Web Algorithmen - Ranking 3

Sommersemester 2005

Dr. Michael Brinkmeier

HITS

Der Name

HITS ist ein anderes linkbasiertes Ranking-Verfahren [Kleinberg 99].

Hypertext

Induced

Topic

Search

Wie Topic andeutet, ist es als lokales (themenweises) Ranking konzipiert.

Aber es kann als globales Ranking auf einem Teilgraphen verstanden werden.

Autoritäten und Hubs

Jeder Knoten kann zwei Rollen erfüllen:

- Er kann eine Autorität sein, d.h. Informationen enthalten.
- Er kann ein Hub sein und auf andere Seiten verweisen.
- (Hub = (Rad-)Nabe, Mittelpunkt, Zentrum)

Jeder Knoten kann beide Rollen gleichzeitig ausfüllen.

Ziel ist es, zu bewerten, wie stark/gut er welche Rolle spielt.

Dazu werden jedem Knoten v zwei Werte zugeordnet:

- die Autorität $a(v)$ und
- der Hub-Score $h(v)$.

Autoritäten und Hubs

Beide Werte beeinflussen sich gegenseitig:

- Ein Hub ist so gut, wie die Autoritäten auf die er zeigt.

$$h(v) = \sum_{u \mid v \rightarrow u} a(u)$$

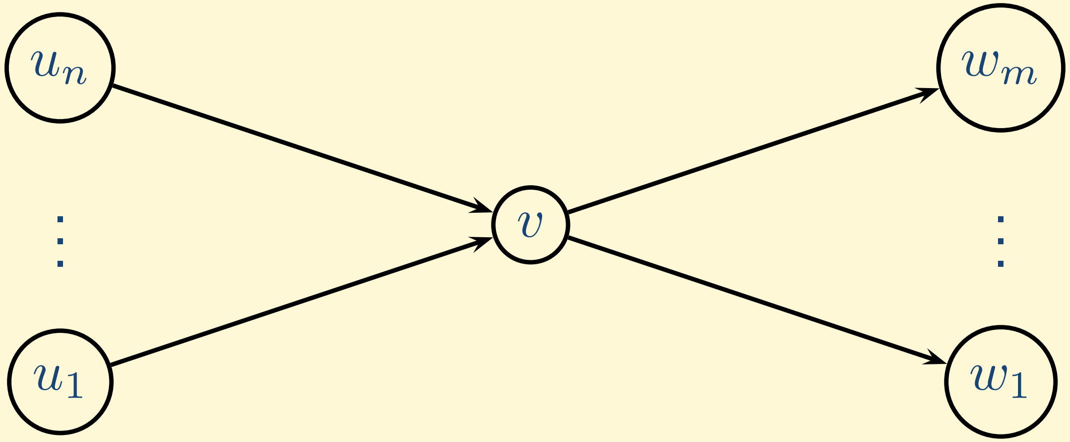
- Eine Autorität ist so gut, wie die Hubs die auf sie zeigen.

$$a(v) = \sum_{u \mid u \rightarrow v} h(u)$$

Man beachte die beiden unterschiedlichen Richtungen!

Frage: Wie können die Werte berechnet werden?

Autoritäten und Hubs



$$a(v) = \sum_{i=1}^{n} h(u_i)$$

$$h(v) = \sum_{i=1}^{m} a(w_i)$$

Die Berechnung

Diesmal benutzen wir die Adjazenzmatrix A.

$$A = (a_{uv})_{u,v \in V} \quad \text{mit} \quad a_{uv} = \begin{cases} 1 & \text{falls } u \longrightarrow v \\ 0 & \text{sonst} \end{cases}$$

D.h. in der Zeile für u steht in Spalte v genau dann eine 1, wenn eine Kante von u nach v existiert.

Es ergibt sich

$$h(v) = \sum_u a_{vu} a(u)$$

$$a(v) = \sum_u a_{uv} h(u)$$

und mittels der Adjazenzmatrix A

$$h = Aa \quad \text{und} \quad a = A^T h.$$

Die Berechnung

Kombination der beiden Gleichungen führt zu:

$$\boxed{h = AA^T h \quad \text{und} \quad a = A^T A a}$$

Damit stellt sich HITS als Eigenvektor-Problem dar:

- a ist ein Eigenvektor von $A^T A$ zum Eigenwert 1.
- h ist ein Eigenvektor von AA^T zum Eigenwert 1.

Frage: Existieren die Vektoren und wie können sie berechnet werden?

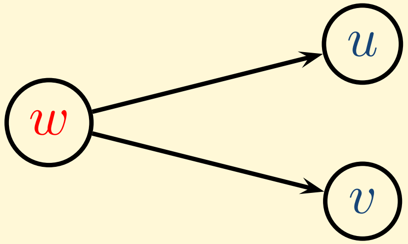
Die Autoritätsmatrix $A^T A$

Die Einträge b_{uv} von $A^T A$ ergeben sich als:

$$b_{uv} = \sum_{w} a_{wu} a_{wv}.$$

Dabei gilt

$$a_{wu} a_{wv} = \begin{cases} 1 & \text{falls } a_{wu} = 1 = a_{wv} \\ 0 & \text{sonst} \end{cases}$$



Damit ist b_{uv} die Anzahl der Knoten w, die sowohl auf u, als auch auf v verlinken.

$\Rightarrow \qquad b_{vv} = \text{in}(v).$

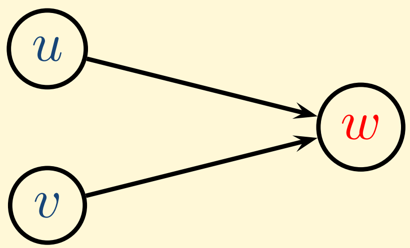
Die Hubmatrix AA^T

Die Einträge c_{uv} von AA^T ergeben sich als:

$$c_{uv} = \sum_{w} a_{uw} a_{vw}.$$

Dabei gilt

$$a_{uw} a_{vw} = \begin{cases} 1 & \text{falls } a_{uw} = 1 = a_{vw} \\ 0 & \text{sonst} \end{cases}$$



Damit ist c_{uv} die Anzahl der Knoten w, die sowohl von u, als auch von v verlinkt werden.

$\Rightarrow \qquad c_{vv} = \mathrm{out}(v).$

Existenz der Lösung

Frage: Existieren Lösungen?

Beobachtung: $A^T A$ und AA^T sind beide symmetrisch, d.h. $b_{uv} = b_{vu}$ und $c_{uv} = c_{vu}$.

Damit gibt es $|V|$ Eigenwerte $\lambda_1, \ldots, \lambda_{|V|}$ mit

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{|V|}|.$$

(Resultat der linearen Algebra)

Problem: Der Eigenwert 1 ist nicht grantiert!

Aber: Kleinberg's ursprünglicher Ansatz ist etwas anders.

Kleinberg's Algorithmus

Kleinberg verwendete von Anfang an einen iterativen Ansatz.

Statt die Gleichungen zu lösen, verwendete er eine Iteration um ein Gleichgewicht für alle Werte zu finden.

Input: Ein Graph $G = (V, E)$ und $k \in \mathbb{N}$

Output: Die Autoritäts- und Hubvektoren a und h

Setze $a = h = (1, \ldots, 1)$
für $1, \ldots, k$ **tue**

$$a = A^T h$$
$$h = Aa$$
$$a = \frac{a}{\|a\|_1}$$
$$h = \frac{h}{\|h\|_1}$$

Ende

Konvergenz der Iteration

Satz: (Potenzmethode)

Sei A eine diagonalisierbare $n \times n$-Matrix mit den Eigenwerten

$$|\lambda_1| = |\lambda_2| = \cdots = |\lambda_r| > |\lambda_{r+1}| \cdots \geq |\lambda_n|$$

und den zugehörigen Eigenvektoren $v_1, \ldots, v_n$. Ferner sei $x^{(0)}$ ein Vektor aus $\mathbb{R}^n$, so dass

$$< x^{(0)}, v_i > \neq 0 \text{ für ein } 1 \leq i \leq r$$

und

$$x^{(i+1)} = \frac{Ax^{(i)}}{\|Ax^{(i)}\|_1},$$

dann existiert ein $c \neq 0$, so dass:

$$\lim_{i \to \infty} x^{(i)} = c \left(< x^{(0)}, v_1 > v_1 + \cdots + < x^{(0)}, v_r > v_r \right)$$

Konvergenz der Iteration

Konsequenz: Defacto wird die folgende Iteration durchgeführt:

$$a^{(0)} = h^{(o)} = (1, \ldots, 1)$$

$$a^{(i+1)} = \frac{A^T h^{(i)}}{\|A^T h^{(i)}\|_1}$$

$$h^{(i+1)} = \frac{A a^{(i+1)}}{\|A a^{(i+1)}\|_1}$$

und somit

$$a^{(i+1)} = \frac{A^T A a^{(i)}}{\|A^T A a^{(i)}\|_1}$$

$$h^{(i+1)} = \frac{A A^T h^{(i)}}{\|A A^T h^{(i)}\|_1}$$

Damit konvergieren h und a gegen einen Eigenvektor des betragsmässig grössten Eigenwertes von $A^T A$ bzw. AA^T.

Konvergenz der Iteration

Problem: Die Lösung ist nicht eindeutig, wenn der grösste Eigenwert mehrfach auftritt.

Dann gibt es einen ganzen Vektorraum von gleichwertigen Rankings.

Aber: Scheint in der Praxis kein Problem zu sein.

Weitere Beobachtung: Die Iteration konvergiert in der Praxis relativ schnell.

Das liegt an dem Verhältnis zwischen grösstem und zweitgrösstem Eigenwert.

Die Iteration konvergiert umso schneller, je kleiner

$$\left|\frac{\lambda_{r+1}}{\lambda_1}\right|$$

ist.

HITS als lokales Ranking

Kleinberg schlug HITS als lokales Ranking vor.

Input: Eine Suchanfrage Q und eine Zahl $d \in \mathbb{N}$

Output: Eine Menge R von Seiten und ein Ranking r auf R

$R =$ Ergebnis der textbasierten Anfrage Q;

für alle $v \in Q$ **tue**
  Füge d verschiedene Vorgänger von v zu R hinzu;
  Füge d verschiedene Nachfolger von v zu R hinzu;
Ende

$r = \mathrm{HITS}(R)$;

Stärken und Schwächen von HITS

Stärken

- Die Seiten können zwei verschiedene Rollen wahrnehmen. Dadurch wird eine differenziertere Betrachtung erlaubt.
- Lokales Ranking
  - Anfrage-spezifisch
  - Einflüsse anderer Themen werden (im Idealfall) ausgeschlossen

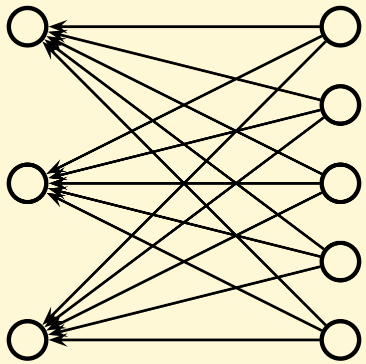
Swächen

- Ranking muss bei jeder Anfrage neu berechnet werden.
- *Tightly Knit Community Effekt*
- Konsequenz: Einfaches Spamming
  - Hub Ranking kann durch ausgehende Links leicht beeinflusst werden.
  - Dadurch wird auch der Autoritätswert der verlinkten Seiten gesteigert.
- Konsequenz: *Topic Drift*

Der Tightly Knit Community Effekt

HITS bevorzugt *dicht vernetzte* Gruppen von Seiten.

Lempel und Moran (2001): HITS hat eine Tendenz dazu, kleine vollständige bipartite Graphen hoch zu bewerten.



Experimente bestätigen diese Tendenz für Gruppen von Seiten im Web, die nicht bipartit sind.

Konsequenz: Spamming

Die eigene Website muss einen vollständigen bipartiten Graphen darstellen.

Weitere Konsequenz: *Topic Drift*

Die Ergebnisse einer Anfrage driften in eine themenfremde, dichtere Community ab.

Beispiele:

- jaguar +car driftet zu car ab (Bharat & Henzinger, 1998)
- Java driftet zu EarthWeb ab (Lempel & Moran, 2000)
- movies driftet zu go.msn.com ab (L&M)
- abortion bevorzugte die pro-life Community (L&M)
- genetics bevorzugte Genetic Algorithms (L&M)

Variationen

- **Bharat & Henzinger, 1998**
  - Normalisiere die verteilten Werte, d.h.

$$h(v) = \sum_{u|v\rightarrow u} \frac{a(u)}{\mathrm{in}(u)} \qquad a(v) = \sum_{u|u\rightarrow v} \frac{h(u)}{\mathrm{out}(u)}.$$

    Dies läuft auf die Verwendung von normalisierten Adjazenzmatrizen hinaus und reduziert den Tightly Knit Community Effekt.

  - Schränke die Menge der zusätzlichen Seiten ein.
    - Bestimme die 100 häufigsten Stichworte des ursprünglichen Ergebnisses.
    - Bewerte die Häufigkeit dieser Stichworte in den benachbarten Seiten.
    - Füge nur Seiten mit hoher Übereinstimmung hinzu.

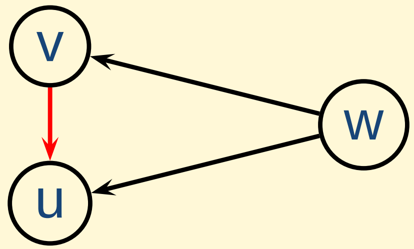
Stochastic **A**pproach for **L**ink **S**tructure **A**nalysis

Lempel & Moran (2000): Kombination von PageRank und HITS

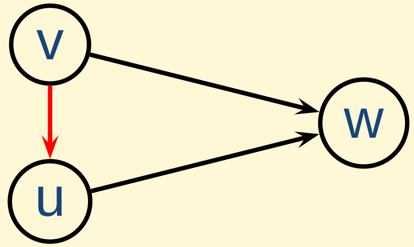
Idee: Führe zwei Zufallsmärsche auf dem Web aus, benutze aber nicht direkt die Links, sondern die Kozitate.

Damit besteht ein einzelner Schritt des Zufalls-Surfers aus zwei Teilschritten:

- **Autoritätenmarsch:**
  - Wähle einen Vorgänger w aus.
  - Wähle einen Nachfolger u von w aus.



- **Hubmarsch:**
  - Wähle einen Nachfolger w aus.
  - Wähle einen Vorgänger u von w aus.



SALSA

Damit ergeben sich die folgenden Übergangswahrscheinlichkeiten:

$$P_A(v,u) = \sum_{w \mid v \leftarrow w \rightarrow u} \frac{1}{\mathrm{in}(v)} \frac{1}{\mathrm{out}(w)}$$

und

$$P_H(v,u) = \sum_{w \mid v \rightarrow w \leftarrow u} \frac{1}{\mathrm{out}(v)} \frac{1}{\mathrm{in}(w)}.$$

Man sieht leicht:

- $P_A(u,v) \neq 0 \quad \Leftrightarrow \quad P_A(v,u) \neq 0$
- $P_H(u,v) \neq 0 \quad \Leftrightarrow \quad P_H(v,u) \neq 0$

v ist also genau dann von u aus erreichbar, wenn auch umgekehrt u von v aus erreicht werden kann.

Damit ergeben sich für beide Zufallsmärsche disjunkte Komponenten von gegenseitig erreichbaren Knoten.

Aber: Lempel und Moran stellten Folgendes fest:

Seien $A_1, \ldots, A_k$ die Komponenten des Autoritätsmarsches, dann ergibt sich das SALSA Autoritäts-Ranking ${}_A(v)$ eines Knotens $v \in A_i$ als

$$r_A(v) = \frac{|V(A_i)|}{|V|} \frac{\text{in}(v)}{|E(A_i)|},$$

wobei $V(A_i)$ die Menge der Knoten in A_i und $E(A_i)$ die Menge der Kanten in A_i ist.

Analog ergibt sich, dass das Hub-Ranking proportional zum Ausgangsgrad ist.

Ergänzungen

Probabilistisches Ranking

Idee: Die Relevanz einer Seite ist proportional zur wahrscheinlichkeit, dass Sie verlinkt wird.

Global betrachtet, kann jeder Link als ein Experiment betrachtet werden. Damit gilt:

$$r(v) = P(* \to v) = \frac{\mathrm{in}(v)}{|E|}.$$

Damit sind wir wieder bei Eingangsgraden.

Aber: Nicht jede Seite kommt als Ziel eines Links in Betracht.

Möglichkeit: Versuche das Ranking themenweise durchzuführen.

Cohn & Chang (2000): Gleichzeitige Unterteilung in *vorgegebene* Themen

Probabilistisches Ranking

Prinzipielle Vorgehensweise:

- **Ausgangspunkt:** Statistisches Modell für das WWW.
- **Relevanz von** v **=** $P(* \to v)$, d.h. WS, dass v verlinkt wird.
- Eventuell themenbasiertes Ranking möglich, d.h.

  $$P(* \to v \mid t) = \text{ WS, dass } v \text{ wegen Thema } t \text{ verlinkt wird.}$$

- Eventuell auch andere Beschränkungen möglich (Bekanntheit, textbasiertes Ranking etc.)
- Bestimmung der Wahrscheinlichkeiten mittels geeigneter Verfahren
  - **Maximum-Likelihood-Methode**, d.h. wähle die WS so, dass die beobachtete Instanz (das WWW) mit möglichst hoher Wahrscheinlichkeit eintritt.
  - Schätze einige Parameter des Modells mit Hilfe von Messungen und berechne daraus die Wahrscheinlichkeiten

Textbasiertes Ranking

- **Inhaltliches Ranking:**
  - Versehe jeden Begriff mit einer Relevanz
  - Messe die Relevanz einer Seite mittels der häufigkeit der Begriffe und ihrer Gewichte
  - **Kritisch:** Wahl der Gewichte der einzelnen Begriffe
  - **Problem:** Relevanz von Begriffen wird sehr individuell empfunden

- **Anfrage-bezogenes Ranking:**
  - Bewerte eine Seite des Ergebnisses bezogen auf die Begriffe in der Suchanfrage
  - Häufigkeit und Position der Begriffe auf der Seite
  - Globale Häufigkeit der Begriffe

Inhalt und linkbasiertes Ranking

Inhaltliche Informationen können in linkbasierte Rankings einfliessen:

- **PageRank**
  - Wähle Personalisierungsvektor mittels textbasiertem Ranking
  - Wähle die Übergangswahrscheinlichkeiten nicht gleichmäßig, sondern mittels
    - Textbasierter Relevanz
    - Textähnlichkeit
  - Verwende nicht (nur) die echten Links, sondern *künstliche* Links, basierend auf der Ähnlichkeit der Inhalte
- **HITS** und andere Verfahren
  - wie oben