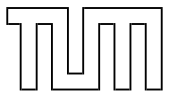
Fortgeschrittene Netzwerk- und Graph-Algorithmen

Prof. Dr. Hanjo Täubig

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

Wintersemester 2010/11



PageRank

Lawrence Page, Sergey Brin, Rajeev Motwani & Terry Winograd (1998): PageRank

- wesentlicher Bestandteil der Suchmaschine von Google
- Idee: Bewertung der Wichtigkeit einer Webseite bezüglich der topologischen Eigenschaften (seiner Position im Web), unabhängig vom Inhalt (mal abgesehen von den Links)
- ist eine Feedback-Zentralität: Wert einer Webseite hängt von der Anzahl und der Zentralität der Webseiten ab, die darauf zeigen
- Gewicht einer Seite wird gleichmäßig an die verlinkten Seiten weitergegeben

$$\mathbf{c}_{\mathrm{PR}}(p) = \frac{1-d}{n} + d \sum_{q \in N^-(p)} \frac{\mathbf{c}_{\mathrm{PR}}(q)}{d^+(q)}$$

d: Dämpfungsfaktor
$N^-(p)$: Knoten mit einer Kante zu p

PageRank

- definiere Transitionsmatrix P:

$$p_{ij} = \begin{cases} \frac{1}{d^+(j)}, & \text{falls } (j,i) \in E \\ 0, & \text{sonst} \end{cases}$$

- oder anders: $p_{ij} = \frac{1}{d^+(j)} a_{ji}$ bzw. $P = (D^+ A)^T$,
  wobei D^+ die Diagonalmatrix ist, die im i-ten Diagonaleintrag den Kehrwert des Ausgangsgrads $d^+(i)$ von Knoten i enthält (falls > 0)

- Matrixform:

$$\mathbf{c}_{\text{PR}} = dP\mathbf{c}_{\text{PR}} + \frac{1-d}{n}\mathbf{1}_n$$

- Lösung meist durch Power (oder Jacobi) Iteration:

$$\mathbf{c}^k_{\text{PR}} = dP\mathbf{c}^{k-1}_{\text{PR}} + \frac{1-d}{n}\mathbf{1}_n$$

PageRank

Folgender Satz garantiert eindeutige Lösung und Konvergenz für $d < 1$:

Satz

Falls $0 \leq d < 1$ *dann hat die Gleichung*

$$\mathbf{c}_{PR} = dP\mathbf{c}_{PR} + \frac{1-d}{n}\mathbf{1}_n$$

eine eindeutige Lösung $\mathbf{c}^*_{PR} = \frac{1-d}{n}\left(I_n - dP\right)^{-1}\mathbf{1}_n$

und die Lösungen des dynamischen Systems

$$\mathbf{c}^k_{PR} = dP\mathbf{c}^{k-1}_{PR} + \frac{1-d}{n}\mathbf{1}_n$$

erfüllen

$$\lim_{k\to\infty} \mathbf{c}^k_{PR} = \mathbf{c}^*_{PR}$$

für jeden Startvektor $\mathbf{c}^0_{PR}$.

PageRank

Etwas anderer Ansatz:

$$\mathbf{c}_{\text{PR}}^{k} = dP\mathbf{c}_{\text{PR}}^{k-1} + \frac{||\mathbf{c}^{k-1}|| - ||dP\mathbf{c}_{\text{PR}}^{k-1}||}{n}\mathbf{1}_n$$

Die Lösungen dieses System konvergieren gegen $\frac{\mathbf{c}_{\text{PR}}^{*}}{||\mathbf{c}_{\text{PR}}^{*}||}$, die normalisierte Lösung des vorherigen Ansatzes.

Hubs & Authorities

Jon Kleinberg, 1997:

- HITS: Hyperlink-Induced Topic Search
- "A good hub is a page that points to many good authorities."
- "A good authority is a page that is pointed to by many good hubs."
- im Gegensatz zu PageRank berücksichtigt HITS auch den Inhalt von Webseiten
- 2 Phasen:
  - 1. Phase hängt von der Suchanfrage ab
  - 2. Phase beschäftigt sich nur mit der Linkstruktur des entsprechenden Netzwerks

Hubs & Authorities

- Suchanfrage σ
- 1. Phase berechnet $V_\sigma \in V$ mit
  - V_σ ist klein im Vergleich zu V,
  - V_σ enthält viele Seiten, die für die Suchanfrage σ relevant sind,
  - V_σ enthält viele wichtige Authorities

Hubs & Authorities

Algorithmus 5 : Hubs & Authorities, 1. Phase

Output : V_σ = Menge relevanter Seiten

Benutze eine textbasierte Suchmaschine für Suchanfrage σ;
Sei W_σ die Liste von Ergebnissen;
Wähle $t \in \mathbb{N}$;
Sei $W_\sigma^t \subset W_\sigma$ die Menge der t Seiten, die am höchsten bewertet wurden;
$V_\sigma := W_\sigma^t$;
forall the $i \in W_\sigma^t$ **do**
    $V_\sigma := V_\sigma \cup N^+(i)$;
    if $|N^-(i)| \leq r$ *(r ist eine benutzerspezifizierte Schranke)* **then**
        $V_\sigma := V_\sigma \cup N^-(i)$;
    else
        Wähle $N_r^-(i) \subseteq N^-(i)$ so, dass $|N_r^-(i)| = r$;
        $V_\sigma := V_\sigma \cup N_r^-(i)$;
return V_σ;

Hubs & Authorities

- 2. Phase berechnet die Hub/Authority-Werte der Seiten im induzierten Graphen $G[V_\sigma]$ aus der gegenseitigen Abhängigkeit zwischen Hubs und Authorities:

$$\begin{aligned} \mathbf{c}_{\text{HA-H}} &= A_\sigma \mathbf{c}_{\text{HA-A}} \text{ unter der Annahme, dass } \mathbf{c}_{\text{HA-A}} \text{ bekannt ist} \\ \mathbf{c}_{\text{HA-A}} &= A_\sigma^T \mathbf{c}_{\text{HA-H}} \text{ unter der Annahme, dass } \mathbf{c}_{\text{HA-H}} \text{ bekannt ist} \end{aligned}$$

A_σ: Adjazenzmatrix von $G[V_\sigma]$

- Da weder $\mathbf{c}_{\text{HA-A}}$ noch $\mathbf{c}_{\text{HA-H}}$ bekannt sind, schlägt Kleinberg eine iterative Bestimmung mit Normalisierung vor.

Hubs & Authorities

Algorithmus 6 : Hubs & Authorities Iteration

Output : Approximationen für $\mathbf{c}_{\text{HA-H}}$ and $\mathbf{c}_{\text{HA-A}}$

$\mathbf{c}^0_{\text{HA-A}} := \mathbf{1}_n$;
for $k = 1 \ldots$ **do**
$\mathbf{c}^k_{\text{HA-H}} := A_\sigma \mathbf{c}^{k-1}_{\text{HA-A}}$;
$\mathbf{c}^k_{\text{HA-A}} := A_\sigma^T \mathbf{c}^k_{\text{HA-H}}$;
$\mathbf{c}^k_{\text{HA-H}} := \frac{\mathbf{c}^k_{\text{HA-H}}}{\|\mathbf{c}^k_{\text{HA-H}}\|}$;
$\mathbf{c}^k_{\text{HA-A}} := \frac{\mathbf{c}^k_{\text{HA-A}}}{\|\mathbf{c}^k_{\text{HA-A}}\|}$;

Hubs & Authorities

Satz

Sei A_σ die Adjazenzmatrix von $G[V_\sigma]$.
Dann sind die Grenzwerte

$$\lim_{k \to \infty} \mathbf{c}_{HA\text{-}A}^k = \mathbf{c}_{HA\text{-}A}$$

und

$$\lim_{k \to \infty} \mathbf{c}_{HA\text{-}H}^k = \mathbf{c}_{HA\text{-}H}$$

wobei $\mathbf{c}_{HA\text{-}A}$ ($\mathbf{c}_{HA\text{-}H}$) der erste Eigenvektor von $A_\sigma^T A_\sigma$ ($A_\sigma A_\sigma^T$) ist.

Hubs & Authorities

$\Rightarrow$ iterative Methode ist nichts anderes als die Lösung der Eigenvektor-Gleichungen

$$\begin{aligned} \lambda \mathbf{c}_{\text{HA-A}} &= (A_\sigma^T A_\sigma)\mathbf{c}_{\text{HA-A}} \\ \lambda \mathbf{c}_{\text{HA-H}} &= (A_\sigma A_\sigma^T)\mathbf{c}_{\text{HA-H}} \end{aligned}$$

für den größten Eigenwert per Power Iteration

$\mathbf{c}_{\text{HA-A}}$: enthält dann die Authority-Werte
$\mathbf{c}_{\text{HA-H}}$: enthält dann die Hub-Werte

Übersicht

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

- Einige 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.

Eine Clique U in G ist eine Maximum-Clique, falls es keine Clique U' mit $|U| < |U'|$ 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 mindestens eine maximale Clique, meistens 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.

Die Ordnung auf den Cliquen sei wie folgt definiert ($U, U' \subseteq V$):
$U < U' \Leftrightarrow$ Der kleinste Knoten aus $U \cup U'$, der nicht in $U \cap U'$ ist, ist in U.

- Annahme: V ist eine geordnete Menge
- Starte mit $U = U'$
- Iteriere über alle $v \in V \setminus U$ in aufsteigender Reihenfolge
  - 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

Entscheidungsproblem:

Problem

Problem: **Clique**
Eingabe: *Graph G, Parameter $k \in \mathbb{N}$*
Frage: *Existiert eine Clique U der Kardinalität $|U| \geq k$ 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 $\mathcal{NP}$-vollständig.*

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