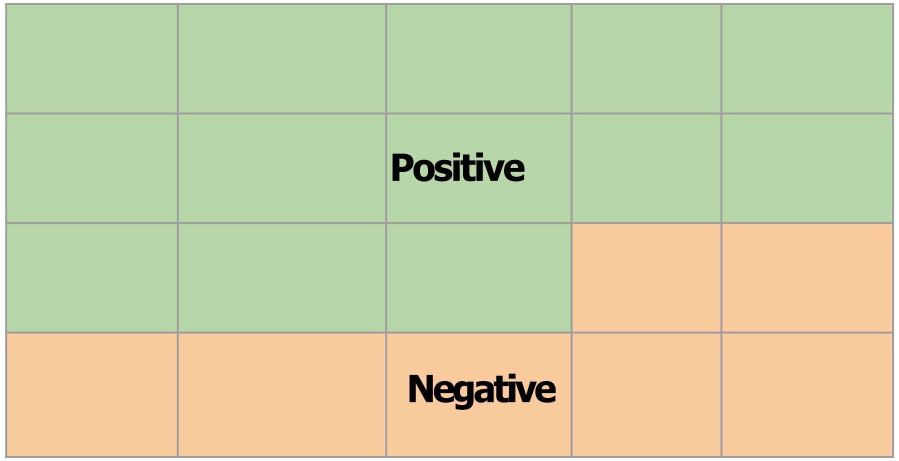
Setiment Analysis with Naive Bayes

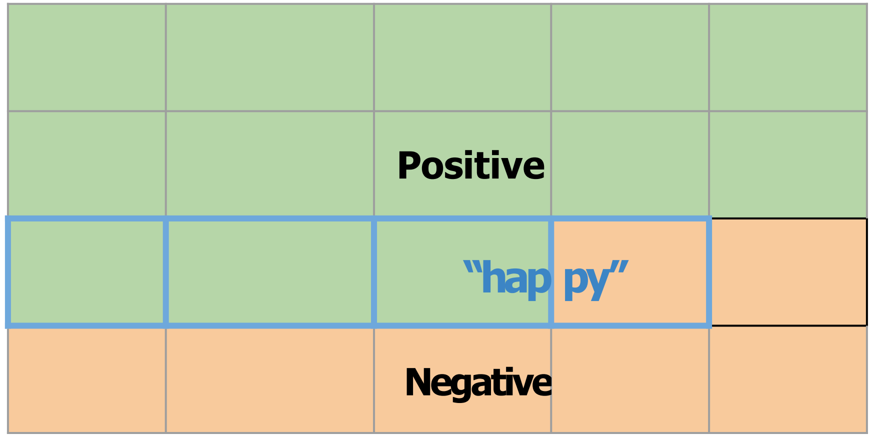
- Conditional Probability
- Bayes' rule
- Laplace smoothing
- Ratio of probabilities
- Log likelihood analysis
- Training, testing Naïve Bayes
- Applications, Sources of Errors in in Naïve Bayes

Probability and Bayes' Rule

Corpus of tweets



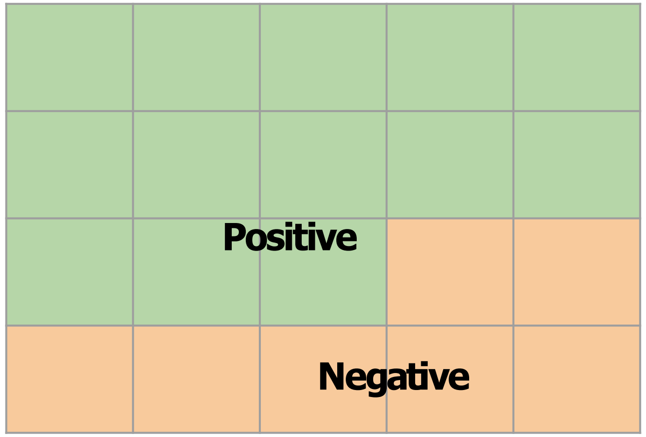
Tweets containing the word "happy"



- Corpus of tweets
  - Tweets that can be categorized as either positive or negative sentiment
  - The word happy is sometimes being labeled positive and sometimes negative

Probabilities

Corpus of tweets



$A \rightarrow$ Positive tweet

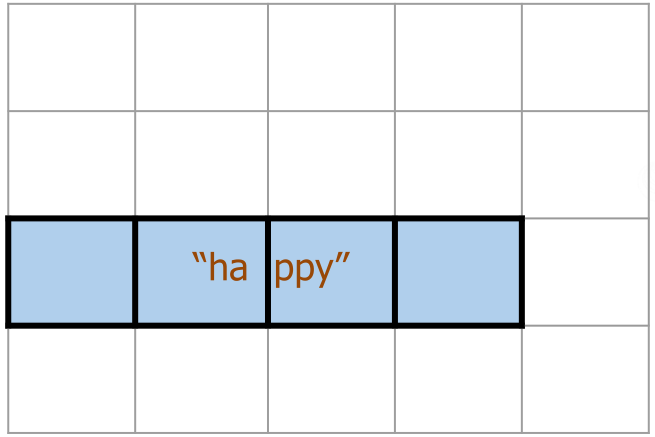
$P(A) = P(\text{Positive}) = N_{pos} / N$

$A \rightarrow$ Positive tweet

$P(A) = N_{pos} / N = 13 / 20 = 0.65$

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

Tweets containing the word "happy"

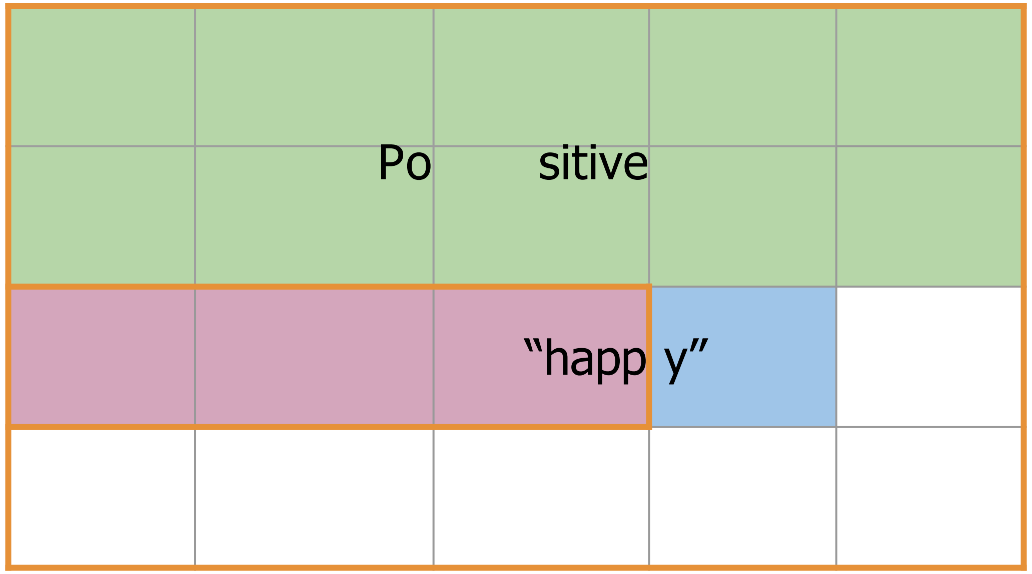


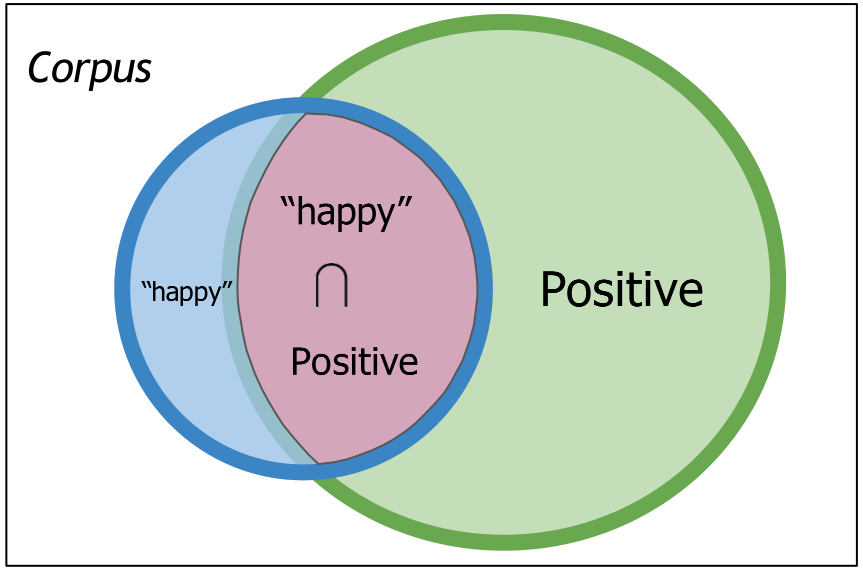
$B \rightarrow$ tweet contains "happy".

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

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

Probability of the intersection





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

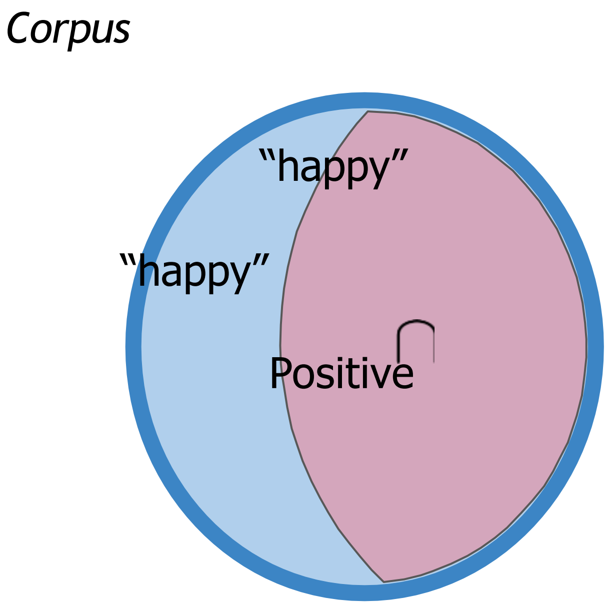
Bayes’ Rule

- Conditional Probabilities

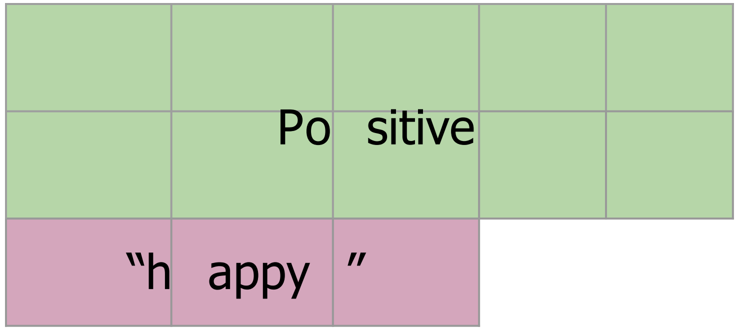
Pos itive “happ y”

$P(A \mid B) = P(\text{Positive} \mid \text{“happy”})$

$P(A \mid B) = 3 / 4 = 0.75$

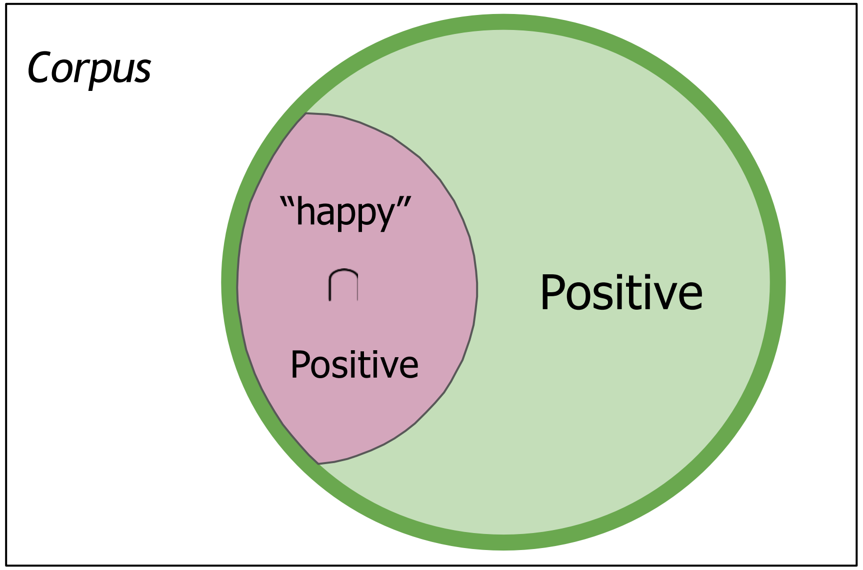


Conditional Probabilities



$$P(B \mid A) = P(\text{"happy"} \mid \text{Positive})$$

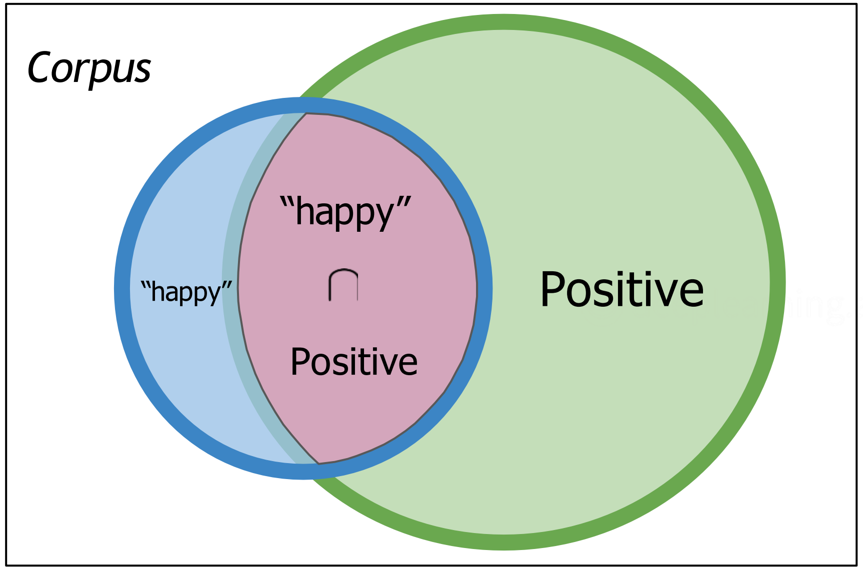
$$P(B \mid A) = 3 / 13 = 0.231$$



Conditional probabilities

Probability of B, given A happened

Looking at the elements of set A, the chance that one also belongs to set B



$$P(\text{Positive} | \text{“happy”}) =$$

$$\frac{P(\text{Positive} \cap \text{“happy”})}{P(\text{“happy”})}$$

Bayes' rule

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

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

Naïve Bayes for Sentiment Analysis

Positive tweets

I am happy because I am learning NLP
I am happy, not sad.

Negative tweets

I am sad, I am not learning NLP
I am sad, not happy

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
N_{class}	**13**	**12**

Naïve Bayes for Sentiment Analysis

FPT Education
FPT UNIVERSITY

- For a document d, out of all classes returns the maximum posterior probability

$$\hat{c} = \operatorname{argmax} P(\text{class}|W) = \operatorname{argmax} \frac{P(W|\text{class})P(\text{class})}{P(W)}$$

- $P(W)$ does not change for each class => drop denominator

$$\operatorname{argmax} P(\text{class}|W) = \operatorname{argmax} P(W|\text{class})P(\text{class})$$

$$\hat{c} = \operatorname{argmax} \underbrace{P(w_1, w_2, \ldots, w_n|\text{class})}_{\text{likelihood}} \underbrace{P(\text{class})}_{\text{prior}}$$

- Assumption that the probabilities $P(w_i|class)$ are independent

$$P(w_1, w_2 \ldots, w_n|\text{class}) = P(w_1|\text{class}).\, P(w_2, |\text{class}) \ldots .\, P(w_n, |\text{class})$$

$$\hat{c}_{NB} = \operatorname{argmax} P(\text{class}) \prod P(w_i|\text{class})$$

P(wi | 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**	**12**

$$p(I|Neg) = \frac{3}{12}$$

word	Pos	Neg
I	0.24	0.25
am	0.24	0.25
happy	0.15	0.08
because	0.08	0.00
learning	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.08
not	0.08	0.17
Sum	**1**	**1**

P(wi|class)

word	Pos	Neg
I	0.24	0.25
am	0.24	0.25
happy	0.15	0.08
because	0.08	0
learning	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.08
not	0.08	0.17

Naïve Bayes

Tweet: I am happy today; I am learning.

$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)} = \frac{0.14}{0.10} = 1.4 \quad \mathbf{>1}$$

$$\cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.20}{0.20}} * \boxed{\frac{0.14}{0.10}} * \cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.10}{0.10}}$$

word	Pos	Neg
I	0.20	0.20
am	0.20	0.20
happy	0.14	0.10
because	0.10	0.05
learning	0.10	0.10
NLP	0.10	0.10
sad	0.10	0.10
not	0.10	0.15

Laplacian Smoothing

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

class ∈ {Positive, Negative}

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

N_{class} = frequency of all words in class

V_{class} = number of unique words in class

FPT Education
FPT UNIVERSITY

Introducing $P(w_i \mid class)$ with smoothing

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	12

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

V = 8

word	Pos	Neg
I	0.19	

?

Introducing $P(w_i \mid class)$ with smoothing

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	12

$V = 8$

word	Pos	Neg
I	0.19	0.20
am	0.19	0.20
happy	0.14	0.10
because	0.10	0.05
learning	0.10	0.10
NLP	0.10	0.10
sad	0.10	0.15
not	0.10	0.15
Sum	1	1

Log Likelihood, Part 1- Ratio of probabilities

Positive ∞

Neutral 1

Negative 0

word	Pos	Neg	ratio
I	0.19	0.20	1
am	0.19	0.20	1
happy	0.14	0.10	1.4
because	0.10	0.05	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

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

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

Naïve Bayes' inference

class ∈ *{pos, neg}*
w *-> Set of m words in a tweet*

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

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

Log Likelihood

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

- Products bring risk of underflow
- $log(a * b) = log(a) + log(b)$
- $$log(\frac{P(pos)}{P(neg)} \prod_{i=1}^{n} \frac{P(w_i|pos)}{P(w_i|neg)}) \Rightarrow log\frac{P(pos)}{P(neg)} + \sum_{i=1}^{n} log\frac{P(w_i|pos)}{P(w_i|neg)}$$

log prior + log likelihood

Calculating Lambda

tweet: I am happy because I am learning.

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

$$\lambda(I) = log\frac{0.05}{0.05} = log(1) = 0$$

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

word	Pos	Neg	λ
I	0.05	0.05	0
am	0.04	0.04	0
happy	0.09	0.01	2.2
because	0.01	0.01	0
learning	0.03	0.01	0
NLP	0.02	0.02	1.1
sad	0.01	0.09	-2.2
not	0.02	0.03	-0.4

Summing the Lambdas

doc: I am happy because I am learning.

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

$$\lambda(\text{happy}) = log\frac{0.09}{0.01} \approx 2.2$$

Word sentiment

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

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

word	Pos	Neg	λ
I	0.05	0.05	0
am	0.04	0.04	0
happy	0.09	0.01	2.2
because	0.01	0.01	0
learning	0.03	0.01	1.1
NLP	0.02	0.02	0
sad	0.01	0.09	-2.2
not	0.02	0.03	-0.4

Log Likelihood Part 2

doc: I am happy because I am learning.

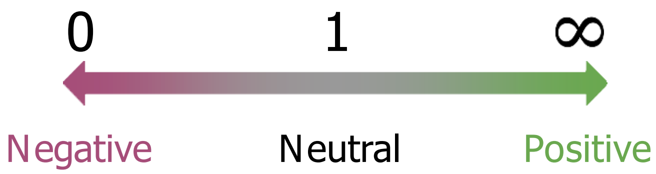
$$\sum_{i=1}^{m} log \frac{P(w_i|pos)}{P(w_i|neg)} = \sum_{i=1}^{m} \lambda(w_i)$$

word	Pos	Neg	λ
I	0.05	0.05	0
am	0.04	0.04	0
happy	0.09	0.01	2.2
because	0.01	0.01	0
learning	0.03	0.01	1.1
NLP	0.02	0.02	0
sad	0.01	0.09	-2.2
not	0.02	0.03	-0.4

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

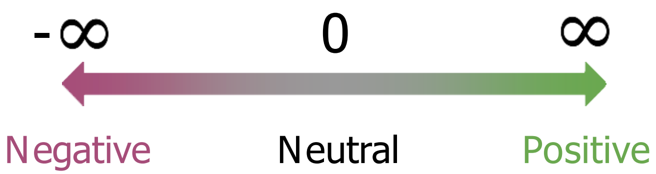
Log Likelihood

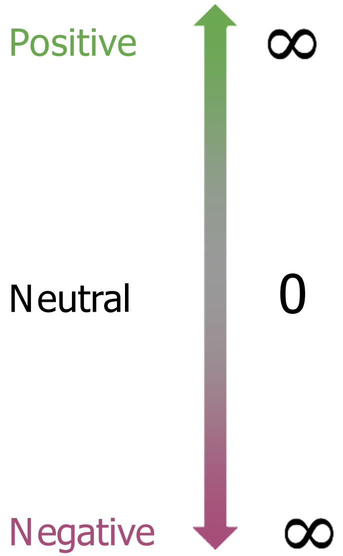
$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)} > 1$$



3.3 > 0

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





Tweet sentiment:

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

Training Naïve Bayes

Step 0: Collect and annotate corpus

Positive tweets

I am happy because I am learning NLP @NLP

I am happy, not sad.

Negative tweets

I am sad, I am not learning NLP

I am sad, not happy!!

Step 1: Preprocess

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

Positive tweets

[happi, because, learn, NLP]

[happi, not, sad]

Negative tweets

[sad, not, learn, NLP]

[sad, not, happi]

Training Naïve Bayes

Positive tweets

[happi, because, learn, NLP]

[happi, not, sad]

Negative tweets

[sad, not, learn, NLP]

[sad, not, happi]

Step 2: Word count

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

Training Naïve Bayes

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: $P(\mathrm{w}|\mathrm{class})$

$$V_{\text{class}} = 6$$

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

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

Step 4: Get lambda

word	Pos	Neg	λ
happy	0.23	0.15	0.43
because	0.15	0.07	0.6
learning	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

Training Naïve Bayes

FPT Education
FPT UNIVERSITY

Step 5:
Get the
log prior

D_{pos} = Number of positive tweets
D_{neg} = Number of negative tweets

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

If dataset is balanced, $D_{pos} = D_{neg}$ and logprior $= 0$.

Summary

FPT Education
FPT UNIVERSITY

1. Get or annotate a dataset with positive and negative tweets
2. Preprocess the tweets: process_tweet(tweet) ➞ $[w_1, w_2, w_3, \ldots]$
3. Compute freq(w, class)
4. Get $P(w \mid pos)$, $P(w \mid neg)$
5. Get $\lambda(w)$
6. Compute $logprior = \log(P(pos) / P(neg))$

Testing NaïveBayes
Predict using Naïve Bayes

- 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

Testing Naïve Bayes

- X_{val} Y_{val} λ $logprior$

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

$$pred = score > 0$$

$$\begin{bmatrix} 0.5 \\ -1 \\ 1.3 \\ \vdots \\ score_m \end{bmatrix} > 0 = \begin{bmatrix} 0.5 > 0 \\ -1 > 0 \\ 1.3 > 0 \\ \vdots \\ score_m > 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ pred_m \end{bmatrix}$$

Testing Naïve Bayes

- X_{val} Y_{val} λ $logprior$

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

$$pred = score > 0$$

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

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

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

Summary

- X_{val} Y_{val} $\longrightarrow$ Performance on unseen data
- Predict using λ and $logprior$ for each new tweet
- Accuracy $\longrightarrow$ $$\frac{1}{m}\sum_{i=1}^{m}(pred_i == Y_{val_i})$$
- What about words that do not appear in λ(w)?

Applications of Naïve Bayes

- Sentiment analysis

$$P(pos|tweet) \approx P(pos)P(tweet|pos)$$

$$P(neg|tweet) \approx P(neg)P(tweet|neg)$$

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

Author identification:

$P($  $|\text{book})$

divided by

$P($  $|\text{book})$

Spam filtering:

$$\frac{P(\text{spam}|\text{email})}{P(\text{nonspam}|\text{email})}$$

Applications of Naïve Bayes

Information retrieval:

$$P(\mathrm{document_k}|\mathrm{query}) \propto \prod_{i=0}^{|query|} P(\mathrm{query_i}|\mathrm{document_k})$$

Retrieve document if $P(\mathrm{document_k}|\mathrm{query}) > \mathrm{threshold}$

Word disambiguation:

$$\frac{P(\mathrm{river}|\mathrm{text})}{P(\mathrm{money}|\mathrm{text})}$$

Bank:  

Naïve Bayes Assumptions

- Independence
- Relative frequency in corpus
- Independence

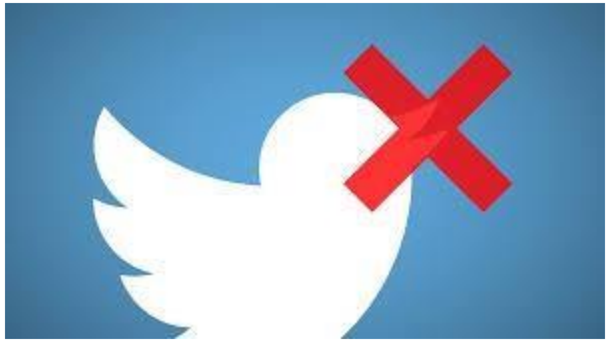
"It is sunny and hot in the Sahara desert."

"It's always cold and snowy in____."

spring?? summer? fall?? winter??

Naïve Bayes Assumptions

- Relative frequencies in corpus



- Relative frequency of training classes affect the model and can be not representative of the real world distribution

Error Analysis

- Outline
  - Removing punctuation and stop words
  - Word order
  - Adversarial attacks

- Processing as a Source of Errors: Punctuation

Tweet: My beloved grandmother~~:~~(

processed_tweet: [belov, grandmoth]

Processing as a Source of Errors

- Removing Words

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

processed_tweet: [good, attitude, close, nice]

- Word Order

Tweet: I am happy because I do not go.

Tweet: I am not happy because I did go.

Adversarial attacks

FPT Education
FPT UNIVERSITY

Sarcasm, Irony and Euphemisms

Tweet: This is a ridiculously powerful movie. The plot was gripping and I cried right through until the ending!

processed_tweet: [ridicul, power, movi, plot, grip, cry, end]

- Adversarial attacks (Easily detected by humans but algorithms are usually terrible at it)
  - Sarcasm, Irony, Euphemisms, etc
  - Example: This is a ridiculously powerful movie. The plot was gripping and I cried right through until the ending