Obligatorisk oppgavesett 2 – MAT1120 H16

Innleveringsfrist: torsdag 03.11.2016, 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 deloppgaver. For å få godkjent Oblig 2 kan høyst 2 av disse 10 deloppgavene leveres blankt og det må komme klart frem fra din besvarelse at du har gjort et *seriøst* forsøk på å løse alle de andre deloppgavene. Minst 6 av de 10 deloppgavene 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.

Om SVD og bildekomprimering

Et digitalt svart-hvitt bilde er ikke noe annet enn en matrise. Tenk deg et rektangulært rutenett hvor hver rute tilsvarer et element i matrisen. En rute kalles gjerne *et piksel* når vi snakker om bilder. I svart-hvitt-bilder har hvert piksel én verdi, og hver verdi angir hvilken gråtone pikselet har. Ofte lagres svart-hvitt bilder med 256 ulike gråtoner, det vil si som matriser med koeffisienter som tar verdier mellom 0 og 255, der 0 tilsvarer svart og 255 tilsvarer hvit. Fargebilder lagres som regel med såkalt rgb-format; hvert piksel får tre komponenter, én for rød (r), én for grønn (g), og den siste for blå (b); når disse tre fargene kombineres på ulike måter kan alle mulige farger produseres. Dette betyr at man trenger 3 matriser for å lagre et fargebilde. Vi skal holde oss til svart-hvitt bilder, slik at én matrise vil være nok for å lagre et bilde.

Her er en beskrivelse av noen Matlab-kommandoer som er relevante for denne obligen:

rank Kommandoen `r = rank(A)` returnerer (den estimerte) rangen til `A`.
Vi er vant til å radredusere matrisen A og telle antall pivot-kolonner for å finne rangen. Dersom vi setter en datamaskin til å gjøre utregningene ved rad-reduksjonen, vil ofte avrundingsfeil kunne føre til at det ser ut som om matrisen har full rang, selv om det egentlig ikke er tilfelle. I stedet kan datamaskinen beregne rangen ved å telle antall positive singulærverdier til A. Matlab bruker denne fremgangsmåten, og den regner da en singulærverdi som positiv dersom den er større enn ε, der toleransen $\varepsilon > 0$ er forutbestemt. Dersom man ønsker å forandre den forhåndsdefinerte toleransen som Matlab bruker i kommandoen `rank` kan man angi dette ved å legge til en ekstra parameter: man skriver da `r = rank(A,`ε`)`, der ε er den ønskede toleransen.

svd Kommandoen `[U,S,V] = svd(A)` returnerer en singulærverdidekomposisjon til matrisen A, det vil si den regner ut matrisene U, S og V slik at $A = USV^T$ blir en SVD for A. Merk: denne kommandoen virker bare hvis elementene i matrisen er av typen `double`. (Vanligvis er ikke dette et problem fordi dette skjer automatisk når du skriver selv inn matriser i Matlab). I noen av oppgavene må derfor matrisene konverteres til denne datatypen, f.eks. ved å skrive `A=double(A)`.

imread Kommandoen `A = imread('filnavn.fmt','fmt')` leser inn et bilde med angitt filnavn og format, og lagrer det som en matrise A (som ikke er av typen `double`, men av typen `uint8`). Formatet ('fmt') kan være 'jpg','tif', 'gif', 'png', med fler.

imwrite Kommandoen `imwrite(A, 'filnavn.fmt','fmt')` skriver bildet bestemt av en matrise A til en fil, med angitt filnavn og format, som i kommandoen imread. For å unngå feilmelding bør matrisen på forhånd være gjort om til typen `uint8` ved å skrive `A=uint8(A)`. Skal du printe ut bilder som du selv har generert, kan du bruke denne kommandoen og så sende filen til printeren med et annet program.

imageview Kommandoen `imageview(A)` viser frem matrisen A som et (lite) bilde. Du kan dessverre ikke skrive ut bildet fra menyen til dette vinduet.

Hovedformålet med denne obligen er å studere hvordan singulærverdidekomposisjonen til en matrise A kan brukes til å komprimere bilder, og derved spare datalagringsplass når bilder lagres som matriser.

Vi begynner med en oppgave om rang 1 matriser.

Oppgave 1

Betrakt $\mathbf{u} \in \mathbb{R}^m$ *og* $\mathbf{v} \in \mathbb{R}^n$ *og anta at begge ikke er null. Siden* $\mathbf{u}$ *er en* $m \times 1$ *matrise og* $\mathbf{v}^T$ *er en* $1 \times n$ *matrise, er* $\mathbf{u}\mathbf{v}^T$ *en* $m \times n$ *matrise.*

Begrunn at rank $\mathbf{u}\mathbf{v}^T = 1$. *Sjekk dette ved hjelp av Matlab: velg en tilfeldig vektor* $\mathbf{u}$ *i* $\mathbb{R}^4$ *og en tilfeldig vektor* $\mathbf{v}$ *i* $\mathbb{R}^5$*, skriv ut matrisen* $\mathbf{u}\mathbf{v}^T$ *og beregn dens rang.*

Vi minner nå om at *singulærverdiene* til en matrise A er definert som kvadratroten til egenverdiene til matrisen A^TA. Deretter minner vi om at hovedresultatet (Teorem 10) i avsnitt 7.4 i læreboka sier følgende:

La A være en $m \times n$ reell matrise, $A \neq 0$. Da har A en *singulærverdidekomposisjon* (forkortes til SVD), dvs. at den kan faktoriseres på formen

$$A = U\,\Sigma\,V^T$$

der U og V er ortogonale matriser av henholdsvis dimensjon $m \times m$ og $n \times n$, mens matrisen Σ er en $m \times n$ på formen

$$\Sigma = \begin{array}{|cccc|ccc|c|}
\hline
\sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 & \\
0 & \sigma_2 & \ddots & \vdots & 0 & \cdots & 0 & r \text{ rader} \\
\vdots & \ddots & \ddots & 0 & \vdots & \cdots & \vdots & \\
0 & \cdots & 0 & \sigma_r & 0 & \ldots & 0 & \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & m-r \text{ rader} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
\hline
\multicolumn{4}{|c|}{r \text{ kolonner}} & \multicolumn{3}{c|}{n-r \text{ kolonner}} & \\
\hline
\end{array}$$

Her er $r = \text{rank } A$. Elementene $\sigma_1, \ldots, \sigma_r$ angir alle singulærverdiene til A *som ikke er null*; alle disse er positive og det er vanlig å ordne disse slik at

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$

Vi antar at dette er gjort (dette vil være tilfelle når du bruker `svd`-kommandoen).

Oppgave 2

Som ovenfor, la $A = U\Sigma V^T$ være en SVD for en $m \times n$ matrise $A \neq 0$, med rank $A = r$. *Skriv U og V på formen*

$$U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r & \cdots & \mathbf{u}_m \end{bmatrix} \quad og \quad V = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_r & \cdots & \mathbf{v}_n \end{bmatrix}.$$

Vi minner om at dersom B er en $m \times n$ matrise og C er en $n \times p$ matrise, så kan matriseproduktet BC skrives som

$$BC = \sum_{j=1}^{n} \mathtt{kol}_j(B)\, \mathtt{rad}_j(C)\,. \tag{1}$$

Bruk likningen (1) *med $B = U\Sigma$ og $C = V^T$ til å begrunne at*

$$A = \sum_{j=1}^{r} \sigma_j\, \mathbf{u}_j \mathbf{v}_j^T\,. \tag{2}$$

Merk at Oppgave 1 gir at alle leddene i summen i likning (2) er matriser med rang 1. Du vil dermed ha vist at en matrise A med rang $r \geq 1$ kan skrives som en sum av r matriser med rang 1.

Vi bruker videre notasjonen fra Oppgave 2. For $1 \leq k \leq r$, definér

$$A^{(k)} = \sum_{j=1}^{k} \sigma_j\, \mathbf{u}_j \mathbf{v}_j^T. \tag{3}$$

Det kan vises at matrisen $A^{(k)}$ har rang k. Den kan derfor sees på som en rang k approksimasjon av A. Hvor stor k må være for å få en akseptabel approksimasjon, vil avhenge av hvordan singulærverdiene til A er fordelt og hvor raskt de avtar mot 0 (når k økes).

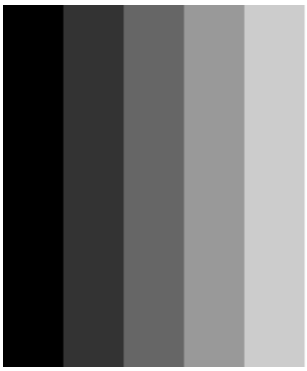
Oppgave 3

a) Lag en Matlab funksjon

$$\mathtt{functionAK = svdApprox(A, k)}$$

som tar inn en $m \times n$ matrise A og et naturlig tall k, regner ut en SVD for matrisen (ved hjelp av `svd`*-kommandoen i Matlab), og returnerer matrisen $A^{(k)}$ definert i* (3). *Dersom $k > r =$* `rank` *A skal funksjonen gi en passende feilmelding.*

b) Les inn bildet `mm.gif` *av Marilyn Monroe og gjør det om til en matrise A. Dette bildet finner du via linken som finnes på hjemmesiden, ved siden av teksten til selve obligen. Husk å konvertere A til* `double`*-type ved kommandoen* `A = double(A)`.



Figur 1: Et bilde med et spesielt enkelt mønster

Sjekk at matrisen A er en 256×256 matrise og bestem dens rang ved hjelp av `rank`*-kommandoen. Prøv også kommanden* $\texttt{rank}(A, \varepsilon)$ *med* $\varepsilon = 0.001$ *på denne matrisen.*

c) Bruk deretter funksjonen du laget i a) til å vise frem bildet som svarer til $A^{(k)}$ *for* $k = 8$ *og for* $k = 32$*. Legg ved en utskrift av disse to bildene.*

d) Når et bilde har (gjentatte) mønstre kan dette ofte utnyttes for å spare lagringsplass. Siden rangen til bildematrisen tilsvarer antall postive singulærverdier, vil lav rang generelt bety at bildet kan komprimeres mye.
Kan du si noe om rangen til matrisen for bildet i figur 1, bare ved å se på bildet?

Siden vi antar at singulærverdiene til A er ordnet i avtagende rekkefølge vil den "viktigste" informasjonen om A som oftest finnes i de første leddene i summen som angir A i (2). Det er dette vi vil utnytte for å komprimere bilder.

Oppgave 4

a) Bruk matrisen fra bildet av Marilyn Monroe og plott dens singulærverdier σ_j *som en funksjon av* j *(la* x*-aksen gå fra 1 til 256). Bruk Matlab-funksjonen* `plot` *til dette. Legg ved utskrift.*

b) Konstruer et tilfeldig bilde av samme størrelse og med intensitetsverdier i samme intervall som bildet av Marilyn Monroe; dette kan du gjøre i Matlab ved å bruke kommandoen

$$\texttt{B} = \texttt{round}(255 * \texttt{rand}(256, 256))$$

Vis fram bildet og legg ved utskrift av det. Plott singulærverdiene til matrisen av det tilfeldige bildet sammen med plottet fra punkt a). Sammenlikn hvordan singulærverdiene avtar for de to matrisene, og prøv å gi en forklaring på det du ser.

Vi bruker videre notasjonen fra Oppgave 2, og antar at A er en *bildematrise*, dvs. en matrise som vi har laget ut ifra et bilde.

Vi merker oss at hvert summeledd i (3) trenger kun to vektorer i tillegg til en singulærverdi. For å lagre matrisen $A^{(k)}$ trenger vi derfor bare å lagre de k første vektorene i begge matrisene U og V og de k første singulærverdiene. En rang k approksimasjon av A (og dermed av bildet), kan vi konstruere fra disse lagrede vektorer og verdier ved å bruke (3). Bildet som vi får frem fra $A^{(k)}$ kaller vi *det komprimerte bildet av rang k.*

Oppgave 5

a) Hvor mange tall trenger man å lagre for å kunne regne ut det komprimerte bildet av rang k til en $m \times n$ bildematrise A ?
Hvor stor k synes du er nødvendig for at det komprimerte rang k bildet av Marilyn Monroe er visuelt sett (så godt som) likt det opprinnelige? (Eksperimenter i Matlab!)

b) Den relative feilen e_{rel} mellom den opprinnelige A og approksimasjonen $A^{(k)}$ er definert ved

$$e_{rel} = \frac{\|A - A^{(k)}\|}{\|A\|}$$

hvor normen $\|B\|$ til en $m \times n$ matrise $B = [b_{i,j}]$ er definert ved

$$\|B\| = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} b_{i,j}^2}\,.$$

Programmer en Matlab funksjon

$$\texttt{functionerror} = \texttt{relError}(\texttt{A}, \texttt{AK})$$

som regner ut den relative feilen mellom en bildematrise `A` *og en approksimasjon* `AK` *av denne. Hva er den relative feilen mellom den opprinnelige A og approksimasjonen $A^{(k)}$ for den k'en som du anga i a) ovenfor?*

Ekstraoppgave (bare for spesielt interesserte!)

Vi bruker samme notasjon som innført rett før og etter Oppgave 2.

a) La $1 \leq k \leq r$. Vis at $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ er en basis for Col $A^{(k)}$. Dette viser spesielt at $A^{(k)}$ har rang k (slik vi nevnte i Oppgave 3). Merk også at denne basisen er da ortonormal.

b) La A svare til bildet av Marilyn Monroe. Vis frem bildet som svarer til A^T. Beskriv hvordan dette bildet fremkommer fra bildet til Marilyn Monroe. Beskriv deretter matrisen som er slik at det bildet du får frem er bildet av Marylin rotert med 90 grader mot urskiva.

c) Du rekonstruerte bildet av Marilyn Monroe i Oppgave 3c) bl.a. ved å bruke de 8 første singulærverdiene, det vil si ved hjelp av $A^{(8)}$. Forsøk nå å rekonstruere dette bildet ved å bruke alle singulærverdiene *bortsett fra* de 8 første, dvs. ved å bruke matrisen

$$\sum_{j=9}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

der $r = \texttt{rank}\, A$. Se på begge bildene, og regn ut den relative feilen for hvert av dem. Er den "visuelle" feilen i begge bildene akseptabel ?