# Park Analysis 1 Visualization

Code ▾

In Woo Park

2/3/2022

## Visualization in Gephi

### 1. EuroSiS WebAtlas (12 pts)

The graph EuroSiS-WebAtlas.gexf is a “WebAtlas” 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. This is a weighted multigraph: there may be multiple edges between nodes and the edges have weights. When reading into Gephi we **merge the edges with strategy Sum**. There should be 1285 vertices and 6462 edges.

The network includes node attributes for ‘country’ and ‘actor_type’ (e.g., government, media, research center, university). We construct visualizations that capture the extent of connections across types, and show whether country and/or actor_type appear to be related connectivity between sites.

#### (a) PDF of the visualization(s):

The Gephi visualizations may be found in: *q1_country.pdf* and *q1_actor_type.pdf*

#### (b) Description of how you constructed the visualization(s):

For the first question, we I used several parameters I defined as presets later on because I thought they were good choices for quick visual feedback. I used two graphs because I found it easier to quickly show relation between the country and actor_type features. For the layout features: scaling 2.5, LinLog mode: on, Prevent Overlap: on. For node sizing, I had it at min: 100, max: 1,000.

I chose these features because they made sure to spread out the nodes but still retain visual clusters. I let the software automatically set the color codes as it wasn’t necessary to specify them.

#### (c) Discussion of features of the real-world network that your visualization shows:

My visualizations does several things. First, the q1_country.pdf visualization uses ForceAtlas2 as the main layout which separates the nodes into clusters based on country. The clusters themselves are visible as they are color coded. Secondly, I’ve done the same process for actor_type using the default clustering done by the software. The node positions don’t change as they are still in the “country”-based layout which makes it easy to see country relation to actor_type.

In my visualization, University and secondary schools dominate the center, which are Poland, Finland, somewhat Belgium, and somewhat Hungary. In addition, media is the majority for Estonia.

I’ve tried having both labels visible for a singular visualization but I could not find a good layout to represent the relation as I wanted. These include, other layouts, having the nodes color coded by country, but the labels having different colors, testing different sizes, and attempting to make edges different colors. Non of these attempts were as clear than having two graphs users can switch between quickly.

### 2. Political Blogs (18 pts)

The network political-blogs.graphml is a network of political blogs from the 2004 election. Vertices are blogs and directed edges are references to other blogs. The vertex attribute ‘value’ indicates political orientation (1.0 is ‘right’ or conservative; 2.0 is ‘left’ or liberal).

The first visualization referenced below shows the Left-Right dichotomy in political blogs, with vertices colored for left (liberal) vs. right (conservative) orientation. It also enables us to identify the blogs that play bridging roles, indicated by higher number of edges that cross between liberal and conservative blogs. Further analysis below identifies the roles that blogs with high in-degree and high out-degree are playing.

### (a) PDF of the visualization(s):

The Gephi visualizations may be found in: *q2_1v1.pdf*

### (b) Description and discussion of visualization(s):

I believe the preset I made for question 1 was adequate for this question as well so I used the same preset values. The only thing I changed was edge size because there were more edges so it was harder to see. I also changed the pallete color to red and blue to signify the different political party the site identifies with. The q2_1v1.pdf currently shows the the disparity between the clusters.

### (c) Blogs that play a bridging role:

I have tried several different filter options which did not solve this problem. What I have done was combine several filters such as giant components, degree range, and out degree ranges, with inter or intra edges. The best visualization I could find was the pairing of in degree range and intra edges. After raising the in degree range setting to roughly 40, I’m left with a graph showing <10 nodes that seem to talk to each other the most. This could potentially represent the bridging roles as instapundit.com and talkingpointsmemo.com seem to be the biggest nodes that are dramatically closer to each other, and in addition receive the most in degree. The website talkingpointsmemo.com seems to be the obvious bridge for conservatives as the arrow is not mutual. It goes from instapundit TO talkingpointsmemo. The other bridging sites I can infer is probably washingtonmonthly.com which is mutual with instapundit.

This can be viewed in *q2_c.pdf*

### (d) Roles of high in-degree and out-degree blogs:

Highest in-degree for conservatives was instapundit.com and for liberals it was dailykos.com. The roles they play is different in that instapundit.com seems to play a bridging role between conservative and liberals sites as dailykos.com plays a more central role in that most liberals websites refer to it the most. It could suggest dailykos.com is the main source for liberal information.

This can be viewed in *q2_d1.pdf*

Interestingly, the highest-out degree for conservatives was blogsforbush.com and for liberals it was newleftblogs.blogspot.com. Both of these nodes and the nodes it neighbors seem to all share simialar connectivity. This suggests two things, it’s kind of a circle jerk network they got going on, or they are owned by the same parent company and are mass producing similar articles based on each others posts. Whatever the case, we may have spotted a group of newsites that like to publish click baity work because those type of articles are usually shared as it basically is a huge advertisement for clicking on non-important news articles. My analysis here is a stretch but this comes from somebody who has worked in media before.

# Projection and Visualization of a Bipartite Graph

In this section we examine a network from Tapped In, a large and successful online community of education professionals that operated from the 1990’s until the late 2000’s. Participants interacted in synchronous chat rooms or in asynchronous discussions, and could also share files. Some were associated with organizations and others were unaffiliated.

The network we examine, TI-Partition-63.graphml, is a small subset of a bipartite graph of actors and media, with edges indicating how the actors interact via the three kinds of media. Here we read it in and summarize it:

Code

```
IGRAPH 6398bc2 D-WB 943 2763 --
+ attr: label (v/c), entity (v/c), room (v/c), tenantorgroup (v/c), actor (v/c),
| Weighted Degree (v/n), type (v/c), id (v/c), Edge Label (e/c), weight (e/n), id (e/c)
```

## 3. Projecting and Visualizing a Bipartite graph (20 pts)

In this section we project the network to be an Actor-Actor network, and write it out for visualization in Gephi. We need to solve a problem before we can do the projection. Our objective is to retain all information that was present in the network we read in, so we can pass that information on to Gephi.

### (a, 10 pts) Projecting with bipartite_projection

```
bipartite_projection(TI)
```

This will return an error that:

```
Error in handle_vertex_type_arg(types, graph) :
  Not a bipartite graph, supply `types' argument
```

Therefore we must do this to make sure Actors == true, and otherwise is false:

```
V(TI)$types <- (V(TI)$entity == "ACTOR")
```

We set the entity == Actor to preserve the information and set the proper data that Actors are true, and else are false. This projection returns a list of the information. We are creating a new attribute and assigning it to TI so no other attributes are taken over.

### (b, 5 pts) Write the actor-actor projection as graphml

```
write_graph(TI, "q2_p1.graphml", format = "graphml")
```

### (c, 5 pts) Visualize the result in Gephi

I was confused on the visualization aspect of this question. It says to color the nodes by tenantorgroup which I have done, but there are more than 2 groups, therefore the visualization is not a bipartite graph.

The Gephi visualizations may be found in: *q3.pdf*

So in addition, I set the node colors to types, the attribute I created during question 3 a), and have also provided a visualization for that in: *q3_2.pdf*

I believe this is more accurate in representing a “bipartite” graph in that my true/false values are finally being used and each node is actually linked by the properties of bipartite. Not sure what I’m supposed to inference from this other than Naomi and AviB being the majority of actors who are connected to other actors.

The obvious conclusion should be that groups of people should talk to a single entity, and then that entity belongs in their own group who talks to another entity.

# Manipulations and Visualization in igraph

Here we return to the European Science in Society network EuroSiS-WebAtlas.gexf. We show how to convert the network to a format that can be read into igraph, and then we demonstrate how to collapse and simplify the graph so that one can see the relationships between countries (similar to the French political party example in class.)

## 4. Preparing a graph for igraph (5 pts)

We want to work in igraph, which does not read gexf files, so we use Gephi to convert to a format igraph can use. (The resulting file is included in this notebook folder.) After doing that we continue with igraph below:

### (a) Format chosen and reason:

I opened up the EuroSIS-WebAtlas.gexf using gephi in order to export it as a graphml file. I chose graphml because I the file extension has been working for previous problems so I thought it would be a good choice for now.

### (b) Reading into igraph and Summary:

ESWA <- read_graph(“ESWA.graphml”, format=“graphml”)

### (c) Verifying the import is correct:

With ESWA.graphml opened on gephi, the top right in the contents tab has metadata on nodes and edges. Because we did the setting with SUM during the load, the total number of nodes are 1285, and the total number of edges are 6462. These numbers correspond to the metadata provided by the console when we call summary(ESWA):

```
IGRAPH bad7dc2 U-W- 1285 6462 --
```

The last two integers have matching node and edge count. We also can tell from this meta data that additional Gephi information was not included because r,g,b,x,y,size are not included.

### (d) Unique values for country:

summary(ESWA)

ESWA_country <- sort(unique(V(ESWA)$country))

Code

```
 [1] "Armenia"        "Belgium"        "Bulgaria"       "Czech Republic" "Estonia"        "Finla
nd"        "France"
 [8] "Hungary"        "International"  "Italy"          "Montenegro"     "Poland"         "Portu
gal"
```

## 5. Collapsed Visualization (20 pts)

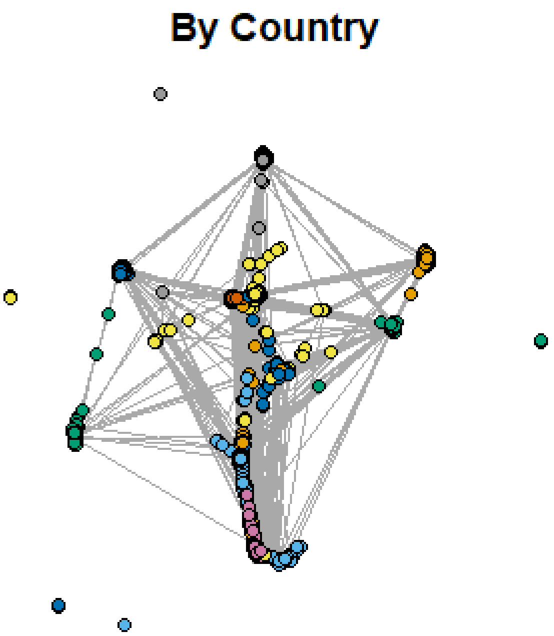
We want to see how different countries connect to each other. We will first show that visualization of the unmodified graph is not helpful. We will then collapse and simplify the graph to be a weighted graph between countries and then visualize that.

### (a) Plotting the full graph (5 pts)

A force directed layout with parameters chosen to make the nodes visible without labels, and to give each node a color according to its country:

Institut Piaget
Philosophy
And Science
Biotechnology
Prywatnych
Tomas Bata University Zlin
Youth Academy
Department
WyÅ¼sza SzkoÅ‚a
PaÅ„stwowa
High School
Danilovgrad

The plot without any features added looks like a clump of hot mess due to default settings, we will try and make the graph look more visually appealing and resourceful.



Using R, it is difficult to see the relation and connectivity between countries. Below is my attempt at making this graph more readable.

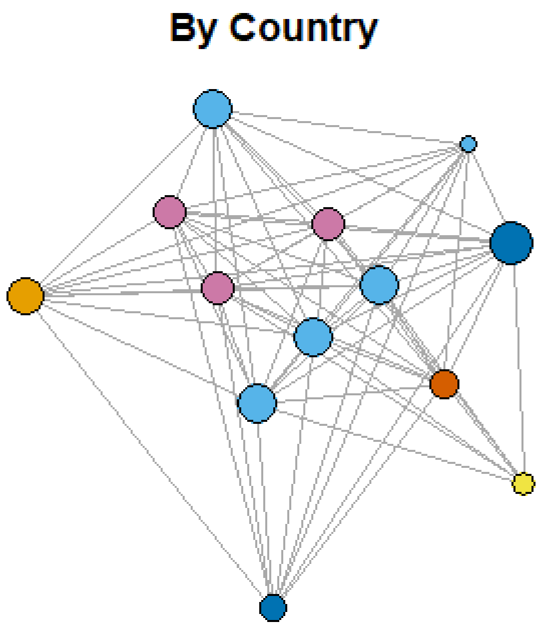
First I use the contract and simple functions in order to atttempt to make it more readable.

```
ESWA_contract <- contract(ESWA, as.numeric(as.factor(V(ESWA)$country)))
summary(ESWA_contract)

is_simple(ESWA_contract)

ESWA.lfr <- layout_with_fr(ESWA_contract)
```

Hide

```
plot(ESWA_contract, main = "By Country",
    layout = ESWA.lfr,
    vertex.label = NA,
    vertex.color = as.numeric(as.factor(V(ESWA_simple)$country)),
    vertex.size = 5*sqrt(as.numeric(as.factor(V(ESWA_simple)$country)))
)
```

```
Warning in layout[, 1] + label.dist * cos(-label.degree) * (vertex.size +  :
  longer object length is not a multiple of shorter object length
Warning in layout[, 2] + label.dist * sin(-label.degree) * (vertex.size +  :
  longer object length is not a multiple of shorter object length
```



It is difficult to see what each edge is doing here so I attempted to consolidate the edges using the weights of the edges as demonstrated in class. In this example, the vertex sizes were messing with our interpretation so we try another method.

### (b) Collapsing the network by Country (5 pts)

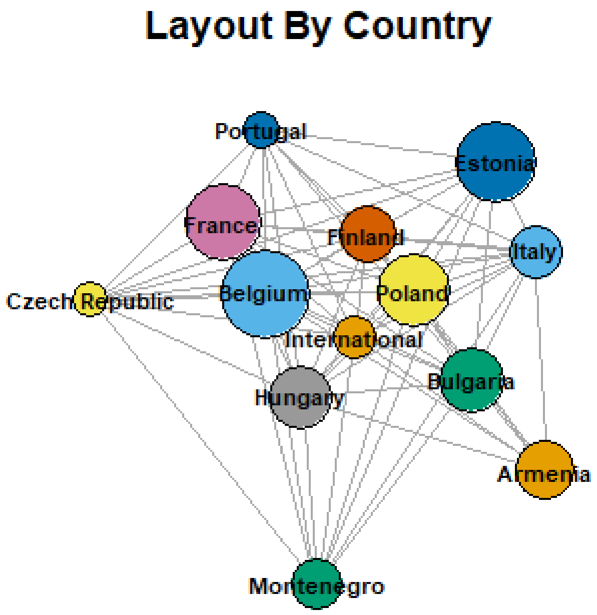
We use contract and simplify to collapse the network into one node for each country, summing the *existing* edge weights.

```
summary(ESWA_contract)
E(ESWA_contract)$weight
E(ESWA_contract)$weight <- 1
ESWA_contract <- simplify(ESWA_contract)
summary(ESWA_contract)
is_simple(ESWA_contract)
E(ESWA_contract)$weight
```

```
(ESWA_name <- sort(unique(V(ESWA)$country)))

table(V(ESWA)$country)
(ESWA_size <- as.vector(table(V(ESWA)$country)))
```

### (c) Plotting the Country graph (5 pts)

```
plot(ESWA_contract, main = "Layout By Country",
    layout = ESWA.lfr,
    vertex.size = 2.8*sqrt(ESWA_size),
    vertex.label = ESWA_name,
    vertex.label.cex = 0.7,
    vertex.label.color = "Black",
    vertex.label.family =NA,
    vertex.label.font = 2,
    vertex.label.dist = 0.0,
    vertex.color = V(ESWA),
    edge.width = sqrt(E(ESWA)$weight),
    edge.arrow.size = 0
    )
```



## (d) Discussion of connectivity between countries (5 pts)

The issue with this graph is that we have lost a lot of information as we consolidated the edges based on repeats. The Warning is valid from the homework prompt, I do in fact appreciate Gephi because I cannot get R to produce the graph I want. At least for now, we can still see how each country connects with others. This should be cleared up in our gephi analysis.

However, from what we have right now, we can infer several things from our visualization. Countries that are geographically closer to each other definitely have more connections. This is probably due to geographic location relating to populations having similar languages. It would be more common to speak to someone with the same language, which is the obvious conclusion.

# 6. Returning to Gephi for Visualization (5 pts)

### (a) Annotating Vertices

Recording the country names and number of web sites on vertices of the Country graph:

Hide

```
V(ESWA_contract)$country <- ESWA_name
V(ESWA_contract)$website_count <- ESWA_size
summary(ESWA_contract)
```

summary(ESWA_contract)

### (b) Writing the graph out in format Gephi can read

Hide

```
write_graph(ESWA_contract, "q6_ESWA.graphml", format="graphml")
```

### (c) Visualizing in Gephi

The Gephi visualization may be found in: *q6.pdf*

Yes, however, I had to use my classmates Gephi visualization because my Gephi seems to be currently bugged and I cannot set edge weights properly. Even when changing the necessary values, the edge weights stay the same at 1.0, not by, in our case, the ESWA_size. I have linked a picture showing my attempt.

The conclusion is the same, however, using Gephi is far superior than plotting in R, because we can visually see the impact of connectivity between countries easier.

Below is a sreenshot of the bug.

help (https://cdn.discordapp.com/attachments/804939291884912745/938713184666288168/unknown.png)

---

Pau