

8 Web-Retrieval mit Autoritäts-Ranking

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)

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-1

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

Def.: $r(q) = \varepsilon / n + (1 - \varepsilon) \sum_{(p,q) \in G} r(p) / \text{outdegree}(p)$ mit $0 < \varepsilon \leq 0.2$

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

$$\vec{r} = \frac{\vec{e}}{n} + (1 - \varepsilon) A^* \vec{r} \Leftrightarrow \frac{1}{1 - \varepsilon} \vec{r} = \left(\frac{\varepsilon}{(1 - \varepsilon)n} \vec{1}^T + A^* \right) \vec{r}$$

d.h. \vec{r} ist Eigenvektor einer modifizierten Transitionsmatrix

Iterative Berechnung von $r(q)$ (auf großem Web-Crawl):

- Initialisierung mit $r(q) := 1/n$
- Verbesserung durch Auswerten der rekursiven Definitionsgleichung konvergiert typischerweise mit ca. 100 Iterationen

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-2

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, \dots, t_{n+1} aus dem Parameterraum und für beliebige x_1, \dots, x_{n+1} aus dem Zustandsraum gilt:

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

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 $n = 0, 1, 2, \dots$

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-3

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

Die Markov-Kette X_n 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:

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

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

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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-4

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

Die Markov-Kette X_n mit diskretem Parameterraum heißt

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

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

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

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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-5

Resultate über Markov-Ketten mit diskretem Parameterraum (1)

Für die **n-Schritt-Transitionswahrscheinlichkeiten**

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

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

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

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

Für die **Zustandswahrscheinlichkeiten nach n Schritten**

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

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

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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-6

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** $\pi_j := \lim_{n \rightarrow \infty} \pi_j^{(n)}$
Diese sind unabhängig von $\Pi^{(0)}$

und durch das folgende lineare Gleichungssystem bestimmt:

$$\pi_j = \sum_i \pi_i p_{ij} \quad \text{für alle } j \quad (\text{Gleichgewichtsgleichungen})$$

$$\sum_j \pi_j = 1$$

in Matrix-Notation: $\Pi^T = \Pi^T P$

$$\Pi^T \bar{1} = 1$$

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-7

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

12. Januar 2001

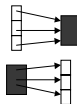
Stammvorlesung „Information Retrieval“

8-8

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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-9

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:

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

\vec{x} und \vec{y} sind also Eigenvektoren von $A^T A$ bzw. $A A^T$.

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

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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-10

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 $x_q := y_p := 1 / |\text{Basismenge}|$
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 \vec{x})

Kritik am HITS-Algorithmus:

Relevanz-Ranking innerhalb der Wurzelmenge bleibt unberücksichtigt

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-11

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

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

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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-12

Bestimmung verwandter URLs

Cocitation-Algorithmus:

- Bestimme bis zu B Vorgänger der gegebenen URL u
- Für jeden Vorgänger p bestimme bis zu BF Nachfolger $\neq u$
- 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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-13

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 $\neq u$ sowie
 - bis zu F Nachfolger von u und für jeden Nachfolger c bis zu FB Vorgänger $\neq u$ mit Elimination von Stop-URLs (wie z.B. www.yahoo.com)
- 2) Duplikateliminiierung:
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

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-14

HITS-Algorithmus zur „Community Detection“

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

Ansatz:

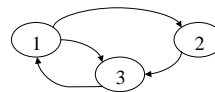
- Bestimmung der k betragsgrößten Eigenwerte von $A^T A$ und der zugehörigen Eigenvektoren x
- In jedem dieser k Eigenvektoren x reflektieren die größten Authority-Scores eine eng vernetzte „Community“

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-15

Beispiel: Page-Rank-Berechnung



$$\epsilon = 0.2 \quad P = \begin{pmatrix} 0.0 & 0.5 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.9 & 0.1 & 0.0 \end{pmatrix}$$

$$\begin{aligned} \Pi^{(0)} &\approx \begin{pmatrix} 0.333 \\ 0.333 \\ 0.333 \end{pmatrix} \Rightarrow \Pi^{(1)} \approx \begin{pmatrix} 0.333 \\ 0.200 \\ 0.466 \end{pmatrix} \Rightarrow \Pi^{(2)} \approx \begin{pmatrix} 0.439 \\ 0.212 \\ 0.346 \end{pmatrix} \Rightarrow \Pi^{(3)} \approx \begin{pmatrix} 0.332 \\ 0.253 \\ 0.401 \end{pmatrix} \\ &\Rightarrow \Pi^{(4)} \approx \begin{pmatrix} 0.385 \\ 0.176 \\ 0.527 \end{pmatrix} \Rightarrow \Pi^{(5)} \approx \begin{pmatrix} 0.491 \\ 0.244 \\ 0.350 \end{pmatrix} \\ \pi_1 &= 0.1 \pi_2 + 0.9 \pi_3 \\ \pi_2 &= 0.5 \pi_1 + 0.1 \pi_3 \\ \pi_3 &= 0.5 \pi_1 + 0.9 \pi_3 \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned} \Rightarrow \pi_1 \approx 0.433, \pi_2 \approx 0.094, \pi_3 \approx 0.471$$

12. Januar 2001

Stammvorlesung „Information Retrieval“

8-16