Introduction to Graph Concepts for Data Science

Aris Anagnostopoulos

1 Introduction

In this chapter we start by giving some basic definitions from graph theory. This will serve as a refresher and will establish notation for the rest of the text. We then give some definitions that are important for the analysis of networks. Finally we describe the power-law distribution as it appear in several occasions in network mining.

2 Basic Definitions in Graph Theory

In this section we first give some basic definitions of graph theory. We assume that the reader knows basic graph theory and this section is only for reference of the terms and to define notation. Then we define some of the terms that are often used in network analysis.

A *graph* $G = (V, E)$ consists of a set of *nodes* V, and a set of *edges* $E \subset V \times V$. Unless specified otherwise, we assume that $|V| = n$ and that $|E| = m$. Depending on the literature, a node is also called *vertex*, *site*, *actor*, or *agent*. An edge is also called *bond*, *link*, *connection*, or *tie*. A graph can be *directed* or *undirected*. For simplicity, for the rest of the section we deal with undirected graphs, although the definitions can be extended to directed graphs as well. If graph G is undirected then an edge (u, v) is considered an unordered pair, in other words we assume that (u, v) and (v, u) are the same edge. If G is directed then (u, v) and (v, u) are different edges.

If an edge $e = (u, v) \in E$ we say that nodes u and v are *adjacent* or *neighboring*, and that nodes u and v are *incident* with the edge e. Informally, we will often call two adjacent nodes *friends*, or *peers*, or *neighbors*.

A *loop* is an edge from a node to itself: (v, v). Two or more edges that have the same endpoints (u, v) are called *multiple edges*. The graph is called *simple* if it does not have any loops or multiple edges. We will be dealing almost exclusively with simple graphs.

A *path* of length k is a sequence of nodes $(v_0, v_1, \ldots, v_k)$, where we have $(v_i, v_{i+1}) \in E$. If $v_i \neq v_j$ for all $0 \leq i < j \leq k$ we call the path *simple*. If, $v_i \neq v_j$ for all $0 \leq i < j < k$ and $v_0 = v_k$ the path is a *cycle*. A path from node u to node v is a path $(v_0, v_1, \ldots, v_k)$ such that $v_0 = u$ and $v_k = v$.

A *subgraph* G' of a graph $G = (V, E)$ is a graph $G' = (V', E')$ where $V' \subset V$ and $E' \subset E$.

For an undirected graph, the *degree* of a node v (sometimes called *connectivity* in the sociology literature) is the number of edges incident with v and is denoted by d_v. For a directed graph we have the *indegree*, d_v^-, which is the number of edges that go into node v, and the *outdegree*, d_v^+, which is the number of edges that go out of node v.

A *triangle* or a *triad* in an undirected graph is a triplet (u, v, w), where $u, v, w \in V$ such that $(u, v), (v, w), (w, u) \in E$.

Two nodes u and v are *connected* if there is a path from u to v. A graph G is *connected* if each pair of nodes is connected, otherwise we say that the graph is *disconnected*. Any graph can be decomposed into a set of one or more *connected components*, where each connected component is a maximal connected subgraph of G.

A simple graph that does not contain any cycles is called a *forest.* A forest that is connected is called a *tree.* A tree has $n-1$ edges. Actually any two of the following three statements imply that the graph is a tree (and thus they also imply the third one):

1. The graph has $n-1$ edges.

2. The graph does not contain any cycles.

3. The graph is connected.

A *shortest path* (sometimes also called *geodesic path*, or *degree of separation*) between nodes u and v is a path from u to v of minimum length. The *distance* $d(u,v)$ between nodes u and v is the length of a shortest path between u and v. If u and v are in different connected component then $d(u,v)=\infty$.

The *diameter* D of a connected graph is the maximum (over all pairs of nodes in the graph) distance. If a graph is disconnected then we define the diameter to be the maximum of the diameters of the connected components. In other words we define

$$D = \max_{(u,v):u,v \text{ are connected}} d(u,v).$$

The *average diameter* of graph G is the average distance between all the connected nodes of G. Some authors use the term diameter to call this quantity but we avoid that here.

The *effective diameter* is the smallest distance that is larger than 90% of the distances between connected nodes. In other words, it is computed according to the following process: compute the distances between all connected nodes in G, ignore the 10% largest distances, and look at the maximum distance left. This is a quantity often used instead of the diameter as it is more robust with respect to outliers.

Another notion important in the analysis of networks is the *correlation coefficient*, which is a measure of transitivity, that is, a measure of how much do friends of friends tend to be friends. There are a few different variations of the correlation coefficient that capture this concept, but the most commonly used is the following. We define the clustering coefficient of node v C_v to be the ratio of all the edges that exist between the friends of v over all the edges that could possibly exist between the friends of v (see Figure 1). Formally, let us define $\hat{d}_v$ to be the number of nodes different than v that are adjacent to node v; note that for a simple graph $\hat{d}_v$ is just the degree d_v. Then the clustering coefficient (recall that we consider the graph to be undirected) is defined as

$$C_v = \frac{|\{(u,w) \in E : u, w \text{ are adjacent to } v\}|}{\binom{\hat{d}_v}{2}}.$$

Note that if the graph is simple then the denominator equals $\binom{d_v}{2}$, and we have

$$C_v = \frac{2\,|\{(u,w) \in E : u, w \text{ are adjacent to } v\}|}{d_v(d_v-1)}.$$

The clustering coefficient of graph G is denoted by C and is the average clustering coefficient among all the nodes:

$$C = \frac{1}{n}\sum_{v \in V} C_v.$$

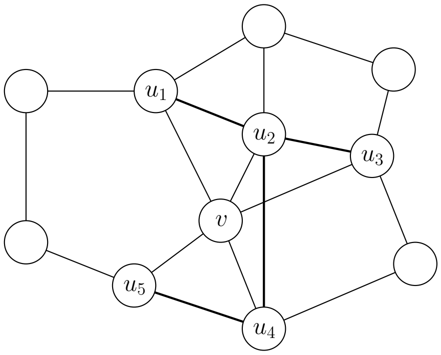

Figure 1: Clustering coefficient of node v. Node v has 5 neighbors and there are 4 edges between those neighbors (the bold edges). Therefore the clustering coefficient of node v is $C_v = \frac{4}{\binom{5}{2}} = 0.4$.

3 Centrality

Another notion that is often used by sociologists at the study of networks is that of *centrality*. As the name implies, centrality measures how central is the node in the graph. Depending on what we mean by "central," there are versions of centrality measure. The most common ones are the *degree centrality*, the *closeness centrality*, and the *betweeness centrality*. We will not be using them here but we describe them for completeness, as they are sometimes found in papers.

The *degree centrality* is the simplest one. The unnormalized one equals to the number of neighboring nodes (the degree in the case of a simple graph). To be able to compare values between different graphs, we define the normalized version, which is normalized by the maximum possible value, $n-1$. So we have

$$\text{Degree centrality of node } v = \frac{d_v}{n-1}.$$

The second notion of centrality, the *closeness centrality*, or just *closeness*, measures how close is the node to the rest of the network. The total distance of node v to the rest of the nodes equals

$$\sum_{u \in V} d(v,u),$$

and since we want the centrality to be large when the distance is small (intuitively a node is central if its distance from the other nodes is small) we take the reciprocal of that. Furthermore, we again normalize so that the value ranges between 0 and 1 by dividing by the maximum possible value, $(n-1)^{-1}$ (which is the value when a single node is connected with $n-1$ other nodes). In other words we define

$$\text{Closeness centrality of node } v = \frac{\frac{1}{\sum_{u \in V} d(v,u)}}{\frac{1}{n-1}} = \frac{n-1}{\sum_{u \in V} d(v,u)}.$$

The third notion of centrality that we define here is the *betweenness centrality*, or just *betweenness*. Assume that two nodes, u and w need to communicate with each other. Then they will ideally use a shortest path. Any node v that is in that path has the ability to affect the communication by distorting it or slowing it down, for example. A node that belongs to a lot of such paths therefore is central in the sense that it can be in the middle and can affect a lot of such communications. That is what betweeness measures.

To define it formally, assume that nodes u and w have g_{uw} shortest paths that connect them (not necessarily disjoint). Then the probability that they use a particular one when they need to communicate is $1/g_{uw}$, assuming that they choose a shortest path uniformly at random among all shortest paths. For a node v define g_{uw}^{v} to be the set of those shortest paths between u and w that contain node v. Then the absolute centrality can be defined as

$$\sum_{u,w \in V \setminus \{v\}} \frac{g_{uw}^{v}}{g_{uw}}.$$

To make it a value between 0 and 1 we normalize it with the maximum value that it can take, which is for the center of the star graph (the graph where a node is connected with the rest $n-1$ nodes and there are no more connections), and in which case the value is $\binom{n-1}{2} = \frac{n^2-3n+2}{2}$. (This is the number of pairs of vertices not including node v, and in the star graph there is a unique shortest path between two nodes and it has to go through the center.) Thus we can define the relative betweeness of a node v as

$$\text{Betweeness centrality of node } v = \frac{\sum_{u,w \in V \setminus \{v\}} \frac{g_{uw}^{v}}{g_{uw}}}{\binom{n-1}{2}} = \frac{2\sum_{u,w \in V \setminus \{v\}} \frac{g_{uw}^{v}}{g_{uw}}}{n^2-3n+2}.$$

This quantity can be computed in polynomial time, the fastest algorithm currently being by Brandes [1] and having running time $O(nm)$, for computing the betweeness of all nodes.

To compare the different version of centrality, degree centrality measures the ability of a node to develop communication. The closeness centrality measures the proximity of a node to the rest of the network, while the betweeness centrality measures in a sense the extent to which a node can control communications in the network.

References

[1] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.