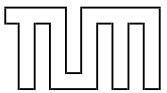
Fortgeschrittene Netzwerk- und Graph-Algorithmen

Dr. Hanjo Täubig

Lehrstuhl für Effiziente Algorithmen
(Prof. Dr. Ernst W. Mayr)
Institut für Informatik
Technische Universität München

Wintersemester 2007/08



Übersicht

- Approximation von Closeness
- Approximation von Betweenness

Approximation von Zentralitätsindizes

- Obwohl die behandelten Zentralitätsindizes in polynomieller Zeit berechnet werden können, heißt das nicht unbedingt , dass die entsprechenden Algorithmen in der Praxis anwendbar sind.

- Beispiel:
  Betweenness-Zentralität für die Knoten des Web-Graphen lässt sich selbst mit dem Algorithmus von Brandes nicht in akzeptabler Zeit berechnen.

$\Rightarrow$ Möglichst wenige Traversierungen bzw. SSSP-Durchläufe auf dem Graphen

$\Rightarrow$ Näherungslösungen mit möglichst geringer Abweichung (mit hoher Wahrscheinlichkeit)

Approximation von Closeness

- Closeness:

$$c_C(v) = \frac{1}{\sum_{w \in V} d(v, w)}$$

- Approximation: wähle k andere Knoten $v_1, \ldots, v_k \in V$

$$\hat{c}_C(v) = \frac{k}{n} \frac{1}{\sum_{i=1}^{k} d(v, v_i)}$$

- Vorgehen (Eppstein / Wang):
  1. Wähle k Knoten $v_1, \ldots, v_k$ gleichverteilt zufällig
  2. Löse für jeden Knoten v_i das SSSP mit diesem Knoten als Startknoten und
  3. berechne für jeden Knoten $v \in V$ die Zentralität

$$\hat{c}_C(v) = \frac{k}{n \cdot \sum_{i=1}^{k} d(v, v_i)}$$

Hoeffdings Ungleichung

Satz (Hoeffdings Ungleichung)

Seien $X_1, \ldots, X_k$ unabhängige Zufallsvariablen mit $a_i \leq X_i \leq b_i$ und $\mu = \mathbb{E}\left[\sum_{i=1}^{k} X_i/k\right]$ der erwartete Durchschnitt. Dann gilt

$$\Pr\left\{\left|\frac{\sum_{i=1}^{k} X_i}{k} - \mu\right| \geq \xi\right\} \leq 2 \cdot e^{-2k^2\xi^2 / \sum_{i=1}^{k}(b_i - a_i)^2}$$

bzw. im Fall $a_i = a$, $b_i = b$ $(\forall i)$

$$\Pr\left\{\left|\frac{\sum_{i=1}^{k} X_i}{k} - \mu\right| \geq \xi\right\} \leq 2 \cdot e^{-2k\xi^2/(b-a)^2}$$

Anwendung der Hoeffding-Ungleichung

Setze

$$
\begin{aligned}
X_i &= n \cdot \frac{d(v_i, u)}{n-1} \\
\mu &= \frac{1}{c_C(v)} \\
a_i &= 0 \\
b_i &= \frac{n \cdot \mathrm{diam}(G)}{n-1}
\end{aligned}
$$

Anwendung der Hoeffding-Ungleichung

$$\begin{aligned}
\Pr\left\{\left|\frac{\sum_{i=1}^{k} X_i}{k} - \mu\right| \geq \xi\right\} &\leq 2 \cdot e^{-2k^2\xi^2 / \sum_{i=1}^{k}(b_i - a_i)^2} \\
&= 2 \cdot e^{-2k^2\xi^2 / \left(k\left(\frac{n \cdot \mathrm{diam}(G)}{n-1}\right)^2\right)} \\
&= 2 \cdot e^{-\Omega\left(k\xi^2 / \mathrm{diam}(G)^2\right)}
\end{aligned}$$

- Wähle $\xi = \epsilon \cdot \mathrm{diam}(G)$
- $k = \Theta\left(\frac{\log n}{\epsilon^2}\right)$ SSSP-Läufe

$\Rightarrow$ Wahrscheinlichkeit, einen Fehler größer als $\epsilon \cdot \mathrm{diam}(G)$ zu machen ist höchstens $\frac{1}{n}$ für jeden Wert

Laufzeit der Closeness-Approximation

- Komplexität eines SSSP-Laufs
  - $\mathcal{O}(m+n)$ in ungewichteten Graphen
  - $\mathcal{O}(m+n\log n)$ in gewichteten Graphen
- Komplexität von k SSSP-Läufen
  - $\mathcal{O}(k\cdot(m+n))$ in ungewichteten Graphen
  - $\mathcal{O}(k\cdot(m+n\log n)$ in gewichteten Graphen

$\Rightarrow$ Komplexität von $\Theta\left(\frac{\log n}{\epsilon^2}\right)$ SSSP-Läufen

- $\mathcal{O}\left(\frac{\log n}{\epsilon^2}\cdot(m+n)\right)$ in ungewichteten Graphen
- $\mathcal{O}\left(\frac{\log n}{\epsilon^2}\cdot(m+n\log n)\right)$ in gewichteten Graphen

Approximation von Betweenness

- gewichtete gerichtete Graphen
- wähle wieder k Knoten zufällig (gleichverteilt) aus
- Berechne für jeden Startknoten v_i die totalen Abhängigkeiten $\delta_{v_i*}(v)$ aller anderen Knoten v
- Berechne

$$\hat{c}_B(v) = \sum_{i=1}^{k} \frac{n}{k} \cdot \delta_{v_i*}(v)$$

- $\mathbb{E}[\hat{c}_B(v)] = c_B(v)$ für alle k und v

Anwendung der Hoeffding-Ungleichung

Setze

$$\begin{aligned} X_i &= n \cdot \delta_{v_i *} \\ \mu &= c_B(v) \\ a_i &= 0 \\ b_i &= n(n-2) \end{aligned}$$

$\delta_{v_i *}$ kann höchstens $n-2$ sein, und zwar wenn alle kürzesten Pfade, die von v_i ausgehen, über v laufen. Also ist X_i durch $n(n-2)$ begrenzt.

Anwendung der Hoeffding-Ungleichung

$$\begin{aligned}\Pr\left\{|\hat{c}_B(v) - c_B(v)| \geq \xi\right\} &\leq 2 \cdot e^{-2k^2\xi^2 / \sum_{i=1}^{k}(b_i - a_i)^2} \\ &= 2 \cdot e^{-2k^2\xi^2 / \left(k(n(n-2))^2\right)} \\ &= 2 \cdot e^{-2k\xi^2 / (n(n-2))^2}\end{aligned}$$

- Wähle $\xi = \epsilon \cdot n(n-2)$
- $k = \Theta\left(\frac{\log n}{\epsilon^2}\right)$ Startknoten / Läufe

$\Rightarrow$ Wahrscheinlichkeit, einen Fehler größer als $\epsilon \cdot n(n-2)$ zu machen ist höchstens $\frac{1}{n}$ für jeden Wert

Laufzeit der Betweenness-Approximation

- Komplexität eines Laufs (für $\delta_{v_i*}(v)$)
  - $\mathcal{O}(m+n)$ in ungewichteten Graphen
  - $\mathcal{O}(m+n\log n)$ in gewichteten Graphen
- Komplexität von k Läufen
  - $\mathcal{O}(k\cdot(m+n))$ in ungewichteten Graphen
  - $\mathcal{O}(k\cdot(m+n\log n)$ in gewichteten Graphen

$\Rightarrow$ Komplexität von $\Theta\left(\frac{\log n}{\epsilon^2}\right)$ Läufen

- $\mathcal{O}\left(\frac{\log n}{\epsilon^2}\cdot(m+n)\right)$ in ungewichteten Graphen
- $\mathcal{O}\left(\frac{\log n}{\epsilon^2}\cdot(m+n\log n)\right)$ in gewichteten Graphen

Ergebnis

- Gewinn: k anstatt n SSSP-artige Läufe
- Verfahren des normalisierten Durchschnitts basierend auf zufälligem Knoten-Sampling läßt sich auf viele andere Zentralitäten übertragen

Kohäsive Gruppen

Eigenschaften:

- Gegenseitigkeit
  Gruppenmitglieder wählen sich gegenseitig in die Gruppe und sind im graphtheoretischen Sinn benachbart
- Kompaktheit / Erreichbarkeit:
  Gruppenmitglieder sind gegenseitig gut erreichbar
  (wenn auch nicht unbedingt adjazent), insbesondere
  - auf kurzen Wegen
  - auf vielen verschiedenen Wegen
- Dichte:
  Gruppenmitglieder haben eine große Nachbarschaft innerhalb der Gruppe
- Separation:
  Gruppenmitglieder haben mit größerer Wahrscheinlichkeit Kontakt zu einem anderen Mitglied der Gruppe als zu einem Nicht-Gruppenmitglied

Lokale Dichte

- Eine Gruppeneigenschaft heißt *lokal*, wenn sie bestimmt werden kann, indem man nur den von der Gruppe induzierten Teilgraphen betrachtet.

$\Rightarrow$ Separation ist *nicht lokal*, weil hier auch die Verbindungen zu den anderen Knoten betrachtet werden

- Viele Definitionen von kohäsiven Gruppen verlangen außer einer Eigenschaft Π auch *Maximalität* (im Sinne von Nichterweiterbarkeit), d.h. die Gruppe darf nicht in einer anderen größeren Gruppe enthalten sein.
- Maximalität verletzt die Lokalitätsbedingung

$\Rightarrow$ Betrachten diese Eigenschaften ohne Maximalitätsbedingung

- Lokalität reflektiert wichtige Eigenschaft von Gruppen:
  - Invarianz unter Veränderung des Netzwerks außerhalb der Gruppe
  - Innere Robustheit und Stabilität ist eine wichtige Gruppeneigenschaft

Cliquen

Definition

Sei $G = (V, E)$ ein ungerichteter Graph. Ein Knotenteilmenge $U \subseteq V$ heißt Clique genau dann, wenn der von U in G induzierte Teilgraph $G[U]$ ein vollständiger Graph ist.
Eine Clique U in G ist eine maximale Clique, falls es keine Clique U' mit $U \subset U'$ in G gibt.

Cliquen – ideale kohäsive Gruppen

Cliquen sind ideale kohäsive Gruppen:
(Sei U eine Clique der Kardinalität k.)

- Cliquen haben größtmögliche Dichte

$$\delta(G[U]) = \bar{d}(G[U]) = \Delta(G[U]) = k - 1$$

- Cliquen besitzen größtmögliche Kompaktheit

$$\text{diam}(G[U]) = 1$$

- Cliquen sind bestmöglich verbunden
  U ist $(k-1)$-fach knotenzusammenhängend und
  $(k-1)$-fach kantenzusammenhängend

Satz von Turán

Satz (Turán, 1941)

Sei $G = (V, E)$ ein ungerichteter Graph mit $n = |V|$ und $m = |E|$. Falls $m > \frac{n^2}{2} \cdot \frac{k-2}{k-1}$, dann existiert eine Clique der Größe k in G.

Spezialfall:

Satz (Mantel, 1907)

Die maximale Anzahl von Kanten in einem dreiecksfreien Graphen ist $\left\lfloor \frac{n^2}{4} \right\rfloor$.

Da die meisten (z.B. soziale) Netzwerke eher dünn sind (also $o(n^2)$ Kanten haben), müssen sie nicht unbedingt von vornherein Cliquen einer bestimmten Größe > 2 enthalten.

Maximale Cliquen

- Graphen enthalten immer maximale Cliquen.
- Meistens sind es sogar viele.
- Sie können sich überlappen (ohne identisch zu sein).

Satz (Moon & Moser, 1965)

Jeder ungerichtete Graph G mit n Knoten hat höchstens $3^{\lceil \frac{n}{3} \rceil}$ maximale Cliquen.

Cliquen-Struktur

- Cliquen sind abgeschlossen unter Exklusion, d.h. wenn U eine Clique in G ist und v ein Knoten aus U, dann ist $U \setminus \{v\}$ auch eine Clique.
  Oder anders gesagt:
  Die Cliqueneigenschaft ist eine *hereditäre* Grapheigenschaft, denn sie vererbt sich auf induzierte Teilgraphen.

- Cliquen sind geschachtelt, d.h. jede Clique der Größe n enthält eine Clique der Größe $n-1$ (sogar n davon).
  (Das folgt hier sofort aus dem Abschluss unter Exklusion. Für andere Eigenschaften, die nicht unter Exklusion abgeschlossen sind, muss man das aber extra beweisen.)

Generalisierte Cliquen

Sei $G = (V, E)$ ein ungerichteter Graph, U eine Teilmenge der Knoten und $k > 0$ eine natürliche Zahl.
Generalisierte (distanz-basierte) Cliquen:

- U heißt k-Clique g.d.w. $\forall u, v \in U : \quad d_G(u, v) \leq k$.
- U heißt k-Club g.d.w. $\text{diam}(G[U]) \leq k$.
- U heißt k-Clan g.d.w. U ist eine maximale k-Clique und U ist ein k-Club.

- k-Cliquen sind nicht lokal definiert (die Distanzen können sich aus Pfaden ergeben, die über Knoten außerhalb von U führen).
- Obwohl k-Clubs und k-Clans lokal definiert sind (abgesehen von der Maximalitätsbedingung), sind sie nur von geringerem Interesse. Distanz-basierte Cliquen sind i.A. nicht abgeschlossen unter Exklusion und nicht geschachtelt.

Grundfunktionen

In $\mathcal{O}(m+n)$ können folgende Funktionen berechnet werden:

- Bestimme, ob eine gegebene Knotenteilmenge $U \subseteq V$ eine Clique in G ist.
  Bestimme für jede Kante in G, ob beide Endknoten in U sind.
  Zähle die Fälle und vergleiche mit $|U| \cdot (|U|-1)/2$.

- Bestimme, ob eine gegebene Clique $U \subseteq V$ maximal ist in G.
  Teste, ob es einen Knoten in $V \setminus U$ gibt, der adjazent zu allen Knoten in U ist.

Maximale Clique

Bestimme die lexikographisch kleinste maximale Clique U, die eine gegebene Clique $U' \subseteq V$ enthält.

- Annahme: V ist eine geordnete Menge
- Seien $U, U' \in V$. Def.: $U \leq U' \Leftrightarrow$ der kleinste Knoten der nicht in $U \cap U'$ ist, ist in U.
- Starte mit $U = U'$
- Itereriere über alle $v \in V \setminus U$ in aufsteigender Reihenfolge und teste, ob $U \subseteq N(v)$.
- Falls ja, dann füge v zu U hinzu.
- Am Ende ist U eine maximale Clique, die U' enthält.

$\Rightarrow$ ebenfalls $\mathcal{O}(m + n)$

Cliquen maximaler Kardinalität

Maximum-Clique: Clique der größtmöglichen Kardinalität in einem gegebenen Graphen

Primitiver Algorithmus: erschöpfende Suche
Zähle alle Kandidatensets $U \subseteq V$ auf
und bestimme, ob U eine Clique ist.
Gib die größte gefundene Clique aus.
$\Rightarrow$ Laufzeit $\mathcal{O}(n^2 \cdot 2^n)$

Clique-Problem

Entscheidungproblem:

Problem

Problem: ***Clique***
Eingabe: *Graph G, Parameter $k \in \mathbb{N}$*
Frage: *Existiert eine Clique der Kardinalität $\geq k$ in G?*

Härte des Clique-Problems

Sei $\omega(G)$ die Größe der Maximum-Clique(n) in G.
Wenn wir einen Algorithmus hätten, der **Clique** in Zeit $T(n)$ entscheidet, dann könnten wir $\omega(G)$ in Zeit $\mathcal{O}(T(n) \cdot \log n)$ mit binärer Suche berechnen.
Andererseits ergibt sich aus jedem Algorithmus zur Berechnung von $\omega(G)$ in Zeit $T(n)$ ein Algorithmus, der **Clique** in $T(n)$ entscheidet.
Ein polynomieller Algorithmus für das eine Problem würde als einen polynomiellen Algorithmus für das jeweils andere Problem implizieren.

Aber:

Satz

Clique *ist $\mathcal{NP}$-complete.*

Beweis: Reduktion von **Satisfiability** (Erfüllbarkeit)

Versteckte Cliquen

Folgerung

Falls $\mathcal{P} \neq \mathcal{NP}$ gilt, gibt es keinen Polynomialzeit-Algorithmus, der eine Clique der Größe k in einem Graphen findet, der garantiert eine solche Clique der Größe k enthält.

Bemerkung: die Schwierigkeit, eine versteckte Clique zu finden, hängt nicht von der Größe der Clique ab. Auch Cliquen der Größe $(1 - \varepsilon) \cdot n$ können nicht in Polynomialzeit gefunden werden.