Page Rank

Grundlagen

Michaela Bachmaier, Florian Babl 23.01.2020

Gliederung

- Mathematische Grundbegriffe
- Idee und Algorithmus
- Probleme und Lösungen
- Potenzmethode
- Erweiterungen

Eigenwerte und Eigenvektoren
Eigenvalues and Eigenvectors

- **Eigenvektor** x einer Matrix M: vom Nullvektor verschieden, Richtung ändert sich durch Multiplikation mit der Matrix nicht → Streckung bzw. Stauchung
- **Formel: Mx = λx**
- "Streckungsfaktor" **λ** heißt **Eigenwert**
- eine Matrix kann mehrere Eigenwerte haben; zu jedem Eigenwert gibt es passende Eigenvektoren
- dominanter Eigenwert: betragsmäßig größter Eigenwert einer Matrix

Eigenwerte und Eigenvektoren
Eigenvalues and Eigenvectors

For the matrix A

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

the vector

$$\mathbf{x} = \begin{bmatrix} 3 \\ -3 \end{bmatrix}$$

is an eigenvector with eigenvalue 1. Indeed,

$$A\mathbf{x} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 3 \\ -3 \end{bmatrix} = \begin{bmatrix} (2 \cdot 3) + (1 \cdot (-3)) \\ (1 \cdot 3) + (2 \cdot (-3)) \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \end{bmatrix} = 1 \cdot \begin{bmatrix} 3 \\ -3 \end{bmatrix}.$$

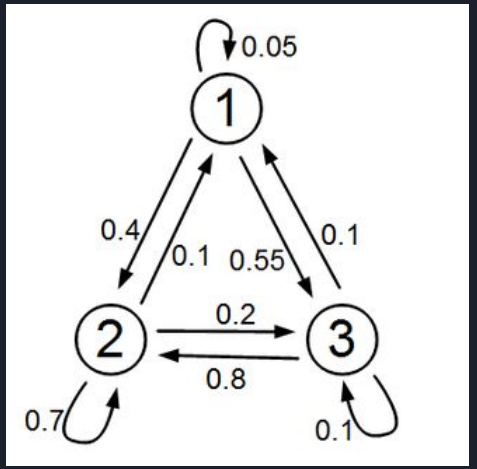
Übergangsmatrix
Stochastic Matrix

- drückt Übergangswahrscheinlichkeiten von diskreten und kontinuierlichen Markow-Ketten aus
- quadratische Matrix
- zeilenstochastische Übergangsmatrix: Zeilensumme 1
- spaltenstochastische Übergangsmatrix: Spaltensumme 1
- Werte der Einträge liegen zwischen 0 und 1

Übergangsmatrix
Stochastic Matrix

	München	Paris	Rom
München	0,05	0,4	0,55
Paris	0,1	0,7	0,2
Rom	0,1	0,8	0,1

$$P = \begin{pmatrix} 0{,}05 & 0{,}4 & 0{,}55 \\ 0{,}1 & 0{,}7 & 0{,}2 \\ 0{,}1 & 0{,}8 & 0{,}1 \end{pmatrix}$$



Irreduzible Matrix
Irreducible Matrix

Eine Matrix M heißt irreduzibel, wenn von jedem Zustand aus ein Übergang zu jedem Zustand existiert.

Ersetzt man alle von 0 verschiedenen Einträge durch 1 und betrachtet die Matrix als Adjazenzmatrix eines gerichteten Graphen, muss ein stark zusammenhängender Graph entstehen, damit man von einer irreduziblen Matrix sprechen kann.

Page Rank

Idee und Algorithmus

Problemstellung

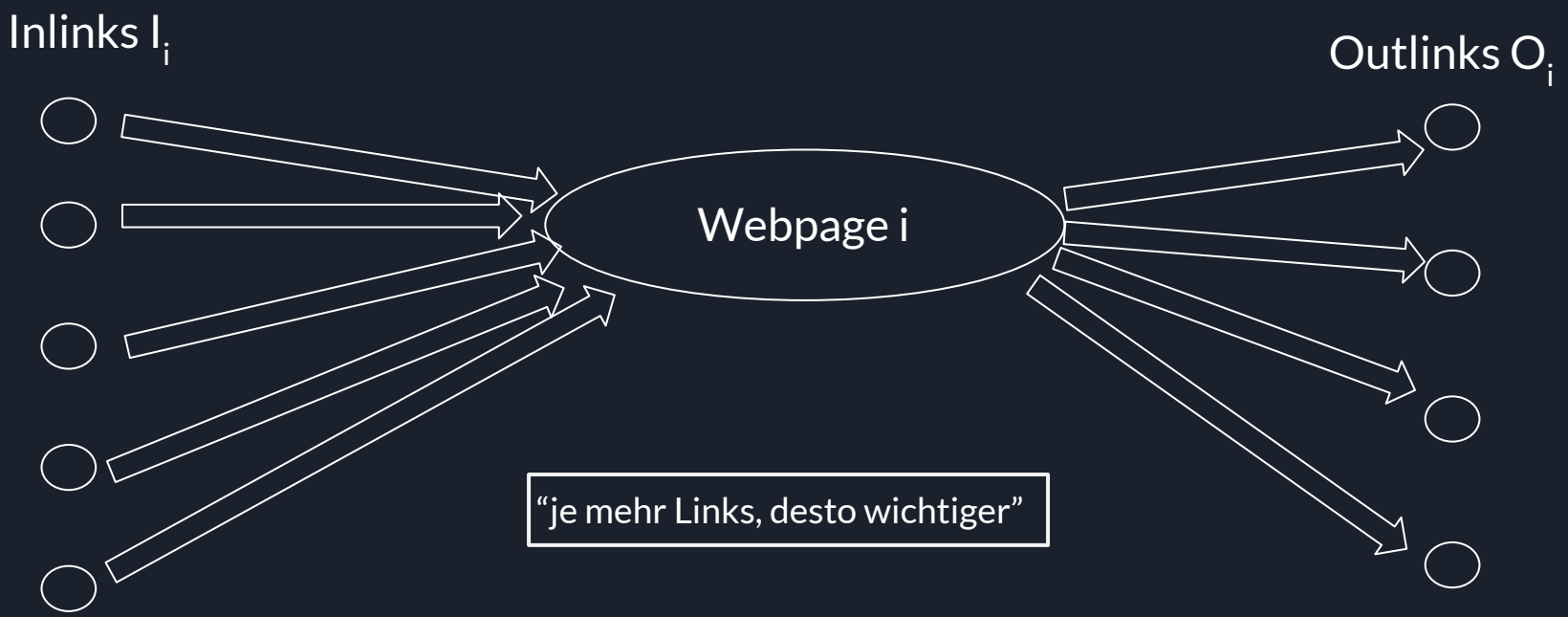
- Suchmaschine: Finden eines Suchbegriffs in einer Vielzahl von Dokumenten
- Mit dem gesamten Internet aufgrund seiner Größe problematisch umzusetzen

→ Nach Relevanz sortieren

→ Page Rank: Sortierung basierend auf Verlinkungsstruktur

Grundidee:

Anzahl an Links zu bzw. von einer Seite i trifft Aussage über ihre Relevanz



Problem: leicht zu manipulieren

Lösung: Page Rankdes Inlinks einbeziehen

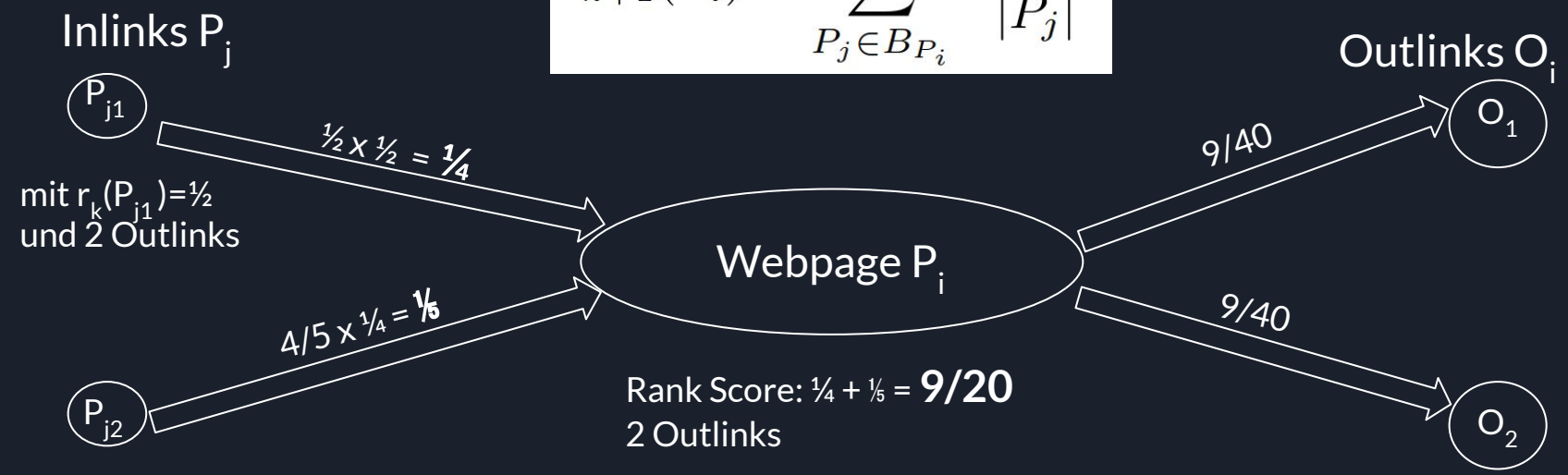
- Page Rank der Seite i ist die gewichtete Summe der Rank Scores der Seiten j_n, die Outlinks zu i haben
- Rank Score der Seite j wird gleichmäßig auf alle Outlinks verteilt
- Formel:

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}$$

→ **Wenn eine Seite j_1 mit einem hohen Rank Score auf i verweist, trägt das zu einem größeren Teil zum Rank Score von i bei, als wenn eine Seite j_2 mit niedrigem Rank Score auf i verlinkt.**

Beispiel

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



Rank Score von O1 und O2:
9/40 + Scores von anderen Inlinks

Mathematische Darstellung

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

Vorhergehende Formel berechnet den Page Rank für jede Seite einzeln.
⇒ Übergangsmatrix *H* im Format nxn (n = Seitenanzahl)

$$\boldsymbol{\pi}^{(k+1)T} = \boldsymbol{\pi}^{(k)T}\mathbf{H}.$$

$\boldsymbol{\pi}^{(k+1)T}$ = Rank-Vektor in der *k+1*-ten (nächsten) Iteration

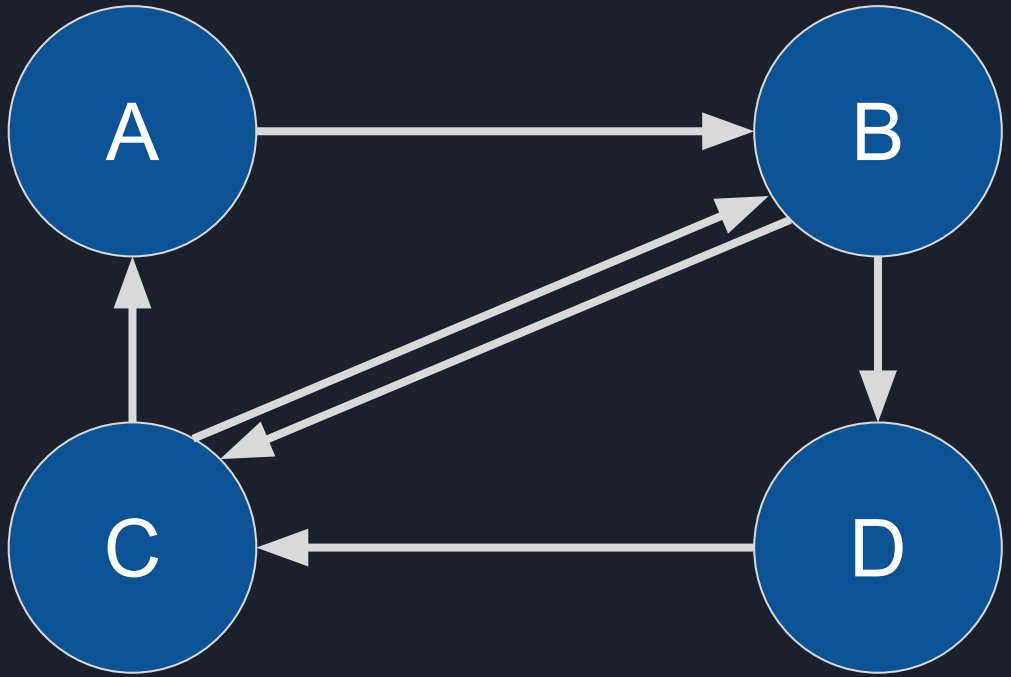
Random Surfer / Markov Kette

- Websurfer, der Hyperlinks (Outlinks) nutzt, um von Seite zu Seite zu springen.
- Wählt nächsten Link zufällig.

Resultat:

- Websurfer verbringt auf manchen Seiten mehr Zeit als auf anderen
- Diese Seiten werden als relevanter eingestuft

Random Surfer / Markov Kette



$$\boldsymbol{\pi}^{(k+1)T} = \boldsymbol{\pi}^{(k)T}\mathbf{H}.$$

H =

	A	B	C	D
A	0	0	½	½
B	1	0	0	½
C	0	½	0	0
D	0	½	½	0

$\boldsymbol{\pi}^{(0)T}$ =

[0,25 0,25, 0,25, 0,25]

Random Surfer / Markov Kette

- Matrixmultiplikation gibt uns

$$\boldsymbol{\pi}^{(1)} = \begin{bmatrix} 0{,}25 \\ 0{,}375 \\ 0{,}125 \\ 0{,}25 \end{bmatrix}$$

- Vorgang wiederholen, bis die Werte konvergieren
- $\boldsymbol{\pi}^{(k+1)T}$ stellt Eigenvektor von H mit Eigenwert 1 dar

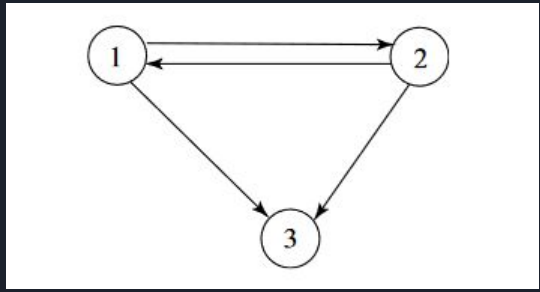
Page Rank

Probleme und Lösungen

Probleme: Sinks

- Seiten, die in jeder Iteration mehr und mehr PageRank anhäufen, aber keine Outlinks haben.

  ⇒ PageRank mancher Seiten bleibt bei 0.
  Beispiel: PDFs, Bilder etc.



- Lösung: stochasticity adjustment
  Jedes “Loch” bekommt Outlinks mit *1/n* zu jeder Seite.

$$\mathbf{H} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{array} \begin{array}{c} \begin{array}{cccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

$$\Rightarrow \quad \mathbf{S} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Anpassung der Formel

$$\mathbf{S} = \mathbf{H} + \mathbf{a}(1/n\,\mathbf{e}^T)$$

H = Ürsprüngliche Übergangsmatrix

n = Anzahl der Seiten

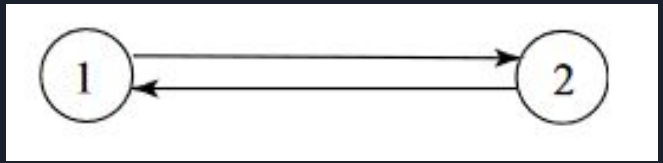
a = 1 wenn $page_i$ ein Sink sonst 0

e = $(1\ 1\ 1\ 1\ 1)^T$

Probleme: Cycles

- Werte sind nach jeder Iteration vertauscht
- Random Surfer kann hängen bleiben

$$(1\ 0) \Leftrightarrows (0\ 1)$$



Lösung: Random Teleportation

Nach einer gewissen Zeit wird dem Surfer langweilig und er springt zu einer zufälligen Website.

Erweiterung der Formel

$$\mathbf{S} = \mathbf{H} + \mathbf{a}(1/n\,\mathbf{e}^T)$$

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)1/n\,\mathbf{e}\mathbf{e}^T$$

H = Ürsprüngliche Übergangsmatrix

n = Anzahl der Seiten

a = 1 wenn $page_i$ ein Sink sonst 0

e = $(1\ 1\ 1\ 1\ 1)^T$

α= Wert zw. 1 und 0. Gibt an wie oft der Surfer teleportiert.

ee^T= uniforme Matrix *nxn* mit Values *1/n*

Google Matrix G

$$= \alpha \mathbf{H} + (\alpha \mathbf{a} + (1 - \alpha)\mathbf{e})\, 1/n\, \mathbf{e}^T$$

$$\mathbf{G} = .9\mathbf{H} + (.9 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + .1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}) \, 1/6 \, (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$$

$$= \begin{pmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{pmatrix}.$$

$$\boldsymbol{\pi}^T = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix}$$
$$\begin{pmatrix} .03721 & .05396 & .04151 & .3751 & .206 & .2862 \end{pmatrix}$$

Problem: Berechnung

Mit Eigenvektor-Berechnung auf π^T kommen:
$\pi^T = \pi^T G$
$\pi^T e = 1$

⇒ $\pi^T e = 1$ stellt sicher, dass π^T ein Wahrscheinlichkeitsvektor ist
⇒ Aber wie berechnet man den Eigenvektor für eine Matrix mit >8 Billionen auf >8 Billionen Einträgen?

$G(G(.....(G\pi^T) = G^n\pi^T$ bis π^T konvergiert. ⇒ Potenzmethode

Page Rank

Potenzmethode

Potenzmethode

- Intuitivste Methode den dominanten Eigenwert und Eigenvektor einer Matrix zu finden
- Eigentlich langsam und es würde modernere Methoden geben
- G eigentlich dense ⇒ lange Berechnung
- Matrixmult. mit sparse Matrix H ⇒ schnelle Berechnung

$$
\begin{aligned}
\boldsymbol{\pi}^{(k+1)T} &= \boldsymbol{\pi}^{(k)T}\mathbf{G} \\
&= \alpha\,\boldsymbol{\pi}^{(k)T}\mathbf{S} + \frac{1-\alpha}{n}\,\boldsymbol{\pi}^{(k)T}\,\mathbf{e}\,\mathbf{e}^{T} \\
&= \alpha\,\boldsymbol{\pi}^{(k)T}\mathbf{H} + \left(\alpha\,\boldsymbol{\pi}^{(k)T}\mathbf{a} + 1 - \alpha\right)\mathbf{e}^{T}/n
\end{aligned}
$$

Potenzmethode

- Matrix wird nicht manipuliert (siehe stochasticity adjustment)
- Andere Methoden manipulieren die Matrix bei jedem Schritt
- Sparse Matrix spart Speicherplatz
- Jede Iteration benötigt O(n) Laufzeit

⇒ Meistens ca. 50 Iterationen benötigt bis Ranking-Vektor π^T konvergiert.

Warum? *asymptotic rate of convergence* von Markov-Ketten

Page Rank

Erweiterungen

Erweiterungen

- Search Engine Optimization
- User-Profil
- Suchort
- Wonach andere Nutzer suchen
- alter Suchverlauf

⇒ Unklar, wie all das in der Berechnung eine Rolle spielt und Google verrät es auch nicht.

Quellen

Empfehlung:

- Langville, A. & Meyer, C. (2011). *Google's PageRank and Beyond. The Science of Search Engine Rankings*. Princeton: Princeton University Press

Sonstige:

- https://www.mathebibel.de/eigenwerte-eigenvektoren
- https://de.wikipedia.org/wiki/%C3%9Cbergangsmatrix
- https://en.wikipedia.org/wiki/Irreducibility_(mathematics)
- Eldén Lars. (2007). *Matrix methods in data mining and pattern recognition*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- https://www.youtube.com/watch?v=qxEkY8OScYY
- Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry (1999) *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.