자연어 처리 딥러닝 캠프

8장 텍스트 분류, 9장 언어 모델링

최우진

2020/01/30

목차

8.1 텍스트 분류

- 텍스트 분류: 텍스트, 문장 또는 문서를 입력으로 받아 사전에 정의된 클래스 중에 어디에 속하는지 분류하는 과정.

8.2.0 베이즈 정리

- 베이즈 정리

$$\underbrace{\mathbf{P}(\boldsymbol{c} \mid \boldsymbol{D})}_{\text{사후 확률}} = \frac{\overbrace{\boldsymbol{P}(\boldsymbol{D} \mid \boldsymbol{c})}^{\text{우도}}\ \overbrace{\boldsymbol{P}(\boldsymbol{c})}^{\text{사전 확률}}}{\underbrace{\boldsymbol{P}(\boldsymbol{D})}_{\text{증거}}} = \frac{\boldsymbol{P}(\boldsymbol{D} \mid \boldsymbol{c})\,\boldsymbol{P}(\boldsymbol{c})}{\sum_{i=1}^{|c|} \boldsymbol{P}(\boldsymbol{D} \mid \boldsymbol{c_i})\,\boldsymbol{P}(\boldsymbol{c_i})}$$

우도	어떤 모델에서 해당 관측값이 나올 확률
사전확률	관측자가 관측을 하기 전에 시스템 또는 모델에 대해 가지고 있는 선험적 확률
사후확률	사건이 발생한 후 그 사건이 특정 모델에서 발생했을 확률

- 대부분 $\boldsymbol{P}(\boldsymbol{D})$을 구하기 어려우므로, 아래 식으로 접근해도 된다.

$$\mathbf{P}(\boldsymbol{c} \mid \boldsymbol{D}) \propto \boldsymbol{P}(\boldsymbol{D} \mid \boldsymbol{c})\,\boldsymbol{P}(\boldsymbol{c})$$

8.2.1 MAP(사후 확률 최대화)와 MLE(최대가능도 추정)

- D(데이터)가 주어졌을 때 가능한 클래스의 집합 c 중에서 사후 확률을 최대로 하는 클래스 D를 선택

$$\hat{c}_{MAP} = arg\max_{c \in C} P(C = c \mid D)$$

- 데이터 D가 나타날 가능도를 최대로 하는 클래스 D를 선택

$$\hat{c}_{MLE} = arg\max_{c \in C} P(D \mid C = c)$$

8.2.2 나이브 베이즈

- 사후 확률 최대화(MAP)를 기반으로 동작
- 사후 확률, N개의 단어 $w_1, w_2, \ldots, w_n$가 주어졌을 때, 문장이 c 클래스에 속할 확률

$$\mathbf{P(y = c \mid x = w_1, w_2, \ldots, w_n)}$$

- x가 다양한 특징으로 이루어진 데이터라면 훈련 데이터에서 매우 희소
- 나이브 베이즈 가정: 각 특징이 독립적

$$\begin{aligned}\mathbf{P(y = c \mid x = w_1, w_2, \ldots, w_n)} &\propto \mathbf{P(x = w_1, w_2, \ldots, w_n \mid y = c)P(y = c)} \\ &\approx \boldsymbol{P(w_1 \mid c)P(w_2 \mid c) \cdots P(w_n \mid c)P(c)} \\ &= \boldsymbol{\prod_{i=1}^{n} P(w_i \mid c)P(c)}\end{aligned}$$

8.2.2 나이브 베이즈

- 나이브 베이즈의 가정에 따라 각 특징들의 확률의 곱에 사전 확률을 곱한 값을 최대화하는 클래스를 계산

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \mathbf{P}(\mathbf{y} = \mathbf{c} \mid \mathbf{x} = \mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_n})$$
$$\approx \underset{c \in C}{\operatorname{argmax}} \prod_{i=1}^{n} \boldsymbol{P}(\boldsymbol{w_i} \mid \boldsymbol{c}) \boldsymbol{P}(\boldsymbol{c})$$

- 사전 확률은 실제 데이터(코퍼스)에서 출현한 빈도를 통해 추정

$$\boldsymbol{P}(\boldsymbol{y} = \boldsymbol{c}) \approx \frac{\boldsymbol{Count}(\boldsymbol{c})}{\sum_{i=1}^{|C|} \boldsymbol{Count}(\boldsymbol{c_i})}$$

8.2.2 나이브 베이즈

- 특징 별 가능도 확률

$$P(w \mid c) \approx \frac{Count(w, c)}{\sum_{j=1}^{|V|} Count(w_j, c)}$$

8.2.3 감성 분석

- 감성 분석은 주로 텍스트 분류에서 활용
- 사용자의 댓글이나 리뷰 등을 긍정 또는 부정으로 분류

$$P(pos) \approx \frac{Count(pos)}{|D|}$$

- 예제

$$C = \{pos, neg\}$$
$$D = \{d_1, d_2, \cdots\}$$

$$P(happy \mid pos) \approx \frac{Count(happy, pos)}{\sum_{j=1}^{|V|} Count(w_j, pos)}$$

- "I am happy to see this movie!" 문장

$$P(pos|I, am, happy, to, see, this, movie, !)$$
$$= \frac{P(I, am, happy, to, see, this, movie, !|pos)P(pos)}{P(I, am, happy, to, see, this, movie, !)}$$
$$\approx \frac{P(I|pos)P(am|pos)P(happy|pos)\cdots P(!|pos)P(pos)}{P(I, am, happy, to, see, this, movie, !)}$$

8.2.4 add-one smoothing

- $Count(happy, pos)$가 0이었다면 $P(happy \mid pos) = 0$이 됨.
- 따라서 가능도를 구하는 식에 1을 더하여 문제를 해결.

$$\tilde{P}(w \mid c) = \frac{Count(w, c) + 1}{\left(\sum_{j=1}^{|V|} Count(w_j, c)\right) + |V|}$$

- 완벽한 해결법은 아니지만, 매우 간단하면서도 강력

8.2.5 장점과 한계

- "I am not happy to see this movie!" 문장은 기존의 문장과 정반대의 뜻을 가진다.

$$P(pos|I, am, not, happy, to, see, this, movie, !)$$
$$P(neg|I, am, not, happy, to, see, this, movie, !)$$

- 'not'은 'happy'를 수식하므로 두 단어를 독립으로 가정하는 것은 옳지 않다

$$P(not, happy) \neq P(not)P(happy)$$

8.2.6 흔한 오해 2

- 표제어 추출, 어간 추출을 수행하여 접사 등을 제거한 후, 텍스트를 분류해야 하는가?

단계	문장
원문	나는 학교에 가요.
전처리	나 는 학교 에 가 요 .
추출	나 학교 가 .

- 위와 같은 어간 추출 결과를 나오는 문장들

번호	문장
1	나만 학교에 가요.
2	나도 학교로 가요.
3	나는 학교를 가요.

- 따라서 희소성 문제를 줄일 수 있다.

9.1.1 언어 모델링

- 언어 모델(LM): 문장의 확률을 나타내는 모델
- 즉, 언어 모델을 통해 문장 자체의 출현 확률을 예측하거나, 이전 단어들이 주어졌을 때 다음 단어를 예측할 수 있으며, 결과적으로 주어진 문장이 얼마나 자연스럽고 유창한 표현인지 계산할 수 있다.
- 예제

번호	버스 정류장에서 방금 버스를 ___.
1	사랑해
2	고양이
3	놓쳤다
4	사고남

번호	문장
1	저는 어제 점심을 먹었습니다.
2	저는 2015년 3월 18일 점심을 먹었습니다.

9.1.2 한국어

- 한국어: 교착어, 영어: 고립어(+굴절어), 중국어: 고립어
- 교착어의 특징

  - 단어의 의미 또는 역할은 어순보다 접사 또는 조사에 의해 결정

  - 같은 의미의 단어라도 붙는 접사나 조사에 따라 단어의 형태나 단어가 변형

- 예제

번호	문장
1	나는 학교에 갑니다 버스를 타고 .
2	나는 버스를 타고 학교에 갑니다 .
3	버스를 타고 나는 학교에 갑니다 .
4	(나는) 버스를 타고 학교에 갑니다 .

9.1.3 문장의 확률 표현

- w_1, w_2라는 2개의 단어가 문장 안에서 순서대로 나온 경우

$$\mathrm{P}(\mathrm{w}_1, \mathrm{w}_2)$$

- 연쇄법칙을 통해 문장 전체를 표현

$$\begin{aligned}\mathrm{P}(\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_n) &= \mathrm{P}(\mathrm{w}_1)\mathrm{P}(\mathrm{w}_2|\mathrm{w}_1)\mathrm{P}(\mathrm{w}_3|\mathrm{w}_1, \mathrm{w}_2)\cdots P(w_n|w_1, w_2, \cdots, w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i \mid w_{<i})\end{aligned}$$

- 곱셈보다 덧셈이 계산 속도가 빠르므로

$$logP(\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_n) = \sum_{i=1}^{n} logP(w_i \mid w_{<i})$$

9.2.2 마르코프 가정

- 특정 시점의 상태 확률은 단지 그 직전 상태에만 의존한다는 가정
- 즉, 앞서 출현한 k개의 단어만 보고 다음 단어의 출현확률을 계산

$$P(x_i|x_1, x_2, \cdots, x_{i-1}) \approx P(x_i|x_{i-k}, \cdots, x_{i-1})$$

- 보통 k값은 0~3정도로 정함
- 단어를 예측할 때, 전체 단어를 조합하는 대신 바로 앞의 일부 조합만 출현 빈도로 계산하여 확률을 추정하는 방법 "n-gram" 이다.

k	n-gram	명칭
0	1-gram	uni-gram
1	2-gram	bi-gram
2	3-gram	tri-gram

9.2.3 일반화

- 스무딩

$$P(w_i \mid w_{<i}) \approx \frac{Count(w_{<i}, w_i) + 1}{Count(w_{<i}) + V}$$

- 스무딩 일반화

$$P(w_i \mid w_{<i}) \approx \frac{Count(w_{<i}, w_i) + k}{Count(w_{<i}) + kV}$$

9.2.3 일반화

- Kneser-Ney 디스카운팅

  - 단어 w가 다른 단어 v의 뒤에 출현할 때, 얼마나 다양한 단어 뒤에서 출현하는지(즉, v가 얼마나 다양한지)를 알아보는 것

  - 다양한 단어 뒤에 나타나는 단어일수록, 훈련 코퍼스에서 보지 못한 단어 시퀀스로 나타날 가능성이 높다

- 예제 문서: { machine learning, deep learning, laptop}
- 단어의 빈도: learning > laptop
- 자유도: learning < laptop ('learning'은 특정 단어 뒤에만 자주 나오는 경향이 있기 때문에)

9.2.3 일반화

- Kneser-Ney 디스카운팅

  - w와 함께 나타난 v들의 집합 {v:Count(v,w)>0}의 크기가 클수록,

    Score은 클 것이라는 가정

$$Score_{continuation} \propto |\{v: Count(v,w) > 0\}|$$

  - Score 식

$$Score_{continuation} = \frac{|\{v: Count(v,w) > 0\}|}{\sum_{w'} |\{v: Count(v,w') > 0\}|}$$

9.2.3 일반화

- 보간(interpolation)

  - 두 개의 다른 언어 모델을 선형적으로 일정 비율(λ)로 섞음

- 예제

  - 일반 영역
    - P(진정제|준비,된) = 0.00001
    - P(사나이|준비,된) = 0.01
  - 특화 영역
    - P(진정제|준비,된) = 0.09
    - P(약|준비,된) = 0.04
  - 보간 결과
    - P(진정제|준비,된)=0.5*0.09 + (1-0.5)*0.00001 = 0.045005

9.2.3 일반화

- 백오프

  - 특정 n-gram의 확률을 n보다 더 작은 시퀀스에 대해 확률을 구하여 보간

  - 예) 3-gram = 2-gram + 1-gram

  - 더 높은 스무딩 및 일반화 효과를 얻을 수 있음

9.3 언어 모델의 평가 방법

- 좋은 언어 모델: 실제 우리가 쓰는 언어와 최대한 비슷하게 확률 분포를 근사하는 모델
- 즉, 많이 쓰이는 문장이나 표현일수록 높은 확률
- 9.3.1 퍼블렉서티(perplexity(PPL))

  - 문장의 길이를 반영하여 확률 값을 정규화 한 값

$$PPL(w_1, w_2, \cdots, w_n) = P(w_1, w_2, \cdots, w_n)^{\frac{1}{n}}$$
$$= \sqrt[n]{\frac{1}{\prod_{i=1}^{n} P(w_i | w_1, \cdots, w_{i-1})}}$$

- PPL은 우리가 뽑어나갈 수 있는 가지의 숫자를 의미(즉, 경우의 수)

8장 텍스트 분류, 9장 언어 모델링

감사합니다