Probabilistic Models

In real world, there are a lot of scenerios where certainty of something is not confirmed.

In order to represent uncertain knowledge where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning

When there are ① unpredictable outcomes

② too large probabilities

③ Unknown errors / events

In order to tackle uncertainty, probabilistic reasoning is used.

Probabilistic models represent the relationships between events in an probabilistic way

Then inference uses the probability to take decisions.

Few egs of Probabilistic models are BBN, Naive Bayes

Naive Bayes Classifier

Baysian classifier works on probability
gives probabilistic prediction

(Unlike SVM & KNN which are based on distance, Baysian & Decision tree are based on probability)

Handles independant events only [Assumption of independant events]

Used to make bayesian belief networks

Very simple Model

Useful for spam filtering, sentiment analysis and medical diagnosis

Works on catagorical data (Unlike SVM)

Used for classification

Called as Naive classifier because

1. Very simple
2. Empirically useful
3. Scales very well

Assumption → all variables are independant

→ Dependant events → H & T of coin

if there is Head, it means Tails is not there definitely

→ Independant events → 2 coins tossed simultaneously

If one has head, we dont know what is the event of first coin

"Naive" because it considers events independant

In actual sense, even if the events are dependant Naive bayes assumes that the events are independant

Bayesian classifier principle →

"If it walks like a duck, quacks like a duck, then it probably is a duck"

In many applications, the relationship between the attributes set and class is non-deterministic

This means that a test cannot be classified to a class label with certainty

eg

age	income	Buy	
Young	High	Yes	1
Young	Low	No	
Young	High	No	2
Young	High	Yes	3
Old	High	Yes	

same tuple different outcomes

∴ No inference rules can be drawn.

It is like two points are coinciding with one another in an SVM

But, we can probabilistically say that given tuple young high, the probability

Yes : $\frac{2}{3}$ No : $\frac{1}{3}$

Bayes classifier is an approach for modelling such probabilistic relationships.

Example Training data →

id	age	income	student	crediting	buy Car
1	Youth	high	no	fair	no
2	y	h	no	excellent	no
3	m-a	h	no	f	y
4	S	medium	no	f	y
5	S	low	yes	f	y
6	S	L	yes	e	n
7	m-a	L	yes	e	y
8	y	m	no	f	n
9	y	L	y	f	y
10	S	m	y	f	y
11	y	m	y	e	y
12	ma	m	n	e	y
13	ma	h	y	f	y
14	S	m	n	e	n

A young student with fair credit score, has medium income, will he buy car?

Here income & age might be dependant but we assume them to be independant for naive bayes

* H — Hypothesis (eg buy or not buy)
* X — evidance (data type)
* prior probabilty $P(H)$ → knowledge is advanced. that is used

Example buy computer event has a probibility 0.7 is known in advanced by historical data

* Posterior probability $P(H|x)$

ie posterior probability of H conditioned on X

* Goal → Calculate $P(H|X)$

$$P(H|x) = P(H) \frac{P(x|H)}{P(x)}$$

Depending on values of income, age, etc will student buy the laptop?

* Posterior probability = Prior probability × likelyhood

* likelyhood $P(x|H)$

Since variables are independant

$$P(X|H) = \prod P(x_i|H)$$

* Marginal probability $P(x)$

It is the overall probability of observing feature vector x

eg $x = [age = y, income = m, \ldots]$

$P(x)$ is constant for all H, and since we only want to compare,

$$P(H|x) \propto P(H)\, P(x|H)$$

No need to calculate $P(x)$ & divide

We want to find out the probability that the student buys computer given age = youth, income = medium, ... etc

ie $P(H = yes \mid age = y, income = m, student = y, creditscore = f)$

$$P(H \mid X) = \frac{P(H)}{P(x)} \; P(X \mid H)$$

(From Bayes theorem)

Since we assume age, income, student etc to be independant variables, we can use the bayes theorem to split it as

$$P(X \mid H) = \prod_{i=1}^{n} P(X_i \mid H)$$

Here H is the output ie buy computer = True
X is the parameters in testing tuple
ie $X = [age = y, income = m, \ldots]$

$\therefore P(H \mid X) = P(H = yes) \times P(age = y \mid H = yes) \times P(income = m \mid H = yes) \ldots\ldots$

In order to predict if computer buy or not, we calculate $P(H = yes/X)$ & $P(H = No/X)$ and compare them

$P(H) \rightarrow$ Probability student buys computer

$P(H = yes) =$ yes: $9/14$
$P(H = No) =$ No: $5/14$

$P(X_1 \mid H) \rightarrow$ Probability that the age is youth when computer is bought

$P(age = youth \mid H = yes) = \frac{2}{9}$

$P(age = Youth \mid H = No) = \frac{3}{5}$

Here we considered age = youth because it is in our testing tuple)

$P(X_2 \mid H)$

$P(income = medium \mid H = yes) = \frac{4}{9}$

$P(income = medium \mid H = no) = \frac{2}{5}$

$P(X_3 \mid H)$

$P(student = yes \mid H = yes) = 6/9$

$P(student = yes \mid H = no) = 1/5$

$P(X_4 \mid H)$

$P(creditscore = fair \mid H = yes) = 6/9$

$P(creditscore = fair \mid H = no) = 2/5$

Using these probabilities,

$P(x \mid buy\ computer = yes)$ ie the probability of a customer being a youth student with medium income

$$P(X \mid H = yes) = P(X_1 \mid H = yes) \times P(X_2 \mid H = yes) \times \ldots\ldots$$

$$= \frac{2}{9} \times \frac{4}{9} \times \frac{6}{9} \times \frac{6}{9}$$

$$= 0.044$$

$$P(x \mid H = No) = \frac{3}{5} \times \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} = 0.019$$

$$P(H = yes \mid x) = P(x \mid H = yes) \frac{P(H = yes)}{P(x)}$$

$$= 0.044 \times \frac{9}{14} = \frac{0.028}{P(x)}$$

$$P(H = No \mid x) = P(X \mid H = No) \; P(H = No)$$

$$= 0.019 \times \frac{5}{14} = \frac{0.006}{P(x)}$$

$P(H = yes) > P(H = No)$

$\therefore$ The tuple can buy computer

$P(x)$ is same for all H (both buy & not buy, so we can ignore it without calculation

Q2

Color	Cloth	Brand	On sale
Blue	Jeans	Y	yes
Blue	Shirt	X	No
Blue	J	Y	Y
Black	J	Y	Y
Black	S	X	N
Black	S	X	Y
Black	S	Y	Y
Blue	J	Y	N
Black	J	X	Y
Blue	S	X	Y

Classify tuple Blue, jeans brand X would be on sale

→ $P(H = yes) = P(\text{Onsale}) = \frac{7}{10}$

$P(H = No) = \frac{3}{10}$

$P(\text{Color} = \text{Blue} \mid H = yes) = 3/7$

$P(\text{color} = \text{Blue} \mid H = No) = 2/3$

$P(\text{cloth} = \text{Jeans} \mid H = yes) = 4/7$

$P(\text{cloth} = \text{Jeans} \mid H = No) = 1/3$

$P(\text{brand} = X \mid H = yes) = 3/7$

$P(\text{brand} = X \mid H = No) = 2/3$

$P(H = yes \mid X, B, J) = P(H = yes) \times P(\text{Color} = \text{Blue} \mid H = yes)$
$\times \ldots\ldots$

$= \frac{7}{10} \times \frac{3}{7} \times \frac{4}{7} \times \frac{3}{7} = 0.073$

$P(H = No \mid XBJ) = P(H = No) \times P(\text{Color} = \text{blue} \mid H = No)$
$\times \ldots.$

$= \frac{3}{10} \times \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} = 0.044$

$P(H = yes) > P(H = No)$

Since probability of Onsale is higher, we can say it will be on sale

Multinomial Naive Bayes

Used for numeric data

n(free)	n(offer)	n(money)	email
3	0	1	Spam
0	1	0	Not spam
2	1	1	Spam
0	1	0	Not Spam
4	2	1	Spam
0	0	0	Not spam

find class where "free" : 2 times, "offer" : 1 time, "money" : 0 time } string occurance

→ $P(\text{Spam}) = \frac{3}{6} = 0.5$

$P(\text{not spam}) = \frac{3}{6} = 0.5$

Vocabulary size = 3 (3 features)

Count (free) & spam = 9
Count (offer) & spam = 3
Count (money) & spam = 3

Total words = 15

$P(\text{"free"} \mid \text{spam}) = \frac{9}{15}$

$P(\text{"offer"} \mid \text{spam}) = \frac{3}{15}$

$P(\text{"money"} \mid \text{spam}) = \frac{3}{15}$

$$P(\text{Spam} \mid \text{free}:2, \text{offer}:1, \text{money}:0) = P(\text{free} \mid \text{Spam})^2 \times P(\text{offer} \mid \text{spam})^1 \times P(\text{money} \mid \text{spam})^0 \times P(\text{free}:2, \text{offer}:1, \text{money}:0)$$

$$= \left(\frac{9}{15}\right)^2 \times \frac{3}{15} \times P(\text{free}:2, \text{offer}:1, \text{money}:0)$$

$$= 0.072 \times P(\text{free}:2, \text{offer}:1, \text{money}:0)$$

Similarly for non spam

$P(\text{"free"} \mid \text{non spam}) = 1/2 = 1/2$

$P(\text{"offer"} \mid \text{non spam}) = 2/2 = 1$

$P(\text{"money"} \mid \text{non Spam}) = 0/2 = 0$

$$P(\text{nonspam} \mid \text{free}:2, \text{offer}:1, \text{money}:0) = P(\text{free} \mid \text{non Spam})^2 \times P(\text{offer} \mid \text{non spam})^1 \times P(\text{money} \mid \text{non Spam})^0 \times P(\text{free}:2, \text{offer}:1, \text{money}:0)$$

$$= \left(\frac{1}{2}\right)^2 \times 1 \times P(\text{free}:2, \text{offer}:1, \text{money}:0)$$

$$= 0.25 \times P(\text{free}:2, \text{offer}:1, \text{money}:0)$$

$P(\text{Spam}) < P(\text{non Spam})$

$\therefore$ Not spam

General formulae for Multinomial

for class C_k, prior probability =

$$P(C_k) = \frac{N_k}{N}$$

N_k ← No of training examples in C_k

N ← total no of training samples

likelyhood

$$P(x_i \mid C_k) = \frac{n_i}{N_k}$$

n_i ← total (sum) of all feature counts in C_k

N_k ← No of training examples in C_k

$$P(C_k \mid x_1^{a_1}, x_2^{a_2} \dots) \propto P(C_k) \prod P(x_i \mid C_k)^{a_i}$$

$a_i \rightarrow$ count of feature in test tuple

Instead of raising probability floats to powers, we can simplify by taking log scale

$$\log P(C_k \mid x) \propto \log \left(P(C_k) \prod_{i=1} P(x_i \mid C_k)^{a_i} \right)$$

$$\propto \log(P(C_k)) + \sum_{i=1}^{n} a_i \log(P(x_i \mid C_k))$$

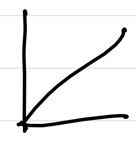
making it linear! No products needed.

We can now calculate the log probabilities instead during training time

And since we are only comparing, no need to take inverse

Note → In multinomial we assume only in one direction.

eg Too high occurences = high occurences

 & not  Possible

eg
0 – 5 occurances → spam
5 – 10 occurances → Non spam.
10 + occurances → spam
} Not Possible

Note → in some references

$$P(X \mid C_k) = \frac{(\Sigma n_i)!}{\prod_{i=1}^{V} n_i!} \cdot \prod_{i}^{V} P(x_i \mid C_k)^{n_i}$$

Here the $\frac{(\Sigma n_i)!}{\prod_{i=1}^{n} n_i!}$ represents multinomial coefficient, or no of ways the features in test tuple can be arranged.

Same for all classes C_k, hence neglected

matters only when the arrangements matter, not in Naive bayes.

eg No of possible arrangement of

"free" × 2
"offer" × 1
"money" × 2
} $\frac{5!}{2! \times 2!} = \frac{120}{4} = 30$

eg free free offer money
money free offer free
⋮

Zero Probability Smoothening

What if $(buy \mid Brand = z)$ asked and z not in dataset?

If a test data has a value never observed in dataset training phase, then $P = 0$ which may lead to incorrect classification

Entire product $P(x_1, x_2 \ldots \mid buy)$ will become 0

$$P(x_i \mid C_k) = \frac{Count(x_i, C_k)}{Count(C_k)}$$

$$Count(x_i, C_k) = 0 \therefore P = 0$$

To overcome this, smoothening is used

① Laplacian smoothening

$$P(x_i \mid C_k) = \frac{Count(x_i, C_k) + 1}{Count(C_k) + V}$$

V: total number of Unique features (Vocabulary size)

Probability of unseen feature is small, but still not zero

eg $P(\text{"money"} \mid spam) = \frac{3+1}{\underset{\substack{\uparrow \\ V=3}}{15+3}} = \frac{4}{18}$

② Additive smoothening

$$P(x_i \mid C_k) = \frac{Count(x_i, C_k) + \alpha}{Count(C_k) + V}$$

α - small constant

Useful when feature set is large

Gaussian Naive Bayes

for continnouse data we can use Gaussian Naive bayes which assumes that the features are in normal distribution

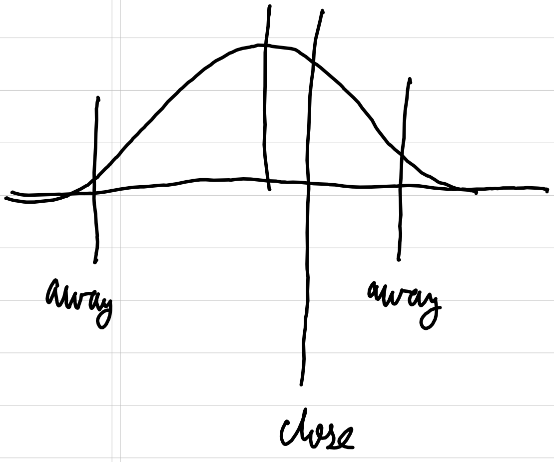
$$P(x_i = x \mid C_k) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x_i - \mu)^2}{2\sigma^2}}$$

σ & μ calculated from the data

$$P(C_k \mid x_1 = a_i, x_2 = a_2, \vdots) = P(C_k) \prod P(x_i = a_i)$$

continnouse variable

μ & σ for a class C_k calculated as training.



close to mean → high prob
away from mean → low prob

Two way can be accomodated

prob never 0, hence no need for regularization

Bernoulli Naive Bayes

Only 2 class data (binary) features are allowed

specialization of general case

eg income : high, low ✓
income : high, medium, low ✗

$$P(x \mid C_k) = \prod_{i=1}^{n} P(x_i = 1 \mid C_k)^{x_i} \left(1 - P(x_i = 1 \mid C)\right)^{1 - x_i}$$

$x_i \rightarrow$ 1 : Present
0 : Absent

Same as the general case

when present $P(x = 1 \mid C_k)$

when absent $1 - P(x = 1 \mid C_k) = P(x = 0 \mid C_k)$

Generative vs Descriminative Models

Generative	Descriminative
Learn joint probability $P(x \cap y)$	Learn conditional prob $P(x \mid y)$
Model how data is generated to make predictions	Find decision boundary between different classes
Estimate probability of both the input features x and output labels y	Directly map input x and output y without modelling how data is generated
Can generate new data	Can only classify or predict
eg N.B., GMM, HMM, BBN	eg D.T., SVM, N.N., logistic regression

N.B. models the distribution for every class, thus making it generative model

BBNs can make new synthetic samples from the conditional probability table

N.B. is generally used for classification only; but here is how it can be used for generation of new samples.

1. Sample class label based on $P(y)$

   eg $P(y = \text{spam}) = 0.6$, then pick spam 60% time

2. Sample each feature independantly given y

   $P(\text{Free} \mid \text{spam}) = 0.9$, then pick word free 90% of time

3. repeat for all features

Can be used for data augmentation

Advantages → 1. Simple

2. Fast

3. Low Power

4. Can handle large datasets

Disadvantages → 1. Assumption of independant predictors may not hold

2. Cannot capture complex relationships in the data. Very simple "naive" model

3. Doesnt work well for small datasets because probability wont be correct.

Applications → 1. Spam filtering
2. Sentiment analysis
3. Topic classification
4. Recommendation systems
5. Anamoly detection
6. Credit scoring

Output of training of naive bayes on a dataset is the calculation of prior probabilities & likelyhoods

Data Type	Model	Example
Binary	Bernoli N.B.	High / Low
Catagorical	N.B.	High / Medium / Low
Discrete	Multinomial N.B.	0, 1, 2, 3
Continouse	Gaussian N.B.	0.1, -10.3, 4.3