



Aalto-yliopisto

MS-A0002 Matriisilaskenta

SVD-demo

(Perustuu Juha-Matti Perkkiön laatimaan hahmontunnistusharjoitukseen.)

Käynnistä Matlab ja käy lataamassa kurssin MyCourses-sivulta tiedosto `kuvatiedosto.mat`, joka löytyy zip-muodossa viimeisen luennon materiaalina. Vaihda Matlabin työhakemistoksi se, johon tallensit kyseisen tiedoston, ja anna komennot

```
load kuvatiedosto
who
```

Matlabin muistiin ilmestyi 10304×400 -kokoinen matriisi `KUVA`, jonka kussakin sarakkeessa on pystyvektoriksi venytetty 112×92 -kokoinen harmaasävykuva. Kuvissa esiintyy 40 eri henkilöä, joista kustakin on 10 eri valokuvaa. Kuvat on järjestetty niin, että matriisin sarakkeissa

$$10(n-1) + 1, \dots, 10n$$

on kuvat samasta henkilöstä, jolle on tässä annettu järjestysnumero $1 \leq n \leq 40$. Kuvat on saatavilla ilmaiseksi PNG-muodossa AT&T Laboratories Cambridgen [www-sivulta](http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html)

<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> .

Sieltä löytyy myös kätevä esikatseluversio, jossa näkyy yhtä aikaa kaikki 400 kuvaa.

Luodaan funktio, jonka avulla voidaan kätevästi katsella sarakkeista löytyviä kuvia. Muodosta tiedosto nimeltä `nayta.m`, ja kirjoita sinne seuraava koodinpätkä:

```
function nayta(kuva)
imagesc(reshape(kuva, 112, 92))
colormap gray
colorbar
axis equal tight
```

Nyt sarakkeesta n löytyvä kuva saadaan näkyviin kirjoittamalla

```
figure(1)
nayta(KUVA(:,n))
```

Selaa läpi matriisista `KUVA` löytyviä kuvia.

Yritetään selvittää, miten hyvin tietokoneen saa tunnistamaan ihmiskasvoja. Muodostetaan ensin keskimääräinen kasvokuva `KA`, ja sen jälkeen matriisi `X`, joka sisältää kuvat, joista on poistettu tämä keskiarvokuva:

```
KA = sum(KUVA')'/400; % Laskee sarakkeiden keskiarvon
X = zeros(size(KUVA));
for n=1:400; X(:,n) = KUVA(:,n)-KA; end
```

Katso `nayta`-komennolla, miltä keskiarvokuva `KA` ja matriisista `X` löytyvät kuvat näyttävät.

Seuraavaksi muodostetaan matriisin `X` singulaariarvohajotelma komennolla

```
[U,S,V] = svd(X,0);
```

Tässä saattaa mennä hetki. Matriisin $X \in \mathbb{C}^{m \times n}$ singulaariarvohajotelma on $X = USV^*$, missä matriisien $U \in \mathbb{C}^{m \times n}$ ja $V \in \mathbb{C}^{n \times n}$ sarakevektorit ovat ortonormeerattuja, ja $S \in \mathbb{C}^{n \times n}$ on diagonaalimatriisi, jonka alkiot ovat ei-negatiiviset singulaariarvot laskevassa järjestyksessä.¹ Erityisesti V on unitaarinen, eli $V^{-1} = V^*$. Alkuperäisen matriisin `X` sarakkeet saadaan esiin seuraavalla käskyllä:

```
X(:,n) = U*S*V(n,:)' ;
```

Singulaariarvohajotelmassa matriisin `X` sarakevektorit esitetään siis lineaarikombinaatioina ortonormaaleista sarakevektoreista `U(:,n)`, joista ensimmäisiä vastaavat kertoimet ovat tyypillisesti häntäpäin kertoimia suurempia johtuen singulaariarvojen esitysjärjestyksestä. Matriisin `U` sarakkeet ovat tietysti mielessä kuvakirjaston “merkittäviä piirteitä”. Tarkastele näitä kirjoittamalla `nayta(U(:,n))` jollain $n = 1, 2, \dots, 400$.

Matriisia `X` voidaan approksimoida singulaariarvohajotelman avulla. Singulaariarvojen suhteista voidaan päätellä, kuinka monta ensimmäistä matriisin `U` pystyriviä kannattaa ottaa mukaan. Anna komento

```
var = (diag(S).^2);
varpart = cumsum(var)/sum(var);
plot(varpart)
```

Nyt kuvaajasta näkyy, kuinka suuri osa singulaariarvojen neliösummasta on mukana, jos matriisista `U` otetaan mukaan annettu määrä sarakkeita.

Katsotaan, miltä eripituiset approksimaatiot näyttävät. Pystyvektoria `X(:,n)` vastaava approksimaatio `Y`, jossa on mukana k ensimmäistä komponenttia, saadaan näkyviin käskyillä

```
Y = U(:,1:k)*S(1:k,1:k)*V(n,1:k)';
nayta(KA + Y)
```

Kokeile, miltä kuvat näyttävät eripituisilla approksimaatioilla.

Pudotetaan seuraavaksi singulaariarvohajotelmasta 300 viimeistä komponenttia:

¹Huomaa, että luennoilla esitetystä muodosta singulaariarvohajotelman matriiseille pätee $U \in \mathbb{C}^{m \times m}$, $S \in \mathbb{C}^{m \times n}$ ja $V \in \mathbb{C}^{n \times n}$. Tällaisen hajotelman Matlab tuottaisi komennolla `[U,S,V] = svd(X)`. Jos $m > n$, lisävipu “0” `svd`-komennon argumentissa jättää matriisin `S` alareunasta pois turhan nolliasta koostuvan lohkon sekä matriisista `U` viimeiset $m - n$ saraketta, jotka eivät itse asiassa vaikuta matriisin `X` rakenteeseen.

```
U = U(:, 1:100);
C = S(1:100, 1:100)*V(:, 1:100)';
```

Nyt siis approksimaatio kuvalle KUVA(:, n) saadaan seuraavasti:

```
nayta(KA + U*C(:, n))
```

Huomaa, kuinka paljon datan määrä tippui, ja kuinka kunkin yksittäisen kasvon (suurpiirteiseen) koodaamiseen riittää alkuperäisen 10304 luvun sijasta vain 100 lukua, eli yksi 100×400 matriisin C sarake. Lisäksi olisi hyvin todennäköistä, että jonkin aivan uuden kasvokuvan koodaamiseen kelpaisi sama 10304×100 -matriisi U, eli jopa kirjastosta puuttuvien kasvojen koodaus onnistuisi todennäköisesti ainoastaan sadalla luvulla.

Kokeillaan seuraavaksi, miten hahmontunnistus onnistuu edellä muodostettujen pakattujen kuvien avulla. Oletetaan, että tietokone on ‘nähty’ kustakin neljästäkymmenestä henkilöstä yhdeksän eri kuvaa, jotka virittävät yhdeksänulotteiset aliavaruudet $W^n \subset \mathbb{R}^{103042}$ kullekin henkilölle $n = 1, \dots, 40$, ja annetaan tietokoneen laskea lopuille 40 kuvalle — yksi kutakin koehenkilöä kohden — etäisyydet näistä aliavaruuksista. Pienin etäisyys vastaa suurella todennäköisyydellä sitä, että kyseessä on sama henkilö. Koska matriisin U pystyvektorit ovat ortonormeerattuja, etäisyyksien laskemisen voi yhtä hyvin suorittaa sataulotteisessa komponenttiavaruudessa, jossa pakattuja kasvokuvia vastaavat matriisin C pystyvektorit, eli kunkin kuvan painot ‘piirrekirjaston’ U suhteen. Toisin sanon kuvan nro n approksimaatiota $U*C(:, n)$ vastaa sataulotteinen komponenttivektori $C(:, n)$, ja sisätuloille (ja sitä kautta etäisyyksille) pätee

$$(U*C(:, n))' * (U*C(:, k)) = C(:, n)' * C(:, k).$$

Muodosta tiedosto nimeltä tunnistuskoe.m, ja kirjoita sinne seuraava koodinpätkä:

```
P = cell(1, 40);

for n=1:40
    [Q,R] = qr(C(:, (n-1)*10+1:n*10-1), 0);
    P{n} = Q*Q';

    for m=1:40
        D(m,n) = norm(C(:, 10*m) - P{n}*C(:, 10*m));
    end
end
```

Nyt P{n} on projektiomatriisi sarakevektoreiden

```
C(:, (n-1)*10+1), ..., C(:, n*10-1)
```

²Aliavaruus W^n sisältää kaikki annettujen yhdeksän ‘vektoroidun kuvan lineaarikombinaatiot.

virittämään aliavaruuteen, ja $D(m, n)$ on vektorin $U^*C(:, 10*n)$ etäisyys aliavaruudesta $W^m \subset \mathbb{R}^{10304}$. Aja tämä pieni skripti komennolla `tunnistus` ja tutki matriisia D . Kuinka monen henkilön kohdalla tunnistus meni oikein?

Vihje: Kokeile esim. komentoa `[d, I] = min(D)`, jossa vaakavektorin I alkio nro j kertoo rivin, jolla matriisin D sarakkeen nro j pienin alkio sijaitsee. Tunnistus on onnistunut, jos pienimmät alkiot/etäisyydet ovat matriisin diagonaalilla.