Data Validation for Betweeness Centrality based off of ORD

Michael Garbus mgarbus2

2022-12-12

Rationale

As most methods using Betweeness Centrality use a variation of Brandes' Algorithim (Brandes' has made multiple papers for multiple algorithms regarding calculation of Betweeness Centrality) which leads to variations in the exact value, and because some tools simply lack documentation (such as https://github.com/franktakes/teexgraph) and others apply a normalization (https://graph-tool.skewed.de/static/doc/centrality.html#graph_tool.centrality.betweenness) (to be fair, this is recommended in Brandes' later papers and books), it is a little difficult to get an exact 1:1 verification of our Betweeness Centrality, especially since our Betweeness Centrality is implemented for a specific input airport. Because of this, we created our own test file as shown below. We expect to get a very similar, albeit scaled, value of the number of paths for a specific airport divided by the total number of routes.

```
#install.packages('tidyverse')
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyverse)
airports = read_csv("routes.dat", col_names = c("Airline",
'ID',
'Source',
'SourceID',
'Destination',
'DestinationID',
'Codeshare',
"Stops",
"Equipment") )
```

```
## Rows: 67663 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): Airline, ID, Source, SourceID, Destination, DestinationID, Codeshar...
## dbl (1): Stops
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
airports %>% filter(Source %in% "ORD") %>% group_by(Destination) %>% count() %>% arrange(-n)
```

```
## # A tibble: 206 x 2
## # Groups:   Destination [206]
##    Destination     n
##    <chr>       <int>
##  1 ATL            20
##  2 MSY            13
##  3 CDG            10
##  4 LHR             9
##  5 DUS             8
##  6 DUB             7
##  7 FCO             7
##  8 LAX             6
##  9 MEX             6
## 10 NRT             6
## # ... with 196 more rows
```

```
total_flights = airports %>% filter(Source %in% "ORD") %>%
  group_by(Destination) %>% count() %>% arrange(-n)

sum(total_flights$n)
```

```
## [1] 558
```

This is the number of total flights, Not the number of total paths. We expect to get a number somewhat similar to this, but a LOT smaller, sort of a similar ratio.

For example, 1/558 = 0.001792115, and our Betweness Centrality is 0.00016469 for single path centrality. This is a ratio of around 11x, and is expected and indicates we are getting correct values.

Test File

```
ORD_Routes <- airports %>% filter(Source %in% "ORD") %>% group_by(Destination)

write_csv(ORD_Routes, "ORD_Routes.dat") #how the test file is written.
```