No.: Date:

Naive Bayes Classifier

Bayes Theorem

$$P(H|D) = \frac{P(D|H)\,P(H)}{P(D)}$$

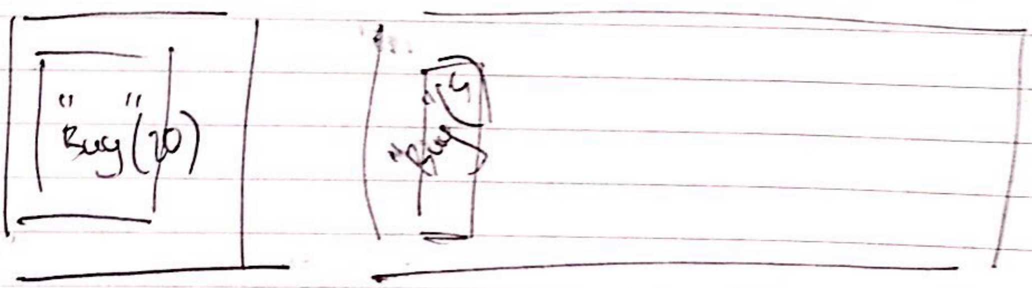
where $P(H)$ is prior probability of data holding this hypothesis b4 seeing data.
$P(D|H)$ is probability of data given a hypothesis.

Example

$H \in \{ SPAM, HAM \}$

$D \rightarrow$ "Buy"

SPAM (25) HAM (75)



$$P(\text{"SPAM"} \mid \text{"Buy"}) = \frac{P(\text{"Buy"} \mid \text{"SPAM"})\,P(\text{"SPAM"})}{P(\text{"Buy"})}$$

$P(\text{"Buy"} \mid \text{"SPAM"}) = \frac{20}{25}$

$P(\text{"SPAM"}) = \frac{25}{100}$

$P(\text{"Buy"}) = \frac{25}{100}$

No.: Date:

$$P(\text{"SPAM"} \mid \text{"Buy"}) = \frac{\frac{20}{25} \times \frac{25}{100}}{\frac{25}{100}}$$

$$= 0.8$$

$$\rightarrow 80\%$$

$$P(\text{"HAM"} \mid \text{"Buy"}) = \frac{\frac{5}{75} \times \frac{75}{100}}{\frac{25}{100}}$$

$$= 0.2$$

$$\rightarrow 20\%$$

$$h_{MAP} \rightarrow \underset{h \in \{SPAM, HAM\}}{\operatorname{argmax}} \frac{P(\text{"Buy"} \mid h_i)\, P(h_i)}{P(\text{"Buy"})}$$

(maximum a posteriori)

$$= \underset{h \in \{SPAM, HAM\}}{\operatorname{argmax}} P(\text{"Buy"} \mid h_i)\, P(h_i)$$

No.: Date:

BUT WAIT, WHAT IF WE HAVE ~~MULTIPLE~~ FEATURES.

$$H_{map} \rightarrow \underset{h \in \{SPAM, HAM\}}{\arg\max} \frac{P(\text{"Buy"} \cap \text{"Cheap"} \mid h_i)\, P(h_i)}{P(\text{"Buy"} \cap \text{"Cheap"})}$$

How we calculate for $P(\text{"Buy"} \cap \text{"Cheap"})$?

We make a NAIVE assumption
(hence the name)

that "Buy" and "Cheap" (features)
are INDEPENDENT.

and we recall what is characteristic
if 2 variables are INDEPENDENT?

e.g A and B are independent.

$$P(A) \times P(B) = P(A \cap B)$$

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

when occurences of one does not
affect probability ~~happ~~ of other occuring.

No.:

Date:

hence,

$$H_{map} \rightarrow \underset{h_i \in \{\text{SPAM}, \text{HAM}\}}{\operatorname{argmax}} P(\text{"Buy"}|h_i)\,P(\text{"Cheap"}|h_i)\,P(h_i)$$

My TEXT Classifier

all distinct words in Vocabulary.

n = number of features

$$T_{map} \Rightarrow \underset{t_i \in \{\text{electronics}, \text{photos}, \text{android}, \ldots\}}{\operatorname{argmax}} \left[\prod_{i=1}^{n} P(a_j | t_i)\right] \cancel{P(t_i)}$$

Comment: I treat as uniform, each $\frac{1}{10}$ so I can take out of argmax.

LEARN: How I find $P(a_j | t_i)$

<training> Vocabulary (distinct words>

for each topic:

<training> text

for each w in Vocab:

$$\frac{\text{number of } \sout{\text{occurrence}} \ w}{\text{total number of words in text}_i}$$

(of ~~class~~ topic!)

No.: Date:

CLASSIFY:

~~For each word in inputString:~~

~~if word is in Vocabulary~~

~~get for each topic:~~

~~get P(word|topic)~~

for each topic:

for each word in ~~Vocab~~ input String:

if word is in Vocabulary

get P(word|topic)

store $\prod$ P(word|topic) in maxMap

(with topic as key).

Check maxMap for $\prod$MAX ~~and return~~

~~topic in key for $\prod$~~

(key with highest value)