

나이프 베이지 스팸 필터



부산대학교 정보·의생명공학대학
정보컴퓨터공학부



개요

- ❖ 베イズ 정리
- ❖ 나이브 베이즈를 이용한 스팸 필터
- ❖ 나이브 베이즈를 이용한 스팸 필터 예제 코드

나이브 베이즈를 이용한 스팸 필터

나이브 베이즈 분류기 Naïve Bayes Classifier

- ❖ Bayesian approach to classifying a new instance is to assign the most probable target value, v_{MAP} , given an attribute values $\langle a_1, a_2, \dots, a_n \rangle$ that describe an instance.

$$v_{MAP} = \operatorname{argmax}_{v_j \in V} P(v_j | a_1, a_2 \dots a_n)$$

$$= \operatorname{argmax}_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)}$$

$$= \operatorname{argmax}_{v_j \in V} P(a_1, a_2 \dots a_n | v_j) P(v_j)$$

Bayes rule applied

Output by the Naïve Bayes classifier

$$v_{NB} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Assumption: attribute values are conditionally independent given the target value

$P(v)$, $P(a|v)$ terms can be estimated from the training data

나이브 베이즈 분류기 적용 예

❖ task:


- Predict the target value(yes or no) of PlayTennis
given a new instance with *<unny, cool, high, strong>*

$$v_{NB} = \operatorname{argmax}_{v_j \in \{yes, no\}} P(v_j) \prod_i P(a_i | v_j)$$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$P(yes) P(sunny|yes) P(cool|yes) P(high|yes) P(strong|yes) = .0053$$

$$P(no) P(sunny|no) P(cool|no) P(high|no) P(strong|no) = .0206$$

 *no*

$$❖ P(no | \langle sunny, cool, high, strong \rangle) = \frac{.0206}{.0206 + .0053} = .795$$

스팸 필터

Assume that instance D described by n -dimensional vector of attributes

$$D = \langle w_1, w_2, \dots, w_n \rangle$$

Naïve Bayes Conditional Independence Assumption:

then $c_{MAP} = \operatorname{argmax}_{c \in C} P(c \mid w_1, w_2, \dots, w_n)$

$$= \operatorname{argmax}_{c \in C} \frac{P(w_1, w_2, \dots, w_n \mid c)P(c)}{P(w_1, w_2, \dots, w_n)}$$

$$= \operatorname{argmax}_{c \in C} P(w_1, w_2, \dots, w_n \mid c)P(c)$$

$$P(w_1, w_2, \dots, w_n \mid c_j) = \prod_i P(w_i \mid c_j)$$

- ❖ $C = \{\text{spam}, \text{ham}\}$, x_n : keyword (e.g., bitcoin, gold, interest)
 D : bag of words model

스팸 필터 학습

- From training corpus, extract *Vocabulary*
- Calculate required estimates of $P(c)$ and $P(w|c)$ terms,
 - For each c_j in C do

$$P(c) \leftarrow \frac{\text{count}_{docs}(C = c)}{|docs|}$$

where $\text{count}_{docs}(x)$ is the number of documents for which x is true.

- For each word $w_i \in \text{Vocabulary}$ and $c \in C$, where $\text{count}_{doctokens}(x)$ is the number of tokens over *all* documents for which x is true of that document and that token...

$$P(w_i | c) \leftarrow \frac{\text{count}_{doctokens}(W = w_i, C = c)}{\text{count}_{doctokens}(C = c)}$$

Issues

- ❖ Underflow Problem: multiplying lots of probabilities can results in floating-point underflow.

➡
$$c_{NB} = \operatorname{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(w_i | c_j)$$

- ❖ Zero conditional probability Problem

➡ **Laplace smoothing**

$$P(w_i | c) \leftarrow \frac{\text{count}_{\text{doctokens}}(W = w_i, C = c) + \alpha}{\text{count}_{\text{doctokens}}(C = c) + \beta}$$