2018학년도 2학기 언어와 컴퓨터

제17–18강 단순 베이즈 분류기와 감정분석 (2–3)

박수지

서울대학교 인문대학 언어학과

2018년 11월 21일 수요일, 26일 월요일

오늘의 목표

1. 분류기를 평가하는 세 가지 척도를 각각 계산할 수 있다.
2. 단순 베이즈 분류기를 파이썬에서 구현할 수 있다.
    1. 데이터 전처리
    2. 훈련: 로그사전확률 및 로그가능도 계산
    3. 실험: 데이터에 대한 확률이 가장 높은 범주 선택

분류기를 평가하는 척도

정확도Accuracy 전체적으로 얼마나 맞추었는가?

정밀도Precision 특정 범주라고 예측된 것이 실제로 얼마나 맞았는가?

재현율Recall 실제로 그 범주인 것이 얼마나 제대로 예측되었는가?

F1 정밀도와 재현율의 조화평균

주의

기준이 되는 범주에 따라 정밀도, 재현율, F1의 값이 달라진다.

정확도·정밀도·재현율의 예시

범죄자와 무고한 시민을 구별하는 문제 (Thanks to Derek Scott)

- 범죄자를 모두 잡고 무고한 시민을 모두 놓아주는 경우 — 이상적
- 범죄자를 모두 놓아주고 무고한 시민을 모두 잡는 경우 — 최악
- 무고한 시민을 놓아주느라 (일부) 범죄자까지 놓아주는 경우
  - 높은 정밀도: 일단 잡힌 사람만 놓고 보면 모두 범죄자이므로
  - 낮은 재현율: 범죄자 중에 잡히지 않은 사람이 있으므로
- 범죄자를 모두 잡느라 (일부) 무고한 시민까지 잡은 경우
  - 낮은 정밀도: 잡힌 사람 중에 무고한 시민이 포함되어 있으므로
  - 높은 재현율: 일단 범죄자만 놓고 보면 모두 잡았으므로

Confusion matrix

	실제 양성	실제: 음성
예측:양성	진양성	위양성
예측:음성	위음성	진음성

$$\text{정확도} = \frac{\text{진양성} + \text{진음성}}{\text{진양성} + \text{위양성} + \text{위음성} + \text{진음성}}$$

$$\text{정밀도} = \frac{\text{진양성}}{\text{진양성} + \text{위양성}}$$

$$\text{재현율} = \frac{\text{진양성}}{\text{진양성} + \text{위음성}}$$

통계적 분류기

$\arg\max_{c \in C} P(c|d)$ 구하기 (d는 데이터, C는 가능한 범주의 집합)

$\arg\max_{c \in C} P(c|d)$ $P(c|d)$가 최대일 때의 c의 값

- 예: $\arg\max_{x \in \mathbb{R}}[-(x-1)^2 + 3] = 1$

SciPy에서 함수 가져오기

```
>>> from scipy import argmax
>>> argmax([1,6,4,2]) # [1,6,4,2] 중 가장 큰 값의 인덱스
1
>>> argmax([2,1,6,4])
2
```

단순 베이즈 분류기

$$P(c|d) = \cdots = \frac{P(c)\prod_{i=1}^{K} P(w_i|c)}{P(d)}$$

$$\arg\max_{c\in C} P(c|d) = \cdots = \arg\max_{c\in C} \left[\log P(c) + \sum_{i=1}^{N} \log P(w_i|c)\right]$$

연습 문제

위에서 $\cdots$ 에 들어갈 식을 도출해 보자.

단순 베이즈 분류기의 목표

$$\arg\max_{c \in C} \left[\log P(c) + \sum_{i=1}^{N} \log P(w_i|c) \right]$$ 구하기

계산할 것

$\log P(c)$ 범주 c의 **로그사전확률**

- 훈련 집합에서 범주 c에 속한 문서가 차지하는 비율

$\log P(w_i|c)$ 단어 w_i에 대한 c의 **로그가능도**

- 범주 c에서 단어 w_i가 차지하는 비율 (+평탄화)

분류기 훈련

로그사전확률

훈련 집합에서 범주 c에 속한 문서가 차지하는 비율

$$\log P(c) = \log \frac{N_c}{N_{doc}}$$

N_c 범주 c에 속한 문서의 개수

N_{doc} 훈련 집합 전체 문서의 개수

코딩에 필요한 것

- 훈련 집합: 문서(단어의 목록)의 목록
- 훈련 집합의 문서를 범주별로 분류해서 저장하기

분류기 훈련

로그가능도

범주 c에서 단어 w가 차지하는 비율 (+평탄화)

$$\log P(w|c) = \log \frac{count(w,c)+1}{\sum_{w' \in V} [count(w',c)+1]}$$

$count(w,c)$ 범주 c에서 단어 w가 출현한 횟수
V 훈련 집합의 어휘 목록

코딩에 필요한 것

- 범주별로 문서를 합쳐서 단어(token)를 세기
- 훈련 집합에 출현한 어휘(type) 목록 만들기

훈련 결과 시험

훈련 결과 얻은 것

로그사전확률 $\log P(c)$, 로그가능도 $\log P(w|c)$

시험할 것

실험 집합의 문서 $testdoc$에 대한 $\arg\max_{c \in C} \log P(c|testdoc)$

$$\arg\max_{c \in C} \log P(c|testdoc)$$

$$= \arg\max_{c \in C} \left[\log P(c) + \sum_{w \in testdoc \cap V} \log P(w|c) \right]$$

function TRAIN NAIVE BAYES(D, C) **returns** log $P(c)$ and log $P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return $logprior$, $loglikelihood$, V

function TEST NAIVE BAYES($testdoc$, $logprior$, $loglikelihood$, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in $testdoc$
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$

Figure 4.2 The naive Bayes algorithm, using add-1 smoothing. To use add-α smoothing instead, change the $+1$ to $+\alpha$ for loglikelihood counts in training.

실습 코드: naivebayes_practice.py

주어진 것

- 코퍼스 (훈련 집합 + 실험 집합)
- 범주 목록

산출할 것

- 로그사전확률 — 범주별
- 로그가능도 — 단어별, 범주별
- 사후확률이 가장 높은 범주 — 문서별

남은 문제

- 모든 단어(혹은 형태소)를 특성으로 쓰는 것이 적절한가?
- 세 가지 이상의 범주에 대한 분류

다른 분류기

로지스틱 회귀분석 / 최대 엔트로피 분류기

- $P(d|c)$를 사용하지 않고 $P(c|d)$를 직접 추정한다.
- 단어 이외의 특성을 다양하게 사용할 수 있다.