Fingerprinting for movie titles

Khudayar Farmanli
Politecnico Di Torino
Data Science and Engineering
Computer Aided Simulations and Performance Evaluation
Torino, Italy
s276310@studenti.polito.it

***Abstract*—The objective of this homework is storing the list of titles from IMDB dataset where region is "IT" by using b-bit fingerprinting based on the b lowest bit of md5 hash function, applied to the utf-8 encoding scheme of the word string.**

I. Input Parameters

- Dataset - contains various characteristics related different movies, we just take the rows contain "IT" as region then eliminate all the other things and take the "title" part.
- m - the number of elements in final set which is 176339 in our particular case, there were more but when I create set it directly eliminated the duplicates, probably more pre-processing is needed but for our purpose it is not crucial .

II. Output parameters

- B_{exp} - minimum number of bits required to store set of titles in a fingerprint set where there is no collision, defined experimentally.
- B_{teo} - minimum number of bits required to store set of titles where probability of collision is at most 0.5 (probability of false positive which we consider it as ϵ based on $Pr(FP) \leq \epsilon$), defined theoritically by exploiting below formula:

$$B \geq \log_2 \frac{m}{\epsilon}$$

- The probability of false positive based on experimentally defined B_{exp} by exploiting above formula with considering $\epsilon = Pr(FP)$, where actual formula is this:

$$Pr(FalsePositive) = 1 - (1 - \frac{1}{n})^m$$

- Based on this probability again we calculate B_{teo}.
- Memory - taken by titles and fingerprint sets.

III. Main data structures

For developing this simulation I have used pandas library to read the tsv file from directory and after doing needed operations, I stored what I got to python set and then continue all the operations with python sets.

IV. Development of Simulation

- Reading tsv file from directory as pandas dataframe.
- Taking only the parts which contain "IT" as region.
- Storing only "title" part to python set.
- Defining the number of elements inside the set and assign it to value m to use later in formulas.
- Creating for loop to iterate over numbers which are considered number of bits, starts from 0 and continue till 100 to define minimum number of bits required to store fingerprints to the set without any collision by exploiting provided methods.
- Calculating 0.5 False Positive probability by exploiting formula.
- Calculating False Positive probability (ϵ) by considering number of bits the one we get from loop.
- Calculating B_{teo} again based on ϵ we got from previous step .
- Calculating the storage taken by original set of titles and fingerprint set.

V. Results

The obtained results are demonstrated below:

- The minimum number of B_{exp} is 34 to store fingerprints without collision.
- For Pr(FP) = 0.5, the required B_{teo} is 18.428 .
- By using B_{exp} we calculate Pr(FP) = 0.00001026 .
- By using value of Pr(FP) calculated based on B_{exp} we find $B_{teo} = 34.00060180140242$
- The storage required for set of original titles set is 21535600 and for fingerprints, it is equal to 14031672 bytes.

VI. Conclusion

At the end, I obtained below conclusions based on my experiments:

- The values obtained experimentally provide similar values when we use them to calculate theoritical ones in terms of minimum number of required bits and the probability of collision both, so we can say B_{teo} is good approximation of B_{exp}.
- The trade-off between Pr(FP) and memory when adopting fingerprinting with respect to not using it is that fingerprinting takes less memory capacity with respect to

original one. The point is, even in the case of Pr(FP) = 0, where it means we do not have any collision and the number of elements is exactly the same for fingerprint set with original set, we get less storage requirement for fingerprint set and if we let some collision occur and set become much smaller (because set only contains unique elements) we will even get a way smaller storage requirement.