NAIVE BAYES

Presented By:Sanjeev A
Reg No:2022503057

Contents

BAYES THEOREM.

- Bayes' theorem gives the probability of a hypothesis C given evidence X as

$$P(C \mid X) = \frac{P(X \mid C) \cdot P(C)}{P(X)}$$

- Where :

- $P(C \mid X)$ → Posterior probability (probability of class C given features X)
- $P(X \mid C)$ → Likelihood (probability of features given the class)
- $P(C)$ → Prior probability of class C
- $P(X)$ → Evidence (probability of features occurring, regardless of class)

NAIVE BAYES ASSUMPTION.

- The Naive Bayes assumption is where the determining features are conditionally independent.
- Let's say the output of class C depends on two features X and Y then P(X|C) and P(Y|C) are independent events.
- This assumption is taken to reduce the mathematical complexity
- This is given as :

$$P(X \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

    - Where x_i is the ith features (word or a token)
- This is also called ***"CLASS CONDITIONAL INDEPENDENCE"***

NAIVE BAYES CLASSIFIER.

- Bayes theorem can be used to determine the class with highest probability given a set of features using class conditional independence.
- Let C be the class X be the sequence of input tokens, then according to bayes theorem we can estimate the probability of input sequence X belongs to class C as:

$$P(C \mid X) = \frac{P(C) \cdot \prod_{i=1}^{n} P(x_i \mid C)}{P(X)}$$

- Now we have to find the class with the highest probability which can be obtained by taking ***"argmax"*** of the above equation. We also notice that P(X) is constant and hence the final equation becomes:

$$\hat{C} = \arg\max_{C} \left[P(C) \cdot \prod_{i=1}^{n} P(x_i \mid C) \right]$$

LOGS AND LAPLACE SMOOTHING

- Multiplication of probabilities often leads to underflow as it is usually very small value. So to avoid it Logs are used and addition of results take place as follows

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

- Another Edge case is where an unseen word occurs. In such a case the value of $P(x_i|C)$ for the unseen word becomes zero resulting in the entire probability becoming zero even though other words in the input sequence are of high probability.
- To overcome this ***"Laplace Smoothing"*** is used as follows

$$P(x_i \mid C) = \frac{\text{count}(x_i, C) + 1}{\sum_{w \in V} \text{count}(w, C) + |V|}$$

Where V is the vocabulary set and |V| is the size of vocabulary.

LOGS AND LAPLACE SMOOTHING

- Multiplication of probabilities often leads to underflow as it is usually very small value. So to avoid it Logs are used and addition of results take place as follows

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

- Another Edge case is where an unseen word occurs. In such a case the value of $P(x_i|C)$ for the unseen word becomes zero resulting in the entire probability becoming zero even though other words in the input sequence are of high probability.
- To overcome this ***"Laplace Smoothing"*** is used as follows

$$P(x_i \mid C) = \frac{\text{count}(x_i, C) + 1}{\sum_{w \in V} \text{count}(w, C) + |V|}$$

Where V is the vocabulary set and |V| is the size of vocabulary.

Company Name

SPAM Vs NON SPAM USING MAP REDUCE

Step 1-Input Splits

Split the large document into chunks of manageable sizes and feed as input to mapper

Step 2-Mapper

Each Mapper gets a subset of (doc_id,c_l,text) and emits the CLASS_LABEL key as well as word count per class key.

Step 3-shuffler

Groups values by identical keys and then maps them to respective reducer task using hash of this key

Step 4-Reducer

Aggregates the output from mapper and finds the occurrence of each word per class and the number of words per class

Model Building and predication

Calculate the prior probabilities for each class and the posterior probability of each word per class using laplace smoothing. Now predict the class with the highest probability

Year/Quarter/Month

Code-LINK

Year/Quarter/Month

Thank you