Exercises: Emails at a large European research institution

YSC2210 - DAVis with R

Michael T. Gastner

Introduction

In their paper 'Local Higher-order Graph Clustering', Yin *et al.* (2017) introduced a data set for anonymised email data from a large European research institution. There is a directed edge $u \to v$ in the network if person u sent person v at least one email. The emails only represent communication within the institution (i.e. any emails from or to the rest of the world were discarded).

There are 1005 individuals in the network. Each individual belongs to exactly one of 42 departments at the research institute. The departments appear as node attributes in the data set.

Objectives

The network is moderately large, which poses a challenge for network visualisation. We explore how to condense the data into informative plots. Along the way, we gain more familiarity with the **tidygraph** and **ggraph** packages.

Data

Go to https://snap.stanford.edu/data/email-Eu-core.html and download

- the edge list that represents email communication links between members of the institution: `email-Eu-core.txt.gz`
- a list of numeric node identifiers with the department as a node attribute: `email-Eu-core-department-labels.txt.gz`

Tasks

(1) Import the node list and the edge list as tibbles. You need to change some of the default settings in the Import Dataset GUI because

   - the source files use spaces, not commas, as delimiters.
   - the top row does not contain column names.

   Let us give the columns in the node tibble the names `person` and `dept`.

(2) In the source data, person identifiers are counted from 0. For later convenience, it is best to consistently add 1 to every person identifier. (Department identifiers should remain unchanged.) Change the corresponding columns with `mutate()`, both in the node tibble and the edge tibble.

(3) Assemble a `tbl_graph` called `eu_netw` from the tibbles.

(4) Let us plot the complete network with **ggraph**. You might want to use the chunk option `cache=TRUE` because the code may run for a little while.

```
ggraph(eu_netw, layout = "kk") +
  geom_edge_bend(
    alpha = 0.1,
    arrow = arrow(length = unit(2, "mm"))
  ) +
  geom_node_point(colour = "brown", alpha = 0.5)
```

Do you find the result useful? Why or why not? Does changing the layout help?

(5) Contract the network so that, in the new network,

- there is one node for each department.
- there is one directed edge from department x to department y if x sent an email to y.
- each edge $x \to y$ has an attribute called `weight` that is equal to the number of people in department x who sent an email to department y.

Note: As discussed in the notes, there is a bug in the **tidygraph** function `to_contracted()`. The following code chunk bypasses the bug by using the function `contract()` in the **igraph** package. You do not need to understand the details.

```
contracted_eu_igraph <-
  eu_netw |>
  as.igraph() |>
  contract(eu_nodes$dept + 1)
vertex_attr(contracted_eu_igraph, "dept") <-
  0:(gorder(contracted_eu_igraph) - 1)
contracted_directed_netw <-
  as_tbl_graph(contracted_eu_igraph) |>
  convert(to_simple) |>
  activate(edges) |>
  mutate(weight = lengths(.tidygraph_edge_index))
```

(6) Make a plot of the contracted network.

```
ggraph(contracted_directed_netw, layout = "stress") +
  geom_edge_fan(
    aes(colour = weight, end_cap = label_rect(node2.dept)),
    alpha = 0.5,
    arrow = arrow(length = unit(0.2, "cm"))
  ) +
  geom_node_label(aes(label = dept), colour = "brown") +
  theme(legend.position = c(1, 1), legend.justification = c(1, 1))
```

Do you find the result useful?

(7) In this and the next step, we write code for an alternative visualisation of the data: a dendrogram that shows how close relations between departments are (figure 1). We can think of the dendrogram as a 'family tree' with the 'root' on the right. In this metaphor, departments that exchange more emails with each other are more closely related. The dendrogram reveals, for example, that department 33 is only distantly related to all other departments. It also shows that departments 0 and 7 are closely related.

As preparation for the dendrogram, create an **undirected** version of `contracted_directed_netw` from task (5). Then turn the undirected network into a **simple** network. Append a column called `weight` to each undirected edge $x \leftrightarrow y$ that is equal to the sum of the weights on the directed edges $x \to y$ and $y \to x$.

Call the result `undirected_simple_netw`.

(8) Use the **tidygraph** function `to_hierarchical_clusters()` with the argument `weights = weight` to obtain a new network that contains the hierarchy of the dendrogram.

```
hierarchy <-
  undirected_simple_netw |>
  convert(to_hierarchical_clusters, weights = weight)
```

(9) Use `ggraph()` to plot the dendrogram similar to figure 1. Here is how the beginning of your code may look like.

```
layout <- create_layout(hierarchy, "dendrogram")
ggraph(layout) +
  geom_edge_elbow() + ...
```

Make a sensible choice for the figure dimensions. Labels should be clearly legible without appearing disproportionately large.

(10) Briefly reflect on the advantages and disadvantages of representing the network data with the dendrogram of figure 1 compared to the network figure you produced in task (6).

(11) An alternative visualisation of the data is the heat map shown in figure 2. (No need to make this plot yourself.) What do you think about that way to represent the intensity of email contacts between departments?

References

Yin, H., Benson, A. R., Leskovec, J., and Gleich, D. F. (2017). Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 555–564, New York, NY, USA. Association for Computing Machinery.

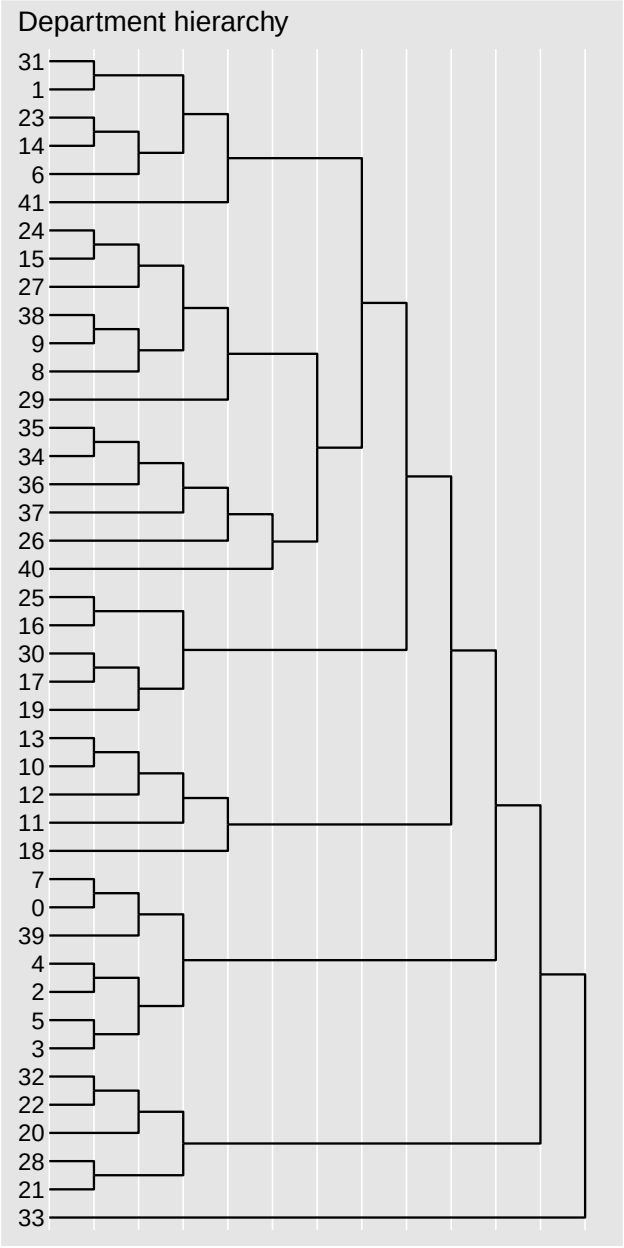

Figure 1: Dendrogram that represents approximate hierarchical clustering of the departments.

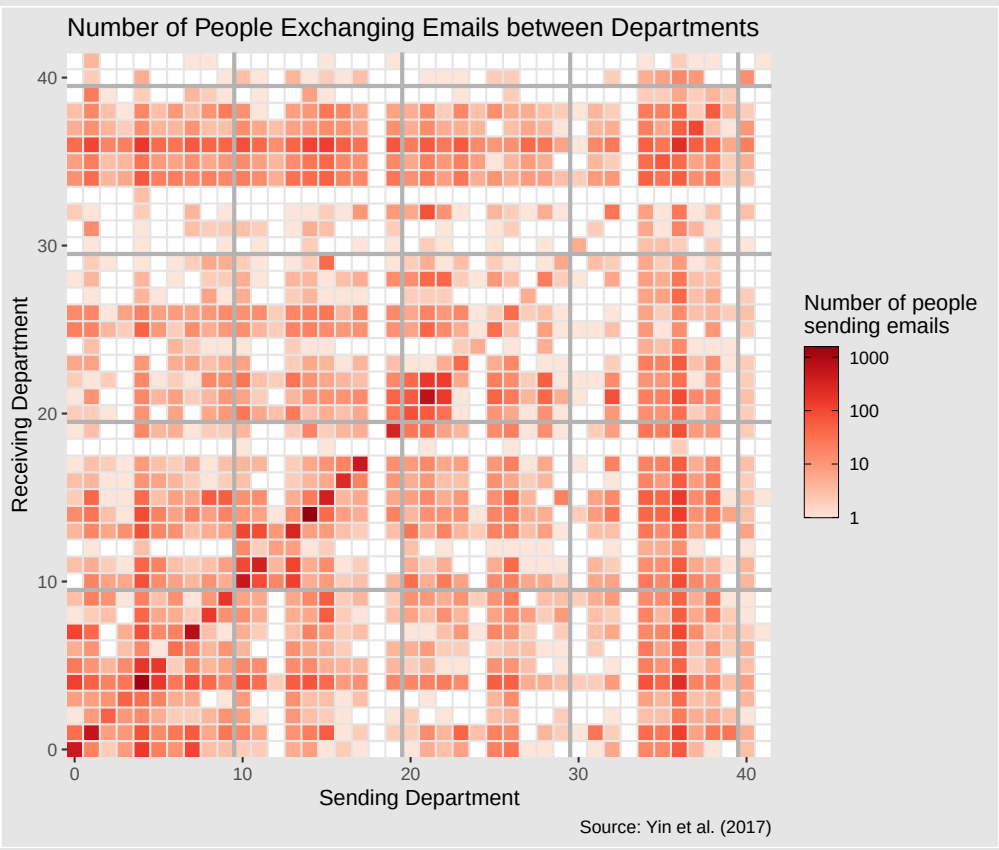

Figure 2: Heatmap of email contacts