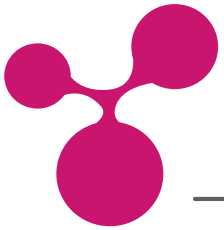
Technische Universität Dresden
Medienzentrum

Prof. Dr. Thomas Köhler
Jun.-Prof. Dr. Nina Kahnwald
(Hrsg.)



GeNeMe '13

Gemeinschaften in Neuen Medien

an der
Technischen Universität Dresden
mit Unterstützung der

BPS Bildungsportal Sachsen GmbH
Campus M21
Communardo Software GmbH
Dresden International University
eScience – Forschungsnetzwerk Sachsen
Gesellschaft der Freunde und Förderer der TU Dresden e.V.
Gesellschaft für Informatik e.V.
Gesellschaft für Medien in der Wissenschaft e.V.
IBM Deutschland
itsax – pludoni GmbH
Kontext E GmbH
Learnical GbR
Medienzentrum, TU Dresden
ObjectFab GmbH
Transinsight GmbH
T-Systems Multimedia Solutions GmbH
Universität Siegen

am 07. und 08. Oktober 2013 in Dresden

www.geneme.de
info@geneme.de

C Community Topics

C.1 Der Einfluss der Länge von Beobachtungszeiträumen auf die Identifizierung von Subgruppen in Online Communities

Sam Zeini[1], Tilman Göhnert[1], Tobias Hecking[1], Lothar Krempel[2], H. Ulrich Hoppe[1]
[1]Universität Duisburg-Essen, Campus Duisburg
[2]Max Plank Institut für Gesellschaftsforschung, Köln

1 Einleitung

Die Verbreitung von Social Media und damit verbunden die entstehenden und wachsenden Communities im Internet führen zu einer Zunahme von auswertbaren, digitalen Spuren, die häufig öffentlich zugänglich sind. Diese lassen sich durch verschiedene analytische Verfahren wie z.B. die Methode der Sozialen Netzwerkanalyse [1] auswerten. Insbesondere Ansätze für „Community Detection“ erfreuen sich besonderer Beliebtheit, wodurch sich unter anderem innovative Untergemeinschaften und Subgruppen beispielsweise in großen „Open Source“-Projekten identifizieren lassen [2]. Im Rahmen dieser Anwendungen ergeben sich neue methodische und grundlegende Fragen, darunter die nach der Rolle der von Zeit in solchen Analysen. Während die Darstellung dynamischer Effekte (z.B. durch Animationen) die Zeit als expliziten Parameter enthält, geht die Wahl der Zeitintervalle für die Aggregation von Daten, aus denen dann Netzwerke gewonnen werden, nur implizit in die Prämissen des Verfahrens ein. Diese Effekte wurden im Gegensatz zur Analyse von Dynamik bisher kaum untersucht. Im Fall der Sozialen Netzwerkanalyse ist die Zielrepräsentation selbst nicht mehr zeitbehaftet sondern sozusagen ein „statischer Schnappschuss“, wodurch etwa zeitabhängige Interaktionsmuster nicht erkannt werden können.

Während der Untersuchung von innovativen Gemeinschaften im Bereich der „Open Source“-Softwareentwicklung (ebd.) beobachteten wir, dass die Wahl unterschiedlicher Längen bei Zeitscheiben einen Einfluss auf die Ergebnisse von „Community Detection“-Verfahren hatte. Die ursprüngliche Untersuchung sah vor, mittels verschiedener Rollenmodelle nach potentiellen Innovatoren in „Open Source“-Netzwerken zu suchen. Dabei haben wir klassische Brokermodelle [3],[4], die individuelle Informationsmaklern als innovationskritische Rollen betrachten, mit überlappenden Mitgliedern von Cliquen verglichen [5], die unserer Annahme nach als quasi Gruppen von Vermittlern arbeitsteilig zwischen Teams in komplexen Projekten fungieren. Bei der Anwendung der Methode der „Clique Percolation“ (siehe Abschnitt über Ansatz) [6] fanden wir unterschiedliche Ergebnisse vor, je nachdem

welche Länge für die Zeitintervalle gewählt wurde. Wählt man beispielsweise Scheiben der Länge eines Jahres in einem dicht verwobenen Developer-Netzwerk, so kann hierbei eine einzige große Clique identifiziert werden. Wählt man im Gegensatz z.B. Zeitscheiben der Länge von einem Tag oder wenigen Tagen in stark spezialisierten Mailinglisten, so ist es wahrscheinlich, dass gar keine Clique gefunden wird. Der beobachtete Effekt kann anschaulich mir einer Metapher aus der Fotografie umschrieben werden: Es ist eine bestimmte Länge der Zeitscheiben notwendig, um ein klares Bild der vorhandenen Subgruppen zu erhalten. Wählt man eine zu lange Zeitscheibe, wird das Bild unscharf bzw. wirkt verwackelt, da Bewegungen zwischen den Cliquen innerhalb der langen Beobachtungsphase nicht mehr aufgelöst werden.

Wir gehen davon aus, dass die „Uhren“ in unterschiedlichen Gemeinschaften jeweils spezifisch „ticken“, sodass die Wahl geeigneter Messintervalle an diese „Eigenzeit“ der jeweiligen Situation angepasst werden muss. Um Möglichkeiten einer optimalen Annäherung auszuloten, haben wir zwei unterschiedliche „Community Detection“-Methoden auf drei unterschiedliche Communities angewandt. Aus Platzgründen gehen wir in diesen Beitrag hauptsächlich nur auf „Clique Percolation“ und deren Anwendung auf zwei idealtypische Communities ein. Im Vergleich zu [7] validieren wir unseren Ansatzes durch ein anderes „Community Detection“-Verfahren, den Link Communities Ansatz, und erweitern unsere Samples durch eine weitere Fallstudie.

2 Verwandte Arbeiten

Mittlerweile existieren zahlreiche Forschungsarbeiten zu Dynamik in Netzwerken, die insbesondere Evolution und Wachstum im Fokus haben. Falkowski [8] liefert in ihrer Dissertation einen ausführlichen Überblick und entwickelt darin Ansätze für das Clustering in dynamischen Netzwerken. Jedoch fehlt hier eine Auseinandersetzung mit der Problematik des Einflusses der Variation von Zeitscheibenlängen auf Clustering. Es existiert eine frühe Arbeit aus den 1980er Jahren aus dem Bereich der Psychologe zu geben [9], die sich unter der Frage nach Zugehörigkeitsempfinden zu Cliquen diesem Phänomen mit der Annahme annähert, dass Akteure bei entsprechend langer Beobachtung mathematisch zwar einer Clique zugeordnet werden können, jedoch sich nicht daran erinnern, jemals in dieser Gruppe gewesen zu sein. Diese Annahme ist sicher auch für die aktuelle Frage nach „Community Detection“ in dynamischen Netzwerken von Relevanz, auch wenn sie nicht mehr von Community-Forschern aufgegriffen wurde.

Andere Ansätze, welche die zeitlicher Dynamik in Netzwerken in Betracht ziehen, sind z.B. die sogenannten „Moving Structures“ bei dynamischen positionalen Analysen in Netzwerken [10] oder aber auch die Arbeiten von Matthias Trier zu Zuverlässigkeit von Zentralitätsmaßen in zeitlicher Betrachtung [11].

Konkrete Arbeiten zum Einsatz der „Clique Percolation“-Methode in dynamischen Netzen versuchen, mögliche temporale Effekte durch eine auf Basis von Gesamtproportionen voraus definierte Initialgröße von Sub-Communities [12] oder Größe und Alter von Communites [13] zu adressieren, oder aber auch relevante Ereignisse [14] als heuristische Hilfsmittel verwenden. Erst die zeitgleich zu unserer Initaluntersuchung [7] erschienene Arbeit von Budka et al. [15] thematisiert das gleiche Problem der Ermittlung von passenden Zeitscheiben bei der Suche nach bestimmten Mustern in dynamischen Netzwerken. Einen ausführlichen Gesamtüberblick über die Analyse dynamische Communities in Netzwerken liefern Coscia et al. [16].

3 Ansatz

Im Vergleich zu anderen Clustering-Verfahren erlauben die „Clique Percolation“-Methode (CPM) [6] und der Ansatz der „Link Communities“ (LC) [17] die Zugehörigkeit von einem Knoten zu mehr als einem Cluster. Somit können sogenannte überlappende Mitglieder in verschiedenen Sub-Communities ermittelt werden. Dies entspricht häufig der empirischen Realität, wo beispielsweise ein Softwareentwickler Mitglied von verschiedenen Teams sein kann oder gleichzeitig verschiedene Rollen innehaben kann (z.B. Entwickler und Koordinator).

Eine Community ist im Sinne von CPM definiert als eine Einheit aller k-Cliquen, die einander durch eine Reihe von adjazenten k-Cliquen erreichen können. k-Cliquen sind bei dieser Definition vollständige Subgraphen der Größe k und zwei k-Cliquen sind adjazent, wenn sie mindestens k-1 Knoten teilen. Percolation bezieht sich in diesem Falle auf eine Komponente, die über die adjazenten k-Cliquen durch den Graphen perkoliert, wobei in jedem Zug nur ein Knoten seine Position ändert (s. Abbildung 1 als Beispiel für eine 4-Cliquen Perkolation).

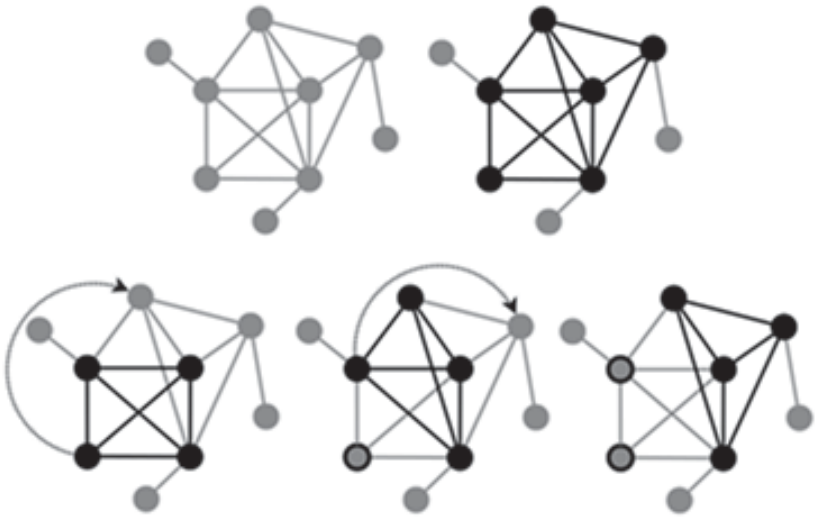


Abbildung 1: Perkolation einer 4-Clique

Wie CPM erlaubt auch die LC-Methode [17] das Finden von überlappenden Gruppen. Allerdings unterscheidet sich die Interpretation dieser Überlappungen. In der LC-Methode werden Gruppen als soziale Dimensionen oder Kontexte definiert (z.B. über gemeinsame Interessen), zwischen denen hierarchische Strukturen und Überschneidungen auftreten können, im Gegensatz zur sonst üblichen Definition über Teile von Netzwerken, die intern stärker vernetzt sind als nach außen hin. Während Knoten üblicherweise in mehreren Kontexten auftreten, wird für Kanten hingegen angenommen, dass sie eindeutig einem Kontext zuzuordnen sind. Um hierarchische und überlappende Gruppenstrukturen erkennen zu können, wird ein hierarchisches Clustering von Kanten anstelle von Knoten durchgeführt.

Wir haben mehrere mögliche Indikatoren identifiziert, die die Gruppenstruktur von Netzwerken wiederspiegeln und als Basis dienen können für einen Mechanismus, der das Finden geeigneter Zeitscheibengrößen für die Untersuchung eines gegebenen Netzwerks unterstützt. Wir definieren einen Indikator hier als Funktion , welche ein Netzwerk and eine durch ein Clusteringverfahren gefundene Gruppenstruktur auf eine reale Zahl abbildet. Die Basisindikatoren, die wir nutzen, sind die Zahl der gefundenen Cluster, eine aggregierte Variante von Clustergröße, Coverage und Overlap-Coverage. Coverage und Overlap-Coverage sind dabei definiert als der Anteil von Knoten, die in mindestens einem Cluster (Coverage) oder in mindestens zwei Clustern (Overlap-Coverage) enthalten sind zur Gesamtanzahl der Knoten des Netzwerks:

$$cov\,(N,D) = \frac{|\{v \in V(N) : \exists c \in C(N,D), v \in c\}|}{|V(N)|}$$

$$ol\_cov\,(N,D) = \frac{|\{v\ |IN\ V(N) : |\{c : c \in C(N,D), v \in c\}| > 1\}|}{|V(N)|}$$

In beiden obigen Formeln bezeichnet die Knotenmenge eines Netzwerks und die Menge der Cluster, die durch Anwendung des Clusteringverfahrens auf das Netzwerk gefunden wurden.

Im Gegensatz zu den anderen Indikatoren muss die Clustergröße für ein komplettes Netzwerk aggregiert werden. Für diese Aggregation nutzen wir Maximum, Durchschnitt und Varianz bzw. Standardabweichung, sodass die aggregierten Formen dann unserer Definition eines Basisindikators entsprechen.

Zusätzlich zu diesen einfachen Indikatoren führen wir noch einen kombinierten Indikator aus Anzahl der Cluster und Overlap-Coverage ein. Dieser Indikator ist für den Fall $|\{c: c \in C(N,D)\}| > 1$ definiert als

$$ci(N,D) = \frac{|\{c: c \in C(N,D)\}| \cdot m}{|V(N)|} + ol\_cov(N,D)$$

und als $ci(N,D) = 0$ für alle anderen Fälle. Hier bezeichnet m die minimale Clustergröße. Dieser Skalierungsfaktor wurde eingeführt, um einen Vergleich zwischen CPM und anderen Clustering-Verfahren zu ermöglichen, da in CPM die minimale Clustergröße durch den Parameter k vorgegeben ist. Um nun die Ergebnisse von CPM mit den Ergebnissen von Clusteringverfahren, die keine Beschränkung der Mindestgröße von Clustern haben, vergleichen zu können, setzen wir $k = m$ und ignorieren alle Cluster, deren Größe kleiner als m ist. Wenn nicht anders angegeben gilt für die präsentierten Ergebnisse, dass sie mit $k = m = 4$ gewonnen wurden. Die Fallunterscheidung für diesen Indikator ist notwendig, da für den Fall, dass nur ein Cluster gefunden wird, dass Ergebnis sonst dem Kehrwert der Knotenzahl des Netzes entspricht, was keine Aussage über die Subgruppenstruktur des Netzes erlaubt.

Auf der Basis dieser Indikatorfunktionen definieren wir nun Zielfunktionen der Form $o(T,D) -> r$, wobei T eine Zeitreihe, also eine Folge von Zeitscheiben eines Netzwerks gleicher Länge ist. Um aus den vorgestellten Basisindikatoren Zielfunktionen zu erzeugen, wird erneut aggregiert, diesmal über alle Zeitscheiben einer Zeitreihe. Es werden die gleichen Formen der Aggregation verwendet, die auch bei der Clustergröße angewendet werden. Die Maxima einer so erzeugten Zielfunktion über verschiedene Zeitreihen sollten nun auf die ideale Zeitscheibengröße (repräsentiert durch die jeweilige Zeitreihe) hinweisen und damit Einblicke in die Dynamik des untersuchten Netzes bieten.

4 Fallstudien

Für die vorliegende Studie wählen wir als Untersuchungsfälle die „Open Source"-Projekte Asterisk und OpenSimulator (OpenSim) sowie das Dojo Toolkit. Sie stellen damit unterschiedlich große Fälle aus verschiedenen Domänen dar.

Asterisk ist eine Software-basierte Telefonanlage, die von der Firma Digium entwickelt wird. Es hat neben dem offenen GNU Lizenzmodell auch eine kommerzielle Lizenz. Auch wenn das Projekt von einer Firma vorangetrieben wird, steht eine große Community hinter dem Projekt. Die Software wird vielfach in Callcentern eingesetzt

und durch das „Open Source“-Modell sind die Unternehmen, die sie einsetzen, in der Lage sie nach eigenen Bedürfnissen anzupassen. Es ist ein typisches, großes Projekt mit einem relativ stabilen Kern von Entwicklern und Nutzern sowie eine stark fluktuierende Nutzergruppe, die das Projekt zeitweise begleiten. Für die vorliegende Fallstudie haben wir Netzwerkdaten aus Mailinglisten und Quellcode-Repositories (SVN) zu extrahiert (s. [18]). Für Asterisk wurden die Jahre 2006 und 2007 erhoben. Die Entwickler-Mailingliste beinhaltet 13542 Nachrichten verteilt auf 4692 Themen, die von 1324 Entwicklern und Nutzern diskutiert werden. Die Community-Mailingliste beinhaltet 67949 Mails verteilt auf 26095 Themen, die von 4642 Nutzern diskutiert werden. Das SVN Archiv setzt sich aus 17868 Revisionen von 1866 Artefakten zusammen, die von 30 Kernentwicklern gepflegt werden.

OpenSimulator ist ein „Open Source“-Projekt, das sich mit der Entwicklung einer Server-seitigen Software befasst, um 3D-Simulationswelten vergleichbar mit dem proprietären System „Second Life“ zu betreiben. Das Lizenzmodell hier ist die BSD Lizenz. OpenSimulator stellt eine mittelgroße Community dar. Im untersuchten Zeitraum von September 2007 bis Februar 2009 haben insgesamt 198 Personen über 1185 Themen (5505 E-Mails) in der Entwickler-Mailingliste diskutiert. Interessanterweise haben sich in der Community-Mailingliste für den gleichen Zeitraum weniger Personen an weniger Themen beteiligt (175Personen, 634Themen, 1582E-Mails). Beim Sourcecode-Management wird im Falle von OpenSimulater Subversion (SVN) verwendet. Hier haben im untersuchten Zeitraum mit 26 Personen vergleichsweise wenige Entwickler Schreibrechte auf dem System. Diese haben an 6012 Objekten gearbeitet (insgesamt 32867 Objekte bei Berücksichtigung aller Versionen).

Das Dojo Toolkit ist eine „Open Source“ Javascript Bibliothek, die von der Dojo Foundation entwickelt wird. Ziel der Bibliothek ist es die Client-seitige Webentwicklung zu unterstützen. Dojo verwendet die BSD Lizenz sowie die Academic Free License. Daten liegen hier von der Entwickler-Mailingliste für die Jahre 2006 und 2007 vor sowie für das SVN Archiv von Januar 2006 bis August 2007. Die Mailingliste beinhaltet 7207 Nachrichten in 1477 Threads, die von 114 Entwicklern diskutiert werden. Das SVN-Quellcodearchiv beinhaltet 15845 Revisionen von 4151 Artefakten, die durch 29 Kernentwickler gepflegt werden. Zum Zeitpunkt der Erhebung stellte das Projekt ein eher typisch kleines „Open Source“-Projekt dar, bestehend hauptsächlich aus einem stabilen Kern.

Für die Evaluation unseres Ansatzes zur Ermittlung optimaler Zeitfenster verwenden wir im Weiteren nur die Entwickler-Mailinglisten. Die Community-Mailinglisten stellen sich als zu fragmentiert dar. Die SVN Daten bestätigen unsere Annahme,

dass durch die kontrollierten Schreibrechte der wenigen Kernentwickler, die Netze sich insbesondere über Zeit zu einer großen Clique hin entwickeln, wo am Ende praktisch jeder alles macht. Die Fälle unterscheiden sich insbesondere durch Größe. Während Asterisk ein großes Projekt mit über 1000 Entwicklern ist, sind OpenSimulator und Dojo kleine bis mittlere Projekte bestehend aus 100 bis 200 aktiven Mitgliedern. OpenSimulator und Dojo unterscheiden sich u.a. wiederum dadurch, dass OpenSimulator eine höhere Fluktuation in der Peripherie aufweist, wenngleich beide einen stabilen Kern besitzen.

5 Ergebnisse

Die ursprüngliche Annahme unserer Untersuchungen war, dass die Dynamik produktiver Communities eine inhärente Eigenzeit beinhaltet. Die Arbeits- und Produktionsphasen der Gemeinschaft unterliegen einem Tempo, der je nach Community stark variieren. Darüber hinaus unterliegt dieser Zeiteffekt selbst einer gewissen Dynamik, die durch interne oder externe Ereignisse beeinflusst wird, beispielsweise als Eile im Projekt kurz vor einer Deadline [2]. Die eigentliche Dynamik, also die Entwicklung strukturelle Eigenschaften des Netzwerks über Zeit unterscheidet sich bei den unterschiedlichen Communities.

Am stabilsten kann hierbei Dojo angesehen werden. Bei Asterisk steigt die Dichte über die Zeit. Zugleich nimmt die Anzahl der Teilnehmer in der Mailingliste ab. Es sind also zunehmend weniger Entwickler dabei, die zunehmend intensiver kommunizieren. OpenSimulator ist ein typisches wachsendes Netzwerk. Während die Dichte am Anfang abnimmt und sich dann einpendelt, nimmt die Anzahl der Knoten im Laufe der Zeit zu. Bezogen auf den CPM Algorithmus bestätigt die Beobachtung, dass der höchste k Wert bei der mindestens noch eine Gruppe gefunden wird bei k=5 für Asterisk und k=6 für OpenSimualtor bei einer Zeitscheibenlänge von drei Monaten liegt. Dies passt zu unserer ursprünglichen Annahme, dass Zeitscheiben der Länge von zwei bis drei Monaten für die untersuchten Communities die geeignetsten seien. Daher haben wir im Folgenden den Fokus auf k Werte zwischen k=3 und k=6 gelegt.

Die Veränderung der Dichte in den Netzwerken hat mögliche Effekte für unsere Indikatoren. So kann die zunehmende Dichte zunächst durch mehr Interaktionen zu einer größeren Anzahl von Clustern führen, die aber im Laufe der Zeit immer mehr miteinander verschmelzen. Eine solche Verschmelzung kann bei Open Source Netzen beispielsweise dazu führen, dass zwar mit der Zeit die Anzahl der Cluster abnimmt, aber die Verschmelzung sich ungleich verteilt. So findet man oft ein großes Cluster, die aus mehreren kleinen Clustern entstanden ist, neben wenigen kleinen Clustern, die nicht weiter verschmelzen über Zeit.

Die Einbeziehung von Dynamik bedeutet für den Coverage-Indikator, dass bei wachsender Zeitscheibenlänge auch die Zahl der innerhalb der Zeitscheibe beobachteten Akteure steigen kann, so dass der Indikator dadurch für eine längere Zeitscheibe einen kleineren Wert annehmen kann, als er ihn für kleinere Zeitscheibe im gleichen Zeitraum hat. Allerdings tritt dieser Fall in den beobachteten Daten selten auf, so dass dieser Indikator mit länger werden Zeitscheiben ansteigt, weshalb wir ihn im Folgenden nicht weiter betrachten.

Aus diesen Überlegungen heraus lässt sich ableiten, dass Coverage und Clustergröße keine zuverlässigen Indikatoren darstellen. Sie neigen dazu einfach mit zunehmender Länge der Zeitscheiben anzusteigen, ohne etwas über die Dynamik der Gemeinschaften auszusagen. Overlap-Coverage und Anzahl der Cluster hingegen sind unserer Beobachtung nach zuverlässige Indikatoren.

Sowohl CP als auch das LC- Verfahren sind mit der Zielvorgabe definiert worden, insbesondere überlappende Subgruppen in Netzwerken zuzulassen und zu finden. Daher bietet sich die Overlap-Coverage der mit diesen Verfahren identifizierten Cluster für verschiedene Zeitscheibenlängen zu betrachten. Abbildung 2 zeigt am Beispiel von Asterisk, wie ein mögliches optimales Zeitfenster identifiziert werden kann. Hierbei ist zu beachten, dass die Overlap-Coverage stark mit Anzahl von Clustern zusammenhängt. Je mehr Subgruppen gefunden werden, desto mehr Überlappungen sind möglich.

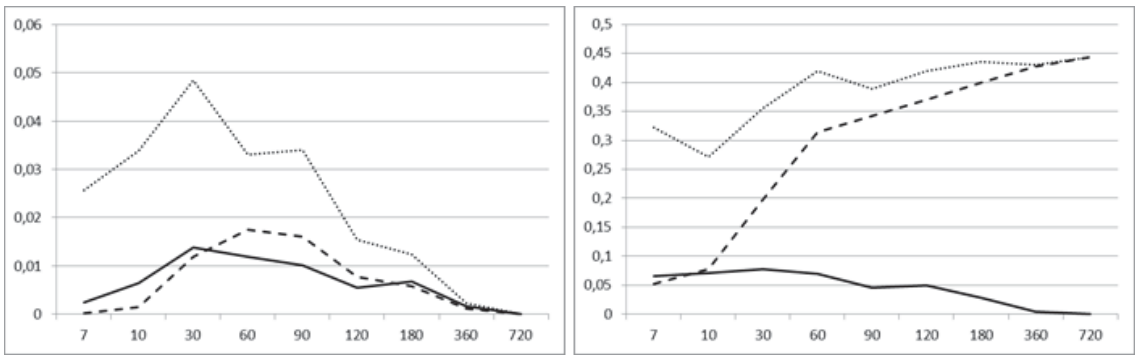


Abbildung 2: Overlap-Coverage für Asterisk, links CPM, rechts LC (gestrichelt= Durchschnitt, gepunktet = Maximum und durchgängig= Standardabeichung)

Asterisk ist das geeignetste Beispiel aus den drei Communities, da es größer ist und zugleich die geringste Dichte aufweist. OpenSimulator und Dojo sind kleiner, weisen aber insgesamt eine höhere Dichte auf. Dies führt für die CPM Methode dazu, dass bei Zeitscheiben der Länge von drei Monaten und mehr bei OpenSimulator und Dojo nicht mehr als ein großer Cluster identifiziert wird (bei k=4). Das Beispiel Asterisk

verdeutlicht zudem, dass die höhere Dichte bei längeren Zeitscheiben einen eindeutig starken Einfluss auf „Community Detection“-Verfahren hat. Hier identifiziert LC im Kontrast zu CPM bei der Verlängerung der Zeitscheiben tendenziell mehr Cluster, da diese in dem Falle über Kanten und nicht über Knoten identifiziert werden. Hierbei kann sowohl bei CPM als auch bei LC ein Peak bei der Varianz von Clustergrößen bei identifizierten Clustern sich als ein guter Indikator für geeignete Längen der Zeitscheiben erweisen, auch wenn diese Abweichungen in den von uns beobachteten Fällen mit LC nicht so deutlich ausgeprägt sind wie bei CPM.

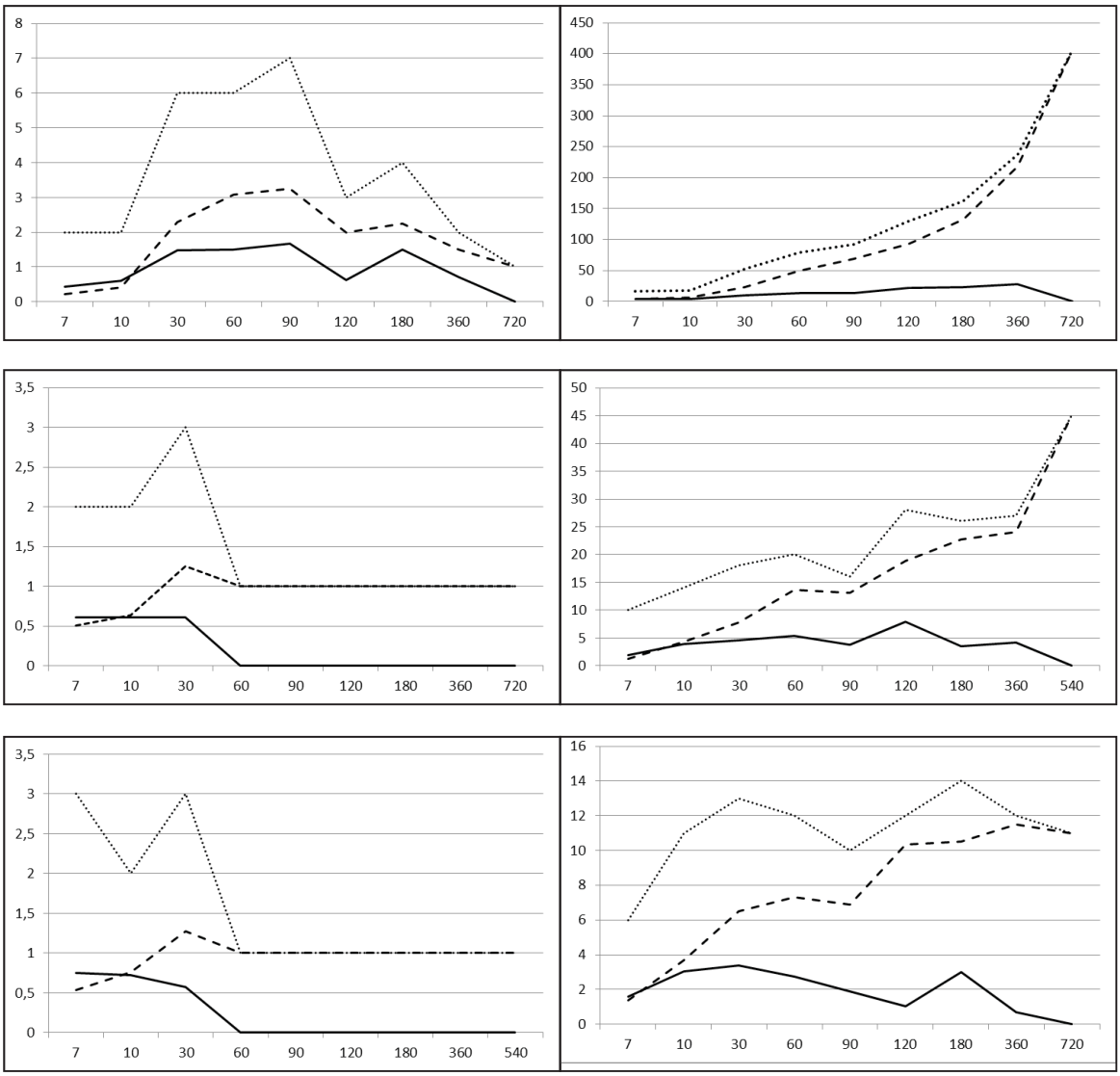


Abbildung 3: Anzahl der Cluster bei unterschiedlich langen Zeitscheiben (CPM (links), LC (rechts), Asterisk (oben), OpenSimulator (Mitte), Dojo (unten), gepunktet = Maximum, gestrichelt = Durchschnitt, durchgängig = Standardabweichung)

Insgesamt zeigen die Ergebnisse deutlich mit dem komplexeren Indikator Overlap-

Coverage aber auch dem einfacheren Indikator der Anzahl von Clustern, wie in Abbildung 3 für alle Communities mit beiden Verfahren, dass die von uns durch qualitative Beobachtung ermittelte Optimallänge von zwei bis drei Monaten [2] für die Communities durch die Indikatorenmodelle bestätigt wird. Auch due Varianz bei Clustergrößen deckt die Annahme, jedoch kann Clustergröße wie oben erwähnt von anderen Faktoren beeinflusst sein.

Bezogen auf die tatsächlichen Personen und deren Rollen im Netzwerk zeigt sich die Tendenz, dass wichtige überlappende Mitglieder von Subgruppen wie Innovatoren sich gut in den Zeitscheiben der Längen von 7 Tagen und 30 Tagen verteilen. So taucht z.B. im OpenSimulator-Netzwerk Christa C.L., die als Wissenschaftlerin neue Ideen wie z.B. Hypergrids (Hyperlinks in 3D Simulationswelten) einbringt, am häufigsten in Zeitscheibenlängen von 7 Tagen auf. Längere Zeitscheiben von etwa einem halben Jahr beinhalten tendenziell häufig Personen mit repräsentativen Rollen, wie z.B. Stefan A. und Dirk H. in der OpenSimulator Community. Auch wenn sich Rollen aufgrund vielfältiger Einflußfaktoren schwer einer bestimmten Länge von Zeitscheiben zuordnen lassen, so lässt sich zumindest eine Tendenz ablesen, dass in dem Fall Längen zwischen einem und drei Monaten einen repräsentativen Schnitt für die untersuchten Communities darstellen.

6 Diskussion

Unsere Untersuchung verdeutlicht, dass die Variation der Länge von Zeitscheiben für die Analyse von produktiven Online-Communities einen systematischen Effekt auf die Identifizierung von Subgruppen hat. Die eingesetzten Verfahren CPM und LC haben jeweils unterschiedliche Definitionen von Communities und verhalten sich demnach auch unterschiedlich bei unserer Untersuchung. Der beobachtete Effekt lässt sich dennoch bei beiden Verfahren feststellen.

Allerdings spiegeln sich nach abschließender Beurteilung unserer Verfahren auch weitere Effekte als Einflussfaktoren für das Clustering von dynamischen Netzwerken in den Resultaten. So ist insbesondere bei produktiven Gemeinschaften davon auszugehen, dass Ereignisse wie Deadlines, Meilensteine, Beginn von Unterprojekten, Teilung usw. eine bedeutende Rolle für die inhärente „Zeit“ bzw. der Geschwindigkeit einer Community. So können sich gut während einer Deadline-Phase beispielsweise ähnlich viele Subgruppen bei einem kürzeren Zeitraum identifizieren, die bei einer gewöhnlichen Phase erst bei einer längeren Zeitscheibe identifiziert werden würden. Daher können wir nach der systematischen Betrachtung davon ausgehen, dass die Ermittlung der korrekten Länge von Zeiträumen für die Beobachtung und Analyse

von Communities über Zeit mit Blick auf Ergebnisse vieler Verfahren des Clustering und der Community Detection von großer Bedeutung ist. Auch wenn Forscher durch implizite Annahmen im Zuge von Recherchen und allgemeinen Beobachtungen über die Produktionstempi ihrer untersuchten Gemeinschaften treffen, die intuitiv passen, so wird sich die Frage spätestens im Kontext von zunehmend anfallenden, großen Datenmengen stellen, wie sie derzeit in der „Big Data“ Diskussion bereits skizziert werden.

Literatur

[1] Wasserman, S. & Faust, K. Social Network Analysis: Methods and Applications. CUP, 1994.

[2] Zeini, S. & Hoppe. “Community Detection” als Ansatz zur Identifikation von Innovatoren in Sozialen Netzwerken. K. Meißner, M. Engelien (Hrsg.), Virtual Enterprises, Communities and Social Networks, Dresden: TUDpress. 2010.

[3] Burt, R. S. Structural holes and good ideas. American Journal of Sociology, 110(2), S.349–399, September, 2004.

[4] Fleming, L. & Waguespack, D. M. Brokerage, boundary spanning, and leadership in Open Innovation communities. Organization Science 18 (2), S.165–180, March–April 2007.

[5] Stark, D. & Vedres, B. Structural Folds: Generative disruption in overlapping groups, American Journal of Sociology, January 2010, 15(4).

[6] Palla, G., Derényi, I., Farkas, I., & Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):S.814-8, 2005.

[7] Zeini, S., Göhnert T., Krempel L., and Hoppe H. U. The Impact of Measurement Time on Subgroup Detection in Online Communities. The 2012 IEEE/ ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), 2012.

[8] Falkowski, T. Community Analysis in Dynamic Social Networks. Sierke, 2009.

[9] Bernard, H.R., Killworth, P. D., & Sailer L. Informant accuracy in social network research IV: A comparison of clique-level structure in behavioral and cognitive data. Social Networks, 2:S.191–218. 1980.

[10] Stegbauer, Ch. & Rausch, A. Moving Structure: Möglichkeiten der positionalen Analyse von Verlaufsdaten am Beispiel von Mailinglisten. In: U. Serdült, and V. Täube (Eds), Applications in Social Network Analysis. „Züricher Politik- und Evaluationsstudien“, 2005. S.75–98,

[11] Trier, M. & Bobrik, A. Analyzing the Dynamics of Community Formation using Brokering Activities. Proceedings of the Third Communities and Technologies Conference, Michigan, Springer, 2007.

[12] Wang, Q. & Fleury, E.Mining time-dependent communities. LAWDN - Latin-

American Workshop on Dynamic Networks. 2010
[13] Palla, G., Barabási, A.L., & Vicsek, T. Quantifying social group evolution, in Nature 446, S.664–667, 5. April 2007.
[14] Greene, D., Doyle, D., & Cunningham, P. Tracking the Evolution of Communities in Dynamic Social Networks, International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp.176–183, 2010.
[15] Budka,M., Musial, K. & Juszczyszyn, K. Predicting the evolution of social networks: Optimal time window size for increased accuracy, 2012 ASE/ IEEE International Conference on Social Computing (SocialCom 2012), 2012, S. 21–30.
[16] Coscia, M., Giannotti, F. & Pedreschi, D. A classification for community discovery methods in complex networks. Statistical Analy Data Mining, vol. 4: S.512–546, 2011.
[17] Ahn, Y. Y., Bagrow, J. P., & Lehmann, S. Link communities reveal multi-scale complexity in networks, Nature, 466 (7307), S.761–764, 2010.
[18] Zeini, S., Malzahn, N., &Hoppe, H. U., 2009, Entstehung von Innovation in Open Source-Netzwerken am Beispiel von Open Simulator, in: Meißner, K./Engelien, M. (Hrsg.), Virtuelle Organisation und Neue Medien 2009. Dresden: TUDpress.