Methods for Data Science (M345 A50)

Coursework 2 – Networks

Deadline: Friday 11 January 2019, 5pm.

General instructions

The goal of this project is to analyse a dataset using the tools from Part 2 of our course. Note that coursework projects are different from exams. They are more open-ended and may require going beyond what we did in lectures. Initiative and creativity are important as is the ability to pull together the course content, draw new links between subjects and back up your analysis with relevant computations. **The quality of presentation and communication are very important**, so use good combinations of tables and figures to present your results.

You can find detailed submission instructions at the end of this document.

Dataset

In this project we will analyse a network dataset. The nodes in the network are doctors, and we have information about when they first prescribed a new drug. This offers the opportunity to explore this social network of doctors, and to relate the network structure to the prescribing behaviour.

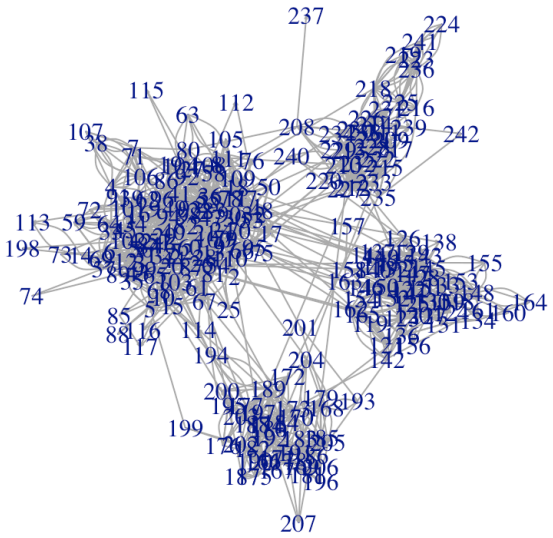
The dataset is in the file `doctornet.Rdata`

```
library(igraph)

load("doctornet.Rdata")

par(oma = c(5,4,0,0) + 0.1, mar = c(0,0,1,1) + 0.1) # margin settings

plot(docnet2,vertex.size=0.3,edge.arrow.size=0.1, vertex.color="blue", verte
x.frame.color=NA,xlim=c(-1,1),ylim=c(-1,1))
```



The vertices represent about 85% of the doctors in 4 cities. The edges are derived from answers to questions about who doctors turned to for advice, with whom they had discussions about medical practice, and with whom they socialised.

Prescription records were used to obtain the month in which each doctor prescribed the new drug (if they did).

Here are descriptions of what all the vertex attributes mean:
http://moreno.ss.uci.edu/data.html#ckm

https://rdrr.io/cran/spatialprobit/man/CKM.html

You can access vertex attributes in `igraph` using e.g. `V(docnet2)$nodeCity` to get information of the city that each doctor belongs to.

This dataset also has four kinds of edges, for whether the link is based on giving advice, having a discussion, being friends, or a random link. In this coursework you won't use most of this information, but it's left there for completeness.

For simplicity, in all questions we will use the undirected version of the graph. Use `as.undirected` to convert the graph.

Question 1 (Rmd only, 10 marks**)**

Compute some basic statistics about this graph. How many vertices and edges does it have? What is the mean degree? Show a histogram of the degree distribution.

The "scale free" property of a graph relates to the degree distribution. A graph with a power-law degree distribution is called "scale-free" because the power law has the same functional form at all scales. Empirically, this is the reason why, for example, there are cities 100 times as big as other cities (city size has a power law distribution), but in contrast, no one is 100 times as tall as anyone else (human height does not follow a power law distribution).

Do you think that the graph has the "scale free" property? Why or why not? Here we are not expecting you to do a statistical test, just a visual demonstration that you understand what the scale free property is.

Question 2 (20 marks)

Perform community detection in this graph using Newman's eigenvector method.

Compare the communities you find to the `nodeCity` vertex attribute using a confusion matrix (a table). Plot the network, illustrating the results of your community detection with vertex colours. What do you notice?

Use any other number of community detection methods from those at the bottom of the page http://igraph.org/r/doc/communities.html (e.g. `cluster_walktrap`, `cluster_edge_betweenness`, `cluster_leading_eigen`, etc). Compare the results with Newman's and the `nodeCity` vertex attribute. Discuss similarities and differences.

Question 3 (30 marks)

A "giant component" is a component of a graph G on n vertices whose size is $O(n)$. Loosely speaking, this means that the component contains a high fraction of the vertices (often understood in the context of $n \to \infty$). Does the docnet2 network have a giant component? What is its size?

Though in practice we cannot explore how a real network changes as we add more nodes or edges, we can study the network connectivity by removing them. With reference to the theory of Erdos-Renyi random graphs we saw in class, if our graph *were* an Erdos-Renyi random graph, how many edges would you have to delete for it to be likely to be very disconnected (i.e. for there to be no giant component)?

Explore deleting edges at random in the docnet2 graph. Approximately, or on average, how many edges do you need to delete to have a substantial effect on the size of the largest component? Comment on how this compares to the theoretical result. **Hint:** think of plotting the size of largest component vs fraction of nodes removed, using `for` loops to randomly remove nodes and averaging over many iterations.

Question 4 (20 marks)

We discussed a number of different centrality measures in class. Use some of these and:

- compare them against each other.
- use them to identify the best set of vertices which, when removed, will disrupt the network.

Test your results: informed by the centrality scores, remove vertices and test to what extent they disconnect the network. Illustrate your results with one or more plots. Compare between the different centrality measures.

Mastery Question (Rmd only, 20 marks**)**

Study the paper "Biological network comparison using graphlet degree distribution" by Natasa Przulj, which generalises the concept of degree distribution:
https://academic.oup.com/bioinformatics/article/23/2/e177/202080

This paper is highly cited and the method is implemented in the `ergm.graphlets` package in R.

Give a brief description of the paper. What do you think are its key strengths? What would you improve or develop? Suggest a research direction that follows on from this work.

Submission instructions

You will hand in three documents, wrapped into **a single .zip file**:

1) Your code in .Rmd format.
2) The html that you create by knitting your .Rmd file.
3) A poster in pdf format.

You are also required to comply with these specific requirements:

- Name your files as 'SurnameCID.zip', e.g. Smith1234567.zip. **Do not submit multiple files**. This will slow down the marking and reduce our ability to give you detailed feedback on your work.
- Your .Rmd file must produce all plots that appear in your poster.
- Your html must make it clear where the answers to each question are. Use clear headings 'Question 1(a)', etc.
- Your poster should include the phrase "*The contents of this work and the associated code are my own unless otherwise stated*".

Note on Rmd files: Your .Rmd file should not be an unstructured long file that has everything you ever tried in it. The .Rmd is not a diary or lab notebook. Be succinct, concise and comment your code in enough detail to make it clear what each code block is doing. As we discussed in class, it is important you demonstrate you understand what the code is doing. A good strategy is first doing the project in your own R scripts, and then preparing an .Rmd file that summarizes your final answers.

R packages

You are strongly advised to use the `igraph` package. As mentioned in class, there are other related network packages such as `network`, `sna`, `statnet`, and `ergm`. **However** their documentation is more scarce and there seems to be less information about how to use them. Most importantly, some of these packages do not play well with `igraph`. Check our tutorial T4 for more information.

It is best not to try to work with `igraph` and `statnet`, `sna`, `network`, etc, at the same time.

If you do want to use tools from these other packages, you may need to detach `igraph` using `detach("package:igraph")`,to unload `igraph` before using the others. The package `intergraph` can convert graph objects between the various formats.

R markdown

You may use any R libraries you want. But do list them at the **top** of your .Rmd file in the setup section by using `require(library_name)`. This way, markers can ensure that they have installed the necessary libraries when running your code.

- **Do not** refer to files that need to be downloaded or read from your hard drive, other than 'doctornet.Rdata'. Do not forget the basic rule: make sure that your files can be opened and run on a computer different from yours. Failure to do this will significantly delay the marking and you may lose marks.
- You may define your own functions.

- Use text and comments to make it clear what your code is doing. Failure to explain your analyses or code may cause you to lose marks.
- Where a question has been labelled "**Rmd only**", you **do not** need to include the result in your poster.

Poster

Your poster should tell the story of your data science project. Imagine showing it to a potential employer or tweeting about it. Your reader should be able to understand the data you had, the question you asked, how you answered it, and what you found from the analysis.

Avoid using lots of text paragraphs - the poster should look good and be accessible to read. Of course, you will need to use *some* text to explain what you did and why, but try to keep the text concise and clear. Bullet points are typically more effective than long text paragraphs.

You can create your poster with any software of choice, but you are encouraged to use LaTeX. There are poster templates at www.overleaf.com which you can use without even having a LaTeX installation on your computer.

Keep in mind:

- The poster should be written clearly and should communicate your project's story with correct spelling and grammar. Plots must have titles and axis labels with legible font size. Unlabelled plots may cause you to lose marks.
- Because of its format, it is impossible to include every detail in a poster. This is OK: a key to good communication is having the 1-minute, 3-minute, 10-minute versions of your story. Think about how to describe your project in three sentences: what you asked, what you did, something you found.
- The poster does not need to have your answers to the "Rmd only" questions.
- All figures in your poster must be produced in your .Rmd file.

Marking scheme

Clarity/legibility of Rmd file (5 marks), Html file (80 marks without mastery), Poster (15 marks). Total: 100 marks. The mastery question has 20 additional marks, out of 120.

Marks may be deducted if the code cannot be run in the markers' computer, e.g. due to external files being required, etc.