Homework 3: Mining Data Streams

ID2222 Data Mining

Homework Groups 70,
Márk Antal Csizmadia
macs@kth.se

November 2021

1 Introduction

In this document, methods and results of mining data streams in the context of counting distinct elements in streams is discussed. The application is harmonic centrality estimation of nodes on streamed graph data with the HyperBall algorithm from the paper titled "In-Core Computation of Geometric Centralities with HyperBall: A Hundred Billion Nodes and Beyond" [1]. The HyperBall algorithm relies on the HyperLogLog algorithm from the paper titled "HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm" [2] for estimating the set size of distinct elements in data stream.

The discussion is accompanied by the exploration of the architecture of the code base submitted alongside with this report. Firstly, the methods used in the code are reviewed, and then some results are shown and discussed.

The accompanying code base is written in Python. For comparison with ground-truth centrality measures, the NetworkX Python library was used. Follow the `README.md` file on how to setup and run the code. The methods of the HyperLogLog algorithm used for the implementation of HyperLogLog counters are in the `hyperloglogcounter.py` file and the implementation of the HyperBall algorithm is in the `hyperball.py` file. To run the HyperBall algorithm with HyperLogLogCounters consult the `main.py` file.

2 Data

The data sets used in the project include the email-Eu-core network data [3] [4] and the Wikipedia vote network data [5] [6]. Both data sets are provided by the Stanford Large Network Dataset Collection and include directed, unweighted graphs with 1005 and 7115 nodes, and 25571 and 103689 edges, respectively. To download the data sets, run the `data.sh` bash script, or visit the URLs.

3 Methods

The theory described in this section is based on the main paper [1].

The **harmonic centrality** of a node `x` in a graph is defined as the sum of the reciprocals of its distances to nodes `y` such that

$$\sum_{y \neq x} \frac{1}{d\left(y, x\right)} = \sum_{d(y,x)<\infty, y \neq x} \frac{1}{d\left(y, x\right)} \tag{3.1}$$

where the distance from node `x` to `y` is the length of the shortest path from node `x` to node `y` in the graph. For nodes `x` that are not reachable from any node `y`, the harmonic centrality is null.

HyperLogLogCounters are probabilistic counters that allow to estimate the set size of distinct components in data streams. The memory requirement of these counters increase in loglog with the number of elements in the stream, hence the name (i.e.: very memory efficient). The counters are parameterised by `b`, the number of bits in the counter with $p = 2^b$ registers, that ultimately define the precision of the counters. The higher `b`, the more accurate the counters are.

A **ball** of radius `r` around node `x` is defined as

$$\mathcal{B}_G\left(x, r\right) = \{y | d\left(x, y\right) \leq t\} \tag{3.2}$$

where `t` is the a distance, that is a threshold on the shortest paths between nodes `x` and `y`. Furthermore, a ball of radius `r` around `x` satisfies

$$\mathcal{B}_G\left(x, 0\right) = \{x\} \tag{3.3}$$

$$\mathcal{B}_G\left(x, r+1\right) = \bigcup_{x \to y} \mathcal{B}_G\left(y, r\right) \cup \{x\} \tag{3.4}$$

As a result of this, the harmonic centrality of node `x` can be iteratively estimated such that

$$\sum_{y \neq x} \frac{1}{d\left(y,x\right)} = \sum_{t>0} \frac{1}{t} |\{y | d\left(y,x\right) = t\}| = \sum_{t>0} \frac{1}{t} \left(|\mathcal{B}_{G^T}\left(x,t\right)| - |\mathcal{B}_{G^T}\left(x,t-1\right)| \right) \tag{3.5}$$

HyperBall makes use of the latter finding by employing a HperLogLogCounter per node in the graph to estimate ball sizes at different distances, and use the differences between ball sizes to estimate the number of nodes at distance t appeared, hence making it possible to estimate geometric centralities, such as the harmonic centrality, on streamed graph data (i.e.: graph streamed by edges).

4 Relation to the Flajolet-Martin Algorithm

The Flajolet-Martin algorithm [7] is a probabilistic approach to count the size of multi sets in data streams. The HyperLogLog counter algorithm is based on the aforementioned algorithm but builds on top of it. Therefore, in the code, the the implementation of the Flajolet-Martin algorithm is not included.

5 Results

In the experiments discussed below, the ground-truth harmonic node centralities are computed with the networkx.algorithms.centrality.harmonic_centrality method. The true harmonic centralities are used to estimate the approximation error, or absolute mean error, of the probabilistic harmonic centralitiy estimates by HyperBall.

Results of varying the parameter `b` (thus the precision) of the HyperBall algorithm on the email dataset (1005 nodes and 25571 edges) and the wiki dataset (7115 nodes and 103689 edges) are shown in Table 1, and Table 2, respectively.

	Execution Time (seconds)	Error [%] (compared to networkx)
HyperBall (b=4)	9.2753	9.4024
HyperBall (b=5)	12.6544	3.4193
NetworkX	1.3376	0

Table 1: Results of HyperBall on the email dataset.

	Execution Time (seconds)	Error [%] (compared to networkx)	
HyperBall (b=4)	88.3968	9.1198	—
HyperBall (b=5)	106.6898	4.0030	—
NetworkX	25.1015	0	—

Table 2: Results of HyperBall on the wiki dataset.

The execution times in the tables are from single runs, and are merely to illustrate the trends discussed below. As the tables show, HyperBall has a very low discrepancy when compared to the true values of node centralities (networkx). As the parameter `b` is increased, the accuracy of HyperBall increases, and so does its execution time. The decrease in error and increaese in execution time is due to the HyperLogLogCounters having more registers ($p = 2^b$). When run on the wiki dataset (approximately 7x nodes and 4x edges), the execution time of HyperBall is approx. x10, while the execution time of networkx is approx. x20. The error rates remain comparable across datasets regardless of the graph size. To summarise:

- the parameter `b` affects the number of registers of the HyperLogLogCounters in the HyperBall algorithm, therefore the accuracy of the algorithm. Higher `b` results in more accurate estimates of the harmonic centralities of the nodes

- the execution time slowly increases as the graph size increases - which is one of the points of the HyperBall algorithm
- HyperBall requires low in-memory storage of variables (not investigated here)

6 Conclusion

The HyperBall algorithm, that is based on the HyperLogLog algorithm, is a fast and efficient approach to estimate the geometric centrality of nodes in a streamed graph.

References

[1] P. Boldi and S. Vigna, "In-core computation of geometric centralities with hyperball: A hundred billion nodes and beyond," in *2013 IEEE 13th International Conference on Data Mining Workshops*, 2013, pp. 621–628. DOI: 10.1109/ICDMW.2013.10.

[2] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier, "Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm," in *Discrete Mathematics and Theoretical Computer Science*, Discrete Mathematics and Theoretical Computer Science, 2007, pp. 137–156.

[3] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17, Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 555–564, ISBN: 9781450348874. DOI: 10.1145/3097983.3098069. [Online]. Available: https://doi.org/10.1145/3097983.3098069.

[4] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, 2–es, Mar. 2007, ISSN: 1556-4681. DOI: 10.1145/1217299.1217301. [Online]. Available: https://doi.org/10.1145/1217299.1217301.

[5] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 1361–1370, ISBN: 9781605589299. DOI: 10.1145/1753326.1753532. [Online]. Available: https://doi.org/10.1145/1753326.1753532.

[6] ——, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10, Raleigh, North Carolina, USA: Association for Computing Machinery, 2010, pp. 641–650, ISBN: 9781605587998. DOI: 10.1145/1772690.1772756. [Online]. Available: https://doi.org/10.1145/1772690.1772756.

[7] P. Flajolet and G. Nigel Martin, "Probabilistic counting algorithms for data base applications," *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 182–209, 1985, ISSN: 0022-0000. DOI: https://doi.org/10.1016/0022-0000(85)90041-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022000085900418.