WEEK 2

SENTIMENT ANALYSIS WITH NAÏVE BAYES.

• Probability and Bayes' Rule

A → Positive tweet

$P(A) = P(\text{Positive}) = N_{pos} / N = 0.65$

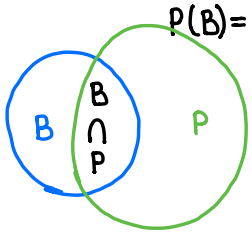
$P(\text{Negative}) = 1 - P(\text{Positive}) = 0.35$

Tweets containing the word "happy"

B → tweet contains "happy"

$P(B) = P(\text{happy}) = N_{happy} / N$

$P(B) = 4/20 = 0.2$



$P(B \cap P) = P(B, P) = \frac{3}{20} = 0.15$

$P(P|B) = P(\text{Positive} | \text{"happy"}) = 3/4 = 0.75 \rightarrow$ 75% of likelihood of being positive if tweet contains the word "happy".

$P(B|P) = P(\text{"happy"} | \text{Positive}) = 3/13 = 0.231 \rightarrow$ 23.1% of likelihood of containing the word "happy" if the tweet is positive.

$$P(\text{Positive} | \text{"happy"}) = \frac{P(\text{Positive} \cap \text{"happy"})}{P(\text{"happy"})}$$

$$P(\text{"happy"} | \text{Positive}) = \frac{P(\text{"happy"} \cap \text{Positive})}{P(\text{Positive})}$$

BAYES' RULE

$$P(\text{Positive} | \text{"happy"}) = P(\text{"happy"} | \text{Positive}) \times \frac{P(\text{Positive})}{P(\text{"happy"})}$$

$$P(X|Y) = P(Y|X)\frac{P(X)}{P(Y)}$$

• Naïve Bayes

this method makes the assumption that the features we are using for classification are all independent, which in reality is rarely the case.

$P(w_i|class)$

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2
Nclass	13	13

word	Pos	Neg
I	0.24	0.24 → 3/12
am	0.24	0.24
happy	0.15	0.08
because	0.08	0
learning	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.15
not	0.08	0.15

power words.

Naïve Bayes inference condition rule for binary classification

$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}$$

"I am happy today; I am learning"

$$\frac{0.2}{0.2} \times \frac{0.2}{0.2} \times \frac{0.14}{0.10} \times \frac{0.2}{0.2} \times \frac{0.2}{0.2} \times \frac{0.1}{0.1}$$

→ smooth the values.

$= 1.4 > 1$ (positive class).

• Laplacian Smoothing

(to avoid probabilities being zero)

$$P(w_i|class) = \frac{freq(w_i, class)}{N_{class}}$$

(smoothing)

$$\rightarrow P(w_i|class) = \frac{freq(w_i, class) + 1}{N_{class} + V}$$

V: # of unique words in a vocabulary.

word	Pos	Neg
I	3	3
⋮	⋮	⋮
Nclass	13	13

$$P(I|Pos) = \frac{3+1}{13+8} = 0.19$$

•Log Likelihood.

$$ratio(w_i) = \frac{P(w_i|Pos)}{P(w_i|Neg)}$$

∞ ↑ positive
1 neutral
0 ↓ negative

Naïve Bayes' inference:

$$\underbrace{\frac{P(pos)}{P(neg)}}_{\text{prior ratio}} \underbrace{\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}}_{\text{likelihood}}$$

- products bring risk of underflow
- $\log(a \cdot b) = \log(a) + \log(b)$
- $\log\left[\frac{P(pos)}{P(neg)} \prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}\right]$

$$\underbrace{\log \frac{P(pos)}{P(neg)}}_{\text{log prior}} + \underbrace{\sum_{i=1}^{m} \log \frac{P(w_i|pos)}{P(w_i|neg)}}_{\text{log likelihood}}$$

-calculating Lambda

$$\lambda(w) = \log \frac{P(w|pos)}{P(w|neg)}$$

word	Pos	Neg	λ
I	0.05	0.05	$0 \rightarrow \log \frac{0.05}{0.05} = \log 1 = 0$
am	0.04	0.04	0
happy	0.09	0.01	$2.2 \rightarrow \log \frac{0.09}{0.01}$

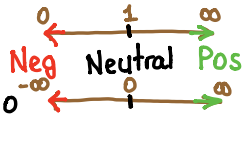
word sentiment $\begin{cases} ratio(w) = \frac{P(w|pos)}{P(w|neg)} \\ \lambda(w) = \log \frac{P(w|pos)}{P(w|neg)} \end{cases}$

tweet: "I am happy because I am learning."

log likelihood = 0+0+2.2+0+0+0+1.1 = 3.3 > 0

$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)} > 1$

$\sum_{i=1}^{m} \log \frac{P(w_i|pos)}{P(w_i|neg)} > 0$



•Training Naïve Bayes

0. Collect and annotate corpus.
1. Preprocess (lowercase, remove punctuation, urls, names, stop words, stemming, tokenize sentences)
2. Word count

3. P(w|class) by using Laplacian smoothing $\frac{freq(w,class)+1}{N_{class}+V_{class}}$

4. Get lambda $\lambda(w) = \log \frac{P(w|pos)}{P(w|neg)}$

5. Get the log prior: $logprior = \log \frac{D_{pos}}{D_{neg}}$ → # of positive tweets

• Testing Naïve Bayes.

predict

- log-likelihood dictionary $\lambda(w) = \log \frac{P(w|pos)}{P(w|neg)}$
- $logprior = \log \frac{D_{pos}}{D_{neg}}$

"I passed the NLP interview"

↓

[I, pass, the, NLP, interview]

↓

$score = -0.01 + 0.5 - 0.01 + 0 + \cancel{logprior}^{0} = 0.48 > 0 \rightarrow$ positive.

- X_{val} Y_{val} λ logprior

$score = predict(X_{val}, \lambda, logprior)$

$pred = score > 0$

$accuracy = \frac{1}{m} \sum_{i=1} (pred_i == Y_{val_i})$

(note: the words that do not appear in $\lambda(w)$ are treated as neutral words)

• Applications of Naïve Bayes

$P(pos|tweet) \approx P(pos)\ P(tweet|pos)$ → Author identification

$P(neg|tweet) \approx P(neg)\ P(tweet|neg)$ → Spam filtering

$\frac{P(pos|tweet)}{P(neg|tweet)} = \underbrace{\frac{P(pos)}{P(neg)}}_{priors} \underbrace{\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}}_{likelihood}$ → Information retrieval

→ Word disambiguation

• Naïve Bayes Assumptions.

- Independence btw the predictors or features.
- Relative frequency in corpus

→Independence assumption could lead us to under or over estimate the conditional probabilities of individual words.

→Relative frequency assumption relies on the distribution of the training data sets: proportion of positive and negative classes → affect the model.

• Error Analysis

→removing punctuation and stop words → semantic may lost

→word order → affects the meaning of a sentence

→adversarial attacks → confuse naive Bayes models.

"My beloved grandmother :("

"I am happy because I did not go." ↔ "I am not happy because I did go."