PageRank: Wie Google funktioniert

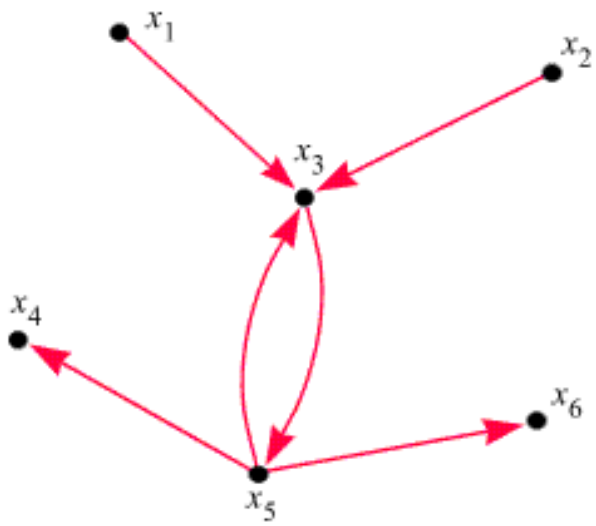
Außermathematische Anwendungen im Mathematikunterricht
WS 2014/15
Franz Embacher, Universität Wien

Das Erfolgsrezept der Suchmaschine von Google…

…lag zunächst in der überzeugenden Reihung von Treffern. Gibt ein User Suchbegriffe wie „Quantencomputer“ oder „Spezielle Relativitätstheorie“ ein, so findet Google im ersten Fall 15100, im zweiten Fall 84100 Seiten, die diese Begriffe enthalten. Die Reihung, in der die Treffer angeboten werden, entspricht in den meisten Fällen den Erwartungen der User: Die „relevantesten“, „nützlichsten“ Seiten werden zuerst angezeigt, so dass es oft gar nicht nötig ist, sich mehr als die ersten paar anzusehen. Bei Suchmaschinen der 1990er-Jahre konnte es leicht passieren, dass sich die gesuchte Seite irgendwo weit hinten in der Liste der Treffer befand. Das Verhalten der Suchmaschine von Google revolutionierte die Art und Weise, in der Suchmaschinen benutzt werden konnten. Zugrunde liegt ihm ein Algorithmus namens PageRank, der von Larry Page (daher der Name) und Sergei Brin entwickelt und im Jahr 1998 veröffentlicht wurde.

Das WWW als gerichteter Graph

PageRank setzt an der **Verlinkungsstruktur** von Webseiten an. Rechts ist ein Mini-WWW abgebildet: Es besteht aus 6 Seiten. Jeder Pfeil stellt einen Link dar, der von einer Seite auf eine andere Seite verweist. Das ist die einzige Struktur, von der PageRank Gebrauch macht. Sie kann als **gerichteter Graph** aufgefasst werden: Jeder Knoten (im Beispiel rechts sind das die Punkte x_1, x_2,… x_6) stellt eine Webseite dar, und jede gerichtete Kante (d.h. jede rote Linie mit Pfeilsymbol) einen Link. Der „Webcrawler“ von Google besucht regelmäßig die ihm bekannten Webseiten, registriert die von ihnen ausgehenden Links und folgt ihnen, um die Seiten, auf die verwiesen wird, ebenfalls zu registrieren. Auf diese Weise bildet Google die Verlinkungsstruktur großer Teile des WWW ab. Woher weiß die Suchmaschine, welche Seiten die „relevantesten“ und „nützlichsten“ sind?



PageRank

Die Ausgangsidee von PageRank besteht darin, Seiten, auf die viele Links verweisen, als die populärsten, relevantesten und nützlichen anzusehen. So einfach geht das aber nicht, denn dann könnte man ja auf einer Seite hunderte von Links versammeln, die auf jene Seiten verlinken, die man gerne ganz oben aufgelistet haben möchte. Also bekommt jede Seite sozusagen nur ein „**Stimmgewicht**“ von 1, das gerecht auf alle ausgehenden Links aufgeteilt wird. Gehen von einer Webseite n Links aus, so bekommt jeder das Stimmgewicht $1/n$. Damit kann die Verlinkungsstruktur des WWW als „Abstimmung“ aufgefasst werden. Dabei könnten wir jeder Seite einen „Score“ zuweisen, der gleich der Summe der Stimmgewichte aller Links ist, die auf sie zeigen. Die Reihung aller Treffer einer Suchanfrage würde dann entsprechend dieses Scores vorgenommen werden.

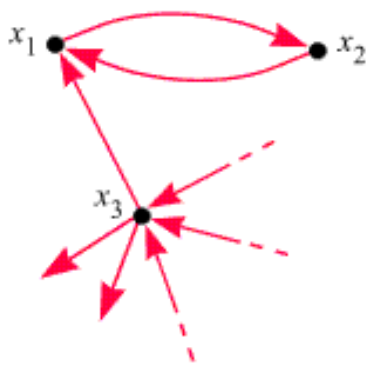
Das hätte jedoch den Effekt, dass man den Score einer Seite bewusst verbessern kann, indem viele weitere Seiten angelegt werden, die alle auf sie verlinken. Um dies zu verhindern, werden die von einer Seite ausgehenden „Stimmen“ ein **weiteres Mal gewichtet**, und zwar mit dem Score, die die Seite bekommt! Eine schöne Zirkeldefinition! Wie viel eine Seite im Abstimmungspoker zu den Scores anderer Seiten beiträgt, hängt von ihrem *eigenen* Score ab: **„Eine Webseite ist relevant, wenn viele relevante Seiten auf sie verweisen“**. Mathematisch gesehen ist dieses Konzept aber kein Problem – es wird einfach als Gleichungssystem formuliert. Das geht ganz einfach:

Wir wollen den Score, den eine Seite x bekommt, als $R(x)$ bezeichnen. Sei nun y eine Seite, die auf x verweist. Sie besitzt den Score $R(y)$ und enthält $C(y)$ Links, von denen einer auf x verweist. (Falls mehrere Links von y auf x verweisen, so werden sie nur als ein Link gezählt). Dieser eine Link bekommt also ein Stimmgewicht von $1/C(y)$, und gewichtet mit dem Score von y ergibt sich, dass die Seite y einen Beitrag $R(y)/C(y)$ zum Score von x als „Stimme“ abgibt. Auf diese Weise erhalten wir die Formel

$$R(x) = \sum_{\substack{\text{Seiten } y, \text{ die auf} \\ x \text{ verweisen}}} \frac{R(y)}{C(y)} \qquad \text{für alle } x \in \text{WWW}. \tag{1}$$

Ein lineares Gleichungssystem, das genauso viele Gleichungen und Variable enthält wie das WWW Seiten (die Google bekannt sind)!

Dummerweise kann das aber **nicht** wie gewünscht **klappen**! Haben wir etwa eine Situation wie die rechts abgebildete vor uns, so wäre $R(x_1) = R(x_2) + \frac{R(x_3)}{C(x_3)}$ und $R(x_2) = R(x_1)$. Damit wäre automatisch $R(x_3) = 0$. Würden wir x_1 und x_2 entfernen, so bekäme die Seite x_3 vielleicht einen ordentlichen Score von den vielen Links, die auf sie verweisen. Der eine, harmlos erscheinende Link von x_3



auf eine beiden aufeinander verweisenden Seiten x_1 und x_2 nimmt x_3 ihren ansonsten wohlverdienten Score *gänzlich* weg! Das soll natürlich nicht sein!

Die **Lösung**, die Page und Brin vorschlugen, besteht darin, jeder Seite – einfach dafür, dass sie existiert – zunächst den Score 1 zu geben und dann ein gewichtetes Mittel dieses Vorab-Scores mit dem in (1) definierten Wert $\sum_{\substack{\text{Seiten } y, \text{ die auf} \\ x \text{ verweisen}}} \frac{R(y)}{C(y)}$ zu bilden. Die Gewichtungsfaktoren werden in der Form $1-d$ und d geschrieben, wobei d eine fest gewählte Konstante (der so genannte „Dämpfungsfaktor“) ist, die $0 < d < 1$ erfüllt. Damit änderten Page und Brin die ursprüngliche Definition (1) des Scores in

$$R(x) = 1-d + d \sum_{\substack{\text{Seiten } y, \text{ die auf} \\ x \text{ verweisen}}} \frac{R(y)}{C(y)} \qquad \text{für alle } x \in \text{WWW}. \tag{2}$$

Der so ermittelte Score einer Seite ist ihr **PageRank**. Der Wert des Dämpfungsfaktors d ist ein Betriebsgeheimnis von Google, aber Gerüchte wollen wissen, dass er in der Gegend von 0.85 angesetzt wird.

Das Gleichungssystem (2) ist das ursprüngliche Erfolgsrezept von Google. Mit ihm kann kein Malheur wie mit (1) passieren, und man kann zeigen, dass es stets (wie auch immer d gewählt wird, solange $0 < d < 1$ ist) eine **eindeutig bestimmte Lösung** besitzt. Jede in der Datenbank von Google registrierte Webseite hat einen eindeutig bestimmten PageRank. Die Anzeige der Treffer wird nach abnehmendem PageRank vorgenommen (gegebenenfalls modifiziert durch zusätzliche Kriterien, die etwa die Güte der Übereinstimmung der eingegebenen Suchbegriffe mit den in der Seite enthaltenen Worten messen oder Manipulationsversuche wie das Anlegen fast identischer Seiten bestrafen).

Für das oben abgebildete Mini-WWW nimmt das Gleichungssystem (2) di e Form

$$\begin{aligned}
R(x_1) &= 1-d \\
R(x_2) &= 1-d \\
R(x_3) &= 1-d+d\left(\frac{R(x_1)}{1}+\frac{R(x_2)}{1}+\frac{R(x_5)}{3}\right) \\
R(x_4) &= 1-d+d\,\frac{R(x_5)}{3} \\
R(x_5) &= 1-d+d\,\frac{R(x_3)}{1} \\
R(x_6) &= 1-d+d\,\frac{R(x_5)}{3}
\end{aligned}$$

für die sechs Variablen $R(x_1)$, $R(x_2)$, … $R(x_6)$ an. Gerechterweise haben die Seiten x_1 und x_2, auf die überhaupt keine Links verweisen, den kleinsten PageRank. Sie werden als letzte gereiht, so dass also die Vergabe eines Vorab-Scores keine Überbewertung von Seiten, die keine „Stimme“ erhalten, darstellt. Mit $d = 0.85$ lautet die Lösung (gerundet): $R(x_1) = 0.15$,

$R(x_2) = 0.15$, $R(x_3) = 0.59$, $R(x_4) = 0.33$, $R(x_5) = 0.65$ und $R(x_6) = 0.33$. Die Reihung der Seiten mit abnehmendem PageRank lautet daher x_5, x_3, (x_4, x_6), (x_1, x_2), wobei die Klammern Seiten mit gleichem PageRank anzeigen.

MathematikerInnen können sich mit den bisherigen Informationen natürlich nicht zufrieden geben! Zwei Fragen drängen sich auf: Wie lässt sich zeigen, dass das Gleichungssystem (2) stets genau eine Lösung besitzt? Und wie schafft es Google, dieses System für Milliarden von Webseiten zu lösen?

Existenz und Eindeutigkeit des PageRank

Um zu zeigen, dass das Gleichungssystem (2) stets genau eine Lösung besitzt, wird es am besten in Matrixform

$$\vec{R} = \vec{D} + d\,A\,\vec{R} \qquad (3)$$

angeschrieben. Dabei sind Spaltenvektoren durch Pfeile gekennzeichnet, und es ist

$$\vec{R} = \begin{pmatrix} R(x_1) \\ R(x_2) \\ \vdots \\ R(x_n) \end{pmatrix}$$ der Vektor der (gesuchten) PageRanks und $$\vec{D} = \begin{pmatrix} 1-d \\ 1-d \\ \vdots \\ 1-d \end{pmatrix}.$$

n ist die Gesamtzahl der erfassten Webseiten, die in beliebiger Weise als x_1, x_2, … x_n durchnummeriert worden sind. Die $n \times n$-Matrix A ist so gewählt, dass

$$A\,\vec{R} = \begin{pmatrix} \sum\limits_{\substack{\text{Seiten } y, \text{ die auf} \\ x_1 \text{ verweisen}}} \frac{R(y)}{C(y)} \\ \sum\limits_{\substack{\text{Seiten } y, \text{ die auf} \\ x_2 \text{ verweisen}}} \frac{R(y)}{C(y)} \\ \vdots \\ \sum\limits_{\substack{\text{Seiten } y, \text{ die auf} \\ x_n \text{ verweisen}}} \frac{R(y)}{C(y)} \end{pmatrix}$$

ist. Ihre Koeffizienten A_{jk} sind durch

$$A_{jk} = \begin{cases} 0 & \text{wenn } x_k \text{ nicht auf } x_j \text{ verweist} \\ 1/C(x_k) & \text{wenn } x_k \text{ auf } x_j \text{ verweist} \end{cases}$$

gegeben. Die Summe der Koeffizienten der k-ten Spalte ist entweder gleich 1 (wenn x_k auf zumindest *eine* andere Seite verweist) oder gleich 0 (wenn x_k auf keine andere Seite verweist). Beweis für den ersten Fall: $\sum_{j=1}^{n} A_{jk} = \sum_{\substack{\text{Links von } x_k \\ \text{irgendwohin}}} \frac{1}{C(x_k)} = \frac{1}{C(x_k)} C(x_k) = 1$.

Zur Illustration schreiben wir die Matrix A für unser obiges Mini-WWW an:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

Jede Spalte enthält entweder lauter Nullen oder eine diskrete Wahrscheinlichkeitsverteilung.

Mit Hilfe der $n \times n$-Einheitsmatrix I lässt sich (3) in der Form

$$(I - d\,A)\vec{R} = \vec{D} \qquad (4)$$

schreiben. Es bleibt zu zeigen, dass die Matrix $I - d\,A$ invertierbar ist, denn in diesem Fall ist die (eindeutige) Lösung durch $\vec{R} = (I - d\,A)^{-1}\vec{D}$ gegeben.

Satz: Die Matrix $I - d\,A$ ist invertierbar.

Beweis: Die Invertierbarkeit von $I - d\,A$ ist gleichbedeutend mit jener der transponierten Matrix $I - d\,A^T$, mit der sich der Beweis leichter formulieren lässt. $I - d\,A^T$ ist genau dann invertierbar, wenn $(I - d\,A^T)\vec{S} = 0$ nur für $\vec{S} = 0$ möglich ist. Sei also $(I - d\,A^T)\vec{S} = 0$ oder, gleichbedeutend, $\vec{S} = d\,A^T\vec{S}$. Gemäß der Definition von A ist die k-te Komponente des Vektors $A^T\vec{S}$ entweder ein Mittelwert einiger der Komponenten von $\vec{S}$ (wenn x_k auf zumindest *eine* andere Seite verweist) oder 0 (wenn x_k auf keine andere Seite verweist). Nun wählen wir eine Komponente S_k, deren Betrag größer-gleich der Beträge aller anderen Komponenten von $\vec{S}$ ist. Wir nehmen o.B.d.A. an, dass $S_k \geq 0$ ist (ansonsten betrachten wir $-\vec{S}$ anstelle von $\vec{S}$). Die k-te Gleichung des Systems $\vec{S} = d\,A^T\vec{S}$ besagt nun, dass S_k entweder 0 ist oder d mal dem Mittelwert einiger der Komponenten von $\vec{S}$. Ein solcher Mittelwert kann aber nie größer als das maximale Element sein – er ist also $\leq S_k$. Nun wird er noch mit der positiven Zahl d, die kleiner als 1 ist, multipliziert. Wäre $S_k \neq 0$, so wäre S_k gleich einer Zahl, die *kleiner* als S_k ist. Da das offensichtlich nicht der Fall sein kann, folgt

$S_k = 0$. Wenn das für die Komponente mit dem größten Betrag gilt, verschwindet der Betrag *aller* Komponenten, woraus folgt: $\vec{S} = 0$. Damit ist der Beweis beendet: $I - d\,A^T$ ist invertierbar und damit auch $I - d\,A$.

PageRank für viele Webseiten berechnen

Das Gleichungssystem (2) – bzw. in der Form (3) oder (4) ausgedrückt – besitzt also genau eine Lösung. Wie kommen wir aber zu ihr, wenn die Zahl der Gleichungen und Variablen in die Milliarden geht? Google gibt sich mit einer **Näherungslösung** zufrieden.

Das Gleichungssystem (3) lässt sich **rekursiv** lösen. Dazu wird ein beliebiger Start-Vektor $\vec{R}_0$ angenommen und in die rechte Seite von (3) eingesetzt. Der daraus entstehende Vektor auf der rechten Seite wird als $\vec{R}_1$ bezeichnet:

$$\vec{R}_1 = \vec{D} + d\,A\,\vec{R}_0 \,.$$

Die gleiche Operation wird nun mit $\vec{R}_1$ als neuem Start-Vektor durchgeführt,

$$\vec{R}_2 = \vec{D} + d\,A\,\vec{R}_1 \,,$$

und oft wiederholt. Das allgemeine Schema lautet also

$$\vec{R}_n = \vec{D} + d\,A\,\vec{R}_{n-1} \,. \qquad (5)$$

Nun lässt sich zeigen, dass die Folge der Vektoren $\vec{R}_n$ für $n \to \infty$ gegen die (eindeutige) Lösung **konvergiert**. (Die **Beweisidee** ist einfach: Bezeichnet $\vec{R}$ die exakte Lösung, so kann die Differenz $\vec{X}_n = \vec{R}_n - \vec{R}$ durch $\vec{X}_n = d\,A\,\vec{X}_{n-1}$ erhalten werden. Daher gilt $\vec{X}_n = (d\,A)^n\,\vec{X}_0$, und es ist nur noch zu zeigen, dass $(d\,A)^n$ für $n \to \infty$ gegen 0, d.h. gegen die Matrix, die nur Nullen als Koeffizienten besitzt, konvergiert. Überlegen Sie selbst, warum das so ist!) Wird das Verfahren nach einer endlichen Zahl von Schritten abgebrochen, so ergibt sich eine Näherungslösung. Im *Mathematica*-Notebook
http://homepage.univie.ac.at/franz.embacher/Lehre/aussermathAnw2011/PageRank.nb
ist dieser Algorithmus für das obige Mini-WWW durchgeführt.

Google geht noch ein bisschen ökonomischer vor: Die Anwendung von (5) ist eine langwierige Angelegenheit mit Milliarden von Rechnungen, bei der sich nach und nach die Komponenten von $\vec{R}_n$ ergeben. Wann immer eine solche Komponente berechnet wurde, wird der erhaltene Wert in $\vec{R}_{n-1}$ ersetzt. In der üblichen Logik von Programmiersprachen kann das so ausgedrückt werden (wobei nun wieder die ursprüngliche Form (2) verwendet wird): Zunächst bekommen alle $R(x_j)$ irgendeinen Startwert(z.B. $1-d$). Dann wird berechnet:

$$R(x_1) = 1-d + d \sum_{\substack{\text{Seiten } y, \text{ die auf} \\ x_1 \text{ verweisen}}} \frac{R(y)}{C(y)}$$

$$R(x_2) = 1-d + d \sum_{\substack{\text{Seiten } y, \text{ die auf} \\ x_2 \text{ verweisen}}} \frac{R(y)}{C(y)}$$

…

$$R(x_n) = 1-d + d \sum_{\substack{\text{Seiten } y, \text{ die auf} \\ x_n \text{ verweisen}}} \frac{R(y)}{C(y)}$$

(Noch einmal, zur Sicherheit: Damit ist *kein* Gleichungssystem gemeint, sondern tatsächlich eine *Berechnung*. In Programmiersprachen wird in genau demselben Sinn etwa $x = x+1$ als Anweisung für eine *Berechnung* geschrieben). Ist etwa die erste Zeile einmal erledigt, so wird ab sofort für $R(x_1)$ der bereits ermittelte Wert verwendet, wann immer er in den folgenden Zeilen benötigt wird. Nach etwa 50 Iterationen und einigen Stunden Rechenzeit ist das Ergebnis für Google genau genug. Da täglich neue Webseiten entstehen und sich die Verlinkungsstruktur ändert, wendet Google dieses Verfahren praktisch kontinuierlich an, um den PageRank der einzelnen Webseiten stets aktuell zu halten.