MATEMATIKK OG INFORMASJONSSØKNING

PÅ NETTET.

Eskilstuna

5 september 02

Leting på nettet[1].

Notasjon

1 Webblesere: Alle som har tilgang til en *webbleser*, som Explorer, Mozilla, Netscape, eller Opera, og som har brukt en av de mange *søkemaskinene* som kommer med leseren har blitt mektig imponert over hvor raskt og presist søkemaskinene fungerer. I dag finnes det mer enn 3 milliarder[2] *hjemmesider* spredt over hele verden, og mange av dem inneholder store mengder med dokumenter. Det er vanskelig å fatte at det er mulig å lete gjennom disse enorme datamengdene, og på bare noen sekunder komme opp med de viktigste hjemmesidene som inneholder informasjonen vi søker etter. En oppmerksom bruker vil oppdage at det er betydelige forskjeller mellom de ulike søkemotorene, som Alta Vista, Google, Lycos, eller Yahoo, både når det gjelder hastighet og den informasjonen de finner. For å forstå forskjellen mellom programmene og hvordan det er mulig å oppnå slike enorme hastigheter er det nødvendig å kjenne til prinsippene og teoriene bak programmene.

anvendelser

Som så ofte når det gjelder spektakulære tekniske anvendelser, og spesielt når det gjelder *verktøyene* på nettet, så er det matematiske resultater og formler som ligger til grunn for anvendelsene, og som forklarer hvordan en slik fantastisk effektivitet og presisjon er mulig. Det er også karakteristisk for slike anvendelser at den matematikken som brukes er *klassisk* og ble funnet uten tanke på disse anvendelsene.

matematikk

Vi skal her forklare i hvilken del av søkningene matematikken spiller en avgjørende rolle, og gi et lite innblikk i den matematikken som behøves.

Søkomoterer

2 Søkemotorer: En enkel modell for en søkemotor på nettet består av tre deler:

* Den første ingrediensen er en *robot* som døgnet rundt søker opp hjemmesider og lader ned dokumentene på hjemmesiden i en *database*. Den mest kjente søkemotoren er *google* som kan søke

databaser

[1] En varm takk til Tommy Ekola (KTH) for all hjelp med denne artikkelen.

[2] Alle oppgaver om tider og antall må taes med en klype salt. De endrer seg hele tiden. Nye brukere kommer til, maskinene blir hurtigere, og hukommelsene større med en skremmende hastighet

opp og lade ned mer enn 2 milliarder hjemmesider i løpet av en uke. Det vil si, den klarer 100 000 hjemmesider per sekund.

Grunnen til at dette Sisyfos arbeidet er mulig er at hver maskin med hjemmesider har et internetnummer som er forholdsvis lett å finne på nettet. Har man først funnet en maskin er det lett å finne hjemmesidene på maskinen, og dermed også alle *lenkene* på hver hjemmeside. Ved hjelp av lenkene kan roboten også komme videre til nye hjemmesider.

ordliste

* Den andre ingrediensen er en *ordliste*, som inneholder de fleste viktige ordene som forekommer på dokumentene på de hjemmesidene vi har ladet ned ved hjelp av roboten. En slik liste kan inneholde mer enn 100 millioner ord, hvilket kan sammenliknes med de par tusen ordene vi bruker i daglig skrift og tale. Til hvert ord i ordlisten finnes en peker til de hjemmesidene der ordet forekommer. Hver gang vi skriver inn ett, eller flere, ord vi vil søke på, leter søkemaskinen i ordlisten til den finner alle ordene, og den vet da også hvilke hjemmesider ordene forekommer på. Med de hurtigste søkemaskinene tar det mindre enn et sekund å finne ordkombinasjoner med 5-6 ord.

  Det tar selvsagt lang tid å sette opp en ordliste over ordene som forekommer i en database med milliarder av dokumenter, men det er mindre krevende enn å finne og lade ned dokumentene i databasen.

ranking

* Den tredje ingrediensen består av å rangordne hjemmesidene etter hvor *viktige* de er. Hver gang vi søker på et, eller flere, ord, finnes ordet, eller ordene, oftest på tusentalls hjemmesider. For at en søkning skal være meningsfull er det derfor helt avgjørende at de hjemmesidene som kommer først opp på skjermen er de som har størst betydning for den som søker. Prøver vi for eksempel å finne opplysninger om Edvard Grieg vil vi ikke gjerne komme til tusenvis av hjemmesider som handler om noe helt annet, og der Grieg bare er nevnt i forbifarten. Vi vil komme direkte til de store sentrale arkivene som inneholder informasjon om Grieg og hans verk.

  Det er lett å forstå at det er en vanskelig oppgave å rangordne alle hjemmesidene på nettet. At en hjemmeside er "viktigere" enn en annen virker å bero på en *subjektiv* vurdering. Men subjektive bedømmelser av milliarder av hjemmesider, mange av dem med store mengder dokumenter, er selvsagt umulig. Dessuten er hjemmesidene veldig ulike. De spenner fra personlige hjemmesider som innholder noen få linjer tekst til enorme databaser med millioner av dokumenter. For å rangordne sidene må vi derfor prøve å finne *objektive* kriterier for hva som er "viktig", og som bare avhenger av den *formelle strukturen* av dokumentene, og som ikke beror på deres

innhold. Det finnes en rekke forslag på hvordan man skal rangordne hjemmesider etter "viktighet", og leseren kan selv tenke gjennom hvilke løsninger som kan være interessante. Skal man bruke antallet ganger et ord forekommer på en hjemmeside, eller kanskje antallet henvisninger til andre sider, eller kanskje antallet dokumenter på hjemmesiden? Den store forskjellen mellom de ulike søkemaskinene ligger nettopp i hvordan de løser denne oppgaven. De som klarer det best blir mest benyttet kommersielt og får derfor best råd til å kjøpe inn flere datorer og mer minne, og dermed bli markedsledende. I dag har de største søkemaskinene titusentalls datamaskiner som arbeider parallellt med å søke og lagre data.

Vi skal i resten av denne artikkelen forklare en genial metode for å rangordne hjemmesider, eller i det hele tatt store datamengder, som bare avhenger av *lenkene* på hjemmesidene. Metoden kalles *PageRank* og ble foreslått av Sergey Brin og Larry Page [B-P] for omkring 5 år siden, da de var studenter ved Stanford University. PageRank er hjertet i den fenomenale søkemotoren *google* og bygger på enkle og fundamentale matematiske ideer og resultater.

PageRank

PageRank.

Lenker

3 Lenker: Idéen bak PageRank er å bruke *lenkene* mellom hjemmesidene til å avgjøre hvor viktige de er. En lenke fra en hjemmeside a til en hjemmeside b er en henvisning i et dokument på hjemmesiden a som er slik at om man *klikker* på henvisningen så kommer man til hjemmesiden b. Vi skal bruke litt *matematisk terminologi* og si at a peker på b og skriver

$$a \qquad b \qquad c$$

om a, b og c er hjemmesidene og a og c peker på b, og hjemmesiden b peker på hjemmesiden c.

Problemet er å gi hver hjemmeside en *rang* R_a. Det vil si, vi vil tilordne et tall R_a til hver hjemmeside som forteller hvor viktig hjemmesiden er i forhold til de andre hjemmesidene. Med andre ord, vi vil at om R_a er større enn R_b så er a viktigere enn b, og a skal derfor komme opp på skjermen før b om begge hjemmesidene a og b inneholder ordene vi søker på. Det er for å løse dette problemet at vi behøver matematikk. For å forklare metoden begynner vi derfor med litt matematisk *notasjon*:

For hver hjemmeside a betegner vi dens rang med R_a. Det er disse tallene vi skal bestemme ved å finne ligninger som de

tilfredsstiller. Vi betegner med $|F_a|$ antallet lenker som finnes på hjemmesiden a. Videre betegner vi med B_a alle hjemmesidene som peker på a, det vil si alle hjemmesidene som har et dokument som inneholder en lenke til a. En litt foreklet form for ligningene som bestemmer tallene R_a i PageRank er

ligningen

$$\lambda R_a = \sum_{b \in B_a} \frac{R_b}{|F_b|}, \tag{PR}$$

der λ er et tall som vi også må bestemme.

Det er et par ting vi bør merke oss med dette ligningssystemet før vi kan forstå hvordan denne modellen for ranking fungerer.

4 Bemerkning: Rankingen R_a er bestemt av de hjemmesidene som peker på a og hvor mange lenker som finnes på disse hjemmesidene. Bidraget til R_a blir stort om mange sider med høy ranking, og med få lenker, peker på den. Dette er en rimelig *modell* ettersom en hjemmeside bør være viktig om den har mange viktige lenker til seg. Dessuten er disse lenkene mer verdt om de kommer fra hjemmesider med få lenker, hvilket også er tiltalende. Merk også at rangordningen ikke avhenger av hvilket ord vi søker. Om to ord står på de samme hjemmesidene vil de samme hjemmesidene komme først opp på skjermen.

modell

En annen fordel ved denne modellen er at antallet hjemmesider som peker på en gitt side er vanskelig å manipulere for de som har kommersielle interessert og vil at deres side skal ha høy ranking for å synes først ved en søkning. Vi kan selv velge hvilke sider vi vil lenke til, men ikke hvilke sider som skal lenke til oss.

5 Bemerkning: Rankingen til en side R_b blir like fordelt mellom alle hjemmesidene den peker på, og bidrar til hver hjemmeside med $\frac{R_b}{|F_b|}$, det vil si, termen $\frac{R_b}{|F_b|}$ forekommer i alle ligningen for R_a der b peker på a, og den forekommer bare i disse ligningene. Summerer vi over alle hjemmesidene H får vi derfor

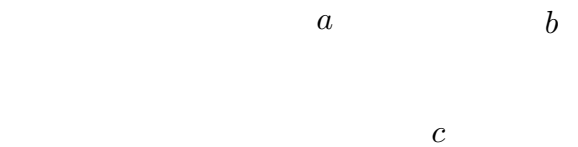
$$\lambda \sum_{a \in H} R_a = \sum_{a \in H} \sum_{b \in B_a} \frac{R_b}{|F_b|} = \sum_{a \in H'} R_a,$$

der H' er hjemmesidene som inneholder lenker. Vi ser at tallet λ måler kvoten mellom summen av rankingen til de hjemmesidene som inneholder lenker, og summen av rankingen til alle hjemmesidene. Ettersom det alltid finnes hjemmesider uten lenker vil $0 < \lambda < 1$.

6 Bemerkning: Vi vet at hjemmesidene, i gjennomsnitt, har 11 lenker. Det vil derfor også være i gjennomsnitt 11 lenker som

→ peker på en gitt hjemmeside. Derfor vil hver ligning (PR) stort sett inneholde 12 ukjente med ikke null koeffsienter.

7 Eksempel: (Redusible tilfellet)



gir ligningene

$$\begin{array}{lcccc} \lambda R_a & = & & & \frac{1}{2}R_c \\ \lambda R_b & = & R_a & & +\frac{1}{2}R_c \\ \lambda R_c & = & & R_b & . \end{array}$$

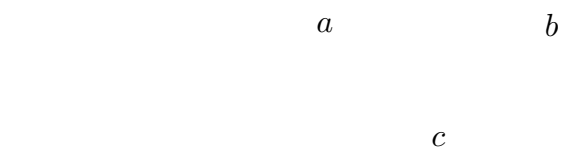
Legger vi sammen de venstre og høyre termene i disse tre ligningene får vi at

$$\lambda(R_a + R_b + R_c) = R_a + R_b + R_c,$$

som viser at enten er $R_a = R_b = R_c = 0$, som vi ikke vil ha, eller vi har at $\lambda = 1$. Med $\lambda = 1$ kan vi løse ligningssystemet og får at $R_a = \frac{1}{2}R_b = \frac{1}{2}R_c$. Vi kan velge R_a vilkårlig til $R_a = 1$, og får rankingen $R_a = 1, R_b = \frac{1}{2}, R_c = \frac{1}{2}$.

Små eksempler er selvsagt ikke *realistiske*. For eksempel fant vi at $\lambda = 1$, som ikke forekommer i praksis, som vi så i Bemerkning
→ (?). Vi må tenke oss at eksempelet er en del av et større nettverk.

8 Eksempel: (Loop)



gir ligningene

$$\begin{array}{lcccc} \lambda R_a & = & 0 & & \\ \lambda R_b & = & R_a & & +R_c \\ \lambda R_c & = & & R_b & . \end{array}$$

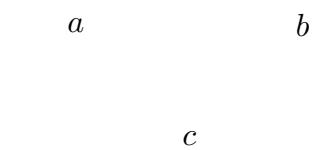
Legger vi sammen termene til venstre og høyre i disse tre ligningen får vi at

$$\lambda(R_a + R_b + R_c) = R_a + R_b + R_c,$$

som viser at enten vil $R_a = R_b = R_c = 0$, som vi ikke vil ha, eller så er $\lambda = 1$. Med $\lambda = 1$ kan vi løse ligningene, og vi får

$R_a = 0$ og $R_b = R_c$. Vi kan velge R_b vilkårlig til $R_b = 1$, og får rangordningen $R_a = 0, R_b = 1, R_c = 1$. Som i forrige eksempel er ikke dette eksempelet heller spesielt realistisk, og vi må tenke oss at dette også er en del av et større nettverk.

9 Eksempel: (Hengende hjemmesider)



gir ligningene

$$\begin{array}{lcll} \lambda R_a & = & & \frac{1}{2}R_b \\ \lambda R_b & = & \frac{1}{2}R_a & \\ \lambda R_c & = & \frac{1}{2}R_a & +\frac{1}{2}R_b \quad . \end{array}$$

Legger vi sammen termene til venstre og høyre i disse tre ligningene får vi at

$$\lambda(R_a + R_b + R_c) = R_a + R_b.$$

For $\lambda = 0$ får vi $R_a = R_b = 0$, og velger vi R_c vilkårlig til 1 får vi rangordningen $R_a = 0$, $R_b = 0$, $R_c = 1$. Når $\lambda \neq 0$ kan vi løse ligningene for R_a og får $R_a = 2\lambda R_b = 4\lambda^2 R_a$. Om vi vil ha $R_a \neq 0$ får vi at $\lambda = \frac{1}{2}$ Dette gir $R_a = R_b$, og $R_c = R_a + R_b$. Vi kan velge R_a vilkårlig til $R_a = \frac{1}{2}$ får vi rankingen $R_a = \frac{1}{2}, R_b = \frac{1}{2}, R_c = 1$.

hengende sid.

10 Bemerkning: I praksis vil man unngå *hengende hjemmesider*, det vil si hjemmesider som ikke inneholder noen lenker. Disse taes derfor bort under beregningene og settes tilbake til slutt.

looper

Vi vil også unngå *looper*, det vil si kjeder av hjemmesder der hver hjemmeside peker til den neste i kjeden, og der det finnes en hjemmeside som ikke inngår i kjeden, men som peker til en av medlemmene av kjeden. Det er for å håndtere slike looper at man

→ bruker en variant på ligningene (PR).

løsning →

11 Løsninger: Vi har ikke løst problemet med å ranke sider bare fordi vi har satt opp ligningssystemet (PR). Vi må også finne en metode for å løse disse ligningene i løpet av en rimelig tid. De tradisjonelle metodene som vi lærer på universitetene, for eksempel *Gauss-Jordan eliminasjon*, er altfor langsomme og for vanskelige å håndtere for ligningssystemer som inneholder så mange som 2 milliarder ligninger i like mange ukjente, selv når de fleste koeffisientene er 0. Når koeffisientene er positive eller null finnes det imidlertid andre metoder, som bruker *iterasjon*. Dette viser seg å

→ være veldig effektivt for ligningene (PR). I praksis rekker det med omkring 50 iterasjoner for å få en meget god ranking for hjemmesidene. Vi skal i de neste seksjonene gi en matematisk forklaring til hvorfor det fungerer så bra å *iterere* disse ligningene.

Litt matematikk.

12 Ligningene på matriseform: La $A = (R_{ab})$ være matrisen med koeffisienter $R_{ab} = \frac{1}{|F_a|}$ om b peker mot a og der $R_{ab} = 0$ om b *ikke* peker mot a. Videre la v være vektoren hvis a'te koordinat er lik R_a. Da kan ligningene (PR) skrives på *matrise form*:

matriserform

$$Av = \lambda v.$$

Alle som har lest litt *lineær* algebra vil kjenne igjen denne ligningen. Vi sier at λ er en *egenverdi* for matrisen A, og at v er en *egenvektor* for matrisen A *tilhørende* egenverdien λ.

→ **13 Eksempel:** I Eksempel (?) ovenfor vil matrisen være

$$A = \begin{pmatrix} 0 & 0 & \frac{1}{2} \\ 1 & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix}.$$

→ I Eksempel (?) er matrisen

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

→ og i Eksempel (?) er matrisen

$$A = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}.$$

Matrisen A har ikke negative koordinater R_{ab}. Når den tilfredsstiller visse tilleggsbetingelser, som vi skal beskrive nedenfor, finnes det en vakker *klassisk* teori for egenverdier og egenvektorer.

14 Definisjon: En matrise $A = (a_{ij})$ kalles *ikke negativ* om $a_{ij} \geq 0$ for alle i, j, og den kalles *positiv* om $a_{ij} > 0$ for alle i, j.

positiv

redusibel

15 Definisjon: En $n \times n$-matrise er *redusibel* om den er på *blokkformen*

$$\begin{pmatrix}
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\vdots & & & & & & & & & & & & & & \vdots \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\ast & \cdots & \ast & 0 & \ast & \cdots & \ast & 0 & \ast & \cdots & \ast & 0 & \ast & \cdots & \ast \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\vdots & & & & & & & & & & & & & & \vdots \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\ast & \cdots & \ast & 0 & \ast & \cdots & \ast & 0 & \ast & \cdots & \ast & 0 & \ast & \cdots & \ast \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\vdots & & & & & & & & & & & & & & \vdots \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\ast & \cdots & \ast & 0 & \ast & \cdots & \ast & 0 & \ast & \cdots & \ast & 0 & \ast & \cdots & \ast \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast \\
\vdots & & & & & & & & & & & & & & \vdots \\
\ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast & \ast & \ast & \cdots & \ast,
\end{pmatrix}$$

det vil si, den inneholder en *blokk* av nuller, der 0'ene står i rekkene $i_1, \ldots, i_p$ og i søylene $j_1, \ldots, j_q$, og der $p + q = n$ med n lik størrelsen av matrisen, og $\{i_1, \ldots, i_p, j_1, \ldots, j_q\} = \{1, 2, \ldots, n\}$. Ekvivalent er den redusibel om den ved en permutasjon av tallene $1, 2, \ldots, n$ kan skrives på formen

$$\begin{pmatrix} B & 0 \\ C & D \end{pmatrix}$$

der B er en $p \times p$-matrise, D er en $q \times q$-matrise, og 0-matrisen i øverste høyre hjørne er en $p \times q$-matrise. En matrise som ikke er redusibel kaller vi *irredusibel*.

De viktigste resultatene om positive og ikke-negative redusible matriser er (se for eksempel [G]):

Perron

16 Theorem: (Perron 1907) *Om A er en positiv matrise har den en positiv egenverdi $\lambda(A)$ som er en enkel rot i det karakteristiske polynomet, og som er strikt større enn absoluttverdien for de andre egenverdiene. Til $\lambda(A)$ svarer en egenvektor som er positiv.*

Frobenius

17 Theorem: (Frobenius 1908-1912) *En irredusibel ikke negativ matrise A har en positiv egenverdi $\lambda(A)$ som er en enkel rot i det karakteristiske polynomet, og som er strikt større enn absoluttverdien for de andre egenverdiene. Til $\lambda(A)$ svarer en egenvektor som er positiv.*

Følgende to resultater er også ofte nyttige (see for eksempel [H-J]):

18 Proposisjon: *Om A er en ikke-negativ irrredusibel matrise $(I + A^m)^{n-1}$ være positiv, der I er enhetsmatrisen og n er antall rader og søyler i A.*

19 Proposisjon: *Om A er en ikke-negativ irredusibel matrise slik at $a_{ii} > 0$ for noen i så finnes det en positiv egenverdi som er strikt større enn absoluttverdien til de andre egenverdiene.*

hjelpesats

20 Proposition: *La A være en matrise og ρ et positivt tall som er større enn absoluttverdien til egenverdiene til A. For hver vektor v vil da*

$$\lim_{n\to\infty} \frac{A^n v}{\rho^n} = 0.$$

Bevis. Ved Schurs metode kan vi lett finne en unitær matrise U slik at U^*AU er øvre triangulær. La D være diagonalmatrisen med koordinater $d_{ii} = \varepsilon^{i-1}$. Da vil $D^{-1}U^*AUD$ være en øvre diagonal matrise, og alle koordinatene ovenfor diagonalen vil være et produkt med en positiv potens av ε. Det følger at om vi velger ε liten så vil $0 = \lim_{n\to\infty} \frac{(D^{-1}U^*AUD)^n w}{\rho^n}$ for hver vektor w. Men vi har at $\lim_{n\to\infty} \frac{(D^{-1}U^*AUD)^n w}{\rho^n} = \lim_{n\to\infty} \frac{D^{-1}U^*A^nUDw}{\rho^n} = D^{-1}U^*\left(\lim_{n\to\infty} \frac{A^nUDw}{\rho^n}\right)$. Derfor vil $\lim_{n\to\infty} \frac{A^nUDw}{\rho^n} = 0$, og velger vi $w = D^{-1}U^*v$ får vi at $\lim_{n\to\infty} \frac{A^n v}{\rho_n} = 0$.

rekusrjon

21 Iterasjon: La A være en matrise og u en ikke null vektor. Vi definierer en følge av *vektorer* $v_0, v_1, v_2, \ldots$ *rekursivt* ved

$$v_0 = \frac{u}{|u|}, \quad v_1 = \frac{Av_0}{|Av_0|}, \quad v_2 = \frac{A^2v_1}{|A^2v_1|}, \ldots,$$

det vil si

$$v_{n+1} = \frac{Av_n}{|Av_n|} \quad \text{for} \quad n = 0, 1, 2, \ldots.$$

Da vil

$$v_n = \frac{A^n u}{|A^n u|} \quad \text{for} \quad n = 0, 1, 2, \ldots.$$

Spesielt har alle vektorene v_n lengde 1. Vi skal gi noen enkle betingelser for at sekvensen $v_0, v_1, v_2, \ldots$ *konvergerer* mot en vektor v som er en egenvektor for A. Det der dette vi mener med å *løse ligningen* $Av = \lambda v$ ved *iterasjon.*

Hovedsats

22 Setning: *La A være en matrise med en positiv egenverdi $\lambda(A)$ som er en enkel rot i det karakteristiske polynomet til A,*

og som er større en absoluttverdien av de andre egenvektorene til A. For "nesten alle" vektorer u vil da sekvensen av vektorer $v_0 = \frac{u}{|u|}, v_1 = \frac{Au}{|Au|}, v_2 = \frac{A^2u}{|A^2u|}, \ldots$ konvergere og

$$\lim_{n\to\infty} \frac{A^n u}{|A^n u|} = v,$$

der v er en egenvektor for A av lengde 1 som tilhører egenverdien $\lambda(A)$.

Bevis. Ettersom $\lambda(A)$ er en enkel rot i minimalpolynomet kan vi finne en basis $u_1, u_2, \ldots, u_n$ for vektorrommet slik at v_1 er en egenvektor for matrisen A av lengde 1 tilsvarende egenverdien $\lambda(A)$, og slik at A med hensyn til denne basen kan skrives på formen

$$A = \begin{pmatrix} \lambda(A) & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{pmatrix},$$

der A_1 er en $(n-1)\times(n-1)$-matrise. Skriver vi $u = a_1u_1 + a_2u_2 + \cdots + a_nu_n$ og setter $u' = a_2u_2 + a_3u_3 + \cdots + a_nu_n$ så får vi at

$$A^n u = \begin{pmatrix} \lambda(A)^n a_1 \\ (A')^n u' \end{pmatrix}.$$

Bortsett fra $\lambda(A)$ har A og A' samme egenverdier. Det følger
→ derfor av Proposisjon (?) at

$$\lim_{n\to\infty} \frac{(A')^n u'}{\lambda(A)^n} = 0$$

og derfor at

$$\lim_{n\to\infty} \frac{A^n u}{\lambda(A)^n} = \begin{pmatrix} a_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = a_1 u_1.$$

Videre følger det at

$$\lim_{n\to\infty} \frac{|A^n u|}{\lambda(A)^n} = |a_1|\,|u_1| = |a_1|.$$

Om $|a_1| \neq 0$, det vil si, for alle vektorer som ikke ligger i vektorrommet *spent* av $u_2, u_3, \ldots, u_n$ får vi derfor

$$\begin{aligned} \lim_{n\to\infty} \frac{A^n u}{|A^n u|} &= \lim_{n\to\infty} \left(\frac{A^n u}{\lambda(A)^n} \frac{\lambda(A)^n}{|A^n u|} \right) \\ &= \lim_{n\to\infty} \left(\frac{A^n u}{\lambda(A)^n} \right) \lim_{n\to\infty} \left(\frac{\lambda(A)^n}{|A^n u|} \right) = \frac{a_1}{|a_1|} u_1 = \pm u_1. \end{aligned}$$

Tilbake til PageRank.

praksis

23 Matematikk og praksis: Vi har tidligere bemerket at matrisen $A = (R_{ab})$ er ganske spesiell. I søylen b er det nøyaktig $|F_b|$ koordinater som ikke er null, og alle ikke null koordinater er lik $\frac{1}{|F_b|}$. Videre har hver rekke omkring $|F_b|$ koordinater som er forskjellig fra null. I praksis er $|F_b|$ omtrent lik 11, så "nesten alle" koordinatene i $A = (R_{ab})$ er like null. Derfor er det ganske lett å regne ut vektorene $v_n = \frac{A^n u}{|A^n u|}$ for en vilkårlig vektor u. Bestemmer man kvotientene $\frac{A^n u}{|A^n u|}$ for $n = 1, 2, \ldots$ så viser det seg at når n nærmer seg 50 så skiller vektorene seg $\frac{A^n u}{|A^n u|}$ og $\frac{A^{n+1} u}{|A^{n+1} u|}$ veldig lite. Det er derfor rimelig å bruke vektoren $\frac{A^n u}{|A^n u|}$ for noe $n \geq 50$ som en *rang vektor*. Som vi merker av den fabelaktige prestasjonsevnen til søkemaskinene som bruker PageRank fungerer dette aldeles utmerket i praksis.

Matematikken vi skisset i forrige seksjon er relevant for å sannsynliggjøre at sekvensen $\frac{u}{|u|}, \frac{Au}{|Au|}, \frac{A^2 u}{|A^2 u|}, \ldots$ konvergerer mot en
→ egenvektor. For å bruke Setning (?) er det imidlertid nødvendig at forutsetningen om at $A = (R_{ab})$ har en positiv egenverdi som er en enkel rot i det karakteristiske polynomet, og som er større enn absoluttverdien av de andre egenverdiene. Om A var positiv
→ ville dette følge av Perrons Setning (?), men som vi har sett er A
→ langt fra positiv. For å bruke Frobenius Setning (?) må vi, blandt annet, vite at matrisen er irredusibel. Dette er langt fra klart. At den er irredusibel betyr, litt upresist, at det ikke finnes grupper av hjemmesider som bare henviser til hverandre. Det er blandt annet
→ dette vi prøver å unngå ved å modifisere ligningene (PR), og ved å ta bort hengende hjemmesider. Om A er irredusibel må vi, for å bruke Frobenius Setning, dessuten vite at den positive egenverdien $\lambda(A)$ er strikt større enn absoluttverdien for de andre egenverdi-
→ ene. Dette holder, ved Proposisjon (?), om R_{aa} er forskjellig fra 0 for noe a. Dette vet vi ikke holder. Derimot vil dette bli tilfredsstilt om vi modifiserer ligningssystemet ved å betrakte hver hjemmeside som lenket til seg selv. En slik modifikasjon gjør at vi
→ også kan bruke Lemma (?) som sier at $(I + A)^{n-1}$ er positiv, og dermed Perrons Setning. Dette er imidlertid upraktisk ettersom n er så stor.

Man kan imidlertid spørre seg om disse resonnementene er interssante eller nødvendige, ettersom iterasjon fungerer i praksis. Svaret er at vi klarer oss uten resonnementene, men at det er takket være matematikerne, og resultatene vi har nevnt, at vi i det hele tatt skulle komme på tanken å løse ligningene ved iterasjon. Det er dette som gjør matematikken så fundamental.

Det er matematikken som gjør det mulig å sette opp de rette modellene, og som antyder hvordan modellene skal analyseres.

References

[B-P] Sergey Brin and Larry Page, *The PageRank Citation Ranking: Bringing order to the web*, google search engine, `http://google.stanford.edu`.

[G] F.R. Gantmacher, *Applications of the theory of matrices*, Interscience publishers, inc., London-New York, 1959.

[H-J] Roger A. Horn & Charles R. Johnson, *Matrix Analysis*, Cambridge Univ. Press, 1985.