Kapitel 4: Linkanalyse für Autoritäts-Ranking

4.1 Page-Rank-Verfahren
4.2 Exkurs: Grundlagen aus der Stochastik
4.3 HITS-Verfahren
4.4 Themenspezifisches Page-Rank-Verfahren

Verbessertes Ranking durch Autoritäts-Scores

Ziel:
Höheres Ranking von URLs mit hoher Autorität bzgl.
Umfang, Signifikanz, Aktualität und Korrektheit von Information
→ verbesserte Präzision von Suchresultaten

Ansätze (mit Interpretation des Web als gerichtetem Graphen G):
• Citation- oder Impact-Rank (q) ~ indegree (q)
• Page-Rank (nach Lawrence Page)
• HITS-Algorithmus (nach Jon Kleinberg)

Kombination von Relevanz- und Autoritäts-Ranking:
• gewichtete Summe mit geeigneten Koeffizienten (Google)
• initiales Relevanz-Ranking und iterative
  Verbesserung durch Autoritäts-Ranking (HITS)

4.1 Page-Rank r(q)

gegeben: gerichteter Web-Graph G=(V,E) mit |V|=n und
Adjazenzmatrix A: $A_{ij} = 1$ falls $(i,j) \in E$, 0 sonst

Idee: $r(q) \approx k \sum_{(p,q) \in G} r(p) / outdegree(p)$

Def.: $r(q) = \varepsilon / n + (1-\varepsilon) \sum_{(p,q) \in G} r(p) / outdegree(p)$ mit $0 < \varepsilon \le 0.25$

Satz: Mit $A'_{ij} = 1/\text{outdegree}(i)$ falls $(i,j) \in E$, 0 sonst, gilt:

$$\vec{r} = \frac{\vec{\varepsilon}}{n} + (1-\varepsilon) A' \vec{r} \quad \Leftrightarrow \quad \vec{r} = \left(\frac{\vec{\varepsilon}}{n} \vec{1}^T + (1-\varepsilon) A' \right) \vec{r}$$

d.h. r ist Eigenvektor einer modifizierten Transitionsmatrix

Iterative Berechnung von r(q):
• Initialisierung mit r(q) := 1/n
• Verbesserung durch Auswerten der rekursiven Definitionsgleichung
konvergiert typischerweise mit ca. 100 Iterationen

4.2 Exkurs: Markov-Ketten

Ein **stochastischer Prozeß** ist eine Familie von
Zufallsvariablen $\{X(t) \mid t \in T\}$.
T heißt Parameterraum, und der Definitionsbereich M der X(t)
heißt Zustandsraum. T und M können diskret oder kontinuierlich sein.

Ein stochastischer Prozeß heißt **Markov-Prozeß**, wenn
für beliebige $t_1, ..., t_{n+1}$ aus dem Parameterraum und
für beliebige $x_1, ..., x_{n+1}$ aus dem Zustandsraum gilt:

$$P[\, X(t_{n+1}) = x_{n+1} \mid X(t_1) = x_1 \wedge X(t_2) = x_2 \wedge ... \wedge X(t_n) = x_n \,]$$
$$= P[\, X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n \,]$$

Ein Markov-Prozeß mit diskretem Zustandsraum heißt **Markov-Kette**.
O.B.d.A. werden die natürlichen Zahlen als Zustandsraum gewählt.
Notation für Markov-Ketten mit diskretem Parameterraum:
X_n statt $X(t_n)$ mit n = 0, 1, 2, ...

Exkurs: Eigenschaften von Markov-Ketten mit diskretem Parameterraum (1)

Die Markov-Kette Xn mit diskretem Parameterraum heißt

homogen, wenn die Übergangswahrscheinlichkeiten
$p_{ij} := P[X_{n+1} = j \mid X_n = i]$ unabhängig von n sind

irreduzibel, wenn jeder Zustand von jedem Zustand mit
positiver Wahrscheinlichkeit erreichbar ist:

$$\sum_{n=1}^{\infty} P[\, X_n = j \mid X_0 = i \,] > 0 \quad \text{für all i, j}$$

aperiodisch, wenn alle Zustände i die Periode 1 haben, wobei
die Periode von i der ggT aller Werte n ist, für die gilt:

$$P[\, X_n = i \wedge X_k \ne i \text{ für } k = 1,...,n-1 \mid X_0 = i \,] > 0$$

Exkurs: Eigenschaften von Markov-Ketten mit diskretem Parameterraum (2)

Die Markov-Kette Xn mit diskretem Parameterraum heißt

positiv rekurrent, wenn für jeden Zustand i die Rückkehr-
wahrscheinlichkeit gleich 1 ist und mittlere Rekurrenzzeit endlich:

$$\sum_{n=1}^{\infty} P[\, X_n = i \wedge X_k \ne i \text{ für } k = 1,...,n-1 \mid X_0 = i \,] = 1$$

$$\sum_{n=1}^{\infty} n\, P[\, X_n = i \wedge X_k \ne i \text{ für } k = 1,...,n-1 \mid X_0 = i \,] < \infty$$

ergodisch, wenn sie homogen, irreduzibel, aperiodisch und
positiv rekurrent ist.

Resultate über Markov-Ketten mit diskretem Parameterraum (1)

Für die **n-Schritt-Transitionswahrscheinlichkeiten**

$p_{ij}^{(n)} := P[\,X_n = j \,|\, X_0 = i\,]$ gilt:

$$p_{ij}^{(n)} = \sum_k p_{ik}^{(n-1)} p_{kj} \quad \text{mit} \quad p_{ij}^{(1)} := p_{ik}$$

$$= \sum_k p_{ik}^{(n-l)} p_{kj}^{(l)} \quad \textit{für}\ 1 \le l \le n-1$$

in Matrix-Notation: $P^{(n)} = P^n$

Für die **Zustandswahrscheinlichkeiten nach n Schritten**

$\pi_j^{(n)} := P[\,X_n = j\,]$ gilt:

$\pi_j^{(n)} = \sum_i \pi_i^{(0)} p_{ij}^{(n)}$ mit Anfangswahrscheinlichkeiten $\pi_i^{(0)}$

in Matrix-Notation: $\Pi^{(n)} = \Pi^{(0)} P^{(n)}$ *(Chapman-Kolmogorov-Gleichung)*

Resultate über Markov-Ketten mit diskretem Parameterraum (2)

Jede homogene, irreduzible, aperiodische Markov-Kette mit endlich vielen Zuständen ist positiv rekurrent und ergodisch.

Für jede ergodische Markov-Kette existieren **stationäre Zustandswahrscheinlichkeiten** $\pi_j := \lim_{n\to\infty} \pi_j^{(n)}$

Diese sind unabhängig von $\Pi^{(0)}$

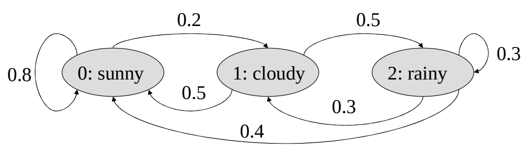
und durch das folgende lineare Gleichungssystem bestimmt:

$$\pi_j = \sum_i \pi_i p_{ij} \quad \textit{für alle } j \qquad \textit{(Gleichgewichts-gleichungen)}$$

$$\sum_j \pi_j = 1$$

in Matrix-Notation (mit $1\times n$-Vektor Π): $\Pi = \Pi P$, $\Pi \vec{1} = 1$

Beispiel: Markov-Kette



$\pi0 = 0.8\ \pi0 + 0.5\ \pi1 + 0.4\ \pi2$
$\pi1 = 0.2\ \pi0 + 0.3\ \pi2$
$\pi2 = 0.5\ \pi1 + 0.3\ \pi2$
$\pi0 + \pi1 + \pi2 = 1$

$\Rightarrow \pi0 = 330/474 \approx 0.696$
$\pi1 = 84/474 \approx 0.177$
$\pi2 = 10/79 \approx 0.126$

Page-Ranks im Kontext von Markov-Ketten

Modellierung des **Random Walks** eines Web-Surfers durch

- Verfolgen von Hyperlinks mit gleichverteilten Wahrscheinlichkeiten
- „Random Jumps" mit Wahrscheinlichkeit ε

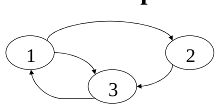
→ ergodische Markov-Kette

Der Page-Rank einer URL ist die stationäre Besuchswahrscheinlichkeit der URL für diese Markov-Kette.

Verallgemeinerungen sind denkbar (z.B. Random Walk mit Back-Button u.ä.)

Kritik am Page-Rank-Verfahren:
Page-Rank ist query-unabhängig und orthogonal zur Relevanz

Beispiel: Page-Rank-Berechnung



$\varepsilon = 0.2$

$$P = \begin{pmatrix} 0.0 & 0.5 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.9 & 0.1 & 0.0 \end{pmatrix}$$

$$\Pi^{(0)} \approx \begin{pmatrix} 0.333 \\ 0.333 \\ 0.333 \end{pmatrix}^T \Rightarrow \Pi^{(1)} \approx \begin{pmatrix} 0.333 \\ 0.200 \\ 0.466 \end{pmatrix}^T \Rightarrow \Pi^{(2)} \approx \begin{pmatrix} 0.439 \\ 0.212 \\ 0.346 \end{pmatrix}^T \Rightarrow \Pi^{(3)} \approx \begin{pmatrix} 0.332 \\ 0.253 \\ 0.401 \end{pmatrix}^T$$

$$\Rightarrow \Pi^{(4)} \approx \begin{pmatrix} 0.385 \\ 0.176 \\ 0.527 \end{pmatrix}^T \Rightarrow \Pi^{(5)} \approx \begin{pmatrix} 0.491 \\ 0.244 \\ 0.350 \end{pmatrix}^T$$

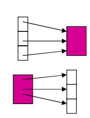
$\pi1 = 0.1\ \pi2 + 0.9\ \pi3$
$\pi2 = 0.5\ \pi1 + 0.1\ \pi3$
$\pi3 = 0.5\ \pi1 + 0.9\ \pi2$
$\pi1 + \pi2 + \pi3 = 1$

$\Rightarrow \pi1 \approx 0.3776,\ \pi2 \approx 0.2282,\ \pi3 \approx 0.3942$

4.3 HITS-Algorithmus: Hyperlink-Induced Topic Search (1)

Idee:

Bestimme
- gute Inhaltsquellen: **Authorities** (großer indegree)
- gute Linkquellen: **Hubs** (großer outdegree)



Finde
- bessere Authorities mit guten Hubs als Vorgängern
- bessere Hubs mit guten Authorities als Nachfolgern

Für Web-Graph G=(V,E) definiere für Knoten p, q $\in$V

Authority-Score $x_q = \sum_{(p,q)\in E} y_p$ und

Hub-Score $y_p = \sum_{(p,q)\in E} x_q$

HITS-Algorithmus (2)

Authority- und Hub-Scores in Matrix-Notation:

$\vec{x} = A^T \vec{y}$ $\vec{y} = A \vec{x}$

Iteration mit Adjazenz-Matrix A:

$\vec{x} := A^T \vec{y} := A^T A \vec{x}$ $\vec{y} := A \vec{x} := A A^T \vec{y}$

x und y sind also Eigenvektoren von $A^T A$ bzw. AA^T.

Intuitive Interpretation:

$M^{(auth)} := A^T A$ ist die Cocitation-Matrix: $M^{(auth)}_{ij}$ ist die Anzahl der Knoten, die auf i und j zeigen

$M^{(hub)} := AA^T$ ist die Bibliographic-Coupling-Matrix: $M^{(hub)}_{ij}$ ist die Anzahl der Knoten, auf die i und j zeigen

Implementierung des HITS-Algorithmus

1) Bestimme hinreichend viele (z.B. 50-200) „Wurzelseiten" per Relevanz-Ranking (z.B. mittels tf*idf-Ranking)
2) Füge alle Nachfolger von Wurzelseiten hinzu
3) Füge für jede Wurzelseite max. d Vorgänger hinzu
4) Bestimme durch Iteration die Authority- und Hub-Scores dieser „Basismenge" (von 1000-5000 Seiten) mit Initialisierung $x_q := y_p := 1 / |Basismenge|$ und Normalisierung nach jedem Schritt
   → konvergiert gegen die Eigenvektoren mit dem betragsgrößten Eigenwert (falls dieser Multiplizität 1 hat)
5) Gib Seiten nach absteigend sortierten Authority-Scores aus (z.B. die 10 größten Komponenten von x)

Kritik am HITS-Algorithmus:
Relevanz-Ranking innerhalb der Wurzelmenge bleibt unberücksichtigt

Verbesserter HITS-Algorithmus

Potentielle Schwachstellen des HITS-Algorithmus:
- irritierende Links (automatisch generierte Links, Spam, etc.)
- Themendrift (z.B. von „Jaguar car" zu „car" generell)

Verbesserung:
- Einführung von Kantengewichten:
  0 für Links auf demselben Host,
  1/k bei k Links von k URLs desselben Host zu 1 URL (xweight)
  1/m bei m Links von 1 URL zu m URLs desselben Host (yweight)
- Berücksichtigung von thematischen Relevanzgewichten (z.B. tf*idf)

→ Iterative Berechnung von

Authority-Score $x_q = \sum_{(p,q)\in E} y_p * topic\,score(p) * xweight(p,q)$

Hub-Score $y_p = \sum_{(p,q)\in E} x_q * topic\,score(q) * yweight(p,q)$

Bestimmung verwandter URLs

Cocitation-Algorithmus:
- Bestimme bis zu B Vorgänger der gegebenen URL u
- Für jeden Vorgänger p bestimme bis zu BF Nachfolger ≠ u
- Bestimme unter allen Geschwistern s von u diejenigen mit der größten Anzahl von Vorgängern, die sowohl auf s als auch auf u zeigen (Cocitation-Grad)

Companion-Algorithmus:
- Bestimme geeignete Basismenge um die gegebene URL u herum
- Wende den HITS-Algorithmus auf diese Basismenge an

Companion-Algorithmus zur Bestimmung verwandter URLs

1) Bestimmung der Basismenge: u sowie
   - bis zu B Vorgänger von u und für jeden Vorgänger p bis zu BF Nachfolger ≠ u sowie
   - bis zu F Nachfolger von u und für jeden Nachfolger c bis zu FB Vorgänger ≠ u

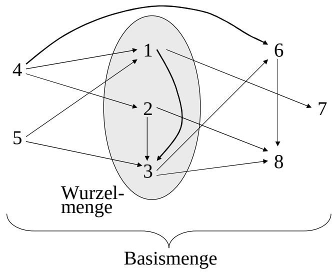
   mit Elimination von Stop-URLs (wie z.B. www.yahoo.com)
2) Duplikateliminierung:
   Verschmelze Knoten, die jeweils mehr als 10 Nachfolger haben und mehr als 95 % ihrer Nachfolger gemeinsam haben
3) Bestimme Authority-Scores mit dem verbesserten HITS-Algorithmus

HITS-Algorithmus zur „Community Detection"

Wurzelmenge kann mehrere Themen bzw. „Communities" beinhalten, z.B. bei Queries „jaguar", „Java" oder „randomized algorithm"

Ansatz:
- Bestimmung der k betragsgrößten Eigenwerte von $A^T A$ und der zugehörigen Eigenvektoren x
- In jedem dieser k Eigenvektoren x reflektieren die größten Authority-Scores eine eng vernetzte „Community"

Beispiel: HITS-Algorithmus



4.4 Themenspezifisches Page-Rank-Verfahren

für verschiedene thematische Klassen (Sport, Musik, Jazz, etc.), wobei jede Klasse c_k durch eine Menge T_k einschlägiger Autoritäten charakterisiert ist (z.B. aus Verzeichnissen von yahoo.com, dmoz.org)

Kernidee :
Ändere den Random Walk durch themenspezifische Random-Jump-Wahrscheinlichkeiten für Seiten aus T_k:

$\vec{r}_k = \varepsilon\, \vec{p}_k + (1-\varepsilon) A' \vec{r}_k$ mit $A'_{ij} = 1/\text{outdegree}(i)$ für $(i,j) \in E$, 0 sonst

mit $(p_k)_j = 1/|T_k|$ für $j \in T_k$, 0 sonst (anstatt $p_j = 1/n$)

Verfahren:
1) Berechne für jede Klasse c_k thematische Page-Rank-Vektoren r_k
2) Klassifiziere Query q (inkl. Kontext) bzgl. Klasse c_k
$\rightarrow$ Wahrscheinlichkeit $w_k := P[c_k \mid q]$
3) Der Autoritäts-Score von Seite d ist $\sum_k w_k\, r_k(d)$

Experimentelle Evaluation: Qualitätsmaße

basierend auf Stanford WebBase (120 Mio. Seiten, Jan. 2001)
enthält ca. 300 000 von 3 Mio. Seiten aus dmoz.org
aus 16 Themen der obersten Stufe von dmoz.org;
Link-Graph mit 80 Mio. Knoten und der Größe 4 GB
auf 1.5 GHz Dual Athlon mit 2.5 GB Speicher und 500 GB RAID
25 Iterationen für alle 16+1 PR-Vektoren brauchen 20 Stunden
Random-Jump-W. ε gesetzt auf 0.25 (themenspezifisch?)
35 Test-Queries, z.B.: classical guitar, lyme disease, sushi, etc.

Qualitätsmaße: Betrachte Top-k zweier Ranglisten τ1 und τ2 (k=20)

- ***Überlappung OSim*** (τ1,τ2) = | top(k,τ1) ∩ top(k,τ2) | / k
- ***Kendall's τ KSim*** (τ1,τ2) =

$$\frac{|\{(u,v) \mid u,v \in U, u \neq v, \text{und } \tau 1, \tau 2 \text{ haben dieselbe Ordnung von } u,v\}|}{|U| \cdot (|U|-1)}$$

mit U = top(k,τ1) ∪ top(k,τ2)

Experimentelle Resultate (1)

- Ranglistenähnlichkeit zwischen den ähnlichsten PR Vektoren:

	OSim	KSim
(Games, Sports)	0.18	0.13
(No Bias, Regional)	0.18	0.12
(Kids&Teens, Society)	0.18	0.11
(Health, Home)	0.17	0.12
(Health, Kids&Teens)	0.17	0.11

- Präzision für Top-10 (# relevante Dok. / 10) von 5 Benutzern:

	Standard	Themenspezifisch
alcoholism	0.12	0.7
bicycling	0.36	0.78
death valley	0.28	0.5
HIV	0.58	0.41
Shakespeare	0.29	0.33
micro average	0.276	0.512

Experimentelle Resultate (2)

- Top-3 für Query "bicycling"
(klassifiziert auf sports mit W. 0.52, regional mit 0.13, health mit 0.07)

Standard	Recreation	Sports
1 www.RailRiders.com	www.gorp.com	www.multisports.com
2 www.waypoint.org	www.GrownupCamps.com	www.BikeRacing.com
3 www.gorp.com	www.outdoor-pursuits.com	www.CycleCanada.com

- Top-5 für Query-Kontext "blues" (Benutzer wählt Seite aus)
(klassifiziert auf arts mit W. 0.52, shopping mit 0.12, news mit 0.08)

No Bias	Arts	Health
1 news.tucows.com	www.britannia.com	www.baltimorepsych.com
2 www.emusic.com	www.bandhunt.com	www.ncpamd.com/seasonal
3 www.johnholleman.com	www.artistinformation.com	www.ncpamd.com/Women's_
4 www.majorleaguebaseball	www.billboard.com	www.wingofmadness.com
5 www.mp3.com	www.soul-patrol.com	www.countrynurse.com

Persönliche Page-Rank-Werte

Ziel: Effiziente Berechnung und Speicherung auf einzelne Benutzerpräferenzen zugeschnittener Page-Rank-Vektoren

Page-Rank-Gleichung: $\vec{r}_k = \varepsilon\, \vec{p}_k + (1-\varepsilon) A' \vec{r}_k$

Theorem:
Seien u1 und u2 persönliche Präferenzvektoren (für Random-Jump-Ziele) und seien r1 und r2 die zugehörigen Page-Rank-Vektoren. Dann gilt für alle α1, α2 ≥ 0 mit α1 + α2 = 1:
α1 r1 + α2 r2 = (1-ε) A‘ (α1 r1 + α2 r2) + ε (α1 u1 + α2 u2)

Korollar:
Für einen Präferenzvektor u mit m von 0 verschiedenen Komponenten und Basisvektoren e_p mit $(e_p)_i = 1$ für i=p, 0 für i≠p gilt:

$u = \sum_{p=1}^{m} \alpha_p \cdot e_p$ mit Konstanten $\alpha_1 \ldots \alpha_m$

und $r = \sum_{p=1}^{m} \alpha_p \cdot r_p$ für den persönlichen Page-Rank-Vektor r