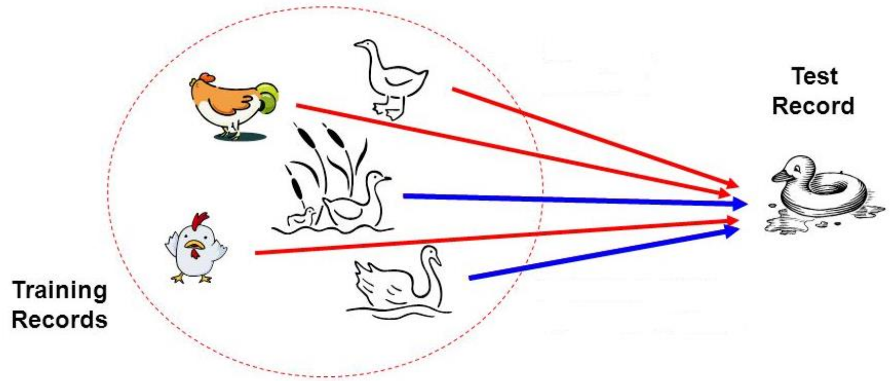
Bayesian Classifier

Naïve Bayes Classifier

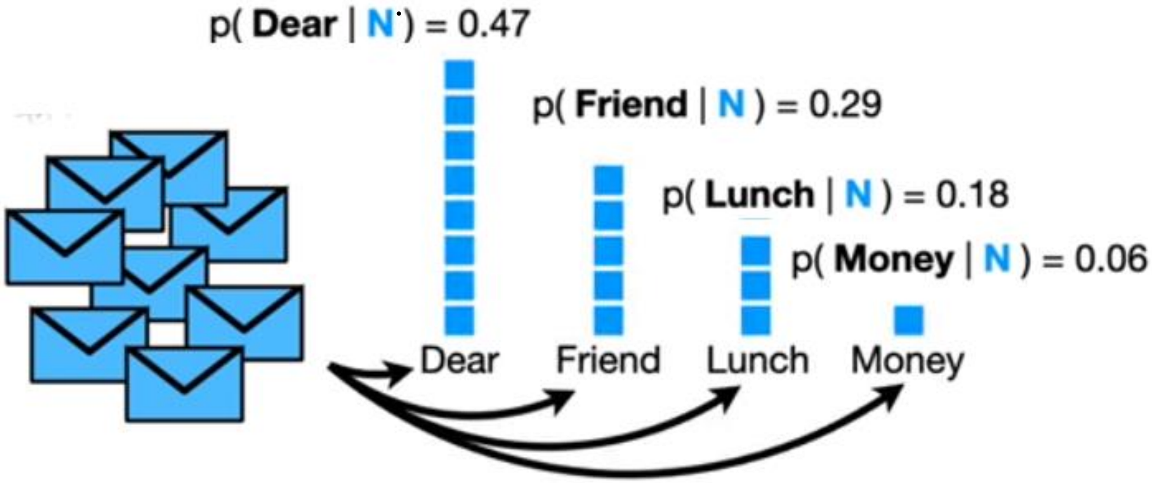
- Given a dataset X = $\{x_1, x_2, \ldots\ldots, x_m\}$ a set of classes C = $\{c_1, c_2, \ldots\ldots, c_k\}$, the classification problem is to define a mapping $f: X \rightarrow C$, Where each x_i is assigned to one class.
- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems
- Probabilistic Approach to Learning. Instead of learning F: X → C, learn P(C|X).
- can design algorithms that learn functions with uncertain outcomes



Applications

- Face Recognition
- Weather Prediction
- Medical Diagnosis
- News Classification
- …

Example: Normal / Spam mail Classification

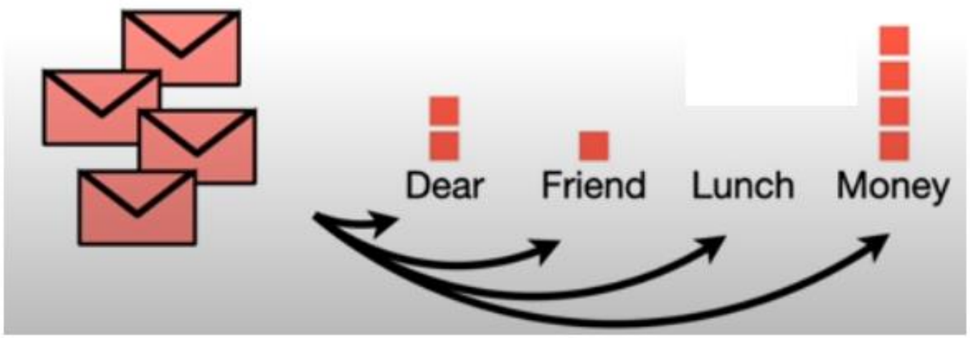


$$p(\textbf{Dear} \mid \text{Normal}) = \frac{8}{17} = 0.47$$

$$p(\textbf{Friend} \mid \text{Normal}) = \frac{5}{17} = 0.29$$

$$p(\textbf{Lunch} \mid \text{Normal}) = \frac{3}{17} = 0.18$$

$p(\textbf{Money} \mid N) = 0.06$



$p(\textbf{Dear} \mid S) = 0.29$

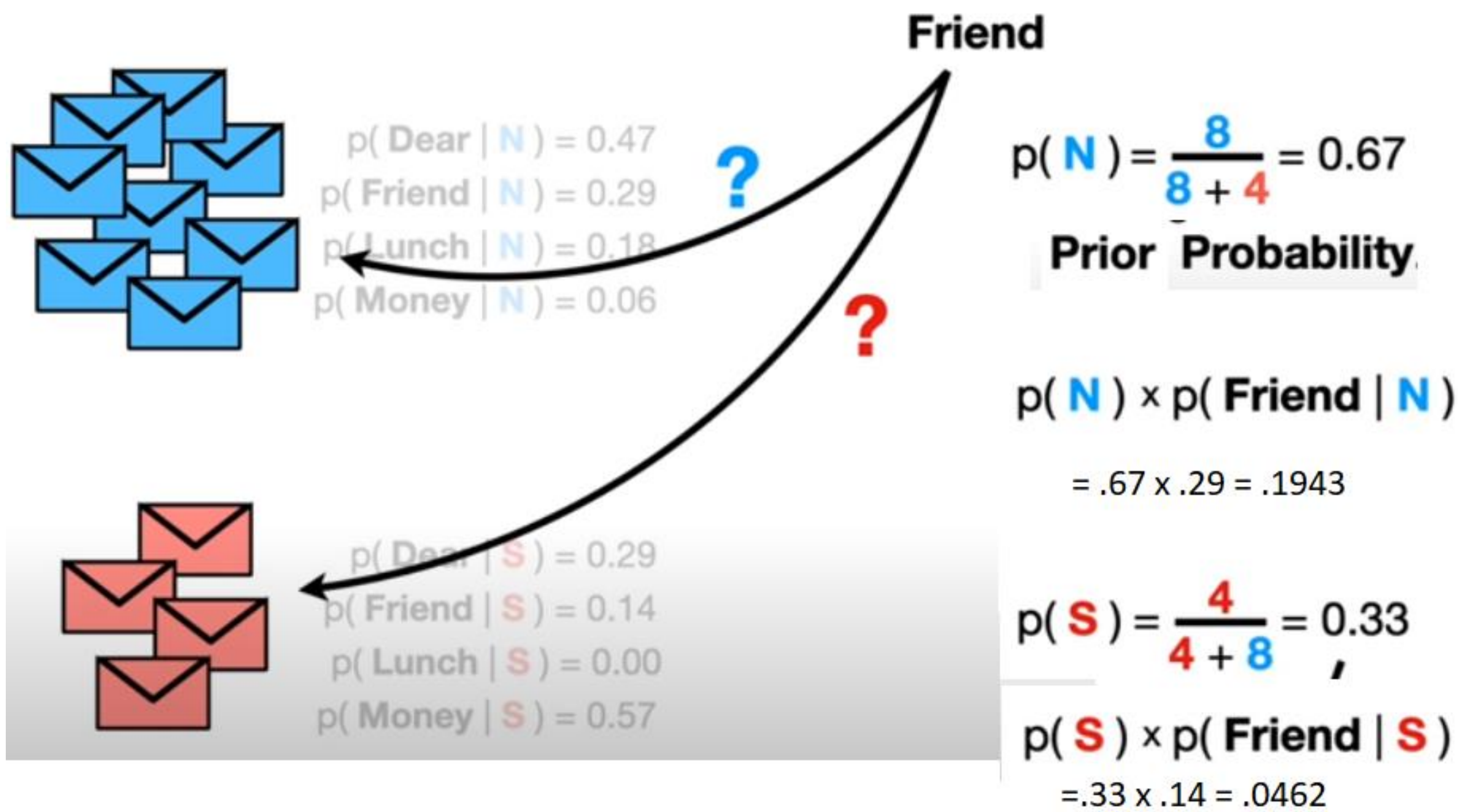
$p(\textbf{Friend} \mid S) = 0.14$

$p(\textbf{Lunch} \mid S) = 0.00$

$p(\textbf{Money} \mid S) = 0.57$

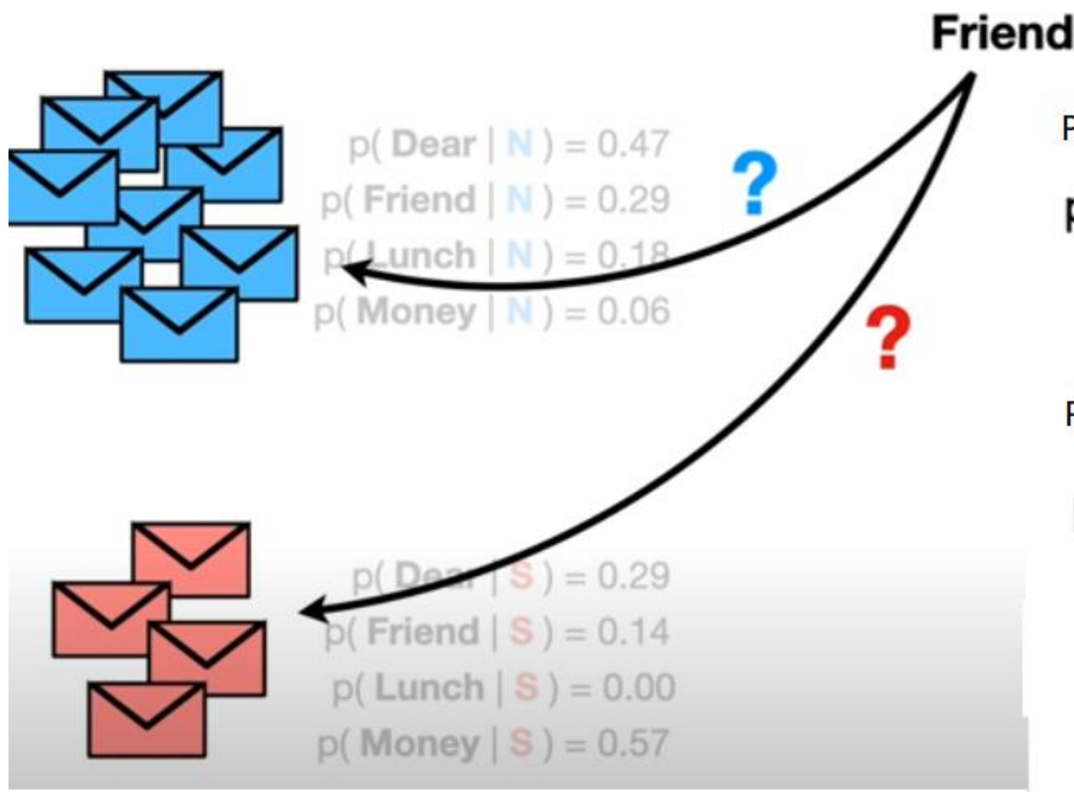
Slides taken from Josh Starmer, statquest

Example: Normal / Spam mail Classification



Slides taken from Josh Starmer, statquest

Example: Normal / Spam mail Classification



P(N|Friend)

p(**N**) x p(**Friend** | **N**)

= .67 x .29 = .1943

P(S|Friend)

p(**S**) x p(**Friend** | **S**)

=.33 x .14 = .0462

Slides taken from Josh Starmer, statquest

Naïve Bayesian Classifier

- Suppose, c is a class variable and X = $\{X_1, X_2, \ldots\ldots, X_n\}$ is a set of attributes, with instance of *c*.
- Naïve Bayesian classifier calculate this posterior probability using Bayes' theorem

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- The probability *P(c|X)* (also called class conditional probability) is therefore proportional to *P(X|c)*$\cdot\, P(c)$.
- Thus, *P(c|X)* can be taken as a measure of *c* given that *X*.

$$P(c/X) \approx P(X|c) \cdot P(c)$$

Weather data set

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

https://www.saedsayad.com/naive_bayesian.htm

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

Likelihood Table

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

$$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$$

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(Yes) = 9/14 = 0.64$$

$$P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$$

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Prediction on test data

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057$$

Naïve Bayesian Classifier Algorithm

Input: Given a set of k mutually exclusive and exhaustive classes $C = \{c_1, c_2, \ldots\ldots, c_k\}$, which have prior probabilities $P(C_1)$, $P(C_2)$,….. $P(C_k)$.

There are n-attribute set $A = \{A_1, A_2, \ldots\ldots, A_n\}$, which for a given instance have values $A_1 = a_1$, $A_2 = a_2$,….., $A_n = a_n$

Step: For each $c_i \in C$, calculate the class condition probabilities, $i = 1,2,\ldots..,k$

$$p_i = P(C_i) \times \prod_{j=1}^{n} P(A_j = a_j | C_i)$$

$$p_x = \max\{p_1, p_2, \ldots\ldots, p_k\}$$

Output: C_x is the classification

Naïve Bayesian Classifier Pros and Cons

Pros

- simple and easy to implement
- doesn’t require as much training data
- handles both continuous and discrete data
- fast and can be used to make real-time predictions
- not sensitive to irrelevant features

Cons

- assumption of independent predictors
- zero frequency problem

Example

No.	Swim	Fly	Crawl	Class Label
1	Fast	No	No	Fish
2	Fast	No	Yes	Animal
3	Slow	No	No	Animal
4	Fast	No	No	Animal
5	No	Short	No	Bird
6	No	Short	No	Bird
7	No	Rarely	No	Animal
8	Slow	No	Yes	Animal
9	Slow	No	No	Fish
10	Slow	No	Yes	Fish
11	No	Long	No	Bird
12	Fast	No	No	Bird
13	Slow	Rarely	No	?

Approach to overcome zero frequency problem

Classifying a new instance

- Consider the instance: (Sunny, Cool, High, Strong)
  - We can estimate each term using the data, for example:
    - Pr(Yes) = 9/14
    - Pr(No) = 5/14
    - Pr(Outlook = Sunny | Yes) = 2/9
    - Pr(Outlook = Sunny | No) = 3/5
    - We end up with $\frac{9}{14} \cdot \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} = .0053$ for Yes and $\frac{5}{14} \cdot \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} = .0206$ for No.
  - We thus predict that No is the output

If the posterior probability for one of the attributes is zero, then the overall class-conditional probability for the class vanishes.

This problem can be addressed by using the M-estimate approach.

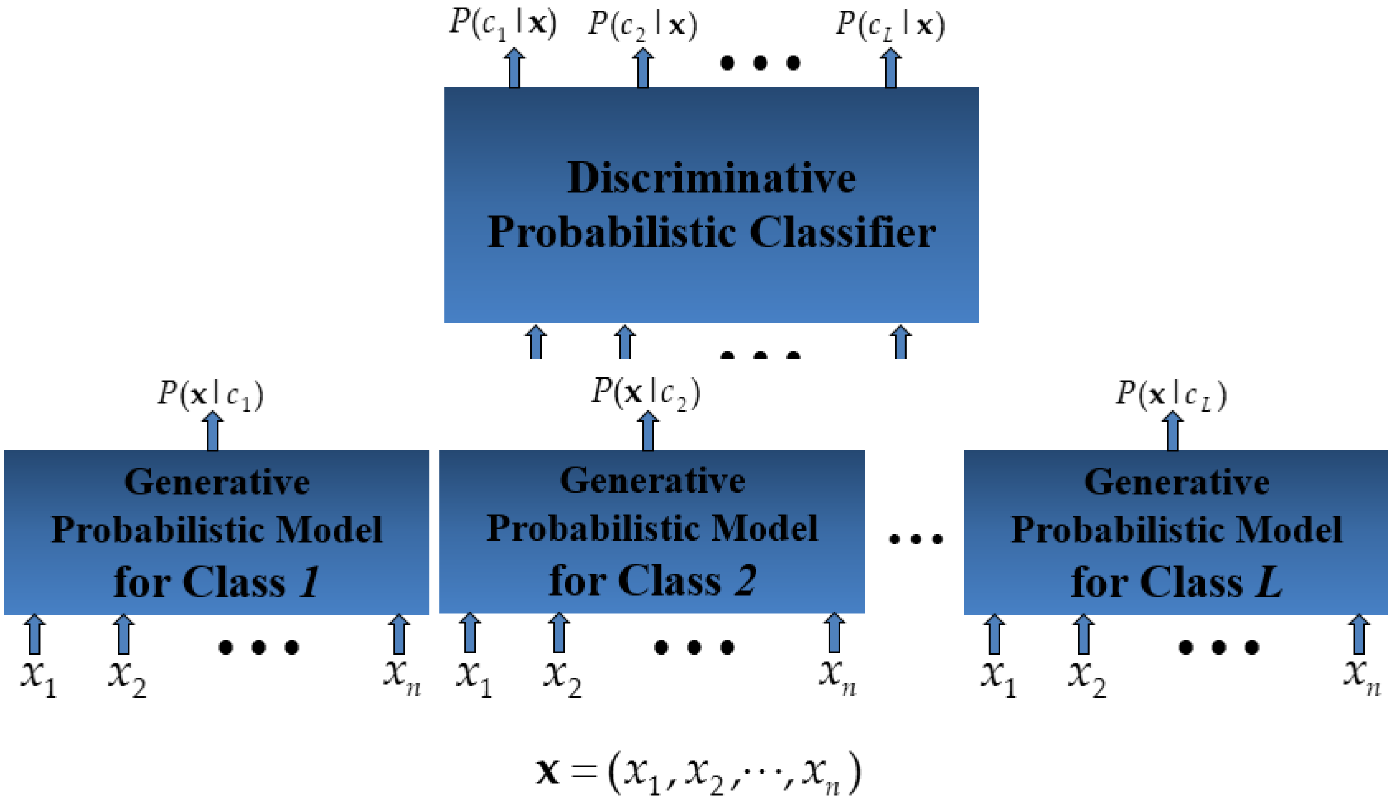
- This happened due to insufficient training data
- This problem can be avoided by using the **m-estimate**

M-Estimate Approach

- Note that we estimated conditional probabilities $\Pr(A \mid B)$ by $\frac{n_c}{n}$ where n_c is the number of times $A \wedge B$ happened and n is the number of times B happened in the training data
- This can cause trouble if $n_c = 0$
- fix the following numbers p and m
  - A nonzero prior estimate p for $\Pr(A \mid B)$, and
  - A number m that says how confident we are of our prior estimate p, as measured in number of samples
- Then instead of using $\frac{n_c}{n}$ for the estimate, use $\frac{n_c+mp}{n+m}$

- p: prior estimate of the probability
- In the absence of other information, assume a uniform prior:
  - P= 1 /k
  - where k is the number of values that the attribute x can take.
- m: equivalent sample size (constant)

Generative Model



Generative classifier learn joint probability p(x,y) and use bayes'rule for computing P(y|x)

Discriminative models directly learn p(y|x) from data

Text Classification using Naïve Bayes

Text Classification Problem

Natural Language Processing (NLP) task
Applications include

- **Social media monitoring:**
- **Customer feedback:**
- **Market research:**

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- *Output*: a predicted class $c \in C$

Bag of words representation

$\gamma($

I love this movie! It's sweet,
but with satirical humor. The
dialogue is great and the
adventure scenes are fun… It
manages to be whimsical and
romantic while laughing at the
conventions of the fairy tale
genre. I would recommend it to
just about anyone. I've seen
it several times, and I'm
always happy to see it again
whenever I have a friend who
hasn't seen it yet.

$)=c$

I **love** this movie! It's **sweet**,
but with **satirical** humor. The
dialogue is **great** and the
adventure scenes are **fun…** It
manages to be **whimsical** and
romantic while **laughing** at the
conventions of the fairy tale
genre. I would **recommend** it to
just about anyone. I've seen
it **several** times, and I'm
always **happy** to see it **again**
whenever I have a friend who
hasn't seen it yet.

x **love** xxxxxxxxxxxxxxxx **sweet**
xxxxxxx **satirical** xxxxxxxxxx
xxxxxxxxxxx **great** xxxxxxx
xxxxxxxxxxxxxxxxxxx **fun** xxxx
xxxxxxxxxxxxx **whimsical** xxxx
romantic xxxx **laughing**
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx **recommend** xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx **several** xxxxxxxxxxxxxxxxx
xxxxx **happy** xxxxxxxxx **again**
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$)=c$

A bag-of-words model, is a way of extracting features from text to describe vocabulary of known words

To reduce vocabulary size, apply text cleaning techniques

Naïve Bayes Classifier

- For a document *d* and a class *c*

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

$$c_{MAP} = \operatorname*{argmax}_{c \in C} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \operatorname*{argmax}_{c \in C} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname*{argmax}_{c \in C} P(d \mid c)P(c)$$

Dropping the denominator

$$= \operatorname*{argmax}_{c \in C} P(x_1, x_2, \ldots, x_n \mid c)P(c)$$

Document d represented as features x1..xn

Assumptions

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i|c_j)$ are independent given the class *c*.

$$P(x_1,\ldots,x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c_j) \prod_{x \in X} P(x \mid c)$$

Naïve Bayes Learning

- From training corpus, extract *Vocabulary*
- Calculate $P(c_j)$ terms
  - For each c_j in C do

    $docs_j \leftarrow$ all docs with class $= c_j$

    $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$
- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in *Vocabulary*

    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

    $$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

$P(c|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14$
≈ 0.0003

$P(j|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9$
≈ 0.0001

Summary

- Naive Bayes is a probabilistic supervised learning algorithm
- Based on the independence assumption . Called “Naïve” because of this assumption
- Uses prior knowledge with observed data
- Training and testing is very easy and fast
- Generative classifier model
- Uses Bag of Words representation
- probability of the class of document is computed using Bayes theorem
- Select a class with highest posterior probability

Reference

Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014