피쳐 엔지니어링

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

04

APPLICATION

ARTIFICIAL INTELLIGENCE

Notice

이 교육과정은 교육부 '성인학습자 역량 강화 교육콘텐츠 개발 ' 사업의 일환으로써 교육부로부터 예산을 지원 받아 고려사이버대학교가 개발하여 운영하고 있습니다.

THINKING

생각해보기

✓ 피쳐 엔지니어링이란 무엇일까요?

학습목표

GOALS

1. 확률 변수와 확률 분포에 대해 설명할 수 있다.
2. 기댓값에 대해 이해하고 설명할 수 있다.
3. 통계적 언어 모델(SLM)에 대해 설명할 수 있다.
4. N-gram 언어 모델에 대해 설명할 수 있다.
5. 문서의 유사도 개념에 대해 설명할 수 있다.
6. 유사도를 계산하는 다양한 방법들에 대해 설명할 수 있다.

1 기초 확률과 통계

2 언어 모델

3 유사도 계산하기

CONTENTS

학습내용

APPLICATION

01 기초 확률과 통계

확률 변수(random variable)

여러 값을 무작위하게 가지는 변수

일반적으로 확률 변수 자체는 보통 글꼴의 영문 소문자로,
확률 변수의 값은 이탤릭 영문 소문자로 표기함

확률 변수(random variable)

여러 값을 무작위하게 가지는 변수

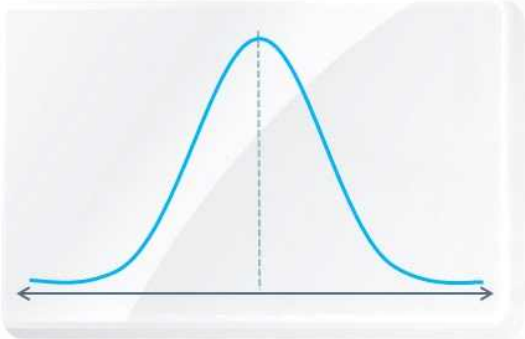
확률변수 x가 값 x 가 나올 확률값 p

$$P(x=x)=P(x)=p \; where \; 0 \leq p \leq 1$$

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

확률 분포(probability distribution)

하나의 확률 변수 또는 확률 변수들의 집합이 각자의 상태를 가질 가능도를 정의



확률 분포(probability distribution)

확률변수 X를 입력으로 받고 X의 각 값에 해당될 때의 확률을 출력하는 일종의 함수

확률변수가 이산이냐 연속이냐에 따라 서술하는 방식이 나뉨

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

이산 확률 분포(discrete probability distribution)

확률 변수가 이산적인(상태들의 개수가 유한 하여 그 개수를 셀 수 있는) 경우의 확률 분포

- 확률값의 총 합은 1이다.

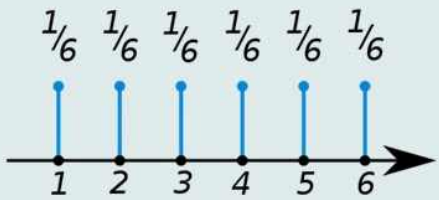
$$\Sigma P(x=x) = 1, \; where \; 0 \leq P(x=x) \leq 1$$

확률 질량 함수(probability mass function, PMF)

이산 확률 변수에서 특정 값에 대한 확률을 나타내는 함수

예시

주사위의 확률 질량 함수



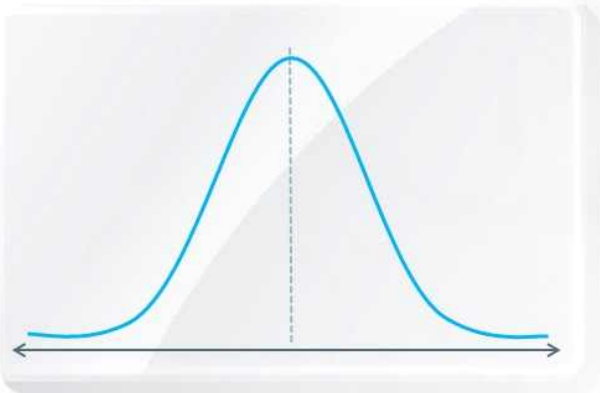
연속 확률 분포(continuous probability distribution)

확률 변수가 연속적인 경우의 확률 분포

- 셀 수 없기 때문에 구간을 정해 놓고 해당 구간에 속한 표본의 수로 나타냄
- 연속적인 확률 분포에서 확률값은 보통 어떤 구간의 넓이를 의미함

확률 밀도 함수(probability density function, PDF)

확률 변수가 연속적인 경우의 확률 분포



06 확률 밀도 함수

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

확률 밀도 함수(probability density function, PDF)

확률 변수가 연속적인 경우의 확률 분포

연속적이므로 특정 점보다 영역에 대한 확률값을 구함

> **어떤 값에 대한 확률 밀도 함수의 면적의 합은 항상 1임**

결합 확률

두 개 이상의 사건이 동시에 일어날 확률

> 두 개 이상의 확률 변수를 가짐

예시

주사위 2개(A, B)를 던질 때의 확률

$$P(A,B)$$

07 결합 확률

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

예시

주사위 2개(A, B)를 던졌을 때 A는 5, B는 3가 나올 확률

$$P(A=5,B=3)$$

- 위의 경우처럼 각각의 사건이 서로에게 영향을 끼치지 않는 경우 확률에서는 이를 "독립"이라고 부르고 아래의 조건을 만족한다.

$$P(A,B)=P(A)P(B)$$

> **어떤 확률이 발생했을 때,**
> **다른 어떤 한 사건이 발생할 확률**

$$P(A|B) = \frac{P(A,B)}{P(B)}, P(A,B) = P(A|B)P(B)$$

예시

주사위 B가 3이 나온 상황에서 주사위 A에서 얻을 수 있는 값의 확률 분포

$$P(A|B=3)$$

08 조건부 확률

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

조건부 확률의 연쇄법칙

다수의 확률변수에 관한 임의의 결합 확률 분포를 조건부 확률의 정의를 사용해 아래와 같이 조건부 분포들로 분해 할 수 있음

$$P\left(x^{(1)}, \ldots, x^{(n)}\right) = P(x^{(1)}) \prod_{i=2}^{n} P(x^{(i)} | x^{(1)}, \ldots, x^{(i-1)})$$

> **위와 같은 관계를 확률의 연쇄 법칙,**
> **또는 곱의 법칙이라고 부름**

$$P(a,b,c) = P(a|b,c)P(b,c)$$

$$P(b,c) = P(b|c)P(c)$$

$$P(a,b,c) = P(a|b,c)P(b,c) = P(a|b,c)P(b|c)P(c)$$

베이즈 정리(Bayes theorem)

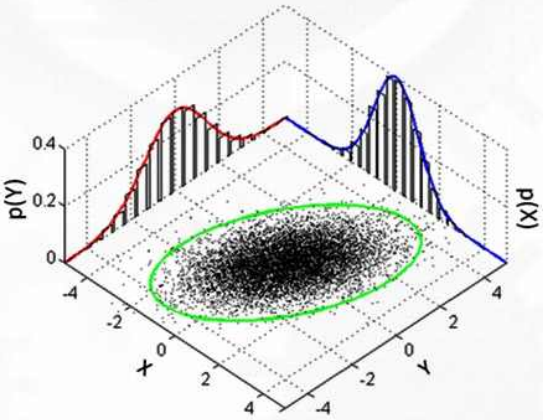
- $P(y \mid x)$를 아는 상태에서 $P(x \mid y)$를 구해야 하는 상태에서 쓰임

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)}$$

- 조건부확률의 정의를 통해 유도가 가능함

주변확률분포(Marginal probability distribution)

- 두 개 이상의 확률 변수의 결합 확률 분포가 있을 때, 하나의 확률 변수에 대해서 적분을 수행한 결과
- 즉, 부분 집합에 대한 확률분포를 의미



출처: https://en.wikipedia.org/wiki/Marginal_distribution

기댓값(Expectaion)

- 확률 변수의 기댓값은 각 사건이 벌어졌을 때의 이득과 그 사건이 벌어질 확률을 곱한 것을 전체 사건에 대해 합한 값
- 이산 변수의 경우 합산으로, 연속 변수의 경우에는 적분으로 계산

$$\mathbb{E}_{x\sim P}[f(x)] = \sum_x P(x)f(x) \ , \int p(x)f(x)dx$$

예시

실제 주사위 결과의 기댓값

$$\mathbb{E}_{x\sim P}[result(x)\,] = \sum_{x} P(x)f(x)$$

$$= \sum_{x=1}^{6} P(x = x) \times result(x)$$

$$= \frac{1}{6} \times (1 + 2 + 3 + 4 + 5 + 6) = 3.5$$

APPLICATION

02 언어 모델

언어 모델(language model)

단어 시퀀스에 확률을 할당하는 일을 하는 모델

- 언어 모델을 통해 문장 자체의 출현 확률을 예측하거나,
  이전 단어들이 주어졌을 때 다음 단어를 예측하는 등의 작업이 가능함
- 통계를 이용하거나 인공신경망을 이용해 언어 모델을 만들 수 있음

오늘 저녁은 치킨을 먹기로 _____.	
1	결정했다
2	엘레베이터
3	먹었다
4	딥러닝

	문 장
1	저는 커서 개발자가 되는게 꿈이에요
2	저는 2040년 3월에 개발자가 되는게 꿈이에요

01 언어 모델

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

단어 시퀀스에 확률을 할당하는 이유

1 **기계 번역**

- P(우리는 치킨을 먹는다) > P(우리는 치킨을 먹혔다)

오타 교정 2

- 그는 생선 가시를 → P(발랐다) > P(잘랐다)

주어진 이전 단어들로부터 다음 단어 예측하기

> **이전 단어들이 주어졌을 때,
> 다음 단어를 예측하는 방식을 단어 시퀀스에
> 확률을 할당하기 위해 가장 보편적으로 사용한다.**

01 언어 모델

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

주어진 이전 단어들로부터 다음 단어 예측하기

1. 하나의 단어를 w, 단어 시퀀스를 W 라고 할 때,
n 개의 단어가 등장하는 단어시퀀스 W의 확률

$$P(W) = P(w_1, w_2, \cdots, w_n)$$

주어진 이전 단어들로부터 다음 단어 예측하기

2 $n-1$개의 단어가 나열된 상태에서 n 번째 단어의 확률

$$P(w_n|w_1, w_2, \ldots, w_{n-1})$$

01 언어 모델

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

주어진 이전 단어들로부터 다음 단어 예측하기

3 전체 단어 시퀀스의 확률

$$P(W) = P(w_1, w_2, \cdots, w_n) = \prod_{i=1}^{n} P(w_n|w_1, w_2, \cdots, w_{n-1})$$

통계적 언어 모델(Statistical Language Model)

예시문장

"My dream is having cute baby"

- $P(W) = P(w_1, w_2, \cdots, w_n) = \prod_{i=1}^{n} P(w_n | w_1, w_2, \cdots, w_{n-1})$
- $P(\text{My dream is having cute baby}) = P(\text{My}) \times$
  $P(\text{dream}|\text{My}) \times P(\text{is}|\text{My dream}) \times$
  $P(\text{having}|\text{My dream is}) \times P(\text{cute}|\text{My dream is having}) \times$
  $P(\text{baby}|\text{My dream is having cute}) \times$

02 통계적 언어 모델

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

카운트 기반의 확률 계산

예시

My dream is 가 나왔을 때 having이 나올 확률

- $P(\text{having}|\text{My dream is}) = \frac{count(My\ dream\ is\ having)}{count(My\ dream\ is)}$

코퍼스 데이터에서 My dream is가 100번 나왔는데
그 중 뒤에 having이 나온 경우가 10번이라면 위의 결과는 10%임

카운트 기반 접근의 한계

$$P(\text{having}|\text{My dream is}) = \frac{count(My\ dream\ is\ having)}{count(My\ dream\ is)}$$

- 언어 모델이 학습한 코퍼스에 My dream is having이라는 시퀀스가 없으면 확률은 0이 되고, My dream is 라는 시퀀스가 없으면 확률 자체를 정의할 수 없음
- 그러나 실제 세계에서의 자연어의 경우 My dream is having이라는 시퀀스는 정답일 가능성이 높음
- 충분한 데이터를 관측하지 못하여 모델링에 문제가 생기는 경우, 이를 희소 문제라고 함

훈련 코퍼스에 확률을 계산하고 싶은 문장이나 단어가 없을 수 있다는 통계적 언어 모델의 단점을 해결하고자 함

$$P(\text{having}|\text{My dream is}) \approx P(\text{having}|\text{dream is})$$

단어의 확률을 구하고자 기준 단어의 앞 단어를 전부 포함해서 카운트 하는 것이 아니라, 앞 단어 중 임의의 개수의 단어만 참고하여 카운트함

N-gram

N개의 연속적인 단어의 나열을 의미함

갖고 있는 코퍼스에서 n개의 단어 뭉치로 끊어서 이를 하나의 토큰으로 간주함

- N = 1 → 유니그램
- N = 2 → 바이그램
- N = 3 → 트라이그램
- N = 4 이상
  → 앞에 그대로 숫자를 붙여서 부름

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

	My dream is having cute baby
unigram	My, dream, is, having, cute, baby
bigram	My dream, dream is, is having, having cute, cute baby
trigram	My dream is, dream is having, is having cute, having cute baby
4-gram	My dream is having, dream is having cute, is having cute baby

~~My dream~~ is having cute ???

$$P(???|\text{is having cute}) = \frac{count(is\ having\ cute\ ???)}{count(is\ having\ cute)}$$

ⓥ 현재 코퍼스에서 is having cute가 10번 등장 한 경우,
그 뒤에 baby가 나온 경우는 5번, bag이 나온 경우는 2번 이라 했을 때,
확률적으로 bag보다는 baby가 맞는 문장이라고 판단함

여전히 카운트 기반의 SLM모델이다.

→ 여전히 **희소문제가 존재**함

Q N을 선택하는 데에 대한 문제

A N을 크게 선택하면 희소문제가 심각해지고, N을 작게 선택하면 현실 세계의 자연어 처리에서 적용이 힘들 수 있음

APPLICATION

03 유사도 계산하기

문서의 유사도(Document Similarity)

문서의 유사도를 계산하는 일은 자연어 처리의 주요 주제 중 하나

- 사람의 경우 주로 문서들간에 동일한 단어 또는 비슷한 단어가 얼마나 공통적으로 많이 사용 되었는지에 의존함
- 기계도 이와 유사하게 문서 내 단어들을 어떻게 수치화했는지, 단어간의 차이를 다양한 방법으로 계싼하여 유사도를 구함

코사인 유사도(Cosine Similarity)란?

> 두 벡터 간의 코사인 각도를 이용하여
> 구할 수 있는 두 벡터의 유사도

02 코사인 유사도

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

코사인 유사도 (Cosine Similarity)

- 두 벡터의 방향이 완전히 동일한 경우에는 1의 값을, 수직을 이루면 0, 180도로 반대의 방향을 가지면 -1의 값을 갖게 됨
- -1부터 1 사이의 값을 가지며 값이 1에 가까울 수록 유사도가 높다고 판단함

문서간 코사인 유사도 계산해보기

예시

문장과 TF-IDF

문서 1 : 저는 사과가 좋아요

문서 2 : 저는 바나나가 좋아요

문서 3 : 저는 바나나가 좋아요

저는 바나나가 좋아요

	바나나	사과	저는	좋아요
문서1	0	1	1	1
문서2	1	0	1	1
문서3	2	0	2	2

02 코사인 유사도

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

코사인 유사도(Cosine Similarity)

코사인 유사도
: -1

코사인 유사도
: 0

코사인 유사도
: 1

코사인 유사도 수식

$$cosine\ similarity = (A \cdot B)/(\| A \| \| B \|)$$

코사인 유사도를 계산하는 함수 만들기

In [1]:
```
import numpy as np
from numpy import dot
from numpy.linalg import norm

def cos_sim(A, B):
    return dot(A, B)/(norm(A)*norm(B))
```

02 코사인 유사도

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

코사인 유사도 계산하기

In [2]:
```
doc1 = np.array([0,1,1,1])
doc2 = np.array([1,0,1,1])
doc3 = np.array([2,0,2,2])
```

In [3]:
```
print("문서1과 문서2간의 코사인 유사도 : ", cos_sim(doc1, doc2))
print("문서2과 문서3간의 코사인 유사도 : ", cos_sim(doc2, doc3))
print("문서3과 문서1간의 코사인 유사도 : ", cos_sim(doc3, doc1))
```

```
문서1과 문서2간의 코사인 유사도 :  0.6666666666666667
문서2과 문서3간의 코사인 유사도 :  1.0000000000000002
문서3과 문서1간의 코사인 유사도 :  0.6666666666666667
```

- 문서 2와 문서 3은 빈도수가 다르지만 방향은 같은 벡터이므로 문서2와 문서3 간의 코사인 유사도 값은 1을 가짐
- 코사인 유사도는 크기가 아닌 각도를 기준으로 계산되므로문서 1과 문서2, 문서 1과 문서 3간의 유사도는 동일함

유클리드 거리(Cosine Similarity)

- 다차원 공간에서 두개의 점 p와 q가 각각 $p=(p_1, p_2, \cdots, p_n)$과 $q=(q\_1, q\_2, \cdots, q\_n)$ 의 좌표를 가질 때, 두 점 사이의 거리를 계산하는 유클리드 거리공식은 아래와 같음

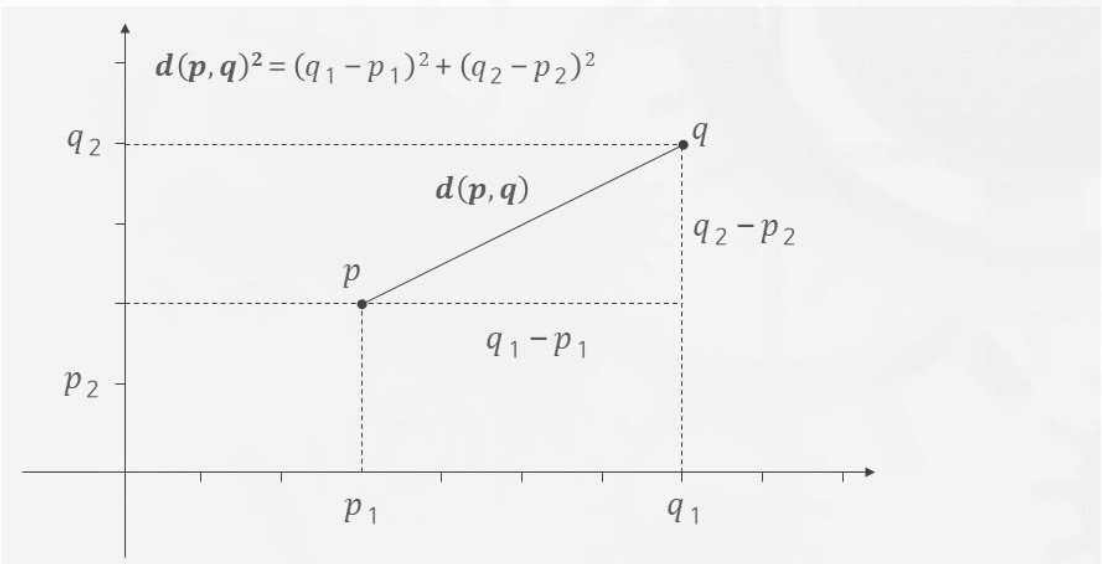
$$\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

03 유클리드 거리

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

유클리드 거리(Cosine Similarity)

- 2차원 공간의 경우 우리에게 익숙한 그림



자카드 유사도(Jaccard Similarity)

- A, B 두개의 집합이 있다고 할 때, 합집합에서 교집합의 비율을 구함으로써 유사도를 계산할 수 있음

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- 자카드 유사도는 0과 1 사이의 값을 가지며 두 집합이 동일하면 1의 값을, 두 집합의 공통 원소가 없다면 0의 값을 가짐

04 유사도 계산

DEEP LEARNING AND NATURAL LANGUAGE PROCESSING

문서에서의 자카드 유사도(Jaccard Similarity)

$$J(doc1, doc2) = \frac{doc1 \cap doc2}{doc1 \cup doc2}$$

In [1]:
```
doc1 = "red blue we like watching rainbow love you"
doc2 = "red blue why he want that car"
```

In [2]:
```
token1 = doc1.split()
token2 = doc2.split()

print(token1)
print(token2)
```

```
['red', 'blue', 'we', 'like', 'watching', 'rainbow', 'love', 'you']
['red', 'blue', 'why', 'he', 'want', 'that', 'car']
```

코사인 유사도 계산하기

```
In [3]:
1 union = set(token1) | set(token2)
2 union
```

```
Out[3]: {'blue',
 'car',
 'he',
 'like',
 'love',
 'rainbow',
 'red',
 'that',
 'want',
 'watching',
 'we',
 'why',
 'you'}
```

```
In [4]:
1 intersection = set(token1) & set(token2)
2 intersection
```

```
Out[4]: {'blue', 'red'}
```

```
In [5]:
1 jaccard_similarity = len(intersection) / len(union)
2 jaccard_similarity
```

```
Out[5]: 0.15384615384615385
```

SUMMARY

학습정리

- 확률 변수와 확률 분포
- 어떤 확률이 발생했을 때,
  다른 어떤 한 사건이 발생할 확률, 조건부 확률
- 카운트 기반의 언어모델 SLM과 N-gram
- 문서간의 유사도를 계산하는 다양한 방법

EXPANSION

확장하기

1. 확률 분포는 무엇이고 그 종류에는 어떤 것들이 있을까요?
2. 베이즈 정리와 관련된 유명한 문제 중 하나인 몬티홀 문제에 대해 살펴보세요.
3. 언어 모델에서 확률을 사용하는 이유는 무엇일까요?
4. 통계적 언어모델의 한계는 무엇이고 해결하기 위한 모델에는 어떤 것이 있을까요?
5. 문서 간 유사도를 계산하는 방법에는 어떤 것들이 있을까요?

참고 문헌

REFERENCE

◆ 참고 사이트
- 용어들에 대한 정의 : https://ko.wikipedia.org/wiki.
- 점프 투 파이썬 : https://wikidocs.net/1669
- 딥러닝을 이용한 자연어 처리 입문 : https://wikidocs.net/book/2155
- 퍼블릭에이아이(www.public.co.kr)

◆ 참고 서적
- 김기현, 『김기현의 자연어 처리 딥러닝 캠프 파이토치 편』, 한빛미디어, 2019
- 이안 굿펠로, 요슈아 벤지오, 에런 쿠빌, 『심층 학습』, 제이펍

서체 출처 : 에스코어드림체-㈜에스코어, 나눔글꼴체-㈜네이버