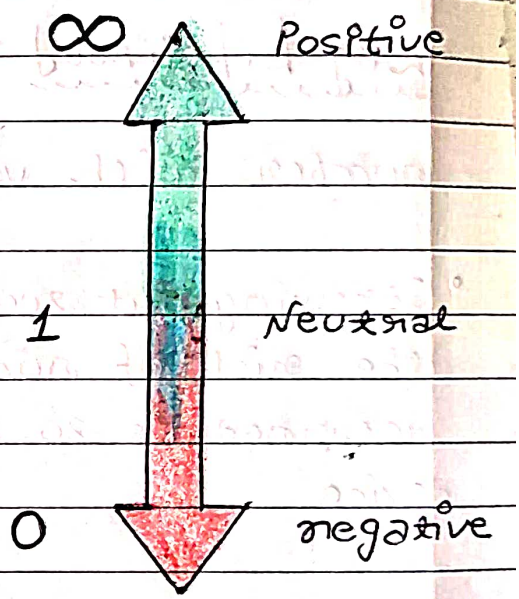
Log Likelihoods

Words can have many shades of emotional meaning, but for the purpose of sentiment classification, they are simplified into three categories : neutral, positive & negative

All can be identified by using their conditional probabilities →

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

word	Pos	neg	ratio
I	0.20	0.20	1
am	0.20	0.20	1
happy	0.14	0.10	1.4
because	0.10	0.10	1
learning	0.10	0.10	1
NLP	0.10	0.10	1
sad	0.10	0.15	0.6
not	0.10	0.15	0.6



The larger the ratio the more positive the word is going to be.

Now, on the other hands, negative words have a ratio smaller than 1.

$$\approx \frac{freq(w_i, 1) + 1}{freq(w_i, 0) + 1}$$

Naive Bayes Inference

class $\in$ {pos, neg}

$w \rightarrow$ set of m words in a tweet

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

- A simple, fast and powerful baseline.
- A probabilistic method used for classification

Log Likelihood → • Sentiments probability ca
requires multiplication of
numbers with values between 0 an 1.

- Carrying out such multiplications on a computer
the risk of numerical underflow when the num
returned is so small if it can't be stored
device

- mathematical trick to solve this —
$\log(a * b) = \log(a) + \log(b)$

- $$\log\left(\frac{P(pos)}{P(neg)} \prod_{i=1}^{n} \frac{P(w_i|Pos)}{P(w_i|Neg)}\right)$$

log (prior multiplied by likelihood)

=> log (prior) + log (likelihood)

or

$$\log \frac{P(Pos)}{P(neg)} + \sum_{i=1}^{n} \log \frac{P(w_i|Pos)}{P(w_i|neg)}$$

Calculating Lambda —

	pos	neg	λ
	0.05	0.05	0
	0.04	0.04	0
	0.09	0.01	2.2
	0.01	0.01	0
	0.03	0.01	1.1
	0.02	0.02	0
	0.01	0.09	-2.2
	0.02	0.03	-0.4

$\lambda(I) = \log \frac{0.05}{0.05} \Rightarrow 0$

$\lambda(am) = \log \frac{0.04}{0.04} \Rightarrow 0$

$\lambda(happy) = \log \frac{0.09}{0.01} \rightarrow \log 9 \rightarrow 2.2$

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

word sentiment

$$ratio(w) = \frac{P(w|Pos)}{P(w|neg)}$$

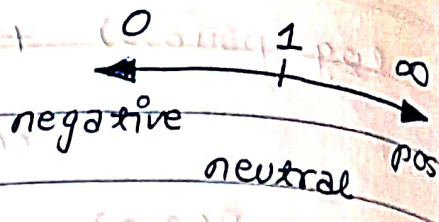
$$\lambda(w) = \log \frac{P(w|Pos)}{P(w|neg)}$$

I am happy because I am learning

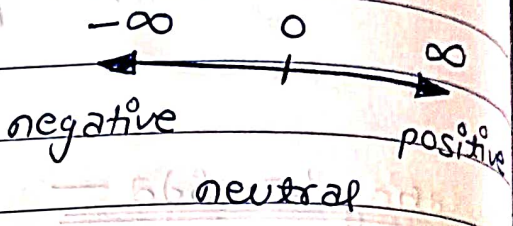
⇒ 0 + 0 + 2.2 + 0 + 0 + 0 + 1.1

⇒ 3.3

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



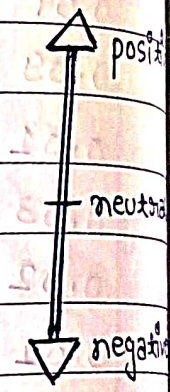
$$\sum_{i=1}^{m} \log \frac{p(w_i|pos)}{p(w_i|neg)} > 0$$



$3.3 > 0$

Tweet sentiment

$$\log \prod_{i=1}^{m} ratio(w_i) = \sum_{i=1}^{m} \lambda(w_i) > 0$$



aining Naive Bayes

p0: Collect and Annotate Corpus

positive tweets
- am happy because I am learning NLP
- am happy, not sad. @ NLP

negative tweets
- I am sad, I am not learning NLP
- I am sad, not happy !!

p1: Data-Preprocessing

- Lowercase
- Remove punctuation, urls, names
- Remove Stopwords
- Stemming
- Tokenize Sentences

positive tweets
- [happi, because, learn, NLP]
- [happi, not, sad]

negative tweets
- [sad, not, learn, NLP]
- [sad, not, happi].

p2: word-count

ce, you have a clean corpus of processed tweets, ou will start by computing the vocabulary for each ? ord in class,

freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

Step 3: $P(w|class)$

$V_{class} = 6$

$$\frac{freq(w, class) + 1}{N_{class} + V_{class}}$$

word	Pos	Neg
happi	0.23	0.15
because	0.15	0.07
learn	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.17
not	0.08	0.17

Step 4: Get Lambda (λ)

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

word	pos	neg	λ
happi	0.23	0.15	0.43
because	0.15	0.07	0.6
learn	0.08	0.08	0
NLP	0.08	0.08	0
sad	0.08	0.17	-0.75
not	0.08	0.17	-0.75

Step5: Get the log prior

D_{pos} = Nos of positive tweets

D_{neg} = Nos of negative tweets

$$\text{log prior} = \log \frac{D_{pos}}{D_{neg}}$$

If dataset is balanced, $D_{pos} = D_{neg}$ and log prior = 0

Testing Naive Bayes →

- Predict using a Naive Bayes Model
- Using your validation set to compute model accuracy

word	λ
I	-0.01
the	-0.01
happi	0.63
because	0.01
pass	0.5
NLP	0
sad	-0.75
not	-0.75

- log-likelihood dictionary

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

- log prior $= \log \frac{D_{pos}}{D_{neg}} = 0$

- Tweet : (check)

[I, pass, the, NLP, interview]

score : $= -0.01 + 0.5 - 0.01 + 0 + 0 + \text{log prior}$

$\Rightarrow 0.48$

pred = score > 0 (positive)

- Xval, yval, λ, log_prior

score = predict(Xval, λ, log_prior)

$$\text{pred} = \text{score} > 0 \begin{bmatrix} 0.5 \\ -1 \\ 1.3 \\ \vdots \\ \text{score}_m \end{bmatrix} > 0 \Rightarrow \begin{bmatrix} 0.5 > 0 \\ -1 > 0 \\ 1.3 > 0 \\ \vdots \\ \text{score}_m > 0 \end{bmatrix} \Rightarrow$$

produces a vector populated with 0's & 1's indicating if the predicted sentiment is negative or positive.

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ pred_m \end{bmatrix} == \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ y_{val_m} \end{bmatrix}$$

0 - 0 → 1
1 - 0 → 0
1 - 1 → 1

$$\frac{1}{m}\sum_{i=1}^{m}(pred_i == y_{val_i}) \Rightarrow \begin{bmatrix} 1 \\ 0 \\ 1 \\ \\ pred_m == y_{val_m} \end{bmatrix}$$

Summary • X_{val} y_{val} → performance on unseen data

predict using λ and logprior for each new tweet

Accuracy → $\frac{1}{m}\sum_{i=1}^{m}(pred_i == y_{val_i})$

What about words that don't appear in $\lambda(w)$?

Applications of Naive Bayes :

① Twitter Sentiment Analysis

$P(pos \mid tweet) = P(pos)\, P(tweet \mid pos)$

$P(neg \mid tweet) = P(neg)\, P(tweet \mid neg)$

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

priors — likelihood

Estimating the probability for each class by using the joint probability of the words in classes

(2) Identification system

i.e. Author Identification : $\frac{P(\text{"shakespeare"} \mid \text{book})}{P(\text{"meadly"} \mid \text{book})}$

If you have two large corporal, each written by two different authors, you could train a model to recognize whether a new document was written by one or another

(3) Word disambiguation : Consider that you have only two possible interpretations of a given word with text.

Let, say you don't know if the word bank in reading is reffering to the bank of a river or to financial institution

To disambiguate your word calculate the score of documents, given that it reffers to each one of the possible meanings.

In this case, if the text refers to the concept of river instead of the concept of money, then the score will be bigger than one.

$\frac{P(\text{river} \mid \text{text})}{P(\text{money} \mid \text{text})}$