Kapitel 3
Information erschließen

Im Februar 2011 veröffentlichten MARTIN HILBERT von der University of Southern California und PRISCILA LÓPEZ von der Open University of Catalonia eine Arbeit mit dem Titel *The World's Technological Capacity to Store, Communicate, and Compute Information*. Darin schätzen sie, daß die Menschheit im Jahre 2007 bei optimaler Datenkomprimierung $2{,}9 \cdot 19^{20}$ Byte Information speichern konnte und daß die Summe aller kommunizierten Information in diesem Jahr sogar bei $2 \cdot 10^{21}$ Byte lag. Unter der gespeicherten Information befinden sich natürlich auch viele Musikstücke auf privaten mp3-Playern und Filme auf privaten DVDs, aber auch die allgemein zugängliche Information liegt weit über dem, was ein Einzelner überblicken kann. Spätestens seit der Jahrtausendwende ist allein das World Wide Web so groß geworden, daß keine Suchmaschine mehr seinen Inhalt von einer Redaktion aus menschlichen Spezialisten ordnen lassen kann; benötigt werden Algorithmen, mit denen dies Computer automatisch erledigen können. Da die verfügbare Information zumindest bislang ungefähr im gleichen Tempo anstieg wie die Rechenkraft pro Euro der jeweils aktuellen Computer, hat ein solcher Ansatz Chancen, auch langfristig durchführbar zu bleiben.

Künstliche Intelligenz, Computerlinguistik und ähnliche Forschungsgebiete sind allerdings noch weit davon entfernt, einen Computer allgemeine Texte „verstehen" zu lassen; lediglich bei experimentellen Systemen mit sehr reduziertem Vokabular können Computer ein gewisses Textverständnis simulieren.

Reale Systeme für praktische Anwendungen müssen daher mit ziemlich groben und einfachen Methoden arbeiten; perfekte Ergebnisse kann man

unter diesen Umständen zwar nicht erwarten, aber – wie der Erfolg von Suchmaschinen wie Google zeigt – sind die Resultate doch erstaunlich gut.

§ 1: Vektorraummodelle

Das erste funktionierende System zur Informationssuche in Textdatenbanken hieß *Smart;* es wurde zwischen 1962 und 1965 unter Leitung von GERARD SALTON an der Harvard University entwickelt. Damals arbeitete es im wesentlichen mit reiner Textsuche; später an der Cornell University entwickelten SALTON und seine Mitarbeiter wesentlich feinere Methoden. Insbesondere verwendeten sie ab Anfang der Siebzigerjahre zunehmend Methoden aus der Linearen Algebra.

Grundlage für den Einsatz entsprechender Algorithmen ist die Term-Dokument-Matrix der Dokumentsammlung: Wir betrachten eine gewisse Menge von Begriffen; bei Fachdatenbanken kann es sich dabei um eine vordefinierte Liste von Stichworten handeln, bei Internetsuchmaschinen aber auch um die Menge aller möglicher Wörter einer Sprache (etwa dreißig Tausend) und eventuell auch noch Falschschreibungen, Eigennamen und so weiter.

Gerade bei Internetsuchmaschinen wird vorher oft auch noch das sogenannte *stemming* praktiziert: Als mögliche Terme gelten nicht die Wörter, sondern die Wortstämme, so daß Flektionsendungen, Verwendung als Substantiv, Adjektiv oder Verb keine Rolle spielen: Beispielsweise werden Information, Informationen, informieren, informierte, informativ *usw.* als *ein* einziger Suchbegriff behandelt.

Manche Suchbegriffe wie Artikel oder häufige Präpositionen sind so unspezifisch, daß sie kaum zum Auffinden geeigneter Dokumente beitragen können; diese werden oft auf eine *Stopliste* gesetzt und zumindest bei Suchanfragen, die auch noch andere Begriffe enthalten, nicht berücksichtigt. (Google findet allerdings auch auf die Anfrage „die" noch Seiten; ganz unter den Tisch fallen sie also zumindest dort nicht.) Was auf die Stopliste kommt, hängt natürlich von der Art der Anwendung ab; einer der Pioniere der automatischen Textsuche, die Firma Boeing,

erschließt ihren Serviceingenieuren die sämtlichen Handbücher durch eine Suchmaschine, die beispielsweise das Wort „Flugzeug“ auf ihrer Stopliste hat – die Firma stellt schließlich keine Rasenmäher her.

Zur Menge aller verbliebener Begriffe wird üblicherweise zunächst ein sogenannter *invertierter Index* gebildet, d.h. für jeden Begriff wird die Liste aller Dokumente zusammengestellt, in denen er vorkommt, gegebenenfalls mit Zusatzinformationen wo und wie oft. Dieser Index wird von sogenannten *Crawlern* zusammengestellt, die periodisch das *world wide web* nach Dokumenten durchsuchen.

Die Term-Dokument-Matrix hat für jeden möglichen Suchbegriff eine Zeile und für jedes Dokument eine Spalte; der Eintrag in der i-ten Zeile und j-ten Spalte gibt an, wie wichtig der i-te Begriff für das j-te Dokument ist.

Im einfachsten Fall sind alle Einträge entweder 0 oder 1, je nachdem, ob der Begriff vorkommt oder nicht; es gibt aber eine ganze Reihe weiterer Strategien, von denen wir noch einige betrachten werden. Oft werden auch die Spalten auf Länge eins normiert; der Grund dafür wird gleich klar werden. In jedem Fall ist die Matrix *spärlich besetzt*, d.h. nur ein Bruchteil der Einträge ist von null verschieden.

Eine Suchanfrage kann als Vektor betrachtet werden, der für jeden möglichen Suchbegriff einen Eintrag hat; die nicht verschwindenden Einträge geben an, welche Begriffe, gegebenenfalls mit welcher Wichtigkeit, in der Anfrage vorkommen.

Suchanfrage und Dokumente werden also dargestellt durch Vektoren aus ein und demselben Vektorraum; ein Dokument paßt umso besser zur Anfrage, je ähnlicher die beiden Vektoren zueinander sind.

Um die „Ähnlichkeit“ zweier Vektoren in diesem Zusammenhang zu definieren, betrachten wir ein Beispiel:

Wir haben vier Dokumente und die drei Suchbegriffe *Jean*, *Paul* und *Sartre*. Im ersten Dokument kommen *Jean* und *Paul* je zweimal vor, im zweiten *Jean* zweimal und *Paul* dreimal; von *Sartre* ist in beiden Dokumenten nicht die Rede. Im dritten Dokument kommen alle drei Begriffe je zweimal vor, im vierten schließlich *Jean* und *Paul* je zweimal,

Sartre dreimal. Die Suchanfrage sei *Jean Paul;* wir interessieren uns also für den deutschen Schriftsteller JOHANN PAUL FRIEDRICH RICHTER (1763–1825), der unter dem Pseudonym JEAN PAUL publizierte.

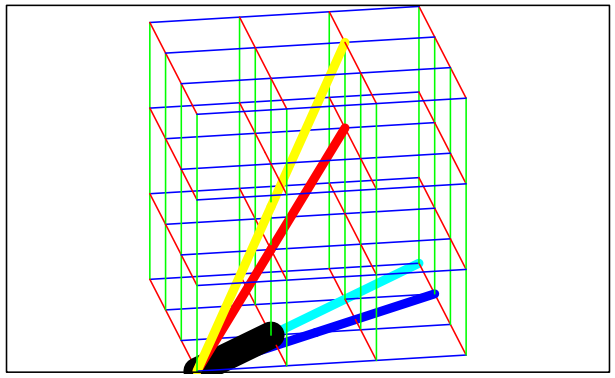
Es ist ziemlich klar, daß wohl nur die ersten beiden Dokumente für diese Anfrage relevant sind; die letzten beiden dürften von dem französischen Philosophen und Schriftsteller JEAN PAUL SARTRE (1905–1980) handeln, nach dem man wohl eher nicht mit der Anfrage *Jean Paul* suchen dürfte.

Die heute gebräuchlichen Suchmaschinen erkennen dies und liefern in erster Linie Dokumente über JEAN PAUL; es gibt aber immer noch online Buchhandlungen (bei meinem Test im Mai 2011 etwa libri.de), die bei der Suche nach JEAN PAUL hauptsächlich Bücher von JEAN PAUL SARTRE finden.

Um zu sehen, wie sich das vermeiden läßt, stellen wir zunächst die Term-Dokument-Matrix auf, wobei wir hier als *Wichtigkeit* einfach die Anzahl der Vorkommen nehmen:

	1	2	3	4
Jean	2	2	2	2
Paul	2	3	2	2
Sartre	0	0	2	3

Die Suchanfrage entspricht dem Spaltenvektor zu (1, 1, 0).



Die Abbildung zeigt die vier Spaltenvektoren der Dokumente in deren jeweiligen Farben sowie den schwarzen Vektor der Suchanfrage. Natürlich sind die Dokumentenvektoren allesamt deutlich länger als der Fragevektor, aber wie man sieht, unterscheiden sich die *Richtungen* der ersten beiden Dokumentenvektoren gar nicht oder kaum von der des Anfragevektors, wohingegen der dritte und der vierte deutlich andere Richtungen haben.

Somit bietet sich als eine einfache Strategie zum Vergleich zwischen Anfrage- und Dokumentenvektoren an, die Berechnung des Winkels an. Da es uns nicht wirklich auf die genauen Werte der Winkel ankommt, sondern nur darauf, ob sie nahe beim Nullwinkel liegen, können wir stattdessen auch die einfacher zu berechnenden Kosinuswerte nehmen und ein Dokument dann als relevant im Sinne der Suchanfrage betrachten, wenn dieser Kosinus hinreichend nahe bei eins liegt. Sofern wir alle Spaltenvektoren der Term-Dokument-Matrix sowie auch den Vektor der Suchanfrage auf Länge eins normieren, können wir diesen Kosinuswert einfach als Skalarprodukt der beiden Vektoren berechnen. Im Falle einer Suchmaschine handelt es sich dabei zwar um Vektoren in einem Vektorraum, dessen Dimension bei rund dreißig Tausend liegen dürfte; da aber kaum eine Suchanfrage mehr als drei Terme enthält, müssen wir tatsächlich nur wenige Produkte berechnen und aufaddieren.

§ 2: Glätten durch orthogonale Projektion

In der Term-Dokument-Matrix ist jedes Dokument repräsentiert durch einen Vektor, der angibt, mit welchem Gewicht welche Terme im Dokument vorkommen. Offensichtlich steckt in der Wahl dieses Vektors viel Willkür, und auch wenn man nach einem einheitlichen Verfahren arbeitet, können inhaltlich sehr ähnlichen Dokumenten recht unterschiedliche Vektoren zugeordnet werden. Hinzu kommt, daß Suchanfragen zwangsläufig zu sehr einfachen Vektoren führen, die in ein sehr viel gröberes Raster passen als die Dokumentenvektoren. Wir können hoffen, daß die Qualität der Suchergebnisse steigt und der Speicherplatzbedarf für die Term-Dokument-Matrix sinkt, wenn wir die Dokumentenvektoren zu Äquivalenzklassen zusammenfassen.

Eine solche Zusammenfassung muß natürlich automatisch erfolgen; alles andere wäre bei wirklich umfangreichen Sammlungen von Dokumenten völlig unrealistisch.

Wir können zumindest informell so tun, als sei der Dokumentenvektor zusammengesetzt aus zwei Komponenten: dem „wirklichen“ Inhalt des Dokuments und einer Art „Rauschen“, das von Zufälligkeiten der Wortwahl und Ähnlichem abhängt. Auch wenn zumindest ich keine Chance sehe, diese beiden Komponenten auch nur einigermaßen präzise zu definieren, befinden wir uns damit doch in einer Situation, mit der wir von anderen Anwendungen her vertraut sind:

a) Lotfußpunkte

Auch hier zerlegen wir einen Vektor in zwei Komponenten: Wenn, im einfachsten Fall, ein Vektor $w \in \mathbb{R}^2$ senkrecht projiziert werden soll auf die Gerade durch den Nullpunkt mit Steigungsvektor u, wollen wir w darstellen als Summe eines Vektors parallel zu u und eines auf u senkrecht stehenden Vektors v:

$$w = \lambda u + v \quad \text{mit} \quad \lambda \in \mathbb{R} \quad \text{und} \quad v \perp u\,.$$

Bilden wir auf beiden Seiten das Skalarprodukt mit u, erhalten wir die Gleichung

$$\langle w, u\rangle = \lambda \langle u, u\rangle + \langle v, u\rangle = \lambda \langle u, u\rangle \quad \text{oder} \quad \lambda = \frac{\langle w, u\rangle}{\langle u, u\rangle}\,.$$

Entsprechend können wir auch im Höherdimensionalen vorgehen: Um einen Vektor $w \in \mathbb{R}^n$ zu projizieren auf den Unterraum, der von den Vektoren $u_1, \ldots, u_r$ aufgespannt wird, zerlegen wir w in eine Linearkombination der u_i sowie einen Lotvektor v, der auf allen u_i senkrecht steht:

$$w = \lambda_1 u_1 + \cdots + \lambda_r u_r + v \quad \text{mit} \quad v \perp u_1, \ldots, v \perp u_r\,.$$

Skalarmultiplikation mit u_i macht daraus

$$\langle w, u_i\rangle = \lambda_1 \langle u_1, u_i\rangle + \cdots + \lambda_r \langle u_r, u_i\rangle\,,$$

und die r so erhaltenen linearen Gleichungen bilden ein lineares Gleichungssystem für die gesuchten Parameter $\lambda_1, \ldots, \lambda_r$.

Besonders einfach wird die Situation, wenn die u_i eine Orthonormalbasis des betrachteten Unterraums bilden, wenn also $\langle u_i, u_j \rangle$ für $i \neq j$ verschwindet und für $i = j$ eins ist. Dann werden die Gleichungen einfach zu $\langle w, u_i \rangle = \lambda_i$. Man beachte, daß wir in keinem Fall den Vektor v wirklich ausrechnen müssen.

b) Überbestimmte lineare Gleichungssysteme

Wenn in einem linearen Gleichungssystem

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \quad \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

die Anzahl m der Gleichungen größer ist als die Anzahl n der Unbekannten, gibt es im allgemeinen keine Lösung. Nun kann es, je nach Anwendung, allerdings vorkommen, daß das Gleichungssystem aus physikalischen oder sonstigen Gründen eine Lösung haben müßte, daß aber die a_{ij} und oder b_i durch Meß- oder Rundungsfehler verfälscht sind und das Gleichungssystem erst dadurch unlösbar wird. Unsere Aufgabe in solchen Fällen besteht darin, eine „Lösung“ $(x_1, \ldots, x_n)$ zu finden derart, daß die Unterschiede zwischen den linken und den rechten Seiten möglichst gering sind.

Fassen wir die Koeffizienten $a_{1j}, \ldots, a_{mj}$ zusammen zu einem Vektor $a_j \in \mathbb{R}^m$ und die rechten Seiten zu einem Vektor $b \in \mathbb{R}^m$, suchen wir also Zahlen $x_1, \ldots, x_n$ derart, daß $x_1a_1 + \cdots + x_na_n$ möglichst nahe bei b liegt.

Das Gleichungssystem ist genau dann exakt lösbar, wenn b im von den Vektoren a_j erzeugten Unterraum U von $\mathbb{R}^m$ liegt. Andernfalls besteht unsere beste Strategie darin, daß wir b senkrecht in diesen Untervektorraum projizieren und das Gleichungssystem mit dem projizierten Vektor c als neuer rechter Seite lösen. Da $v = b - c$ senkrecht auf U steht, ist das wieder äquivalent dazu, daß wir reelle Zahlen x_i suchen, für die gilt

$$x_1a_1 + \cdots + x_na_n + v = b \quad \text{mit} \quad v \perp a_1, \ldots, v \perp a_n \,.$$

Das ist genau das Problem, das wir im vorigen Abschnitt gelöst haben.

Besonders übersichtlich wird die Lösung, wenn wir das Gleichungssystem in Matrixform schreiben als $Ax = b$, wobei A diejenige $m \times n$-Matrix bezeichnet, deren Spalten die Vektoren a_j sind. Da dieses Gleichungssystem unlösbar ist, suchen wir tatsächlich eine Lösung von

$$Ax + v = b \,,$$

wobei v auf allen a_j senkrecht stehen soll; die Skalarprodukte $\langle a_j, v \rangle$ müssen also für $j = 1, \ldots, m$ verschwinden.

Diese Orthogonalitätsbedingung läßt sich ebenfalls kompakter mit Matrizen schreiben: Die transponierte Matrix A^T hat in der j-ten Zeile die Einträge des Vektors a_j stehen; die j-te Komponente von A^Tv ist also das Skalarprodukt $\langle a_j, v \rangle$. Somit muß für den obigen Vektor v das Produkt A^Tv gleich dem Nullvektor sein. Multiplizieren wir daher die Gleichung $Ax + v = b$ von links mit der Matrix A^T, erhalten wir das neue, lösbare Gleichungssystem

$$(A^TA)x = A^Tb \,,$$

dessen Lösungsvektor(en) x für das überbestimmte Gleichungssystem das beste ist (sind), was wir bezüglich der EUKLIDischen Norm bekommen können.

c) Lineare Regression

Hauptanwendung solcher überbestimmter linearer Gleichungssystem ist die Ausgleichsrechnung; das bekannteste Beispiel dazu wiederum sind Ausgleichsgeraden: Hier haben wir zwei Meßgrößen x, y, zwischen denen wir einen Zusammenhang der Form $y = ax + b$ erwarten mit unbekannten Parametern a und b. Für x und y haben wir eine Reihe von Messungen (x_i, y_i) für $i = 1, \ldots, N$, und natürlich wird es im allgemeinen keine zwei reellen Zahlen a, b geben, so daß für alle i gilt $y_i = ax_i + b$.

Die N Gleichungen $y_i = ax_i + b$ bilden für $N > 2$ ein solches überbestimmtes lineares Gleichungssystem aus N Gleichungen für die beiden Unbekannten a und b. (Im Gegensatz zu sonst sind hier also a, b unbekannt, während die x_i, y_i bekannt sind.)

Die Matrix A dieses Gleichungssystem hat zwei Spalten: In der ersten stehen die x_i, in der zweiten stehen lauter Einsen. Der Vektor auf der rechten Seite ist der Vektor y, dessen Komponenten die y_i sind.

A^T hat entsprechend zwei Zeilen, wobei in der ersten die x_i stehen und in der zweiten lauter Einsen. Somit ist

$$A^T A = \begin{pmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & N \end{pmatrix} \quad \text{und} \quad A^T b = \begin{pmatrix} \sum_{i=1}^N x_i y_i \\ \sum_{i=1}^N y_i \end{pmatrix} .$$

a und b sind somit Lösungen des linearen Gleichungssystems

$$\sum_{i=1}^N x_i^2 \cdot a + \sum_{i=1}^N x_i \cdot b = \sum_{i=1}^N x_i y_i \quad \text{und} \quad \sum_{i=1}^N x_i \cdot a + N \cdot b = \sum_{i=1}^N y_i ,$$

die man auch leicht als geschlossene Formel angeben kann.

Entsprechend lassen sich auch im Höherdimensionalen zu vorgegebenen Datenpunkten $(x_{i1}, \ldots, x_{i_m}, y_i) \in \mathbb{R}^{m+1}$ beliebige lineare Regressionsansätze der Form

$$y_i = \sum_{j=1}^m a_j f_j(x_{i1}, \ldots, x_{im})$$

mit unbestimmten Koeffizienten a_j auf überbestimmte lineare Gleichungssysteme zurückführen, die nach Multiplikation mit der Transponierten der Matrix des Gleichungssystems ein neues System liefern, dessen Lösungen die nach der Methode der kleinsten Quadrate bestmöglichen Koeffizienten sind. Man beachte, daß der Ansatz *nur* in den a_j linear sein muß; die Funktionen f_j können beliebig gewählt werden.

d) Projektion auf optimale affine Teilräume

Oftmals haben wir in unserem Modell keine expliziten Gleichungen, die eine der Variablen als Funktion der anderen darstellen, sondern wir suchen einfach eine Relation, die eine Wolke von Datenpunkten möglichst gut beschreibt. Hier wollen wir uns auf den einfachsten Fall einer linearen Relation beschränken; wir suchen also einen affinen Teilraum einer vorgegebenen Dimension, in dessen „Nähe“ die Datenpunkte liegen. Da

alle wesentlichen Ideen schon im Eindiemsionalen auftreten, wollen wir zunächst der Fall einer Geraden ausführlich betrachten.

Wir haben also N Punkte $p_1, \ldots, p_N \in \mathbb{R}^n$ und suchen dazu eine im Sinne der kleinsten Quadrate optimale Gerade g. Eine Gerade läßt sich schreiben in der Form

$$g = \left\{ a + tm \;\middle|\; t \in \mathbb{R} \right\} ,$$

wobei wir annehmen können, daß der Vektor m die Länge eins hat.

Ist $q_i = a + t_i m$ die orthogonale Projektion von p_i auf g, so steht der Differenzvektor $p_i - q_i$ senkrecht auf m, d.h.

$$\langle p_i - a - t_i m, m \rangle = \langle p_i - a_i, m \rangle - t_i = 0 ;$$

somit ist $q_i = a + \langle p_i - a, m \rangle m$.

Gesucht ist jene Gerade g, für die die Summe der Abstandsquadrate zu g minimal wird; mit $\|x\| \underset{\text{def}}{=} \sqrt{\langle x, x \rangle}$ soll also

$$\sum_{i=1}^N \|p_i - q_i\|^2 = \sum_{i=1}^N \|(p_i - a - \langle p_i - a, m \rangle m\|^2$$

minimal werden.

Wir überlegen uns zunächst, daß dann das arithmetische Mittel (der Schwerpunkt) der p_i auf g liegen muß: Für

$$v \underset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^N (p_i - q_i) \quad \text{und} \quad r_i \underset{\text{def}}{=} p_i - q_i - v$$

ist

$$\sum_{i=1}^N \|p_i - q_i\|^2 = \sum_{i=1}^N \|v + r_i\|^2 = N \langle v, v \rangle + 2 \left\langle v, \sum_{i=1}^N r_i \right\rangle + \sum_{i=1}^N \langle r_i, r_i \rangle .$$

Dabei ist $\sum_{i=1}^N r_i = \sum_{i=1}^N (p_i - q_i) - Nv = 0$, also

$$\sum_{i=1}^N \|p_i - q_i\|^2 = N \|v\|^2 + \sum_{i=1}^N \|r_i\|^2 .$$

Geometrisch betrachtet ist $r_i = p_i - (q_i + v)$ der Differenzvektor zwischen p_i und dem Vektor $q_i + v$ auf der Geraden

$$\widetilde{g} = \left\{ a + v + tm \;\middle|\; t \in \mathbb{R} \right\} .$$

Der Abstand von p_i zur Geraden $\widetilde{g}$ ist also höchstens gleich der Länge von r_i, und wäre v nicht der Nullvektor, so wäre die Summe der Abstandsquadrate für $\widetilde{g}$ kleiner als für g. Nach Wahl von g muß somit $v = 0$ sein und

$$s \underset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} p_i = \frac{1}{N} \sum_{i=1}^{N} q_i$$

liegt als Schwerpunkt von Punkten auf der Geraden g selbst auf g.

Wir können daher in der Geradengleichung $a = s$ setzen und müssen nun noch einen Vektor m der Länge eins finden, für den die Summe der Abstandsquadrate zu g minimal wird.

Schreiben wir zur Abkürzung $b_i = p_i - s$, so ist $q_i = s + < p_i - s, m > m = s + < b_i, m > m$, also $p_i - q_i = b_i - < b_i, m > m$ und

$$\begin{aligned} \sum_{i=1}^{N} \|p_i - q_i\|^2 &= \sum_{i=1}^{N} \|b_i - \langle b_i, m\rangle\, m\|^2 \\ &= \sum_{i=1}^{N} \|b_i\|^2 - 2\sum_{i=1}^{N} \langle b_i, m\rangle \langle b_i, m\rangle + \sum_{i=1}^{N} \|\langle b_i, m\rangle\, m\|^2 \\ &= \sum_{i=1}^{N} \|b_i\|^2 - \sum_{i=1}^{N} \|\langle b_i, m\rangle\, m\|^2 \ . \end{aligned}$$

Da die erste Summe in der zweiten Zeile nicht von m abhängt, muß der gesuchte Vektor m die zweite Summe dort maximal machen.

Bezeichnet B die $N \times N$-Matrix, deren Zeilen die Vektoren b_i sind, so ist Bm der Spaltenvektor mit Einträgen $b_i m$, und das Skalarprodukt dieses Vektors mit sich selbst ist gleich der zu maximierenden Summe.

Identifizieren wir Vektoren mit einspaltigen Matrizen, so ist das Skalarprodukt zweier Vektoren u, v gleich dem Matrixprodukt $u^T \cdot v$, wobei

u^T die transponierte Matrix bezeichnet von u bezeichnet, also den zugehörigen Zeilenvektor. Das Skalarprodukt von Bm mit sich selbst ist somit gleich

$$(Bm)^T(Bm) = (m^T B^T)(Bm) = m^T(B^T B)m \ .$$

$B^T B$ ist eine symmetrische reelle $N \times N$-Matrix; wie wir wissen, sind alle ihre Eigenwerte reell, und der $\mathbb{R}^N$ hat eine Basis aus Eigenvektoren von $B^T B$.

Wenn wir in dieser Basis rechnen, wird $B^T B$ zur Diagonalmatrix mit den Eigenwerten $\lambda_1, \ldots, \lambda_N$ als Einträgen, und sind $m_1, \ldots, m_N$ die Komponenten von m bezüglich der Basis aus Eigenvektoren, so ist

$$\begin{aligned} m^T(B^T B)m &= (m_1\ m_2\ \ldots\ m_N) \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{pmatrix} \\ &= \sum_{i=1}^{N} \lambda_i m_i^2 \ . \end{aligned}$$

Da $m^T(B^T B)m$ als Längenquadrat des Vektors Bm nicht negativ werden kann, sind dabei alle Eigenwerte $\lambda_i \geq 0$.

Damit ist klar, daß der gesuchte Vektor m Eigenvektor der Länge eins zum größten Eigenwert von $B^T B$ sein muß. Falls dieser Eigenwert Vielfachheit eins hat, ist m bis aufs Vorzeichen eindeutig bestimmt, und es gibt nur eine Lösungsgerade; andernfalls sind alle Geraden im affinen Teilraum durch v mit einem Richtungsvektor aus dem Eigenraum Lösungen.

Mit minimalen Veränderungen bei den obigen Argumenten ist nun auch klar, was der optimale r-dimensionale affine Teilraum

$$A = \left\{ a + t_1 m_1 + \cdots + t_r m_r \;\middle|\; t_1, \ldots, t_r \in \mathbb{R} \right\}$$

zu den vorgegebenen Punkten ist: Für a können wir wieder den Schwerpunkt der p_i nehmen, und $m_1, \ldots, m_r$ sind die Eigenvektoren zu den r größten Eigenwerten von $B^T B$.

e) Orthogonalität bei Matrizen

Bei den bisherigen Beispielen wußten wir stets, in welchem Untervektorraum die gesuchte Projektion liegen sollte; im Falle der Term-Dokument-Matrix war bislang nur vage die Rede von einem „Inhalt“, der nie exakt definiert wurde.

Angenommen, wir haben zwei Dokumente, die so ähnlich sind, daß wir ihre Inhalte bezüglich praktisch jeder Suchanfrage als äquivalent betrachten. Angesichts der automatischen und rein formalen Zuordnung von Dokumentenvektoren wird es selbst dann immer wieder vorkommen, daß für die beiden Dokumente verschiedene Vektoren berechnet werden. Wenn wir die Länge der Vektoren auf eins oder eine sonstige Konstante normieren, bedeutet diese Verschiedenheit insbesondere, daß die Vektoren linear unabhängig sind.

Für die gesamte Term-Dokument-Matrix hat dies zur Folge, daß es viel mehr linear unabhängige Spalten gibt als eigentlich notwendig. Von daher bietet sich an, auf einen Raum von Matrizen niedrigeren Ranges zu projizieren; um konkrete Zahlenwerte wollen wir uns im Augenblick noch nicht kümmern, und auch die Tatsache, daß Matrizen eines vorgegebenen Rangs (oder auch höchstens eines vorgegebenen Rangs) nur in den seltensten Fällen einen Untervektorraum bilden, soll uns nicht stören.

Von orthogonalen Projektionen können wir erst reden, wenn wir einen Orthogonalitätsbegriff, d.h. also ein Skalarprodukt haben. Wir wählen dazu die einfachste Möglichkeit: Wir fassen eine $n \times m$-Matrix auf als einen Vektor aus $\mathbb{R}^{nm}$ und nehmen dort das übliche Standardskalarprodukt, d.h.

$$\langle A, B\rangle \underset{\text{def}}{=} \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}b_{ij} \; .$$

Wer will, kann das auch kompakter formulieren: Wie stumpfsinniges Nachrechnen zeigt, ist

$$\langle A, B\rangle = \text{Spur}(A^T B) \; ,$$

was wir allerdings im folgenden nicht brauchen werden. Die Norm $||A|| = \sqrt{\langle A, A\rangle}$ zu diesem Skalarprodukt wird, um sie von den vielen

anderen möglichen Matrixnormen zu unterscheiden, als FROBENIUS-Norm bezeichnet.

f) Orthonormalbasen im Vektorraum der Matrizen

Da wir den Vektorraum $\mathbb{R}^{m\times n}$ der $m \times n$-Matrizen mit dem Vektorraum $\mathbb{R}^{mn}$ identifizieren, haben wir eine Standardbasis, bestehend aus Matrizen, bei denen genau ein Eintrag gleich eins ist und alle anderen gleich null; wie jede Standardbasis eines $\mathbb{R}^N$ ist das natürlich eine Orthonormalbasis. Wir wollen uns überlegen, daß wir uns auch aus zwei beliebigen Orthonormalbasen von $\mathbb{R}^n$ und $\mathbb{R}^m$ eine Orthonormalbasis von $\mathbb{R}^{n\times m}$ konstruieren können.

Dazu definieren wir zunächst für zwei Vektoren

$$v = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \in \mathbb{R}^m \quad \text{und} \quad w = \begin{pmatrix} w_1 \\ \vdots \\ v_n \end{pmatrix} \in \mathbb{R}^n$$

deren *Tensorprodukt* als die Matrix

$$v \otimes w = \begin{pmatrix} v_1w_1 & v_1w_2 & \dots v_1w_n \\ v_2w_1 & v_2w_2 & \dots v_2w_n \\ \vdots & \vdots & \ddots \quad \vdots \\ v_mw_1 & v_mw_2 & \dots v_mw_n \end{pmatrix} \in \mathbb{R}^{m\times n} \; .$$

(Das Zeichen „$\otimes$“ wird in diesem Zusammenhang ausgesprochen als „Tensor“).

Lemma: $(v_1, \dots, v_m)$ und $(w_1, \dots, w_n)$ seien Orthonormalbasen von $\mathbb{R}^n$ *bzw.* $\mathbb{R}^m$. Dann bilden die Vektoren $v_i \otimes w_j$ eine Orthonormalbasis des Vektorraums $\mathbb{R}^{n\times m}$.

Beweis: Wir bezeichnen die Komponenten der Vektoren v_i mit $v_{i\mu}$, die

der Vektoren w_j mit $w_{j\nu}$. Dann ist

$$\begin{aligned}\langle v_i \otimes w_j, v_k \otimes w_\ell \rangle &= \sum_{\mu=1}^{m}\sum_{\nu=1}^{n}(v_{i\mu}w_{j\nu})(v_{k\mu}w_{\ell\nu}) \\ &= \sum_{\mu=1}^{m} v_{i\mu}w_{k\mu}\sum_{\nu=1}^{n} w_{j\nu}v_{\ell\nu} \\ &= \langle v_i, v_k\rangle \langle w_j, w_\ell\rangle \ .\end{aligned}$$

Ist $i \neq k$ oder $j \neq \ell$, so verschwindet rechts mindestens einer der beiden Faktoren, also auch das Produkt. Ist aber $i = k$ und $j = \ell$, so sind beide Skalarprodukte rechts gleich eins, also auch das Skalarprodukt links. ∎

§3: Die Singulärwertzerlegung

Unser nächstes Ziel ist es, zu einer gegebenen Matrix $A \in \mathbb{R}^{m\times n}$ Orthonormalbasen $v_1, \dots v_m$ von $\mathbb{R}^m$ und $w_1, \dots, w_n$ von $\mathbb{R}^n$ zu finden, derart, daß A in der Orthonormalbasis aus den Matrizen $v_i \otimes w_j$ eine möglichst kurze Basisdarstellung hat. Offensichtlich hat jede der Matrizen $v_i \otimes w_j$ den Rang eins, denn jede ihrer Spalten ist proportional zu w_j, und jede ihrer Zeilen ist proportional zu v_i. Eine Matrix vom Rang r muß daher eine Linearkombination von mindestens r Basismatrizen sein; wir wollen eine Basis finden, in der wir mit genau r auskommen.

Satz: Zu jeder linearen Abbildung

$$\varphi\colon \begin{cases} \mathbb{R}^m \to \mathbb{R}^n \\ v \mapsto Av \end{cases}$$

gibt es Orthonormalbasen $v_1, \dots v_m$ von $\mathbb{R}^m$ und $u_1, \dots, u_n$ von $\mathbb{R}^n$ derart, daß in der Abbildungsmatrix Σ von φ bezüglich dieser Basen alle Einträge σ_{ij} mit $i \neq j$ verschwinden und für die Einträge σ_{ii} gilt:

$$\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{rr} \ .$$

Für $n = m$ ist Σ also eine Diagonalmatrix, für $n > m$ eine durch Nullspalten und für $m > n$ eine durch Nullzeilen erweiterte Diagonalmatrix.

Den *Beweis* führen wir durch Induktion nach dem Minimum der beiden Zahlen m und n:

Ist dieses Minimum gleich eins, so ist $m = 1$ oder $n = 1$ oder beides.

Im Falle $m = 1$ nehmen wir für $\mathbb{R}^m = \mathbb{R}$ die Orthonormalbasis bestehend aus der Eins. Falls A die Nullmatrix ist, können wir für $\mathbb{R}^n$ eine beliebige Orthonormalbasis wählen; andernfalls nehmen wir als ersten Basisvektor den Vektor $\varphi(1)$ dividiert durch seine Länge und ergänzen ihn zu einer Orthonormalbasis von $\mathbb{R}^n$.

Ist $n = 1$, nehmen wir entsprechend für $\mathbb{R}^n = \mathbb{R}$ die Orthonormalbasis bestehend aus der Eins. In $\mathbb{R}^m$ nehmen wir irgendeine Orthonormalbasis des Kerns und ergänzen sie durch einen weiteren Vektor zu einer Orthonormalbasis von ganz $\mathbb{R}^n$; diesen weiteren Vektor betrachten wir als ersten Basisvektor.

Wenn das Minimum größer als eins ist und A die Nullmatrix, können wir für $\mathbb{R}^n$ und $\mathbb{R}^m$ beliebige Orthonormalbasen wählen und alle $\sigma_i = 0$ setzen.

Für alle anderen Matrizen A betrachten wir in $\mathbb{R}^m$ die Einheitssphäre

$$S = \left\{ v \in \mathbb{R}^m \;\middle|\; |v| = 1 \right\}$$

bestehend aus allen Vektoren der Länge eins, und darauf die Abbildung

$$\psi\colon \begin{cases} S \to \mathbb{R} \\ v \mapsto |\varphi(v)| \end{cases} ,$$

die jedem Vektor $v \in S$ die Länge des Vektors $\varphi(v) = Av \in \mathbb{R}^n$ zuordnet. Da S kompakt ist, nimmt ψ sein Maximum an; dieses sei σ_1 und werde für den Vektor $v_1 \in S$ angenommen. Da A nicht die Nullmatrix ist, kann σ_1 nicht verschwinden; wir können daher dividieren und setzen

$$u_1 = \frac{\varphi(v_1)}{\sigma_1} \ .$$

Dann ist $\varphi(v_1) = \sigma_1 u_1$, und u_1 ist wie v_1 ein Vektor der Länge eins.

Nach dem Basisergänzungssatz in Verbindung mit dem Orthogonalisierungsverfahren von GRAM und SCHMIDT können wir dazu weitere

Vektoren $v_2, \ldots, v_n$ und $u_2, \ldots, u_m$ finden derart, daß die Vektoren v_i eine Orthonormalbasis von $\mathbb{R}^n$ bilden und die u_j eine von $\mathbb{R}^m$. Bezüglich dieser beiden Basen habe φ die Abbildungsmatrix A_1.

Da die Spaltenvektoren der Abbildungsmatrix die Koeffizienten der Basisdarstellung der Bildvektoren sind und v_1 aus $\sigma_1 u_1$ abgebildet wird, hat die erste Spalte von A_1 in der ersten Zeile den Eintrag σ_1, und alle anderen Einträge verschwinden. Wir wollen uns überlegen, daß auch in der ersten *Zeile* von A_1 alle anderen Einträge verschwinden müssen.

Wir schreiben

$$A_1 = \begin{pmatrix} \sigma_1 & b_{01} & \cdots & b_{0,m-1} \\ 0 & b_{11} & \cdots & b_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b_{n-1,1} & \cdots & b_{n-1,m-1} \end{pmatrix}$$

und betrachten den Vektor

$$v = \sigma_1 v_1 + b_{01} v_2 + \cdots + b_{0,m-1} v_m \in \mathbb{R}^m \, .$$

Da er bezüglich einer Orthonormalbasis dargestellt ist, können wir das Quadrat seiner Länge einfach als Summe der Koeffizientenquadrate berechnen, d.h.

$$\|v\|^2 = \sigma_1^2 + \sum_{j=1}^{m-1} b_{0j}^2 \, .$$

Sein Bild unter φ ist

$$\begin{aligned} \varphi(v) &= \sigma_1 \varphi(v_1) + b_{01} \varphi(v_2) + \cdots + b_{0,m-1} \varphi(v_{m-1}) \\ &= \sigma_1^2 u_1 + b_{01} b^{(1)} + \cdots b_{0,m-1} b^{(m-1)} \, , \end{aligned}$$

wobei $b^{(j)}$ den $(j-1)$-ten Spaltenvektor von A_1 bezeichnet. In Koordinaten ausgedrückt ist somit

$$\varphi(v) = \begin{pmatrix} \sigma_1^2 + b_{01}^2 + \cdots + b_{0,m-1}^2 \\ b_{01} b_{11} + \cdots + b_{0,m-1} b_{1,m-1} \\ \vdots \\ b_{01} b_{n-1,1} + \cdots + b_{0,m-1} b_{n-1,m-1} \end{pmatrix} \, .$$

Das Längenquadrat dieses Vektors ist Quadratsumme der Einträge, d.h.

$$\|\varphi(v)\|^2 \geq \left(\sigma_1^2 + b_{01}^2 + \cdots + b_{0,m-1}^2\right)^2 \, .$$

Der Einheitsvektor

$$v_0 \underset{\text{def}}{=} \frac{v}{\|v\|} = \frac{v}{\sqrt{\sigma_1^2 + b_{01}^2 + \cdots + b_{0,m-1}^2}}$$

wird dementsprechend abgebildet auf den Vektor $\varphi(v)/\|v\|$, dessen Länge größer oder gleich

$$\frac{\sigma_1^2 + b_{01}^2 + \cdots + b_{0,m-1}^2}{\sqrt{\sigma_1^2 + b_{01}^2 + \cdots + b_{0,m-1}^2}} \geq \sqrt{\sigma_1^2 + b_{01}^2 + \cdots + b_{0,m-1}^2}$$

ist. Diese Länge kann aber höchstens gleich σ_1 sein, denn nach Konstruktion ist das ja die größtmögliche Länge für das Bild eines Vektors der Länge eins. Somit müssen alle b_{0j} verschwinden; die Matrix A_1 hat also die Form

$$A_1 = \begin{pmatrix} \sigma_1 & 0 \\ 0 & B \end{pmatrix}$$

mit einer $(n-1) \times (m-1)$-Matrix B. Dies zeigt, daß φ den von v_2 bis v_m erzeugten Untervektorraum des $\mathbb{R}^m$ auf den von u_2 bis u_n erzeugten Untervektorraum des $\mathbb{R}^n$ abbildet. Schränken wir die Abbildung φ ein auf diese beiden Untervektorräume, haben wir jeweils um eins kleinere Dimensionen; nach Induktionsannahme gibt es also Orthonormalbasen dieser Untervektorräume, bezüglich derer die Einschränkung von φ Diagonalgestalt hat. Nach Wahl von σ_1 ist klar, daß alle Diagolaneinträge kleiner oder gleich σ_1 sein müssen.

Ersetzen wir $v_2, \ldots, v_m$ und $u_2, \ldots, u_n$ durch die Vektoren dieser Basen, erhalten wir zusammen mit v_1 und u_1 Orthonormalbasen von $\mathbb{R}^m$ und $\mathbb{R}^n$, bezüglich derer die Abbildungsmatrix Σ von φ keine Einträge σ_{ij} mit $i \neq j$ hat und für die $\sigma_{ii} = \sigma_i$ gilt $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, wobei r das Minimum der beiden Dimensionen m und n bezeichnet. ∎

Der Wechsel von der Standardbasis zu dieser Orthonormalbasis wird jeweils durch eine orthogonale Matrix beschrieben; somit haben wir bewiesen:

Satz: Jede reelle $m \times n$-Matrix A läßt sich als ein Produkt $A = U\Sigma V^T$ schreiben mit orthogonalen Matrizen $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$ sowie einer Matrix $\Sigma \in \mathbb{R}^{m\times n}$, in der alle Einträge σ_{ij} mit $i \neq j$ verschwinden und für die restlichen Einträge gilt $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{rr}$. ∎

Definition: Die Zahlen $\sigma_i \underset{\text{def}}{=} \sigma_{ii}$ heißen *singuläre Werte* von A; die Spaltenvektoren von U und von V bezeichnen wir als *singuläre Vektoren* von A. Die Zerlegung $A = U\Sigma V^T$ heißt *Singulärwertzerlegung* der Matrix A.

Da die Matrizen U und V orthogonal sind, sind ihre inversen Matrizen einfach die transponierten. Dies können wir ausnützen um die singulären Werte und Vektoren mit bekannten Größen in Verbindung zu bringen:

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = U\Delta U^T = U\Delta U^{-1},$$

wobei Δ eine Diagonalmatrix mit Einträgen σ_i^2 ist. Somit sind die σ_i die Wurzeln der Eigenwerte von AA^T.

Multiplizieren wir die Gleichung $A = U\Sigma V^T$ von rechts mit V, erhalten wir die Gleichung $AV = U\Sigma$, denn für eine orthogonale Matrix V ist $V^T V = VV^T$ gleich der Einheitsmatrix. Für den i-ten Spaltenvektor v_i von V ist daher $Av_i = \sigma_i u_i$.

Entsprechend können wir $A^T = V\Sigma^T U^T$ von rechts mit U multiplizieren und erhalten $A^T U = V\Sigma^T$; für den i-ten Spaltenvektor u_i von U ist daher $A^T u_i = \sigma_i v_i$.

Fassen wir beides zusammen, erhalten wir die Gleichungen

$$A^T A v_i = \sigma_i^2 v_i \quad \text{und} \quad AA^T u_i = \sigma_i^2 u_i\,;$$

die singulären Vektoren sind also die Eigenvektoren von $A^T A$ *bzw.* AA^T.

Da die Matrizen U und V als orthogonale Matrizen invertierbar sind, ist der Rang der Ausgangsmatrix A gleich dem der Matrix Σ, d.h. gleich der Anzahl r der von Null verschiedenen Singulärwerte.

Da die Einträge der Matrix Σ höchstens dann von Null verschieden sein können, wenn der Zeilenindex gleich dem Spaltenindex ist, wird im

Produkt $U\Sigma$ der i-te Spaltenvektor von U mit σ_i multipliziert. Entsprechend wird im Produkt ΣV^T der i-te Zeilenvektor von V^T, d.h. also die i-te Spalte von V, mit σ_i multipliziert. Für $i > r$ ist $\sigma_i = 0$, die entsprechenden Spalten von U und V spielen also für die Berechnung von $A = U\Sigma V^T$ keinerlei Rolle und müssen somit auch nicht abgespeichert werden. Bezeichnet U_1 die $n \times r$-Matrix aus den ersten r Spaltenvektoren von U, V_1 die $m \times r$-Matrix aus den ersten r Spaltenvektoren von V und Σ_1 die $r \times r$-Diagonalmatrix mit Einträgen $\sigma_1, \ldots . \sigma_r$, so ist also auch

$$A = U_1\Sigma_1 V_1^T \quad \text{mit} \quad U_1 \in \mathbb{R}^{n\times r}, \quad V_1 \in \mathbb{R}^{m\times r} \quad \text{und} \quad \Sigma_1 \in \mathbb{R}^{r\times r}\,.$$

Ausgedrückt durch die Spaltenvektoren u_i, v_i von U_1 und V_1 können wir dies auch schreiben als

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i$$

mit dem in §2*f)* eingeführten Tensorprodukt.

Die Projektion auf den Raum aller Matrizen vom Rang höchstens s für ein $s \leq r$ liefert uns der folgende

Satz: A sei eine Matrix vom Rang r; ihre Singulärwertzerlegung sei

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i\,.$$

Dann ist für jedes $s \leq r$ die Matrix

$$A_s = \sum_{i=1}^{s} \sigma_i u_i \otimes v_i$$

eine orthogonale Projektion von A auf einen Vektorraum von Matrizen mit Rang höchstens s, und für jede Matrix B vom Rank höchstens s gilt: $\|A - B\| \geq \|A_s - A\|$.

(Als Matrix können wirflA_s wie folgt definieren: Die Matrix Σ_s entstehe aus Σ dadurch, daß alle σ_{ii} mit $i > s$ auf Null gesetzt werden. Dann ist $A_s = U\Sigma_s V^T$. Die Matrix A_s ist genau dann eindeutig bestimmt, wenn $\sigma_s > \sigma_{s+1}$ ist; andernfalls gibt es mehrere Lösungen.)

Beweis: Da die sämtlichen u_i *bzw.* v_j jeweils eine Orthonormalbasis des zu Grunde liegenden Vektorraums bilden, bilden die $u_i \otimes v_j$ nach §2*f)* eine Orthonormalbasis des entsprechenden Vektorraums von Matrizen. Wir betrachten eine beliebige Matrix B aus diesem Raum und schreiben sie als

$$B = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij} u_i \otimes v_j \ .$$

Da das Quadrat der EUKLIDischen Norm bezüglich einer Orthonormalbasis einfach als Summe der Koeffizientenquadrate berechnet werden kann, ist

$$\|B - A\|^2 = \sum_{i=1}^{\min(m,n)} (b_{ii} - \sigma_i)^2 + \sum_{i \neq j} b_{ij}^2 \ .$$

Wenn dies minimal werden soll, müssen also zunächst alle b_{ij} mit $i \neq j$ verschwinden.

Von den Koeffizienten b_{ii} können für eine Matrix B vom Rang höchstens $s < r \leq \min(n,m)$ nicht mehr als s von Null verschieden sein; wir können daher nicht alle r Koeffizienten $b_{ii} = \sigma_i$ setzen, sondern müssen einige auch auf null setzen. Ein solcher Koeffizient liefert dann einen Beitrag von σ_i^2 zur obigen Summe. Da die σ_i der Größe nach geordnet sind, setzen wir somit $b_{ii} = \sigma_s$ für $i \leq s$ und $b_{ii} = 0$ sonst. ∎

§4: Latente semantische Analyse

Wir haben orthogonale Projektionen und die Singulärwertzerlegung betrachtet, um damit die Term-Dokument-Matrix zu „entrauschen“; wir wollen also den Vektor v zu einem Dokument auffassen als Summe zweier Vektoren, von denen der eine den „wirklichen“ Inhalt des Dokuments beschreibt, während im anderen all die Zufälligkeiten stecken, die individuelle Wortwahl (Karotte/Möhre, Auto/PKW) und Stil mit sich bringen. Wenn wir diese Zerlegung wirklich definieren und auch durchführen könnten, hätte die Matrix der „Inhaltsvektoren“ sicherlich einen kleineren Rang als die Term-Dokument-Matrix.

Ansatzpunkt der latenten semantischen Analyse ist die logisch unzulässige, aber praktisch bewährte Umkehrung dieser Aussage: Wenn wir die Term-Dokument-Matrix durch eine „benachbarte“ Matrix niedrigeren Rangs ersetzen, steht zu hoffen, daß deren Spalten eher den „Inhaltsvektoren“ entsprechen als die Spalten der Term-Dokument-Matrix.

Ein ähnliches Problem hat auch die numerische Mathematik beim Rechnen mit Gleitkomma-Matrizen: Falls die Spalten a_i einer Matrix einer linearen Gleichung $\sum \lambda_i a_i = 0$ genügen sollten, wird durch allfällige Rundungsfehler die rechte Seite tatsächlich oftmals verschieden vom Nullvektor; der Rang der Gleitkomma-Matrix wird also größer als der Rang der exakten Matrix.

Betrachtet man die singulären Werte einer solchen Matrix, so werden diese meist hinter einem festen Index plötzlich sehr viel kleiner. Diesen Index bezeichnet man als den (nicht wirklich exakt definierten) *numerischen Rang* der Matrix.

Als Beispiel betrachten wir die Matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \text{mit} \quad AA^T = \begin{pmatrix} 14 & 32 & 50 \\ 32 & 77 & 122 \\ 50 & 122 & 194 \end{pmatrix}$$

Die Eigenwerte von AA^T sind 0 und $\frac{1}{2}(285 \pm 3\sqrt{8881})$, und tatsächlich hat A natürlich nur den Rang zwei.

Ein Programm wie MatLab berechnet jedoch auch zu A eine „inverse“ Matrix (wenn auch mit Warnung über die schlechte Konditionszahl) und kommt auf die singulären Werte 16,85, 1,07 und $4{,}42 \cdot 10^{-16}$. Hier ist der numerische Rang offensichtlich gleich dem Rang zwei der exakten Matrix.

So extrem wie in diesem Beispiel ist der Abfall der singulären Werte im Falle von Term-Dokument-Matrizen natürlich nur selten; trotzdem ist meist *ungefähr* klar, ab wann sie klein genug werden, um vernachlässigt zu werden. Der wohl populärste Ansatz zur latenten semantischen Analyse besteht daher darin, die Term-Dokument-Matrix so auf eine Matrix niedrigeren Rangs zu projizieren und mit dieser zu arbeiten.

Als Beispiel für eine latente semantische Analyse betrachtet

LARS ELDÉN: Matrix Methods in Data Mining and Pattern Recognition, SIAM, 2007

fünf Dokumente folgenden Inhalts:

1. The Google^{TM} matrix P is a model of the internet.
2. P_{ij} is nonzero, if there is a link from Web page j to i.
3. The Google matrix is used to rank all Web pages.
4. The ranking is done by solving a matrix eigenvalue problem.
5. England dropped out of the top ten in the FIFA ranking.

Wenn wir die zehn Suchbegriffe *eigenvalue, England, FIFA, Google, internet, link, matrix, page, rank, Web* zulassen, erhalten wir die Term-Dokumentmatrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Die Suchanfrage „Ranking of Web Pages“ entspricht dem Spaltenvektor zu $(0,0,0,0,0,0,0,1,1,1)$; berechnen wir den Kosinus seines Winkels mit den fünf Spaltenvektoren von A erhalten wir die Werte $0, \frac{2}{3}, \frac{3}{\sqrt{15}} \approx 0{,}775$ und für die beiden letzten Spalten jeweils $\frac{1}{3}$. Demnach wäre das dritte Dokument das passendste, gefolgt vom zweiten, danach gleichrangig das vierte und das fünfte und am unpassendsten das erste.

Tatsächlich ist klar, daß für diese Anfrage das letzte Dokument völlig irrelevant ist, während das erste trotz disjunkter Suchbegriffe durchaus eine gewisse Bedeutung hat.

Die Singulärwertzerlegung von A ist $A = U_1 \Sigma V^T$ mit

$$U_1 = \begin{pmatrix} -0{,}142 & -0{,}243 & 0 & -0{,}578 & 0{,}364 \\ -0{,}0787 & -0{,}261 & -0{,}385 & 0{,}392 & 0{,}168 \\ -0{,}0787 & -0{,}261 & -0{,}385 & 0{,}392 & 0{,}168 \\ -0{,}392 & 0{,}0274 & 0{,}385 & 0{,}399 & -0{,}251 \\ -0{,}130 & -0{,}0740 & 0{,}385 & 0{,}375 & 0{,}495 \\ -0{,}102 & 0{,}373 & -0{,}192 & -0{,}0346 & 0{,}654 \\ -0{,}535 & -0{,}216 & 0{,}385 & -0{,}179 & 0{,}113 \\ -0{,}365 & 0{,}475 & -0{,}192 & -0{,}0102 & -0{,}0917 \\ -0{,}484 & -0{,}402 & -0{,}385 & -0{,}161 & -0{,}215 \\ -0{,}365 & 0{,}475 & -0{,}192 & -0{,}0102 & -0{,}0917 \end{pmatrix},$$

$$\Sigma = \begin{pmatrix} 2{,}85 & 0 & 0 & 0 & 0 \\ 0 & 1{,}88 & 0 & 0 & 0 \\ 0 & 0 & 1{,}73 & 0 & 0 \\ 0 & 0 & 0 & 1{,}26 & 0 \\ 0 & 0 & 0 & 0 & 0{,}848 \end{pmatrix}$$

und

$$V = \begin{pmatrix} -0{,}370 & -0{,}139 & 0{,}667 & 0{,}472 & 0{,}420 \\ -0{,}291 & 0{,}703 & -0{,}333 & -0{,}0436 & 0{,}555 \\ -0{,}750 & 0{,}191 & 0 & 0{,}0308 & -0{,}633 \\ -0{,}407 & -0{,}457 & 0 & -0{,}728 & 0{,}309 \\ -0{,}225 & -0{,}491 & -0{,}667 & 0{,}494 & 0{,}142 \end{pmatrix}.$$

Setzen wir zur latenten semantischen Analyse die letzten drei dieser Diagonaleinträge auf Null, erhalten wir die neue Matrix

$$\begin{pmatrix} -0{,}142 & -0{,}243 \\ -0{,}.0787 & -0{,}261 \\ -0{,}.0787 & -0{,}261 \\ -0{,}392 & 0{,}0274 \\ -0{,}130 & -0{,}0740 \\ -0{,}102 & 0{,}373 \\ -0{,}535 & -0{,}216 \\ -0{,}365 & 0{,}475 \\ -0{,}484 & -0{,}402 \\ -0{,}365 & 0{,}475 \end{pmatrix} \begin{pmatrix} 2{,}85 & 0 \\ 0 & 1{,}88 \end{pmatrix} \begin{pmatrix} -0{,}370 & -0{,}291 \\ -0{,}139 & 0{,}703 \\ 0{,}667 & -0{,}333 \\ 0{,}472 & 0{,}044 \\ 0{,}420 & 0{,}555 \end{pmatrix}$$

$$= \begin{pmatrix} 0{,}214 & -0{,}203 & 0{,}218 & 0{,}375 & 0{,}316 \\ 0{,}152 & -0{,}280 & 0{,}0748 & 0{,}316 & 0{,}291 \\ 0{,}152 & -0{,}280 & 0{,}0748 & 0{,}316 & 0{,}291 \\ 0{,}407 & 0{,}362 & 0{,}850 & 0{,}432 & 0{,}226 \\ 0{,}156 & 0{,}00992 & 0{,}251 & 0{,}214 & 0{,}152 \\ 0{,}00992 & 0{,}579 & 0{,}353 & -0{,}203 & -0{,}280 \\ 0{,}622 & 0{,}159 & 1.07 & 0{,}807 & 0{,}542 \\ 0{,}261 & 0{,}932 & 0{,}951 & 0{,}0147 & -0{,}205 \\ 0{,}617 & -0{,}130 & 0{,}891 & 0{,}908 & 0{,}682 \\ 0{,}261 & 0{,}932 & 0{,}951 & 0{,}0147 & -0{,}205 \end{pmatrix}.$$

Berechnen wir nun die Kosinuswerte der Winkel zwischen den Spalten und den Suchanfragen, erhalten wir die neuen Werte

$$0.604, \quad 0.640, \quad 0.743, \quad 0.374 \quad \text{und} \quad 0.140,$$

nach zwar weiterhin das dritte Dokument die beste Antwort ist, allerdings liegen nun sowohl das erste als auch das vierte deutlich vor dem irrelevanten fünften. Der Grund liegt natürlich darin, daß die Projektionen der entsprechenden Spaltenvektoren von A auf den von den ersten beiden Spaltenvektoren von U_1 aufgespannten Untervektorraum des $\mathbb{R}^{10}$ weitaus besser übereinstimmen als die Originale im $\mathbb{R}^{10}$.

§5: Der PageRank von Google

Google begann als studentisches Forschungsprojekt der beiden Doktoranden SERGEY BRIN und LAWRENCE PAGE an der Stanford University im kalifornischen Palo Alto; seine ersten Vorläufer waren auch nur dort auf dem Campus zugänglich. Als Stanfords Präsident JOHN HENNESSY, ein technischer Informatiker, Mitte der neunziger Jahre erstmals von der neuen Suchmaschine hörte, tippte er seinen Namen ein und erhielt gleich als erstes eine Seite von Stanford. Das war ihm bei der damals führenden Suchmaschine AltaVista noch nie passiert und trug sicherlich mit dazu bei, daß Stanford später die Kommerzialisierung von Google nach Kräften förderte.

Die ersten Prototypen berücksichtigten zur Anordnung der Suchergebnisse selbstverständlich noch nicht die heute üblichen über zweihundert

„Signale“; damals gab es im wesentlichen nur ein Kriterium, den *PageRank*. Er ist benannt nach LAWRENCE PAGE und patentiert als US Patent 6 285 999 vom 4. September 2001 mit Anschlußpatent 7 058 628 vom 6. Juni 2006*) Inhaber der Patente ist die Stanford University; als Erfinder ist jeweils LAWRENCE PAGE angegeben.

Im Gegensatz zu den meisten anderen Kriterien ist der PageRank unabhängig von jeder Suchanfrage: Durch ihn sollen *alle* (dem System bekannten) Webseiten nach ihrer Wichtigkeit geordnet werden. Getreu der allgemeinen Philosophie von Google muß diese Wichtigkeit nach einem gut skalierbaren Verfahren ohne menschliche Intervention berechenbar sein.

Die Grundidee dazu ist einfach und auch nicht neu: Seit langem wird immer wieder versucht, die Wichtigkeit wissenschaftlicher Arbeiten rein mechanisch zu bestimmen. Ein einfacher und deshalb gerne verwendeter Ansatz besteht darin, eine Arbeiten nach der Anzahl jener anderer Arbeiten zu beurteilen, in denen sie zitiert wird. Mit Hilfe des *Science Citation Index* und inzwischen auch *CiteSeer* (`citeseer.ist.psu.edu`) und ähnlichen Datenbanken läßt sich diese Anzahl leicht feststellen (oder zumindest schätzen, denn sie hängt natürlich ab vom Umfang der verwendeten Datenbank), und man erhält ein objektives Maß.

Weniger klar ist, was durch dieses Maß gemessen wird, denn an der Spitze stehen praktisch nie Arbeiten, die ein Fachwissenschaftler der entsprechenden Disziplin zu den wichtigsten aus dem betreffenden Zeitraum rechnen würde. Der Grund ist ziemlich klar: Ein kleines Licht mit einer großen Schar mittelmäßiger Schüler, die allesamt ständig den großen Meister zitieren, schneidet hier besser ab als ein Autor, der nur von wenigen hochkarätigen Spezialisten zitiert wird.

Ähnlich sieht es aus, wenn man diese Vorgehensweise auf das *World Wide Web* überträgt. Da eine Volltextsuchmaschine ohnehin Kopien aller ihr bekannter Webseiten im Speicher hat, kann sie zu jeder dieser Seiten

*) US-Patente sind auf dem offiziellen Server `patft.uspto.gov` des *United States Patent and Trademark Office* in HTML zu finden, pdf-Dateien bei `www.pat2pdf.org`.

leicht ermitteln, wie viele andere Seiten darauf verweisen und kann dann als Wichtigkeitsmäs für eine Seite S definieren

$$w_0(S) = \text{Anzahl der Seiten, die auf } S \text{ verweisen .}$$

Die Probleme mit diesem Maß sind im wesentliche dieselben wie oben:

1. Ein Verweis von einer wichtigen Seite sagt mehr aus, als ein Verweis von einer unwichtigen.
2. Wenn eine wichtige Seite auf hundert andere Seiten verweist, kann man das nicht vergleichen mit einem Verweis auf nur eine einzige Seite.
3. Durch Massenproduktion inhaltsleerer Seiten, deren einziger Zweck der Verweis auf eine zu pushende Webseite ist, läßt sich das Maß leicht manipulieren.

PAGE benutzt trotzdem die Informationen, die in der Verweisstruktur des World Wide Web steckt, allerdings mit einer Modifikation, die den ersten beiden Problemen entgegenwirkt und damit zumindest teilweise auch das dritte löst: Eine Seite ist wichtig, wenn wichtige Seiten auf sie verweisen, insbesondere dann, wenn diese nur auf wenige andere Seiten verweisen.

Ein erster Ansatz, dies in eine mathematische Formel umzusetzen, könnte folgender sein: Jede Seite S erhält eine Wichtigkeit $w(S)$, für die folgendes gilt: Sind $R_1, \ldots, R_n$ die Seiten, die auf S verweisen und verweist R_i auf m_i Seiten, so ist

$$w(S) = \sum_{i=1}^{n} \frac{w(R_i)}{m_i} . \qquad (*)$$

Dies ist sicherlich eine sinnvolle Forderung, jedoch ist *a priori* nicht klar, ob dadurch eine eindeutige Rangordnung definiert wird: Erst einmal muß untersucht werden, ob es überhaupt eine von der Nullfunktion verschiedene Lösungsfunktion w gibt, danach stellt sich noch das Problem der Eindeutigkeit.

Diese Frage läßt sich einfach beantworten, denn die obige Formel definiert offensichtlich ein lineares Gleichungssystem für die Unbekannten $w(S)$. Um es in eine üblichere Form zu bringen, bezeichnen wir die

der Suchmaschine bekannten Dokumente mit $S_1, \ldots, S_N$, die Anzahl der von Seite S_j ausgehenden Verweise mit m_j und setzen

$$a_{ij} = \begin{cases} \frac{1}{m_j} & \text{falls es einen Verweis } S_j \to S_i \text{ mit } j \neq i \text{ gibt} \\ 0 & \text{sonst} \end{cases} .$$

Dann wird die obige Gleichung zu

$$w_i = \sum_{j=1}^{N} a_{ij} w_j \quad \text{für } i = 1, \ldots, N .$$

Bringen wir hier noch w_i auf die andere Seite, haben wir die Standardform eines homogenen linearen Gleichungssystems:

$$\sum_{j=1}^{n} b_{ij} w_j = 0 \quad \text{mit} \quad b_{ij} = \begin{cases} a_{ij} & \text{falls } i \neq j \\ -1 & \text{falls } i = j \end{cases} .$$

Ein solches System hat immer den Nullvektor als Lösung; weitere Lösungen gibt es genau dann, wenn die Gleichungen linear abhängig sind.

Für eine Seite S_j, von der mindestens ein Verweis ausgeht, ist

$$\sum_{i=1}^{N} a_{ij} = m_j \cdot \frac{1}{m_j} = 1 \quad \text{und damit} \quad \sum_{i=1}^{N} b_{ij} = 0 ;$$

falls von jeder Seite mindestens ein Verweis ausgeht, ist also die Summe aller N linker Seiten gleich Null. In diesem Fall sind daher die Gleichungen linear abhängig und es gibt auch nichttriviale Lösungen.

Nun gibt es allerdings viele Seiten, die auf keine anderen Seiten verweisen, zum Beispiel die pdf-Datei mit dem Text dieses Skriptums. Um trotzdem die Existenz nichttrivialer Lösungen zu garantieren, werden vor allem zwei Strategien angewandt:

1. Man ignoriert zunächst alle Seiten, die auf keine anderen verweisen. Für den Rest löst man das lineare Gleichungssystem und setzt die Lösung dann mittels der Formel $(*)$ fort auf die restlichen Seiten. Da diese Seiten auf nichts verweisen, können sie nie auf der rechten Seite von $(*)$ auftreten; rechts stehen immer nur bereits aus dem ersten Schritt bekannte Gewichte.

2. Man behandelt diese Seiten so, als würden sie auf *jede* andere Seite verweisen, setzt also für eine solche Seite S_j den Wert von a_{ij} für jedes i auf $1/N$. Dann ist auch für solche j die Summe aller a_{ij} gleich eins, so daß obiges Argument die Existenz einer nichttrivialen Lösung zeigt.

Ein weiteres Problem sind Verweise auf nicht (mehr) existente oder einfach nur unzugängliche Seiten, z.B. solche mit Paßwortschutz oder Gebühr. Diese muß die Suchmaschine entweder ignorieren oder aber wie eine Seite ohne ausgehende Verweise behandeln.

Nachdem die Existenz nichttrivialer Lösungen geklärt ist, stellt sich als nächstes die Frage der Eindeutigkeit. Natürlich erfüllen mit jedem Lösungsvektor auch dessen sämtliche Vielfachen das Gleichungssystem; sofern es aber eine Lösung mit ausschließlich nichtnegativen Wichtigkeiten gibt, führen alle positiven Vielfachen davon auf dieselbe Rangordnung. Wir müssen also sicherstellen, daß der Lösungsraum erstens eindimensional ist und zweitens Vektoren ohne negative Komponenten enthält.

Leider kann die Dimension des Lösungsraums deutlich größer sein als eins: Wenn wir die Webseiten $S_1, \ldots, S_N$ einteilen können in zwei Klassen A und B mit der Eigenschaft, daß keine Seite aus A auf eine Seite aus B verweist und umgekehrt, sind die Gleichungen für die Seiten aus A und die für die Seiten aus B offensichtlich unabhängig voneinander und jedes der beiden Teilsysteme hat nach obiger Diskussion einen mindestens eindimensionales Lösungsraum, und läßt sich jede Lösung des ersten Teilsystems mit jeder Lösung des zweiten zu einer Lösung des Gesamtsystems kombinieren, so daß dessen Lösungsraum mindestens zweidimensional ist. Bei mehr als zwei Klassen, die nur innerhalb der eigenen Klasse zitieren, wird die Dimension noch größer, und für die Praxis am schlimmsten ist die Tatsache, daß uns das Gleichungssystem keinerlei Anhaltspunkte gibt, wie wir die relative Wichtigkeit der einzelnen Klassen festlegen sollen.

Aus diesem Grund muß Gleichung $(*)$ erweitert werden um einen Term, der die Existenz disjunkter Klassen verhindert. Die Idee dazu erklären BRIN und PAGE folgendermaßen:

Gleichung $(*)$ läßt sich auch stochastisch interpretieren: Angenommen, ein Surfer beginnt mit einer zufällig ausgewählten Webseite und klickt, sobald er sie auf dem Bildschirm hat, zufällig auf irgendeinen der dort gefundenen Verweise. Mit der nun erscheinenden Seite verfährt er genauso, und so weiter. Falls dieses Experiment hinreichend oft wiederholt und hinreichend lange durchgeführt wird, ergibt sich als Grenzwert eine Wahrscheinlichkeitsverteilung über alle Webseiten, d.h. wir können für jede Seite S die Wahrscheinlichkeit $p(S)$ ermitteln, daß der Surfer dort ankommt. Offensichtlich genügt auch die Funktion p der Gleichung $(*)$.

Nun wird das Modell etwas modifiziert: Der Surfer klickt nicht mehr unbedingt auf einen der Verweise der aktuellen Webseite, sondern nur mit einer gewissen Wahrscheinlichkeit α. Alternativ geht er mit Wahrscheinlichkeit $1-\alpha$ zu irgendeiner zufällig ausgewählten Webseite. Jetzt wird die Wahrscheinlichkeit für das Ankommen auf einer Seite S beschrieben durch eine Funktion, die der Rekursionsbedingung

$$p(S) = \alpha \sum_{j=1}^{n} \frac{p(R_j)}{m_j} + \frac{1-\alpha}{N} .$$

Laut BRIN und PAGE ist $\alpha \approx 0{,}85$ eine vernünftige Wahl.

Auch gemäß dieser Rekursionsvorschrift können wir wieder Wichtigkeiten $w(S)$ definieren, die einem linearen Gleichungssystem genügen: Sind $S_1, \ldots, S_N$ die sämtlichen Webseiten (wobei wir annehmen, daß entweder nur Webseiten berücksichtigt werden, die auf andere verweisen, oder aber, daß eine Webseite ohne externe Verweise so behandelt wird, als verwiese sie auf alle Webseiten) und soll S_i die Wichtigkeit w_i bekommen, so muß nun gelten

$$w_i = \alpha \sum_{j=1}^{N} a_{ij} w_j + \frac{1-\alpha}{N} ,$$

wobei die Koeffizienten a_{ij} wie oben definiert sind. Im Gegensatz zum dortigen Ansatz haben wir hier aber für $\alpha \neq 1$ ein inhomogenes lineares Gleichungssystem,

Dieses Gleichungssystem kann für $0 \leq \alpha < 1$ höchstens eine Lösung haben: Sind nämlich $(w_1, \ldots, w_N)$ und $(u_1, \ldots, u_N)$ beides Lösungs-

vektoren, so können wir einen Index i finden, für den $|w_i - u_i|$ maximal ist. Für dieses i ist dann

$$\begin{aligned}
|w_i - u_i| &= \left| \left(\alpha \sum_{j=1}^{N} a_{ij} w_j + \frac{1-\alpha}{N} \right) - \left(\alpha \sum_{j=1}^{N} a_{ij} u_j + \frac{1-\alpha}{N} \right) \right| \\
&= \left| \alpha \sum_{j=1}^{N} a_{ij} (w_j - u_j) \right| \leq \alpha \sum_{j=1}^{N} a_{ij} \left| w_j - u_j \right| \\
&\leq \alpha \sum_{j=1}^{N} a_{ij} \left| w_i - u_i \right| = \left(\alpha \sum_{j=1}^{N} a_{ij} \right) |w_i - u_i| \\
&= \alpha \left| w_i - u_i \right| \text{ , denn } \quad \sum_{j=1}^{N} a_{ij} = 1 \, .
\end{aligned}$$

Das ist aber nur möglich, wenn $|w_i - u_i|$ verschwindet und damit, wegen dessen Maximaleigenschaft, auch alle anderen Differenzen $w_j - u_j$. Dies zeigt, daß die beiden Lösungen übereinstimmen,

Noch nicht gezeigt ist die *Existenz* einer Lösung, aber jeder, der mit dem BANACHschen Fixpunktsatz vertraut ist, wird wohl wissen, wie es nun weitergeht: Wir starten mit irgendeinem N-tupel $(w_1^{(0)}, \ldots, w_N^{(0)})$ positiver Zahlen und konstruieren dazu sukzessive neue N-tupel gemäß der Vorschrift

$$w_i^{(k+1)} = \alpha \sum_{j=1}^{N} a_{ij} w_j^{(k)} + \frac{1-\alpha}{N} \, .$$

Falls das Tupel $(w_1^{(k)}, \ldots, w_N^{(k)})$ eine Lösung ist, stimmt es natürlich mit seinem Nachfolger $(w_1^{(k+1)}, \ldots, w_N^{(k+1)})$ überein, aber das können wir nicht realistischerweise erwarten.

Als Maß der Abweichung zwischen den beiden Tupeln betrachten wir wie oben einem Index i, für den $\left| w_i^{(k+1)} - w_i^{(k)} \right|$ maximal wird. Für

$k \geq 1$ ist dann nach einer völlig analogen Rechnung

$$\begin{aligned}
\left| w_i^{(k+1)} - w_i^{(k)} \right| &= \left| \left(\alpha \sum_{j=1}^{N} a_{ij} w_j^{(k)} + \frac{1-\alpha}{N} \right) - \left(\alpha \sum_{j=1}^{N} a_{ij} w_j^{(k-1)} + \frac{1-\alpha}{N} \right) \right| \\
&= \left| \alpha \sum_{j=1}^{N} a_{ij} (w_j^{(k)} - w_j^{(k-1)}) \right| \leq \alpha \sum_{j=1}^{N} a_{ij} \left| w_j^{(k)} - w_j^{(k-1)} \right| \\
&\leq \alpha \sum_{j=1}^{N} a_{ij} \left| w_i^{(k)} - w_i^{(k-1)} \right| = \left(\alpha \sum_{j=1}^{N} a_{ij} \right) \left| w_i^{(k)} - w_i^{(k-1)} \right| \\
&= \alpha \left| w_i^{(k)} - w_i^{(k-1)} \right| \text{ , denn } \quad \sum_{j=1}^{N} a_{ij} = 1 \, .
\end{aligned}$$

Da wir $\alpha < 1$ vorausgesetzt haben, werden die Abweichungen also immer kleiner, und im Limes erhalten wir eine Lösung.

Damit ist bewiesen, daß es genau eine Lösung gibt, und nach dem, was wir in der Linearen Algebra gelernt haben, können wir diese mit dem GAUSS-Algorithmus bestimmen.

Es gibt allerdings einen wesentlichen Unterschied zwischen dem hier zu lösenden Gleichungssystem und den aus Übungsblättern und Klausuren bekannten: Zwar geht es in beiden Fällen (meist) um N Gleichungen in N Unbekannten, aber im Studium ist N selten mehr als vier, während es bei Google im Augenblick bei etwas über 80 Milliarden liegt.

Um den GAUSS-Algorithmus für ein Gleichungssystem aus N Gleichungen mit N Unbekannten durchzuführen, braucht man asymptotisch etwa N^3 Rechenoperationen. Bei $N \approx 8 \cdot 10^9$ sind das ungefähr $2^9 \cdot 10^{27}$, mit der Näherung $2^{10} \approx 10^3$ also ungefähr 2^{99}.

Bei der Beurteilung der Sicherheit elektronischer Unterschriften geht das Bundesamt für Sicherheit in der Informationstechnik derzeit aus von einem Sicherheitsniveau 2^{100}; ein Verfahren gilt somit als sicher, wenn anzunehmen ist, daß ein Gegner mindestens 2^{100} Versuche benötigt, um

das Verfahren zu knacken. Diese „Versuche“ sind zwar etwas komplexer als einfache Rechenoperationen; andererseits muß man aber bei der Beurteilung der Sicherheit von Kryptoverfahren auch Gegner berücksichtigen, die einen Rechenaufwand von einem Jahr oder gar mehr nicht scheuen, was weit jenseits dessen liegt, was für die periodisch zu aktualisierende Rangfolge der Webseiten liegt. Daher können wir davon ausgehen, daß 2^{99} Rechenoperationen zumindest für diese Aufgabe derzeit nicht im Bereich des Realisierbaren liegen.

Andererseits sind wir hier auch nicht im Bereich der Reinen Mathematik, sondern es geht um eine Anwendung der Mathematik auf reale Probleme. Dabei müssen wir uns gerade bei so einem Thema auch stets bewußt sein, daß unsere Modelle mit ziemlicher Sicherheit nur eine Approximation an die Wirklichkeit sind – falls es hier überhaupt irgendeine „Wirklichkeit“ geben sollte.

Von daher wäre es Unsinn, mit riesigem Aufwand ein Problem, das bestenfalls eine grobe Approximation an eine vielleicht gar nicht vorhandene Wirklichkeit beschreibt, mathematisch exakt zu lösen: Eine approximative Lösung reicht vollkommen.

Diese wiederum bietet uns gerade der theoretische Ansatz, mit dem wir die Existenz einer Lösung bewiesen haben: Die dazu verwendete Iteration gestattet uns schließlich eine beliebig genaue Annäherung an die Lösung. Da wir von acht Milliarden Gleichungen in genauso vielen Unbekannten ausgehen, ist schließlich auch die Iterationsvorschrift keine Aufgabe für das Rechnen mit Bleistift und Papier: Um die Gleichung

$$w_i^{(k+1)} = \alpha \sum_{j=1}^{N} a_{ij} w_j(k) + \frac{1-\alpha}{N}$$

für alle i auszuwerten, brauchen wir größenordnungsmäßig N^2 Rechenoperationen *je Iteration.*

Hier hilft uns eine praktische Beobachtung, die wohl keinen Surfer im World Wide Web erstaunen dürfte: Bekanntlich ist $a_{ij} = 0$, wenn die j-te Webseite nicht auf die i-te verweist, und natürlich gibt es kaum Webseiten, die auch nur auf einen Bruchteil aller vorhandener Webseiten verweisen. Experimentelle Untersuchungen zeigen, daß eine Webseite

im Durchschnitt nur sieben externe Verweise hat. In der obigen Summe über acht Milliarden Summanden sind also im Durchschnitt nur sieben von Null verschieden, wir brauchen also tatsächlich nur etwa $7N$ Additionen und Multiplikationen. Damit sind wir wieder im Bereich der für die langfristige Existenz von Google so wichtigen Skalierbarkeit: Die Zahl N wird natürlich im Laufe der Jahre ziemlich ansteigen, aber zumindest nach bisheriger Erfahrung wird die Rechenkraft pro Dollar (oder Euro) ungefähr im gleichen Maße steigen. Bei der Anzahl sieben für den Durchschnitt für die Verweise auf andere Webseiten sind zumindest mittelfristig keine wesentlichen Änderungen zu erwarten: Die Erzeuger von Webseiten werden wohl auch in Zukunft nicht wesentlich mehr Verweise auf ihren Seiten anbringen, der Aufwand pro Iteration bleibt also ungefähr derselbe, wenn auch künftig der Umfang des World Wide Web im selben Maße ansteigt wie die Rechenkraft der Computer einer festen (inflationsbereinigten) Preisklasse.

Was die Anzahl der Iterationen betrifft, die für ein vorgegebenes Genauigkeitsniveau notwendig sind, sagt uns die Summenformel für geometrische Reihen, daß es nicht auf die Anzahl der Webseiten ankommt: Sei $(w_1, \dots, w_N)$ die Lösung des linearen Gleichungssystems, $(w_1^{(k)}, \dots, w_N^{(k)})$ die k-te Iteration und i derjenige Index, für den $\left|w_i - w_i^{(k)}\right|$ maximal ist. Dann ist

$$\left|w_i - w_i^{(k)}\right| = \left|\sum_{\ell=k}^{\infty} (w_i^{(\ell+1)} - w_i^{(\ell)}\right| \le \sum_{\ell=k}^{\infty} \left|w_i^{(\ell+1)} - w_i^{(\ell)}\right|$$

$$\le \sum_{\ell=0}^{\infty} \alpha^{\ell} \left|w_i^{(k+1)} - w_i^{(k)}\right| = \frac{\left|w_i^{(k+1)} - w_i^{(k)}\right|}{1-\alpha}$$

nach der Summenformel für die geometrische Reihe. Wir können daher nach jedem Iterationsschritt abschätzen, wie groß der maximale Fehler ist und abbrechen, sobald dieser eine akzeptable Größenordnung erreicht hat.

In der Arbeit von BRIN und PAGE von 1998

> SERGEY BRIN, LAWRENCE PAGE: *The Anatomy of a Large-Scale Hypertextual Web Search Engine,* Computer networks and ISDN

systems, 1998, Elsevier;
http://db.stanford.edu/pub/public_html/papers/google.pdf

ist davon die Rede, daß die Berechnung für das damals untersuchte Netz von 26 Millionen Seiten auf einer (nach damaligen Standards) mittelgroßen *workstation* einige Stunden dauerte. In

LAWRENCE PAGE, SERGEY BRIN, RAJEEV MOTWANI, TERRY WINOGRAD: *The PageRank Citation Ranking: Bringing Order to the Web,*Technical Report. Stanford InfoLab 1999,
http://ilpubs.stanford.edu:8090/422/

wird das Konvergenzverhalten genauer untersucht und gezeigt, daß für ein Web mit 322 Millionen Links etwa 45 Iterationen ausreichen. Neuere Zahlen werden von Google nicht veröffentlicht; nach externen Schätzungen soll Google einige Tage benötigen, um den PageRank komplett neu zu berechnen.

Die Werte für die Wichtigkeiten können beträchtlich schwanken: der minimale Wert ist offensichtlich gleich $1-d$, also für $d = 0{,}85$ gleich 0,15; die theoretische Obergrenze liegt bei dN, also im Milliardenbereich. Google zerteilt diesen Bereich in elf Teilintervalle, denen die PageRanks null bis zehn zugeordnet werden. Über die Definition dieser Intervalle ist nichts bekannt, allerdings wird vermutet, daß die Intervallängen ungefähr in einer geometrischen Progression ansteigen, so daß der PageRank ungefähr gleich einem Logarithmus der gerade bestimmten Wichtigkeit ist, dessen Basis in der Gegend von sechs oder sieben liegen dürfte ($6^{10} = 60\,466\,176$ und $7^{10} = 282\,475\,249$). Auch wenn für interne Berechnungen die exakten Werte der w_i verwendet werden, veröffentlicht Google nur die Grobwerte – wahrscheinlich auch dies wieder, um Suchmaschinenoptimierern nicht zuviel Information an die Hand zu geben.

§6: Der HITS-Algorithmus

Zur gleichen Zeit, als BRIN und PAGE in Stanford im Rahmen ihres Dissertationsprojekts den PageRank-Algorithmus entwickelte, war rund zwanzig Kilometer südlich JON KLEINBERG von der Cornell University

als *visiting professor* am IBM Almaden Research Center bei San José und befaßte sich ebenfalls mit dem Problem, Webseiten nach Wichtigkeit und Relevanz zu ordnen. Sein Ansatz war etwas komplizierter:

Nach dem Ansatz von BRIN und PAGE gibt eine Webseite mit m Verweisen an jede der aufgeführten Webseiten ein m-tel ihrer eigenen Wichtigkeit weiter, bei großen Werten von m also fast nichts.

Im Falle einer Seite, die wahllos so ziemlich alles zitiert, ist dies sicherlich sinnvoll; gerade damals gab es aber auch noch eine ganze Reihe von Webseiten, auf denen ein oder mehrere Autoren mit großer Mühe eine Sammlung von Referenzen für ein bestimmtes Thema zusammengestellt hatten, teilweise sogar mit Kommentaren zu den einzelnen Seiten. Wenn eine solche Seite gut gemacht ist, verweist sie auf wichtige Seiten, und das sollte bei *deren* Beurteilung auch gebührend gewürdigt werden.

In seiner Arbeit

JON M. KLEINBERG: *Authoritative sources in a hyperlinked environment*, Journal of the ACM Volume 46 Issue 5, Sept. 1999
http://portal.acm.org/citation.cfm?doid=324133.324140

betrachtet er Webseiten deshalb unter den beiden Gesichtspunkten *hub* und *authority*.

Das englische Wort *hub* hat viele deutsche Übersetzungen; unter anderem bezeichnet es im Luftverkehr ein Drehkreuz, d.h. einen Flughafen, über den eine Gesellschaft beispielsweise die Passagiere ihrer Fernflüge auf die Anschlußflüge zu den weniger zentralen Zielen verteilt, und entsprechend auch die Verteilzentren von Logistikunternehmen. Außerdem steht das Wort für die Nabe eines Rads (von der nach allen Richtungen die Speichen ausgehen), und nicht zuletzt bezeichnet sich auch KLEINBERGs Geburtsstadt Boston als *hub of the universe,* frei übersetzt also als Nabel der Welt.

Auch das Wort *authority* hat viele mögliche Übersetzungen, unter anderem Behörde, Berechtigung, Befehlsgewalt, Ermächtigung, Obrigkeit, Vollmacht; was KLEINBERG meint sind allerdings die alternativen Bedeutungen im Sinne von Fachmann oder Fachkompetenz.

Die Idee ist also, daß *hubs* zentrale Anlaufstellen sind, die auf *authorities* verweisen, die etwas zu einem bestimmten Thema zu sagen haben. Das Prinzip, nach dem er Webseiten ordnet, faßt er in der zitierten Arbeit so zusammen:

> A good *hub* is a page that points to many good authorities; a good *authority* is a page pointed to by many good hubs.

Wie bei den Maximen für den PageRank ist auch diese Definition zirkulär, und wie dort kann die Zirkularität mit Hilfe der Linearen Algebra leicht aufgelöst werden:

KLEINBERG ordnet jeder der Seite S_i *zwei* Gewichte zu: Das *authority*-Gewicht x_i und das *hub*-Gewicht y_i; sie sind so normalisiert, daß der Vektor x mit Komponenten x_i und der Vektor y mit Komponenten y_i jeweils die (EUKLIDische) Länge eins haben. Die obige Maxime wird im wesentlichen so umgesetzt, daß die *authority*-Wichtigkeit einer Seite proportional zur Summe der *hub*-Wichtigkeiten aller darauf verweisender Seiten ist, wohingegen die *hub*-Wichtigkeit proportional zur Summe der *authority*-Wichtigkeiten jener Seiten ist, auf die sie verweist. Die Proportionalitätskonstanten sind jeweils dadurch bestimmt, daß sowohl x als auch y Einheitsvektoren mit nichtnegativen Einträgen sind.

Bezeichnen wir mit A die Matrix mit Einträgen

$$a_{ij} = \begin{cases} 1 & \text{falls es einen Verweis } S_j \to S_i \text{ mit } j \neq i \text{ gibt} \\ 0 & \text{sonst} \end{cases},$$

soll es also Konstanten $\alpha, \beta \in \mathbb{R}_{\geq 0}$ geben, so daß

$$y = \alpha A x \quad \text{und} \quad x = \beta A^T y$$

ist, d.h.

$$x = \alpha\beta A^T A x \quad \text{und} \quad y = \alpha\beta A A^T y\,.$$

Somit ist x ein Eigenvektor von $A^T A$ und y einer von AA^T, beide zum selben Eigenwert.

Tatsächlich geht KLEINBERG nicht von dieser Charakterisierung der beiden Vektoren x und y aus, sondern gibt stattdessen eine Methode zur

Konstruktion der beiden Vektoren an: Er startet mit zwei beliebigen Vektoren $x^{(0)}, y^{(0)}$ mit nichtnegativen Einträgen und setzt sukzessive

$$x^{(k)} = A^T y^{(k-1)} \quad \text{und} \quad y^{(k)} = A x^{(k)}\,.$$

Nach einer gewissen Anzahl von Iterationen, wenn sich die Richtungen von $x^{(k)}$ und $x^{(k-1)}$ sowie die von $y^{(k)}$ und $y^{(k-1)}$ nicht mehr wesentlich voneinander unterscheiden, nimmt er für x den Einheitsvektor in Richtung $x^{(k)}$ und für y den in Richtung $y^{(k)}$.

Dieses Vorgehen erinnert an die Berechnungsmethode für den PageRank. Um zu sehen, ob und wohin KLEINBERGs Folgen konvergieren, beachten wir, daß

$$x^{(k)} = A^T A x^{(k-1)} \quad \text{und} \quad y^{(k)} = A A^T y^{(k-1)}$$

ist; wir müssen uns also nur überlegen, wohin für eine symmetrische Matrix M die durch $z^{(k)} = M z^{(k-1)}$ definierte Folge für einen gegebenen Anfangsvektor $z^{(0)}$ konvergiert.

Wie wir wissen, gibt es zu einer symmetrischen Matrix eine Basis aus Einheitsvektoren, bezüglich derer sie Diagonalgestalt hat; durch Umordnen erhalten wir eine Basis $(b^{(1)}, \ldots, b^{(N)})$, bezüglich derer M Diagonalgestalt hat mit Diagonaleinträgen $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. Für $z^{(0)} = z_1 b^{(1)} + \ldots + z_N b^{(N)}$ mit $z_i > 0$ ist dann

$$z^{(k)} = \lambda_1^k z_1 b^{(1)} + \ldots + \lambda_N^k z_n b^{(N)}\,.$$

Für alle $\lambda_i < \lambda_1$ geht der Quotient $\lambda_i^k / \lambda_1^k$ gegen null; im Einheitsvektor zu $z^{(k)}$ kommen daher für große Werte von k praktisch nur noch die Koeffizienten vor, die mit λ_1^k multipliziert wurden. Ist $\lambda_1 > \lambda_2$, bekommen wir somit als Ergebnis nach Normalisierung den Eigenvektor $b^{(1)}$ zum größten Eigenwert λ_1; ansonsten erhalten wir einen von $z^{(0)}$ abhängigen Vektor aus dem Eigenraum zum Eigenwert λ_1. Wie im Falle von PageRank kann man letzteres ausschließen, indem man die Matrix M ersetzt durch eine konvexe Linearkombination mit der Matrix, deren sämtliche Einträge eins sind, allerdings zeigen experimentelle Untersuchungen, daß man auch ohne diese Maßnahme auskommt.

Das Verfahren von KLEINBERG unterscheidet sich noch in einem zweiten Punkt von PageRank: Dort werden bekanntlich sämtliche von den

Crawlern gefundene Webseiten unabhängig von jeder Suchanfrage global nach ihrer Wichtigkeit geordnet. KLEINBERG dagegen betrachtet nur einen Ausschnitt des Webs: In der oben zitierten Originalarbeit schlägt er vor, beispielsweise mit den ersten zweihundert Ergebnissen der damals populären Suchmaschine Altavista zu starten; in heutigen Darstellungen ist die Rede davon, mit allen bekannten Seiten anzufangen, die die Suchbegriffe enthalten.

In einem nächsten Schritt wird diese Ausgangsmenge erweitert um alle Webseiten, die entweder einen Verweis auf eine der betrachteten Seiten enthalten oder aber Ziel eines Links von einer derartigen Seite sind. (Dieses Verfahren kann man gegebenenfalls noch ein oder mehrere Male wiederholen.) Danach wird die Matrix A für das so erhaltene Teilnetz aufgestellt, und nur für dessen Seiten werden die *hub*- und *authority*-Wichtigkeiten aufgestellt. Man beachte, daß durch die Erweiterung der ursprünglichen Menge von Webseiten auch Seiten einbezogen werden und möglicherweise sogar hohe Wichtigkeiten bekommen, die überhaupt keinen der Suchbegriffe enthalten. Auf diese Weise erreicht der Algorithmus auch ohne Untersuchung der Term-Dokument-Matrix eine Art latente semantische Analyse.

§ 7: Die Gewichte der Terme

Kehren wir zurück zur Term-Dokument-Matrix. Ihr Eintrag a_{ij} soll eine Art Gewicht des i-ten Term im j-ten Dokument sein. Im einfachsten Fall nimmt a_{ij} nur die beiden Werte 0 und 1 an, je nachdem ob der der Begriff im Dokument vorkommt oder nicht; eine andere offensichtliche Lösung wäre, daß a_{ij} zählt, wie oft der Begriff auftritt. Beides ist zwar einfach, führt aber auch zu offensichtlichen Problemen: Nicht jedes Wort, das irgendwo in einem Dokument vorkommt, läßt auf den Inhalt des Dokuments schließen, und wenn ein Wort sehr häufig vorkommt, kann das auch einfach nur bedeuten, daß das Dokument entweder sehr lang oder sehr geschwätzig ist.

Seit es Textdatenbanken gibt werden daher auch kompliziertere Schemata diskutiert und immer weiter verfeinert; Google etwa benutzt nach

eigenen Angaben rund zweihundert sogenannte „Signale“, um die Relevanz eines Dokuments für eine Suchanfrage zu bestimmen.

Schon bei deutlich einfacheren Vorgehensweisen empfiehlt es sich, zwischen der *lokalen* und der *globalen* Wichtigkeit eines Begriffs zu unterscheiden. Dabei soll die lokale Wichtigkeit messen, welche relative Bedeutung ein Begriff innerhalb eines speziellen Dokuments hat, während die globale Wichtigkeit angibt, wie wichtig der Begriff für die gesamte Dokumentensammlung ist. Der Eintrag in der Term-Dokument-Matrix ist das Produkt der beiden Wichtigkeiten, wobei anschließend eventuell noch alle Spalten in geeigneter Weise normalisiert werden.

Beginnen wir mit der lokalen Wichtigkeit. Die beiden einfachsten Schemata wurden bereits erwähnt: Wir setzten die lokale Wichtigkeit ℓ_{ij} des i-ten Begriffs für das j-te Dokument entweder nur auf 0 oder 1, je nachdem ob der Begriff im Dokument vorkommt, oder aber wir setzten ℓ_{ij} auf die Anzahl f_{ij} der Vorkommen des Begriffs im Dokument. Um die damit verbundene Bevorzugung langer Dokumente abzumildern, wird teilweise auch der Logarithmus verwendet; da dieser für den Wert null nicht definiert ist, setzt man hier

$$\ell_{ij} = \log(1 + f_{ij}) .$$

Um völlig unabhängig von der Dokumentlänge zu werden, kann man auch die *durchschnittliche* Häufigkeit f_j der Terme im j-ten Dokument betrachten: Ist u_j die Anzahl verschiedener Terme im Dokument, so setzten wir

$$f_j = \frac{1}{u_j} \sum_{i=1}^{m} f_{ij} \quad \text{und} \quad \ell_{ij} = \frac{\log(1 + f_{ij})}{\log(1 + f_j)} .$$

Hier ist $\ell_{ij} = 1$ für jeden Begriff, der genau die mittlere Häufigkeit hat: kommt der Term überdurchschnittlich oft vor, ist $\ell_{ij} > 1$, ansonsten kleiner.

Ein Kompromiss zwischen bloßem Vorkommen und (relativer) Häufigkeit ist die bereits von SALTON vorgeschlagene vergrößerte normalisierte Häufigkeit

$$\ell_{ij} = \frac{1}{2} \left(\chi(f_{ij}) + \frac{f_{ij}}{\max_\nu f_{\nu j}} \right) \quad \text{mit} \quad \chi(x) = \begin{cases} 1 & \text{falls } x \neq 0 \\ 0 & \text{falls } x = 0 \end{cases} .$$

Die globale Wichtigkeit g_i hängt nur vom Suchbegriff ab. Durch sie soll berücksichtigt werden, daß eher seltene Begriffe meist deutlich spezifischer sind als Allerweltsbegriffe, die in praktisch jedem Dokument vorkommen. Das extremste Beispiel dafür sind die bereits erwähnten *Nullen* auf der Stopliste, die überhaupt nicht berücksichtigt werden; im Rahmen der jetzigen Betrachtungsweise können wir sie definieren als die Wörter mit $g_i = 0$.

Ein Begriff ist unter dem Gesichtspunkt der Informationssuche umso spezifischer, je ungleichmäßiger er über die Dokumente verteilt ist. Da die SHANNONsche Entropie derartige Ungleichmäßigkeiten quantifiziert, liegt es nahe, sie auch für die Definition einer globalen Wichtigkeit einzusetzen. Wir betrachten alle Vorkommen des i-ten Begriffs in den n Dokumenten der Sammlung und definieren als

$$p_{ij} = \frac{f_{ij}}{\sum_{j=1}^{n} f_{ij}}$$

den Anteil dieser Vorkommen im j-ten Dokument. Die Summe

$$-\sum_{j=1}^{n} p_{ij} \log p_{ij}$$

hat ihren maximalen Wert $\log n$, wenn der Begriff in jedem Dokument gleich häufig auftritt; den minimalen Wert Null nimmt sie an, wenn er nur in einem einzigen Dokument vorkommt. Damit bietet sich

$$g_i = 1 + \frac{\sum_{j=1}^{n} p_{ij} \log p_{ij}}{\log n}$$

als eine Möglichkeit zur Definition der globalen Wichtigkeit an: Im Falle der gleichmäßigen Verteilung erhalten wir den Wert Null, für einen Begriff, der nur in einem einzigen Dokument vorkommt dagegen den maximal möglichen Wert eins.

Für ein einfacheres Maß können wir auch einfach nur zählen, in wie vielen Dokumenten der Begriff vorkommt. Ist n_i diese Anzahl, so ist

$$g_i = \log \frac{n}{n_i}$$

gleich Null für einen Begriff, der in jedem Dokument vorkommt, wohingegen der Maximalwert $\log n$ angenommen wird. falls das Wort nur in einem Dokument steht. Diese sogenannte *inverse Dokumenthäufigkeit* wird vor allem gerne eingesetzt für Sammlungen, deren Inhalt sich nicht allzu häufig ändert. Alternativ wird auch gelegentlich das sogenannte probabilistische Inverse

$$g_i = \log \frac{n - n_i}{n_i}$$

verwendet, das die Anzahlen von Dokumenten mit *bzw.* ohne den Begriff zueinander in Beziehung setzt.

Eine völlig andere Strategie besteht darin, die Wichtigkeit eines Begriffs danach zu beurteilen, wie oft er in den Dokumenten auftritt, in denen er überhaupt vorkommt; damit hätten wir also

$$g_i = \frac{1}{n} \sum_{j=1}^{n} f_{ij} \,.$$

Dieses Maß ist offensichtlich nur sinnvoll, wenn Nullen vorher ausgeschlossen wurden, denn es würde auch beispielsweise für bestimmte Artikel eine hohe Wichtigkeit liefern.

Möchte man die globalen Wichtigkeiten nach Seltenheit des Suchbegriffs festlegen, bietet sich auch an, den Vektor $(f_{i1}, \ldots, f_{in}) \in \mathbb{R}^n$ der Anzahlen zu betrachten; da seltene Begriffe kurzen Vektoren entsprechen, kann die globale Wichtigkeit als Kehrwert

$$g_i = \frac{1}{\sqrt{\sum_{j=1}^{n} f_{ij}^2}}$$

der EUKLIDischen Länge definiert werden.

Durch Multiplikation der lokalen und globalen Wichtigkeiten erhält man erhält man ein Maß für die Relevanz des i-ten Terms im j-ten Dokument. Bei den meisten Wahlen erhält man dabei Werte, die lange Dokumente gleich in zweierlei Hinsicht bevorzugen: Einmal stehen in einem langen Dokument im allgemeinen mehr verschiedene Begriffe; die Wahrscheinlichkeit, daß ein Begriff aus der Suchanfrage überhaupt vorkommt, ist

also größer. Zum andern wird ein fester Begriff, so er uberhaupt vorkommt, in einem langen Dokument meist häufiger vorkommen als in einem kurzen.

Diesen Effekt kann man durch Normalisierung abmildern oder sogar aufheben. Wir kennen bereits die häufig angewendete Strategie, den Kosinus des Winkels zwischen dem Spaltenvektor des Dokuments und dem Anfragevektor als Ähnlichkeitsmaß zu verwenden; dies entspricht der Normierung der Spalten auf EUKLIDische Länge eins und der Skalarproduktbildung zur Berechnung der Relevanz. Wie experimentelle Untersuchungen zeigen, führt diese Strategie zu einer Bevorzugung *kurzer* Dokumente. Auch der Vergleich mit Mittel- oder Maximalwerten kann zur Normierung eingesetzt werden – ein Beispiel haben wir bereits oben bei den lokalen Gewichten betrachtet.

In der Arbeit

> AMIT SINGHAL. CHRIS BUCKLEY, MANDAR MITRA: *Pivoted Document Length Normalization*, Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, 1976
> `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.9950&rep=rep1&type=pdf`

wird untersucht, wie für fünfzig Suchanfragen an eine Datenbank mit 741 856 Dokumenten einerseits, wie jeweils einerseits die Anzahl der relevanten Dokumente und andererseits die der gefundenen Dokumente von der Dokumentlänge abhängt. Als Ergebnis erhielten die Autoren zwei Kurven, die sich in einem bestimmten Punkt schneiden; vor diesem Punkt liegt die eine Kurve oben, danach die andere.

Idealerweise sollten natürlich beide Kurven übereinstimmen; die Übereinstimmung kann verbessert werden, indem durch geeignete Renormierung die Kurve der gefundenen Dokumente um den Schnittpunkt so gedreht wird, daß die Tangenten beider Kurven dort übereinstimmen. Dazu werden verschiedene Ansätze diskutiert, beispielsweise die Definition

$$a_{ij} = \frac{(1+\log f_{ij})/(1+log\overline{f}_j)}{(1-s)p+su_j},$$

wobei wieder $\overline{f}_j$ die mittlere Anzahl der Vorkommen eines Worts im j-ten Dokument ist, p ist die mittlere Anzahl verschiedener Begriffe pro Dokument und u_j die Anzahl verschiedener Wörter in Dokument j. Der *Slope* s definiert den Winkel, um den gedreht wird, z.B. $s \approx 0{,}2$. Damit erhielten sie um 13,7% bessere Ergebnisse als mit der üblichen Kosinusstrategie.

Natürlich sind die hier vorgestellten Maße nur ein kleiner Ausschnitt aus der Vielfalt aller denkbarer Möglichkeiten, und es sind vor allem die in der akademischen Welt diskutierten. Die kommerziell erfolgreichen Suchmaschinen und sonstigen Textverwaltungssysteme dürften wohl deutlich kompliziertere Maße verwenden, deren Einzelheiten sie aus gutem Grund für sich behalten. Publizierte experimentelle Tests gibt es im wesentlichen nur für kleine Systeme; ihre Ergebnisse lassen sich nur sehr bedingt auf große Zeitschriftendatenbanken oder gar das gesamte World Wide Web übertragen – insbesondere da im letzteren mittlerweile viele Webseitenoptimierer damit beschäftigt sind, Rangfolgen zu analysieren und die Seiten ihrer Kunden nach vorne zu bringen. Bevor das Problem der optimalen Gewichtung wirklich verstanden ist, sind sicherlich noch viele theoretische wie auch experimentelle Studien notwendig.

§8: Mehr über Matrixzerlegungen

Wir haben bislang nur die Singulärwertzerlegung einer Matrix betrachtet und gesehen, daß wir mit ihrer Hilfe die im Sinne der FROBENIUS-Norm nächstgelegene Matrix finden können, deren Rang eine vorgegebene Schranke nicht überschreitet; dies war der Ausgangspunkt zur latenten semantischen Analyse.

Sowohl in der theoretischen Literatur als auch in praktisch implementierten Systemen werden daneben noch eine ganze Reihe weiterer Zerlegungen diskutiert *bzw.* angewendet, die teils einfacher zu berechnen sind (die Singulärwertzerlegung der Term-Dokument-Matrix des gesamten *World Wide Web* ist mit den heute zur Verfügung stehenden Computern und Algorithmen definitiv nicht berechenbar), teils auch theoretische