History of Networks

In graph theory, the Erdos Renyi model is a model using for generating random graphs. They are named after Paul Erdos and Alfred Renyi, In the model introduced by Erdos and Renyi, all graphs on a fixed vertex set with a fixed number of edges are equally likely; in the model introduced by Gilbert, each edge has a fixed probability of being present or absent, independently of the other edges. These random graphical models can be used in the probabilistic method to prove the existence of graphs satisfying various properties, or to provide a rigorous definition of what it means for a property to hold for almost all graphs

A graph generated by the binomial model of Erdos and Renyi (p = 0.01)

* In the G(n, M) model, a graph is chosen uniformly at random from the collection of all graphs which have n nodes and M edges. For example, in the G(3, 2) model, each of the three possible graphs on three vertices and two edges are included with probability 1/3.
* In the G(n, p) model, a graph is constructed by connecting nodes randomly. Each edge is included in the graph with probability p independent from every other edge. Equivalently, all graphs with n nodes and M edges have equal probability of being connected.

{\displaystyle p^{M}(1-p)^{{n \choose 2}-M}.}

The parameter p in this model can be thought of as a weighting function; as p increases from 0 to 1, the model becomes more and more likely to include graphs with more edges and less

Then we moved on to computing the number of friends that any two nodes have in common, motivated by the problem of friend recommendations on social networks. The underlying idea can be traced back to Granovetter: two people are likely to know each other if they have many mutual friends. To compute the number of mutual friends between all pairs of nodes, we exploit the fact that the neighbors of every node share that node as a common friend. To count all mutual friends we simply loop over each node and increment a counter for every pair of its neighbors. For each node this scales as the square of its degree, so the whole algorithm scales as the sum of the squared degrees of all nodes. This can quickly become expensive if we have even a few high-degree nodes, which are quite common in practice.

he most widely known generative model for a subset of scale-free Albert's rich get richer generative model in which each new Web page creates links to existing Web pages with a probability distribution which is not uniform, but proportional to the current in-degree of Web pages. This model was originally invented by Derek J. de Solla Price in the year 1965 under the term cumulative advantage) According to this process, a page with many in-links will attract more in-links than a regular page. This generates a power law but the resulting graph differs from the actual Web graph in other properties such as the presence of small tightly connected communities.

The Watts–Strogatz model is a [random graph](https://en.wikipedia.org/wiki/Random_graph) generation model that produces graphs with [small-world properties](https://en.wikipedia.org/wiki/Small-world_network), including short [average path lengths](https://en.wikipedia.org/wiki/Average_path_length) and high [clustering](https://en.wikipedia.org/wiki/Clustering_coefficient). It was proposed by Duncan J. Watts and [Steven Strogatz](https://en.wikipedia.org/wiki/Steven_Strogatz) in their joint 1998 [Nature](https://en.wikipedia.org/wiki/Nature_(journal)) paper However the [ER](https://en.wikipedia.org/wiki/Erd%C5%91s%E2%80%93R%C3%A9nyi_model) graphs do not have two important properties observed in many real-world networks:

1. They do not generate local clustering and [triadic closures](https://en.wikipedia.org/wiki/Triadic_closure). Instead because they have a constant, random, and independent probability of two nodes being connected, ER graphs have a low [clustering coefficient](https://en.wikipedia.org/wiki/Clustering_coefficient).
2. They do not account for the formation of hubs. Formally, the [degree](https://en.wikipedia.org/wiki/Degree_(graph_theory)) distribution of ER graphs converges to a [Poisson distribution](https://en.wikipedia.org/wiki/Poisson_distribution), rather than a [power law](https://en.wikipedia.org/wiki/Power_law) observed in many real-world, [scale-free networks](https://en.wikipedia.org/wiki/Scale-free_networks).[[3]](https://en.wikipedia.org/wiki/Watts_and_Strogatz_model#cite_note-Ravasz2002-3)

The Watts and Strogatz model was designed as the simplest possible model that addresses the first of the two limitations. It accounts for clustering while retaining the short average path lengths of the ER model.

Homophily, or the formation of social ties due to matching individual traits

 Social Contagion, also known as social influence;

 The causal effect of an individual's covariates on their behavior or other measurable responses

The political blogosphere and the 2004 U.S. election: divided they blog

Lada A. Adamic from HP Labs and Natalie Glance from Intelliseek Applied Research Center studied the linking patterns and discussion topics of political bloggers and in particular, the degree of interaction between liberal and conservative blogs and to discover any difference in the structure of the two communities by analyzing top 40 A-list blogs, over the period of 2 months preceding US 2004 elections and to quantify how often they referred to each other, to understand the intersections of the topics discussed, in a both intra-community and inter-community manner.

Jonathan Bright from Oxford University in his paper Explaining the emergence of echo chambers on social media: the role of ideology and extremism explains the concept of echo chambers which refers to the idea that online conversations about politics are typically divided into a variety of subgroups, and that this division takes place along ideological lines with people only talking to others with which they are already in agreement. Therefore, it can be said there is subgroup or subcommunity formulation wheremost of the interaction takes place and intercommunity interaction is a lot lesser.

Types of networks:

Social Networks like Facebook, Google Plus, Twitter

Information Networks, World Wide Web

Activity Networks like email

Biological Networks like protein interactions

Geographical Networks like roads, railway tracks etc.

The professor also mentioned in the class that Facebook is slowly moving towards becoming an information network rather than a social network.

Network Representations

Networks can be represented in many ways.

Networks can be directed or undirected. Directed graphs are those graphs where the edges between two nodes has a direction property associated with it such that an edge from node A to node B doesn’t necessarily imply an edge from node B to node A. Undirected graphs are graphs where edges don’t have the direction property such that an edge form node A to B implies that there is also an edge from node B to node A.

Unweighted graphs are those graphs where the edges have no weights associated with them, whereas weighted graphs have edges that have a certain weight associated with them which usually denotes the strength of connection between two nodes.

There are other kinds of abstractions for representing networks such as metadata networks (where there is certain metadata associated with the nodes of the graph).

While creating networks, it is also suggested to do a lot of preprocessing to ensure that we are able to create a good network representation.

Data Structures

Networks can be represented by various data structures. Mainly two types of data structures are used for representing graphs. In an adjacency matrix representation, we have a matrix having N rows and N columns for a graph having N nodes and each entry in the matrix representing the presence or absence of an edge (or the weight of the edge in a weighted graph). In this representation, edge lookup is constant time but it requires a lot of storage space. It is possible to represent the adjacency matrix in a sparse fashion.

Another possible representation for a graph is an adjacency list representation such that for each node in the graph, we have a list of nodes it is connected to (basically there is an edge emanating from the source node connecting the nodes in the adjacency list). In an adjacency list representation, it takes O(E) time to check for the presence or absence of edge between two nodes. Adjacency list is convenient for certain graph traversals like BFS and DFS.

Describing Networks

There are many kind of descriptive statistics to describe networks such as

Indegree is the number of incoming connections for a particular node.

Outdegree is the number of outgoing connections for a particular node.

Using the degree statistics, we can create degree distributions which can be used as modeling tools in many network analysis tasks.

Clustering tries to ascertain how many friends of friends are also friends. This can be done using cycle counting (like triangle counting).

Measures like clustering coefficient can be used as descriptive statistics for discovering subcommunities in the graph, the size of the subcommunities, the density of interactions within the subcommunities and deriving the bounds on interactions between different subcommunities.

Number of connected components is a measure of how many disconnected parts does the network have. The number of connected components can be counted using a graph traversal algorithm like depth first search or breadth first search.

Path Length basically measures the length of the shortest path between two nodes. We can use breath first search for finding the shortest path between two nodes.

To find the shortest path between two nodes we can use breadth first search starting from the source node and terminating the breadth first traversal when we first encounter the destination node.

Triangle Counting is basically counting the number of cycles in the graph of length 3. This can basically be done using done several algorithms. It is also a measure of the number of such nodes, A, B and C in a graph such that A and C, and B and C are connected to each other and also A and B are connected to each other.

We learnt in the class that we should experiment with toy networks initially while creating visualizations. In the class, a road network of a particular city was shown and it was found that a lot of interesting inferences could be drawn from the visualization. One such inference was that were some private gated communities at the edge of the city. Also, from the density of the edges, one could reason about the density of the road network for that part of the city.

We also discussed the Wikipedia voting network, the degree distributions of number of voters. And we inferred that very few people actually vote on Wikipedia.

BFS Pseudocode

breadth first search:

unmark all vertices

choose some starting vertex x

mark x

list L = x

tree T = x

while L nonempty

choose some vertex v from front of list

visit v

for each unmarked neighbor w

mark w

add it to end of list

add edge vw to T

Depth first search:

dfs(vertex v)

{

visit(v);

for each neighbor w of v

if w is unvisited

{

dfs(w);

add edge vw to tree T

}

}

Finding Components:

DFS(G)

{

make a new vertex x with edges x->v for all v

initialize a counter N to zero

initialize list L to empty

build directed tree T, initially a single vertex {x}

visit(x)

}

visit(p)

{

add p to L

dfsnum(p) = N

increment N

low(p) = dfsnum(p)

for each edge p->q

if q is not already in T

{

add p->q to T

visit(q)

low(p) = min(low(p), low(q))

} else low(p) = min(low(p), dfsnum(q))

if low(p)=dfsnum(p)

{

output "component:"

repeat

remove last element v from L

output v

remove v from G

until v=p

}

}

Algorithms for finding mutual friends:

Algorithm:

For each node in the graph, run over all the pairs of its neighbors using an adjacency list. All these pairs of nodes, have at least 1 mutual friend.

This has a time complexity which is sum for all nodes the square of the length of their adjacency list. In a densely connected graph, this algorithm has a very bad time complexity. Also, in case a node is like a celebrity, then it may be connected with majority of the other nodes in the graph. In such a situation, the runtime of the algorithm becomes quadratic or more than quadratic overall in the number of number of the graph which can be very bad for large networks.