10

비정형데이터분석

벡터공간모형을 이용한 텍스트 데이터 표현(1)

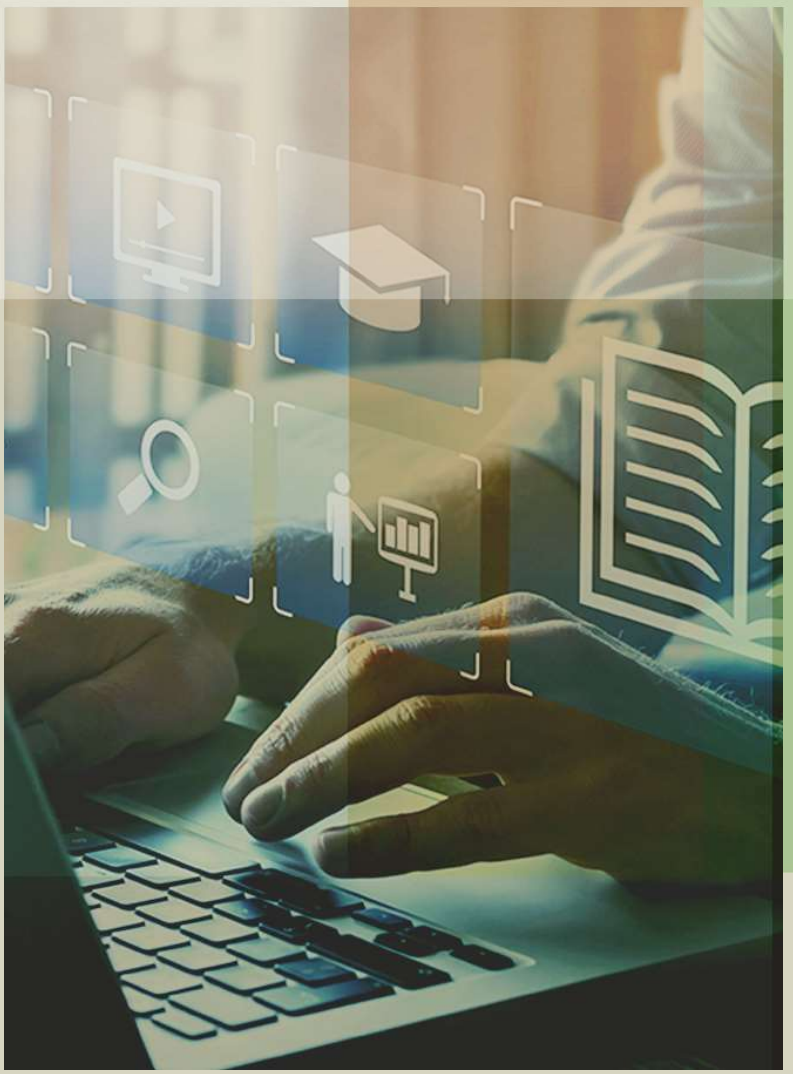
통계·데이터과학과 장영재 교수

학습목차

1. 벡터공간모형(Vector Space Model)
2. 문서-단어행렬(Document-Term Matrix, DTM)
3. 단어빈도-역문서빈도(Term Frequency -Inverse Document Frequency, TF-IDF)

01

벡터공간모형
(Vector Space Model)



1. 벡터공간모형(Vector Space Model)

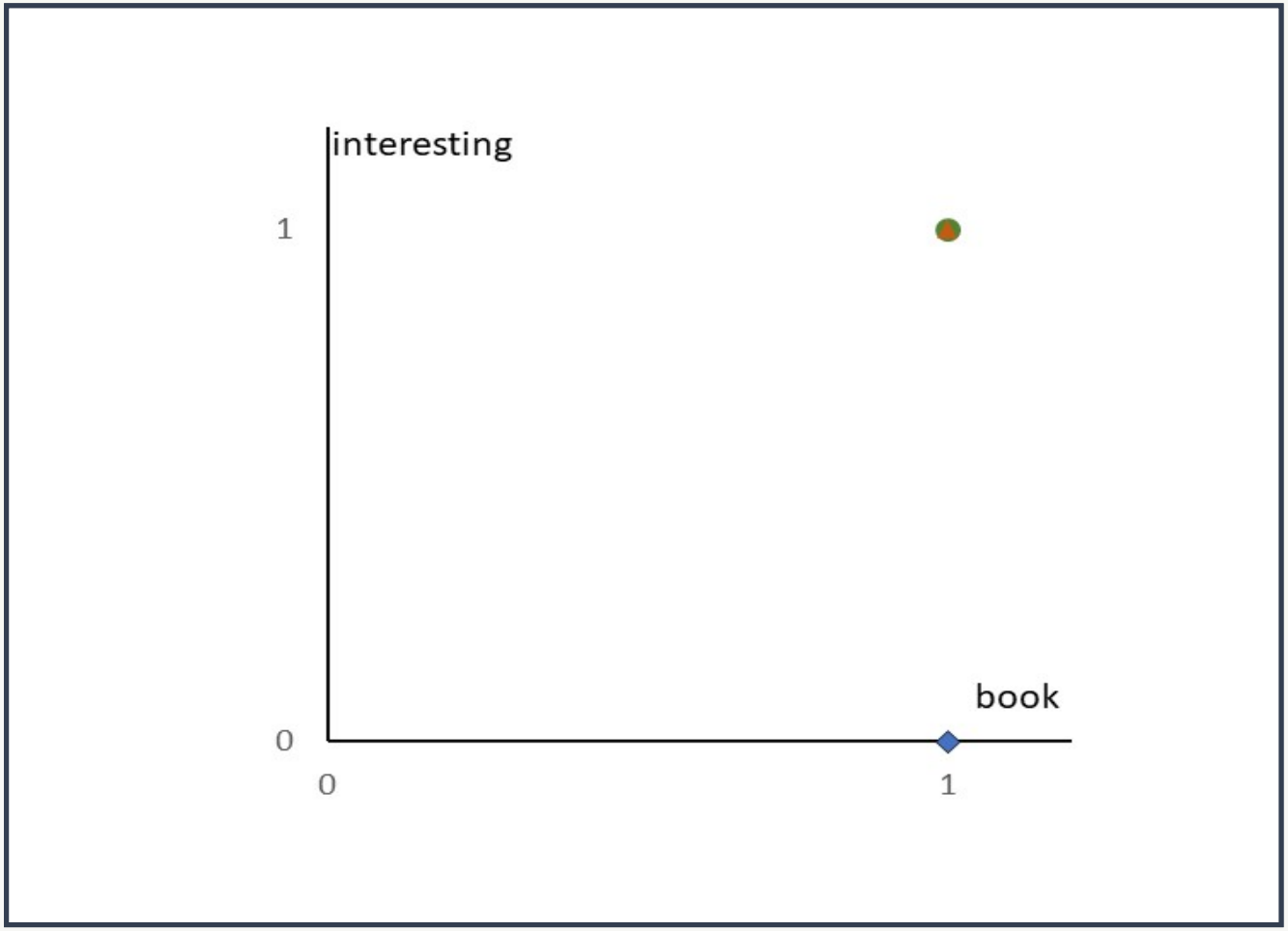
- 벡터공간모형이란 텍스트 데이터를 벡터공간의 한 점으로 표현하는 방식 (Kapetanios et al., 2013)
  - 단어주머니의 각 토큰*을 벡터공간의 각 축으로 하고 토큰들의 출현횟수를 각 축의 좌표 값으로 정의하는 방식
    - ▷ 편의상 토큰이라 표현하지만 엄밀하게는 타입을 의미함. 타입은 공통되는 토큰을 하나로 간주

      A : This is a book.

      B : This is an interesting book.

      C : This book is interesting.
  - 토큰화, 불용어 삭제 등의 전처리 과정을 거치면 세 단어주머니 {"book"}, {"interesting", "book"}, {"book", "interesting"}을 얻음
  - 단어주머니들은 한 축은 "book", 다른 한 축은 "interesting"인 2차원 벡터공간에 표현

1. 벡터공간모형(Vector Space Model)



02

문서-단어행렬(Document-Term Matrix, DTM)

2. 문서-단어행렬(Document-Term Matrix, DTM)

- 문서-단어행렬은 텍스트 데이터의 각 문서와 해당 문서에 등장한 각 단어, 즉 토큰의 출현빈도를 나타낸 행렬
  - 각 열은 각 문서의 토큰들에 대응되고 행과 열이 만나는 문서-단어행렬의 각 셀에는 문서별 토큰들의 출현횟수를 기록

  → "the best theater in New_York", "the best hotel in New_York", "the best gift for kids"의 단어주머니는

  {"good", "in", "New_York", "the", "theater"}, {"good", "in", "New_York", "the", "hotel"}, {"for", "good", "gift", "kid", "the"}

- 필요에 따라 각 문서를 열에, 각 단어를 행에 대응되도록 표현하는 단어-문서행렬(Term-Document Matrix, TDM)을 활용하기도 함
  - 벡터공간의 차원은 단어-문서행렬의 열의 수에 해당되고 벡터공간 내의 관측값들의 수는 단어-문서행렬의 행의 수에 해당

2. 문서-단어행렬(Document-Term Matrix, DTM)

문서-단어행렬

	for	gift	good	hotel	in	kid	New_York	the	theater
문서1	0	0	1	0	1	0	1	1	1
문서2	0	0	1	1	1	0	1	1	0
문서3	1	1	1	0	0	1	0	1	0

<표> 문서-단어행렬의 예

2. 문서-단어행렬(Document-Term Matrix, DTM)

- R 명령문을 통해 문서-단어행렬을 생성할 수 있음
  - 문장이 저장된 변수 x에 대한 전처리 작업을 수행하고 대소문자변환, 문장부호 삭제, 원형복원 과정을 거쳐 각 문장을 토큰별로 분해

→ 결과는 bows라는 이름을 가진 리스트로 저장됨

→ unlist( ) 함수로 리스트를 벡터로 변환하고 unique( ) 함수를 사용하여 중복되지 않는 단어들을 구한 후 sort( ) 함수로 정렬

→ lapply( ) 함수를 이용하여 세 문장의 문자열 벡터에 테이블을 생성하는 함수를 적용하고 생성된 테이블 행을 결합

2. 문서-단어행렬(Document-Term Matrix, DTM)

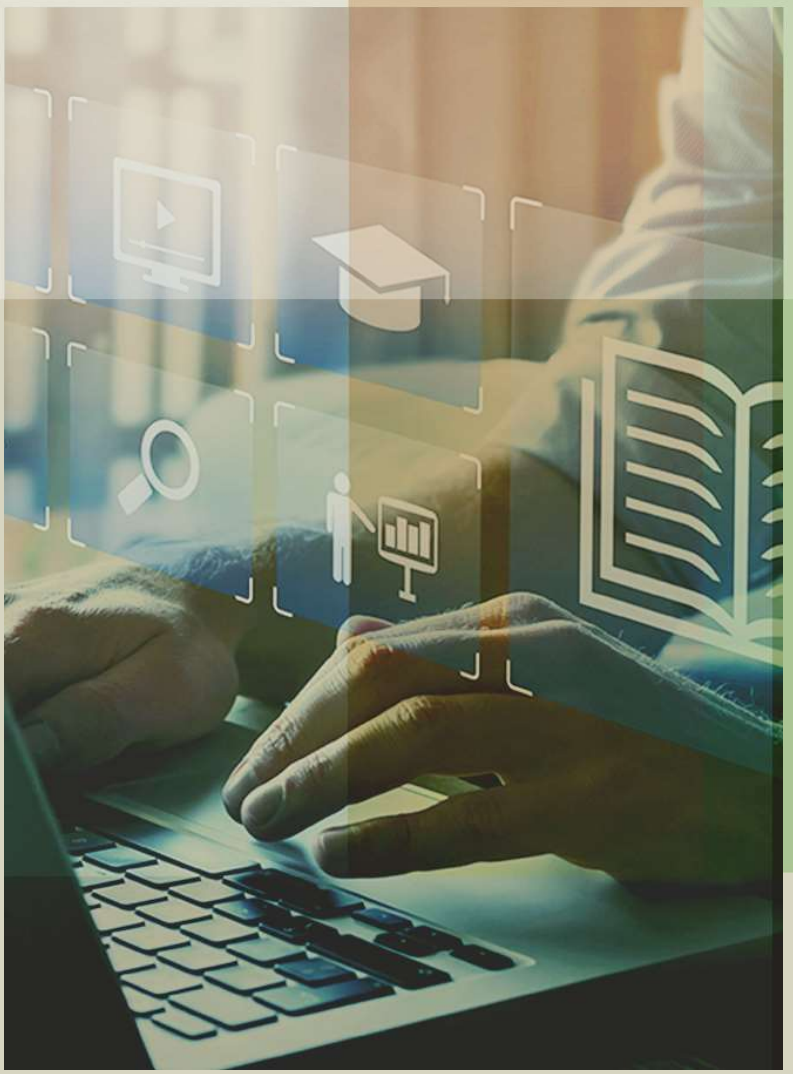
```
library(textstem)
x <- c("the best theater in New_York",
       "the best hotel in New_York", "the best gift for kids")
x <- tolower(x)
x <- gsub(x, pattern = "([^[:alnum:][:blank:]'-])", replacement = "")
x <- lemmatize_strings(x)
bows <- strsplit(x, " ")
lev <- sort(unique(unlist(bows)))
DTM <- lapply(bows, FUN = function(y, lev){table(factor(y, lev, ordered = T))}, lev = lev )
DTM <- matrix(unlist(DTM), nrow = length(DTM), byrow = TRUE)
colnames(DTM) <- lev
rownames(DTM) <- paste('doc', 1:dim(DTM)[1], sep="")
```

2. 문서-단어행렬(Document-Term Matrix, DTM)

- 문법적인 구조나 출현 순서 등에 대한 정보가 사라지고 단어들의 출현빈도에 대한 정보만 포함하고 있기 때문에 문맥에 숨겨져 있는 미묘한 의미까지는 표현하기 어려움
  - 문서의 주제 등 대략적인 의미를 파악하는 데에는 유용
    문서-단어행렬의 단어 빈도만으로도 주제의 유사성을 파악할 수 있음

03

단어빈도-역문서빈도 (Term Frequency-Inverse Document Frequency, TF-IDF)



3. 단어빈도-역문서빈도 (TF-IDF)

1 단어빈도(Term Frequency), 문서빈도(Document Frequency) 및 장서빈도(Collection Frequency)

- 문서에 포함된 단어들이 일반적으로 널리 사용되는 단어인지 특정 주제를 가진 문서에서만 사용되는 단어인지를 구분해내기 위한 지표가 필요
  - 단어빈도, 문서빈도, 장서빈도 등의 개념을 정의
    - → 단어빈도 $tf_{i,j}$ 는 특정문서 d_j 에서의 단어 w_i의 출현빈도
    - → 문서빈도 df_i 는 전체 문서들 중에서 단어 w_i를 포함한 문서의 수
    - → 장서빈도 cf_i 는 전체 문서에서의 단어 w_i의 출현빈도를 의미

3. 단어빈도-역문서빈도 (TF-IDF)

- 다음의 예에서 단어빈도, 문서빈도, 장서빈도를 산출

→ 문서2에서의 단어빈도는 $tf_{A,2} = 1,\ tf_{B,2} = 4$

→ 단어 A의 문서빈도 df_A 는 75, 단어 B의 문서빈도 df_B 는 20

→ 단어 A의 장서빈도 cf_A 와는 단어 B의 장서빈도 cf_B 는 모두 80

▷ 장서빈도는 단어빈도들의 합이고 항상 문서빈도보다 크거나 같음

	출현빈도								출현빈도 1이상인 문서 수
	문서1	문서2	문서3	문서4	…	문서99	문서100	합계	
단어A	1	1	1	2	…	0	1	80	75
단어B	3	4	0	0	…	3	0	80	20
…	…	…	…	…	…	…	…	…	…

<표> 단어-문서행렬의 예

3. 단어빈도-역문서빈도 (TF-IDF)

- 뉴욕타임즈(New York Times)의 기사들에서 "insurance"라는 단어와 "try"라는 단어의 장서빈도와 문서빈도 비교

→ 출현빈도는 각각 10,440, 10,422로 비슷하지만 문서빈도는 "try"가 "insurance"에 비해 두 배 이상

→ "insurance"의 평균 출현빈도는 문서당 2.612번(=10440/3997)으로 "try"의 문서당 1.190번(=10422/8760)에 비해 더 높음

p	장서빈도	문서빈도
insurance	10440	3997
try	10422	8760

<표> 뉴욕타임즈 기사에서의 단어 출현빈도
(Manning and Schütze, 1999)

3. 단어빈도-역문서빈도 (TF-IDF)

1 단어빈도-역문서빈도 (TF-IDF, Term Frequency-Inverse Document Frequency)

- 단어빈도만을 이용하여 문서-단어행렬을 만드는 것이 바람직하지 않을 수도 있음
  - "try"와 같이 일반적인 의미로 자주 사용되는 단어보다 "insurance"처럼 자주 사용되지는 않지만 특정 주제에 밀접하게 연관되어 있는 단어들이 데이터의 의미를 파악하는 데에 더 큰 도움

3. 단어빈도-역문서빈도 (TF-IDF)

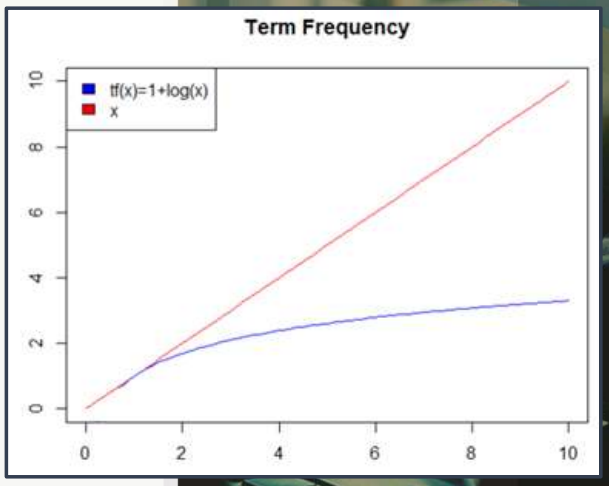
- 자주 사용되는 단어들의 가중치는 낮추고 특별한 주제를 가진 문서에만 주로 사용되는 단어들의 가중치는 높이는 방식을 고려

  - 단어빈도(TF)에 문서빈도(DF)의 역수에 해당하는 가중치의 곱

    : 단어빈도-역문서빈도(TF-IDF) 방식 활용 : $TF\text{-}IDF = TF \times IDF$

  - 단어빈도 TF는 문서 내에서의 상대출현빈도 $\left(rtf_{i,j} = \frac{tf_{i,j}}{\sum_k tf_{k,j}}\right)$ 나 로그함수를 적용한 형태 $\log(1 + tf_{i,j})$ 혹은 $1 + \log(tf_{i,j})$ (단, $tf_{i,j} = 0$ 경우에는 0으로 정의)도 사용할 수 있음

    → 출현빈도에 로그함수를 적용하면 그림에서와 같이 출현빈도가 큰 단어의 중요성이 상대적으로 축소되어 반영



<그림> 로그-단어 빈도

3. 단어빈도-역문서빈도 (TF-IDF)

- 자주 사용되는 단어들의 가중치는 낮추고 특별한 주제를 가진 문서에만 주로 사용되는 단어들의 가중치는 높이는 방식을 고려

  - 역문서빈도 IDF는 문서빈도 df, 즉 특정단어 w_i 가 출현한 문서의 수를 역수로 나타낸 것인데 일반적으로 이 역수에 로그함수를 취한 형태

    $log\frac{N}{df_i}$ 를 많이 사용

    → 단어 w_i 가 모든 문서에서 등장하였다면 $df_i = N$ 이 되어

    $$log\frac{N}{df_i} = log1 = 0$$

    ▷ 세 문장 " the best theater in New_York " , " the best hotel in New_York " , " the best gift for kids " 예제를 적용

3. 단어빈도-역문서빈도 (TF-IDF)

- 앞서 정의한 바대로 TF-IDF값을 사용하여 변환을 실시
  - 한 문서에만 나타난 단어는 1.099로 두 문서에서 등장한 단어의 0.405에 비해 큰 값을 나타냄
    - → 관사 "the", "a" 등과 같이 모든 문서에서 널리 사용되는 단어들은 불용어 삭제 과정을 거치지 않더라도 TF-IDF 방식으로 가중치를 부여하면 값이 0이 되어 불용어 삭제를 거친 것과 같은 결과

3. 단어빈도-역문서빈도 (TF-IDF)

	for	gift	good	hotel	in	kid	New_York	the	theater
문서1	0	0	1	0	1	0	1	1	1
문서2	0	0	1	1	1	0	1	1	0
문서3	1	1	1	0	0	1	0	1	0

$$TF\text{-}IDF = (1 + \log(tf_{i,j})) \times log\frac{N}{df_i}$$

	good	for	gift	hotel	in	kid	New_York	the	theater
문서1	0	0	0	0	0.405	0	0.405	0	1.099
문서2	0	0	0	1.099	0.405	0	0.405	0	0
문서3	0	1.099	1.099	0	0	1.099	0	0	0

3. 단어빈도-역문서빈도 (TF-IDF)

3 R을 이용한 TF-IDF 행렬 작성

- TF행렬과 IDF행렬을 작성하여 두 행렬의 원소들을 각각 곱하는 방식을 R로 구현하되 음의 무한대는 -Inf로 표현되므로 이 값들을 0으로 변환할 필요
  - "try"와 같이 일반적인 의미로 자주 사용되는 단어보다 "insurance"처럼 자주 사용되지는 않지만 특정 주제에 밀접하게 연관되어 있는 단어들이 데이터의 의미를 파악하는 데에 더 큰 도움

3. 단어빈도-역문서빈도 (TF-IDF)

```
TF <- 1+log(DTM)
TF[TF==-Inf] <- 0
DTM[DTM>0] <- 1
DF <- colSums(DTM)
IDF <- log(dim(DTM)[1]/DF)
TFIDF <- t(t(TF)*IDF)
```

또는

```
IDFmat <- matrix(IDF, nrow = dim(TF)[1], ncol = dim(TF)[2], byrow = T)
TFIDF <- TF*IDFmat
```

실습하기

다음시간안내

11

벡터공간모형을 이용한 텍스트 데이터 표현(2)