Identifying influential nodes in film actor communities

Joan Sánchez García
University of Edinburgh
J.Sanchez-Garcia-1@sms.ed.ac.uk
STN Project, Fall 2018

Abstract

Today we live in a society submerged in social networks, massive structures of information that seem impossible to treat and study. That is why it is necessary to find basic properties that allow us to locate what we are looking for, quickly and effectively. In this project we use an extract from the IMDB database to study if it is possible to easily find the most influential nodes within the communities of a graph. For this, a specific definition of "influence" will be used, along with the community detection algorithm k-means and the ranking algorithm PageRank. All the code and the scripts to plot the data developed for this article are available here [4].

Problem statement

The intention of this project is to try to find the most influential nodes within a community from the representation of the graph in the form of an adjacency matrix.
The influence of a node on the graph is based on the impact of its fame on the set of nearby nodes. That is why different methods to calculate this influence are studied in this article.
The importance of this project is based on the rapid growth of networks and the need to exploit their information. Finding the most important node of a community is a vital task in many fields, including the world of cinema. To date, there are many studies that are being carried out on this subject. The challenge is to find a simple solution.
This project uses an excerpt from the IMDB database with the intention of locating those actors that are important within their community. This location can help us to focus attention centers, either to improve the casting of a movie, its probability of generating more success at the box office…
See Annex 1 to consult the structure of the database that will be used in this project.

Related work

Networks have ceased to be a laboratory tool to become one of the foundations of modern life. As a result, it is not surprising that recent years have witnessed an explosion of research on community structure in graphs, and a huge number of methods or techniques have been designed [1].
In Yang Wang, Zengru Di and Ying Fan paper [2], they introduce a new approach to characterize the node importance to communities. First, they propose a centrality metric to measure the importance of network nodes to community structure using the spectrum of the adjacency matrix. The node importance is defined to communities as the relative change in the eigenvalues of the network adjacency matrix upon their removal.
In this study, we will also use the adjacency matrix to find the important nodes, but we will not use the eigenvalues. We will use some of the fields of the data on which we will work to give a weight to each node. This weight will be used in algorithms such as PageRank.
In Malliaros, Rossi and Vazirgiannis work [3], they capitalize on the properties of the K-truss decomposition, a triangle-based extension of the core decomposition of graphs, to locate individual influential nodes. Unlike their work, decomposition doesn't apply in this.

The solution

First approach: use the adjacencies of the nodes The first approach to the solution is to use the degree of each node to determine its importance within the community to which it belongs in the graph. Since this is a highly connected graph, each node has a way to reach any other, therefore, the node with the most adjacencies will be the one that has more facility to reach the others, given that it will know different routes, and therefore, could be the most influential.
To find these nodes, we generate the adjacency matrix by extracting the information from the database. The adjacency matrix tells us, for each actor, the id of the actors with whom he/she has worked. Once we have calculated it, we apply the k-means algorithm with random initial nodes, since we are only interested in finding communities so we can have more than one result per execution (if the entire graph was a single community, we would find a single "maximum influential " node. In this way, if we ask to calculate 4 communities, we can compare in blocks of four if the results are the desired ones).
For each of the communities found, we look in the adjacency matrix which is the node with the most adjacencies and compare it with the validator.

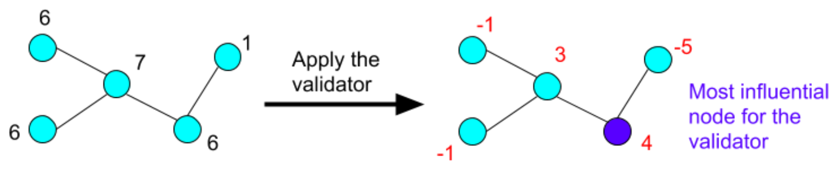
Second approach: PageRank The second approach will be the PageRank algorithm. We will test two versions of this algorithm: one assigning a PageRank of $1/(numbers\ of\ actors\ in\ the\ system)$ to each actor *n*. The other, assigning $(\sum_{i=0} films'\ fame(n))\ /\ total\ fame\ of\ the\ system$. Iteratively, the actor *n* will distribute his current fame in an equitable manner among all the nodes with which he is connected, and he will update his with all the portions that arrive to him.
As our graph is not acyclic, after a few iterations of applying the standard definition of the algorithm, all the PageRank of the system would remain locked in the cycles. To prevent this from happening, we apply scaled PageRank, which scales each of the values of the nodes on a *s* scale and distributes (1-s) among all the nodes of the system, helping the PageRank transfusion does not stagnate.
In this way, when the algorithm stabilizes after *x* iterations, the fame of each node will represent the level of influence of this. With this method, not only the degree of a node will be taken into account (PageRank assigns the same initial value to each node), but we introduce different initial values to each node, so we vary the definition of "important node".

Third approach: k-means initial nodes Although unfortunately we will not be able to introduce this part of the study due to lack of time, it could also be interesting to execute the k-means algorithm choosing as initial nodes those provided by the validator, that is, those that our validation algorithm considers are the most appropriate and see how the final result varies.

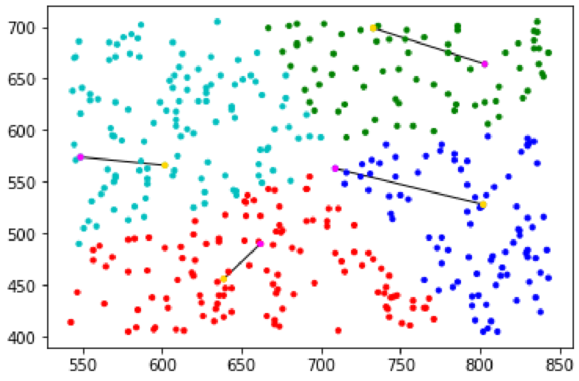
Validator Our validation algorithm performs a simple comparison between the fame of a node and that of all its neighbors. For a node *n*, we would calculate its fame as fame(n)= $\sum_{\forall neighbour(n)\ i} fame(n) - fame(i)$. In this way, a node will be considered more influential if it has worked in films with people less famous than him. In the same way, if an actor works with someone with more fame, his influence can be negative.

Results

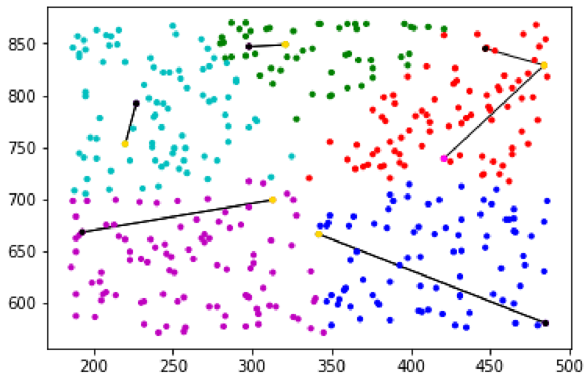
The intention of the validator is to have a fast and minimally effective method that allows us to find the values we are looking for in order to compare the results with other more complex methods. In our case, the idea of the validator that we use comes from the definition of influence "to cause an effect (on someone)" [5]. An actor will be important if it can directly impact the actors with whom he works. Therefore, a famous actor who works with a single not known person will be more influential than someone who works with many people of equal fame. This definition takes us away from the conventional idea that someone who is influential is simply connected to many people. In our case, the validator would give the following answer:



Our first approach is to find the nodes with the highest degree within a community. In the example on the left, we see the 4 communities of the graph, the nodes with a greater degree within them (in fuchsia) and the nodes that the validator provides us (in gold). As we can see, the two answers are far from being considered close. This happens for the same reason that we explained in the previous example: the importance of an actor is not determined by the number of actors with whom he has worked, but by the impact that may have had on them. It is for this reason that the method of choosing the node with the highest degree will not be a guarantee of choosing the most influential node



Our second approach is to calculate the influence of a node using PageRank. The first thing that we realize is that if we apply PageRank in the standard way (assigning 1 / n fame to each node), the algorithm finds that the most important nodes are also those that have a higher degree, since they are the ones that have more chances of receiving the PageRank (our graph is not directed). In addition, we will not be using the concept of influence.



An idea to compensate this, is to make that nodes start with a different PageRank depending on the fame of each actor. To achieve this, and at the same time ensure that the total amount of PageRank of the system does not vary, each actor is assigned actors' fame/total fame in the system.
Even so, PageRank tends to accumulate again in the actors with greater degree, it simply takes more iterations to stabilize. As in the first approach, this method would not bring us closer to the desired result. It is not a surprise that on many occasions, the node with the highest degree and the one with the highest PageRank are the same. In the example itself, it occurs in ¾ communities.

So what's wrong? As we have seen, what fails is our definition of influence, which has nothing to do with how the network is constructed. We are asking the graph to return us information that has not been needed to be built, it is secondary.
To look for this information only taking into account the external form and the properties of the network, as we are doing in this work, we should have built it weighing on the influence. In our case, an edge was created if two actors had collaborated, regardless of the differences in fame between them.
Therefore, it will be difficult to find influential nodes with the definition that we have given, given that this does not address neither the topology nor the intrinsic properties of the network.

Bibliography

[1] Community detection in graphs:
https://www.sciencedirect.com/science/article/pii/S0370157309002841
[2]Identifying and Characterizing Nodes Important to Community Structure Using the Spectrum of the Graph: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0027418
[3]Locating influential nodes in complex networks:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4725982/
[4]Code developed for this article: https://github.com/joansanchez/STN
[5]Definition of influence: http://www.wordreference.com/definition/influence

Annex 1

The database consists of 3 files: name.basics.tsv, which contains the id of the actor and the movies in which he/she has participated, title.basics.tsv, which contains all the information related to a movie and title.ratings. tsv, which includes the ratings and the number of votes of the films in the database.

Index	Field	Type	Examples/Notes
			name.basics.tsv
0	Name actor	str	Unique person/crew ID
1	Known for titles	list[str]	‘tconst1,tconst2,tconst3’
			title.basics.tsv
0	Film name	str	
1	Title type	Optional[str]	
2	Primary title	Optional[str]	
3	Original title	Optional[str]	
4	Is adult?	int	
5	Start year	Optional[int]	
6	End year	Optional[int]	
7	Runtime minutes	Optional[int]	
8	Genres	Optional[List[str]]	
			title.ratings.tsv
0	Name film	str	
1	Average rating	float	
2	Num votes	int	