Park Analysis 2 Graph Metrics

Code

In Woo Park

2/5/2021

Introduction

In this report we demonstrate how to compute common graph metrics in igraph; compare the results obtained to those that can be computed in Gephi; and explain any discrepancies we find between the two sets of results. We also interpret the meanings of the metrics with respect to an example network.

We work with a simplified EuroSiS WebAtlas network, constructed for the European Science in Society program. It consists of a partial sampling of Science in Society websites in Europe, and how they link to each other. Although the original network supplied in EuroSiS-WebAtlas.gexf is a multigraph, we work with a simple graph version. We have constructed a graphml version of the network by reading the gexf file into Gephi with parallel edges merged using "Sum", and writing it out as graphml. We read in the graphml below (the resulting simple graph should be undirected and weighted with 1285 vertices and 6462 edges):

Hide

```
WA <- read_graph("Networks/EuroSiS-WebAtlas-Simplified.graphml", format="graphml")
summary(WA)
is_simple(WA)
```

1. Degree and Weighted Degree

a. Average Degree Results

We compute these metrics in Gephi and igraph, and compare them in the table below.

Hide

```
(WA.mdeg  <- mean(degree(WA)))
(WA.mwdeg <- sum(strength(WA))/length(strength(WA)))
```

Metric	Value
Gephi Average Degree:	10.05759
igraph mean degree:	10.0575875
Gephi Avg. Weighted Degree:	11.807
igraph mean weighted degree:	11.8070039

b. Why EuroSiS average weighted degree is larger

Take for instance, two nodes that share one edge with a weight of 2. Both nodes have d=1 degrees, but share an edge with w=2 weights. This would represent node1 requesting something from node2 and node2 replying back to node1. This creates the situation where both nodes still have d=1 but the degree weight is now w=2. It must be that the degree only cares for unique degrees (connections) regardless of degree weight.

Our example is from ID: 98, Poland, where we can see a degree of 1, but a weight of 2. The nodes must be requesting and replying, but that is considered one unique connection regardless of multiple requests and replies.

Assume the base case, and apply to other nodes, you would end up with the degree average being strictly less than weighted degree average.

Also,

c. Plot of Degree Distribution

Hide

```
dd <- degree.distribution(WA)

plot(ifelse(dd == 0, NA, dd),
    main = "Degree Distribution No Zeros",
    xlab= "Nd",
    ylab = "p(Nd)",
    #log = "xy"
    )
```

d. Discussion of Degree Distribution

Discuss the degree distribution plot as if you are explaining to a client what the degree distribution plot tells us. Explain the specific plot, not just how these plots work in general. Relate the features of the degree distribution plot to the application domain (EuroSiS web sites) so your client can understand!

The degree distribution plot tells us several things. To begin with, nodes with degree 1 is the most popular so you can assume that most actor_types share connections with other nodes with a degree of 1 or a single connection. On the other end of the spectrum, the maximum degree is 100 which shares a neighbor with a slightly less accurate maximum at 99. If you also consider the third highest maximum, all three “maximum” nodes share a common country in that they all belong to the Belgium subgroup. This suggests that Belgium plays a big role in application domain as it has the most connections out of the other countries.

We can confirm this hypothesis by viewing the data set in Gephi and setting the node partitions by country. In this instance, Belgium does in fact represent the majority node partition at 15.33% with Estonia and France following closely at 12.76% and 11.75%.

Degree count between 10 and 60 seem to be the most dense but are closer to the count of 0. This could suggest that website to website connections happen in “bursts” where connections themselves aren’t regularly flowing but rather websites tend to connect with multiple websites once and this relationship isn’t held. Rather, websites prefer creating new connections than reusing old ones.

---

2. Distances

a. Mean Geodesic Distance and Diameter

We compute these metrics in Gephi and igraph, and compare them in the table below.

Hide

```
(WA.mdist <- mean_distance(WA))
(WA.diam  <- diameter(WA))
```

Metric	Value
Gephi Mean Geodesic Distance:	3.8958939492904223
igraph Mean Geodesic Distance:	3.8958939
Gephi Diameter:	10
igraph Diameter:	10

b. Explanation of Discrepancy

The discrepancy between Gephi's Diameter and igraphs' diameter is by a value of 2. There is a discrepancy between Gephi and igraph results because I believe igraph doesn't consider the weights during the diameter calculation. Since igraph gives us more control over computations, we can get the igraph to match the Gephi results as follows:

Hide

```
(WA.diam  <- diameter(WA, weights=NA))
```

c. Plot of Distance Distribution

Hide

```
plot(distance_table(WA)$res,
     main = "Distance Distribution",
     xlab = "Value",
     ylab = "Count",
     type = 'h',
     lwd = 6)
```

d. Comparison of Degree and Distance Distributions

Describe the difference in general shape and range of values between the distances plot of this question and the degree distribution plot of question 1c.

The plot for 1c. has shows the degree distribution in a different plot format. It seems inverse actually, the highest value in 1c. is values closest to p(Nd) of 0 whilst the plot for 2d. has the values closest to 4 as the highest count. Without much change, the plot for 2d. also goes from 0 - 10 instead of 0-100, they should represent the same thing as 2d. works out in between values in approximate decimals. Also, it is important to remember that 2d. showcases the distance distribution whilst 1c. showcases the degree distribution.

---

3. Clustering and Transitivity

a. Computations and Results

We compute both average local clustering coefficient and global transitivity in igraph, and compare them to Gephi's average clustering coefficient in the table below.

Hide

```
(WA.avglocalcc  <- transitivity(WA, type ="average"))
(WA.globalcc    <- transitivity(WA, type = "global"))
```

Metric	Value
Gephi Average Local CC:	0.381 / 0.23477402329444885
igraph Average Local CC:	0.381242
igraph Global Transitivity:	0.234774

b. Explanation of Difference

The values that differ from each other are the igraph average local cc and igraph global transitivity. The *direction* of this difference makes sense because both are computing completely different things. One computes the local clustering coefficient and the other computes the global clustering coefficient.

---

4. Components

a. Component Analysis in Gephi

Number of connected components identified by Gephi: 6 weakly connected components / 1285 nodes

Members of first smallest component:

Web Site (Label)	What they have in common
International Forum for Women	Country: Poland
Polska Konfederacja Pracodawców	Country: Poland
Gender Index Project	Country: Poland

Members of second smallest component:

Web Site (Label)	What they have in common
National Museum	Country: Montenegro
Natural History Museum	Country: Montenegro
State Archive of Montenegro	Country: Montenegro
Ministry of Culture, Sports	Country: Montenegro

Members of third smallest component:

Web Site (Label)	What they have in common
Semmelweis Doctors’ History	Country: Hungary
List of Budapest’s museums	Country: Hungary

The similarities between each smallest component and what they have in common is that they share the same country.

b. Component Analysis in igraph

The number of components and their sizes:

Hide

```
(WA.CC <- components(WA)$no)
(WA.CS <- components(WA)$csize)
```

The $label of vertices in each of the three smallest components:

Hide

```
(V(WA)[components(WA)$membership==2]$label)
```

Hide

```
(V(WA)[components(WA)$membership==3]$label)
```

Hide

```
(V(WA)[components(WA)$membership==4]$label)
```

Pau