Social Network Analysis 2022

Class 9 – Summary & Feedback

What We've Done So Far ~ Basics

- How to create a network in R:
  - From a manual command, a list of edges, or data from a social media API like Twitter.
- How to "prune" a network:
  - Removing isolates, identifying and removing small connected components.
  - Removing or hiding unimportant node, edges, or labels.
- How to calculate basic statistics about a network:
  - Network size, diameter, mean distance, edge density.
- Similarly, how to calculate statistics which tell us about each node's role in the network:
  - Indegree / Outdegree, and various measures of centrality

What We've Done So Far ~ Communities

- Community Detection
  - How to use various algorithms (Fast-Greedy, Infomap, etc.) to find the densely-connected communities that exist within your overall graph.
  - How to "trim" those communities – focusing on the most important ones, or combining smaller communities logically to allow for easier analysis.
  - How to separate out communities from the overall graph, so you can analyse them separately – finding their most important / central nodes, etc.

- Community Identification
  - Using a combination of network analysis (centrality algorithms etc.) and other techniques like text analysis to interpret communities and uncover their identities or unifying properties.

What We've Done So Far ~ Visualisation

- Network Layout
  - Using a variety of algorithms (Fruchterman-Reingold, Kamada-Kawai, Graphopt etc.) and parameters to find the optimal layout for viewing your network.
  - Saving that layout in an R object, so you can reliably re-create it as you work on other aspects of the visualisation.

- Colours, Sizes and Transparencies
  - Setting the colours, sizes, transparency levels and other aspects of your nodes and edges to create an easy to read visualisation that highlights key aspects of your data.

- Selective Plotting
  - Plotting only the most important aspects of a graph – key nodes, edges and labels – in order to focus viewers' attention on the key points of your research.

Today…

- Today we'll look at one final tool that you may wish to use as you work with networks – a free piece of software called Gephi, which is primarily used by people working with biological networks, but has many powerful features that can be useful to us as social scientists as well.

- This class will also be an opportunity for feedback and any questions you may have about your final research project for the module – I've left the class schedule quite open so there should be plenty of time for anything you'd like to ask or discuss.

General Feedback

Final Assignment

How much data is enough data?

- This question depends very much on what kind of data you're working with, and what you hope to find out.

- If you're doing a small-scale study of the behaviour of a handful of entities (people, companies, Twitter accounts, etc.), then a relatively small number of network edges can still be enough – a few hundred, or possibly as few as several dozen.

- If you're working with something like a retweet network on a popular topic, I'd suggest getting as much data as you can, within reason. The structure of your network will be more clear, and the communities more clearly defined, when you have more data to work with.

A note on retweet network data…

- We collected 18,000 tweets using the keyword “vaccine”, and were able to construct a fairly interesting network from that. However – recall that this data only spanned a **20 hour period**.
- Consequently, this cannot be considered as representative of the debate overall!

- Students who want to examine similarly popular and contentious keywords – some of you have proposed studying topics like immigration and climate change policy – need to be aware of this limitation.
- If possible, try to collect more data (using the rtweet “retryonratelimit” option) to help overcome this problem and get a more representative data set.

Problems with the Twitter API

- A number of you had issues with the Twitter API and rtweet. I'm still not entirely sure about the reason why it works *sometimes*, but Paolo has probably identified the root of the problems – rtweet only works version 1.1 of the API, not the current v.2.0.
- The slightly messy solution here seems to be to download and install v.0.70 of rtweet, which is an older version. I'm not sure why exactly this gets around the problem, but it seems to.
- You may wish to look at RTwitter2 as a possible replacement for rtweet in the long term: https://github.com/MaelKubli/RTwitterV2 - but note that you'll have to rewrite a lot of the code we've been using, as this package creates quite different data frames to store the tweets.

Unusual Patterns in Downloaded Tweets

- Speaking of the Twitter API, there is also a fundamental problem with the API being a “black box” – i.e., we don’t know what algorithm Twitter uses to select tweets when we search for them.
- It’s suspected that the popularity of a tweet (engagements such as Likes, RTs, etc.) plays a major role in whether it is selected – so if a topic has recently had declining engagement, you might get a lot of older tweets in your results, even if there are more recent ones that have been posted.
- One option to improve this situation: limit your search to only give you tweets from the past 24 hours, and re-run that search at the same time every day for several days. This should give you a decent time-series for the keywords.
- (If you have the time and resources, it’s also possible to do this every 15 minutes, limiting your search to only the previous 15 minutes.)

Data Collection

- Some of you have decided to supplement your data collection by adding extra hashtags or keywords to your searches.
- This is fine – it's often a a totally reasonable approach that can help you to identify a broader community and discourse around your chosen topic.
- However: Be careful to make sure that you're choosing keywords that actually reflect discussion of the same topic.
    - If your keywords are too different, the data for them may not overlap – so you end up with a lot of connected components – i.e. "subnetworks" in your data that are totally disconnected from each other.

A note on hashtags

- A few people suggested gathering data based on #hashtags (although I don't think anyone is actually using this approach in the end).
- This is a common approach in social science, but to be honest… I think it might be an idea that's mostly coming from researchers who don't actually use social media very much.
- The decision by a Twitter user to use a #hashtag instead of just writing out a keyword normally may be suggestive of a very specific perspective – in other words, searching for "#vaccine" instead of just "vaccine" might introduce a major bias to your data, because people using the hashtag are more likely to see themselves as campaigners or influencers on this issue.

- Note that a search for "vaccine" would also return tweets containing "#vaccine" by default.

Working with Friend/Follower Networks

- Some of you have chosen to work with Friend/Follower networks rather than RT networks.
- This is a great approach, but it has some challenges in terms of data collection. One of the major ones is that unless you download the FULL list of Friends/Followers for a given account, your data will be highly biased, as the lists are downloaded in reverse chronological order (not at random).
- Downloading the full list for a popular account could be quite time-consuming; the absolute maximum you can download in a single 15 minute period is 75,000 IDs (15 “pages” of 5000 IDs each).
- The resulting networks also tend to be very large, of course, since each of those downloaded IDs becomes a node.

More Efficient Friend/Follower Networks

- Depending on what you are trying to show, there are a few ways to reduce the size and complexity of friend and follower networks.
- **Remove all nodes with degree = 1:** Nodes with degree=1 are connected to only one of your target accounts, and thus tell us nothing about network structure. They can be safely removed from the graph right at the outset.
- **Calculate Jaccard Similarity**: By using the lists of followers for each target account to calculate a Jaccard Similarity value (Intersection / Union), you can create a simple network of your target accounts where the similarity between the accounts is represented as the weight of the edge between them.
- For two vectors of Twitter IDs **a** and **b**:
  jaccard_similarity <- length(intersect(a, b)) / (length(a) + length(b) – (length(intersect(a, b))))

Connected Components

- It's worth revising again the difference between connected components and communities.
- Connected components are *separate networks* that exist in your data – there are no connections between them.
    - They usually indicate that you've gathered data for multiple unconnected topics, or just some data that's peripheral to the *main* conversation about a topic.
- Communities are *clusters of tightly-connected users within a single network*. They are connected to the other communities – they're not separate networks entirely.
    - They indicate "closeness" between a group of nodes; for example, a group of characters in a book who interact with each other a lot, or a group of people on Twitter who mostly agree with each other.

Best Practices with Connected Components

- Generally speaking, when we're analysing a network we want to ignore any small components and focus exclusively on the largest component – called the **giant component** – since this is where the actual community we're interested in can be found.
- In a situation where there's one giant component (80% - 90% of your nodes), you can simply extract that component and ignore everything else.
- However, you may find that your network contains two or more similarly sized large components. **This usually indicates a problem in your data collection / specification.** It means you gathered data related to topics or communities that aren't connected to each other at all.
    - You won't be able to analyse this data in a single network – you'll need to extract each of the large components and analyse them separately. You should also try to understand and explain why they were disconnected from each other; why didn't your nodes have any connections?

Assignment 9

- Your assignment for tomorrow is to prepare a presentation of a research plan outlining your final project for this module.
- This plan should include:
  - Your research question (and a brief introduction of your topic).
  - Your hypothesis (this can be quite broadly phrased)
  - Details of the data you're going to use and your progress in acquiring it
  - Your intended research methodology

- This research plan should be presented in tomorrow's class.
  - You should prepare some slides for your presentation.
  - Students who cannot present in class for any reason may submit a detailed research plan over email instead.