

## Wie Googles PageRank-Algorithmus funktioniert

- $\Gamma = (V, E)$  sei ein gerichteter Graph mit  $V = \{1, 2, \dots, k\}$  und  $E \subseteq V \times V$
- Interpretation  $V$  ist eine (grosse) Menge von Webseite,  $E$  beschreibt die Verlinkungen unter diesen Seiten
- Von  $\Gamma$  werden keine besonderen Eigenschaften verlangt (Zusammenhang, starker Zusammenhang, Primitivität);  $\Gamma$  kann auch Nullzeilen enthalten; in der Regele ist  $A$  eine dünn besetzte Matrix
- $A = [A_{i,j}]_{1 \leq i, j \leq k} = A_\Gamma$  sei die Adjazenzmatrix von  $G$ :

$$A_{i,j} = \begin{cases} 1 & \text{falls } (i, j) \in E \\ 0 & \text{sonst} \end{cases}$$

- $\mathbf{1}_k = (1, 1, \dots, 1)$  Einsvektor der Länge  $k$  (Zeilenvektor)  
Beachte

$$\mathbf{1}_k \cdot \mathbf{1}_k^\top = k$$

$$\mathbf{1}_k^\top \cdot \mathbf{1}_k = \mathbb{J}_k = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

- Vektor der Zeilensummen von  $A$

$$A \cdot \mathbf{1}_k^\top = \mathbf{r}^\top = (r_1, r_2, \dots, r_k)^\top$$

Ferner seien

$$\mathbf{u} = (u_1, u_2, \dots, u_k) \quad \text{mit} \quad u_i = \begin{cases} 1 & \text{falls } r_i > 0 \\ 0 & \text{falls } r_i = 0 \end{cases}$$

$$\mathbf{v} = (v_1, v_2, \dots, v_k) \quad \text{mit} \quad v_i = \begin{cases} 0 & \text{falls } r_i > 0 \\ 1 & \text{falls } r_i = 0 \end{cases}$$

also  $\mathbf{u} + \mathbf{v} = \mathbf{1}_k$

- Für die Matrix  $B$

$$B = [B_{i,j}]_{1 \leq i, j \leq k} \quad \text{mit} \quad B_{i,j} = \begin{cases} \frac{A_{i,j}}{r_i} & \text{falls } r_i > 0 \\ 0 & \text{falls } r_i = 0 \end{cases}$$

gilt also

$$B \cdot \mathbf{1}_k^\top = \mathbf{u}^\top$$

- Die Matrix

$$C = [C_{i,j}]_{1 \leq i,j \leq k}$$

entsteht dadurch, dass man in  $B$  eventuelle Nullzeilen durch stochastische Vektoren  $\frac{1}{k} \mathbf{1}_k$  ersetzt:

$$C = B + \frac{1}{k} \mathbf{v}^t \cdot \mathbf{1}_k$$

$C$  ist eine stochastische Matrix. Nichtnegativ ist sie offensichtlich und es ist

$$C \cdot \mathbf{1}_k^t = B \cdot \mathbf{1}_k^t + \frac{1}{k} \mathbf{v}^t \cdot \mathbf{1}_k \cdot \mathbf{1}_k^t = \mathbf{u}^t + \mathbf{v}^t = \mathbf{1}_k^t$$

$C$  muss aber nicht primitiv sein!

- Sei nun  $\gamma$  mit  $0 < \gamma < 1$  ein Parameter: die sogenannte *Teleportationskonstante*. Sie gibt die Wahrscheinlichkeit dafür an, dass ein Surfer von einer Webseite “ $i$ ” aus einem der  $r_i$  Links folgt, und zwar mit Gleichverteilung. Mit Wahrscheinlichkeit  $1 - \gamma$  springt er zu irgendeiner der anderen Seiten (wiederum gleichverteilt). Die *Googlematrix*  $G$  ist nun

$$G = \gamma C + (1 - \gamma) \frac{1}{k} \mathbb{J}_k$$

Als konvexe Kombination von stochastischen Matrizen ist  $G$  wiederum eine stochastische Matrix. Ausserdem ist  $G$  primitiv, da  $G$  ein positive Matrix ist.

- Gesucht ist also der eindeutig bestimmte (positive) Linkseigenvektor  $\mathbf{x}$  von  $G$  mit  $\|\mathbf{x}\|_1 = \sum_{1 \leq i \leq k} x_i = 1$  zum Eigenwert  $\lambda = 1$ : das ist der PageRank-Vektor.
- Wie berechnet man  $\mathbf{x}$ ? Lösen eines riesigen Eigenwertproblems kommt nicht in Frage. Man macht es durch Approximation!

$$\begin{aligned} \mathbf{x}^{(n+1)} &= \mathbf{x}^{(n)} \cdot G \\ &= \gamma \mathbf{x}^{(n)} \cdot B + \frac{1}{k} \left( \mathbf{x}^{(n)} (\gamma \mathbf{v}^t + (1 - \gamma) \mathbf{1}_k^t) \right) \mathbf{1}_k \end{aligned}$$

Das hat den Vorteil, dass im Matrix-Vektorprodukt  $\mathbf{x}^{(n)} \cdot G$  die Matrix mutmasslich dünn besetzt ist.