텍스트 마이닝과 데이터 마이닝

Part 05. 토픽 모델링과 워드 클라우드

정 정 민

programmers

Chapter 12. 토픽 모델링 대표 모델

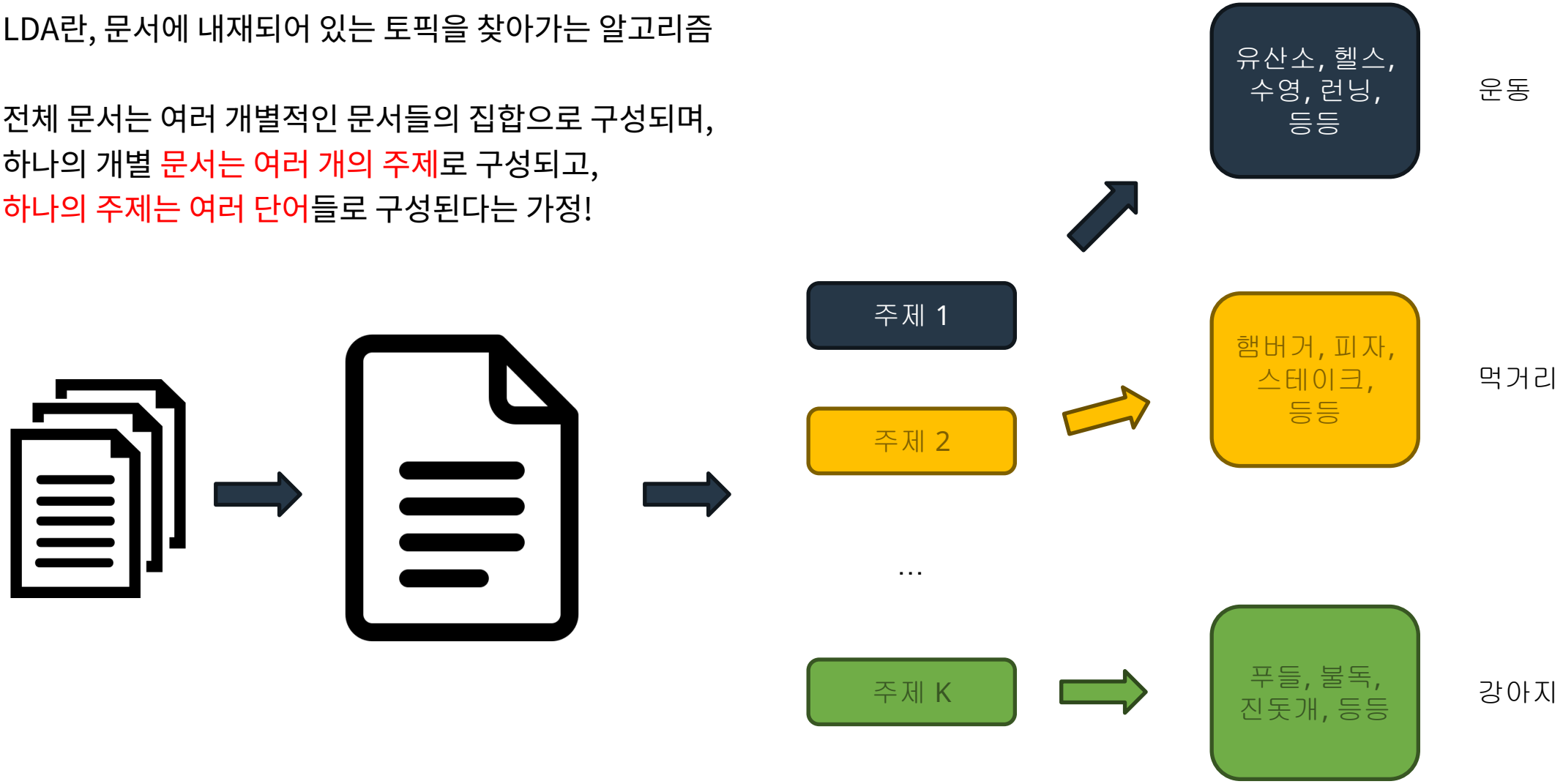
1. LDA 기본 과정
2. LDA 알고리즘

LDA 기본 과정

LDA의 기본 가정

- LDA란, 문서에 내재되어 있는 토픽을 찾아가는 알고리즘

- 전체 문서는 여러 개별적인 문서들의 집합으로 구성되며,
- 하나의 개별 문서는 여러 개의 주제로 구성되고,
- 하나의 주제는 여러 단어들로 구성된다는 가정!



LDA 예시 문장으로 확인하기

- 아래와 같은 예시 문장

문서 1 : 우리 부모님은 건강을 위해 아침마다 수영을 하시고 저녁에는 산책을 합니다.
문서 2 : 나와 동생은 햄버거를 좋아합니다. 특히 치킨이 들어간 햄버거를 좋아하고, 어제는 피자를 먹었습니다.
문서 3 : 오늘은 나의 생일이라 햄버거를 먹었습니다. 그런데 살이 너무 많이 쪄서 산책과 수영을 시작했습니다.

- 문서 내에 몇 개의 토픽이 있을지는 사용자가 정의
  - 위 예에서는 편의상 2개의 토픽(A, B)가 있다고 가정

- LDA를 진행하면
  - 문서 1은 A 토픽이 존재하며, 그 단어는 빨간색으로 표시
  - 문서 2는 B 토픽이 존재하고, 파란색으로 표시
  - 문서 3은 A와 B 토픽이 둘 다 존재
  - 할당된 단어를 확인할 결과 A는 '운동' B는 '먹거리'로 토픽의 이름을 정할 수 있음

LDA 조금 더 구체적으로

문서 1 : 우리 부모님은 건강을 위해 아침마다 수영을 하시고 저녁에는 산책을 합니다.
문서 2 : 나와 동생은 햄버거를 좋아합니다. 특히 치킨이 들어간 햄버거를 좋아하고, 어제는 피자를 먹었습니다.
문서 3 : 오늘은 나의 생일이라 햄버거를 먹었습니다. 그런데 살이 너무 많이 쪄서 산책과 수영을 시작했습니다.

- 각 문서를 구성하는 토픽의 구성을 보면
  - **문서 1** : 100% 토픽 A / **문서 2** : 100% 토픽 B / **문서 3** : 67% 토픽 A & 33% 토픽 B

- 또한, 각 토픽을 구성하는 단어를 보면
  - 토픽 A : **건강** (20%) / **수영** (40%) / **산책** (40%)
  - 토픽 B : **햄버거** (60%) / **치킨** (20%) / **피자** (20%)

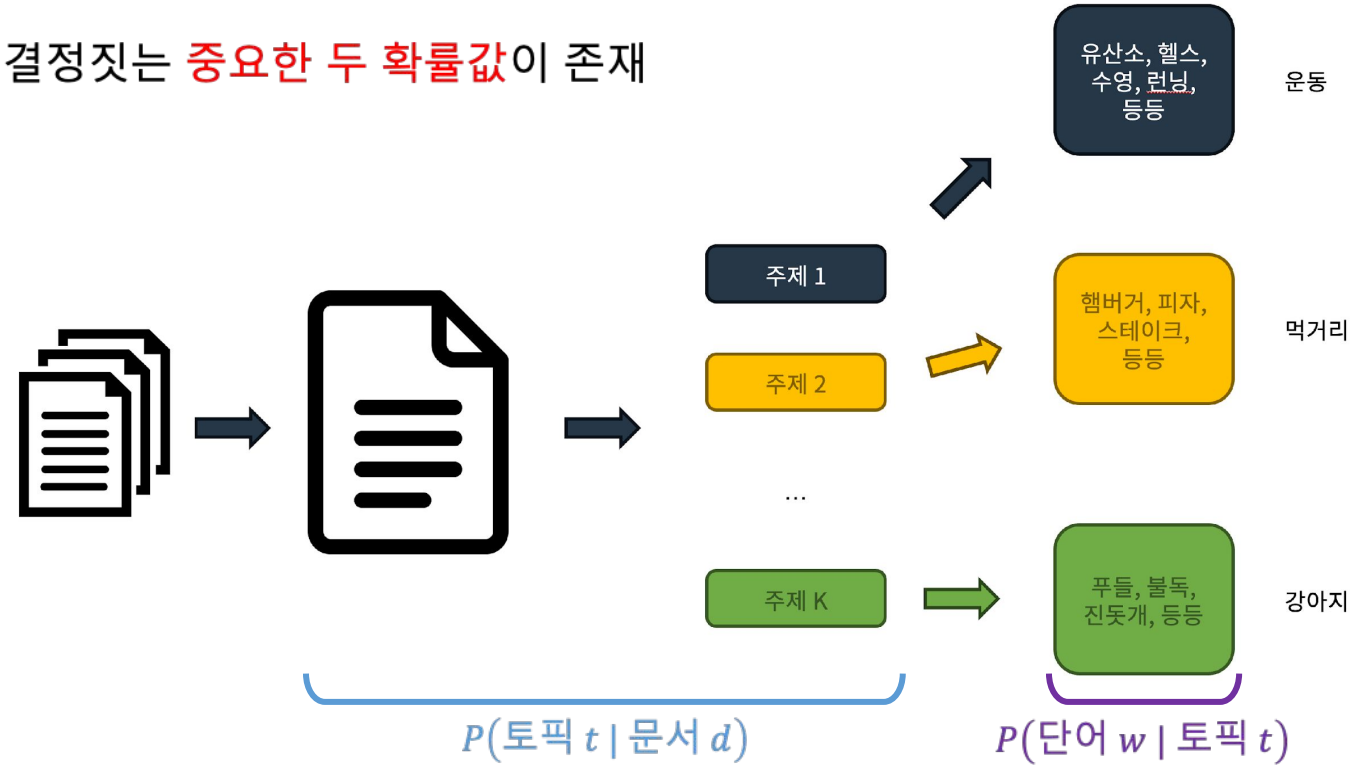
- 이렇듯 확률이나 비율의 집합을 분포로 표현한 것을 Dirichlet 분포라고 함
- LDA는 문서 표면에 드러나지 않은 숨어있는 토픽의 확률 분포(Latent Dirichlet)를 가정하고 각 단어를 토픽에 할당(Allocation)하는 분석 방법
- LDA : Latent Dirichlet Allocation

LDA 알고리즘

programmers

LDA의 중요한 두 확률값

- LDA 알고리즘을 결정짓는 중요한 두 확률값이 존재



1. 문서에 어떤 토픽이 들어있는가
- 이를 $P(\text{토픽 } t \mid \text{문서 } d)$ 라고 함

2. 각 토픽에 어떤 단어가 들어있는가
- 이를 $P(\text{단어 } w \mid \text{토픽 } t)$ 라고 함

확률 값의 의미

- $P(\text{토픽 } t \mid \text{문서 } d)$
  - 특정 문서 d에서 토픽 t가 차지하는 비율
  - 문서에서 각 토픽이 얼마나 중요한지를 나타냄

- $P(\text{단어 } w \mid \text{토픽 } t)$
  - 특정 토픽 t에서 단어 w가 차지하는 비율
  - 토픽에 특정 단어가 나타낼 확률

- $P(\text{토픽 } t \mid \text{문서 } d, \text{단어 } w)$
  - 특정 단어가 어떤 문서의 주제에 속할 확률
    - 즉, 어떤 단어가 문서의 주제와 얼마나 잘 맞는지를 나타냄
    - 이 값이 크면, 특정 단어가 그 문서의 주제와 매우 밀접한 관련이 있음을 의미
  - LDA에서 최종적으로 유추해야하는 값이지만 직접적으로 구하기가 어려움
  - $\propto P(\text{토픽 } t \mid \text{문서 } d) \times P(\text{단어 } w \mid \text{토픽 } t)$

알고리즘 적용 과정

1. 토픽 개수 K 설정 (사용자의 몫)

2. 문서 내 모든 단어에 무작위로 K 토픽 중 하나를 할당

3. 단어 w의 토픽 할당을 결정하기 위해 나머지 단어들의 할당 결과를 활용 $P(\text{토픽 } t \mid \text{문서 } d) \times P(\text{단어 } w \mid \text{토픽 } t)$ 계산
   - 이 값이 제일 커지는 t를 w에 재 할당
   - 전체 문서의 모든 단어들을 대상으로 연산 진행
   - 종료 시점에 도달할 때까지 반복 진행
     - w에 할당된 t의 변화가 없는 시점까지
     - 정해진 업데이트 횟수 도달까지
     - 등등

4. 최종 결과 분석
   - 토픽에 존재하는 단어를 보고 토픽이 의미하는 주제를 사용자가 정의 (토픽 1은 '먹거리'구나!)
   - 할당된 토픽을 기준으로 문서에 존재하는 토픽을 분석 (문서 1은 토픽이 2,6,8이 있네!)

알고리즘 적용 예시 (문서 3개, K=2(A, B)로 설정!)

초기 임의 토픽 설정

문서 1 내 단어	
건강	A
수영	A
산책	A

문서 2 내 단어	
햄버거	A
치킨	A
햄버거	B
피자	B

문서 3 내 단어	
햄버거	B
산책	B
수영	B

P(토픽 t | 문서 d) 계산

문서 1 내 단어	
건강	A
수영	A
산책	A

문서 2 내 단어	
햄버거	???
치킨	A
햄버거	B
피자	B

문서 3 내 단어	
햄버거	B
산책	B
수영	B

- $P(\text{토픽 } A \mid \text{문서 } 2) = \frac{1}{3}$
- $P(\text{토픽 } B \mid \text{문서 } 2) = \frac{2}{3}$

P(단어 w | 토픽 t) 계산

문서 1 내 단어	
건강	A
수영	A
산책	A

문서 2 내 단어	
햄버거	???
치킨	A
햄버거	B
피자	B

문서 3 내 단어	
햄버거	B
산책	B
수영	B

- $P(\text{단어 “햄버거” } \mid \text{토픽 } A) = \frac{0}{4}$
- $P(\text{단어 “햄버거” } \mid \text{토픽 } B) = \frac{2}{5}$

⇒ B

- $P(\text{토픽 } A \mid \text{문서 } 2) \times P(\text{단어 “햄버거” } \mid \text{토픽 } A) = 0$
- $P(\text{토픽 } B \mid \text{문서 } 2) \times P(\text{단어 “햄버거” } \mid \text{토픽 } B) = \frac{4}{15}$
- 수치적으로 큰 B를 선택!
- 문서 2에 토픽 B도 적당히 있고, 토픽 B 안에 “햄버거”가 또 있어서 충분히 B로 할당해도 무방

E.O.D