

MAT 1120: Obligatorisk oppgave 1, H-09

Innlevering: *Senest fredag 25. september, 2009, kl.14.30, på Ekspedisjonskontoret til Matematisk institutt (7. etasje NHA). Du kan skrive for hånd eller med datamaskin, men besvarelsen skal uansett leveres på papir. Erfaringsmessig blir det lange køer både ved skriverne og utenfor ekspedisjonskontoret rett før innleveringsfristen, så det er smart å levere tidligere. **Obligen skal leveres med en egen forside som du finner på***

<http://www.uio.no/studier/emner/matnat/math/MAT1100/h07/obliger.xml>

(det vil også være papirkopier av forsiden tilgjengelig ved innlevering). På samme side finner du regelverket for obliger ved Matematisk institutt. Husk spesielt å søke om utsettelse til studieinfo@math.uio.no før innleveringsfristen dersom du blir syk!

Instruksjoner: *Oppgaven er obligatorisk, og studenter som ikke får besvarelsen godkjent, vil ikke få adgang til avsluttende eksamen. For å få besvarelsen godkjent, må man ha minst 60% score (vekten til hver oppgave står på), og det vil bli lagt vekt på at man har en klar og ryddig besvarelse med gode begrunnelser. Alle svar skal begrunnes. Du kan få poeng på en oppgave selv om du ikke er kommet frem til et svar, og det er derfor viktig at du leverer inn alt det du har kommet frem til. Studenter som ikke får sin opprinnelige besvarelse godkjent, men som gjennom besvarelsen viser at de har gjort et reelt forsøk på å løse oppgavene, vil få én mulighet til å levere en revidert besvarelse.*

Det er lov å samarbeide og å bruke alle slags hjelpemidler. Den innleverte besvarelsen (tekst og programmer) skal imidlertid være skrevet av deg og gjenspeile din forståelse av stoffet. Er vi i tvil om at du virkelig har forstått det du har levert inn, kan vi be deg om en muntlig redegjørelse.

Anvendelser av lineær algebra på rangering av websider

Øyvind Ryan (oyvindry@ifi.uio.no)

7. september 2009

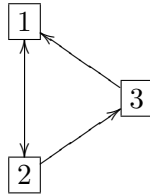
Ved søk på internett skriver du inn et par søkeord, og nettleseren kommer med forslag på websider som inneholder disse ordene. Du har sikkert lagt merke til at sidene kommer opp i en rangert rekkefølge, og at de mest interessante sidene listes først. Hvordan websider kan rangeres på en god måte, også kalt *pageranking*, er en liten vitenskap i seg selv. Noe av suksessen til Google ligger i at de er gode på dette. Å regne ut en rangering av alle sider på weben er en enormt regnekrevende operasjon, og selskaper som Google gjør dette med jevne mellomrom (en gang i måneden i følge [2]). Når man først har regnet ut en rangering, så kan denne brukes for alle resultater fra søk på internett, for å vise resultatene i rangert rekkefølge. Er du klar over at algoritmer for rangering av websider kan forklares ut fra lineær algebra, slik vi lærer i MAT1120? En del av koblingen mellom lineær algebra og rangering kommer gjennom teorien for *Markov-kjeder* (kapittel 4.9 i læreboka).

Denne obligatoriske oppgaven gir deg noen innførende oppgaver i sammenhengen mellom lineær algebra og rangering av websider. De av dere som er interessert i å vite mer bør lese [1], som fremstillingen her er delvis basert på. Hvis du er interessert i å vite enda mer så finner du mye i [2], som også inneholder mer om metoder i lineær algebra i bruk i dagens web-søkemotorer. En del av dette stoffet går utover pensumet i MAT1120.

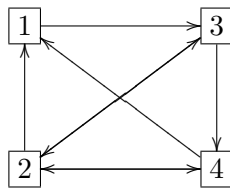
Terminologi

Vi representerer et nettverk av websider (her kalt en *web*) ved hjelp av diagrammer som i figur 1, 2, og 3. Hver node (firkant) i diagrammet svarer til et web-dokument. Vi antar at vår web inneholder n dokumenter, og nummererer disse med $1, 2, \dots, n$. En pil fra dokument j til dokument i svarer til en hyperlenke i dokument j , med referanse til dokument i . For en web skal vi lage det som kalles en *link-matrise*:

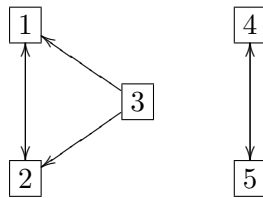
Definisjon 1 Vi definerer n_j til å være antall dokumenter som dokument



Figur 1: Et enkelt web med tre dokumenter



Figur 2: Web for oppgave 1



Figur 3: Web for oppgave 2

j refererer til ved hjelp av hyperlenker. Vi teller **ikke** referanser fra et dokument til seg selv.

Definisjon 2 Vi definerer link-matrisen $A = [a_{ij}]$ ved at $a_{ij} = \frac{1}{n_j}$ hvis dokument j har en hyperlenke til dokument i , og $a_{ij} = 0$ ellers.

Det er enkelt å sette opp link-matrisen for ethvert web-diagram: Kolonne j indikerer hyperlenker fra dokument j til andre dokumenter. Som et eksempel, la oss sette opp link-matrisen for weben fra figur 1. Dette vil være en 3×3 -matrise, siden denne weben har tre dokumenter. Ved å telle antall hyperlenker fra hvert enkelt dokument ser vi at $n_1 = 1$, $n_2 = 2$, og $n_3 = 1$.

Den første kolonnen i link-matrisen får vi ved å ta for oss alle hyperlenkene fra dokument 1. Siden dokument 1 bare har en hyperlenke til dokument 2, og $n_1 = 1$, så får vi $(0, 1, 0)$. Andre kolonne blir $(\frac{1}{2}, 0, \frac{1}{2})$ siden dokument 2 har hyperlenker til dokument 1 og 3, og $n_2 = 2$. Tredje kolonne blir $(1, 0, 0)$ siden dokument 3 bare har en hyperlenke til dokument 1. Setter vi disse tre kolonne-vektorene sammen til en link-matrise får vi

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

Link-matrisen består alltid av kun ikke-negative tall.

Vi vil også trenge følgende definisjon:

Definisjon 3 Vi definerer mengden $L_i \subset \{1, 2, \dots, n\}$ som dokumentene som inneholder hyperlenker til dokument i .

I [1] kalles disse for *backlinks*.

Vi er ute etter å bestemme fornuftige "scores" x_1, x_2, \dots, x_n , der x_i er score til dokument i . En score skal være et ikke-negativt tall, der en høy verdi betyr stor relevans. Hvis tallene x_1, \dots, x_n har sum 1, så vil vi kalle vektoren (x_1, \dots, x_n) for en *score-vektor*. I kapittel 4.9 i læreboka kalles dette også for en *sannsynlighetsvektor*.

En måte å regne ut en score-vektor er å kreve at

$$x_i = \sum_{j \in L_i} \frac{x_j}{n_j}. \quad (1)$$

En hyperlenke fra dokument j til dokument i gjenspeiles med et summeledd $\frac{x_j}{n_j}$ på høyre side i (1). Hvis det ikke forekommer en hyperlenke fra dokument j til dokument i , så vil $\frac{x_j}{n_j}$ -leddet for ligning i i (1) ikke være til stede. Vi har samlet litt stoff rundt motivasjonen for ligning (1) i et appendix, som spesielt interesserte anbefales å lese.

(1) beskriver et likningssystem som bruker link-matrisen:

$$A\mathbf{x} = \mathbf{x}, \quad (2)$$

alternativt

$$(A - I)\mathbf{x} = 0, \quad (3)$$

Som vi har lært, slike ligningssystemer kan ha uendelig mange løsninger, så det er ikke sikkert at det finnes en *unik* score-vektor.

Gitt en score-vektor så kan vi liste opp dokumentene etter avtagende score. En slik opplisting kalles en *rangering*. Vi vil si at rangeringen er *unik* hvis score-vektoren er unik. Vi bryr oss ikke om detaljer som hvilken side vi skal rangere først i tilfelle to sider får samme score, da det først og fremst er selve score-vektoren vi er ute etter.

Legg merke til at (3) sier at en score-vektor \mathbf{x} hører til nullrommet for matrisen $A - I$. Det er denne sammenhengen med lineær algebra vi skal bruke i oppgavene nedenfor.

Oppgave 1: (Teller 10%)

Skriv opp link-matrisen A for figur 2. Skriv opp en basis for nullrommet til matrisen $A - I$, og finn den unike score-vektoren (husk at summen av elementene i score-vektoren skal være 1). Angi den tilhørende rangeringen.

Oppgave 2: (Teller 10%)

Figur 3 viser det vi kaller en *usammenhengende web*, det vil si at weben kan splittes i delmengder som ikke refererer til hverandre. Skriv opp link-matrisen, og skriv opp en basis for nullrommet til matrisen $A - I$. Finnes det en unik score-vektor? Er rangeringen unik?

Oppgave 3: (Teller 30%)

I seksjon 4.9 i læreboka ble det definert hva det vil si at en matrise er stokastisk: I en stokastisk matrise er elementene ikke-negative og summen av elementene i hver kolonne er 1. Videre kalte vi en stokastisk matrise regulær hvis en potens av den inneholder bare ekte positive tall.

a) Er link-matrisene fra oppgave 1 og 2 stokastiske? Kan du forklare med enkle ord hvordan en web kan gi en link-matrise som ikke er stokastisk?

Problemet med unik rangering viser seg å være lettere å adressere hvis link-matrisen er stokastisk. Vi definerer

$$M = (1 - m)A + mS, \quad (4)$$

der $0 < m < 1$ og S er en $n \times n$ -matrise hvor alle elementer er $\frac{1}{n}$. I [2] kalles M for *Google-matrisen*. Legg merke til at M kan ses på som link-matrisen til en sammenhengende web, selv om A representerer link-matrisen til en usammenhengende web.

b) Vis at M er regulær og stokastisk når A er stokastisk.

Fra b) og teorem 18 på side 294 følger det nå at M har en unik scorevektor når A er stokastisk.

c) Skriv opp matrisen M når link-matrisen A er som i figur 3, og $m = 0.15$. Bruk Matlab-funksjonen `null(M-I)` til å finne nullrommet til $M - I$. Skriv opp den unike score-vektoren du nå får.

Oppgave 4: (Teller 20%)

a) I denne oppgaven skal du studere følgende Matlab-kode:

```
function A=randlinkmatrix(n)
    A = round(rand(n,n));
    for k=1:(n-1)
        A(k,k) = 0;
        if (A(:,k) == 0)
            A(n,k) = 1;
        end
        s = sum(A(:,k));
        A(:,k) = (1/s) * A(:,k);
    end
    A(n,n) = 0;
    if (A(:,n) == 0)
        A(1,n) = 1;
    end
    s = sum(A(:,n));
    A(:,n) = (1/s) * A(:,n);
```

Forklar med enkle ord hva de forskjellige delene i denne koden gjør, ut fra de begrepene som vi har innført denne obligatoriske oppgaven. Hva er det funksjonen returnerer?

b) Kjør metoden fra a) med $n = 5$, og tegn den tilhørende weben for link-matrisen som metoden returnerer (slik det er tegnet i figur 2 og 3).

Oppgave 5: (Teller 30%)

a) Programmer en Matlab-funksjon

`ranking(A)`

som returnerer (den unike) score-vektoren \mathbf{x} for matrisen M definert ved (4) med $m = 0.15$ når A er stokastisk. Hvis A ikke er stokastisk skal programmet returnere en feilmelding. Bruk Matlab-funksjonen `null(M-eye(5,5))` som i oppgave 3c).

b) Siden world wide web etterhvert inneholder forferdelig mange dokumenter, så vil funksjonen `ranking` regne ut nullrommet til veldig store matriser. Dette er regnekrevende. Vi kan i stedet forsøke å regne ut approksimasjoner til den (unike) score-vektoren på følgende måte: Definer $\mathbf{x}_0 = (\frac{1}{n}, \dots, \frac{1}{n})$, og

$$\mathbf{x}_k = M\mathbf{x}_{k-1}, k \geq 1,^1 \quad (5)$$

der M er som i a) og A forutsettes stokastisk. Sekvensen $\mathbf{x}_0, \mathbf{x}_1, \dots$ er da en *Markov-kjede*.

Programmer en Matlab-funksjon

```
rankingapprox(A,m,delta)
```

som beregner \mathbf{x}_k ved hjelp av (5) helt til vi støter på en k som er slik at $\max_j |\mathbf{x}_k(j) - \mathbf{x}_{k-1}(j)| < \text{delta}$; funksjonen returnerer da vektoren \mathbf{x}_k .

c) For `delta=0.005`, `m = 0.15` og link-matrisen fra oppgave 1, sammenlign vektorene du får fra funksjonene `ranking(A)` og `rankingapprox(A,delta)`.

Appendix

Motivasjon for ligning (1)

(1) kan motiveres ut fra en demokratisk modell for dokumenter på weben, der et dokument kan stemme på andre dokumenter ved å ha hyperlenker til disse. Dokumenter som får flest stemmer får høyest score.

Divisjonen med n_j i (1) er tatt med for at et web-dokument ikke skal få ekstra innflytelse ved kun å legge til hyperlenker til mange andre sider, og for at de n_j hyperlenkene fra dokument j skal telle like mye ved "opptelling" av stemmer [1].

I analogien med stemmegiving svarer kravet om at score-vektoren kun skal inneholde ikke-negative tall med sum 1 til at summen av alle stemmer blir 1 (i et demokratisk valg ville vi sagt 100%). I vår demokratiske modell for weben tillater vi ikke at dokumenter stemmer på seg selv, det vil si vi teller ikke hyperlenker fra et dokument til seg selv. Dette er et prinsipp som brukes ved stemmegiving i for eksempel Melodi Grand Prix, men det brukes ikke ved stortingsvalg (partiledere har da lov til å stemme på sitt eget parti?).

Ved hjelp av (3) kan vi tenke på surfing på weben som et *diskret dynamisk system* (seksjon 5.6), der overgangen fra en tilstand til en annen skjer når vi surfer fra en side til en annen. Vi kan tenke oss en viktig tolkning av score-vektoren i kontekst av dynamiske systemer: Anta at vi surfer fra side til side på weben, og hele tiden velger blant hyperlenkene på en side med like stor sannsynlighet. Anta og at vi bruker like mye tid på hver side før

¹Dette vil også kunne være regnekrevende, men i en praktisk implementasjon ville man kunne utnytte at A vil inneholde mange elementer som er 0.

vi surfer til en ny. Da vil score-vektoren gi oss andeler i tid vi bruker på hver side i det lange løp. I kapittel 4.9 blir også score-vektoren kalt for en *likevektstilstand*.

Referanser

- [1] K. Bryan and T. Leise, The 25,000,000,000 Eigenvector: The Linear Algebra behind Google, *SIAM Review*, **48**, 569-581, 2006
- [2] A. N. Langville and C. D. Meyer, *Google's pagerank and beyond: the science of search engine rankings*, Princeton University Press, 2006