Estimating Completeness in Streaming Graphs

Malay Bhattacharyya
Department of C.S.E.
University of Kalyani
malaybhattacharyya@klyuniv.ac.in

Supratim Bhattacharya
Department of C.S.E.
University of Kalyani
bhattacharya.supratim@gmail.com

Sanghamitra Bandyopadhyay*
Machine Intelligence Unit
Indian Statistical Institute
sanghami@isical.ac.in

ABSTRACT

Finding the completeness of a graph is important from various aspects. Considering the massive growth and dynamics of real-life networks, we readdress this problem in a streaming setting. We approach the problem of verifying the completeness of a graph by estimating the eigen values of a sketch of its adjacency matrix. Here, we provide the first approximation algorithm for estimating the completeness of a bipartite graph in the streaming model. The approach is further generalized for any arbitrary simple graph. We employ some useful recent results on ℓ_1 heavy eigen-hitters to construct the algorithms working in linear time and consuming sublinear space. The implementation of the algorithms have also been done and tested on a couple of networks. We illustrate the effectiveness of the proposed approaches in analyzing social, biological and other real-life networks.

Categories and Subject Descriptors

E.1 [**Data Structures**]: Graphs and Networks; F.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms

Theory, Design, Analysis

Keywords

Streaming model, complete graphs, heavy eigen-hitter

1. INTRODUCTION

Graphs and networks are suitable descriptors of various real-life environments like social activity, professional collaboration, web activity, etc. [12, 16]. They reflect local and global relationships between the objects, which they model. Studying how these objects interact with each other is useful from different perspectives. A graph is complete if all of its objects are connected to each other [5]. We are often interested to find out whether a graph is complete or not. Verifying the completeness of a graph consumes quadratic space and time with respect to its order. Considering the massive growth and dynamics of real-life networks, this becomes time/space inefficient. Therefore, designing sublinear algorithms is very important in massive data analytics [15].

Due to the explosive growth of volume of the real-life datasets (the emergence of *big data*), many of the computational problems have been redefined to overcome the bottlenecks of time/space complexity. In this paper, we readdress the problem of verifying the completeness of a graph in a streaming model. In streaming models, the data are available as a sequence of items (stream) and the data cannot be stored entirely [20]. Therefore, we have to examine the data within a few passes (may be single) as the available memory is also limited. Again, the processing time per item has to be sublinear. This imposes a new kind of uncertainty in computing beyond approximation and randomization.

Here, we consider that the adjacency matrix of a graph is available as a stream. Adopting a turnstile model, we estimate completeness of the corresponding graph based on the ℓ_1 norm. Initially, we study the problem for a bipartite graph in the streaming model and generalize it further for any arbitrary simple graph. We employ some recent approximation results on ℓ_1 heavy eigen-hitters to find out top k eigen values, respectively [3]. The proposed algorithms run in linear time and consumes space proportional to k^2 and the error parameters. We also demonstrate the effectiveness of the approaches in analyzing social and other real-life networks.

The current paper is organized as follows. Some background details and motivating applications are included in section 2 and section 3, respectively. Section 4 describes the state-of-the-art. Some theoretical results are provided in section 5 and based on this the proposed method is presented in section 6. Section 7 and section 8 cover some empirical results and discussions. Finally, section 9 concludes the paper.

2. PRELIMINARIES

Let us introduce some formal notations and standard definitions that will be used throughout the paper. We assume that $|S|$ denotes the size (cardinality) of a set S. A graph is a doublet $G = (V, E)$, where V denotes the set of vertices and $E \subseteq V \times V$ denotes the set of edges. The term *graph*

*Corresponding author.

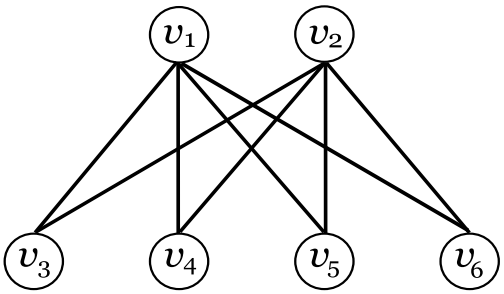

Figure 1: A complete bipartite graph with the sets of disjoint vertices $\{v_1, v_2\}$ and $\{v_3, v_4, v_5, v_6\}$.

is used to refer to a simple graph (without self-loops or parallel edges [7]) that is undirected and labeled. Suppose the adjacency matrix of a graph G is denoted as A_G. A subgraph of a graph contains a subset of the vertices and edges. A subgraph is said to be induced by a vertex set if it has exactly the edges that appear in the original graph over the same vertex set. A graph is complete if all of its vertices are connected to each other, i.e. $E = V \times V$. A clique is a complete subgraph (often restricted to be maximal) of a graph [5]. A graph is said to be bipartite if its vertices can be segregated into two disjoint subsets, say V_1, V_2, such that $V_1 \cap V_2 = \phi$, $V_1 \cup V_2 = V$ and $E \subseteq V_1 \times V_2$. A complete bipartite graph has exactly $|V_1| \times |V_2|$ edges (see Fig. 1). The other notations and graph-theoretic terminologies have their usual meaning, unless specified otherwise.

In this study, we assume that graphs are available under a streaming setting. In conventional data streaming models, the input stream $< s_1, s_2, \cdots >$ arrives sequentially (itemwise) and describes an underlying signal [11]. The streaming models vary one from the other depending upon how the s_i's represent the signal. Here, we consider a turnstile model where the underlying signal S is a one-dimensional function $S : [1 \ldots N] \rightarrow R$, R denoting the real space, where the s_i's are updates to $S[j]$'s [20]. Note that in case of streaming graphs, represented as a real symmetric adjacency matrix, we obtain a strict turnstile model by default where $S[j]$'s are always non-negative. Inspired from the earlier formalizations [9], we define a streaming graph in a strict turnstile model as follows.

Definition 1. A *streaming graph in a strict turnstile model* is a simple graph on n vertices $V = \{v_1, v_2, \ldots, v_n\}$ with edges $E = \{(v_i, v_j) : s_k = (i, j) \text{ for some } k \in [m]\}$, where the data items $s_k \in [n] \times [n]$ are available as an input stream $\mathcal{S} =< s_1, s_2, \ldots, s_m >$ pursuing a strict turnstile model.

In this paper, we address a stronger version of the streaming problems involving linear time and sublinear space. To formalize, we would like the per-item processing time, storage and overall computing time to be simultaneously $O(N, t)$, preferably $polylog(N, t)$, at any time instant t in the data stream. A sketch is often necessary to map the original space to a reduced space, retaining the necessary properties, to achieve this. We formally define a sketch as follows.

Definition 2. A *sketch* ψ of a data set x, with respect to some function f, is a projection of $x \rightarrow \psi$ from which one can compute $f(x)$.

Our proposed algorithms and the related theoretical results are mainly based on approximating the heavy eigen-hitters in a streaming graph. We include the definition of heavy eigen-hitters below.

Definition 3. The *ϕ-heavy eigen-hitters* of a graph G are the eigen values that are at least ϕ-fraction of the total mass of all the eigen values of the matrix A_G.

In Definition 3, the total mass of eigen values represents, in a simpler understanding, the summation of all the eigen values. A norm is a function that assigns a strictly positive length (or size) to each vector in a vector space, other than the zero vectors (having a length zero). In general, we define the ℓ_p norm as follows.

Definition 4. For any non-zero vector x, the *ℓ_p norm* is defined as

$$||x||_p = (\sum_{i}^{n} |x|^p)^{1/p}, \quad (1)$$

where $p \geq 1$ denotes a real constant.

Definition 5. The *ℓ_1 heavy eigen-hitters* are the heavy eigen-hitter values based on the total mass in ℓ_1 norm.

We discuss some real-life applications of estimating completeness in a streaming graph in the subsequent section.

3. MOTIVATING APPLICATIONS

Recent efforts in analyzing the available massive volume of data have assisted in both productivity growth and innovation in the industry and academia. It has immense potential in understanding the World Wide Web (WWW), financial sectors, medical analytics, public service domains, etc. Graphs can highlight large-scale global relations in effective ways for streaming data. Therefore, many futuristic applications are addressable with graph problems. Our target problem of estimating the completeness of a graph in a streaming model is also of high importance. We foresee a number of applications of this problem in various emerging areas of *big data*. Three of these are highlighted below.

- **Social network analysis:** Social communication at large-scale, rooted in WWW, has enabled the modeling of trillions of interactions between various social groups (e.g., researchers, students, actors, etc.). For the last decade or so, there is a massive growth of un-analyzed data in this area. Various other social communication methods like social networking websites (Facebook, Twitter, etc.), smart phones, multimedia applications, etc. are also contributing to these growing volumes of networked data. This increases the amount of dataflow per unit time and area. In accordance with this growth of data, analyses started with representing a network as a graph where the vertices are the elements and the edges denote their relations. Studying such large-scale graphs and their topologies might provide important features about the participating elements. Analyzing the dynamics of social networks is also interesting from different perspectives. Completeness can be verified for a portion of the streaming data so as to ensure whether the corresponding set of vertices (or a subgraph) arriving currently is

forming a clique or not. Again, the completeness of bipartite graphs may reveal interactions at the maximum scale between two different social groups. These are very important in a streaming setting. A recent attempt has shown that spreading rumors in real-life social networks is (surprisingly) faster than in complete graphs [8]. Therefore, studying graph completeness is also important for benchmarking analyses.

- **Analysis of biological networks:** Biological systems are often modeled as a network of biomolecules for understanding their cooperative activity. With the advancements of high-throughput technologies, enormous amount of experimental data is becoming accessible day by day. Biological networks may not be available as a stream but analyzing networks in linear/sublinear time/space is useful for dynamic sequencing of genes or for studying protein-protein interactions. Estimating completeness in such large-scale biological networks might help in prompt identification of strongly connected components. In a spreading disease network, biomolecules get rapidly affected by one another and such behaviors can be analyzed with completeness verification models. The broader goal is certainly to facilitate the system level understanding of cell-to-cellular components and its subprocesses.

- **Studying communication networks:** The number of communication service providers has rapidly increased over the last few decades around the world. Their growth has not only increased the volume of data but also its variability. This type of data can also be modeled as a network to understand various properties. Analyzing such networks might help the service providers to decide whether they should involve new services. Estimating completeness in such networks will be helpful in identifying the saturation of connectivity. This will invoke the demand of new communication providers. Again for the better understanding of a dynamic setting, we need to study the communication networks modeled on streaming graphs.

In the following section, we discuss the state-of-the-art of finding completeness of graphs, estimating cliques and the related progresses in streaming algorithms.

4. RELATED WORKS

Verifying the completeness of a graph is a special case of clique problems. A graph is complete if its clique number is of the order of the graph. Finding the maximum order clique in a graph is known to be an NP-hard problem [5]. An important study about a decade ago showed that the approximation of a maximal clique in polynomial time is hard within a factor of $n^{1-\varepsilon}$ (for any $\varepsilon > 0$), unless NP = ZPP (where n is the number of vertices in the graph) [14]. ZPP stands for zero-error probabilistic polynomial time. The problems in ZPP can be exactly solved in expected polynomial time by a probabilistic algorithm. It is strongly believed that ZPP $\subset$ NP and the hypothesis NP $\neq$ ZPP is almost as strong as P $\neq$ NP [14], where P denotes the decision problems solvable in polynomial time using a deterministic Turing machine. For this reason, many of the recent algorithms to solve the maximum clique problem (MCP) are based on metaheuristic approaches [4]. To approximate cliques, spectral approaches showed promise in earlier studies. Spectral graph theory is also important in analyzing the bounds of completeness. A few attempts were made earlier to estimate the chromatic number of a graph using eigen values [22], which can be further related with the clique number of a graph. Based on eigen value computations, several upper bounds on the clique number were derived previously [2]. Recently, these bounds (and also lower bounds) were tightened further [6]. Current studies indicate that a relation with the spectral radius with the clique might help us to estimate the upper bound of the clique in a streaming model [17].

Streaming algorithms have been in focus for more than a decade. But this domain is still in a nascent stage. The limited earlier contributions before 2005 have been well reviewed in [20]. While presenting this survey, Muthukrishnan also addressed some real life problems based on streaming models. Following this, diverse efforts were made to revisit and solve a number of problems in a streaming setting. There were studies on matrix approximation, matrix decomposition, low rank approximation, ℓ_p regression, etc. [13, 17, 18]. There has been an influential line of work on computing a low-rank approximation of a given matrix, starting with the works of [10, 21]. A lot of works were done on linear algebra in a streaming model. Also low rank approximation made the analysis of massive data less complicated. Very recently, the ℓ_1 and ℓ_2 heavy eigen-hitter problems have been estimated in the streaming model in a lower dimension [3]. Notably, the heavy eigen-hitters problem was first proposed in [20]. Andoni and Huy achieved a success probability of $\frac{5}{9}$ [3]. They also estimated the residual error with the same probabilistic accuracy. Sampling and sketching methods for producing low-complexity approximations of large matrices is in focus for the last few decades. We estimated the completeness of a graph in the streaming model based on the computations of ℓ_1 heavy eigen-hitters.

5. THEORETICAL RESULTS

In this section, we present some useful theoretical outcomes and derive some new results that will be helpful in devising the proposed algorithms. Let A_G be a real symmetric $n \times n$ ($n \geq 1$) matrix denoting the adjacency relations in a graph G. Further assume $\lambda_i(A_G)$ be the i^{th} largest eigen value of A_G in absolute value. Now, if ψ represents a sketch of the matrix A_G where $\psi = PA_GP^T$. Then, we have the following important result from a recent study [3].

THEOREM 1. *There is a linear sketch of the real symmetric matrix A_G, of dimension $n \times n$, using space $O(k^2\epsilon^{-4})(\epsilon > 0,\ k \in \{1, 2, \ldots, n\})$, from which one can produce values $\tilde{\lambda}_i$, for $i \in [k]$, satisfying the following with at least $\frac{5}{9}$ success probability*

$$|\lambda_i(A_G) - \tilde{\lambda}_i| \leq \epsilon|\lambda_i(A_G)| + \frac{1}{k}S_1^{k+1},$$

where $S_1^{k+1} = \sum_{i>k} |\lambda_i(A_G)|$ denotes the residual "ℓ_1 error".

Now, we derive the following result on bipartite graphs using the previous claim in Theorem 1.

THEOREM 2. *On fixing a value of $\epsilon > 0$, one can ensure whether G is a complete bipartite graph by deriving a linear sketch ψ from A_G whose top two heavy eigen-hitters in absolute value should be the same satisfying*

$$\lambda_1(\psi) = (1 \pm \epsilon)\lambda_1(A_G) \pm S_1^2,$$

and

$$\lambda_2(\psi) = (1 \pm \epsilon)\lambda_2(A_G) \pm 0.5S_1^3,$$

and the third largest eigen value satisfies

$$\lambda_3(\psi) = \pm 0.\dot{3}S_1^4.$$

PROOF. The eigen values of a complete bipartite graph G can be ordered as $\{\lambda_1(A_G), 0, \ldots, 0, \lambda_n(A_G)\}$, where $\lambda_1(A_G) = -\lambda_n(A_G) = \lambda$ (say) [2]. Therefore, if we obtain a decreasing order of the eigen values of A_G in absolute value, we would get $\{\lambda, \lambda, 0, \ldots, 0\}$. Since G does not contain any self-loops, the trace of A_G should be zero. Then, we can write

$$\sum_{i=1}^{n} \lambda_i(A_G) = 0.$$

It is understandable that if the eigen values are decreasingly ordered by absolute value, say $\{\lambda_1'(A_G), \lambda_2'(A_G), \ldots, \lambda_n'(A_G)\}$, and if $\lambda_1'(A_G) = \lambda_2'(A_G)$ and $\lambda_3'(A_G) = 0$, then rest of the eigen values of A_G will be certainly zero. This is because the rest of the eigen values cannot be negative (being in absolute value) and no more than zero (being in decreasing order). So, it is sufficient for A_G, to have the first two largest eigen values same in absolute value and the third one zero, for claiming that the corresponding graph G is complete bipartite. Now, from Theorem 1, we can derive that the first k eigen values, for a particular $\epsilon > 0$, will satisfy the following for a linear sketch ψ

$$\lambda_i(\psi) = (1 \pm \epsilon)\lambda_i(A_G) \pm \frac{1}{k}S_1^{k+1}, \tag{2}$$

Using Eqn. (2), one can verify whether the first two largest eigen values are same and estimate their values from the sketch ψ satisfying

$$\lambda_1(\psi) = (1 \pm \epsilon)\lambda_1(A_G) \pm \frac{1}{1}S_1^{1+1}$$
$$\implies \lambda_1(\psi) = (1 \pm \epsilon)\lambda_1(A_G) \pm S_1^2.$$

and similarly

$$\lambda_2(\psi) = (1 \pm \epsilon)\lambda_2(A_G) \pm \frac{1}{2}S_1^{2+1}$$
$$\implies \lambda_2(\psi) = (1 \pm \epsilon)\lambda_2(A_G) \pm 0.5S_1^3.$$

Again, one can verify whether the third largest eigen value is zero and estimate its value from the sketch ψ satisfying

$$\lambda_3(\psi) = (1 \pm \epsilon)\lambda_3(A_G) \pm \frac{1}{3}S_1^{3+1}$$
$$\implies \lambda_3(\psi) = (1 \pm \epsilon).0 \pm 0.\dot{3}S_1^4$$
$$\implies \lambda_3(\psi) = \pm 0.\dot{3}S_1^4.$$

This altogether completes the required proof. □

THEOREM 3. *On fixing a value of $\epsilon > 0$, one can ensure whether G is a complete graph by deriving a linear sketch ψ from A_G whose top two heavy eigen-hitters in absolute value satisfy the following*

$$\lambda_1(\psi) = (1 \pm \epsilon)(n-1) \pm S_1^2.$$

and

$$\lambda_2(\psi) = (\epsilon \pm 1) \pm 0.5S_1^3.$$

PROOF. The eigen values of a complete graph G can be ordered as $\{n-1, -1, ..., -1\}$ [2]. Therefore, if we obtain a decreasing order of the eigen values of A_G in absolute value, we would get $\{n-1, 1, \ldots, 1\}$. Since G does not contain any self-loops, the trace of A_G should be zero. Then, we can write

$$\sum_{i=1}^{n} \lambda_i(A_G) = 0.$$

It is understandable that if the eigen values are decreasingly ordered by absolute value, say $\{\lambda_1'(A_G), \lambda_2'(A_G), \ldots, \lambda_n'(A_G)\}$, and if $\lambda_1'(A_G) = n-1$ and $\lambda_2'(A_G) = 1$, then certainly the rest of the eigen values of A_G should also be one. This is because the rest of the eigen values cannot be negative (being in absolute value) and no more than one (being in decreasing order). So, it is sufficient for A_G, to have the first two largest eigen values as $n-1$ and one, respectively, for claiming that the corresponding graph G is complete. We have already derived that the first k eigen values, for a particular $\epsilon > 0$, will satisfy Eqn. (2) for a linear sketch ψ. Then using this, the first largest eigen value can be estimated as

$$\lambda_1(\psi) = (1 \pm \epsilon).\lambda_1(A_G) \pm \frac{1}{1}S_1^{1+1}$$
$$\implies \lambda_1(\psi) = (1 \pm \epsilon).(n-1) \pm S_1^2.$$

and the second largest eigen value can be estimated as

$$\lambda_2(\psi) = (1 \pm \epsilon).\lambda_2(A_G) \pm \frac{1}{2}S_1^{2+1}$$
$$\implies \lambda_2(\psi) = (1 \pm \epsilon).(-1) \pm 0.5S_1^3$$
$$\implies \lambda_2(\psi) = (\epsilon \pm 1) \pm 0.5S_1^3.$$

This altogether completes the required proof. □

In the next section, we present our approaches for completeness verification of bipartite graphs and any arbitrary graph in a streaming setting.

6. PROPOSED METHOD

Our algorithms are principally based on the concept of generating a projection P, of size $O(k/\epsilon^2)$ by n, and computing the sketch $\psi = PA_GP^T$ for ℓ_1 to retain (and thus estimate) the properties of the heavy eigen-hitters as close as possible. This saves the space requirements and computational time together. The method of verifying completeness of bipartite graphs is formally presented as Algorithm 1. Theorem 2 serves as the base of this approach. Note that, for estimating the completeness of bipartite graphs, we require the top two eigen values in absolute value to be same in original matrix A_G, i.e. $\lambda_1(A_G) = \lambda_2(A_G) = \lambda$ (say). Therefore, the eigen value differences in the sketch matrix ψ result into the following relation (using Theorem 2).

$$\lambda_1(\psi) - \lambda_2(\psi) = \pm 2\epsilon\lambda \pm S_1^2 \pm 0.5S_1^3.$$

This provides an error estimation of the relation derived in Theorem 2.

In Algorithm 2, we present the formal approach of verifying completeness of any arbitrary graph. This is a more generalized approach with a fewer number of eigen value estimations. We mainly use the results from Theorem 3 to devise this algorithm. In both the algorithms described, for

Algorithm 1 An algorithm for estimating completeness of bipartite graphs

Input: The adjacency matrix A_G of the bipartite graph G.
Output: The decision about the completeness of G.
Algorithmic Steps:

1: Obtain a sketch $\psi = PA_GP^T$, where P is a $t \times n$ matrix with $\Theta(\frac{\log^2 n}{\epsilon^2})$-wise independent entries identically distributed as $N(0, \frac{1}{t})$.
2: Compute the top three largest eigen values of ψ in the decreasing order denoted as $\lambda_1(\psi)$, $\lambda_2(\psi)$ and $\lambda_3(\psi)$, respectively.
3: **if** $\lambda_1(\psi) = \lambda_2(\psi)$ and $\lambda_3(\psi) = \pm 0.\dot{3}S_1^4$ **then**
4: G is a complete bipartite graph.
5: **end if**

bipartite and general graphs, the $\Theta(\frac{\log^2 n}{\epsilon^2})$-wise independent entries for the random matrix P are generated following an earlier approach [1].

Algorithm 2 An algorithm for estimating completeness of any arbitrary graph

Input: The adjacency matrix A_G of the graph G.
Output: The decision about the completeness of G.
Algorithmic Steps:

1: Obtain a sketch $\psi = PA_GP^T$, where P is a $t \times n$ matrix with $\Theta(\frac{\log^2 n}{\epsilon^2})$-wise independent entries identically distributed as $N(0, \frac{1}{t})$.
2: Compute the top two largest eigen values of ψ in the decreasing order denoted as $\lambda_1(\psi)$ and $\lambda_2(\psi)$, respectively.
3: **if** $\lambda_1(\psi) = (1 \pm \epsilon)(n-1) \pm S_1^2$ and $\lambda_2(\psi) = (\epsilon \pm 1) \pm 0.5S_1^3$ **then**
4: G is a complete graph.
5: **end if**

7. EMPIRICAL STUDY

We considered two real-life networks for testing the outcome of the proposed algorithms. The algorithms were implemented in MATLAB and the simulations were performed on an HP Laptop with Intel(R) Core(TM) i5-2410M processor running at 2.30 GHz speed and having 4 GB primary memory. one of these networks is a complete bipartite graph and the other one is sparse. The experimental procedures are briefly discussed below.

7.1 Study on Synthetic Networks

We have constructed two synthetic networks, one complete bipartite and another complete network, both having orders 40 for performance analysis of the proposed approaches. The complete bipartite network has equal number of partitions. In both these cases, dimension of the sketch matrix becomes $t \times 40$. We have varied t from 10 to 25 and several arbitrary matrices are generated by employing random selection method on a normal distribution (identically) with parameters $(0, 0.01)$. Finally, the eigen values are estimated (using Algorithm 1 and Algorithm 2) and compared with the original values. The obtained eigen values indicate their completeness. The Figs. 2(a-b) show the accuracies of the eigen values against the difference of dimensions between the sketch and the original matrix. It becomes clear

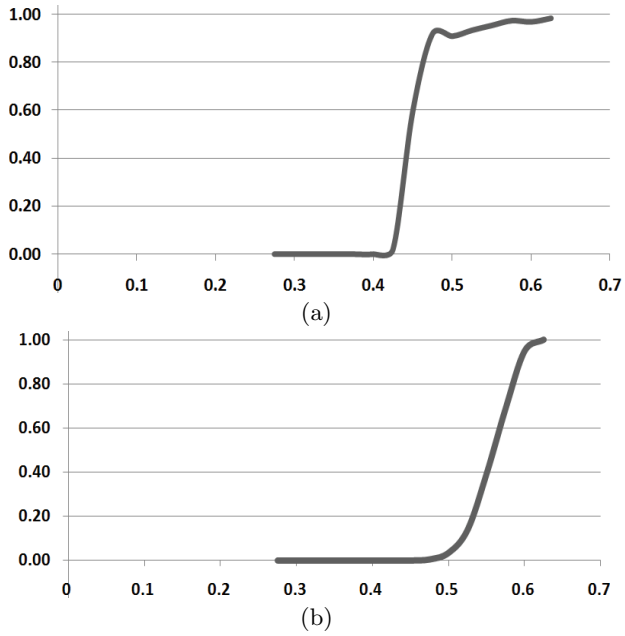

Figure 2: The average accuracy obtained against the sketch difference of the synthetic (a) complete bipartite network and (b) complete network.

that the performance rapidly improves after certain threshold. So, proper selection of the dimension of the sketch vector is very much important.

7.2 Study on Social Networks

We have used a large-scale social interaction data of Facebook, consisting of 'circles' (denoting 'friends lists'), from a recent study [19]. This interaction data is used to construct a large undirected unweighted social network having 4039 vertices and 88234 edges. The average clustering coefficient of the network is found to be 0.61, establishing that it is not complete. We have analyzed this and computed a sketch of dimension 100×4039 with elements identically distributed in $N(0, 0.01)$. Finally, Algorithm 2 is applied on this. The obtained eigen values are found to be quite far from the values supporting its completeness (as per Theorem 3).

8. DISCUSSION

The approaches to completeness verification presented in the current paper is important from two different perspectives. First, the theoretical results provided might be useful in estimating the clique number of a graph that depends on the number of eigen values no greater than '-1' [2]. Secondly, the implementation details might be useful is developing many other algorithms that work in a streaming setting. Our assumption of a strict turnstile model does not weaken the results because the problem demands so. The real symmetric form of adjacency matrices of streaming graphs can be well captured using a turnstile model. Our attempts of utilizing heavy eigen-hitters are also very promising. The approaches to standard vector heavy hitters return the elements that are most frequent (heavy coordinates) [20]. On the contrary, our methods do not find the elements that are heavy hitters. This saves an additional factor of $O(\log n)$ to the space requirements (ignoring the random seed size).

Another significant advantage of the proposed algorithms is that their performances are independent of the seed selection for generating random matrices. It might appear that the algorithms work only on static graphs (i.e. on a fixed adjacency matrix) to compute the sketches. But they also work in a streaming setting because the algorithms, being linear in computational time, are also capable of supporting arbitrary updates to the matrix. The major limitation of our approaches is that the success probabilities are still low since they rely on multiple applications of Theorem 1. Again, it might be criticized that large real-life networks, like social entities and their interactions, are rarely complete. But the approaches of estimating completeness are still applicable where time is a major constraint. Therefore, the proposed methods are very much generalized.

9. CONCLUSION

In this paper, we have provided the first approximation algorithms for estimating the completeness of bipartite graphs and, in general, any arbitrary graph. Our results are promising and useful for diverse applications. We have also implemented the proposed algorithms and verified results on two test cases. The approaches are promising for many further directions of research on big data at a network level. However, the success probabilities of our algorithms are still poor that we would like to improve in near future. Again, we wish to extend the current analysis using property testing.

10. ACKNOWLEDGMENTS

The authors are thankful to Huy L. Nguyen in the department of Computer Science of Princeton University for his important feedback over an initial draft of the paper.

11. REFERENCES

[1] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Stuctures and Algorithms*, 3(3):289–304, 1992.

[2] A. T. Amin and S. L. Hakimi. Upper bounds on the order of a clique of a graph. *SIAM Journal on Applied Mathematics*, 22(4):569–573, 1972.

[3] A. Andoni and H. L. Nguyen. Eigenvalues of a matrix in a streaming model. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1729–1737, New Orleans, USA, 2013.

[4] S. Bandyopadhyay and M. Bhattacharyya. Mining the largest dense vertexlet in a weighted scale-free graph. *Fundamenta Informaticae*, 96:1–25, 2009.

[5] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D. Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization: Supplementary Volume A*, pages 1–74. Kluwer Academic, Dordrecht, 1999.

[6] M. Budinich. Exact bounds on the order of the maximum clique of a graph. *Discrete Applied Mathematics*, 127:535–543, 2003.

[7] R. Diestel. *Graph Theory*. Springer-Verlag Heidelberg, New York, 2005.

[8] B. Doerr, M. Fouz, and T. Friedrich. Why rumors spread fast in social networks. *Communications of the ACM*, 55(6):70–75, 2012.

[9] J. Feigenbaum, S. Kannan, A. McGregor, and S. Suri. Graph distances in the data stream model. *SIAM Journal of Computing*, 38(5):1709–1727, 2008.

[10] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[11] A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Surfing wavelets on streams: One pass summaries for approximate aggregate queries. In *Proceedings of the* 27^{th} *International Conference on Very Large Data Bases*, pages 79–88, San Francisco, CA, USA, 2001.

[12] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proccedings of the National Academy of Sciences USA*, 99:7821–7826, 2002.

[13] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[14] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[15] P. Indyk, R. Levi, and R. Rubinfeld. Approximating and testing k-histogram distributions in sub-linear time. In *Proceedings of the* 31^{st} *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 15–22, Scottsdale, Arizona, 2012.

[16] H. Ino, M. Kudo, and A. Nakamura. Partitioning of web graphs by community topology. In *Proceedings of the* 14^{th} *International World Wide Web Conference*, pages 661–669, Chiba, Japan, 2005.

[17] R. Kannan and S. Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4(3-4):157–288, 2009.

[18] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[19] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *Proceedings of the Neural Information Processing Systems*, pages 548–556, 2012.

[20] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005.

[21] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: a probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–235, 2000.

[22] H. S. Wilf. The eigenvalues of a graph and its chromatic number. *Journal of the London Mathematical Society*, 42(1):330–332, 1967.