Zeg nooit zomaar googelen tegen zoeken op het internet



De advocaten van Google moeten de bovenstaande inverse bananenslogan in gedachten hebben gehad toen ze protesteerden tegen deze betekenis van het woord googelen dat door verschillende Nederlandse woordenboeken verklaard werd als "zoeken op het internet". Googelen kan je enkel met Google en dat is een beschermde merknaam. In feite is de naam Google een tikfoutje want het was de bedoeling van de stichters Larry Page en Sergey Brin, twee studenten aan de Stanford universiteit in 1998, om Googol te laten registreren. Gezien de complexiteit die het web toen reeds had, leek hen Googol een geschikte naam. Het betekent immers 10^{100}, een 1 gevolgd door 100 nullen wat groter is dan het aantal elementaire deeltjes in het gekende heelal. Edward Kasner, een Amerikaans wiskundige, populariseerde de term googol rond 1920 om uit te leggen dat heel groot nog steeds niet hetzelfde is als oneindig. Het was in feite zijn negenjarig neefje dat googol suggereerde als naam. Misschien heeft die wel google gezegd maar heeft Kasner het verkeerd genoteerd.

Hoe kan Google in april 2007, in minder dan 10 jaar na de oprichting, uitgroeien tot de nummer één van de Brandz Top 100 lijst van meest waardevolle merknamen (een plaats die lang bezet is door Microsoft). De motor die het bedrijf zijn impact geeft is de Google zoekmachine die erin slaagt om bijna steeds de meest relevante webpagina's corresponderend met de ingetikte zoekterm vooraan te plaatsen. Die volgorde wordt bepaald door een algoritme dat PageRank heet (een woordspeling omdat Larry Page één van de auteurs is). Het gebeurt niet gauw dat wiskunde in de media komt, maar een artikel als *"The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google"* van Kurt Bryan en Tanya Leise in SIAM Review (2006), trekt algauw de aandacht. Het bedrag in de titel verwijst naar de geschatte waarde van Google toen het in 2004 naar de beurs ging.

Hoewel het *not-done* is om toe te geven dat je niet zonder dt-fouten kan schrijven is het nog steeds *bon-ton* om te verkondigen dat men niets van wiskunde kent maar is het anderzijds dan weer hip om met IT-termen te goochelen (met ch!). Misschien is deze tekst voor een aantal niet-wiskundigen een inleiding op de googlogie die met enige gretigheid zal gesavoureerd worden en kan dit stukje wiskunde in het (hopelijk) zoete google-broodje verstopt worden. Dat gebeurt trouwens ook in het boek *Google's PageRank and Beyond, the science of search engine rankings* dat geschreven is door A.N. Langville en C.D. Meyer, allebei professoren wiskunde.



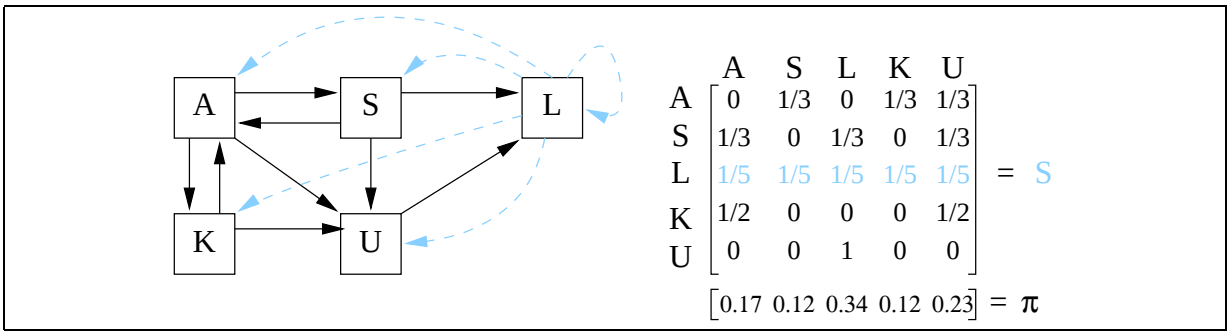
De PageRank wordt aangegeven door het groene streepje op de Google taakbalk

Hoe bepaalt PageRank dat een webpagina populairder, en dus belangrijker is dan een andere? Op een doorsnee webpagina staan gemiddeld ongeveer tien verwijzingen naar andere pagina's. Dat zijn de *outlinks* voor deze pagina. Omgekeerd zullen andere pagina's dan weer verwijzen naar deze pagina. Dat zijn dan de *inlinks* ervan. Als pagina A verwijst naar pagina S, dan is dit een stem die A uitbrengt op S in de *pop poll* die op het web continu aan de gang is. Men zou zo de pagina met

de meeste inlinks als de meest populaire kunnen beschouwen. Volgens het democratisch principe krijgt elke pagina echter maar één stem om uit te delen. Pagina A kan die ene stem helemaal aan pagina S geven door enkel naar S te linken, maar A kan ook zijn stem gelijk verdelen over meerdere pagina's. Als bijvoorbeeld A linkt naar S, K en U, dan krijgen die elk een derde van de stem van A. Zo krijgt elke link een waarde. De populariteit van A wordt dan gemeten door de som te maken van de waardes van haar inlinks. Goede poging, maar zo zit de reële, en dus ook de virtuele wereld niet echt in elkaar. Je zal maar tot de jetset behoren als je populair bent bij de jetset-populatie. In een democratie is immers iedereen gelijk, maar de ene al wat meer gelijk dan de andere. Daarom zal een stem van een populaire pagina uiteindelijk zwaarder wegen dan een stem van een niet-populaire pagina. De stem die A kan uitdelen wordt gewogen met haar populariteitsindex. Met andere woorden een pagina verdeelt niet haar stem maar verdeelt haar eigen populariteit over haar outlinks en wat de pagina aan populariteit binnenkrijgt via haar inlinks bepaalt haar eigen populariteit. Hiermee hebben we de moeder-versie van PageRank gedefinieerd.

Die laatste formulering is een zelf-refererend systeem. Om de populariteit van pagina A te berekenen moeten we de populariteit kennen van alle pagina's die naar A verwijzen, maar als A ook verwijst naar andere pagina's, dan kunnen we de populariteit van die andere pagina's niet berekenen voor we de populariteit van A kennen, en we zitten met een patstelling. Het is ook een transcendent probleem dat men nooit exact zal kunnen oplossen met eindig veel klassieke wiskundige bewerkingen. Hoe slaagt Google er dan in toch een rangschikking te bepalen? Heel eenvoudig: men moet het probleem niet exact oplossen. Een benadering van de exacte oplossing die toelaat om een zekere rangschikking te maken, is voldoende. Dit is een algemeen principe. Wiskundige modellen zijn benaderingen van de realiteit maar ze gelden slechts in een ideale, virtuele wereld. Theoretisch werkt de wiskunde met getallen die oneindig veel cijfers kunnen hebben, met lijnen en vlakken die geen dikte hebben enz. Maar die bestaan niet. Hoe geavanceerd onze toestellen ook zijn om het allerkleinste te meten, we zullen steeds op een grens stoten waaronder we geen verdere onderverdeling meer kunnen maken. De waarneembare realiteit is dus niet continu, maar discreet. Dit besef en de daarmee gepaard gaande ontwikkeling van de wetenschap is er vooral gekomen na het ontstaan van digitale computers, waar de kleinst mogelijke korrel aan informatie één bit is.

Maar laten we terugkeren naar PageRank. Stel dat we 5 pagina's (A, S, L, K, U) moeten sorteren volgens PageRank. (Zie figuur.) We discretiseren de tijd in ogenblikken 0,1,2,... en starten op tijdstip 0 met de veronderstelling dat alle pagina's even belangrijk zijn. Dit bewaart men in een rij $\pi_0 = [1/5,\ 1/5,\ 1/5,\ 1/5,\ 1/5]$. Op tijdstip 1 verdeelt iedere pagina haar rang over bevriende pagina's (haar outlinks). Dit herverdeelt de populariteit en men krijgt een nieuwe rij van getalletjes met voor elke pagina de rang na de herverdeling. Op tijdstip 2 verdeelt weer iedere pagina zijn huidige rang over dezelfde vrienden als de eerste keer en dit proces blijf je herhalen. Zo kom je dichter en dichter bij de exacte oplossing die je weliswaar maar zal bereiken na oneindig veel stappen, maar praktisch heeft PageRank maar een vijftigtal stappen nodig om een voldoende goede benadering te vinden die toelaat alle pagina's te sorteren.



Het algoritme kan wiskundig zeer eenvoudig neergeschreven worden. Stel dat H de tabel is met 5 rijen en 5 kolommen waarin op rij A rij vermeld staat hoeveel pagina A van haar stem toekent aan

elk van de 5 mogelijke pagina's. Dat is voor elke outlink evenveel, dus 1/3 omdat er 3 outlinks zijn. De pagina's waar niet naar gelinkt wordt krijgen een 0. Analoog voor de andere pagina's. Pagina L heeft geen outlinks en de hele rij L is daarom 0. We noemen het een bengelende pagina. De tabel H is als de tabel S op de figuur maar met nullen op de rij L. Lees je de tabel per kolom, dan staat bijvoorbeeld in kolom U hoeveel pagina U ontvangt van elk van de andere pagina's via haar inlinks. De tabel H beschrijft de regels voor herverdeling. Met $\pi_{k+1} = \pi_k H$ geven we aan hoe π_k na herverdeling π_{k+1} wordt. Na oneindig veel stappen is geen verbetering meer mogelijk is en dan is $\pi = \pi H$. Een dergelijk probleem noemt men een eigenwaardeprobleem en π is een eigenvector. Dat is een wiskundige term die komt uit het Duits en voor het eerst gebruikt is door David Hilbert in 1904. Het heeft dus niets te maken met het feit dat een pagina haar eigen waarde zou kunnen bepalen. Eigenwaardeproblemen krijg je als iets op een veelvoud van zichzelf wordt afgebeeld. Een televisietoestel gefilmd door een televisiecamera wordt door de televisiezender gecapteerd en weer naar het scherm gestuurd. Dan zie je op het scherm een toestel met op zijn scherm een toestel... (het zg. Droste-effect). Als het toestel in het beeld maar half zo groot is als het gefilmde toestel is dit een eigenwaardeprobleem met eigenwaarde 0.5. Als de camera precies het beeldscherm filmt, is het uitgezonden beeld gelijk aan wat er gecapteerd wordt en is de eigenwaarde 1. Andere voorbeelden: als je een microfoon voor een luidspreker houdt, zal de versterker de eigenfrequenties versterken (eigenwaarde groter dan 1) met onaangename fluittonen als gevolg; gebouwen hebben ook eigenfrequenties waardoor sommige gebouwen instorten bij een aardbeving en andere gebouwen met andere eigenfrequenties er vlak naast bijna ongeschonden de schok overleven enz.

Bengelende pagina's vormen echter een praktisch probleem voor het PageRank algoritme. Laten we veronderstellen dat een surfer domweg (ze hoeft niet noodzakelijk blond te zijn) van pagina naar pagina fladdert door op elke pagina lukraak een outlink aan te klikken. Dit model noemt men in de wiskunde een dronkenmanswandeling (dus toch een gender-onderscheid? Laten we in het vervolg dan maar over *het* surfkonijn spreken). Dit lukraak surfen kan doorgaan tot het konijn op een bengelende pagina terechtkomt. Omdat het geen uitweg ziet stuurt het de boodschap "Beam me up Scotty" en zap! het wordt lukraak geteleporteerd naar een willekeurige pagina van het web. De kans dat het zich na k stappen op een bepaalde pagina zal bevinden wordt weergegeven door de rij getalletjes π_k en een surfstap later door π_{k+1}. Het verband is $\pi_{k+1} = \pi_k S$ waarbij S hetzelfde is als de tabel H maar de nulrij L is vervangen door een rij met waarden 1/5: het kan van L met gelijke kans naar om het even welke pagina gezapt worden. (Zie figuur.) De bengelende pagina en de berekeningsproblemen zijn verdwenen. Het vraagstuk heeft nu een unieke oplossing die bestaat uit positieve getalletjes waarvan de som 1 is en de methode om ze te berekenen werkt. PageRank gaat nog één stapje verder. Als het konijn op een niet-bengelende pagina aankomt laat men de keuze tussen ofwel met een zekere waarschijnlijkheid verder te surfen via de outlinks ofwel, in het andere geval, om geteleporteerd te worden naar een willekeurige pagina. De tabel S wordt dan vervangen door een nieuwe tabel $G = \alpha S + (1 - \alpha)E$ waarbij E de teleporteermatrix is. In ons mini-voorbeeld is E een tabel helemaal opgevuld met de waarden 1/5, omdat alle sprongen even waarschijnlijk zijn, maar men zou dat eventueel kunnen personaliseren. Het dronken konijn kan bijvoorbeeld een voorkeur hebben om naar websites van oenologische verenigingen te zappen. En hiermee zijn we tot de $E = mc^2$ gekomen van de webzoekmachines: $\pi = \pi(\alpha S + (1 - \alpha)E)$. Google kiest $\alpha = 0.85$.

Een voordeel van dit PageRank-algoritme is dat dit een rangschikking geeft van alle webpagina's, onafhankelijk van de zoektermen die je intikt. Dit in tegenstelling tot bijvoorbeeld HITS (Hypertext Induced Topic Search) geconcipieerd door Jon Kleinberg, zoals PageRank eveneens in 1998. Omdat het web voortdurend verandert, moet de PageRank voor de ongeveer 6 miljard websites regelmatig herberekend worden. Webrobots zijn constant in de weer met het uitzenden van spinnen die het web afsnuffelen om de wijzigingen op te sporen. De informatie wordt opgeslagen en geïndexeerd zodat pagina's relevant voor een vraag van een "gebruiker" (het Google-busje komt zo) gemakkelijk kunnen gevonden worden. Google heeft steeds een tamelijk open politiek gevoerd

en zo is het mogelijk de maandelijkse herberekening van de PageRank te volgen op het web. Men krijgt de PageRank van een pagina te zien op drie verschillende servers. Waarschijnlijk is één ervan operationeel terwijl op de andere twee de herberekening gebeurt. Eenmaal voltooid wordt de nieuwe PageRank operationeel. Tijdens die berekeningen ziet men de PageRank op en neer springen, wat gekend is als de Google-dans. Ondanks het eenvoudige wiskundige principe blijft het maandelijks (her)berekenen van de PageRank voor miljarden pagina's een technologisch hoogstandje: een knap staaltje van software-engineering en wiskundige ingenieurstechnieken is nodig alleen al omdat men de Googlematrix G maar in stukjes in de computer kan binnenhalen. Alle beschikbare rekenkracht wordt ingezet en de computers lopen warm. Eén van de redenen waarom Google in Ghlin-Baudour 250 miljoen wil investeren, is de nabijheid van water om de computers te koelen.

Het Internet Resultaten **1** - **10** van circa **1.060.000.000** voor **google** (**0,05** seconden)

Zoek op het trefwoord "google" en je krijgt meer dan 1 miljard pagina's (op 18 mei 2007)

De Google-dans wordt vol spanning gevolgd door firma's die webpagina's van hun klanten zo hoog mogelijk willen geklasseerd zien. Ze kunnen bijvoorbeeld link melkerijen (*link farms*) opzetten. Dat is een verzameling pagina's die naar elkaar verwijzen om daardoor hun PageRank op te drijven. Dit is een vorm van spam, een probleem waar alle zoekmachines, niet alleen Google, mee te maken hebben. Een andere manier om Google om de tuin te leiden is de Google-bom. De eerste Google-bom werd gelanceerd in 2001 door Adam Mathes, een student computerwetenschappen in Stanford. Een van de bekendste bommen is de "miserable failure" bom. Als je deze zoekterm intikte kwam Google je als belangrijkste pagina de webstek van George W. Bush aanbieden. Het principe is eenvoudig. Om dit effect te bekomen moet je aan vrienden vragen op hun pagina's een link te leggen naar de Bush webstek met als verwijzende tekst "miserable failure". Linkanalysesystemen zoals TrustRank van Yahoo! zijn speciaal ontworpen om relevante pagina's van spam te onderscheiden.

Natuurlijk is PageRank niet het enige systeem en zijn linkanalyse, data mining en aanverwante domeinen in volle ontwikkeling. Toepassingen zijn er overal waar gezocht wordt naar een naald in een hooiberg: bibliotheeksystemen, bibliometrie, bioinformatica,... en we exploreren nog maar het topje van de hooiberg.

Adhemar Bultheel



A.N. Langville en C.D. Meyer
Google's PageRank and Beyond, the science of search engine rankings
Princeton University Press (July 3, 2006)