

Graph Theoretical Analysis for Small Components of Complex Network : A Survey

Jian Li 042021151

June 19, 2005

Fudan University, Department of Computer Science and Engineering
Shanghai 200433, China

Abstract

Recently, complex networks are hot topics among many research area, such as in the domain of bioinformatics, there are many such as protein networks, gene networks, neuro networks. Some other examples are ecological network, social network and world wide web. Because they play some certain functional role, their topological structures are different from each other for distinct networks, for instance, the degrees of some networks behave differently, some subgraph occurs frequently etc. So using the topological structure discovery methodology, say graph theory approach, to study the complex networks could be effective and significant useful way to discover the feature of the network. The papers surveys many important but not all literatures and results of studies of complex network by graph theory.

Keywords: complex network, graph theory, network motif, subgraph discovery, random graph

1 Introduction

The real world is so complex that it is hard to understand every aspect. So the study of the world need us to carefully collect the data and properly represent the object we study. Integrating the information collected about the world requires breaking the studied systems into comprehensible small parts as well as know how these parts interact with each other. In many cases, the mutual relationships between the components are best described as complex networks where the small component are described as the node and interrelation as edges. This network modelling offer us a new way to categorize systems of very different types.

The study of networks pervades all of science, from neurobiology to statistical physics. The most basic issues are structural: how does one characterize the wiring diagram of a food web or the Internet or the metabolic network of the bacterium *Escherichia coli*? Are there any unifying principles underlying their topology? Biological, social and world wide web network are recently been shown to have statistical features, such as "small world" property[9] that short paths between any two nodes, namely the small perimeter of the network, local interaction property, scale free[8] property and so on. Graph theory is a powerful tool for discovering these surprising features. Comparing with such young area such as bioinformatics, graph theory is a relative maturing subject. Since Euler introduce the graph theory to the mathematical world, there undergoes a long and splendid way in the area. Modern graph theory researches many interesting problem such as random graph, random walk on graphs and extremal graph theory which are stimulated by both the theoretical interest and the practical usage[12].

The study of networks pervades all of science, from neurobiology to statistical physics to bioinformatics. The most basic issues are structural: how does one characterize the wiring diagram of a food web or the Internet or the metabolic network of the bacterium *Escherichia coli*? Are there any unifying principles underlying their topology? The graph theoretical work focus on the topological feature of the network, such as degree distribution, perimeter, subgraphs and difference with random graph or artificial randomized graph. Some of the work will be list here and survey with some details to some extent.

2 Analysis Complex Network

2.1 Counting Subgraph in Large Network

Generally speaking, testing whether a graph is a subgraph of another large graph is NP-complete[14], so counting subgraphs is #P-complete[1]. But for some particular subgraph, there is some efficient way to counting them. And for some #P-complete problem, many techniques are designed to estimate the number of the subgraph approximately. Here we will list some important ones.

1. counting spanning trees[2].

The spanning tree is a maximal acyclic subgraph for a connected graph, that is a tree containing all the vertices of the graph. The following theorem which is known as *the matrix tree theorem*, gives the way for counting the spanning trees in a directed graph. For undirected graph the theorem is similar.

Definition 1 *Laplacian:*The laplacian of the graph is a matrix defined by $L=D-A$,where D is the diagonal matrix of which the nonzero entry is the degree of the node and A is the adjacency matrix.

Theorem 2 Suppose L is the laplacian of the graph G , L_0 is the matrix that is obtained by deleting the first row and column of L , then $\text{Det}(L_0)=\text{number of oriented spanning tree rooted at } V_0$,where $\text{Det}(L_0)$ is the determinant of L_0 .

2.counting given length cycles[3]. There are many result about counting cycles in a given graph. I will list a few important one.

Theorem 3 [4] Let $G = (V, E)$ be a directed or undirected graph, let $v \in V$ and let $v \geq 3$. A G_k that passes through v , if one exists, can be found in $O(E)$ time.

Theorem 4 [3] Deciding whether a directed or undirected graph $G = (V, E)$ contains simple cycles of length exactly $2k - 1$ and of length exactly $2k$, and finding such cycles if it does, can be done in $O(E^{2-1/k})$ time.

Note that the number k in the above two theorem is a fixed number and not a part of input. In fact, if the k is a input, the decision version of the problem is NP-complete, imply there is quit unlikely that there is some polynomial algorithm. Many other result can be found in [3].

3.approximate the number of perfect matching in bipartite graph[6].

A bipartite graph is a graph which vertices can be divided into two classes, each class contain no induced edge. A perfect match is a set of disjoint edges and every vertex are exactly one endpoint of some edge in this set. It has been proved that given a bipartite graph, to count how many perfect matches it contains is #P-complete, and it is equivalent to calculating a permanent of a matrix[1, 6].

Definition 5 the permanent of an $n \times n$ matrix $A = (a_{i,j})$ is

$$\text{Perm}(A) = \sum_{\pi \in S_n} \prod_{i=1}^n a_{i,\pi(i)}$$

where $S_n = \{\pi : \pi \text{ is a permutation of } \{1, 2, \dots, n\}\}$.

To approximating the permanent, several important technique are introduced. First, the problem are reduced the problem of sampling uniformly at random from all perfect matching in a bipartite graph,then reduced to a easier problem that nearly uniform sampling in a aperiodic markov chain. Finally by canonical path augmentation argument, the transition of the markov chain are carefully defined and by relate the probability mix property of the markov chain with the second eigenvalue of the transition matrix and conductance, the final approximating counting problem is settled.

Here, before we state the main result, we carefully define what is "approximate counting".

Definition 6 An (ϵ, δ) -FPRAS for a counting problem is a fully polynomial randomized approximation scheme that computes an ϵ -approximation with probability at least $1 - \delta$ in time polynomial in $n, 1/\epsilon, \log(1/\delta)$.

Theorem 7 *There exist an (ϵ, δ) -FPRAS for the problem of estimating the number of perfect matching in a bipartite graph of minimum degree at least $n/2$. where n is the number of nodes on each side of the bipartition.*

The details are interesting but complicated, interested reader can refer [6] for future reading.

4.approximatly counting Hamilton cycles in dense graphs[5].

Hamilton cycle is a simple cycle(i.e. a closed non-self-intersected path) that visited every vertex in the graph. Counting Hamilton cycles in a graph is also a #P-complete problem. To deal with the problem, the technique developed so far is also the randomized approximation algorithm which we discussed above. By sampling a reasonable part of the entire space we can estimate the number we want to obtain in the entire space. Although for different problem, the details is quit different. For counting Hamilton cycles, [5] makes use of a structure called 2-factor¹, and use it as an upper bound of the number of Hamilton cycles. And by carefully estimating the number of 2-factor, and dedicated analysis of the relation between the number of 2-factor and the number of Hamilton Path,they finally get the following strong result.

Theorem 8 *There exist an (ϵ, δ) -FPRAS for the problem of estimating the number of Hamilton cycles in dense graph of minimum degree at least $n/2$. where n is the number of nodes in the graph.*

2.2 Frequent Subgraph Mining

In data mining community, to discover frequent pattern from large transaction database draws much attention. If we model relations as a graph then the problem of finding frequent patterns then becomes that of discovering subgraphs which occur frequently enough over the entire set of graphs.

This work is quit different from counting some given subgraphs in a specified network. It work a set of graphs, and the task is to find a subgraph that appears more in the set, but not in one given graph. A number of works has been done, here we give a demonstrative one[31].

In [31],they give a algorithm,named FSG, for finding all connected subgraphs that appear frequently in a large graph database. Their algorithm finds frequent subgraphs using a level-by-level expansion. The key features of FSG are the following: (1) it uses a sparse graph representation which minimizes both storage and computation, (2) it increases the size of frequent subgraphs by adding one edge at a time, allowing to generate the candidates eciently, (3) it uses simple algorithms of canonical labeling and graph isomorphism which work eciently for small graphs, and (4) it incorporates various optimizations for candidate generation and counting which allow it to scale to large graph databases. We briefly sketch the algorithm as following.

First we specify the notation we will use: D : A dataset of graph transactions. t : A transaction of a graph in D , k -(sub)graph: A (sub)graph with k edges. g^k : A k -subgraph.

¹a factor of a graph is a spanning subgraphs: a subgraph whose vertex set is the whole graph. If every vertex of a factor has degree r , we call it r -factor

C^k : A set of candidates with k edges. F^k : A set of frequent k -subgraphs. $cl(g^k)$: A canonical label of a k -graph g^k .

The algorithm use adjacency-list representation to store the graph, and use canonical labeling to store the frequent item so as to easy test the isomorphism.

Algorithm 1 $fsg(D, \sigma)$ (Frequent Subgraph)

```

1:  $F^1 \leftarrow$  detect all frequent 1-subgraphs in  $D$ 
2:  $F^2 \leftarrow$  detect all frequent 2-subgraphs in  $D$ 
3:  $k \leftarrow 3$ 
4: while  $F^{k-1} = \emptyset$  do
5:    $C^k \leftarrow fsg - gen(F^{k-1})$ 
6:   for each candidate  $g^k \in C^k$  do
7:      $g^k.count \leftarrow 0$ 
8:     for each transaction  $t \in D$  do
9:       if candidate  $g^k$  is included in transaction  $t$  then
10:         $g^k.count \leftarrow g^k.count + 1$ 
11:    $F^k \leftarrow \{g^k \in C^k | g^k.count \geq \sigma |D|\}$ 
12:    $k = k + 1$ 
13: return  $F_1, F_2, \dots, F_k$ 

```

The following algorithm is used for candidate generation. For each pair of frequent subgraphs that share the same core (ie. a maximal part which all these subgraph share), they use fsg-join algorithm to generate all possible candidates of size $k+1$. For each candidate, the algorithm first check if it is already in C^{k+1} . If not, it verify if all its k -subgraphs are frequent.

Algorithm 2 $fsg-gen(F^k)$ (Candidate Generation)

```

1:  $C^{k+1} = \emptyset$ 
2: for each pair of  $g_i^k, g_j^k \in F^k, i < j$  such that  $cl(g_i^k) \leq cl(g_j^k)$  do
3:   for each edge  $e \in g_i^k$  do {create a  $(k-1)$ -subgraph of  $g_i^k$  by removing an edge  $e$ }
4:      $g_i^{k-1} \leftarrow g_i^k - e$ 
5:     if  $g_i^{k-1}$  is included in  $g_j^k$  then { $g_i^k$  and  $g_j^k$  share the same core}
6:        $T^{k+1} \leftarrow fsg - join(g_i^k, g_j^k)$ 
7:       for each  $g_j^{k+1} \in T^{k+1}$  do
8:         {test if the downward closure property holds for  $g_j^{k+1}$  }
9:         flag  $\leftarrow$  true
10:        for each edge  $f_i \in g_j^{k+1}$  do
11:           $h_i^k \leftarrow g_j^{k+1} - f_i$ 
12:          if  $h_i^k$  is connected and  $h_i^k \notin F^k$  then
13:            flag  $\leftarrow$  false
14:            break
15:          if flag = true then
16:             $C^{k+1} \leftarrow C^{k+1} \cup \{g_j^{k+1}\}$ 
17: return  $C^{k+1}$ 

```

Algorithm 3 join two k node subgraph share the same core to one $k+1$ node subgraph.

Algorithm 3 fsg-join(g_1^k, g_2^k, h^{k-1}) (Join)

```

1:  $M \leftarrow$  detect all automorphisms of  $h_{k-1}$ 
2: {determine an edge  $e_1 \in g_1^k$  that does not appear in  $h^{k-1}$ }
3:  $e_1 \leftarrow NULL$ 
4: for each edge  $e_i \in g_1^k$  do
5:   if  $e_i \notin h^{k-1}$  then
6:      $e_1 \leftarrow e_i$ 
7:     break
8: {determine an edge  $e_2 \in g_2^k$  that does not appear in  $h^{k-1}$ }
9:  $e_2 \leftarrow NULL$ 
10: for each edge  $e_i \in g_2^k$  do
11:   if  $e_i \notin h^{k-1}$  then
12:      $e_2 \leftarrow e_i$ 
13:     break
14:  $G \leftarrow$  generate all possible graphs of size  $k+1$  from  $g_1^k$  and  $g_2^k$ 

```

Although above algorithm doesn't settle the frequent subgraph mining problem with a analyzed low complexity, it performs good in experiment[31].

2.3 Some Surprising Property of Real Network

2.3.1 small world property

Taking a connected graph or network with a high graph diameter and adding a very small number of edges randomly, the diameter tends to drop drastically. This is known as the small world phenomenon. It is sometimes also know as "six degrees of separation" since, in the social network of the world, any person turns out to be linked to any other person by roughly six connections. Short-term memory uses small world networks between neurons to remember this sentence.

In modern mathematics, the center of the network of coauthorship is considered to be P. Erdős, resulting in the so-called Erdős number. [25]

3 Theoretical Model:Random Networks and Randomized Networks

3.1 Random networks

Random network have been studies as models of complex systems. The classic random network model are given by Erdős in 1940s and 1950s, and he did many initial research on this model. His initial motivation is to use probabilistic method to demonstrate the existence of graphs with seemingly contradictory properties but without to construct them explicitly. But soon it is found useful in many theoretical and practical area and it became a nature model for the real network, so it incur a systematic study by many research communities. For a more deep understanding of random network theoretically, you can refer to [13, 12].

We first give some definitions of classic random network model. Two closely related space stand out: $\mathcal{G}(n, M), \mathcal{G}(n, p)$. We note that a graph with n node can have at most $N = (n - 1)n/2$ edges and 2^N subgraphs.

for $0 \leq M \leq N$, the space $\mathcal{G}(n, M)$ consists of all n -node subgraphs with M edges, and we turn $\mathcal{G}(n, M)$ into a probability space by taking its elements to be equiprobable.

The space $\mathcal{G}(n, p)$ is defined for $0 \leq p \leq 1$. To get a element of this space, we select the edges independently with probability p .

3.2 subgraphs in random networks

The property of random graphs has been studied so abundantly that it is even impossible to survey all the important results. Here we give out some introductory results and some results that are close related to analysis the real complex networks.

Theorem 9 [12] *1 ≤ h ≤ k be fixed natural number, then in $\mathcal{G}(n, p)$ almost everywhere G_p is such that for every sequence of k vertices x_1, x_2, \dots, x_k , there exists a vertex x such that $xx_i \in E(G_p)$ if $1 \leq i \leq h$ and $xx_i \in E(G_p)$ if $h < i \leq k$.*

The theorem seems sophisticated but it will have some immediate consequence that would be great useful in analysis networks.

1. For a fixed integer k , almost everywhere graph $G_{n,p}$ has minimal degree at least k . That is to see, if a real network are modeled as $\mathcal{G}(n, p)$, we can see that it is quit likely that every node have a required connectivity. (note that p should be related with n)

2. Almost every graph $G_{n,p}$ has diameter² 2. It means that it has quit short path from one node of the network to another node of the network for some p which is related to n . This coincides with the result of the small world property of the real network system which we will discuss later.

One of the most important work in analyzing networks is to extract some subnetwork from the whole network. But unfortunately, the subgraph detecting problem is well known to be NP-Complete[14], which means to find a given subgraph in the whole graph is very expensive in the sense of running time of any algorithm in general case. But for some particular network model it maybe become feasible.

The following theorem, proved by Erdős and Rényi and appears in [15], says that whether a random graph is likely to have a *balanced graph*³ as a subgraph.

Theorem 10 [15] *Let $k \geq 2, k - 1 \leq l \leq k(k - 1)/2$, and let $F = G(k, l)$ be a balanced graph with k vertice and l edges, if $pn^{k/l} \rightarrow 0$ then almost no $G_{n,p}$ contains F , and if $pn^{k/l} \rightarrow \infty$ then almost every $G_{n,p}$ contains F .*

The clique is a complete graph. Find a clique is somehow a center question in the topic of finding a specific subgraph which is also NP-complete[14]. For one reason it is a special case of general subgraph finding problem, for another reason, every subgraph

²In graph theory, diameter of a graph is defined as the length of longest path which is chosen among shortest paths between every pair of nodes in the network

³We call a graph balanced if not subgraph of it has strictly larger average degree, that means the connection of the graph are quit uniform that no where is very dense and very sparse

finding problem can be reduced to the problem of finding a clique by a simple polynomial reduction, the proof can be find in Appendix A.

The following theorem says that some thing about the clique number, namely the size of the largest induced clique, in the random network model. In practice, such as in WWW, a subgraph a clique is a highly interconnected component, so it is reasonable to guess that it is in fact a local network for a community.

Theorem 11 [11] *Let $0 < p < 1$ be fixed, the clique number of almost every $G_{n,p}$ is d or $d + 1$, where $d = 2\log_b n + O(\log\log n)$*

There are so many result that consider the subgraph appearance in the random graph, such as a hamitonian cycle, a dense subgraph, or a spanner with some specified connectivity[13]. Following is a rough but more general result consider the number of appearance of a specified subgraph.

Theorem 12 [8] *The average number of appearance G of a subgraph with n nodes and g edges in a directed network of N node is*

$$\langle G \rangle = \lambda \binom{N}{n} p^g (1-p)^{n(n-1)-g} \sim \lambda N^n \left(\frac{\langle K \rangle}{N}\right)^g$$

where λ is a term of order 1 which is dependent on the subgraph and $\langle K \rangle = Np$.

3.3 scale-free network

Erdős networks exhibit a Possionian degree distribution, this means that the distribution of the degree of the node is so that nodes with a much higher degree is rare. Many naturally occurring network obey a long-tail degree sequence, so sometimes it is reasonable to describe it as a power law, $Prob(\text{a node has degree } k) \sim k^{-\gamma}$, where γ often between 2 and 3. These networks are termed scale-free network, with the property that some nodes with high degree-called hubs-usually appear.

scale-free network behaves quit differently from the Erdős model. Here we give a general result consider the number of subgraphs in scale-free network. Considering the power exponent of the degree distribution, γ , as a control parameter, we show that random networks exhibit transitions between three regimes. In each regime, the subgraph number of appearances follows a different scaling law.

Theorem 13 [8]

The subgraph number of subgraph appearances follows a different scaling law $\langle G \rangle \sim N^a$, where $a = n - g + s - 1$ for $\gamma < 2$, $a = n - g + s + 1 - \gamma$ for $2 \leq \gamma \leq \gamma_c$, and $a = n - g$ for $\gamma > \gamma_c$, where s is the maximal outdegree in the subgraph, and $\gamma_c = s + 1$.

In fact, certain subgraphs appear much more frequently than in Erdős networks. These results are in very good agreement with numerical simulations[8]. This has implications for detecting network motifs, which we will discuss later, subgraphs that occur in natural networks significantly more than in their randomized counterparts.

3.4 Randomized Networks

the generation of randomized graphs is intensively used for simulations of various kinds. Many recent studies however gave evidence of the fact that most real-world network have several properties in common which make them very different from random graphs. So randomized network appears, it can be defined as some a space of network which share some property, and pick the element from the space with some probability.

Many models have been introduced to capture this feature. In particular, the Molloy and Reed model[30], generates a random graph with prescribed degree sequence in linear time. However, their model produces graphs that are neither simple nor connected. To bypass this problem, one generally simply removes multiple edges and loops, and then keeps only the largest connected component.

In [29]given a degree sequence, we want to generate a random simple connected graph having exactly this degree sequence. Although it has been widely investigated, it is still an open problem to directly generate such a random graph, or even to enumerate them in polynomial time, even without the connectivity requirement[28].

Instead generating the network with prescribed degree sequence, there are many other models that will generate many other network with different properties.

4 Some possible functional building blocks:Network Motifs

4.1 Motivation and Definition

There are many types of networks in the world – computer webs like the Internet, connections among components in electronics, relationships among friends and acquaintances, transportation grids, food relationships among animals, connections among neurons, and interactions among genes. it is possible to categorize networks by looking at certain recurring circuits, or motifs, within the networks. There are some small, local, wiring patterns that occur throughout the network and play some important functional role in the entire system. This is probably because they are functional units important to whatever function the network was designed or evolved to perform. Identifying and examining these simple small patterns can help explain how networks function. Ron Milo, Nadav, Kashi-tan, Shalev Itzkovitz ,Shai Shen-Or and Uri Alon at the Weizmann Institute of Science, and Dmitry Chklovskiie at Cold Spring Harbor Laboratory published their research about finding such simple recurring simple building blocks in the October 25, 2002 issue of the journal Science.They call them Network Motifs[16].

Formally,a n-node network motifs are the n-node recurring subgraphs which occur more often far more than that in randomized networks, where the randomized networks satisfy that the degree sequences and n-1 node subgraphs appearance are the same as the real network we want to detect.

In fact, such network motifs has already been found to play important function in the real world.In the case of biological regulation networks, it has been suggested that network motifs play key information processing roles[19]. In the transcription network of bacteria

and yeast, three network motifs are found. One of these, the feed-forward loop (FFL), has been shown theoretically to perform information processing tasks such as sign-sensitive filtering, response acceleration and pulse-generation[20]. Some other motifs are found such as Autoregulatory motif, Multi-component loop, Single input module, Multi-input module.

4.2 Specific Algorithm Used for discover some specific motifs

Assume the we have already have the topological structure of simple building and we want to test for specific network. For different motif structure we can design different discover strategy. One example algorithm used for analyzing Hepatocyte and Pancreatic Islet Gene Expression are sketch below[18].

In order to discover network motifs, two data matrices were created. The overall matrix D consists of binary entries D_{ij} , where a 1 indicates binding of regulator j to intergenic region i , a 0 indicates no binding. The regulator matrix R is a subset of D , containing only the rows corresponding to the intergenic region assigned to each regulator, in the same order as the columns of regulators.

The algorithms used to find each motif are described below.

(1) Autoregulatory motif: Find each non-zero entry on the diagonal of R . (2) Feedforward loop: For each master regulator (column of R), find non-zero entries, which correspond to regulators bound. For each master regulator/secondary regulator pair, find all rows in D bound by both regulators. (3) Multi-component loop: For each regulator (column of R), find the regulators to which it binds. For each of these, find the regulators it binds. If any of these are the original regulator, you have a multi-component loop of two. For all others, find regulators to which they bind. If any of these are the original, you have a multi-component loop of three. Repeat to find larger loops. (4) Single input module: Find the intergenic regions bound by only one regulator. That is, take the subset of rows of D such that the sum of each row is 1. Then for each regulator (column), find non-zero entries. Each set (greater than three promoter regions) is a SIM. (5) Multi-input module: Find the intergenic regions bound by more than one regulator. That is, take the subset of rows of D such that the sum of each row is greater than 1. Then, for each row, find any other row bound by the same regulators. The collection of rows bound by the same regulators correspond to a MIM. Once a row is assigned to a MIM, remove it from further analysis. (6) Regulator cascade: For each regulator (column of R), use a recursive algorithm to find chains of all lengths. That is, for each regulator whose promoter is bound by the regulator before it in the chain, find the regulator promoters to which it binds. Repeat until the chain ends. There are three possible ways to end a chain: a regulator that does not bind to the promoter of any other regulator, a regulator that binds to its own promoter, or one that binds to the promoter of another regulator earlier in the chain.

4.3 Generic Algorithm I : Counting

First, for the purpose of comparing, we should construct an ensemble of randomized networks which share the same degree sequence and $n-1$ node subgraph appearance with the real networks given.

4.3.1 Prescribed Randomized Network used for comparing

We employed a Markov-chain algorithm[17], based on starting with the real network and repeatedly swapping randomly chosen pairs of connections ($X_1 \rightarrow Y_1, X_2 \rightarrow Y_2$ is replaced by $X_1 \rightarrow Y_2, X_2 \rightarrow Y_1$) until the network is well randomized. Switching is prohibited if the either of the connections $X_1 \rightarrow Y_2$ or $X_2 \rightarrow Y_1$ already exist.

We generate a series of randomized network ensembles, each of which has the same (n-1)-node subgraph count as the real network, as a null hypothesis for detecting n-node motifs. This is done to avoid assigning high significance to a structure only because of the fact that it includes a highly significant substructure.

So The remain work is to Control for Appearances of (n-1)-Node subgraphs. Metropolis Monte-Carlo approach are used[27, 17]. $V_{real,k}$ be the number of appearances of each of the k-th (n-1)-node subgraphs in the real network and $V_{rand,k}$ be the corresponding vector in the randomized network. We define an energy

$$E = k(|V_{real,k} - V_{rand,k}|/(V_{real,k} + V_{rand,k})).$$

The energy E is zero only when all the three-node subgraph counts of the real and randomized graphs are equal.

we generate a random switch ($X_1 \rightarrow Y_1, X_2 \rightarrow Y_2$ to $X_1 \rightarrow Y_2, X_2 \rightarrow Y_1$), and similarly for double edges, as described above). If this switch lowers E , it is accepted. Otherwise, it is accepted with probability $exp(-\delta E/T)$, where δE is the difference in energy before and after the switch and T is an effective temperature.

This process is repeated, with a simulated annealing regiment to lower T slowly until a solution with $E = 0$ is obtained. This can be readily generalized to form (n-1)-node null-hypothesis networks

4.3.2 Algorithm for Counting

Do brute-force search algorithm for both real network and randomized network Simply enumerate all the possible n node subgraphs, classify them into non-isomorphic class. Count the number of subgraphs in each class.

A table is formed that counts the number of appearances of each type of subgraph in the network, This process is repeated for each of the randomized networks. The number of appearances of each type of subgraph in the random ensemble is recorded, to assess its statistical significance.

4.3.3 Criteria for Network Motif Selection

if the appearance of a certain type of subgraph satisfy three condition below, we claim it as a network motif.

- (i) The probability that it appears in a randomized network an equal or greater number of times than in the real network is smaller than $P = 0.01$.
- (ii) The number of times it appears in the real network with distinct sets of nodes is at least 4.
- (iii)The number of appearances in the real network is significantly larger than in the randomized networks: $N_{real} - N_{rand} > 0.1N_{rand}$. This is done to avoid detecting as motifs

some common subgraphs that have only a slight difference between N_{rand} and N_{real} but have a narrow distribution in the randomized networks.

4.4 Generic Algorithm II : Sampling

The counting algorithm can exactly enumerate the number of subgraph, but to detect network motifs, we only need to know which type of subgraph occur more frequently in real network than in randomized network. So [21] gives a clever trade-off between accuracy and efficiency, say using sampling method. Surprisingly, the complexity of their algorithm doesn't depend on the size of the network, but their randomized sampling converges the counting method and fortunately it performs very well in practice. The sampling method is useful for analyzing very large networks or for detection of high-order motifs, which are beyond the reach of exhaustive enumeration algorithms.

In fact, randomize sampling is not a new thing but has a longer history and rich application to many areas. Such as in machine learning community, sampling are used for learning input character distribution. In theoretical computer science community, randomized sampling are used for approximating some NP-hard problem, see arora[22]. Some other application such as estimate the volume of a higher-dimension convex shape can be found everywhere.

The method are sketched below.

Definition: E_s is the set of picked edges, V_s is the set of all node that are touch be the edges in E_s .

ALGORITHM Sampling:

Initiate $V_s = \emptyset$ and $E_s = \emptyset$

1. Pick a random edge $e_1 = (v_i, v_j)$, update $E_s = \{e_1\}$, $V_s = \{v_i, v_j\}$
2. Make a list L of all neighboring edges of E_s , omit all edges between V_s .
if $L = \emptyset$ return to Step 1.
3. pick a random edge $e = (v_k, v_l)$ from L .
Update $E_s = E_s \cup \{e\}$, $V_s = V_s \cup \{v_k, v_l\}$
4. Repeat steps 2-3 until completing n-node subgraph S .
5. Calculate the probability P to sample S .

The probability of sampling the subgraph is the sum of the probabilities of all such possible ordered sets of n-1 edges:

$$P = \sum_{\sigma \in S_m} \prod_{e_j \in \sigma} Pr[E_j = e_j | E_1, E_2, \dots, E_{j-1} = e_1, e_2, \dots, e_{j-1}]$$

where Where S_m is a set of all (n-1)-permutations of the edges from the specific subgraph edges that could lead to a sample of the subgraph. E_j is the j-th edge in a specific (n-1)-permutation (σ).

Add score $W = 1/P$ to the accumulated score, S_i , of the relevant subgraph type i: $S_i = S_i + W$. After S_T samples, assuming we sampled L different subgraph types, we calculate the estimated subgraph concentrations

$$C_i = \frac{S_i}{\sum_{k=1}^L S_k}$$

. Finally we calculate Z – scores:

$$Z = (C_{real} - \langle C_{rand} \rangle) / \text{Var}_{rand}$$

where C_{real} is the concentration in the real network, $\langle C_{rand} \rangle$ and Var_{rand} are the mean and SD in the randomized networks.

The criterion used for judging the network motifs are the same with counting algorithm.

Theorem 14 *Subgraph concentrations calculated by the sampling algorithm converges to the fully enumerated concentrations.*

The theorem means that we are trying to do the right thing although with some probabilistic uncertainty. We remark that different numbers of samples were required for achieving good estimations for different subgraphs and in different networks. All of the simulations we performed, on a variety of networks, showed that the results converge toward the real values within $ST = 10^5$ samples or less. In fact, as the authors mentioned, theoretical analysis can be use for deciding to using how many samples instead of a experience number, see [24]. This method shows surprising experimental result, see [21].

5 Conclusion

This paper surveys some recent results about using graph theory to analyze complex networks. In Particular, the random network, a natural model of some real network, are discussed and several property are listed. Randomized network, a randomization of real network, which are often used for comparing with real network and thus discovering the speciality of topological structure of the real network, are also mentioned. In order to find a functional component or a particular building block of the complex network, it reduces to find a specific subgraph in the given graph or to count the frequency of a specific subgraph. We uses most of the parts to concentrate on this important topic. Some theoretical works to detect the subgraph and count the subgraph frequency are listed, meanwhile the some data mining method are mentioned and a algorithm to count the frequency of subgraph in a ensemble of graph transactions are briefly sketched. Network motifs are defined as subnetworks that appears more frequently in real networks than in randomized networks. This paper discusses two algorithm for finding the network motifs, brute-force enumeration algorithm and random sampling algorithm, and make comparison between this two algorithms.

References

- [1] Oded Goldreich, Introduction to Complexity Theory-Lecture Notes. page 115-134, 1999
- [2] D.E.Knuth. The Art of Computer Programming, VOL 1, Fundamental Algorithms, 3rd Edition. Addison-Wesley. page 378-379. 1997

- [3] N.Alon, R.Yuster. and U.Zwick. Finding and counting given length cycles. *Algorithmica*, 17, 209C223.1997
- [4] B.Monien, How to find long paths efficiently. *Annals of Discrete Mathematics*, 25,page 239-254,1985
- [5] M.Dyer, A.M.Frieze and M.Jerrum. Approximately counting Hamilton cycles in dense graphs. *Proceedings of the 5th ACM/SIAM Symposium on Discrete Algorithms*. ACM/SIAM Press, page. 336C343.1994
- [6] R.Motwani and P.Raghavan, *Randomized Algorithms*. Cambridge Press, page 315-329.1995
- [7] R.Duke, H.Lefmann and V.Rödl. A fast approximation algorithm for computing the frequencies of subgraphs in a given graph. *SIAM J. Comput.* 1995
- [8] Itzkovitz,S.,Milo,R.,Ksahtan,N.,Ziv,G. and Alon,U. Subgraphs in random networks. *Physics Review E*, 2003.
- [9] S.H.Strogatz, *Nature*, page 410, 268 ,2001.
- [10] D.J.Watts and S.H.Strogatz, Collective dynamics of 'small-world' networks, *Nature*, page 440, 1998.
- [11] B.Bollobas and P.Erdős, Cliques in random graphs, *Mathematics Proceeding of Cambridge Philosophy Society*. 1976.
- [12] B.Bollobas, *Modern Graph Theory*, Springer-Verlag, page 216,225,252 cited.
- [13] B.Bollobás, *Random Graphs*, Academic press, London, 1985
- [14] M.R.Garey and D.S.Johnson, *Computer and Intractability-A Guide to the Theory of NP-Completeness*. W.H.Freeman,1979
- [15] R.Fagin, Probabilities on finite models, *Journal of Symbolic Logic*,1976
- [16] Milo,R., Shen-Orr,S., Itzkovitz,S., Kashtan,N., Chklovskii,D. and Alon,U., Network motifs: simple building blocks of complex networks. *Science*, page 824-827.2002.
- [17] Supplement Web material for [16],<http://www.weizmann.ac.il/mcb/UriAlon>
- [18] Odom, Control of Hepatocyte and Pancreatic Islet Gene Expression by HNF Transcription Factors, *Science* 303, page 1378-1381,2004
- [19] Shen-Orr,S., Milo,R., Mangan,S. and Alon,U. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics*, 2002
- [20] Mangan and U. Alon ,Structure and function of the feed-forward loop network motif. *Proceeding of National Academy of Sciences of USA*, 2003

- [21] N.Kashtan, S.Itkovitz, R.Milo, and U.Alon efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs, Bioinformatics, Vol.20, page 1746-1758,2004
- [22] S.Arora, David Karger and Marek Karpinski, Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problem
- [23] M.Dyer, A.Frieze and R.Kannan, A random polynomial algorithm for approximating the volume of convex bodies, Journal of the ACM,38:1-17,1991
- [24] S.Chaudhuri, R.Motwani, and V.Narassaya, Using random sampling for histogram construction:how much is enough? ,Proceeding of the ACM SIGMOD International Conference on Management of Data, Seattle, page 436-447, 1998
- [25] <http://mathworld.wolfram.com/SmallWorldNetwork.html>
- [26] http://jura.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=22&f=network-analysis
- [27] Lecture Note of Statistical Physics: Chapter 11 :Monte Carlo methods in statistical physics. available at http://www.physics.ohio-state.edu/~ntg/780/computational_physics_2003_11.pdf
- [28] M.R. Henzinger and V. King, Randomized fully dynamic graph algorithms with polylogarithmic time per operation. Journal of the ACM 46(4), page 502-516, July 1999
- [29] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman and U. Alon, Uniform generation of random graphs with arbitrary degree sequences, Physics Review. E64(2), 2001
- [30] M. Molloy and B. Reed, A critical point for random graphs with a given degree sequence, Random Structures and Algorithms, page 161-179, 1995
- [31] M.Kuramochi and G.Karypis , Frequent Subgraph Discovery, IEEE International Conference on Data Mining,2001

6 appendix A

Subgraph finding problem: Given two graphs G_1, G_2 , the question is whether G_1 is a subgraph of G_2 , that is G_1 can be obtained from G_2 by deleting some nodes and edges.

clique problem: Give a graph G , is there a clique of size k which is a subgraph of G ?

By the NP-completeness theory, every two NP-completeness problem have polynomial time reduction to each other, but what we will prove that there is simple reduction so that you can even use algorithms for solving clique problem as subroutines to solve the subgraph finding problem.

Theorem 15 *Suppose we have two graphs G_1, G_2 , if we have a oracle which can answer clique problem in $O(1)$ time, then we can answer whether G_1 is a subgraph of G_2 in at most $O(|G_1| \times |G_2|)$ time.*

Proof: Construct a new graph G' with $|G_1| \times |G_2|$ vertices, which we label them as a Cartesian product (v_1, v_2) , where $v_1 \in G_1, v_2 \in G_2$. And then we add edge to this graph. If $e(v_1, v'_1) \in G_1$ and $e(v_2, v'_2) \in G_2$, then connect two vertices $e(v_1, v_2)$ and $e(v'_1, v'_2)$. If $e(v_1, v'_1) \notin G_1$ then connect two vertices $e(v_1, v_2)$ and $e(v'_1, v'_2)$ for all $v_2, v'_2 \in G_2$ except $v_2 = v'_2$. Then we ask clique problem for graph G' that is there a clique of size $|G_1|$ in G' , we can easily proof that if answer is yes if and only if G_1 is a subgraph of G_2 . \square