Graph models and generators

Anna Yakovlieva

November 14, 2021

Abstract

Traditional graph models, for example, the Erdös-Renyi and Barabashi-Albert models, have been known for a long time and are successfully used to analyze complex networks. However, they have significant drawbacks and, above all, inflexibility and lack of realism. These are rough results of analysis and forecasting in complex algorithms. The relevance of the topic of this work is that in the real conditions of the current COVID-19 pandemic, machine learning models could use the properties of complex networks to analyze and predict epidemiological data. The modern development of graph model generators is deep graph generation (DGG), which uses machine learning approaches.

This is a draft version of the report.

The latest version can be found here: report.pdf.

1 Introduction

Graphs are an essential representation of data in the form of objects and their relationships, which naturally arises in many areas. Graphs are a mathematical model of networks, including complex networks such as social networks, for example, Facebook, Youtube, Tik-Tok, molecular structures for pharmaceuticals, transport networks, citation networks, the WWW hypertext network itself, etc. The study of real networks deals with network theory and the theory of complex networks in features, which has many of its specific problems and applications.

The problems of studying networks and their models as graphs can be divided into two types [7]: 1) recognition and prediction of graph patterns, and 2) studying the distribution of graphs and generating new ones. The first type includes such tasks as classification of nodes, graphs, prediction of connections, detection of communities, detection of anomalies. The second type is graph generation.

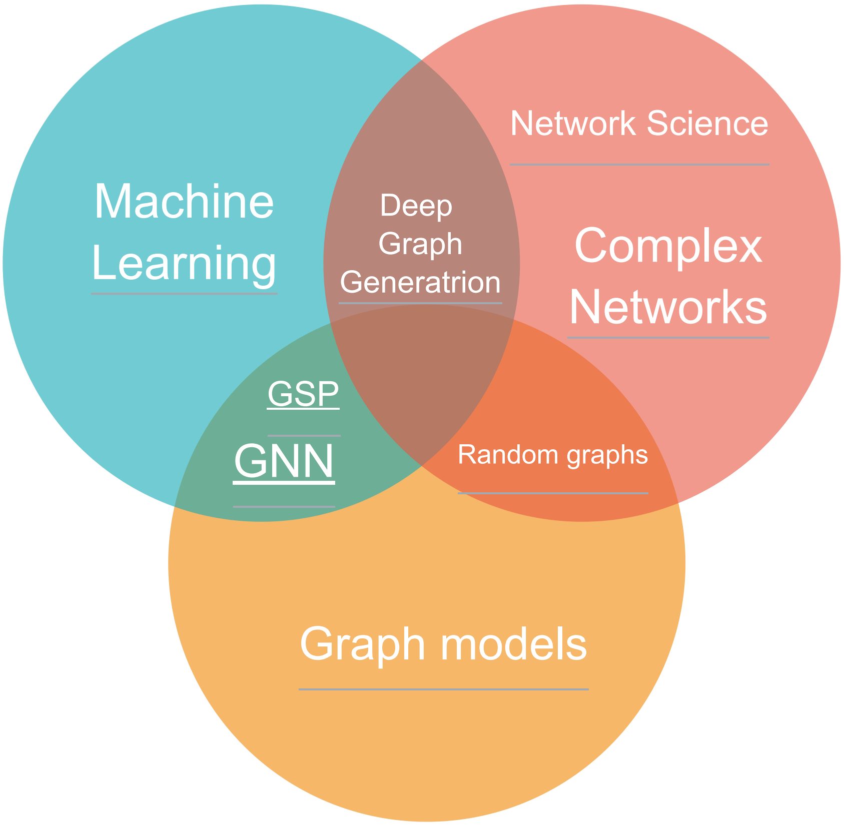

Figure 1: Machine learning vs Complex Networks

Graph generators differ from graph models in that they can use different algorithmic elements to give a synthetic graph certain properties. Thus, several generators can correspond to one graph model [5]. Thus, a generator is an algorithm or program, and a graph is a mathematical model of a network.

1.1 Graph Generation Surveys

The survey, [2], published in 2020, looks at traditional graph models, and only mentions DGGs. In this review, more than 40 generators are collected, their classification is carried out according to several parameters.

This survey brings a definite line under traditional graph generation. The overwhelming majority of works were published more than five years ago. From this we can conclude that the traditional generation of graphs is now more an applied technical problem than a scientific one.

In this respect, the older work of partially the same authors [1] from 2017 is indicative, where an example of a generator is given and its properties are studied using statistical modeling.

The survey [7] was published in 2020 and is the first on the DGG topic. It contains a fairly complete description of all models, as well as examples of their application.

The survey [4] published in 2021 is the second on the DGG topic and competes with the survey [7]. It contains a clearer classification of DGG methods, links to datasets, and used implementations.

There are currently no other new DGG surveys.

2 Graphs, network theory and complex networks

2.1 Local characteristics of graphs

Graph theory introduces definitions of local properties of graphs, which creates the basis for solving many algorithmic problems to determine these characteristics [15], [12], [11].

These are properties such as vertices (nodes) and arcs, digraphs and graphs with weights, subgraphs, cliques, degree of a node, paths, distances, shortest paths, strong and weak connectivity, and the incidence matrix.

2.2 Topological characteristics of graphs

In contrast to local characteristics, global ones characterize the graph and its properties as a whole, they are determined through local ones, but they are reduced to them.

For example: mean degree, degree distribution, mean path, eccentricity, radius and diameter, clustering coefficient, efficiency, centrality, degree of centrality, web centrality, prestige, PageRank, hubs, power ratings.

2.3 The structure of complex networks

The following characteristics of complex networks are distinguished: betweenness centrality, subgraphs and motives, cliques, communities, communities, groups, communes, Assortative and dissortative mixing.

2.4 Tasks for complex networks

Collaboration on social media, resistant to attacks, endurance assessment, safety assessment, synchronization in networks, self-organized criticality, cascading damage and spread of infections, community search, maximizing the spread of influence, complexity score.

3 Graph models

3.1 Small World Model (Watts-Strogatz)

In 1967, Harvard sociologist Stanley Milgram, based on his sociological research, made a statement that surprised many: every person on the globe can be associated with any other person by a chain of six acquaintances.

He sent out 296 letters with the same content to randomly selected people in two different cities of the United States.

These letters stated that this letter should reach a certain person in Boston whom these 296 people did not know, and the address this man was not known to them. Moreover, these people should forward letters only to those friends who, in their opinion, can help them achieve the desired final destination. In S. Milgram's experiment of 296 letters, 69 reached the goal that makes up 29% of the total. The average path length was found to be 6.2. This experiment is considered the first empirical evidence of the "Small world phenomenon".

36 years later, an experiment was carried out on a larger scale using E-mail. 24,163 volunteers were selected and sent emails letters to your friends. The final recipients were 18 people from 13 countries. Only 384 (!) Chains were completed, which indicates that social networks globally, they are quite sparse. However, the average path length turned out to be about four degrees of separation, that is, even less than "six steps".

In the mid-1990s, Steven Strogatz and his graduate student Duncan Watts of Cornell University in Ithaca, NY decided to study properties networks that have the property of a "small world". Computer modelling different types of networks has shown that this property is possessed by networks with a high degree of clustering and small average path length between nodes. Network of actors Hollywood, nematode worm neural network, Structure of the Internet and the World Wide Web also possesses the phenomenon of "small world".

3.2 Erdös-Renyi model

The simplest random networks are the so-called classical random graphs (Erdos-Renyi model) in the statistical ensemble of which all possible graphs with the number of nodes N and the number of links L have the same statistical weight of the implementation.

That is, for such networks, the probability of the existence of a connection between any two nodes is the same.

Strictly speaking, the concept of a random network cannot be applied to a single end network. Indeed, if you look at a network of a certain size, it is impossible understand which algorithm (deterministic or non-deterministic) it was built. Therefore, in the spirit of statistical physics, a random network can be define not as a unitary network, but as a statistical ensemble, that is, as a set networks in which each specific network has a certain probability of being realized, that is, each network in the ensemble has its own statistical weight. From this definition, it follows that a given random network is a network with a given a certain probability, another random network is a network with a different probability, etc. In order to get the average value for some quantity in a random network, we we average this value over all realizations, taking into account their statistical the weight.

3.3 Barabashi-Albert model

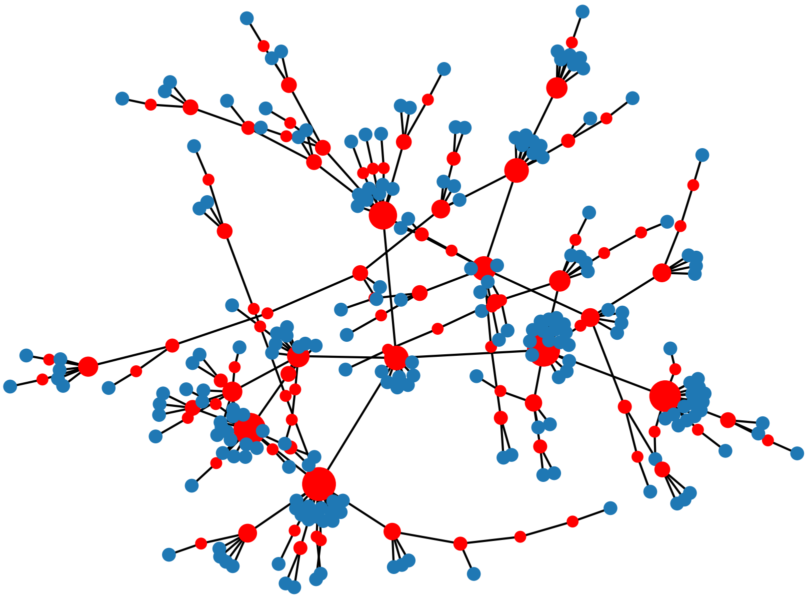

Figure 2: Barabási–Albert preferential attachment model

In 1999, a physicist from the University of Notre Dame (USA) Laszlo Barabasi together with his graduate student Reka Albert studied the properties of real networks with a slightly different point of view. If Strogatz and Watts in their study of networks proceeded from the phenomenon of "small world then Barabashi and Albert decided to investigate the law distribution of nodes of some real networks by the number of connections. The result was also unexpected.

Instead of the distribution of the number of bonds according to Poisson's law, which has a strict maximum near the average, for many real networks, such as metabolic networks and protein interactions in cells, the structure of air traffic in the USA, the structure of the Internet and its virtual counterpart, the World Wide Web, etc., such an average value does not exist, and the corresponding probability distribution obeys power law characteristic of all critical states.

Barabashi and Albert proposed a simple and elegant model for the emergence and evolution of scale-free networks. They showed that for the emergence of scale-free networks require two conditions:

1. Growth. Starting with a small number of m nodes, at each time step we add one new node with links that connect this new node to various already existing nodes.

2. Preferencial attachment. When nodes are selected to join new node, it is assumed that the probability with which the new the node will connect to an already existing node, depends on the number of links that this node is already linked to other nodes.

Scale-free networks are one of the manifestations of the phenomenology of critical phenomena in complex systems, since their structure obeys a power law.

3.4 Other models

There are several models that complement and develop the Barabashi-Albert model.

For example: Bollobashi-Riordan model, LCD model, Buckley-Ostguts model, Mori model, copy model, Chung-lu model, Yanson-Luchak model, Kronecker model.

4 Deep Graph Generation Models

The graph models described in the previous section were created relatively long ago and are well studied. They are used to efficiently generate synthetic graphs with specific properties that can represent networks in the real world. However, these models have several disadvantages and can only simulate some of the statistical characteristics of real networks. Due to their simplicity, these models have limited capabilities for modeling complex structures of real graphs. For example, the Erdos-Renyi model cannot simulate heavy tails of distributions, which is typical for real networks [7].

The main limitation of these approaches is that they cannot learn the graph structure of real networks from data. Moreover, often a priori knowledge about the structure of graphs is not available.

The modern approach is to form a graph model based on a training set of graphs. Instead of formulating a mathematical model, we use a given set of graphs to obtain a graph generation model with similar characteristics. After that, the quality of the resulting graph model can be checked with special metrics.

This overview report is not intended to use a mathematically accurate description of deep networks, since knowledge of VAE, GAN, LSTM is not assumed. Instead, he concentrates on a descriptive and substantive approach, following [4], [7], [6], [8].

4.1 VAE model

VAEs are the most popular approach in deep generative models. The theory and motivation of VAEs is related to statistical models of Bayesian variational inference. Its purpose is to train the probabilistic decoder $p_\theta(A|Z)$ with which you can do sampling, i.e., generating a realistic graph in the form of its adjacency matrix $\hat{A} \sim p_\theta(A|Z)$, due to the hidden variable Z. In order to train VAE, we combine a probabilistic decoder with a probabilistic encoder that encodes the input graph G into a hidden variable Z.

The idea in VAE is to use an encoder and a decoder together, training them to reconstruct the input graph using a posterior distribution $Z \sim q_\theta(Z|G)$. After that, we can drop the encoder and use unconditional sampling $Z \sim p(Z)$ to generate output graphs G based on a random hidden variable. This idea was proposed in [9] and called the Variational Graph Autoencoder (VGAE).

GNN is used for enocoder [14], [16], [13]. For a decoder a simple multilevel perceptron (MLP) is used.

4.2 GAN model

The VAE model has significant limitations, despite the fact that this model has a good probabilistic basis and much work has been devoted to studying the structure of the hidden space studied by VAE models. One of these limitations is the same feature as the well-known blurry images in the output of the image decoder. An alternative to the VAE framework is GAN. This model is one of the most popular.

The basic idea behind the GAN model is as follows. We define a generator and a discriminator and train together in a competitive game. Generator is a trainable function $g_\theta : R^d \to X$. The generator network is trained to generate realistic graphs $x \in X$. In this case, the input receives a random variable $z \in R^d$, for example, sampled from a normal distribution. At the same time, we define the discriminator $d_\theta : X \to [0, 1]$. The purpose of the discriminator is to separate real graphs and graphs obtained by the generator. The discriminator determines the probability that $x \in X$ is a fake.

For example, [3] uses a simple multilevel perceptron (MLP) as a generator to generate an adjacency matrix based on a random vector z:

$$\hat{A} = \sigma(MLP(z))$$

GNN is used as a discriminator for classification.

4.3 Autoregressive models

The autoregressive method combines VAE, GAN models. These models assume edges are generated independently. Instead, in the autoregressive approach, we assume that edges are generated sequentially and the probability of each edge is the conditional probability of the edge that was previously generated.

Autoregressive models can be recurrent and non-recurrent.

Recurrent DGGs are a bunch of autoregressive deep graph generators that use RNNs, namely LSTMs or GRUs, to account for the impact of generation history on the current decision. In turn, they can be divided into node-by-node generators and edge-by-edge generatores. Examples: GraphRNN, MolecularRNN.

Non-recurrent DGGs can be divided into attention-based methods, such as GRAM or AGE, which use the attention mechanism plays a major role, and other methods that are non-recurrent and do not use attention. These are, for example, DeepGMG or DeepGG models.

4.4 RL-based models

It uses a reinforcement learning approach. That being said, the grid generation process is reviewed at the Markov Decision Process (MDP).

At each step, the RL agent adds a subgraph or connects two nodes based on the current graph. As a result, the agent receives a reward similar to the one used in the discriminator in the GAN model based on the similarity of real graphs and the one that was generated.

4.5 Evaluating DGG

Various methods for generating graphs raise the question: how to evaluate the quality of different models [10]? How can we say that one graph generation method is better than others. The adopted method is a comparison of different statistical indicators of the generated graphs, comparable distributions of these characteristics with graphs from the test set. Among such characteristics, the distribution of degrees, cluster coefficients, etc. are used. It is possible to calculate the function of the distance between a group of synthetic graphs and graphs of the test sample.

5 Deep generative models, Complex Networks and applications

Among the applications of deep graph generation are: generation in molecular chemistry, as well as non-molecular generation: semantic parsing in NLP, protein modeling, code modeling, scene generation.

Generation of new molecules and their optimization is a fundamental problem in drug discovery, pharmaceuticals and chemistry [7], [4]. The goal here is to create new molecules with the desired properties. This is a very complex mathematical and computational combinatorial problem. Small variations in chemical structure can lead to significant changes in the properties of the molecule. Nowadays, most molecules are created by hand by experts in the field of chemistry and pharmacology. Recently, it was shown how it can use deep graph generation to solve these problems, if you represent the atoms of a molecule as nodes and the interactions of atoms as arcs.

Until a few years ago, the generation of molecules accounted for the majority of DGG applications, while now this proportion is about half.

6 Summary

Deep graph generation is a fast-growing area of graph theory and deep learning theory that should address many questions [7]. Among them: scalability, taking into account additional restrictions, increasing the variety of generation without data binding, generation of dynamic graphs, and others. It is hoped that the development of deep graph generation and progress in the field of machine learning in general and in GNN [14], [16], [13] in particular will make it possible to predict the behavior of complex networks much better than now to solve important problems, including medical and epidemiological ones.

References

[1] G. Bagan, A. Bonifati, R. Ciucanu, G. H. L. Fletcher, A. Lemay, and N. Advokaat. gmark: Schema-driven generation of graphs and queries. *IEEE Transactions on Knowledge and Data Engineering*, 29(4):856–869, 2017.

[2] A. Bonifati, I. Holubová, A. Prat-Pérez, and S. Sakr. Graph generators: State of the art and open challenges. *ACM Computing Surveys (CSUR)*, 53(2):1–30, 2020.

[3] N. De Cao and T. Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[4] F. Faez, Y. Ommi, M. S. Baghshah, and H. R. Rabiee. Deep graph generators: A survey. *IEEE Access*, 9:106675–106702, 2021.

[5] D. Goldenberg. Social network analysis: From graph theory to applications with python. *CoRR*, abs/2102.10014, 2021.

[6] X. Gu. Explore deep graph generation. 2019.

[7] X. Guo and L. Zhao. A systematic survey on deep generative models for graph generation. *CoRR*, abs/2007.06686, 2020.

[8] W. L. Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[9] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[10] V. Mostofi and S. Aliakbary. Towards quantitative methods to assess network generative models, 2018.

[11] M. Newman. *Networks*. Oxford university press, 2018.

[12] A.-L. Parabasi. Network science by albert-lászló barabási.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[14] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

[15] M. J. Zaki and W. Meira. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition.* Cambridge University Press, 2020.

[16] Z. Zhang, P. Cui, and W. Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.