

DATA STREAM COMPUTATION FOR MONITORING STATISTICS OF MASSIVE WEBGRAPHS

LUCIANA S. BURIOL*, DEBORA DONATO*, STEFANO LEONARDI*
AND TOBIAS MATZNER+

ABSTRACT. We are interested in computing properties of large graphs, as the webgraph, using data stream algorithms. In this work we report results on computing the indegree rarity distribution of a graph obtained as a stream of edges. We implement a rarity algorithm proposed in the literature and show experimentally that the results approximate very well the optimal value with very limited use of memory and time. Moreover, considering some structure in the stream, we present results for the algorithm adapted for maintaining the rarity distribution of the number of cliques of size three.

1. INTRODUCTION

Data stream algorithms aim to maintain the underlying information of a stream of data, using small memory space. The data is processed on the fly, as it is generated, or it can also be read from second memory devices. Typical applications of data stream algorithms are originated from massive datasets such as network traffic measurements, telephone call records, biological datasets and atmospheric observations. In these applications is unnecessary or impractical to read data multiple times. In many cases, the data is not even stored. This work focuses on a new natural application for data streams. We are interested in using data stream algorithms for monitoring statistical and topological properties of large graphs such as the webgraph.

Several theoretical results have been proposed in this new research field, some of them have not yet been implemented and experimented, some of them are not practical. In this paper we observe how a data stream algorithm behaves in practice for computing the indegree rarity distribution of a graph over the arc arrivals. More specifically, we maintain the distribution of the number of nodes that has a given indegree over the total number of different nodes seen in the stream so far. We use the algorithm proposed by Datar and Muthukrishnan [4] and show experimentally that the results are very close to the optima even when a low precision is requested. The original algorithm proposes the use of min-wise hash functions, whereas we use universal hashing [4]. This decision is due to the fact that computing min-wise hashing consumes about two orders of magnitude more time than

universal hashing without providing better results in practice for the graphs we have tested.

When considering a specific structure in the data stream, other properties can be computed. For example, reading the stream in an adjacency list fashion, the same rarity algorithm can be used for estimating the density of minors such as small bipartite cliques.

The indegree of webpages is an important measure of their popularity. The experimental observation of the indegree distribution has been the subject of seminal works aimed to characterize the structure of the web-graph [1, 2]. This study has also revealed a surprising number of dense subgraphs, specifically bipartite cliques, of moderately small size [7], considered as cores of hidden web communities.

We use the α -rare algorithm of Datar and Muthukrishnan [4] for driving our experiments. Consider a stream of data. An item i is called α -rare if it appears exactly α times in the stream. Let's call $\#\alpha$ -rare the number of elements that appear exactly α times in the stream. Likewise, $\#distinct$ denotes the number of distinct items in the stream. The α -rarity ρ_α is defined as the ratio $\rho_\alpha = \frac{\#\alpha\text{-rare}}{\#distinct}$. In other words, the α -rarity of a stream is the measure of number of items that repeat exactly α times in the stream. The algorithm uses sublinear space and per item processing time, and use a constant number of hash functions, that depends on the precision required to the algorithm.

The algorithm uses min-wise hash functions proposed by [3]. This family of hash functions requires that all elements of a given set X have an equal chances to become the minimum element of the image of X under a random permutation π .

We conducted our experiments on streams of Wikipedia graphs. A graph of this type is generated from the link structure of the online and free-content encyclopedia Wikipedia (www.wikipedia.org).

The algorithms were coded in g++ version 3.3.2. The experiments were conducted in a Intel Pentium IV, with 1GB RAM, running Mandrake 9.0.

Due to the excessive computational time spent by min-wise hash functions, we use universal hash functions instead. We optimized an online available implementation of min-wise hash functions, but still it takes too long. The graph is read as a sequence of arcs. The endpoint of each edge is hashed for the rarity indegree hashing, and triples of nodes are hashed for counting the number of bipartite cliques of size three.

For a good approximation, a larger number of hash functions are required. But we observed, that even with a small number of hash functions, the results are close to the optima. Figure 1 presents results when using only 100 and 1000 hash functions. The lines are plot for a logarithmic number of indegree values. The plot omits results for indegree higher than 63 for the sake of clarity of the figure, but a complete plot would present additional lines on the bottom of the figure, appearing on increasing order of the number of edges processed.

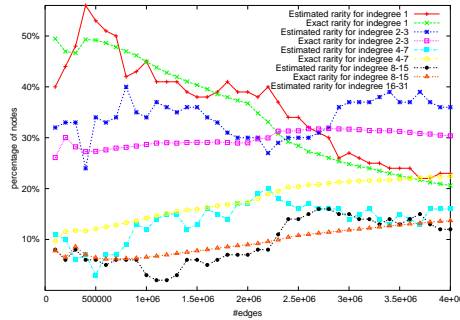


FIGURE 1. Estimated and exact indegree rarity distribution computed for edges arrivals of graph `wikiEN`. The estimation makes use of 1000 (graph on the left) and 100 (graph on the right) universal hashing functions. Values are presented to α up to 63, presented as \log_2 plot. This plot presents the percentage of nodes with a given indegree (y-basis) considering the amount of edges processed so far (x-basis). Results are plot every 100,000 items processed.

Tests with the algorithm using min-wise hash function are not presented due to the excessive time spent. For example, for computing 10,000 items and using 100 hash functions, for the indegree rarity distribution, the algorithm takes on average more than 200 seconds, whereas only 0.03 s. is spent using universal hashing.

Concluding remarks. We conclude that using universal hashing by this algorithm speed up a lot the codes, maintaining good approximations. As further work, we would like to test other algorithms that estimates interesting statistical and topological properties of webgraphs. Moreover, dynamic aspects of webgraphs also could be explored, as edges being inserted and removed over time. The α -rarity algorithm does not have solution for deletions. But a recent publication of Cormode, Muthukrishnan and Rozenbaum [5] presents an algorithm that maintain results considering also deletions. Likewise, a sampling algorithm was presented by Frahling, Indyk and Sohler [6], also for maintaining distributions under insertions and deletions.

REFERENCES

- [1] A.L. Barabasi and A. Albert. Emergence of scaling in random networks. *Science*, (286):509, 1999.

- [2] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, S. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33:309–320, June 2000.
- [3] A.Z. Broder, M. Charikar, A.M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Proc. of STOC*, pages 327–336, 1998.
- [4] M. Datar and S. Muthukrishnan. Estimating rarity and similarity over data stream windows. *LNCS*, 2461:323–334, 2002.
- [5] Irina Rozemberka G. Cormogode, S. Muthukrishnan. Summarizing and mining inverse distributions on data streams via dynamic inverse sampling. *Proceedings of the 31st VLDB Conferenct*, 2005.
- [6] C. Sohler G. Frahling, P. Indyk. Sampling in dynamic data streams and applications. *21st Annual Symposium on Computational Geometry*, 2005.
- [7] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber communities. pages 403–416, 1999.

(*) DIPARTIMENTO DI INFORMATICA E SISTEMISTICA, UNIVERSITÀ DI ROMA “LA SAPIENZA”,
VIA SALARIA 113, 00198 ROMA, ITALY
E-mail address: {buriol,donato,Stefano.Leonardi}@dis.uniroma1.it

(+) FAKULTÄT FÜR INFORMATIK, UNIVERSITÄT KARLSRUHE, KARLSRUHE, GERMANY
E-mail address: tobias.matzner@rechnerpost.de