

Universität Konstanz  
Seminar Netzwerkanalyse  
SS 2006  
Dozent: Prof. Dr. Ulrik Brandes  
Betreuer: Christian Pich

## **The Link Prediction Problem**

[Basierend auf: "The Link Prediction Problem for Social Networks" von  
David Liben-Nowell und John Kleinberg; 8 Januar 2004]

Sebastian Faller  
Matrikelnummer: 496721  
faller@inf.uni-konstanz.de

# Inhaltsverzeichnis

<b>1</b>	<b>Motivation und Anwendungsgebiete</b>	<b>2</b>
<b>2</b>	<b>Grundlegendes Konzept</b>	<b>3</b>
<b>3</b>	<b>Algorithmen</b>	<b>5</b>
3.1	Gemeinsame Nachbarn . . . . .	5
3.1.1	Common Neighbours . . . . .	5
3.1.2	Jaccard-Koeffizient . . . . .	5
3.1.3	Adamic/Adar . . . . .	6
3.1.4	Preferential Attachment . . . . .	6
3.2	Kürzeste Wege . . . . .	7
3.2.1	Katz . . . . .	7
3.2.2	Hitting Time, Page Rank, und Variationen . . . . .	7
3.2.3	Sim Rank . . . . .	9
3.3	Höher wertige Methoden - "Meta Methoden" . . . . .	9
3.3.1	Schwach eingestufte Aproximation . . . . .	9
3.3.2	Unseen Bigrams . . . . .	10
3.3.3	Clustering . . . . .	10
<b>4</b>	<b>Ergebnisse und Ausblicke</b>	<b>11</b>

# 1 Motivation und Anwendungsgebiete

Soziale Netzwerke sind sehr dynamische und schnell wachsende Netzwerke. Und gerade dieses hohe Maß an Veränderung das dadurch entsteht ist für die Wissenschaft so interessant. Denn es wirft die Frage nach dem Mechanismus auf, der hinter dieser Dynamik und dem Wachstum liegt. Das heißt nach welchen Regeln sich das Netzwerk verändert, entwickelt und ob man diese Entwicklung vorhersagen, berechnen kann. Und das ist genau der Ansatz des Zugrunde liegenden Artikels von David Liben-Nowell und John Kleinberg.

Sie versuchen eben nun Formeln und Gesetzmäßigkeiten zu finden, die diese Entwicklungen vorhersagen. Also wann und zwischen welchen Knoten eine neue Kante entsteht, die für sie Interaktion, Kollaboration oder Einfluss zwischen zwei Entitäten bedeutet.

Als Beispiel für ihre gesamte Arbeit ziehen sie immer die Frage nach Kollaboration zwischen zwei Autoren die möglicherweise zusammen einen Artikel schreiben, dass dann im Falle dass es eintritt als Kante symbolisiert wird. Ihre Motivation ist also vorher zusagen welche zwei Autoren eines Abgegrenzten Bereichs, hier aus dem Bereich Physik, zusammen in nächster Zeit ein Paper zusammen schreiben werden. Sie wollen also ein Modell entwickeln, mit dem sie genau diese Fragestellung beantworten können und so die Entwicklung eines Sozialen Netzwerks abbilden können.

Laut ihrer Aussage gibt es für die Erstellung eines solchen Modells genug Hinweise in der Topologie eines Netzwerks, Beispiele sind gemeinsame Bekannte oder das Verkehren in den gleichen Kreisen, weil man zu einer bestimmten Gruppe gehört oder bestimmte Interessen hat. Was allerdings laut den Autoren nicht in die Vorhersage mit eingerechnet werden kann sind Faktoren, wie der dass zwei Personen aus welchen Gründen auch immer plötzlich geographisch näher zu einander stehen als vorher und auf Grund dessen gemeinsam ein Paper schreiben. Diese Faktoren sind wie das genannte Beispiel nur schwer in die Topologie eines Graphen zu integrieren.

Anwendungsgebiete für ihren Ansatz finden sich laut Kleinberg und Liben-Nowell in der Forschung im Bereich Künstliche Intelligenz und data mining, sowie in der Sicherheitsforschung und der Überwachung terroristischer Netzwerke.

Zu Beginn dieser Arbeit wird das Link Prediction Problem und dessen grundlegenden Prinzipien besprochen. Anschließend wird das Kernstück des Problems die Algorithmen zur Vorhersage vorgestellt, an die sich zusätzliche Methoden zur Verbesserung der Leistung der Algorithmen anschließen. Abschließend werden die Ergebnisse präsentiert und kritisch besprochen. Danach folgt noch ein kleiner Ausblick auf weitere Möglichkeiten, aber auch noch bestehender Arbeit, an den Ideen.

## 2 Grundlegendes Konzept

Nachdem wir nun einen groben Überblick über die Thematik und die Anwendungsmöglichkeiten bekommen haben wollen wir uns den dahinter liegenden theoretischen Grundlagen zuwenden, die genauen Algorithmen die zur Anwendung kommen und den Kern der Theorie darstellen werden später im Detail besprochen. Vorerst soll die Grundlage und die Definition des "Link Prediction Problems" wie es Kleinberg und Liben-Nowell sehen erläutert werden.

Betrachtet wird ein dynamischer Graphen  $G\{V, E\}$  zu bestimmten Zeitintervallen, für die gilt,

$$t_0 < t'_0 < t_1 < t'_1$$

. Sie betrachten somit also immer zwei Momentaufnahmen des selben Graphen, wobei es sich bei der einen Momentaufnahme um das so genannte Trainingsintervall,  $G[t_0, t'_0]$ , handelt, auf dem die später besprochenen Algorithmen angewendet werden, und dem Testintervall,  $G[t_1, t'_1]$ , das zur Kontrolle und Bewertung der Ergebnisse der Vorhersage auf dem Testintervall herangezogen wird.

Das heißt, sie nehmen das Intervall  $G[t_0, t'_0]$  und tun so als ob dies der aktuelle Zustand des Graphen sei und versuchen dann unter Anwendung der, noch zu besprechenden, Algorithmen auf  $G[t_0, t'_0]$  eine Vorhersage der neu eingefügten Kanten zwischen zwei Knoten des Graphen zum Zeitpunkt

$$[t_1, t'_1], e = \{u, v\} \in G[t_1, t'_1] \wedge e = \{u, v\} \notin G[t_0, t'_0]$$

, wobei gilt,

$$u, v \in G[t_0, t'_0] \wedge u, v \in G[t_1, t'_1],$$

zu machen. Anschließend ziehen sie dann das Testintervall  $G[t_1, t'_1]$  heran um zu überprüfen, ob ihre vorhergesagten Kanten auch wirklich entstanden sind und berechnen dann daraus, anhand der richtig vorausgesagten Kanten die Genauigkeit des Algorithmus, angewandt auf diesen Graphen.

Als Ausgabe der Algorithmen, angewendet auf  $G[t_0, t'_0]$ , erhält man eine einfache sortierte Liste von Kanten

$$\begin{aligned} L_p &:= A \times A - E_{old}^1 \\ E_{old} &:= \{\forall e \mid e \in G[t_0, t'_0]\} \end{aligned}$$

mit bestimmten Wahrscheinlichkeiten, nach der die Liste auch absteigend sortiert ist.

Diese Liste repräsentiert neu entstehende Kanten, die mit einer bestimmten Wahrscheinlichkeit, geliefert vom verwendeten Algorithmus, in  $G[t_1, t'_1]$

---

<sup>1</sup>A steht für die Autoren und entspricht somit den Knoten  $V$  des Graphen, d.h.  $L_p$  könnte auch folgendermaßen definiert sein:  $L_p := V \times V - E_{old}$

gebildet werden könnten. Allerdings enthält diese Liste alle Paare von Knoten und ihre Wahrscheinlichkeit eine Kante auszubilden, auch wenn diese 0 sein sollte. Da das aber keine wirkliche Voraussage ist, ist es notwendig einen Schätzer einzuführen, anhand dessen man abgrenzt welche Kanten in die Voraussage aufgenommen werden und welche nicht.

In diesem Fall bedienen sich Kleinberg und Liben-Nowell eines Tricks. Sie gehen einfach hin und nehmen alle  $n$  ersten Kanten der Liste in die Vorhersage auf, wobei

$$\begin{aligned} n &:= |E_{new}^*| \\ E_{new}^* &:= E_{new} \cap (Core \times Core) \\ Core &:= \{Core \subset A^1 \mid \# \text{geschriebene Artikel} \geq 3 \text{ in } G[t_0, t'_0] \wedge G[t_1, t'_1]\} \\ E_{new} &:= \{\forall e \mid e \in G[t_1, t'_1] \wedge e \notin G[t_0, t'_0]\} \end{aligned}$$

der Anzahl Kanten entspricht, die wie sie aus dem Testintervall wissen, neu entstehen, egal wie hoch oder niedrig die ersten  $n$  Wahrscheinlichkeitswerte in der Liste sind.

Dieses Vorgehen wirft allerdings einige Fragen und Probleme auf. Denn im eigentlichen Anwendungsfall möchte man ja eine Vorhersage machen bei der das Ergebnis noch nicht bekannt ist, das heißt man die Zahl neu entstehender Kanten, im vorher zu sagenden Graphen  $G[t_1, t'_1]$ , nicht kennt, und man diese Information somit auch nicht als Schätzer verwenden kann. Außerdem spielen bei dieser Vorgehensweise die Wahrscheinlichkeiten, wie vorhin schon kurz angesprochen, nur eine untergeordnete Rolle und sorgen im Prinzip nur für die absteigende Sortierung der Liste.

Diese entsteht nun aber auch, wenn man nur ganz geringe Wahrscheinlichkeiten der Knotenpaare eine Kante auszubilden vom Algorithmus erhält. Im Umkehrschluss bedeutet das, dass die einzelne Vorhersage einer Kante zwischen zwei Knoten auf die eigentliche Vorhersage, aller neu entstehenden Kanten, keinen Einfluss hat.

Das wäre aber das was man von einem Maß erwarten würde, das es eine Aussage über den gewünschten Sachverhalt macht, in diesem Fall eine Vorhersage der Kanten ermöglicht. Demnach wäre es wohl sinnvoller einen Schätzer zu verwenden, der erstens nicht abhängig von der Kenntnis des Ergebnisses ist und zweitens die einzelnen Wahrscheinlichkeiten der Knotenpaare eine Kante auszubilden stärker beziehungsweise besser berücksichtigt. Denkbar wäre ein Grenzwert, der darüber entscheidet bis zu welcher Wahrscheinlichkeit Kanten in die Vorhersage aufgenommen werden. Allerdings müsste dieser Grenzwert empirisch oder mathematisch belegt werden um von dieser doch sehr "willkürlichen" Methode weg zu kommen.

---

<sup>1</sup>A steht für die Autoren und entspricht somit den Knoten  $V$  des Graphen

### 3 Algorithmen

Bei den von Kleinberg und Liben-Nowell verwendeten Algorithmen kann man zwei Kategorien unterscheiden. Und zwar die, die die Wahrscheinlichkeit, dass zwei Knoten eine Kante bilden, über die gemeinsamen Nachbarn der beiden Knoten berechnet und die, die die kürzesten Wege zwischen den beiden Knoten zur Berechnung hernehmen.

Es sei für einen Knoten  $x$ ,  $\Gamma(x)$  die Menge der Nachbarn von  $x$  im Graphen  $G\{V, E\}$ .

#### 3.1 Gemeinsame Nachbarn

##### 3.1.1 Common Neighbours

$$score(x, y) := |\Gamma(x) \cap \Gamma(y)|$$

Dieser Algorithmus verwendet einen Ansatz der sehr intuitiv ist, er macht nämlich nichts anderes als die Übereinstimmung der Nachbarn der beiden Knoten auf gemeinsame Nachbarn zu überprüfen. Und je höher diese Übereinstimmung in den gemeinsamen Nachbarn ist, um so größer ist die Wahrscheinlichkeit, dass die beiden Knoten eine neue Kante ausbilden. Das bedeutet im Prinzip nichts anderes als dass zum Beispiel zwei Wissenschaftler, die einen großen gemeinsamen Bekanntenkreis haben mit einer höheren Wahrscheinlichkeit zusammenarbeiten als zwei Wissenschaftler, die einen kleinen oder gar keinen gemeinsamen Bekanntenkreis haben.

Dementsprechend könnte man auch schreiben:

$$score(x, y) := |\Gamma(x) \cap \Gamma(y)| = \# \text{ gemeinsame Nachbarn}$$

##### 3.1.2 Jaccard-Koeffizient

$$score(x, y) := \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

Jaccard's coefficient erweitert das Prinzip aus dem obigen Algorithmus, Common Neighbours, indem er das Ergebnis davon durch die Anzahl aller Nachbarn der beiden Knoten gewichtet. Wenn man sich das klar gemacht hat folgt daraus, dass es sich bei diesem Algorithmus eigentlich nur um eine Variante der Abzählregel von Laplace handelt, Anzahl günstiger Fälle, in diesem Fall die Anzahl gemeinsamer Nachbarn, durch die Anzahl aller möglichen Fälle, hier die gesamte Menge Nachbarn die die beiden Knoten haben.

### 3.1.3 Adamic/Adar

$$score(x, y) := \sum_{z \in |\Gamma(x) \cap \Gamma(y)|} \frac{1}{\log|\Gamma(z)|}$$

Adamic/Adar gehen nun sogar noch einen Schritt weiter, sie machen ihr Maß nicht nur von den direkten Nachbarn abhängig, sondern betrachten auch noch wie wichtig beziehungsweise zentral diese gemeinsamen Nachbarn der beiden betrachteten Knoten sind. Für ihr Maß bedeutet das, je zentraler ein gemeinsamer Nachbar ist, das heißt je mehr Nachbarn dieser hat, um so geringer wird dieser gewichtet. Hier sollen also die gemeinsamen Nachbarn entscheidend sein die eher seltene Nachbarn sind. Grund dafür ist, dass man vermeiden möchte, dass plötzlich jeder mit jedem ein Paper schreiben könnte, nur weil beide eine Zentrale Figur ihres Themengebietes kennen, die bekanntermaßen viele Verbindungen zu anderen Personen hat aber im Prinzip nichts über die Nähe der einzelnen Personen untereinander hat. Vergleichbar ist die Problematik mit der Indexierung von Texten, bei denen man auch die Wörter die zu häufig vorkommen nicht verwendet, da diese nichts über den Inhalt des Textes aussagen. Der Logarithmus in der Formel von Adamic/Adar misst dann nun noch die Größenordnung.

### 3.1.4 Preferential Attachment

$$score(x, y) := |\Gamma(x)| \cdot |\Gamma(y)|$$

Das Preferential Attachment basiert auf der einfachen Grundlage wer hat dem wird gegeben. Also je mehr Nachbarn  $x$  und/oder  $y$  haben um so größer ist die Wahrscheinlichkeit dass sie eine Kante ausbilden. Der Grund für die Aussage ist der, dass man davon ausgeht je mehr Nachbarn ein Knoten hat um so wahrscheinlicher existiert ein Weg von diesem Knoten zu einem bestimmten anderen Knoten den man betrachtet. Wenn dieser andere Knoten dann selbst noch viele Nachbarn hat steigt die Wahrscheinlichkeit eines Weges, und besonders der eines kurzen Weges noch mehr. Wie von Kleinberg und Liben-Nowell schon erwähnt haben dies Barabasi et al. und Newman in ihren Studien herausgefunden. Wobei noch dazu zu sagen wäre, dass das Produkt als Maß nur die Approximation eines Integrals ist, dass die Funktion exakt bestimmen würde. Und Barabasi et al. formulieren diese Approximation auch mehr als Annahme aus ihren Untersuchungen, denn als bewiesene Behauptung.[2]

## 3.2 Kürzeste Wege

### 3.2.1 Katz

$$score(x, y) := \sum_{l=1}^{\infty} \beta^l \cdot |Patch_{x,y}^{<l>}|$$

Der Katz Algorithmus ist nun einer der Algorithmen, der das Prinzip der kürzesten Wege verwendet. Und er tut dies, in dem er über die Menge der Pfade von einem zum anderen Knoten summiert. Und je mehr Pfade es zwischen den beiden Knoten gibt, um so größer bewertet er die Wahrscheinlichkeit, dass zwischen den beiden Knoten eine neue Kante entsteht. Um kurze Pfade stärker zu gewichten, denn diese sagen mehr über die "nähe" von zwei Knoten aus, als Lange, dämpft Katz alle Pfade um den Faktor  $\beta^l$ . Denn je länger die Pfade werden um so kleiner wird dieser Faktor. Zum  $\beta$  selbst und der Wahl des Wertes für  $\beta$  sagen Liben-Nowell und Kleinberg nicht viel, nur, dass die Wahl eines sehr kleinen  $\beta$  zu sehr ähnlichen Vorhersagen wie beim common neighbours Algorithmus. Was auch verständlich ist, denn je kleiner  $\beta$  zu Beginn ist führen schon geringe Pfadlängen zu einer "starken Dämpfung", das heißt das im extrem Fall nur Pfade stark in das Maß einfließen, die direkte Nachbarn beider Knoten sind, und damit hat man wieder näherungsweise die Definition des Maßes wie wir es schon aus common neighbours kennen.

Kleinberg und Liben-Nowell unterscheiden noch zwei Varianten dieses Algorithmus. Die ungerichtete Variante, bei der es keine Rolle spielt, wie oft zwei Knoten, respektive Personen, miteinander Kollaborieren, das heißt es entsteht eine Kante bei Kollaboration, aber keine weitere sollten die Personen nochmals miteinander Kollaborieren. Und die gewichtete Variante, die mehrfache Kollaboration berücksichtigt, indem sie die Kante entsprechend der Häufigkeit der Kollaboration gewichtet.

### 3.2.2 Hitting Time, Page Rank, und Variationen

$$\begin{aligned} score(x, y) &:= -H_{x,y} \\ score(x, y) &:= -C_{x,y} = -(H_{x,y} + H_{y,x}) = -H_{x,y} - H_{y,x} \end{aligned}$$

Hitting time dagegen, geht einen anderen Weg, dieser Algorithmus basiert auf dem Prinzip eines Zufallslaufs auf dem Graphen, beginnend bei einem Knoten  $x$ , der solange zufällig zu Nachbarn des jeweiligen Knoten bei dem er sich befindet, bis er seinen Zielknoten  $y$  erreicht hat. Die Auswahl des jeweiligen Nachbarn zu dem er als nächstes springt, ist für alle Nachbarn des aktuellen Knotens gleich Wahrscheinlich.

$$H_{x,y},$$

die hitting time, ist nun die erwartete Anzahl von Schritten die der Zufallslauf von  $x$  nach  $y$  benötigt. Da die hitting time, also der Zufallslauf von  $x$  nach  $y$  nicht zwingend symmetrisch ist, das heißt, dass ein Zufallslauf von  $y$  nach  $x$  eine andere Anzahl Schritte haben kann, definieren Kleinberg und Liben-Nowell ein weiteres Maß, die so genannte commute time,

$$C_{x,y} = H_{x,y} + H_{y,x},$$

die genau dieser Tatsache Rechnung trägt. Wie man in der Formel vom Anfang schon sieht werden die beiden Werte, hitting time und commute time, negiert um als Maß verwendet zu werden. Warum dies geschieht wird nicht näher ausgeführt.

Ein Problem hat die obige Definition allerdings, was Kleinberg und Liben-Nowell auch ansprechen, die hitting time  $H_{x,y}$  bleibt auch dann relative klein, wenn der Zielknoten  $y$  eine hohe stationäre Wahrscheinlichkeit hat. Das heißt, selbst wenn  $y$  von vielen anderen Knoten aus erreichbar ist, also viele mittelbare und unmittelbare Nachbarn hat, bleibt die hitting time relative klein. Deshalb schlagen die Autoren eine normalisierte Variante ihrer obigen Definition vor, die genau dieses Problem ausgleichen soll.

$$\begin{aligned} \text{score}(x, y) &:= -H_{x,y} \cdot \pi_y \\ \text{score}(x, y) &:= -(H_{x,y} \cdot \pi_y + H_{y,x} \cdot \pi_x) \end{aligned}$$

Ein weiteres Problem stellt die Tatsache dar, dass fast jeder Knoten mit jedem anderen Knoten in einem Graphen verbunden werden kann, solange die Pfadlänge nur lang genug ist. Das heißt es besteht die Möglichkeit, zwei Knoten eines Graphen miteinander zu verbinden, obwohl diese eigentlich nichts miteinander zu tun haben. Dies geschieht sehr leicht, wenn es im Graphen einen, oder mehrere „dominierende“ Knoten gibt, die sehr viele eingehende Kanten besitzen, weil sie sehr Zentral sind.

Um es mit dem ganz zu Anfang eingeführten Anwendungsbeispiel auszudrücken, es gibt in jedem Bereich, sei es nun die Physik oder ein anderer, Personen, die sehr dominant sind und sehr viele Artikel, mit den unterschiedlichsten Leuten geschrieben haben, da sie führend in ihrem Bereich sind. Das führt dann dazu, dass diese Personen in einem Graphen Personen über eine Unmenge anderer Personen miteinander verbinden können, wobei die Wahrscheinlichkeit dass diese Personen miteinander einen Artikel schreiben werden sehr gering bleibt, da sie sich nicht wirklich kennen.

Somit sagen Verbindungen mit sehr langen Pfadlängen nicht wirklich etwas über die Wahrscheinlichkeit aus, dass zwei Personen miteinander einen Artikel schreiben. Um diesem Phänomen Herr zu werden, wird eine so genannte Reset-Wahrscheinlichkeit  $\alpha$  eingeführt, die jedes mal mit berücksichtigt wird, wenn der Zufallslauf einen neuen Nachbarn eines Knoten besuchen möchte. Das bedeutet, vor jeder Entscheidung welchen Nachbarn des aktuellen Knoten der Zufallslauf besucht wird vorher über  $\alpha$  entschieden, ob der

Zufallslauf weiter geht, oder zum Ausgangsknoten zurück kehrt und einen neuen Durchlauf startet. Laut Liben-Nowell und Kleinberg verhindert dieser Zusatz in den meisten Fällen, dass weit entfernte Teile des Graphen untersucht werden.

### 3.2.3 Sim Rank

$$score(x, y) := \begin{cases} 1, & x = y \\ \gamma \cdot \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} similarity(a, b)}{|\Gamma(x)| \cdot |\Gamma(y)|}, & sonst \end{cases}$$

Sim Rank geht einen etwas anderen Weg, es berechnet als Maß für die Vorhersage die Ähnlichkeit der beiden Knoten des betrachteten Knotenpaares  $x$  und  $y$ . Und dies geschieht, wie man in der oben dargestellten Formel sehen kann, über eine rekursive Definition, denn um die Ähnlichkeit von  $x$  und  $y$  zu berechnen, berechnet der Algorithmus jeweils erst die Ähnlichkeit der Nachbarn von  $x$  und  $y$  und gewichtet diesen Wert durch die Division mit dem Preferential Attachment. Der dabei entstehende Wert wird mit dem Wahrscheinlichkeitsfaktor  $\gamma$  multipliziert, der ähnlich wie bei Page Rank Sprünge angibt und im Intervall  $0 < \gamma < 1$  liegt.

Ähnlichkeit ist in der Graphentheorie folgendermaßen definiert,

$$(x, v) \in_k E \Rightarrow (x, w) \in_k E \text{ und } (v, x) \in_k E \Rightarrow (w, x) \in_k E,$$

und bedeutet, zwei Knoten sind sich ähnlich, wenn man von einem zum anderen über einen dritten Knoten kommen kann.

## 3.3 Höher wertige Methoden - "Meta Methoden"

Diese im folgenden vorgestellten Methoden sind keine für sich stehende Ansätze, sondern dienen als Vorarbeit und zur Verbesserung der Ergebnisse beziehungsweise der Laufzeit der vorhin vorgestellten Algorithmen. Es ist allerdings nicht Ziel dieser Arbeit sie im Detail vorzustellen, sondern nur die Idee zu skizzieren.

### 3.3.1 Schwach eingestufte Aproximation

Dieser Ansatz macht sich die Tatsache zunutze, dass alle Algorithmen, die bis jetzt vorgestellt wurden, eine äquivalente Formulierung der adjazenz Matrix, die zur Repräsentierung des Graphen  $G_{collab}$  verwendet werden kann, haben. So besteht der common neighbour Algorithmus, laut Liben-Nowell und Kleinberg, einfach aus der Zuordnung jedes Knoten  $x$ , zu seiner Reihe  $r(x)$  in  $M$ . Der  $score(x, y)$  wird dann als inneres Produkt der Reihen  $r(x)$  und  $r(y)$  definiert.

Die Idee der beiden Autoren ist nun, über die Berechnung der *rang* – *k* Matrix  $M_k$ , wobei laut den Autoren *k* „frei“, aber unter der Voraussetzung dass es ein kleiner Wert sein muss, wählbar ist, die  $M$  näherungsweise am besten darstellt, einen Vorteil zu erreichen, in dem auf dieser „vereinfachten“ Matrix gearbeitet wird. Sie selbst bezeichnen das Arbeiten mit  $M_k$  anstatt mit  $M$  als eine Art von „noise-reduction“.

### 3.3.2 Unseen Bigrams

$$\begin{aligned} score_{ungewichtet}^*(x, y) &:= |\{z : z \in \Gamma(y) \cap S_x^{<\delta>}|\} \\ score_{gewichtet}^*(x, y) &:= \sum_{z \in \Gamma(y) \cap S_x^{<\delta>}} score(x, z) \end{aligned}$$

Mit diesem Ansatz wird versucht das Dilemma, dass neue Kanten, die erst noch entstehen müssen, die Bildung anderer Kanten die etwas später entstehen beeinflussen. Indem durch diese zuvor entstandenen Kanten „neue“ Nachbarschaften, beziehungsweise kürzeste Wege entstehen die wiederum die Bildung einer Kante zwischen zwei anderen Kanten theoretisch ermöglichen. Und genau diese Tatsache versuchen Liben-Nowell und Kleinberg durch die obigen beiden Formeln in ihren *score* einfließen zu lassen, um die Genauigkeit ihrer Aussage zu verbessern. Man kann diese Formeln daher auch als Glättung oder Gaußfilter betrachten

$S_x^{<\delta>}$  steht hier für die Menge aller  $\delta$ ,  $\delta \in \mathbb{Z}^+$ , Knoten, die unter einem bestimmten *score*, leider führen Kleinberg und Liben-Nowell wieder nicht aus um was für einen *score*, beziehungsweise Größenordnung es sich dabei handelt, „ähnlich“ gegenüber  $x$  sind. Gemeint ist wohl, welche Knoten eine Kante ausbilden könnten, die einen positiven Einfluss auf die Kantenbildung zwischen den beiden im Moment betrachteten Knoten haben. Unterschieden wird noch zwischen einer ungewichteten Variante, in der einfach alle nützlichen Knoten, die auch Nachbarn von  $y$  sind addiert werden und so deren Menge den *score* bildet. Natürlich muss zur Beurteilung der Knoten  $z$  zuerst deren *score* in Bezug auf  $x$  berechnet werden. Und einer gewichteten Variante, bei der einfach über die  $score(x, z)$  Werte, aufsummiert wird, wieder unter der Voraussetzung, dass die  $z$  Knoten in der Menge der Nachbarn von  $y$  liegen und in der Menge  $S_x^{<\delta>}$ .

### 3.3.3 Clustering

Beim clustering geht es darum, den Graphen zu „lichten“, das heißt Kanten die für die Berechnung des *score* nicht gebraucht werden, beziehungsweise nicht relevant sind heraus zu filtern und aus der anschließenden Vorhersage heraus zu nehmen, um die Laufzeit des verwendeten Algorithmus zu verbessern. Es ist allerdings schwierig festzulegen, welche Kanten nicht relevant sind, diese Frage kann sehr kontrovers diskutiert werden. So sind für den

einen Kanten nicht relevant, die zwei Gruppen von Knotenmengen miteinander verbinden, weil diese einzelne Verbindung angeblich keine Bedeutung über die Beziehung der Knotenmengen hat und deshalb eine Ausnahme darstellt, ähnlich wie ein Ausreißer bei einer Messreihe, was eben schon mal passieren kann aber für die Auswertung keine Rolle spielt. Für andere ist gerade das Gegenteil der Fall, sie halten eine solche Kante, die zwei Knotenmengen miteinander verbindet für sehr wichtig, da sie die einzige Verbindung darstellt und ohne sie mögliche Ergebnisse verfälscht werden können. Hier soll allerdings nur die Idee von Kleinberg und Liben-Nowell vorgestellt werden und nicht diese Kontroverse gelöst werden.

Sie „löschen“ alle Kanten, deren  $score(u, v)$ , den sie zu Anfang für alle Kanten des Graphen berechnet haben, die geringsten Werte dieses berechneten scores aufweisen. Leider geben sie, ähnlich wie bei dem Problem des Schätzers, keinen Grenzwert an bis zu dem die Kanten gelöscht, entfernt, werden und welche Kanten demnach relevant sind. Auch hängt dieses clustering, zu mindest soweit aus dem Artikel ersichtlich, nicht davon ab welches Knotenpaar eigentlich zur Berechnung des  $score$  betrachtet wird. Und das ist nicht ganz unkritisch, denn global betrachtet mag eine bestimmte Kante keine Bedeutung haben, kann aber lokal für diesen einen Fall enorm wichtig sein. Nachdem dies nun getan ist, wird der  $score(x, y)$  der beiden eigentlich betrachteten Knoten, für die eine Vorhersage gemacht werden soll, auf diesem, wie sie es nennen, „bereinigten“ Teilgraphen berechnet.

Ziel dieser Methode ist es im Vorfeld Kanten aus zu sortieren, die für die eigentliche Vorhersage keine beziehungsweise eine so geringe Rolle spielen, dass sie nicht ins Gewicht fallen. Dadurch soll bei der eigentlichen Vorhersage Rechenzeit gespart werden.

## 4 Ergebnisse und Ausblicke

Die Analyse der Ergebnisse von Kleinberg und Liben-Nowell kann durchaus sehr kontrovers gesehen werden. So wird zum Beispiel als Vergleichsreferenz, auf deren Ergebnissen Aufbauend all ihre Algorithmen bewertet werden, ein Zufallsvorhersagealgorithmus verwendet, über den man nur weiß mit welcher Wahrscheinlichkeit er bei den einzelnen Graphen eine richtige Vorhersage macht. Aber keine weiteren Informationen darüber, wie diese Wahrscheinlichkeiten zu Stande kommen, noch wie dieser Vergleichsalgorithmus funktioniert, das heißt es gibt keine Informationen darüber wie oder mit welcher „Zufalls-“ Wahrscheinlichkeit der Algorithmus eine Kante zwischen einem Knotenpaar vorhersagt. Eine Möglichkeit wäre, dass der Zufallsalgorithmus jedes betrachtete Knotenpaar mit gleicher Wahrscheinlichkeit in die Vorhersage aufnimmt, und somit unter den Knotenpaaren eine Gleichverteilung der Kollaboration angenommen wird.

Es gibt zwar eine Tabelle[1, Figure 8, Seite 13], die angibt wie viele richti-

ge Vorhersagen jeder einzelne Algorithmus gemacht hat und wie viele richtige Vorhersagen mit denen von anderen Algorithmen übereinstimmen. Aber es wird daraus nicht ersichtlich ob es sich dabei um die richtige Vorhersage ganzer Durchläufe, also um komplett richtige Vorhersagen von Kantenmengen im neuen Graphen, oder um die einzelne Menge aller in allen Durchläufen richtig vorhergesagten Kanten.

Diese Aussage ist allerdings abhängig von der Definition einer Vorhersage, hier wird sie so verstanden, dass es sich bei einer Vorhersage um eine Menge neuer Kanten im Graphen zu einem bestimmten Zeitpunkt  $t_1 < t'_1$  handelt, was somit keine aussage über die Genauigkeit der Vorhersage einer einzelnen Kante zulässt, das wiederum wird aus der Definition des "Link Prediction Problems" und den Erläuterungen der Autoren zu ihren Algorithmen abgeleitet. Des weiteren sind die Informationen über die verwendeten Graphen etwas spärlich, so gibt es im gesamten Artikel nur eine Tabelle die die Stammdaten der Graphen enthält, Anzahl Autoren, Artikl, Kante und wie sich die Anzahl Kanten von Trainings- zum Testset verhält.

Es gibt aber keine schlüssigen Hinweise auf die Topologie und Komplexität der Graphen, mit denen man das Abschneiden der Algorithmen vergleichen könnte um eine Aussage darüber zu machen, ob das erzielte Ergebnis auf diesem Graphen nun gut ist, da er eine sehr komplexe Topologie hat, oder eher als schlecht zu bewerten ist, da die Topologie sehr simpel ist. Die einzige Information in dieser Hinsicht gibt es zu den Graphen astro-ph, der als „schwierige“ Datenstruktur bezeichnet wird, und gr-qc, der als ein Graph mit „einfacher“ Struktur. Daher fällt es auch schwer ihre Aussagen darüber welcher Algorithmus nun für welchen Fall, „einfache“ oder „schwere“ Datenstruktur, besser geeignet sei entsprechend nachzuvollziehen und gegebenenfalls zu überprüfen. Man ist in diesem Fall ganz und gar von den Aussagen, beziehungsweise interpretierten Ergebnissen, der Autoren abhängig.

Die erste Aussage von Liben-Nowell und Kleinberg über ihre Algorithmen ist die, dass die Ergebnisse so „schlecht“ ausfielen, weil viele Faktoren, wie früher schon besprochen, außerhalb der in einem Netzwerk abbildbaren Faktoren liege. Die zweite Aussage ist, dass es keinen klaren Gewinner gibt. An dieser Stelle werfen sich einige Fragen auf.

- Gibt es überhaupt einen Gewinner?
- Wie sinnvoll ist ein Algorithmus, der im Idealfall mit ungefähr 16% Wahrscheinlichkeit eine richtige Vorhersage liefert?
- Was sagt der Vergleich mit einem Zufallsvorhersagealgorithmus, der mit einem Intervall zwischen 0,15% und 0,48% liegt eine richtige Vorhersag zu machen, über die anderen Algorithmen aus?

Was auffällt ist, dass die Ergebnisse der Algorithmen als Faktoren angegeben werden, die die Verbesserung gegenüber dem Zufallsvorhersagealgorithmus

angeben. Man kann sich nun fragen warum diese Darstellungform gewählt wurde und die eigentliche Korrektheit der Vorhersage, der Algorithmen, nicht in Prozent angegeben wurde.

Was sagen nun Kleinberg und Liben-Nowell über ihre Algorithmen bei den vorliegenden Testbedingungen. Heraussticht, dass der Katz Algorithmus und seine Varianten, mit clustering und low-rank approximation, im Vergleich mit den anderen Algorithmen konsistent "gute" Ergebnisse liefert. Bei drei von fünf arXiv, Sammlung von e-Print Papers aus dem Bereich Physik, Abschnitten, eine Variante von Katz die beste Leistung. Außerdem hätten die sehr einfachen Maße, wie zum Beispiel common neighbours und Adamic/Adar, überraschend gut abgeschnitten.

Außerdem fanden sie heraus, dass es bei den unterschiedlichen Algorithmen häufig zu Überschneidungen bei den Vorhersagen gekommen ist. Dies ist allerdings nicht sonderlich verwunderlich, denn einige Algorithmen unterscheiden sich unter bestimmten Bedingungen kaum, oder sind sogar fast gleich, wie zum Beispiel common neighbours und der Katz Algorithmus bei kleinem  $\beta$ , denn dann spielen Pfade der Länge drei oder länger in der Berechnung kaum noch eine Rolle und entscheidend sind dann wie bei common neighbours die direkten Nachbarn der beiden betrachteten Knoten.

Ein großes Problem, gerade für die einfachen Algorithmen sei laut intensiver Forschung von Kleinberg und Liben-Nowell das "small world problem". Denn das "small world problem" sagt aus, dass es grob gesagt zwischen zwei völlig unabhängigen Personen häufig kurze Pfade gibt. Für das Link Prediction Problem handelt es sich bei den kritischen Pfaden um Pfade der Länge 2, die Probleme verursachen. Das wirft nun für die einfachen Algorithmen, die sich das Prinzip der gemeinsamen Nachbarn zu nutze machen, das Problem auf, dass sie in diesen Fällen mit hoher Wahrscheinlichkeit eine Kollaboration vorhersagen werden, die aber nie stattfinden wird, da beide Knoten völlig unabhängig von einander sind und diese Verbindung keine Aussagekraft hat. Daher sind alle Algorithmen, die auf dem Prinzip der gemeinsamen Nachbarn beruhen, in diesem Fall nicht mehr geeignet. Die einzigen Algorithmen, bei denen man dieses Problem lösen kann sind also demnach die Algorithmen die als Grundlage ihrer Vorhersage die kürzesten Wege nehmen, denn bei ihnen kann man dies Wege der Länge zwei sehr einfach ausschließen, indem man sie nur Pfade der Länge drei und größer verwenden lässt.

Als Ergebnisse und Ziele für weitere Arbeit auf diesem Gebiet führen Liben-Nowell und Kleinberg an, das man die Informationen, die im Datensatz des Trainingsgraphen enthalten ist besser auszuschöpfen und zu nutzen. Außerdem müsste die Effektivität, in Bezug auf große Netzwerke, der Methoden, die auf dem Prinzip der gemeinsamen Nachbarn beruhen, verbessert werden.

## Literatur

- [1] David Liben-Nowell, John Kleinberg "The Link Prediction Problem", 2004
- [2] A. L. Barabasi, H. Jeong, Z. Nda, E. Ravasz, A. Schubert, and T. Vicsek. "Evolution of the social network of scientific collaboration.", *Physica A*, 311(3-4):590-614, 2002.
- [3] Lilian Lee "Measures of distributional similarity.", In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 25-32, 1999.