Naïve Bayes

21 September 2024 23:16

→ Probabilistic approach. → find posterior prob : what we believe after seeing data.

	+ve	−ve	Total
Cancer	78	2	80
No cancer	298	9622	9920
Total	376	9624	10000

total both +ve and cancer

$P(+ve|Cancer) = \frac{78}{80}$ ← total w/ cancer

$P(+ve|\text{No cancer}) = \frac{298}{9920}$

$P(-ve|Cancer) = \frac{2}{80}$

$P(-ve|\text{No cancer}) = \frac{9662}{9920}$

$P(cancer|+ve) = \frac{78}{376}$ ← total +ve

$P(\text{no cancer}/+ve)$

$P(cancer|-ve) = \frac{2}{9624}$

Okay, so net net,

$$\underbrace{P(canc|+ve)}_{\text{posterior}} = \frac{\overbrace{P(cancer)}^{\text{prior}} \cdot \overbrace{P(+ve|cancer)}^{\text{likelihood}}}{\underbrace{P(+ve)}_{\text{evidence.}}}$$

$h \in H$ → set of all possible hypotheses.

$$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$$

$$h_{MAP} = \text{argmax} P(h|D)$$

$$= \text{argmax} \frac{P(D|h) \cdot P(h)}{P(D)}$$

if all h are equally likely (uniform prior) + P(D) is same.

$\underset{\text{maximum likelihood}}{h_{ML}} = h_{MAP}$ for uniform prior $= P(D|h)$

Bayes optimal classifier : $\hat{v} = \text{argmax } P(v_j|D)$

$$= \text{argmax} \sum P(v_j|h_i)\, P(h_i|D)$$

weight is posterior probability.

Gibbs Algorithm : if $|H| \uparrow$, computation intensive.
Step 1 – randomly pick h_i from $P(h_i|D)$
Step 2 – classify target instance using h_i

Worst case : misclassification error = 2 × NB

Naive Bayes :

① For each v_j, $P(v_j)$
② For each a_i, $P(a_i|v_j)$
③ $v_{NB} = \text{argmax } P(v_j) \cdot \prod P(a_i|v_j)$

Zero frequency problem : If even one attr. $P(a_i|v_j)$ is 0, then $\prod P(a_i|v_j) = 0$

→ n_c = no. of examples w/ $v = v_j$ and $a = a_i$
→ n = no. of ex w/ $v = v_j$

Current skissue = $P(a_i|v_j) = \frac{n_c}{n} = 0$

Moving on to text documents.

Set of docs = examples

Prior prob = |docs_j| ... = P(v_j)

Namita Achyuthan

Moving on to text documents.

Set of docs = examples
Set of target vals = V
Set of distinct words = $|vocab|$

Prior prob = $\frac{|docs_j|}{|Examples|} = P(v_j)$

Function learns $P(w_k | v_j)$ that randomly drawn word from a doc in class v_j will be w_k.

Consider:

- $text_j$ = concatenated form of $docs_j$
- n = total no. of words in $text_j$

For each word in $|vocab|$: n_k = no. of times w_k appears in $text_j$

$$P(w_k | v_j) = \frac{n_k + 1}{n + |vocab|}$$

Laplace smoothing.

$|vocab|$ = dimension
if dimension ↑
no. of events ↑
exponential increase in size of dataset.

Namita Achyuthan