11

비정형데이터분석

# 벡터공간모형을 이용한 텍스트 데이터 표현(2)

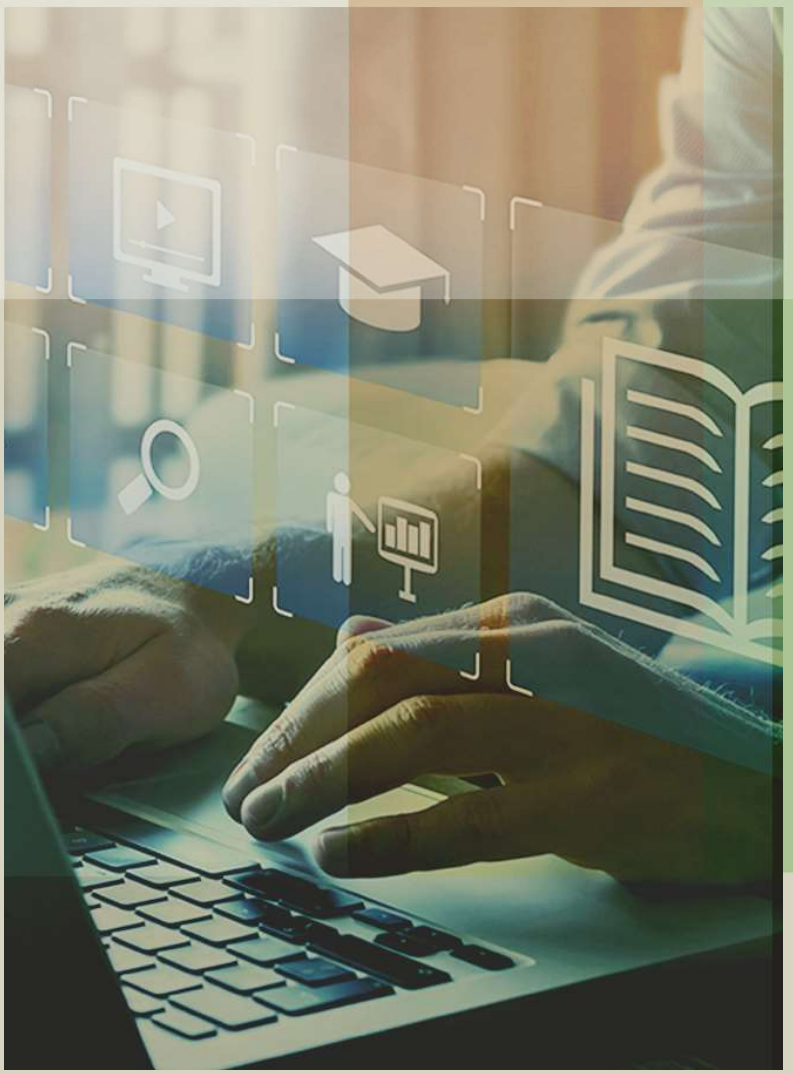
통계·데이터과학과 장영재 교수

# 학습목차

1. 문서-단어행렬의 특징
2. 단어 의미의 유사성
3. 워드 임베딩(Word Embedding)

01

# 문서-단어행렬의 특징



# 1. 문서-단어행렬의 특징

## 1 고차원 데이터

- 문서들을 모아 놓은 전체 텍스트 데이터에는 대체로 매우 많은 단어들이 포함되어 있음
  - 일반적인 영어 사전에는 수만 개에서 수십만 개, 국립국어원에서 작성한 우리말샘 사전에는 백만 개 이상의 단어가 등록

→ 사전에는 등록되어 있지 않지만 텍스트 데이터에 포함되어 있는 전문용어와 고유명사 등과 같은 단어들까지 고려하면 더 많은 단어들이 존재

# 1. 문서-단어행렬의 특징

- 텍스트 데이터를 문서-단어행렬로 표현하면 일반적으로 단어의 수, 즉 행렬에서 열의 수가 매우 많은 고차원(high-dimensional) 데이터 (독립변수의 수가 매우 많음)
  - 고차원 데이터를 분석하는 경우 과적합(overfitting), 변수들 사이의 다중공선성(multicollinearity) 문제 등에 유의할 필요

# 1. 문서-단어행렬의 특징

## 2 희소성(sparseness)

- 문서-단어행렬은 토큰의 종류가 매우 많은 고차원행렬이지만, 특정 문서 내에 포함된 토큰의 종류는 그렇게 많지 않은 것이 일반적
  - 한 문서에 사용할 수 있는 단어의 종류에는 한계가 있기 때문에 문서-단어행렬에서는 많은 항목들이 0의 값으로 채워짐

# 1. 문서-단어행렬의 특징

## 3 동의어와 동음이의어 등의 구분 곤란

- 텍스트 데이터를 문서-단어행렬로 표현할 때에는 동의어와 동음이의어에 대한 정보를 충분히 반영해주기 어려운 한계점이 있음
  - 집을 뜻하는 영어 단어들 "home", "house", "dwelling", "residence" 등은 서로 비슷한 의미를 가지고 있으나 별개의 단어
  - "object" : "사물", "물건", "목적", "반대하다", "이의를 제기하다"

02

# 단어 의미의 유사성

# 2. 단어 의미의 유사성

## 1 단어별 상호연관성 정보(Pointwise Mutual Information, PMI)

- 단어들의 의미의 연관성을 확인하는 지표로 단어별 PMI를 사용
  - 비슷한 주제에 사용되는 단어들은 통계적으로 같은 문서에 등장할 가능성이 높다는 점에 착안

→ 비슷한 의미를 가진 단어와 구절이 모여 글의 주제를 표현하는 공통성이 생기게 된다는 점에 착안하여 단어들의 의미의 연관성을 확인하는 지표가 PMI임

# 2. 단어 의미의 유사성

## 2 연관성 규칙(Association Rule)

- PMI는 데이터마이닝에서 활용되고 있는 연관성 규칙과 매우 유사
  - 동시에 구매되는 빈도가 높은 상품들을 파악해볼 수 있는 장바구니 분석 (market basket analysis)을 활용. 품목 A와 B에 대해,

→ 품목 A와 B에 대해, $A \rightarrow B$의 지지율, 신뢰도, 향상도를 다음과 같이 정의

지지율: $supp(A \cup B) = \frac{A와\ B가\ 동시에\ 포함된\ 거래수}{전체\ 거래수}$

* 품목 A를 구하는 사건을 A', 품목 B를 구하는 사건을 B'라고 정의하면 지지율을 확률로 표현하여 P(A' ∩ B')와 같이 나타내기도 함

신뢰도: $conf(A \Rightarrow B) = (supp(A \cup B))/(supp(A))$

향상도: $lift(A \Rightarrow B) = (supp(A \cup B))/(supp(A)supp(B)) = conf(A \Rightarrow B)/(supp(B))$

# 2. 단어 의미의 유사성

## 3 단어별 상호연관성 정보(Pointwise Mutual Information, PMI)의 확률표현

- PMI는 위에서 살펴본 연관성 규칙과 매우 유사하며 판매물품이 단어로, 장바구니가 텍스트 데이터의 문서로 바뀐 것으로 이해할 수 있음
  - 한 문서에 단어 $w_i$ 가 등장할 확률을 $P(w_i)$라 할 때,

$$PMI(w_i, w_j) = log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

$$PMI(w_i, w_j) = log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} = log \frac{P(w_i|w_j)}{P(w_i)} = log \frac{P(w_j|w_i)}{P(w_j)}$$

# 2. 단어 의미의 유사성

## 4 연속형 단어주머니(Continuous Bag-of-Words)와 스킵-그램(Skip-gram)

- 같은 문장이나 문서에 나타나지 않은 단어들 중에서도 연관성을 가지는 단어들이 있을 수 있음
  - 앞서 살펴본 PMI에서는 단어들이 같은 문장, 혹은 같은 문서에 나오는지가 단어들의 연관성을 측정하는 중요한 요소였음

# 2. 단어 의미의 유사성

- 아래 문장에 주어에 해당하는 "People" 대신 "We", "You", "They" 등의 복수형 대명사를 사용하여도 의미에 다소 차이는 있지만 문맥상으로는 큰 문제가 없음

  People think great science is done by luck. (Hamming(1986))

  → 동사 "think"를 "thinks"로 바꾸면 단수형 대명사나 고유명사도 사용할 수 있지만 "Lion", "Strawberry" 등의 사람이 아닌 생물이나 "Desk", "Water" 등의 무생물은 의미상 주어로 부적합

- 단어의 의미는 함께 사용되는 단어들을 통해 알 수 있음(Firth, 1957)

# 2. 단어 의미의 유사성

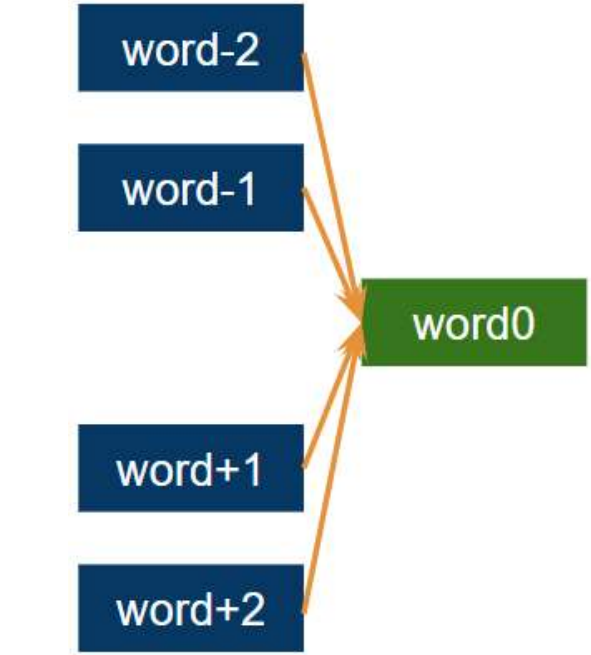
- 사람들이 사용하는 문장을 많이 모아보면 문맥상으로 빈 칸에 들어갈 수 있는 단어와 그렇지 않은 단어를 파악할 수 있음
  - " People"이 포함된 문장은 수없이 많으며 이들 문장을 모아보면 "People"이라는 단어의 의미와 "People"과 비슷한 의미를 가지는 단어들을 자연스럽게 찾아낼 수 있음

  → 주변 단어로부터 문맥상 유사한 의미를 가지는 단어를 유추해내는 방식 : 연속형 단어주머니(Continuous Bag-of-Words, CBoW 방식)

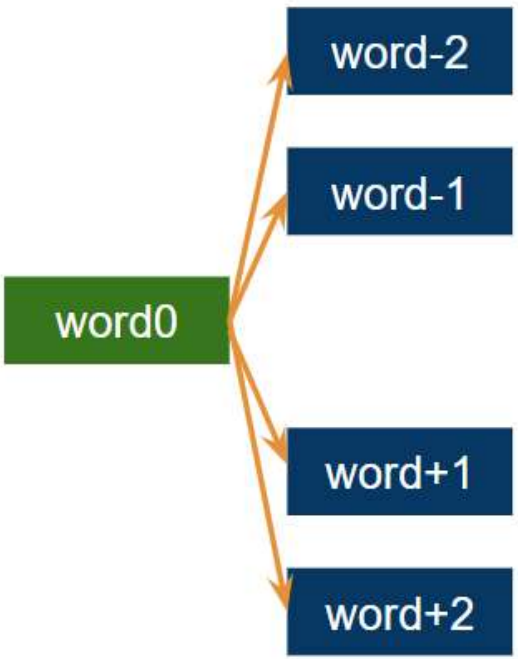
- 반대로 중심단어를 알고 있을 때 이 중심단어를 통해 주변에 배치될 수 있는 단어들을 추측해내는 방식으로 단어의 유사성을 파악해 낼 수도 있음

  → 중심 단어를 통해 주변에 배치될 수 있는 단어를 추측해 내는 방식으로 단어의 유사성을 판단 : 스킵-그램(Skip-gram) 방식

# 2. 단어 의미의 유사성




<continuous bag-of-words>




<skip gram>

# 2. 단어 의미의 유사성

- 비슷한 문맥에서 등장하는 단어들은 비슷한 의미를 가지는 경향이 있으며 다음 문장과 같이 분포가설(Distributional Hypothesis)로 요약

"Words that occur in similar contexts tend to have similar meanings." (Turney and Pantel, 2010)

  - 다만, 유사성이 높은 것으로 제시된 단어들 중에서도 의미상으로는 서로 반대되는 단어들도 있을 수 있음에 유의

→ ex) "이번 달은 가격이 x.x% 상승하였다." 문장과

"이번 달은 가격이 x.x% 하락하였다." 문장

03

# 워드 임베딩(Word Embedding)

# 3. 워드 임베딩(Word Embedding)

- 문서-단어행렬과 이를 변형한 TF-IDF 방식의 큰 특징 가운데 하나는 단어 하나가 행렬 내의 하나의 열을 차지한다는 점
  - 텍스트 데이터의 크기가 커질수록 행렬의 차원이 확대(고차원성)되고 행렬 내에서 0으로 채워지는 칸이 많아짐(희소성)

→ 워드 임베딩을 통해 단어들을 보다 낮은 차원의 공간에 표현함으로써 텍스트 데이터 분석의 효율성을 높일 수 있고 단어들 사이의 유사성을 잘 표현할 수 있음

# 3. 워드 임베딩(Word Embedding)

## 1 원-핫 표현방식(One-hot Encoding)

- 하나의 문서가 문서-단어행렬의 $i$째 단어 $w_i$ 하나로만 이루어져 있다면 이 단어 $w_i$에 해당되는 $i$번째 열만 1로 기록되고 이외의 열들은 모두 0으로 채워짐

  - 1을 "참(TRUE)" 또는 "뜨거운(hot)" 상태, 0을 "거짓(FALSE)" 또는 "차가운(cold)" 상태로 표현 : 원-핫 표현방식(Harris & Harris, 2015)

  → 단어1을 벡터로 표현하면 첫 번째 칸만 1이고 나머지 칸들은 모두 0인 벡터 $(1, 0, 0, \cdots, 0)$, 단어2는 벡터 $(0, 1, 0, \cdots, 0)$ 등

# 3. 워드 임베딩(Word Embedding)

- 단어1의 수를 $cnt_1$, 단어2의 수를 $cnt_2$, …, 단어 $i$의 수를 $cnt_i$ 라고 할 때 문서-단어행렬은 이들 단어 수를 가중치로 각 단어벡터들을 더해서 표현할 수 있음

$$Doc_k = \sum_i cnt_i \times w_i$$

# 3. 워드 임베딩(Word Embedding)

## 2 코사인 유사도(Cosine Similarity)

- 텍스트 데이터의 유사성을 측정하기 위해 흔히 사용되고 있는 코사인 유사도는 상관계수와 비슷한 개념
  - $x_i$는 문서 $x$에서의 단어 $w_i$의 출현빈도, $y_i$는 문서 $y$에서의 단어 $w_i$의 출현빈도라고 할 때,

$$\cos(x, y) = \frac{x_1 y_1 + x_2 y_2 + \cdots + x_p y_p}{\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}\sqrt{y_1^2 + y_2^2 + \cdots + y_p^2}}$$

→ $w_i$와 $w_j$가 서로 같은 단어가 아닌 경우, 값이 모두 0이 되므로 의미가 비슷한 단어라 하더라도 항상 0의 값을 가지게 되어 원-핫 표현방식에서는 단어 유사성을 표현하기에 부적합

# 3. 워드 임베딩(Word Embedding)

## 3 워드 임베딩(Word Embedding)

- 워드 임베딩이란 고차원 이산형 벡터공간에서 표현되었던 단어들을 저차원의 연속형 벡터공간에 집어넣는다(embed)는 뜻 (Eisenstein, 2019)

  - 워드 임베딩의 대표적인 기법 중 하나로 word2vec (Mikolov et al., 2013)

    → 방대한 텍스트 데이터를 기반으로 단어들 사이의 관계를 학습하여 단어들 사이의 관계를 표현할 적절한 벡터공간을 생성(50~300차원의 실수 공간)

    → 인공신경망(neural network) 모형을 사용하여 비슷한 문맥에서 자주 등장하는 단어들, 즉 비슷한 의미를 가진 단어들의 내적을 가급적 크게 하고 같은 문맥에서 출현하지 않는 단어들의 내적은 가급적 0에 가깝도록 벡터공간 상 배치

# 3. 워드 임베딩(Word Embedding)

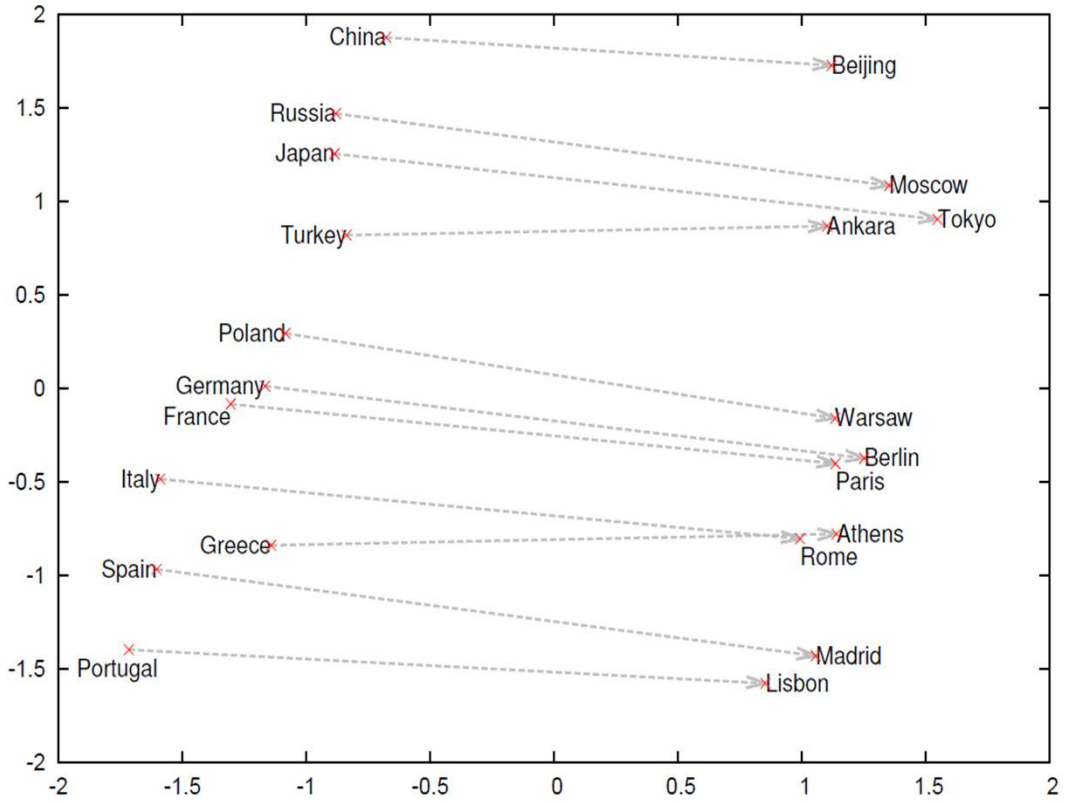
- word2vec 방식으로 학습된 단어 벡터들은 덧셈, 뺄셈 등의 연산을 통해 단어 사이의 관계를 표현할 수 있어 텍스트 데이터 분석에 있어서 유용성이 높음

→ 다양한 문맥을 통해 단어들의 의미를 학습해 나가므로 방대한 문서를 활용해야 좋은 학습 결과를 얻을 수 있음. 따라서 초기 학습과정에 많은 노력이 필요

## 사례 : 수도와 국가명을 2차원 공간의 점으로 표현 (Mikolov et al., 2013)

- 각 국가명과 해당 국가의 수도를 나타내는 점을 이은 선분들은 대체로 평행하고 길이도 비슷한 관계

→ 'B국가의 수도 = B국가명 - A국가명 + A국가의 수도' 식을 유추




**<그림> word2vec을 이용해 표현한 국가와 수도명의 관계**

다음시간안내

12

# 텍스트 데이터의 통계적 분석(1)