network and community visualization

Fabio Morea

2023-05-13

introduction

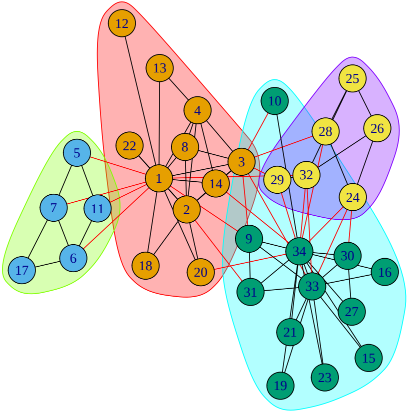
Networks and communities

Network analysis is a powerful tool for understanding complex systems in both the hard and social sciences. By representing data as a network, it is possible to identify communities of similar members, or nodes, that are connected to each other by various relationships.

This allows to gain insights into the structure of the system, as well as the behavior of the nodes within it.

For example, in social networks, nodes can represent people, and the relationships between them can represent various types of interactions, such as friendships or collaborations. By analyzing the network, researchers can identify clusters of people who are more likely to interact with each other, or who share similar characteristics.

Networks and communities: a toy example



Communities in large networks

As the size of a network increases, the complexity of the network also increases, making it difficult to represent the network in a simple visual format. As a result, a simple visualization may not provide much clarity in understanding the structure of the network. To gain a better understanding of a large network, more sophisticated visualizations and analysis techniques are needed.

Contents of this seminar

1. A family of benchmark networks (with built-in communities and controlled *fuzzyness*)
2. Visualization of network and communities (standard and custom layouts, k-core VS strength)
3. Modularity-based community detection
4. robustness of results in *fuzzy* networks
5. consensus and “robustness of membership”
6. sample results in a “labour market network”

A family of benchmark networks (with built-in communities and controlled *fuzzyness*)

About LFR benchmark networks

LFR (Lancichinetti-Fortunato-Radicchi) benchmark graphs are synthetic networks used to evaluate the performance of network analysis algorithms. They are generated using a stochastic block model and are designed to have realistic community structure, degree distributions, and other properties of real-world networks.

They are used to test the accuracy of algorithms for network analysis tasks such as community detection, link prediction, and node classification.

Benchmark graphs for testing community detection algorithms, Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi, Phys. Rev. E 78, 046110 2008

LFR_benchmark_graph — NetworkX 3.1 documentation

Building a family of LFR benchmark networks

in Python , networkx LFR_benchmark_graph()

https://networkx.org/documentation/stable/reference/generated/network

```python
for mui in range(10,99,10):
    print("generating benchmark for mu = ", mui/100)
    gb = LFR_benchmark_graph( mu=mui/100,
        n= 250,
        tau1 = 2,
        tau2 = 2,
        average_degree=5,
        min_community=30,
        seed=42)
    gt = add_true_labels(gb)
    nx.write_gml(gt, f"FLR_benchmark_{mui}.gml")
```

```
## generating benchmark for mu =  0.1
## generating benchmark for mu =  0.2
## generating benchmark for mu =  0.3
## generating benchmark for mu =  0.4
```

2- Visualization of network and communities (standard and custom layouts, k-core VS strength)

print mu = 10%

```
...

## [1] "Loading graph... FLR_benchmark_10.gml"

##              [,1]      [,2]
##  [1,]  1.3052332 -2.854755
##  [2,]  0.2228602 -4.668300
##  [3,] -5.5707272  6.727587
##  [4,]  3.1711022  8.814313
##  [5,] -8.7578044  5.732778
##  [6,] 12.0735977 -2.162243
##  [7,] -8.9116410  2.263359
##  [8,]  2.6776370  1.081074
##  [9,] -3.6850453 -5.952591
## [10,] -6.6124565 -7.022790
```

plot function

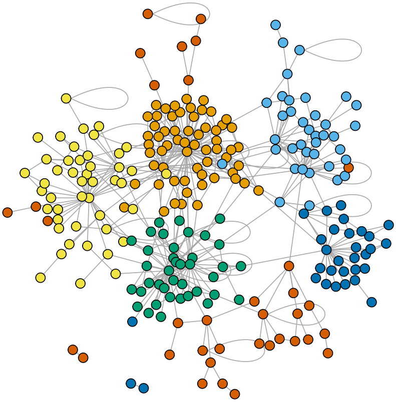
```
plot_graph_and_comms <- function(mu , show_comms = TRUE) {
  filename = paste0("FLR_benchmark_", mu, ".gml")
  print(paste("Benchmark network ", filename,"and communit
  g <- read_graph(filename, format = "gml")
  LO = layout_with_fr(g)
  if (show_comms == TRUE) {
    comms = table(V(g)$community)
    comms = make_clusters(g,
                          membership = V(g)$community,
                          modularity = FALSE)

    plot(
      comms,
      g,
      vertex.color = V(g)$community,
      layout = LO,
      vertex.size = 5,
      vertex.label = NA
    )
```

plot mu = 10%

```
plot_graph_and_comms(mu = 10 , show_comms = FALSE)
```

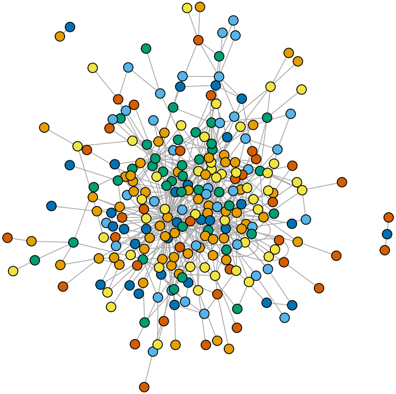
```
## [1] "Benchmark network  FLR_benchmark_10.gml and communi
```



plot mu = 50%

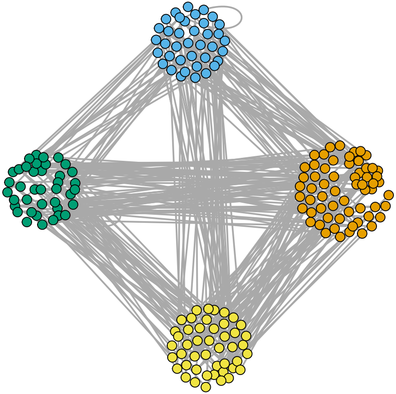
```
plot_graph_and_comms(mu = 50 , show_comms = FALSE)
```

```
## [1] "Benchmark network  FLR_benchmark_50.gml and communi
```



customized layout

plot customized layout



unsupervised community detection

What is a community

the definition of ‘community’ in network analysis depends on the context and the researcher’s goals. for example:

1. a community is a group of nodes that are more densely connected to each other than to the rest of the network.
2. a community is group of nodes that share certain characteristics, such as having similar properties
3. a community is a group of nodes that are more likely to interact with each other than with other nodes in the network.
4. ...

Modularity based community detection

- **Assumptions**:
  - community is subset of nodes that are more densely connected to each other than to the rest of the network
  - partition
  - unsupervised ("true labels" are not available - or not used for community detection)
- **Definition of Modularity**: Given a network G partitioned into a number of communities Gi, modularity Q(G,Gi) is a function measuring the extent to which edge density is higher within than between communities. i.e. a partition of G that maximises Q results in communities that have strong internal connections and weak connections with other communities

A partition with a higher modularity score indicates that the edges within the partition are denser than the edges between partitions, suggesting strong internal connections and weak connections with other communities. ***The optimal partition maximizes modularity***