

Dynamic Network Evolution: Models, Clustering, Anomaly Detection

Cemal Cagatay Bilgin and Bülent Yener
Rensselaer Polytechnic Institute, Troy NY, 12180.,

Abstract—Traditionally, research on graph theory focused on studying graphs that are static. However, almost all real networks are dynamic in nature and large in size. Quite recently, research areas for studying the topology, evolution, applications of complex evolving networks and processes occurring in them and governing them attracted attention from researchers.

In this work, we review the significant contributions in the literature on complex evolving networks; metrics used from degree distribution to spectral graph analysis, real world applications from biology to social sciences, problem domains from anomaly detection, dynamic graph clustering to community detection.

I. INTRODUCTION

The solution to the famous historical problem of Seven Bridges of Königsberg by Leonhard Euler laid the foundations of the graph theory in 1736 and regarded as the first paper in history of the graph theory [36]. Since then graph theory has established and became one of the most active areas of research.

Many real world complex systems take the form of networks, with a set of nodes or vertices and links or edges connecting a pair or more of the nodes. Such networks are found in diverse application domains such as computer science, sociology, chemistry, biology, anthropology, psychology, geography, history, engineering. Until recently, the research on network theory has mainly focused on graphs with the assumption that they remain static, ie they do not change over time. A wealth of knowledge has been developed for this type of static graph theory.

We explain some of the widely used graph metrics in the literature to study dynamic graphs, present the real dynamic networks in almost all fields of science and give a taxonomy of models to capture the evolution of these dynamic networks.

II. BACKGROUND

For completeness we start with the objects under discussion. We formally define a graph, a digraph, the four types of dynamic networks, and the graph metrics heavily used in the literature.

- A graph G is formally defined by $G = (V, E)$ where V is the finite set of vertices and E is the finite set of edges each being an unordered pair of distinct vertices.
- A digraph G is formally defined by $G = (V, A)$ where V is the finite set of vertices and A is the set of arcs each being a pair of distinct vertices.

We use interchangeably use node, vertex and edge, link, connection to refer the same concept.

Let f be a function defined on the vertex set as $f : V \rightarrow N$ and g be the function defined on the edge set as $f : E \rightarrow N$. A node weighted graph is then defined as the triple $G = (V, E, f)$ and likewise an edge weighted graph is defined as the triple $G = (V, E, g)$. A fully weighted graph is the quadruple $G = (V, E, f, g)$.

For some applications, the existence of a smaller graph in a larger graph evoke interest.

Let $G_1 = (V_1, E_1, f_1, g_1)$ and $G_2 = (V_2, E_2, f_2, g_2)$ be graphs. G_1 is called a subgraph of G_2 if the following conditions hold.

- $V_1 \subseteq V_2$
- $E_1 \subseteq E_2$
- $f_1(u) = f_2(u) \forall u \in V_1$
- $g_1(u, v) = g_2(u, v) \forall (u, v) \in E_1$

A dynamic graph can be obtained by changing any of the V, E, f or g . Harary classifies the dynamic graphs by the change of any of these [37].

- Node dynamic (di)/graphs where the vertex set V change over time.
- Edge/Arc dynamic (di)/graphs where the edge set E change over time.
- Node weighted dynamic (di)/graphs where the f function varies with time.
- Edge/Arc weighted dynamic (di)/graphs where the g function varies with time.

A. Graph Metrics

1) *Degree*: The simplest and one of most intensively studied graph metric is the degree. The degree k of a node u is defined as the total connections of that node. The in-degree k_i is the incoming edges to node u and the out-degree k_o is the outgoing edges from node u . The distributions of in-degree, out-degree and the joint degree distributions have been studied.

2) *Clustering Coefficient*: Metrics that quantify how close the graph from being a clique and whether a node's neighbors are also neighbors of each other are defined. For an undirected graph, let node u have k neighbors and among these k neighbors let y of them have links to each other. The clustering coefficient for node u is then defined as the ratio of actual number of links between node u 's neighbors and the total number of connections that could possibly exist. Defined in equation (1), clustering coefficient is was first introduced in the context of sociology [69] and then in computer science [70].

$$C = \frac{y}{k(k-1)/2} \quad (1)$$

In the original calculation of the clustering coefficient node u is not included, instead of this another related metric have also been proposed that accounts node u in the calculation as well. The actual number of links in the neighborhood is $k+y$ and the total possible links that can exist is $(k+1)k/2$, giving a value as in (2)

$$D = \frac{k+y}{(k+1)k/2}. \quad (2)$$

3) *Shortest Path*: The shortest path between two nodes in an undirected graph is defined as the geodesical distance of these nodes with unit length edges. Several metrics using the shortest path distance have been defined. The **path length** or **hop count** between two nodes is defined as their shortest path length in the graph, taking the weight of each link as a unit length. Given the shortest path lengths between a node u_i and all the reachable nodes from u_i , the **eccentricity** and the **closeness** of node u_i are defined as the maximum and the average of these shortest path lengths, respectively. The minimum eccentricity value in a graph eccentricity is referred as the **radius** and the maximum eccentricity is referred as the **diameter** of the graph. Nodes that have an eccentricity equal to the radius are defined to be **central points** of the graph.

B. Spectral Graph Analysis

Apart from the metrics defined above, graph spectra has also been used. The spectral analysis of graphs deal with the eigenvalues of the adjacency matrix or other matrices derived from the adjacency matrix [20]. Spectral graph analysis reveals important structural properties of graphs. In particular, the second smallest eigenvalue is a measure of the compactness (more precisely, algebraic connectivity) of a graph. A large second eigenvalue indicates a compact graph, whereas a small eigenvalue implies an elongated topology.

The spectral analysis can be directly carried on the adjacency matrix, or the laplacian or normalized laplacian matrix. The adjacency matrix A of a graph $G(V, E)$ where V is the vertex set and E is the edge set is defined as in (3).

$$A(u, v) = \begin{cases} 1, & \text{if } (u, v) \in E \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The degree matrix D is constructed by $D(u, u) = \sum_{v \in V} A(u, v)$. More specifically, the degree matrix is a diagonal matrix that holds the degree of node i at (i, i) entry. The Laplacian matrix L is then given by the difference between the degree matrix and the adjacency matrix, $L = D - A$. Laplacian matrix is strongly connected to the gradient of the graph, nevertheless we use the normalized version of the Laplacian matrix given as (4).

$$\mathcal{L}(G) = \begin{cases} 1, & \text{if } i = j \text{ and } d_i \neq 0 \\ -\frac{1}{\sqrt{d_i d_j}}, & \text{if } i \text{ and } j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The spectral decomposition of the normalized Laplacian matrix is given by $\mathcal{L} = \Phi \Lambda \Phi^T$ where $\Lambda = \text{diag}(\lambda_1, \lambda_1 \dots \lambda_{|V|})$ with the eigenvalues as its elements and Φ with the eigenvectors as columns. The normalized Laplacian matrix and its spectral decomposition provide insight to the structural properties of the graph. Since \mathcal{L} is a symmetric positive semi-definite matrix the eigenvalues of normalized Laplacian matrix are all between 0 and 2.

For normalized Laplacian matrices the number of zero eigenvalues gives the number of connected components in the graph. Number of zero eigenvalues, the number of eigenvalues equal to one, and the number of eigenvalues equal to two in our feature set have been used in [8], [19] as well as the slope of the line that fits the plotted eigenvalues between $[0, 1]$ and $[1, 2]$. Trace and the energy of the normalized Laplacian are defined as $\sum_i \lambda_i$ and $\sum_i \lambda_i^2$ respectively. It has been reported that the eigenvalues of the normalized Laplacian graph is more distinguishing than that of the Laplacian or the adjacency matrices [20].

III. EVOLVING NETWORKS IN NATURE

The question of whether there is a relation between the structure and the functionality has attract attention in many fields. In this section we discuss some of the most prominent real complex networks and some of the findings in these networks.

A. Citation Networks

The vertices of a citation network are scientific papers and the directed edges of the network connect a paper to another if the former cites the later. The evolution of a citation network is therefore simple: A new vertex is added to the network for a newly published paper and links between this new paper and the papers it cites are added. New edges between old vertices are not possible in citation networks.

Often times citation networks are sparse with an average out-degree of order 10^1 .

B. Collaboration Networks

Collaboration networks of movie databases such as IMDB, and scientific paper databases such as DBPL have been studied in the literature. These collaboration networks can be expressed bipartite graphs where the first set of nodes represent the collaborators and the second set of nodes represent the acts of the collaborations. Although less information one-mode projections of bipartite graphs are used to represent such networks. That is the act of collaborations are not represented in the network instead two collaborators are linked if they appeared in the same collaboration.

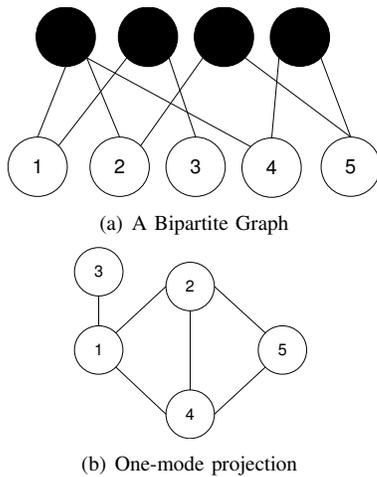


Fig. 1. In figure 1(a) a bipartite graph is given. Vertices that is the collaborators in this graph are connected to acts of the collaborations. Vertices 1, 2 and 4 collaborates together for a specific task, vertices 1,3 collaborates in another task and so on. The one-mode projection of the same network is provided in 1(b) where the collaborators are linked to each other directly.

C. Communication Networks

Several networks fall under the category of communication networks i.e. world wide web (WWW), the Internet, call-data. The nodes of the WWW graph are the homepages and the directed edges of the graph are the hyperlinks from one page to another. As of July 2008 google announced that their search engine indexed 1 trillion pages.

After it has been shown that the degree distribution of the Web obeys the power law, work on the structure of the Web graph has been fostered. As the Web is one of the largest real networks, studying this graph is a hard problem, therefore, most of the studies covered portions of the whole Web graph. Although these studies all cover different portion, they all found that both the in-degree and the out-degree distributions follow the power law with different exponents.

The clustering coefficient of the Web has also been considered to understand the structure. Adamic showed that the clustering coefficient of the Web is 0.1078 compared to 0.00023 of the random graph with the same number of edges and nodes. This finding implies that the Web is a small world network.

D. Biological Networks

Many biological networks have been introduced and the topology and evolution of such networks have interested researchers. Among these networks neural networks, metabolic reaction networks, protein protein interaction networks, human sexual contacts have been studied extensively. The following subsections will give an overview of each of these networks.

1) *Neural Networks*: The neural network of the *Caenorhabditis Elegans* worm has been considered in [70]. The nodes of the network are the neurons and the edges between these nodes are synapses. This network has an

average clustering coefficient of 0.26 compared to 0.05 of the random network with the same number of edges and vertices. The degree distribution however differs from other networks found in nature. An exponential distribution fits both to the in- and out-degree distributions.

Using functional magnetic resonance imaging to extract functional networks connecting correlated human brain sites, Eguluz et al. studied the structure of brain functional networks [27]. This network is found to follow the power law degree distribution, with high clustering coefficient and small average path lengths.

2) *Metabolic Networks*: The substrates or in general molecular compounds are the vertices and the reactions connecting these molecules are the edges in metabolic networks. Like collaboration networks, these graphs can be represented with the informative bipartite graphs but in practice the less informative one-mode projects have been used.

A substrate graph and reaction graph from the *Escherichia coli* intermediary metabolism for energy generation and small building block synthesis is considered in [68]. The authors define two graphs G_s and G_r . In G_s , two substrates are linked when they occur in the same chemical reaction and likewise in G_r two reactions are linked when there is at least one chemical compound that occurs in those reactions, either as substrate or as a product. Both frequency-degree and rank-degree distributions of the substrate graph follows a power law distribution and neither the frequency degree nor the rank degree distribution follows a simple probability function. In the substrate graph coenzyme A, 2-oxoglutarate, pyruvate and glutamine having the highest degree which were viewed as an evolutionary core of the *E.coli*.

3) *Protein Networks*: In [40] and [67] protein protein interactions (PPI) in yeast *S. cerevisiae* have been studied. The vertices of PPI networks are the proteins and the edges direct from the regulating to the regulated component. These networks might contain self loops contrary to the most of the networks considered here. Jeong et al. showed that random deletion in PPI network of yeast *S. cerevisiae* is not lethal, but targeted deletion of proteins is lethal [40]. The in-degree distribution of such networks are shown to be power law distributions by Wagner et al [67].

4) *Food Networks*: Food networks (food chains) are used by ecologist to describe the eating interactions between various species. The nodes of the graph are therefore organisms, and the directed edges connect a specie eater and its food. These networks might be either unweighted or weighted with weights being the energy transferred from the producer to the eater. Since cannibalism is possible in food networks these networks might have unit loops as well. Studies on these networks up to this date suffered from the small size of these networks. It has been suggested that the statistical analysis of these networks are hindered because of this fact [23].

Although the size of these networks are small, they appear to follow power law distributions [52], though with a small exponent compared to the other networks. Indeed, it has also been shown that an exponential function fits as good as a power law function to some food webs that power laws [16].

E. Economics

Dynamic networks have applications in economics as well. A network representation of the stock market is analysed in [10]. Nodes in this network are the instruments in the stock. A correlation function between instruments i and j is defined by equation (5)

$$C_{ij} = \frac{\langle R_i R_j \rangle - \langle R_i \rangle \langle R_j \rangle}{\sqrt{\langle R_i^2 - \langle R_i \rangle^2 \rangle \langle R_j^2 - \langle R_j \rangle^2 \rangle}} \quad (5)$$

The instruments related to each other are found by pruning the ones with high correlation. More specifically two instruments are highly correlated if $C_{ij} > 0.2$. The stock market graph has been found to follow power law degree distributions. These graphs get denser as new vertices and edges are added every day. They also show high clustering coefficients.

F. Aerospace

Dynamic graph evolution is also studied in the control theory and aerospace engineering. In [50] the importance of studying dynamic graphs in the context of a distributed space system is presented as well as a theoretical framework for the study is given. The evolution of the graph here is quite different than the other proposed and observed methods since the working assumption is the complete controllability of the dynamic system which is quite different than the rest of network presented in this survey ie citation networks, biological networks.

The evolution of the system is defined through a difference equation of the form:

$$x_i(k+1) = f_i(x_i(k), u_i(k)), \quad i = 1, \dots, n; \quad (6)$$

The topology of the information channel, represented by a graph, is then mapped from control states to a graph by $G(x(t))$.

A graph G_r is considered as a neighboring graph to G_s if there is a control sequence $u(k_0), u(k_0+1), \dots, u(k_0+p-1)$ that transforms $x(k_0) \in \chi_r$ to $x(k_0+p) \in \chi_s$ and if $x(k) \in \chi_q$ for some $k_0 \leq k \leq k_0+p$ then $q = r$ or $q = s$.

G. Other Dynamic Networks in Computer Science

Several computer science application areas for dynamic networks are studied in [37]. In logic programming languages, functional languages and data flow languages the given program is converted to a graph. This graph is changed over the time with the operations specified by the program hence representing a dynamic graph. In Article Intelligence tree searches are involved to find the optimum answer. Building and pruning trees dynamically is involved in Game tree search.

Computer networks are dynamic networks as well. The edges and the vertices fail, or new edges and vertices introduced during the life time of a network hence forming a dynamic network. Process communication networks are also dynamic, processes representing the vertices and links representing the data exchange between them.

Data structures such as B-trees are used in accessing databases. In B-trees key-value pairs are stored and given a key the corresponding value can be accessed. As new values are added to B-tree the tree changes overtime, representing the evolution of the tree.

IV. DYNAMIC NETWORK MODELS

A. Random Graphs and Variants

We will start with the mother of all network models, namely the Erdős and Renyi random graph model. The simplest and the first graph evolution model proposed is the Erdős and Renyi random graph model [28], [29]. The method starts with N vertices and randomly links two nodes with a probability of p . Such graphs with N nodes, and edge probability p form a set of graphs $G_{N,p}$. Many properties of random graphs have been studied such as degree distribution, clustering coefficient, size of the largest connected component, diameter of the graph.

The degree distribution of Erdős Renyi random graphs have been shown to follow a Poisson distribution given by equation (7) as the number of nodes in the network goes to infinity.

$$p_k = \frac{z^k (e^{-z})^k}{k!} \quad \text{with } z = p(N-1) \quad (7)$$

The probability of having exactly k neighbors is given by the binomial distribution

$$p_k = \binom{n}{k} p^k (1-p)^{(N-k)}. \quad (8)$$

For different values of p the size of the largest connected component and the degree distribution of the network attracted attention. For smaller values of p , the size of the components are relatively close to each other and have a finite mean size. These components follow an exponential degree distribution. With higher values of p a giant connected component appears in the graph with $O(N)$ vertices. The rest of the vertices in the graph follow an exponential degree distribution. Between these two states that are dependent on the choice of p , a phase transition point is found at $p = 1/N$.

The probability of two neighbors being connected to each other, referred also as the clustering coefficient, of random networks are shown to be small compared to the real networks. The clustering coefficient of a random network with N vertices is given by

$$p = \frac{\hat{k}}{N}, \quad (9)$$

where \hat{k} is the average node degree in the network. Another metric studied for these networks is the diameter of the network defined as the longest shortest path in the network. The diameter of random networks are given by equation (10)

$$\text{Diameter} = \frac{\log N}{\log \hat{k}}, \quad (10)$$

where \hat{k} is the average node degree. From equation 10, as the number of vertices in the network increases, the diameter of the network slowly increases as well.

1) *Power Law Random Graph*: Several other random networks have been proposed that change some of the properties of the initial Erdos and Renyi random networks. The most obvious change to $G_{N,p}$ is the change of the degree distribution from a poisson to a power law. Aiello et al proposed Power Law Random Graphs that randomly assigns a degree to each node to match the distribution [1]. The number of k degree nodes is given by

$$e^\alpha/k^\beta, \quad (11)$$

whereas for the initial random graph method this quantity was given by 8. This new setup for graph formation results in power law distributions for the proposed PLRG method with a powerlaw exponent of β . In the same work the authors compare the degree distribution to real call networks and show that the method is not fully capable of capturing the network. Nonetheless, they show that for different choices of the parameters α and β , the graphs have variety of properties. For

- When $\beta > \beta_0 \approx 3.48$ the graph has almost surely has no giant component. When $\beta < \beta_0 \approx 3.48$ the graph has almost surely have a giant connected component.
- When $2 < \beta < \beta_0 \approx 3.48$, the second largest components are of size $\Theta(\log n)$. For any $2 \leq x < \Theta(\log n)$, there is almost surely a component of size x .
- In the case of $\beta = 2$ the graph is almost surely connected with more complicated second largest component sizes.
- When $1 < \beta < 2$, the graph is almost surely not connected and smaller components are of size $\Theta(1)$.
- When $0 < \beta < 1$, the graph is almost surely connected.
- $\beta = \beta_0$ corresponds to the phase transition observed at $p = 1/N$ at Erdos Renyi graphs.

[71], [55] [62], [9]

B. Preferential Attachment and Variants

Traditional Erdos Renyi random graph models have position degree distributions. However, it has been found that many real life networks follow power law distributions(see section III). Generalized random graph models have been proposed to mimic the power law degree distribution of the real networks but these models do not explain how such a phenomena occurs in these graphs. Barabasi et al. [3] introduced the concept of preferential attachment for this purpose.

1) *BA Model*: In this proposed model, nodes arriving one at a time to the network. Each new arriving node then creates m edges where m is constant parameter. The edge creation is random but preferential. The probability Π that a new vertex u connecting to vertex v depends on the degree k_i of vertex v . After m node insertions the model leads to a random network with $t_i + m$ vertices and mt edges. Networks with this model evolves into a scale-invariant state with the probability that a

vertex has k edges following a power-law with an exponent $\gamma = 2.90 \pm 1$. Moreover, in this model the diameter of the network is also low. The diameter in preferential attachment model grows slowly, i.e., logarithmically with the number of nodes [48].

2) *Initial Attractiveness*: While the Barabasi Albert model have the power law degree distribution the power law exponent is constant regardless of the choice of m . Dorogovtsev et al. [24] added one more parameter to the probability calculation.

$$\Pi = \frac{A + k(v)}{\sum_i (A + k(i))} \quad (12)$$

The degree distribution of the network is given by (13)

$$\gamma = 2 + \frac{A}{m}. \quad (13)$$

3) *Edge Copying Methods*: Another class of network models is the edge copying models that take into account the fact a new node i.e., webpage will most likely be familiar to the topics of interest to it and therefore will have some of its new edges from the existing similar webpages. Several edge copying methods have been introduced using this principle [42], [44], [47].

a) *Edge Copying*: The model introduced by Kleinberg et al consists of three steps

- Node creation and deletion In each iteration, nodes may be independently created and deleted under some probability distribution. All edges incident on the deleted nodes are also removed.
- Edge creation In each iteration, we choose some node v and some number of edges k to add to node v . With probability β , these k edges are linked to nodes chosen uniformly and independently at random. With probability $1 - \beta$, edges are copied from another node: we choose a node u at random, choose k of its edges (u, w) , and create edges (v, w) . If the chosen node u does not have enough edges, all its edges are copied and the remaining edges are copied from another randomly chosen node.
- Edge deletion Random edges can be picked and deleted according to some probability distribution.

b) *Community Guided*: After the observation of power law degree distributions several methods have been proposed to model real life network scenarios. Preferential attachment is one of the most commonly used technique in most applications. Leskovec et al [47] shows that there are missing understandings of real life networks and using existing graph generation techniques therefore do not fill in those misunderstandings. Temporal evolution of citation graphs, an Internet graph, bipartite affiliation graphs, a recommendation network and an email communication network are studied in his work. Two striking observations of the work are the networks are becoming denser, $e(t) n(t)^a$, and the diameter of the network is decreasing as the network increases. Comparing these to the diameter of random networks, in Erdos Renyi random

networks diameter grows slowly as a function of $\log N / \log Z$ where N is the number of nodes in the network and z is the average degree. In the case of preferential attachment the diameter is growing as $\log n$ or $\log \log n$.

One difficulty of generators are the significance of the proposed technique. In this specific case, one might form a graph by forcing $n(t)^a$ links for each new node introduced to the network. This clearly is not a justifiable as the method would mean each new paper must cite $n(t)^a$ papers in citation networks. In the same work of [47], two rather meaningful methods to fill this gap are proposed, forest fire model and community guided attachment. Community guided attachment takes into account the fact that power law distributions are observed together with the self similarity property of the network. The model represents the recursive structure of communities within communities as a tree. For every node in the graph there is a leaf node in the tree. Edges between nodes are added as a function of the tree distance of the nodes. That is if two nodes are closer to each in the tree, they are more likely to form links between them. Since scale free property is desired the edge forming function should be level independent, that is for any distance h in the tree, $f(h)/f(h-1)$ is a constant. The whole process of forming the graph can be thought of as first forming the communities and then linking the nodes.

c) Forest Fire Model: The second method proposed is the forest fire model that forms by the a directed graph. The model has two parameters, a forward burning parameter and a backward burning parameter, p and q respectively. The graphs grow one node at a time by choosing an ambassador node w uniformly random for the newly introduced node u . Selecting x outlinks and y inlinks of the ambassador where x and y are geometrically distributed with means $p/(1-p)$ and $rp/(1-rp)$ node u forms outlinks to these $x+y$ nodes. This process is repeated for each of the outlinks u choses. This rather basic model has heavy tail in/out degrees, community structure, densification power law and shrinking diameter properties. Extensions to the method are provided to fit the data better.

4) Fitness: In preferential attachment model nodes that arrive early will end up having highest degrees. However, one could envision that each node has an inherent competitive factor that nodes may have, capable of affecting the network's evolution refered as node fitness. In these models a fitness parameter, given in (14), is attached to each node that does not change over time [7], [24], [30]. A new node can still attract others if the associated fitness value for that node is high. That is the most efficient (or "fit") nodes are able to gather more edges.

$$\Pi = \frac{n_v k_v}{\sum_i n_i k(i)} \quad (14)$$

Likewise, [54] proposes growth models of social networks using some predefined principles. Preferential attachment is included, the acquaintances decreases over time if the

individuals are meeting occasionally and there is an upper-bound on the number of friends a node can have.

In [22] the authors presents a method for understanding the governing dynamics of evolving networks that relies on attachment kernel functions. Attachment kernel functions are simply a scalar function of some of the node properties. By defining the kernel function in terms of the potentially important vertex properties the evolution of the network is modeled. In the case of preferential attachment method, the kernel function is given by the node degree, that is $A(d_i(t))$ where A is the kernel function $d_i(t)$ is the degree of node i at time t . In the case of citation networks, using degree of a node as the kernel function the probability of a paper i citing a d degree paper j is given by $P[\text{cites } d \text{ degree node}] = P_i(d) = \frac{A(d)N_d(t)}{\sum_k A(d_k(t))}$ where $N_d(t)$ is the number of d degree nodes in the network at time step t . Using this formula $A(d)$ is then given by $A(d) = P_i(d)S(t)/N_d t$ where $S(t)$ is the sum of the kernel attachment function for each node of the graph at time t . The authors use an iterative approach that starts with $S_0(t) = 1$ and measure $A_0(t)$ which is then used to give a better estimate for $S(t)$ to find out the exact values. In practice after 5 iterations are the algorithm converges and an approximation of the kernel function is found.

5) Opposition to Power Laws: Although power laws have been reported in various application domains [32], [65], [33], [10], [41], [72], [13], [25], [53], [73], [43], [61] Pennock et al and others have observed deviations from a pure power law distribution in several datasets [58], [6], [66], [2]. The common deviations are found to be exponential cutoffs and lognormals. The authors discover qualitatively different link distribution among subcategories of pages for example, among all university homepages or all newspaper homepages. The distributions within these specific categories are typically unimodal on a log scale, with the location of the mode, and thus the extent of the rich get richer phenomenon, varying across different categories. The authors also note that the connectivity distribution over the entire web close to a pure power law.

There has been some oppositions to power laws in biological data as well. The authors in [66] proposes not to use frequency-degree plots to decide whether the degree distribution follows a power law, but rather to use rank-degree plots which they claim is a more accurate way of expressing the distribution. The results suggest that FYI(filtered yeast interactome) and HPI(human protein interaction) are following an exponential distribution rather than the expected power law distribution when rank-degree distribution is used. When the frequency-degree plots used however, the distribution is still a power law.

The exponential cutoff models distributions look like power laws over the lower range of values on the x axis but decay exponentially quickly for higher values. Lognormal distributions look like truncated parabolas on log-log scales and model situations where the plot dips downwards in the lower range of values on the x axis.

C. Window Based Methods

In the study of [21] a data structure and updating scheme that captures the graph and its evolution through time is given for the AT&T call data. The studied graphs in this work have hundreds of millions of nodes and edges, and the rate of change is hundreds of thousands of new nodes and edges per day. That is the graphs are highly dynamic.

The main contribution of the paper is that the authors defines means to summary the past behavior of the dynamic graph in hand and then predict the graph for the next time slot. Let g_t be the transactions at time t , and let the sum of two graphs be defined as;

$$G = \alpha g \oplus \beta h$$

where α and β are non-negative scalars and the nodes and edges in G are obtained by the union of nodes and edges in g and h . The weighted graph is also defined similarly.

One way to summarize the past behavior is then the cumulative sum of the graphs. Another way is to use only recent activity, that is the sum of last k time steps:

$$\mathcal{G}_t = g_{t-k+1} \oplus g_{t-k+2} \oplus \dots \oplus g_t = \bigoplus_{i=t-k+1}^t g_i$$

\mathcal{G}_t can be used as a predictor for the graph at time $t + 1$. The simplest such prediction for the graph is surely g_t which does not store any past activity and prone to errors. A better prediction is possible by a windowing scheme where $\mathcal{G}_t = \bigoplus w_i g_i$ with $\sum w_i = 1$. Then a particular convenient way for the weights of the graph is $w_i = \theta^{t-i}(1-\theta)$ which in turn gives a predictor of the form

$$\mathcal{G}_t = \theta \mathcal{G}_{t-1} \oplus (1-\theta)g_t$$

which is also referred as the exponential smoothing.

D. Kronecker Graphs

Another graph generation technique is presented in which heavy tails for in and out-degree distribution as well as eigenvalues and eigenvectors, small diameters and the densification power law [47] are preserved in [46]. The previously proposed forest fire model cannot be analyzed rigorously although being successful in capturing the observed properties of the real networks. To capture the shrinking diameter property and the densification power law and simple enough to be analyzed, Kronecker graphs are proposed. In Kronecker graphs evolves recursively creating self similar graphs. The growing sequence of graphs are Kronecker product of of the initiator adjacency matrix.

$$C = A \oplus B = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1m}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & a_{n2}\mathbf{B} & \dots & a_{nm}\mathbf{B} \end{pmatrix} \quad (15)$$

To smooth out the discrete nature of the process, stochastic Kronecker graphs are also introduced. In this model the initiator matrix is a probability matrix rather than an adjacency

matrix and the Kronecker multiplication is applied as above. Each “1” and “0” of G_1 is replaced with α and β to reduce the number of parameters from N_1^2 , the number of entries in the matrix, to two. The authors proceed with a detailed mathematical analysis of the model.

E. Entropy Based Methods

The evolution of several blogs and blog post are studied in [49]. Topological features of these blogs are found to distinguishing, the behavior of the evolution is bursty. That is, it is not uniform but yet it is self similar and therefore compactly described by the bias factor. The authors use the entropy of post at time t over total number of posts. That is $H_p = -\sum_{t=1}^T p(t) \log p(t)$ where $p(t) = P(t)/P_{Total}$. The entropies are found to be much lower than the entropies of the uniform sequences. A more interesting result is the fit of a b-model for these blogs. In the b-model if the total activity on a blog is P messages, in the first half of the evolution b fraction of the posts appear and in the second half, $1-b$ fraction appears. Recursively repeating this for the first and the second half the authors obtains a b-model. The b-model produces a linear entropy as a function of time. Interestingly enough the blog evolution shows the same property. The authors provide how to extract the value of the b from a given history of the evolution.

V. DYNAMIC NETWORK ANALYSIS

There is a strong correlation between finding patterns in static graphs, modeling dynamically and clustering evolving graphs and anomaly detection in dynamic graphs. In this section we give graph clustering techniques, anomaly detection algorithms in particular for dynamic graphs.

A. Graph Similarity

Measures We start with the graph similarity functions as they are excessively used in clustering, spotting anomalies and for various other problems. Graph similarity functions are categorized in two groups:

- feature based similarity measures
- structure based similarity measures

Using the topology of the graphs, two similarity metrics has been defined, maximum common subgraph distance and the graph edit distance.

Let $|mcs|$ be the size of the maximum common subgraph and $|g_1|$ and $|g_2|$ be the size of the input graphs. The maximum common subgraph distance is then defined as

$$d(g_1, g_2) = 1 - \frac{|mcs(g_1, g_2)|}{\max(|g_1|, |g_2|)}, \quad (16)$$

in [15].

The edit distance between two graphs is the total cost of edit operations (i.e. deletion of vertex, addition of edge) that transform a graph to another graph.

Let e be an edit operation and $c(e)$ its cost and the cost of a sequence of edit operations be the sum of the individual edit operations. The graph edit distance is then given by the minimum cost that transforms g_1 to g_2 . The graph

edit algorithms search amongst the possibly non-unique edit sequences that results in a minimum edit cost. Several graph edit distance computation procedures are discussed in [51].

In feature based similarity graphs, each graph is represented as a feature vector and the similarity is defined as the distance between the corresponding vectors. The diameter, average clustering coefficient, graph spectra are the most commonly used feature vectors.

B. Anomaly Detection

One important area of data mining is the anomaly detection in a time series of data. A good deal of research has been put together on network intrusion and fraud detection systems. However little work is carried on discovering anomalies in graph based data. We give a brief overview of anomaly detection techniques in graph based data in this section.

1) *Time Series Analysis of Graph Data*: There has been a tremendous work on time series analysis. Being one of the pioneers, Box and Jenkins' ARIMA models are one of the mostly used method in times series analysis [11], [12].

The general idea of ARIMA models is

- Formulate a class of models assuming certain hypotheses,
- Identify a model for the observed data,
- Estimate the model parameters,
- If the hypothesis of the model are validated, use the model for forecasting. If not, identify a new model and estimate the parameters until the hypothesis of the model are validated.

The first order and second order autoregressive models are given by:

$$\begin{aligned}\dot{w}_t &= \phi_1 \dot{w}_{t-1} + a_t \\ \dot{w}_t &= \phi_1 \dot{w}_{t-1} + \phi_2 \dot{w}_{t-2} + a_t\end{aligned}\quad (17)$$

The backward shift operator is defined by equation (18) to represent (17) in a more compact form:

$$Bw_t = w_{t-1} \quad (18)$$

Using the backward operation (18) and equation (17) we have:

$$\begin{aligned}0 &= \dot{w}_t - (\phi_1 \dot{w}_{t-1} + \phi_2 \dot{w}_{t-2} \dots + a_t) \\ &= \dot{w}_t \left\{ 1 - \phi_1 \frac{\dot{w}_{t-1}}{\dot{w}_t} - \phi_2 \frac{\dot{w}_{t-2}}{\dot{w}_t} \dots \right\} - a_t \\ &= \dot{w}_t \{ 1 - \phi_1 B - \phi_2 B^2 \dots \} - a_t.\end{aligned}\quad (19)$$

Alternatively in moving average models \dot{w}_t is linearly dependent on a_t and on one or more of the previous a's.

$$\begin{aligned}\dot{w}_t &= a_t - \theta_1 a_{t-1} \\ \dot{w}_t &= a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}\end{aligned}\quad (20)$$

Using the same backward operation equation (20) becomes

$$\begin{aligned}\dot{w}_t &= a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} \dots \\ &= a_t \left\{ 1 - \frac{\theta_1 a_{t-1}}{a_t} - \frac{\theta_2 a_{t-2}}{a_t} \dots \right\} \\ &= a_t \{ 1 - \theta_1 B - \theta_2 B^2 - \dots \}\end{aligned}\quad (21)$$

An autoregressive model cannot be used to represent moving average behaviour unless it uses an infinite number of autoregressive terms, so merging auto regressive method with the moving average model we obtain the *general mixed autorregressive-moving average model of order (p, q)* given by the following equation:

$$\begin{aligned}\dot{w}_t - \phi_1 \dot{w}_{t-1} - \phi_2 \dot{w}_{t-2} \dots - \phi_p \dot{w}_{t-p} \\ = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} \dots - \theta_q a_{t-q}\end{aligned}$$

Again using the operator B (22) this can be rewritten by

$$\phi_p(B) \dot{w}_t = \theta_q(B) a_t \quad (22)$$

where

$$\begin{aligned}\phi_p(B) &= 1 - \phi_1 B - \phi_2 B^2 \dots - \phi_p B^p \\ \theta_q(B) &= 1 - \theta_1 B - \theta_2 B^2 \dots - \theta_q B^q\end{aligned}$$

In order to make this series stationary the roots of $\phi(B) = 0$ and $\theta(B) = 0$ must lie outside the interval $[-1, 1]$. With this condition satisfied the model defined in (22) becomes a valueable method to represent *stationary* time series. There are infinite ways a time series can be unstationary and the authors give two types of non-stationary series and how to convert it to a stationary time series.

In the work of Pincombe et al. [59] the idea of ARIMA models is borrowed for the purposes of graph based anomaly detections. With the assumption that a single feature can accurately represent a graph, the method extracts graph based features like diameter distance, edit distance, maximum common subgraph vertex distance and so on. The algorithm then models the time series of each of the individual features as an ARMA model and compares which of the features is better in terms of capturing the anomalies. In the general case though the assumption that a single feature can represent the whole graph is too aggressive to be true.

2) *Anomaly Detection using Minimum Description Length*: Several methods have been proposed using the minimum description length principle to detect graph anomalies. The MDL in the context of graph theory is the number of bits required to encode the graph. Using this principle the most common substructure, "pattern" can be calculated by

$$M(S, G) = DL(G|S) + DL(S) \quad (23)$$

where G is the entire graph, S a substructure in graph G , $DL(S)$ is the description length of the substructure and

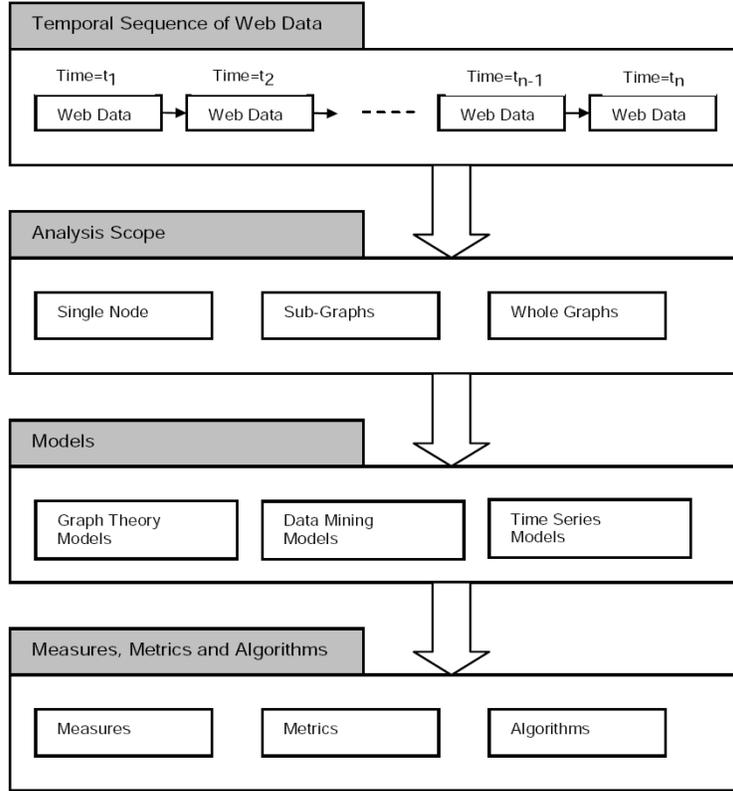


Fig. 2. An overview of studying evolving graphs. A temporal sequence of graph data is the input. The scope of the analysis is application specific. Dependent on the specific application it can be the evolution of features of a single, a group of nodes more specifically sub-graphs or the whole graph itself. Researchers have tackled the problems mainly in three different ways, using graph theoretical approaches, data mining models and time series analysis.

$DL(G|S)$ is the description length of the graph after compressing the graph with the substructure S . The substructures that minimize this function are the “patterns” in the graph and using these patterns several methods have been proposed to detect graph anomalies. The simplest idea is one that finds the structures that maximize equation (23). This however is not suitable as no compression at all, or other extreme case compressing the whole graph turns out to be the high. A heuristic approach which calculates the value of the size of a substructure multiplied by the frequency of that substructure is therefore used rather than directly trying to maximize equation (23) in [56]. In the same work Noble et al also proposes studying the different substructures of the graph amongst each other to determine how anomalous each of the subgraphs are compared to the others. One of the main restrictions of the methods proposed by [56] is that the found anomalies are always connected. Recall however that in [59] the anomaly can be anything but the algorithm says nothing about where the anomaly is.

The method proposed in Noble et al is biased towards finding small substructures and this issue is addressed by Eberle et al in [26] again using the MDL principle. The algorithm starts with finding the best substructures, S_i 's of the graph using MDL. Then rather finding the structures that are least similar to this substructure, the authors find matching instances, I_j , such that the cost of transforming

I_j to S_i is positive. In the final step if the frequency of these matching structures multiplied by the cost of the transformation is less than a threshold, the substructure is output as an anomaly. The intuition here is that an anomaly is a structure that is infrequent and slightly different than the most common pattern. In the same work a probabilistic approach is also given. The extensions to the substructure are assigned a probability and the substructure with the least probability is output as the result of the algorithm. The authors also present another algorithm which examines the ancestral substructures. All these methods are prone to miss the anomalies if the dataset contains more than one type of anomaly ie a deletion followed by modification.

3) *Window Based Approaches*: The idea of scan statistics, also known as moving windows analysis is also applied in anomaly detection in graphs [60]. Briefly in scan statistics method, a small window of the data is analyzed and local statistics in that window are calculated. The maximum of such statistics is called the scan statistics of the data, and represented by $M(x)$. The approach is then to check whether this scan statistics is more than a threshold value. If there is a data for which $M(x)$ is greater than the threshold, then there is an anomaly according to the scan statistics method.

In the case of graph theory, the window to be scanned is the k neighborhood of a vertex v . The authors defined the scan statistics to be the outdegree of the k^{th} neighborhood

where for $k = 0, 1, 2$ of vertex v . Some standardization is performed by vertex dependent mean and variance lest to miss a vertex that increases its communication but goes unnoticed due a vertex that has a higher and constant communication. By restricting the anomaly detection only to scan statistics of the second neighborhood, a “chatter” is detected in the famous Enron dataset.

Another partly window based algorithm is proposed by [39]. The dependency graph for application layer of web based systems are constructed. The weights of the graph represent how much a service requested the service at the other of the edge in a given time interval. The authors define the activity vector $u(t)$, which is merely the eigenvector corresponding to the maximum eigenvalue. The activities are stored in the matrix $U(t) = [u(t), u(t-1) \dots u(t-h+1)]$. Then the “typical pattern” is defined to be the left singular vector of $U(t)$. For any new data point at time t , this $U(t-1)$ matrix is constructed, the typical pattern is extracted and then the typical pattern vector is compared against the activity vector at time t . The angle between the vectors give how similar the activity vector and the typical pattern of the data in the last h time spots. The authors provide an online algorithm to calculate the threshold for the angle as well.

4) *Approaches using Vertex Edge Properties:* A recent study by Papadimitriou et al uses vertex and edge properties to find how similar the web graphs are [57]. If two consecutive graphs are too similar or too different than each other, this may indicate an anomaly in the series. The first such vertex property is the rank of a vertex defined by Kleinberg et al [42]. Given two graphs the ranks of the vertices are calculated and sorted. Using the similarity function of

$$sim_{VR}(G, G') = 1 - \frac{2 \sum_{v \in V \cup V'} w_v \times (\pi_v - \pi'_v)^2}{D} \quad (24)$$

where π_v and π'_v are the ranks of vertex v in G and G' respectively, w_v is the quality of v and D as a normalization factor, the (dis)similarity of the graphs are calculated.

Another method proposed in the same work is to use the edge weight similarity of the graphs. The edges shared by both of the graphs are used for this approach. Similar to the vertex ranking approach, the weight of an edge is multiplied by the quality of the vertex that it originates from. This value is normalized by a factor to give a measure for an edge similar to the rank of a vertex, but defined for the edge. This measure is then used for similarity checking between the graphs. Another methods have been defined as well, based on the idea that if two graphs are similar they share a common vertex and edge set.

Two other methods have been defined in the same work. The experiments carried on show that they are not as useful as the first three methods, nonetheless we will mention them here briefly for the sake of completeness. A sequence comparison scheme can be used, if a method that can convert the graph data to a sequence while preserving the topological features of the graph can be proposed. This problem is also known as graph seriation problem. The other method assigns

a set of features to vertices and edges, namely quality of the vertex to vertex and normalized quality to an edge. These values are then hashed and a similarity check between two signatures are performed for consecutive graphs.

[64] uses a similar idea for bipartite graphs. A relevance score for each node is defined. Using random walks originating from a vertex u , the number of times vertex v is seen counted. Intuitively the higher the probability of visiting v the higher the two nodes are relevant. For a node u in the graph, the neighbors should have a high relevance for u to be normal. Otherwise the node is considered as abnormality. [17]

Finding Hidden Groups in Graph Data: [5], [4], [31]. In [34] using the centrality indices the authors find the community structure in social and biological networks. This work is not focused on the dynamic evolution though.

C. Clustering Dynamic Graphs

Data clustering in general seeks to find homogeneous groups where within group similarity is minimized and the between group similarity is maximized. In Han and Kamber [35] clustering methods for static data are classified into five major categories, partitioning methods, hierarchical methods, density based methods, grid based and model based methods.

The partitioning methods construct k partitions of the data. These partitions are crisp if each and every object in the dataset is allowed to belong to only one partition. In the fuzzy partitioning however, an object can belong to more than one partition with varying probabilities. Lloyd’s algorithm which is commonly referred as the k-means algorithm and the k-medoids algorithms are the two well known crisp partitioning heuristics where c-means and the c-medoids are the counterparts. The second class of clustering methods work by clustering the data in a hierarchical manner. These algorithms construct a tree of the dataset to represent the hierarchy. Hierarchical algorithms are either divisive or agglomerative where in divisive methods the whole data is considered a cluster and split in each iteration. The reverse strategy is taken in the agglomerative methods. Density based methods continue to grow clusters as long as the number of neighbors (density) is below a threshold. If it is more than the specified threshold than a new cluster is formed. Grid based techniques have been also proposed for static data clustering where a grid structure is placed on the object space. All the clustering operations take place on this grid structure. The last set of methods used for clustering is the model based algorithms. These methods try to fit models for each of the cluster and fit the data to the assumed model as much as possible.

Recent studies introduced techniques to cluster dynamic data. These studies either try to modify the existing clustering algorithms for static data to fit in a dynamic environment or try to convert the dynamic data into a form so that the existing algorithms can be used directly.

1) *Modifying Existing Algorithms:* There is a vast literature on how to cluster static graphs. In the literature graph clustering usually refers to clustering nodes of a graph. However the emphasis in this survey is clustering a set

of graphs that evolve over time rather than clustering the nodes in the individual graphs. Birkhäuser discusses what is needed to extend different types of clustering algorithms from n-dimensional real space to graph domains [14]. Two modifications are in need, a graph distance function $f(g_1, g_2)$ for any given two graphs and a method to calculate the center of a cluster i.e. the center of a set of graphs. Several distance metrics have been proposed as discussed in section V-A and can be used for this purpose. In the work of [14] to find the center of a cluster, a median graph definition is given as in equation (25).

$$\bar{g} = \operatorname{argmin}_{g \in U} \sum_{i=1}^n d(g, g_i). \quad (25)$$

Using the median graph definition and the distance functions, centroid based and hierarchical clustering algorithms proposed for n-dimensional real space can be modified to work in graph domain as well.

In [38] Hopcroft et al studied NEC CiteSeer database to track the evolving communities. They periodically create agglomerative clusters and examine the evolution of the communities. A similarity metric between two papers is defined as the cosine angle between the associated references. If two papers cite the same set of papers, then they are similar to each. Using this distance metric agglomerative clustering is performed on the data. Similar to the cosine similarity, the distance between two clusters is given as

$$\operatorname{distance}(C, C') = \sqrt{n_c n_{c'} / (n_c + n_{c'})} (1 - \cos(r_c, r_{c'}))$$

where r_c and $r_{c'}$ are the associated reference vectors.

Finding clusters does not necessarily mean the the natural communities are found. To find the communities a series of subgraphs $G_1, G_2 \dots G_n$ are produced from the original network G where each G_i is a graph induced by the 95% of the vertices of G . The clustering algorithm is then used to produce a set of trees $T = \{T_1, T_2 \dots T_n\}$. A community in C in T_1 is natural if in a fraction of f of the clustering trees of T the distance of the best match of a community is greater than a threshold value of p . Tuning these parameters is application specific.

2) *Minimum Description Length*: Sun et al used the minimum description length idea both to cluster a given graph stream into graph segments and the sources and destinations in each of the graph segments [63] for bipartite graphs. That is, the authors try to answer two questions: given a graph segment find partitions of source and destination nodes and second given a graph stream find the graph segments. The cost of encoding the graph segment is provided and then the algorithm seeks to find graph segments such that the sum of the cost of encoding graphs segments is minimized.

This problem is NP-hard and a minimization algorithm is provided. Given the graph segment, the algorithm first finds source partition pairs that when merged gives the minimum encoding cost. If the total encoding cost decreases then these nodes are merged. The same idea is applied to split as well. The partition with the largest entropy is found. Foreach node

in this partition if the encoding cost decreases without the node being in this partition, then a new partition starts. This process is repeated until there is no decrease in the encoding cost. By this way a given graph segment is partitioned. A new graph is combined to the current graph segment if the difference between the cost of encoding the segment with the merged graph and the cost of encoding the individual graph is less than a threshold.

VI. CONCLUSIONS

Many real world complex systems can be represented as graphs. The entities in these system represent the nodes or vertices and links or edges connect a pair or more of the nodes. We encounter such networks in almost any application domain i.e. computer science, sociology, chemistry, biology, anthropology, psychology, geography, history, engineering. Until recently, the research on network theory has mainly focused on graphs with the assumption that they remain static, ie they donot change over time. A wealth of knowledge has been developed for this type of static graph theory.

Work on dynamic graph theory have been motivated by finding patterns and laws. Power laws, small diameters, shrinking diameters have been observed. Graph generation models that try to capture these properties are proposed to synthecially generate such networks. There are several problems to be answered in these complex networks. Is the network evolving normally? What is normal behaviour? Is there a phase transition?

This study mainly focussed on evolution models, graph similarity measures, anomaly detection in large network based data and clustering similar graphs together. This field is incredibly rapid, and inevitably we have not mentioned all the aspects of dynamic graphs to have a compact and rather self-contained survey. We refer interested users to [18] and [45].

REFERENCES

- [1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, 2000.
- [2] LAN Amaral, A. Scala, M. Barthelemy, and HE Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences*, 2000.
- [3] A.L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509, 1999.
- [4] J. Baumes, M. Goldberg, M. Hayvanovych, M. Magdon-Ismael, W. Wallace, and M. Zaki. Finding Hidden Group Structure in a Stream of Communications? *LECTURE NOTES IN COMPUTER SCIENCE*, 3975:201, 2006.
- [5] J. Baumes, M. Goldberg, MS Krishnamoorthy, M. Magdon-Ismael, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. *Proceedings of the IADIS International Conference on Applied Computing*, pages 97–104, 2005.
- [6] Z. Bi, C. Faloutsos, and F. Korn. The "DGX" distribution for mining massive, skewed data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 17–26. ACM New York, NY, USA, 2001.
- [7] G. Bianconi and A.L. Barabasi. Competition and multiscaling in evolving networks. *Europhysics Letters*, 54(4):436–442, 2001.
- [8] C. Bilgin, C. Demir, C. Nagi, and B. Yener. Cell-Graph Mining for Breast Tissue Modeling and Classification. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5311–5314, 2007.

- [9] J. Blasiak and R. Durrett. Random Oxford graphs. *Stochastic Processes and their Applications*, 115(8):1257–1278, 2005.
- [10] V. Boginski, S. Butenko, and P.M. Pardalos. Mining market data: A network approach. *Computers and Operations Research*, 33(11):3171–3184, 2006.
- [11] G. E. P. Box, G. M. Jenkins, and MacGregor. G. M. Some Recent Advances in Forecasting and Control. *Applied Statistics*, 17(2):91–109, 1968.
- [12] G.E.P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.
- [13] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. *Computer Networks*, 33(1-6):309–320, 2000.
- [14] H. Bunke. *A graph-theoretic approach to enterprise network dynamics*. Birkhauser.
- [15] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.
- [16] J. Camacho, R. Guimerà, and L.A. Nunes Amaral. Robust Patterns in Food Web Structure. *Physical Review Letters*, 88(22):228102, 2002.
- [17] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. *Conference on Principles and Practice of Knowledge Discovery in Databases*, 2004.
- [18] D. CHAKRABARTI and C. FALOUTSOS. Graph Mining: Laws, Generators, and Algorithms. *ACM Computing Surveys*, 38:2, 2006.
- [19] A. Chapanond, M.S. Krishnamoorthy, and B. Yener. Graph Theoretic and Spectral Analysis of Enron Email Data. *Computational & Mathematical Organization Theory*, 11(3):265–281, 2005.
- [20] F.R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- [21] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Computational methods for dynamic graphs. *Journal of Computational and Graphical Statistics*, 2003.
- [22] G. Csardi, K. Strandburg, L. Zalanyi, J. Tobochnik, and P. Erdi. Estimating the dynamics of kernel-based evolving networks. *Arxiv preprint cond-mat/0605497*, 2006.
- [23] SN Dorogovtsev and JFF Mendes. Evolution of networks. *Advances In Physics*, 51(4):1079–1187, 2002.
- [24] SN Dorogovtsev, JFF Mendes, and AN Samukhin. Structure of Growing Networks with Preferential Linking. *Physical Review Letters*, 85(21):4633–4636, 2000.
- [25] R. Durrett and J. Schweinsberg. Power laws for family sizes in a duplication model. *Ann. Probab.*, 33(6):2094–2126, 2005.
- [26] W. Eberle and L. Holder. Discovering Structural Anomalies in Graph-Based Data. *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 393–398, 2007.
- [27] V.M. Eguíluz, D.R. Chialvo, G.A. Cecchi, M. Baliki, and A.V. Apkarian. Scale-Free Brain Functional Networks. *Physical Review Letters*, 94(1):18102, 2005.
- [28] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17–61, 1960.
- [29] P. Erdős and A. Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Scientia Hungary*, 12:261–267, 1961.
- [30] G. Ergün and GJ Rodgers. Growing random networks with fitness. *Physica A: Statistical Mechanics and its Applications*, 303(1-2):261–272, 2002.
- [31] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou. Community Dynamics Mining. *Proc. of 14th European Conference on Information Systems*, 2006.
- [32] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262, 1999.
- [33] P. Fu and K. Liao. An Evolving Scale-free Network with Large Clustering Coefficient. *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–4, 2006.
- [34] M. Girvan and MEJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821, 2002.
- [35] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.
- [36] F. Harary. Graph theory. *Reading*, 1969.
- [37] F. Harary and G. Gupta. Dynamic graph models. *Mathl. Comput. Modelling*, pages 79–87, 1997.
- [38] J. Hopcroft. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences*, 101(suppl. 1):5249–5253, 2004.
- [39] T. Ide and H. Kashima. Eigenspace-based anomaly detection in computer systems. *Conference on Knowledge Discovery in Data: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 22(25):440–449, 2004.
- [40] H. Jeong, S.P. Mason, A.L. Barabasi, Z.N. Oltvai, et al. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.
- [41] H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, A.L. Barabasi, et al. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.
- [42] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [43] J. Kleinberg and S. Lawrence. The structure of the web. *Science*, 294(5548):1849–1850, 2001.
- [44] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale Knowledge Bases from the Web. In *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES*, pages 639–650, 1999.
- [45] Jure Leskovec. *Dynamics of Read-World Networks*. PhD thesis, Cornell University, 2008.
- [46] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, and Christos Faloutsos. Realistic mathematically tractable graph generation and evolution using kronecker multiplication. *European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2005.
- [47] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*, 2007.
- [48] L. Lu. The diameter of random massive graphs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 912–921. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2001.
- [49] M. McGlohon, J. Leskovec, C. Faloutsos, M. Hurst, and N. Glance. Finding Patterns in Blog Shapes and Blog Evolution.
- [50] M Mesbahi. On a dynamic extension of the theory of graphs. *Proceedings of the American Control Conference*, 2:1234–1239, 2002.
- [51] BT Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(5):493–504, 1998.
- [52] J.M. MONTROYA and R.V. SOLÉ. Small World Patterns in Food Webs. *Journal of Theoretical Biology*, 214(3):405–412, 2002.
- [53] MEJ Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):25102, 2001.
- [54] MEJ Newman. The structure of growing social networks. *Phys. Rev. E*, 64:046132, 2001.
- [55] MEJ Newman, SH Strogatz, and DJ Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):26118, 2001.
- [56] C.C. Noble and D.J. Cook. Graph-based anomaly detection. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, 2003.
- [57] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina. Web graph similarity for anomaly detection. Technical report, Technical Report 2008–1, Stanford University, 2008. URL: <http://dbpubs.stanford.edu/pub/2008–1>.
- [58] D.M. Pennock, G.W. Flake, S. Lawrence, E.J. Glover, and C.L. Giles. Winners don't take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Sciences of the United States of America*, 99(8):5207, 2002.
- [59] B. Pincombe. Anomaly Detection in Time Series of Graphs using ARMA Processes. *ASOR BULLETIN*, 24(4):2, 2005.
- [60] C.E. Priebe, J.M. Conroy, D.J. Marchette, and Y. Park. Scan Statistics on Enron Graphs. *Computational & Mathematical Organization Theory*, 11(3):229–247, 2005.
- [61] E. Ravasz, AL Somera, DA Mongru, ZN Oltvai, and A.L. Barabasi. Hierarchical Organization of Modularity in Metabolic Networks, 2002.
- [62] H. Reittu and I. Norros. Random graph models of communication network topologies. *Arxiv preprint arXiv:0712.1690*, 2007.
- [63] J. Sun, C. Faloutsos, S. Papadimitriou, and P.S. Yu. GraphScope: parameter-free mining of large time-evolving graphs. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696, 2007.

- [64] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. *Proc. IEEE Intl. Conf. Data Mining*, pages 418–425, 2005.
- [65] K. Takemoto and C. Oosawa. Evolving networks by merging cliques. *Physical Review E*, 72(4):46116, 2005.
- [66] Reiko Tanaka, Tau-Mu Yi, and John Doyle. Some protein interaction data do not exhibit power law statistics. *FEBS Letters*, 579:5140–5144, 2005.
- [67] A. Wagner. The Yeast Protein Interaction Network Evolves Rapidly and Contains Few Redundant Duplicate Genes. *Molecular Biology and Evolution*, 18(7):1283–1292, 2001.
- [68] Andreas Wagner and David A. Fell. The small world inside large metabolic networks. *Proceedings of Biological Sciences*, (1478):1803–1810, 2001.
- [69] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [70] DJ WATTS and SH STROGATZ. Collective dynamics of ‘small-world’ networks. *Nature(London)*, 393(6684):440–442, 1998.
- [71] B. Wemmenhove and NS Skantzos. Slowly evolving random graphs II: adaptive geometry in finite-connectivity Hopfield models. *Journal of Physics A Mathematical and General*, 37(32):7843–7858, 2004.
- [72] S. Wuchty, E. Ravasz, and A.L. Barabasi. The Architecture of Biological Networks. *Complex Systems in Biomedicine*, 2003.
- [73] H. Zhou and R. Lipowsky. Dynamic pattern evolution on scale-free networks. *Proceedings of the National Academy of Sciences*, 102(29):10052, 2005.