احتمالات

$$P(A) = \frac{n_A}{n_{total}}$$

احتمال وقوع A به شرط که اتفاق افتاده B باشد

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

$$P(A \cap B) = P(A|B)\,P(B) = P(B|A)\,P(A)$$

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$ → Bayse Rule

نکته: کلماتی که در کلاس + و − احتمال تقریباً یکسانی دارند معمولاً احساسی به جمله القا نمی‌کنند.

توی مثال الان: $class \in \{+, -\}$

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

n_{class} → تعداد کلمات کلاس

$$= \frac{freq(w_i, class) + 1}{n_{class} + V}$$

nominator (+1), demenator (V) → سایز vocab

laplacian smoothing

اگر کلمه توی tweet نباشه صفر می‌شه. برای جلوگیری از این موضوع از تکنیک smoothing استفاده می‌کنیم. همه فرمول رو به رو

(چون صفر جلوی قیاس بین کلاس + و − رو می‌گیره)

$$ratio(w_i) = \frac{P(w_i|+)}{P(w_i|-)} \approx \frac{freq(w_i, 1) + 1}{freq(w_i, 0) + 1}$$

منفی (0) ← خنثی (1) → مثبت (∞)

ضرب می‌کنیم تا کلمات خنثی با ratio=1 بی‌اثر باشن.

naive baysian

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

زمانی که تعداد m زیاد باشه دچار underflow می‌شیم و با log گرفتن میشه جلوشو گرفت

$$\log \frac{P(+)}{P(-)} + \sum_{i=1}^{m} \log \frac{P(w_i|+)}{P(w_i|-)}$$

log prior ($\log \frac{P(+)}{P(-)}$), log liklihood ($\log \frac{P(w_i|+)}{P(w_i|-)} = \lambda(w)$)

یادآوری: $\log(1) = 0$

منفی ($-\infty$) ← خنثی (0) → مثبت ($+\infty$)

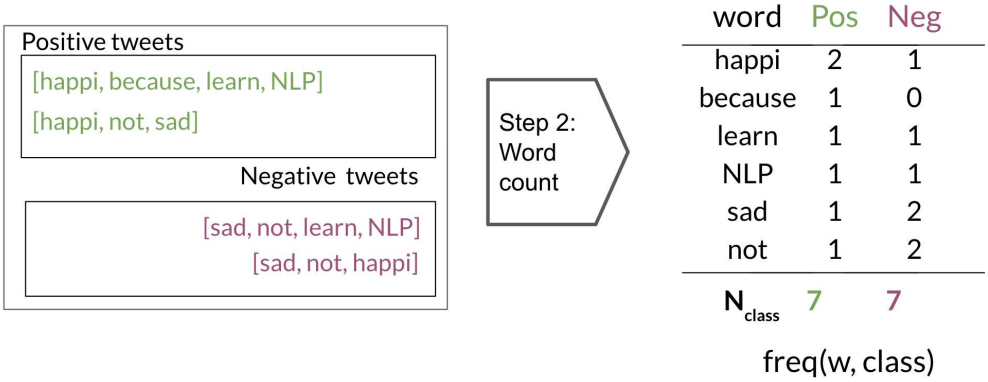
To train your naïve Bayes classifier, you have to perform the following steps:

1) Get or annotate a dataset with positive and negative tweets

2) Preprocess the tweets: process_tweet(tweet) ➞ [w1, w2, w3, ...]:

- Lowercase
- Remove punctuation, urls, names
- Remove stop words
- Stemming
- Tokenize sentences

3) Compute 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

freq(w, class)

4) Get $P(w|pos), P(w|neg)$

You can use the table above to compute the probabilities.

5) Get $\lambda(w)$

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

6) Compute $logprior = \log(P(pos)/P(neg))$

$\text{logprior} = \log \frac{D_{pos}}{D_{neg}}$, where D_{pos} and D_{neg} correspond to the number of positive and negative documents respectively.

- log-likelihood dictionary $\lambda(w) = log\frac{P(w|pos)}{P(w|neg)}$
- $logprior = log\frac{D_{pos}}{D_{neg}} = 0$
- Tweet: [I, pass, the, NLP, interview]

$$score = -0.01 + 0.5 - 0.01 + 0 + logprior = 0.48$$

$$pred = score > 0$$

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

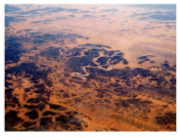
The example above shows how you can make a prediction given your λ dictionary. In this example the *logprior* is 0 because we have the same amount of positive and negative documents (i.e. $\log 1 = 0$).

There are many applications of naive Bayes including:

- Author identification
- Spam filtering → $\frac{P(spam|email)}{P(nonspam|email)}$
- Information retrieval → $P(document|query) \propto \prod_{i=0} (query_i|document)$
- Word disambiguation → $\frac{P(river|text)}{P(money|text)}$

This method is usually used as a simple baseline. It also really fast.

Naïve Bayes makes the independence assumption and is affected by the word frequencies in the corpus. For example, if you had the following



"It is sunny and hot in the Sahara desert."

"It's always cold and snowy in ___."

In the first image, you can see the word sunny and hot tend to depend on each other and are correlated to a certain extent with the word "desert". Naive Bayes assumes independence throughout. Furthermore, if you were to fill in the sentence on the right, this naive model will assign equal weight to the words "spring, summer, fall, winter".

Relative frequencies in corpus

On Twitter, there are usually more positive tweets than negative ones. However, some "clean" datasets you may find are artificially balanced to have to the same amount of positive and negative tweets. Just keep in mind, that in the real world, the data could be much noisier.

There are several mistakes that could cause you to misclassify an example or a tweet. For example,

- Removing punctuation
- Removing words

Tweet: This is not good, because your attitude is not even close to being nice.

processed_tweet: [good, attitude, close, nice]

Tweet: My beloved grandmother :(

processed_tweet: [belov, grandmoth]

- Word order

Tweet: I am happy because I did not go.

Tweet: I am not happy because I did go.

- Adversarial attacks

These include sarcasm, irony, euphemisms.