Friday, March 22, 2024 16:07

2. You are classifying emails as legitimate or spam and in doing so you would like to estimate the probability that a new email e containing the keywords (w_1 , w_2 , ..., w_n) is spam by taking all the emails in the training set with those keywords. In order words the probability is calculated as:

$$P(spam|e) = \frac{Number\ of\ spam\ emails\ with\ keywords\ (w_1, w_2, ..., w_n)}{Number\ of\ total\ emails\ with\ keywords\ (w_1, w_2, ..., w_n)} \qquad (1)$$

a. Explain why the plan may not work.
b. Describe the data sets for which this plan might work.
c. Using Naive Bayes assumption, explain how to get the probability of an email being spam? Show it mathematically.
d. How does Naive Bayes assumption caters the problem with your plan?

[a] - The keywords are assumed independent of each other, which is unlikely as it might be possible that more than one keywords are dependent on each other.
- Also some keywords might be more important than another.
- Curse of dimensionality is also a issue. As for decently sized dataset, no. of spam emails with keywords $w_1, w_2, ..., w_n$ might be '0' ~~or~~ very small, as it is unlikely for an email to contain 'n' keywords for large n.
- This means we might estimate probability '0' for most new emails or undefined if denominator is '0'.

[b] The idea will work for large datasets. We want emails with each possible set of keywords.
e.g. for 'n' key words, we need 2^n emails.
S: set of 'n' keywords.
$P(S) = 2^n$ → powerset of S.

[c] Let $y \to 1$ be spam
$y \to 0$ be not-spam

Bayes Theorem:-

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad -(a)$$

$$P(y=1) = 1 - P(y=0)$$

$$\therefore P(spam|w_1, w_2, ... w_n) = \frac{P(w_1, w_2, ..., w_n | spam) \cdot P(spam)}{P(w_1, w_2, ... w_n)} \quad -(b)$$

Now, for independent events A & B,

$$P(A, B) = P(A) \cdot P(B) \quad -(c)$$

By a, b, c

$$\therefore P(spam | w_1, ..., w_n) = \frac{P(spam) \cdot P(w_1, w_2, ..., w_n}{P(w_1) \cdot P(w_2) ... P(w_n)}$$

$$= \frac{P(spam) \prod_{i=1}^{n} P(w_i|spam)}{P(w_1|spam) ... P(w_n|spam) + P(w_1|not\ spam) ... P(w_n|not\ spam)}$$

$$= \frac{P(spam) \prod_{i=1}^{n} P(w_i|spam)}{P(spam) \prod_{i=1}^{n} P(w_i|spam) + P(not\ spam) \cdot \prod_{i=1}^{n} P(w_i|not\ spam)}$$

[d] In Naive Bayes solution, we are calculating $P(w_i|spam)$ for each word w_i.
Thus instead of set of keywords we are just looking for a single w_i, it decreases the chance of getting probability zero.
Also, the probability of each word is calculated instead of joint probability.