Page No.
Date :

Naive Bayes :-

→ Baye's theorem :-

A, B are two events – their probability $P(A)$ $P(B)$

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

↓
posterior

$P(B|A)$ – conditional prob B given A

$P(A)$ = prior prob A

$P(B)$ = prob of B

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

A	B	C
↓	↓	↓
$P(A)$	$P(B)$	$P(C)$
↓	↓	↓
$P(E/A)$	$P(E/B)$	$P(E/C)$ – common event.

$$P = \frac{P(A) \cdot P(E/A)}{P(A) \cdot P(E/A) + P(B)\, P(E/B) + P(C)\, P(E/C)}$$

Baye's theorem for classification

spam

not spam

$$P(y|x) = \frac{P(x/y)\, P(y)}{P(x)}$$

$$P(y=1/x) = \frac{P(x/y=1)\, P(y=1)}{P(x)}$$

$$P(y=1/x) = \frac{P(x/y=1)\, P(y=1)}{P(x)}$$

$$P(y=0|x) = \frac{P(x/y=0)\, P(y=0)}{P(x)}$$

Final prediction = $\arg\max P(y_i|x)$

Date :

$P(y=c \mid x) \;\alpha\;$ likelihood . prior

ex does patient has disease or not ?

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

let's say a patient takes a lab test & result comes back +ve. The test returns a correct +ve 98% time & correct -ve 97% of time. 0.008% of entire popn has this disease.

$$P(+ \mid \text{disease}) = 0.98$$

$$P(- \mid \text{disease}) = 1 - 0.98 = 0.02$$

$$P(- \mid \neg\,\text{disease}) = 0.97$$

$$P(+ \mid \neg\,\text{disease}) = 1 - 0.97 = 0.03$$

$$P(\text{disease} \mid +) = \frac{P(+ve \mid \text{disease}) \cdot P(\text{disease})}{P(+ve)}$$

$$= \frac{0.98 \times 0.008}{0.98 \times 0.008 + 0.03 \times 0.992}$$

$$P(+ve) = P(+ve \mid \text{disease}) \cdot P(\text{disease}) + P(+ \mid \neg\,\text{disease}) \cdot P(\neg\,\text{disease})$$

Page No.
Date :

Naive Bayes Classifier :-

for text classification

sms — spam
— not spam

email — inbox
— promotional
— social

movie review — +ve
— −ve
— neutral

discrete feature :- presence or absence of words.

$$P(y=1|X) = \frac{P(X|y=1)\,P(y=1)}{P(X)}$$

$$P(y=0|X) = \frac{P(X|y=0)\,P(y=0)}{P(X)}$$

pred : $\text{argmax}(P(y_i|X)) = \text{argmax}(P(X|y_i)P(y_i))$

$$P(y=1) = \frac{\text{count all +ve reviews}}{\text{total reviews}}$$

$$P(y=1) = \frac{\sum_{i=1}^{m} 1\{y^i = 1\}}{m}$$

$P(X|y=1) =$

$X = \langle x_1, x_2\, x_3 \ldots x_{|V|} \rangle$

$P(x_1, x_2\, x_3 \ldots x_{|V|} | y=1) =$

$P(x_1|y=1)\,P(x_2|y=1, x_1)\,P(x_3|y=1, x_1 x_2)$
$\ldots P(x_{|V|}|y=1, x_1 x_2 \ldots x_n)$

x_1 … good
x_2 … awesome

P(awesome | +ve, good)
P(like | +ve, good, awesome)

After naive bayes assumption

$P(x|y=1) = P(x_1|y=1)\, P(x_2|y=1) \ldots P(x_{|v|}|y=1)$

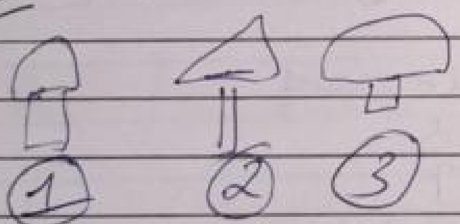
$= \prod P(x_i|y=1)$

$P(y=1|x) = \dfrac{\prod P(x_i|y=1) \cdot P(y=1)}{P(x)}$

$P(x) = P(x|y=0)\, P(y=0) + P(X|y=1)\, P(y=1)$

$P(y=0|x) = \dfrac{\prod (x_i|y=0)\ P(y=0)}{P(X)}$

ex Mushroom classification Example



$X =$ feature of mushroom $\begin{bmatrix} x_1 & x_2 & x_3 \\ \vdots & \vdots & \vdots \end{bmatrix}$ $\quad y =$ type of mushroom $\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$

$P(y=1|X) \longrightarrow \dfrac{P(X|y=1)\, P(y=1)}{P(x|y=1) + P(x|y=2) + P(x|y=3)}$

$P(y=2|X)$

$P(y=3|X)$ $\quad (x,y) \quad x = \langle x_1, x_2, x_3 \rangle$

naive bayes assumption:- independent features.

$P(X|y=1) = \prod P(x_1|y=1)\, P(x_2|y=1)\, P(x_3|y=1)$

$P(y=1|x) \propto \prod_{i=1}^{m} P(x_i|y=1) \cdot P(y=1)$

Date :

let's says $y \in C$

conditional probablity

$$P(y=c|x) \;\alpha\; \underbrace{\prod_{i=1}^{n} \overbrace{P(x_i|y=c)}^{\text{conditional probablity}}\; \underbrace{P(y=c)}_{\text{Prior}}}_{\text{likelihood}}$$

$$= \prod_{i=1}^{n}$$

# posterior prob = prior prob x likelihood
likelihood = likelihood x cond^n prob

```
def prior_prob(y_train, label)
    total_examples = y_train.shape[0]        → total no feature
    class_examples = np.sum(y_train = label)  ↳ total no. of feature with label
```

1 2 3 → class feature
1 –
2 –
3 –
4 –

type of mushroom

```
def cond_prob(x_train, y_train, feature_col, feature_val, label):
    x_filtered = x_train[y_train == label]
    numerator = np.sum(x_filtered          ↳ only get featur
        [:, feature_col == feature_val])      this label
    # all rows     particular feature occruence
```

get green mushroom in class 2.

$= \dfrac{\text{green mushroom in class 2}}{\text{total mushrooms in class 2.}}$

```
    denominator = np.sum(y_train == label)
    return numerator / float(denominator)
```

```
def predict(x_train, y_train, xtest):

    classes = np.unique(y_train)  0,1,2,3...
    n_features = x_train.shape[1]
    post_prob = []  list of prob of all classes
    for label in classes:

        likelihood = 1.0
        for f in range(n_features):
        likelihood = 1.0
        cond = cond_prob(x_train, y_train, f,
                         x test[f], label)
        likelihood *= cond

        prior = prior_prob(y_train, label)
        post = likelihood * prior
        post_prob.append(post)

    pred = np.argmax(post_prob)
    return pred.
```

Naive Bayes for text classification

$\boxed{\text{Text review}}$ movie review y $\boxed{}$ {1, 2, 3 ... k} rating

x

1. Bag of words - order of words doesn't matter
2. conditional independence

$$P(y = c) = \frac{[\text{no. of example in class } c]}{\text{total examples}}$$

$$P(x_i \mid y_i = c) = \frac{[\text{count of } c]}{[\text{total examples}]}$$

how many times happy occurs in class +ve review

count of word x_i that occured in class c

Page No.

Date :

$$P(x_i / y_i = C) = \frac{[\text{count of } x_i, y_i = C]}{[\Sigma w \ \text{count}(w, y_i = C)]}$$

cond^n prob of word =

sum of counts of each words in vocab in class c

class +ve review

happy, awesome, beautiful, fun, great, good — count how many times these words occur

$P(y = +ve \mid x) = \Pi \, P(x_i / y = +ve) \, P(y = +ve)$

- good —
- awesome —
- liked —
- ....

training

"I was overjoyed"

Test

now P("overjoyed / y = "+ve") comes to be 0 in +ve class

↳ wrong prediction

Solution :- Laplace Correction

↳ add 1 to our numerator & denominator

$$P(x_i / y = C) = \frac{[\text{count}(x_i, y = C) + 1)}{\Sigma w \in V \, [\text{count}(w, y = C) + 1]}$$

this term can never be zero now

$$= \frac{\text{count}(x_i, y = C) + 1}{\Sigma w \in V \, (\text{count}(w, y = C) + |V|}$$

vocab size

Multivariate Bernoulli Naive Bayes

y=spam D_1 = "like I like to swim" [1 1 0 1 1]

y=code D_2 = "I like coding" [1 1 1 0 0]

$$\text{vocab} = [\underset{x_0}{I}, \underset{x_1}{like}, \underset{x_2}{coding}, \underset{x_3}{swim}, \underset{x_4}{to}]$$

(Bag of words model)

$$P(x \mid y=c) = \pi\, P(x_i \mid y=c)^{b} \cdot (1 - P(x_i \mid y=c))^{(1-b)}$$

cond^n prob of generating this sentence in class c

$$P(y \mid x) = \pi\, (P(x_i \mid y=c)\ P(y=c)$$

$$P(x_i \mid y=c) = \frac{df_{x_i, y=c} + 1}{\text{count of docs in class } c + 2}$$

$df_{x_i,y=c}$ = count of document having class c & they contain this feature x_i

Multinomial Event Naive Bayes

- Frequency of term $tf(t, d)$
- normalized term freq $= \frac{tf(t,d)}{n_d}$ ← raw freq

$$P(y \mid x) = \pi\, (P(x \mid y)\ P(y))$$

← no. of doc — total no. of doc in class (with word x)

$$\hat{P}(x_i \mid w) = \frac{\sum tf(x_i, d \in c) + |\alpha|}{\sum N_{doc} + |\alpha|\, V}$$

$|\alpha|$ — laplace hyper parameter

$\sum N_{doc}$ — total no. of doc in class c

Suitable for discrete feature values

Page No.
Date :

Multivariate Bernoulli Event

"offer for you"

$L(\theta) = \prod_{i=1}^{n} P(x_i | y = \text{spam})^{(1-?)} (1 - P(x_i | y = \text{spam}))\, P(\text{spam})$

Multinomial Event Model

"offer for you"

$\prod P(x_i | y)\, P(y)$

comes from multinomial distribution

$P(\text{offer} | y = \text{"spam"})$

Gaussian Naive Bayes

- suitable for continous variables - gaussian distribution

$$P(x_i | y = c) = \frac{1}{\sqrt{2\pi}\,\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$P(x_i | y = c)$ = normal dist^n (μ, σ)

cond^n prob