به نام او، به یاد او، برای او

Machine Learning

Naive Bayes Classifier

(NB)

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n|y)P(y)}{P(x_1, x_2, \dots, x_n)}$$

Dr. *Ali Valinejad*

valinejad.ir
valinejad@umz.ac.ir

Probabilistic Classification

Data Set:

$(\boldsymbol{x}^{(i)})^T$	$y^{(i)}$
$(\boldsymbol{x}^{(1)})^T$	$y^{(1)}$
$(\boldsymbol{x}^{(2)})^T$	$y^{(2)}$
$\vdots$	$\vdots$
$(\boldsymbol{x}^{(m)})^T$	$y^{(m)}$

$$\boldsymbol{x}^{(i)} = \left[x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}\right]^T$$

$$y^{(i)} \in \{c_1, c_2, \dots, c_K\}$$

Problem statement:

Given features $(x_1, x_2, \dots, x_n)$

Predict a label y= $c_1, c_2, \dots, c_K$

A good strategy is to predict:

$$\underset{y}{\operatorname{argmax}}\ P(y|x_1, x_2, \dots, x_n)$$

Car Theft Example

Data Set:

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Features:

Color, *Type*, *Origin*

(x_1: *Color*, x_2: *Type*, x_3: *Origin*)

(x_1=*Red*, x_2= *SUV*, x_3= *Domestic*)

Label:

Stolen (can be either Yes or No). (y= *Yes* or *No*)

Car Theft Example

Problem statement:

Given features $x_1, x_2, \ldots, x_n$

Predict a label y= *Yes*, *No*

if x_1=Red, x_2=SUV, x_3=Domestic, **predict a label for** *Stolen* (y = *Yes* or *No*)

$$\underset{y}{\text{argmax}}\ P(y|x_1=\text{Red}, x_2=\text{SUV}, x_3=\text{Domestic})$$

Probabilistic Classification

Establishing a probabilistic model for classification

- Discriminative model

$$P(y|X) \qquad y = c_1, c_2, \ldots, c_K,\ X = (x_1, x_2, \ldots, x_n)$$

- Generative model

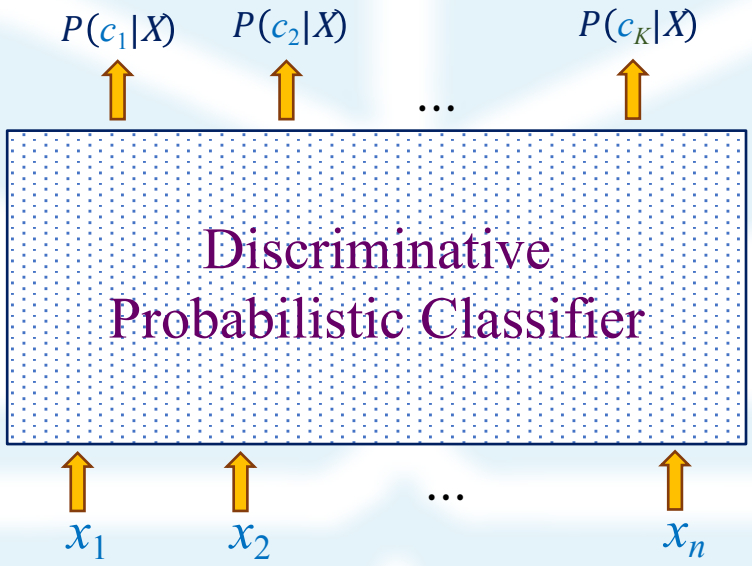
$$P(X|y) \qquad y = c_1, c_2, \ldots, c_K,\ X = (x_1, x_2, \ldots, x_n)$$

Probabilistic Classification

Establishing a probabilistic model for classification

- Discriminative model

$$P(y|X) \qquad y = c_1, c_2, \ldots, c_K,\ X = (x_1, x_2, \ldots, x_n)$$



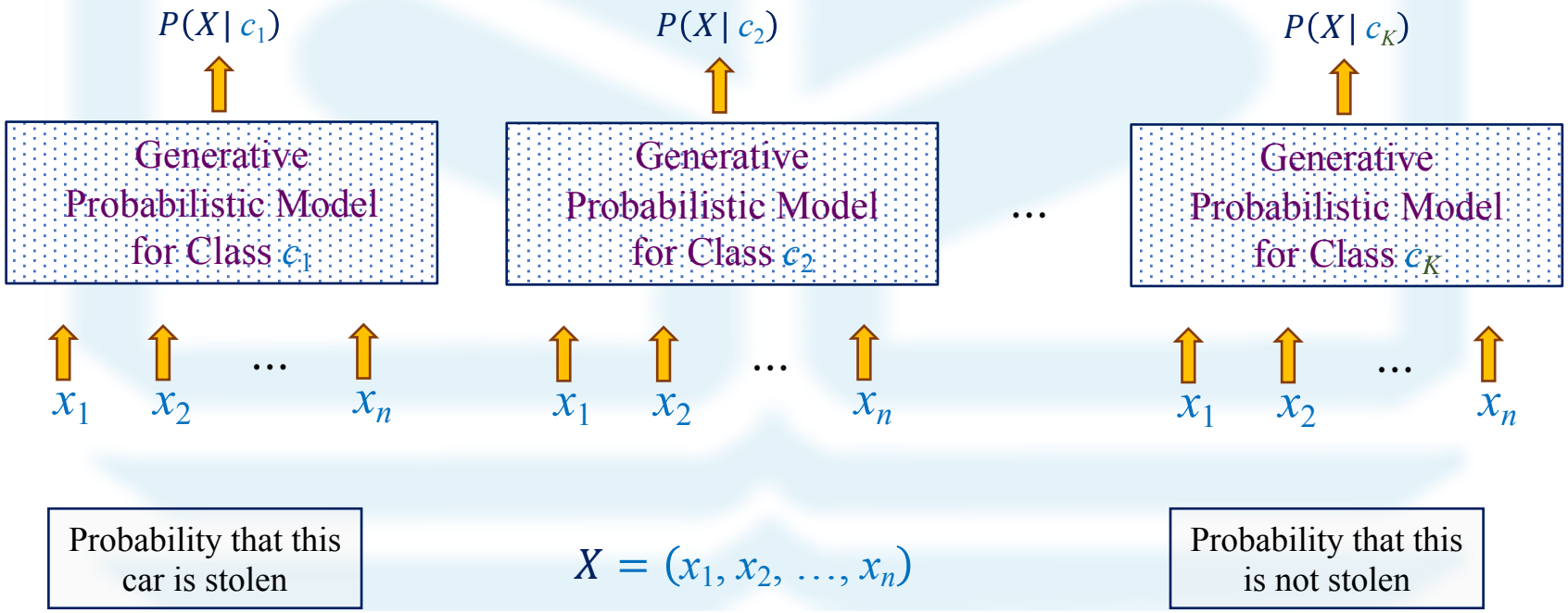
$$X = (x_1, x_2, \ldots, x_n)$$

Probabilistic Classification

Establishing a probabilistic model for classification

- Generative model

$P(X|y)$ $\quad y = c_1, c_2, \ldots, c_K,\ X = (x_1, x_2, \ldots, x_n)$



The Bayes Classifier

Establishing a probabilistic model for classification

- Generative model

$P(X|y)$ $\quad y = c_1, c_2, \ldots, c_K,\ X = (x_1, x_2, \ldots, x_n)$

Likelihood Prior

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

Normalization Constant

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|y)P(y)}{P(x_1, x_2, \ldots, x_n)}$$

Why did this help?

We might be able to specify how features are "generated" by the class label.

The Bayes Classifier

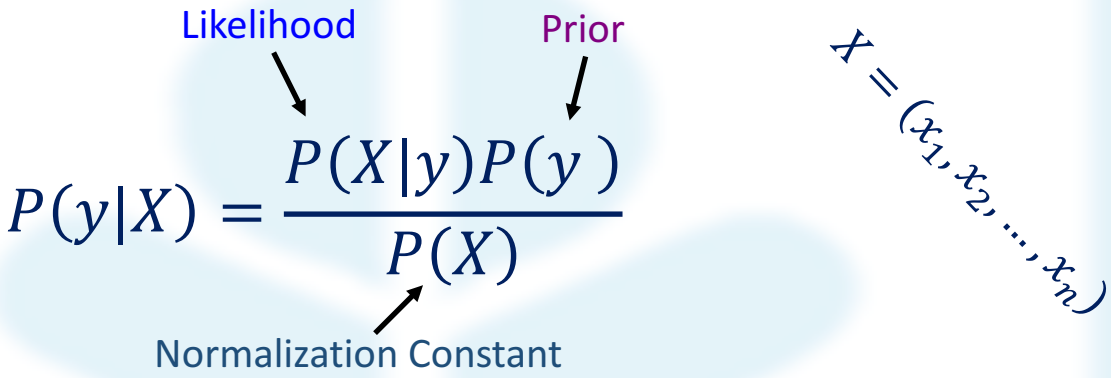
$$P(y = c|x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n|y = c)P(y = c)}{P(x_1, x_2, \dots, x_n)}$$
$$= \frac{P(x_1,x_2,\dots,x_n|y=c)P(y=c)}{P(x_1,x_2,\dots,x_n|y=Yes)P(y=Yes)+P(x_1,x_2,\dots,x_n|y=No)P(y=No)}$$

$$P(y = Yes|x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic})$$
$$= \frac{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = Yes)P(y = Yes)}{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic})}$$
$$= \frac{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = Yes)P(y = Yes)}{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = Yes)P(y = Yes) + P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = No)P(y = No)}$$

$$P(y = No|x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic})$$
$$= \frac{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = No)P(y = No)}{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic})}$$
$$= \frac{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = No)P(y = No)}{P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = No)P(y = No) + P(x_1\text{=Red}, x_2\text{=SUV}, x_3\text{=Domestic}|y = Yes)P(y = Yes)}$$

To classify, we'll simply compute these two probabilities and predict based on which one is greater

The Bayes Classifier



For the Bayes classifier, we need to "**learn**" two functions,

- the **likelihood,** $P(X|y)$
- the **prior,** $P(y)$

Model Parameters

1- How many parameters are required to specify the **prior**?

2- How many parameters are required to specify the **likelihood**?

The problem with explicitly modeling $P(X|y)$ is that there are usually too many parameters:

We'll run out of space
We'll run out of time
And we'll need tons of training data (which is usually not available)

MAP classification rule

MAP classification rule

- **MAP**: **M**aximum **A** **P**osterior
- Assign $X = (x_1, x_2, \dots, x_n)$ to c^* if

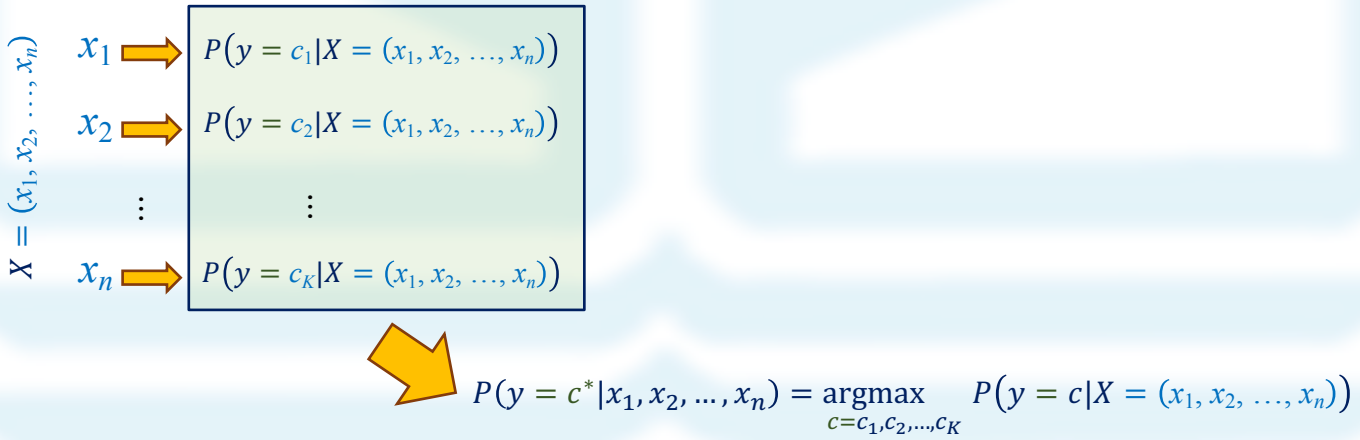
$$P(y = c^*|X = (x_1, x_2, \dots, x_n)) > P(y = c|X = (x_1, x_2, \dots, x_n)) \quad c \neq c^*,\ c = c_1, c_2, \dots, c_K$$

Algorithm. ***MAP classification rule***

Step1: Compute $P(y = c_i|X = (x_1, x_2, \dots, x_n))$ for $i=1,2,\dots, K$

Step2: Compute $c^* = \underset{c=c_1,c_2,\dots,c_K}{\text{argmax}} \; P(y = c|X = (x_1, x_2, \dots, x_n))$

Step3: Assign $X = (x_1, x_2, \dots, x_n)$ to c^* as it's predicted label



MAP classification rule

Example:

Assume $X = (x_1$=Red, x_2=SUV, x_3=Domestic),

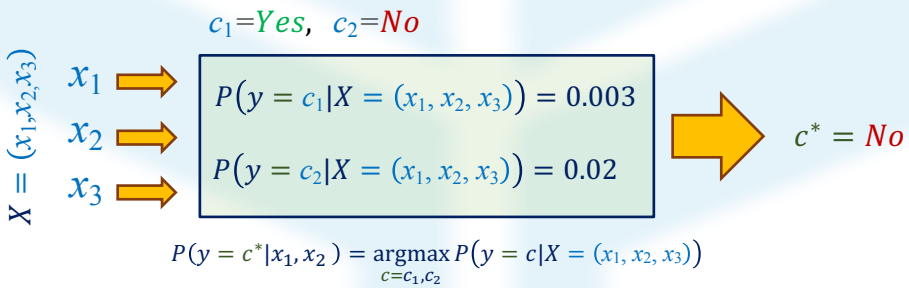
if $P(y = Yes|x_1$=Red, x_2=SUV, x_3=Domestic$) = 0.003$

and

$P(y = No|x_1$=Red, x_2=SUV, x_3=Domestic$) = 0.02$

then

$c^* = No$



$$P(y = c^*|x_1, x_2) = \underset{c=c_1,c_2}{\operatorname{argmax}} P(y = c|X = (x_1, x_2, x_3))$$

Method of *Generative classification* with the MAP rule

1. Apply Bayesian rule to convert them into posterior probabilities

$$P(y = c_i|x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n|y = c_i)P(y = c_i)}{P(x_1, x_2, \dots, x_n)} \propto P(x_1, x_2, \dots, x_n|y = c_i)P(y = c_i)$$

for i=1,2,..., K

2. Then apply the MAP rule:

$$P(y = c^*|x_1, x_2, \dots, x_n) > P(y = c|x_1, x_2, \dots, x_n) \quad c \neq c^*, \qquad c = c_1, c_2, \dots, c_K$$

Algorithm. *Generative classification* with the MAP rule

Step1: Compute $P(y = c_i)$ for *i=1,2,..., K*

Step2: Compute $P(x_1, x_2, \dots, x_n|y = c_i)$ for *i=1,2,..., K*

Step3: Apply Bayesian rule to compute $P\big(y = c_i|X = (x_1, x_2, \dots, x_n)\big)$ for *i=1,2,..., K*

Step4: Compute $c^* = \underset{c=c_1,c_2,\dots,c_K}{\text{argmax}} \; P\big(y = c|X = (x_1, x_2, \dots, x_n)\big)$

Step5: Assign $X = (x_1, x_2, \dots, x_n)$ to c^* as it's predicted label

Bayes classification

$$P(y|X) \propto P(X|y)P(y) = P(x_1, x_2, \dots, x_n|y)P(y)$$

Difficulty

learning the joint probability

The Naïve Bayes Model

- The ***Naïve Bayes Assumption***:

  Assume that all features are *independent* **given the class label y**

$$P(x_1, x_2, \dots, x_n|y) = \prod_{i=1}^{n} P(x_i|y)$$

Naïve Bayes Algorithm
(for *discrete valued Features*)

Input: Data set D $= \{(\boldsymbol{x}^{(i)}, y^{(i)})\}$, for i=1,2,…, m, such that $\boldsymbol{x}^{(i)}, y^{(i)} \in \{c_1, c_2, \dots, c_K\}$ and each feature $x_j^{(i)}$ can be take it's discrete value from set $\{x_{j1}^{(i)}, x_{j2}^{(i)}, \dots, x_{jn_j}^{(i)}\}$, for $j = 1,2, \dots, n$.

Output:

- Given an unknown instance $X^{new} = (x_1^{new}, x_2^{new}, \dots, x_n^{new})$, predict it's label.

***Training Phase*:**

- Estimate $P(y = c_i)$ with examples in D for i=1,2,…, K:

$$\hat{P}(y = c_i) \approx P(y = c_i)$$

- For every feature value x_{jk} of each feature x_j, estimate $P(x_j = x_{jk} | y = c_i)$ for k=1,2,…, n_j with examples in D:

$$\hat{P}(x_j = x_{jk} | y = c_i) \approx P(x_j = x_{jk} | y = c_i)$$

Output of training phase: Conditional probability table for each feature x_j, for $j = 1,2, \dots, n$.

Training is easy, just create probability tables.

***Testing Phase*:**

Given an unknown instance $X^{new} = (x_1^{new}, x_2^{new}, \dots, x_n^{new})$, predict it's label:

- Look up tables to assign the label c^* to X^{new}:

$$c^* = \underset{c=c_1,c_2,\dots,c_K}{\operatorname{argmax}} \; P(y = c) \prod_{j=1}^{n} \hat{P}(x_j^{new} | y = c)$$

Classification is easy, just multiply probabilities

Naïve Bayes Algorithm
(for *discrete valued Features*)

Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$P(\text{Outlook} = \text{Sunny}, \text{Play} = \textit{Yes}) = \frac{\#(\text{Outlook}=\text{Sunny}, \text{Play}=\textit{Yes})}{\#\ total\ samples} = \frac{2}{14}$ $P(\text{Play}=\textit{Yes}) = 9/14$

$P(\text{Outlook} = \text{Sunny}, \text{Play} = \textit{No}) = \frac{\#(\text{Outlook}=\text{Sunny}, \text{Play}=\textit{No})}{\#\ total\ samples} = \frac{3}{14}$ $P(\text{Play}=\textit{No}) = 5/14$

Naïve Bayes Algorithm
(for *discrete valued Features*)

Example: Play Tennis

Cont.

Training Phase

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

Temperature

$$P(Hot|Yes) = \frac{P(Hot, Yes)}{P(Yes)} = \frac{\frac{2}{14}}{\frac{9}{14}} = \frac{2}{9}$$

$$P(Cool|Yes) = \frac{P(Cool, Yes)}{P(Yes)} = \frac{\frac{3}{14}}{\frac{9}{14}} = \frac{3}{9}$$

$$P(Hot|No) = \frac{P(Hot, No)}{P(No)} = \frac{\frac{2}{14}}{\frac{5}{14}} = \frac{2}{5}$$

$$P(Cool|No) = \frac{P(Cool, No)}{P(No)} = \frac{\frac{1}{14}}{\frac{5}{14}} = \frac{1}{5}$$

$$P(Mild|Yes) = \frac{P(Mild, Yes)}{P(Yes)} = \frac{\frac{4}{14}}{\frac{9}{14}} = \frac{4}{9}$$

$$P(Mild|No) = \frac{P(Mild, No)}{P(No)} = \frac{\frac{2}{14}}{\frac{5}{14}} = \frac{2}{5}$$

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Naïve Bayes Algorithm
(for *discrete valued Features*)

Example: Play Tennis

Cont.

Training Phase

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

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

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

Naïve Bayes Algorithm
(for *discrete valued Features*)

Example: Play Tennis

Cont.

Testing Phase

Given a new instance X^{new} = (Outlook = *Sunny*, Temperature = *Cool*, Humidity = *High*, Wind = *Strong*), what is it's label?

- Look up tables achieved in the learning phrase:

P(Outlook = *Sunny* | Play=*Yes*) = 2/9
P(Outlook = S*unny* | Play=*No*) = 3/5
P(Temperature = *Cool* | Play=*Yes*) = 3/9
P(Temperature = *Cool* | Play==*No*) = 1/5
P(Huminity = *High* | Play=*Yes*) = 3/9
P(Huminity = *High* | Play=*No*) = 4/5
P(Wind = *Strong* | Play=*Yes*) = 3/9
P(Wind = *Strong* | Play=*No*) = 3/5
P(Play=*Yes*) = 9/14
P(Play=*No*) = 5/14

- Decision making with the MAP rule:

P(*Yes*|X^{new}) ≈ [P(Sunny|Y*es*)P(Cool|*Yes*)P(High|Y*es*)P(Strong|*Yes*)]P(Play=*Yes*) = 0.0053

P(*No*|X^{new}) ≈ [P(Sunny|N*o*) P(Cool|N*o*)P(High|*No*)P(Strong|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|X^{new}) < P(*No*|X^{new}), we label X^{new} to be *No*.

Naïve Bayes Algorithm
(*Zero conditional probability*)
If no example contains the feature value

m-estimate of probability

- Note that we estimated conditional probabilities P(A|B) by $\frac{n_c}{n}$ where n_c is the number of times A and B happened in same time and n is the number of times B happened in the training data
- *This can cause trouble if n_c=0.*
- To avoid this, we fix the numbers p and m beforehand:
  - A nonzero prior estimate p for P(A|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 $\frac{n_c+pm}{n+m}$.
- Just think of this as adding a bunch of samples to start the whole process
- If we don't have any knowledge of p, assume the attribute is uniformly distributed over all possible values

$$\hat{P}(x_{jk}|y = c_t) \approx \frac{n_c + pm}{n + m}$$

n_c: number of training examples for which $x_j = x_{jk}$ and $y = c_t$.

n: number of training examples for $y = c_t$.

p: prior estimate (usually, $p = \frac{1}{n_j}$, $(x_j \in \{x_{j1}^{(i)}, x_{j2}^{(i)}, \dots, x_{jn_j}^{(i)}\})$.

m: weight to prior (number of *virtual* examples, $m \geq 1$)

The parameter m is also known as equivalent sample size. It prevents the probabilities from being 0.

Naïve Bayes Algorithm
(for *Continuous valued Features*)

Input: Data set D $= \{(\boldsymbol{x}^{(i)}, y^{(i)})\}$, for i=1,2,..., m, such that $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$, $y^{(i)} \in \{c_1, c_2, \dots, c_K\}$.

Output:

- Given an unknown instance $X^{new} \in \mathbb{R}^n$, predict it's label.

***Training Phase*:**

- Estimate $P(y = c_i)$ with examples in D for i=1,2,..., K:
- Numberless values taken by a continuous-valued feature
- For each feature x_j estmate conditional probability $P(x_j|y = c_t)$ using *normal distribution*

$$\hat{P}(x_j|y = c_t) \approx P(x_j|y = c_t) = \frac{1}{\sqrt{2\pi}\,\sigma_{jt}} e^{-\frac{(x_j-\mu_j)^2}{2{\sigma_{jt}}^2}}$$

μ_j: mean(average) of feature values x_j of examples for which $y = c_t$

σ_{jt}: standard deviation of feature values x_j of examples for which $y = c_t$

Output of training phase: Conditional probability table for each feature x_j, for $j = 1,2, \dots, n$.

***Testing Phase*:**

Given an unknown instance $X^{new} = (x_1^{new}, x_2^{new}, \dots, x_n^{new}) \in \mathbb{R}^n$, predict it's label:

- Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
- Apply the MAP rule to assign the label c^* to X^{new}:

$$c^* = \underset{c=c_1,c_2,\dots,c_K}{\operatorname{argmax}} \; P(y = c) \prod_{j=1}^{n} \hat{P}(x_j^{new}|y = c)$$

Naïve Bayes Algorithm
(for *Continuous valued Features*)

Cont.

Example

- Temperature is naturally of continuous value.

  Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

  No: 27.3, 30.1, 17.4, 29.5, 15.1

***Training Phase*:**

Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n \quad \Longrightarrow \quad \mu_{Yes} = 21.64, \qquad \mu_{No} = 23.88$$

$$\sigma^2 = \frac{1}{N-1}\sum_{n=1}^{N} (x_n - \mu)^2 \quad \Longrightarrow \quad \sigma_{Yes} = 2.35, \qquad \sigma_{No} = 7.09$$

Output: two Gaussian models for $P(temp|c)$

$$\hat{P}(x_j|Yes) \approx \frac{1}{2.35\sqrt{2\pi}} e^{-\frac{(x_j-21.64)^2}{2\times 2.35^2}} = \frac{1}{2.35\sqrt{2\pi}} e^{-\frac{(x_j-21.64)^2}{11.09}}$$

$$\hat{P}(x_j|No) \approx \frac{1}{7.09\sqrt{2\pi}} e^{-\frac{(x_j-23.88)^2}{2\times 7.09^2}} = \frac{1}{7.09\sqrt{2\pi}} e^{-\frac{(x_j-23.88)^2}{50.25}}$$

References

1- https://cse.sc.edu/~rose/587/PPT/NaiveBayes.ppt
2- http://euclid.nmu.edu/~mkowalcz/cs495/notes/NaiveBayesSlides/lesson004.pdf