Gaurang Dangyach, Ujjwal Kumar,
Aditya Singh Kaurav,
Pankaj Nirmal, Harsit Sinha **Assignment-2 CS771** 2022-23

Problem 1

Give detailed calculations explaining the various design decisions you took to develop your decision tree algorithm. This includes the criterion to choose the splitting criterion at each internal node (which essentially decides the query word that Melbo asks when that node is reached), criterion to decide when to stop expanding the decision tree and make the node a leaf, any pruning strategies and hyperparameters etc. (10 marks)

.

Solution: **Splitting criteria at each internal node:**

We used a variation of the ID3 algorithm.

Description of the algorithm used: First of all, instead of choosing a query from the complete dictionary, we chose word from the list of the words reaching that particular node. This automatically ensures that the query asked also satisfies the mask created by the previous all queries made to reach that node (i.e. If we know because of the previous query that letter at 2^{nd} position is '**e**' the new query will also be chosen from the list of words having letter '**e**' at 2^{nd} position) and therefore we will be able to make more efficient queries.

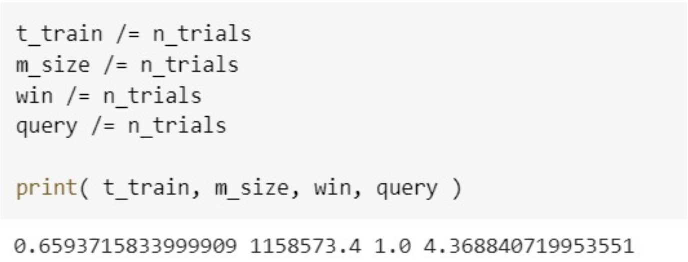
Secondly, instead of choosing a random word as query from my_words_idx (the list of wards reaching the node). We wanted to choose word which maximises entropy reduction but instead of testing all the words from my_words_idx, to reduce training time, we chose $int(\log_2(len(my\_words\_idx)))$ random words from my_words_idx and amongst those words we chose the one finally which reduced the entropy by highest value.

The reason for choosing this particular number is that so that for the first split, the algorithm checks about 12 words (if the dictionary length is about 5000 words) and eventually if there are only two words it chooses one word

Problem 2

Write code implementing your decision tree learning algorithm. You are not allowed to use any library other than numpy. This means that even use of scikit-learn is prohibited. Use of other libraries such as scipy, skopt, etc is also forbidden. Submit code for your chosen method in submit.py. Your code must implement a my_fit() method that takes a dictionary as a list of words and returns a trained decision tree as a model. The trained decision tree as a model should be a tree object. Every node in that tree should be a node object. There is no restriction on what attributes the tree object or the node objects may have and what methods those classes implement (i.e. feel free to implement your own Tree and Node classes) but the Node class must implement at least 2 methods: (a) Every non-leaf node should implement a get_child() method that takes a response and decides which child node to move to. (b) Every node (leaf as well as non-leaf) should implement a get_query() method that tells what query Melbo should ask when at that node. For leaf nodes, this would be the final query in that round after which the round will be terminated. We will evaluate your method on a different dictionary than the one we have given you and check how good is the algorithm you submitted (see below for details). Please go over the Google Colab validation code and the dummy submission file dummy_submit.py to clarify any doubts about data formats, protocol, etc. (30 marks)

Solution



```
t_train /= n_trials
m_size /= n_trials
win /= n_trials
query /= n_trials

print( t_train, m_size, win, query )
```

```
0.6593715833999909 1158573.4 1.0 4.368840719953551
```

Figure 1: Results obtained