Obligatorisk oppgavesett 1 – MAT1120 H16

Innleveringsfrist: torsdag 22/09 – 2016, innen kl 14.30.

Besvarelsen leveres på Matematisk institutt, 7. etasje i N.H. Abels hus. Husk å bruke forsiden som du finner via hjemmesiden.

Dersom du på grunn av sykdom eller andre tungtveiende grunner har behov for å utsette innleveringen, må du i god tid før innleveringsfristen sende søknad til:

studieinfo@math.uio.no

Husk at sykdom må dokumenteres ved legeattest.

Oppgavesettet består av tilsammen 10 oppgaver. For å få godkjent Oblig 1 kan høyst to av disse 10 oppgavene leveres blankt og det må komme klart frem fra din besvarelse at du har gjort et *seriøst* forsøk på å løse alle de andre oppagvene. Minst 6 av de 10 oppgavene må være besvart på en tilfredstillende måte, med en ryddig fremstilling og gode begrunnelser. Det vil også bli lagt vekt på at Matlab-delene i oppgavesettet er rimelig godt besvart – en besvarelse som viser mangelfulle Matlab-ferdigheter kan bli underkjent selv om den tilfredstiller de andre kravene. Der det står at Matlab skal brukes, må det vedlegges passende utskrifter med kommentarer. Det er tillatt å bruke Python (eller en annen programpakke enn Matlab), men husk at det vil kunne bli stilt spørsmål som kreves kjennskap til Matlab ved slutteksamen.

Studenter som ikke får sin opprinnelige besvarelse godkjent, men som har gjort et reelt forsøk på å løse oppgavesettet, vil få en mulighet til å levere en revidert besvarelse. Studenter som ikke får godkjent begge sine besvarelser til Oblig 1 og Oblig 2 vil ikke få adgang til avsluttende eksamen.

Det er lov å samarbeide om oppgavene. Men alle må levere sin egen personlige besvarelse og selv ha gjennomført alle Matlab-kjøringer. Er vi i tvil om at du virkelig ha forstått det du har levert inn, kan vi be deg om en muntlig redegjørelse.

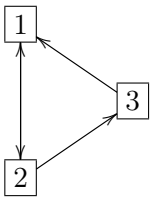
Det vises ellers til regelverket for obligatoriske oppgaver, som du finner via hjemmesiden til emnet.

Lineær algebra og Google-PageRank

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. 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.

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* (avsnitt 4.9 i læreboka). Oppgavene i denne obligen er ment å gi et lite innblikk om sammenhengen mellom lineær algebra og rangering av websider.

Vi vil representere 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 en webside, her kalt et dokument. Vi antar at vår web inneholder n dokumenter, og nummererer disse med $1, 2, ..., n$. En pil fra dokument nr. j til dokument nr. i svarer til en hyperlenke i dokument nr. j, som refererer til dokument nr. i.
Merk : vi betrakter bare weber der ingen dokument har hyperlenker til seg selv. Vi antar også at et dokument kan høyst ha en hyperlenke til et annet dokument.

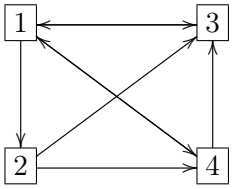


Figur 1: Et enkelt web med tre dokumenter

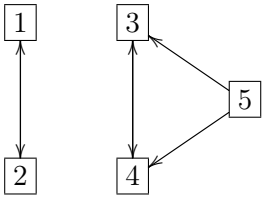
Til en gitt web med n dokumenter kan vi tilordne en $n \times n$ matrise A som kalles en *link-matrisen* til weben. Vi trenger først en definisjon.

Definisjon 1 *For hver $j \in \{1, 2, \ldots, n\}$ definerer vi n_j til å være antall dokumenter som dokument nr. j refererer til ved hjelp av hyperlenker.*

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



Figur 2: Web for oppgave 1



Figur 3: Web for oppgave 2

Alle koeffisientene i link-matrisen er ikke-negative tall mindre eller lik 1. Det er enkelt å sette opp link-matrisen til en web.

Som et eksempel, la oss sette opp link-matrisen for weben fra Figur 1. Den 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 nr. 1. Siden dokument nr. 1 bare har en hyperlenke til dokument nr. 2, og $n_1 = 1$, så får vi $(0, 1, 0)$.

Andre kolonne blir $(\frac{1}{2}, 0, \frac{1}{2})$ siden dokument nr. 2 har hyperlenker til dokument nr. 1 og dokument nr. 3, og $n_2 = 2$.

Tredje kolonne blir $(1, 0, 0)$ siden dokument nr. 3 har en hyperlenke bare til dokument nr. 1.

Setter vi disse tre kolonne-vektorene sammen til en matrise får vi link-matrisen

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

Vi er ute etter å bestemme fornuftige "scores" $x_1, x_2, \ldots, x_n$, der x_i skal være score til dokument nr. i. En score skal være et ikke-negativt tall mindre eller lik 1, og en høy verdi skal indikere stor relevans. Hvis tallene $x_1, x_2, \ldots, x_n$ har sum 1, så vil vi kalle vektoren $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ for en *score-vektor*. I avsnitt 4.9 i læreboka kalles en slik vektor $\mathbf{x}$ for en *sannsynlighetsvektor*.

En måte å regne ut en score-vektor $\mathbf{x}$ for en web med link-matrise A er å kreve at $\mathbf{x}$ er en løsning av likningen[1]

$$A\mathbf{x} = \mathbf{x} \tag{1}$$

eller alternativt av det homogene systemet

$$(A - I)\mathbf{x} = \mathbf{0} \tag{2}$$

der I betegner identitetsmatrisen. Som vi har lært, kan slike systemer 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 (2) sier at en score-vektor $\mathbf{x}$ tilhører nullrommet for matrisen $A-I$. Det er denne sammenhengen med lineær algebra vi skal bruke i oppgavene nedenfor.

Oppgave 1 *Skriv opp link-matrisen A for Figur 2. Bestem deretter 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 av dokumentene.*

Oppgave 2 *Figur 3 viser det vi kaller en* usammenhengende web*, det vil si at weben kan splittes i deler som ikke refererer til hverandre. Skriv opp link-matrisen, og bestem en basis for nullrommet til matrisen $A - I$. Finnes det en unik score-vektor?*

I avsnitt 4.9 i læreboka defineres det hva det vil si at en kvadratisk matrise er *stokastisk*: I en stokastisk matrise er alle koeffisientene ikke-negative og summen av koeffisientene i hver kolonne er lik 1. Videre kalles en stokastisk matrise *regulær* hvis en eller annen potens av matrisen inneholder bare ekte positive tall. Ut fra Teorem 18 i avsnitt 4.9 i læreboka vet vi at en regulær stokastisk matrise har en unik score-vektor.

Oppgave 3 *Er link-matrisene fra Oppgave 1 og 2 stokastiske? Er de regulære? Kan du forklare med enkle ord hvordan en web kan gi en link-matrise som ikke er stokastisk?*

[1] I et appendiks på slutten av oppgavesettet gir vi en motivasjon for likningen (5), som spesielt interesserte anbefales å lese.

Problemet med unik rangering viser seg å være lettere å adressere hvis link-matrisen er stokastisk. Hvis A er link-matrisen til en web med n dokumenter, og m er et valgt tall slik at $0 < m < 1$, definerer vi en ny $n \times n$ matrise M ved

$$M = (1 - m)A + mS, \tag{3}$$

der S er $n \times n$ matrisen der alle koeffisientene er $\frac{1}{n}$. M kalles ofte *Google-matrisen* til weben (for den valgte m). Legg merke til at M kan betraktes som link-matrisen til en sammenhengende web, selv om A representerer link-matrisen til en usammenhengende web.

Oppgave 4 *Anta at A er stokastisk. Begrunn at M da er stokastisk og regulær (og dermed har en unik score-vektor).*

Oppgave 5 *Skriv opp Google-matrisen M når link-matrisen A er den du fant i Oppgave 2, og $m = 0.15$. Bruk Matlab-kommandoen* `null(M-I)` *til å bestemme nullrommet til $M - I$. Angi så den unike score-vektoren til M.*

Oppgave 6 *I denne oppgaven skal du studere Matlab-koden under, som returnerer en $n \times n$ matrise A. Her gir kommandoen* `round(rand(n,n)` *en tilfeldig $n \times n$ matrise med bare nuller og enere.*

```
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 hvorfor denne funksjonen returnerer en link-matrise.

Oppgave 7 *Kjør Matlab-funksjonen fra Oppgave 6 med $n = 5$, og tegn den tilhørende weben for link-matrisen som funksjonen returnerer.*

Oppgave 8 *Skriv en Matlab-funksjon*

`ranking(A)`

som returnerer (den unike) score-vektoren $\mathbf{x}$ *for Google-matrisen* M *definert ved* (3) *med* $m = 0.15$ *når input-matrisen* A *er en stokastisk link-matrise. Hvis* A *ikke er stokastisk skal funksjonen returnere en feilmelding. Du kan gjerne bruke Matlab-kommandoen* `null(M-I)` *som i Oppgave 5.*

Siden World Wide Web etterhvert inneholder forferdelig mange dokumenter, så vil funksjonen `ranking` måtte regne ut nullrommet til veldig store matriser. Dette er regnekrevende. Vi kan i stedet forsøke å regne ut approksimasjoner til den (unike) score-vektoren for en Google-matrise M som i Oppgave 8 på følgende måte[2] (jf. Teorem 18 i avsnitt 4.9):

Vi setter $\mathbf{x}_0 = (\frac{1}{n}, \ldots, \frac{1}{n})$ og definerer $\mathbf{x}_1 = M\mathbf{x}_0$, $\mathbf{x}_2 = M\mathbf{x}_1$, osv.
Med andre ord,

$$\mathbf{x}_k = M\mathbf{x}_{k-1}\,, \quad \text{for } k \geq 1. \tag{4}$$

Sekvensen $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ er da en *Markov-kjede* assosiert med M. Husk at M er definert i (3) og at vi bruker $m = 0.15$.

Oppgave 9

Skriv en Matlab-funksjon

`rankingapprox(A, delta)`

som beregner $\mathbf{x}_k$ *ved hjelp av* (4) *helt til den støter på en* k *som er slik at* $\max_j |\mathbf{x}_k(j) - \mathbf{x}_{k-1}(j)| <$ `delta`*; funksjonen skal da returnere vektoren* $\mathbf{x}_k$*. Her antaes det at* A *er stokastisk og at* δ *er et positivt (lite) tall.*

Oppgave 10

Sammenlign vektorene som du får ut når du anvender funksjonene `ranking(A)` *og* `rankingapprox(A,delta)` *på link-matrisen* A *fra Oppgave 1 og du velger* `delta = 0.005`.

[2]Dette vil også kunne være regnekrevende, men i en praktisk implementasjon kan man utnytte at A vil inneholde veldig mange 0'ere.

Appendiks

Likning (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. Denne stemmegivningen gjøres gjentatte ganger. Hvert dokument begynner med en score, som oppgraderes etter hver avstemning.

Betrakt en web med n dokumenter. For $i \leq n$ definerer vi

$$L_i = \Big\{ j \in \{1, 2, \cdots, n\} \mid a_{ij} \neq 0 \Big\}$$

som svarer til dokumentene som har en hyper-link til dokument (nr.) i. Merk at L_i kan være den tomme mengden (dvs ingen hyper-link går til i), og dette svarer til at rad i i link-matrisen er nullvektoren. For weben i Figur 1 har vi for eksempel at $L_1 = \{2, 3\}, L_2 = \{1\}$ og $L_3 = \{2\}$.

For en web med n dokumenter og link-matrise A vil en vektor $\mathbf{x}$ tilfredstille likning (1) hvis og bare hvis vi har at

$$\sum_{j \in L_i} \frac{x_j}{n_j} = x_i \tag{5}$$

for alle i som er slik at $L_i \neq \emptyset$, mens $x_i = 0$ ellers.

En hyperlenke fra dokument j til dokument i (med andre ord, en stemme fra j til i) gjenspeiles altså med et summeledd $\frac{x_j}{n_j}$ på venstre side i (5). Divisjonen med n_j i (5) er tatt med for at de n_j hyperlenkene fra dokument j skal telle like mye ved "opptelling" av stemmer. 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 skal være lik 1 (i et demokratisk valg ville vi sagt 100%).

Når score-vektoren er unik kan den gis følgende tolkning: Anta at vi surfer fra side til side på weben, og hele tiden velger blant hyperlenkene på en side med like stor sannsynlighet. Anta også at vi bruker like mye tid på hver side før vi surfer til en ny. Da vil score-vektoren gi oss andeler i tid vi bruker på hver side i det lange løp.