Subgraph Frequencies and Network Classification

Quaizar Vohra

1. Introduction

Current metrics used in summarizing networks, such as degree distribution, average diameter and clustering coefficient, provide a very coarse understanding of their structure. One would like to have finer-grained summaries of networks which allow making distinction between structurally different networks. A class of such interesting graph statistics are the Small Subgraph Frequencies. These subgraph frequencies allow us to study the underlying structure of networks by distinguishing mathematical properties from social behavior related properties. Given a small graph *H* (template) of *k* nodes and a large graph (network) *G*, the subgraph frequency *#(H,G)* is the fraction of *k*-tuples of nodes of *G* such that the induced graph is *H*. One can consider the subgraph frequencies for all such graphs *H* on *k* nodes and summarize the network by stacking the frequencies into a vector. This method was proposed in [1] and was studied as a local property for a large collection of small subgraphs in a large social network. The results from this study suggest that studying the frequency statistics of large networks as a single unit may provide unique insights. To our knowledge, this statistic has not been studied as a global property of a large network. To this end we want to explore advanced methods for computing subgraph frequencies for very large networks. We will then compute the statistics on some real networks and use it for classifying these networks into groups or categories.

This document has the following layout Section 2 summarizes and critiques 4 papers related to this topic. The first paper makes a case for using subgraph frequencies as a statistics and the remaining three describe advanced methods for computing subgraph frequencies. Section 3 describes a proposal for computing subgraph frequencies and classifying networks based on the critique and comparison of these papers

2. Summary of Research Papers

2.1 Subgraph Frequencies: Mapping the Empirical and Extremal Geography of Large Graph Collections [Ugander et. al]

The main contributions of this paper are as follows:

- The paper demonstrates that subgraph frequencies in larger graphs are constrained by 2 properties: a) Pure Mathematical constraints which limit the occurrence of certain

structures (subgraphs) and b) Social behavior which prevent creation of certain structures, e.g a triangle with an edge missing.

- It claims (and demonstrates) that by studying the frequencies of these structures or subgraphs, it can distinguish between real and artificial networks and between different types of social networks..
- It proposes use of subgraph frequencies as a statistic to study the structure of social graphs. It creates a coordinate system based on a vector of subgraph frequencies where each coordinate encodes the relative frequency of a distinct k-node subgraph H in a larger graph G. Typical value of k is small, i.e. 3 or 4. For example, for k = 3 there are only 4 distinct subgraph types and hence the vector is 4-dimensional.
- It demonstrates the usefulness of subgraph frequencies as features for classification of networks by accurately classifying 3 types of networks on facebook, e.g. friendship neighborhoods, facebook groups and event networks. Networks with different structures (represented by different subgraph frequency vectors) point to different human behavior when faced with different settings. This allows us to predict the structure and growth of future networks.
- The measurements done in this paper use a large collection of small dense graphs induced from the real facebook networks. One Subgraph frequency vector is computed per induced network. This is done for a very large collection of smaller induced graphs of size 50 to 200 nodes.
- The results of this paper consistently show that subgraph frequencies of real world networks lie very close to a one-dimensional band and this band becomes narrower as the size of the network grows. The largest network size they used is 200 nodes.
- This result naturally begs the characterization of a very large network as one unit. We expect subgraph frequency vector to be a finer grained and accurate summary of the network compared to other global properties like degree distribution and clustering coefficients.

2.2 Counting Arbitrary Subgraphs in Data Streams [Kane et. al]

This paper presents efficient way of counting the occurrences of an arbitrary subgraph H of of a large graph G. This is particularly efficient when the number of edges in H are small (as compared to a clique formed by vertices of H).

The main idea here is to build an unbiased estimator which approximates the occurrence of H in G. The error is reduced by running a very large number of parallel instances of this estimator.

The estimator itself is build on the idea of counting all subgraphs (not necessarily induced) of G which are isomorphic to H. It starts with all possible k-tuples of edges in G (where k is the number of edges in H) and tries to match each of them with the edges of H. The estimator itself is composed of 3 types of random variables which in conjunction are used for testing whether a subgraph G' of G (induced by some k-tuple of edges in G) is isomorphic to H. The first type of

random variable X_c uses a random deg(c)th root of unity for a node c in H (there are as many instances of this random variable as nodes in H). Product of the expectations of these random variables is taken. Expectation of X_c is 1 only if there is a node w in subgraph G' such that all edges of c map onto the edges of w. The product ensures that every node c in H uniquely maps to a node w in G' and its edges map to the edges of w. This ensures that H is homomorphic to G'. The next 2 types of random variables ensure that the number of nodes of G' are same as that of H. Again there are several instances of each of these 2 types (as many as the number of vertices in H) and the product of their expectation is non-zero only if |G'(V)| = |H'(V)|. Please refer to the paper for more details

The paper then uses Chebyshev's Inequality to bound the error by using several instances of the estimator.

The main result of the paper is that it takes $O(\frac{m^k \cdot (\Delta G)^k}{\varepsilon^2 \cdot (\#H)^2})$ instances of the estimator for the error to be $(1 \pm \varepsilon)$ - bounded with probability greater than ⅔. Here m is the number of edges in the large graph G, k is the number of edges in template H, #H is the number of occurrences of H in G and ε is the required error

The benefit of this scheme is that all the estimators can be run completely in parallel and they require no communication (other than adding the results from all of them). The disadvantage is that it is not very efficient for large k. It is quite efficient for small values of k and can be used for finding 3-node or 4-node subgraphs with 4 or fewer edges. The efficiency comes from the fact that $\#H \gg m$ in large real networks. For larger values of k, e.g. for a 4-clique, we need to use a different method of computing frequency. This leads us to the scheme described in the second paper.

2.3 Clique counting in MapReduce: theory and experiments [Finocchi et al]

This paper uses a map-reduce scheme for counting cliques which is highly efficient taking only $O(m^{k/2})$ run-time complexity and $O(m)$ memory.

It uses the idea of total ordering of graph nodes by node-degree, i.e. the number of edges incident to a node in the graph. A node x $\prec$ node y if either degree(x) < degree(y) or x has a lower label than y. Each node u is then responsible for counting cliques among all its neighbors v such that $u \prec v$.

The algorithm has 3 steps.

1. It first computes high-neighborhood of each node u which is defined as all the nodes v such that $u \prec v$.

2. Next it finds all edges between nodes in the high-neighborhood of each node u.
3. Finally it counts k-1 cliques within the high neighborhood of each node u (adding node u to each of the k-1 cliques results in k-cliques). Each of these steps require a map-reduce phase.

The ordering by node-degree guarantees that each high-neighborhood has atmost $O(\sqrt{m})$ nodes. Finding k-1 cliques among $\sqrt{m}$ nodes sequentially takes $O(m^{k-1/2})$ steps. And since there are at most $\sqrt{m}$ high-neighborhoods with more $\sqrt{m}$ Nodes, the total complexity of the algorithm is $O(m^{k/2})$.

The algorithm can further be combined with various flavors of sampling to make it more efficient

The results shown on large real world graphs show significant speedup compared to the rest of the state of art mechanism. Given the run-time complexity of $O(m^{k/2})$, this method seems much more efficient compared to the first method (based on using multiple instance of an unbiased estimator). The downside of this scheme is that it only counts cliques, i.e. it cannot count other types of subgraphs which are smaller than cliques in terms of number of edges.

2.4 Fast Approximate Subgraph Counting and Enumeration [Slota & Madduri]

This paper proposes the last scheme for efficient counting of subgraph frequencies for subgraphs which are trees.

It uses a k-coloring mechanism for finding subgraphs of size k (number of vertices). It uses dynamic programming by partitioning the subgraph H, also known as template graph, into sub-templates recursively down to a single node. Frequencies are computed in the reverse order of partitioning of the templates. The idea is to find occurrences of a parent subtemplate by combining occurrences of child sub-templates such that they together form a C-colorset (i.e. a unique coloring of each of the C vertices in the parent template)

The paper presents a scheme for parallelizing the above computation using multi-threading and shared memory to achieve further speedup. This algorithm has O(m) complexity and is very efficient for computing specific type of subgraphs, i.e. trees.

The disadvantage here again is the limited types of subgraphs that can be counted by this scheme. Also this scheme might benefit significantly from a map-reduce implementation.

3. Project Proposal

Based on our stated goal in the introduction and the study of the 4 papers above, we make the following proposal.

- We will implement a combination of subgraph counting schemes from the 3 papers presented above as they are specialized in counting different types of graphs. The first scheme using unbiased estimator may suffice for counting all types of 3-node subgraphs while it is not efficient for counting very dense 4-node subgraphs. We will use scheme 2 for counting 4-cliques and a variant of scheme 3 which allows subgraphs which are trees with triangles.
- We will use the above implementation for computing subgraph frequency statistics for a large number of artificial networks and real world networks found in the SNAP database
- We will use the frequency statistics in classification of these networks as the frequency statistics allow us to distinguish their underlying structure.

4. Dataset

- We will use artificially generated graphs, e.g small-world graph as well as Kronecker Graphs.
- We will use several real networks from the SNAP database

5. References

[1] Subgraph Frequencies: Mapping the Empirical and Extremal Geography of Large Graph Collections [Ugander et. al]
[2] Counting Arbitrary Subgraphs in Data Streams [Kane et. al]
[3] Clique counting in MapReduce: theory and experiments [Finocchi et al]
[4] Fast Approximate Subgraph Counting and Enumeration [Slota & Madduri]