

Universität Stuttgart - Institut für formale Methoden der Informatik

Hauptseminar:

Inside Google - Algorithmen für Suchmaschine

Web Community

Changsheng Qian

22. Januar 07

Betreuer: Prof. Dr. Javier Esparza

Inhaltsverzeichnis

1.	Einführung	2
2.	Relevante Definitionen	4
2.1	Maximum-Flow und Minimal-Cuts	4
2.2	Ford-Fulkerson Algorithmus	6
3.	Die Web Community	7
3.1	Übertragung der MaxFlow und MiniCut auf Web Community	8
3.2	Exact-Flow Community	9
3.3	Approximate Community	10
4.	Experimentelle Ergebnisse	14
4.1	Support Vector Maschine Community	14
4.2	The Internet Archive Community	15
4.3	The Ronald Rivest Community	15
5.	Zusammenfassung und Fazit	16
6.	Referenzen	17

Abbildungsverzeichnis

1.	Teil der Webgraphen	3
2.	Beispiel eines Maximum-Flow Problems	4
3.	Ein- und Ausgehende Links	7
4.	Trennen der Web Community von Webgraphen	8
5.	Focused Community crawling, entstehenden Graphen	10
6.	Lokalisierung eines minimalen Schnitts mit virtueller Senke	11
7.	Der Focused Crawl Algorithmus	12
8.	Beispiel der Verwendung des Focused Crawl Algorithmus	13

1. Einführung

Das World Wide Web enthält die Informationen, die sich schwierig komplett erfassen lassen. Es handelt sich um ein selbstorganisiertes Medium, in dem die Seiten thematisch zusammengehören und miteinander verlinkt sind. Das Internet hat ein exponentielles Wachstum seit seiner Erfindung: Im Jahr 2000 waren bereits 10^9 Webseiten von Suchmaschinen indexiert. Mehrere tausend Seiten werden täglich erstellt. Wegen verschiedenen Gründen z.B. temporärer Seiten oder illegalen Inhalten verschwinden auch bereits erstellte Seiten.

Die Suchmaschinen bieten die Möglichkeiten, Informationen thematisch zu finden. Keine Suchmaschine deckte bis 2002 mehr als 16% des Webs ab. Die Vereinigung der elf größten Websuchmaschinen (Google, Yahoo etc.) erreicht auch nur die Hälfte des gesamten Webs.

Die Suchmaschinen können die Webseiten mit Hilfe von Spider/Crawler auffinden. Spider beginnt von einer Seite und springt zu der nächsten bis alle erreichbaren Seiten indexiert sind. Dadurch bilden die Seiten, die untereinander stark verlinkt und thematisch verwandt sind, eine Webgemeinschaft (Web Community), damit die Suchmaschinen in einer sich auf einem Thema konzentrierende Seitengemeinschaften suchen um möglichst eine effiziente Lösung zurückzugeben.

Das Internet wird hier als Web-Graph betrachtet, bei dem die Webseiten als Knoten, die Hyperlinks als Kanten dargestellt werden. Dabei ist die Web Community ein Teilgraph des Webgraphen. Somit lässt sich das Auffinden einer Web Community im Internet auf

Graphenproblem reduzieren.

In diesen Bericht wird eine gesuchte Web Community folgendermaßen definiert.

Eine Web Community besteht aus Mitgliedern, die mehr Links zu anderen Mitgliedern als zu Nichtmitgliedern.

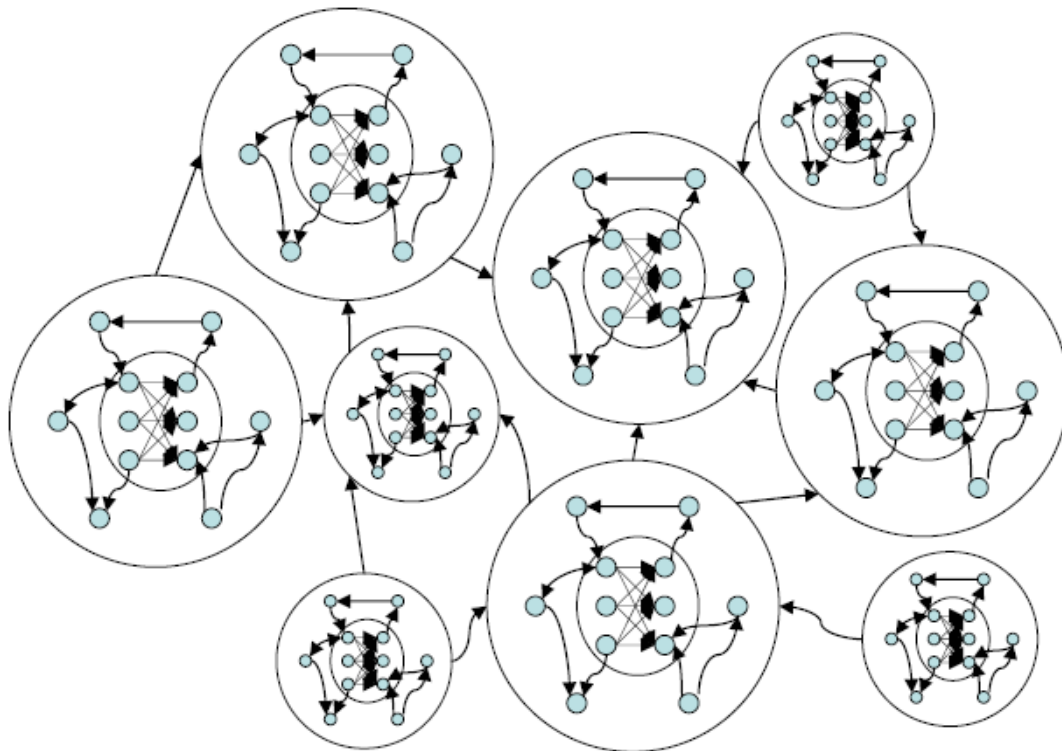


Abbildung 1:

Teil der Webgraphen, wobei Web Communities (in Kreisen) deutlich stärker untereinander verbunden sind.

Quelle: [NEC Research Institute](#) [5]

In meinen Bericht werden die Methoden vorgestellt, um eine solche Web Community zu finden, sowie der Such-Algorithmus. Am Anfang werden einige Definitionen, die für den Lösungseinsatz erforderlich sind, eingeführt. Zum Schluss werden die experimentelle Ergebnisse, die das Verfahren praktisch funktioniert bewiesen haben, angegeben.

2. Relevante Definitionen

2.1 Maximum-Flow und Minimal-Cuts

Das Maximum-Flow Problem beschäftigt sich mit der maximalen Menge, die von einer Quelle s zu einer Senke t gelangen kann. Dabei stellt man sich einen Warentransport vor, bei dem die Produkte von s nach t transportiert werden. Es existieren mehrere Zwischenstationen, die die Produkte durchlaufen müssen. Zwischen 2 beliebigen Stationen ist die Kapazität beschränkt. (Es darf nur eine bestimmte Menge von Produkten transportiert werden.)

Auf dem Weg darf die Kapazität nicht überschritten werden. Weniger ist jedoch möglich.

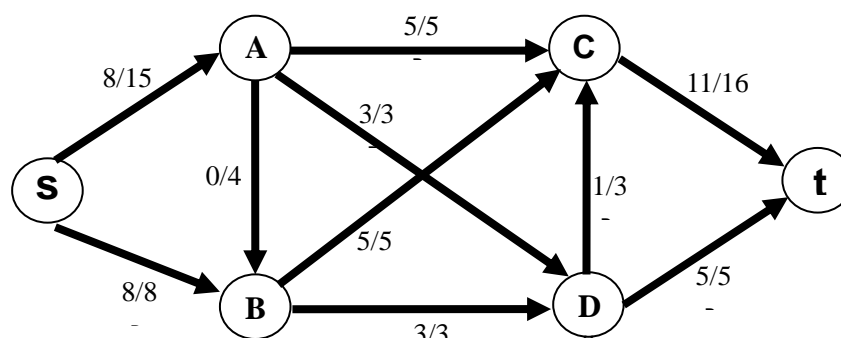


Abbildung 2:

**Beispiel des Maximum-Flow Problems
und dessen möglichen MaxFluss Weg**

Wir betrachten die obige Abbildung als einen gerichteten Graphen mit zwei ausgezeichneten Knoten s (Quelle) und t (Senke). Es existiert eine Kapazitätsfunktion c , die jeder Kante (u,v) eine Kapazität $c(u,v)$ aus dem Bereich der nicht negativen reellen Zahlen zuordnet.

Ein s-t Fluss muss folgende Bedingungen erfüllen:

1. Kapazitätsbeschränkung: Der Fluss einer Kante erlaubt maximal die Kapazität der Kante.
2. Flusserhaltung: Aus jedem Knoten muss genauso viel hineinfließen wie herausfließen.

Ein Fluss F ist genau dann ein Maximumfluss von s nach t , wenn er keinen erweiternden Weg erlaubt.

Offensichtlich sehen wir, dass der MaxFluss obiges Beispiel 16 ist.

Max-Flow-Min-Cut Theorem [1]

Sei $G=(V,E)$ ein gerichteter Graph und jede Kante (u,v) hat eine Kapazität $c(u,v)$.

Seien $s,t \in V$ zwei unterschiedliche Knoten, die die Quelle und Senke bezeichnen.

$A,B \subset V$ werden Schnitt genannt falls:

- $s \in A$ und $t \in B$
- $A \cap B = \emptyset$ und $A \cup B = V$

Die Kapazität von Cut (s,t) ist

$$c(s,t) = \sum_{u \in A, v \in B \mid (u,v) \in E} c(u,v).$$

die Summe der Kapazität aller Kanten, die an Schnitt A und B hängen.

Die MinCut ist die minimale Kapazität von Cut.

Wir nehmen die Abbildung 2 noch mal als Beispiel. Der Graph wird bei der MinCut wie folgt geschnitten:

$$S = \{s,A,B\}, T = \{C,D,t\}$$

Die zugehörige Kapazität ist :

$$c(s,t) = c(A,C) + c(A,D) + c(B,C) + c(B,D) = 5 + 3 + 5 + 3 = 16.$$

Der Wert der MinCut ist in einem Netzwerk gleich der MaxFluss.[2]
 Der Algorithmus von Ford und Fulkerson dient der Berechnung eines maximalen Flusses in einem Netzwerk.

2.2 Ford-Fulkerson Algorithmus[1]

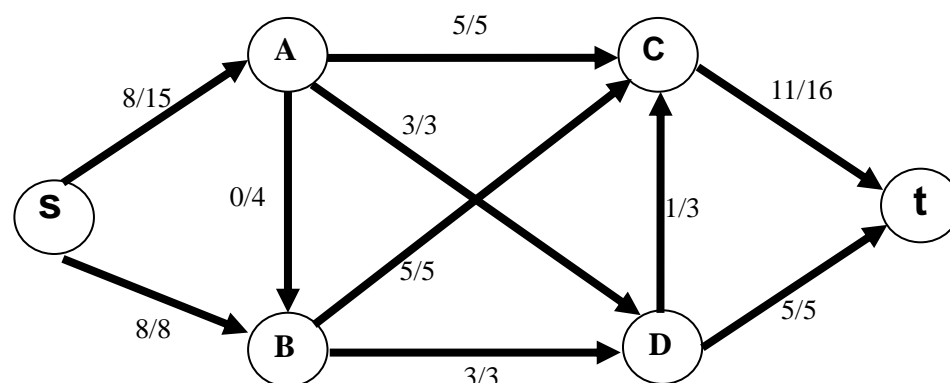
Ford und Fulkerson beschreiben einen iterativen Algorithmus. Man geht von einem Flussnetzwerk N aus. Weiterhin bezeichne s die Quelle und t die Senke.

Nach Ende des Algorithmus enthält den MaxFluss f in N.

```

1 ford_fulkerson_methode (N, s, t) {
2   f = 0; ----- Anzahl des Flusses
3   while (existiert ein flusserhöhrendem Weg p von s nach t) {
4     erweitere f um p;
5   }
6   return f;
7 }
  
```

Zunächst wird der Fluss mit 0 initialisiert. Dann wird der Fluss jeweils erweitert, bis es keinen erweiternden Weg in Netzwerk mehr erlaubt.



Von s nach t können die Flüsse wie folgt verteilt werden:

5 Flüsse S-A-C-T

3 Flüsse S-A-D-T

5 Flüsse S-B-C-T

2 Flüsse S-B-D-T
1 Fluss S-B-D-C-T
MaxFlow = 16

Die Komplexität des Algorithmus ist abhängig von der Anzahl der Kanten m und der Anzahl der Knoten. Obwohl jeder Schleiferdurchlauf des Algorithmus lediglich $O(m)$ Zeit benötigt, ist er bei ungünstiger Wahl des flusserweiternden Weges vom maximalen Fluss f abhängig. Falls die Kapazität Integer ist, ist die Laufzeitkomplexität des Ford-Fulkerson Algorithmus von $O(E * f)$.

3. Die Web Community

Eine Web Gemeinschaft ist eine Menge von Seiten, die mehr eingehende und ausgehende Kanten zu anderen Mitgliedern einer Gemeinschaft haben als zu Nichtmitgliedern.

Hier werden nicht nur die ausgehenden Links beachtet, sondern auch die eingehenden Links. Die ausgehenden Links können im Quelltext der Webseiten gefunden werden, und die eingehenden Links können mit Hilfe des Schlüsselworts **Link: Adresse der Webseite** bei vielen Suchmaschinen gesucht werden.

```
...  
<table>  
<tr>  
<td>  
<a href="http://stuttgart.de/termin.html">  
</td>  
</tr>  
</table>  
...  
http://www.google.de/search?q=link:http://stuttg  
art.de
```

Abbildung 3:
Ein- und Ausgehende Links

Wir betrachten das Internet als einen Graphen, die Web-Seiten werden als Knoten und die Hyperlinks als Kanten dargestellt. Dann ist eine Web Community ein Teilgraph aus dem Web-Graphen, der eine spezielle Eigenschaft hat, nämlich die, dass weniger Kanten die Gemeinschaft verlassen als Mitglieder verbinden. Das Finden einer Gemeinschaft lässt sich auf ein Graphenproblem reduzieren.

3.1 Übertragung der MaxFlow-MinCut auf Web Community

Der Maximum-Flow Algorithmus findet die Kanten, die Quelle und Senke verbinden, und beseitigt diese. Somit wird ein Teilgraph vom Web getrennt und eine Web Community gebildet. Die Kanten sind so gewählt, dass deren Kapazitätssumme das Minimum aller Kanten ist. Das bildet genau den Minimum-Cut.

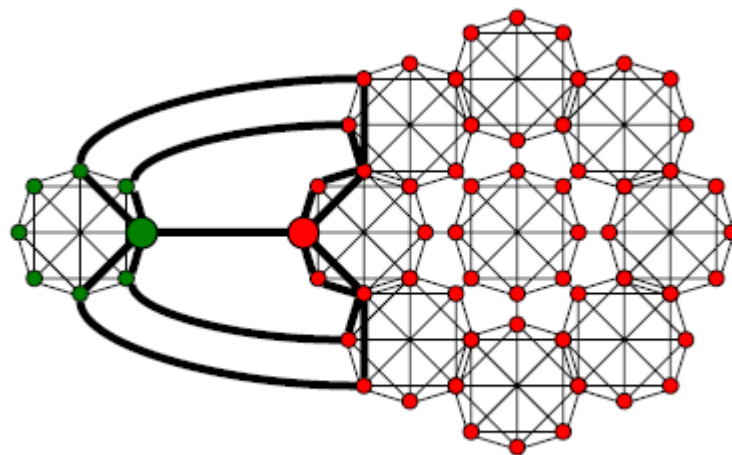


Abbildung 4:

Links wird die Quelle (grüner Knoten mit grünem Schatten) von der Senke (roter Knoten mit rotem Schatten) getrennt, somit repräsentiert der grüne Teilgraph eine Web Community, die sich vom Rest des Webgraphen trennen lässt.

Quelle: [NEC Research Institute](#) [5]

3.2 Exact-Flow Community

Bei der Variante wird ein Bündel von Seiten als Kern der Gemeinschaft verwendet (Quellseiten). Von ihnen ausgehend wird die Gemeinschaft ermittelt. Um diese Seiten zusammenzufassen, wird eine künstliche Quelle mit Kanten zu den Quellseiten erstellt.

Es ist keine explizite Senke erforderlich, man verwendet eine künstliche damit die Quellseiten niemals getrennt werden können, somit ist die Kapazität unendlich.

Definition 1:

Eine Community ist eine Knoten-Untermenge $C \subset V$, so dass für alle Knoten $v \in C$ gilt, v hat mindestens so viele Kanten, die mit Knoten in C verbunden sind wie in $(V-C)$.

Definition 2:

Sei $s\#$ Anzahl der Kanten zwischen s und allen Knoten in $(C-S)$ und sei $t\#$ Anzahl der Kanten zwischen t und allen Knoten in $(V-C-t)$.

Eine Community C kann identifiziert werden, indem der s - t Minimum Cut vom Graph G berechnet wird, mit s als Quelle und t als Senke und mit $s\#$ und $t\#$ größer als die Schnittmenge. Nach diesem Schnitt gehören die von s erreichbaren Knoten zur Community.

Der Beweis ist einfach. Nehmen wir an, es existiert in C ein Knoten v , das von s erreichbar ist und $v \notin C$. Daraus folgt, dass v mehrere Verbindungen mit den Knoten in $(V - C)$ hat als in C . Mit der MinCut Definition können wir den Graphen noch effizienter schneiden, so dass v in $(V - C)$ bleibt und von t erreichbar ist. Das ist ein Widerspruch zu der Annahme. Also bilden die Knoten, die von s erreichbar sind, nach dem Schnitt eine Community.

Dieses Verfahren liefert eine eindeutige Community und ist in der Theorie von keinem externen Faktor abhängig.

Doch leider ist es kaum zu verwirklichen. Einen Graphen dieser Größe kann kein Computer speichern und es ist ein enormer Aufwand jede Seite, die im WWW existiert, zu finden. Das WWW wächst ständig an und es gibt Seiten, die weder durch Suchmaschinen auffindbar sind, noch von anderen Seiten verlinkt werden.

3.3 Approximate-Flow Community

Das Verfahren basiert auf dem obigen, und hängt von einem externen Faktor ab: Die Suchmaschine.

Eine angenäherte Community wird in mehreren Iterationen durch Aufnahme einiger Nichtmitglieder in die Gemeinschaft erreicht. So wächst die Menge der Quellseiten und die Größe des zu bearbeitenden Webgraphen.

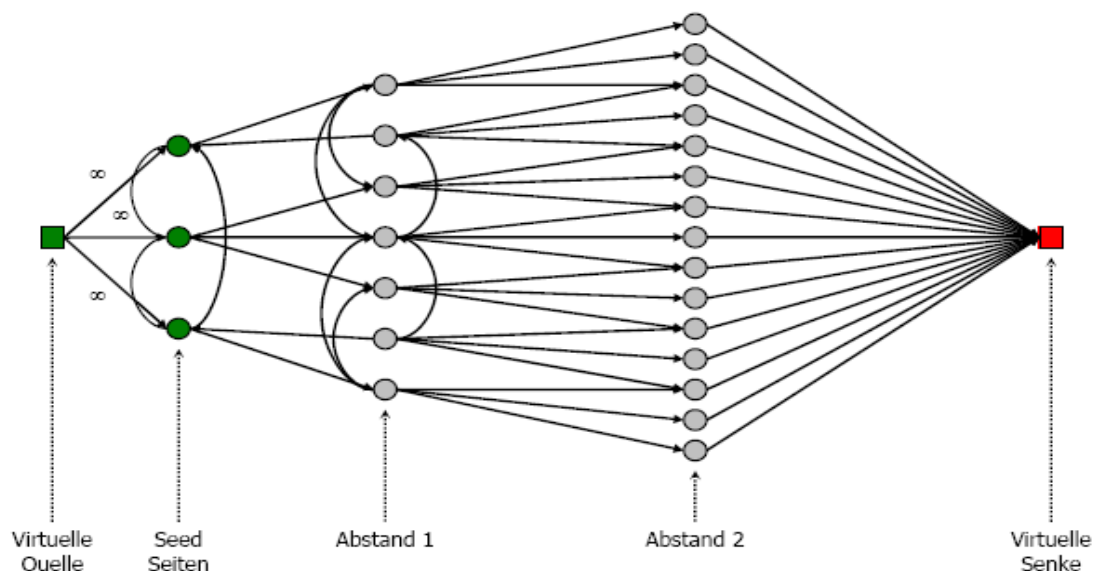


Abbildung 5:

Focused community crawling und entstehende Graphen [3]

In Abbildung 5 beginnt das Crawler von der Quelle aus und findet alle Seiten, die einen ein- oder ausgehenden Link besitzen. Dessen HTML-Code wird heruntergeladen, falls die gewünschten Seiten (Abstand 1) gefunden werden. Dann werden die ein- und ausgehenden Links gespeichert. Der Vorgang wird beendet, wenn ein solcher Knoten (Abstand 2) gefunden wird, der nicht zur Gemeinschaft gehört, da er sich in der Nähe der Senke befindet. Hier werden also die Quellseiten von den Senkeseiten getrennt: MinCut.

Da keine Webseiten als Senke benutzt werden können, wird eine virtuelle Senke eingesetzt, die garantiert, dass der approximierte Schnitt identisch zum idealen Schnitt ist.

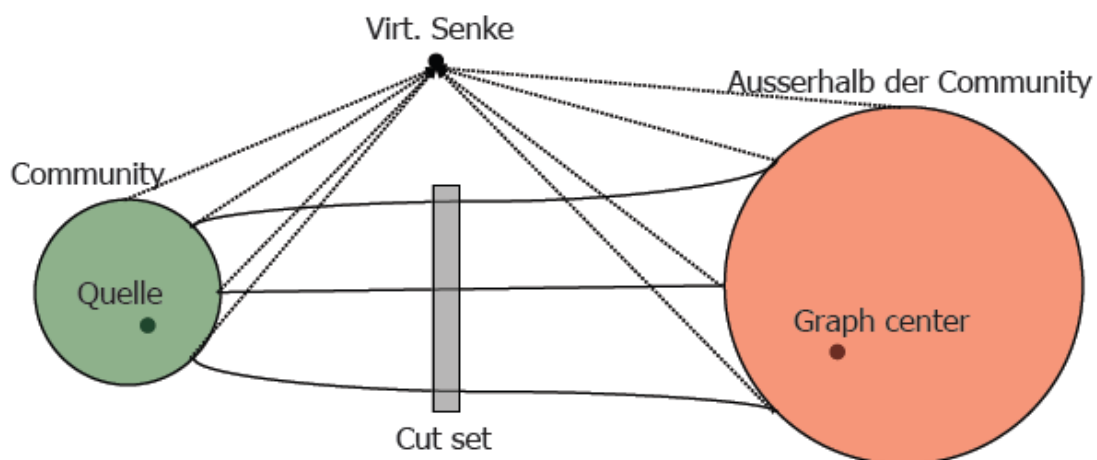


Abbildung 6:

Lokalisieren eines minimalen Schnitts, auch wenn keine gute Senke zur Verfügung steht. Eine virtuelle Senke wird benutzt. [3]

Es kommt zu einem Punkt, an dem die Verlinkungen innerhalb der Gemeinschaften so dicht sind, dass keine externe Seite mehr ausreichend Links aufweist, um aufgenommen zu werden.

Wenn wenige Webseiten als Seed-Seiten benutzt werden, hat die

Methode weniger Erfolg. Mit wenigen Webseiten kann nur ein Teil einer Community gefunden werden, nicht die ganze gesuchte Web Community. Folgender Algorithmus löst das Problem:

```
1 procedure FOCUSED CRAWL(graph:  $G = (V, E)$ ;  
2 vertex:  $s, t \in V$ )  
3 while number of iterations is less than desired do  
4   Set  $k$  equal to the number of vertices in seed set.  
5   Perform maximum flow analysis of  $G$ ,  
6   yielding community,  $C$ .  
7   Identify non-seed vertex,  $v^* \in C$ ,  
8   with the highest in-degree relative to  $G$ .  
9   for all  $v \in C$  such that in-degree of  $v$  equals  $v^*$   
10    Add  $v$  to seed set.  
11    Add edge  $(s, v)$  to  $E$  with infinite capacity.  
12  end for  
13  Identify non-seed vertex,  $u^*$ ,  
14  with the highest out-degree relative to  $G$ .  
15  for all  $u \in C$  such that out-degree of  $u$  equals  $u^*$   
16    Add  $u$  to seed set.  
17    Add edge  $(s, u)$  to  $E$  with infinite capacity.  
18  end for  
19  Re-crawl so that  $G$  uses all seeds.  
20  Let  $G$  reflect new information from the crawl.  
21 end while  
22 end procedure
```

Abbildung 7:

Der Focused Crawl Algorithmus[3]

Der Algorithmus identifiziert eine Untermenge der Community. Die stärkste gefundene Seite, je nach Anzahl der Links, wird als neue Seed-Seite benutzt. Die ausgehenden Links von neuen Seed-Seiten bilden einen neuen Teilgraph bis keine mehr gefunden werden. Die gefundene Community wird vom Rest des Webgraphens getrennt.

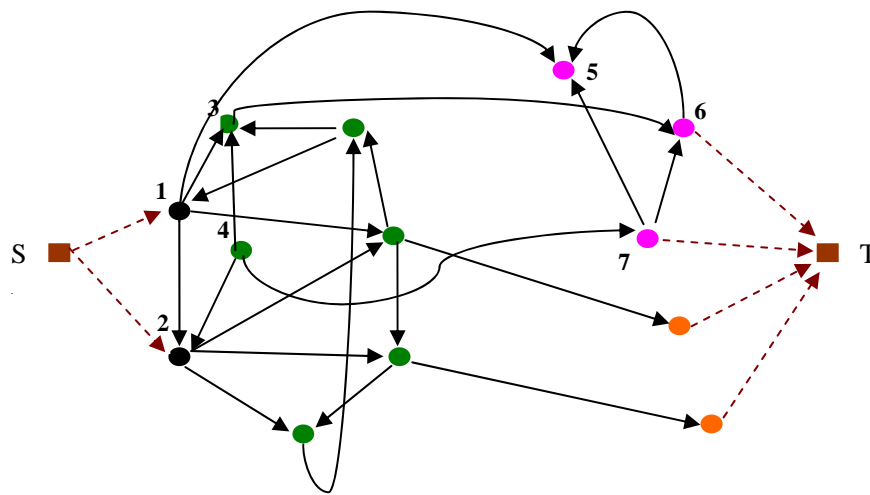


Abbildung 8:
Beispiel der Verwendung des Focused Crawl Algorithmus
1,2 SeedSeiten 3,4 durch Algorithmus aufgenommene Seedseiten
5,6,7 neu in die Community aufgenommene Seedseiten

Abbildung 8 zeigt ein Beispiel der Verwendung des Focused Crawl Algorithmus. Von SeedSeiten 1,2 aus werden eine Community aus den grünen Knoten gebildet. Durch Focused Crawl werden innerhalb der Community die Seiten 3,4 in die Seedseiten aufgenommen, da 3 die höchsten eingehenden Links und 4 die höchsten ausgehenden Links hat.

Danach wird die Community durch SeedSeiten 1,2,3,4 neu berechnet, und die Seiten 5,6,7 werden in die Community aufgenommen. Der Algorithmus verläuft iterativ, so wird die Community jedesmal durch Aufnahme Nichtmitglieder in die Community vergrößert.

Diese Community ist aufgrund unterschiedlicher Suchmaschinen nicht eindeutig. Es existieren verschiedene Suchalgorithmen und sie liefern unterschiedliche Ergebnisse.

Ein weiteres Problem besteht darin, dass in kurzer Zeit viele Links abgefragt werden müssen, welches eine große Menge Arbeit für die Suchmaschinen wegen der HTTP Request ist.

4. Experimentelle Ergebnisse

Um die Maximum-Flow Methode von Identifikation der Gemeinschaft zu testen, benutzen wir hier drei verschiedene Webseiten-Bündel als Seedseiten.

4.1 Support Vektor Maschine Community

Wir wählen die Support Vektor Maschine Community für den Test, weil sie schon 6 Jahre alt in diesem Forschungsbereich und im Vergleich zu anderen Forschungsbereichen noch klein ist, eine beachtliche Anzahl renommierter Forscher verweisen kann und nicht in bekannten Portalen gelistet ist.

Die Links, deren Quellen und Ziele in gleichen Domänen liegen, werden ignoriert, um interne Links zu vermeiden

Es wurden 4 URLs als Seed-Seiten verwendet:

<http://svm.first.gmd.de/>

<http://svm.research.bell-labs.com/>

<http://www.clrc.rhbnc.ac.uk/research/SVM/>

<http://www.support-vector.net/>

Das Ergebnis sieht folgender Maßen aus:

Graph Größe: 11000

Gefundene Gemeinschaftgröße: 252

4.2 The Internet Archive Community

Als zweites Beispiel wählen wir IA, weil sie diverse Bereiche beinhaltet: Information Retrieval, Library Science, Archivierung, Visualisierung etc. Da diese Community so vielfältig ist, wurden 11 URLs als Seed-Seiten benutzt.

Das Ergebnis sieht wie folgt aus:

Graph Größe: 7000

Gefundene Gemeinschaftgröße: 289

Die ersten 40 Ergebnisse bestehen aus einem Gemisch von Internet-Statistiken, digitalen Bibliotheken, Archiv-Organisationen und weiteren Webseiten, die nahe zur IA stehen. Sogar die letzten drei Links, die am wenigsten bewertet wurden, gehören auch zur IA Community.

4.3 The Ronald Rivest Community

Für das letzte Beispiel wurde beschlossen, nach einer Personengemeinschaft zu suchen. Aus mehreren Kandidaten wurde Ronald Rivest ausgewählt, weil er eine sehr bekannte Person ist und viele Webseiten auf seine Arbeit über Verschlüsselung und seine Veröffentlichung „Introduction to Algorithms“ [4] verweisen.

Hier wurde eine einzige Seed-Seite verwendet, deshalb wurden bei der ersten Iteration die internen Links auch als extern betrachtet.

Seed-URL: <http://theory.lcs.mit.edu/~rivest>.

Das Ergebnis sieht wie folgt aus:

Graph Größe: 38000

Gefundene Gemeinschaftsgröße: 150

5. Zusammenfassung und Fazit

Das Verfahren funktioniert in der Theorie und wurde von den Autoren durch 3 Experimente bewiesen, dass es möglich ist, mit den entsprechenden Quellseiten die gesuchte Webgemeinschaft zu finden und deren Mitglieder aufzulisten.

Das Verfahren ist noch vergleichsweise jung und könnte praktisch eingesetzt werden wie z.B. :

Verbesserte Suchmaschinen: Es wäre möglich, effizientere Suchergebnisse zu erreichen, wenn die Suchmaschinen ihre Suche auf Web Communities beschränken können.

Verbesserte Filterung: Mit Identifikation der unerwünschten Gemeinschaften können die Suchmaschinen sehr einfach die entsprechenden Seiten rausfiltern.

Da das Verfahren auch eingehende Links berücksichtigt, könnte es passieren, dass die Spam-Seiten in die Community eingebracht werden. Diese sollten die Suchmaschinen einer solchen Seite herausfiltern können.

Der Approximate-Flow Algorithmus hängt sehr von einer Suchmaschine ab und hat deswegen den Nachteil, dass unterschiedliche Suchmaschinen verschiedene Resultate liefern und somit verschiedene Gemeinschaften. Ein anderer Nachteil dieses Verfahrens liegt in ausgewählten Quellseiten, was entscheidend für

die Suchergebnisse ist. Sind die initialen Quellseiten ungünstig gewählt und werden bei der Iteration unerwünschte Mitglieder in die Gemeinschaft aufgenommen, weicht der Themenschwerpunkt weiter ab. Der Fehler kann sich fortpflanzen und ein total falsches Ergebnis als eine Web Community wird entstehen.

6. Referenzen

- [1] <http://www.wikipedia.org>
- [2] Lester Ford und Delber Fulkerson. Maximal flow through a network (context). 1956.
- [3] Gary William Flake, Steve Lawrence und C. Lee Giles. Efficient Identification of web communities. Boston, MA, USA, Aug 20-23 2000.
- [4] T.H.Cormen, C.E. Leiserson, and R.L. Riverst. Introduction to algorithms. MIT Press and McGraw-Hill Book Company, 6th edition, 1992.
- [5] <http://www.nec-labs.com>
- [6] Gary William Flake, Steve Lawrence und C. Lee Giles, Frans M. Coetzee. Self-Organization and Identification of Web Communities, 2002.
- [7] David Gibson. Inferring Web Communities from link topologies (Research report RJ. International Business Machines Corporation Research Division), IBM Research Division, 1998.