**인공지능 과제2: 영화리뷰 긍정/부정 분류하기**

2014005014

컴퓨터전공

이지민

1. Naïve Bayes Classification 계산 구현

* Label출연율 계산: P(L)

- 학습데이터의 각 label의 document 수를 총 document 수로 나눈 값을 **label출연율 P(L)**로 한다.

* Label 내의 Document출연율 계산: P(D|L)

- 각 label 내의 단어 출연수를 각 label 내의 전체 단어 수로 나눈 값을 **label 내의 단어출연율 P(Wn|L)**로 한다.

- label 내의 전체 단어 수는 중복을 허용한 합계로 한다.

- 해당 document에 포함되는 모든 단어의 P(Wn|L)들의 곱을 **label 내의 document출연율 P(D|L)**로 한다.

* Classification 계산: P(L|D)

- 대소관계만을 보기 때문에 P(L)과 P(D|C)의 곱을 **P(L|D)**로 한다.

- P(L|D)을 최대값으로 하는 label로 구분한다.

2. zero빈도 문제

학습데이터에 없는 새로운 단어가 있으면 P(D|L)이 0이 되면서 label의 확률도 같이 0이 된다. 이것을 해결하기 위해 Additive smoothing방법을 사용한다. P(Wn|L) 계산을 이하와 같은 방법으로 한다.

3. 형태소 분석 사용

데이터 가공을 목적으로 **KoNLPy** 외부 라이브러리를 사용했다.

4. Underflow 대책

log합산을 통해 대소관계를 비교한다.

5. 코드 분석

* NaiveBayes 클래스

1. member 변수
2. vocab

- 학습데이터 전체의 단어의 집합이다.

- set() 초기화를 통해 집합구조를 가지며 중복이 배제된다.

- len()를 통해 학습데이터의 전체 단어 수를 알 수 있다.

- additive smoothing에서 사용된다.

1. word\_count

- 학습데이터의 label별 단어의 집합이다.

- 사전형 자료구조를 사용하며 단어 출연 횟수를 관리한다.

- ex). { label : { word : count } }

1. document\_count

- 학습데이터의 label별 document 수의 집합이다.

- 사전형 자료구조를 사용한다.

- ex). { label : document 수}

1. method 함수
2. train(self,document,label)

- 인자로 받은 document를 형태소 분석을 하고 단어 배열을 생성한다.

- 단어마다 label과 같이 word\_count의 단어 출연 횟수를 갱신한다.

- 단어마다 vocab집합에 추가한다.

- document\_coun집합의 document 수를 갱신한다.

1. \_\_word\_count\_up(self,word,label)

- 인자로 단어와 label을 받아 word\_count의 해당 단어 출연 횟수를 +1증가한다.

- 새로운 label이 들어오면 word\_count에 추가시킨다.

- label 내에 새로운 단어가 들어오면 word\_count에 추가시킨다.

1. \_\_document\_count\_up(self,label)

- 인자로 label을 받아 document\_count의 해당 document 수를 +1증가한다.

- 새로운 label이 들어오면 document\_count에 추가시킨다.

1. classifier(self,document)

- 인자로 받은 document의 classification을 한다.

- 형태소 분석을 통해 단어배열을 생성한다.

- 각 label 별로 P(L|D)를 구하며 이것을 최대값으로 한 label를 return한다.

1. \_\_score(self,label,word\_list)

- 인자로 label과 단어배열을 받아 P(L|D)을 계산 후 return한다.

- underflow 대책으로 log합산을 통해 계산한다.

1. \_\_label\_prob(self,label)

- 인자로 label을 받아 P(L)을 계산 후 return한다.

1. \_\_word\_prob(self,word,label)

- 인자로 단어와 label을 받아 P(Wn|L)을 계산 후 return한다.

- Additive smoothing을 적용하고 있다.

1. \_\_word\_count\_inclueded(self,word,label)

- word\_count에서 label 내의 단어 출연 횟수를 return한다.

* training\_data(nb,path)

- 파일을 읽어 각 행마다 학습시킨다.

* evaluation(nb,path)

- label이 붙어있는 파일을 읽어 classification한 값과 비교해 평가한다.

* test(nb,input\_p,output\_p)

- 파일을 읽어 classification된 새로운 파일을 출력한다.

6. 성능 평가

위와 같은 Naïve Bayes 모델은 additive smoothing에 사용되는 α값에 의해 성능의 차이가 생긴다. ratings\_valid.txt 파일을 이용해 성능 평가를 한다.

|  |  |
| --- | --- |
| **α** | **성능(정답률)** |
| 0.25 | 84.96000000000001% |
| 0.5 | 85.1199% |
| 0.75 | 85.09% |
| 1.0 | 85.15% |

-> 전체적으로 미묘한 차이지만 제일 높게 나온 α = 1.0을 채택한다.