Amazon Network Analysis

Assignment im Rahmen der Vorlesung 'Social Network Analyis'

Ferdinand Bubeck

2021-11-15

Inhaltsverzeichnis

1 Einleitung

1.1 Zielsetzung

1.2 Vorgehensweise

Als Vorgehensweise wird in diesem Projekt das für das Feld Data Science etablierte Standard-Vorgehen CRISP-DM gewählt (Cross Industry Standard Process for Data Mining). In mehreren Phasen werden so von dem richtigen Verständnis der Daten, dem Data Wrangling und Data Preprocessing bis hin zum Modelfitting und der Evaluation alle entscheidenen Schritte strukturiert durchlaufen, um ein optimales Ergebnis aus den Daten zu generieren. In der Abbildung 1 ist das Vorgehensmodell abgebildet. Da es sich in diesem Projekt um ein PoC handelt, wird die letzte Phase 'Deployment' ausgelassen.

Figure 1: CRISP-DM (Source: https://statistik-dresden.de/archives/1128))

2 Hauptteil

2.1 Business Understanding

- Welche Produkte werden nur in Verbindung mit anderen Produkten gekauft?
- Welche Produkte sind zentral?

2.1.1 Laden der Libraries

```
library("tidyverse")
library("tidygraph")
library("igraph")
library("ggraph")
```

2.1.2 Importieren der Daten

Die Daten stammen aus der Datensatz-Bibliothek der Stanford University und können als .txt unter folgendem Lin heruntergeladen werden. (Link: https://snap.stanford.edu/data/amazon0302.html)

Zum Einlesen der Daten kommt im Folgenden die Funktion *read.table* zum Einsatz.

```
amazon <- read.table("Data/Amazon0302.txt")
```

2.2 Data Understanding

Die Daten basieren auf dem Grundsatz "Kunden, die Artikel A gekauft haben, haben auch Artikel B gekauft". Wenn ein Produkt i häufig zusammen mit Produkt j gekauft wird , enthält der Graph eine gerichtete Kante von i nach j .

Um einen ersten Einblick in die Daten zu erhalten, wird mit der Funktion *head* die ersten Zeilen des Datensatzes ausgegeben. Zusätzlich dazu ist es von entscheidender Rolle, die Qualität der Daten zu bewerten. Aus diesem Grund werden alle fehlenden Werte, sogenannte NAs gezählt und ausgegeben.

```
head(amazon)
```

```
##   V1 V2
## 1  0  1
## 2  0  2
## 3  0  3
## 4  0  4
## 5  0  5
## 6  1  0
```

```
# Count NAs
which(is.na(amazon))
```

```
## integer(0)
```

Der Dataframe besteht aus 3 Spalten: einer ID Spalte, und zwei Kantenspalten. Des Weiteren weisen die Daten keine Lücken und fehlenden Werte auf, sodass der komplette Datensatz für das weitere Vorgehen genutzt werden kann.

2.3 Data Preparation

Auf Basis der vorangegangen Schritte müssen nun weitere Anpassungen der Daten erfolgen, um damit arbeiten zu können. Zum Einen werden die Kantenspalten von ihren ursprünglichen Namen in sprechendere Bezeichnungen umbenannt. Im gleichen Schritt werden alle Werte um 1 erhöht, sodass keine Nullen mehr existieren.

```
dat <- amazon %>%
  rename(
    from = V1,
    to = V2
  ) %>%
  mutate(
    from = from+1,
    to = to+1
  )
```

2.4 Modeling

Nach der Datenbearbeitung kann nun das Netz gefittet werden. Hierzu wird die Funktion *as tbl graph* angewendet, um ein Netz zu erstellen.

```
net <- as_tbl_graph(dat)
net
```

```
## # A tbl_graph: 262111 nodes and 1234877 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 262,111 x 1 (active)
##   name
##   <chr>
## 1 1
## 2 2
## 3 3
```

```
## 4 4
## 5 5
## 6 6
## # ... with 262,105 more rows
## #
## # Edge Data: 1,234,877 x 2
##    from    to
##   <int> <int>
## 1     1     2
## 2     1     3
## 3     1     4
## # ... with 1,234,874 more rows
```

Die beiden Spalten aus dem Ursprungsdatensatz wurden in ein Netz, bestehend aus 262111 Knoten und 1234877 Kanten, konvertiert. Es handelt sich, wie aus der Zusammenfassung des Netzes zu entnehmen ist, um einen gericheteten Graphen. Die Knotennamen sind in diesem Fall die Ziffern der Kantendaten. Leider liegt dem Autor dieser Arbeit keine Zuordnungsliste von Knotenziffern zu realen Amazonprodukten vor. Aus diesem Grund wird im Folgenden mit den Ziffern der Knoten weitergearbeitet.

```
# Calculate Degree of Vertices
degree <- degree(net)

# Adjacency Matrix
adjacencyMatrix <- net[]
```

Aus dem Netz kann nun der Degree abgeleitet und abgespeichert werden. Der Degree oder Grad eines Knoten ist die Anzahl von Kanten, die an ihn angrenzen. Für die Analyse ist die Verteilung der Grade der Knoten interessant. Gibt es eine überwiegende Mehrheit an Knoten, welche die gleiche Anzahl an Kanten besitzen? Gibt es Ausreißer mit vielen Kanten? Ähnelt die Verteilung einer Normalverteilung, ist die links oder rechts verschoben?
Um diese Fragen zu beantworten, wird im nächsten Schritt ein Histogramm erzeugt, welches die Knotengrade des Netzwerkes visualisiert.

2.5 Data Visualization

Um die Degrees für die Visualisierung nutzen zu können, müssen diese zuvor in ein Dataframe umgewandelt werden. Dies geschieht mit der Funktion *as.data.frame*. Anschließend wird die Library *ggplot2* für das Histogramm angewendet.

```
degree_df <- as.data.frame(degree)


hist_of_degrees <- ggplot(data = degree_df, aes(x=degree))+
```

```
  geom_bar(fill = "#e2001a", colour = "#e2001a", alpha=.5)+
  scale_y_continuous(trans='log10')+
  xlim(0,120)+
  labs(title = "Histogram of Node-Degrees",
       subtitle = "Amazon Network Analysis",
       y = "Frequency (log10 scale)",
       x = "Degree of Vertices (xlim = 120)")+
  theme_classic()

hist_of_degrees
```

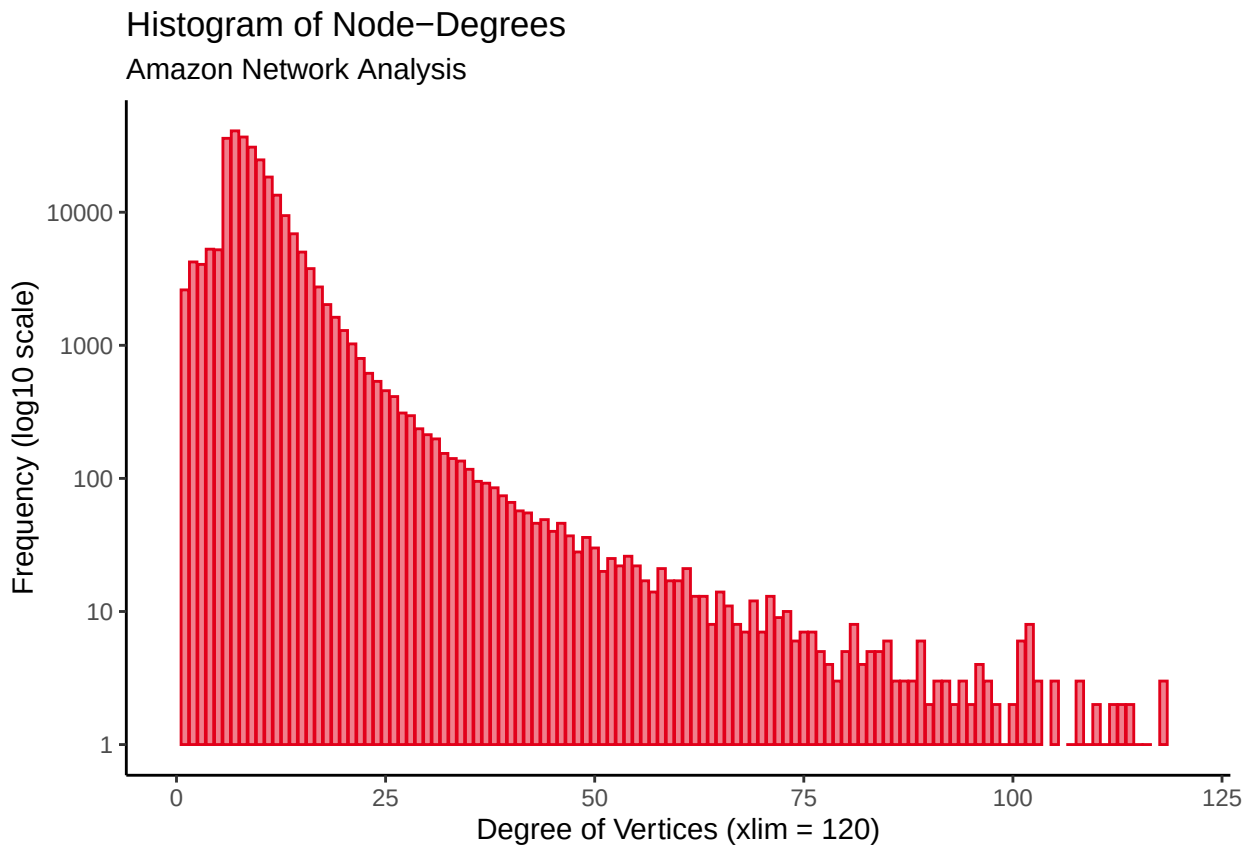

Figure 2: Knotengrad Histogramm

Das Histogramm der Knotengrade zeigt eine Mehrheit der Grade im Bereich 5-20. Dies bedeutet, dass eine Mehrheit der Knoten im Datensatz eine durchschnittliche Anzahl an Kanten von 5-20 aufweist. Weiterhin ist zu erkennen, dass einige Knoten 100 und mehr Kanten besitzen. Die großen Ausreißer wurden in diesem Plot weggelassen, doch selbst in dieser Darstellungsweise zeigt sich ein abflachender Bereich Richtung x –> unendlich. Um eine übersichtlichere Darstellung der Observationen um den Nullbereich der y-Achse zu gewährleisten, wurde die y-Achse nach dem dekadischen Logarithmus skaliert.

2.6 Experimental Data

```
# Subsetting Data
dat_exp <- dat[1:200,]

net_exp <- as_tbl_graph(dat_exp)

net_exp <- net_exp %>%
  activate(nodes) %>%
  mutate(
    degree = centrality_degree()
  )
```

```
# Data Viz for Subset
# network diagramm
ggraph(net_exp, layout = 'fr', maxiter = 100) +
  geom_node_point(colour="#e2001a") +
  geom_edge_link(alpha = .4) +
  theme_graph()
```

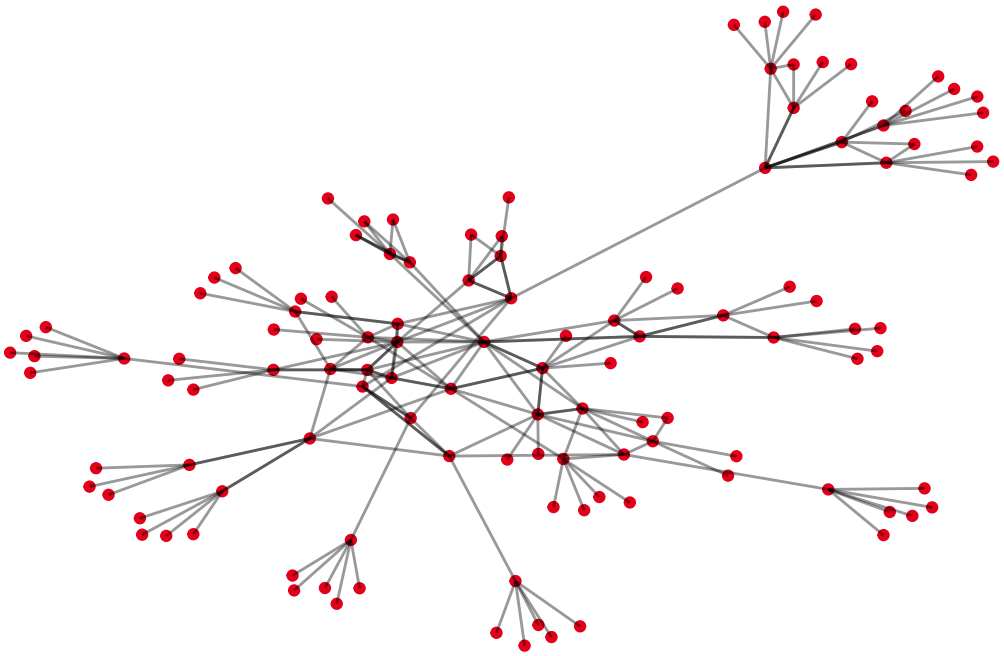

Figure 3: Netzwerk Visualisierung 1

```
ggraph(net_exp, layout = 'kk', maxiter = 100) +
  geom_node_point(colour="#e2001a") +
  geom_edge_link(alpha = .4) +
  theme_graph()
```

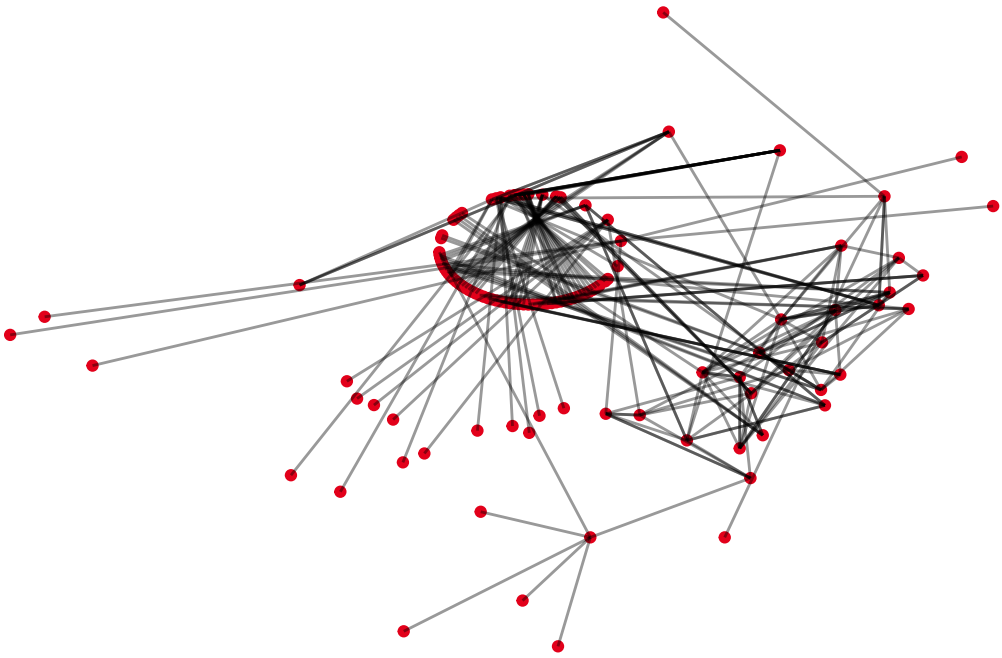

Figure 4: Netzwerk Visualisierung 2

```
# coord diagramm
ggraph(net_exp, layout = 'linear', circular = TRUE) +
  geom_node_point(colour="#e2001a") +
  geom_edge_arc(alpha = .4) +
  theme_graph()
```

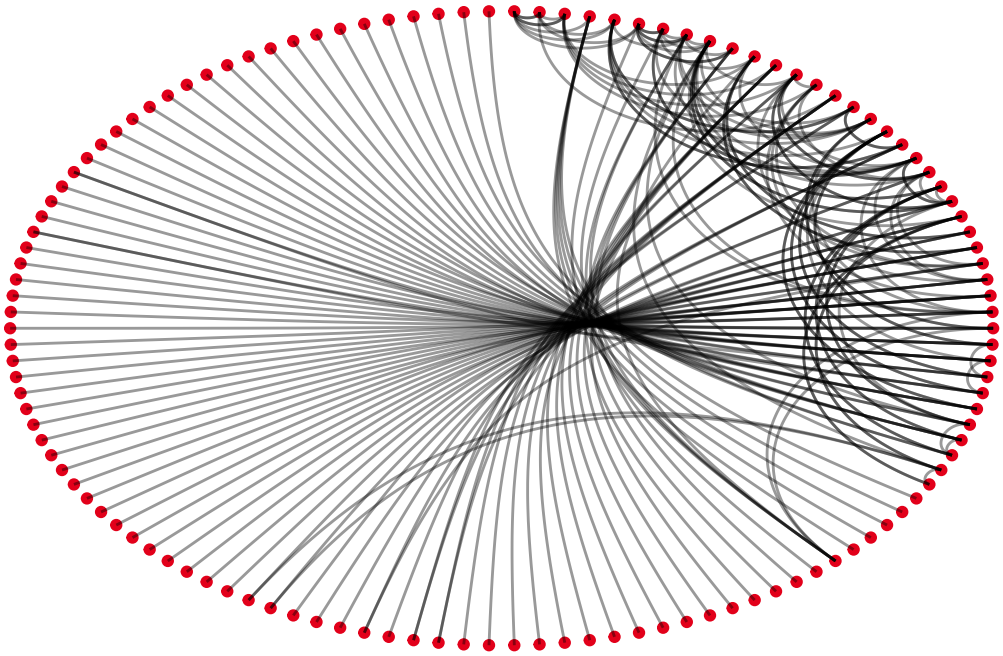

Figure 5: Netzwerk Visualisierung 3

3 Fazit

3.1 Evaluation der Ergebnisse

tbd

3.2 kritische Reflexion

tbd