

# Google's Page Rank

Sommerakademie Salem 2008 – AG 4

Martin Linnartz, Benedikt Rehle und Sven Pirner

Studienstiftung des deutschen Volkes

29. August 2008

- **Einführung in den Algorithmus (Martin)**
- Theorie zu Markov-Chains (Benedikt)
- Implementierung und Optimierungsansätze (Sven)

## Funktionsweise von Google

- Grundgedanke: Eine Webseite ist wichtig, wenn andere wichtige Seiten auf diese verlinken (vgl. Zitate in akademischen Papers)
- Etwa monatlich werden sämtliche Seiten des Internet von Google-Crawlern durchforstet und auf Linkstrukturen hin untersucht
- Es entsteht der Google-Index mit über 8 Milliarden Einträgen
- Der zugeordnete "Rang" entspricht der Wichtigkeit einer Seite

## Wie wird der Rang einer Webseite ermittelt?

Der grundlegende Algorithmus wurde von Brin und Page entwickelt und ist seit 2001 in den USA patentiert:

$$r(P_i) := \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}$$

Mit  $B_{P_i}$  als Menge aller Seiten  $P_j$ , die auf  $P_i$  verlinken, sowie  $|P_j|$  als Anzahl aller ausgehenden Links von  $P_j$ .

Aber: Da der PageRank  $r(P_j)$  zunächst unbekannt ist, kann  $r(P_i)$  nur iterativ bestimmt werden!

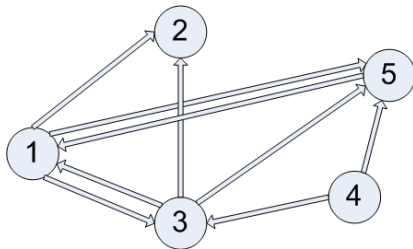
Daraus ergibt sich die folgende **Iterations-Vorschrift**:

$$r_{k+1}(P_i) := \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

Annahme: Alle Seiten haben zu Beginn die gleiche Gewichtung  $r_0(P_i) = \frac{1}{n} \forall P_i$  mit  $n =$  Anzahl der Webseiten im Google-Index.

# Hyperlinkmatrix H

Betrachte folgenden Untergraphen des Internet:



Die zugehörige  $n \times n$ -Hyperlinkmatrix  $H$  mit  $H_{ij} = \frac{1}{|P_i|}$ , wenn ein Outlink von  $P_i$  auf  $P_j$  vorhanden ist, sonst  $H_{ij} = 0$ :

$$H = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Der **PageRank-Vektor** mit  $n$  Einträgen wird definiert als Zeilenvektor  $\pi$  und für den Iterationsschritt  $k = 0$  initialisiert mit  $\pi^{(0)T} = \frac{1}{n} \cdot e^T$ , wobei  $e = \text{ones}(n)$ . Zu Beginn der Berechnung haben also alle Webseiten den gleichen Ranking Score  $\frac{1}{n}$ .

Mithilfe der Hyperlinkmatrix  $H$  wird die Potenzmethode angewandt:

## Einfache Potenzmethode

$$\pi^{(k+1)T} = \pi^{(k)T} \cdot H$$

## Schwierigkeiten bei diesem Verfahren

- Rank Sinks/ Senken: Sogenannte Dangling Nodes (Webseiten ohne Outlinks) erzeugen in der H-Matrix Nullzeilen und akkumulieren im Laufe der Iteration Ranking Score.
- Cluster: Auch ein Untergraph (eines Netzwerkes), der nur intern verlinkt, aber keine Outlinks zur Umgebung hat, agiert als Senke.
- Cycles: Zwei Webseiten, die sich ausschließlich gegenseitig verlinken, verhindern Konvergenz des PageRank-Vektors.

## Markov-Chain-Theory

Die H-Matrix wird so angepasst, dass sie zu einer Markov-Matrix und damit *stochastisch*, *irreduzibel* und *aperiodisch* wird. Wird die Potenzmethode mit einer Markov-Matrix durchgeführt, so ist sichergestellt, dass der PageRank-Vektor konvergiert und eindeutig bestimmt ist (dazu mehr im zweiten Teil dieses Vortrags)!

# Anpassung der H-Matrix

H wird stochastisch, indem alle Nullzeilen durch  $\frac{1}{n}e^T$  ersetzt werden:

$$S = H + a \left( \frac{1}{n} e^T \right)$$

mit  $a_i = 1$ , wenn  $P_i$  dangling node ist, und  $a_i = 0$  sonst.

S wird dann über die Teleportationsmatrix  $E = \frac{1}{n}ee^T$  primitiv angepasst, es entsteht die **Google-Matrix**:

$$G = \alpha S + (1 - \alpha)E$$

mit  $0 < \alpha < 1$  Wahrscheinlichkeitsfaktor.  $\alpha$  gibt an, zu welchen prozentualen Anteilen ein **random surfer** der Hyperlinkstruktur des Webs folgen (Matrix S) oder per zufällig eingegebener URL auf eine andere Seite wechseln würde (Matrix E).

Mit der neu erzeugten Google-Matrix  $G$  (anstelle der einfacheren Hyperlinkmatrix  $H$ ) führt die Potenzmethode nun zum gewünschten Ergebnis, d.h. zu einem eindeutigen, konvergierenden PageRank-Vektor.

$$\pi^{(k+1)T} = \pi^{(k)T} \cdot G$$

Da  $G$  allerdings vollbesetzt ist, wird für die numerische Berechnung doch wieder die schwachbesetzte Hyperlinkmatrix  $H$  herangezogen (geringerer Rechenaufwand):

$$\pi^{(k+1)T} = \alpha \pi^{(k)T} H + \left( \alpha \pi^{(k)T} \mathbf{a} + 1 - \alpha \right) \frac{1}{n} \mathbf{e}^T$$

**Quelle:** Langville A.N., Meyer C. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, Princeton, NJ, 2006.