

Hyperlink Induced Topic Search (HITS)

Ekaterina Tikhoncheva

Seminar „Information Retrieval“

Universität Heidelberg

19.01.2014

Agenda

- 1 Einführung
- 2 Grundidee
- 3 Algorithmus
 - Schritt 1
 - Schritt 2
 - Konvergenz
- 4 Ergebnisse
- 5 Erweiterungen des Algorithmus
- 6 HITS vs PageRank
- 7 Zusammenfassung
 - Modifikationen

Einführung

- HITS ist ein Algorithmus zum Finden und Bestimmen der Rangordnung der zu einer gegebenen Suchanfrage relevanten Seiten im WWW*
- Entwickelt von Jon M. Kleinberg, 1997 IBM Research Report RJ 10076
- Idee: finde angesehene Informationsquellen (engl. **authorities**) zu der gegebenen Suchanfrage
- Mathematischer Hintergrund: Eigenvektoren der mit dem Web-Graphen verbundenen Matrix

*in einer Hyperlinked-Umgebung

Drei exemplarische Typen von Suchanfragen

- spezielle:
„Does Netscape support the JDK 1.1 - code-signing API?“[6]
- breit angelegte:
„Find information about the Java programming language“[6]
- Suche nach ähnlichen Seiten
„Find pages similar to *java.sun.com*“[6]

Drei exemplarische Typen von Suchanfragen

- spezielle: **Problem der Knappheit**
„Does Netscape support the JDK 1.1 - code-signing API?“[6]
- breit angelegte:
„Find information about the Java programming language“[6]
- Suche nach ähnlichen Seiten
„Find pages similar to *java.sun.com*“[6]

Drei exemplarische Typen von Suchanfragen

- spezielle: **Problem der Knappheit**
„Does Netscape support the JDK 1.1 - code-signing API?“[6]
- breit angelegte: **Problem der Vielfältigkeit**
„Find information about the Java programming language“[6]
- Suche nach ähnlichen Seiten
„Find pages similar to *java.sun.com*“[6]

Ranking

- Man möchte die angesehensten Seiten (Authorities) aus der Menge aller zu der Anfrage relevanten Seiten finden
- Mögliche Hindernisse:
 - die höchst relevanten Seiten werden nicht unbedingt durch ein textbasiertes Ranking vorgezogen
 - es kann sein, dass die relevanten Seiten die Wörter aus der Suchanfrage gar nicht enthalten

Authorities and Hubs I

Annahme

Die Relevanz zwischen zwei Seiten wurde vom Ersteller des Links zwischen diesen Seiten geprüft

Stimmt im Allgemeinen nicht (z.B. Navigationslinks, Werbung)

Aber unter dieser Annahme reicht es, nur die Linkstruktur des WWW zu betrachten, um die Autorität einer Seite im Bezug auf eine andere zu bestimmen

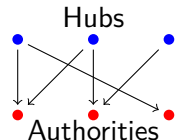
Authorities and Hubs II

Authorities (Autoritätsseiten)

Relevante Seiten, auf die viele weitere relevante Seiten zeigen

Problem: populäre Seiten kommen immer als Authorities vor

Es muss eine Überlappung in den Links, die auf Authorities zeigen, berücksichtigt werden



Hubs

Seiten, die auf viele Authorities zeigen

HITS Algorithmus

Der Algorithmus besteht aus zwei Schritten:

- 1 Konstruiere einen auf die Suchanfrage fokussierten Teilgraphen des WWW
- 2 Berechne rekursiv die Authority- und Hubgewichte für alle Knoten im Teilgraphen aus Schritt 1

Teilgraph des WWW

Wir stellen das ganze Web als gerichteten Graphen $G = (V, E)$ dar. Gesucht wird ein auf der Anfrage basierter Teilgraph des WWW

Sei eine Suchanfrage σ gegeben. Wir bezeichnen mit Q_σ das Ergebnis der textbasierten Suche

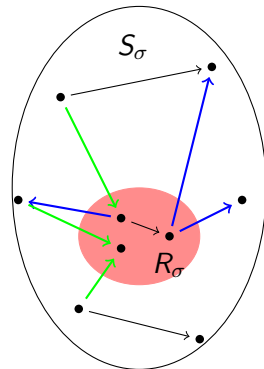
- Q_σ ist groß
- Gute Authorities müssen nicht in Q_σ liegen

Gesucht wird eine Menge S_σ von Seiten, die

- klein ist
- relevante Seiten enthält
- viel gute Authorities enthält

Algorithmus 1 Teilgraphen(σ, E, t, d) [6]

- 1: R_σ ist die Menge der t besten Ergebnisse des textbasierten Suchalgorithmus E angewendet auf die Anfrage (engl. **root set**) σ
 - 2: $S_\sigma := R_\sigma$
 - 3: **for** p in R_σ **do**
 - 4: $S_\sigma = S_\sigma \cup \{ \text{Menge der Seiten, auf die } p \text{ zeigt} \}$
 - 5: $S_\sigma = S_\sigma \cup \{ \text{Menge der } d \text{ Seiten, die auf } p \text{ zeigen} \}$
 - 6: **end for**
-



Bemerkungen

Algorithmus 1 Teilgraphen(σ, E, t, d)[6]

- 1: R_σ ist die Menge der **t** besten Ergebnisse des textbasierten Suchalgorithmus E angewendet auf die Anfrage (engl. **root set**)
- 2: $S_\sigma := R_\sigma$
- 3: **for** p in R_σ **do**
- 4: $S_\sigma = S_\sigma \cup \{ \text{Menge der Seiten, auf die } p \text{ zeigt} \}$
- 5: $S_\sigma = S_\sigma \cup \{ \text{Menge der } d \text{ Seiten, die auf } p \text{ zeigen} \}$
- 6: **end for**

In der Implementierung vom Autor [6]:

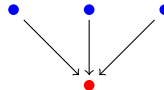
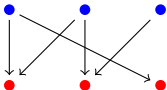
- $t = 200$
- $d = 50$
- $|S_\sigma| \sim O(10^3)$
- Lösche Link zwischen den gleichen Domains (Navigationslinks)
- Beschränke Anzahl der Seiten aus einer Domain, die auf die gleiche Seite verweisen (z.B. Werbung)

Berechnen von Authorities und Hubs

Der erste Schritt erlaubt es uns, sich auf die Menge S_σ der Seiten zu konzentrieren, die viele angesehene Seiten (Authorities) enthält und gleichzeitig klein genug ist, um die nicht trivialen Algorithmen anwenden zu können

Annahmen

- Ein guter Hub zeigt auf viele gute Authorities
- Eine gute Autorität ist das Ziel von vielen guten Hubs



Sei Graph $G = G[S_\sigma]$ aus dem Schritt 1 gegeben. Wir bezeichnen mit A die Adjazentmatrix des Graphen G

Für jede Seite $p \in V$ werden ein nicht negatives **Autoritätsgewicht** $x(p)$ und nicht negatives **Hubgewicht** $y(p)$ definiert, so dass es

$\sum_{p \in V(G)} (x(p))^2 = 1$ und $\sum_{p \in V(G)} (y(p))^2 = 1$ gelten

Wir betrachten weiter die Gewichte $\{x(p) | p \in V(G)\}$ und $\{y(p) | p \in V(G)\}$ als n - dimensionale Vektoren x und y ($n = |V(G)|$), wobei jeder Eintrag dem Gewicht $x(p)$ ($y(p)$) der Seite p entspricht, $p \in V(G)$

Es werden zwei Operationen definiert:

$$\mathcal{I}(y) = \sum_{q:(q,p) \in E(G)} y(q) = A^T y$$

$$\mathcal{O}(x) = \sum_{q:(p,q) \in E(G)} x(q) = Ax$$

Algorithmus 2 Iteriere(G, k)[6]

```

1:  $z = (1, \dots, 1) \in \mathbb{R}^n$ 
2:  $x_0 := z$ 
3:  $y_0 := z$ 
4: for  $i = 1, \dots, k$  do
5:    $x'_i := \mathcal{I}(y_{i-1})$ 
6:    $y'_i := \mathcal{O}(x'_i)$ 
7:   normiere  $x'_i \rightarrow x_i$ 
        $\sum_{p \in V(G)} (x_i(p))^2 = 1$ 
8:   normiere  $y'_i \rightarrow y_i$ 
        $\sum_{p \in V(G)} (y_i(p))^2 = 1$ 
9: end for
10: return  $(x_k, y_k)$ 

```

Algorithmus 3 Filter(G, k, c)[6]

```

1:  $(x_k, y_k) = \text{Iteriere}(G, k)$ 
2: Authorities  $:=$ 
   {Seiten mit  $c$  größten Einträgen von  $x_k$ }
3: Hubs  $:=$ 
   {Seiten mit  $c$  größten Einträgen von  $y_k$ }
4: return (Authorities, Hubs)

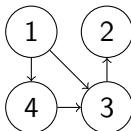
```

$x_i, y_i \in \mathbb{R}^n$ bezeichnen Authority- und Hubgewichte nach der i -ten Iteration des Algorithmus

$$\mathcal{I}(y_i) = \sum_{q: qp \in E(G)} y_i(q)$$

$$\mathcal{O}(x_i) = \sum_{q: pq \in E(G)} x_i(q)$$

Beispiel



$$\text{Adjazenzmatrix } A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

x_0	(1,1,1,1)	y_0	(1,1,1,1)
x_1	(0, 0.41, 0.82, 0.41)	y_1	(0.80, 0, 0.27, 0.53)
x_2	(0, 0.17, 0.85, 0.51)	y_2	(0.84, 0, 0.11, 0.53)
x_3	(0, 0.07, 0.85, 0.52)	y_3	(0.85, 0, 0.04, 0.53)
x_4	(0, 0.03, 0.85, 0.53)	y_4	(0.85, 0, 0.02, 0.53)
x_5	(0, 0.01, 0.85, 0.53)	y_5	(0.85, 0, 0.01, 0.53)
x_6	(0, 0, 0.85, 0.53)	y_6	(0.85, 0, 0, 0.53)
x_7	(0, 0, 0.85, 0.53)	y_7	(0.85, 0, 0, 0.53)

Konvergenz

Aus dem Algorithmus folgt $y_k = (AA^T)^k z$

Sei $w_1(AA^T)$ der Haupteigenvektor von AA^T , d.h. Vektor zu dem betragsgrößten Eigenwert

Man kann zeigen, dass der Vektor z zu $w_1(AA^T)$ nicht orthogonal ist, und da die Matrix AA^T symmetrisch ist, konvergiert die Folge $\{y_k\}$ gegen $w_1(AA^T)$ (Power Iteration, Potenzmethode [4]). Außerdem hat $w_1(AA^T)$ nur positive Einträge (siehe Perron-Frobenius Theorem)

Analog konvergiert $x_k = (A^T A)^{k-1} A^T z$ gegen $w_1(A^T A)$

In der Praxis lässt man den Algorithmus nicht bis zu Konvergenz laufen, sondern führt eine feste Anzahl k^* von Iterationen durch

* $k=20$ sei ausreichend [6]

Beispiel (Fortsetzung)

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$A^T A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Eigenwerte

$$\lambda_1 = 0$$

$$\lambda_1 = 0.38$$

$$\lambda_1 = 1$$

$$\lambda_1 = 2.62$$

Eigenvektoren

$$(1, 0, 0, 0)$$

$$(0, 0, -0.53, 0.85)$$

$$(0, 1, 0, 0)$$

$$(0, 0, 0.85, 0.53)$$

$$AA^T = \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Eigenwerte

$$\lambda_1 = 0$$

$$\lambda_1 = 0.38$$

$$\lambda_1 = 1$$

$$\lambda_1 = 2.62$$

Eigenvektoren

$$(0, 1, 0, 0)$$

$$(-0.53, 0, 0, 0.85)$$

$$(0, 0, 1, 0)$$

$$(0.85, 0, 0, 0.53)$$

Somit sehen wir, dass die Folgen $\{x_k\}$ und $\{y_k\}$ konvergieren
entsprechend gegen $w_1(A^T A)$ und $w_1(AA^T)$.

Ergebnisse: Suchanfrage „Gates“ I

HITS [6]

(Gates) Authorities

.643 <http://www.roadahead.com/>

.458 <http://www.microsoft.com/>

.440 <http://www.microsoft.com/corpinfo/bill-g.htm>

Bill Gates: The Road Ahead

Welcome to Microsoft

www.ask.com

http://www.ask.com/wiki/Bill_Gates?lang=en

www.gates.com/

www.gates.com/all-search-tools/automotive-interchange-search-results

usmilitary.about.com/od/glossaryterms/g/g2641.htm

www.gatesfoundation.org/

Gates Corporation

Gates Corporation (manufacture)

gate

Bill & Melinda Gates Foundation

Ergebnisse: Suchanfrage „Gates“ II

www.google.com

www.gates.com/	Gates Corporation
ww2.gates.com/germany/	Gates Corporation
de.wikipedia.org/wiki/Bill_Gates	
http://www.gatesbbq.com	Gates Bar B. Q.
http://www.gatesfoundation.org	Gates Foundation

www.search.yahoo.com

www.gates.com/	Gates Corporation
www.gateschevyworld.com	Gates Chevy World - Mishawaka, Elkhart, South Bend
www.gatecrafters.com/design_your_gates.aspx	Driveway Gates Automatic Gates Electric Gates
www.gates.com/industries/automotive	Gates Automotive Aftermarket Solutions Gates Corporation
http://www.gatesbbq.com	Gates Bar B. Q.

Ergebnisse: Suchanfrage „Java“ I

HITS [6]

(java) Authorities	<i>Gamelan</i>
.328 http://www.gamelan.com/	<i>JavaSoft Home Page</i>
.251 http://java.sun.com/	<i>The Java Developer: How Do I...</i>
.190 http://www.digitalfocus.com/digitalfocus/faq/howdoi.html	<i>The Java Book Pages</i>
.190 http://lightyear.ncsa.uiuc.edu/~srp/java/javabooks.html	<i>comp.lang.java FAQ</i>
.183 http://sunsite.unc.edu/javafaq/javafaq.html	

www.ask.com

www.ehow.com	What is Java?
www.ask.com/wiki/Java_(programming_language)?lang=en	Wiki
java.com/download	Download java
java.about.com/	about.com
www.oracle.com/technetwork/java/index.html	ORACLE

Ergebnisse: Suchanfrage „Java“ II

www.google.com

www.java.com/de/download/
www.java.com/de/
www.java.com/download
www.java.com/en/
[de.wikipedia.org/wiki/Java_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Java_(Programmiersprache))

Download der kostenlosen Java-Software
java.com: Java + Sie
Download Free Java Software
java.com: Java + You

www.search.yahoo.com

www.java.com
java.com/en/download
www.oracle.com/technetwork/es/java/javase/overview/index.html
windows.microsoft.com/en-US/internet-explorer/install-java
www.java.net

Java - Official Site
Download Java
Java SE - Downloads
Install Java in Internet Explorer
java.net - Official Site

Bewertung der Ergebnisse

In seiner Arbeit [6] gibt Kleinberg folgende Evaluationsergebnisse an:

Es wurden die Suchergebnisse von Yahoo!, CLEVER*[2] und AltaVista[1] verglichen, indem 37 Freiwillige die Qualität der von verschiedenen Suchmaschinen vorgeschlagenen Seiten zu den fest gewählten Themen abschätzen sollten

Die Ergebnisse sind:

In 50% der Fälle: CLEVER war besser.

In 31% der Fälle: Yahoo! und CLEVER waren gleich gut.

In 19% der Fälle: Yahoo! war besser.

*Implementation von HITS mit Erweiterungen

Suche nach ähnlichen Seiten

- Die Suchanfrage σ sei eine Internetseite
 $R_\sigma = \{\text{Menge der Seiten, die auf } \sigma \text{ zeigen}\}$
- Wende den HITS Algorithmus an

(www.honda.com) Authorities

.202 <http://www.toyota.com/>

.199 <http://www.honda.com/>

.192 <http://www.ford.com/>

.173 <http://www.bmwusa.com/>

.162 <http://www.volvocars.com/>

.158 <http://www.saturncars.com/>

.155 <http://www.nissanmotors.com/>

.145 <http://www.audi.com/>

.139 <http://www.4adodge.com/>

.136 <http://www.chryslercars.com/>

Welcome to @Toyota

Honda

Ford Motor Company

BMW of North America, Inc.

VOLVO

Welcome to the Saturn Web Site

NISSAN - ENJOY THE RIDE

Audi Homepage

1997 Dodge Site

Welcome to Chrysler

Mehrere Mengen von Hubs und Authorities

- Der betrachtete Algorithmus liefert die am dichtesten vernetzte Menge der Authorities und Hubs im auf der Suchanfrage basierten Teilgraphen des WWWs
- Es gibt eventuell weitere dicht vernetzte Mengen, die zu der Anfrage relevant sind (z.B. mehrdeutige Anfrage)
- betrachte weitere Eigenvektoren, sortiert nach Betragsgrößen
- betrachte positive und negative Teile eines Nicht-Haupteigenvektors

HITS

- wird auf einem Teilgraphen des WWW angewendet
- einfacher, aber langsam in Echtzeit
- berechnet Autoritätsseiten und Hubs
- beruht auf textbasierter Suche
- kann „diffundieren“

PageRank

- benutzt ganzen Graphen des WWW
- dauert lange, wird aber nur einmal beim Indexieren berechnet
- berechnet nur Autoritätsseiten
- unabhängig vom Kontext

Zusammenfassung: I

- Wegen der hohen Anzahl von Webseiten ist bei der Internetsuche ein Ranking-Verfahren sehr nützlich
- Als Ergebnis der Suche werden t Autoritätsseiten zurückgeliefert
- Jeder Link enthält einen gewissen Anteil der Autorität des Erstellers des Links
- Autoritätsseiten sind zu der Anfrage relevante Seiten mit vielen eingehenden Links
- Hubs sind Seiten, die auf viele Autoritätsseiten verlinken
- Der ganze WWW-Graph ist zu groß \Rightarrow betrachte auf der Anfrage basierten Teilgraphen

Zusammenfassung: II

- HITS-Algorithmus
 - arbeitet auf dem generierten Teilgraphen
 - ist ein rekursiver Algorithmus
 - konvergiert gegen Haupteigenvektoren der Matrizen $A^T A$ und AA^T , wobei A die Adjazentmatrix des betrachteten Teilgraphen ist
 - liefert Seiten zurück, die den c größten Einträgen des Haupteigenvektors der Matrix $A^T A$ entsprechen
 - kann für die Suche nach ähnlichen Seiten benutzt werden
 - kann die für verschiedene Bedeutungen der Anfrage relevanten Seiten liefern
- HITS wurde in der Suchmaschine CLEVER (IBM Project) implementiert (nicht mehr in Betrieb)
- HITS wird in der Suchmaschine Ask.com benutzt ([10])

Modifikationen

- Problem :
 - Topic Drift
 - Ergebnisse bei bestimmten Strukturen des Webs sind nicht intuitiv klar [8]
- Modifikationen des Algorithmus
 - Gewichteter HITS-Algorithmus ([3, 7, 8])
 - HITS-Algorithmus mit potenzierte Eingabe ([8])

The End

Vielen Dank für Ihre Aufmerksamkeit!



References I

- [1] Altavista. www.de.wikipedia.org/wiki/AltaVista. Accessed: 2015-15-01.
- [2] CLEVER project.
www.en.wikipedia.org/wiki/CLEVER_project. Accessed: 2015-15-01.
- [3] Krishna Bharat and M. R. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.
- [4] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

References II

- [5] Joni Pajarinen. The pagerank/hits algorithms. www.cis.hut.fi/Opinnot/T-61.6020/2008/pagerank_hits.pdf. Accessed: 2015-15-01.
- [6] John M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.
- [7] Yi Shang Li, Longzhuang and Wei Zhang. Improvement of HITS-based algorithms on web documents. *Proceedings of the 11th international conference on World Wide Web*. ACM, 2002.

References III

- [8] J. C Miller, G. Rae, and F. Schaefer. Modifications of Kleinberg's HITS Algorithm Using Matrix Exponentiation and Web Log Records. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 444–445, 2001.
- [9] Jannik Strötgen. Beamer theme.
<http://dbs.ifi.uni-heidelberg.de/fileadmin/Team/jannik/beamerthemeHeidelberg.sty>. Accessed: 2015-15-01.

References IV

- [10] Raluca Tanase and Remus Radu. The mathematics of web search, lecture 4: Hits algorithm - hubs and authorities on the internet. www.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture4/lecture4.html. Accessed: 2015-15-01.
- [11] Zhe Zhao. Authoritative sources in a hyperlinked environment, Jon M. Kleinberg. web.eecs.umich.edu/~michjc/eecs584/notes/lecture19-kleinberg.pdf. Accessed: 2015-15-01.