Wintersemester 2015/2016

Deskriptive Analyse von Netzwerken

Bachelorarbeit

Verfasser: Elisabeth Krätzschmar
Matrikelnummer
Betreuer: Prof. Dr. Göran Kauermann
Prüfer: Prof. Dr. Göran Kauermann
Institut für Statistik
Ludwig–Maximilians–Universität München
Abgabedatum: 29. März 2016

Inhaltsverzeichnis

Nomenklatur

$G = (V, E)$	Graph mit Knotenmenge V und Kantenmenge E
N_v	Anzahl der Knoten
N_e	Anzahl der Kanten
$\mathbf{A}$	Adjazenzmatrix
$deg(v)$	Grad des Knotens v
$deg_{in}(v)$	Eingangsgrad des Knotens v
$deg_{out}(v)$	Ausgangsgrad des Knotens v
$\{deg(v)\}_{v \in V}$	Gradfolge
$dist(u, v)$	geodätische Distanz zwischen Knoten u und v
$c_{cl}(v)$	Nähezentralität von Knoten v
$c_B(v)$	Intermediationszentralität von Knoten v
$c_{Ei}(v)$	Eigenwertzentralität von Knoten v
$den(G)$	Dichte des Graphen G
$\tau_3(G)$	Anzahl der 2-Stars in G
$\tau_\Delta(G)$	Anzahl der Triangles in G
$cl(G)$	Clusterkoeffizient von G
$cl_T(G)$	Cluster-Transitivitätskoeffizient
$\mathcal{C} = \{C_1, \ldots, C_K\}$	Partition
$mod(\mathcal{C})$	Modularität der Partition
$\mathbf{L}$	Laplacematrix
λ	Eigenwert
$\mathbf{x}$	Eigenvektor
$\mathbf{B}$	Modularitätsmatrix

Abstract

Diese Bachelorarbeit beschäftigt sich mit den deskriptiven Analysemöglichkeiten von Netzwerken und wendet sie auf einen Patentdatensatz des Max-Planck-Instituts für Innovation und Wettbewerb an.

Deskriptive Analysemethoden solcher Daten umfassen zum einen die Untersuchung von Eigenschaften der Netzwerkbausteine. Hierbei stehen insbesondere Maße für die verschiedenen Konzepte zur Messung des Einflusses einzelner Knoten im Fokus. Zum anderen werden Methoden zur Messung der Netzwerkkohäsion, also dem Grad der Vernetzung der einzelnen Akteure im Netzwerk, und geeignete Mittel zum Identifizieren von Gruppenstrukturen vorgestellt.

1 Einleitung

In Zeiten boomender sozialer Netzwerke besteht ein starkes Interesse daran, die Struktur solcher Netzwerke zu untersuchen und Rückschlüsse daraus zu ziehen. Derartige Netzwerkstrukturen kann man auch in zahlreichen technologischen und biologischen Konzepten finden, insbesondere auch bei der Erforschung von auf neuronalen Netzen beruhender künstlicher Intelligenz.

Thema dieser Arbeit ist die deskriptive Analyse von Netzwerken. Mit der wachsenden Untersuchung von Netzwerkstrukturen sowohl im Alltag als auch in der Forschung gewinnt die statistische Analyse der hierbei anfallenden Daten zunehmend an Bedeutung. Aufgrund der besonderen Struktur dieser Daten sind hierzu spezielle Werkzeuge und Methoden erforderlich. Dazu werden im Kapitel 2 zunächst die Grundbegriffe und Konzepte der Netzwerkanalyse vorgestellt, die aus der mathematischen Graphentheorie stammen. Anschließend wird in den Kapiteln 3 und 4 auf Methoden zur Beschreibung von Netzwerken eingegangen. Dabei werden in Kapitel 3 zunächst Maße besprochen, die auf dem Grad der Vernetzung der Akteure im Netzwerk basieren, und anschließend werden verschiedene Konzepte zur Messung der Wichtigkeit einzelner Bestandteile des Graphen vorgestellt. Kapitel 4 konzentriert sich auf den Zusammenhang des Netzwerks. Hierzu wird die lokale und globale Struktur des Netzwerkgraphen genauer untersucht und im Anschluss werden Möglichkeiten der sinnvollen Unterteilung des Netzwerks in einzelne Gruppen vorgestellt.

1.1 Patentdatensatz

Ein Großteil der vorgestellten deskriptiven Analysemethoden in der vorliegenden Arbeit wird auf ein Netzwerk von Patententwicklern angewandt. Es umfasst 10208 Entwickler, die Patente angemeldet haben. Wenn zwei Entwickler bei einem Patent zusammengearbeitet haben, besteht eine Verbindung zwischen diesen beiden Entwicklern.

Die Datenanalyse wurde mit **R** basierend auf den Methoden, die in Kolaczyk and Csàrdi (2014) vorgestellt werden, durchgeführt. Dabei wurde in erster Linie die Netzwerkanalyse-Pakete *igraph*, *sand* und *ergm* benutzt.

2 Einführung in die Netzwerkanalyse

Zunächst werden im folgenden Kapitel einige Grundbegriffe aus der Graphentheorie eingeführt und häufig vorkommende Strukturen in Graphen vorgestellt. Im Anschluss wird auf Möglichkeiten, Netzwerkgraphen in komprimierter Form darzustellen, und Rechenaspekte eingegangen.

2.1 Grundbegriffe

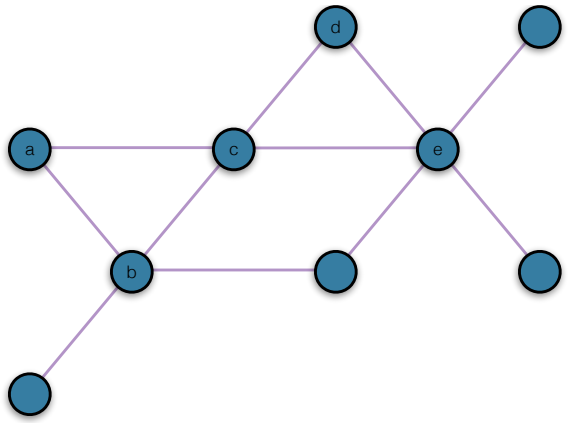

Abbildung 1: Beispiel eines Netzwerkgraphen

Die Struktur von Netzwerken lässt sich mathematisch durch einen Graphen modellieren, wie in Abbildung 1 dargestellt. Ein **Graph** $G = (V, E)$ ist eine mathematische Struktur, die aus einer **Menge von Knoten** V (*Vertex*) und einer **Menge von Kanten** E (*Edge*) besteht. Die Kantenmenge E selbst besteht wiederum aus Knotenpaaren $\{u, v\}$, wobei $u, v \in V$, $u \neq v$ gilt, und beschreibt, wie die einzelnen Knoten miteinander verbunden sind. Bei einem einfachen, ungerichteten Graphen ist die Menge der Kanten eine Teilmenge aller zweielementigen Teilmengen von V. Die Anzahl der Knoten $N_v = |V|$ wird die **Ordnung** und die Anzahl der Kanten $N_e = |E|$ die **Größe** eines Graphen G genannt.

Betrachtet man nur einen Teil eines Graphen, so spricht man von einem **Subgraphen** $H = (V_H, E_H)$ von $G = (V, E)$, wenn man eine Untermenge der Knotenmenge $V_H \subset V$ und Kanten aus E betrachtet, deren Knoten in V_H liegen. Von einem **induzierten Subgraphen** $G' = (V', E')$ spricht man, wenn zu einer vorgegebenen Knoten-Untermenge $V' \subset V$ alle Kanten $E' \subset E$, deren zugehörige Knoten in V' liegen, betrachtet werden.

Wählt man aus dem Netzwerk in Abbildung 1 die Knoten $V' = \{a, b, c, d, e\}$ aus, so ist der dazugehörige induzierte Subgraph in Abbildung 2 abgebildet.

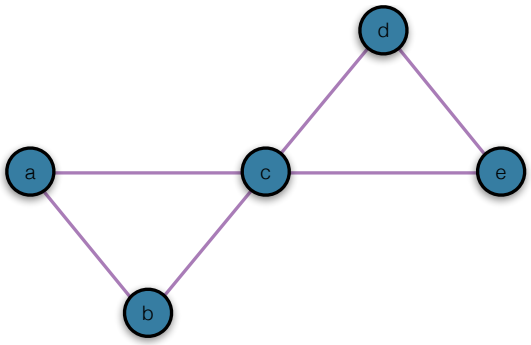

Abbildung 2: Durch $V' = \{a, b, c, d, e\}$ induzierter Subgraph von Abbildung 1

Multigraphen sind eine Erweiterung der einfachen Graphen, die Loops und Multi-Edges zulassen. **Loop** bezeichnet hierbei eine Kante, bei der Anfangs- und Endknoten identisch sind. **Multi-Edges** bezeichnet den Fall, dass zwischen zwei Knoten mehr als eine Kante existiert. Solche Multigraphen können beispielsweise benutzt werden, um die verschiedenen Arten von Beziehungen in einem sozialen Netzwerk zu modellieren. Dabei könnte beispielsweise dazwischen unterschieden werden, ob man miteinander befreundet ist, oder ob auch eine Verwandtschaft oder andere Beziehung zueinander besteht, wie in Abbildung 3 skizziert. Dabei könnten die grünen Kanten Freundschaften zwischen den Akteuren darstellen, während die gelben Kanten Verwandtschaft und die orangenen Kanten die Zugehörigkeit zum selben Sportverein anzeigen.

Ein anderer Spezialfall von Graphen sind **Digraphen** bzw. **gerichtete Graphen** $G* = (V^*, E^*)$, wie in Abbildung 4 skizziert. Die Kanten in einem gerichteten Graphen werden dann **gerichtete Kanten** oder **Bögen** genannt. Im Vergleich zu Kanten ist **Bögen** zusätzlich zu den zwei Knoten auch eine Richtung zugeordnet und bestehen im Gegensatz zu vorher aus *geordneten* Knotenpaaren (u^*, v^*), $u^*, v^* \in V^*, u^* \neq v^*$, wobei u^* den Anfangsknoten und v^* den Endknoten des Bogens bezeichnet. (u^*, v^*) und (v^*, u^*) bezeichnen also zwei verschiedene Bögen. Gerichtete Graphen sind nicht notwendigerweise Multigraphen. Zwischen zwei Knoten können bei gerichteten Graphen zwei Bögen mit entgegengesetzter Richtung bestehen.

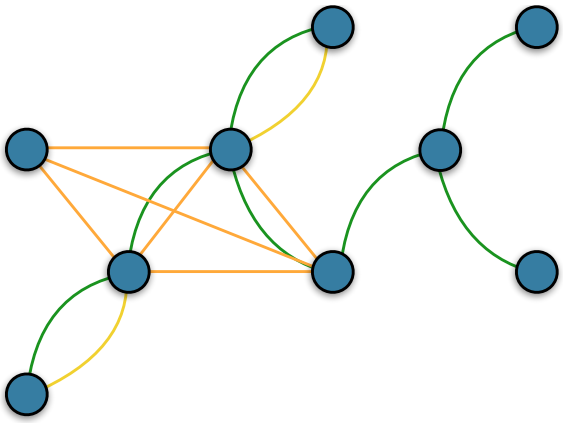

Abbildung 3: Beispiel eines Multigraphen

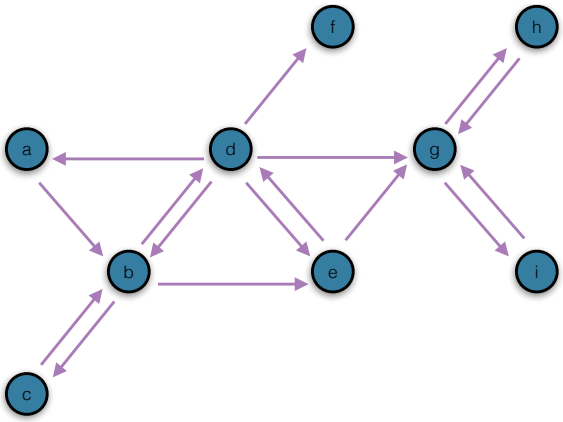

Abbildung 4: Beispiel eines Digraphen

Zwei Knoten eines ungerichteten Graphen heißen **adjazent**, wenn es eine Kante gibt, die beide Knoten miteinander verbindet. Analog dazu heißen zwei Kanten adjazent, wenn beide über einen gemeinsamen Knoten verbunden sind. Man spricht davon, dass ein Knoten **inzident** zu einer Kante ist, wenn der Knoten ein Endpunkt dieser Kante ist. Wenn man für einen gegebenen Knoten $v \in V$ die Menge der zugehörigen adjazenten Knoten $N(v) = \{u \in V | \{u, v\} \in E\}$ betrachtet, die auch als **Nachbarschaft** eines Knotens bezeichnet wird, so wird die Kardinalität $|N(v)|$ als **Grad** $deg(v)$ des Knotens v bezeichnet. Die Anordnung der Knotengrade eines Graphen nach aufsteigender Größe nennt man **Gradfolge**. Summiert man die Elemente einer solchen Gradfolge für einen Graphen G auf, so erhält man die doppelte Anzahl der Kanten in diesem Graphen, es gilt also $\sum_{v \in V} deg(v) = 2|E|$. Das lässt sich dadurch erklären, dass man den Grad eines Knotens statt über die Anzahl der adjazenten Knoten auch über die Anzahl der angrenzenden Kan-

ten berechnen kann. Eine Kante fließt daher immer zweimal, also einmal pro zugehörigen Knoten, in die Gradsumme mit ein. Daraus lässt sich folgern, dass die Gradsumme für jeden Graphen eine gerade Zahl ist. Für den Beispielgraphen in Abbildung 1 ist die resultierende Gradfolge beispielsweise $\{1, 1, 1, 2, 2, 2, 4, 4, 5\}$.

Bei gerichteten Graphen betrachtet man sowohl den **Eingangsgrad** $deg_{in}(v^*)$ als auch den **Ausgangsgrad** $deg_{out}(v^*)$ eines Knotens $v^* \in V$. Der Eingangsgrad $deg_{in}(v^*)$ gibt die Anzahl der Bögen an, die in v^* enden, und der Ausgangsgrad $deg_{out}(v^*)$ beschreibt die Anzahl der Bögen, für die v^* der Startknoten ist. Analog zum ungerichteten Fall lässt sich auch hier die **Eingangs-** und **Ausgangsgradfolge** eines gerichteten Graphen G^* definieren. Für den Beispielgraphen in 4 ist die Ausgangsgradfolge dementsprechend $\{0, 1, 1, 1, 1, 2, 2, 3, 5\}$ und die Eingangsgradfolge $\{1, 1, 1, 1, 1, 2, 2, 3, 4\}$.

Auf Graphen sind verschiedene Arten von Routen definiert, je nachdem, ob man Knoten oder Kanten mehrmals oder höchstens einmal passieren darf. Die grundlegende Route ohne Restriktionen wird **Weg** genannt. Auf einem Graphen $G = (V, E)$ wird ein Weg von einem Anfangsknoten $v_0 \in V$ zu einem Endknoten $v_l \in V$ durch eine abwechselnde Folge von Knoten und Kanten, die durchlaufen werden beschrieben werden, $(v_0, e_1, v_1, \ldots, v_{l-1}, e_l, v_l)$. Dabei ist e_i die Kante, die v_{i-1} und v_i miteinander verbindet. Die **Länge** l eines Weges ist die Anzahl der Kanten, die bei diesem Weg durchlaufen werden. Verbietet man nun das mehrmalige Durchlaufen eines Knotens, spricht man von einem **Pfad**, und verbietet man das mehrmalige Durchlaufen von Kanten, liegt ein **Trail** vor. Man beachte dabei, dass zwar jeder Pfad ein Trail ist, aber nicht jeder Trail auch ein Pfad, wie in Abbildung 5 dargestellt. Da der Trail (rechts) den Knoten b mehrmals passiert, liegt hier kein Pfad vor.

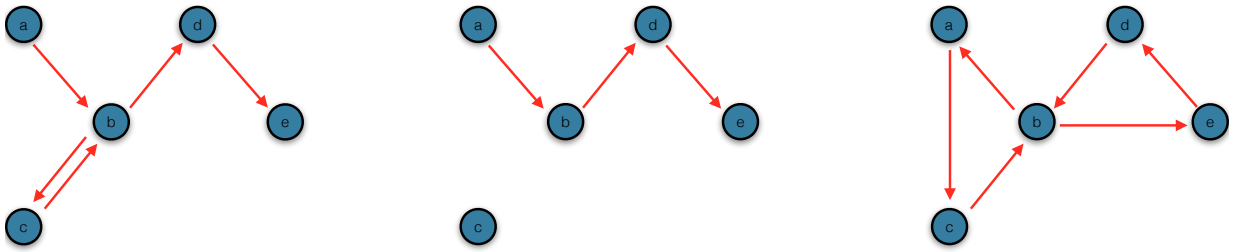

Abbildung 5: Beispiele für einen Weg, einen Pfad und einen Trail, der zugleich auch ein Kreis ist

Eine weitere Art von Routen in einem Graphen sind solche mit identischem Anfangs- und Endknoten. Ein Trail mit $v_0 = v_l$ nennt man einen **Kreis**. Ein Kreis der Länge $l \geq 3$, bei dem aber sonst alle Knoten nur einmal passiert werden, heißt **Zyklus**. Kommt kein Zyklus im gesamten Graphen vor, so spricht man von einem **azyklischen Graphen**. Die Definitionen können direkt auf den Fall von gerichteten Graphen übertragen werden, indem man statt Kanten die Bögen des Graphen betrachtet.

Die Knoten und Kanten eines Graphen können auch mit Gewichten versehen werden. Werden Kanten Gewichte zugewiesen, spricht man von **Kantengewichten** w_e, die Notation für **Knotengewichte** ist analog w_v. Die Länge eines Weges bei gewichteten Kanten wird nun durch Aufsummierung der einzelnen Kantengewichte berechnet. Die Länge von ungewichteten Graphen ist also ein Spezialfall mit $w_e = 1, \forall e \in E$. Gewichtete Kanten werden in der Praxis dazu benutzt, die Länge, die Wichtigkeit oder auch die Kapazität einer Kante darzustellen.

Oft ist es von Interesse, die **Kohäsion**, also den Grad der Vernetzung eines Graphen zu betrachten. Wenn man ein Knotenpaar v und u in einem Graphen G betrachtet, nennt man u **erreichbar** von v, wenn es einen Weg gibt, der von v nach u führt. Gilt, dass jeder Knoten u von jedem beliebigen anderen Knoten v in dem Netzwerk erreichbar ist, also $\exists l \in \mathbb{N} \forall u \in V \ \forall v \in V : \exists \ \{v = v_0, e_1, \ldots, e_l, u = v_l\}$, so nennt man den Graphen **verbunden**. Ein **unverbundener** Graph zerfällt in mehrere Komponenten. **Komponenten** eines Graphen sind maximale Subgraphen, die verbunden sind. Maximal bedeutet in diesem Zusammenhang, dass es keinen weiteren Knoten im Graphen gibt, den man zu der Komponente hinzunehmen könnte, ohne dass die Komponente nicht mehr verbunden ist. In Abbildung 6 ist rechts ein verbundener Graph mit der Knotenmenge $\{a, b, c, d, e, f, g, h, i\}$ und links ein unverbundener Graph, der aus zwei Komponenten besteht, zu sehen. Die erste Komponente besteht aus der Knotenmenge $\{a, b, c, d, e\}$ und die zweite Komponente aus der Knotenmenge $\{f, g, h, i\}$.

Für gerichtete Graphen unterscheidet man zwischen zwei Arten von Vernetzung. Man nennt einen gerichteten Graphen **schwach verbunden**, wenn es zwischen allen Knoten eine Verbindung gibt, falls man die Richtung ignoriert, also nur den zugrundeliegenden ungerichteten Graphen betrachtet. Im Gegensatz dazu heißt ein gerichteter Graph **stark verbunden**, wenn jeder Knoten von jedem anderen Knoten aus unter Berücksichtigung der Richtung

der Bögen erreichbar ist.

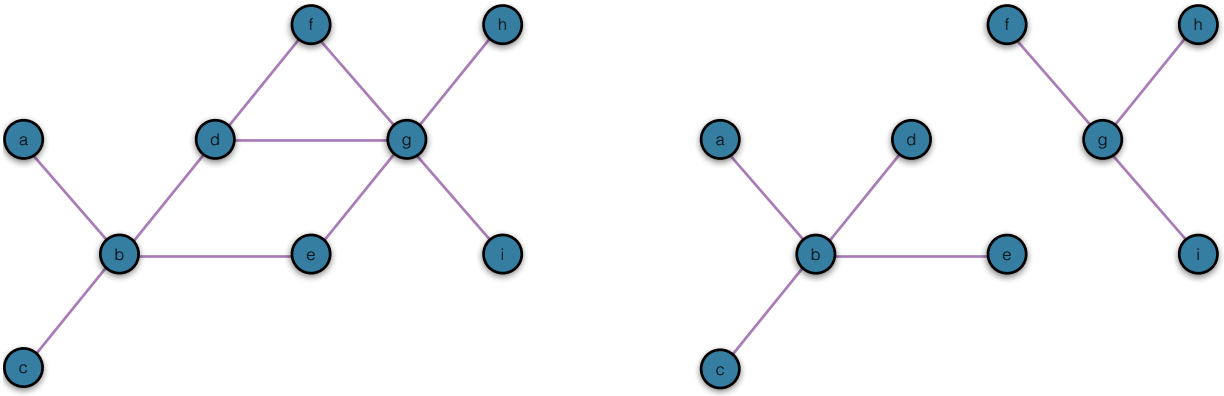

Abbildung 6: Beispiele für einen verbundenen Graphen (links), sowie einen unverbundenen Graphen mit zwei Komponenten (rechts)

Die **Distanz** oder auch **geodätische Distanz** $dist(u, v)$ zwischen zwei Knoten u und v in einem Graphen ist die Länge des kürzesten Pfades zwischen den beiden. Dabei muss der kürzeste Pfad nicht eindeutig sein, es kann auch mehrere Pfade mit minimaler Länge geben. $\max_{u,v \in V} d(u, v)$, also die größte Distanz in einem Graphen, wird **Durchmesser** des Graphen genannt. Auf gerichtete und gewichtete Graphen wird diese Definition entsprechend übertragen. Liegt ein unverbundener Graph vor, so wird für ungewichtete Graphen entweder die Anzahl der Knoten N_v als maximal mögliche Distanz in dem Graphen oder aber die maximale Distanz zweier Knoten in verbundenen Komponenten des Graphen angegeben.

2.2 Formen von Graphen

Teile von Graphen haben oft eine besondere Struktur. Einige solcher Strukturen, die besonders häufig vorkommen, sollen in diesem Unterkapitel vorgestellt werden.

In einem **kompletten** Graphen sind alle Knoten direkt miteinander verbunden. Für einen Graphen ohne Kantengewichte heißt das, dass $\forall u, v \in V : d(u, v) = 1$ gilt. Ist eine Teilmenge eines Graphen, also ein Subgraph, komplett, so nennt man diesen Subgraphen **Clique**. Eine solche Clique nennt man **maximal**, wenn man keinen Knoten, der noch nicht in der

Clique enthalten ist, zu ihr dazu nehmen kann, ohne dass der Subgraph nicht mehr komplett ist. Ein Beispiel für eines kompletten Graphen ist in Abbildung 7 zu sehen.

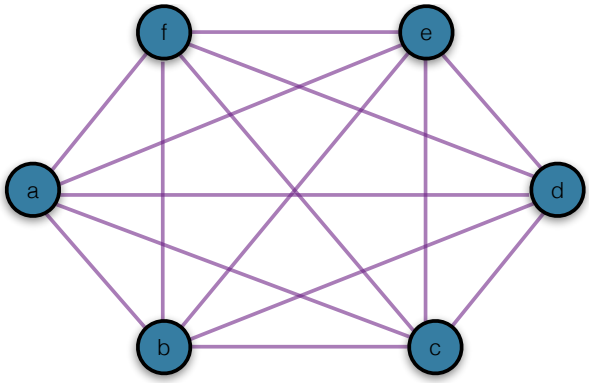

Abbildung 7: Beispiel eines kompletten Graphen

Haben alle Knoten in einem Graphen denselben Grad, so nennt man den Graphen **regulär**. Haben alle Knoten in dem Graphen den Grad d, so sagt man auch der Graph ist **d-regulär**. Solche d-regulären Graphen können inhaltlich so interpretiert werden, dass man von jedem Knoten aus immer d verschiedene andere Knoten erreichen kann, also eine spezielle Gitterstruktur in dem Netzwerk vorliegt. Ein Beispiel für einen 3-regulären Graphen ist in Abbildung 8 dargestellt.

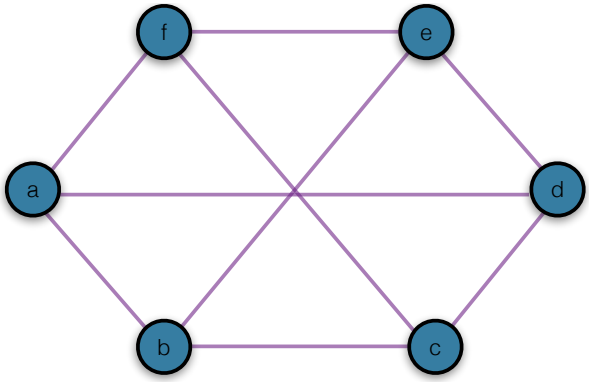

Abbildung 8: Beispiel eines 3-regulären Graphen

Ein azyklischer, zusammenhängender Graph heißt **Baum**. Besteht ein Graph aus mehreren unzusammenhängenden Subgraphen, wo jeder für sich alleine genommen ein Baum ist, so

nennt man den Graphen einen **Wald**. Sind die Kanten des Graphen mit einer Richtung versehen, so spricht man von einem **gerichteten** Baum. Bäume dieser Art haben oft einen Knoten, der der einzige Knoten in dem Baum ist, von dem aus man alle anderen Knoten erreicht und dieser Knoten heißt **Wurzel**. Entscheidungsbäume, wie in Abbildung 9 gezeigt, sind Beispiele für Bäume mit Wurzelnknoten, die man dann **gewurzelte Bäume** nennt.

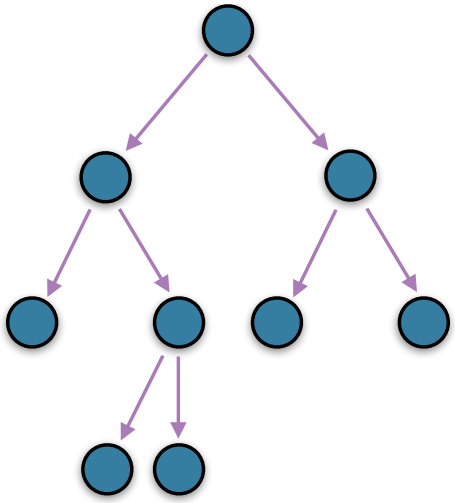

Abbildung 9: Beispiel eines Entscheidungsbaums

Der Knoten u^*, von dem ein Pfad zu einem anderen Knoten v^* führt, heißt **Vorfahre** von v^*. v^* wird dann **Nachkomme** von u^* genannt. Knoten unmittelbar vor anderen Knoten werden **Eltern**, Knoten unmittelbar nach anderen Knoten werden **Kinder** genannt. Ein Knoten, von dem kein Bogen wegführt, der also keine Kinder hat, wird **Blatt** genannt. Ein **DAG** ist ein gerichteter (directed), azyklischer Graph. Im Gegensatz zu einem gerichteten Baum, enthält ein DAG einen Zyklus, wenn man die Richtung der Kanten ignoriert. Ein solcher DAG ist in Abbildung 10 dargestellt. DAGs oder Baumstrukturen findet gerade im Design von effizienten Berechnungsalgorithmen Anwendung.

Manche Graphen erfüllen die Eigenschaft der Bipartitheit. **Bipartitheit** bedeutet, dass die Menge der Knoten V in zwei disjunkte Klassen V_1 und V_2 zerfällt, sodass $V_1 \cup V_2 = V$ gilt, und nur Knoten verschiedener Klassen mit einer Kante verbunden werden können. Man könnte zum Beispiel die Vereine und Spieler in der Fußball Bundesliga als ein Netzwerk

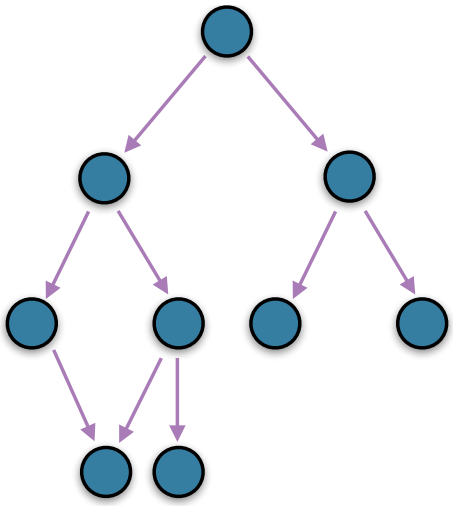

Abbildung 10: Beispiel eines DAG

darstellen, bei dem eine Kante zwischen einem Spieler-Knoten und einem Vereins-Knoten besteht, wenn der Spieler bereits für den Verein aktiv war. Für bipartite Graphen wird oft ein induzierter Graph $G_1 = (V_1, E_1)$ durch eine Knotenklasse V_1 definiert, wobei zwischen zwei Knoten aus V_1 eine Kante besteht, wenn beide im ursprünglichen bipartiten Graphen mindestens einen gemeinsamen Nachbarn in V_2 hatten. Eine Skizze von einem bipartiten Graphen und dem dazugehörigen durch die roten Knoten induzierten Graphen ist in Abbildung 11 dargestellt.

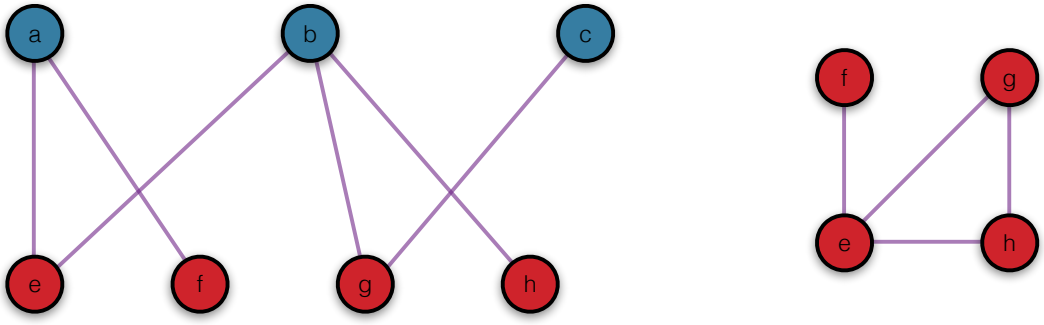

Abbildung 11: Beispiel eines bipartiten Graphen (links) und der durch die roten Knoten induzierte Graph (rechts)

2.3 Matrixdarstellung eines Graphen

Gerade bei größeren Graphen, die graphisch nur schwer darstellbar sind, macht es Sinn, die Struktur und die Eigenschaften eines Graphen G mithilfe von Matrizen in komprimierter Form darzustellen.

Die wichtigste Matrix, die die grundlegenden Verbindungen in einem ungerichteten Graphen wiedergibt, ist die sogenannte **Adjazenzmatrix A**. Dabei handelt es sich um eine symmetrische $N_v \times N_v$-Matrix mit binären Einträgen. Nummeriert man die Knoten eines Graphen mit 1 bis N_v durch, sind die Einträge der Matrix $\mathbf{A}$ definiert durch

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{für } \{i,j\} \in E \\ 0, & \text{sonst} \end{cases}$$

wobei $\{i,j\}$ für die Kante zwischen zwei Knoten i und j steht. Wenn also eine Kante zwischen dem i-ten und j-ten Knoten besteht, so ist der Eintrag in der i-ten Zeile und j-ten Spalte und der Eintrag der j-ten Zeile und i-ten Spalte der Adjazenzmatrix eine 1, ansonsten 0. Einen Beispielgraphen und die daraus resultierende Adjazenzmatrix finden sich in Abbildung 12.

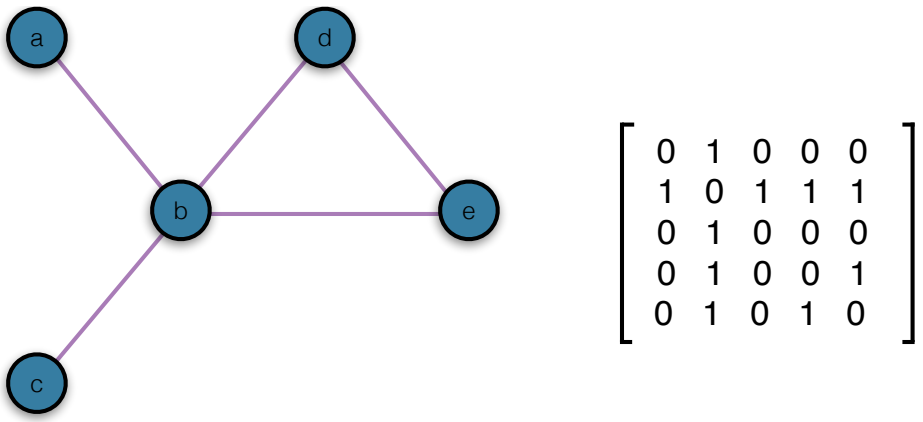

Abbildung 12: Beispiel eines Netzwerks und der dazugehörigen Adjazenzmatrix

Durch diese Struktur kann man über die Adjazenzatrix noch andere Informationen über den zugrundeliegenden Graphen extrahieren. Bildet man die Zeilensumme in der i-ten Zeile

$\mathbf{A}_{i+} = \sum_{j=1}^{N_v} \mathbf{A}_{ij}$, so erhält man den Grad von Knoten i. Den Grad von Knoten i würde man aufgrund der Symmetrie der Matrix auch durch die Spaltensumme der i-ten Spalte erhalten. Bildet man die r-te Potenz der Adjazenzmatrix $\mathbf{A}^r$, so erhält man in den Einträgen $\mathbf{A}^r_{ij}$ die Anzahl der Walks der Länge r zwischen Knoten i und j. Es gilt außerdem, dass G genau dann ein regulärer Graph ist, wenn der größte Grad des Graphen d_{max} ein Eigenwert von $\mathbf{A}$ ist. Zur Eigenwertberechnung wird auf den Appendix verwiesen.

Für Digraphen wird die Definition der Adjazenzmatrix insofern abgewandelt, dass die Einträge $\mathbf{A}_{ij}$ der Matrix nur dann 1 sind, wenn ein Bogen von i nach j existiert. Da ein Bogen von i nach j nicht per se auch einen Bogen von j nach i impliziert, ist die Adjazenzmatrix für Digraphen nur noch in Sonderfällen symmetrisch. Auch sind Spalten- und Zeilensumme nun entsprechend anders zu interpretieren. $\mathbf{A}_{i+}$ entspricht dem Ausgangsgrad des i-ten Knotens $deg(i)_{out}$ und $\mathbf{A}_{+j}$ dem Eingangsgrad des j-ten Knotens $deg(j)_{in}$.

Für gewichtete Graphen kann man die Adjazenzmatrix dahingehend abwandeln, dass die Einträge für die existierenden Kanten nicht 1 sind, sondern dem Gewicht entsprechen, welches der Kante zwischen i und j zugeordnet wurde.

Eine Abwandlung der Adjazenzmatrix ist die **Inzidenzmatrix B**. Hierbei handelt es sich um eine $N_v \times N_e$-Matrix mit binären Einträgen

$$\mathbf{B}_{ij} = \begin{cases} 1, \text{ wenn Knoten } i \text{ inzident zu Kante } j \text{ ist} \\ 0, \text{ sonst} \end{cases}$$

2.4 Datenstruktur und Algorithmen

2.4.1 Datenstruktur

Die Daten eines Netzwerkgraphen können üblicherweise auf zwei grundlegende Arten repräsentiert werden. Die Adjazenzmatrix ist eine naheliegende Art, die Verbindungen des Netzwerkes darzustellen, jedoch bringt sie gerade für große Netzwerke auch einige Probleme mit sich. Der Speicherbedarf einer solchen Matrix ist mit $\mathcal{O}(N_v^2)$ quadratischer Natur, was für große Graphen einen hohen Speicherbedarf bedeutet. Insbesondere wenn die ein-

zelnen Knoten eines solchen großen Graphen nicht eng miteinander vernetzt sind, ist der Speicheraufwand für eine solche Adjazenzmatrix groß, obwohl die Adjazenzmatrix zum Großteil nur Nulleinträge enthält.

Daher bietet es sich für solche Graphen an, statt der Adjazenzmatrix eine **Adjazenzliste** zu betrachten. Eine Adjazenzliste ist eine Liste, in der der i-te Eintrag die Knoten aufführt, zu denen der i-te Knoten eine direkte Kante hat. Die Länge der Einträge der Adjazenzliste ist im einfachen Graphen $2N_e$, im Digraphen N_e. Damit reduziert sich der Speicheraufwand zu $\mathcal{O}(N_v + N_e)$. Für den Fall, dass $N_e \sim N_v$, reduziert sich der Speicheraufwand also, während er für den Fall eines dichten Graphen, für den $N_e \sim N_v^2$ gilt, ähnlich wie für die Adjazenzmatrix ist.

Eine andere Variation der Adjazenzliste ist eine Liste der im Graphen vorkommenden Kanten. Hier besteht jeder Eintrag aus einem Knotenpaar, das durch eine Kante miteinander verbunden ist.

2.4.2 Algorithmen

Während viele Eigenschaften eines Graphen direkt aus der Datenstruktur abgelesen werden können, erfordern andere Fragestellungen komplexere Algorithmen. Solche Fragestellungen werden in zwei Kategorien eingeteilt. Die **lösbaren** Fragestellungen erfordern einen Rechenaufwand polynomer Ordnung $\mathcal{O}(n^p)$. Manche Fragestellungen erfordern aber einen exponentiellen Rechenaufwand der Ordnung $\mathcal{O}(a^n)$. Gerade bei großen Graphen ist ein Algorithmus mit exponentiellem Rechenaufwand, oder auch schon mit polynomialen Rechenaufwand bei großem p, gar nicht oder nur mit extrem großen Zeitaufwand durchführbar.

Oft wird ein Graph mittels Algorithmen nach bestimmten Strukturen, zum Beispiel Zyklen oder maximalen Komponenten, durchsucht. Dabei startet man bei einem Anfangsknoten und bewegt sich dann systematisch von diesem Knoten aus zu allen von hier erreichbaren Knoten. Man unterscheidet zwischen zwei grundlegenden Suchstrategien.

Beim **breadth-first-search**, kurz BFS, werden ausgehend von einem Ausgangsknoten zuerst direkt benachbarte Knoten erforscht, dann die Knoten, die zwei Kanten entfernt sind, usw., bis alle erreichbaren Knoten durchlaufen wurden. Ein Schema dieser Suchmethode

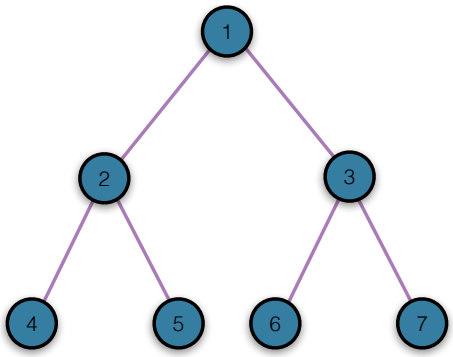

Abbildung 13: Suchschema eines breadth-first-Suchalgorithmus

wird in Abbildung 13 gezeigt. Die zugrundeliegende Struktur dieses Algorithmus ist ein Baum, bei dem der Pfad vom Anfangsknoten zu einem anderen Knoten dem kürzesten Pfad entspricht.

Die andere Variante, einen Graphen zu durchsuchen, ist **depth-first-search**, kurz DFS. Hierbei wird, wieder ausgehend von einem Anfangsknoten, zuerst über einen Nachbarknoten so weit wie möglich durch den Graphen geschritten, bevor dann nach und nach die Abzweigungen von diesem Pfad erforscht werden. Eine Skizze dieses Suchalgorithmus ist in Abbildung 14 dargestellt.

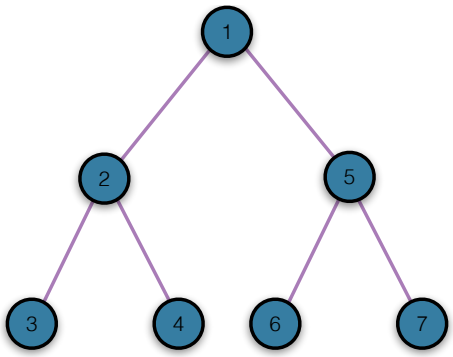

Abbildung 14: Suchschema eines depth-first-Suchalgorithmus

Welcher von beiden Suchalgorithmen gegebenenfalls sinnvoller ist, hängt von der Fragestellung ab, nach der der Graph durchsucht wird. Der BFS-Algorithmus wird oft für die Berechnung des kürzesten Pfade zwischen zwei Knoten benutzt, siehe auch den Algorith-

mus von Brandes im Appendix, während der DFS sich z.B. als Teil eines komplexeren Algorithmus zur Untersuchung, ob ein Graph azyklisch ist, bewährt hat.

2.5 Grundlegende Grapheigenschaften des Patentdatensatz

Bei dem zu untersuchenden Patentdatensatz handelt es sich um einen einfachen, ungerichteten, ungewichteten und nicht-bipartiten Graphen mit $N_v = 10\,208$ Knoten und N_e=21 976 Kanten. Der Graph ist nicht verbunden und zerfällt in mehrere Komponenten. Der Durchmesser des Graphen beträgt 30, wobei hier beachtet werden sollte, dass diese Zahl für einen unverbundenen Graphen dem maximalen Durchmesser einer verbundenen Komponente im Graphen entspricht.

3 Eigenschaften von Knoten und Kanten

In diesem Kapitel sollen nun einige Charakteristiken der Elemente eines Netzwerkgraphen präsentiert werden. Diese Charakteristiken lassen sich in zwei Kategorien unterteilen. In Unterkapitel 3.1 werden die Eigenschaften erläutert, die auf dem Knotengrad basieren, und in 3.2 werden verschiedene Konzepte zur Messung der Zentralität bzw. der Wichtigkeit eines Knotens sowie die Erweiterung auf die Wichtigkeit von Kanten vorgestellt.

3.1 Gradmaße

In einem Netzwerkgraphen $G = (V, E)$ ist der Grad d_v eines Knotens v als Anzahl der in v inzidenten Kanten des Graphen definiert. Aufgrund dieser Definition ist der Grad der Knoten ein Maß für die Vernetzung des Graphen. Hierfür betrachtet man dann die Gradfolge $\{deg(v)\}_{v \in V} = \{deg(1), \ldots, deg(N_v)\}$ und definiert auf dieser Grundlage verschiedene Maße. Für den Fall eines Digraphen wird anstelle des Grads jeweils der Eingangsgrad $deg_{in}(v)$ und der Ausgangsgrad $deg_{out}(v)$, sowie die entsprechenden Gradfolgen $\{deg_{in}(v)\}_{v \in V}$ und $\{deg_{out}(v)\}_{v \in V}$ betrachtet.

3.1.1 Gradverteilung

Für den Grad der einzelnen Knoten eines Netzwerkgraphen kann man nun eine Dichtefunktion f definieren. Dabei ist f_d der Anteil an Knoten $v \in V$ mit Grad $deg(v) = d$. Die dazugehörige Verteilungsfunktion ist gegeben durch $F_d = \sum_{k=0}^{d} f_k$.

Gerade für große Netzwerke ist die Gradverteilung eine einfache Möglichkeit die Konnektivität des Graphen zusammenzufassen. Für den Patentdatensatz ist die Gradverteilung als Histogramm in Abbildung 15 dargestellt. Der minimale Grad des Patentnetzwerks ist 1, der maximale Grad 56. Der Knotengrad 2 kommt mit 2147-mal am häufigsten im Datensatz vor. Insgesamt ist die Verteilung stark rechtsschief. Der Durchschnittsgrad liegt etwa bei 4.3, der Mediangrad bei 3.

Plottet man die Häufigkeit der verschiedenen Grade auf einer log-log-Skala, erhält man für den Patentdatensatz Abbildung 16. Das ist gerade bei solchen Verteilungen wie hier sinnvoll, da zwar ein Großteil der Knoten einen geringen Grad hat, es aber auch viele einzelne Knoten mit bedeutend größerem Grad gibt. Oft liegt eine **Power-Law** Komponente in der

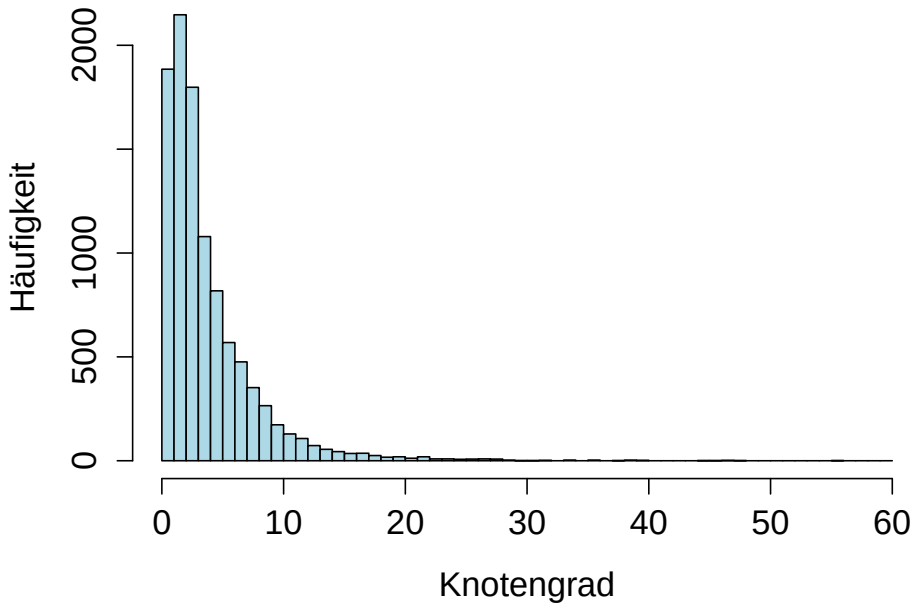

Abbildung 15: Knotenverteilung des Patentdatensatzes

Verteilung vor, d.h.

$$f_d \propto d^{-\alpha},\ \alpha \in \mathbb{R}. \tag{1}$$

Die Unbekannte in der Gleichung, die es zu schätzen gilt, ist also α. Hierzu gibt es verschiedene mehr oder weniger vorteilhafte Ansätze. Logarithmiert man beide Seiten der Gleichung (1), folgt

$$\log(f_d) \sim C - \alpha \log(d), \tag{2}$$

wobei C eine Konstante ist. Womöglich ist der intuitive Ansatz hierzu, mithilfe einer linearen Regression von $\log(d)$ auf $\log(f_d)$ einen Schätzer für α zu erhalten. Ein Beispiel für die resultierende Regressionsgerade ist 16 gegeben. Die lineare Regression liefert $\hat{\alpha} = 2.544$ und eine Anpassungsgüte von $R^2 = 0.907$.

In der Praxis kann dieser Ansatz jedoch problematisch sein, da es wegen der hohen Knotengrade mit geringer Häufigkeit zu großer Ungenauigkeit der Schätzung kommen kann. Eine Möglichkeit, dieses Problem zu umgehen, ist, statt der normalen Häufigkeit die kumulierte Häufigkeit $F(d)$ zu benutzen. Die kumulierte Randhäufigkeit nimmt dann folgende Form

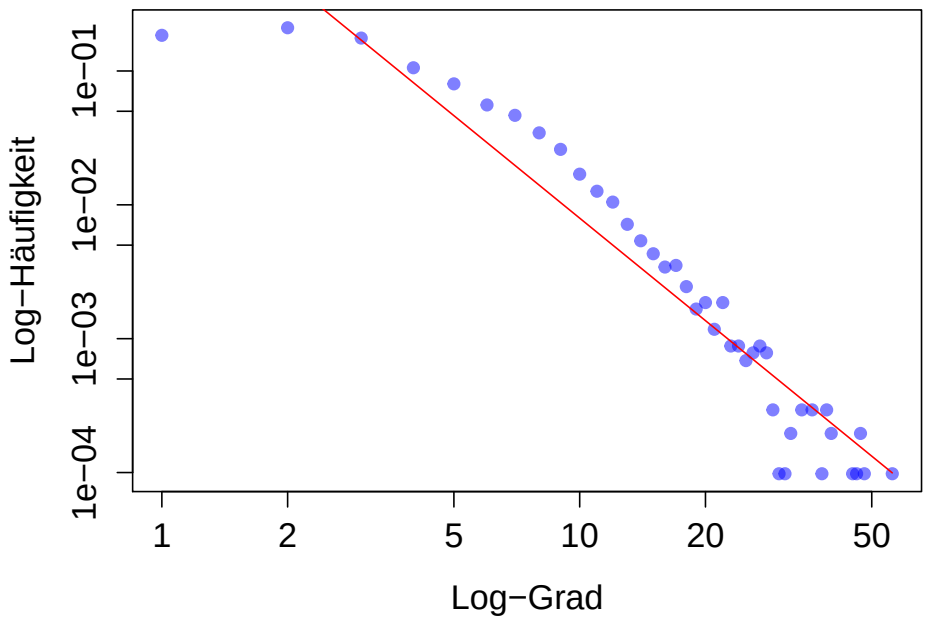

Abbildung 16: Knotenverteilung des Patentdatensatzes in einer log-log-Skala mit linearer Regressionsgerade

an

$$\bar{F}(d) = 1 - F(d) \sim d^{-(\alpha-1)}. \tag{3}$$

Im Anschluss kann hier wieder ein regressionsbasierter Ansatz benutzt werden, um α zu schätzen.

Eine weitere Variante ist die Schätzung von α über die Hill-Schätzer $\hat{\gamma}_k$

$$\hat{\alpha}_k = 1 + \hat{\gamma}_k^{-1}, \tag{4}$$

$$\text{mit } \hat{\gamma}_k = \frac{1}{k} \sum_{i=0}^{k-1} \log \left(\frac{d_{(N_v-i)}}{d_{(N_v-k)}} \right), \tag{5}$$

wobei $d_{(1)} \leq \ldots \leq d_{(N_v)}$ die geordneten Knotengrade sind. k ist dabei ein Wert, der selbst gewählt werden muss. Die Wahl von k erfolgt, indem $\hat{\gamma}_k$ für verschiedene Werte von k geplottet wird und einen Wert für k gewählt wird, bei dem die Werte $\hat{\gamma}_k$ sich stabilisiert haben. Ein solcher Hill-Plot für das Patentnetzwerk ist in Abbildung 17 gegeben, siehe

auch Drees et al. (2000). Es wurde ein Wert bei einem Knotengrad von 9 ausgewählt. Der Hill-Schätzer beträgt $\hat{\gamma}_k = 0.43$. Damit ist der entsprechende Schätzer $\hat{\alpha} \approx 3.33$ und es gilt $\bar{F}(d) = 1 - F(d) \sim d^{-2.33}$.

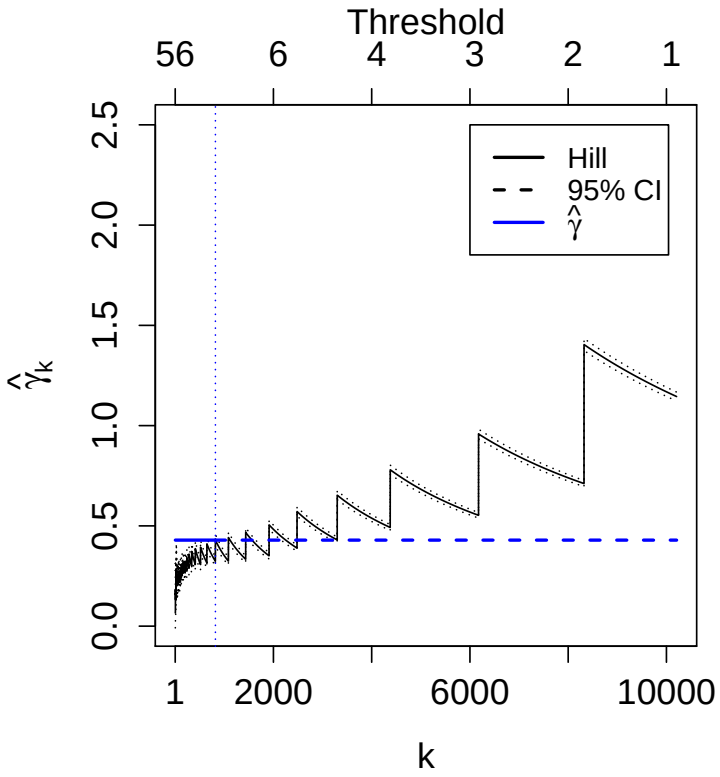

Abbildung 17: Hillplot für den Patentdatensatz

Eine andere interessierende Eigenschaft eines Netzwerkes kann sein, welche Knoten mit welchen Knoten verbunden sind. Dazu ist die Gradfolge alleine nicht ausreichend, da sie nicht spezifiziert, wie die einzelnen Knoten miteinander verbunden sind. Zwei Netzwerkgraphen können dieselbe Gradfolge haben, jedoch trotzdem strukturell komplett unterschiedlich aufgebaut sein.

3.1.2 Gradkorrelation

Um den Zusammenhang zwischen zwei Knoten in Abhängigkeit von deren Graden genauer zu beschreiben, führt man ein 2-dimensionales Pendant zur Gradverteilung ein. Hierzu betrachtet man die Häufigkeit, mit der zwei Knoten verbunden sind, von denen der eine Knoten den Grad d_1 und der andere den Grad d_2 hat. Bei Digraphen ist das geordnete Knotenpaar $e = (v_1, v_2)$, das eine Kante beschreibt, klar definiert, bei ungerichteten Graphen muss man die Knoten jedoch nach einer bestimmten Logik ordnen.

Eine Möglichkeit besteht darin, die Kanten $e = (v_1, v_2) \in E$ derart zu sortieren, dass $d(v_1) \leq d(v_2)$. Für jedes Paar $d_1 < d_2$ wird dann die Hälfte der relativen Häufigkeit zu f_{d_1,d_2} zugeordnet und die andere Hälfte zu f_{d_2,d_1}. Für den Fall $d_1 = d_2$ wird f_{d_1,d_2} die relative Häufigkeit komplett zugeordnet. Damit ist die so definierte Verteilung $\{f_{d,d'}\}$ symmetrisch.

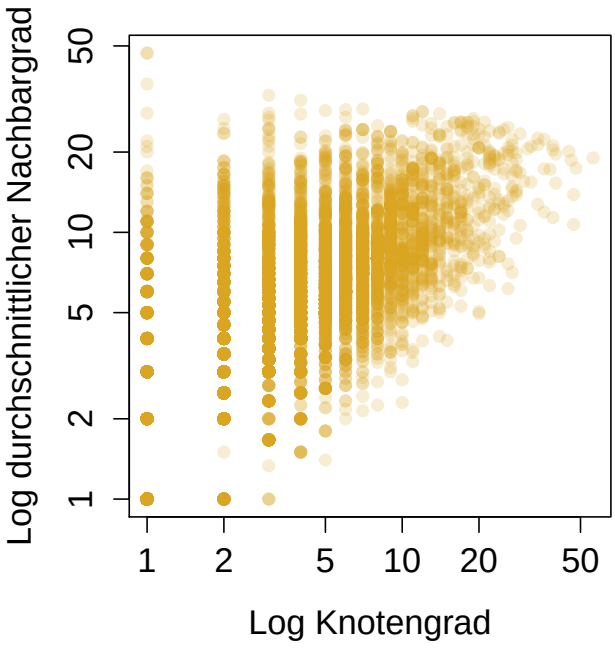

Abbildung 18: Gradkorrelation

Eine graphische Darstellung der Knoten, die miteinander verbunden sind, ist in Abbildung 18 dargestellt. Hier wurde jedem Knoten der durchschnittliche Knotengrad seiner Nachbarn

zugeordnet. Wie hier zu sehen ist, sind Knoten mit niedrigem Grad sowohl mit anderen Knoten mit niedrigem Grad als auch mit Knoten mit höherem Grad verbunden. Knoten höheren Grades sind jedoch tendenziell auch mit anderen Knoten höheren Grades verbunden.

Aufbauend auf dieser Verteilung kann man auch die bedingte Verteilung $f_{d'|d}$ betrachten. Mit dieser bedingten Verteilung wird die Wahrscheinlichkeit beschrieben, dass ein Knoten mit Grad d mit einem anderen Knoten mit Grad d' verbunden ist. Abbildungen der Mittelwerte dieser bedingten Verteilungen

$$\bar{d}(d) = \sum_{d'} d' f_{d'|d} \tag{6}$$

als von d abhängige Funktion können die Art des Zusammenhangs zwischen Knoten mit hohem und niedrigem Grad wiedergeben.

Eine andere, einfache Maßzahl für die Gradkorrelation ist die Korrelation, die durch die gemeinsame Verteilung $f_{d,d'}$ und ihren marginalen Verteilungen definiert ist. Trotz der Definition der verschiedenen Maßzahlen für die Gradkorrelation ist es ratsam zu betrachten, was inhaltlich für das entsprechende Netzwerk Sinn macht oder überhaupt möglich ist.

3.2 Zentralitätsmaße

Zentralitätsmaße zielen darauf ab, die “Wichtigkeit” eines Knotens in einem Netzwerk zu quantifizieren. Es gibt verschiedene Auffassungen darüber, was die Wichtigkeit und damit die Zentralität eines Knotens ist, und daher gibt es auch zahlreiche unterschiedliche Zentralitätsmaße. Im Folgenden werden die gängigsten Zentralitätsmaße vorgestellt und erläutert.

3.2.1 Gradzentralität

Ein gängiges Maß für die Zentralität eines Knotens haben wurde bereits in Kapitel 3.1 vorgestellt: den Knotengrad. Die Gradzentralität $C_D(v)$ eines gegebenen Knotens v, für einen Graphen $G = (V, E)$ mit N_v Knoten und N_e Kanten ist definiert als der Grad des Knotens $deg(v)$. Für Digraphen wird entsprechend zwischen Eingangsgradzentralität und Ausgangszentralität unterschieden. Im Gegensatz zu anderen Zentralitätsmaßen ist die Gradzentralität ein lokales Maß. Zur Berechnung für einen bestimmten Knoten ist nur

die Anzahl der direkten Nachbarn relevant, der Rest des Graphen hat keinen Einfluss.

Everett and Borgatti (1999) erweitern die Definition von Gradzentralität auf Gruppen von Knoten. Die Gradzentralität einer Gruppe ist definiert als die Anzahl von Knoten, die mit Knoten der Gruppe verbunden sind. Ist ein Knoten mit mehreren Knoten der Gruppe verbunden, wird dies trotzdem nur einmal gezählt.

3.2.2 Nähezentralität

Die grundlegende Idee der Nähezentralität ist, dass die Wichtigkeit eines Knotens darüber definiert ist, wie nahe er zu anderen Knoten des Netzwerkes ist. Sei $G = (V, E)$ ein ungerichteter Graph. Die Nähezentralität eines Knotens v ist definiert als

$$c_{Cl}(v) = \frac{1}{\sum_{u \in V} dist(v, u)}, \tag{7}$$

wobei $dist(v, u)$ die geodätische Distanz zwischen den Knoten u und v bezeichnet. Um die Vergleichbarkeit der Nähezentralität zwischen Graphen verschiedener Größen zu erhalten, wird das Maß auf das Intervall $[0, 1]$ normiert, indem man es mit dem Faktor $N_v - 1$ multipliziert. Dabei bedeutet ein Wert von 1, dass alle Knoten $u \in V$ in der direkten Nachbarschaft von v liegen. Für Knoten, die nicht miteinander verbunden sind, wird hier meist die Anzahl der Knoten N_v als Distanz genommen.

Für die Berechnung der geodätischen Distanz zwischen zwei Knoten muss der kürzeste Weg zwischen den Knoten gefunden werden. Um die Länge des kürzesten Wegs zwischen zwei Knoten zu berechnen, wird der Dijkstra-Algorithmus benutzt, der im Appendix genauer erläutert wird.

Die Definition von Nähezentralität wird kompliziert, wenn der zu untersuchende Graph nicht verbunden ist, da die geodätische Distanz dann für ein Knotenpaar, das nicht miteinander verbunden ist, den Wert ∞ annimmt und $c_{Cl}(v)$ damit 0 wird. Eine Möglichkeit, auch für solche Graphen eine Aussage über die Nähezentralität zu treffen, besteht darin, die Nähezentralität für verbundene Komponenten des Graphen separat zu berechnen. Wenn der Graph aus einer giant component, also einer verbundenen Komponente, die einen Großteil der Knoten in dem Graphen enthält, besteht, so beschränkt man die Analyse häufig nur

	$\mathbf{N_v}$	$\mathbf{N_e}$	Mean	Minimum	Maximum
Komponente 1	969	3353	0.1851	0.1172	0.2715
Komponente 2	754	3101	0.1098	0.0659	0.1591
Komponente 3	386	1085	0.0934	0.0592	0.1322
Komponente 4	325	930	0.1913	0.1020	0.2962
Komponente 5	149	548	0.2795	0.1263	0.4077
Komponente 6	125	488	0.2717	0.1981	0.4052
Komponente 7	109	403	0.3087	0.1878	0.4887

Tabelle 1: Übersicht über die Nähezentralität der größten Komponenten

auf diese giant component. Ist das nicht der Fall, so kann man die geodätische Distanz für zwei unzusammenhängende Knoten umdefinieren. Statt dem Wert ∞ kann der Distanz zum Beispiel der Wert N_v zugeordnet, werden. Die Wahl von N_v ist damit zu begründen, dass die maximal mögliche geodätische Distanz von zwei Knoten in einem zusammenhängenden Graphen $N_v - 1$ ist.

Für die Knoten im Patentdatensatz wird die normierte Nähezentralität betrachtet. Die durchschnittliche Nähezentralität beträgt hier $9.988\ 10^{-5}$, der Knoten mit der kleinsten Nähezentralität hat einen Wert von $9.797\ 10^{-5}$, der Knoten mit der höchsten Nähezentralität hat einen Wert von $1.082\ 10^{-4}$. Ingesamt sind die Werte sehr nahe an 0. Ein Grund dafür ist, dass der Graph unverbunden ist und daher in einzelne Komponenten zerfällt. Für Knoten v, u aus verschiedenen Komponenten wird daher $dist(v, u) = N_v = 10208$ gewählt. Um das zu umgehen, wurden minimale, maximale und durchschnittliche Nähezentralität in den Komponenten des Patentdatensatz mit mehr als 100 Knoten in Tabelle 1 dargestellt. Hier ist zu sehen, dass sich für die einzelnen Komponenten wesentlich größere Werte ergeben.

Wie schon zuvor die Gradzentralität kann auch die Definition von Nähezentralität auf Gruppen von Knoten angewandt werden, wie von Everett and Borgatti (1999) vorgestellt.

3.2.3 Intermediationszentralität

Intermediationszentralität beschreibt, inwiefern ein Knoten auf den einzelnen Pfaden eines Graphen liegt. Knoten, die auf vielen Pfaden liegen, werden hier als wichtiger bzw. zentraler für das Netzwerk bewertet als andere. Freeman (1977) definiert die Intermediati-

	$\mathbf{N_v}$	$\mathbf{N_e}$	**Median in** 10^{-3}	**Minimum**	**Maximum**
Komponente 1	969	3353	0.0000	0.0000	0.1817
Komponente 2	754	3101	0.0000	0.0000	0.5146
Komponente 3	386	1085	0.0000	0.0000	0.5152
Komponente 4	325	930	0.0000	0.0000	0.4795
Komponente 5	149	548	1.5300	0.0000	0.2151
Komponente 6	125	488	0.0164	0.0000	0.2516
Komponente 7	109	403	0.0000	0.0000	0.3827

Tabelle 2: Übersicht über die Intermediationszentralität der größten Komponenten

onszentralität eines Knotens v als

$$c_B(v) = \sum_{s \neq t \neq v \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}, \tag{8}$$

wobei $\sigma(s,t|v)$ die Anzahl der kürzesten Pfade zwischen s und t, die durch v führen, ist und $\sigma(s,t)$ die Gesamtanzahl der kürzesten Pfade zwischen s und t ist. Wenn die kürzesten Pfade zwischen zwei Knoten in einem Graphen eindeutig sind, so misst $c_B(v)$ die Anzahl an kürzesten Pfade in G, die durch v gehen. Eine Normierung kann auch hier durch die Division mit dem Faktor $(N_v - 1)(N_v - 2)/2$ erfolgen.

Um die Intermediationszentralität für alle Knoten v in G zu berechnen, muss die Länge aller kürzesten Pfade zwischen allen Knotenpaaren bestimmt und für jeden Knoten aufsummiert werden. Daher ist die Berechnung der Ordnung $\mathcal{O}(N_v^3)$ gerade für große Netzwerke sehr aufwändig. Eine Berechnungsalternative mit geringerem Aufwand ist mit dem Brandes-Algorithmus gegeben, der die Rechenzeit auf $\mathcal{O}(N_v N_e)$ verkürzt. Eine genauere Beschreibung des Algorithmus ist im Appendix gegeben.

Für den Patentdatensatz wird wieder die normierte Intermediationszentralität betrachtet. Statt des Mittelwerts der Intermediationszentralität wird hier nun der Median betrachtet, da extreme Werte vorliegen, die den Mittelwert verzerren würden. Die minimale Intermediationszentralität sowie der Medianwert liegen bei 0, die maximale Intermediationszentralität bei 2.797 10^{-3}. In Tabelle 2 sind die entsprechenden Werte wieder für die sieben größten Komponenten gegeben. Das Minimum liegt jeweils immer bei 0, der Median bis auf Komponenten 5 und 6 ebenfalls. Die Maxima variieren zwischen 0.18 und 0.52.

	$\mathbf{N_v}$	$\mathbf{N_e}$	**Median in** 10^{-2}	**Minimum in** 10^{-2}	**Maximum**
Komponente 1	969	3353	0.00150	0.00000	1.00000
Komponente 2	754	3101	0.00003	0.00000	1.00000
Komponente 3	386	1085	0.00118	0.00000	1.00000
Komponente 4	325	930	2.15500	0.00001	1.00000
Komponente 5	149	548	10.17000	0.00002	1.00000
Komponente 6	125	488	8.59100	0.01728	1.00000
Komponente 7	109	403	8.22800	0.05720	1.00000

Tabelle 3: Übersicht über die Eigenvektorzentralität der größten Komponenten

3.2.4 Eigenvektorzentralität

Ein viertes Zentralitätsmaß misst die Wichtigkeit bzw. die Zentralität eines Knotens danach, wie zentral die Nachbarn eines Knotens sind. Bonacich (1972), basierend auf Katz (1953), definiert ein Eigenvektorzentralitätsmaß

$$c_{Ei}(v) = \alpha \sum_{\{u,v\} \in E} c_{Ei}(u), \tag{9}$$

wobei $c_{Ei} = (c_{Ei}(1), \ldots, c_{Ei}(N_v))^T$ die Lösung zum Eigenwertproblem $Ac_{Ei} = \alpha^{-1} c_{Ei}$ mit der Adjazenzmatrix $\mathbf{A}$ ist. Eine Wiederholung der Berechnung von Eigenwerten und Eigenvektoren einer Matrix ist im Appendix gegeben. Nach Bonacich (1972) ist der größte Eigenwert von $\mathbf{A}$ die optimale Wahl von α^{-1}. Ist ein Graph G zusammenhängend und ungerichtet, so ist der größte Eigenwert eindeutig und der dazugehörige Eigenvektor besteht aus Einträgen ungleich null, die alle dasselbe Vorzeichen haben.

Für den Patentdatensatz liegt das Minimum sowie der Medianwert bei 0. Die maximale Eigenvektorzentralität eines Knoten beträgt 1. In Tabelle 3 sind die entsprechenden Werte für die sieben größten Komponenten gegeben. Das Maximum liegt jeweils bei 1, das Minimum reicht von 0 bis zu 5.72 10^{-4}.

Die vorgestellten vier Zentralitätsmaße haben unterschiedliche Auffassungen von Wichtigkeit von Knoten, daher können die Zentralitätsbewertungen für einzelne Knoten voneinander abweichen. In Abbildung 19 sind vier Targetplots der Komponente 7 für die eben vorgestellten Zentralitätsmaße gegeben. Der rote Knoten mit dem höchsten Wert ist in jedem Fall derselbe Knoten, er bekommt also von allen vier Maßen die höchste Zentralitätsbewertung. Dieses Ergebnis lässt sich jedoch nicht auf den gesamten Graphen ausdehnen.

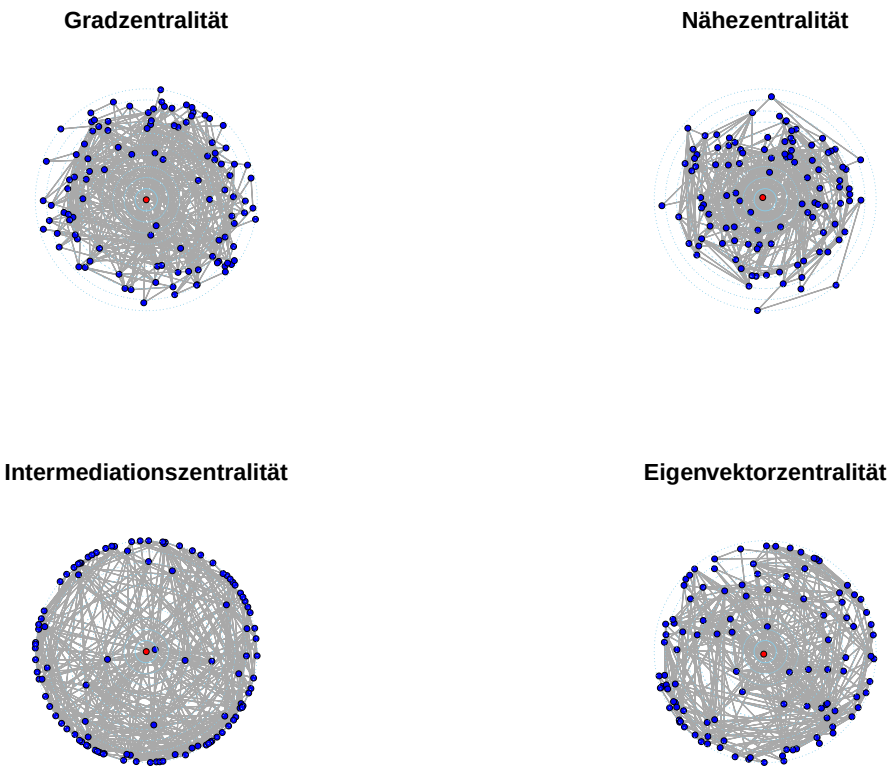

Abbildung 19: Target-Plot für die vier Zentralitätsmaße.

3.2.5 Erweiterung auf Kantenzentralität

Alle bisher vorgestellten Maße haben gemeinsam, dass sie die Zentralität von Knoten betrachtet. Es gibt jedoch auch Zentralitätsmaße, die statt auf Knoten auf Kanten angewandt werden können, wie zum Beispiel die Intermediationszentralität. Statt den Anteilen der kürzesten Pfade, die durch einen Knoten v gehen, werden hier die Anteile der kürzesten Pfade, die durch eine Kante $e = (u, v), u, v/inV$ gehen, betrachtet, siehe Girvan and Newman (Girvan and Newman).

Andere Zentralitätsmaße sind nicht direkt auf Kanten übertragbar. Um solche Maße trotzdem anwenden zu können, wird der **duale Graph** $G'' = (V'', E'')$ eines Graphen $G = (V, E)$ verwendet. Bei einem dualen Graphen werden Kanten und Knoten gewissermaßen vertauscht. Das heißt, dass die Knoten $v'' \in V''$ die Kanten $e \in E$ darstellen. Die Kanten $e'' \in E''$ stehen dafür, dass die beiden dazugehörigen Kanten im ursprünglichen Graphen

G inzident in einem gemeinsamen Knoten sind, siehe Brandes and Erlebach (2005). Ein Beispiel für einen Graphen und den dazugehörigen dualen Graphen ist in Abbildung 20 gegeben.

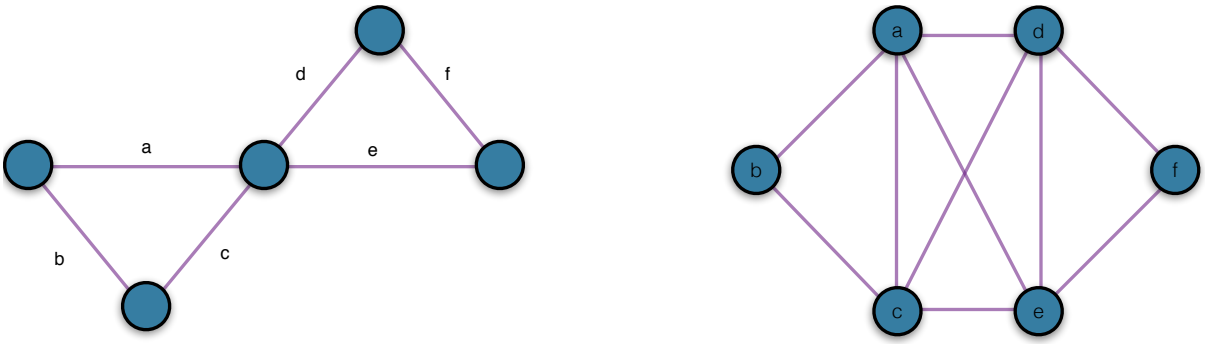

Abbildung 20: links: Beispielgraph mit Kanten a, b, c, d, e, f; rechts: dazugehöriger dualer Graph

4 Netzwerkkohäsion

Während in Kapitel 3 Netzwerkcharakteristika basierend auf den Knoten präsentiert wurden, soll nun der Zusammenhang des Netzwerks genauer untersucht werden. Wieder gibt es verschiedene Maße für den Zusammenhalt eines Netzwerkes, die auf unterschiedliche Fragestellungen eine Antwort geben sollen. Wie schon bei den Zentralitätsmaßen gibt es auch hier lokale und globale Herangehensweisen.

4.1 Lokale Dichte

Oft ist es von Interesse, ob eine Teilmenge der Knoten lokal dicht beieinander liegt. Das klassische Beispiel für solche eng-vernetzten Knoten ist die Clique, also eine Teilmenge von Knoten, bei der jeder Knoten ein direkter Nachbar des anderen ist. Je größer eine Clique ist, desto seltener kommt sie in der Praxis vor, da das Netzwerk dazu sehr dicht vernetzt sein muss. Turán (1941) hat gezeigt, dass es für die Existenz einer n-Clique in einem Graphen hinreichend ist, wenn $N_e > (N_v^2/2)[(n-2)/(n-1)]$ gilt. Jedoch ist diese Bedingung in der Praxis gerade für größere n selten erfüllt, da N_e und N_v oft von ähnlicher Ordnung sind. Für den Patentdatensatz ist diese Bedingung bereits für Cliquen der Ordnung $n = 3$ nicht erfüllt. Dabei ist darauf hinzuweisen, dass trotzdem Cliquen der Ordnung $n = 3$ oder höher im Patentdatensatz bestehen können. Die **Cliquenzahl** $\omega(G)$ gibt die Anzahl der Knoten in der maximalen Clique von G an. Für den Patentdatensatz ergibt sich eine Cliquenzahl von $\omega(G) = 16$.

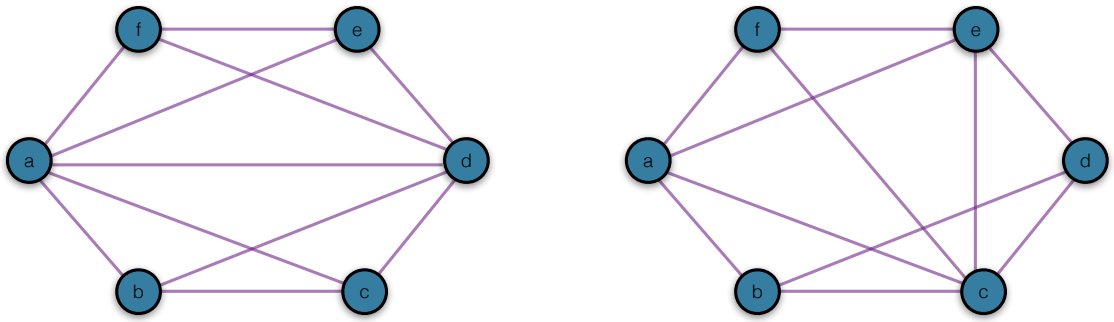

Abbildung 21: links: Beispiel eines 3-Cores; rechts: Beispiel eines 3-Plex

Eine andere, weniger restriktive Art als Cliquen, Netzwerkkohäsion zu messen, sind Plexe. Ein **n-Plex** ist ein Subgraph, der aus m Knoten besteht, wobei $m > n$ gilt, und in dem

kein Knoten einen geringeren Grad als $m - n$ hat. Wenn kein Knoten einen geringeren Grad als $m-n$ hat, darf jeder Knoten mit maximal $n-1$ anderen Knoten nicht verbunden sein. Da in einer Clique jeder Knoten den Grad $m-1$ hat, ist jede Clique auch ein 1-Plex.

In der Praxis hat es sich jedoch als sehr aufwändig erwiesen, maximale Cliquen oder Plexe zu finden. Eine weitere Lockerung der Anforderungen an eine lokale Struktur führt zu Cores. Ein **k-Core** ist ein Subgraph, in dem jeder Knoten mindestens den Grad k hat, und von dem es keine Obermenge mit dieser Eigenschaft gibt, der diesen Subgraphen enthält. In Abbildung 21 ist ein Beispiel für einen 3-Core und einen 3-Plex gegeben. Ein maximaler Core kann in $\mathcal{O}(N_v+N_e)$ berechnet werden, verglichen mit $\mathcal{O}(N_v^{2,376})$ für maximale Cliquen. Ein Targetplot der Cores in Komponente 7 ist in Abbildung 22 gegeben. Die schwarzen Kreise am äußersten Rand stehen für 1-Cores, die roten Kreise für 2-Cores, usw.. Die maximalen Cores sind 8-Cores, von denen es 9 in der Komponente gibt und die durch die grauen Kreise in der Mitte des Targetplots dargestellt werden.

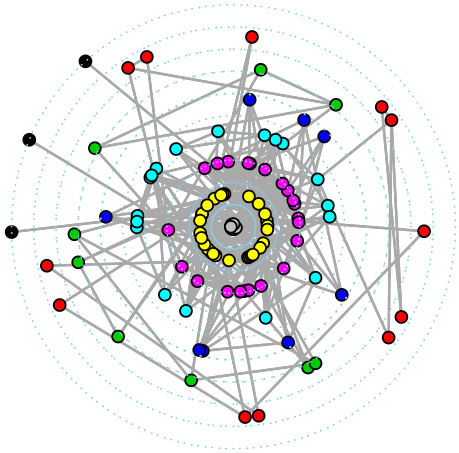

Abbildung 22: Targetplot der Cores in Komponente 7

Die Maße, die wir bis jetzt kennengelernt haben, basieren alle darauf, dass man im Graphen nach vordefinierten Strukturen sucht. Alternativ kann man Maße für die lokale Dichte eines Graphen definieren, indem man die Anzahl der Kanten mit der Anzahl der möglichen Kanten vergleicht. So folgt für die Dichte eines Subgraphen $H = (V_H, E_H)$

$$den(H) = \frac{|E_H|}{|V_H|(|V_H| - 1)/2}. \tag{10}$$

Die Dichte ist auf das Intervall $[0, 1]$ normiert und sagt aus, wie sehr der Subgraph einer Clique ähnelt. Ist die Dichte 1, so existieren in dem Subgraph alle möglichen Kanten und es liegt eine Clique vor. Wählt man $H = G$, so erhält man ein Maß für die Dichte des gesamten Graphen. Für den Patentdatensatz ergibt sich hier eine Dichte von $den(G) = 4.2183\ 10^{-4}$.

Eine andere Herangehensweise, den gesamten Graphen zu beschreiben, wurde von Watts and Strogatz (1998) beschrieben und liegt in der Berechnung der Dichte für die Nachbarschaft $N(v)$ eines jeden Knotens v. Bildet man dann das arithmetische Mittel über $den(N(v))$ für alle Knoten $v \in V_H$, so kann man das Ergebnis als einen Clusterkoeffizienten für den gesamten Graphen G betrachten.

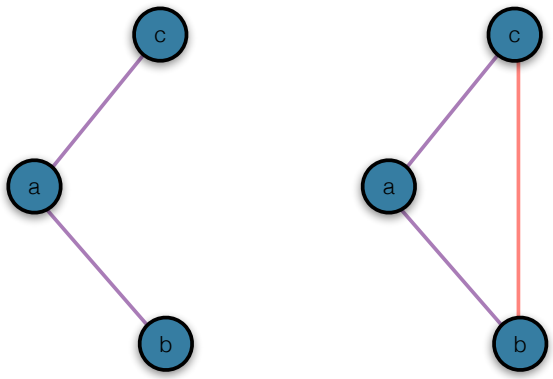

Abbildung 23: Links: 2-Star, Rechts: Triangle

Eine weitere Möglichkeit, den Grad der Clusterbildung in dem Graphen zu beschreiben ist, wie häufig 2-Stars, also drei Knoten, die durch zwei Edges verbunden sind, zu Triangles, also einer 3er-Clique, werden, wie in Abbildung 23 dargestellt. Sei dazu $\tau_\Delta(v)$ die Anzahl der Triangles, zu denen ein Knoten v gehört, und $\tau_3(v)$ die Anzahl an 2-Stars, zu denen v gehört. $\tau_3(v)$ lässt sich als $\binom{n}{d_v}$ berechnen. Der Clusterkoeffizient nach Watz Strogatz

$den(H_v)$ lässt sich damit für Knoten v mit $\tau_\Delta(v) > 0$ umschreiben zu

$$den(H_v) = cl(v) = \frac{\tau_\Delta(v)}{\tau_3(v)}. \tag{11}$$

Analog wird dann der Clusterkoeffizient für ganz G geschrieben als

$$cl(G) = \frac{1}{V'} \sum_{v \in V'} cl(v), \tag{12}$$

wobei $V' \subset V$ die Menge an Knoten beschreibt, für die $d_v \geq 2$ gilt.

Da der Clusterkoeffizient in Gleichung (12) jedoch ein Mittelwert über die Mittelwerte $cl(v)$ ist, kann es informativer sein, stattdessen das gewichtete Mittel

$$\frac{\sum_{v \in V'} \tau_3(v) cl(v)}{\sum_{v \in V'} \tau_3(v)} \tag{13}$$

zu betrachten. Das kann umgeschrieben werden zu

$$cl_T(G) = \frac{3\tau_\Delta(G)}{\tau_3(G)}, \tag{14}$$

wobei $\tau_\Delta(G) = 1/3 \sum_{v \in V} \tau_\Delta(v)$ der Anzahl von Triangles im Graphen und $\tau_3(G) = \sum_{v \in V} \tau_3(v)$ der Anzahl von 2-Stars entspricht. Die Kennzahl $cl_T(G)$ in Gleichung (14) wird auch **Transitivität** eines Graphen genannt. Sie beschreibt, in welchem Anteil von Fällen die Knoten der 2-Stars auch ein Triangle bilden.

Da sich die beiden Clusterkoeffizienten cl und cl_T zwar oft ähnlich verhalten, es aber Extremfälle gibt, in denen beide stark voneinander abweichen, ist es wichtig, immer genau anzugeben, welcher von beiden Koeffizienten benutzt wird.

Im Patentdatensatz liegen 30 055 Triangles und 157 398 2-Stars vor. Damit ergibt sich für den Clusterkoeffizienten $cl_T(G) = 0.5728$. Während die beiden vorgestellten Clusterkoeffizienten auf der Basis von Triangles, also 3-Cycles, berechnet werden, können solche Clusterkoeffizienten auch für Cycles mit höherem k berechnet werden, siehe Newman (2010) oder Fronczak et al. (2002).

Knotenanzahl	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Häufigkeit	722	433	236	104	60	33	30	33	17	18	11	6	4
Knotenanzahl	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**24**	**26**	**27**	**28**	**29**
Häufigkeit	6	3	6	2	3	2	2	1	2	1	1	1	2
Knotenanzahl	**30**	**31**	**32**	**33**	**34**	**45**	**49**	**55**	**60**	**78**	**80**	**82**	**95**
Häufigkeit	1	1	1	1	1	2	2	1	1	1	1	1	1
Knotenanzahl	**109**	**125**	**149**	**325**	**386**	**754**	**969**						
Häufigkeit	1	1	1	1	1	1	1						

Tabelle 4: Übersicht über die Komponenten des Patentdatensatz

4.2 Konnektivität

Während im vorhergehenden Unterkapitel in erster Linie nach zusammenhängenden Unterstrukturen geringer Größe gesucht wurde, soll nun untersucht werden, wie der Graph vernetzt ist, und, falls es sich um ein unverbundenes Netzwerk handelt, wie der Graph in verbundene Komponenten zerfällt.

4.2.1 Verbundene Komponenten und "Small Worlds"

Eine verbundene Komponente eines Graphen ist ein möglichst großer verbundener Subgraph. Ein Graph, der nicht verbunden ist, lässt sich in einzelne verbundene Komponenten unterteilen. Ob ein Graph verbunden ist, oder nicht, sowie seine Unterteilung in verbundene Komponenten lässt sich mit BFS oder DFS-Algorithmen in $\mathcal{O}(N_v + N_e)$ Rechenzeit herausfinden.

Bei nicht-verbundenen Graphen gibt es häufig eine giant component. Eine **giant component** ist ein verbundener Subgraph, der den Großteil der Knoten des Graphen enthält. In solchen Fällen werden dann die weiteren Analysen nur auf die giant component angewandt.

Für den Patentdatensatz ist in Tabelle 4 eine Übersicht der einzelnen maximalen Komponenten gegeben. Da die größte Komponente mit 969 Knoten nicht einmal 10% der Knoten umfasst, kann hier nicht von einer giant component gesprochen werden. Besonders die hohe Anzahl an kleinen Komponenten ist auffällig. Über 40% der Knoten entfallen auf Komponenten mit 5 oder weniger Knoten.

Zusammenhängende Graphen oder giant components weisen zudem manchmal die "small-

world"-Eigenschaft auf. Die erste Idee dazu kam von Milgram (1967), der behauptet hat, dass jeder Mensch durchschnittlich über nur sechs Bekannte mit jedem anderen Menschen auf der Erde verbunden ist. Generell beschreibt das "small-world"-Phänomen die Eigenschaft von großen Netzwerken, bei denen die durchschnittliche Distanz zwischen zwei Knoten im Vergleich zu ihrer Größe relativ klein ist. Formal ausgedrückt gilt, dass ein Netzwerkgraph diese Eigenschaft erfüllt, wenn

$$\bar{l} = \frac{1}{N_v(N_v+1)/2} \sum_{u \neq v \in V} dist(u, v) \tag{15}$$

kleiner gleich $\mathcal{O}(\log N_v)$ ist. Watts and Strogatz (1998) haben beobachtet, dass eine geringe durchschnittliche Distanz in dem Graphen mit einem hohen Clusterkoeffizienten einhergeht.

Eine andere interessante Fragestellung hinsichtlich Konnektivität ist, wieviel Einfluss einzelne Knoten oder Kanten eines Netzwerkes auf die Konnektivität eines Netzwerkgraphen haben. In Abbildung 24 ist ein Spezialfall zu sehen, bei dem der rote Knoten die einzige Verbindung zwischen dem linken und dem rechten Teil des Graphen ist. Würde man diesen Knoten entfernen, wäre der Graph nicht mehr verbunden.

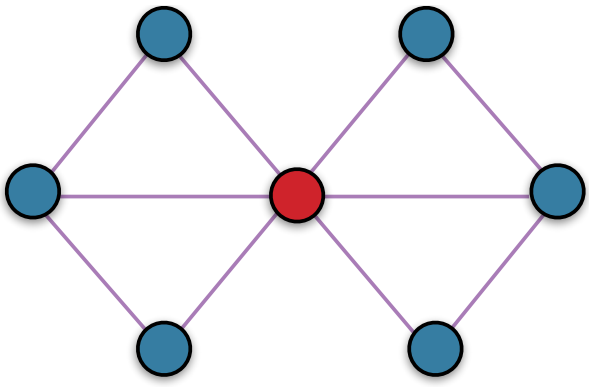

Abbildung 24: Beispiel eines 1-Knoten verbundenen Graphen

Ein Graph ist **k-Knoten-verbunden**, wenn die zwei folgenden Bedingungen erfüllt sind:

1. $N_v > k$
2. Wenn man weniger als k Knoten des Graphen entfernt, ist der überbleibende Graph

verbunden.

Dieselbe Definition lässt sich auch auf Kanten eines Graphen beziehen. Ein Graph ist demnach **k-Kanten-verbunden**, wenn der Graph mindestens $N_e = 2$ Kanten hat und das Entfernen von weniger als $k < N_e$ Kanten den Graphen verbunden lässt.

Basierend auf dieser Definition ist die **Knoten/Kanten-Konnektivität** durch den größten ganzzahligen Wert k gegeben, für den der Graph k-Knoten/Kanten-verbunden ist. Die Knoten-Konnektivität ist nach oben durch die Kanten-Konnektivität beschränkt, während die Kanten-Konnektivität wiederum durch den geringsten Knotengrad $\min deg(v)$ im Graphen nach oben beschränkt ist.

Die sieben größten Komponenten des Patentdatensatz haben alle eine Knoten- und Kanten-Konnektivität von 1. Das heißt, schon das Entfernen von einem bestimmten Knoten oder einer bestimmten Kanten würde die Komponenten in mehrere Subkomponenten zerfallen lassen.

Ein direktes Kriterium, wann ein Graph k-Knoten/Kanten-verbunden ist, liefert das Theorem von Menger. Es besagt, dass ein nichttrivialer Graph genau dann k-Knoten/Kanten-verbunden ist, wenn alle nicht-adjazenten Knoten $u, v \in V, u \neq v$ über einen Pfad mit k unterschiedlichen Knoten/Kanten erreicht werden können.

Eine Menge von Knoten oder Kanten, ohne die der Graph unverbunden ist, nennt man **Knoten/Kanten-Cut**. Meist ist man daran interessiert, herauszufinden, was der kleinstmöglichste Knoten/Kanten-Cut ist. Muss nur ein wohlgewählter Knoten aus dem Graphen entfernt werden, um ihn in Subkomponenten zerfallen zu lassen, nennt man einen solchen Knoten **Cut-Knoten**. Solche Cuts lassen sich genauer beschreiben. So ist ein u-v-Cut eine Partition der Knoten in zwei unverbundene, nichtleere Knotenmengen $S, \bar{S} \subset V$, bei der Knoten $u \in S$ und Knoten $v \in \bar{S}$. Liegen an den Kanten zusätzlich Gewichte w_e vor, so nennt man einen solchen u-v-Cut minimal, wenn die Summe der Gewichte an Kanten, die Knoten in S mit Knoten in $\bar{S}$ verknüpfen, minimal ist. Sind alle Kanten mit $w_e = 1$ gewichtet, so ist es äquivalent, den minimalen u-v-Cut oder den Kanten-Cut mit der geringsten Anzahl an Kanten zu finden, der den Graphen in zwei Komponenten teilt, wobei u ein Teil der ersten und v ein Teil der zweiten Komponente ist. Wenn die geringste Mächtigkeit solcher minimaler Kanten-Cut Mengen für Knoten $u, v \in V, u \neq v$ gleich k ist, so ist die

	N_v	Anzahl an Cut-Knoten	relativer Anteil an Cut-Knoten
Komponente 1	969	87	0.0898
Komponente 2	754	76	0.1008
Komponente 3	386	63	0.1633
Komponente 4	325	39	0.1200
Komponente 5	149	13	0.0872
Komponente 6	125	11	0.0880
Komponente 7	109	10	0.0917

Tabelle 5: Übersicht über Anzahl und Anteil der Knoten, die durch ihr Entfernen die Komponente in Subkomponenten zerfallen lassen in den größten Komponenten

Kanten-Konnektivität des Graphen $k - 1$.

Betrachtet man einen Digraphen statt eines ungerichteten Graphen lassen sich die zuvor präsentierten Konzepte anpassen. Da man bei der Verbundenheit von Digraphen zwischen schwach und stark verbunden unterscheidet, wird auch bei der Definition von k-Knoten- und k-Kanten-verbunden zwischen schwach und stark unterschieden. Ein Digraph ist schwach k-Knoten/Kanten-verbunden, wenn die Bedingungen für einfache Graphen für das zugrundeliegende Netzwerk ohne Richtung erfüllt ist. Sind die Bedingungen sogar erfüllt, wenn man die Richtungen der Kanten berücksichtigt, so ist er stark k-Knoten/Kanten-verbunden. Die Definition für Cuts bleibt unverändert, bis auf dass man nun die Richtung der Kanten berücksichtigt. Daher wird nun bei einem cut $(S, \bar{S})$ eine der Mengen, z.B. S als **Source** und $\bar{S}$ als **Sink** bezeichnet, um die Richtung der Bewegung von S zu $\bar{S}$ wiederzugeben.

Da für die sieben größten Komponenten des Patentdatensatz schon das Entfernen von einem ausgewählten Knoten zum Zerfallen führt, wurde nun zudem in Tabelle 5 betrachtet, wieviele Knoten diese Eigenschaft besitzen. Komponente 3 scheint am instabilsten zu sein. Der Anteil an Cut-Knoten ist hier mit ca. 16% fast doppelt so hoch wie in Komponenten 1, 5 oder 6.

4.3 Graphenpartitionierung

Oft macht es Sinn, einen Graphen zu partitionieren. Eine Partition einer Menge S ist ganz allgemein eine Unterteilung der Menge in disjunkte, nichtleere Untermengen $\mathcal{C} =$

$(C_1, \ldots, C_K)$ von S, sodass gilt $\dot{\bigcup}_{k=1}^{K} C_k = S$. Bei Netzwerkgraphen wird eine Partition vorgenommen, um zusammenhängende Cluster von Knoten im Netzwerk zu finden. Das Ziel einer Partitionierung ist also, die Knoten zu Untermengen zusammenzufassen, die in sich eine besonders hohe Kohäsion aufweisen. Dabei wird eine Untermenge als besonders kohäsiv angesehen, wenn die enthaltenen Knoten unter sich stark vernetzt und gleichzeitig von den anderen Knoten des Netzwerks relativ gut getrennt sind.

Die Menge der Kanten, die die Knoten aus zwei beliebigen Mengen $C_k, C_l \in \mathcal{C}, k \neq l$ verbinden, sei $E(C_k, C_l)$. $E(C_k)$ sei analog die Menge der Kanten, die Knoten innerhalb der Menge C_k miteinander verbinden. Formal ausgedrückt sucht man für gegebenen Graphen $G = (V, E)$ nach einer Partition $\mathcal{C} = (C_1, \ldots, C_K)$ der Menge der Knoten V, sodass $E(C_k, C_l)$ im Vergleich zu $E(C_k)$ und $E(C_l)$ klein ist. Zwei Methoden der Graphenpartitionierung werden im Folgenden genauer vorgestellt.

4.3.1 Hierarchisches Clustering

Hierarchisches Clustering ist ein generelles Konzept, aus dem viele andere Partitionierungstechniken abgeleitet wurden, die sich hinsichtlich ihrer Clusterkriterien und zugrundeliegenden Optimierungsalgorithmen unterscheiden. Man unterscheidet zwischen agglomerativen und divisiven Verfahren. **Agglomerative** Verfahren gehen zunächst von jedem Knoten einzeln aus, um eng verbundene Knoten nach und nach in Cluster zusammenzufassen, während **divisive** Verfahren von der Gesamtmenge der Knoten V ausgehen, um diese nach und nach in möglichst weit entfernte Cluster zu unterteilen.

Beide Arten von Verfahren gibt es mit verschiedenen Feinheiten der Endpartition, von dem Extremfall, in dem jeder Knoten für sich alleine einen Cluster bildet, also die Partition $\mathcal{C} = \{\{v_1\}, \ldots, \{v_{N_v}\}\}$, bis hin zu dem anderen Extrem, in dem das Netzwerk nicht unterteilt ist. Um die Cluster auf den verschiedenen Ebenen graphisch darzustellen, wird im Allgemeinen auf ein Dendrogramm zurückgegriffen, wie in Abbildung 25 dargestellt.

Sowohl für agglomerative als auch für divisive Verfahren muss man vorher Kohäsion quantifizieren. Dafür gibt es verschiedene Maße, die zumeist darauf basieren, die (Un-)Ähnlichkeit x_{ij} zwischen zwei Knoten $v_i, v_j \in V$ oder auch zwischen zwei Knotenmengen C_i und C_j mit $i \neq j$ zu beschreiben. Zwei übliche Ansätze, um die Ähnlichkeit von Knotenmengen zu beschreiben, sind das single-linkage und das complete-linkage Verfahren. Das **single-**

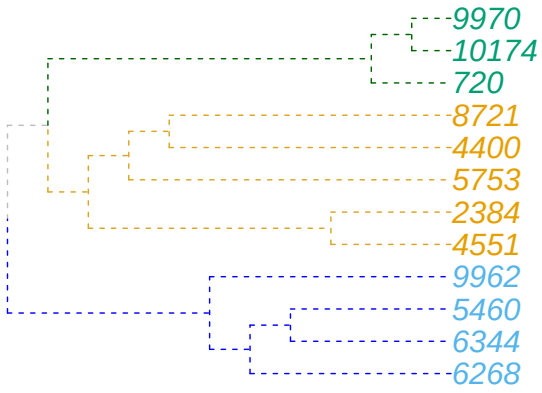

Abbildung 25: Beispiel eines Dendogramms

linkage Verfahren definiert die Unähnlichkeit bzw. den Abstand $D_{single-linkage}(C_1, C_2)$ für zwei Knotenmengen C_1 und C_2 als das Minimum über alle x_{ij}, für die $v_i \in C_1$ und $v_j \in C_2$ ist, also

$$D_{single-linkage}(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2} x_{ij}. \tag{16}$$

Das **complete-linkage** Verfahren hingegen definiert die Unähnlichkeit zwischen Mengen C_1 und C_2 als

$$D_{complete-linkage}(C_1, C_2) = \max_{v_i \in C_1, v_j \in C_2} x_{ij}. \tag{17}$$

Auch für die Unähnlichkeit x_{ij} zwischen zwei Knoten selbst gibt es verschiedene Maße. Die "Euklidische Distanz" Unähnlichkeit beispielsweise ist definiert als

$$x_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}, \tag{18}$$

woebei $\mathbf{A}$ die Adjazenzmatrix des Graphen ist. Diese Unähnlichkeit misst die euklidische Distanz zwischen den Zeilen i und j.

Jedoch benutzen nicht alle hierarchischen Clustermethoden ein Unähnlichkeitsmaß für Knoten. So optimiert Newman (2010) stattdessen die sogenannte Modularität einer Parti-

tion. Die Modularität $\text{mod}\,(\mathcal{C})$ einer Partition $\mathcal{C} = (C_1, \ldots, C_K)$ ist gegeben durch

$$\text{mod}\,(\mathcal{C}) = \sum_{k=1}^{K} [f_{kk}(\mathcal{C}) - f_{kk}^*(\mathcal{C})]^2, \tag{19}$$

wobei $f_{kk}(\mathcal{C})$ der Anteil der Kanten E ist, der Knoten aus C_k miteinander verbindet, und f_{kk}^* der erwarteten Anteil der Kanten ist, die bei einer zufälligen Kantenzuweisung Knoten aus C_k miteinander verbinden würden. Häufig wird f_{kk}^* als $f_{k+}f_{+k}$ definiert, also als Produkt der k-ten Zeilen- und Spaltensumme der Matrix $K = (f_{ij})$. Große Modularitätswerte deuten darauf hin, dass die Vernetzung innerhalb der einzelnen Mengen aus der Partition $\mathcal{C}$ über die zufällige Vernetzung hinausgehen, und weisen damit auf eine Gruppenstruktur hin.

Der Vorteil an dieser Herangehensweise ist, dass ein einziges Qualitätsmaß auf alle möglichen Partitionen angewandt und damit nicht nur eine Hierarchie erstellt wird, sondern direkt auch die optimale Partition in dieser Hierarchie ausgewählt wird.

Für Komponente 7 des Patentdatensatzes ist in Abbildung 26 eine Partitionierung durch hierarchisches agglomeratives Clustering mithilfe der Optimierung der Modularität dargestellt. Das dazugehörige Dendogramm wird aus Gründen der Übersichtlichkeit nicht abgebildet.

4.3.2 Spektralpartitionierung

Eine andere Herangehensweise zur Partitionierung von Netzwerkgraphen ist die Spektralpartitionierung. Sie benutzt die Eigenwertanalyse von Graphmatrizen, um Rückschlüsse auf die Konnektivität des Graphen zu ziehen. Die zwei gängigsten Methoden basieren auf der Adjazenz- und der Laplace-Matrix eines Graphen G.

Für die erste Methode wird zunächst eine Spektralanalyse der Adjazenzmatrix durchgeführt. Hierbei werden die (maximal) N_v Eigenwerte sowie die dazugehörigen Eigenvektoren bestimmt. Für die genaue Berechnung wird auf den Appendix verwiesen. Die resultierenden, nach der Größe geordneten Eigenwerte $\lambda_1 \leq \cdots \leq \lambda_{N_v}$ sowie die dazugehörigen Eigenvektoren $x_1, \ldots, x_{N_v}$ erfüllen dann die Gleichung

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i. \tag{20}$$

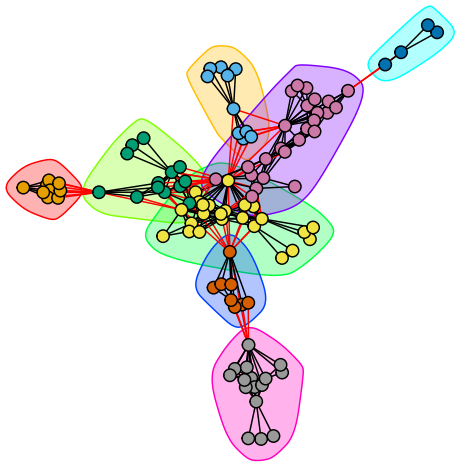

Abbildung 26: Partitionierung von Komponente 7 durch hierarchisches agglomeratives Clustering mithilfe der Optimierung der Modularität

Dann wird zuerst der betragsmäßig größte Eigenwert betrachtet und dessen Eigenvektor-Einträge ebenfalls der Größe nach geordnet. Die Knoten mit besonders großen negativen oder positiven Einträgen in diesem Eigenvektor, sowie ihre direkte Nachbarschaft, wird dann zu einem Cluster zusammengefasst. In der Praxis werden so normalerweise nur die K größten Eigenwert-Eigenvektorpaare untersucht, wobei $K \sim \log N_v$.

Die Idee hinter dieser Herangehensweise ist folgende: Wenn ein Graph eigentlich aus zwei d-regulären Graphen besteht, die nur durch wenige Knoten miteinander verbunden sind, so werden die zwei größten Eigenwerte der Matrix ähnlich groß wie d sein und die anderen Eigenwerte der Adjazenzmatrix werden deutlich geringer sein. Es wird also einen deutlichen Unterschied zwischen dem zweit- und dem drittgrößten Eigenwert geben. Die dazugehörigen beiden Eigenvektoren werden zudem für Knoten des einen Clusters stark positive Werte und für die Knoten des anderen Clusters stark negative Werte aufweisen.

Das Problem dieser Methode liegt in der Idee selbst. Oft sind die Cluster, in die ein Graph unterteilt werden soll, deutlich nicht regulär, sondern es liegt in dem Graphen eine starke

Streuung der Knotengrade vor. Als Resultat wird die resultierende Partition eine Trennung nach Knotengrad sein, die die zugrundeliegende Gruppenstruktur oft nicht erfasst. Eine Lösung, die Gkantsidis et al. (2003) vorgeschlagen haben, umfasst eine Umformung der Adjazenzmatrix, sodass die Zeilensummen alle 1 sind.

Eine andere Methode basiert auf der Spektralanalyse der Laplace-Matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{21}$$

wobei $\mathbf{A}$ die Adjazenzmatrix ist und $\mathbf{D} = diag[(d_v)]$. Laut Kolaczyk (2009) besteht ein Graph G genau dann aus K verbundenen Komponenten wenn gilt

$$\lambda_1(\mathbf{L}) = \ldots = \lambda_K(\mathbf{L}) = 0 \tag{22}$$

und

$$\lambda_{K+1}(\mathbf{L}) > 0. \tag{23}$$

Der kleinste Eigenwert der Laplace-Matrix $\mathbf{L}$ ist immer gleich 0 und hat den dazugehörigen Eigenvektor $\mathbf{x}_1 = (1, \ldots, 1)^T$. Wenn also vermutet wird, dass der Graph G annähern aus $K = 2$ Komponenten besteht, würde man nach obiger Aussage erwarten, dass auch $\lambda_2(\mathbf{L}) \approx 0$.

Da eine Bisektion, also eine Partition des Graphen in zwei Teile, oft von Interesse ist, wird in diesen Fällen der **isoperimetrische Wert** eines Graphen betrachtet. $\phi(S, \bar{S}) = |E(S, \bar{S})|/|S|$ ist der Anteil des Cuts $(S, \bar{S})$. Um eine möglichst gute Bisektion des Graphen zu erreichen, muss man eine derartige Menge S finden, wo der Anteil der Kanten, die Knoten aus S und $\bar{S}$ verbinden, möglichst klein ist. Der isoperimetrische Wert ist daher definiert als

$$\phi(G) = \min_{S \subset V : |S| \leq N_{v/2}} \phi(S, \bar{S}). \tag{24}$$

Die Minimierung der Anteile ist rechnerisch sehr aufwändig, jedoch kann man untere und obere Schranken für den isoperimetrischen Wert angeben:

$$\frac{\lambda_2}{2} \leq \phi(G) \leq \sqrt{\lambda_2(2d_{\max} - \lambda_2)}, \tag{25}$$

wobei $d_{\max}$ den höchsten Grad in G und λ_2 der zweitgrößte Eigenwert der Laplace-Matrix

ist. Nimmt λ_2 also kleine Werte an, so wird auch $\phi(G)$ klein.

Nach Fiedler (1973) wird eine Bisektion nun vorgenommen, indem man den zu λ_2 gehörigen Eigenvektor $\mathbf{x}_2$ betrachtet. Die Knoten, für die der Eigenvektor positive Einträge hat, werden der einen Knotenmenge zugeordnet:

$$S = \{v \in V : \mathbf{x}_2(v) \geq 0\}. \tag{26}$$

Die Knoten, für die der Eigenvektor negative Einträge hat, werden $\bar{S}$ zugeordnet:

$$\bar{S} = \{v \in V : \mathbf{x}_2(v) < 0\}. \tag{27}$$

Der Eigenwert λ_2 wird auch **Fiedler-Wert** und der dazugehörige Eigenvektor $\mathbf{x}_2$ **Fiedler-Vektor** gennant. Eine solche Spektralbisektion wird daher als Approximation genutzt, um den besten Cut für $\phi(G)$ zu erhalten.

Oftmals soll ein Graph in mehr als zwei Partitionen unterteilt werden. Eine Herangehensweise ist hier, die oben vorgestellte Bisektion als iteratives Verfahren an den erhaltenen Partitionen nochmals durchzuführen. Eine Verbesserung dieses Verfahren schlägt Newman (2006) vor. Hierbei wird statt der Laplacematrix die Modularitätsmatrix $\mathbf{B}$ mit den Einträgen

$$\mathbf{B}_{ij} = \mathbf{A}_{ij} - \frac{deg(i)deg(j)}{N_e} \tag{28}$$

optimiert.

Das Resultat einer solchen Spektralpartition von Komponente 7 ist in Abbildung 27 abgebildet. Dabei werden Knoten zwar verglichen mit der hierarchischen Partitionierung in Abbildung 26 oft in ähnliche Gruppen zusammengefasst, jedoch bildet die Spektralpartitionierung nur 6 Gruppen, während die hierarchische Partitionierung in 8 Clustern resultiert.

Das Problem bei den vorgestellten Methoden zur Spektralpartition ist, dass eine Eigenwertzerlegung für große Graphen relativ aufwändig ist. Der Rechenaufwand verringert sich jedoch für Graphen mit wenigen Kanten und auch bei einer Bisektion, wenn der Abstand zwischen zweitgrößtem und drittgrößtem Eigenwert groß ist.

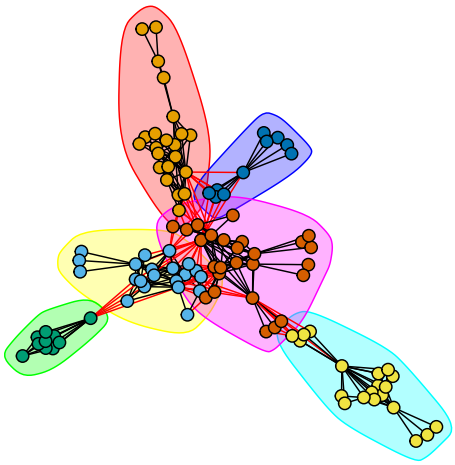

Abbildung 27: Partitionierung von Komponente 7 durch Spektralpartitionierung mithilfe der Optimierung der Modularität

4.4 Assortativity & Mixing

Im vorangegangen Kapitel wurde nach einer Clusterstruktur von ähnlichen Knoten im Netzwerkgraphen gesucht. Oft haben ähnliche Knoten auch ähnliche Eigenschaften oder Attribute, die jedoch latent sind. Graphenpartitionen sind in solchen Fällen daher sehr nützlich, um die Knoten trotzdem Clustern zuordnen zu können.

Sind diese Knoteneigenschaften nicht latent, sondern bekannt, kann man mit einigen Kennzahlen quantifizieren, inwiefern eine Eigenschaft auf die Partition einen Einfluss hat. Haben diese Eigenschaften einen Einfluss auf die Vernetzung zwischen den Knoten, so spricht man von **assortativem Mixing**. Um nun den Einfluss dieser Eigenschaften zu quantifizieren, kann man diverse Assortativitäts-Koeffizienten berechnen. Einige dieser Maßzahlen, die im Grunde auf dem Konzept der Korrelation basieren, sollen nun vorgestellt werden.

Das Knoten-Merkmal, das für die Assortativität betrachtet wird, kann sowohl kategorial, ordinal als auch metrisch sein. Angenommen, das betrachtete Merkmal sei kategorial und

hat M verschiedene Kategorien. Für jeden Knoten im Graphen sei die Merkmalsausprägung, also die Kategorie, bekannt. Sei $\mathbf{F}$ eine Matrix, deren Einträge f_{ij} der relative Anteil an Kanten im Graphen ist, die Knoten der i-ten Kategorie mit Knoten der j-ten Kategorie verbinden. f_{i+} sei dann die i-te Zeilensumme und f_{+j} die j-te Spaltensumme. Dann ist der **Assortativititäts-Koeffizient** r_a definiert als

$$r_a = \frac{\sum_i f_{ii} - \sum_i f_{i+} f_{+i}}{1 - \sum_i f_{i+} f_{+i}}. \tag{29}$$

Wenn der Anteil der Verbindungen innerhalb einer Kategorie sich nicht von dem erwarteten Wert der Verbindungen bei zufälliger Kantenanordnung unterscheidet, so ist der Zähler und damit r_a Null. Analog ist der Koeffizient gleich 1, wenn nur Knoten derselben Kategorie miteinander verbunden sind, da dann $\sum_i f_{ii} = 1$ gilt. Wenn die Kantenanordnung perfekt disassortativ ist, also nur Kanten zwischen Knoten unterschiedlicher Kategorie bestehen und $\sum_i f_{ii} = 0$ ist, erreicht der Koeffizient seinen minimalen Wert von

$$r_a^{\min} = -\frac{\sum_i f_{i+} f_{+i}}{1 - \sum_i f_{i+} f_{+i}}. \tag{30}$$

Der Wertebereich des Koeffizienten ist also das Intervall $(-1, 1]$, wobei darauf hingewiesen wird, dass man die untere Grenze von -1 selbst bei perfekten dissortativem Verhalten der Kanten nicht erreicht.

Liegt statt eines kategorialen ein ordinales oder metrisches Merkmal vor, wird ein anderer Assortativititäts-Koeffizient benutzt. Seien (x_e, y_e) die Merkmalsausprägungen der Knoten, die durch eine Kante $e \in E$ verbunden sind. Um die Assortativität im Graphen zu beschreiben, wird nun der **Pearson-Korrelationskoeffizient** des Paares (x_e, y_e)

$$r = \frac{\sum_{x,y} xy(f_{xy} - f_{x+} f_{+y})}{\sigma_x \sigma_y} \tag{31}$$

benutzt. Die Summe wird über alle beobachteten Merkmalsausprägungskombinationen (x, y) gebildet und f_{xy}, f_{x+}, f_{+y} sind analog wie im kategorialen Fall definiert. σ_x und σ_y entsprechen den Standardabweichungen der Verteilungen der Häufigkeiten $\{f_{x+}\}$ und $\{f_{y+}\}$.

Diese Methoden lassen sich nur auf Netzwerke anwenden, bei denen weitere Informationen zu Eigenschaften und Attributen der Akteure in dem Netzwerk vorliegen. Im Patentnetz-

werk ist das nicht der Fall. Würden weitere Informationen jedoch vorliegen, wäre es auch hier von Interesse, die Clusterstruktur vor dem Hintergrund der Fachdisziplin, der Nationalität, des Alters oder Geschlechts zu analysieren.

5 Zusammenfassung

Die vorgestellten Methoden zur deskriptiven Analyse von Netzwerken fokussieren sich auf die Eigenschaften der Kanten und Knoten eines Netzwerks und die Beschreibung des Zusammenhangs eines Netzwerks. Dabei sind nicht immer alle Methoden für jedes Netzwerk sinnvoll oder anwendbar. Ergebnisse sollten daher immer vor dem Hintergrund interpretiert werden, was das Netzwerk überhaupt darstellt. Insbesondere für große Netzwerkgraphen spielt zudem die Effizienz von Algorithmen und damit deren Rechenzeit eine große Rolle.

Für den Patentdatensatz bewerten in den sieben größten Komponenten alle vier vorgestellten Zentralitätsmaße denselben Knoten im Netzwerk als am "wichtigsten" für den Graphen, für den gesamten Graphen ist das jedoch nicht der Fall. Insgesamt ist der Grad der Kohäsion des Netzwerks eher gering, was auch daran liegt, dass das Netzwerk unverbunden ist und aus mehreren Komponenten besteht. Insbesondere entfallen sehr viele Knoten auf sehr kleine Subkomponenten mit weniger als 5 Knoten und es liegt keine giant component vor, auf die sich eine weitere Analyse und die Modellerstellung konzentrieren könnte.

Literatur

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology 2*(1), 113–120.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology 25*(2), 163–177.

Brandes, U. and T. Erlebach (2005). *Network Analysis - Methodological Foundations*. Springer.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik 1*(1), 269–271.

Drees, H., L. de Haan, and S. Resnick (2000). How to make a hill plot. *The Annals of Statistics 102*(41), 14497–14502.

Everett, M. and S. Borgatti (1999). The centrality of groups and classes. *Journal of Mathematical Sociology 23*, 181–202.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal 23*(98), 298–305.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry 40*(1), 35–41.

Fronczak, A., J. M. Holyst, J., and J. Sienkiewicz (2002). Higher order clustering coefficients in barabási-albert networks. *Physica A 316*(1-4), 688–694.

Girvan, M. and M. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences 99*(12).

Gkantsidis, C., M. Mihail, and E. Zegura (2003). Spectral analysis of interenet topologies. *Proceedings of the 22nd Annual INFOCOM Conference*, 364–374.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika 18*(1), 39–43.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data - Methods and Models*. Springer.

Kolaczyk, E. D. and G. Csàrdi (2014). *Statistical Analysis of Network Data with R*. Springer.

Milgram, S. (1967). The small world problem. *Psychology Today 2*(1), 60–67.

Newman, M. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review 74*(3), 036104.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

R Development Core Team (2014). R: eine freie Programmiersprache für statistisches Rechnen und statistische Grafiken.

Turán, P. (1941). On an extremal problem in graph theory. *Matematikai és Fizikai Lapok 48*, 436–452.

Watts, D. and S. Strogatz (1998). Collective dynamics of 'small world' networks. *Nature 393*(6684), 440–442.

A Appendix

A.1 Eigenwerttheorie

Für den Kontext dieser Arbeit ist nur die Eigenwerttheorie für endliche Vektorräume relevant. Zudem werden Eigenwerte nur in $\mathbb{R}$ betrachtet. Gegeben sei also ein Vektorraum V mit $dim(\mathbf{V}) = n \in \mathbb{N}$. Dann lässt sich jeder Endomorphismus $f : V \to V$ durch eine $n \times n$-Matrix darstellen. Gilt für ein $\lambda \in \mathbb{R}$ und einen Nicht-Nullvektor $x \in \mathbb{R}^n$ die Gleichung

$$\mathbf{A}x = \lambda x, \tag{32}$$

so nennt man λ einen reellen **Eigenwert** der Matrix $\mathbf{A}$. Der Vektor $x \neq 0$ ist dann der dazugehörige **Eigenvektor**.

Zur Berechnung der Eigenwerte und Eigenvektoren wird die Gleichung (32) zu

$$\mathbf{A}x - \lambda\mathbf{E}x = 0 \tag{33}$$

umgeschrieben, wobei $\mathbf{E}$ die n-dimensionale Einheitsmatrix ist. Ein Ausklammern des Vektors x liefert

$$(\mathbf{A} - \lambda\mathbf{E})x = 0. \tag{34}$$

Wegen $x \neq 0$, ist dieses Gleichungssystem genau dann lösbar, wenn

$$\det(\mathbf{A} - \lambda\mathbf{E}) = 0. \tag{35}$$

Diese Determinante ist ein Polynom n-ten Grades in λ und wird auch **charakteristisches Polynom** genannt. Dessen Nullstellen

$$\lambda^n + \alpha_{n-1}\lambda^{n-1} + \ldots + \alpha_1\lambda + \alpha_0 = 0 \tag{36}$$

in $\mathbb{R}$ sind die reellen Eigenwerte der Matrix. Die dazugehörigen Eigenvektoren x berechnet man dann durch Einsetzen der Eigenwerte in Gleichung (34) und Lösen des homogenen linearen Gleichungssystems. Eigenwerte sind nicht immer einfach, sondern es können auch mehrfache Nullstellen im charakteristischen Polynom vorkommen. Die Häufigkeit eines Eigenwerts wird als **algebraische Vielfachheit** bezeichnet. Die Anzahl der linear unabhängigen Eigenvektoren zu einem Eigenwert wird als **geometrische Vielfachheit** des

Eigenwertes bezeichnet.

A.2 Algorithmen

A.2.1 Dijkstra

Der Dijkstra-Algorithmus wurde 1959 vom niederländischen Informatiker Edsger Dijkstra veröffentlicht, siehe Dijkstra (1959). Es handelt sich hierbei um einen Algorithmus zum Finden des kürzesten Pfades von einem gegebenen Startknoten zu allen anderen Knoten eines Netzwerks oder aber zu einem einzelnen vorher spezifizierten Knoten. Die Rechenzeit ist mit $\mathcal{O}(n^2)$ quadratisch. Ein prominentes Anwendungsbeispiel für den Algorithmus sind Routenplaner, deren Ziel es ist, einen möglichst kurzen Weg von einem Ort zu einem anderen zu finden.

Dabei geht der Algorithmus für einen vorgegebenen Startknoten s und Zielknoten z wie folgt vor:

1. Zunächst wird jedem Knoten ein provisorischer Distanzwert zugewiesen. Dem Startknoten wird dabei der Wert Null und jedem anderen Knoten der Distanzwert ∞ zugewiesen.

2. Der Startknoten s wird nun als der momentan betrachtete Knoten u gesehen und alle anderen Knoten werden zu der Menge U der Knoten, die noch nicht überprüft wurden, zusammengefasst.

3. Die Distanz zu allen Nachbarknoten v des momentan betrachteten Knotens u, die noch nicht durchschritten wurden, wird berechnet. Sie wird zu der Distanz von u zu s dazuaddiert. Hat der Nachbarknoten v zuvor einen Wert größer als den nun berechneten (also zuvor z.B. ∞) gehabt, wird dieser zuvorige Wert durch den neuen Wert ersetzt.

4. Nachdem Schritt 3 für alle Nachbarknoten von u durchgeführt wurde, wird der momentan betrachtete Knoten u aus U entfernt.

5. Liegt der Zielknoten z nun nicht mehr in der Menge der unüberprüften Knoten U, so ist der Algorithmus beendet. Ist die kleinste Distanz der unüberprüften Knoten ∞, so ist der Algorithmus auch beendet, da dann die unüberprüften Knoten nicht von dem Startknoten s aus erreichbar sind.

6. Trifft keine der beiden Bedingungen in Schritt 5 zu, so wird nun der Knoten mit der geringsten Distanz zum Startwert s ausgewählt und mit Schritt 3 weitergemacht.

A.2.2 Brandes

Um die Intermediatätszentralität der Knoten eines Netzwerks zu berechnen, wird meist der 2001 von Ulrik Brandes entwickelte Brandes-Algorithmus verwendet, siehe Brandes (2001). Wie in 3.2 beschrieben, ist die Intermediationszentralität eines der gängigsten Zentralitätsmaße. Zur Berechnung für einen Knoten $u \in V$ wird für alle Knotenpaare $s, t \in V$ der Anteil der kürzesten Pfade zwischen s und t berechnet, die durch u gehen. Die Summe dieser Anteile über alle Knoten in dem Graphen wird dann als Intermediationszentralität $c_B(u)$ bezeichnet.

Für ungewichtete Graphen benötigt der Algorithmus eine Rechenzeit von $\mathcal{O}(N_v N_e)$ und für gewichtete Graphen $\mathcal{O}(N_v N_e + N_v^2 log N_v)$, verglichen mit $\mathcal{O}(N_v^3)$ Rechenzeit für ungewichtete Graphen bei einer direkten Berechnung. Im Folgenden soll das Prinzip für ungewichtete Digraphen dargestellt werden, wie es auch in der Veröffentlichung von 2001 der Fall war.

Für den Algorithmus müssen zunächst einige weitere Kennzahlen definiert werden. Sei $G = (V, E)$ ein Graph und $s, t \in V$ ein fixes Knotenpaar. $\sigma(s, t)$ sei die Anzahl an kürzesten Pfaden zwischen s und t. $\sigma(s, t|v)$ ist dann die Anzahl an kürzesten Pfaden, die durch $v \in V$ gehen. Als Dependency eines Startknoten s auf einen Knoten v wird dann

$$\delta_s(v) = \sum_{t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)} \tag{37}$$

definiert. Hierfür werden für alle Knoten $t \in V$ die Anteile der kürzesten Pfade von s nach t, die durch v gehen, aufsummiert. Die Intermediationszentralität von v kann dann als die Summe über die Dependencies aller Knoten $s \in V, s \neq v$ dargestellt werden:

$$c_B(v) = \sum_{s \neq v \in V} \delta_s(v). \tag{38}$$

Die essentielle Umformung, auf der der Algorithmus basiert, ist dann, dass man die De-

pendency (37) eines Startknoten s auf einen Knoten v auch darstellen kann als

$$\delta_s(v) = \sum_{w:v \in P(s,w)} \frac{\sigma(s,v)}{\sigma(s,w)} \Big(1 + \delta_s(w)\Big), \tag{39}$$

wobei $P(s,w)$ die Menge aller Vorfahren von w auf dem kürzesten Weg von s zu w sind.

Der Algorithmus von Brandes funktioniert dann wie folgt:

1. Für jeden Knoten $s \in V$ werden die kürzeste Pfade berechnet. Bei einem ungewichteten Graphen ist das äquivalent zu einer BFS.

2. Bei der Berechnung in Schritt 1 werden auch die Menge der Vorfahren $P(s,v)$ und die Anzahl der kürzesten Pfade $\sigma(s,v)$ bestimmt.

3. Für jeden möglichen Startknoten werden mithilfe der Menge der Vorfahren und der Anzahl der kürzesten Pfade die Dependencies für alle anderen Knoten $v \in V$ berechnet.

4. Um nun die Intermediationszentralität eines Knotens $v \in V$ zu berechnen, werden alle Dependencies für alle Startknoten s aufsummiert.

Abbildungsverzeichnis

Tabellenverzeichnis

Inhalt der CD

Die beigelegte CD-ROM enthält folgende Dateien:

- Die **Bachelorarbeit** als PDF-Datei
- Den Ordner **Grafiken**, in dem alle in der Bachelorarbeit enthaltenen Grafiken im PDF-Format enthalten sind
- Den **Patentdatensatz** inventornet.dta
- Die folgenden **R-Skripte**:
    1. Datenerstellung.R liest den Patentdatensatz ein
    2. Gradverteilung.R enthält alle Analysen zu 3.1
    3. Zentralitaet.R enthält alle Analysen zu 3.2
    4. Kohaesion.R enthält alle Analysen zu 4

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

München, den 29.03.2016

Elisabeth Krätzschmar