

[Sign In](#)[Get started](#)

Published in Python in Plain English

You have **2** free member-only stories left this month.
[Sign up for Medium and get an extra one](#)



Valentina Alto

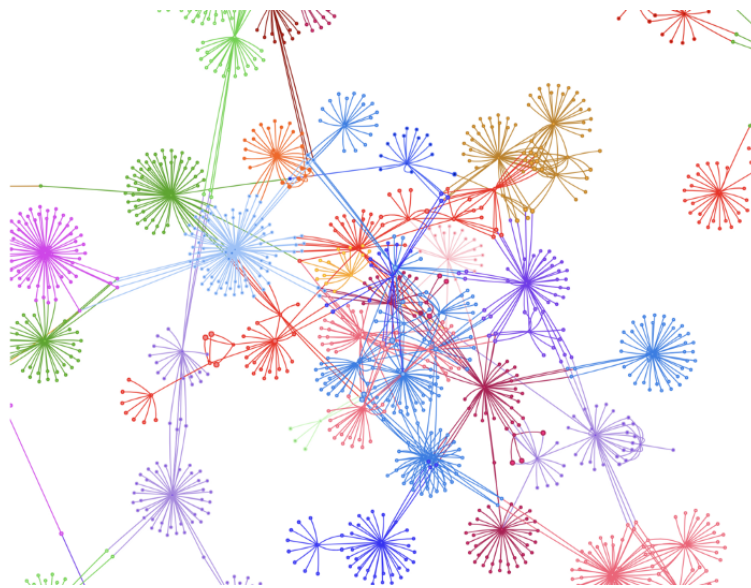
[Follow](#)Apr 11, 2020 · 7 min read · Member-only · [Listen](#)

Image source: <http://carrefax.com/articles-blog/2018/10/14/case-law-network-graph>

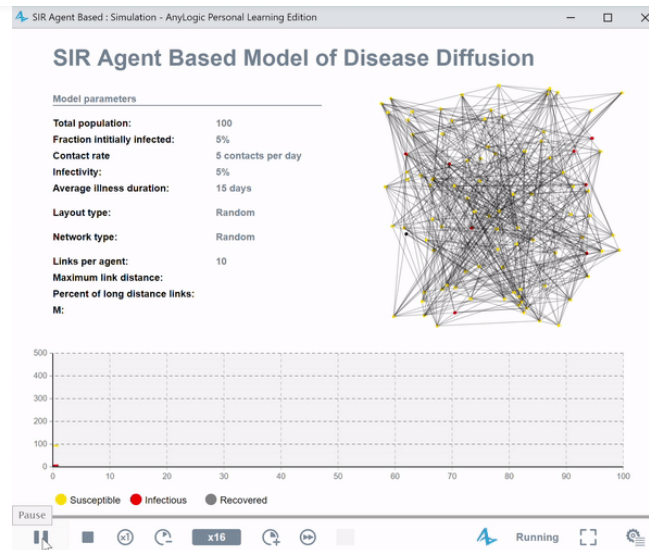
Introduction to Network science with NetworkX

Part 1: Understanding the math behind graph theory

Whenever we have to simulate and analyze complex systems, it is pivotal to use models that are able to capture the relationships among agents. Those relations — and their relative weights and distributions — can be modeled with the mathematical representation of a graph.

Generally speaking, whenever a system can be modeled by a graph, we say that this system is a network. Network science is the discipline whose goal is understanding phenomena whose underlying structure is that of a graph. Namely, in the field of epidemiology, networks are used to simulate the spread of diseases, where nodes are individuals and links represent the contacts among individuals. In the following example, I used a network representation to visualize the dynamics of a [SIR model](#):





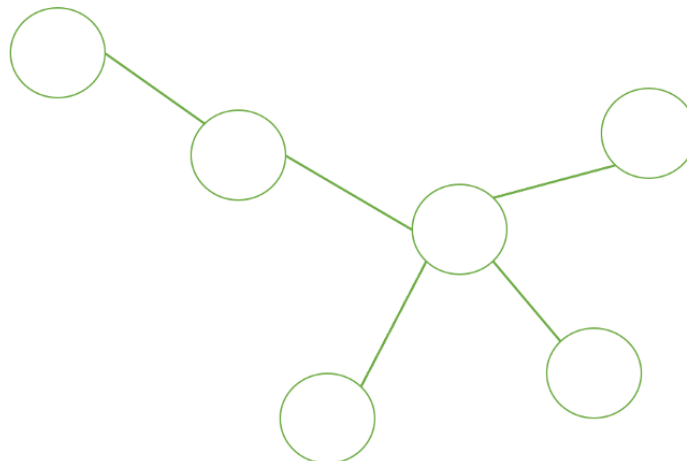
<https://www.anylogic.com/>

In this series of articles, I'm going to go through the math behind a network, its metrics and how to use it to simulate events. Furthermore, I will use the Python package NetworkX to provide visual examples of what I've been explaining in theory and show you different types and shapes of networks.

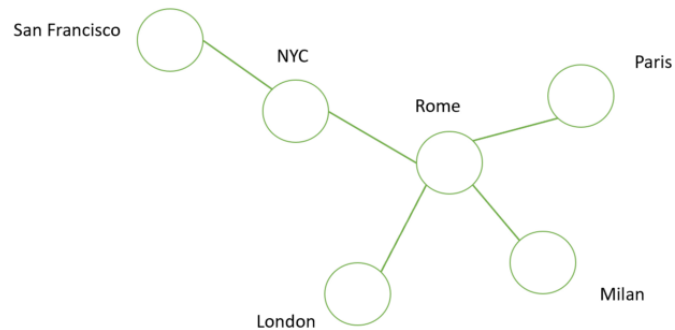
In this first part, I will be focusing on the math behind networks and graphs, providing an introduction to some useful metrics.

The math behind graphs

As mentioned above, graphs are represented by a set of nodes (or vertices) joined by lines defined as links (or edges).



Graphs can describe many relational systems. Namely, the graph above might represent air traffic among different airports:



The links represent direct routes among airports. Namely, to go from Rome to San Francisco, you necessarily need to pass via NYC.

In mathematical language, we can represent this graph with a so-called Adjacent Matrix, described as follows:

	San Francisco	NYC	Rome	London	Milan	Paris
San Francisco	0	1	0	0	0	0
NYC	1	0	1	0	0	0
Rome	0	1	0	1	1	1
London	0	0	1	0	0	0
Milan	0	0	1	0	0	0
Paris	0	0	1	0	0	0

Rows and columns represent nodes: if the element corresponding to the entry i, j is 1, it means that there is a link between nodes i and j , otherwise, the element is equal to 0.

From the graph above, you can see that each route can be taken in any direction (indeed, the adjacent matrix is symmetric). That's hold whenever the graph is *undirected*. On the other hand, if we had such a situation:



you can see that, from San Francisco, you cannot go back to Rome. If this is the case, the adjacent matrix is not symmetric anymore:

Those types of graphs are called *directed*.

Regardless of its nature (directed versus undirected), a graph of size N (that means, a graph with N nodes) can have at most $N(N-1)/2$ edges. A graph with as many edges is said to be **complete**. On the other hand, when the number of edges E is approximately equal to N^a (with $a < 2$), the graph is said to be **sparse**.



I've already mentioned the feature of the *size* of a graph: it is the number of nodes of the system. However, this is not the only metric we can use to describe our graph. We can cluster all of those metrics into three categories:

- distance metrics
- centrality metrics
- statistical properties

So let's examine all of them.

Distance Metrics

The concept of distance can be translated into that of “path” within the network framework. A path is simply the set of links connecting two or more nodes. More interestingly, we can define, for each couple of nodes i, j , the shortest path l_{ij} connecting those nodes. Namely, in the example above (slightly changed for this purpose), the shortest path between London and San Francisco is:

With this measure in mind, we can define the **diameter of the network, which is the maximum value among all the possible shortest paths:**

In the above example, the diameter of the network is equal to 3, since it is the highest value among all the shortest paths. For the sake of clarity, here it is the list of all possible shortest paths:

- San Francisco \leftrightarrow NYC = 1
- San Francisco \leftrightarrow Rome = 2
- San Francisco \leftrightarrow London = 3
- San Francisco \leftrightarrow Milan = 3
- San Francisco \leftrightarrow Paris = 3
- NYC \leftrightarrow Rome = 1
- NYC \leftrightarrow Milan = 2
- NYC \leftrightarrow London = 2
- NYC \leftrightarrow Paris = 2



- Rome \leftrightarrow Paris = 1

Centrality Metrics

Among those measures, we can further differentiate among:

- **Degree centrality:** it measures the number of edges incident on a node i . Namely, in the above picture, the degree centrality of the node Rome is 4. Note that, if we are dealing with directed graphs, each node will have an in-degree and out-degree centrality measure, counting the number of edges arriving at and departing from, respectively, the objective node.
- **Closeness centrality:** it refers to the average distance of a vertex i to all the others and it is computed as:

Where l_{ij} , as defined above, is the shortest path between vertexes i and j . For example, let's compute the closeness centrality for the node Rome:

If we compute that of Milan instead:

We can conclude that Rome is more “central” (aka well connected) than Milan.

- **Betweenness centrality:** this metric assesses the importance of a node in terms of the “bridge” it can create among others nodes. Indeed, one node might be poorly connected, yet it can keep the whole network together. Consider the following example:



Node A has a degree centrality of 2 (pretty low), yet if removed the whole network falls apart. Node A, indeed, is keeping together two clusters of nodes. To take into account this important property, the betweenness centrality comes to aid:

Let's explain this formula:

- σ_{hj} represents the total number of shortest paths between h and j
- $\sigma_{hj}(i)$ is the total number of shortest paths between h and j passing through i .

Intuitively, the greater the number of shortest paths passing through i , the closer this value is to 1. Consider the following graph:

And let's compute the betweenness centrality of node A:



So in that case, the betweenness centrality of node A is equal to 9 (sum of the last column), while its degree centrality is equal to 2 and its closeness centrality is equal to 0.1 ($1/10$).

Statistical properties

We can further characterize a network with a distribution. Namely, we can compute the degree distribution: it is a probability distribution $P(k)$ which tells us the probability of any randomly chosen node to have degree= k . It can be visualized by simply counting the number of nodes having a given degree and plotting the corresponding histogram. Namely, consider the following graph:

The corresponding histogram is:

[Sign In](#)[Get started](#)

As you can see, the most frequent degree among nodes is equal to 2. Note that the same reasoning can be applied to the metric of betweenness centrality, obtaining a *betweenness distribution*.

Final Thoughts

In this first article, I've been providing an introduction to some basic features and metrics to characterize networks. In the next articles, I'll introduce the Python package NetworkX and differentiate among different types of graphs.

Hoping you enjoyed the reading, stay tuned for [Part 2!](#)

 171  1




More from Python in Plain English

[Follow](#)


New Python content every day. Follow to join 500k+ monthly readers.

[Read more from Python in Plain English](#)

Recommended from Medium

 Sanskritijain

Let's Grow More Virtual Internship Experience

 Chris Prusakiewicz


Research Project Management, part 3: Data Collection and Analysis



 Kantida ... in Web Mining [IS688, ...

Clustering a privacy concerns level of online media users across social media platforms



 Shilpa Leo in Towards Data Science


Battle of the Auto ML titans for People Analytics application



 Jinav Gala

Fast Data Science- utility



 Aaron Fred... in Towards Data Sci...

How to Transfer SQL Knowledge to R



 maura.cerow



 Frederick Bott





[Sign In](#)

[Get started](#)



[About](#) [Help](#) [Terms](#) [Privacy](#)

