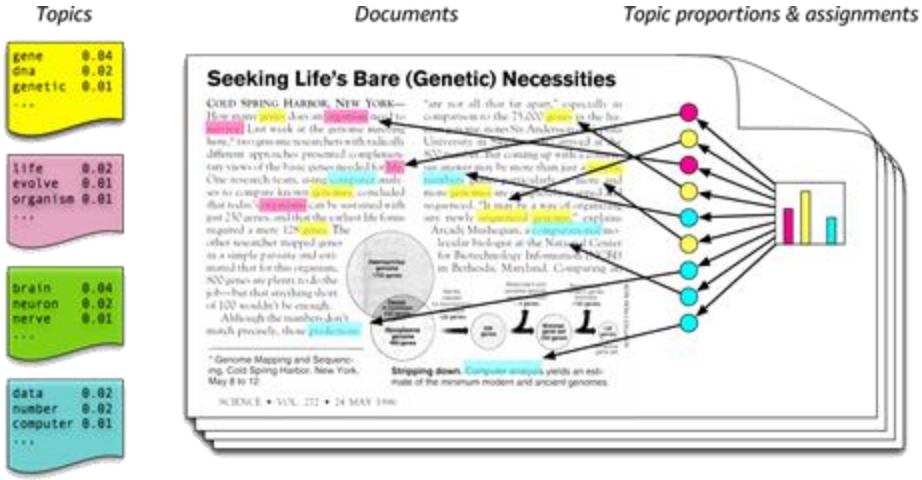
토픽 모델링

최희윤 강사

토픽 모델링(Topic Modeling)

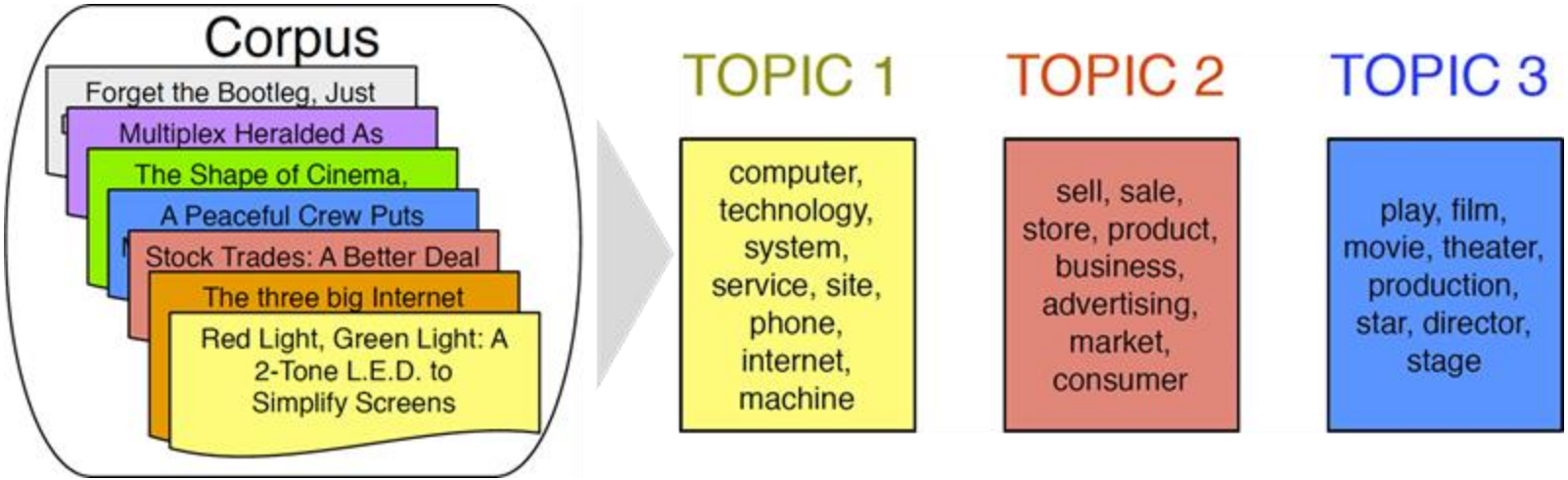
- ✓ 문서 집합의 추상적인 "주제"를 발견하기 위한 통계적 모델 중 하나로,
  텍스트 본문의 숨겨진 의미 구조를 발견하기 위해 사용되는 텍스트 마이닝 기법
- ✓ 특정 주제에 관한 문헌에서는 그 주제에 관한 단어가 다른 단어들에 비해 더 등장,
  그 단어들을 잠재적인 "주제"로 분류



ref)토픽 모델 - 위키백과, 우리 모두의 백과사전 (wikipedia.org)

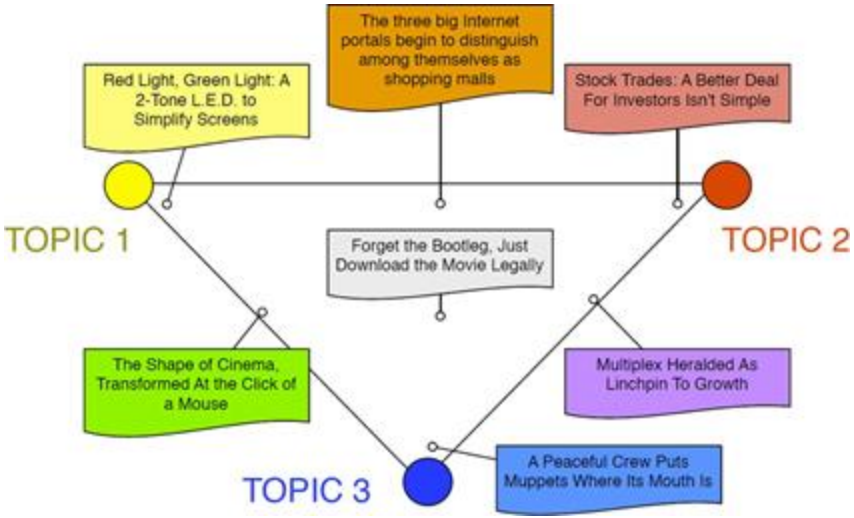
토픽 모델링(Topic Modeling)

✓ Corpus(말뭉치)로부터 K개의 Topic을 추출 -> Word to Topics



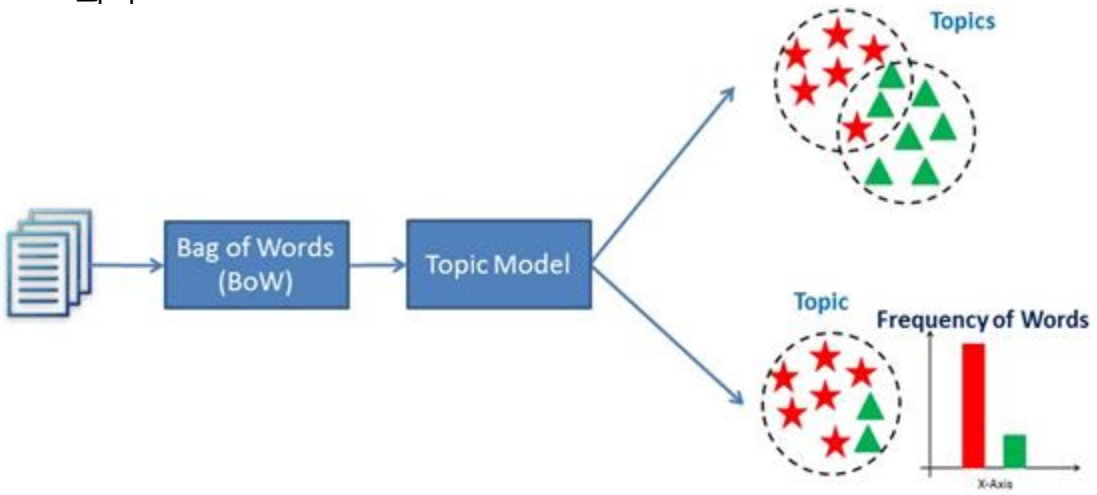
토픽 모델링(Topic Modeling)

- ✓ 각각의 문서(Document)는 추출된 Topic들로 표현 가능
- ✓ 문서 A는 Topic1 위주로 구성 혹은 문서 B는 Topic2 위주로 구성되어 있다고 볼 수 있음



토픽 모델링(Topic Modeling)

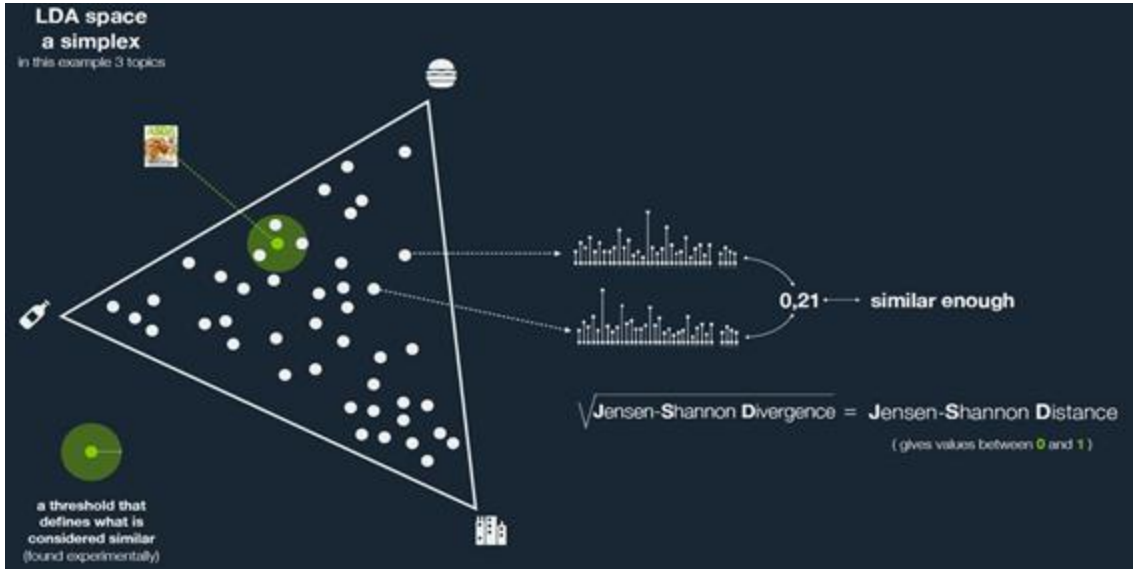
1. 개별 문서는 1개 이상의 주제를 다룰 수 있다는 점을 전제로 수집된 텍스트를 토픽의 확률적 혼합체로 간주
2. 각 토픽을 추출된 키워드들의 분포로 나타냄으로써 텍스트 내의 구조를 파악



ref)토픽 모델 - 위키백과, 우리 모두의 백과사전 (wikipedia.org)

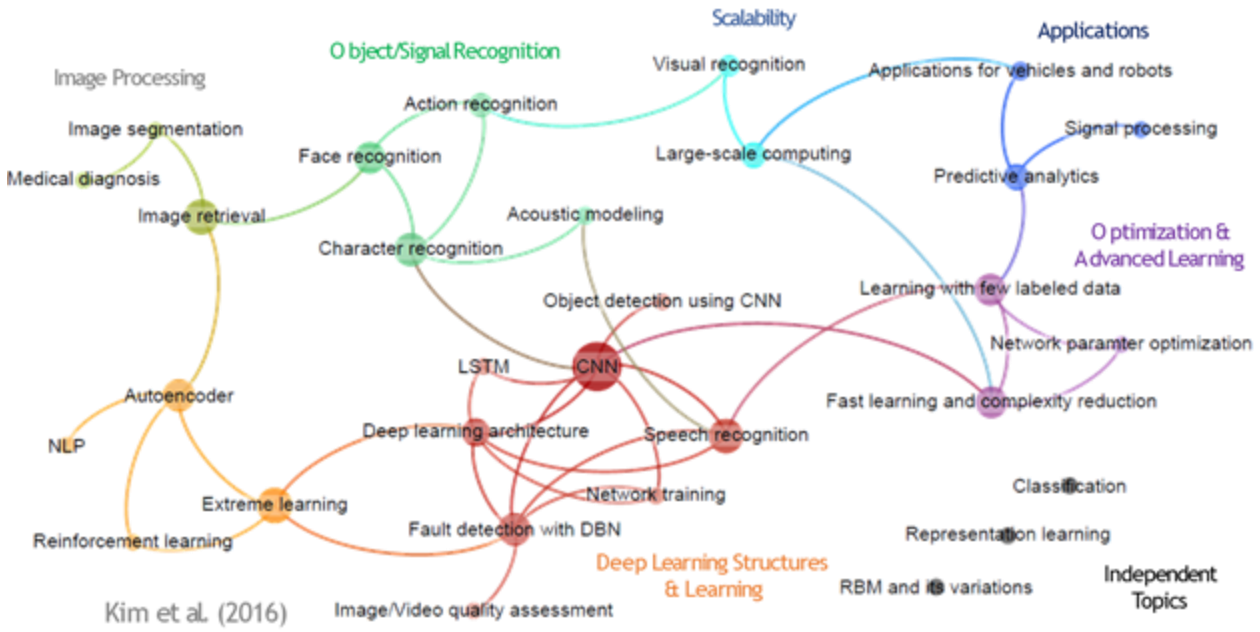
토픽 모델링(Topic Modeling)

1. 각각의 점들은 하나의 문서이고 오른쪽 분포는 Topic의 Distribution(분포)
2. 즉 문서는 토픽으로 이루어진 벡터라고 볼 수 있으므로 각각 문서들의 토픽으로 이루어진 벡터의 유사도를 통해서 각 문서가 얼마나 유사한지 알 수 있음



토픽 모델링(Topic Modeling)의 활용

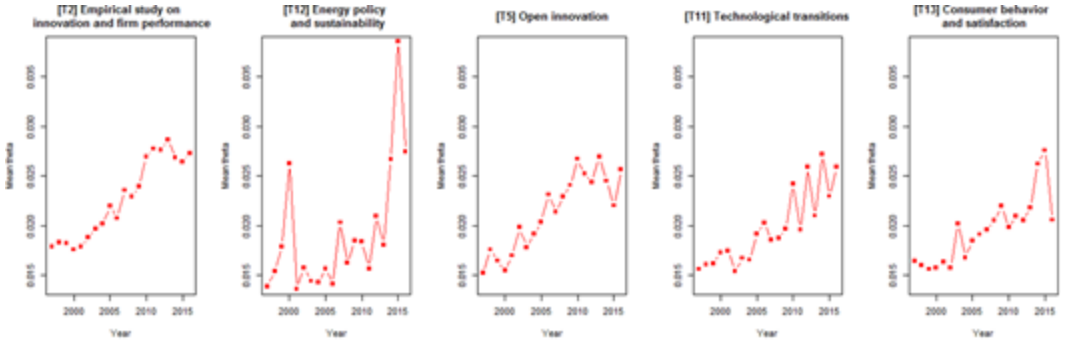
✓ 토픽간의 유사한 토픽을 계산할 수 있음. 딥러닝 아카이브를 토픽 모델링을 했을 때의 결과



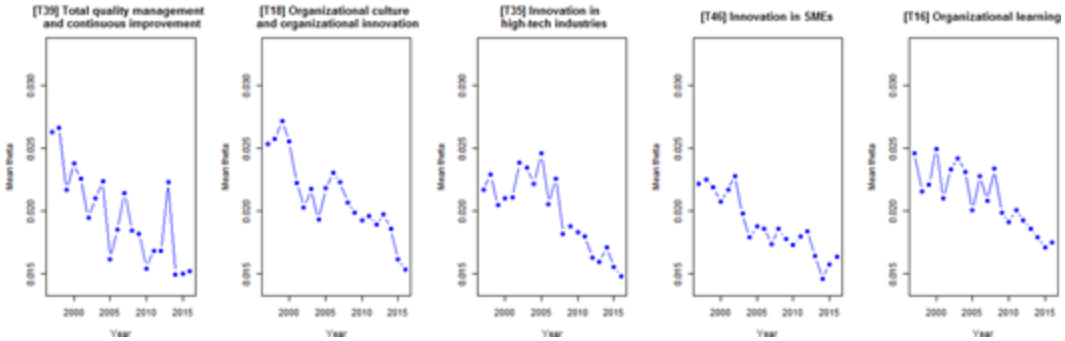
토픽 모델링(Topic Modeling)의 활용

✓ 기간별 토픽의 비중을 비교해봤을 때 트렌드를 확인할 수도 있음.

Hot Topic



Cold Topic



Lee and Kang (2017)

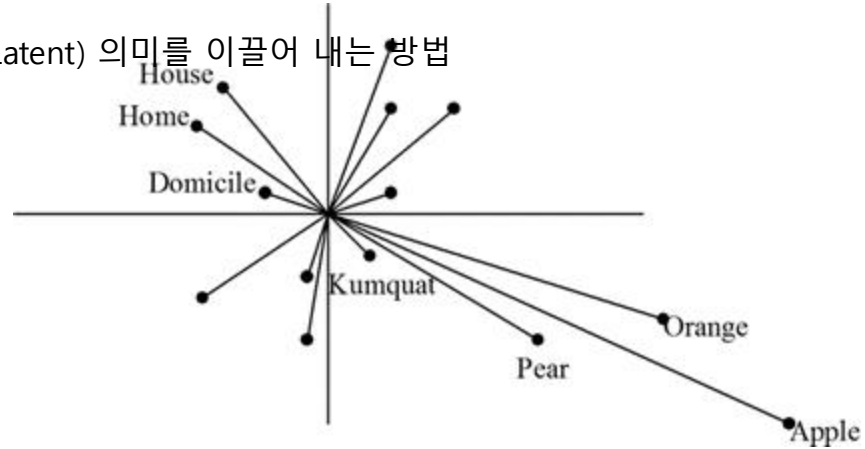
토픽 모델링(Topic Modeling)의 활용

- ✓ 사회 문제를 다루고 있는 대용량 뉴스기사로부터 토픽분석을 적용하여 사회적 이슈에 관한 키워드를 도출하는 시스템을 제안
- ✓ 트위터 데이터를 대상으로 SNS상에서의 주요 이슈를 추출하는 트위터 이슈 트래킹 시스템을 제안
- ✓ 국토교통, 안전, 정보통신기술, 건설과 철강산업 등의 분야에도 토픽 모델링을 적용하여 미래 핵심 기술과 이슈를 발견하고 트렌드를 분석하여 경제적, 사회적 부가가치를 창출하고 국가 전략 및 정책 수립 반영하는데에 활용
- ✓ 토픽을 도출하고 그것을 보면서 사회와 시대를 이해하고 이를 바탕으로 의사결정을 하고 계획을 세울 수 있는 분석이 토픽분석

잠재 의미 분석 (LSA)
Latent Semantic Analysis

잠재 의미 분석 (LSA, Latent Semantic Analysis)

- ✓ 잠재 의미 분석(LSA, Latent Semantic Analysis)은 토픽모델링 방법 중 하나
- ✓ BoW에 기반한 TDM이나 TF-IDF가 빈도로 단어 중요도를 판단하고 있어 의미를 고려하지 못한다는 단점이 있음
- ✓ LSA는 동일한 의미를 공유하는 단어들은 같은 텍스트에서 발생(co-occurrence)한다고 가정하는 벡터 기반 방법
- ✓ TDM 내에 잠재된(Latent) 의미를 이끌어 내는 방법



잠재 의미 분석 (LSA, Latent Semantic Analysis)

1. 대량의 텍스트 문서에서 발생하는 단어들 간의 연관관계를 분석함으로써 잠재적인 의미 구조를 도출
2. 문서 집합 내에서 연관성, 즉 동시출현(co-occurrence, 빈도)이 높은 단어들을 기준으로 유사한 문서를 추출
    a. co-occurrence 정보를 이용한다는 것은 단어의 '형태(morphology)가 아닌 의미(semantic)'를 이용한다는 뜻이다. 예를 들어 '배'라는 단어는 같은 문장에 co-occur 하는 동사가 '타다' 인지 '먹다' 인지에 따라 의미가 달라지게 된다.

단어	문서1	문서2	문서3
고양이	1	0	1
귀엽다	1	1	0
강아지	0	1	1
충성스럽다	0	1	0
애완동물	0	0	1

잠재 의미 분석 (LSA)의 수학적 의미

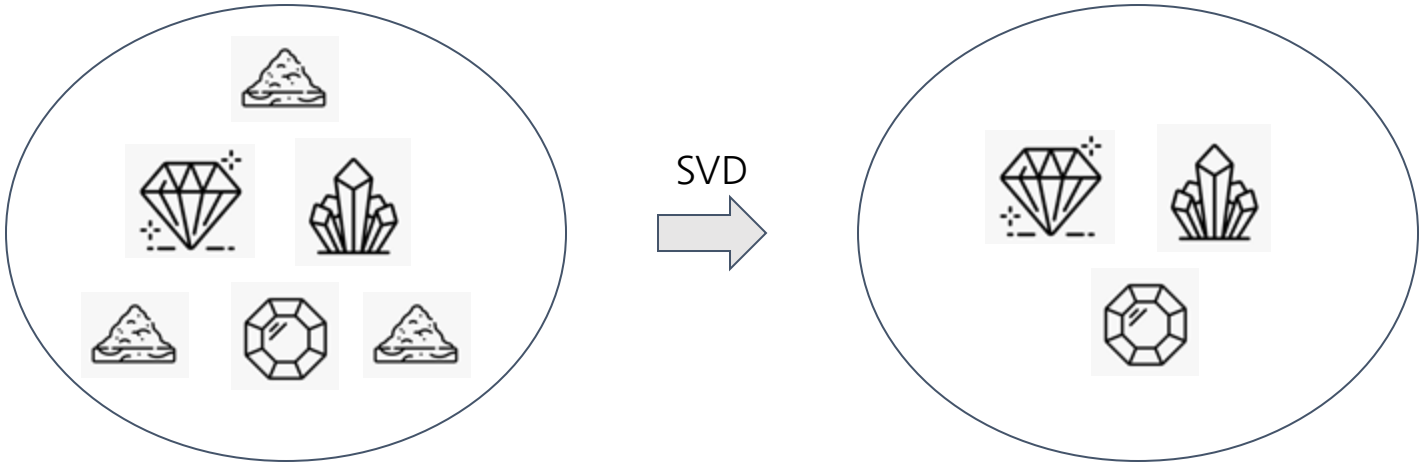
✓ LSA는 SVD를 이용해서 텍스트의 본질 즉 잠재 의미를 뽑아 내는 것이다.

1. LSA는 SVD를 이용해서 용어-문서 행렬을 더 간단한 세 행렬로 분해
2. 3개의 행렬을 곱하면 다시 원래의 행렬이 된다.
    - 단 3개의 행렬을 절단 후 다시 결합함으로써 문서를 표현하는 벡터 공간의 차원을 줄일 수 있다.
3. 다시 합쳐진 행렬은 원본 그대로가 아닌 "잠재 의미"를 담고 있는 행렬로 변한다.
    - 원래의 문서를 보다 더 잘 표현하게 된다.

잠재 의미 분석(LSA)의 수학적 의미

SVD (Singular Value Decomposition)

- ✓ SVD는 형태를 유지하고, 정보를 줄여준다.
- ✓ Singular Value Decomposition and the Fundamental Theorem of Linear Algebra
- ✓ 특이값 분해(SVD) - gaussian37



잠재 의미 분석(LSA)의 수학적 의미

SVD (Singular Value Decomposition)

- ✓ SVD는 형태를 유지하고, 정보를 줄여준다.
- ✓ Singular Value Decomposition and the Fundamental Theorem of Linear Algebra
- ✓ 특이값 분해(SVD) - gaussian37

```
[U,S,V]=svd(X)
```

$$X = U\Sigma V^T = \begin{bmatrix} | & | & | & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_m \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} [v_1 \quad v_2 \quad \cdots \quad v_m]^T$$

잠재 의미 분석(LSA)의 수학적 의미

SVD (Singular Value Decomposition)

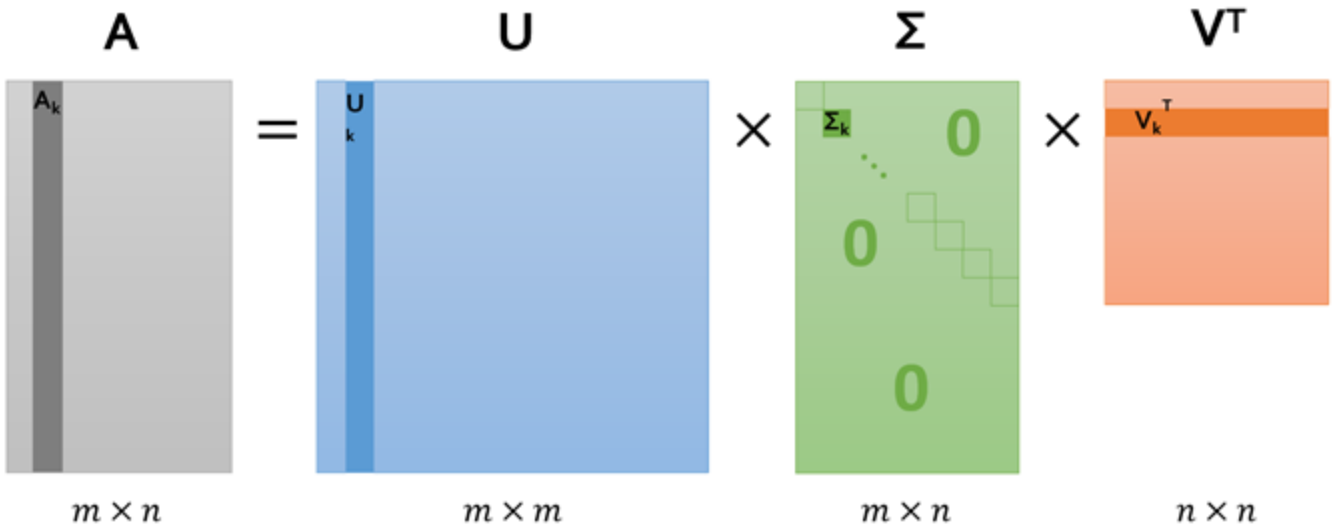
- ✓ SVD는 형태를 유지하고, 정보를 줄여준다.
- ✓ Singular Value Decomposition and the Fundamental Theorem of Linear Algebra
- ✓ 특이값 분해(SVD) - gaussian37

$$X = U\Sigma V^T = \begin{bmatrix} | & | & | & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_m \end{bmatrix} [v_1 \quad v_2 \quad \cdots \quad v_m]^T$$

잠재 의미 분석(LSA)의 수학적 의미

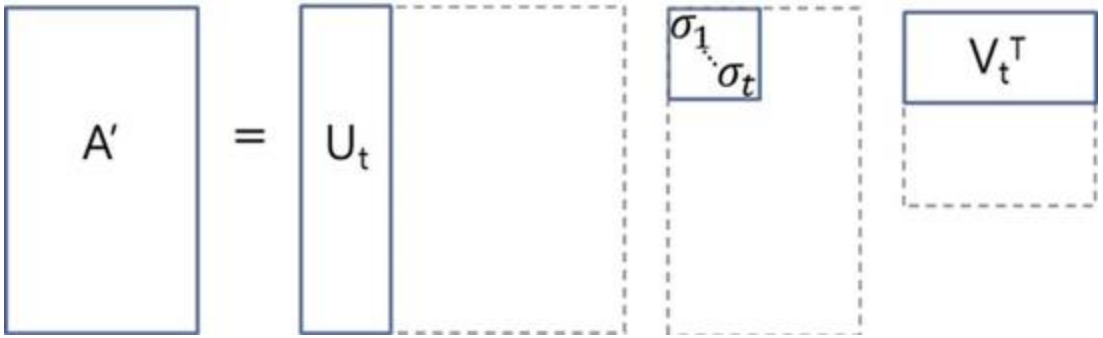
✓ A(원본)은 3개의 행렬로 나누어질 수 있다 -> 3개의 행렬을 이용하여 다시 원본을 만들 수 있음

✓ 이것을 Full SVD라고 부름



잠재 의미 분석(LSA)의 수학적 의미

✓ LSA는 원본의 정보를 근사하여 표현하는 것이 목표이므로 3개의 행렬의 차원을 줄여서 원본 정보와 유사한 A′ 을 만드는 것



잠재 의미 분석 (LSA)

✓ 토픽을 추출된 키워드들의 분포로 나타냄으로써 텍스트 내의 구조를 파악

문서-단어행렬
m x n = 6 x 4

좌특이벡터 (문서벡터)
m x m = 6 x 6

특이값
m x n = 6 x 4

우특이벡터 (단어벡터)
n x n = 4 x 4

잠재 의미 분석 (LSA)

✓ 토픽을 추출된 키워드들의 분포로 나타냄으로써 텍스트 내의 구조를 파악

문서-단어행렬
m x n = 6 x 4

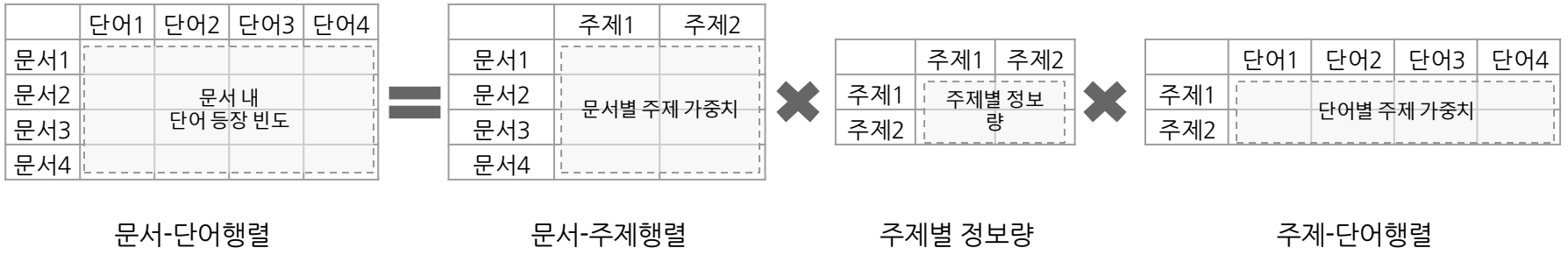
좌특이벡터 (문서벡터)
m x m = 6 x 2

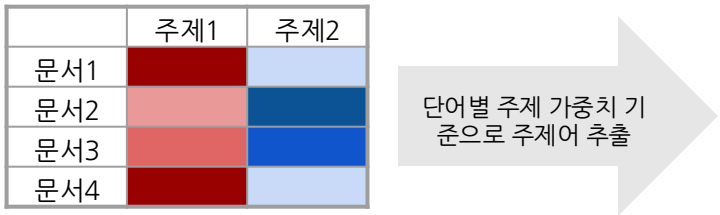
특이값
m x n = 2 x 2

우특이벡터 (단어벡터)
n x n = 2 x 4

잠재 의미 분석 (LSA)

✓ 토픽을 추출된 키워드들의 분포로 나타냄으로써 텍스트 내의 구조를 파악





잠재 의미 분석 (LSA)

- ✓ DTM (문서-단어 행렬)은 sparase 하다.
- ✓ LSA를 활용하여 의미를 보존하며 밀집벡터(Dense Vector)를 생성할 수 있다.

잠재 의미 분석 (LSA)

- ✓ U 는 토픽별 문서의 벡터를 의미함
- ✓ U 는 잠재 의미에 기반한 문서 벡터
- ✓ 문서 벡터간 코사인 유사도를 측정하여 유사도 측정이 가능

잠재 의미 분석 (LSA)

- ✓ V^T 는 k차원의 단어 벡터를 의미함
- ✓ V^T 는 잠재 의미에 기반한 단어 벡터
- ✓ 단어 벡터간 코사인 유사도를 측정하여 유사도 측정이 가능

	단어1	단어2	단어3	단어4
문서1	문서 내 단어 등장 빈도			
문서2				
문서3				
문서4				

=

	차원1	차원2
문서1	문서별 주제 가중치	
문서2		
문서3		
문서4		

×

	차원1	차원2
차원1	특이값	
차원2		

×

	단어1	단어2	단어3	단어4
차원1	단어별 주제 가중치			
차원2				

문서-단어행렬

단어벡터행렬

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

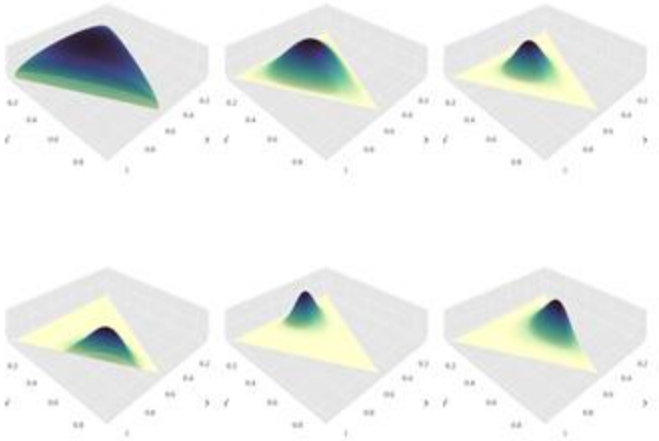
잠재 의미 분석 (LSA)


토픽모델링
(Topic modeling)
문서(Document) / 단어(Term)
DTM 생성 (문서-단어 행렬)
특이값 분해 (SVD)
토픽모델링 (n개 토픽)
예. 5개 토픽모델링
벡터 활용
단어간 유사도 분석
단어 벡터(V^T 벡터)간 유사도 분석
문서간 유사도분석
문서 벡터(U 벡터)간 유사도 분석
문서-단어간 유사도분석
문서-단어간 유사도분석

잠재 디리클레 할당 (LDA)
Latent Dirichlet Allocation

잠재 디리클레 할당 (LDA)

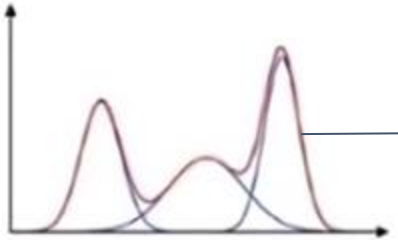
- ✓ 잠재(Latent) : 우리가 볼 수 있는 것은 문서 내의 단어들 뿐 토픽은 알 수 없음. 이 잠재적인 토픽을 찾는 것이 목표
- ✓ 디리클레(Dirichlet) : 문서 내 토픽 비율(토픽 분포)과 토픽 내 단어 비율(단어 분포)을 생성하는데 사용되는 확률 분포
- ✓ 할당(Allocation) : 단어들이 어떤 토픽에 속하는지를 결정하는 과정



디리클레 분포 - 위키백과

잠재 디리클레 할당 (LDA) - Dirichlet Distribution

- ✓ 디리클레 분포는 다항 분포들의 분포, 즉 여러 개의 다항 분포들이 합쳐진 분포
- ✓ k=3, 즉 Topic이 3개에 대한 확률 분포를 디리클레 분포로부터 가져온다



(0.7, 0.15, 0.15)

토픽에서의 단어 빈도
3개의 확률 분포 즉 3개의
토픽에서의 단어 분포

문서 내 토픽의 분포
토픽1: 0.7
토픽2: 0.15
토픽3: 0.15

[논문리뷰] 잠재 디리클레 할당 - LDA(Latent Dirichlet Allocation)

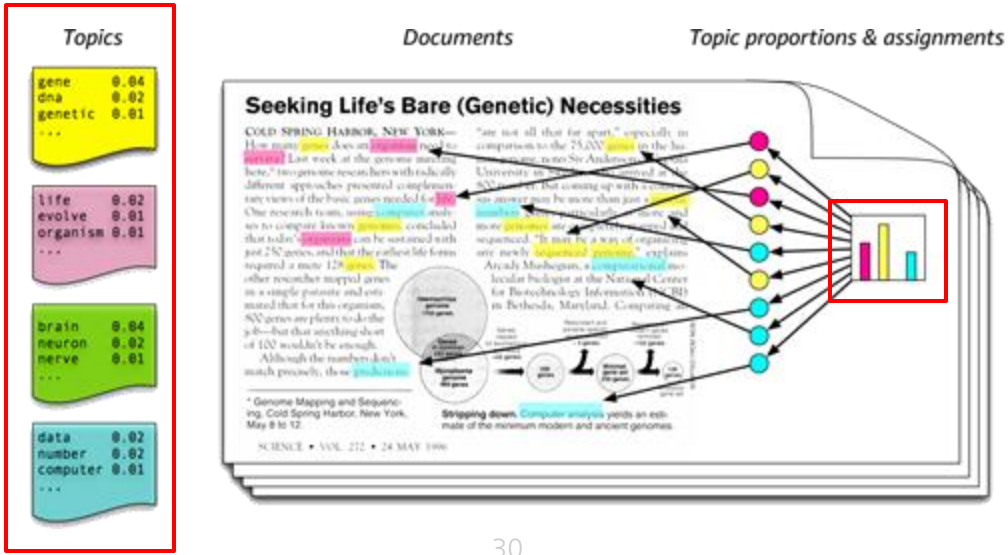
잠재 디리클레 할당 (LDA)

- ✓ LDA는 2가지 파트로 진행된다.
- ✓ Generative / Inference
  - Generative
    - 문서가 어떻게 생성되는가?
  - Inference
    - 주어진 문서 집합으로부터 LDA 모델을 학습하는 과정
  - Genrative는 가정으로 이러한 가정으로 Inference를 통해 파라미터를 찾고 토픽을 찾는 것이 바로 LDA 모델링

잠재 디리클레 할당 (LDA)

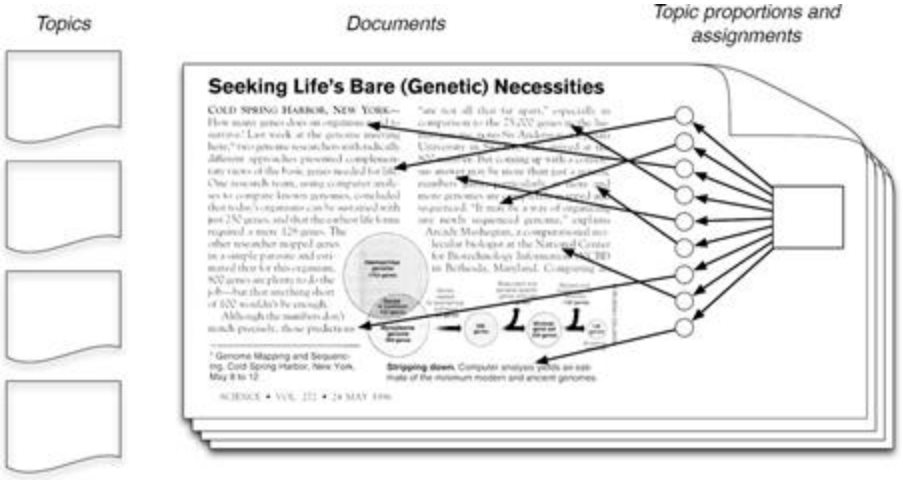
✓ 잠재 디리클레 할당(LDA, Latent Dirichlet Allocation)란 주어진 문서에 대해 어떤 주제가 존재하는지에 대한 확률모형 (토픽모델링)

- 각각의 토픽들은 단어들의 분포(빈도)
- 각각의 문서에서의 토픽들의 비중은 변하지 않는다. (mixture of corpus-wide topics)
- 각각의 단어들은 토픽에서 샘플링을 통해서 생성된다.



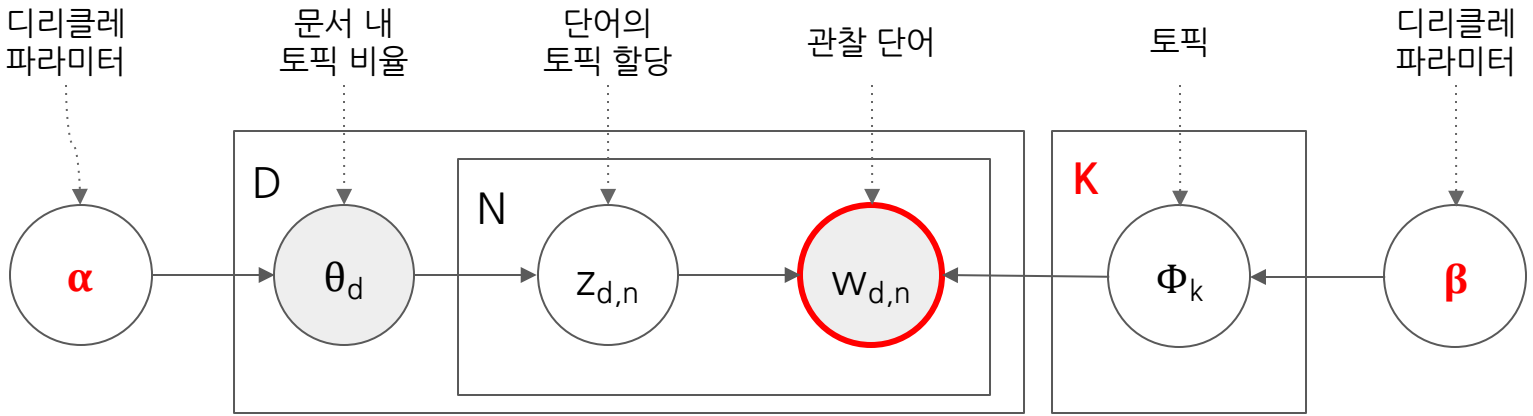
잠재 디리클레 할당 (LDA)

- ✓ 하지만 우리가 보는 입장에서 **토픽이 무엇인지**, **토픽의 분포는 어떻게 생겼는지**, **문서의 단어가 어떤 토픽에서 뽑혀서 결정되는지** 우리는 알 수 없다.
  - p(topics, proportions, assignments | documents)
- ✓ 실질적으로 우리는 문서의 단어들만 보고있다. 그렇기 때문에 역으로 **문서가 생성되는 과정을 유추(Generation 가정)**하고 **이러한 가정을 통해서 LDA를 학습(Inference)하는 과정**이 필요.



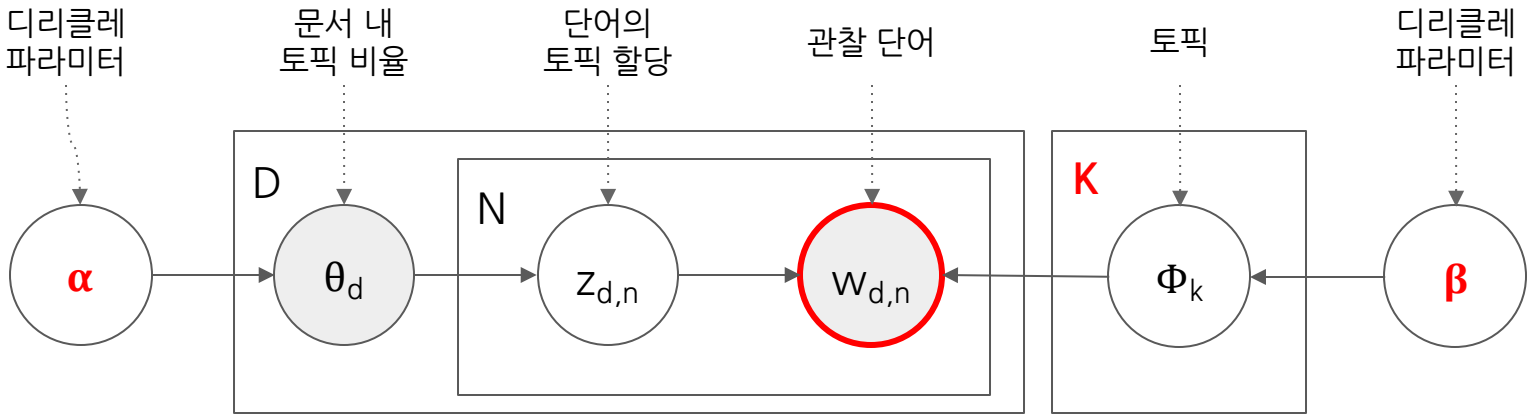
잠재 디리클레 할당 (LDA) - Generation

α	디리클레 파라미터 (보통 0.1)	D	전체 문서 개수
θ_d	문서 내 토픽 비율	Φ_k	K번째 토픽
$Z_{d,n}$	단어의 토픽 할당	K	토픽수
$W_{d,n}$	관찰단어	β	토픽 하이퍼파라미터 (보통 0.001)
N	N은 d번째 문서의 단어 수		



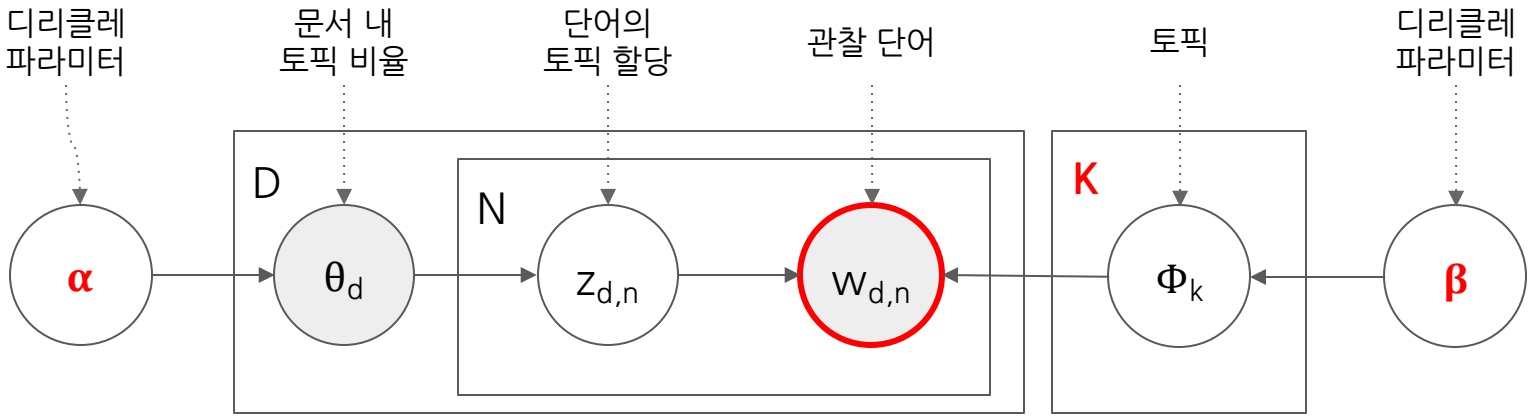
잠재 디리클레 할당 (LDA) - Generation

- ✓ 관찰 가능한 변수는 d번째 문서에 등장한 n번째 단어 $w_{d,n}$가 유일
- ✓ 이 정보를 가지고 하이퍼파라미터(사용자 지정) α,β를 제외한 모든 잠재 변수를 추정
- ✓ 사전에 결정해주어야할 값은 α, β, K값
- ✓ 보통 **α은 0.1, β는 0.001로 사용** 주제를 '할당'한 뒤 그 주제로부터 단어를 추출.



잠재 디리클레 할당 (LDA) - Generation

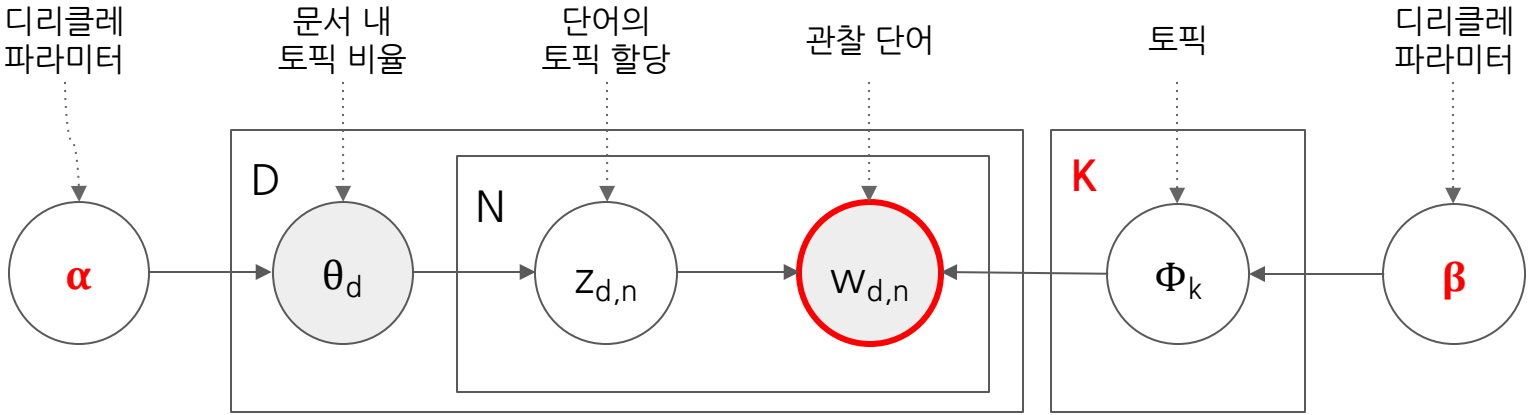
- ✓ α는 문서가 얼만큼의 토픽 비중을 가지고 있는가를 결정(θ_d)
  - θ_d는 d번째 문서에서 n번째 단어가 할당되었는가($z_{d,n}$)을 결정
    - 즉 어떤 토픽에서 왔는가를 결정
- ✓ β는 각각의 토픽들에서 단어들이 얼마만큼의 비중을 가지고 있는가를 결정(Φ_k)
- ✓ 최종적으로 $w_{d,n}$ 은 어떤 토픽에서 왔고, 토픽에서의 단어 비중을 통해서 단어가 결정



잠재 디리클레 할당 (LDA) - Generation

✓ 결국 어떤 토픽에서 단어들이 나왔는지를 가장 잘 설명하는 확률 분포를 찾는다!

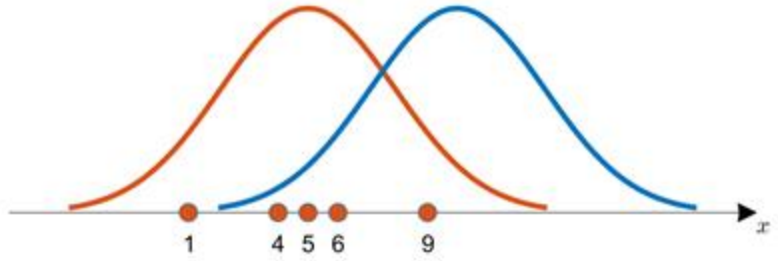
- 해당 수식의 확률을 최대화 하는 것이 목표!



$$p(\phi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\phi_i|\beta) \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\phi_{1:K}, z_{d,n}) \right)$$

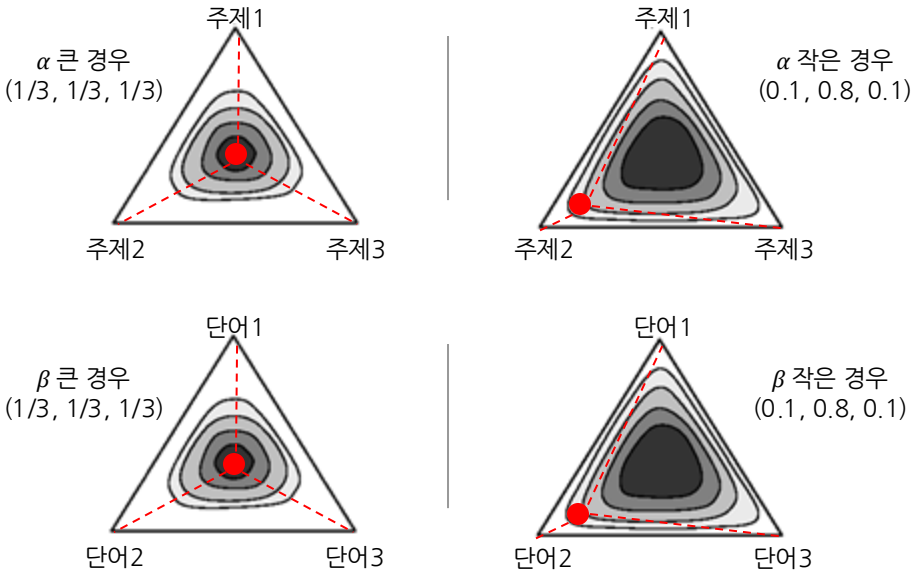
잠재 디리클레 할당 (LDA) - Generation

- ✓ 결국 어떤 토픽에서 단어들이 나왔는지를 가장 잘 설명하는 확률 분포를 찾는다!
  - 해당 수식의 확률을 최대화 하는 것이 목표!
- ✓ 데이터 1,4,5,6,9를 가장 잘 설명하는 확률분포는 확률 값이 크다.
- ✓ 따라서 가장 데이터를 잘 설명하는 확률 분포는 확률의 값이 크다고 볼 수 있다.



잠재 디리클레 할당 (LDA) - Generation

- ✓ θ는 Mixture Model의 가중치 역할
- ✓ α는 문서의 토픽 분포 생성을 위한 파라미터
- ✓ β는 토픽의 단어 분포 생성을 위한 파라미터
- ✓ α가 크면 모든 토픽이 뽑힐 확률이 균등한 다항 분포가 뽑힌다.
- ✓ α가 작아지면 특정 토픽이 뽑힐 확률이 높아지는 다항 분포가 뽑힌다.
- ✓ 정리하자면
  - **α가 낮다는 건 문서가 특정 주제에 집중**
  - **β가 낮다는 건 토픽이 특정 단어에 집중**



2-Simplex

잠재 디리클레 할당 (LDA) - Generation

✓ Generation Model의 요약

1. **토픽 비율 선택**
   a. 먼저 문서가 여러 주제(토픽)로 구성되어 있다고 가정하고, 문서에서 각 주제가 차지하는 비율을 확률적으로 정함. (**문서의 토픽 비율 θ_d**)
2. **토픽별 단어 비중 설정**
   a. 각 주제(토픽)에서 어떤 단어들이 많이 나오는지, 주제별 단어 비중을 확률적으로 설정. (**토픽의 단어 분포 ϕ_k**)
3. **각 단어 선택**
   4. **토픽 선택**:
      - 문서의 토픽 비율 θ_d에 따라, 각 단어 $w_{d,n}$가 어떤 토픽 $z_{d,n}$에서 나올지 확률적으로 선택
   5. **단어 선택**:
      - 선택된 토픽 $z_{d,n}$에서 해당 토픽의 단어 분포 ϕ_k를 기반으로 단어 $w_{d,n}$를 확률적으로 선택

Inference - Gibbs Sampling

✓ 단어들에 대한 결합 분포를 계산하는 것은 현실적으로 불가능

1. 문서 내 단어 수가 많아 고차원임
2. 모든 단어와 토픽의 가능한 조합을 다 계산하기에는 계산 비용이 너무 큼

✓ 따라서 근사치를 내기 위한 과정이 Inference

✓ Approximate한 것을 Sampling을 하여 사용

✓ 보통 Gibbs Sampling을 사용

1. 단어들에 Topic을 Random하게 할당
2. Sampling
3. Assign
4. 2와 3의 과정을 반복

LDA - Gibbs Sampling

- 문서내에 존재하는 단어
    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

LDA - Gibbs Sampling

✓ 불용어를 제거한 단어들을 남기고 각각의 단어의 임의의 주제를 배정 (α=0.1, β=0.001)

✓ 문헌별 주제 분포, 주제별 단어 분포를 계산

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

θ	A	B	C	D	E	F
#1	1.1	2.1	0.1	2.1	2.1	2.1
#2	1.1	1.1	2.1	0.1	1.1	3.1

Φ	cute	kit	eat	rice	cate	ham	bre
#1	1.001	0.001	2.001	1.001	3.001	0.001	2.001
#2	1.001	2.001	1.001	1.001	0.001	2.001	1.001

LDA - Gibbs Sampling

✓ 1번 단어(cute)의 주제를 Masking 후 다시 분포를 계산

✓ 새로 계산된 분포를 가지고 1번 단어(cute)의 토픽을 다시 할당

토픽할당

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

문서 내 토픽 분포

θ	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
	1.2	3.2	2.2	2.2	3.2	5.2

토픽 내 단어 분포

	cute	kit	eat	rice	cake	ham	bre	합계
#1	0.001	0.001	2.001	1.001	3.001	0.001	2.001	8.007
#2	1.001	2.001	1.001	1.001	0.001	2.001	1.001	8.007

LDA - Gibbs Sampling

- ✓ 이제 2번 단어를 Masking 후 분포 를 다시 계산하고 토픽을 할당
- ✓ 이와같은 과정을 계속해서 반복

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

LDA - Gibbs Sampling

✓ 모든 반복 과정을 끝내면 모든 단어에 새로운 토픽이 할당된다.

토픽할당

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

문서 내 토픽 분포

θ	A	B	C	D	E	F
#1	0.1	3.1	0.1	2.1	3.1	3.1
#2	2.1	0.1	2.1	0.1	0.1	2.1
	1.1	3.2	2.2	2.2	3.2	5.2

토픽 내 단어 분포

Φ	cute	kit	eat	rice	cake	ham	bre	합계
#1	0.001	0.001	3.001	2.001	3.001	0.001	3.001	11.007
#2	2.001	2.001	0.001	0.001	0.001	2.001	0.001	6.007

잠재 디리클레 할당 (LDA) 가정

✓ 잠재 디리클레 할당 가정 : **단어 교환성(exchangeability)**

- BoW라고 표현하기도 함. 단어 교환성은 단어들 순서를 고려하지 않고 유무만 중요하다는 가정
- '나는 양념 치킨을 좋아해 하지만 후라이드 치킨을 싫어해', '나는 후라이드 치킨을 좋아해 하지만 양념 치킨을 싫어해' 간에 차이가 없다고 생각
- 단어 빈도수만으로 표현이 가능. 이를 기반으로 교환성을 포함하는 모형을 제시한 것이 LDA 모델

✓ 단순히 단어 하나를 단위로 생각하는 것이 아니라 단어 묶음을 한 단위로 생각하는 **n-gram방식으로 LDA의 교환성 가정을 확장**시킬 수도 있음

잠재 디리클레 할당 (LDA) 한계

✓ 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

✓ 하이퍼 파라미터 설정의 어려움
- 토픽의 수 K값을 얼마로 두는게 적절한지 모름
- 적절한 K값을 설정하고 그에 따르는 α, β값을 잘 튜닝해야 좋은 결과를 얻을 수 있음

잠재 디리클레 할당 (LDA) 절차

1. D개의 전체 문서에 k개 토픽이 분포되어있다고 가정
2. 모든 단어에 k개 토픽 중 하나를 임의 할당
    1. 각 문서는 토픽을 가짐
    2. 토픽은 단어 분포를 가짐
3. 임의 할당 했지만 올바르게 할당되었다고 가정
4. 다음 과정을 반복하여 토픽을 재할당
5. 안정적인 상태(결과가 수렴)까지 반복