

Googles PageRank-Algorithmus

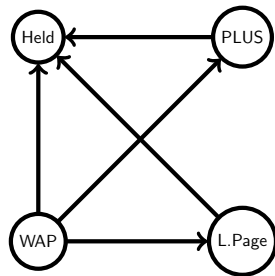
Niklas Leon Draxl, Mateja Jankovic

28. Februar 2026

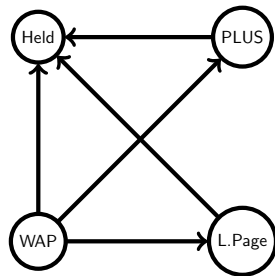
- Wie berechnet man die Wichtigkeit einer Seite im Web?

- Wie berechnet man die Wichtigkeit einer Seite im Web?
- Wie lässt sich dieses Problem mathematisch auffassen?

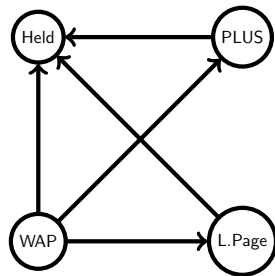
- Links werden als gerichtete Kanten dargestellt



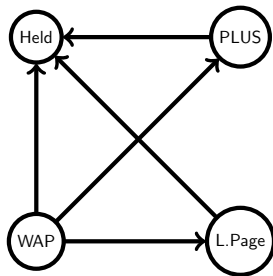
- Links werden als gerichtete Kanten dargestellt
- Jede Seite als Knoten



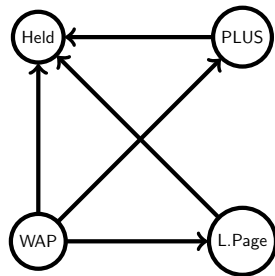
- Links werden als gerichtete Kanten dargestellt
- Jede Seite als Knoten
- jede Seite P_i erhält einen initialen Seitenrang $r(P_i)$



- Links werden als gerichtete Kanten dargestellt
- Jede Seite als Knoten
- jede Seite P_i erhält einen initialen Seitenrang $r(P_i)$
- $r(P_i) := \frac{1}{n}$



- Links werden als gerichtete Kanten dargestellt
- Jede Seite als Knoten
- jede Seite P_i erhält einen initialen Seitenrang $r(P_i)$
- $r(P_i) := \frac{1}{n}$
- $n := \text{Anzahl der Seiten}$



Berechnung des Seitenrangs P_i

$$r(P_i) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$$

- $r(P_i)$: Rang der Seite P_i

Berechnung des Seitenrangs P_i

$$r(P_i) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$$

- $r(P_i)$: Rang der Seite P_i
- $B(P_i)$: Menge aller Seiten, die einen Link auf P_i haben

Berechnung des Seitenrangs P_i

$$r(P_i) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$$

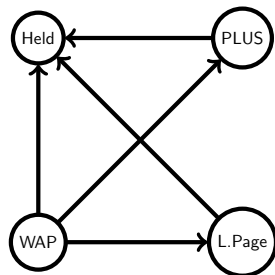
- $r(P_i)$: Rang der Seite P_i
- $B(P_i)$: Menge aller Seiten, die einen Link auf P_i haben
- P_j : eine konkrete Seite, die auf P_i verweist

Berechnung des Seitenrangs P_i

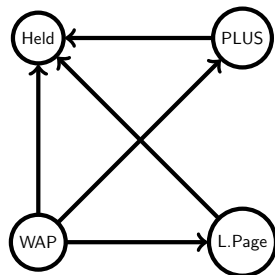
$$r(P_i) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$$

- $r(P_i)$: Rang der Seite P_i
- $B(P_i)$: Menge aller Seiten, die einen Link auf P_i haben
- P_j : eine konkrete Seite, die auf P_i verweist
- $o(P_j)$: Anzahl der ausgehenden Links von P_j

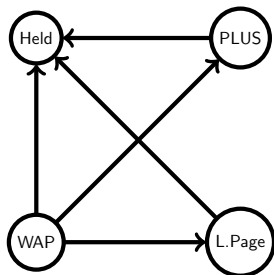
- Initialer Seitenrang = $\frac{1}{4}$



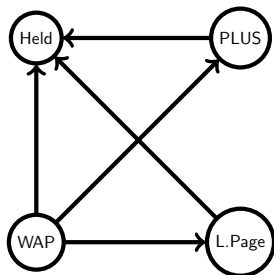
- Initialer Seitenrang = $\frac{1}{4}$
- $r(\text{Held}) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$



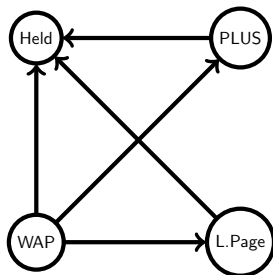
- Initialer Seitenrang = $\frac{1}{4}$
- $r(\text{Held}) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$
- $r(\text{Held}) = \frac{r(\text{PLUS})}{1} + \frac{r(\text{WAP})}{3} + \frac{r(\text{L.Page})}{1}$



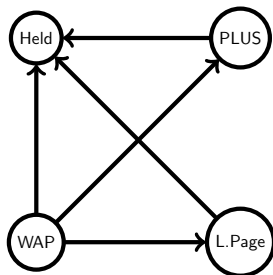
- Initialer Seitenrang $= \frac{1}{4}$
- $r(\text{Held}) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$
- $r(\text{Held}) = \frac{r(\text{PLUS})}{1} + \frac{r(\text{WAP})}{3} + \frac{r(\text{L.Page})}{1}$
- $r(\text{Held}) = \frac{0.25}{1} + \frac{0.25}{3} + \frac{0.25}{1}$



- Initialer Seitenrang = $\frac{1}{4}$
- $r(\text{Held}) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$
- $r(\text{Held}) = \frac{r(\text{PLUS})}{1} + \frac{r(\text{WAP})}{3} + \frac{r(\text{L.Page})}{1}$
- $r(\text{Held}) = \frac{0.25}{1} + \frac{0.25}{3} + \frac{0.25}{1}$
- $r(\text{Held}) \approx 0.583$



- Initialer Seitenrang = $\frac{1}{4}$
- $r(\text{Held}) = \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)}$
- $r(\text{Held}) = \frac{r(\text{PLUS})}{1} + \frac{r(\text{WAP})}{3} + \frac{r(\text{L.Page})}{1}$
- $r(\text{Held}) = \frac{0.25}{1} + \frac{0.25}{3} + \frac{0.25}{1}$
- $r(\text{Held}) \approx 0.583$
- Wir wiederholen dieses Verfahren für alle Knoten und aktualisieren die Seitenrang-Werte, bis sie konvergieren



Problem des Summierungsverfahrens

- Summierungsverfahren ist sehr aufwendig bzw. unübersichtlich

- Summierungsverfahren ist sehr aufwendig bzw. unübersichtlich
- Besonders bei vielen Knoten bzw. Websites

- Summierungsverfahren ist sehr aufwendig bzw. unübersichtlich
- Besonders bei vielen Knoten bzw. Websites
- Lösung: Wir speichern unsere Pageranks in einem Vektor π^T der Länge n

- Summierungsverfahren ist sehr aufwendig bzw. unübersichtlich
- Besonders bei vielen Knoten bzw. Websites
- Lösung: Wir speichern unsere Pageranks in einem Vektor π^T der Länge n
- Die Struktur des Webs in einer $n \times n$ Matrix H

- Summierungsverfahren ist sehr aufwendig bzw. unübersichtlich
- Besonders bei vielen Knoten bzw. Websites
- Lösung: Wir speichern unsere Pageranks in einem Vektor π^T der Länge n
- Die Struktur des Webs in einer $n \times n$ Matrix H
- Dadurch können wir alle Seitenrang-Werte gleichzeitig per Matrix-Vektor-Multiplikation aktualisieren.

Konstruktion der Matrix H

- Wir erstellen die Hyperlink-Matrix H :
 - $H_{i,j} = 1 \iff P_i$ mit ausgehenden Link auf P_j
 - sonst $H_{i,j} = 0$

Konstruktion der Matrix H

- Wir erstellen die Hyperlink-Matrix H :
 - $H_{i,j} = 1 \iff P_i$ mit ausgehenden Link auf P_j
 - sonst $H_{i,j} = 0$
- Anschließend gewichten wir zeilenweise:

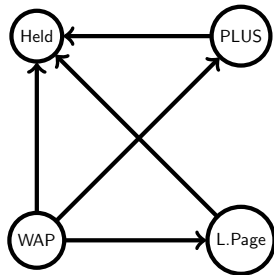
$$H_{i,j} = \frac{H_{i,j}}{o(P_i)}.$$

- P_i beschreibt eine konkrete Seite
- $o(P_i)$ beschreibt die Anzahl der ausgehenden Links der Seite P_i
- Somit speichern wir in $H_{i,j}$ die Wahrscheinlichkeit, dass wir durch zufälliges Klicken von der Seite P_i auf die Seite P_j springen

Beispiel Matrix erstellen

$$H_{i,j} \leftarrow \frac{H_{i,j}}{o(P_i)}.$$

	Held	WAP	L.Page	PLUS
Held	0	0	0	0
WAP	0.33	0	0.33	0.33
L.Page	1	0	0	0
PLUS	1	0	0	0



- wir wissen mit welcher Wahrscheinlichkeit wir von P_i nach P_j kommen

- wir wissen mit welcher Wahrscheinlichkeit wir von P_i nach P_j kommen
- jetzt benötigen wir die Wahrscheinlichkeit auf welcher Website wir uns jetzt gerade befinden

- wir wissen mit welcher Wahrscheinlichkeit wir von P_i nach P_j kommen
- jetzt benötigen wir die Wahrscheinlichkeit auf welcher Website wir uns jetzt gerade befinden
- dies gibt der Vektor π^T an.

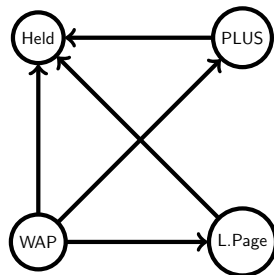
- $\pi^T := \begin{pmatrix} r(P_1) \\ r(P_2) \\ \dots \\ r(P_n) \end{pmatrix}$

- wir wissen mit welcher Wahrscheinlichkeit wir von P_i nach P_j kommen
- jetzt benötigen wir die Wahrscheinlichkeit auf welcher Website wir uns jetzt gerade befinden
- dies gibt der Vektor π^T an.

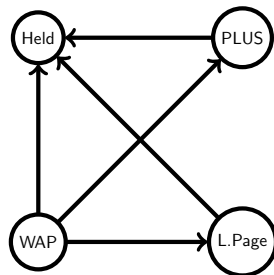
- $\pi^T := \begin{pmatrix} r(P_1) \\ r(P_2) \\ \dots \\ r(P_n) \end{pmatrix}$

- zu beginn entspricht der Wert von $r(P_i) = \frac{1}{n}$

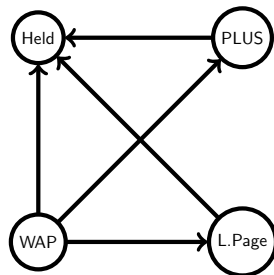
- Initialer Seitenrang = $\frac{1}{4}$



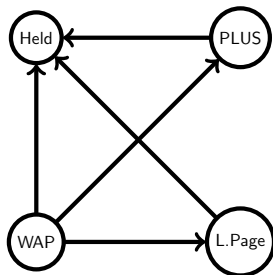
- Initialer Seitenrang = $\frac{1}{4}$
- somit ergibt sich für $\pi_0^T = \begin{pmatrix} r(P_1) \\ r(P_2) \\ \dots \\ r(P_n) \end{pmatrix}$



- Initialer Seitenrang = $\frac{1}{4}$
- somit ergibt sich für $\pi_0^T = \begin{pmatrix} r(P_1) \\ r(P_2) \\ \dots \\ r(P_n) \end{pmatrix}$
- $\pi_0^T = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$



- Initialer Seitenrang = $\frac{1}{4}$
- somit ergibt sich für $\pi_0^T = \begin{pmatrix} r(P_1) \\ r(P_2) \\ \dots \\ r(P_n) \end{pmatrix}$
- $\pi_0^T = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$
- Der Seitenrang der 4 Seiten ist am Anfang gleich groß



- wir multiplizieren den Vektor π_k^T mit unser Matrix H und erhalten somit den Vektor π_{k+1}^T

- wir multiplizieren den Vektor π_k^T mit unser Matrix H und erhalten somit den Vektor π_{k+1}^T
- Dieser gibt die Seitenränge nach $k + 1$ zufälligen Klick-Schritten an

- wir multiplizieren den Vektor π_k^T mit unser Matrix H und erhalten somit den Vektor π_{k+1}^T
- Dieser gibt die Seitenränge nach $k + 1$ zufälligen Klick-Schritten an
- Somit ergibt sich

- wir multiplizieren den Vektor π_k^T mit unser Matrix H und erhalten somit den Vektor π_{k+1}^T
- Dieser gibt die Seitenränge nach $k + 1$ zufälligen Klick-Schritten an
- Somit ergibt sich

Iterationsverfahren

$$\pi_{k+1}^T = H \cdot \pi_k^T$$

Iteration Beispiel

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.5825 \\ 0 \\ 0.08 \\ 0.08 \end{pmatrix}$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.5825 \\ 0 \\ 0.08 \\ 0.08 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.16 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.5825 \\ 0 \\ 0.08 \\ 0.08 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.16 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.5825 \\ 0 \\ 0.08 \\ 0.08 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.16 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad !$

- $\pi_{k+1}^T = H \cdot \pi_k^T$

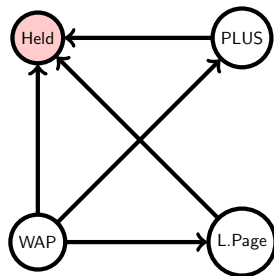
- $\pi_1^T = \begin{bmatrix} & \text{Held} & \text{WAP} & \text{L.Page} & \text{PLUS} \\ \text{Held} & 0 & 0 & 0 & 0 \\ \text{WAP} & 0.33 & 0 & 0.33 & 0.33 \\ \text{L.Page} & 1 & 0 & 0 & 0 \\ \text{PLUS} & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$

- $\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.5825 \\ 0 \\ 0.08 \\ 0.08 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0.16 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad !$

- Problem: in diesem Model verlässt der Benutzer nie die Seite Held

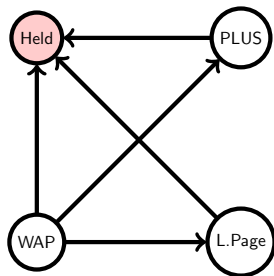
Problem: Dangling Nodes

- **Dangling Node:** Seite ohne ausgehende Links



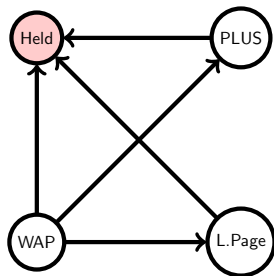
Problem: Dangling Nodes

- **Dangling Node:** Seite ohne ausgehende Links
- Problem: der Seitenrang wird nicht korrekt berechnet

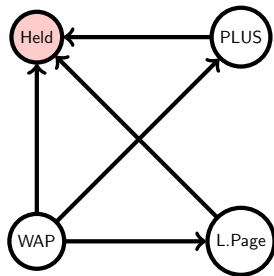


Problem: Dangling Nodes

- **Dangling Node:** Seite ohne ausgehende Links
- Problem: der Seitenrang wird nicht korrekt berechnet
- Lösung: wir springen von einer Dangling Node zufällig zu einer anderen Seite mit Wahrscheinlichkeit von je $\frac{1}{n}$

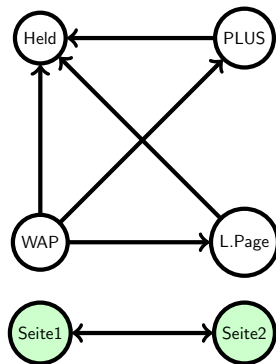


	Held	WAP	L.Page	PLUS
Held	0.25	0.25	0.25	0.25
WAP	0.33	0	0.33	0.33
L.Page	1	0	0	0
PLUS	1	0	0	0



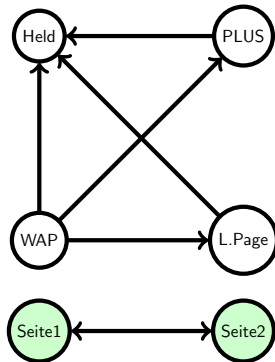
Problem: Zyklen

- **Zyklen:** Benutzer kommt in eine Schleife von Seiten



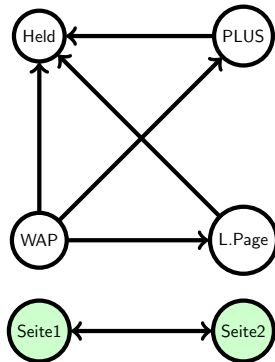
Problem: Zyklen

- **Zyklen:** Benutzer kommt in eine Schleife von Seiten
- Daher falsche Berechnung der Seitenränge



Problem: Zyklen

- **Zyklen:** Benutzer kommt in eine Schleife von Seiten
- Daher falsche Berechnung der Seitenränge
- Lösung: Random Surfer



- Dieser simuliert echtes Nutzerverhalten (z.B. neue URL eingeben)

- Dieser simuliert echtes Nutzerverhalten (z.B. neue URL eingeben)
- Mit Wahrscheinlichkeit d : Folge einem Link

- Dieser simuliert echtes Nutzerverhalten (z.B. neue URL eingeben)
- Mit Wahrscheinlichkeit d : Folge einem Link
- Mit Wahrscheinlichkeit $(1 - d)$: Springe zu einer zufälligen Seite

Modifizierte PageRank Formel

$$r(P_i) = d \cdot \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)} + (1 - d) \cdot \frac{1}{n}$$

- Dieser simuliert echtes Nutzerverhalten (z.B. neue URL eingeben)
- Mit Wahrscheinlichkeit d : Folge einem Link
- Mit Wahrscheinlichkeit $(1 - d)$: Springe zu einer zufälligen Seite

Modifizierte PageRank Formel

$$r(P_i) = d \cdot \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)} + (1 - d) \cdot \frac{1}{n}$$

- d : Damping Factor (typisch: $d = 0,85$)

- Dieser simuliert echtes Nutzerverhalten (z.B. neue URL eingeben)
- Mit Wahrscheinlichkeit d : Folge einem Link
- Mit Wahrscheinlichkeit $(1 - d)$: Springe zu einer zufälligen Seite

Modifizierte PageRank Formel

$$r(P_i) = d \cdot \sum_{P_j \in B(P_i)} \frac{r(P_j)}{o(P_j)} + (1 - d) \cdot \frac{1}{n}$$

- d : Damping Factor (typisch: $d = 0,85$)
- $\frac{1}{n}$: Wahrscheinlichkeit auf eine beliebige Seite zu springen

Verbesserte Matrix G

- Wir bauen die neue Googlematrix G folgendermaßen:

- Wir bauen die neue Googlematrix G folgendermaßen:
- $G_{i,j} := d \cdot H_{i,j} + (1 - d) \cdot \frac{1}{n}$

- Wir bauen die neue Googlematrix G folgendermaßen:
- $G_{i,j} := d \cdot H_{i,j} + (1 - d) \cdot \frac{1}{n}$
- $d \cdot H_{i,j}$ entspricht der Wahrscheinlichkeit, dass wir einen Link von i nach j folgen

- Wir bauen die neue Googlematrix G folgendermaßen:
- $G_{i,j} := d \cdot H_{i,j} + (1 - d) \cdot \frac{1}{n}$
- $d \cdot H_{i,j}$ entspricht der Wahrscheinlichkeit, dass wir einen Link von i nach j folgen
- $(1 - d) \cdot \frac{1}{n}$ entspricht der Wahrscheinlichkeit, dass wir auf eine zufällige Seite springen

verbesserte Matrix G Beispiel

verbesserte Matrix G Beispiel

- | | Held | WAP | L.Page | PLUS | Seite1 | Seite2 |
|--------|-------|-------|--------|-------|--------|--------|
| Held | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| WAP | 0.33 | 0 | 0.33 | 0.33 | 0 | 0 |
| L.Page | 1 | 0 | 0 | 0 | 0 | 0 |
| PLUS | 1 | 0 | 0 | 0 | 0 | 0 |
| Seite1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Seite2 | 0 | 0 | 0 | 0 | 1 | 0 |

- | | Held | WAP | L.Page | PLUS | Seite1 | Seite2 |
|--------|-------|-------|--------|-------|--------|--------|
| Held | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 |
| WAP | 0.33 | 0 | 0.33 | 0.33 | 0 | 0 |
| L.Page | 1 | 0 | 0 | 0 | 0 | 0 |
| PLUS | 1 | 0 | 0 | 0 | 0 | 0 |
| Seite1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Seite2 | 0 | 0 | 0 | 0 | 1 | 0 |

- $$G_{i,j} := 0.85 \cdot H_{i,j} + 0.15 \cdot \frac{1}{6}$$

verbesserte Matrix G Beispiel

$$G = \begin{array}{c|cccccc} & \text{Held} & \text{Youtube} & \text{Hallo} & \text{PLUS} & \text{Seite1} & \text{Seite2} \\ \hline \text{Held} & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 & 0.1667 \\ \text{Youtube} & 0.3083 & 0.0250 & 0.3083 & 0.3083 & 0.0250 & 0.0250 \\ \text{Hallo} & 0.8750 & 0.0250 & 0.0250 & 0.0250 & 0.0250 & 0.0250 \\ \text{PLUS} & 0.8750 & 0.0250 & 0.0250 & 0.0250 & 0.0250 & 0.0250 \\ \text{Seite1} & 0.0250 & 0.0250 & 0.0250 & 0.0250 & 0.0250 & 0.8750 \\ \text{Seite2} & 0.0250 & 0.0250 & 0.0250 & 0.0250 & 0.8750 & 0.0250 \end{array}$$

- Wir wissen

Iterationsverfahren

$$\pi_{k+1}^T = H \cdot \pi_k^T$$

- Wir wissen

Iterationsverfahren

$$\pi_{k+1}^T = H \cdot \pi_k^T$$

- Was passiert mit $k \rightarrow \infty$?

Definition: Markov-Kette

Eine **Markov-Kette** ist ein stochastischer Prozess $(X_0, X_1, X_2, \dots)$ mit der **Markov-Eigenschaft**:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$$

Definition: Markov-Kette

Eine **Markov-Kette** ist ein stochastischer Prozess $(X_0, X_1, X_2, \dots)$ mit der **Markov-Eigenschaft**:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$$

- X_n steht für den Zustand zum Zeitpunkt n

Definition: Markov-Kette

Eine **Markov-Kette** ist ein stochastischer Prozess $(X_0, X_1, X_2, \dots)$ mit der **Markov-Eigenschaft**:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$$

- X_n steht für den Zustand zum Zeitpunkt n
- Für die Berechnung des nächsten Zustands X_{n+1} benötigen wir lediglich den aktuellen Zustand X_n - die vergangenen Zustände sind irrelevant

Definition: Markov-Kette

Eine **Markov-Kette** ist ein stochastischer Prozess $(X_0, X_1, X_2, \dots)$ mit der **Markov-Eigenschaft**:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$$

- X_n steht für den Zustand zum Zeitpunkt n
- Für die Berechnung des nächsten Zustands X_{n+1} benötigen wir lediglich den aktuellen Zustand X_n - die vergangenen Zustände sind irrelevant
- Beschrieben durch eine **Übergangsmatrix** P mit Einträgen $P_{ij} = P(X_{n+1} = j \mid X_n = i)$

Definition: Markov-Kette

Eine **Markov-Kette** ist ein stochastischer Prozess $(X_0, X_1, X_2, \dots)$ mit der **Markov-Eigenschaft**:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$$

- X_n steht für den Zustand zum Zeitpunkt n
- Für die Berechnung des nächsten Zustands X_{n+1} benötigen wir lediglich den aktuellen Zustand X_n - die vergangenen Zustände sind irrelevant
- Beschrieben durch eine **Übergangsmatrix** P mit Einträgen $P_{ij} = P(X_{n+1} = j \mid X_n = i)$
- Für PageRank bedeutet das: Aus $\pi_{k+1}^T = H \cdot \pi_k^T$ folgt, dass wir für die $(k+1)$ -te Iteration nur den Vektor π_k^T zum Zeitpunkt k benötigen

Defintion: Stochastische Matrix

Eine Matrix P heißt **stochastisch**, wenn:

- Alle Einträge nicht-negativ: $P_{ij} \geq 0$
- Jede Zeile summiert sich zu 1: $\sum_j P_{ij} = 1$

Defintion: Stochastische Matrix

Eine Matrix P heißt **stochastisch**, wenn:

- Alle Einträge nicht-negativ: $P_{ij} \geq 0$
- Jede Zeile summiert sich zu 1: $\sum_j P_{ij} = 1$

Überzeugen Sie sich selbst! (Achtung: gerundete Werte)

$G =$		Held	Youtube	Hallo	PLUS	Seite1	Seite2
	Held	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	Youtube	0.3083	0.0250	0.3083	0.3083	0.0250	0.0250
	Hallo	0.8750	0.0250	0.0250	0.0250	0.0250	0.0250
	PLUS	0.8750	0.0250	0.0250	0.0250	0.0250	0.0250
	Seite1	0.0250	0.0250	0.0250	0.0250	0.0250	0.8750
	Seite2	0.0250	0.0250	0.0250	0.0250	0.8750	0.0250

Definition: Stationäre Verteilung

Ein Wahrscheinlichkeitsvektor $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$ heißt **stationäre Verteilung** einer Markov-Kette mit Übergangsmatrix P , wenn:

$$\pi^T = P \cdot \pi^T \quad \text{und} \quad \sum_{i=1}^n \pi_i = 1, \quad \pi_i \geq 0$$

Definition: Stationäre Verteilung

Ein Wahrscheinlichkeitsvektor $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$ heißt **stationäre Verteilung** einer Markov-Kette mit Übergangsmatrix P , wenn:

$$\pi^T = P \cdot \pi^T \quad \text{und} \quad \sum_{i=1}^n \pi_i = 1, \quad \pi_i \geq 0$$

- Die Verteilung bleibt im Gleichgewicht

Definition: Stationäre Verteilung

Ein Wahrscheinlichkeitsvektor $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$ heißt **stationäre Verteilung** einer Markov-Kette mit Übergangsmatrix P , wenn:

$$\pi^T = P \cdot \pi^T \quad \text{und} \quad \sum_{i=1}^n \pi_i = 1, \quad \pi_i \geq 0$$

- Die Verteilung bleibt im Gleichgewicht
- Langfristige Aufenthaltswahrscheinlichkeit in jedem Zustand

Definition: Stationäre Verteilung

Ein Wahrscheinlichkeitsvektor $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$ heißt **stationäre Verteilung** einer Markov-Kette mit Übergangsmatrix P , wenn:

$$\pi^T = P \cdot \pi^T \quad \text{und} \quad \sum_{i=1}^n \pi_i = 1, \quad \pi_i \geq 0$$

- Die Verteilung bleibt im Gleichgewicht
- Langfristige Aufenthaltswahrscheinlichkeit in jedem Zustand
- Beim PageRank: π_i = Wichtigkeit von Website i

Definition: Stationäre Verteilung

Ein Wahrscheinlichkeitsvektor $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$ heißt **stationäre Verteilung** einer Markov-Kette mit Übergangsmatrix P , wenn:

$$\pi^T = P \cdot \pi^T \quad \text{und} \quad \sum_{i=1}^n \pi_i = 1, \quad \pi_i \geq 0$$

- Die Verteilung bleibt im Gleichgewicht
- Langfristige Aufenthaltswahrscheinlichkeit in jedem Zustand
- Beim PageRank: π_i = Wichtigkeit von Website i

Existiert immer eine (eindeutige) stationäre Verteilung?

Theorem: Perron-Frobenius (vereinfacht)

Sei M eine Markov-Matrix, bei der alle Einträge positiv sind ($M_{ij} > 0$ für alle i, j). Dann gilt:

- M hat eine eindeutige stationäre Verteilung π
- Für jeden beliebigen Startvektor π_0 gilt:

$$\pi_k = M^k \cdot \pi_0 \rightarrow \pi \quad \text{für } k \rightarrow \infty$$

Theorem: Perron-Frobenius (vereinfacht)

Sei M eine Markov-Matrix, bei der alle Einträge positiv sind ($M_{ij} > 0$ für alle i, j). Dann gilt:

- M hat eine eindeutige stationäre Verteilung π
- Für jeden beliebigen Startvektor π_0 gilt:

$$\pi_k = M^k \cdot \pi_0 \rightarrow \pi \quad \text{für } k \rightarrow \infty$$

Bedeutung:

- Konvergenz ist unabhängig vom Startvektor
- Konvergenz ist für die Struktur von PageRank garantiert

- Wir definieren einen Threshold ϵ

- Wir definieren einen Threshold ϵ
- Iterative Multiplikation: $\pi_{k+1}^T = G \cdot \pi_k^T$

- Wir definieren einen Threshold ϵ
- Iterative Multiplikation: $\pi_{k+1}^T = G \cdot \pi_k^T$
- Stoppt wenn $|\pi_{k+1}^T - \pi_k^T| < \epsilon$

- Wir betrachten eine naive Implementierung ohne Optimierung der Matrix

- Wir betrachten eine naive Implementierung ohne Optimierung der Matrix
- Pro Iteration berechnen wir $\pi_{t+1}^T = \pi_t^T G$.

- Wir betrachten eine naive Implementierung ohne Optimierung der Matrix
- Pro Iteration berechnen wir $\pi_{t+1}^T = \pi_t^T G$.
- Eine Matrix-Vektor-Multiplikation mit einer dichten $n \times n$ -Matrix kostet $\mathcal{O}(n^2)$.

- Wir betrachten eine naive Implementierung ohne Optimierung der Matrix
- Pro Iteration berechnen wir $\pi_{t+1}^T = \pi_t^T G$.
- Eine Matrix-Vektor-Multiplikation mit einer dichten $n \times n$ -Matrix kostet $\mathcal{O}(n^2)$.
- Bei k Iterationen ergibt sich die Gesamtlaufzeit $\mathcal{O}(k \cdot n^2)$.

- PageRank misst die Wichtigkeit von Websites durch Linkstruktur

- PageRank misst die Wichtigkeit von Websites durch Linkstruktur
- Random Surfer simuliert Benutzerverhalten

- PageRank misst die Wichtigkeit von Websites durch Linkstruktur
- Random Surfer simuliert Benutzerverhalten
- Hyperlink-Matrix H + Damping Factor $\rightarrow$ Google Matrix G

- PageRank misst die Wichtigkeit von Websites durch Linkstruktur
- Random Surfer simuliert Benutzerverhalten
- Hyperlink-Matrix H + Damping Factor $\rightarrow$ Google Matrix G
- Berechnung durch iterativen Prozess

- PageRank misst die Wichtigkeit von Websites durch Linkstruktur
- Random Surfer simuliert Benutzerverhalten
- Hyperlink-Matrix H + Damping Factor $\rightarrow$ Google Matrix G
- Berechnung durch iterativen Prozess

Kernidee

Die Wichtigkeit einer Seite ergibt sich aus der Wichtigkeit der Seiten, die auf sie verlinken.

Googles PageRank-Algorithmus

Niklas Leon Draxl, Mateja Jankovic

28. Februar 2026