Community detection on the Wikipedia hyperlink graph

Armand Boschin, Bojana Ranković, Quentin Rebjock

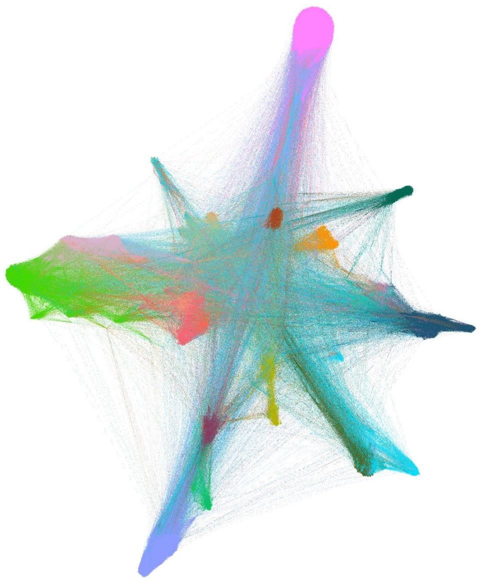
January 25, 2018

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

WIKIPEDIA
The Free Encyclopedia

Table of contents

1) Data acquisition
2) Modelisation of the network
3) Community detection
   a) Louvain algorithm
   b) Spectral clustering
4) Visualization



1) Data Acquisition

Disambiguation pages

WIKIPEDIA
The Free Encyclopedia

Not logged in Talk Contributions Create account Log in

Article Talk

Read Edit View history Search Wikipedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export

Create a book
Download as PDF
Printable version

Languages

Jaguar (disambiguation)

From Wikipedia, the free encyclopedia

A **jaguar** is a large cat native to South and Central America.

Jaguar may also refer to:

Contents [hide]

1 Transportation
2 Entertainment
2.1 Cartoons, comics and print
2.2 Music
2.3 Film
2.4 Other
3 Science and technology
4 Sports
5 Military and weapons
6 Other uses
7 See also

Transportation [edit]

- Jaguar Cars
- Armstrong Siddeley Jaguar, an aircraft engine
- SS *Empire Ballad*, a Panamanian steamship built in 1941 and renamed SS *Jaguar* in 1962

Entertainment [edit]

Cartoons, comics and print [edit]

- Jaguar (Marvel Comics), a SHIELD supervillain
- Jaguar (Insurgent Comix), a superheroine
- Jaguar (Archie Comics), a comics character
- Jaguar (cartoonist), Sérgio Jaguaribe (born 1932) from Brazil
- *Jaguar* (novel), by Roland Smith

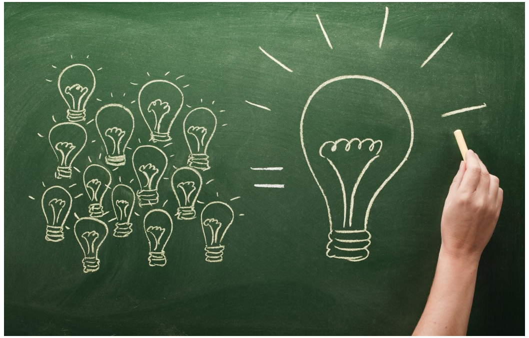
Music [edit]

- Jaguar (band), a new wave British heavy metal band
- Jagúar (band), an Icelandic funk band
- Jaguar (Kenyan musician), pseudonym of Charles Njagua Kanyi
- The Jaguars, an American doo wop group known for their 1956 cover of "The Way You Look Tonight"
- The Jaguars, a 1960s Japanese band featured on the *Banzai!* TV series
- Fender Jaguar, a guitar introduced in 1962
- "Knights of the Jaguar", often shortened as "Jaguar", a techno song by DJ Rolando

Wikipedia API

```
In [1]: import wikipedia

In [2]: wikipedia.page('Jaguar (disambiguation)')

/usr/local/lib/python3.5/dist-packages/bs4/__init__.py:181: UserWarning: No parser was explicitly specified, so I'm u
sing the best available HTML parser for this system ("lxml"). This usually isn't a problem, but if you run this code
on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 193 of the file /usr/lib/python3.5/runpy.py. To get rid of this warning,
change code that looks like this:

 BeautifulSoup(YOUR_MARKUP})

to this:

 BeautifulSoup(YOUR_MARKUP, "lxml")

  markup_type=markup_type))

---------------------------------------------------------------------------
DisambiguationError                       Traceback (most recent call last)
<ipython-input-2-23d0ceef0c05> in <module>()
----> 1 wikipedia.page('Jaguar (disambiguation)')

/usr/local/lib/python3.5/dist-packages/wikipedia/wikipedia.py in page(title, pageid, auto_suggest, redirect, preload)
    274         # if there is no suggestion or search results, the page doesn't exist
    275         raise PageError(title)
--> 276     return WikipediaPage(title, redirect=redirect, preload=preload)
    277   elif pageid is not None:
    278     return WikipediaPage(pageid=pageid, preload=preload)
```

Strategy



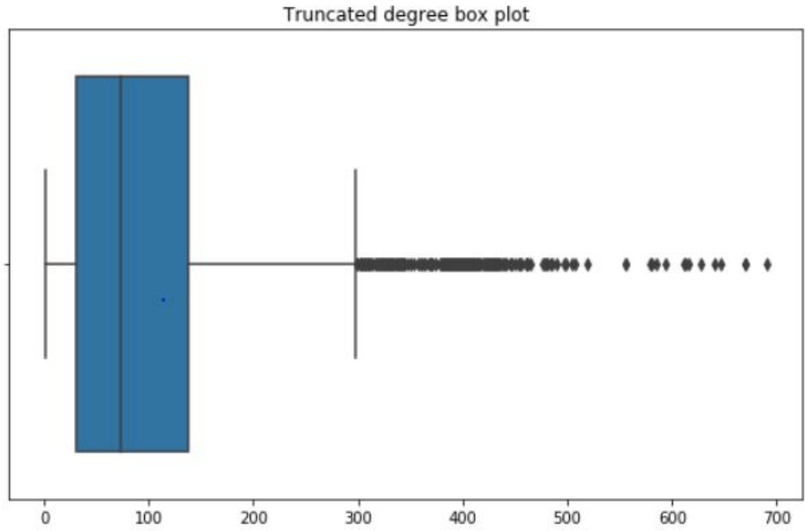
- Start from root node ‘Jaguar (disambiguation)’
- Explore neighbors (first nodes)
- Explore neighbors of first nodes (second nodes)
- Get inner connections

For each node :

- Title
- Neighbors
- Categories (need to be cleaned from “All wikipedia pages with … “
- URL

Principal properties of the collected network

Property	Value
Nodes	6830
Edges	367483
Clustering coefficient	0.643
Size of the largest giant component	6830/6830
Diameter	5
Average degree	107.5



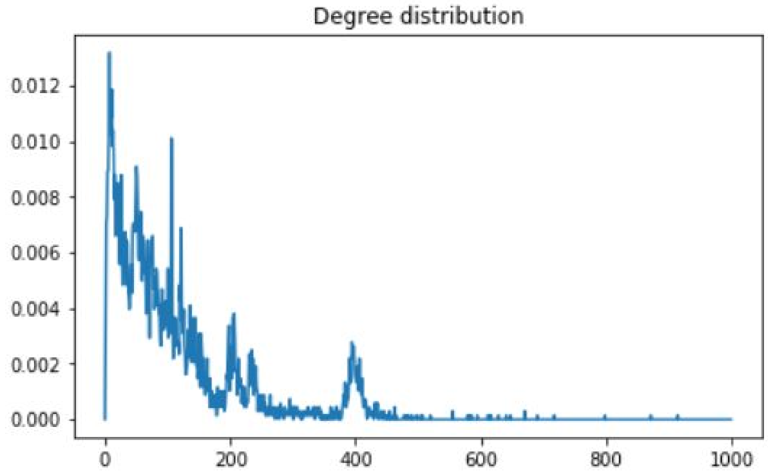
2) Modelisation of the network

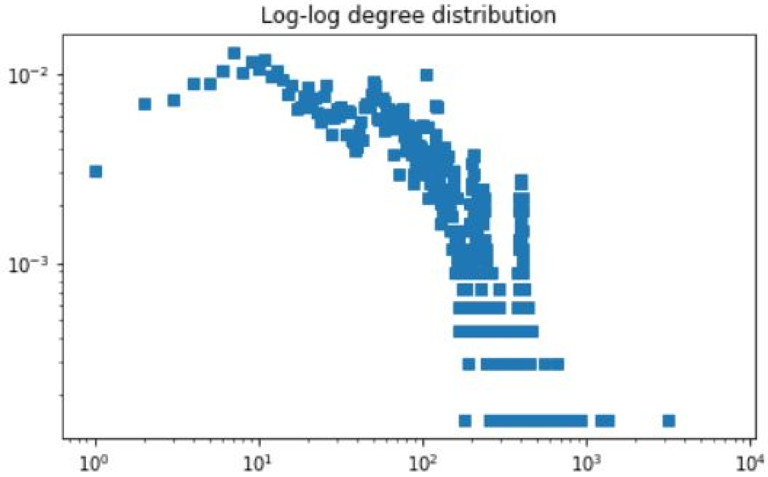
Find a simple network with similar properties

- Approximate the number of nodes/edges, the degree distribution, the clustering coefficient and the giant components

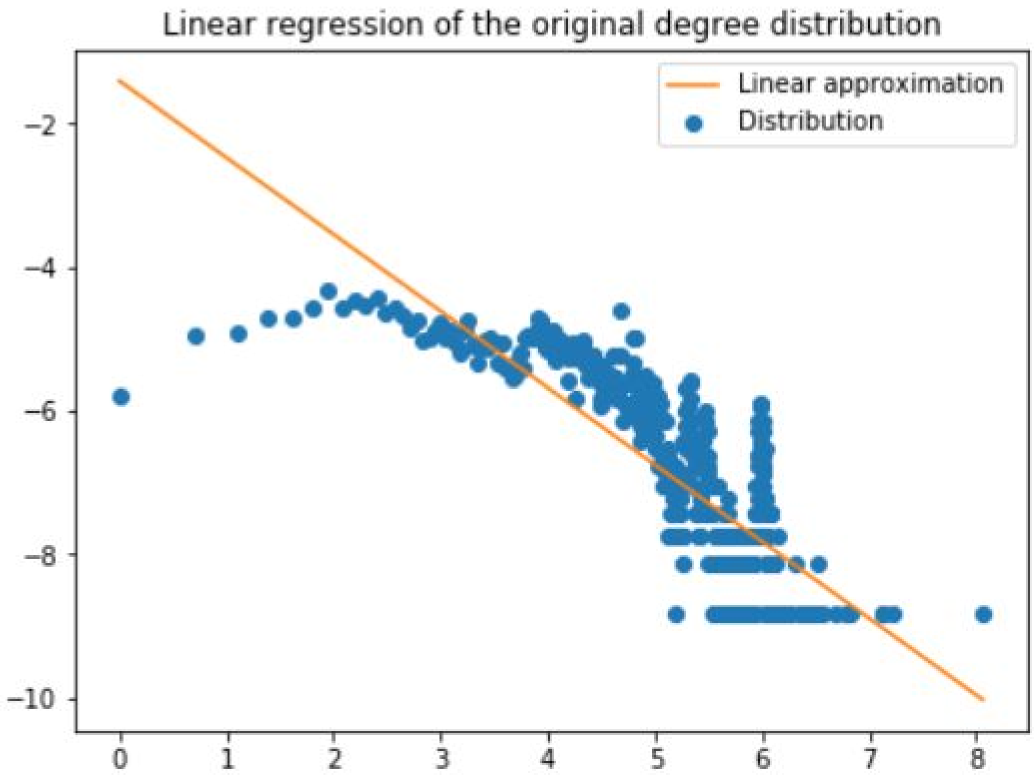
- The degree distribution is very complicated and noisy

- The log-log plot suggests a power law distribution





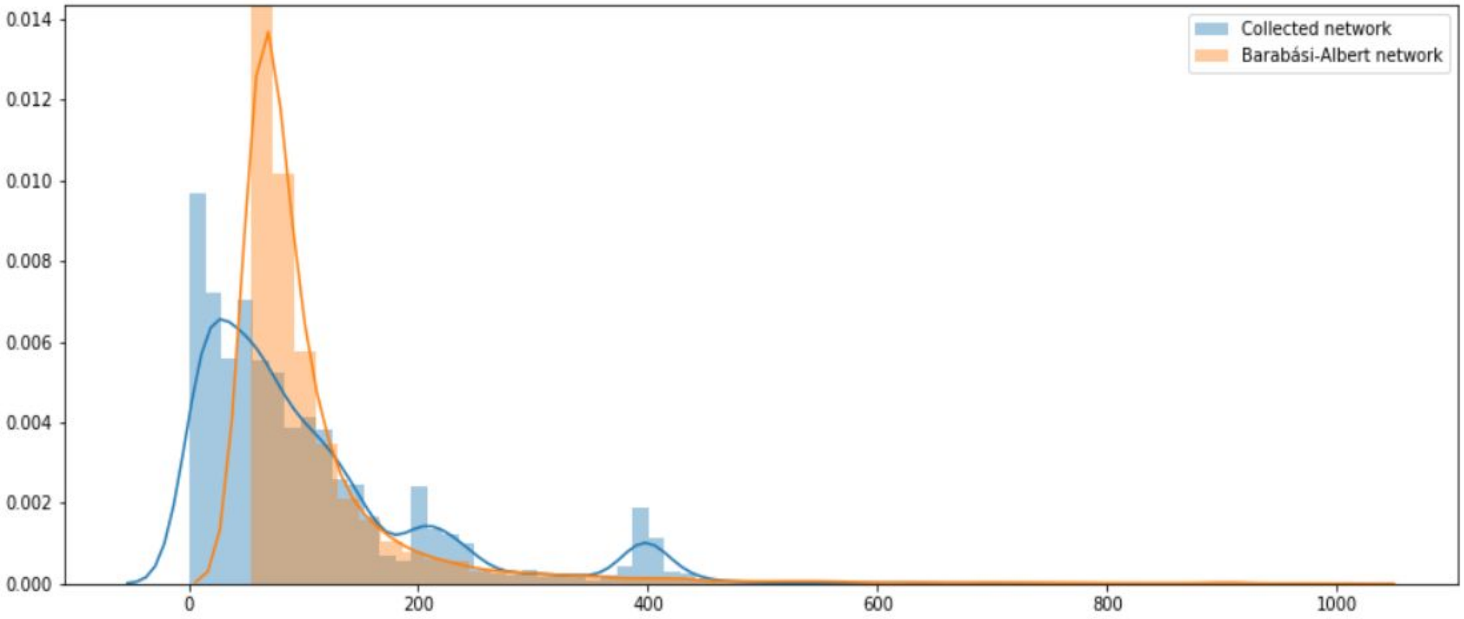
Regression of the degree distribution



- A linear regression can be used to find the power law coefficient
- $y = \mathbf{1.0693} \times x - 1.405$
- $R^2 \simeq 0.627$
- The value of R^2 is not very high but the power law seems to be a good approximation anyway.

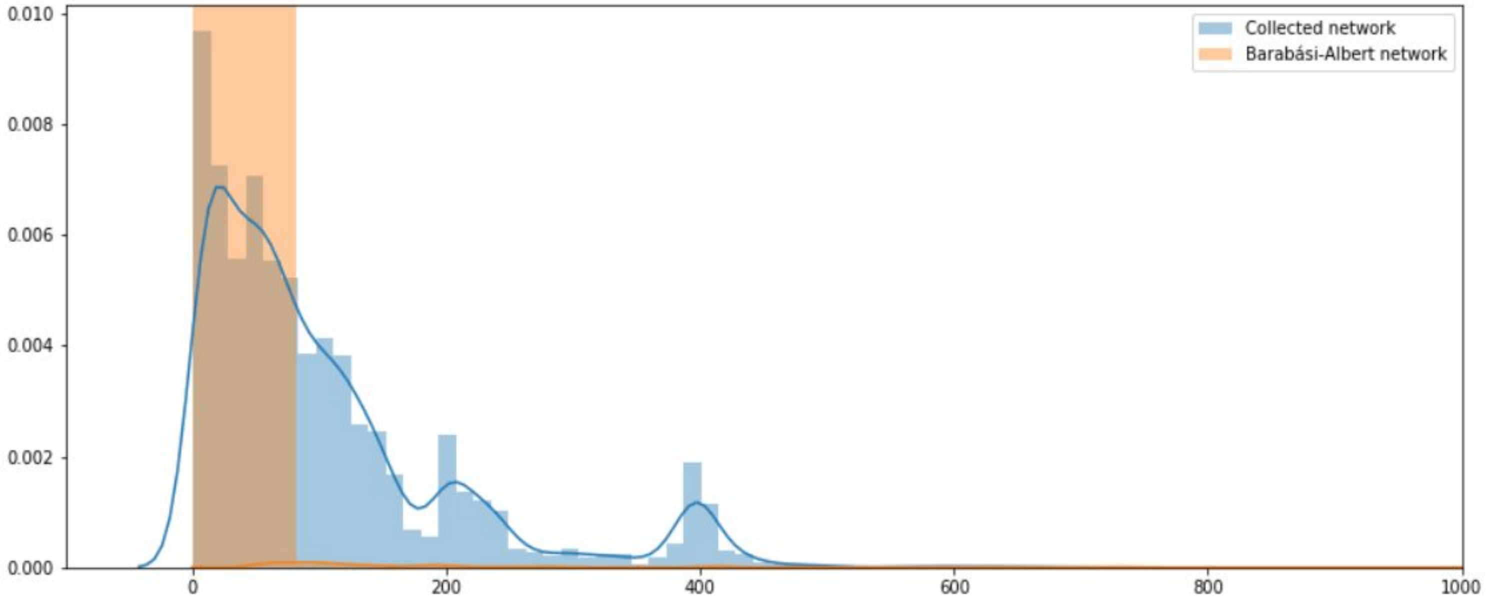
Barabási-Albert model

- The Barabási-Albert graph allows to get a power law distribution with an exponent of 3
- We would prefer a coefficient closer to the one calculated (~1.0693)



Creation of a power law network

- No perfect method to get exactly the wanted coefficient
- Generating a sequence of degrees following the desired law and linking them iteratively



Summary of the properties

	Nodes	**Edges**	**Distribution fit**	**Clustering coefficient**	**Giant components**
Original	6830	367483	X	0.643	6830/6830
Erdős–Rényi	6830	368697	Bad	0.016	6830/6830
Barabási-Albert	6830	365904	Pretty good	0.048	6830/6830
Self-made power law	6830	24675	Pretty bad	0.452	6826/683

- None of the model perfectly fits all the properties

- It seems like the Barabási-Albert network gives the most encouraging results
    - => preferential attachment

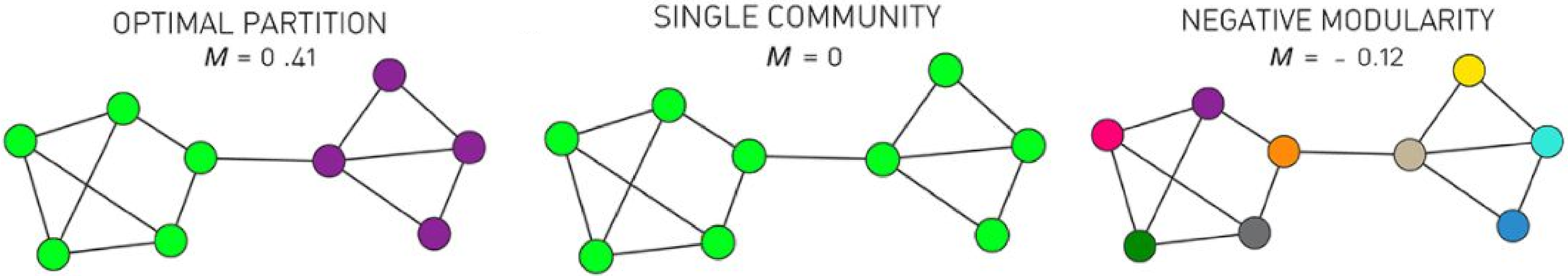
3.a) Community detection: Louvain Algorithm

Louvain Algorithm

- Optimization algorithm for maximizing the modularity of the network

Modularity

- Measure of division of a network into modules (communities)



Louvain Algorithm

Start with a weighted network of N nodes, N different communities

Step I:

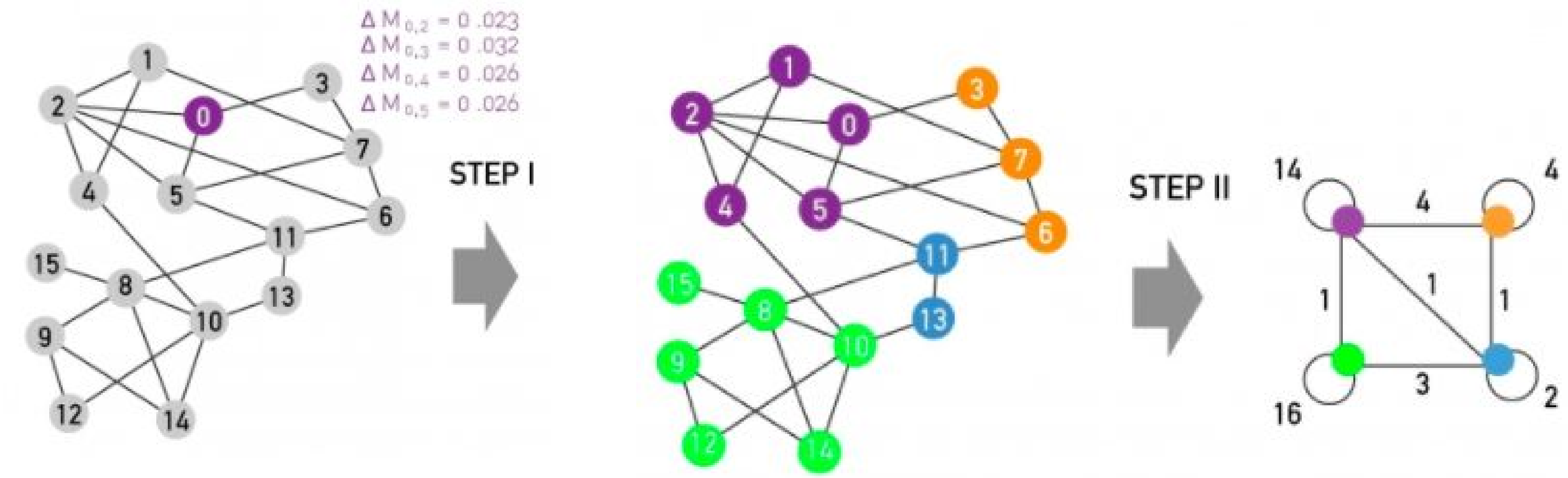
- For each node *i* evaluate the gain in modularity if we place node *i* in the community of one of its neighbors *j*

Step II:

- Construct a new network with communities as nodes
- New weights are the sum of weights on links between communities

Louvain algorithm

- Forces small communities into a larger one



Results

- Modularity of the resulting partition
  - High modularity 0.7878
  - Network has community structures
  - Number of communities: 17

- Topics for community

```
Community 2/17 (561 pages) :
('Jaguar vehicles', 44)
('Ford Motor Company', 43)
('Rear-wheel-drive vehicles', 40)
('Motor vehicle manufacturers of the United Kingdom', 36)
('Defunct motor vehicle manufacturers of England', 33)
('Car manufacturers of the United Kingdom', 30)
('Sports car manufacturers', 29)
('Former defence companies of the United Kingdom', 29)
('Car brands', 29)
('Defunct motor vehicle manufacturers of the United Kingdom', 28
```

Alphabetical Order
Aircrafts
American Footbal
Animals / mammals
Apple inc.
British ships
Cars
Comics and fictional characters
Electronics
Car racing
Luxury in Britain
Mexican soccer
Music instruments
Rugby
Social science
Songwriters
Weapons

3.b) Community detection : Spectral Clustering

Spectral clustering algorithm

- Compute the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ or $\mathbf{L}_{\mathrm{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$
- Compute the first k eigenvalues and eigenvectors
- Gives a n x k matrix (k first eigenvectors)
- Each row can be seen as an embedding of a node in $\mathbb{R}^k$

- Clusterize those n points of $\mathbb{R}^k$ to get the labels

First try : apply on natural graph

- Edges when there are links
- Weights all equal to 1

Poor results :

- Giant community with 99% of the nodes

```
'1, 1, 0, 0, 0, 1, 1, 2, 0, 0, 1, 6809, 1, 1, 1, 4, 1, 2, 2, 1, 1'
```

Better idea : apply a kernel

$$\arg\min_{y_1,\ldots,y_N} \sum_{i\sim j} \mathbf{W}(i,j)\|y_i - y_j\|_2^2$$

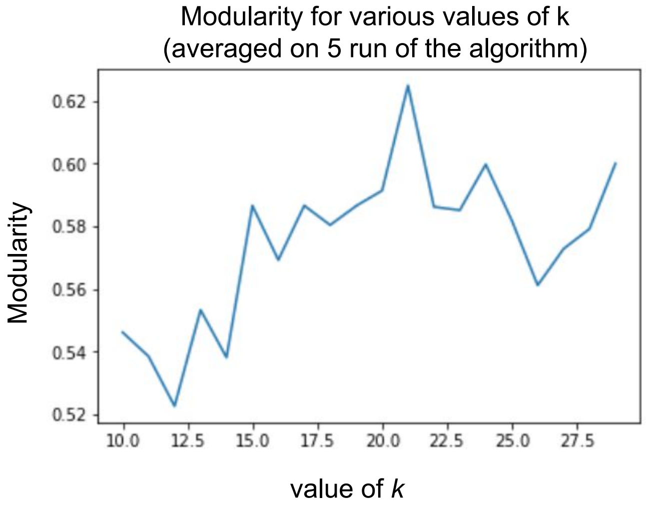
- Compute all distances (costly: Θ(|edges| + |nodes|log|nodes|))
- Create a new complete graph
- $\mathbf{W}(u,v) = \exp\left(\frac{-d^2(u,v)}{\sigma^2}\right)$ (gaussian kernel)

- Sparsify ?

Choose k ?

- The same k returned by Louvain ?

- Elbow rule ? Difficult to apply, high variance in the modularity (on average 0.035 of variance).



Results

- Modularity of the resulting partition : 0.62 ± 0.026

  (reminder of louvain : 0.7878)

- Same communities are detected:
  - Some are doubled (ship incidents three times)
  - One is finer (electronics split in computer hardware and video games)

Interpretation of the results:

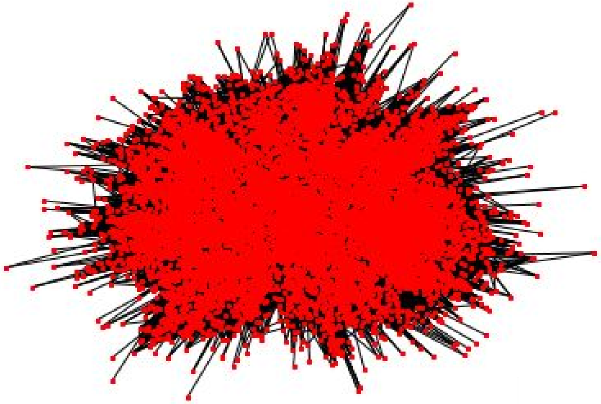
- Communities can be extracted just from the link structure

- Louvain better results in term of modularity

- Spectral clustering more costly but finer partition

4) Time to Visualize

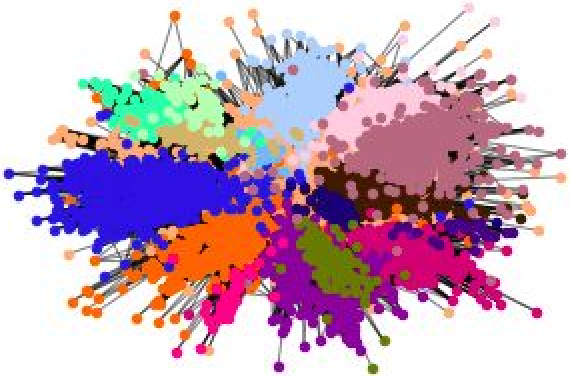
Networkx



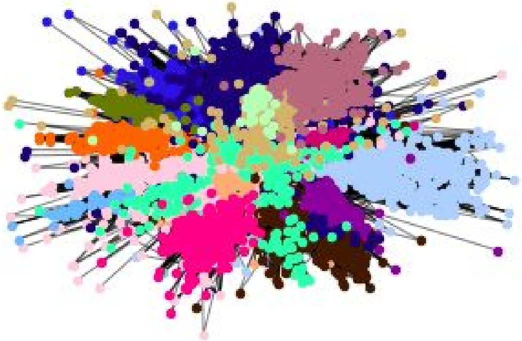
Initial state

Benefits:
- Easy to implement

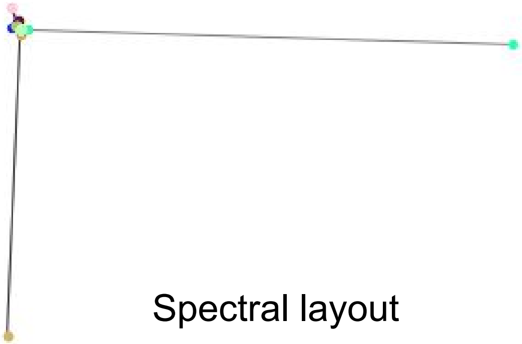
Drawbacks:
- Provides basic options for visualization
- Layouts don't provide enough information



Spring layout

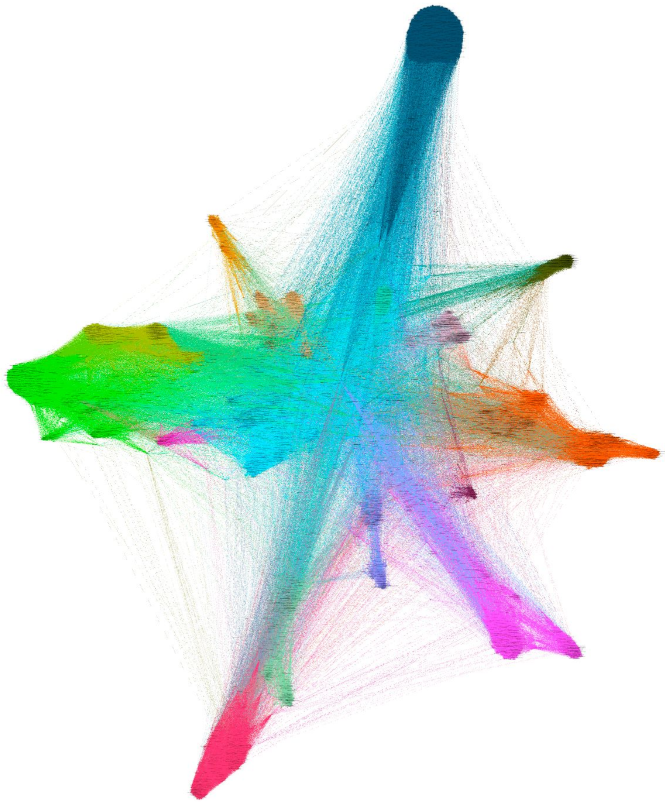


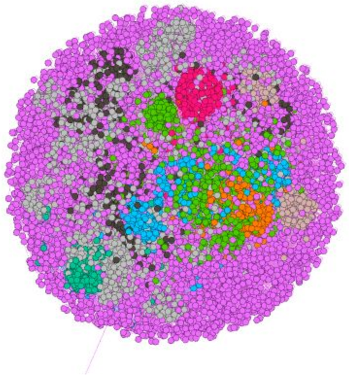
Fruchterman Reingold layout



Spectral layout

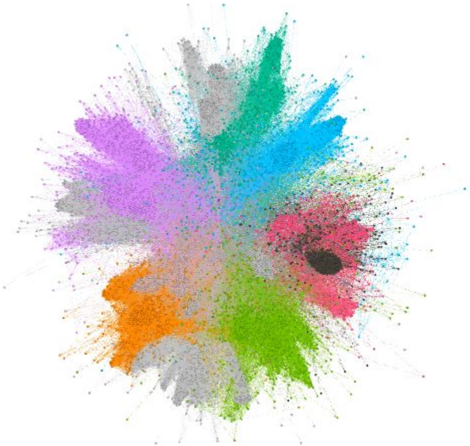
Gephi



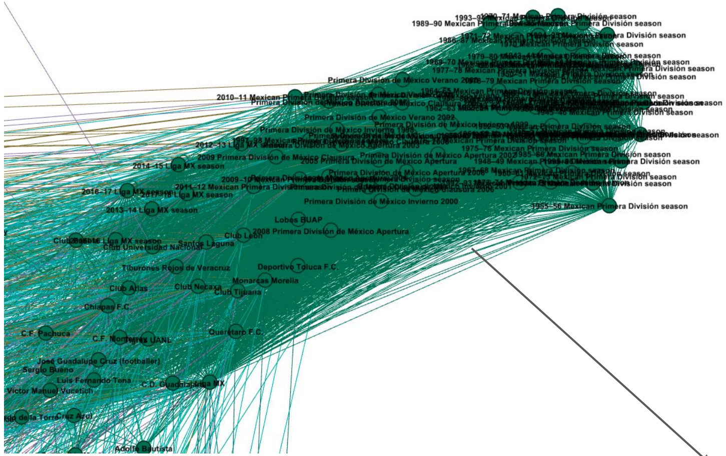


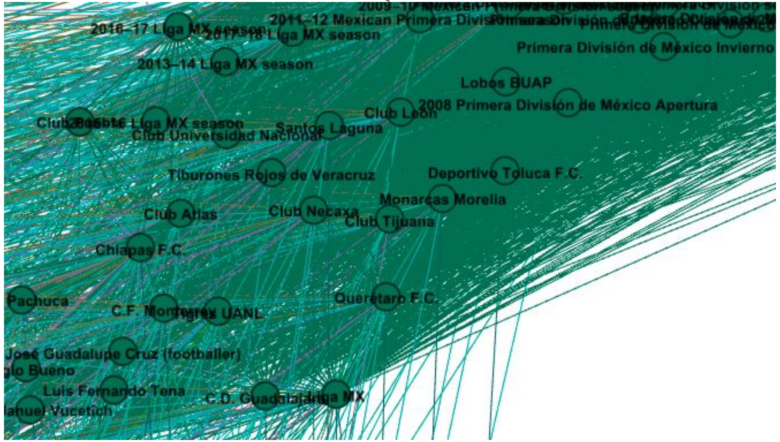
Benefits:
- Customizing layouts and partitions

Drawbacks:
- Slow
- Limited interactivity
- No Jupyter notebook support

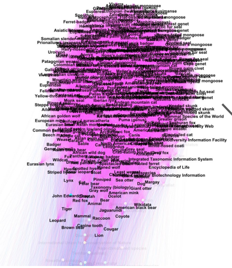


Gephi

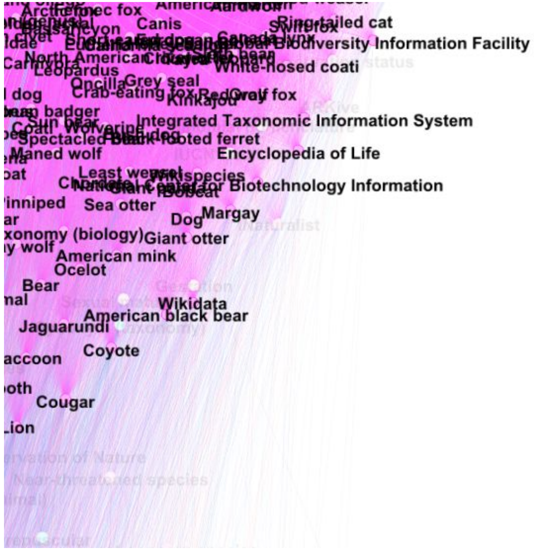




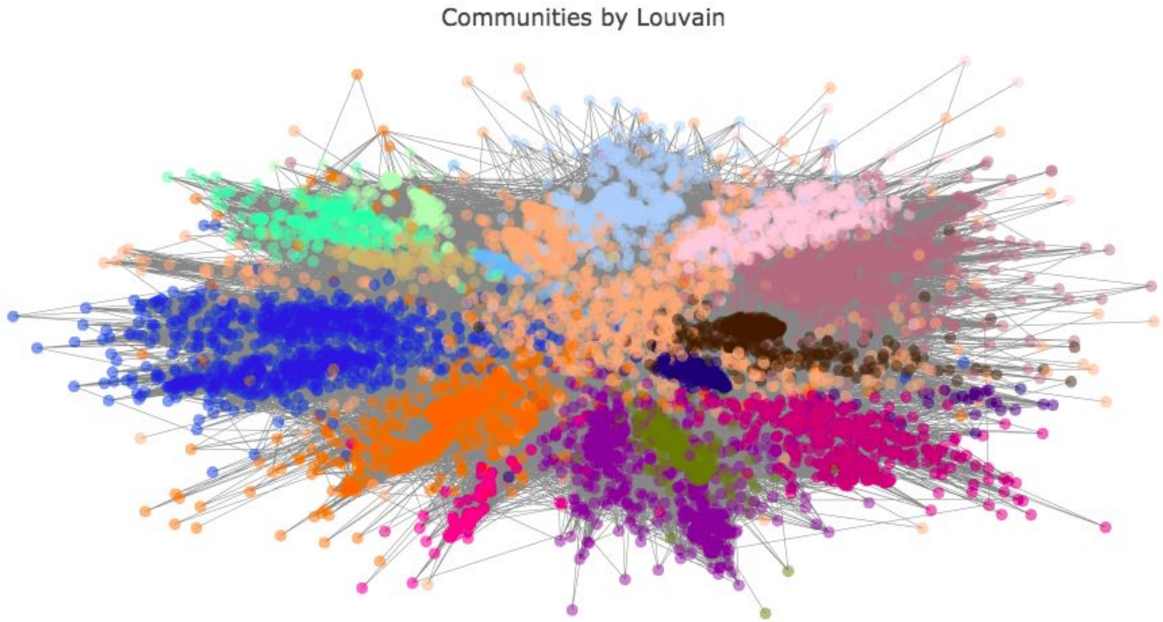
Mexican football



Animals



Plotly

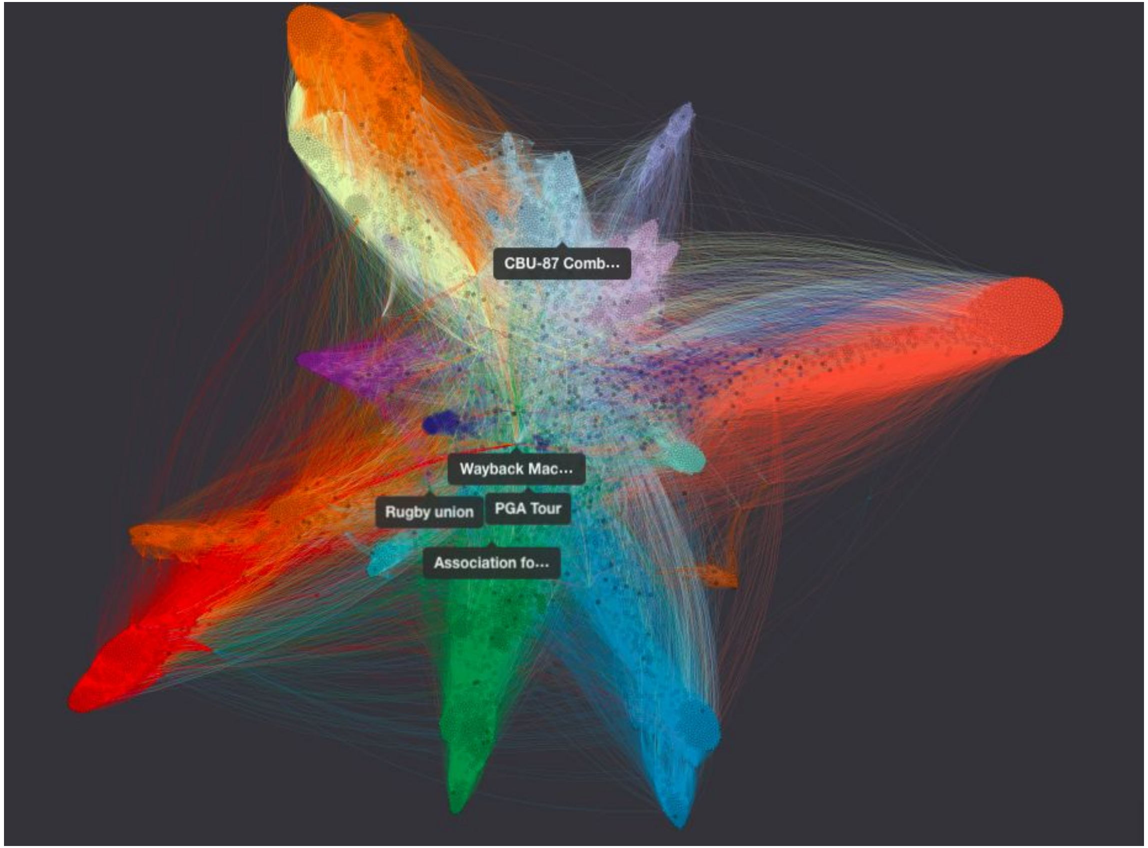


Benefits:
- Allows interactivity
- WebGL support

Drawbacks:
- Still laggy

Plotly interactive graph

Graphistry



Benefits:

- Fast
- Allows interactivity in real time
- Customizing plots
- Support for multiple python network modules

Drawbacks:

- Not really intuitive

Conclusion

Conclusion

- The clustering we made proved that we can extract categories only from structural considerations
- Louvain algorithm seems to provide better results : faster and better modularity

Further work

- Measure properly the fit of the community detection with the categories of the nodes. How ?

  Implement a natural language processing pipeline on those categories in order to extract topics

Additional slides

Modularity

Measure of division of a network into modules (communities)

Hypothesis: Randomly wired networks lack an inherent community structure

$$M = \sum_{c=1}^{nc} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right]$$

L – number of links in network
L_c – number of links within C_c community
k_c – degree of the nodes in C_c community

Modularity

Higher when

- number of edges in the communities high
- number of edges between the communities low

Zero when

- All nodes belong to one community

Negative when

- Each node belongs to separate community