Vorlesung
Big Data Analytics
in Theorie und Praxis — Theorieteil

Prof. Dr. Nicole Schweikardt

Lehrstuhl Logik in der Informatik
Institut für Informatik
Humboldt-Universität zu Berlin

Kapitel 1:

PageRank: Markov-Ketten als Grundlage der Funktionsweise von Suchmaschinen im Internet

Abschnitt 1.1:
Einleitung

Suchmaschine:

Eingabe:

Eine Anfrage, bestehend aus einem oder mehreren Stichworten

Ziel:

Eine nach Relevanz sortierte Liste von Webseiten, die Informationen zu den Stichworten enthalten

Herausforderungen:

- es gibt sehr viele Webseiten
  bereits in 2008 mehr als 1 Billion (also 10^{12}) URLs
  Quelle: Google „Official Blog“, 25. Juli 2008
- ständig kommen neue hinzu
- viele Webseiten werden täglich aktualisiert;
  manche nach einiger Zeit auch wieder gelöscht
- niemand kennt den genauen Inhalt des gesamten Internets
- trotzdem müssen Suchanfragen in „Echtzeit“ beantwortet werden

Die Herausforderung besteht darin, Anfragen für einen sich rasant ändernden Suchraum gigantischer Größe ohne merkliche Reaktionszeit zu beantworten.

Abschnitt 1.2:

Die Architektur von Suchmaschinen

Suchmaschinen nutzen u.a. die folgenden Komponenten:

(1) Web-Crawler: Computerprogramme, die Crawler genannt werden, durchforsten das Internet, um neue oder veränderte Webseiten zu identifizieren. Die von den Crawlern gefundenen Informationen über Webseiten und deren Inhalt werden aufbereitet und gespeichert.

(2) Indexierung: Die Informationen werden in einer Datenstruktur gespeichert, mit deren Hilfe bei Eingabe eines Suchworts in „Echtzeit“ alle Webseiten ermittelt werden können, die das Suchwort enthalten.

(3) Bewertung der Webseiten: Die ausgewählten Webseiten werden im Hinblick auf ihren Informationsgehalt (hinsichtlich möglicher Suchworte sowie hinsichtlich ihrer generellen Bedeutung im Internet) bewertet.

Einige Details dazu:

Zu jeder vom Crawler gefundenen Webseite wird die URL (d.h. die Adresse) sowie der Inhalt der Webseite gespeichert.

Der Inhalt der Webseite wird analysiert und es werden Informationen darüber gespeichert, welches Wort mit welcher Häufigkeit und an welchen Positionen in der Webseite vorkommt (etwa: im Titel, als Überschrift, im Fließtext, mit welcher Schriftgröße etc.).
Diese Informationen werden im so genannten Index gespeichert.

Außerdem werden die Links, die auf Webseiten angegeben sind, analysiert. Enthält Webseite i einen Link auf eine Webseite j, so wird der Text, mit dem der Link beschriftet ist, im zu j gehörenden Index-Eintrag abgelegt. Diese Linkbeschriftungen geben wertvolle Hinweise darüber, welche Informationen die Webseite j enthält.

Der invertierte Index:

Aus dem Index wird der so genannte invertierte Index generiert.

Dies ist eine Datenstruktur, die zu jedem möglichen Suchwort eine Liste aller Webseiten angibt, die dieses Suchwort enthalten.

Dabei werden jeweils auch Zusatzinformationen gespeichert, die die Wichtigkeit des Suchworts innerhalb der Webseite beschreiben, z.B. die Häufigkeit des Stichworts, seine Position und Schriftgröße innerhalb der Webseite sowie das Vorkommen des Stichworts in Beschriftungen von Links auf die Webseite.

Der Web-Graph:

Die Link-Struktur des Internets kann man durch einen gerichteten Graphen modellieren, bei dem jede Webseite (d.h. jede URL) durch einen Knoten repräsentiert wird, und bei dem es eine Kante von Knoten i zu Knoten j gibt, wenn die Webseite i einen Link auf Webseite j enthält.

Dieser Graph wird Link-Index oder Web-Graph genannt.

Der Web-Graph wird üblicherweise als Adjazenzliste gespeichert.

Bearbeitung von Such-Anfragen:

Eingabe: eine Liste von Such-Stichworten

Ziel: finde die hinsichtlich dieser Stichworte informativsten Webseiten und zeige diese sortiert nach ihrer Relevanz an

Dabei werden folgende Kriterien berücksichtigt:

(A) die Häufigkeit und Positionierung der Suchbegriffe auf der jeweiligen Webseite sowie in der Beschriftung von Links, die auf diese Webseite verweisen, und

(B) die grundlegende Bedeutung einer Webseite.

Für (A) können Methoden aus dem Bereich Information Retrieval verwendet werden.

Für (B) wird die Link-Struktur des Internets, d.h. der Web-Graph berücksichtigt.

Die „grundlegende Bedeutung“ einer Webseite:

Das Maß für die „grundlegende Bedeutung“ einer Webseite wird aus der Link-Struktur des Internets gewonnen, ohne dass der textuelle Inhalt einer Webseite dabei berücksichtigt wird.

Als Rechtfertigung für die Güte dieses Ansatzes, geht man von der folgenden Annahme aus:

Wenn eine Webseite i einen Link auf eine Webseite j enthält, dann

- gibt es eine inhaltliche Beziehung zwischen beiden Webseiten, und
- der Autor der Webseite i hält die Informationen auf Webseite j für wertvoll.

Es gibt verschiedene Verfahren, die Maße für die grundlegende Bedeutung einer Webseite liefern, beispielsweise das von *Google* genutzte Page-Rank Verfahren von Brin und Page oder die HITS (Hypertext Induced Topic Search) Methode von Kleinberg.

Beide Ansätze versuchen, die in der Link-Struktur manifestierte „relative Wertschätzung“ zwischen einzelnen Webseiten in eine „grundlegende Bedeutung“ der Webseiten umzurechnen.

Bearbeitung einer Suchanfrage:

Bei der Bearbeitung einer Suchanfrage, bei der eine Liste s von Such-Stichworten eingegeben wird, wird unter Verwendung von (A) und (B) jeder Webseite i ein Wert $Score(i, s)$ zugeordnet, der als Maß für die Relevanz der Webseite i hinsichtlich der Suchanfrage s dient.

Als Trefferliste gibt die Suchmaschine dann eine Liste aller Webseiten aus, deren Score über einer bestimmten Schranke liegt und sortiert die Liste so, dass die Webseiten mit dem höchsten Score am weitesten oben stehen.

Wie der Wert $Score(i, s)$ gewählt wird, ist Betriebsgeheimnis der einzelnen Betreiber von Suchmaschinen.

Im Rest dieses Kapitels werden wir uns anhand des Page-Rank Verfahrens etwas genauer ansehen, wie die „grundlegende Bedeutung“ einer Webseite modelliert und berechnet werden kann.

Abschnitt 1.3:

Der Page-Rank einer Webseite

Der Page-Rank einer Webseite:

Der Page-Rank liefert ein Maß für die „grundlegende Bedeutung“ einer Webseite, das allein also aus der Link-Struktur des Internets bestimmt wird, ohne dass der textuelle Inhalt einer Webseite dabei berücksichtigt wird.

Wir schreiben im Folgenden $G = (V, E)$, um den Web-Graphen zu bezeichnen.

Der Einfachheit halber nehmen wir an, dass die Webseiten mit den Zahlen $1, \ldots, n$ durchnummeriert sind (wobei $n = |V|$ ist), und dass $V = \{1, 2, \ldots, n\}$ ist.

Jeder Knoten von G repräsentiert eine Webseite, und jede Kante $(i, j) \in E$ modelliert einen Link von Webseite i auf Webseite j.

Für jeden Knoten $i \in V$ sei

$$a_i \;:=\; \text{Aus-Grad}_G(i) \;=\; |\{j \in V \;:\; (i,j) \in E\}|$$

der Ausgangsgrad von i in G. D.h. a_i ist die Anzahl der Hyperlinks, die von der Webseite i auf andere Webseiten verweisen.

Für eine Webseite $j \in V$ schreiben wir $\text{Vor}_G(j)$, um die Menge aller Webseiten zu bezeichnen, die einen Link auf j enthalten, d.h.

$$\text{Vor}_G(j) \;=\; \{i \in V \;:\; (i,j) \in E\}.$$

Die Elemente in $\text{Vor}_G(j)$ werden Vorgänger von j genannt.

Die „grundlegende Bedeutung“ einer Webseite i wird im Folgenden durch eine Zahl PR_i modelliert, dem so genannten Page-Rank von i.

Der Wert PR_i soll die Qualität (im Sinne von „Renommee“ oder „Ansehen“) von Webseite i widerspiegeln; die Zahl PR_i soll umso größer sein, je höher das Renommee der Webseite i ist.

Das Renommee (und damit der Wert PR_j) einer Webseite j wird als hoch bewertet, wenn viele Webseiten i mit hohem Page-Rank PR_i einen Link auf die Seite j enthalten.

Die Werte PR_i, die allen Webseiten $i \in V$ zugeordnet werden, werden daher so gewählt, dass Folgendes gilt:

> *Eine Webseite i mit a_i ausgehenden Links „vererbt“ ihren Page-Rank an jede Webseite j mit $(i,j) \in E$ um den Anteil $\frac{\mathrm{PR}_i}{a_i}$.*

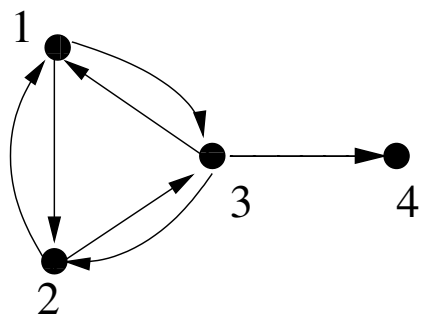
Mit dieser Sichtweise müsste also für alle $j \in V$ mit $\mathrm{Vor}_G(j) \neq \emptyset$ gelten:

$$\mathrm{PR}_j \;=\; \sum_{i \in \mathrm{Vor}_G(j)} \frac{\mathrm{PR}_i}{a_i}. \qquad (1)$$

Ein Problem:

Ein Problem stellen hierbei Knoten dar, deren Ausgangsgrad 0 ist, da solche Knoten ihren Page-Rank nicht an andere Knoten weitervererben und daher zu Werten PR_i führen können, die kein sinvolles Maß für die Bedeutung einer Webseite liefern.

Als Beispiel betrachte man den folgenden Graphen $G = (V, E)$:



Frage: Welche Werte $PR_1, PR_2, PR_3, PR_4 \in \mathbb{R}$ erfüllen die Gleichung (1)?

Antwort: Die einzigen Werte $PR_1, PR_2, PR_3, PR_4 \in \mathbb{R}$, die die Gleichung (1) erfüllen, sind $PR_1 = PR_2 = PR_3 = PR_4 = 0$.

Diese Werte spiegeln aber nicht die intuitive „grundlegende Bedeutung“ wider, die man den Webseiten 1, 2, 3 und 4 zuordnen würde!

Im Folgenden werden Knoten vom Ausgangsgrad 0 auch Senken genannt.

Zur Bestimmung des Page-Ranks betrachtet man in der Regel nur Graphen ohne Senken, d.h. gerichtete Graphen, bei denen jeder Knoten einen Ausgangsgrad $\geqslant 1$ hat.

Natürlich gibt es keine Garantie, dass der Web-Graph keine Senken besitzt. Brin und Page schlagen zwei Möglichkeiten vor, den Web-Graphen in einen Graphen ohne Senken zu transformieren:

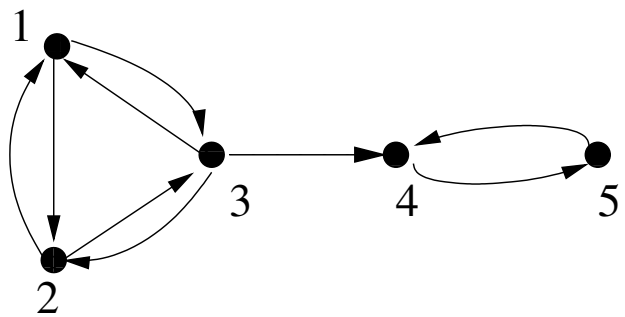
- Die eine Möglichkeit ist, von jeder Senke Kanten zu *allen* Knoten hinzuzufügen.
- Die andere Möglichkeit ist, alle Senken zu löschen und dies rekursiv so lange zu tun, bis ein Graph übrig bleibt, der keine Senke besitzt.

Wir nehmen im Folgenden an, dass eine dieser beiden Transformationen durchgeführt wurde und dass der Web-Graph durch einen endlichen gerichteten Graphen $G = (V, E)$ repräsentiert wird, der keine Senke besitzt.

Ein weiteres Problem:

Ein weiteres Problem stellen Knotenmengen dar, die unter sich zwar verbunden sind, die aber keine Kante zu einem anderen Knoten des Graphen G enthalten.

Als einfaches Beispiel betrachten wir den folgenden Graphen $G = (V, E)$:



Frage: Welche Werte $PR_1, PR_2, PR_3, PR_4, PR_5 \in \mathbb{R}$ erfüllen die Gleichung (1)?

Antwort: Man kann sich leicht davon überzeugen, dass Werte $PR_1, PR_2, PR_3, PR_4, PR_5 \in \mathbb{R}$ genau dann die Gleichung (1) erfüllen, wenn $PR_1 = PR_2 = PR_3 = 0$ und $PR_4 = PR_5$ ist.
Insbesondere kann für $PR_4 = PR_5$ jede beliebige Zahl gewählt werden!

Ähnlich wie im vorherigen Beispiel spiegeln diese Werte nicht die intuitive „grundlegende Bedeutung“ wider, die man den Webseiten 1–5 zuordnen würde!

D.h. die durch die Gleichung (1) gegebenen Werte $PR_1, \ldots, PR_5$ liefern kein sinnvolles Maß, um die grundlegende Bedeutung der einzelnen Webseiten zu bewerten.

Der Dämpfungsfaktor:

Um dieses Problem zu vermeiden, wird die Vererbung von PR_i auf die Nachfolgeseiten j mit $(i,j) \in E$ meistens um einen Dämpfungsfaktor d mit $0 \leqslant d \leqslant 1$ abgeschwächt.

Dies wird in der folgenden Definition präzisiert.

Die Page-Rank-Eigenschaft:

Definition 1.1 (Page-Rank-Eigenschaft)

Sei d eine reelle Zahl mit $0 \leqslant d \leqslant 1$. Die Zahl d wird im Folgenden Dämpfungsfaktor genannt.
Sei $G = (V, E)$ ein gerichteter Graph, der keine Senke besitzt, und sei $n := |V| \in \mathbb{N}_{\geqslant 1}$ und $V = \{1, \ldots, n\}$.
Für alle $i, j \in V$ sei $a_i := \text{Aus-Grad}_G(i)$ und $\text{Vor}_G(j) := \{i \in V \ : \ (i,j) \in E\}$.
Ein Tupel $\text{PR} = (\text{PR}_1, \ldots, \text{PR}_n) \in \mathbb{R}^n$ hat die Page-Rank-Eigenschaft bezüglich d, wenn für alle $j \in V$ gilt:

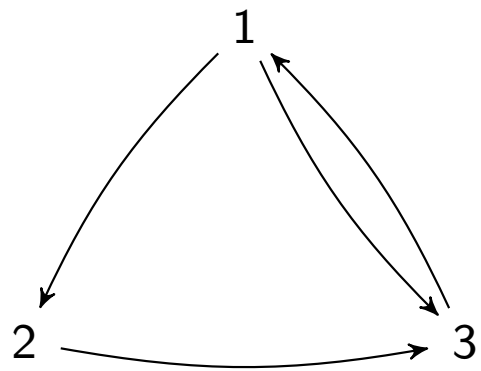
$$\text{PR}_j \quad = \quad \frac{1-d}{n} \quad + \quad d \cdot \sum_{i \in \text{Vor}_G(j)} \frac{\text{PR}_i}{a_i}. \tag{2}$$

Beachte

Für den Dämpfungsfaktor $d = 1$ erhält man gerade die Gleichung (1).
Für den Dämpfungsfaktor $d = 0$ ist $\text{PR}_1 = \text{PR}_2 = \ldots = \text{PR}_n = \frac{1}{n}$.
Brin und Page empfehlen, den Wert $d = 0.85 = \frac{17}{20}$ zu wählen.

Beispiel 1.2

Zur Veranschaulichung der Page-Rank-Eigenschaft betrachten wir den Dämpfungsfaktor $d := \frac{1}{2}$ und den folgenden Graphen $G = (V, E)$:



Wir suchen ein Tupel $\mathrm{PR} = (\mathrm{PR}_1, \mathrm{PR}_2, \mathrm{PR}_3)$ von reellen Zahlen, das die Page-Rank-Eigenschaft bzgl. $d = \frac{1}{2}$ hat, d.h. es gilt:

(1) $\mathrm{PR}_1 = \frac{1}{2 \cdot 3} + \frac{1}{2} \cdot \frac{\mathrm{PR}_3}{1}$

(2) $\mathrm{PR}_2 = \frac{1}{2 \cdot 3} + \frac{1}{2} \cdot \frac{\mathrm{PR}_1}{2}$

(3) $\mathrm{PR}_3 = \frac{1}{2 \cdot 3} + \frac{1}{2} \cdot \left(\frac{\mathrm{PR}_1}{2} + \frac{\mathrm{PR}_2}{1} \right)$.

Die Werte PR_1, PR_2 und PR_3 können wir daher finden, indem wir das Lineare Gleichungssystem lösen, das aus den folgenden drei Gleichungen besteht:

$$(1) \quad 1 \cdot \mathrm{PR}_1 - \tfrac{1}{2} \cdot \mathrm{PR}_3 = \tfrac{1}{6}$$
$$(2) \quad -\tfrac{1}{4} \cdot \mathrm{PR}_1 + 1 \cdot \mathrm{PR}_2 = \tfrac{1}{6}$$
$$(3) \quad -\tfrac{1}{4} \cdot \mathrm{PR}_1 - \tfrac{1}{2} \cdot \mathrm{PR}_2 + 1 \cdot \mathrm{PR}_3 = \tfrac{1}{6}$$

Die Auflösung dieses linearen Gleichungssystems (z.B. mittels Gauß-Elimination) liefert die Werte

$$\mathrm{PR}_1 = \tfrac{14}{39}, \quad \mathrm{PR}_2 = \tfrac{10}{39}, \quad \mathrm{PR}_3 = \tfrac{15}{39}.$$

□Ende Beispiel 1.2

Auf die gleiche Art wie in diesem Beispiel erhält man auch für den Web-Graphen und einen geeigneten Dämpfungsfaktor d ein entsprechendes lineares Gleichungssystem.

Um den Page-Rank der einzelnen Webseiten zu berechnen, müssen wir „nur“ dieses lineare Gleichungssystem lösen.

Dabei stellen sich folgende Probleme:

(1) Zunächst ist völlig unklar, ob dieses lineare Gleichungssystem überhaupt eine Lösung besitzt, und falls ja, ob die Lösung eindeutig ist.

Anhand von Definition 1.1 ist nämlich prinzipiell auch denkbar, dass es gar kein Tupel gibt, das die Page-Rank-Eigenschaft bzgl. d hat, oder dass es mehrere verschiedene Tupel gibt, die die Page-Rank-Eigenschaft bzgl. d besitzen.

(2) Das lineare Gleichungssystem hat n Unbekannte, wobei n die Anzahl der Webseiten im Internet ist — und diese Zahl ist enorm groß.

Um den Page-Rank aller Webseiten zu bestimmen, benötigen wir daher ein extrem effizientes Verfahren zum Lösen dieses linearen Gleichungssystems.

Im Rest des Kapitels werden wir sehen, dass die Theorie der Markov-Ketten uns hilft, diese Probleme zu lösen. Dazu ist die im folgenden Abschnitt dargestellte Sichtweise auf den Page-Rank sehr hilfreich.

Abschnitt 1.4:

Der Zufalls-Surfer

Der Zufalls-Surfer:

Wir nehmen an, dass der Webgraph durch einen gerichteten Graphen $G = (V, E)$ mit Knotenmenge $V = \{1, \ldots, n\}$ repräsentiert wird, der keine Senke besitzt. Des Weiteren sei d eine beliebige reelle Zahl mit $0 \leqslant d \leqslant 1$.

Wir betrachten einen Zufalls-Surfer (englisch: random surfer), der auf einer beliebigen Webseite beginnt und beliebige Links verfolgt, ohne dabei auf Inhalte zu achten.

Wenn der Zufalls-Surfer auf einer Webseite i ist, so wählt er

- mit Wahrscheinlichkeit d einen Link, der von Seite i ausgeht.
  Hierbei wird dann jeder der $a_i = \text{Aus-Grad}_G(i)$ ausgehenden Links mit derselben Wahrscheinlichkeit $\frac{d}{a_i}$ ausgewählt.
- mit Wahrscheinlichkeit $(1 - d)$ eine beliebige Webseite im Web-Graphen.
  Hierbei wird dann jede der n Webseiten mit derselben Wahrscheinlichkeit $\frac{1-d}{n}$ ausgewählt.

Für alle $i, j \in V$ gibt daher

$$p_{i,j} := \begin{cases} \frac{1-d}{n} + \frac{d}{a_i} & \text{, falls } (i,j) \in E \\ \frac{1-d}{n} & \text{, falls } (i,j) \notin E \end{cases} \tag{3}$$

die Wahrscheinlichkeit an, mit der der Zufalls-Surfer in einem Schritt von Seite i zu Seite j wechselt.

Diese Wahrscheinlichkeiten, mit denen sich der Zufalls-Surfer von Knoten zu Knoten bewegt, lassen sich kompakt durch die folgende Matrix darstellen.

Die Page-Rank-Matrix $P(G, d)$

Definition 1.3 (Die Page-Rank-Matrix $P(G, d)$)

Sei $d \in \mathbb{R}$ mit $0 \leqslant d \leqslant 1$, sei $n \in \mathbb{N}_{\geqslant 1}$ und sei $G = (V, E)$ mit $V = \{1, \ldots, n\}$ ein gerichteter Graph ohne Senke. Für jedes $i \in V$ sei $a_i := \text{Aus-Grad}_G(i)$.

Die Page-Rank-Matrix ist die $n \times n$-Matrix

$$P(G, d) := \begin{pmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & & \vdots \\ p_{n,1} & \cdots & p_{n,n} \end{pmatrix},$$

wobei für alle $i, j \in V$ der Eintrag in Zeile i und Spalte j der in Gleichung (3) festgelegte Wert $p_{i,j}$ ist.

Wir schreiben auch kurz $(p_{i,j})_{i,j=1,\ldots,n}$, um die Matrix $P(G, d)$ zu bezeichnen.

Beispiel 1.4

Für den Wert $d = \frac{1}{2}$ und den Graphen G aus Beispiel 1.2 ist beispielsweise $p_{1,1} = \frac{1}{6}$, $p_{1,2} = \frac{1}{6} + \frac{1}{4} = \frac{5}{12}$, $p_{2,3} = \frac{1}{6} + \frac{1}{2} = \frac{2}{3}$ und insgesamt

$$P(G,d) = \begin{pmatrix} \frac{1}{6} & \frac{5}{12} & \frac{5}{12} \\ \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}.$$

□ Ende Beispiel 1.4

Zur Erinnerung: Vektor-Matrix-Produkt

Um den Zusammenhang zwischen dem Zufalls-Surfer, der Page-Rank-Matrix und Tupeln mit der Page-Rank-Eigenschaft beschreiben zu können, benötigen wir folgende Notation für das Rechnen mit Matrizen.

Zur Erinnerung (Vektor-Matrix-Produkt)

Sei $n \in \mathbb{N}_{\geqslant 1}$, und für alle $i, j \in \{1, \ldots, n\}$ sei $p_{i,j}$ eine reelle Zahl. Sei $P := (p_{i,j})_{i,j=1,\ldots,n}$ die $n \times n$-Matrix, die in Zeile i und Spalte j den Eintrag $p_{i,j}$ hat (für alle $i, j \in \{1, \ldots, n\}$).

Ist $X = (X_1, \ldots, X_n)$ ein Tupel aus n reellen Zahlen, so ist das Vektor-Matrix-Produkt

$$X \cdot P$$

das Tupel $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$, bei dem für jedes $j \in \{1, \ldots, n\}$ gilt:

$$Y_j \;=\; \sum_{i=1}^{n} X_i \cdot p_{i,j}.$$

Zusammenhang zwischen Page-Rank-Matrix und Page-Rank-Eigenschaft

Der folgende Satz beschreibt den genauen Zusammenhang zwischen Zufalls-Surfer, Page-Rank-Matrix und Tupeln mit der Page-Rank-Eigenschaft.

Satz 1.5

Sei $d \in \mathbb{R}$ mit $0 \leqslant d < 1$, sei $n \in \mathbb{N}_{\geqslant 1}$ und sei $G = (V, E)$ ein gerichteter Graph mit $V = \{1, \ldots, n\}$, der keine Senke besitzt. Dann gilt:

(a) *Ist $\mathrm{PR} = (\mathrm{PR}_1, \ldots, \mathrm{PR}_n) \in \mathbb{R}^n$ ein Tupel, das die Page-Rank-Eigenschaft bzgl. d besitzt, so ist $\sum_{i=1}^{n} \mathrm{PR}_i = 1$.*

(b) *Für jedes Tupel $X = (X_1, \ldots, X_n) \in \mathbb{R}^n$ mit $\sum_{i=1}^{n} X_i = 1$ gilt:*
X besitzt die Page-Rank-Eigenschaft bzgl. d $\iff$ $X \cdot P(G, d) = X$.

Beweis: Siehe Tafel!

Beachte

Für den Beweis von Satz 1.5 (a) ist wichtig, dass $d \neq 1$ ist und dass G keine Senke besitzt.

Linke Eigenvektoren zum Eigenwert 1

Notation 1.6 (Eigenvektor)

Ein Vektor $X = (X_1, \ldots, X_n)$ heißt linker Eigenvektor zum Eigenwert 1 der $n \times n$-Matrix P, falls gilt: $X \cdot P = X$ und $X \neq (0, \ldots, 0)$.

Satz 1.5 besagt also, dass ein Tupel $\mathrm{PR} = (\mathrm{PR}_1, \ldots, \mathrm{PR}_n) \in \mathbb{R}^n$ genau dann die Page-Rank-Eigenschaft bzgl. d besitzt, wenn es ein linker Eigenvektor zum Eigenwert 1 der Matrix $P(G, d)$ ist, für den $\sum_{i=1}^{n} \mathrm{PR}_i = 1$ ist.

Diese Sichtweise auf den Page-Rank sowie die in den folgenden Abschnitten vorgestellte Theorie der Markov-Ketten helfen uns, die beiden Hauptprobleme (eindeutige Lösbarkeit des linearen Gleichungssystems und effiziente Berechnung der Lösung) zu lösen.

Abschnitt 1.5:

Markov-Ketten

Markov-Ketten und stochastische Matrizen

Markov-Ketten sind nach dem russischen Mathematiker Andrei A. Markov (1856–1922) benannt. In der Literatur werden unterschiedliche Schreibweisen des Namens verwendet, z.B. Markov, Markow oder Markoff.

Definition 1.7 (Markov-Kette)

Eine (homogene) Markov-Kette mit Übergangsmatrix P wird durch eine $n \times n$-Matrix

$$P = \left(p_{i,j}\right)_{i,j=1,\ldots,n}$$

mit $n \in \mathbb{N}_{\geqslant 1}$ beschrieben, für die gilt:

(1) $p_{i,j} \geqslant 0$ für alle $i, j \in \{1, \ldots, n\}$, und

(2) für jede Zeile $i \in \{1, \ldots, n\}$ gilt: $\sum_{j=1}^{n} p_{i,j} = 1$.

Eine Matrix P, die die Eigenschaften (1) und (2) besitzt, wird auch stochastische Matrix genannt.

Der zu P gehörende Graph

Der zu einer stochastischen Matrix P gehörende Graph ist der gerichtete Graph mit Knotenmenge $V = \{1, \ldots, n\}$, so dass für alle $i, j \in \{1, \ldots n\}$ gilt:
Es gibt in G genau dann eine Kante von i nach j, wenn $p_{i,j} > 0$ ist.

Den Eintrag $p_{i,j}$ in Zeile i und Spalte j von P kann man als Wahrscheinlichkeit dafür auffassen, dass ein Zufalls-Surfer im Graphen G in einem Schritt von Knoten i zu Knoten j springt.

Beispiel 1.8

Sei $G = (V, E)$ ein beliebiger gerichteter Graph mit Knotenmenge $V = \{1, \ldots, n\}$ (für $n := |V| \in \mathbb{N}_{\geqslant 1}$), der keine Senke besitzt.
Sei d eine reelle Zahl mit $0 \leqslant d < 1$ und sei $P := P(G, d)$ die zugehörige Page-Rank-Matrix.

Gemäß der Definition von $P(G, d)$ ist $p_{i,j} > 0$ für alle $i, j \in \{1, \ldots, n\}$ (dazu beachte man, dass $0 \leqslant d < 1$ ist).

Außerdem gilt für jede Zeile $i \in \{1, \ldots, n\}$, dass

$$\sum_{j=1}^{n} p_{i,j} \quad = \quad \sum_{j=1}^{n} \frac{1-d}{n} \quad + \sum_{j\,:\,(i,j)\in E} \frac{d}{a_i} \quad \overset{G \text{ ohne Senke}}{=} \quad (1-d) \;+\; a_i \cdot \frac{d}{a_i} \quad = \quad 1.$$

Somit ist P eine stochastische Matrix und beschreibt daher eine Markov-Kette.

Für jedes $i, j \in \{1, \ldots, n\}$ gibt der Wert $p_{i,j}$ die Wahrscheinlichkeit dafür an, dass der Zufalls-Surfer in einem Schritt von Webseite i zu Webseite j springt.

Da $p_{i,j} > 0$ ist, ist der zu P gehörende Graph der vollständige gerichtete Graph auf n Knoten, d.h. der Graph mit Knotenmenge $V = \{1, \ldots, n\}$ und Kantenmenge $V \times V$. Diesen Graphen bezeichnen wir im Folgenden mit $\vec{K}_n$. □Ende Beispiel 1.8

Abschnitt 1.6:

Die effiziente Berechnung des Page-Rank

Um zu sehen, dass die Theorie der Markov-Ketten uns eine Lösung für die beiden Hauptprobleme (eindeutige Lösbarkeit des linearen Gleichungssystems und effiziente Berechnung der Lösung) zu lösen. liefert, schauen wir uns die Bewegungen des Zufalls-Surfers auf dem Web-Graphen etwas genauer an.

Für unsere Betrachtungen ist der folgendermaßen definierte Begriff einer Verteilung sehr nützlich.

Verteilungen

Definition 1.9

Sei $n \in \mathbb{N}_{\geqslant 1}$.
Eine Verteilung auf $V = \{1, \ldots, n\}$ ist ein Tupel $X = (X_1, \ldots, X_n) \in \mathbb{R}^n$, für das gilt:

(1) für alle $i \in \{1, \ldots, n\}$ ist $X_i \geqslant 0$ und

(2) $\sum_{i=1}^{n} X_i = 1$.

Ist G ein gerichteter Graph mit Knotenmenge $V = \{1, \ldots, n\}$ und ist $X = (X_1, \ldots, X_n)$ eine Verteilung auf V, so fassen wir für jedes $i \in V$ die Zahl X_i als Wahrscheinlichkeit dafür auf, dass ein Zufalls-Surfer in G sich auf Knoten i befindet.

Beobachtung 1.10

Sei $n \in \mathbb{N}_{\geqslant 1}$ und sei $P = (p_{i,j})_{i,j=1,\ldots,n}$ eine stochastische Matrix. Ist $X = (X_1, \ldots, X_n)$ eine Verteilung auf $V := \{1, \ldots, n\}$, so gibt das Tupel $Y = (Y_1, \ldots, Y_n)$ mit

$$X \cdot P \;=\; Y$$

Folgendes an: Wenn wir in dem zu P gehörenden Graphen für jedes $i \in V$ den Zufalls-Surfer mit Wahrscheinlichkeit X_i auf Knoten i beginnen lassen, so gibt für jedes $j \in V$ die Zahl

$$Y_j \;=\; \sum_{i=1}^{n} X_i \cdot p_{i,j}$$

die Wahrscheinlichkeit dafür an, dass der Zufalls-Surfer sich nach einem Schritt auf Knoten j befindet.

Rekursiv können wir so für jedes $k \in \mathbb{N}$ eine Verteilung $X^{(k)} = (X_1^{(k)}, \ldots, X_n^{(k)})$ angeben, so dass für jedes $j \in V$ der Wert $X_j^{(k)}$ die Wahrscheinlichkeit dafür angibt, dass der Zufalls-Surfer sich nach k Schritten auf Knoten j befindet. Dazu wählen wir

$$X^{(0)} \;:=\; X \qquad \text{und} \qquad X^{(k+1)} \;:=\; X^{(k)} \cdot P\,, \qquad \text{f.a. } k \in \mathbb{N}.$$

Somit gilt für jedes $k \in \mathbb{N}_{\geqslant 1}$: $\quad X^{(k)} \;=\; X^{(0)} \cdot \underbrace{P \cdot P \cdot \ldots \cdot P}_{k \text{ mal}} \;=\; X \cdot P^k.$

Zur Erinnerung: Das Matrix-Produkt

Zur Erinnerung

Das Produkt $A \cdot B$ zweier $n \times n$-Matrizen $A = (a_{i,j})_{i,j=1,\ldots,n}$ und $B = (b_{i,j})_{i,j=1,\ldots,n}$ ist die $n \times n$-Matrix $C = (c_{i,j})_{i,j=1,\ldots,n}$ mit

$$c_{i,j} \quad = \quad \sum_{\ell=1}^{n} a_{i,\ell} \cdot b_{\ell,j}$$

für alle $i, j \in \{1, \ldots, n\}$.

Für jede Zahl $k \in \mathbb{N}_{\geqslant 1}$ ist die $n \times n$-Matrix A^k rekursiv definiert durch

$$A^1 := A \qquad \text{und} \qquad A^{k+1} := A \cdot A^k \quad \text{(für alle } k \in \mathbb{N}\text{)}.$$

Wir schreiben $(A^k)_{i,j}$, um den Eintrag in Zeile i und Spalte j der Matrix A^k zu bezeichnen.

Man sieht leicht, dass Folgendes gilt:

Beobachtung 1.11

Ist $n \in \mathbb{N}_{\geqslant 1}$ und ist $P = (p_{i,j})_{i,j=1,\ldots,n}$ eine stochastische Matrix, so können wir für jedes $k \in \mathbb{N}_{\geqslant 1}$ den Eintrag $(P^k)_{i,j}$ in Zeile i und Spalte j der Matrix P^k als die Wahrscheinlichkeit dafür auffassen, dass der Zufalls-Surfer auf dem zu P gehörenden Graphen innerhalb von genau k Schritten von Knoten i zu Knoten j gelangt.

Insbesondere gilt auch für jede Zeile i, dass $\sum_{j=1}^{n}(P^k)_{i,j} = 1$.

Wegen

$$P^{k+1} \quad = \quad P^k \cdot P \quad = \quad P \cdot P^k$$

gilt für jedes $k \in \mathbb{N}_{\geqslant 1}$:

$$(P^{k+1})_{i,j} \quad = \quad \sum_{\ell=1}^{n}(P^k)_{i,\ell} \cdot p_{\ell,j} \quad = \quad \sum_{\ell=1}^{n} p_{i,\ell} \cdot (P^k)_{\ell,j}\,.$$

Zur effizienten Berechnung des Page-Ranks machen wir uns zunutze, dass die durch die Page-Rank-Matrix $P(G, d)$ (für $0 \leqslant d < 1$) beschriebene Markov-Kette die folgende Eigenschaft hat.

Der Beweis, dass die Matrix $P(G, d)$ tatsächlich diese Eigenschaft hat, wird am Ende dieses Abschnitts durch Satz 1.14 geliefert.

Ergodische Markov-Ketten

Definition 1.12 (Ergodische Markov-Ketten)

Sei $n \in \mathbb{N}_{\geqslant 1}$ und sei $P = (p_{i,j})_{i,j=1,\ldots,n}$ eine stochastische Matrix.

Die durch P beschriebene Markov-Kette heißt ergodisch, wenn für alle Zeilen $i, i' \in \{1, \ldots, n\}$ und alle Spalten $j \in \{1, \ldots, n\}$ gilt:

Die Grenzwerte

$$\lim_{k\to\infty} (P^k)_{i,j} \qquad \text{und} \qquad \lim_{k\to\infty} (P^k)_{i',j}$$

existieren und es gilt

$$\lim_{k\to\infty} (P^k)_{i,j} \quad = \quad \lim_{k\to\infty} (P^k)_{i',j} \quad > \quad 0.$$

Eigenschaften ergodischer Markov-Ketten

Beobachtung 1.13 (Eigenschaften ergodischer Markov-Ketten)

Ist P eine stochastische Matrix, die eine ergodische Markov-Kette beschreibt, so gilt offensichtlicherweise Folgendes:

(1) *Die Matrix*

$$P' := \left(\lim_{k \to \infty} \left(P^k\right)_{i,j} \right)_{i,j=1,\ldots,n} \tag{4}$$

ist wohldefiniert (da die Grenzwerte existieren), alle Einträge sind > 0 (da die Grenzwerte alle > 0 sind), und

(2) *alle Zeilen von P' sind identisch.*

Wir schreiben $p' := (p'_1, \ldots, p'_n)$, um die erste Zeile von P' zu bezeichnen.

Die Matrix P' sieht daher folgendermaßen aus:

$$P' \quad = \quad \begin{pmatrix} p' \\ p' \\ \vdots \\ p' \end{pmatrix} \quad = \quad \begin{pmatrix} p'_1 & \dots & p'_n \\ p'_1 & \dots & p'_n \\ & \vdots & \\ p'_1 & \dots & p'_n \end{pmatrix} .$$

Wegen Gleichung (4) ist $P' = \lim_{k\to\infty} \left(P^k\right)$.

Somit ist $P' \cdot P = P'$, und daher gilt insbesondere

$$p' \cdot P \;=\; p' \,,$$

d.h. p' ist ein linker Eigenvektor zum Eigenwert 1 der Matrix P.

Da für jedes $k \in \mathbb{N}_{\geqslant 1}$ und jede Zeile $i \in \{1, \dots, n\}$ die Summe aller Einträge der i-ten Zeile von P^k gleich 1 ist und die Grenzwertbildung mit der Bildung endlicher Summen vertauscht werden kann, ist auch die Summe aller Einträge der i-ten Zeile von P' gleich 1.

Daher ist $\sum_{i=1}^{n} p'_i = 1$, d.h. p' ist eine Verteilung.

Notation

Eine Verteilung Y mit $Y \cdot P = Y$ wird auch stationäre Verteilung für P genannt.

Für jede beliebige Verteilung $X = (X_1, \ldots, X_n)$ gilt:

$$X \cdot P' \;=\; p', \tag{5}$$

denn für jedes $j \in V$ ist der j-te Eintrag im Tupel $X \cdot P'$ gerade die Zahl $\sum_{i=1}^{n} X_i \cdot p'_j \;=\; p'_j \cdot \sum_{i=1}^{n} X_i \;=\; p'_j$.

Daher gilt:

(a) $p' = (p'_1, \ldots, p'_n)$ ist die **einzige** stationäre Verteilung, die P besitzt, und

(b) wenn der Zufalls-Surfer im zu P gehörenden Graphen seinen Startknoten gemäß einer beliebigen Anfangsverteilung $X = (X_1, \ldots, X_n)$ wählt und hinreichend viele Schritte macht, so ist für jedes $j \in V$ die Wahrscheinlichkeit, bei Knoten j zu landen, beliebig nah bei p'_j.

Die Wahl des Anfangsknotens ist für einen Zufalls-Surfer, der hinreichend lange surft, also egal.

Auf Grund der Gleichungen (4) und (5) erhalten wir:

$$p' \overset{(5)}{=} X \cdot P' \overset{(4)}{=} X \cdot \lim_{k\to\infty} P^k = \lim_{k\to\infty} \left(X \cdot P^k\right) = \lim_{k\to\infty} X^{(k)},$$

wobei $X^{(0)} := X$ und $X^{(k+1)} := X^{(k)} \cdot P$, f.a. $k \in \mathbb{N}$.

Um eine Näherung für das Tupel p' zu berechnen, können wir daher wie folgt vorgehen:

- Wir starten mit einer beliebigen Verteilung $X^{(0)} = X$ (etwa der Gleichverteilung $X = (\frac{1}{n}, \cdots, \frac{1}{n})$)
- und berechnen nacheinander für $k = 1, 2, 3$ usw. das Tupel $X^{(k+1)} := X^{(k)} \cdot P$.
- Dieser Prozess wird beendet, sobald das Tupel $X^{(k+1)}$ sich nicht mehr viel vom Tupel $X^{(k)}$ unterscheidet, d.h. sobald für jedes $j \in \{1, \ldots, n\}$ die Zahl $|X_j^{(k+1)} - X_j^{(k)}|$ kleiner als eine vorher festgelegte Schranke ε ist (wobei $X_j^{(k+1)}$ und $X_j^{(k)}$ der Eintrag in der j-ten Komponente von $X^{(k+1)}$ bzw. $X^{(k)}$ ist).

□Beobachtung 1.13

Um diese Vorgehensweise für die Berechung des Page-Rank benutzen zu können, benötigen wir noch folgenden Satz.

Satz 1.14

Ist $P = (p_{i,j})_{i,j=1,\ldots,n}$ eine stochastische Matrix mit $p_{i,j} > 0$ f.a. $i,j \in \{1,\ldots,n\}$, so ist die durch P beschriebene Markov-Kette ergodisch.

Beweis: Für jedes $k \in \mathbb{N}_{\geqslant 1}$ und jede Spalte $j \in \{1,\ldots,n\}$ sei

$$m_j^{(k)} := \min_{i \in \{1,\ldots,n\}} (P^k)_{i,j}$$

der kleinste Eintrag der j-ten Spalte von P^k, und sei

$$M_j^{(k)} := \max_{i \in \{1,\ldots,n\}} (P^k)_{i,j}$$

der größte Eintrag der j-ten Spalte von P^k.

Behauptung 1: Für alle $k \in \mathbb{N}_{\geqslant 1}$ und alle $j \in \{1,\ldots,n\}$ gilt

$$m_j^{(k)} \leqslant m_j^{(k+1)} \qquad \text{und} \qquad M_j^{(k)} \geqslant M_j^{(k+1)}.$$

Gemäß Voraussetzung ist $p_{i,j} > 0$ und $p_{i,j} \leqslant 1$ für alle $i, j \in \{1, \ldots, n\}$.
Somit ist

$$a \;:=\; \min_{i,j \in \{1,\ldots,n\}} p_{i,j} \;>\; 0.$$

Gemäß Behauptung 1 gilt für jedes $j \in \{1, \ldots, n\}$:

$$0 \;<\; a \;\leqslant\; m_j^{(1)} \;\leqslant\; m_j^{(2)} \;\leqslant\; m_j^{(3)} \;\leqslant\; \cdots \tag{6}$$

und

$$1 \;\geqslant\; M_j^{(1)} \;\geqslant\; M_j^{(2)} \;\geqslant\; M_j^{(3)} \;\geqslant\; \cdots\,. \tag{7}$$

Behauptung 2: Für jedes $j \in \{1, \ldots, n\}$ ist $\lim_{k \to \infty} \left(M_j^{(k)} - m_j^{(k)}\right) \;=\; 0.$

Bevor wir Behauptung 2 beweisen, schließen wir zunächst den Beweis von Satz 1.14 ab.

Siehe Tafel!

Um den Beweis von Satz 1.14 abzuschließen, müssen wir nur noch Behauptung 2 beweisen.

Die folgende Behauptung 3 liefert uns den Schlüssel zum Beweis von Behauptung 2.

Behauptung 3: Sei $a := \min_{i,j \in \{1,\ldots,n\}} p_{i,j}$. Seien $i_1, i_2 \in \{1,\ldots,n\}$. Sei

$$I_1 := \left\{ \ell \in \{1,\ldots,n\} : p_{i_1,\ell} \geqslant p_{i_2,\ell} \right\} \quad \text{und} \quad I_2 := \{1,\ldots,n\} \setminus I_1,$$

d.h. I_1 ist die Menge aller Spalten, bei denen der Eintrag in Zeile i_1 größer oder gleich dem Eintrag in Zeile i_2 ist, und I_2 ist die Menge aller Spalten, bei denen der Eintrag in Zeile i_1 echt kleiner als der Eintrag in Zeile i_2 ist. Dann gilt:

(a) $\displaystyle\sum_{\ell \in I_1} \left(p_{i_1,\ell} - p_{i_2,\ell}\right) \leqslant 1 - na.$

(b) $\displaystyle\sum_{\ell \in I_2} \left(p_{i_1,\ell} - p_{i_2,\ell}\right) = -\sum_{\ell \in I_1} \left(p_{i_1,\ell} - p_{i_2,\ell}\right).$

(c) $(P^{k+1})_{i_1,j} - (P^{k+1})_{i_2,j} \leqslant (1-na) \cdot \left(M_j^{(k)} - m_j^{(k)}\right)$, für alle $k \in \mathbb{N}_{\geqslant 1}$ und $j \in \{1,\ldots,n\}$.

Unter Verwendung von Teil (c) von Behauptung 3 können wir nun Behauptung 2 beweisen und damit den Beweis von Satz 1.14 abschließen.

Lösung der beiden Hauptprobleme

Folgerung 1.15 (Lösung der beiden Hauptprobleme)

Sei $P := P(G, d)$ die Page-Rank-Matrix für einen Dämpfungsfaktor d mit $0 \leqslant d < 1$ und einen gerichteten Graphen $G = (V, E)$ ohne Senke.

Wegen $d \neq 1$ sind alle Einträge in P echt größer als 0.

Mit Satz 1.14 erhalten wir, dass P ergodisch ist.

Aus Beobachtung 1.13 folgt, dass die stationäre Verteilung p' von P das eindeutig festgelegte Tupel ist, das die Page-Rank-Eigenschaft bzgl. d besitzt.

Das am Ende von Beobachtung 1.13 beschriebene Vorgehen liefert ein Verfahren, um eine Näherung für das Tupel p' zu berechnen:

- *Sei $P := P(G, d)$ die Page-Rank-Matrix für den Dämpfungsfaktor $d := 0,85$, wobei G der Senken-freie Graph ist, der aus dem Web-Graphen durch wiederholtes Löschen von Senken entsteht.*
- *Starte mit einer beliebigen Verteilung $X^{(0)} = (X_1, \ldots, X_n)$ (z.B. $X^{(0)} := (\frac{1}{n}, \ldots, \frac{1}{n})$).*
- *Für $k := 1, 2, \ldots$ berechne $X^{(k+1)} := X^{(k)} \cdot P$.*

Aus der Theorie der Markov-Ketten und den speziellen Eigenschaften der Page-Rank-Matrix ergibt sich, dass auf Grund des hohen Zusammenhangs des Web-Graphen die Folge der Tupel $X^{(k)}$ für $k = 0, 1, 2, \ldots$ sehr schnell gegen die stationäre Verteilung p' konvergiert.

Laut Kapitel 5 des Buchs „Mining of Massive Datasets“ von Leskovec, Rajamaran und Ullman (2014) reichen in der Praxis i.d.R. 50–75 Iterationen aus, so dass die Einträge des Vektors $X^{(k)}$ für $k = 75$ eine hinreichend gute Näherung für die Page-Ranks der einzelnen Webseiten sind.

Abschnitt 1.7:

Praktische Aspekte der Berechnung des Page-Ranks

In der Praxis werden mehrere Tausend PCs eingesetzt, die mehrere Stunden zur Berechnung des Page-Ranks benötigen — was in Anbetracht der Tatsache, dass es mehrere Milliarden Webseiten gibt, erstaunlich gering ist.

Kompakte Speicherung der Page-Rank-Matrix

Für den Web-Graphen müssen wir davon ausgehen, dass die Anzahl n der Knoten extrem groß ist ($n \geqslant 10^{12}$), so dass es nicht ratsam ist, alle n^2 Einträge der Page-Rank-Matrix $P := P(G, d)$ abzuspeichern. Zur kompakten Speicherung von P wird ausgenutzt, dass P viele identische Einträge der Form $\frac{1-d}{n}$ hat. Jede Zeile i von P sieht wie folgt aus:

- In jeder Spalte j mit $(i,j) \notin E$ steht der Wert $\frac{1-d}{n}$.
- In jeder Spalte j mit $(i,j) \in E$ steht der Wert $p_{i,j} = \frac{1-d}{n} + \frac{d}{a_i}$. Die Anzahl dieser Spalten j ist i.d.R. sehr klein (in der Größenordung 10–50), da jede einzelne Webseite i i.d.R. nur recht wenige Links enthält.

Eine kompakte Repräsentation, aus der wir für gegebenes i und j die Zahl $p_{i,j}$ leicht ausrechnen können, speichert folgende Werte:[1]

- Die Werte d und $\frac{1-d}{n}$, und
- für jede Zeile $i \in \{1, \ldots, n\}$
  - (i.1): den Aus-Grad a_i des Knotens i und
  - (i.2): die Liste aller Knoten j mit $(i,j) \in E$.

Der dafür benötigte Speicherplatz ist

$$O(1) + \sum_{i=1}^{n} \big(O(1) + O(a_i)\big) \quad = \quad O(|V| + |E|) \quad = \quad O(|G|)$$

Für den Web-Graphen G können wir davon ausgehen, dass die Knoten einen recht geringen Aus-Grad haben (in der Größenordnung 10 bis 50) und dass daher $|G| = O(|V|)$ ist, wobei der durch die O-Notation unterdrückte konstante Faktor ebenfalls in der Größenordnung 10 bis 50 liegt.

[1] Diese Repräsentation ist eine um die Werte d, $\frac{1-d}{n}$ und a_i angereichterte Adjazenzliste des Web-Graphen G.

Parallelisierte Berechnung des Vektor-Matrix-Produkts

In jedem der $k \approx 75$ Iterationsschritte der Page-Rank-Berechnung muss für die Page-Rank-Matrix P und den Vektor $X^{(k)}$ das Vektor-Matrix-Produkt $X^{(k+1)} := X^{(k)} \cdot P$ berechnet werden. Wir betrachten nun, wie man einen einzelnen Iterationsschritt auf einem Rechnercluster durchführen kann und schreiben dabei $Y = (y_1, \ldots, y_n)$, um den Vektor $X^{(k)}$ zu bezeichnen, und $Y' = (y'_1, \ldots, y'_n)$ um den gesuchten Vektor $X^{(k+1)} := X^{(k)} \cdot P$ zu bezeichnen.

Erster Ansatz:

Wir gehen davon aus, dass für eine Zahl $L \geqslant 1$ im Rechnercluster L Rechner $R_1, \ldots, R_L$ zur Verfügung stehen. Die Matrix P teilen wir auf in L vertikale Streifen $P_1, \ldots, P_L$, so dass der Streifen P_1 die ersten $\frac{n}{L}$ Spalten von P enthält, der Streifen P_2 die nächsten $\frac{n}{L}$ Spalten von P enthält usw. Somit ist

$$P \quad = \quad \Big(P_1 \; P_2 \; \cdots \; P_L \Big)$$

wobei P_s für jedes $s \in \{1, \ldots, L\}$ eine $(n \times \frac{n}{L})$-Matrix ist.
Wir verteilen nun die Page-Rank-Matrix P und den Eingabe-Vektor Y so im Rechnercluster, dass für jedes $s \in \{1, \ldots, \ell\}$ der Rechner R_s den Streifen P_s und den gesamten Vektor Y zur Verfügung hat.
Der Rechner R_s berechnet dann das Vektor-Matrix-Produkt $Z_s := Y \cdot P_s$.

Dann gilt: Für jedes $s \in \{1, \ldots, L\}$ ist Z_s ein Vektor der Länge $\frac{n}{L}$; und der gesuchte Vektor $Y' = Y \cdot P$ ist gerade der Vektor $(Z_1, Z_2, \ldots, Z_L)$.

Nachteil: In der dargestellten kompakten Repräsentation von P werden die Einträge der Page-Rank-Matrix allerdings *zeilenweise* gespeichert. Durch das Bilden von vertikalen Streifen der Matrix P können die Vorteile der kompakten Speicherung zu Nichte gemacht werden (im Extremfall besteht jeder Streifen aus genau einer Spalte von P).

Zweiter Ansatz:

Wir gehen davon aus, dass für eine Zahl $L = \ell^2$ im Rechencluster L Rechner $R_{s,t}$ für $s, t \in \{1, \ldots, \ell\}$ zur Verfügung stehen. Die Matrix P teilen wir auf in ℓ^2 Blöcke $B_{s,t}$, so dass $B_{s,t}$ für jedes $s, t \in \{1, \ldots, \ell\}$ eine $(\frac{n}{\ell} \times \frac{n}{\ell})$-Matrix ist. Somit ist

$$P = \begin{pmatrix} B_{1,1} & B_{1,2} & B_{1,3} & \cdots & B_{1,\ell} \\ B_{2,1} & B_{2,2} & B_{2,3} & \cdots & B_{2,\ell} \\ B_{3,1} & B_{3,2} & B_{3,3} & \cdots & B_{3,\ell} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ B_{\ell,1} & B_{\ell,2} & B_{\ell,3} & \cdots & B_{\ell,\ell} \end{pmatrix}$$

Den Eingabevektor Y teilen wir auf in horizontale Streifen $Y_1, \ldots, Y_\ell$, von denen jeder die Länge $\frac{n}{\ell}$ hat, d.h.

$$Y = \begin{pmatrix} Y_1 & Y_3 & Y_3 & \cdots & Y_\ell \end{pmatrix}$$

Wir verteilen nun die Page-Rank-Matrix P und den Eingabe-Vektor Y so im Rechencluster, dass für alle $s, t \in \{1, \ldots, \ell\}$ der Rechner $R_{s,t}$ den Block $B_{s,t}$ sowie den Streifen Y_s erhält.

Der Rechner $R_{s,t}$ berechnet dann das Vektor-Matrix-Produkt $Z_{s,t} := Y_s \cdot B_{s,t}$ und schickt das Ergebnis $Z_{s,t}$ an den Rechner $R_{t,t}$ (insbesondere ist $Z_{s,t}$ ein Zeilenvektor der Länge $\frac{n}{\ell}$).
Der Rechner $R_{t,t}$ nimmt die Zwischenergebnisse $Z_{1,t}$, $Z_{2,t}$, ..., $Z_{\ell,t}$ entgegen und berechnet deren Summe

$$Z_t \quad := \quad Z_{1,t} + Z_{2,t} + Z_{3,t} + \cdots + Z_{\ell,t}$$

Dann gilt: Für jedes $t \in \{1, \ldots, \ell\}$ ist Z_t ein Vektor der Länge $\frac{n}{\ell}$; und der gesuchte Vektor $Y' = Y \cdot P$ ist gerade der Vektor $(Z_1, Z_2, \ldots, Z_\ell)$.

Im nächsten Iterationsschritt spielt dieser Vektor Y' die Rolle, die bisher der Vektor Y gespielt hat. Um die einzelnen Streifen von Y' an die richtigen Rechner zu verteilen, schickt (für jedes $t \in \{1, \ldots, \ell\}$) Rechner $R_{t,t}$ den Vektor Z_t (der im nächsten Iterationsschritt als Y_t fungiert) direkt an die Rechner $R_{t,1}$, $R_{t,2}$, $R_{t,3}$, $\ldots$, $R_{t,\ell}$.

Vorteil: Jeder Block $B_{s,t}$ der Page-Rank-Matrix kann relativ kompakt gespeichert werden, indem für jede Zeile i von $B_{s,t}$ unter (i.2) nur diejenigen Knoten j mit $(i,j) \in E$ aufgelistet werden, die die Spalten von $B_{s,t}$ betreffen.

Abschnitt 1.8:

Literaturhinweise