MAT1120 Oblig 2

Eirik Lund

1. november 2016

1):

Jeg laget et lite script som genererer tilfeldige verdier for $\vec{U} \in R^4$ og $\vec{V^T} \in R^5$ Verdiene som genereres er mellom 1 og 10.

```
[a,b] = sort(rand(4,10));
 U = b(:,1:1);

 [c,d] = sort(rand(10,5));
 Vt = d(1:1,:);

 U*Vt

 rank(U*Vt)
```

Vi får følgende output:

```
>> oppgave1

ans =

    24    40    40     8    20
     6    10    10     2     5
    18    30    30     6    15
    12    20    20     4    10


ans = 1
```

Forklaring:

Vi bruker Theorem 14 fra Linear Algebra and it applications.
rank A + dim Nul A = n

Antall pivotkolonner + antall ikkepivot kolonner = antall kolonner

$\vec{U}$ består av en kolonne hvor vi klarer å lage en pivot kolonne. Siden den består av kun en pivotkolonne så får vi følgende:

rank(U) + 0 = n, 1 = 1.
$\vec{V^T}$ består av (1×5) hvor n = 5.
rank V^T + dim Nul V^T = n
1 + dim Nul V^T = 5

Hvor nullrommet vil bestå av en (5×4) matrise.
Ved å gange $\vec{U} \cdot \vec{V^T}$
Så vil vi få en (1×1) for rangen til matrisen og en (5×4) for nullrommet.

Dette er og på grunn av at hvis vi ganger en matrise med en annen vil ikke den linæreuavhengigheten endres.

2

Singulærverdiene for en $(m \times n)$ matrise:

Antar A er ortogonalt diagonalisert, hvor $\vec{v_1}..\vec{v_r}...\vec{v_m}$ er egenvektorene til UV^T

Vi har da at egenverdiene $\lambda_n \geq 0$

$\sigma_j = \sqrt{\lambda_j}$
Ortonormaliserer : $\vec{u_j} = \dfrac{1}{\parallel A\vec{v_j} \parallel} A\vec{v_j}$

$A\vec{v_j} = \sigma_j \vec{u_j}$, $(1 \leq j \leq r)$

$U = [\vec{u_1}...\vec{u_m}]$ og $V = [\vec{v_1}...\vec{u_n}]$ Disse multiplisert gir $(m \times n)$

$U\Sigma = [\vec{u_1}...\vec{u_m}][\vec{\sigma_1}...\vec{\sigma_r}]$ Lengden utover r returnerer nuller fordi dette er utenfor størrelsen til ranken = r.

$U\Sigma = [\vec{u_1}\ \vec{\sigma_1}\ ...\ \vec{u_r}\vec{\sigma_r} \quad 0..0_m]$

V^T vil gi radstørrelsen n

$A = \sum_{j=1}^{r} \sigma_j U_j V_j^T$

Så vi itererer kun over størrelsen til rangen og matrisen har en rang mer eller lik.

3

a)
```
function Newpic=svdApprox(A,k)

  if k>rank(A)
    disp('K higher than rank(A)');
    return
  end

  [U,S,V] = svd(A);
  Newpic = zeros(size(A));

  for j=1:(k)
    Newpic = Newpic + S(j,j)*U(:,j)*V(:,j)';

  end

  imwrite(Newpic, 'Newpic.gif','gif');
```
» A = [1 2 3 4 5; 5 4 3 2 1;1 2 3 4 5; 1 2 3 4 5; 1 2 3 4 5]

» Svdapprox(A,3)

print2 =
K is higher than rank

b) » A =imread('mm.gif');
» A = double(A);
» rank(A)

ans =

256

» size(A)

ans =
256 256

» rank(A,0.001)

ans =
256
Vi får ønsket rank og dimensjon.

c) Med en valgt k = 8 så får vi et bilde som er relativt uklart.

Med en valgt k = 32 så får vi et bilde som er mye tydeligere enn forrige.

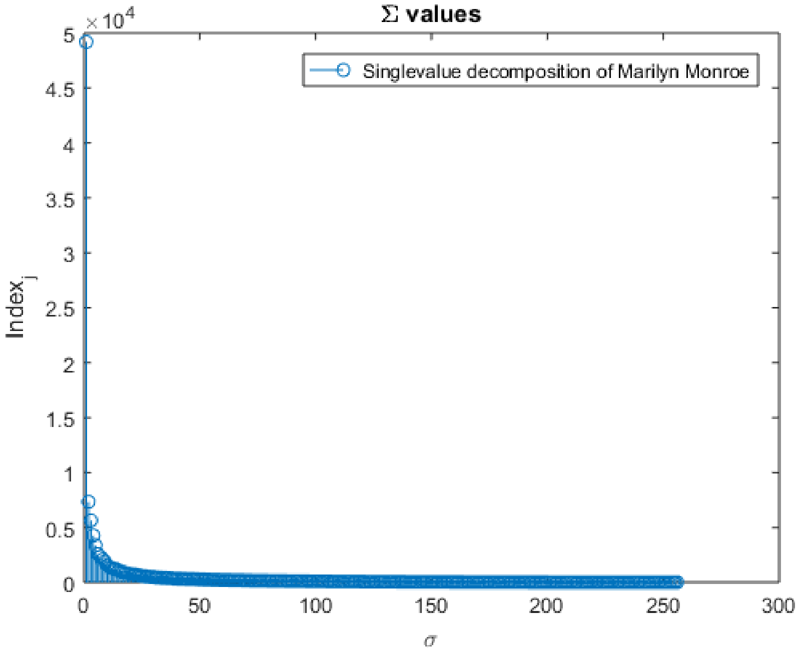
d) Vi ser på figur 1 at fargetonene går fra svart til grå, hvor radene er like store. Dimensjonen til radrommet vil da være 1 og rangen derfor også 1.

4

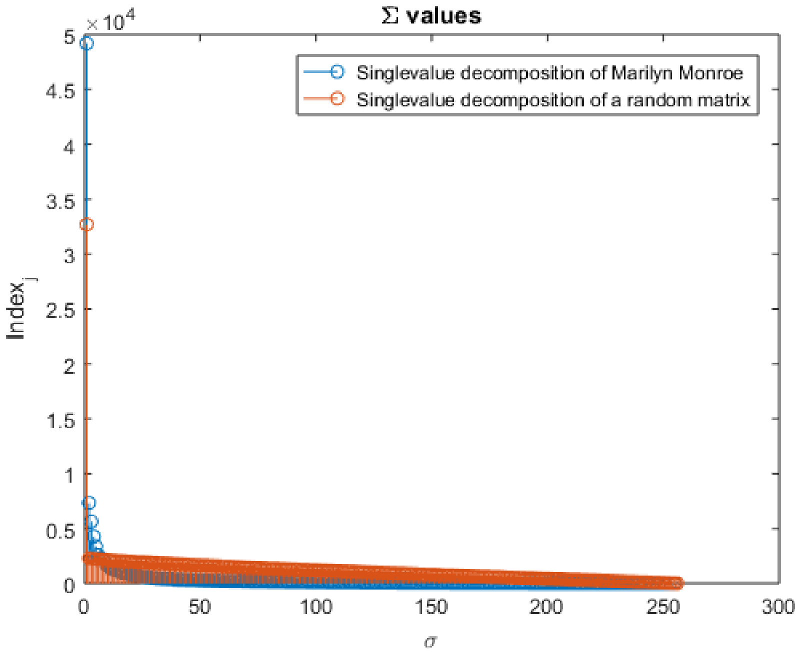
a) Lagde følgende script:

```
A = imread('mm.gif','gif');
B = round(255*rand(256,256));

A = double(A);
B = double(B);

[U,S,V] = svd(A);
[U,SS,V] = svd(B);
s = diag(S);
ss = diag(SS);

n = length(A);
n2 = length(B);
x = linspace(1, n, n);
x2 = linspace(1, n2,n2);

figure(1)
stem(x,s)
title('\Sigma values')
xlabel('\sigma')
ylabel('Index_{j}')
en =('Singlevalue decomposition of Marilyn Monroe')

%figure(2)
hold('on')
stem(x2,ss)
title('\Sigma values')
xlabel('\sigma')
ylabel('Index_{j}')
to =('Singlevalue decomposition of a random matrix')
legend(en,to)
```

Vi får følgende plot:

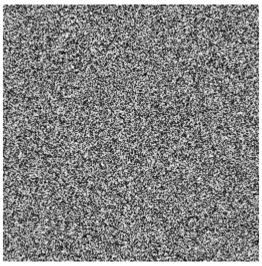


b) Med en tilfeldig matrise får vi følgende plot:



Det kan se ut til at mye av informasjonen om bildet ligger ved få σ'er, som vi så ved plottet av Marilyn Monroe ved k = 8.

Hvis vi skriver til fil: » imwrite(B, 'B.gif','gif')



Som vi ser så består bildet av tilfeldig støy og derfor en linært synkende funksjon for σ siden det krever flere itererasjoner for å gjenskape bildet.

5

a) Vi finner bits:

» info = imfinfo('Newpic.gif');
» bitdepth = info.BitDepth

bitdepth =
8

Så det vil bli: 8 bits $\cdot(k + km + kn)$
Etter å ha testet litt med svdApprox.m så gir ulike verdier for $k \in [50, 60]$ liten visuell forskjell.
Her er et bilde med k = 60, » svdApprox(A, 60);

c)

```
function functionerror = relError(A,AK)
A = double(A);
AK = double (AK);
C = norm(A-AK)/norm(A)
C*100

end
```

fra a) så har jeg lagret Newpic.gif fra svdApprox.m med en valgt k = 60.
» AK = imread('Newpic.gif','gif'); » relError(A,AK)
C =
0.0051
ans =

0.5101

Gir en relativ feil på 0.51 prosent som er ganske bra.