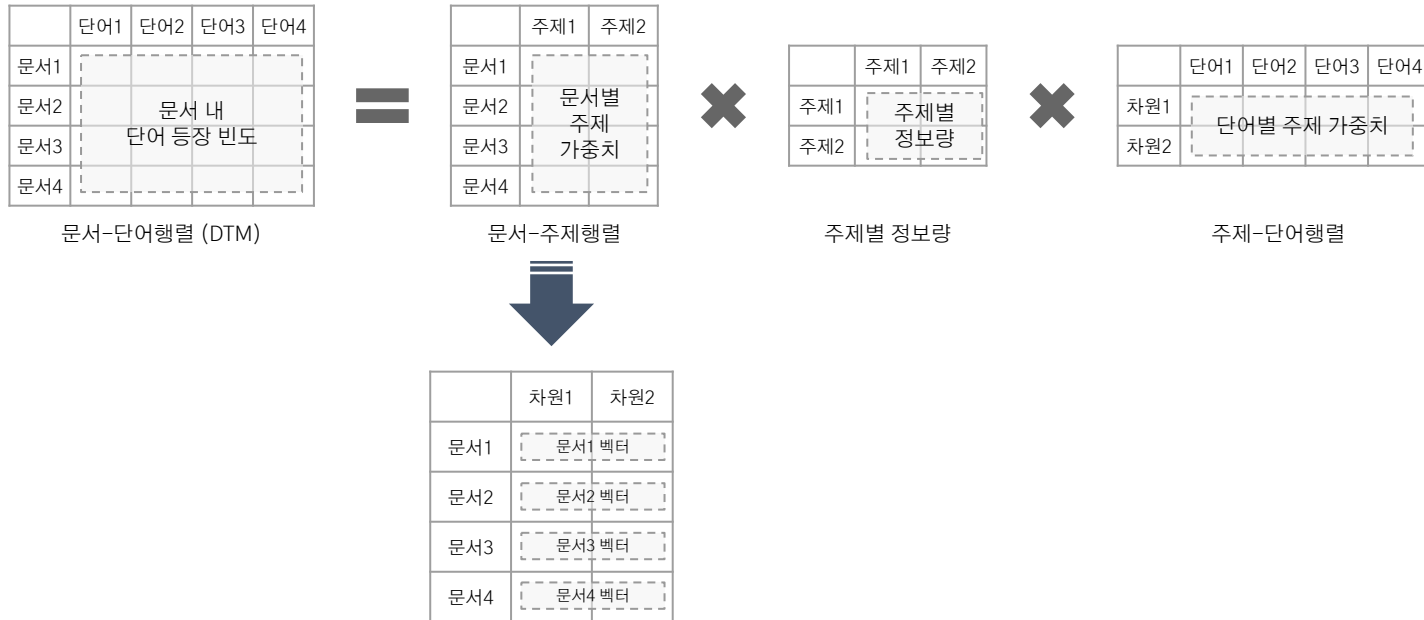


그렇다면, LSA를 어떻게 활용할 수 있을까요?



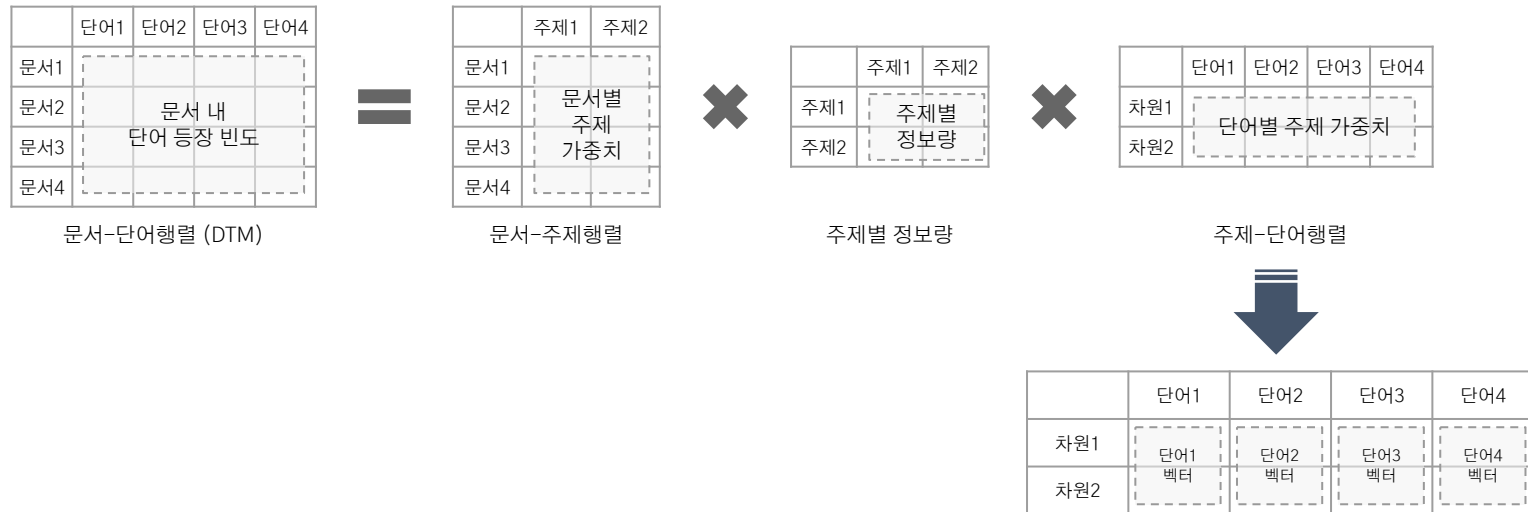
# LSA 활용 (1)

## - 문서벡터로 활용



## LSA 활용 (2)

### - 단어벡터로 활용



# LSA 활용 (3)

- 단어/문서간 유사도 측정
- 단어-단어, 문서-문서, 단어-문서 모두 가능.

	단어1	단어2	단어3	단어4
차원1	단어1 벡터	단어2 벡터	단어3 벡터	단어4 벡터
차원2	단어1 벡터	단어2 벡터	단어3 벡터	단어4 벡터

단어1과 단어2의 유사도 측정가능

	차원1	차원2
문서1	문서1 벡터	
문서2	문서2 벡터	
문서3	문서3 벡터	
문서4	문서4 벡터	

문서2와 문서4의 유사도 측정가능

단어3과 문서4의 유사도 측정가능

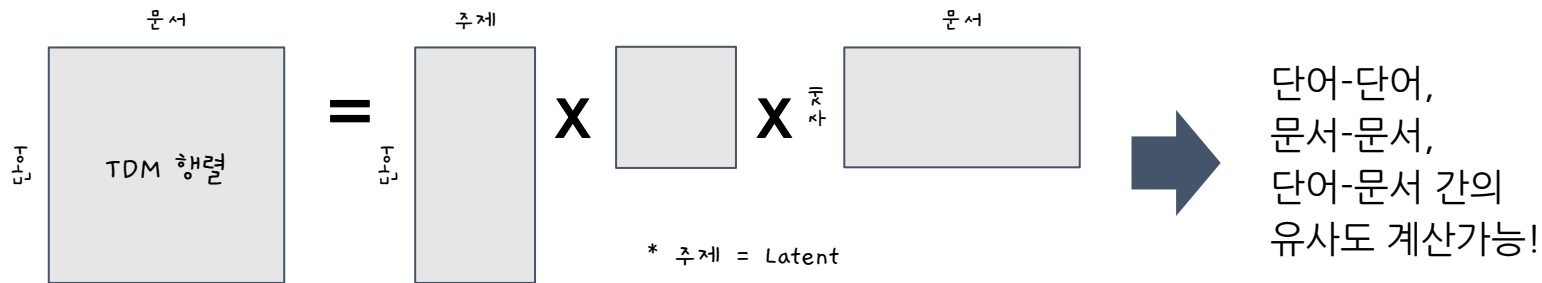
그래서 전체 LSA의 분석과정을 살펴보면

# 잠재 의미 분석 절차



# LSA 정리

- 잠재의미분석
- '동일한 의미를 공유하는 단어들은 같은 텍스트 안에서 발생한다.'는 가정에서 시작한 모델
- 단어 문서 행렬을 SVD를 사용해 행렬을 분해하고 차원을 축소해서 근접한 단어들끼리 유한 주제로 묶어주는 토픽모델링의 방법 중 하나



# 실습1 - 간단한 토픽모델링 구현 LSA

문서 구분	내용	
문서1	바나나 사과 포도 포도	과일
문서2	사과 포도	
문서3	포도 바나나	
문서4	짜장면 짬뽕 탕수육	중식
문서5	볶음밥 탕수육	
문서6	짜장면 짬뽕	
문서7	된장찌개 김치찌개 김치 비빔밥	한식
문서8	김치 된장 비빔밥	
문서9	비빔밥 김치	
문서10	사과 볶음밥 김치 된장	→ 섞여있어요

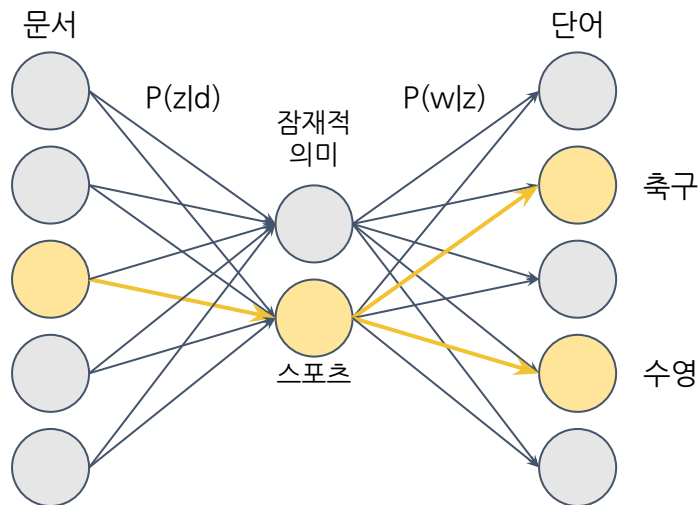
위의 10개의 문서에 총 몇가지 주제가 있을까요?

# pLSA(Probabilistic Latent Semantic Analysis)



# pLSA(Probabilistic Latent Semantic Analysis)

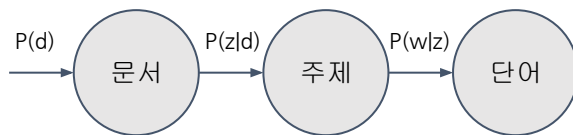
- 확률적 잠재의미분석
- 토픽모델링을 위해 잠재의미분석에서 사용하는 SVD분해 대신, 확률적 방법을 사용
- pLSA는 “잠재적(Latent)의미가 존재하고, 이 잠재적의미가 문서와 단어를 연결한다”고 가정



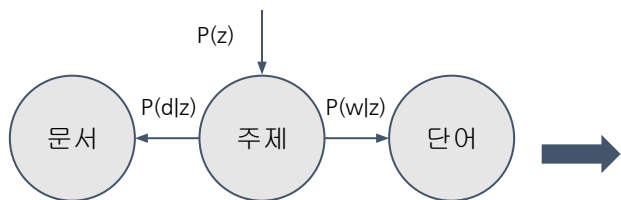
$P(z|d)$  : 문서 하나가 주어졌을때 특정 주제(토픽)가 나타날 확률  
 $P(w|z)$  : 주제가 정해졌을 때 특정 단어가 나타날 확률

# pLSA 모델 구성

- 토픽모델링의 가정 “문서는 여러 주제로 구성되 있고, 각 주제는 단어 집합으로 구성된다”

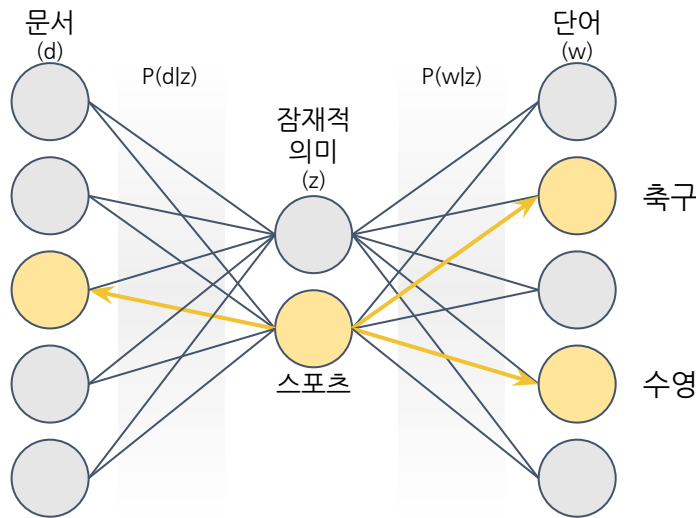


이건 어떨까요?



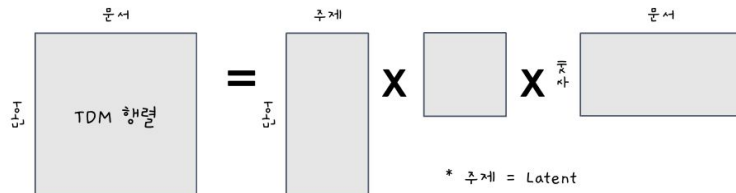
➡ 이러한 원리가 pLSA !

# pLSA(Probabilistic Latent Semantic Analysis)



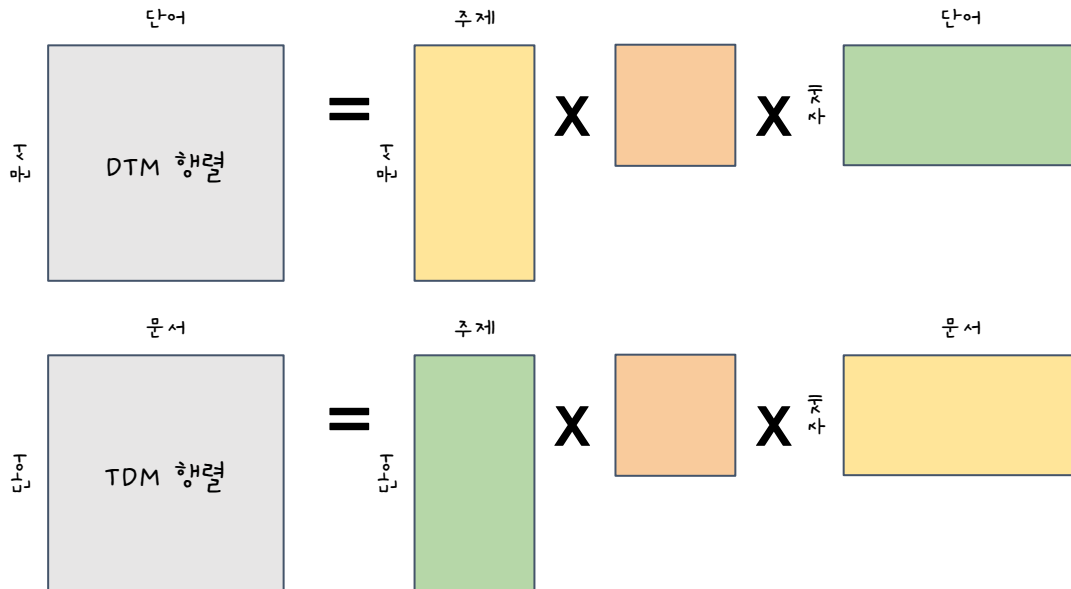
$$P_{pLSA}(d, w) = \sum_z P(d|z)P(z)P(w|z)$$

참고) LSA와 비교



# LSA와 pLSA 비교

$$P_{pLSA}(d, w) = \sum_z P(d|z)P(z)P(w|z)$$



# pLSA 한계

---

- 새로운 문서가 들어왔을때, 이것을 추정하기 어렵다.
- pLSA의 파라미터는 분석할 문서 수에 따라 선형적으로 증가한다.

# LDA(Latent Dirichlet Allocation)

# LDA(Latent Dirichlet Allocation)

- 잠재 디리클레 할당

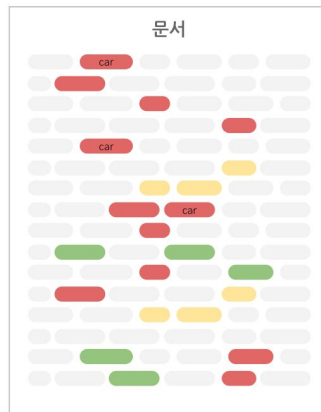
주어진 문서에 대해 어떤 주제가 존재하는지에 대한 확률모형

- '주제 내 단어분포'와 '문서내 주제 분포'를 추정하는 방법

→ 분포를 추정할 때 디리클레 다항분포를 사용한다.

디리클레 다항분포란?

연속확률분포의 하나로  $k$ 차원의 실수 벡터의 요소가 양수이며,  
모든 요소를 더한 값이 1인 경우에 대해 확률값이 정의된 분포



주제 내 단어 분포

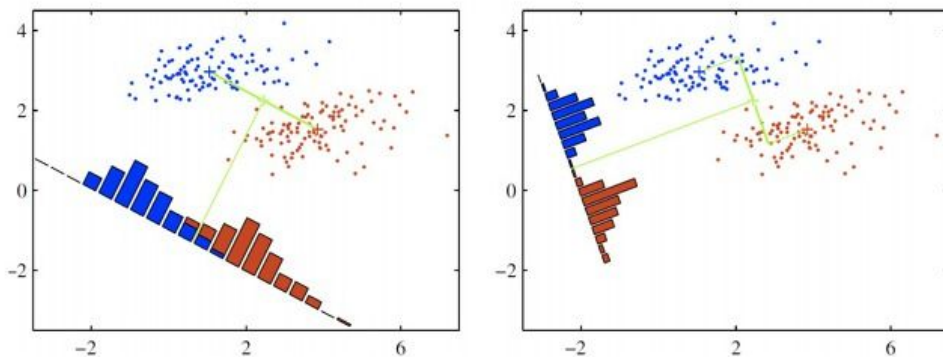
car	0.43
vehicle	0.15
motorcycle	0.07
banana	0.33
apple	0.12
strawberry	0.01

문서 내 주제 분포



## 잠재 디리클레 할당 (LDA) (2)

- 잠재(Latent) : 사전적인 의미는 “잠재적인, 숨어 있는”. 우리가 직접 관찰할 수 있는 것은 문서 내용뿐.  $\alpha$ ,  $\beta$ ,  $\theta$ ,  $z$ 는 모두 감춰진 파라미터
- 디리클레(Dirichlet) : 19세기 독일 수학자의 이름. 디리클레 분포(Dirichlet Distribution)를 사용하고 있음.  
( $\theta$ 를 결정할 때  $\alpha$ 를 파라미터로 디리클레 분포를 사용)
- 할당(Allocation) : ‘할당’. 각 단어를 결정할 때,  $\theta$ 에 대한 다항 분포(Multinomial Distribution)로 주제를 ‘할당’한 뒤 그 주제로부터 단어를 추출.





© 2006 The Authors  
Journal compilation © 2006 Blackwell Publishing Ltd

- 문서의 내용을 관찰하여 감춰진 파라미터들을 디리클레 분포를 사용하여 각 단어에 주제를 할당하는 과정
- 전체 텍스트 문서 집합의 주제(토픽)들, 각 텍스트 문서별 주제의 확률, 각 단어들이 각 주제에 포함될 확률을 디리클레 분포를 사용하여 도출 한다.

# LDA(Latent Dirichlet Allocation)

여기가 핵심!

gene이 등장할 확률 0.04  
dna가 등장할 확률 0.02  
genetic이 등장할 확률 0.01로  
단어를 보고 “유전”과 관련된  
토픽이라고 유추할 수 있음

Topics

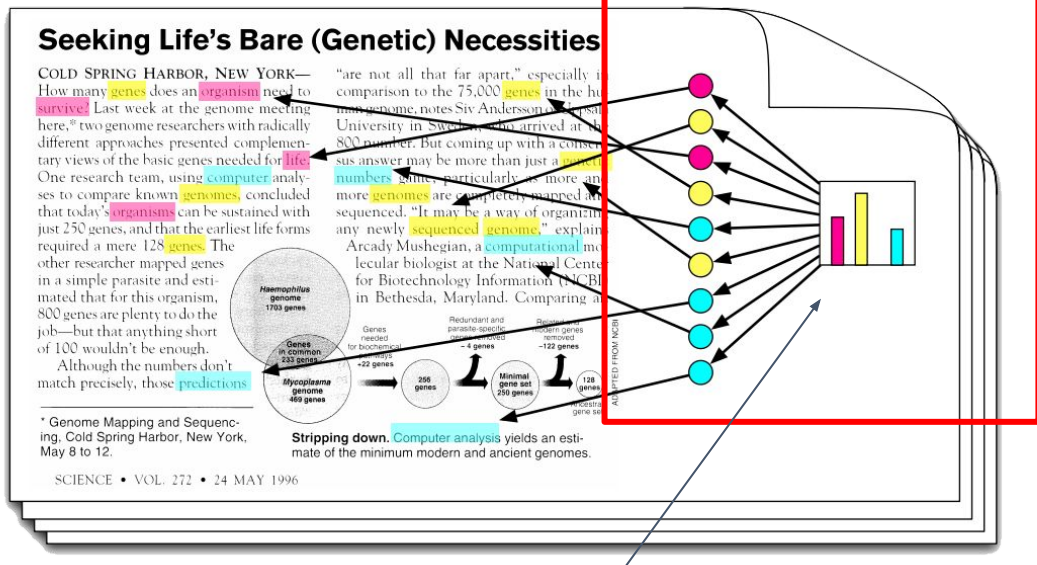
gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

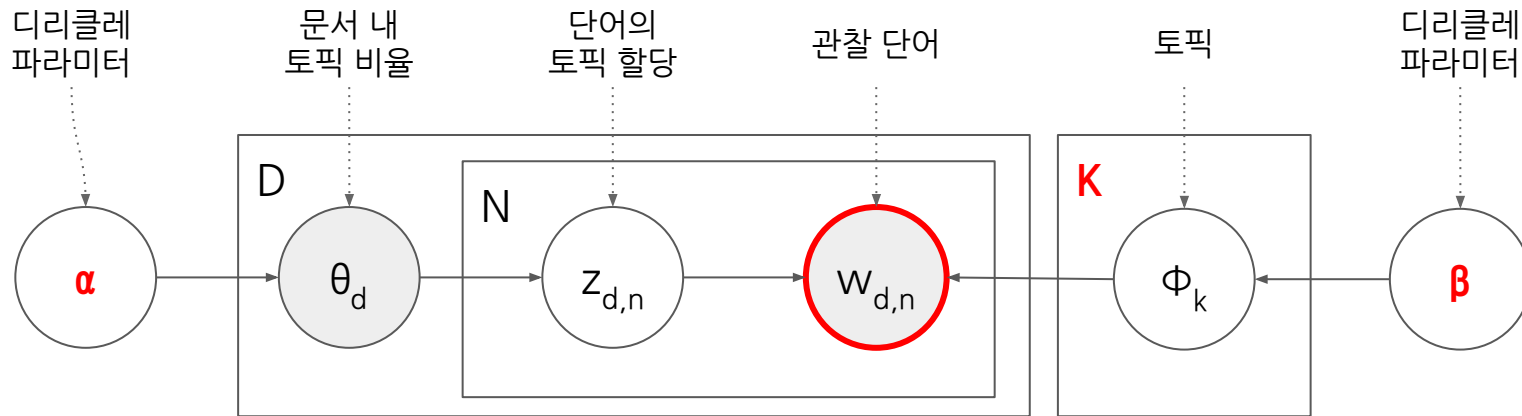
Documents



토픽 비중의 결과 노란색 토픽이 다수 존재하는 것으로 보아,  
이 문서의 주제는 “유전”으로 유추할 수 있음

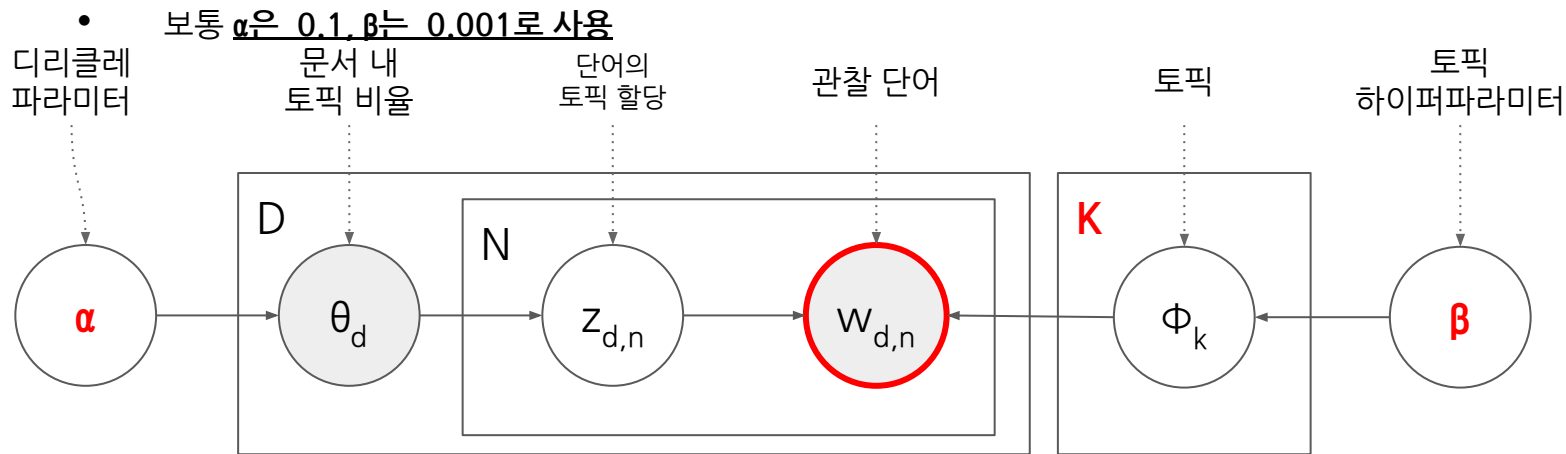
# 잠재 디리클레 할당 모델 (1)

$\alpha$	디리클레 파라미터 (보통 0.1)	$D$	전체 문서 갯수
$\theta_d$	문서 내 토픽 비율	$\phi_k$	토픽
$z_{d,n}$	단어의 토픽 할당	$K$	토픽수
$w_{d,n}$	관찰단어	$\beta$	토픽 하이퍼파라미터 (보통 0.001)
$N$	N은 d번째 문서의 단어 수		



## 잠재 디리클레 할당 모델 (2)

- 관찰 가능한 변수는 d번째 문서에 등장한 n번째 단어  $w_{d,n}$ 가 유일
- 이 정보를 가지고 하이퍼파라미터(사용자 지정)  $\alpha, \beta$ 를 제외한 모든 잠재 변수를 추정
- 사전에 결정해주어야 할 값은  $\alpha, \beta, K$ 값



# 깁스 샘플링(Gibbs sampling)

- 문헌 d에 속하는 어떤 단어 m이 주제 j에 속할 확률은  
(주제 j에 속하는 모든 단어 중에서 단어 m이 차지하는 비중)X(문헌 d에 속하는 모든 주제 중 주제 j가 차지하는 비중) 의 곱에 비례함

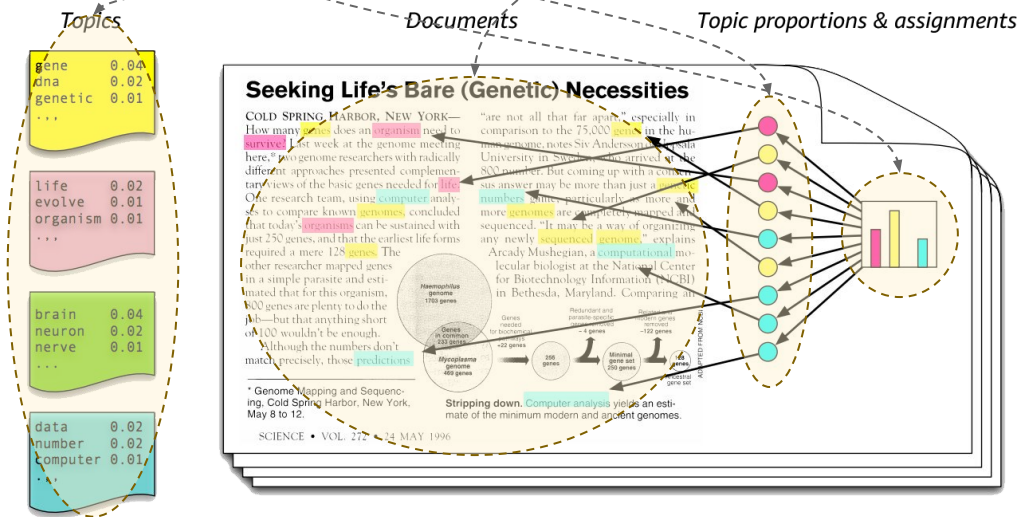
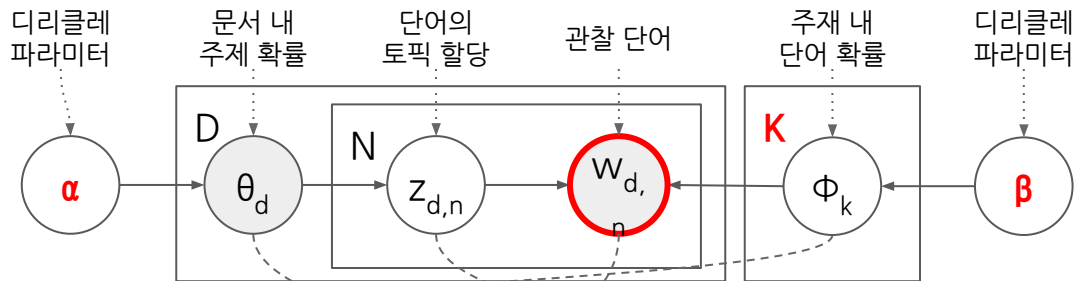
문서  $d_i$ 에서 단어  $w_i$ 가  
토픽 j로 할당될 확률

토픽 j에  
할당된 단어수

문서  $d_i$ 내 토픽 j에  
할당된 단어수

$$P(z = j \mid w_i, d_i) \propto \frac{C_{w_i j}^{WT} + \beta}{\sum_{w=1}^W C_{w j}^{WT} + W\beta} \cdot \frac{C_{d_i j}^{DT} + \alpha}{\sum_{t=1}^T C_{d_i t}^{DT} + T\alpha}$$

## 잠재 디리클레 할당 모델 (2)



# LDA 과정 개요

말뭉치로부터 토픽분포를 뽑는다

토픽분포로부터 토픽을 뽑는다

토픽에 해당하는 단어를 뽑는다.

좋은점?

현재 문서에 등장한 단어들이  
어떤 토픽에서부터 나온것인지를  
알 수 있다.

# LDA 수행과정

LDA 추정분포

각 문서의  
토픽분포 추정

각 토픽내  
단어분포 추정

토픽 개수  $k$  를 설정

모든 단어를  
 $k$ 개 토픽 하나에 임의 할당

재할당 반복

- $D$ 개의 전체 문서에  $k$ 개 토픽이 분포되어있다고 가정
- 모든 단어를  $k$ 개 토픽 중 하나를 임의 할당
  - 각 문서는 토픽을 가짐
  - 토픽은 단어 분포를 가짐
- 임의 할당 했지만 올바르게 할당되었다고 가정
- 다음 과정을 반복하여 토픽을 재할당
  - $p(t|d)$ : 문서 내 주제확률
  - $p(w|t)$ : 주제 내 단어확률
  - $p(t|d) * p(w|t)$ : 주제  $z$ 에 대해, 문서  $d$  내에서 단어  $w$ 가 존재할 확률
- 안정적인 상태(결과가 수렴)까지 반복



# LDA 계산 절차 예제

A: Cute kitty

B: Eat rice or cake

C: Kitty and hamster

D: Eat bread

E: Rice, bread and cake

F: Cute hamster eats bread and cake

위 문서를 LDA를 사용해서 토픽모델링을 해보자!



# LDA 계산 절차 예제

1단계 : 토픽 개수  $k$  설정

-> 2개로 설정하자

토픽 개수  $k$  를 설정

모든 단어를  
 $k$ 개 토픽 하나에 임의 할당

재할당 반복

토픽 개수 k 를 설정

모든 단어를  
k개 토픽 하나에 임의 할당

재할당 반복

# LDA 계산 절차 예제

2단계 : 모든 단어들에 2개의 토픽을 임의로 할당

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	#1	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

토픽 개수 k 를 설정

모든 단어를  
k개 토픽 하나에 임의 할당

재할당 반복

# LDA 계산 절차 예제

3단계 : 재할당 - 준비1

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	?	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

cute라는 단어에 대해, “문서내 토픽 확률( $P(z|d)$ )”과 “토픽내 단어확률( $P(w|z)$ )”을 구한다.

이때,  $\alpha = 0.1$

문서내 토픽등장분포 = 등장빈도 +  $\alpha$

$\theta$	A	B	C	D	E	F
#1	0.1	2.1	0.1	2.1	2.1	2.1
#2	1.1	1.1	2.1	0.1	1.1	3.1
sum	1.2	3.2	2.2	2.2	3.2	5.2

토픽 개수 k 를 설정

모든 단어를  
k개 토픽 하나에 임의 할당

재할당 반복

# LDA 계산 절차 예제

3단계 : 재할당 - 준비2

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	?	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

cute라는 단어에 대해, “문서내 토픽 확률( $P(z|d)$ )”과 “토픽내 단어확률( $P(w|z)$ )”을 구한다.

이때,  $\beta = 0.001$

토픽내 단어분포 = 토픽내 단어빈도 +  $\beta$

$\phi$	cute	kit	eat	rice	cate	ham	bre	sum
#1	0.001	0.001	2.001	1.001	2.001	0.001	2.001	8.007
#2	1.001	2.001	1.001	1.001	0.001	2.001	1.001	8.007

토픽 개수 k 를 설정

모든 단어를  
k개 토픽 하나에 임의 할당

재할당 반복

# LDA 계산 절차 예제

3단계 : 재할당 준비(앞의 1, 2 결과 정리)

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	?	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

$\theta$	A	B	C	D	E	F
#1	0.1	2.1	0.1	2.1	2.1	2.1
#2	1.1	1.1	2.1	0.1	1.1	3.1
sum	1.2	3.2	2.2	2.2	3.2	5.2

$\phi$	cute	kit	eat	rice	cate	ham	bre	sum
#1	0.001	0.001	2.001	1.001	2.001	0.001	2.001	8.007
#2	1.001	2.001	1.001	1.001	0.001	2.001	1.001	8.007

### 3단계 : 재할당 cute

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	?	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

$\theta$	A	B	C	D	E	F
#1	0.1	2.1	0.1	2.1	2.1	2.1
#2	1.1	1.1	2.1	0.1	1.1	3.1
sum	1.2	3.2	2.2	2.2	3.2	5.2

$\phi$	cute	kit	eat	rice	cate	ham	bre	sum
#1	0.001	0.001	2.001	1.001	2.001	0.001	2.001	8.007
#2	1.001	2.001	1.001	1.001	0.001	2.001	1.001	8.007

$$P(cute|\text{토픽1}) = \frac{0.001}{8.007} = 0.000125$$

$$P(\text{토픽1}|A) = \frac{0.1}{0.1+1.1} = 0.0834$$

$$P(\text{토픽1}|cute, A) = 0.000125 * 0.0834 = 0.0000104$$

$$P(cute|\text{토픽2}) = \frac{1.001}{8.007} = 0.125$$

$$P(\text{토픽2}|A) = \frac{1.1}{0.1+1.1} = 0.9167$$

$$P(\text{토픽2}|cute, A) = 0.125 * 0.9167 = 0.1145875$$

**-> cute는 토픽2로 재할당**

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	#2	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	#2	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1



w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	#2	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

w	cute	kit	eat	rice	cake	kit	ham	eat	bre	rice	bre	cake	cute	ham	eat	bre	cake
z	#2	#2	#1	#2	#1	#2	#2	#1	#1	#1	#2	#1	#2	#2	#2	#1	#1

순서대로 재할당  
일정한 값으로 수렴할때까지 반복하거나,  
사용자가 지정한 횟수만큼 반복한다



# 잠재 디리클레 할당 한계

- LDA 분석방법 샘플링을 이용하기 때문에 실행시마다 결과가 달라질 수 있음
  - 문서 수가 적고 단어가 희소 할 수록 결과가 달라질 수 있음
- 단어의 분포만을 가지고 주제를 그룹핑 하기 때문에 사람이 인지하는 주제와 얼마나 일치할까에 대한 문제 
- 파라미터 설정의 어려움 
  - 토픽의 수 K값을 얼마나 두는게 적절한지 모름
  - 적절한 K값을 설정하고 그에 따르는  $\alpha$ ,  $\beta$ 값을 잘 튜닝해야 좋은 결과를 얻을 수 있음

# LDA 정리

---

- 잠재 디리클레 할당(LDA, Latent Dirichlet Allocation)란 주어진 문서에 대해 어떤 주제가 존재하는지에 대한 확률모형 (토픽모델링)
- LDA는 토픽별 단어의 분포, 문서별 토픽의 분포를 추정
- 결과적으로 전체 텍스트 문서 집합의 주제(토픽)들, 각 텍스트 문서별 주제의 확률, 각 단어들이 각 주제에 포함될 확률을 도출(디리클레:확률분포명칭)

# LSA, LDA 모델 비교



	LSA	LDA
이름	잠재의미분석	잠재 디리클레할당
가정	동일한 의미를 공유하는 단어는 같은 텍스트 안에 등장한다.	토픽의 단어분포와 문서의 토픽분포의 결합으로 문서 내 단어들이 생성된다.
특징요약	단어-문서 행렬을 <b>SVD</b> 행렬 분해를 사용해 행렬 차원을 축소해서 축소차원에서 근접단어들로 토픽을 선정	단어가 특정 토픽에 존재할 확률과 문서에 특정 토픽이 존재할 확률을 추정하여 토픽 선정

# 실습1 - 간단한 토픽모델링 구현 LDA

문서 구분	내용	
문서1	바나나 사과 포도 포도	과일
문서2	사과 포도	
문서3	포도 바나나	
문서4	짜장면 짬뽕 탕수육	중식
문서5	볶음밥 탕수육	
문서6	짜장면 짬뽕	
문서7	된장찌개 김치찌개 김치 비빔밥	한식
문서8	김치 된장 비빔밥	
문서9	비빔밥 김치	
문서10	사과 볶음밥 김치 된장	섞여있어요

위의 10개의 문서에 총 몇가지 주제가 있을까요?

## 실습2 - 리뷰데이터로 토픽모델링

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
2	10265843	너무재밌었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0
4	6483659	사이폰페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1
...	...	...	...
9995	8665166	곰티비로 무료로 봤기때문에 5점주려고했는데 1 한국 공포영화의 특징인 깜놀시키려 하...	0
9996	8312675	이편걸드라마라고했냐 수습할수없으면강친자녀아니면되고 간단하네 얼굴을바꿨으면 결말이라도...	0
9997	6386483	웬지 김연아 크면 에리카처럼 될것같음.	1
9998	4452600	솔직히 굿 ㅋㅋㅋㅋ 넘버1씨는 살아남길 바랬는데 2번째극장판 어서 나오길	1
9999	9832698	그냥보다나옴 노답 핵노잼	0

10000 rows × 3 columns

**pyLDAvis** 를 사용해서 토픽모델링을 진행해보고,

**다음 데이터 중 부정적인 리뷰를 몇개의 토픽으로 분리하는 것이 가장적절한지 적절한 토픽갯수를 찾아보세요.**