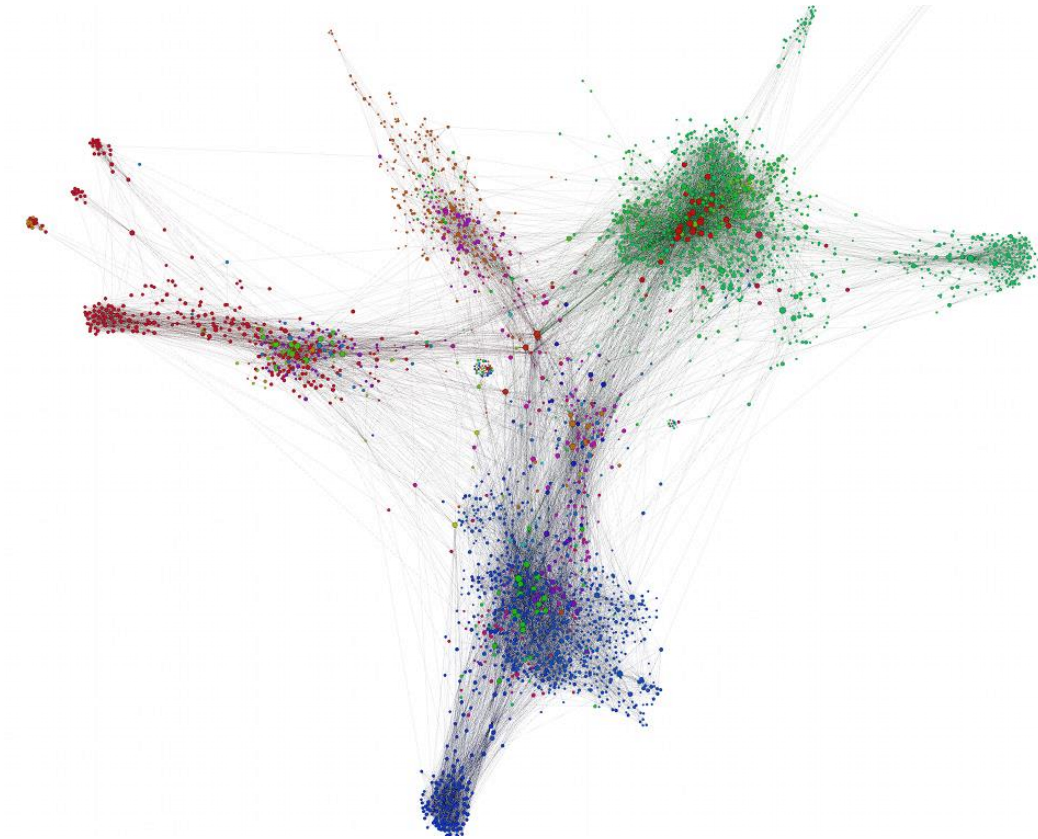


Script: Soziale Netzwerk Analyse



Autor: Michael Henninger
Jahr: 2015

Datum	Änderung
09.08.2015	Update der Referenzen, Überarbeitung des Inhalts
06.08.2015	Neue Themen wie: Datenerhebung, Modellierung, Tie-Strength, Eigenvector-Zentralität, Bridge & Brokers, Netzwerk-Reduktion
09.10.2014	Titelbild, Grafik-Fehler bei Hierarchischem Clustering behoben, Hinzufügen von Netzwerk Zentralitätsmassen
06.10.2014	Erste Version, inkl. Änderungsvorschläge von S. Schnorf

Inhalt

Lernziele	4
Grundlagen	5
Datenerhebung	5
Umfragen.....	5
Beobachtungen	5
Schriftliche Nachweise	5
Abbildung des Sozialen Netzwerks als Graph	6
Modellierung von Sozialen Netzwerken.....	7
One-Mode Netzwerk	7
Multirelational	8
Signed Netzwerk.....	8
Two-Mode Netzwerk.....	8
Tie-Strength	9
Connected Components.....	10
Aktor-Zentralität.....	12
Degree Centrality	12
Closeness Centrality	13
Betweenness Centrality.....	14
Eigenvector Zentralität	17
Netzwerk-Zentralisierung.....	18
Degree Zentralisierung	19
Betweenness Zentralisierung	19
Closeness Zentralisierung.....	20
Prestige.....	22
Indegree (Popularity).....	22
Proximity Prestige	22
Bridges und Brokers	24
Netzwerk-Metriken	25
Graph Density.....	25
Graph Diameter	25
Cluster Coefficient	26
Reduktion des Netzwerks.....	28
Local View.....	28
Global View.....	28

Contextual View	28
Ego Netzwerk.....	28
Filtern anhand von Katen-Attributen	28
Communities	28
Clustering – Auffinden von Communities.....	31
Hierarchical Clustering	31
Edge-Betweenness Clustering	32
Small World	35
Experimente	35
Ursprüngliches Paket-Experiment von Milgram	35
Co-Authorship Experiment von Paul Erdős	35
MSN Studie von Microsoft	35
Eigenschaften von Small World Netzwerken	35
Referenzen	37

Lernziele

Nach den Unterrichtseinheiten und dem Selbststudium sollten folgende Ziele erreicht werden:

- Sie kennen die Grundbegriffe im Zusammenhang mit Graphen
- Sie kennen die verschiedenen Möglichkeiten der Netzwerk-Modellierung und können sich situativ für ein geeignetes Modell entscheiden, um ein Soziales Netzwerk abzubilden.
- Sie können die wichtigsten Aktor- & Netzwerk-Zentralitätsmasse, Prestige-Masse und Graph-Metriken von Hand berechnen und das Resultat interpretieren.
- Sie kennen den Begriff Small World und können entscheiden, ob es sich bei einem gegebenen Graphen um einen Small-World Graphen handelt.
- Sie können ein Netzwerk so reduzieren, dass es sinnvoll analysiert werden kann.
- Sie können einen Graphen in Gephi analysieren. (Dazu gehört neben der Berechnung und Interpretation von Zentralitätsmassen und Metriken auch die Anwendung von Filtern, um das Netzwerk sinnvoll zu reduzieren.)

Grundlagen

Datenerhebung

Bevor mit der Analyse von Sozialen Netzwerken begonnen werden kann, müssen die zu untersuchenden Daten eingesammelt werden. Dazu gibt es verschiedene Möglichkeiten, welche in einer kurzen Einführung kurz erläutert werden. Quelle:

Umfragen

Umfragen (z.B. durch Interviews oder Fragebögen) sind ein gängiges Mittel zur Erhebung von Sozialen Netzwerkdaten bei Personen. Mögliche Fragen wären hier:

- „Mit wem bist du befreundet?“
- „Bei wem holst du Rat ein?“
- „Welche 3 Personen haben dein Leben geprägt?“

Die Daten können entweder symmetrisch oder asymmetrisch behandelt werden. Ein Beispiel für symmetrisch wäre: Falls eine Person A aussagt, dass sie mit Person B befreundet ist, dann wird automatisch angenommen, dass Person B auch mit Person A befreundet ist. Bei asymmetrischen Datenerhebungen würde im vorliegenden Fall nur eine Freundschaftsbeziehung von A nach B, aber nicht von B nach A abgebildet werden. Es muss nach Anwendungsfall entschieden werden, ob es sich bei den untersuchten Beziehungen eher um symmetrische oder asymmetrische Beziehungen handelt. Besteht die Möglichkeit ohne grösseren Mehraufwand eine asymmetrische Analyse durchzuführen empfiehlt sich diese, da so eine genauere Einsicht in die Daten und mehr Analysemöglichkeiten offen bleiben.

Dem Befragten können unterschiedliche Antwortmöglichkeiten gegeben werden. Wird beispielsweise eine Person nach deren Freunden gefragt, kann dies verschieden formuliert werden:

- Freie Auswahl: Nennen Sie mir all ihre engeren Freunde
- Beschränkte Auswahl: Nennen Sie mir all ihre engeren Freunde, die sich in der folgenden Liste von Personen befindet.
- Fixe Anzahl: Nennen Sie mir ihre drei besten Freunde
- Rangfolge: Nennen Sie mir Ihre besten Freunde in absteigender Reihenfolge (bester Freund zuerst).

Der Nachteil von Befragungen ist, dass Personen bei heiklen und privaten Themen sehr zurückhaltend sind und möglicherweise unehrliche Antworten geben.

Beobachtungen

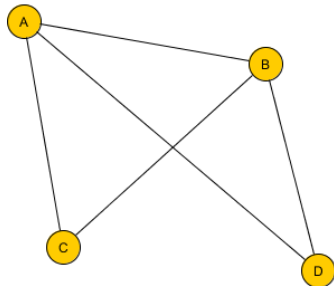
Die Interaktion zwischen Akteuren kann auch über Beobachtungen ermittelt werden. Der Nachteil ist, dass hier die Art der Interaktion durch die alleinige Beobachtung nicht immer eindeutig ermittelbar ist. Vorteilhaft ist jedoch, dass dadurch nicht viele Leute befragt werden müssen und somit die Wahrscheinlichkeit für „unehrliche Angaben“ sinkt. Beispielsweise würden wohl die wenigsten Leute eine Teilnahme an illegalen Veranstaltungen zugeben. Hier würde die Beobachtung der Teilnehmer einen ehrlicheren Einblick geben.

Schriftliche Nachweise

Die sozialen Beziehungen können anhand von vorhandenen Daten abgeleitet werden. Als Quelle dienen beispielsweise Daten von Online-Plattformen, E-Mails, SMS usw. Es ist nicht immer ganz einfach an diese Daten zu kommen, jedoch ermöglichen sie sehr detaillierte Analysen.

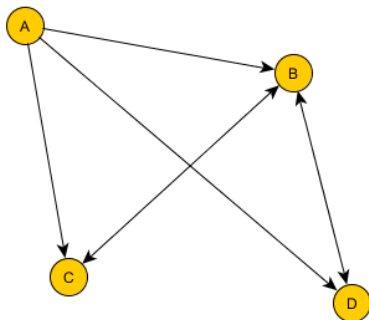
Abbildung des Sozialen Netzwerks als Graph

Ein soziales Netzwerk wird als Graph modelliert und besteht somit aus Knoten (auch *Aktoren* oder *Nodes* genannt) und Kanten (*Edges, Ties*), welche die Interaktion zwischen den Aktoren abbildet. Im folgenden kleinen Beispiel-Graphen sind die Knoten A bis D dargestellt. Die Kanten zwischen den Knoten sind hier ungerichtet was bedeutet, dass die Interaktion in beide Richtungen möglich ist.



Ein Beispiel für solche ungerichtete Graphen (ungerichtete Kanten, auch bidirektionale Kanten genannt) wäre das Facebook Freundes-Netzwerk. Da sind entweder Personen miteinander befreundet oder nicht, aber eine Person A kann nicht mit der Person B befreundet sein, wenn B nicht auch mit A befreundet ist. (Hingegen in Google Plus wäre so eine Konstellation denkbar, da ich jemanden in einen „Kreis“ hinzufügen kann, diese Person muss mich jedoch nicht in einen seiner Kreise hinzufügen). Im obigen Beispiel wäre die Person A mit allen drei anderen Personen befreundet. Die Person C und D jedoch nur mit 2 von den 3 möglichen Personen, da sie untereinander nicht befreundet sind.

Ein Beispiel für gerichtete Graphen wäre ein E-Mail Netzwerk. Hier visualisiert die Pfeilrichtung der Kante, wer wem eine E-Mail schreibt. Deshalb sind die Kanten gerichtet:



Es ist ersichtlich, dass die Person A nur E-Mails verschickt, aber keine empfängt. Dies wäre bei einem Newsletter denkbar, der nur informiert. Person B kommuniziert mit der Person C und D, die Personen C und D jedoch nicht miteinander. Ein solches Konstrukt könnte entstehen, wenn die Personen C und D zu verschiedenen Teams gehören (z.B. Teamleiter sind) und die Person B beide kennt.

In der Sozialen Netzwerkanalyse stellen die Kanten eine direkte oder indirekte Kommunikation / Verbindung zwischen verschiedenen Aktoren dar. Einer Kante kann dabei verschiedene Bedeutungen zugetragen werden. Mögliche Bedeutungen:

- Informationsaustausch
- Ressourceaustausch
- Beeinflussung
- Mitgliedschafts-Beziehung
- Verwandschafts-Beziehung
- Persönliche Beziehung
- usw.

Hier ein paar Beispiele, wie soziale Netzwerke anhand verschiedener Daten konstruiert werden können:

Quelle	Bedeutung der Knoten	Bedeutung der Kanten
E-Mail Kommunikation	Personen (die E-Mails empfangen oder senden)	E-Mails zwischen den Personen
Facebook Freundesbeziehungen	Personen (Freunde)	Freundschaftsbeziehungen zwischen Freunden
Tweets (Retweets)	Twitternde Personen	Tweets (zwischen der tweetenden und retweeteten Person)
Wikipedia Artikelnetzwerk	Wikipedia-Artikel	Beziehungen zwischen versch. Wikipedia-Artikeln
Webseiten Verlinkung	Webseiten	Links zwischen Webseiten

Knoten und Kanten können verschiedene Eigenschaften besitzen. Hier ein paar Beispiele:

- Facebook-Netzwerk: Personen (Knoten) können Herkunft, Alter, Geschlecht usw. als Eigenschaften hinterlegt haben.
- E-Mail Netzwerk: Kanten (E-Mail) können Inhalt des Mails, Subject oder Datum beinhalten
- Tweet-Netzwerk: In den Knoten können Eigenschaften zur Person (Herkunft, Name, Beschreibung usw.), in den Kanten der Tweet selbst sowie das Datum abgelegt werden.
- Personen-Netzwerke: Als Kantengewicht kann angegeben werden, wie stark zwei Personen miteinander befreundet sind.

Oftmals beinhalten die Kanten ein Gewicht (z.B. Anzahl Benachrichtigungen untereinander). Bei solchen Graphen wird auch von **gewichteten Graphen** gesprochen.

Die Netzwerk-Struktur sowie die Eigenschaften von Knoten/Kanten lassen interessante Analysen zu. Zum einen bietet die Netzwerk-Struktur bereits viel Potenzial, um Schlüsselpersonen durch Berechnung von Zentralitätsmassen zu erkennen. Zum anderen kann das Netzwerk anhand der vorhandenen Eigenschaften der Knoten und Kanten gefiltert oder die verschiedenen Eigenschaften unterschiedlich dargestellt werden.

Modellierung von Sozialen Netzwerken

Es gibt verschiedene Möglichkeiten, die gesammelten Daten als Soziales Netzwerk in einem Graphen zu modellieren. Aus einer Datenquelle können auch verschiedene Netzwerk-Modellierungen resultieren.

One-Mode Netzwerk

In einem One-Mode Netzwerk kann jeder Knoten mit jedem anderen verbunden sein. Es gibt nur ein Knotentyp. Ein paar Beispiele:

- E-Mail Netzwerk: Knoten sind Personen und alle Personen können miteinander verbunden sein, da alle einander E-Mails schreiben können.
- Freundschafts-Netzwerk: Knoten sind Personen, beliebige Personen können miteinander befreundet sein.
- Webseiten: Webseiten (hier als Aktoren verwendet) können mit beliebigen anderen Aktoren verknüpft sein.

Multirelational

In einem Multirelationalen Netzwerk werden mehrere Beziehungstypen im selben Netzwerk abgebildet. Für eine genauere Analyse ist es dann erforderlich die Relationstypen wieder zu reduzieren. Jedoch ist es interessant aus einem ursprünglichen Multirelationalen Netzwerk verschiedene Netzwerke einzelner Relationstypen zu extrahieren und die Analysen dieser einzelnen Subnetzwerke wieder miteinander zu vergleichen.

Signed Netzwerk

Ein Signed Graph ist ein Graph, dessen Kanten entweder in positives oder negatives Vorzeichen haben.

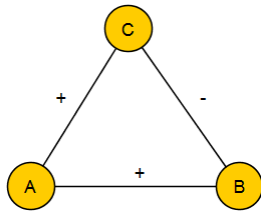


Abbildung 1: Signed Netzwerk

Two-Mode Netzwerk

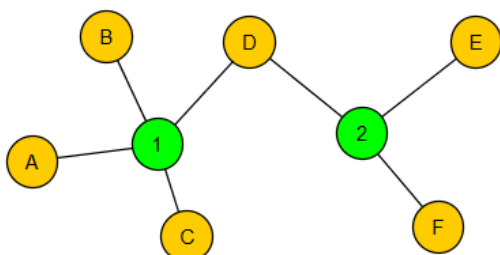
Bei Two-Mode Netzwerken existieren im Gegensatz zu den One-Node Netzwerken zwei verschiedene Typen von Knoten. Kanten existieren nur zwischen Knoten verschiedener Typen (auch Bipartites Netzwerk genannt). Ein paar Beispiele für Two-Mode Netzwerke:

- Personen-Event-Netzwerk: Es wird modelliert, welche Personen an welchen Events teilnehmen.
- Personen auf Webseiten: Personen-Knoten sind verknüpft mit denjenigen Webseiten-Knoten, auf welchen deren Namen erscheint.

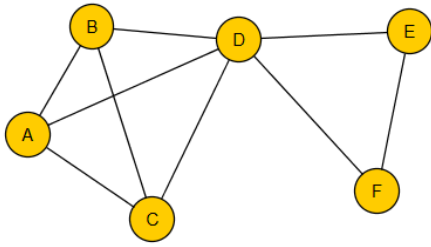
Interessant ist, dass die Anzahl Kanten eines Knotens angeben, wie viele Verbindungen zum anderen Knotentyp bestehen unter der Voraussetzung, dass keine parallele Kanten existieren.

Beachten Sie, dass später bei verschiedenen Netzwerk-Metriken zwischen One- und Two-Mode Netzwerken unterschieden werden muss. Im weiteren Verlauf des Scripts wird immer von einem One-Mode Netzwerk ausgegangen. Diese Analysen (Netzwerk-Metriken, Zentralitätsmasse) für das One-Mode Netzwerk können nicht ohne Anpassung auf ein Two-Mode Netzwerk anwenden.

Ein Two-Mode Netzwerk lässt sich jedoch in ein One-Mode Netzwerk transformieren. Dazu gibt's verschiedene Möglichkeiten. Die einfachste ist, dass alle mit einem Knoten (Typ A) verbundenen Knoten (Typ B) untereinander verbunden werden. Es entsteht eine sogenannte Clique. Hier ein Beispiel: Es ist ein Two-Mode Netzwerk abgebildet welches zeigt, welche Personen (gekennzeichnet mit Buchstaben) in welchen Vereinen (gekennzeichnet mit Zahlen) tätig sind:



Nun wird dieses Two-Mode Netzwerk nach obiger Beschreibung in ein One-Mode Netzwerk umgewandelt:



Da die Person D in beiden Vereinen tätig ist, ist sie nun mit allen anderen Knoten verbunden. Wie später noch ersichtlich wird, hat sie somit aus verschiedenen Gründen eine strategisch wichtige Position im Gesamtnetzwerk.

Tie-Strength

Ein besonderes Konzept von gewichteten Kanten wurde von Mark Granovetter eingeführt. Er bezeichnete die Intensität der Beziehung zwischen Personen als *Tie Strength*. Diese Stärke einer Beziehung wurde anhand von vier Komponenten definiert:

- Wie viel Zeit zwei Personen miteinander verbringen
- Grad der emotionalen Intensität der Beziehung
- Gegenseitiges Vertrauen
- Art der bidirektionalen Hilfeleistungen

Granovetter teilte die *Tie Strength* grob in zwei Kategorien ein:

- **Strong Ties:** Familie und sehr enge Freunde (sehen sich häufig, teilen gleiche Interessen)
- **Weak Ties:** Weniger enge Freunde wie z.B. Arbeitskollegen

Personen, mit denen man überhaupt keinen persönlichen Kontakt pflegt (z.B. Verkäufer) werden in diesem Modell nicht abgebildet. Es wird in diesem Fall auch von *absent Tie* gesprochen.

Die Untersuchung der Tie-Stärke spielt in der Sozialen Netzwerk Analyse eine bedeutende Rolle. Gruppen von Personen, welche über Strong Ties verbunden sind, pflegen ein grosses Vertrauensverhältnis und sind eng miteinander verbunden. Da sich diese Personen stark vertrauen, ist bereits eine Person alleine sehr einflussreich auf deren enge Freunde. Weak Ties eignen sich zur Verbreitung von Informationen über grosse Strecken im Netzwerk. Die Weak Ties eröffnen die Möglichkeit Informationen aus verschiedenen Gruppen zu beziehen.

Die Zahl der Strong Ties ist sehr viel kleiner als der Weak Ties. Der Anthropologe Robert Dunbar hat herausgefunden, dass Menschen im Durchschnitt maximal 150 Weak Ties (Dunbar's Number) haben. Er definierte die Weak Tie Beziehung so, dass mindestens der Name der Person und die wesentlichen Beziehungen zu dieser Person bekannt sein müssen.¹

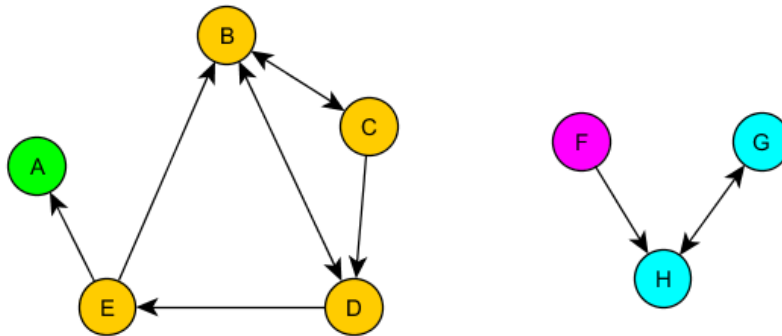
¹ Beschreibung von Dunbar's Number: https://en.wikipedia.org/wiki/Dunbar%27s_number, Aufgerufen am 6. Juli 2015

Connected Components

Als *Connected Component* wird eine Menge von Knoten verstanden, die über beliebige Pfade miteinander verbunden sind. Ist ein Teilnetzwerk abgeschnitten, bildet es einen zweiten *Component*.

Bei gerichteten Graphen wird zusätzlich unterschieden zwischen *Strongly Connected Components* und *Weakly Connected Components*. Wichtig ist: Bei beiden Connected Component Varianten befindet sich schlussendlich jeder Node genau in einem Component.

Ein *Strongly Connected Components* ist dadurch definiert, dass ich alle Knoten entlang der Kantenrichtung erreichen. Hier ein Beispiel:

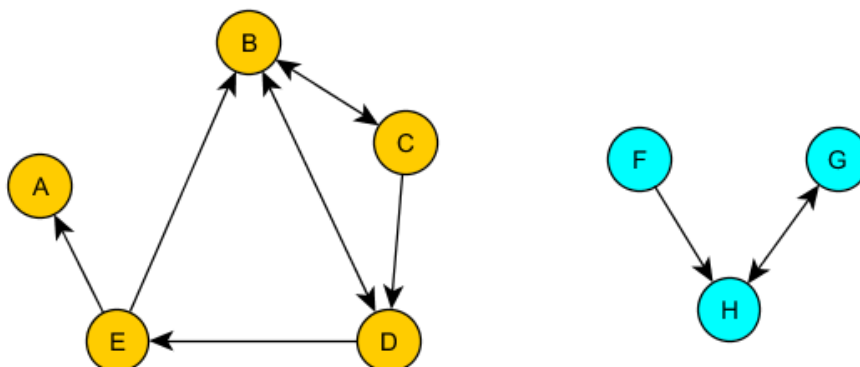


Darin befinden sich die folgenden *Strongly Connected Components*:

- B, C, D, E
- A
- G, H
- F

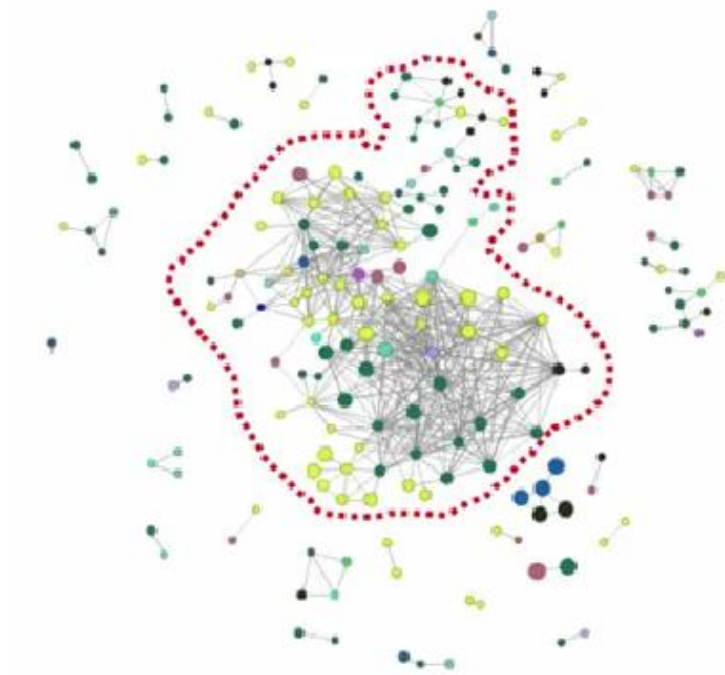
Bei *Weakly Connected Components* wird untersucht, welche Knoten einander erreichen, wobei die Kantenrichtung ignoriert wird und die Kanten einfach als bidirektional (in beide Richtungen zeigend) angesehen werden. Im folgenden Beispiel existieren die Weakly Connected Components:

- A, B, C, D, E
- F, G, H



Bei ungerichteten Graphen wird nur von *Connected Components* gesprochen.

Wenn ein Component einen signifikanten Teil der Knoten enthält wird dieser „**Giant Component**“ genannt. Folgendes Bild zeigt rot eingekreist einen solchen Giant Component:



Interpretation

Falls der Graph aus mehreren Components besteht kann dies dazu führen, dass Informationen nicht an alle Aktoren gelangen, der Informationsfluss ist also nicht mehr gewährleistet. Es zeigt wie stark die Aktoren miteinander verbunden sind. Ebenfalls identifiziert die Analyse, welche Personen im Netzwerk gut miteinander verbunden sind und welche nicht.

Ob in einem Netzwerk nach Strongly oder Weakly Connected Components gesucht wird ist vom Anwendungsfall abhängig (Ist es wirklich wichtig, dass die Kommunikation entlang der Kantenrichtung erfolgt?). Weakly Connected Components sind einfacher zu finden, da diese weniger restriktiv sind.

Befinden sich die Aktoren in verschiedenen Komponenten hat dies auch Einfluss auf gewisse Zentralitätsmasse, welche im Folgekapitel genauer betrachtet werden.

Aktor-Zentralität

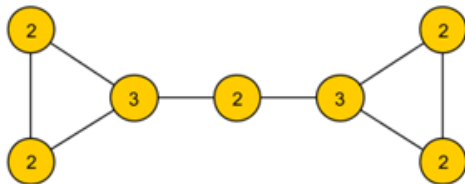
Zentralitätsmasse berechnen, wie wichtig ein einzelner Knoten im Netzwerk ist. Zentralität geht davon aus, dass ein Akteur „sichtbar“ ist. Durch seine Position ist ein zentraler Akteur informiert und hat auch Kontrollmöglichkeiten auf den Informationsfluss.

Es existieren verschiedene Zentralitätsmasse mit unterschiedlichen Bedeutungen. Das Verständnis der Berechnungen dieser Zentralitätsmasse ist wichtig, um die Werte richtig zu interpretieren.

Degree Centrality

Beschreibung:

Die *Degree Centrality* ist ein sehr einfaches Mass. Es zählt die Anzahl Kanten, die von einem Knoten weg gehen oder zu einem Knoten führen, also die direkten Verbindungen zu seinen Nachbarn. Die Summe dieser Kanten bildet dann den *Degree* des Knotens. Es gibt bei gerichteten Graphen auch die Möglichkeit den *Indegree* und den *Outdegree* separat zu berechnen, indem für die Knoten lediglich die eingehenden, resp. die ausgehenden Kanten gezählt werden. Hier ein Beispiel-Netzwerk, in dem die Knotenbezeichnung dem Degree entspricht:



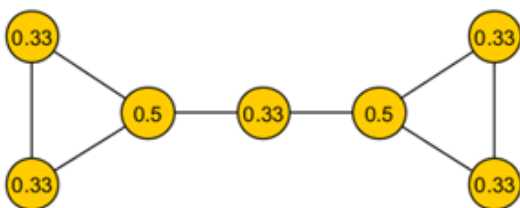
Normalisierung:

Die *Degree Centrality* kann als lokales Mass angesehen werden. Es sagt nichts über das komplette Netzwerk aus, sondern bietet lediglich eine isolierte Sicht auf einen Knoten. Mit der Normalisierung wird die Grösse des Netzwerks berücksichtigt, so dass Werte verschiedener Netzwerke miteinander vergleichbar sind. Die Normalisierung wird jedoch für *Degree Centrality* selten angewendet, da so oftmals sehr kleine Werte entstehen würden (grosse Netzwerke bestehen aus mehreren tausend Knoten).

Für die Normalisierung muss der Degree Centrality Wert des Knotens durch die maximale Anzahl möglichen Verbindungen dividiert werden. Dies sind:

- Bei ungerichteten Graphen: $n-1$
- Bei gerichteten Graphen: $2(n-1)$

So erhalten wir für jeden Knoten einen Wert zwischen 0 und 1. So ergeben sich folgende normalisierte Werte:



Formel

$$C_{D(v)} = \deg(v)$$

Die Degree Centrality für einen Knoten entspricht der Anzahl Verbindungen. Bei gerichteten Netzwerken können In- und Outdegree unterschieden werden.

Interpretation:

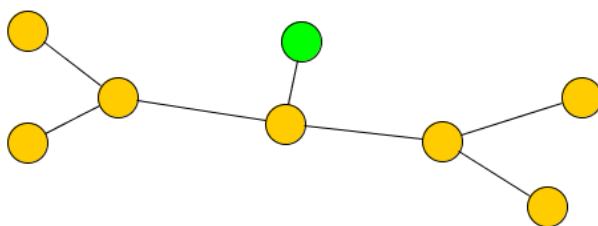
Ein Akteur mit einem hohen Degree-Wert geniesst eine hohe Wahrnehmung und Einfluss. Er ist im Netzwerk sehr prominent. Beispielsweise erreicht eine Person mit hohem Degree-Wert in Facebook mit einer Statusmeldung sehr viele Leute. Solche Personen eignen sich um Informationen zu verbreiten oder andere direkt zu beeinflussen, z.B bei Kaufsentscheidungen. Ein hoher Degree-Wert bedeutet meistens auch eine hohe (Kommunikations-)Aktivität. Hat ein Akteur sehr viele Indegrees gibt es auch viele Wege, über die er informiert werden kann. Somit ist dieser im Allgemeinen sehr schnell informiert.

Ein grosser In-Degree Wert ist auch ein Zeichen für Prestige, da eine solche Person von anderen „gewählt“ wurde und deshalb Einfluss auf diese geniesst. (Siehe Abschnitt Indegree (Popularity) auf der Seite 22)

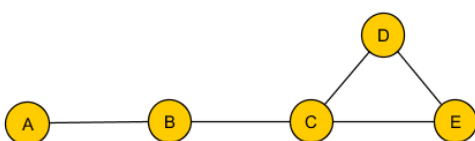
Closeness Centrality

Beschreibung:

Die vorangehend erläuterte Degree Centrality lässt sich leicht berechnen. Ein Schwachpunkt ist jedoch, dass es sich lediglich um ein lokales Mass handelt. Es betrachtet nur gerade die adjazenten Knoten und nicht das komplette Netzwerk. Möglicherweise ist es nicht wichtig sehr viele direkte Kontakte zu besitzen, sondern auch die indirekten Kontakte zu berücksichtigen und nicht zu weit vom Zentrum entfernt zu sein. Ein Beispiel wäre der grün eingefärbte Knoten in der folgenden Visualisierung, welcher nur gerade 1 Hop vom zentralsten Knoten entfernt ist, jedoch lediglich eine Degree Centrality von 1 besitzt.



Die *Closeness Centrality* setzt da an. Sie berechnet für jeden Knoten, wie effizient von diesem Knoten aus alle anderen Knoten erreichbar sind. Die Closeness Centrality für einen Knoten berechnet durch die Summe aller Inversen der kürzesten Distanzen zu allen anderen Knoten. (Siehe Formel). Die Berechnung der Closeness Centrality ist für grosse Graphen sehr zeitintensiv.



$$C_C(A) = 1 + 1/2 + 1/3 + 1/3 = \mathbf{13/6} \quad \Rightarrow \text{Normalisiert: } 1/4 * 13/6 = \mathbf{13 / 24 = 0.54}$$

$$C_C(B) = 1 + 1 + 1/2 + 1/2 = \mathbf{3} \quad \Rightarrow \text{Normalisiert: } 1/4 * 3 = \mathbf{3/4 = 0.75}$$

$$C_C(C) = 1 + 1 + 1 + 1/2 = \mathbf{7/2} \quad \Rightarrow \text{Normalisiert: } 1/4 * 7/2 = \mathbf{7 / 8 = 0.88}$$

$$C_C(D) = 1 + 1 + 1/2 + 1/3 = \mathbf{17/6} \Rightarrow \quad \text{Normalisiert: } 1/4 * 17/6 = \mathbf{17 / 24 = 0.71}$$

$$C_C(E) = 1 + 1 + 1/2 + 1/3 = \mathbf{17/6} \Rightarrow \quad \text{Normalisiert: } 1/4 * 17/6 = \mathbf{17 / 24 = 0.71}$$

Normalisierung:

Für die Normalisierung wird die berechnete Summe der inversen Distanzen dann noch durch die *Anzahl Knoten -1* geteilt.

Formel:

$$C_C(i) = \sum_{j=0}^N [d(i,j)]^{-1}$$

Formeln aus der Literatur berechnen oftmals zuerst die Summe und bilden dann die Inverse davon. Das Problem dabei ist, dass ein Graph mit verschiedenen Komponenten dann *Infinity* zur Gesamtsumme hinzunehmen würde (Knoten erreichen sich nicht), was zu 0 bei der Bildung der Inverse führen würde (wegen 1/Inf). Mit der angegebenen Formel wird für die Distanz zwischen verschiedenen Components lediglich 0 als Summand verwendet.

Für die Normalisierung wird die Closeness Centrality durch die *Anzahl Knoten – 1* geteilt.

$$\text{Normalisiert } C'_C(i) = \frac{C_C(i)}{N - 1}$$

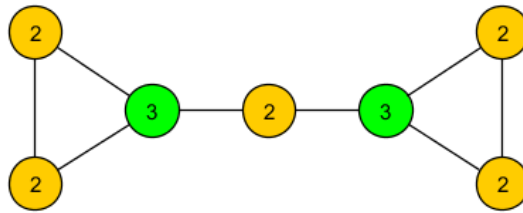
Interpretation:

Ein Akteur mit hoher Closeness Centrality kann alle anderen Knoten im Netzwerk effizient erreichen. Es kann als Mass für die sequenzielle Verbreitung von Informationen ausgehend von einem Knoten interpretiert werden, indem immer die kürzesten Pfade berücksichtigt werden.

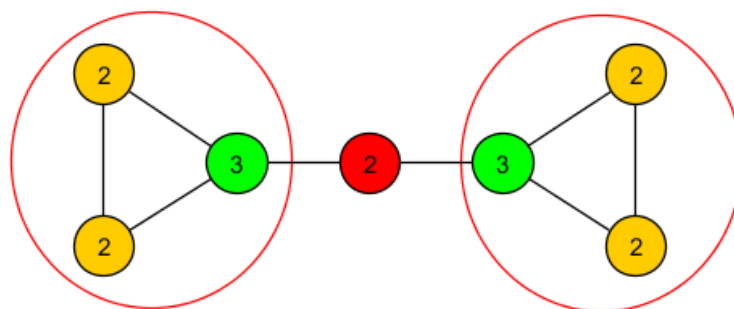
Betweenness Centrality

Beschreibung

Die *Betweenness Centrality* untersucht die Position innerhalb des ganzen Netzwerkes für alle Knoten. Nehmen wir dazu eines unserer Beispiel-Netzwerke mit eingetragenen *Degree Centralities*:



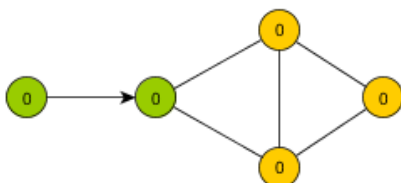
Aus der Sicht der *Degree Centrality* sind die grün eingefärbten Knoten die wichtigsten. Der Knoten zwischen den beiden grünen scheint mit diesem Mass weniger wichtig zu sein. Aus der Sicht des kompletten Netzwerkes ist er jedoch sehr wichtig, da er die beiden Teilnetzwerke miteinander verbindet (zusammen mit den beiden grünen Knoten). Immer wenn jemand vom linken Teilnetz mit dem rechten (und umgekehrt) spricht, geht die Kommunikation über den rot eingefärbten Knoten (sowie auch über die grünen):



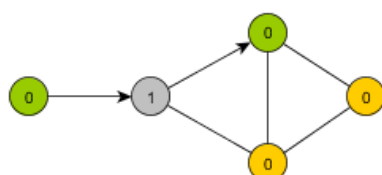
Es wird in diesem Fall auch von **Brokerage Position oder Gatekeeper** (Vermittler-Position) gesprochen, einem „dazwischen sein“. Die *Betweenness Centrality* berechnet für jeden Knoten, wie stark sich dieser in einer *Brokerage Position* befindet. Dazu wird von jedem Knoten der kürzeste Pfad zu allen anderen Knoten gesucht. Liegt ein Knoten auf vielen dieser kürzesten Pfade, so erhält er eine grössere *Betweenness Centrality*.

Der Algorithmus zur Berechnung der *Betweenness Centrality* funktioniert wie folgt (**Wichtig:** Es handelt sich um ein bidirektionales Netzwerk, die Pfeile veranschaulichen hier lediglich den kürzesten Pfad und bedeuten nicht „gerichtete Kante“):

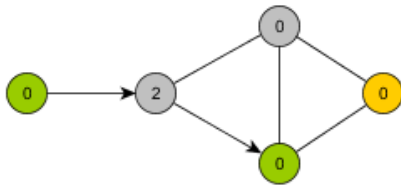
1. Jeder Knoten erhält zu Beginn den *Betweenness Centrality* Wert 0. Im Anschluss wird der kürzeste Pfad von allen Knoten zu allen anderen Knoten berechnet. Wir beginnen mit dem Knoten links. Der kürzeste Pfad zum folgenden Knoten ist direkt. Es hat also keine anderen Knoten dazwischen.



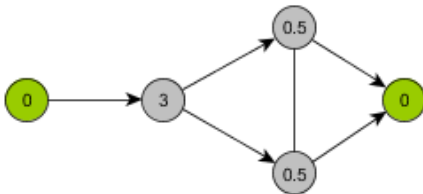
2. Der kürzeste Pfad zum nächsten Knoten geht über den vorherigen. Deshalb wird der *Betweenness Centrality* Wert dieses Knotens um 1 erhöht:



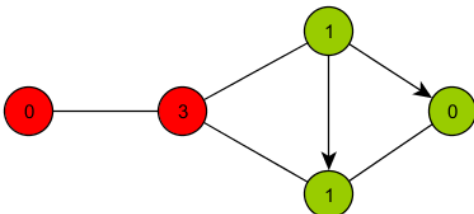
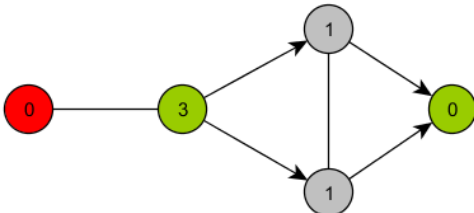
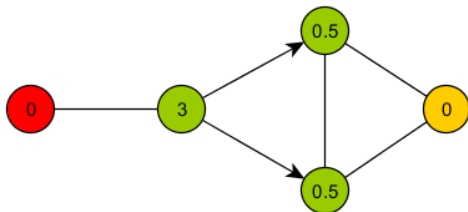
3. Beim unteren Knoten genau dasselbe:



4. Vom ersten zum letzten Knoten finden wir einen **Spezialfall**. Es gibt zwei kürzeste Pfade! In diesem Fall wird jedem Knoten auf dem Pfad jeweils 1 / #kürzeste Pfade addiert. In unserem Fall würden wir also allen Knoten 0.5 addieren auf beiden Wegen. Deshalb erhalten wir die folgenden Werte:



5. Die restlichen Schritte werden nach dem gleichen Schema ausgeführt. Weil es sich um ein bidirektionales Netzwerk handelt muss ein Knotenpaar lediglich einmal berücksichtigt werden.



Die Berechnung der Betweenness Centrality ist für grosse Graphen sehr zeitintensiv.

Normalisierung:

Da diese Werte sehr gross werden können macht es Sinn, sie zu normalisieren. Dazu werden die berechneten Betweenness-Werte durch die Anzahl möglicher Verbindungen zwischen allen anderen Knotenpaaren geteilt. Es muss unterschieden werden, ob es sich um gerichtete oder ungerichtete Graphen handelt. Die maximale Anzahl Verbindungen berechnet sich wie folgt (n = Anzahl Knoten):

- Gerichtete Graphen: $(n-1) * (n-2)$
- Ungerichtete Graphen: $(n-1) * (n-2) / 2$

Formel:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v)$$

Die Betweenness Centrality für einen Knoten setzt sich wie folgt zusammen: Es werden zwischen allen Knotenpaaren s, t die kürzesten Wege σ_{st} berechnet. Für jeden Knoten wird dann ermittelt, wie viele der kürzesten Pfade durch den entsprechenden Knoten verlaufen. Die Summe der kürzesten Pfade durch einen Knoten für alle Knotenpaare bildet dann die Betweenness Centrality.

Für die Normalisierung muss der berechnete Betweenness-Wert C_B dann noch durch die gesamte Anzahl kürzester Pfade geteilt werden:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Interpretation:

Ein Akteur mit hohem Betweenness Centrality spielt eine zentrale Rolle, weil ein grosser Teil der Kommunikation des Netzwerks über diesen Knoten geht. Er hat auch die Möglichkeit den Informationsfluss zu beeinflussen, indem er die Weiterleitung von Nachrichten verhindert oder fördert. Er kann die Nachrichten auch manipulieren oder Profit aus den Informationen schlagen. Akteure mit einem hohen Betweenness Wert stellen zudem einen „kritischen“ Punkt im Netzwerk dar, deren Ausfall die Kommunikation für das gesamte Netzwerk markant beeinflussen kann.

Eigenvektor Zentralität

Bei der Eigenvektor-Zentralität berechnet sich die Wichtigkeit eines Knotens anhand der Wichtigkeit der direkt benachbarten Knoten. Ein Knoten ist umso wichtiger, je wichtiger seine direkten Kontakte sind. Dieses Verfahren wird hier nicht im Detail erläutert, ist jedoch auf der Webseite <http://dijr-courses.wikidot.com/soc180:eigenvector-centrality> simpel und gut beschrieben.

Netzwerk-Zentralisierung

Die Masse des letzten Abschnitts über Zentralität berechnen alle, wie zentral ein Akteur anhand lokaler Gegebenheiten oder seiner Position innerhalb des Netzwerks ist. Die Netzwerk-Zentralisierung kombiniert alle Akteur-Werte zu einem einzelnen Wert. Linton Freeman hat dazu eine allgemeine Zentralisierungs-Formel entwickelt, welche die folgenden beiden Eigenschaften erfüllt:

- Es soll gezeigt werden, in welchem Masse der zentralste Akteur die Zentralität der anderen Akteure überschreitet
- Der berechnete Wert soll auf den maximal erreichbaren Wert (anhand der Netzwerkgrösse) bezogen sein.

Diese Formel liefert einen Wert zwischen 0 (alle Akteure sind gleich) und 1 (maximaler Unterschied), welcher zeigt, wie die Werte verteilt sind. Die Formel hat sich durchgesetzt und lässt sich auf alle vorgestellten Zentralitätsmasse anwenden. Die allgemeine Formel lautet:

$$\frac{\sum_{i=1}^n [C_X(p^*) - C_X(p_i)]}{\max \sum_{i=1}^n [C_X(p^*) - C_X(p_i)]}$$

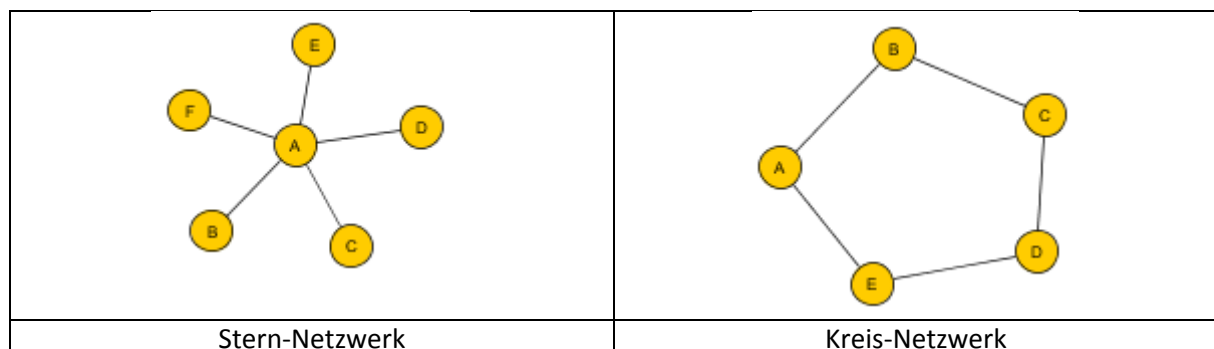
Wobei die einzelnen Funktionen und Variablen folgende Bedeutung zugewiesen bekommen:

- n : Anzahl Akteuren
- $C_X(p_i)$: Zentralitätsmessung für gegebenen Akteuren p_i
- $C_X(p^*)$: Grösster im Netzwerk vorhandener Zentralitätswert, berechnet mit dem Zentralitätsmass C_X
- $\max \sum_{i=1}^n [C_X(p^*) - C_X(p_i)]$: Theoretisch maximal mögliche Summe der Differenzen der Akteur-Zentralitäten für alle Akteure

In Worte gefasst bedeutet die Formel: Es wird im Zähler die Differenz zwischen dem höchsten Zentralitätswert zu allen Akteur-Zentralitätswerte aufsummiert. Der Nenner bekommt als Wert die Summe der theoretisch maximal möglichen Differenzen. So resultiert immer ein Wert zwischen 0 und 1, wobei 0 bedeutet, dass alle Knoten gleich zentral sind und 1, dass ein Akteur maximale Zentralität besitzt und alle anderen die tiefste Zentralität. Es besteht also die höchst mögliche Ungleichheit.

Hinweis: Im Zähler führt die Differenz vom Knoten mit dem grössten Zentralitätswert zu sich selbst zum Summand 0. Deshalb kann dieser ignoriert werden.

Als Beispiele zur Veranschaulichung werden die folgenden zwei Netzwerke verwendet, da beim Stern-Netzwerk der maximale Unterschied ersichtlich ist und beim Kreis-Netzwerk alle Knoten gleich zentral sind:



In den folgenden Abschnitten wird für jedes der betrachteten Zentralitätsmasse gezeigt, wie von den Akteur-Zentralitätsmassen das entsprechende Netzwerk-Zentralitätsmass berechnet werden kann.

Degree Zentralisierung

Die Degree-Zentralisierung, ausgehend von den **nicht-normalisierten Degree Centrality** Werten, berechnet sich in einem **ungerichteten Graphen** mit der folgenden Formel:

$$C_D = \frac{\sum_{i=1}^n [C_D(p^*) - C_D(p_i)]}{(n-1)(n-2)}$$

C_D steht hier für den nicht normalisierten Degree-Centrality-Wert des entsprechenden Akteurs

Der Nenner erklärt sich wie folgt: Das Stern-Netzwerk zeigt die höchst mögliche Ungleichheit. Dabei hat der zentrale Akteur (A im Bild) den Degree von (n-1). Alle anderen haben lediglich den Degree 1. Daraus resultiert eine Differenz zwischen dem zentralsten Akteur und einem aussenstehenden von: (n-1) - 1 = n-2. Diese Maximale Differenz befindet sich zwischen allen aussenstehenden Akteuren (n-1) und dem zentralsten. Deshalb ist die maximal mögliche Differenz im Nenner (n-1) (n-2).

Wird ein **gerichteter Graph** verwendet muss der Nenner noch mit dem Faktor 2 multipliziert werden.

Beim Stern-Netzwerk hat der Knoten A den Degree von 5, normalisiert also 1. Alle anderen Knoten (B-F) haben einen Degree von 1 und normalisiert von 0.2. Die Berechnung lautet also:

$$C_D = \frac{(5-1)+(5-1)+(5-1)+(5-1)+(5-1)}{(6-1)(6-2)} = \frac{20}{20} = 1$$

Beim Kreis-Netzwerk hat jeder Knoten den Degree 2. Somit ergibt sich folgende Berechnung:

$$C_D = \frac{(2-2)+(2-2)+(2-2)+(2-2)+(2-2)}{(5-1)(5-2)} = \frac{0}{12} = 0$$

Sind bereits normalisierte Werte für die Degree-Centrality verfügbar, fällt im Nenner der Term (n-1) weg. Übrig bleibt dann noch:

$$C_D = \frac{\sum_{i=1}^n [C'_D(p^*) - C'_D(p_i)]}{n-2}$$

Betweenness Zentralisierung

Für die Berechnung der Betweenness Centralization wird die folgende Formel verwendet, welche von **bereits normalisierten Akteur-Zentralitäten** ausgeht:

$$C_B = \frac{\sum_{i=1}^n [C'_B(p^*) - C'_B(p_i)]}{n-1}$$

C'_B steht hier für den normalisierten Betweenness-Centrality-Wert des entsprechenden Akteurs

Das Sternnetzwerk zeigt das maximal zentralisierte Netzwerk. Hier hat der mittlere Knoten einen normalisierten BC-Wert von 1, alle anderen einen Wert von 0. Dies zwischen einem aussen stehenden Knoten (z.B. B) und dem zentralsten Knoten A eine Differenz von 1. Und diese Differenz kann maximal für alle aussen stehenden Knoten auftreten, also (n-1) Mal. Somit ist der Nenner 1·(n-1).

Im Star-Netzwerk hat der Knoten A einen normalisierten Betweenness-Centrality Wert von 1, alle anderen Knoten (B-F) haben den Betweenness-Centrality Wert von 0. In die Formel eingesetzt resultiert dies zu:

$$C_B = \frac{(1 - 0) + (1 - 0) + (1 - 0) + (1 - 0) + (1 - 0)}{6 - 1} = \frac{5}{5} = 1$$

Beim Kreis-Netzwerk hat jeder Knoten einen nicht-normalisierten Betweenness-Centrality-Wert von 1 und somit einen normalisierten Betweenness-Centrality Wert von 0.166. Die Netzwerk Betweenness Centrality Wert lautet also:

$$C_b = \frac{(0.16 - 0.16) + (0.16 - 0.16) + (0.16 - 0.16) + (0.16 - 0.16) + (0.16 - 0.16)}{5 - 1} = \frac{0}{4} = 0$$

Closeness Zentralisierung

Um die Formel für die Closeness Centralization herzuleiten muss ein wenig ausgeholt werden. Für die Berechnung Closeness Centrality existieren verschiedene Formeln. Eine häufig gesehene ist die folgende:

$$C'_c(n_i) = \frac{n - 1}{(\sum_{j=i}^n d(n_i, n_j))}$$

Der grosse Nachteil an dieser Formel ist, dass für Graphen mit verschiedenen Komponenten der Closeness-Centrality Wert für alle Knoten 0 wird, da die Distanz zwischen zwei sich nicht erreichbaren Akteuren per Definition ∞ ist und somit der Nenner ebenfalls.

Leider referenzieren in der Literatur alle gefundenen Berechnungen der Closeness Centralization auf das oben aufgezeigte Closeness-Centrality Mass, welches minimal unterschiedliche Werte gegenüber der betrachteten Closeness Centrality Berechnungsweise auf der Seite 13 liefert.

Deshalb musste die Formel für die im Script beschriebene Berechnungsweise für Closeness Centralization hergeleitet werden, was hier auch ein wenig genauer ausgeführt wird:

Der Graph mit der maximal möglichen Zentralisierung ist wieder das Stern-Netzwerk. Dabei ergibt sich für die Closeness-Centrality der äusseren Aktoren (B - F) die allgemeine Formel:

$$C'_c(B) = \frac{\frac{1}{1} + (n - 2) \cdot \frac{1}{2}}{n - 1} = \frac{1 + (n - 2) \cdot \frac{1}{2}}{n - 1} = \frac{1 + \frac{n}{2} - 1}{n - 1} = \frac{\frac{n}{2}}{n - 1} = \frac{n}{2(n - 1)}$$

Der erste Summand bezieht sich auf die direkte Verbindung von B zu A. Der zweite Summand berechnet die Distanz zu allen anderen aussen stehenden Knoten. Diese haben eine Pfadlänge von 2 und von einem aussen stehenden zu allen anderen aussen stehenden zu kommen muss dieser Weg (n-2) Mal addiert werden. Das Ganze wird dann noch normalisiert, indem durch die *Anzahl Knoten - 1* geteilt wird.

Der zentralste Aktor A hat den Closeness-Wert 1, da er alle anderen Knoten direkt erreicht:

$$C'_c(A) = \frac{\frac{1}{1} (n - 1)}{n - 1} = 1$$

Nun muss die maximal mögliche Differenz zwischen dem zentralsten Knoten und einem aussenstehenden Knoten berechnet werden:

$$C'_c(A) - C'_c(B) = 1 - \frac{n}{2(n-1)} = \frac{2(n-1) - n}{2(n-1)} = \frac{2n - 2 - n}{2(n-1)} = \frac{n-2}{2(n-1)}$$

Diese maximale Differenz kann nun im kompletten Graphen genau (n-1) Mal auftreten. Somit berechnet sich die maximale Differenz für alle Closeness Centrality-Werte mit:

$$(n-1) \cdot \frac{(n-2)}{2(n-1)} = \frac{(n-1) \cdot (n-2)}{2(n-1)} = \frac{n-2}{2}$$

Für die Berechnung der Closeness Zentralisierung kann also die folgende Formel verwendet werden:

$$C_c = \frac{\sum_{i=1}^n [C'_c(p^*) - C'_c(p_i)]}{\left(\frac{n-2}{2}\right)}$$

C'_c steht hier für den **normalisierten Closeness-Centrality Wert** entsprechend dem beschriebenen Verfahren im Abschnitt Closeness Centrality auf der Seite 13.

Für das Stern-Netzwerk ergeben sich folgende, normalisierte Closeness-Centrality Werte:

$$C'_D(A) = \frac{5}{5} = 1$$

Und für alle aussen stehenden Aktoren B-F jeweils:

$$C'_D(B, C, D, E, F) = \frac{1 + 4 \cdot \frac{1}{2}}{5} = \frac{3}{5} = 0.6$$

Nun werden die entsprechenden Werte in die Formel eingefügt:

$$C_c = \frac{(1 - 0.6) + (1 - 0.6) + (1 - 0.6) + (1 - 0.6) + (1 - 0.6)}{\left(\frac{(6-2)}{2}\right)} = \frac{2}{\left(\frac{4}{2}\right)} = \frac{2}{2} = 1$$

Beim Kreis-Netzwerk hat jeder Akteur eine normalisierte Closeness Centrality von $\frac{3}{4} = 0.75$. Somit ergibt dies:

$$C_c = \frac{(0.75 - 0.75) + (0.75 - 0.75) + (0.75 - 0.75) + (0.75 - 0.75) + (0.75 - 0.75)}{\left(\frac{(5-2)}{2}\right)} = \frac{0}{1.5} = 0$$

Prestige

Wenn in einem Sozialen Netzwerk die Kanten gerichtet sind, dann kann auch das Ansehen (Prestige) der Personen untersucht werden. Voraussetzung dafür ist, dass die Interpretation der Kantenbedeutung dies zulässt. Zwei Möglichkeiten zur Berechnung eines Prestige-Wertes werden in den kommenden Unterabschnitten genauer beschrieben.

Indegree (Popularity)

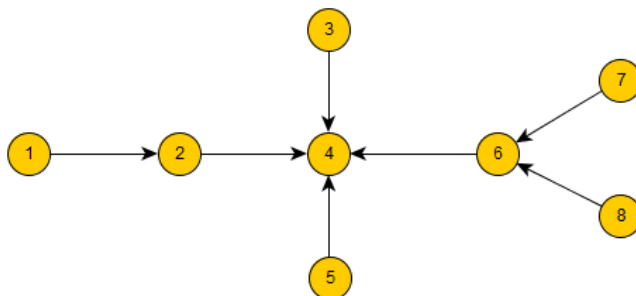
Das simpelste Prestige-Mass ist der Indegree eines Knotens, wo das Prestige-Mass sich einfach anhand der Summe aller eingehenden Kanten für einen Knoten bestimmen lässt. Der Nachteil dieses Masses ist, dass es nur gerade die direkten Nominationen berücksichtigt und dass in sehr dichten Netzwerken sehr viele Knoten einen hohen Indegree haben.

Proximity Prestige

Proximity Prestige Mass berücksichtigt nicht nur die direkten Indegrees, sondern alle Knoten, welche direkt und indirekt auf einen Knoten zeigen. Direkte und kurz entfernte Knoten haben dabei einen grösseren Einfluss als weit entfernte Knoten. Die Berechnung des Proximity Prestiges Wertes funktioniert wie folgt:

$PP(v)$ = Anteil der Knoten, welche direkt oder indirekt auf den Knoten v verweisen (ohne Knoten selbst) dividiert durch den Durchschnitt aller Pfadlängen der auf den Knoten v verweisenden Knoten.

Da die Definition sehr kompliziert klingt ein anschaulicheres Beispiel. Im folgenden Graphen haben die Pfeil-Richtungen die Bedeutung: „Hat um Rat gefragt“:



Im Netzwerk befinden sich insgesamt 8 Knoten.

Alle Knoten ohne eingehenden Kanten haben eine Proximity Prestige von 0. Das wären die Knoten: 1, 3, 5, 7, 8.

Der Knoten 2 hat genau einen Knoten (1), welcher direkt auf ihn verweist. Daraus folgt:

$$PP(2) = \frac{\frac{1}{8-1}}{\frac{1}{7}} = \frac{1}{7} = 0.14$$

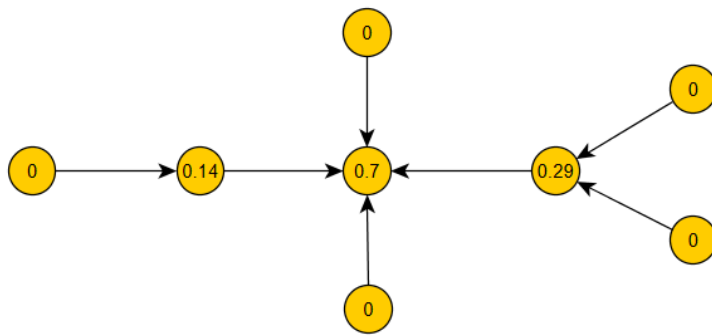
Der Knoten 6 hat einen Indegree von 2 Knoten, die direkt auf ihn verweisen:

$$PP(6) = \frac{\frac{2}{8-1}}{\frac{2}{7}} = \frac{2}{7} = 0.286$$

Interessant wird es beim Knoten 4. Alle anderen Knoten (7 Stück) verweisen direkt oder indirekt auf ihn:

$$PP(4) = \frac{\frac{7}{8-1}}{\frac{10}{7}} = \frac{1}{10} = 0.7$$

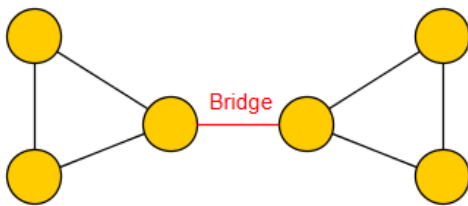
Derselbe Graph nochmals mit den Proximity Prestige-Massen als Labels:



Bridges und Brokers

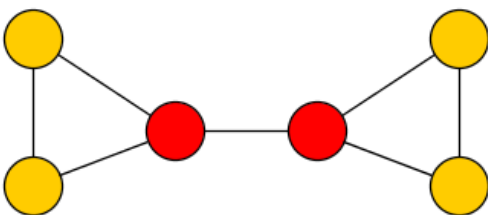
Mit der Betweenness Centrality wurde berechnet, in welchem Ausmass sich Knoten auf den kürzesten Kommunikationspfaden zwischen allen anderen Knoten befindet. Diese Position ist nicht zu unterschätzen. Ist ein Akteur mit anderen Akteuren verbunden, die jedoch untereinander nicht verbunden sind, übernimmt dieser eine Mediator-Funktion und kann daraus auch Profit schlagen. Die Lücke zwischen den Knoten, die nur über den Vermittler-Knoten miteinander kommunizieren, wird auch „**Structural Hole**“ genannt. Grundsätzlich geht es darum, dass in Netzwerken solche „Structural Holes“ vermieden werden, da die Präsenz von Structural Holes auch immer die Machtposition eines einzelnen Akteurs fördert.

Bridge: Von einer Bridge wird gesprochen, wenn die Entfernung einer Kante die Zahl der Komponenten im Netzwerk vergrössern würde. Folgendes Beispiel zeigt eine Bridge-Kante:



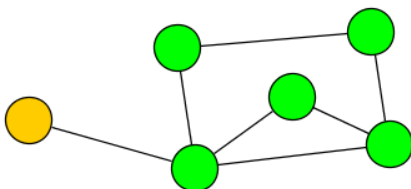
Dasselbe Prinzip der Bridge lässt sich auch auf Knoten anwenden.

Cut-Vertex: Erhöhen sich die Anzahl Komponenten im Falle einer Entfernung eines bestimmten Knotens, wird dieser Cut-Vertex genannt. Im folgenden Beispiel sind die Cut-Vertices rot eingefärbt:



Es ist unschwer zu erkennen, dass Akteure an der rot eingefärbten Position eine sehr einflussreiche Position innerhalb des Netzwerks aufweisen. Sie können die Kommunikation verweigern oder gar manipulieren, ohne dass dies von der gegenüberliegenden Seite erkannt wird. Deshalb ist es wichtig, solche Cut-Vertices zu vermeiden. Dazu wurde die Definition von Bi-Component eingeführt.

Bi-Component: Ein Component mit mindestens drei Knoten, welcher keinen Cut-Vertex besitzt. Im nachfolgenden Graphen bilden die Knoten des grünen Components einen Bi-Component:



Netzwerk-Metriken

Die betrachteten Zentralitätsmasse sind Werte für einen bestimmten Knoten. Die einzelnen Werte sagen jedoch wenig bis nichts über die gesamte Netzwerkstruktur aus, sondern beziehen sich mehr auf die Position einzelner Akteure innerhalb des Netzwerks. An diesem Punkt kommen die Netzwerk-Metriken ins Spiel. Netzwerk-Metriken liefern **Kennzahlen zur kompletten Netzwerkstruktur**, welche es auch ermöglichen, verschiedene Netzwerke miteinander zu vergleichen und diesbezügliche Schlussfolgerungen zu ziehen. Dabei steht die Kommunikation im Zentrum. Je indirekter die Kommunikation ist, desto grösser ist die Wahrscheinlichkeit und Gefahr für Manipulationen.

Graph Density

Die *Graph Density* (Dichte) beschreibt wie gut der Graph insgesamt verbunden ist. Der *Density*-Wert liegt immer im Bereich zwischen 0 und 1. Der Wert 0 steht für einen Graphen, in dem es gar keine Kanten gibt. Der *Density*-Wert 1 beschreibt einen Graphen, bei dem alle Knoten mit allen anderen Knoten verbunden sind. In der Graphentheorie wird dies auch als *Kompletter Graph* oder *Clique* bezeichnet.

Die Density für gerichtete Graphen berechnet sich aus mit folgender Formel, wobei $|E|$ für die Anzahl Kanten und $|V|$ für die Anzahl Knoten steht:

$$D_{\text{Gerichtet}} = \frac{|E|}{|V| (|V| - 1)}$$

In einem gerichteten Netzwerk gibt es $|V| (|V| - 1)$ mögliche Kanten (von jedem Knoten zu jedem anderen). Wenn die Anzahl vorhandener Kanten der Anzahl möglicher Kanten entspricht, dann resultiert daraus der Wert 1.

In einem ungerichteten Graphen gibt es zwischen einem Knotenpaar nur maximal eine Kante und nicht 2 (für den Weg hin und zurück). Deshalb muss in ungerichteten Kanten die Anzahl Kanten mit 2 multipliziert werden (oder der Nenner in der Formel halbiert):

$$D_{\text{Ungerichtet}} = \frac{2|E|}{|V| (|V| - 1)}$$

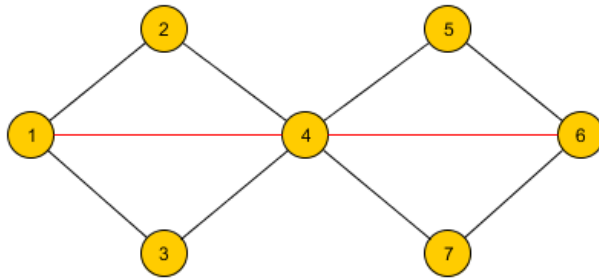
Die Dichte kann einen Hinweis geben, wie schnell sich Informationen in einem Netzwerk verbreiten. In einem dichten Netzwerk verläuft die Kommunikation sehr direkt. Mit steigender Anzahl der Knoten in einem grossen Netzwerk steigt auch die Anzahl der möglichen Kanten. Die Anzahl direkter Kontakte von Personen bleiben aber ungefähr konstant. Deshalb nimmt die Dichte mit der Grösse des Netzwerks ab. Dies ist auch ein Problem, wenn verschiedene Netzwerke verglichen werden.

Graph Diameter

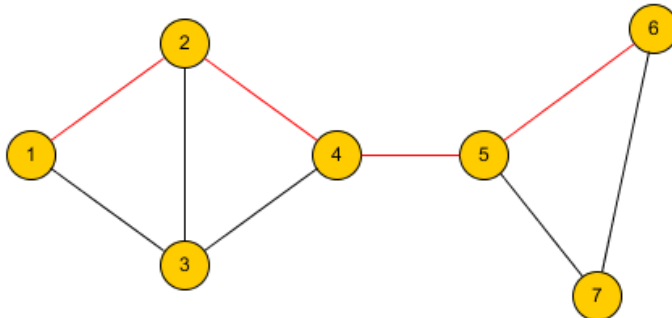
Beschreibung:

Als Graph Diameter (Durchmesser) wird der „längste kürzeste Pfad“ zwischen allen Knotenpaaren in einem Graphen. Falls es mehrere Components gibt, dann ist der Diameter laut Definition unendlich.

Im folgenden Beispiel ist der Diameter $d(1,6) = 2$. Dies ist aber nur einer von vielen „längsten kürzesten Pfade“. Genauso gut könnte $d(2,3)$, $d(2,7)$, $d(1,5)$ usw. als Referenz verwendet werden. Wichtig ist jedoch: Es gibt **keinen** längeren kürzesten Pfad!



Dieser Graph besitzt einen Diameter $d(1,6)$ resp. $d(1,7) = 4$.



Bei hohen Pfadlängen ist die Wahrscheinlichkeit grösser, dass die Nachricht unterwegs manipuliert wird. Deshalb sind kleinere Pfadlängen zu bevorzugen. Ein Vorteil von mehreren Pfaden zwischen zwei Knoten ist, dass Nachrichten mehrmals ankommen und Manipulationen so aufgedeckt werden können.

Die Berechnung des Durchmessers in gewichteten Graphen ist beschrieben unter:

http://www.gitta.info/Accessibiliti/de/html/StructPropNetw_learningObject2.html

Interpretation:

Dieses Mass gibt Auskunft darüber, welches die grösste Entfernung zwischen zwei Knoten ist und somit auch, wie viele Kanten eine Nachricht innerhalb des Netzwerks auf den kürzesten Pfaden maximal passieren muss, um jeden Akteur zu erreichen.

Cluster Coefficient

Der Cluster Coefficient ist ein Mass für Cliquenbildung in einem Graphen. Ein Wert von 1 bedeutet dabei, dass es sich um eine Clique handelt. Es wird zwischen dem lokalen und dem globalen Cluster Coefficient unterschieden, wobei es sich beim globalen Wert lediglich um den Mittelwert der lokalen Werte handelt.

Der lokale Cluster Coefficient berechnet sich aus dem Quotienten der Anzahl Kanten zwischen den Nachbarn eines Knoten (bei ungerichteten Netzwerken multipliziert mit dem Faktor 2) und der maximal möglichen Anzahl Kanten.

Für ungerichtete Graphen wird der lokale Clustering Coefficient C_i für einen Knoten wie folgt berechnet:

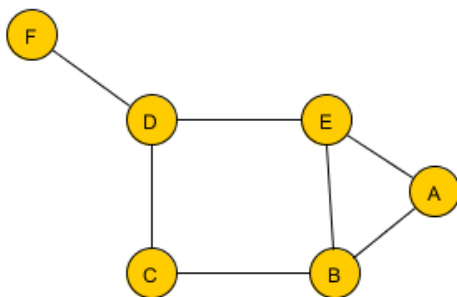
$$C_i = \frac{2n}{k_i(k_i - 1)}$$

Wobei n für die tatsächliche Anzahl Kanten zwischen den benachbarten Knoten steht und k_i für die Anzahl benachbarte Knoten. Bei einem gerichteten Graphen entfällt der Faktor 2 bei den tatsächlichen Kanten.

Der globale Clustering Coefficient C' berechnet sich dann wie folgt:

$$C' = \frac{1}{N} \sum_i^N C_i$$

Gegeben sie folgender Beispielgraph:



Der Knoten A besitzt zwei Nachbarn, B und E. Zwischen diesen existiert eine Kante. Dies führt zum Cluster Coefficient für A:

$$C_A = \frac{2 \cdot 1}{2 \cdot 1} = 1$$

Die Berechnung für die weiteren Knoten:

$$C_B = \frac{2 \cdot 1}{3 \cdot 2} = 1/3$$

$$C_C = \frac{0}{2 \cdot 1} = 0$$

$$C_D = \frac{0}{3 \cdot 2} = 0$$

$$C_E = \frac{2 \cdot 1}{3 \cdot 2} = 1/3$$

$$C_F = \frac{0}{1 \cdot 0} = 0.$$

Für den globalen Clustering Coefficienten C' ergibt sich dann der folgende Wert:

$$C' = \frac{1}{6} * \frac{5}{3} = \frac{5}{18}.$$

Akteure mit einem hohen Clustering Coefficienten sind stark lokal vernetzt und haben möglicherweise aufgrund längerer oder intensiverer Kontakte geschlossene Triaden gebildet (Der Freund meines Freundes ist mein Freund).

Reduktion des Netzwerks

Häufig sind die zu analysierenden Netzwerke sehr gross. Oftmals ist es deshalb erforderlich, das Netzwerk zu reduzieren, bevor es analysiert werden kann. Deshalb werden verschiedene Verfahren zur Reduktion des Netzwerks beschrieben. Diese Verfahren können natürlich auch beliebig kombiniert werden.

Local View

Eine einfache Möglichkeit das Netzwerk zu reduzieren ist das Filtern von Knoten anhand deren Attribute. So könnten die Knoten in einem Firmen-Netzwerk anhand deren Abteilung gefiltert und das Kommunikationsverhalten einzelner Abteilungen isoliert analysiert werden. Weitere Beispiele für das Filtern von Personen-Netzwerken wären Analysen anhand von Alter (nur jüngere Personen analysieren) oder Herkunft (Analyse von Personen aus verschiedenen Ländern).

Global View

Die Global View erlaubt eine allgemeine Sicht auf die Netzwerkstruktur. Dazu werden mehrere Knoten zu einem Knoten zusammengefasst; die detaillierte Sicht auf einzelne Knoten ist hier nicht von Interesse. Im Beispiel eines Unternehmens-Netzwerks, wo bei jeder Person das Attribut „Abteilung“ vorhanden ist, könnten die Knoten nach Abteilung gruppiert werden und das Kommunikationsverhalten zwischen den einzelnen Abteilungen analysiert werden.

Contextual View

Die Contextual View ist eine Kombination zwischen der Local und der Global View. Dazu werden die Knoten nach einem Kriterium gruppiert. Danach werden eine Gruppen in der Local View angezeigt und der Rest in der Global View. In unserem E-Mail Beispiel mit den Angestellten und Abteilungen wäre somit ersichtlich, wie die Personen in einer Abteilung mit allen anderen Abteilungen kommunizieren.

Ego Netzwerk

Ein Ego-Netzwerk beinhaltet ein spezifischer Knoten („Ego“), dessen Nachbarn und die Kanten zwischen all diesen Akteuren.

Filtern anhand von Katen-Attributen

Mindestens so wertvoll ist das Filtern von Kanten. Hier eignet sich das Gewicht der Kanten sehr gut. Wenn bei einem E-Mail Netzwerk das Kantengewicht für die Anzahl E-Mails steht, so können Kanten mit Gewicht unter einem definierten Schwellwert entfernt werden. Die Analyse der Verteilung der Kantengewichte ist sehr hilfreich, um einen solchen Schwellwert festzulegen.

Communities

Eine Community ist ein Subgraph innerhalb eines Graphen, welcher sehr stark und (ziemlich) direkt verbunden ist. Aufgrund dieser engen Beziehung haben diese Personen ein grosses Vertrauensverhältnis zueinander und dadurch auch einen starken Zusammenhalt. Communities zeichnen sich durch einen starken Gemeinschaftssinn aus. Sie haben ähnliche Einstellungen / Verhaltensweisen oder gemeinsame Interessen. Sie haben eine Tendenz zur gegenseitigen Angleichung und Konsensbildung. Communities bilden sich meistens um einen Kontext herum, z.B. Familie, Hobby, Berufskollegen usw. Dort gibt es auch immer Kernmitglieder und periphere Mitglieder, die eher weniger Beziehungen zu anderen aufweisen.

In Cliques gibt es redundante Informations-Lieferanten, die Gruppe kann sich sehr schnell austauschen dank der starken Verbundenheit. Sie ist auch gegen aussen ein wenig abgeschottet und das Verlassen einer Clique kann auch der Verlust vieler Beziehungen zu Cliques-Mitglieder zur Folge haben.

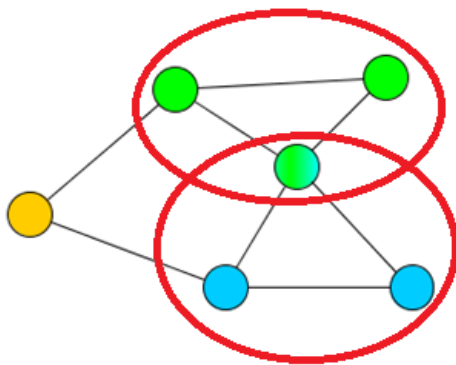
Mitglieder einer Clique entnehmen die Informationen aus dem gleichen Pool und sind gegenüber Innovationen und Entwicklungen von aussen eher abgekoppelt.

Es gibt verschiedene Möglichkeiten, wie eine Community identifiziert werden kann. Hier ein paar davon:

- Alle kennen einander (=Clique, können sich auch überlappen, so dass ein Knoten in mehreren Cliquen vorkommen kann). Hier steht die direkte Beziehung zu anderen Mitglieder im Vordergrund.
- Alle Knoten in einer Gruppe haben mindestens k Links zu anderen Knoten der Gruppe (k -core). Hier besteht nur ein gewisser Grad an Verbundenheit.
- Individuen erreichen einander über maximal n Hops (n -Clique). Hier wird die Nähe der Cliquen-Mitglieder in den Vordergrund gestellt.

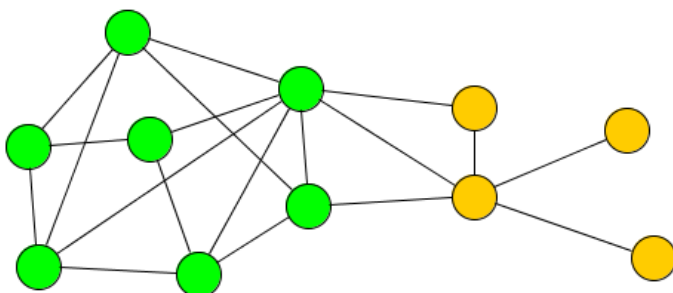
Cliquen kommen meistens nur im Umfang von kleineren Knotenmengen (ca. 3-4 Personen) vor, da die Cliquen-Bedingung sehr restriktiv ist. Aufgrund der Struktur sind auch die Zentralitätsmasse nicht sehr interessant, sehr wohl jedoch Überlappungen von Cliquen, da diejenigen Personen Informationen von beiden Gruppen erhalten.

Hier ein Beispiel eines Graphen mit zwei Cliquen, grün und blau eingefärbt. Der mittlere Knoten befindet sich in beiden Cliquen:



k-Core: k -Core hat eine weniger einschränkende Bedingung als Clique. Hier muss jeder Knoten mindestens k andere Knoten des Clusters verbunden sein.

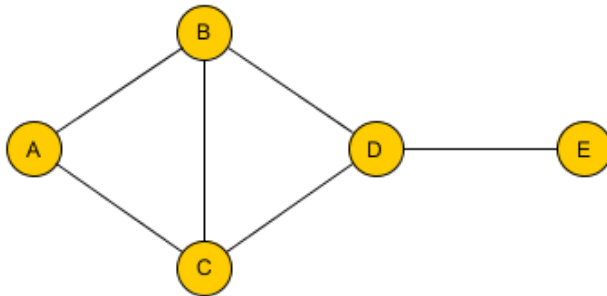
Im folgenden Beispiel bilden die grün eingefärbten Knoten einen 3-Core:



Doch auch diese Definition ist einschränkend. Angenommen es gibt Knoten, welche weniger als k Kanten haben, aber nur zu Individuen einer Community verbunden sind, dann wären dieser nicht Teil des k -Cores. (Oder der komplette Cluster wird auf einen geringeren k -Core gestuft als eigentlich nötig)

n -cliques: Alle innerhalb einer n -Clique erreichen einander mit maximal n Hops. Eine **1-clique** wäre somit eine Clique wie sie vorher dargestellt wurde, wo jeder jeden direkt erreichen kann. Im folgenden Graphen befinden sich drei 1-Cliques, wobei die Knoten B,C und D zu zwei Cliques gehören:

- A, B, C
- B, C, D
- D, E



Der Graph enthält folgende 2-Cliques:

- A, B, C, D
- B, C, D, E

Bei n -Cliques können verschiedene Eigenheiten auftreten, welche alle dieselbe Ursache haben. Bei n -Cliques kann die definierte Hop-Distanz n über Knoten gehen kann, welche schlussendlich nicht im Cluster erscheinen. Hier zwei Beispiele:

<p>Der Durchmesser (Graph Diameter) der n-Clique kann grösser sein als das n. Grün ist die 2-Clique eingefärbt, deren beiden untersten Punkte sich aber nur über ein en Knoten ausserhalb des Clusters kennen:</p>	<p>Es kann passieren, dass die Knoten in der n-Clique nicht miteinander verbunden sind. Z.B können die roten und grünen Knoten als 2-Cliques interpretiert werden, deren Knoten aber nicht direkt miteinander verbunden sind:</p>

p -cliques: Bei p -Cliques muss mindestens ein Bruchteil p (Angabe zwischen 0 und 1) aller Kanten eines Knotens zu anderen Knoten führen, welche sich im Cluster befinden. Somit werden viele der oben erwähnten Nachteile beseitigt.

Clustering – Auffinden von Communities

Beim Clustering geht es darum in einem Graphen Community Strukturen zu entdecken. Dazu gibt es verschiedene Vorgehensweisen. Hier werden zwei vorgestellt: Zum einen ein Top-Down Clustering und zum anderen ein Bottom-Up Clustering.

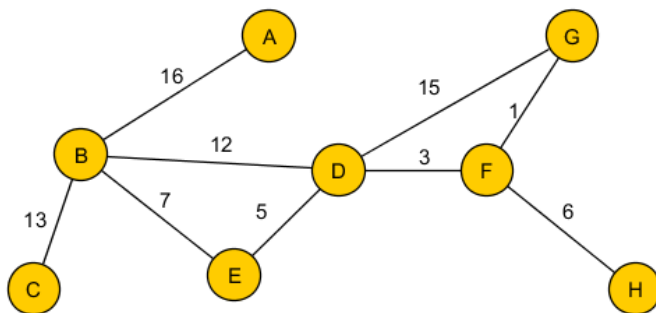
Hierarchical Clustering

Das Hierarchical Clustering gehört zu den Bottom-Up Clustering Verfahren. Dies bedeutet, dass zu Beginn jeder Knoten einen eigenen Cluster bildet und diese Schritt für Schritt zusammengefasst werden. Dabei werden in jedem Durchlauf die Ähnlichkeit zwischen allen Clustern berechnet und jeweils die zwei einander ähnlichsten Clustern zusammengefasst. Dazu muss natürlich die Ähnlichkeit zwischen zwei Clustern definiert werden. Hier zwei Beispiele:

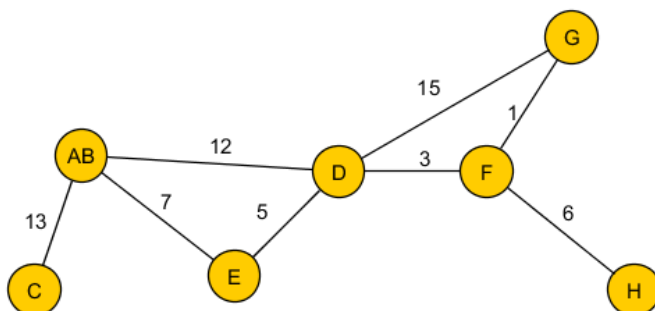
- Anzahl der gemeinsamen Freunde
- Häufigkeit der direkten Kommunikation

Dieser Vorgang wird durchgeführt bis entweder die gewünschte Anzahl Clusters erreicht worden ist oder sich alle Knoten im selben Cluster befinden.

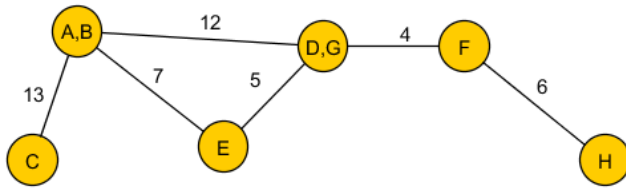
Hier ein Beispiel: Als Ähnlichkeitsmass wird die Anzahl kommunizierter Nachrichten (Kanten-Gewicht) zwischen zwei Clustern verwendet. Die Ausgangslage ist folgender Graph und die alle Knoten sollen in zwei Clustern unterteilt werden.



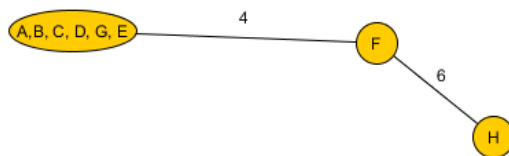
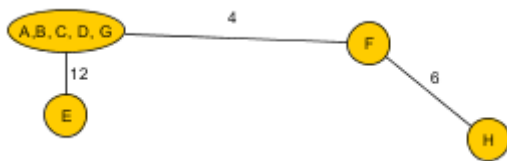
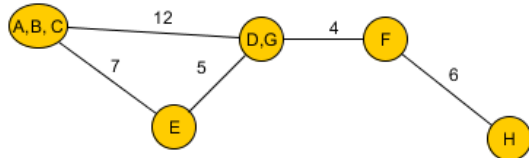
Zu Beginn bildet jeder Knoten seinen eigenen Cluster, es existieren also 8 Clusters. Jetzt werden diejenigen Knoten zusammengeführt, welche am meisten kommunizieren. Das sind die Knoten A und B:



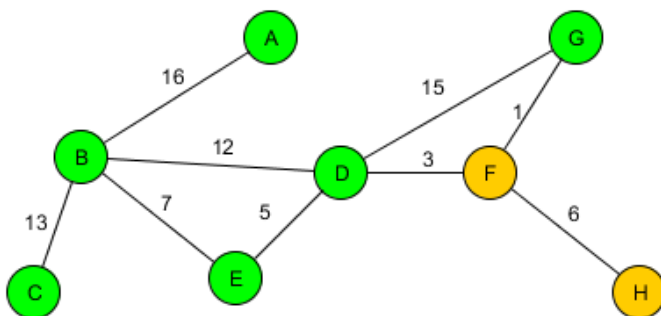
Nun werden die Knoten D und G zusammengefasst. Da diese nun einen neuen Cluster bilden muss die Kommunikation zwischen dem neuen Cluster und aussenstehenden Knoten aktualisiert werden. Beide Knoten kommunizieren mit dem Knoten F. Deshalb muss beim neuen Cluster für die Kommunikation mit dem Knoten F die Summe zwischen den Knotenpaaren G und F (1) sowie D und F (3) verwendet werden:



Dieses Verfahren wird nun Schritt für Schritt fortgesetzt bis schlussendlich zwei Clusters entstehen:



Nun sind die zwei gesuchten Clusters gefunden. Hier noch die Visualisierung innerhalb des Ursprungs-Netzwerk:

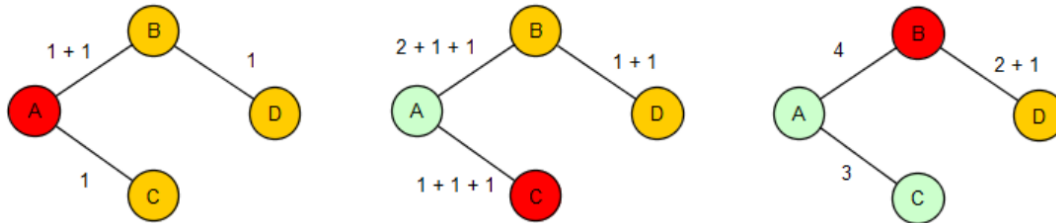


Edge-Betweenness Clustering

Das Edge-Betweenness-Clustering, auch bekannt unter dem Girvan-Neuman Algorithmus, ist ein Top-Down Clustering Verfahren. Dies bedeutet, dass sich zuerst alle Knoten im selben Cluster befinden und dann sukzessive aufgeteilt werden.

Wie der Name bereits vermuten lässt wird als Ähnlichkeitsmass der Edge-Betweenness Centrality Wert berechnet. Die Edge-Betweenness-Centrality Berechnung funktioniert gleich wie bei die Node Betweenness Centrality, nur jetzt halt für Kanten.

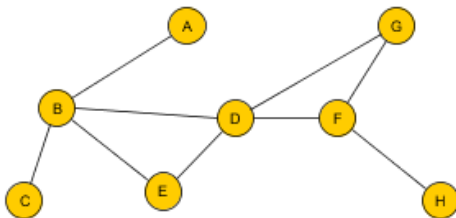
Folgendes kleines Beispiel veranschaulicht die Berechnung der Edge Betweenness: Es wird mit dem Knoten A begonnen (Bild 1) und die kürzesten Pfade zu allen anderen Knoten gesucht. Nun wird bei jeder Kante gezählt, wie viele der kürzesten Pfade über die entsprechende Kante verlaufen. Wie bei der Node Betweenness-Centrality wird dies für jeden Knoten ausgeführt (Bild 2 und 3). Schlussendlich hat die Kante zwischen den Knoten A und B den grössten Edge-Betweenness Wert. Diese Kante liegt am häufigsten auf den kürzesten Kommunikationspfaden zwischen allen Knotenpaaren.



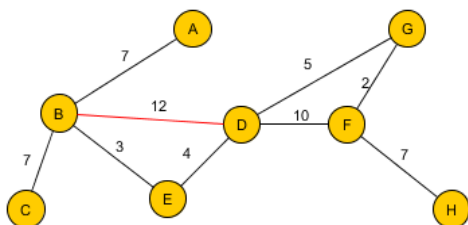
Ein hoher Edge-Betweenness Wert bedeutet, dass es sich um eine Kante zwischen zwei Knoten-Gruppen handelt. Bei diesem Clustering-Verfahren gibt es mehrere Iterationen, wobei in jeder Iteration diejenige(n) Kante(n) mit dem höchsten Betweenness-Wert entfernt wird, bis schlussendlich die gewünschte Anzahl Clusters erreicht worden ist. Dieses Verfahren ist sehr rechenintensiv, da nach jeder Iteration die Edge-Betweenness Werte neu berechnet werden müssen, der komplette Algorithmus liegt in $O(n^3)$. Deshalb wird es selten eingesetzt.

Hier ein weiteres ausführliches Beispiel für das Betweenness Clustering. Die Kante(n) mit dem höchsten Betweenness-Wert sind jeweils rot eingefärbt und werden nacheinander entfernt:

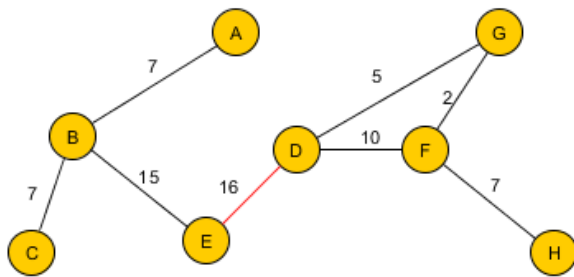
Ausgangslage:



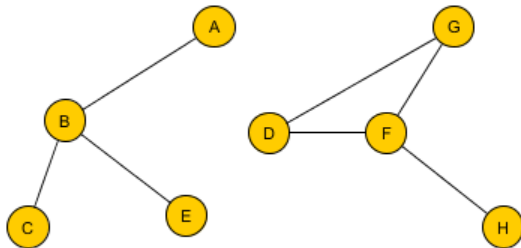
Berechnete Edge-Betweenness Centralities (nicht normalisiert):



Nachdem die Kante mit dem höchsten Betweenness-Wert entfernt wurde, werden die Kanten-Betweenness Werte neu berechnet:



Nachdem wieder die Kante mit dem höchsten Betweenness-Wert entfernt wurde erhalten wir den in zwei Clusters aufgeteilte Graphen:



Small World

Experimente

Ursprüngliches Paket-Experiment von Milgram

Im Jahre 1967 hat Stanley Milgram folgendes Experiment durchgeführt: 60 Teilnehmer der USA mussten ein Paket an eine festgelegte Person in Boston, die sozial und geografisch weit von der Ursprungs-Person entfernt war, zusenden. Sie durften es aber nur der Person direkt zustellen, falls Sie diese auch persönlich kennen und mit Vornamen ansprechen. Ansonsten mussten sie es einer bekannten Person mit gleichem Kriterium weitersenden, bei der die Wahrscheinlichkeit hoch war, dass sie die Zielperson kennt. Der Weg des Paketes wurde protokolliert. So wurde untersucht, wie viele Personen zwischen Absender und Empfänger lagen (Anzahl Hops)

Die durchschnittliche Pfadlänge lag bei 5.5. Daraus schliesst sich, dass im Durchschnitt innerhalb der USA jede Person jede andere über 6 Personen erreichen kann. Es entstand auch der Ausdruck *Six Degrees of Separation* für dieses Phänomen. Dieser Ausdruck wurde jedoch nie von Milgram verwendet.

Co-Authorship Experiment von Paul Erdős

Ein weiteres Small World Experiment wurde vom Mathematiker Paul Erdős durchgeführt. In einem Graphen stellt er alle Autoren von Publikationen als Knoten dar. Kanten zwischen zwei Knoten wurden erzeugt, wenn diese gemeinsam eine Publikation verfasst haben (Co-Autor). Paul Erdős gab sich selbst die Erdős-Zahl 0. Personen, mit denen er publiziert hatte, erhielten die Zahl 1. Autoren, welche mit Co-Autoren von Paul Erdős eine Publikation verfasst haben die Zahl 2 usw.



Autoren, welche nicht erreicht werden konnten, erhielten die Zahl „unendlich“. Es zeigte sich, dass die Zahl entweder unendlich oder sehr klein war. Bei 268'000 Personen konnte ein endlicher Wert ermittelt werden, der Durchschnitt bei 4.65 lag. (Erdős hat in sehr vielen Teilen der Mathematik publiziert).

MSN Studie von Microsoft

Microsoft analysierte 90 Millionen täglich aktive Messenger-Accounts. Jeder Account wurde als Knoten und die Kommunikation zwischen zwei Personen als Kanten dargestellt. Die Analyse ergab schlussendlich, dass zwei beliebige Personen durchschnittlich 6.6 Schritte voneinander getrennt waren. Es gab auch Pfade bis zu einer Länge von 29 Schritten. Damit wurde die Theorie der Small World anhand eines riesigen, globalen Netzwerks bestätigt, auch wenn die beiden Forscher Eric Horvitz und Jure Leskovec eher „Seven Degrees of Separation“ als Mass vorschlagen würden.

Eigenschaften von Small World Netzwerken

In einem Small World Network sind die meisten Personen sind über wenige Hops miteinander verbunden. Die Netzwerke neigen dazu geschlossene Triaden und Cliques (oder zumindest fast-Cliques) zu bilden, weil ein Bekanntenkreis gut miteinander verbunden ist. Ein weiteres Merkmal von Small-World Netzwerken sind Hub-Nodes. Dies sind Knoten, welche sehr viele Verbindungen haben. Die Distanz zwischen zwei zufällig gewählten Knoten entspricht etwa dem Logarithmus der Anzahl Knoten. Duncan Watts and Steven Strogatz haben entdeckt, dass Graphen anhand zwei unabhängigen Metriken klassifiziert werden können:



- Clustering Coefficient
- Average Shortest Path Length

Sogenannte Random Graphs, bei denen die Kanten zwischen den Knoten rein zufällig erzeugt werden, haben einen kleinen *Average Shortest Path Length* und zugleich einen kleinen *Clustering Coefficient*. Small-World Graphen haben einen kleinen Average Shortest Path, jedoch einen deutlich höheren Clustering Coefficient als der Random Graph.

Um zu erkennen, ob es sich um ein Small-World Graphen handelt, werden die beiden Masse mit einem Random-Netzwerk mit ungefähr gleicher Degree-Verteilung verglichen. Für ein Small World Netzwerk sind dann die folgenden beiden Eigenschaften erfüllt:

- $L_{SW} \leq L_{rand}$ (Average Shortest Path Length)
- $C_{SW} \gg C_{rand}$ (Global Clustering Coefficient)

Hier ein Beispiel aus Wikipedia, wo ein Random Graph mit einem Small World Graphen verglichen wird. Es ist ein deutlicher Unterschied beim Clustering Coefficient zu erkennen.

Random Graph	Small World Graph
	
Average vertex degree = 1,417 Average shortest path length = 2.109. Clusterization coefficient = 0.167	Average vertex degree = 1,917 Average shortest path length = 1.803. Clusterization coefficient = 0.522

Referenzen

Adamic, L. (kein Datum). *Corusera - SNA (Webseminar)*. Von <https://www.coursera.org/course/sna> abgerufen

de Nooy, W., Mrvar, A., & Batagelj, V. (2005). *Exploratory Social Network Analysis with Pajek*.

Everton, S. F. (kein Datum). *Tracking, Destabilizing and Disrupting Dark Networks WITH Social Network Analysis*.

Golbeck, J. (2013). *Analyzing the Social Web*.

Jansen, D. (2006). *Einführung in die Netzwerkanalyse, 3. überarbeitete Auflage*. VS Verlag für Sozialwissenschaften.

Scott, J. (2012). *Social Network Analysis*.

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*.

Wikipedia. (1. 10 2014). *Centrality*. Von Centrality: <http://en.wikipedia.org/wiki/Centrality> abgerufen

Wikipedia. (1. 10 2014). *Girvan-Newman Algorithm*. Von http://en.wikipedia.org/wiki/Girvan-Newman_algorithm abgerufen