Assignment 2:
Counting Distinct Elements in Practice

Wojtek Kowalczyk

Motivation

- There are several memory-efficient algorithms for approximate counting of distinct elements in data streams. We will focus on 2 algorithms:

  - The one described in the textbook (MMDS book, Section 4.4)
  - LogLog Counting (Durand-Flajolet)

- To apply these algorithms in practice, one should know the relation between:

  - the magnitude of the expected count (**N**)
  - the amount of the required memory (**M: number of bytes**)
  - **Relative Approximation Error**:

    RAE = abs(true_count - estimated_count)/true_count

- The algorithms and theoretical bounds are described in papers

Your Task

- Study both algorithms

- Implement them

- Run multiple experiments to experimentally establish the relation between:

    - the magnitude of the expected count (**N**)
    - the amount of the required memory (**M: number of bytes**)
    - Relative Approximation Error **(RAE)**

- **Write a report with your own description of both algorithms, their theoretical properties (according to the papers) and your own experimental results. Are your results consistent with theory? Which algorithm is better: from the textbook or from the original paper?**

Your Task in more details:

- Study the relevant papers on:
  - LogLog Counting (Durand-Martin)
  - Section 4.4 from the textbook

- Starting point:
  http://blog.notdot.net/2012/09/Dam-Cool-Algorithms-Cardinality-Estimation

Pay special attention to comments on the Python *hash()* function!
Don’t use it or use a dedicated library (e.g., *hashlib*) or your own implementation

Realize that the ordering and the number of repeating elements in the stream have no impact on the result. Therefore you can simulate your stream by generating a long sequence of random 32-bit long integers (they will look like 32-bit long hashes of distinct objects), so for your experiments you don’t need any hash function!!!

Your Task in more details (2):

- Run multiple experiments (for various numbers of distinct elements, number of buckets, number of setups) to establish/verify the trade-offs between the number of “hashes”/”buckets”, errors, the amount of required memory as a function of the number of distinct elements in the stream.

- required memory="memory required by your algorithm, assuming the most memory-efficient implementation“ – not necessarily 32bits per input record!

Write a report

The report should:

- Describe the algorithms, together with their theoretical properties in an accessible way

- Design/describe your own “experimental setup”

- Perform experiments and document your findings

- Summarize findings and make conclusions in a form of a Practical Guide:

How to count distinct elements in limited memory?

Moreover:

- Treat it as your own research project. In particular:
    - make your own choices concerning the experimental setup
    - decide yourself how much of “theoretical estimates” you want to use
    - decide yourself how you want to present the “practical guide”
    - (optionally): suggest a concrete application of approximate counting, e.g., to cybersecurity, marketing, databases, ...

Deadline: Tuesday, 9 October, 23:59
Submit both the report and the scripts in the same way as you did with A1
(this time ‘A1’ should be replaced with ‘A2’)