Übung zu Peer-to-Peer und Cloud Computing

Übungstermin 03: Besprechung des Übungsblattes 02

Dominik Rauh

14. November 2018

Universität Augsburg
Institut für Informatik
Lehrstuhl für Organic Computing

Mittwoch bis Montag

- Mittwoch bis Montag sollte reichen
- wenn längere Aufgaben kommen, bekommt ihr mehr Zeit
- nächstes Blatt bis: Montag, 26.11., 08:00 Uhr

Verständnisfragen zu Graphentheorie

Warum sind in P2P-Netzwerken hohe Clustering-Koeffizienten oft erstrebenswert? Geben Sie mehrere Gründe an und erklären Sie diese!

Clustering-Koeffizient?

- $C_v \sim$ Anzahl der Verbindungen unter den Nachbarn eines Knoten
- in dieser Vorlesung: *lokales* Maß
- $C_v = 1 \Leftrightarrow$ Nachbarn von v sind eine *Clique*, also ein total vernetzter Teilgraph
- $C_v = 0 \Leftrightarrow$ Nachbarn von v haben keine Verbindungen zueinander
- niedriger $C_v \Rightarrow$ Zufallsgraph um v herum
- hoher $C_v \Rightarrow$ Small World–Netzwerk um v herum

Hohe Clustering-Koeffizienten in P2P-Netzwerken?

höhere Fehlertoleranz redundante Verbindungen ⇒ geringere Wahrscheinlichkeit für „Auseinanderbrechen“ des Graphen bei Knotenausfall

geringere Routing-Komplexität lokal mehr mögliche Pfade ⇒ effizienter Zugriff auf nahe Long Distance–Links etc.

Performanz durchschnittliche Pfadlänge kürzer ⇒ geringere Latenz, schnellere Suche, …

effizientes Bootstrapping neu hinzugefügte Knoten sind schnell gut eingebunden

Weshalb teilt man bei der Berechnung von L_G durch den Term $|V| * (|V| - 1)/2$? Kombinatorik könnte bei der Erklärung helfen!

L_G?

- durchschnittliche Pfadlänge des Graphen G
- die Summe der Längen aller minimalen Pfade (Distanzen) geteilt durch die Anzahl aller minimalen Pfade
- Anzahl aller minimalen Pfade entspricht Anzahl der Knotenpaare

- Wähle einen Knoten v aus $|V|$ Knoten aus!
- Wähle einen zweiten Knoten aus den verbliebenen $|V - 1|$ Knoten aus (v wird nicht mehr betrachtet, da keine Selbstreferenzen)!
- Eigentlich bidirektionale Kanten wurden jetzt doppelt gezählt $\Rightarrow$ Faktor $\frac{1}{2}$

Ziehe zwei Knoten aus $|V|$ ohne Beachtung der Reihenfolge (Bidirektionalität) und ohne Zurücklegen (keine Selbstreferenz)!

$$\binom{|V|}{2} = \frac{|V|!}{2!\,(|V|-2)!} = \frac{|V|\,(|V|-1)\,(|V|-2)!}{2\,(|V|-2)!} = \frac{|V|\,(|V|-1)}{2}$$

Problem: Inseln

- wenn ein Knoten u nicht von v aus erreichbar ist, dann $d_{v,u} = \infty$
- somit $L_G = \infty$

$\Rightarrow$ Aussagekraft von L_G nur bei *verbundenen Graphen* ($\neq$ total/vollständig vernetzt!): Es existiert mindestens ein Pfad zwischen jedem möglichen Paar (v, w) von Knoten

Welche beiden berühmten Algorithmen eignen sich für die *genaue* Ermittlung der für die Berechnung von L_G benötigten Entfernung für jedes Paar von Knoten?

Welche Algorithmen?

- Dijkstra und Floyd-Warshall
- aber auch: Bellman-Ford
- Breitensuche eher nicht
  - entspricht Dijkstra mit 1-gewichteten Kanten
  - hier: Kanten verschieden gewichtet

Welchen Zusammenhang gibt es zwischen der Anzahl von Kanten im Graphen und der Laufzeit dieser Algorithmen?

Laufzeit?

- Floyd-Warshall ist besser in dichten (*dense*) Graphen, die fast alle möglichen Kanten haben (denn $O(|V|^3)$)
- Dijkstra ist besser in dünn besetzten (*sparse*) Graphen, die nur wenige Kanten haben. Dijkstra muss für jeden Knoten einmal durchlaufen, somit $O(|E|\,|V| + |V|^2 \log |V|)$ (mit Fibonacci-Heaps); bei dünn besetzten Graphen, ist $|E| \ll |V|^2$, somit ist Dijkstra besser.
- Bellman-Ford: $O(|E|\,|V|^2)$.

Ist ein Graph mit $|V| = 1$ und $|E| = 0$ wirklich ein Graph?

Ist $(V, \emptyset)$ ein Graph?

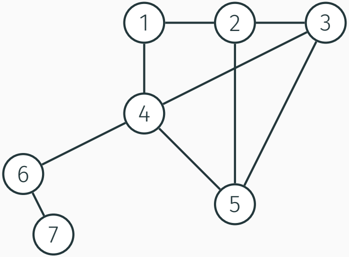
Ja, denn die Definition schränkt die Anzahl an Elementen von V und E nicht ein.

Rechenaufgaben zu Graphentheorie

Berechnen Sie für die *markierten* Knoten v_n der Graphen (a) bis (d): Grad k_v, Nachbarschaft N_v und Clustering-Koeffizient C_v!

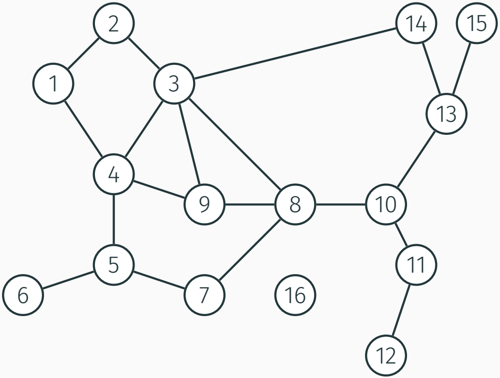
Graph (a)

- $k_{v_3} = 3$
- $N_{v_3} = \{2, 4, 5\}$
- $C_{v_3} = \frac{2}{3*(3-1)*\frac{1}{2}} = \frac{2}{3}$
- $k_{v_4} = 4$
- $N_{v_4} = \{1, 3, 5, 6\}$
- $C_{v_4} = \frac{1}{4*(4-1)*\frac{1}{2}} = \frac{1}{6}$



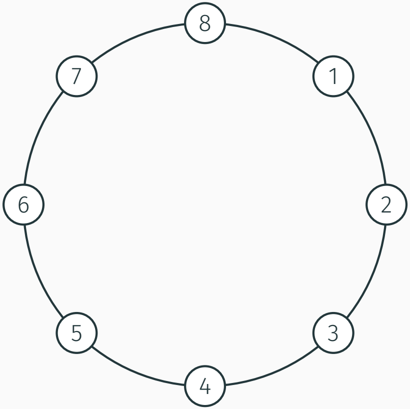
Graph (b)

- $k_{v_2} = 2$
- $N_{v_2} = \{1, 3\}$
- $C_{v_2} = \frac{0}{} = 0$
- $k_{v_8} = 4$
- $N_{v_8} = \{3, 7, 9, 10\}$
- $C_{v_8} = \frac{1}{4*(4-1)*\frac{1}{2}} = \frac{1}{6}$



Graph (c)

- $k_{v_1} = 2$
- $N_{v_1} = \{2, 8\}$
- $C_{v_1} = 0$



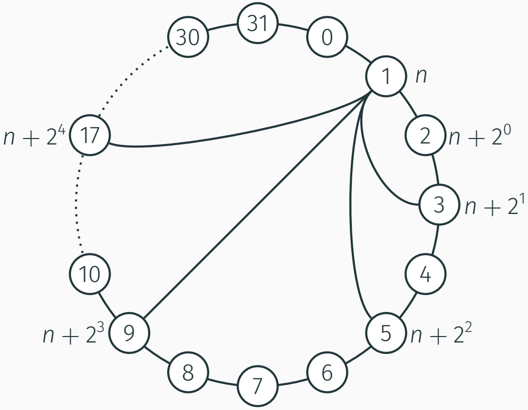
Nachbarschaftstabelle für Graph (d)

	2	3	5	9	17	25	29	31	0
$2^0 = 1$	3	4	6	10	18	26	30	0	1
$2^1 = 2$	4	5	7	11	19	27	31	1	2
$2^2 = 4$	6	7	9	13	21	29	1	3	4
$2^3 = 8$	10	11	13	17	25	1	5	7	8
$2^4 = 16$	18	19	21	25	1	9	13	15	16

- Anzahl der blauen Felder: Anzahl der Verbindungen zwischen Nachbarn von v_1
- v_{25}, v_{29}, v_{31} und v_0 besitzen ihrerseits Verbindungen zu v_1 (gelb)
  $\Rightarrow$ ebenfalls Nachbarn von v_1 (Bidirektionalität)
- $v_9 \leftrightarrow v_{25}$ darf nur einmal gezählt werden (einmal rot markiert)

Graph (d)

- $k_{v_1} = 9$
- $N_{v_1} = \{0, 2, 3, 5, 9, 17, 25, 29, 31\}$
- $|N_{v_1}| = 9$
- $C_{v_1} = \frac{12}{(9*(9-1)*\frac{1}{2})} = \frac{1}{3}$



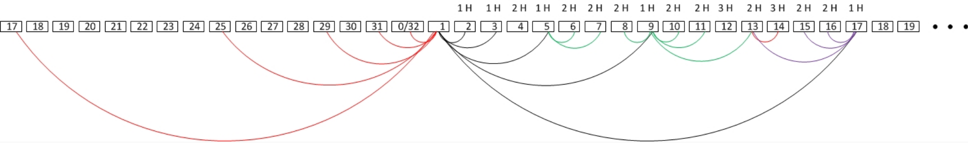
Welchen Durchmesser haben die jeweiligen Graphen?

Durchmesser der Graphen?

Durchmesser: Der längste aller kürzesten Wege.

- $D_{(a)} = |p(2,7)| = 4$
- $D_{(b)} = \infty$ (wegen v_{16})
- $D_{(c)} = 4$
- $D_{(d)} = 3$

Durchmesser von Graph (d)



Berechnen Sie für die Graphen (a) und (b) den Wert L_G.

Distanztabelle für $L_{(a)}$

$v_1 \rightarrow v_2$	1										
$v_1 \rightarrow v_3$	2	$v_2 \rightarrow v_3$	1								
$v_1 \rightarrow v_4$	1	$v_2 \rightarrow v_4$	2	$v_3 \rightarrow v_4$	1						
$v_1 \rightarrow v_5$	2	$v_2 \rightarrow v_5$	1	$v_3 \rightarrow v_5$	1	$v_4 \rightarrow v_5$	1				
$v_1 \rightarrow v_6$	2	$v_2 \rightarrow v_6$	3	$v_3 \rightarrow v_6$	2	$v_4 \rightarrow v_6$	1	$v_5 \rightarrow v_6$	2		
$v_1 \rightarrow v_7$	3	$v_2 \rightarrow v_7$	4	$v_3 \rightarrow v_7$	3	$v_4 \rightarrow v_7$	2	$v_5 \rightarrow v_7$	3	$v_6 \rightarrow v_7$	1

$L_{(a)}$ und $L_{(b)}$

- $L_{(a)} = \frac{39}{7*(7-1)*\frac{1}{2}} = 1,86$
- $L_{(b)} = \infty$