

Assignment Social Network Analysis

Meta Infos

Student:in	Jonathan Baß
Titel	Hier könnte der Titel ihrer Arbeit stehen!
Kurskontext	Social Network Analysis
Datum	insertdate
Dozent	Philipp M. Mendoza, M.Sc.

Für weitere Anleitungen und Tipps siehe den Syllabus und den Eintrag zur Portfolioprüfung.

Wichtig: das ganze sollte in der Form eines Blogpostes geschrieben sein; sprich ein Fließtext! Nachfolgend ein Vorschlag der Strukturierung eurer Arbeit; in jedem Report sollten *zumindest* die hier angeführten Punkte abgedeckt werden.

Executive Summary

Dies ist die einzige Sektion die in Bullet points angeführt werden soll. * Einleitung * Forschungsfrage * Datensatz * Strategie * Ergebnisse

Einleitung und Fragestellung

Die Analyse und Nachverfolgung von Kontaktdaten ist in der Medizin und Forschung schon lange ein wichtiges Thema, momentan durch die aktuelle Corona-Pandemie jedoch relevanter als je zuvor. Bei engen oder langen Kontakten können Keime und Erreger zwischen den Personen ausgetauscht werden und sich Krankheiten auf diesem Weg verbreiten. Dies stellt im Alltag kein großes Problem dar, da die meisten Erreger harmlos sind und das menschliche Immunsystem den Ausbruch der Krankheit verhindern kann. Dies ist jedoch in Krankenhäusern nicht immer der Fall. Durch die hohe Konzentration an angewendeten Medikamenten wie Antibiotika können sich schnell multiresistente Keime bilden, welche auch Krankenhauskeime genannt werden. Eine detaillierte Beschreibung und Quantifizierung der Kontakte in Krankenhäusern kann deshalb wichtige Informationen für die Epidemiologie von Krankenhausinfektionen sowie für die Konzeption und Validierung von Kontrollmaßnahmen liefern. Wie diese Vorgenommen werden und welche Schlüsse aus der Analyse gezogen werden können wir in dem folgenden Beitrag erläutern.

Thema

In der geriatrischen Abteilung eines Krankenhauses in Lyon, Frankreich wurde 2010 ein Experiment gemacht. Dafür wurden allen sich auf der Station befindlichen Personen mit tragbare RFID (Radio Frequency Identification) Sensoren ausgerüstet. Diese haben alle Interaktionen im Nahbereich von etwa 1,5m und einer zeitlichen Auflösung von 20 Sekunden über einem Zeitraum von vier Tagen und vier Nächten gemessen. Die Studie umfasste 46 Mitarbeiter des Gesundheitswesens und 29 Patienten und es wurden insgesamt 14.037 Kontakte erfasst.

Daten

Die Daten liegen als ein `igraph` graph-Objekt vor, welches die graph-Attribute “name” und “Citation”, als vertex-Attribut “Status” und als edge-Attribut “Time” besitzt. “Status” beschreibt dabei die Rolle der Person. Dabei wird in Verwaltungspersonal (*ADM*), Ärzte (*MED*), medizinisches Personal wie Krankenschwestern und -pleger oder Hilfspersonal (*NUR*) und Patienten/Patientinnen (*PAT*) unterschieden. “Time” ist der Zeitstempel der Sekunde, an dem das 20-Sekunden Intervall ausgelaufen und nicht erneuert wurde.

Forschungsfrage

Die Übertragung der Krankheiten geschieht durch persönlichen Kontakt. Personen, die viele Kontakte mit einer Vielzahl an Personen haben, können dabei als Superspreader fungieren und Krankheiten schnell verteilen. In der folgenden Ausarbeitung beschäftige ich mich mit der Frage, ob gewisse Rollen ein höheres Risiko haben solch ein Superspreader zu sein und ob die in Krankenhäusern durch organisatorische und räumlich gegebene Bildung von Gemeinschaften einen Einfluss auf eine mögliche Verberitung hat

Relevanz der Forschungsfrage

Die Kontaktnachverfolgung ist durch die aktuelle Corona-Pandemie ein aktuellers Thema denn je. Es gilt geschwächte und anfällige Personen zu schützen. Gerade in Krankenhäusern kann eine unkontrollierte Ausbreitung des SarsCov2-Virus durch einen Superspreader verheerende Auswirkungen haben. Die Analyse und Beantwortung der Forschungsfrage kann bei der Entwicklung eines Schutzkonzeptes helfen und das allgemeine Risiko einer Ansteckung während eines Krankenhausaufenthaltes verringern.

Analysestrategie

- Operationalisierung eurer Forschungsfrage (welche Maße verwendet ihr um eure Forschungsfrage zu beantworten und warum?)
- Charakterisierung des Netzwerks
- Kontext der verwendeten Daten

Umsetzung

Datenmanipulationen

Erklärung der Schritte

1. Pakete installieren

```
# install.packages("igraph", dependencies = T)
# install.packages("igraphdata", dependencies = T)
# install.packages("tidygraph", dependencies = T)
# install.packages("tidyverse", dependencies = T)
# install.packages("ggraph", dependencies = T)
# install.packages("ggthemes", dependencies = TRUE) # a collection of a lot of themes for dataviz!
# install.packages("gganimate", dependencies = TRUE) # an dynamic animation package
# install.packages("gifski", dependencies = TRUE) # an engine to render gifs
# install.packages("rnetcarto", dependencies = TRUE)
```

2. Bibliotheken deklarieren

```
library(igraph)
library(igraphdata)

library(ggraph)
library(ggthemes)
library(gganimate)
library(gifski)

library(tidyverse)
library(tidygraph)

library(RColorBrewer)
library(rnetcarto)
```

3. Alte Daten aus dem Environment entfernen

```
rm(list = ls())
```

4. Daten von igraphdata instanziiieren

```
data("rfid")
```

5. Schleifen und direktionale Beziehungen entfernen und in Variable speichern

```
df <- as.undirected(simplify(rfid))
```

6. igraph-Objekt in adjacency matrix convertieren

```
df.mat=as_adjacency_matrix(df, sparse = F)
head(df.mat, 1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]    0    1    1    1    1    1    1    1    1    1    1    1    1    1
##      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
## [1,]    1    1    1    1    1    1    1    1    1    1    1    1
##      [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
## [1,]    1    1    1    1    1    0    1    0    1    1    1    0
##      [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
## [1,]    0    1    1    1    1    0    1    1    1    1    1    0
##      [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61] [,62]
## [1,]    1    1    1    1    1    0    0    1    0    1    0    1
##      [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72] [,73] [,74]
## [1,]    1    1    1    0    1    1    1    0    1    1    1    1
##      [,75]
## [1,]    0
```

7. Simulierter annealing Algorithmus

```
rnc=netcarto(df.mat)
rnc
```

```
## [[1]]
##   name module connectivity participation    role
## 58  58      0 -1.85366388      0.6111111 Peripheral
## 59  59      0 -1.60252233      0.6562500 Connector
## 75  75      0 -1.10023921      0.6115702 Peripheral
## 72  72      0 -0.84909765      0.6975309 Connector
## 44  44      0 -0.84909765      0.7100000 Connector
## 43  43      0 -0.84909765      0.7296786 Connector
## 56  56      0 -0.59795609      0.6400000 Connector
## 61  61      0 -0.59795609      0.6640625 Connector
## 53  53      0 -0.59795609      0.7107750 Connector
## 54  54      0 -0.34681453      0.6171875 Peripheral
## 55  55      0 -0.09567297      0.6621315 Connector
## 36  36      0  0.40661014      0.6639232 Connector
## 62  62      0  0.40661014      0.7285156 Connector
## 10  10      0  0.65775170      0.7436050 Connector
## 57  57      0  0.90889326      0.5495868 Peripheral
## 30  30      0  0.90889326      0.7281380 Connector
## 25  25      0  0.90889326      0.7398594 Connector
## 21  21      0  0.90889326      0.7456597 Connector
## 26  26      0  1.16003482      0.7321949 Connector
## 64  64      0  1.16003482      0.7327951 Connector
## 23  23      0  1.91345949      0.7390012 Connector
## 66  66      1 -1.77175914      0.6913580 Connector
## 70  70      1 -1.02575529      0.5400000 Peripheral
## 32  32      1 -1.02575529      0.6224490 Connector
## 40  40      1 -1.02575529      0.7313019 Connector
## 39  39      1 -1.02575529      0.7396694 Connector
## 38  38      1 -0.27975144      0.5680473 Peripheral
## 34  34      1 -0.27975144      0.6015625 Peripheral
## 69  69      1 -0.27975144      0.7258979 Connector
## 48  48      1  0.09325048      0.7436085 Connector
## 49  49      1  0.09325048      0.7491349 Connector
## 8   8       1  0.46625240      0.6426593 Connector
## 3   3       1  0.46625240      0.7297668 Connector
## 5   5       1  1.21225625      0.7457851 Connector
## 24  24      1  1.21225625      0.7466667 Connector
## 6   6       1  1.58525817      0.7427160 Connector
## 37  37      1  1.58525817      0.7487245 Connector
## 67  67      2 -2.36081759      0.5714286 Peripheral
## 47  47      2 -1.29282868      0.5680473 Peripheral
## 68  68      2 -1.29282868      0.7372401 Connector
## 50  50      2 -0.75883423      0.6530612 Connector
## 41  41      2 -0.75883423      0.6818182 Connector
## 42  42      2 -0.75883423      0.7071006 Connector
## 51  51      2 -0.49183700      0.7327824 Connector
## 2   2       2 -0.22483977      0.7015306 Connector
## 52  52      2  0.04215746      0.6862245 Connector
```

```
## 12 12 2 0.04215746 0.6923783 Connector
## 4 4 2 0.30915469 0.7076125 Connector
## 20 20 2 0.30915469 0.7312061 Connector
## 19 19 2 0.57615191 0.7377778 Connector
## 33 33 2 0.84314914 0.7187500 Connector
## 22 22 2 1.11014637 0.6975000 Connector
## 27 27 2 1.11014637 0.7321949 Connector
## 17 17 2 1.11014637 0.7417667 Connector
## 7 7 2 1.11014637 0.7442290 Connector
## 29 29 2 1.37714360 0.7404337 Connector
## 46 46 3 -1.61413862 0.4938272 Peripheral
## 71 71 3 -1.61413862 0.6600000 Connector
## 60 60 3 -1.61413862 0.6938776 Connector
## 14 14 3 -0.82776339 0.6122449 Peripheral
## 73 73 3 -0.82776339 0.7352538 Connector
## 74 74 3 -0.56563832 0.6893424 Connector
## 28 28 3 -0.30351324 0.6840278 Connector
## 18 18 3 -0.30351324 0.6976000 Connector
## 31 31 3 -0.30351324 0.7270408 Connector
## 65 65 3 0.22073690 0.6322314 Connector
## 35 35 3 0.22073690 0.7409972 Connector
## 45 45 3 0.22073690 0.7471743 Connector
## 63 63 3 0.48286198 0.7000000 Connector
## 9 9 3 0.74498705 0.6953125 Connector
## 16 16 3 0.74498705 0.7312061 Connector
## 13 13 3 0.74498705 0.7387153 Connector
## 15 15 3 1.26923720 0.7404770 Connector
## 11 11 3 1.53136228 0.7224000 Connector
## 1 1 3 1.79348735 0.7385112 Connector
##
## [[2]]
## [1] 0.1027329
```

8. Neues igraph-Objekt mit Index als Spalte erstellen

```
df %>% as_tbl_graph() %>% activate(nodes) %>% mutate(name = row_number()) -> df2
df2
```

```
## # A tbl_graph: 75 nodes and 1139 edges
## #
## # An undirected simple graph with 1 component
## #
## # Node Data: 75 x 2 (active)
##   Status name
##   <chr> <int>
## 1 ADM      1
## 2 NUR      2
## 3 NUR      3
## 4 NUR      4
## 5 NUR      5
## 6 NUR      6
## # ... with 69 more rows
```

```
## #
## # Edge Data: 1,139 x 2
##   from    to
##   <int> <int>
## 1     1     2
## 2     1     3
## 3     1     4
## # ... with 1,136 more rows
```

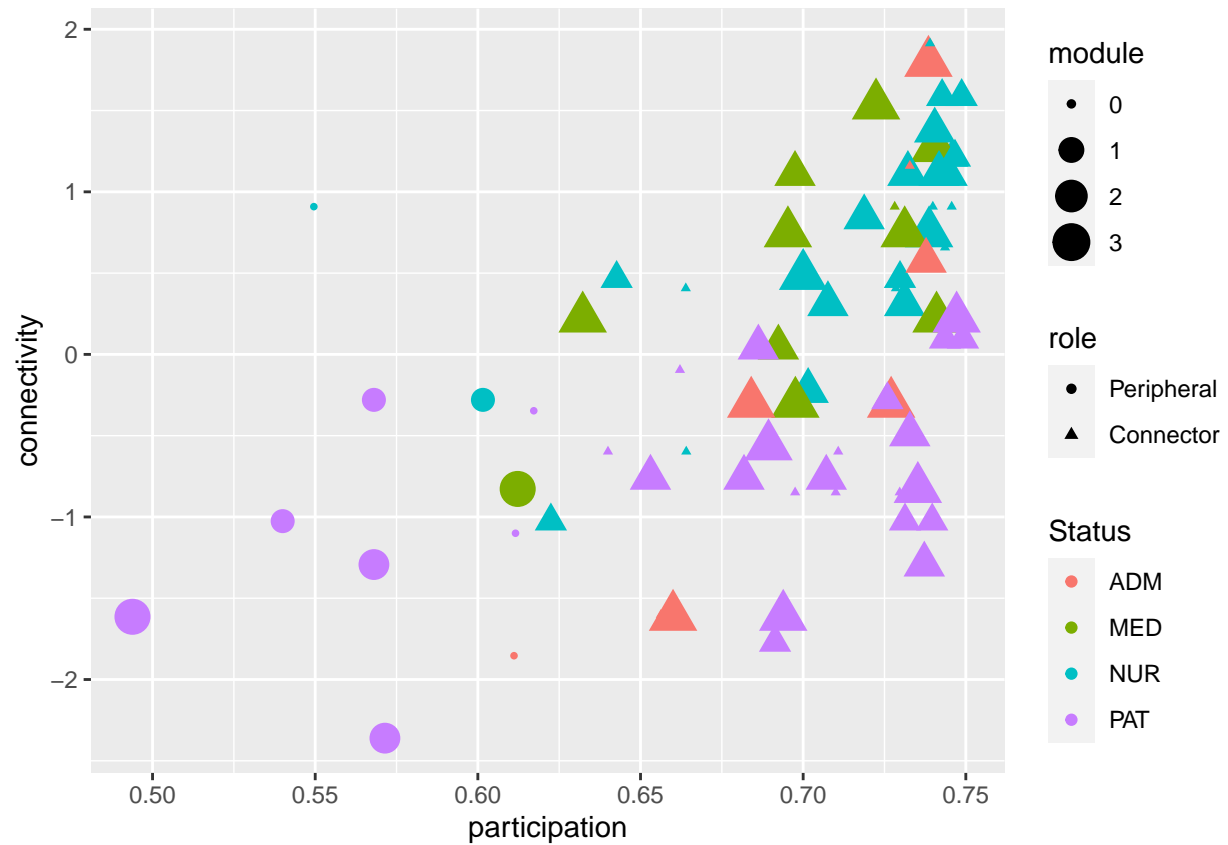
9. Status mergen

```
test <- merge(x = df2 , y = rnc[[1]], by = "name", all = TRUE)
head(test, 10)
```

	name	Status	module	connectivity	participation	role
## 1	1	ADM	3	1.7934874	0.7385112	Connector
## 2	2	NUR	2	-0.2248398	0.7015306	Connector
## 3	3	NUR	1	0.4662524	0.7297668	Connector
## 4	4	NUR	2	0.3091547	0.7076125	Connector
## 5	5	NUR	1	1.2122563	0.7457851	Connector
## 6	6	NUR	1	1.5852582	0.7427160	Connector
## 7	7	NUR	2	1.1101464	0.7442290	Connector
## 8	8	NUR	1	0.4662524	0.6426593	Connector
## 9	9	MED	3	0.7449871	0.6953125	Connector
## 10	10	NUR	0	0.6577517	0.7436050	Connector

10. Plot

```
ggplot(test, aes(y = connectivity, x = participation)) +
  geom_point(aes(color=Status, shape=role, size=module))
```



Interpretation der Visualisierungen

Conclusio

Wiederholung der Fragestellung

Zusammenfassung der zentralen Ergebnisse

(*Limitationen, weiterführende Kommentare, etc.*)