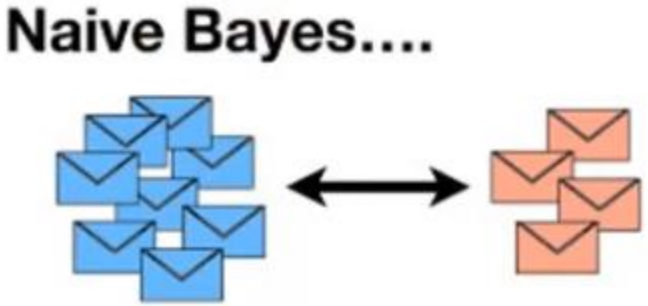
Naïve Bayes

Melika Zare

Naïve Bayes

- In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers“ based on applying Bayes' theorem with strong (naïve) independence assumptions between the features.
- They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve higher accuracy levels.



Example

$\vec{X}$ = email

$\vec{X} = \{x_1, x_2, \dots, xn\}$= all words in emails

(w_1, w_2, …, w_{1000})

P(c_k | $\vec{x}$) = ?

$$P(c_k \mid \vec{x}) = \frac{P(c_k)P(\vec{x}|c_k)}{P(\vec{x})} \qquad \text{Bayesian Rule}$$

Chain Rule

$P(C_k, x_1, \ldots, x_n) = P(x_1, \ldots, x_n, C_k)$

$= P(x_1 | x_2, \ldots, x_n, C_k)\, P(x_2, \ldots, x_n, C_k)$

$= P(x_1 | x_2, \ldots, x_n, C_k)\, P(x_2 | x_3, \ldots, x_n, C_k)\, P(x_3, \ldots, x_n, C_k)$

$= \ldots$

$= P(x_1 | x_2, \ldots, x_n, C_k)\, P(x_2 | x_3, \ldots, x_n, C_k) \ldots P(x_{n-1} | C_k)\, P(x_n | C_k)\, P(C_k)$

Naïve Conditional Independence

$P(x_i \mid x_{i+1}, \ldots, x_n, C_k) = P(x_i \mid C_k)$

$P(x_1 \mid x_2, \ldots, x_n, C_k)\, P(x_2 \mid x_3, \ldots, x_n, C_k) \ldots P(x_{n-1} \mid C_k) P(x_n \mid C_k)$

Conditional Probability

$$P(\vec{x} \mid c_k) = P(x_1 \mid C_k)\, P(x_2 \mid C_k) \ldots P(x_n \mid C_k) = \prod_{i=1}^{n} P(xi \mid C_k)$$

$$P(c_k \mid \vec{x}) = \frac{P(c_k) P(\vec{x} \mid c_k)}{P(\vec{x})} = \frac{P(C_k) \prod_{i=1}^{n} P(xi \mid C_k)}{P(\vec{x})}$$

Binary Classifier

$$P(S \mid \vec{x}) = \frac{P(S)\prod_{i=1}^{n} P(xi \mid S)}{P(\vec{x})}$$

$$P(N \mid \vec{x}) = \frac{P(N)\prod_{i=1}^{n} P(xi \mid N)}{P(\vec{x})}$$

$$P(S \mid \vec{x}) \cong P(S)\prod_{i=1}^{n} P(xi \mid S)$$

$$P(N \mid \vec{x}) \cong P(N)\prod_{i=1}^{n} P(xi \mid N)$$

Multi-class classifier

$$\hat{y} = \arg\max P(c_k) \prod_{i=1} P\left(x_i | c_k\right)$$

Example

Num	Words in Email	Type
1	Click on this link	Spam
2	Share your review	Not spam
3	Share your account number	Spam
4	Send us your review	Not spam
5	Send us your password	Spam

1) Remove Stopwords & commonly used words.
2) Create dictionary

Train Phase

Word	P(word | spam)	P(word | Not spam)
Click	1	0
Link	1	0
Share	½	½
Review	0	1
Account	1	0
Number	1	0
Send	½	½
Password	1	0

Words in Email	Type
Click on this link	Spam
Share your review	Not spam
Share your account number	Spam
Send us your review	Not spam
Send us your password	Spam

P(send | spam) = ½
P(send| Not spam) = ½

$$P(\vec{X} = \text{spam}) = \frac{3}{5}$$

$$P(\vec{X} = \text{Not spam}) = \frac{2}{5}$$

Test Phase

Word	P(word | spam)	P(word | Not spam)
Click	1	0
Link	1	0
Share	½	½
Review	0	1
Account	1	0
Number	1	0
Send	½	½
Password	1	0

$\vec{X}_{new}$ = {send your review to me}

$\vec{X}_{new}$ = {send, review}

$$\hat{y} = \arg\max P(c_k) \prod_{i=1} P(x_i | c_k)$$

$$\begin{cases} spam: \frac{3}{5} * \frac{1}{2} * 0 = 0 \\ Not\ spam: \frac{2}{5} * \frac{1}{2} * 1 = 0.2 \end{cases}$$

argmax

$\hat{y}$ = Not Spam

HAM

Total emails: 72

SPAM

Total emails : 28

Prob. Ham
=72/100
= 0.72

Prob. Spam
=28/100
= 0.28

HAM

Total Words
717

Word	Count	Prob of Word if mail is Ham
Friend	86	0.238227
Rich	41	0.113573
Money	79	0.218837
Beach	80	0.221607
Office	75	0.207756

SPAM

Total Words
741

Word	Count	Prob of Word if mail is Spam
Friend	63	0.184751
Rich	36	0.105572
Money	97	0.284457
Beach	53	0.155425
Office	92	0.269795

$\vec{X}_{new}$ = {you have rich friend}= {rich, friend}

$$\begin{cases} spam: 0.28 * 0.105572 * 0.18475 = 0.0054612 \\ Not\ spam: 0.72 * 0.11357 * 0.23822 = 0.0194804 \end{cases}$$

- log()

$$\begin{cases} spam: 0.54 + (3.144 + 2.1363) = 5.8303 \\ Not\ spam: 0.14 + (3.238 + 2.996) = 6.3476 \end{cases}$$

HAM

Total emails: 72

SPAM

Total emails : 28

Prob. Ham
=72/100
= 0.72

Prob. Spam
=28/100
= 0.28

HAM

Total Words
717

Word	Count	Prob of Word if mail is Ham
Friend	86	0.238227
Rich	41	0.113573
Money	79	0.218837
Beach	80	0.221607
Office	75	0.207756

SPAM

Total Words
741

Word	Count	Prob of Word if mail is Spam
Friend	63	0.184751
Rich	36	0.105572
Money	97	0.284457
Beach	53	0.155425
Office	92	0.269795

$$\xrightarrow{-\log()} \begin{cases} spam: 0.54 + (3.144 + 2.1363) = 5.8303 \\ Not\ spam: 0.14 + (3.238 + 2.996) = 6.3476 \end{cases}$$

$$P(class) * P(x_1|class) * \ldots * P(x_n|class) \Rightarrow$$

$$\log(P(class)) + \log(P(x_1|class)) + \ldots + \log(P(x_n|class))$$

$\vec{\mathrm{X}}_{new}$ = {send your review to me}

$\vec{\mathrm{X}}_{new}$ = {send, review}

$$\hat{y} = \arg\max P(c_k) \prod_{i=1} P\left(x_i | c_k\right)$$

$$\begin{cases} spam: \frac{3}{5} * \frac{1}{2} * 0 = 0 \\ Not\ spam: \frac{2}{5} * \frac{1}{2} * 1 = 0.2 \end{cases}$$

- log()

$$\begin{cases} spam: 0.73 + (1 + Undefiend) \\ Not\ spam: 1.32 + (1 + 0) \end{cases}$$

Laplace Smoothing

Thank you

Train:

$$P(F_{i,j} = f \mid class = c) = \frac{\#\text{ of times } F_{i,j} = f \text{ when } class = c}{\text{Total number of training examples where } class = c}$$

$$P(class = c) = \frac{\#\text{ of training examples where } class = c}{\#\text{ of training examples}}$$

Test:

$$\log(P(class)) + \log(P(f_{1,1} \mid class)) + \log(P(f_{1,2} \mid class)) + \ldots + \log(P(f_{28,28} \mid class))$$

class	posterior probability
0	.3141
1	.432
2	0
3	0
4	.4
5	.004
6	.1
7	.2
8	.7
9	.5