

Websuche

Linkanalyse

1

- Viele Dokumente enthalten *Bibliographien* (oder *Referenzen*), d.h. eindeutige *Zitierungen* anderer vorher veröffentlichter Dokumente.
- Bei Verwendung von Zitaten als Links können solche Korpora als Graph betrachtet werden.
- Die Struktur dieses Graphen kann unabhängig vom Inhalt interessante Informationen über die Ähnlichkeit von Dokumenten und die Struktur von Informationen liefern.

2

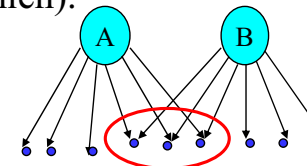
Einflussfaktor (Impact Factor)

- Von Garfield in 1972 entwickelt, um die Bedeutung (Qualität, Einfluss) von wissenschaftlichen Zeitschriften zu messen.
- Maß dafür, wie oft Artikel einer Zeitschrift von anderen Wissenschaftlern zitiert werden.
- Wird jährlich vom Thompson Scientific (<http://www.isinet.com/>) berechnet und veröffentlicht.
- Der *Einflussfaktor* einer Zeitschrift J im Jahr Y ist die durchschnittliche Anzahl von Zitaten (von allen indizierten Dokumenten, die im Jahr Y veröffentlicht wurden) eines Papers, das in J im Jahr $Y-1$ oder $Y-2$ veröffentlicht wurde.
- Berücksichtigt nicht die Qualität des zitierenden Artikels.
- Siehe auch <http://citeseer.ist.psu.edu/impact.html> für einen ähnlichen Index.

3

Bibliographische Kopplung

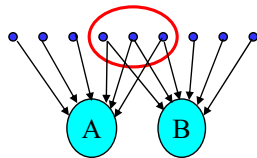
- Maß für die Ähnlichkeit von Dokumenten, das 1963 von Kessler eingeführt wurde.
- Die bibliographische Kopplung von zwei Dokumenten A und B ist die Anzahl der Dokumente, die *sowohl* von A als auch von B zitiert werden, d.h. der Umfang des Durchschnitts ihrer Bibliographien (ggf. normiert durch die Größe der Bibliographien).



4

Ko-Zitation

- Ein alternatives auf Zitaten basierendes Maß der Ähnlichkeit, das 1973 von Small eingeführt wurde.
- Anzahl der Dokumente, die sowohl A als auch B zitieren, ggf. normalisiert durch die gesamte Anzahl von Dokumenten die entweder A oder B zitieren.



5

Zitate im Vergleich zu Links

- Weblinks sind anders als Zitate:
 - Links sind navigationsfähig.
 - Viele Seiten mit hohem In-Grad sind Portale und keine Inhaltsanbieter.
 - Nicht alle Links (aber auch nicht alle Zitate) sind Bestätigungen.
 - Firmenwebseiten verweisen nicht auf ihre Konkurrenten, Zitate relevanter Literatur werden hingegen durch Peer-Reviewing erzwungen.

6

Autoritäten

- *Autoritäten* sind Seiten, die anerkannt sind, und die signifikante, vertrauenswürdige und nützliche Information zu einem Thema zu liefern.
- *In-Grad* (Anzahl von Zeigern auf eine Seite) ist ein einfaches Maß der Autorität.
- Jedoch behandelt ein In-Grad alle Links gleich.
- Sollten nicht Links von Seiten, die selbst Autoritäten sind, mehr zählen?

7

Hubs

- *Hubs* sind Indexseiten, die viele nützliche Links auf relevante Inhaltsseiten (Themenautoritäten) liefern.
- Hubseiten zum Thema “Information Retrieval” sind z.B. unter <http://www.cs.utexas.edu/users/mooney/ir-course> zu finden.

8

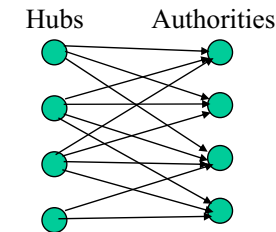
HITS

- Algorithmus, der 1998 von Kleinberg entwickelt wurde.
- Er versucht, Hubs und Autoritäten zu einem bestimmten Thema rechnerisch durch die Analyse eines relevanten Subgraphen des Webs zu bestimmen.
- HITS basiert auf einer rekursiven Definition:
 - Hubs verweisen auf viele Autoritäten.
 - Auf Autoritäten wird von vielen Hubs verwiesen.

9

Hubs und Autoritäten

- Zusammen neigen sie dazu, einen bipartiten Graphen zu bilden:



10

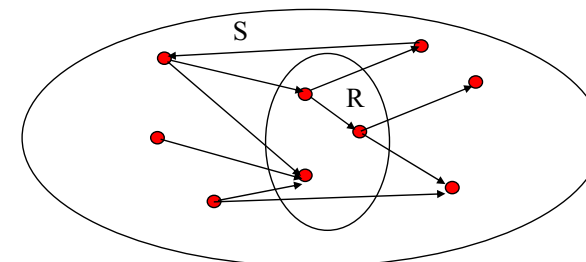
HITS Algorithmus

- Aufgabe: Berechnet Hubs und Autoritäten für ein bestimmtes Thema, das durch eine Anfrage spezifiziert ist.
- Bestimmt zuerst eine Menge relevanter Seiten für die Anfrage, die als *Basis-Menge* S bezeichnet wird.
- Analysiert die Linkstruktur des durch S induzierten Teilgraphen, um Autoritäts- und Hubseiten in dieser Menge zu finden.

11

Konstruieren eines Basis-Subgraphen

- Für eine spezifische Anfrage Q sei die *Wurzel-Menge* R die Menge der von einer Standard-Suchmaschine (z.B. KSM) zurückgegebenen Dokumente.
- $S := R$.
- Füge zu S alle Seiten hinzu, auf die mindestens eine Seite in R verweist.
- Füge zu S alle Seiten hinzu, die auf mindestens eine Seite in R verweisen.



12

Aufwandsbegrenzung

- Um den rechnerischen Aufwand zu limitieren:
 - Begrenze die Anzahl der Wurzelseiten auf die besten 200 Seiten, die für die Anfrage gefunden wurden.
 - Begrenze die Anzahl der “Rückwärts-Link”-Seiten auf eine willkürliche Menge von höchstens 50 Seiten, die von einer “Rückwärts-Link”-Anfrage zurückgegeben wurden.
- Um reine Navigationslinks zu eliminieren:
 - Eliminiere Links zwischen zwei Seiten auf dem gleichen Host.
- Um “nicht-autoritätsfördernde” Links zu eliminieren:
 - Erlaube max. m ($m \cong 4-8$) Seiten von jedem Host als Zeiger auf ein beliebige individuelle Seite.

13

HITS – Iterativer Algorithmus

- Iterativer Algorithmus, der sich langsam einer sich gegenseitig verstärkenden Menge von Hubs und Autoritäten nähert.
- Aufgabe: Bestimme für jede Seite $p \in S$
 - den Autoritätswert a_p (zusammengefasst in einem Vektor \mathbf{a})
 - und den Hubwert h_p (Vektor \mathbf{h})

15

Autorität und In-Grad

- Selbst in der Basismenge S einer gegebenen Anfrage sind die Knoten mit dem höchsten In-Grad nicht notwendigerweise Autoritäten (sondern evtl. nur allgemein bekannte Seiten wie Yahoo oder Amazon).
- Auf ‘wahre’ Autoritätsseiten wird von mehreren Hubs verwiesen (dies sind Seiten, die auf viele Autoritäten verweisen.)

14

HITS-Algorithmus

1. Initialisiere alle $a_p := h_p := 1$
2. Normalisiere die Werte, so dass gilt:
$$\sum_{p \in S} (a_p)^2 = 1 \quad \sum_{p \in S} (h_p)^2 = 1$$
3. Auf Autoritäten wird durch viele gute Hubs verwiesen:

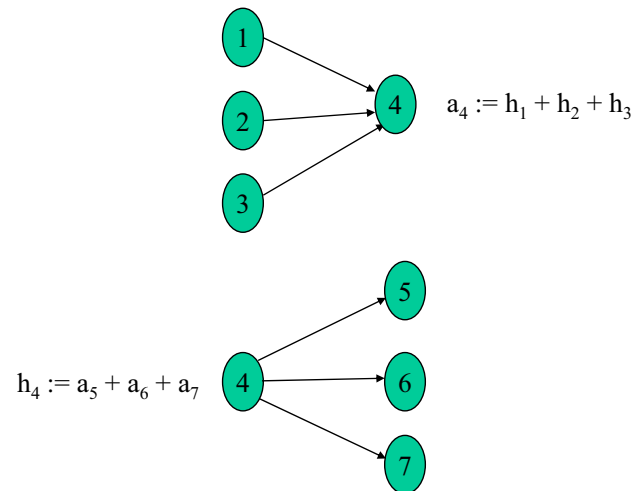
$$a_p = \sum_{q:q \rightarrow p} h_q$$

4. Hubs verweisen auf viele gute Autoritäten:

$$h_p = \sum_{q:p \rightarrow q} a_q$$

5. Solange die Vektoren sich (signifikant) ändern, gehe zu Schritt 2.

Illustrierte Update-Regeln



17

Darstellung in linearer Algebra

- Definiere A als Adjazenzmatrix für den durch S induzierten Subgraphen.
 - $A_{ij} = 1$ für $i \in S, j \in S$ gdw. $i \rightarrow j$ im Graphen.
- Die Autoritätswerte a_p werden in einem Vektor \mathbf{a} zusammengefasst, und die Hubwerte h_p in einem Vektor \mathbf{h} .
- Die Schritte der Iteration ergeben sich zu
 - $\mathbf{h} := A\mathbf{a}$
 - $\mathbf{a} := A^T\mathbf{h}$

19

HITS im Detail

Initialisiere für alle $p \in S$: $a_p := h_p := 1$

Bis Änderung kleiner als gegebener Schwellwert:

Für alle $p \in S$:
/* aktualisiere Autoritätswerte */
$$a_p := \sum_{q:q \rightarrow p} h_q$$

Für alle $p \in S$:
/* aktualisiere Hubwerte */
$$h_p := \sum_{q:p \rightarrow q} a_q$$

Für alle $p \in S$: $a_p := a_p / c$ mit $c := \sqrt{\sum_{p \in S} a_p^2}$
/* \mathbf{a} normalisieren */

Für alle $p \in S$: $h_p := h_p / c$ mit $c := \sqrt{\sum_{p \in S} h_p^2}$
/* \mathbf{h} normalisieren */

18

Konvergenz

- Algorithmus konvergiert zu einem *Fixpunkt*, falls unendlich wiederholt.
- Autoritätsvektor \mathbf{a} konvergiert gegen den ersten Eigenvektor von $A^T A$.
- Hubvektor, \mathbf{h} , konvergiert gegen den ersten Eigenvektor von $A A^T$.
- In der Praxis liefern 20 Wiederholungen ziemlich stabile Ergebnisse.

20

Ergebnisse

- Autoritäten für Anfrage “Java”
 - java.sun.com
 - [comp.lang.java FAQ](#)
- Autoritäten für Anfrage “search engine”
 - Yahoo.com
 - Excite.com
 - Lycos.com
 - Altavista.com
- Autoritäten für Anfrage “Gates”
 - Microsoft.com
 - roadahead.com

(Nach [Kleinberg 1998])

21

Beobachtung

- In den meisten Fällen waren die endgültigen Autoritäten nicht in der anfänglichen Wurzelmenge, die mit Altavista bestimmt wurde.
- Autoritäten wurden durch Vor- und Rückwärtslinks hinzugefügt (und dann durch HITS als Autorität bestimmt).

22

Finden ähnlicher Seiten durch Verwendung der Linkstruktur

- Aufgabe: Bestimmung ähnlicher Seiten zu einer Seite P . (Dieser Ansatz findet Autoritäten in der “Link-Nachbarschaft” von P .)
- Sei t gegeben (z.B. $t = 200$).
- Sei R eine Menge von t Seiten, die auf P verweisen (die Wurzelmenge).
- Bestimme die Basismenge S von R wie o.a.
- Lasse HITS auf S laufen.
- Gebe die besten Autoritäten in S als die “ähnlichsten Seiten von P ” zurück.

23

Ergebnisse der Ähnlichkeitssuche

- Gegeben “honda.com”
 - toyota.com
 - ford.com
 - bmwusa.com
 - saturncars.com
 - nissanmotors.com
 - audi.com
 - volvocars.com

24

PageRank

- Alternative Link-Analyse-Methode, die von Google verwendet wird (Brin & Page, 1998).
- Versucht nicht, die Unterscheidung zwischen Hubs und Autoritäten zu erfassen, sondern klassifiziert Seiten nur nach Autorität.
- Wird eher auf das gesamten Web angewandt als auf eine lokale Nachbarschaft von Seiten, die die Ergebnisse einer Anfrage umgeben.

25

Grund-Idee PageRank

- Die Messung des In-Grades alleine (Zitatzählung) berücksichtigt nicht die Autorität der Quelle eines Links.
- (Vereinfachte) PageRank-Gleichung für Seite p :

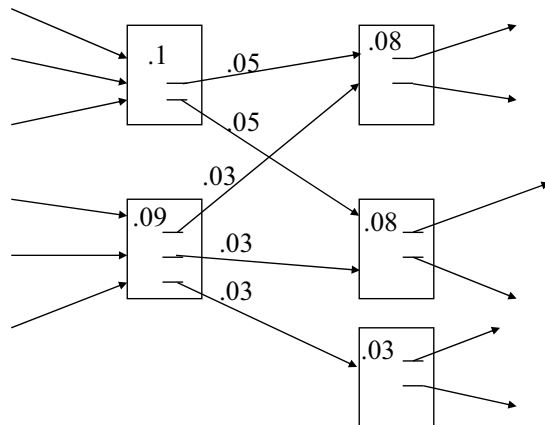
$$R(p) = \sum_{q:q \rightarrow p} \frac{R(q)}{N_q}$$

- N_q ist die Gesamtzahl der Out-Links von Seite q .
- Eine Seite q gibt einen gleichen Anteil ihrer Autorität an alle Seiten weiter, auf die sie verweist (z.B. auf p).

26

Grund-Idee PageRank

- PageRank “fließt” entlang der Kanten:



27

Grundidee PageRank

- Wiederhole den Fluss-Prozess bis zur Konvergenz:

Sei S die Gesamtmenge der Seiten.

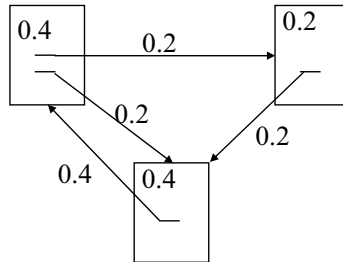
Initialisiere für alle $p \in S$: $R(p) = 1/|S|$

Bis sich Werte nicht mehr (viel) ändern (**Konvergenz**)

$$\text{Für jedes } p \in S: R'(p) = \sum_{q:q \rightarrow p} \frac{R(q)}{N_q}$$

28

Beispiel: stabiler Fixpunkt



29

Lineare-Algebra-Version

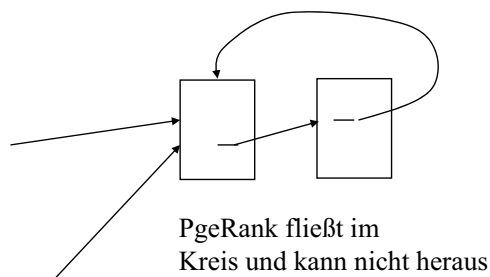
- Betrachte $\mathbf{r} := (R(p))_{p \in S}$ als einen Vektor in $\mathbb{R}^{|S|}$.
- Sei \mathbf{A} die $|S| \times |S|$ -Matrix mit

$$\mathbf{A}_{vu} := 1/N_u \text{ falls } u \rightarrow v, \text{ und } \mathbf{A}_{vu} := 0 \text{ sonst.}$$
- Dann gilt am Ende des Algorithmus $\mathbf{r} = \mathbf{A}\mathbf{r}$, d.h. \mathbf{r} konvergiert zu dem Eigenvektor von \mathbf{A} , der zum Eigenwert 1 gehört.

30

Problem mit anfänglicher Idee

- Eine Gruppe von Seiten, die nur auf sich selbst verweist, aber auf die durch andere Seiten verwiesen wird, agiert als eine Gewichts-Senke, die das ganze Gewicht absorbiert.
- “Suchmaschinenoptimierer” nutzen diesen Effekt in “Linkfarmen” aus.



31

Gewichts-Quelle

- Führe eine Gewichts-Quelle E ein, die kontinuierlich den Rank jeder Seite p durch einen festen Betrag $E(p)$ ergänzt.

$$R(p) = \alpha \sum_{q:q \rightarrow p} \frac{R(q)}{N_q} + (1 - \alpha)E(p)$$

32

PageRank-Algorithmus

Sei S die Gesamtmenge der Seiten.

Sei $\alpha \in (0,1)$, z.B. $\alpha = 0.85$.

Für alle $p \in S$: $E(p) := 1/|S|$

Für alle $p \in S$ initialisiere $R(p) := 1/|S|$.

Bis sich die Gewichte nicht mehr (viel) ändern (*Konvergenz*):

$$R(p) = \alpha \sum_{q:q \rightarrow p} \frac{R(q)}{N_q} + (1-\alpha)E(p)$$

33

Lineare Algebraversion

- Nach Konvergenz gilt $\mathbf{r} = \alpha \mathbf{A} \mathbf{r} + (1-\alpha) \mathbf{E}$.
- Wegen $\|\mathbf{r}\|_1 = 1$ gilt $\mathbf{r} = c(\alpha \mathbf{A} + (1-\alpha) \mathbf{E} \times \mathbf{1}) \mathbf{r}$
wobei $\mathbf{1}$ der Vektor ist, der nur aus 1ern besteht.
- Somit ist \mathbf{r} ein Eigenvektor von $\alpha \mathbf{A} + (1-\alpha) \mathbf{E} \times \mathbf{1}$.

34

Random-Surfer-Modell

- PageRank kann als Modellierung eines “willkürlichen Surfers” betrachtet werden, der auf einer beliebigen Seite startet und dann entweder
 - mit der Wahrscheinlichkeit $E(p)$ willkürlich auf die Seite p springt
 - oder willkürlich einem Link auf der aktuellen Seite folgt.
- $R(p)$ modelliert dann die Wahrscheinlichkeit, dass sich dieser willkürliche Surfer zu jeder gegebenen Zeit auf der Seite p befindet.
- Die “Sprünge” in \mathbf{E} werden benötigt, um zu vermeiden, dass der willkürliche Surfer in Web-Senken “gefangen” wird, aus denen kein Link herausführt.

35

Konvergenz

- Frühe Experimente in Google verwendeten 322 Millionen Links.
- PageRank-Algorithmus konvergiert (mit einer kleinen Toleranz) in ca. 52 Wiederholungen.
- Die Anzahl der für Konvergenz erforderlichen Wiederholungen ist empirisch $O(\log n)$ (wobei n die Anzahl der Links ist).
- Daher ist die Berechnung ziemlich effizient.

36

Einfache Titelsuche mit PageRank

- Verwende zunächst die einfache Boolesche Suche, um Titel von Webseiten zu suchen und klassifiziere die gefundenen Seiten dann nach ihrem PageRank.
- Beispiel-Suche nach “Universität” (aus [Page, Brin 1998]):
 - Altavista gab eine beliebige Menge von Seiten mit “Universität” im Titel wieder (schien kurze URLs zu bevorzugen).
 - Primitives Google gab die Homepages der erstklassigen amerikanischen Universitäten wieder.

37

Google-Suche

- Komplette Google-Suche umfasste vor der Kommerzialisierung (basierend auf wissenschaftlichen Veröffentlichungen):
 - Vektorraummodell
 - Abstandsmaß zu Schlüsselwörtern
 - HTML-Tag-Gewichtung (z.B. Titelpräferenz)
 - PageRank
- Details zu aktuellen Google-Komponenten sind Betriebsgeheimnisse.

38

HTML-Struktur & Merkmalgewichtung

- Gewichte ggf. Tokens unter bestimmten HTML- Tags stärker:
 - `<TITLE>` Token (Google scheint Titelübereinstimmungen zu mögen)
 - `<H1>`, `<H2>`... Token
 - `<META>` Schlüsselwort-Token
- Zerlege eine Seite in verschiedene Abschnitte (z.B. Navigationsleiste und Seiteninhalt) und gewichte auf den unterschiedlichen Abschnitten basierende Token unterschiedlich.

Nachtrag zu Kap. 4 Modelle

39

Personalisierter PageRank

- PageRank kann durch Ändern von \mathbf{E} beeinflusst (personalisiert) werden: Beschränken des “Random Surfers” auf eine Menge als relevant spezifizierter Seiten.
- Zum Beispiel durch Setzen von $E(p) := 0$, außer auf der eigenen Homepage, wo $E(p) := \alpha$
- Dies führt zu einer Ausrichtung auf Seiten, die im Webgraphen näher zu der eigenen Homepage sind.

40

PageRank-basiertes Spidering

- Verwende PageRank, um den Spider auf “wichtige” Seiten zu leiten (zu fokussieren).
- Berechne PageRank unter Verwendung der aktuellen Menge der bearbeiteten Seiten.
- Sortiere die Anfrage-Warteschlange des Spiders auf der Basis des aktuell geschätzten PageRanks.

41

Schlussfolgerungen zur Linkanalyse

- Die Linkanalyse verwendet als Suchhilfe Informationen über die Struktur des Webgraphen.
- Dies ist eine der wesentlichen Innovationen bei der Websuche
- ... und der primäre Grund für den Erfolg von Google.

42