UNIVERSITÀ DEGLI STUDI
DI GENOVA

Department of Computer Science

GA - Graph Analytics A.Y. 2018/2019

Analysis of a Realistic Graph

Andrea Canepa (S 4185248)

I like to know why a video has suddenly gone viral, why a song has broken, why a TV show is suddenly rating out of pattern... I'm pretty good at understanding why things are becoming popular

Simon Cowell

4 April 2020

Contents

1 Introduction and Dataset Presentation

The graph represents the network established between Facebook pages of American Tv Shows. It could be considered a ***social network***.
The nodes represent "blue" verified Facebook pages and exists an edge from node i to node j if page i has a like to page j and vice versa. It is a kind of *mutual relationship* the one between the nodes.
The data goes back to November 2017. The downloaded zip presents two files:

- `fb-pages-tvshow.nodes`: contains the mapping between node number and title of the tv show;
- `fb-pages-tvshow.edges`: contains the edges that constitute the graph.

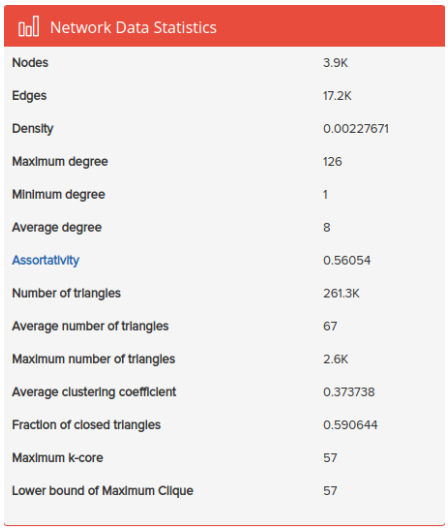


Figure 1: Dataset information

The dataset can be found at

`http://networkrepository.com/fb-pages-tvshow.php`

a very well organized site, thick of datasets subdivided into a lot of categories. My work is entirely written in ***Python3*** language and relies on two fundamental libraries:

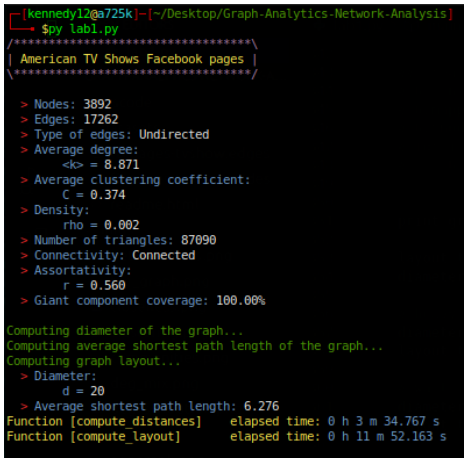
- `networkx` (v. 2.4): to handle everything related to the graph world;
- `matplotlib` (v. 3.1.2): to report data in a user-friendly graphical way.

The script that I have delivered permits two different execution modes, depending on the command-line arguments:

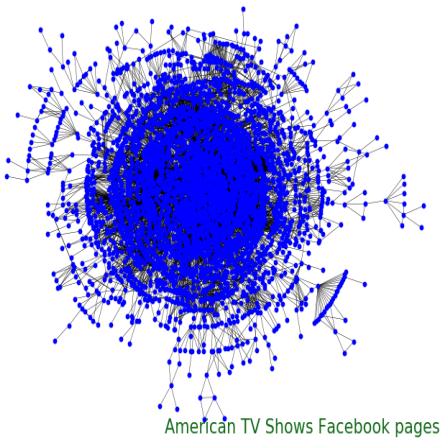
- ***Analysis*** (*no arguments*): it computes metrics, extracts layout and communities from the graph;
- ***Query*** (`--interactive`): gives the user the opportunity to understand which node corresponds to which show, to give a sense to the results computed before.

2 Graph Measures

In this section I will provide an analysis of the graph structure starting from values extracted computing some centrality measures we have seen during the course. First of all, I would like to show an overview of the *dimensions* of the graph in the following pictures, taken after execution in *analysis* mode.



(a) General data about the graph



(b) Representation in `kamada-kawai` layout

Figure 2: Graph representations

Starting from the picture (2.a), we can see that the network consists of 3892 *nodes* and 17262 *edges*. Since the relationship is *mutual*, the network is ***undirected***. Furthermore, we can see that the **giant component** corresponds with the graph itself, since its *coverage* is 100%. We can note that the number of *triangles* differs from the one reported by the site since the latter count each vertex of the triangle, so the real number of them is obtained dividing by 3 that amount.

2.1 Density

It is known that the maximum number of edges in a *dense* graph is $L_{tot} = \frac{N(N-1)}{2}$, where N is the number of nodes of the network. To understand if the graph is ***dense***, I have to compare the number of actual links, L, w.r.t. the previous quantity:

$$\boxed{\frac{L}{L_{tot}} = \frac{2L}{N(N-1)} = \frac{2 \cdot 17262}{3892 \cdot (3892-1)} = \frac{34524}{15143772} = 0.002}$$

Hence we conclude that the network is mostly **sparse**, in fact, $L \ll L_{tot}$. This is also an indicator of the *scaling-free* nature of the graph.

2.2 Degree Distribution

In this section we can see how the ***degree distribution***, p_k, which indicates the probability that a randomly chosen node has degree k. The formula is the following:

$$\boxed{p_k = \frac{N_k}{N}}$$

where N_k is the number of nodes in the network with degree k, while N is the total number of nodes.

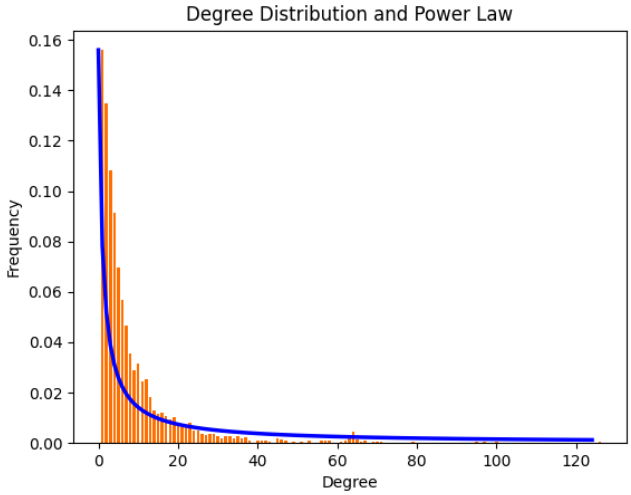


Figure 3: Degree distribution in the network

As we can notice from figure 3, the *histogram* (in orange) tells us that there are few nodes with high degree, which are the ***hubs***, and a lot of nodes with low connectivity. The shape seems to follow the **power-law** distribution, drawn in blue.
To conclude the discussion about this measure, let me show some statistics of the *degree distribution* in my graph:

```
##########################
### Degree distribution ###
##########################
        +++
        |- Maximum: 126 --> ( Queen of the South )
        |- Minimum: 1 --> ( TVR )
        |- Average: 8.870503597122303
        |- Variance: 157.67644753672909
        +++
```

Figure 4: Degree distribution statistics

2.3 Distances and Layout

Here I would like to briefly take a look at two characteristics of the graph. Since both of them required some heavy computation, I decided to perform them in parallel, exploiting the ***multithreading*** features of the *multiprocessing* library of `Python3`.
I decided to compute the graph layout, to give it a prettier shape in figures, following the *Kamada* and *Kawai* cost function, based on an *optimization* problem, in particular, a *minimization* of a cost function.
Regarding the *distances*, I computed the graph ***diameter***, which is the dimension of the greatest shortest path between each couple of nodes, along the ***average shortest path***. Below, in figure 4, we can see the results.

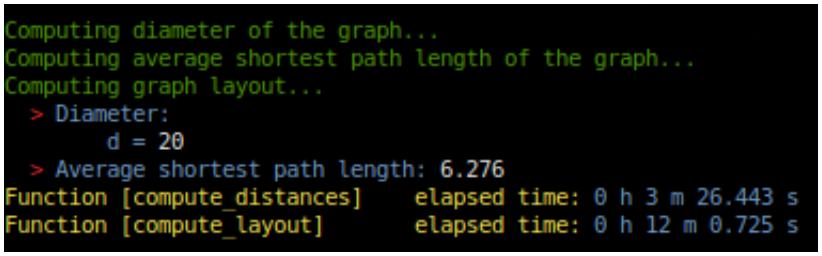


Figure 5: Layout and distances computation

A peculiar fact to notice is that, even the network is relatively small w.r.t. some real networks subject to study, the computation of the results took quite a lot of time, regardless of the fact that I used the optimized library version of the algorithm.
The *ratio* of the latter is near 0, the shortest paths are smaller than the graph size.

$$\boxed{\frac{\bar{d}_{i\neq j}(i,j)}{d} = \frac{6.276}{20} = 0.3138}$$

2.4 Clustering

This is a *local* property of the graph, relative to a node and its neighborhood.

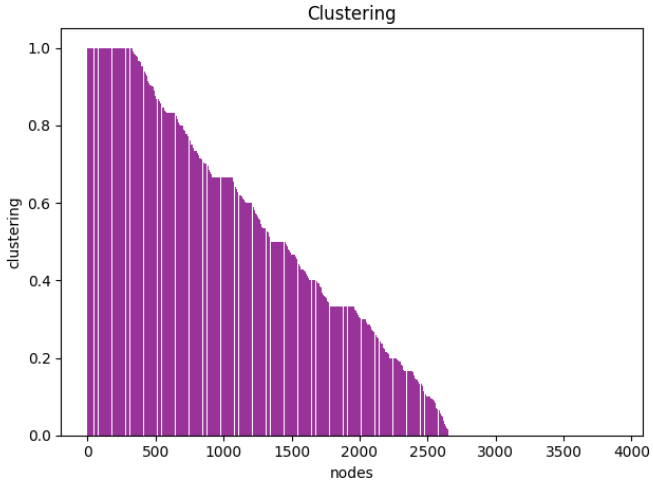


Figure 6: Clustering coefficient in the graph

```
#################
### Clustering ###
#################
        +++
         |- Maximum: 1.0 --> ( Team Wallraff - Reporter undercover )
         |- Minimum: 0 --> ( Twoja twarz brzmi znajomo )
         |- Average: 0.37373843245974014
         |- Variance: 0.12521307515304766
        +++
```

Figure 7: Clustering coefficient statistics

This measure is strictly connected with the ***density***, discussed before, as it shows us how many *structural holes* are present in the graph. It is worth remembering that values near to 1 indicate that those nodes are strongly connected, whilst values near to 0 indicate hubs. Moreover for unweighted graphs, the ***clustering*** of a node i is the fraction of possible *triangles* that exist through that node,

$$\boxed{c(i) = \frac{2T(i)}{k_i(k_i - 1)}}$$

where $T(i)$ is the number of triangles in which the vertex i appears and k_i is the degree of the node i.

2.5 Betweenness Centrality

Starting from this section, the pictures show only the values for the 25 most relevant nodes w.r.t. the measure I am presenting.

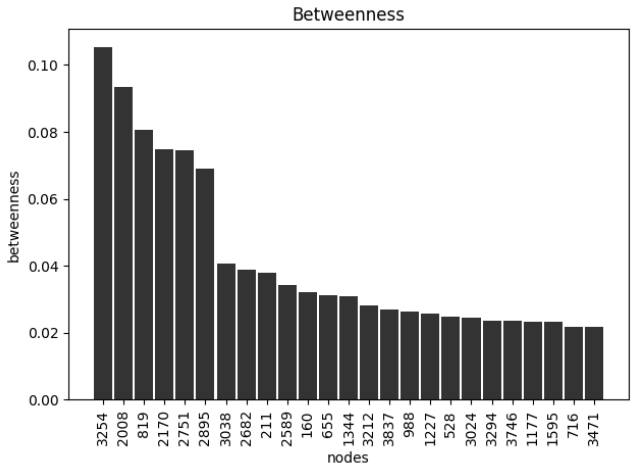


Figure 8: Betweenness centrality in the graph

```
##################
### Betweenness ###
##################
        +++
         |- Maximum: 0.10544488181477074 --> ( Queen of the South )
         |- Minimum: 0.0 --> ( Living Biblically )
         |- Average: 0.0013562754354734274
         |- Variance: 2.0327998076309795e-05
        +++
```

Figure 9: Betweenness statistics

I would like to briefly recall the formula of the ***betweenness centrality***:

$$C_B(i) = \sum_{s,t \in V} \frac{\sigma(s,t \mid i)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t \mid i)$ is the number of those paths passing through some node i other than s and t.

It is quite evident that the values of ***betweenness*** decrease rapidly, showing that only a fraction of the nodes is in the middle of the *communication line* of the network. Those are the ***hubs***. This is another clear evidence of the **power-law** distribution.

2.6 Closeness Centrality

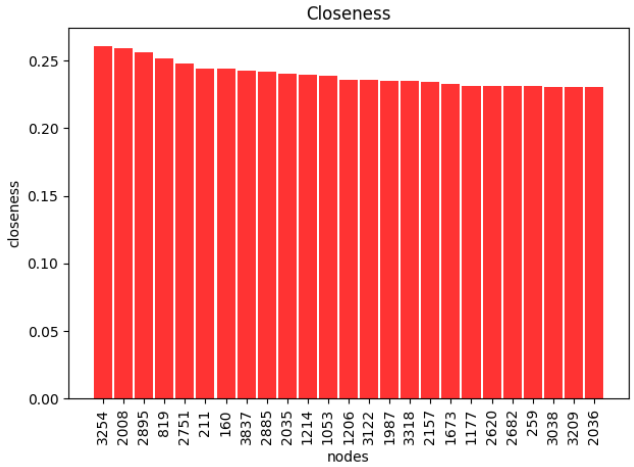


Figure 10: Closeness centrality in the graph

```
################
### Closeness ###
################
        +++
         |- Maximum: 0.2609832986786505 --> ( Queen of the South )
         |- Minimum: 0.07677736340496064 --> ( Rischiatutto )
         |- Average: 0.16521277827746084
         |- Variance: 0.0008839630877975622
        +++
```

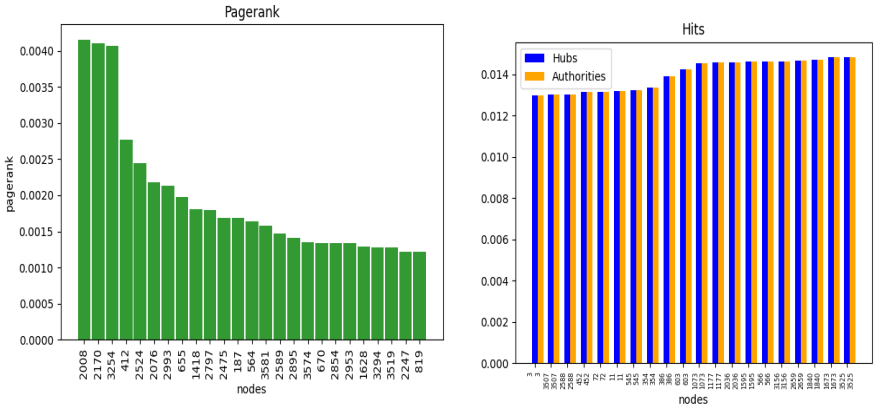
Figure 11: Closeness statistics

The function in the library `networkx`, `nx.closeness_centrality()`, presents a slightly different implementation of the ***closeness*** formula that we have seen during the course. It is due to an optimization proposed by Wasserman and Faust, which is:

$$C_C(i) = \frac{n-1}{N-1} \frac{n-1}{\sum_{j=i}^{n-1} d(i,j)}$$

where $d(i,j)$ is the *shortest path distance* between i and j, n is the number of reachable nodes and N is the cardinality of the nodes set. This formulation of the problem takes into account the number of reachable nodes: since the graph is fully connected and exists only one component, this is reduced to the base case of the problem. From figure 8, we can evince that there are a lot of nodes next to the *"center"* of the graph, all near since the component in which they reside is the same.

2.7 PageRank and HITS



(a) Pagerank (b) HITS

Figure 12: Pagerank and HITS in the graph

Figure 13: Pagerank and HITS statistics

In this section I will talk about *link analysis* techniques.

PageRank algorithm shows us some information related to the *importance* that a node has in the network. Similarly to the Google Matrix, I chose the ***dumping factor***, α, to be 0.8. A node has a higher value of *PageRank* if nodes with a high value of *PageRank* are linked with it.

In figure 10.b, as well as in figure 11, we can see that the value of **hub** and **authority** for the nodes are the same since the network is undirected. The presence of some ***hubs*** in the graph is, once again, a symbol of the presence of the **power-law** distribution.

3 Assortativity and Communities analysis

3.1 Assortativity

An important value that describes the nature of the network is the **assortativity**. In order to understand better to whom the nodes tend to link each other I computed the ***degree correlation matrix***. Since the graph is undirected, it is *symmetric*, as expected. In figure 14, we appreciate the fact that nodes with comparable degrees connect together. This is the common behaviour of ***assortative*** networks, according to the value computed (figure 2.a) using the following formula, referring to the documentation of the `networkx` library:

$$\boxed{r = \frac{\sum_{x,y} xy(e_{xy} - a_x b_y)}{\sigma_a \sigma_b}}$$

Standard Pearson Coefficient

where a_x and b_y are, respectively, the fraction of edges that start and end at vertices with values x and y (on an undirected graph, $a_x = b_x$), σ_a and σ_b are the standard deviations of the distributions a_x and b_y and e_{xy} is the joint probability distribution (mixing matrix) of the degrees.

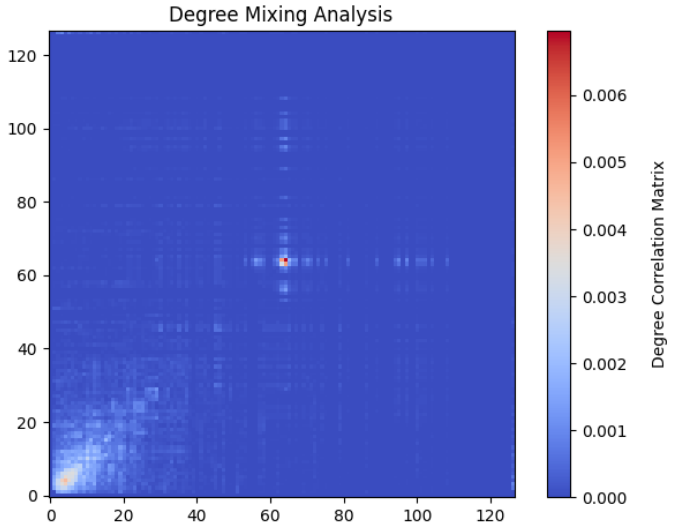


Figure 14: asd

As a matter of fact the *assortativity* value for the graph is greater than zero, $r = 0.56$, underlining one more time that the network is ***social***. This result could be interpreted as a sign of the presence of **communities**, which is the topic of the next discussion.

3.2 Communities

An interesting topic to study, while treating *social networks*, is the ***composition of communities***. I decided to explore my data with the algorithm proposed by *Clauset, Newman* and *Moore* for ***community detection***. Then I would like to discuss briefly the statistics of the performance obtained.

3.2.1 Clauset-Newman-Moore algorithm

This is a *greedy* algorithm that performs an *agglomerative clustering* technique based on a quantity called ***modularity***, Q. The procedure tries to join two subsets of nodes if this leads to a modularity increment, i.e. $\max_{\Delta Q} \Delta Q > 0$.

There are 3 steps:

1. starts with as many communities as nodes, that is one node per community;
2. tries to agglomerate each pair of clusters and compute ΔQ, select the pairs for which ΔQ is maximal and actually join them;
3. goto (2) until one community remains.

For the sake of clarity, here I show the formula of **modularity**:

$$\boxed{Q = \frac{1}{2|E|} \sum_{i,j \in V} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \delta(c_i, c_j)}$$

where $|E|$ is the cardinality of the set of the edges, A_{ij} is the adjacency matrix, V is the set of vertices, k_i is the degree of the node i, c_i is the community of the node i and $\delta(a, b) = 1$ if $a = b$, 0 otherwise.

3.2.2 Evaluation of communities

Here I show some indicators obtained during the community analysis. First of all the graph hosts 59 communities of different sizes. Then, an interesting value of the modularity, $Q \simeq 0.83 \geq 0.3$, tells us that the network has a *significant community structure[1]*.

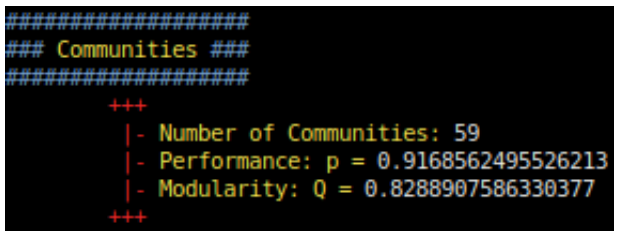


Figure 15: Statistics of communities

To evaluate the goodness of the result I chose to compute the **performance** of the partition, which counts the number of correct *allocations* of pairs of vertices, i.e. two vertices belonging to the *same* community and connected by an edge, or two vertices belonging to *different* communities and not connected by an edge.

$$P(\rho) = \frac{|\{(i,j) \in E, C_i = C_j\}| + |\{(i,j) \notin E, C_i \neq C_j\}|}{\frac{N(N-1)}{2}}$$

By definition $0 \leq P(\rho) \leq 1$. Since, in my case, $P(\rho) \simeq 0.92$, I can conclude that the partition is very accurate. Finally I would like to end the presentation showing in a colorful way the *distribution* of communities in the network.

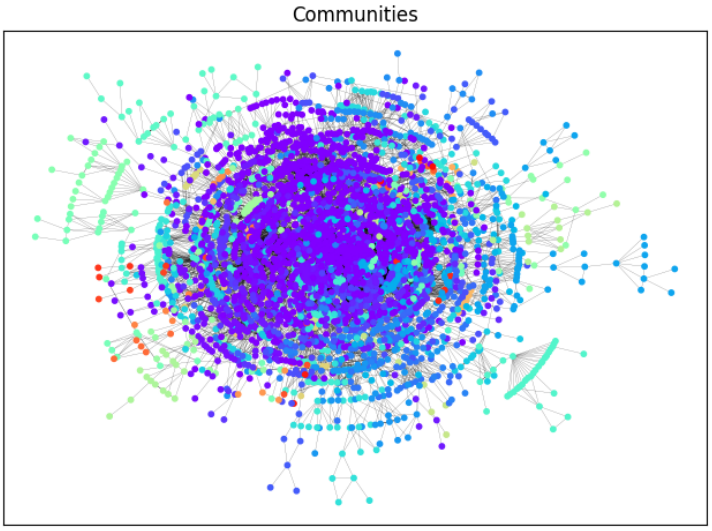


Figure 16: Graphical representation of communities in the network

4 Example: Query mode

Those are some of my all-time favourite shows and important nodes in the graph. To terminate the interactive session is sufficient to type *END*.

```
┌─[kennedy12@a725k]─[~/Desktop/Graph-Analytics-Network-Analysis]─{*CTF3*}
└─► $py lab1.py --interactive
Tell me which node you want to know [0-3891] (END to exit):     819
Node 819 is the show:   The Tonight Show Starring Jimmy Fallon
Tell me which node you want to know [0-3891] (END to exit):     2036
Node 2036 is the show:  Brooklyn Nine-Nine
Tell me which node you want to know [0-3891] (END to exit):     3887
Node 3887 is the show:  Unbreakable Kimmy Schmidt
Tell me which node you want to know [0-3891] (END to exit):     3209
Node 3209 is the show:  The Office
Tell me which node you want to know [0-3891] (END to exit):     3440
Node 3440 is the show:  Sherlock
Tell me which node you want to know [0-3891] (END to exit):     1053
Node 1053 is the show:  Parks and Recreation
Tell me which node you want to know [0-3891] (END to exit):     1723
Node 1723 is the show:  The Late Late Show with James Corden
Tell me which node you want to know [0-3891] (END to exit):     3254
Node 3254 is the show:  Queen of the South
Tell me which node you want to know [0-3891] (END to exit):     2751
Node 2751 is the show:  The Voice
Tell me which node you want to know [0-3891] (END to exit):     2524
Node 2524 is the show:  Good Morning America
Tell me which node you want to know [0-3891] (END to exit):     2008
Node 2008 is the show:  Home & Family
Tell me which node you want to know [0-3891] (END to exit):     1923
Node 1923 is the show:  Family Feud
Tell me which node you want to know [0-3891] (END to exit):     END
┌─[kennedy12@a725k]─[~/Desktop/Graph-Analytics-Network-Analysis]─{*CTF3*}
└─► $
```

Figure 17: Example of a session in query mode

Figure 18: Tv-shows

References

[1] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[2] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2), Feb 2003.

[3] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[4] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019*, pages 65–72. ACM, 2019.