Advanced Algorithms Assignment 3

Diogo Bento 93391

***Resumo* – O presente artigo detalha um estudo do uso de sketches no contexto da Unidade Curricular de Algoritmos Avançados da Universidade de Aveiro.**
Este artigo faz uso de CountMinSketch e analisa os resultados obtidos, que serão comparados aos valores obtidos por um contador normal.

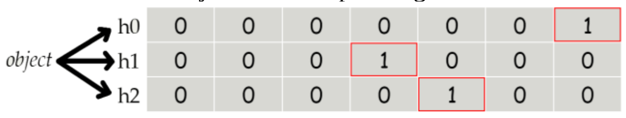
***Abstract* – This paper details a study of the use sketches, tackled within the context of the Advanced Algorithms Curricular Unit of Universidade de Aveiro.**
This article uses CountMinSketch and analyses the results obtained, using values obtained by a normal counter as a benchmark.

I. INTRODUCTION

In software counting event occurrences is commonplace.
Standard procedure is to increment the event's matching variable by 1 whenever the event occurs, but other alternatives may be appealing based on context.
Sketch algorithms are an alternative that does not store what values are being counted to save memory and ultimately results in a lossy, compact synopsis of data.

II. COUNTMINSKETCH

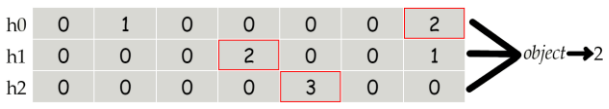
The CountMinSketch algorithm is backed by an integer table with dimensions **d** x **m**, initialized to 0.
When a value is counted using this algorithm **d** hash functions with values ranging from 0 to **m**-1 are calculated.
The table's **d** rows are then incremented by 1 at the indexes returned by their corresponding hash function.



h0	0	0	0	0	0	0	1
h1	0	0	0	1	0	0	0
h2	0	0	0	0	1	0	0

Fig. 1 – Sketch Count Update

As insertion count increases the probability of a object's matching buckets being correct goes down due to collisions, however due to how this algorithm works each bucket can only contain values equal to or greater than the correct value.
As such out of the **d** values related to a given object it is known that the one with the smallest value has the least error.



h0	0	1	0	0	0	0	2
h1	0	0	0	2	0	0	1
h2	0	0	0	0	3	0	0

Fig. 2 – Sketch Value Retrieval

III. ANALYSIS

A. On obtaining data and visualizations

In order to obtain data the German, French, and English versions of 'A Christmas Carol' were downloaded from Project Gutenberg.
The texts had their headers and footers manually removed following conversion to lowercase and stopword/non-alphabetic character removal.
Sketches were performed with 3 to 10 hash functions and hash value ranges varying from 10 to 200% of the unique word count for every language.
The maximum and average errors were then calculated, both on absolute and relative terms, as well as the deviation and these results were plotted as 3D scatter plots.
The obtained results can be found at *results.json* and were obtained by running *counters.py*.
The statistical calculations and plotting code are contained in *analysis.py* and running the program with the **--interactive** flag allows a user to interact with graphs instead of outputting them into an image file.

B. Results

For brevity's sake only a few of the generated graphs will be displayed.
All graphs will be provided in the */graphs* subdirectory of this report.

The Maximum Absolute Error was generally unpredictable across languagues.
Increasing the amount of hash functions seems to be consistently helpful while an increased hash range benefits greatly at first but then falls off in usefulness.

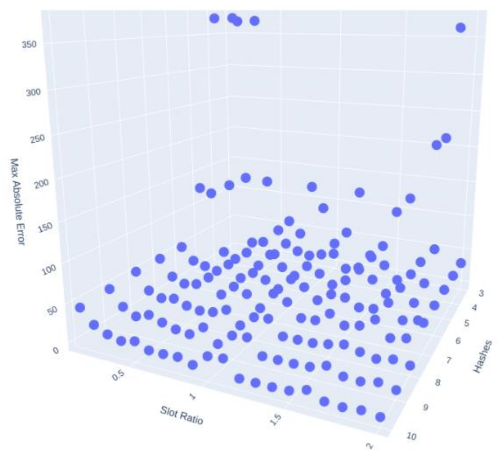


Fig. 3 – Maximum Absolute Error

Analyzing Maximum Relative Error yields similar results.

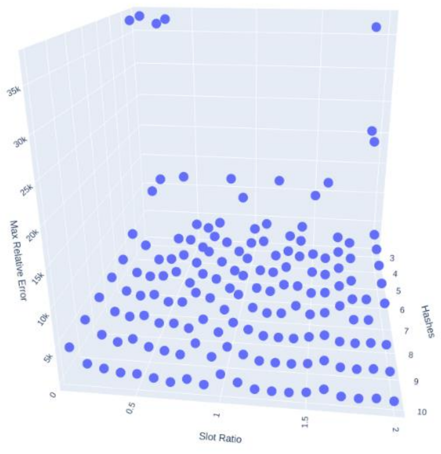


Fig. 4 - Maximum Relative Error

The Average Absolute Error behaves similarly for all languages.
 The error value seems to display early exponential behaviour as the number of hash functions diminuishes.
Similarly increasing the number of hash buckets decreases error greatly roughly until ratio 0.5.

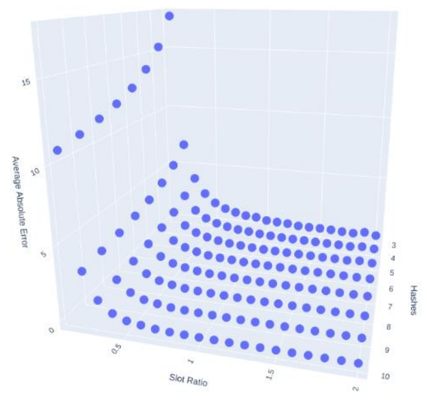


Fig. 5 - Average Absolute Error

The Average Relative error's behaviour is similar.

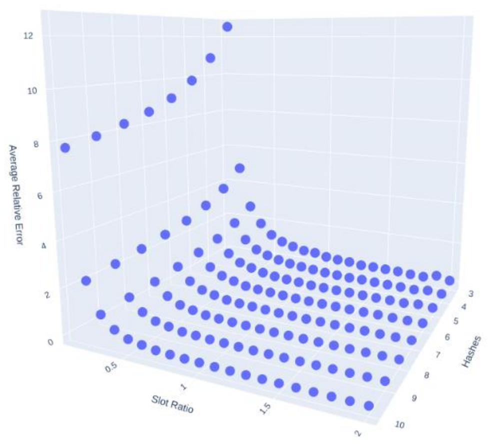


Fig. 6 - Average Relative Error

Deviation behaves similarly to Average Error, with the distinction that when using a low hash function count it is significantly more unstable, even as the size of each table increased.

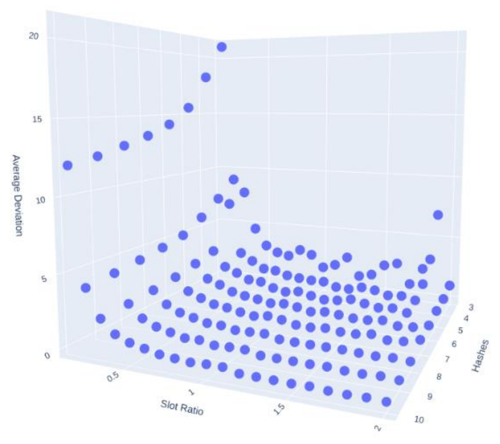


Fig. 7 - Average Deviation