6. Topic Modeling

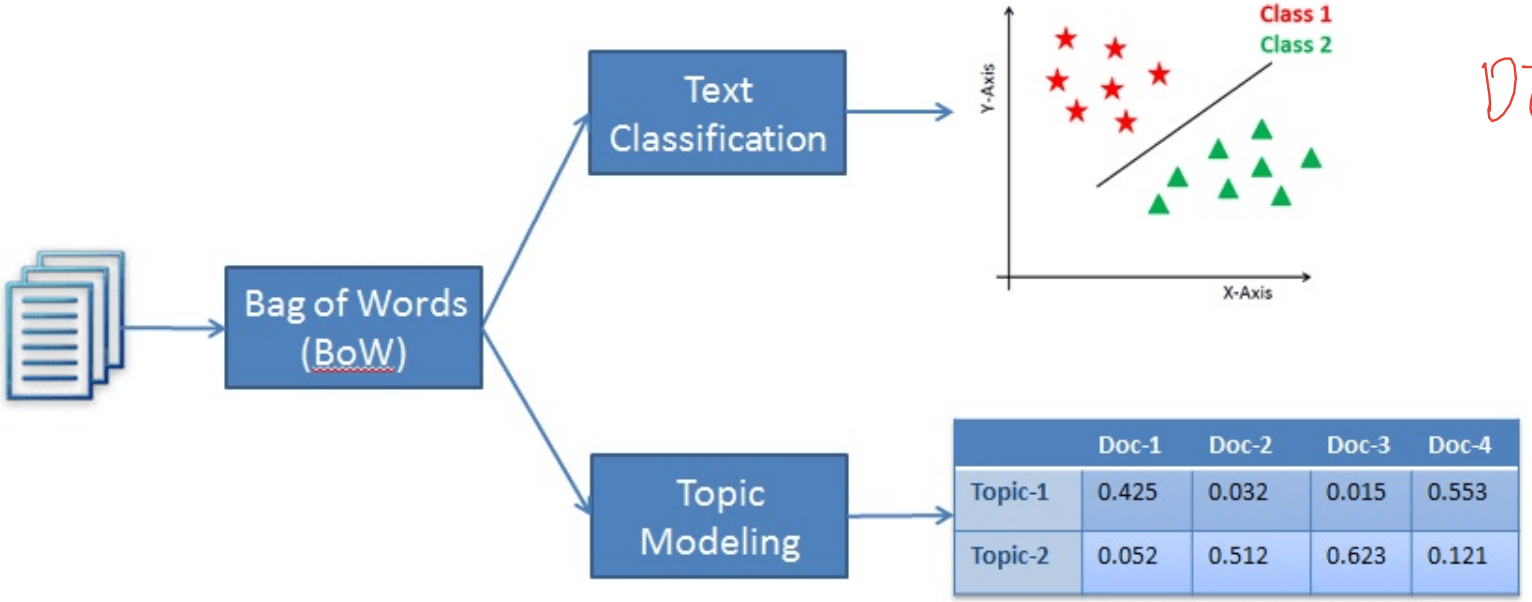
6장 내용

- **Topic model:** 문서의 주제를 찾아내서 문서들을 비교/분류하는 것
- **잠재 의미 분석(Latent Semantic Analysis):** SVD 방식을 이용하여 주제를 찾아냄 LSA 1990
- **잠재 디리클레 할당(Latent Dirichlet Allocation):** 확률 모형을 통해 주제를 찾아냄 LDA 002

Topic modeling

단어들 중에 중요 의미있는 단어들을 분석

- 토픽 모델링은 사용된 단어들을 분석하여 문서에 포함되어 있는 주제를 찾아내고자 함
- 규칙 기반 방식을 사용하지 않고 문서 분석 알고리즘을 사용
- 텍스트 분류와 유사하지만 토픽 모델링에서는 문서에 내재된 주제를 숫자로 나타냄

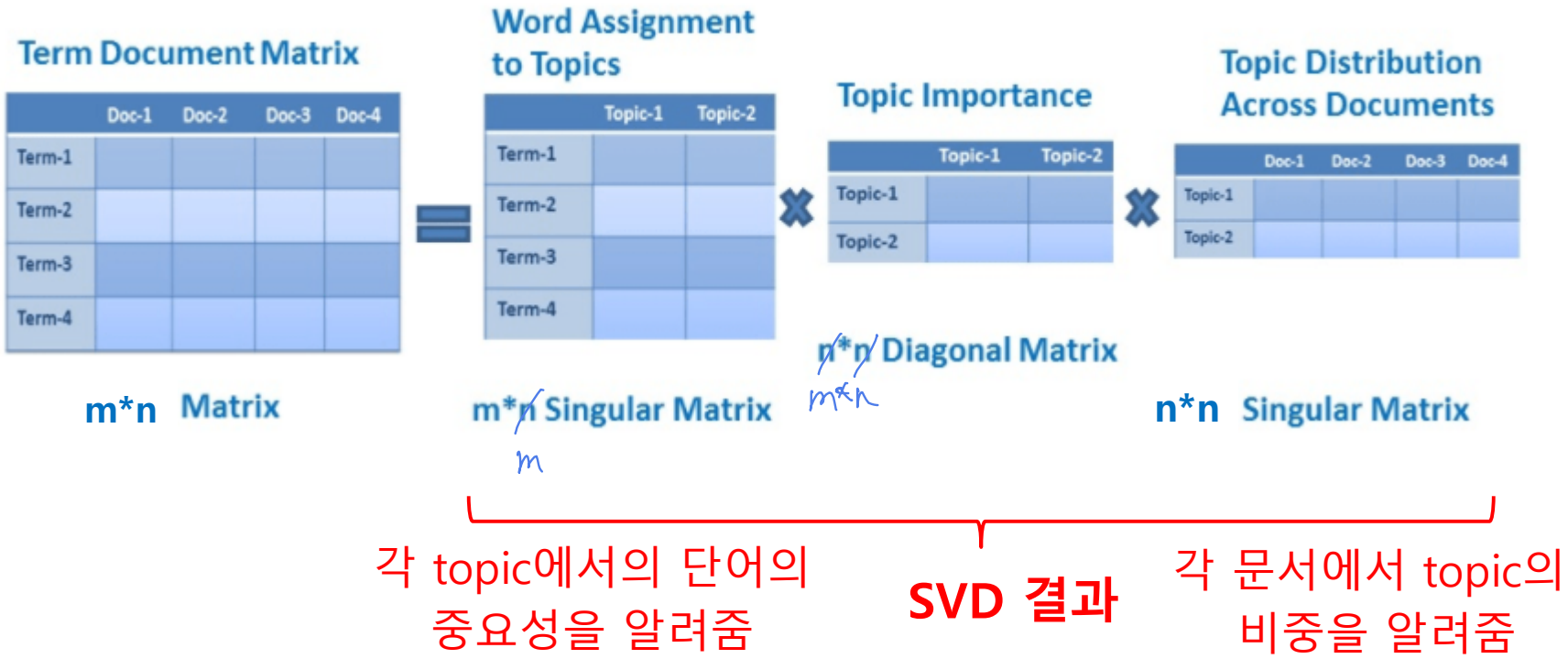


	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1	0.425	0.032	0.015	0.553
Topic-2	0.052	0.512	0.623	0.121

DTM

잠재 의미 분석(Latent Semantic Analysis)

- 여러 문서로부터 추출된 DTM(Document-term matrix)에 Singular Value Decomposition(SVD) 기법을 적용하여 주제를 찾아내는 방식
- DTM의 관심 영역을 축소하여 문서들에 내재된 주제를 찾아내는 것임



특이값 분해(Singular Value Decomposition)

- A가 $m \times n$ 행렬일 때 다음과 같이 3개의 행렬의 곱으로 분해하는 것

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

각 행렬은 다음 성질을 가짐

$U: m \times m$ 직교행렬 $(UU^T = I)$

$\Sigma: m \times n$ 직사각 대각행렬 대각원소의 나머지는 0

$V: n \times n$ 직교행렬 $(VV^T = I)$

*__직교행렬(orthogonal matrix):__ 한 행의 크기(norm)는 1이고, 서로 다른 행들간의 내적은 0임.

- SVD는 numpy와 같은 패키지를 이용하여 수행

특이값 분해 사례

- LSA에서는 문서-단어 행렬(DTM)을 행렬 A로 사용

$A =$

	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

- A는 다음과 같이 분해됨. 이들을 곱하면 다시 A가 됨

U

	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

$\times$

Σ

값이 커일품

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

값이 게일작음

$\times$

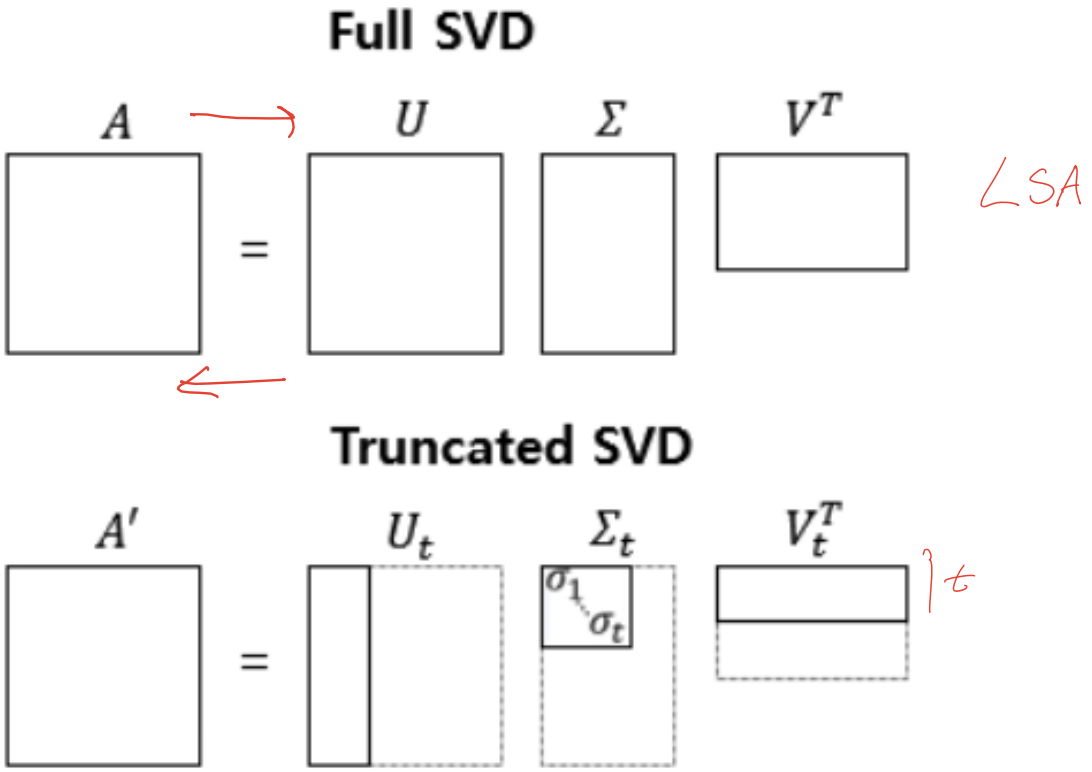
V^T

	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

- Σ의 원소들은 내림차순으로 배열되어 있음

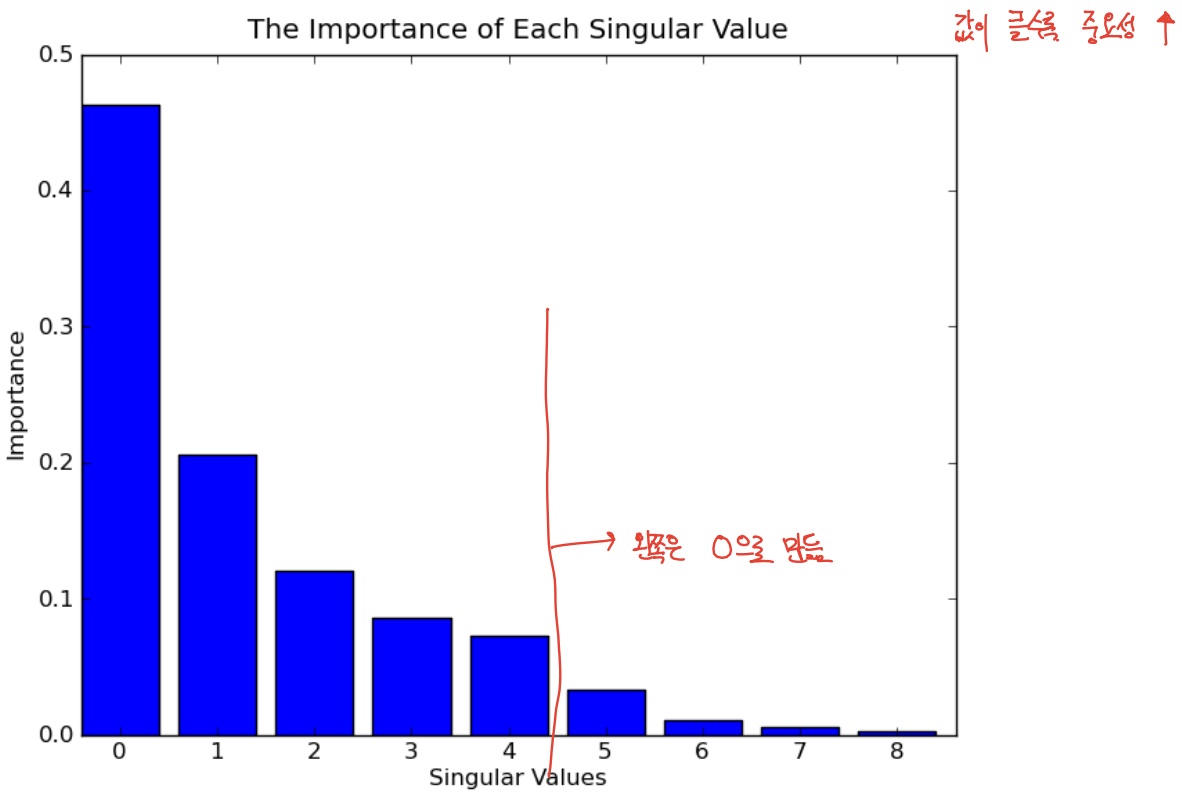
SVD 조절에 의한 차원 축소 (dimensionality reduction)

- Σ 행렬의 원소 숫자(topic 숫자)를 줄여서 DTM A 를 변형시킴
- 이와 같이 적은 수의 topic에 집중함으로써 A의 차원을 줄이게 됨



각 특이값의 중요도

- Σ 행렬의 원소값들은 해당 topic의 중요성을 반영함
- 상대적으로 낮은 값들을 0으로 만들면 덜 중요한 topic을 고려하지 않는 결과가 됨
- 아래 그림은 원소값들의 크기 사례를 보여주는 것임



LSA 처리 절차

numpy

1. 총 n개의 단어를 포함하고 있는 m개의 문서에서 $m \times n$ 크기의 DTM A를 생성. 필요한 경우 단어에서 불용어를 제거
2. A에 대해 SVD를 적용하여 $A = U\Sigma V^T$ 를 만족시키는 세 행렬을 구함
3. Σ의 원소 중 K개의 큰 값을 선정(주요 토픽을 선정하는 과정)
4. U와 V에서 K개의 행과 열 만을 남긴 U_t와 V_t 을 구함. U_t와 V_t에서는 각 문서에서 K개의 주제가 차지하는 비중, 그리고 각 주제에서 단어들이 차지하는 비중을 볼 수 있음

LSA 사례

- 문장 데이터

```
d0 = "He is a good dog."
d1 = "The dog is too lazy."
d2 = "That is a brown cat."
d3 = "The cat is very active."
d4 = "I have brown cat and dog.“
```

- 불용어들을 제거하여 문장들을 수정

	documents	clean_documents
0	He is a good dog.	good dog
1	The dog is too lazy.	dog lazy
2	That is a brown cat.	brown cat
3	The cat is very active.	cat active
4	I have brown cat and dog.	brown cat dog

LSA 사례(계속)

- SVD 수행 결과 (K = 2로 설정)

	documents	topic_1	topic_2
0	good dog	0.3413834191239959	0.7199781067501040
1	dog lazy	0.3413834191239958	0.7199781067501034
2	brown cat	0.8609490919302158	-0.3659836550739513
3	cat active	0.5166658991993210	-0.3850046207843266
4	brown cat dog	0.9494117370834857	0.0236302940661153

Document-topic matrix

0,1은 topic 2, 2~4는 topic 1에 가까움

UΣ(V^T)에 해당됨

	topic_1	topic_2
active	0.2003541259081117	-0.2424408501618364
brown	0.5965117122287049	-0.2018098984872574
cat	0.6293380994160956	-0.3298859088715313
dog	0.4158307960649448	0.6169033286639758
good	0.1323826028466488	0.4533766476433699
lazy	0.1323826028466494	0.4533766476433687

Term-topic matrix

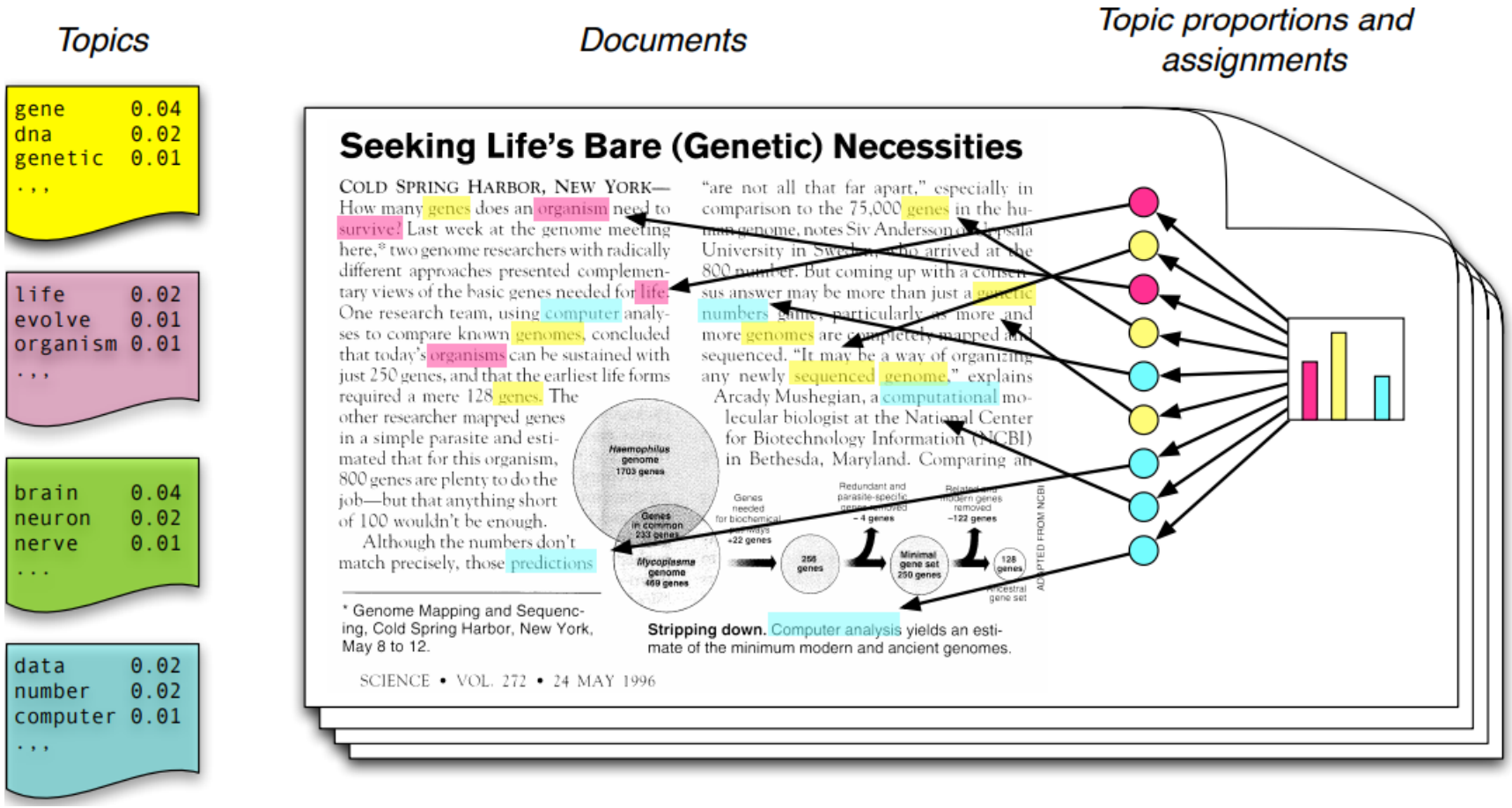
active, brown, cat은 topic 1, dog, good, lazy는 topic 2에 가까움

LSA 응용 분야

- LSA는 **차원 축소(dimensionality reduction)**에 사용됨. 단어가 10만 개, 문서가 10,000개 있는 경우 DTM 규모는 100,000x10,000 크기이지만 수백개 정도의 topic으로 축소할 수 있음
- LSA는 검색 엔진에 사용됨. 이 경우 **Latent Semantic Indexing(LSI)**이라는 명칭도 사용
- **문서 군집화(clustering)**에 사용될 수 있음

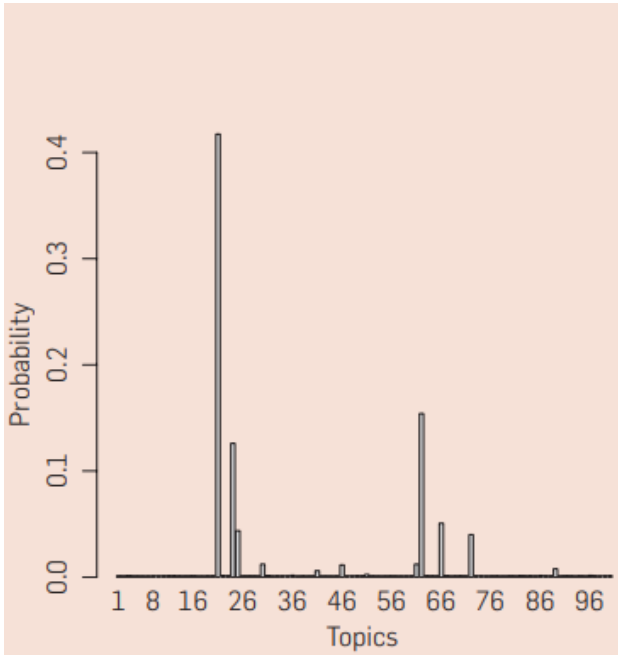
잠재 디리클레 할당(LDA)

- 문서(신문 기사, 논문 등)의 주제와 주제 단어들을 찾음



LDA 분석 사례

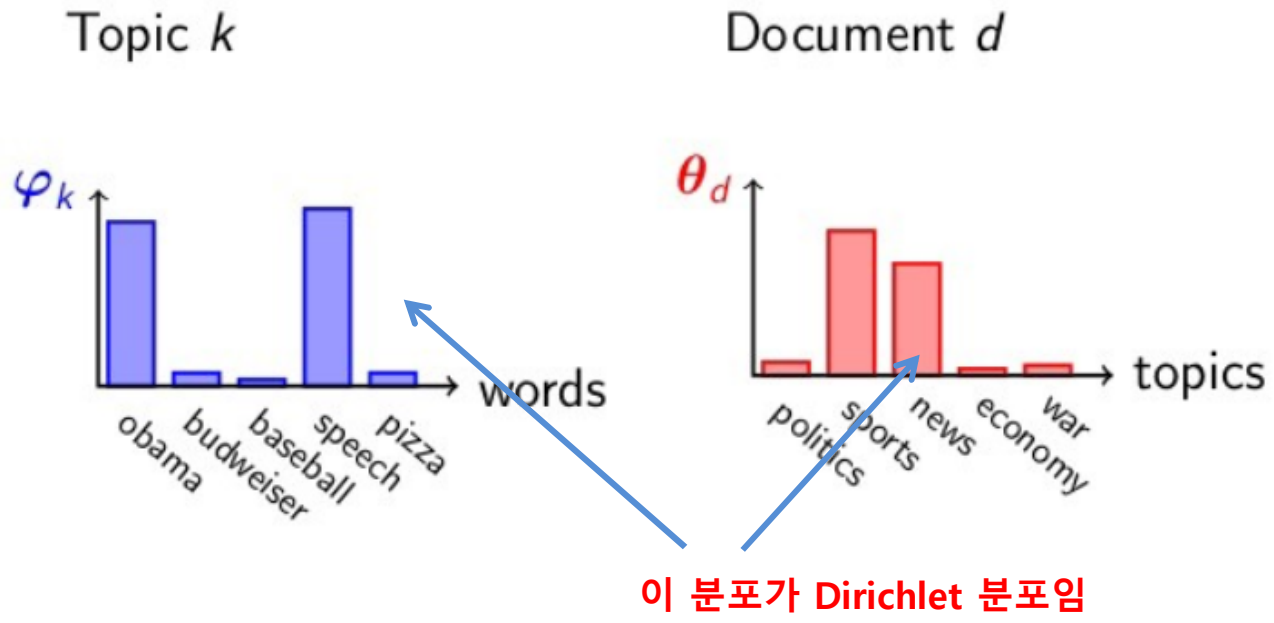
- Science 지의 17,000개의 기사 분석과 각 topic 별 15개 핵심 단어



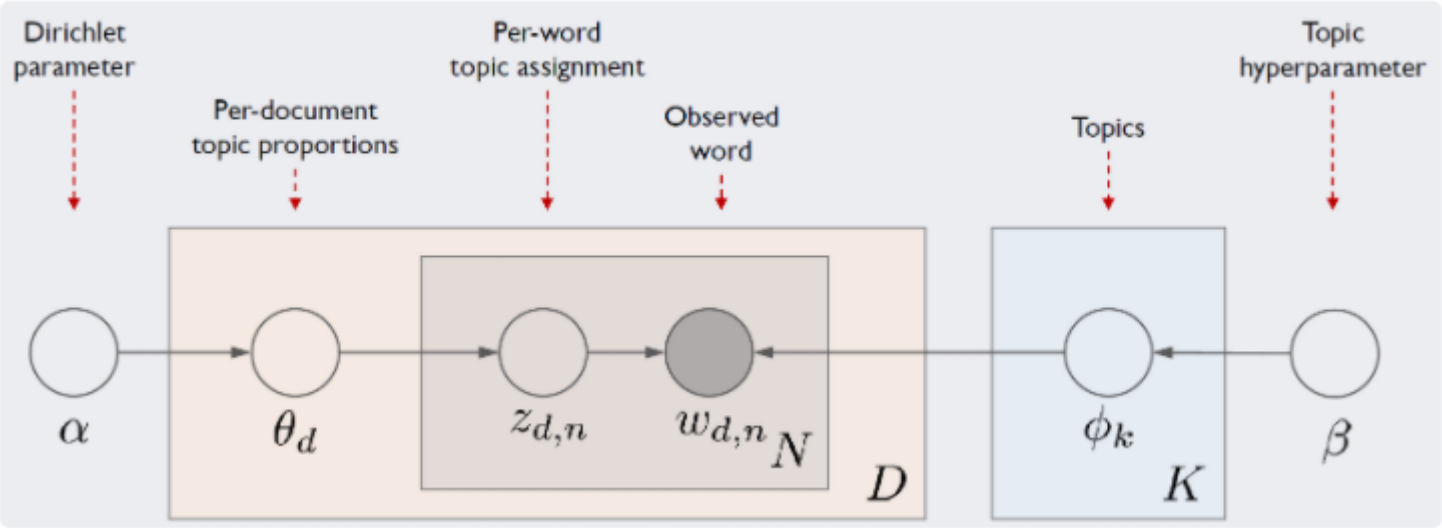
"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

LDA 분석 방법

- **입력:** 다수의 문서(기사, 논문, 웹 문서 등등)에서 추출된 Document-term matrix
- **전제 조건:** Topic의 숫자(예: 100개 등)
- **분석 방법:** Gibbs sampling과 같은 통계적 분석 기법으로 문서별 topic 분포와 각 topic의 주요 단어들을 찾아 냄



LDA Model



D: 문서의 개수

N: d번째 문서의 단어 수

K: Topic 수

θ_d: d 문서에서의 topic 분포(Dirichlet 분포)

φ_k: K topic에서의 단어 분포(Dirichlet 분포)

$w_{d,n}$은 document-term matrix로 관찰가능한 유일한 항목

목표: $w_{d,n}$으로부터 K개의 φ_k와 D개의 θ_d를 찾아 냄

LDA 수행 절차

- 확률적 모델링(probabilistic modeling) 방식을 사용
- 수행 절차는 다소 복잡함
- 데이터 분석 모듈에 기능이 제공되고 있음

- Original 논문: D. Blei, A. Ng, M. Jordan, Latent Dirichlet allocation, *J. Mach. Learn. Res.* 3, 993–1022, 2003.
- Review 논문: D. Blei, Probabilistic topic models, *Communications of ACM*, 2012.

LDA 수행 하기

1. 사용자는 토픽의 개수 k를 지정
2. 모든 단어를 k개 중 하나의 토픽에 임의로 할당. 한 단어가 문서에서 2회 이상 등장한 경우 각 단어는 다른 토픽에 할당될 수도 있음
3. 모든 문서의 모든 단어에 대해 다음 사항을 반복 진행

3-1. 문서의 각 단어 w는 자신은 잘못된 토픽에 할당되어 있지만, 다른 단어들은 전부 올바른 토픽에 할당되어 있는 상태라고 가정. 단어 w는 조건부 확률 계산에 의해 토픽이 재할당됨