Clustering of dynamic graphs

Hauptseminar Networkvisualisation

Moritz Hamann

Abb. 1. In den Wolken: Vancouver von Cypress Mountain. Auf der ersten Seite dürfen keine Grafiken außer dieser optionalen Aufmachgrafik (Teaser) abgebildet sein.

Kurzbeschreibung—This paper presents a summary about various techniques to detect and identify densely connected nodes in a graph, so called clusters. In the first part, we introduce the concept of clusters for static graphs alongside their main properties. For dynamic graphs with time varying edge connections, these cluster may be subject to change with every time step. Therefore additional characteristics have to be introduced.
The second part describes two methods to detect, identify and track cluster in a dynamic graph. A common solution for this problem is the clustering of a static graph at each time step, and the identification of the same clusters over multiple time steps. A method is presented to track these clusters, which is independent of the underlying static graph clustering algorithm. Furthermore, we describe an extension of the k-clique percolation algorithm to dynamic graphs.
Finally, the clique percolation algorithm is applied to two different real world networks, which yields interesting result about group dynamics, with regards to the correlation of various group properties.

1 Motivation

Das mathematische Konzept der Graphen ist ein essentielles Modellierungswerkzeug in der Informatik. Nicht nur lassen sich damit verschiedenste Datenstrukturen anschaulich dargestellen, sondern mit ihrer Hilfe lassen sich auch jegliche Beziehungen zwischen einzelnen Objekten oder Prozessen in einem Netzwerk modellieren und untersuchen. Aus diesem Grund sind sie heutzutage nicht nur in der klassischen Informatik sowie in der Mathematik zu finden, sondern haben auch in vielen anderen Wissenschaften ihren Einzug erhalten. So werden sie genutzt um die Gruppendynamik in biologischen Netzwerken zu beschreiben, dienen als Kontrollalgorithmen für Multiagenten Systeme [?] und beschreiben Kommunikationsmuster in sozialen Netzwerken.

Um die Eigenschaften sehr großer Netzwerke analytisch untersuchen zu können, werden häufig Zufallsgraphen nach dem Model von Edgar Gilbert (nachweise?) oder Erdos-Renyi verwendet. Diese Graphen haben die Besonderheit, dass die Wahrscheinlichkeit für eine Verbindung zwischen je zwei Knoten im gesammten Netzwerk konstant ist. Dadurch ensteht ein gleichmäßiger Graph, dessen Gradverteilung binomial verteilt sind, und somit die meisten Knoten die gleiche Anzahl an Kanten haben. Mit Hilfe der Wahrscheinlichkeitstheorie, lassen sich nun die Eigenschaften dieser Graphen auch für eine sehr hohe Anzahl an Knoten bestimmen und untersuchen.

Institut für Visualisierung und Interaktive Systeme, Universität Stuttgart. Sommersemester 2013.

Allerdings haben Untersuchungen von realen Netzen gezeigt (nachweis), dass sich diese in den meisten Fällen von Zufallsgraphen unterscheiden. Reale Netzwerke sind häufig sogenannte Skalenfreie Netze (im Englischen 'Scale-free networks'), in denen die Anzahl der Verbindungen pro Knoten nicht binomial verteilt ist, sonder nach einem Potentzgesetz. Dadurch einsteht ein Netzwerk, in dem einzelne wenige Knoten eine große Anzahl an Verknüpfungen aufweisen, doch die Mehrzahl der Knoten weniger stark verknüpft ist. Weiterhin ist die Kantenverteilung zwischen den Knoten auch lokal sehr inhomogen, so dass sich Teilgraphen ausbilden, deren Knoten untereinander sehr stark bis komplett verknüpft sind, während sie nach Außen weniger Verbindungen aufweisen. Diese Teilgraphen werden auch 'Cluster' genannt.

Diese Cluster spielen in viele Anwendungsgebieten eine wichtige Rolle. Betrachtet man zum Beispiel den Graph der Freundschafts Beziehungen in einem sozialen Netzwerk, lassen sich mithilfe von Angaben anderer Benutzer, sowie lokaler Cluster, unter anderem Rückschlüße auf gemeinsame Interessen, Wohnorte oder Freunde der einzelnen Benutzer schließen. Diese Informationen bieten dem soziale Marketing eine bis vor kurzem unbekannte Menge an Möglichkeiten ihre Produkte zielgerichteter und persönlicher zu vermarkten. Aber auch in dynamischen Graphen, in denen Knoten und Kanten sehr häufig wechseln können, ist es wichtig Cluster zu finden. Betrachtet man den Verbindungsgraph eines dezentralen, kabellosen Ad-Hoc Netzwerks, so lassen sich mit Hilfe von Clustering Verfahren Teilnetze

finden die geographisch Eine zusätzliche Herrausforderung zur eigentlichen Clusteranalyse ist hierbei allerdings

Übersicht

Diese Arbeit gibt einen Überblick über die Eigenschaften dieser Cluster, ihr

2 GRUNDLAGEN

2.1 Formale Definition eines Clustern

In diesem Kaptiel wird versucht eine formale Definition zu geben, was ein einen Cluster in Graphen zu. Während die gewünschten Eigenschaften

2.1.1 Eigenschaften von Clustern

Der Artikel von S.E. Schaeffer [?] bietet eine umfassende Zusammenfassung über bisherige Clustering Verfahren, und versucht eine Defintion für Cluster anhand gewünschter Eigenschafter zu geben.

Betrachtet man den Teilgraphen Ω eines kompletten Graphen Υ, so müssen mehrere Bedingungen erfüllt sein, damit Ω ein Cluster wird. Natürlich sollten alle Knoten aus Ω verbunden sein, was bedeutet das zwischen jedem Paar aus Knoten u und v mit $u,v \in \Omega$ ein Pfad existiert. Ist dies nicht der Fall so ist der gesammte Graph nicht verbunden, und das Clustering sollte auf den einzelnen Teilgraphen gesondert betrachtet werden. Weiterhin sollte der Teilgraph Ω eine hohe Kantendichte zwischen seinen Knoten aufweisen. Dies ist der Fall, wenn mehere Pfade zwischen den Knoten aus Ω existieren, so dass jeder Pfad möglichst wenig Elemente aus $\Upsilon \setminus \Omega$ enthält.

Der Grad $d(v)$ eines Knoten v ist definiert als die Anzahl der Kanten zu anderen Knoten im Graphen Υ. Ist nun $v \in \Omega$ wobei Ω wieder ein Teilgraph von Υ ist, so lässt sich der Grad in einen externen und internen Teil unterscheiden. Dabei ist der interne Grad die Anzahl der Kanten von v zu anderen Knoten aus Ω, der externe Grad die Anzahl der Kanten von v zu allen anderen Knoten aus $\Upsilon \setminus \Omega$. Dabei gilt:

$$d_{int}(v,\Omega) = |\Gamma(v) \cap \Omega| \tag{1}$$
$$d_{ext}(v,\Omega) = |\Gamma(v) \cap (\Upsilon \setminus \Omega)| \tag{2}$$
$$d(v) = d_{int}(v,\Omega) + d_{ext}(v,\Omega) \tag{3}$$

wobei $\Gamma(v)$ die direkten Nachbarn von v sind. Eine Eigenschaft, die den Teilgraphen Ω zu einem Cluster werden lässt, ist ein hohes Verhältniss von internem zu externem Grad für alle Knoten $v \in \Omega$, d.h. die Knoten eins Cluster haben untereinander wesentlich mehr Verknüpfungen als zu den Knoten des restlichen Graphen.

Eine weiteres Kriterium für die Qualität eines Clusters ist die sogenannte interne Clusterdichte. Die allgemeine Dichte eines Graphen $\Upsilon = (V,E)$ mit der Knotenmenge V und der Kantenmenge E ist definiert als das Verhältniss der Summer aller Kanten durch die Anzahl aller möglichen Kanten in Graph:

$$\rho(\Upsilon) = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)} \tag{4}$$

Somit lässt sich die interne Clusterdichte eines Clusters Ω definieren als

$$\rho_{int}(\Omega) = \frac{|\{\{u,v\}|u,v \in \Omega\}|}{|\Omega|(|\Omega|-1)} \tag{5}$$

wobei $\{u,v\}$ eine Kante zwischen den Knoten u und v darstellt. Die fehlende 2 im Zähler im Vergleich zur allgmeinen Graphendichte ist dadurch zu erklären, dass $\{u,v\}$ und $\{v,u\}$ zwar die gleiche Kante darstellen, aber zwei unterschiedliche Elemente sind, wodurch die Anzahl der Kanten in $\{\{u,v\}|u,v \in \Omega\}$ verdoppelt wird.

Zusätzlich zur internen Clusterdichte, existert noch eine sogenannte externe Clusterdichte zwischen verschiedenen Clustern Ω_i eines Graphen Υ. Sie ist definiert als das Verhältniss der Summe aller Kanten zu der Summe aller möglichen Kanten zwischen den verschiedenen Clustern Ω_i:

$$\rho_{ext}(\Upsilon|\Omega_1 ... \Omega_k) = \frac{|\{\{v,u\}|v \in \Omega_i \, u \in \Omega_j, i \neq j\}|}{|V|(|V|-1) - \sum_{l=1}^{k} |\Omega_k|(|\Omega_k|-1)} \tag{6}$$

wobei $|\Omega_i|$ die Anzahl der Knoten des Teilgraphen Ω_i darstellt. Im Allgmeinen kann man sagen, dass für ein gutes Clustering auf einem Graph Υ gelten sollte:

$$\rho_{int}(\Omega_i) > \rho(\Upsilon) > \rho_{ext}(\Upsilon|\Omega_1 ... \Omega_k) \quad \forall i = 1...k \tag{7}$$

Abb. 2 zeigt drei verschiedene Cluster unterschiedlicher Qualität im Vergleich. Dabei representieren die schwarz hervorgehobenen, beliebig gewählten Teilgraphen jeweils einen Cluster. Der linke Cluster weist eine sehr hohe intere Dichte auf, und hat kaum Kanten mit Knoten ausserhalb. Daher ist Qualität dieses Clusters sehr hoch. Der mittlere Cluster hat zwar die gleiche Anzahl an internen Kanten, weist aber im Gegensatz zum Linken eine wesentliche höhere Kantenanzahl zu Knoten ausserhalb des Cluster auf. Zwar hat der rechte Cluster nur wenige Kanten nach aussen, allerdings ist aber die Kantendichte innerhalb des Clusters minimalst, was ihn zum schlechtesten Cluster der drei macht.

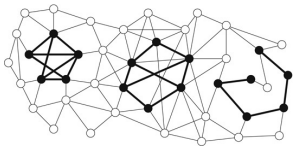

Abb. 2. Drei verschiedene Cluster unterschiedlicher Qualität [referenz]

2.1.2 k-Cliques

Einen Teilgraphen Ω mit k Knoten nennt man k-Clique, falls alle k Knoten dieses Teilgraphen direkt miteinander verbunden sind, und Ω somit vollständig ist. Die entstehende Topologie des Teilgraphen Ω ist natürlich vom Parameter k abhänging. Abb. 3 zeigt k-Cliques für verschiedene Werte von k.

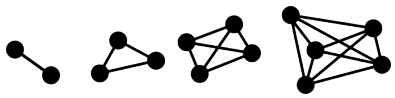

Abb. 3. Topologien für k-Cliques mit k=2,3,4,5

Da jede k-Clique vollständig ist, ist die in Kapitel 2.1.1 definierte interne Graphdichte mit $\rho_{int}(\Omega) = 1$ maximal. Somit stellen k-Cliques theoretisch gute Kandidaten füer Cluster da. Beschränkt man sich bei der Clusterfindung auf einzelne k-Cliques, werden in den meisten Fällen viele Cluster nicht gefunden, da Forderung an einen vollständig verknupften Cluster zu restrektiv ist.

2.2 Dynamische Graphen

Ein klassicher Graph $\Omega = (V,E)$ ist eine Kombination einer Menge an Knoten V und Kanten E zu einem bestimmten Zeitpunkt t. Für viele Anwendungen ist es aber essentiell das Netzwerk über einen Zeitraum mit mehreren Zeitschritten t_i zu betrachten und untersuchen.

Das Konzept der *dynamischen Graphen*[Quelle] stellt die zeitliche Veränderung eines Graphen als geordnete Folge von statischen Teilgraphen für jeden Zeitpunkt $t = 1,...,n$ da:

$$\Upsilon = \{\Omega_1 = (V_1,E_1), \Omega_2 = (V_2,E_2), ..., \Omega_n = (V_n,E_n)\} \tag{8}$$

wobei Ω_i der Konfiguration des dynamischen Graphen Υ zum Zeitpunkt i entspricht. Da alle V_i, E_i für alle Werte von i unabhängig sind, ist es mögich das zu jedem Zeitschritt sowohl Kanten als auch Knoten hinzugefügt oder verschwinden können.

3 CLUSTERING IN STATISCHEN GRAPHEN

Viele Clustering Verfahren für dynamische Graphen basieren auf Verfahren zur Clusterung klassischer, statischer Graphen. Da dynamische Graphen aufgrund der Zeitanhängigkeit weitere Komplexität einführen, ist es sinnvoll als erstes diese Clusteringmethoden für statische Graphen zu betrachten. In diesem Kaptiel wird versucht die Vielzahl verschiedener Verfahren anhand ihrer grundsätzlichen Eigenschaften in verschiedene Kategorien zu klassifizieren, sowie eine kurze Erklärung ihrer Funktionsweise zu geben.

Nach Schaeffer[] lassen sich Clustering Verfahren grundsätzlich in lokale oder globale Verfahren einteilen. Hierbei werden die Verfahren entweder global auf den ganzen Graph angewendet, oder nur lokal auf einen Teilgraphen. Entsprechend benötigen globale Verfahren Informationen üeber die Topologie des gesammten Graphen, während bei bei lokalen Verfahren nur rekursiv die Nachbarschaft eines einzelnen Knotens betrachtet wird, somit auch Netzwerke betrachtet werden können, die a priori nicht komplett bestimmt sind.

Entsprechend bieten lokale Verfahren eine bessere Skalierbarkeit als Globale, falls das Clustering nur auf einen Teilgraphen angewendet werden soll, da die Topologie des restlichen Graphen nicht bekannt sein muss. Weiterhin haben diese Verfahren den Vorteil, dass das Clustering nur von der lokalen Struktur anhängt und eine lokale Änderung im Graphen auch nur das Clusterung in deren Umgebung beeinflusst. Deshalb eignen sich lokale Verfahren in Anwendungen, bei denen die Nachbarschaft einzelner Knoten schnell und häufig untersucht werden soll.

Tabelle 1. Einteilung der Clusteringverfahren für statische Graphen

	Globale Verfahren	Lokale Verfahren
Top-Down	Spektrale Methoden Random Walk Methoden Maximaler Fluss Methoden	
Bottom-Up	Modularitätsoptimierung Nächste Nachbarn Methode	CPM

Globale Verfahren

Die Familie der globalen Verfahren lässt sich noch mal in sogenannte *Top-Down*, sowie *Bottom-Up* Methoden[Schaeffer] unterteilen. Bei Top-Down Verfahren wird der Graph rekursiv anhand verschiedener Kriterien in immer kleinere Methoden unterteilt, während bei Bottom-Up Verfahren viele kleinere Cluster sukzessive zu grösseren zusammen gefasst werden, bis das Clustering einem Abbruchkriterum genügt.

Ein Vertreter der Top-Down Methoden, sind die sogenannten *Spektralen Methoden*. Sie basieren auf den Eigenwerten und Vektoren der Laplace-Matrix des Graphen. Die Laplace Matrix eines Graph $\Omega = (V,E)$ ist definert als

$$L(\Omega) = D(\Omega) - A(\Omega) \tag{9}$$

wobei $D(\Omega) = diag(\{d(v_1),...,d(v_n)\})$ die Degreematrix von Ω ist[Buch üeber graph theorie]. $A(\Omega)$ ist die sogenannte Adjacency Matrix von Ω, und definiert als

$$[A]_{ij} = \begin{cases} 1 & \text{falls zwischen } v_i \text{ und } v_j \text{ eine Kante besteht} \\ 0 & \text{sonst} \end{cases} \tag{10}$$

Da die Laplace Matrix eine symmetrische, positiv semidefinite Matrix ist[Buch], sind alle Eigenwerte $\lambda_i \geq 0$ und die korrespondierenden Eigenvektoren bilden ein orthogonales System. Ordnet man die Eigenwerte in aufsteigener Reihenfolge $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, so folgt aus der positiv Semidefinitheit von L dass $\lambda_1 = 0$. Ist weiterhin $\lambda_2 > 0$ so exisitieren keine isolierten Teilgraphen im kompletten Graph[Buch]. Die Algorithmen der spektralen Clustering Methoden benutzen typischerweise die Komponenten des Fideler Vektors -den Eigenvektor von λ_2- um die Knoten eins Graphen zu vergleichen und clustern[Schaeffer].

Ein weitere Gruppe von Methoden die den Top-Down Ansatz verfolgen, sind die sogenannten *Random Walk* oder *Markov Ketten Methoden*. Diese Verfahren basieren auf einem zufälligen Weg ξ fester Länge durch den Graph. Dabei ξ, ausgehend von einem Startknoten v_{start}, iterativ über eine zufällige Auswahl der direkten Nachbarn des jeweils aktuellen Knoten aufgebaut. Aufgrund der höheren Dichte in Clustern, wird der Weg ξ die Knoten des selben Clusters wie v_{start} häufiger besuchen als Knoten ausserhalb des Clusters. Abb. 4 zeigt einen Beispielgraph mit zwei Clustern. Wird ein Weg ausgehend von einem weissen Knoten aufgebaut, ist es für einen zufälligen Weg nur auf einem Knoten möglich den linken Cluster zu verlassen, und somit werden die weissen Knoten des linken Clusters mit einer höheren Wahrscheinlichkeit besucht.

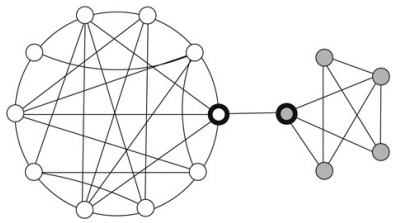

Abb. 4. [Schaeffer]

Für gewichtete Graphen eignen sich auch die *Maximaler Fluss Methoden* um ein Clustering duchzuführen. Sie gehören ebenfalls zu der Gruppe der Top-Down Verfahren, und versuchen einen minimalen Schnitt[Schaeffer] des Graphen zu finden. Dazu werden Strömungsberechnungen auf dem Graphen durchgeführt, wobei ein Fluss zwischen zwei Knoten nur über eine Verbindungskante bestehen kann. Da aufgrund des *Minimum cut, maximum Flow Theorms* der Schnitt eins Graphen beim maximalen Fluss am geringsten ist, lässt sich dieser über den maximalen Fluss auf den Kanten des Graphen finden. Flake et al.[Quelle] berechneten hierzu den Fluss mithilfe künstlich hinzugefügter Senken, und erzeugten daraus einen Minimum-cut Tree [Quelle] um das Clustering auf einem Graphen durchzuführen.

Beispiele für Methoden, die einen Bottom-Up Ansatz werden sind unter anderem die *Modularitätsoptimierung* welche in Kaptiel 3.1 genauer behandelt wird. Eine weitere Vertreter ist die sogenannte *Nächste Nachbarn Methode*[Schaeffer], welche häufig auch bei allgmeinen Klassifizierungsproblmen angewendet wird. Um die Nächste Nachbarn Methode auf Graphen anwenden zu können, muss eine Ähnlichkeit zwischen zwei Knoten definiert werden. Eine solche Ähnlichkeit kann z.B. anhand von vorhandenen Metadaten jedes Knoten erfolgen, oder anhand der Schnittmenge der direkten Nachbarn zweier Knoten. Im ersten Schritt wird füer jeden Knoten des Graphen derjenige Nachbar gesucht, füer welchen die Ähnlichkeit am grössten ist. Diese beiden Knoten bilden nun einen Cluster. In den darauffolgenden Schritten, werden nun die bereits gefunden Cluster auf Ähnlichkeit untersucht, und zu grösseren Cluster zusammengefügt, bis das Clustering beendet ist.

3.1 Modularitätsoptimierung

Das Verfahren der Modularitätsoptimierung ist ein weiterer Vertreter der Bottom-Up Ansätze. Es versucht eine Paritionierung des Graphen zu finden, für die die Modularität maximal wird. Hierbei ist die Modularität ein Maß, in wie fern die gewählte Partitionierung $C(\Omega)$ eines Graphen Ω einem guten Clustering entspricht. Mathematisch ist die Modularität $Q(C)$ definiert als[paper]:

$$Q(C) = \frac{1}{2|E|} \sum_{u,v \in V} \left[A_{uv} - \frac{k_v k_u}{2|E|} \delta\left(c(u), c(v)\right) \right] \tag{11}$$

wobei $A_{uv} = 1$ falls eine Kannte zwischen u und v existiert, ansonsten 0. $k_u = d(u)$ ist der Grad des Knoten, und $c(u)$ diejenige Parition die den Knoten u enthält. Weiterhin ist $\delta\left(c(u), c(v)\right)$ falls $c(u) = c(v)$, ansonsten 0. Somit representiert die Modularität die Summe aller Kan-

ten innerhalb einer Partition minus der Anzahl der Kanten, falls diese zufällig verteilt wären.

3.2 Clique Percolation Method (CPM)

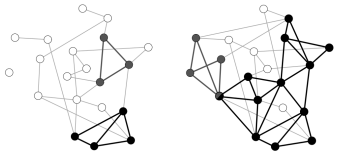

Abb. 5. aus paper

4 Clustering in dynamischen Graphen

4.1 Evolution eins Clusters

4.2 Erweiterung der CPM

4.3 Time step Clustering

5 maybe visualization of dynamic graphs

6 Group Evolution (results)

7 Conclusion

7.1 Abbildungen und Tabellen

Alle Abbildungen (siehe Abb. **??**) und Tabellen (Tabelle **??**) sollten zentriert sein (`\centering`). Abbildungen über beide Textspalten (Abb. 6) können mit `\begin{figure*}...\end{figure*}` eingefügt werden.

7.2 Referenzen

Literaturangaben wie beispielsweise Levoy [1] werden mit Hilfe von BibTeX erzeugt. Dazu werden die Referenzen in die Literaturliste (hier *literatur.bib*) eingetragen und entsprechend mit `\cite` referenziert.

7.3 LaTeX-Übersetzung

Die LaTeX-Datei kann mit *latex* oder *pdflatex* übersetzt werden. Dabei ist zu beachten, dass für die Übersetzung mit *latex* die Grafiken in Postscript (eps) vorliegen, für *pdflatex* entsprechend als jpg, png oder pdf. Der Ablauf ist dabei der folgende:

1. `pdflatex <quelldatei.tex>`
2. `bibtex <quelldatei>`
3. `pdflatex <quelldatei.tex>` (evtl. mehrfach)

Alternativ kann auch das mitgelieferte Makefile verwendet werden.

Literatur

[1] M. Levoy. *Display of Surfaces from Volume Data.* PhD thesis, University of North Carolina at Chapel Hill, 1989.

[2] M. Strengert, T. Klein, R. Botchen, S. Stegmaier, M. Chen, and T. Ertl. Spectral volume rendering using GPU-based raycasting. *The Visual Computer*, 22(8):550–561, 2006.

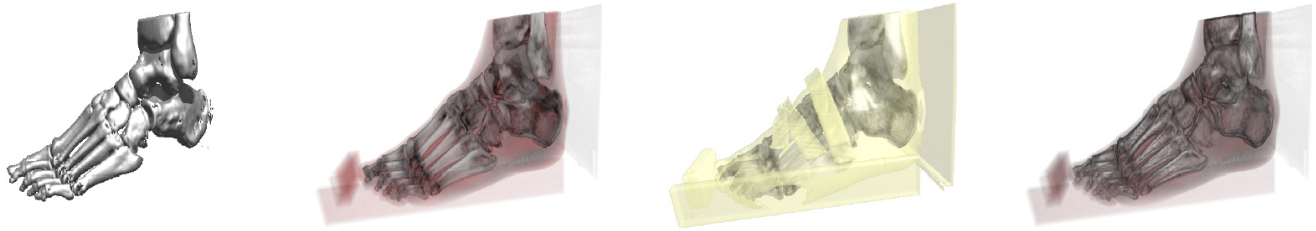

Abb. 6. Illustration über beide Textspalten hinweg. Auch Illustrationen müssen entsprechend den Quellen gekennzeichnet werden. [2]