



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MATIAS ALA-SUNILA
SINGULAARIARVOHAJOTELMA

Kandidaatin työ

Tarkastaja:
Yliopisto-opettaja Mika Mattila

TIIVISTELMÄ

MATIAS ALA-SUNILA: Singulaariarvohajotelma

Tampereen teknillinen yliopisto

Kandidaatin työ, 15 sivua, 2 liitesivua

Syyskuu 2018

Teknis-luonnontieteellinen koulutusohjelma

Pääaine: Matematiikka

Tarkastaja: Yliopisto-opettaja Mika Mattila

Avainsanat: Singulaariarvohajotelma, matriisi, hermiittinen matriisi, häiriöalttius, pseudoinverssi

Tutkielmassa esitellään singulaariarvohajotelma ja sen tärkeitä ominaisuuksia. Singulaariarvohajotelman avulla mikä tahansa matriisi voidaan lausua kolmen matriisin tulona. Suurista matriiseista eli suurista taulukoista, joissa alkioina on numeroita, halutaan usein selville merkittävimmät rivit tai sarakkeet, jotka saadaan helposti selville singulaariarvohajotelman avulla. Tämän jälkeen voidaan kyseistä tietoa hyödyntää suoraan tai sen avulla tehdä paras approksimaatio alkuperäisestä matriisista. Hajotelman avulla voidaan myös muodostaa matriisille yleistetty käänteismatriisi.

Työssä keskitytään todistamaan singulaariarvohajotelman olemassaolo sekä esitellään lyhyesti muutama hyödyllinen sovellus, joissa singulaariarvohajotelman aiemmin esiteltyjä ominaisuuksia voidaan soveltaa käytännössä. Työssä oletetaan lukijan omaavan perustiedot matriisilaskennasta. Perustietämyksen ulkopuolelta teoriaosuudessa käydään läpi huolellisesti muun muassa hermiittinen matriisi sekä semidefiniittisyys. Teoriaosuudessa selvitetään, kuinka käytännössä singulaariarvohajotelma lasketaan koneella. Työhön sisältyy myös itse toteutettu ohjelma kuvan pakkaamiseksi singulaariarvohajotelman avulla.

SISÄLTÖ

1. Johdanto	2
2. Teoreettinen tausta	3
2.1 Singulaariarvohajotelman olemassaolo	4
2.2 Singulaariarvohajotelman ominaisuuksia	5
2.3 Singulaariarvohajotelman laskeminen käsin	6
3. Sovelluskohteita	8
3.1 Datan pakkaaminen	8
3.2 Pienimmän neliösumman menetelmä	11
3.3 Kohinainen data	13
4. Yhteenveto	15
Lähteet	16
Liite 1: Koodi kuvan pakkaamiseksi singulaariarvohajotelman avulla	17

TERMIT JA SYMBOLIT

I	Identiteettimatriisi
A, B	Matriiseja
A^T	Matriisin A transpoosi
A^*	Matriisin A konjugaattitranspoosi
$\ A\ $	Matriisnormi
SVD	Singulaariarvohajotelma (Singular value decomposition)
\mathbf{x}, \mathbf{y}	Vektoreita
\mathbf{x}^T	Vektorin \mathbf{x} transpoosi
\mathbf{x}^*	Vektorin \mathbf{x} konjugaattitranspoosi

1. JOHDANTO

Matriisit ovat taulukoita, joihin on tallennettu tietoa numeroarvoina. Tällaisia taulukoita löytyy melkein kaikilta tieteenaloilta, sillä monessa tutkimuksessa kerätään aineistoa, joka on helposti tallennettavissa matriisina. Tietoa halutaan tallentaa matriisimuodossa, sillä tietokoneet kykenevät tehokkaasti laskemaan laskuja matriisien avulla. Matriisien käytännöllisyydestä johtuen niille kehitettiin paljon käyttökohteita tilastotieteessä. Tilastotieteessä matriisit mahdollistivat työskentelyn isojenkin aineistojen parissa. Suuria aineistoja ei pystytä tutkimaan käsin vaan tietokoneen täytyy muokata aineistoa, jotta päästään käsiksi haluttuun tietoon. Matriisien käsitteelyyn liittyvät olennaisesti erilaiset matriisihajotelmat.

Tässä työssä tarkastellaan erityisesti yhtä yleisintä matriisihajotelmaa eli singulaariarvohajotelmaa. Singulaariarvohajotelmassa alkuperäinen matriisi lausutaan kolmen eri matriisin avulla, joista saadaan tietoa alkuperäisestä matriisista tai voidaan tehdä aproksimaatioita alkuperäisestä matriisista.

Työssä käydään ensin läpi teoriaa ja todistetaan singulaariarvohajotelman olemassaolo sekä perustellaan samalla tiettyjä hajotelman ominaisuuksia. Teoriaosuus perustuu pitkälti Hornin ja Johnsonin kirjan [2] Lukuun 3.1. Teorian jälkeen singulaariarvohajotelmalle voidaan löytää paljon sovelluskohteita, joista muutama esitellään seuraavassa luvussa 3. Lopuksi yhteenvedossa pohditaan hajotelman positiivisia ja negatiivisia puolia.

2. TEOREETTINEN TAUSTA

Singulaariarvohajotelman olemassaolon todistamiseksi täytyy aluksi määritellä matriisin konjugaattitranspoosi sekä hermiittinen matriisi. Matriisin $A \in \mathbb{C}^{m \times n}$ konjugaattitranspoosi saadaan, kun jokaisesta matriisin A alkioista otetaan sen kompleksikonjugaatti ja lopuksi saatu matriisi transponoidaan. Mikäli A olisi reaalinen, niin $A^* = A^T$.

Määritelmä 1. Matriisi $A \in \mathbb{C}^{m \times n}$ on hermiittinen, mikäli $A = A^*$.

Singulaariarvoja varten on hyvä todistaa, että jokaisella matriisilla A matriisitulo A^*A on hermiittinen ja vastaavasti reaalilla matriisilla symmetrinen.

Lause 1. Tulo A^*A on hermiittinen jokaisella matriisilla $A \in \mathbb{C}^{m \times n}$.

Todistus. Olkoon $A \in \mathbb{C}^{m \times n}$. Peruskursseilla on jo todistettu, että $(BC)^* = C^*B^*$ ja $(B^*)^* = B$ millä tahansa matriiseilla $B \in \mathbb{C}^{m \times n}$ ja $C \in \mathbb{C}^{n \times p}$. Näitä hyödyntäen saadaan, että $(AA^*)^* = A^*(A^*)^* = A^*A$. \square

Samalla tavalla voidaan todeta, että myös matriisitulo AA^* on hermiittinen. Hermiittisyys todistettiin, jotta päästään seuraavaan tärkeään lauseeseen.

Lause 2. Hermiittisen matriisin ominaisarvot ovat reaalisia.

Todistus. Olkoon $A \in \mathbb{C}^{m \times n}$ hermiittinen matriisi ja $\mathbf{x} \in \mathbb{C}^m$ matriisin A ominaisarvoa λ vastaava ominaisvektori. Tällöin

$$\begin{aligned}
 A\mathbf{x} &= \lambda\mathbf{x} && \text{(kerrotaan vasemmalta } \mathbf{x}^*) \\
 \Rightarrow \mathbf{x}^*A\mathbf{x} &= \mathbf{x}^*\lambda\mathbf{x} && ((\)^*) \\
 \Rightarrow \mathbf{x}^*A^*\mathbf{x} &= \bar{\lambda}\mathbf{x}^*\mathbf{x} && \text{(Hermiittisyys)} \\
 \Rightarrow \mathbf{x}^*A\mathbf{x} &= \bar{\lambda}\mathbf{x}^*\mathbf{x} && (\|\mathbf{x}\| = \sqrt{\mathbf{x}\mathbf{x}^*}) \\
 \Rightarrow \lambda\|\mathbf{x}\|^2 &= \bar{\lambda}\|\mathbf{x}\|^2 && (\|\mathbf{x}\| > 0, \mathbf{x} \neq \mathbf{0}) \\
 \Rightarrow \lambda &= \bar{\lambda} \Rightarrow \lambda = \mathbb{R}
 \end{aligned}$$

\square

Reaalisuuden lisäksi voidaan tutkia ominaisarvojen etumerkkiä. Seuraavaksi määriteltävän semidefiniitin matriisin ominaisarvot ovat aina ei-negatiivisia [3, p. 230].

Määritelmä 2. Hermittinen matriisi A on positiivisesti semidefiniitti, jos tulon $\mathbf{x}^*A\mathbf{x}$, missä $\mathbf{x} \in \mathbb{C}^n$, tulos on aina ei-negatiivinen reaaliluku.

Aikaisemmin hermiittiseksi todistettu matriisitulo on itseasiassa myös positiivisesti semidefiniitti, joka on helppo todistaa.

Lause 3. Matriisi AA^* on positiivisesti semidefiniitti jokaisella matriisilla $A \in \mathbb{C}^{m \times n}$.

Todistus. Olkoon $A \in \mathbb{C}^{m \times n}$ ja $\mathbf{x} \in \mathbb{R}^n$. Merkitään $\mathbf{y} = A^*\mathbf{x}$. Näin saadaan muokattua tulo $\mathbf{x}^*AA^*\mathbf{x}$ muotoon $\mathbf{x}^*A\mathbf{y}$. Huomataan samalla, että $\mathbf{x}^*A = \mathbf{y}^*$. Näin ollen $\mathbf{x}^*AA^*\mathbf{x} = \mathbf{y}^*\mathbf{y}$ ja skalaari $\mathbf{x}^*AA^*\mathbf{x}$ on ei-negatiivinen, sillä vektorinormin neliön $\mathbf{y}^*\mathbf{y}$ arvo on aina ei-negatiivinen. \square

2.1 Singulaariarvohajotelman olemassaolo

Matriisin A singulaariarvot ovat yhtäsuuret kuin matriisitulon A^*A ominaisarvojen neliöjuuret. Spektraaliteoria toteaa, että mikäli reaalinen matriisi A on symmetrinen, niin se on ortogonaalisesti diagoonisoituva. [4, s. 400]. Spektraaliteoriaa voidaan soveltaa myös matriisitulolle AA^* ja todeta, että neliömatriisilla on aina täysi määrä ominaisarvoja, mutta ne voivat olla kompleksisia. Aiemmin kuitenkin todettiin matriisitulon AA^* olevan hermiittinen [1] sekä positiivisesti semidefiniitti [3], joten ominaisarvot ovat reaalisia ja ei-negatiivisia. Näin ollen matriisin A singulaariarvot ovat myös aina ei-negatiivisia reaalilukuja.

Määritelmä 3 (Singulaariarvohajotelma). Olkoot matriisi $A \in \mathbb{C}^{m \times n}$ ja $\Sigma = \text{diag}(\sigma_1, \sigma_2 \dots \sigma_n)$ diagonaalimatriisi, missä diagonaalilla on ensin matriisin A singulaariarvot $\sigma_1, \sigma_2 \dots \sigma_r$ ja sen jälkeen $\sigma_{r+1} = \sigma_{r+2} = \dots = \sigma_n = 0$. Tällöin on olemassa sellaiset unitaariset matriisit $U \in \mathbb{C}^{m \times m}$ ja $V \in \mathbb{C}^{n \times n}$, että $A = U\Sigma V^*$

Todistus. Olkoon $\mathbf{v} \in \mathbb{C}^n$, jokin sellainen yksikkövektori, että $\sigma_1 = \|A\mathbf{v}\|_2$, missä σ_1 on matriisin A suurin singulaariarvo ja $\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$. Matriisi V voidaan muodostaa matriisin A^*A ominaisvektoreista. Mikäli $m \leq n$, niin U saadaan suoraan muodostettua matriisin V ominaisvektoreista. Muulloin täytyy täydentää matriisin V kantaa, jotta saadaan n kappaletta ortonormaaleja vektoreita. Muodostetaan yksikkövektori $\mathbf{u}_1 = A\mathbf{v}/\sigma_1$. Voidaan löytää sellaiset vektorit $[\mathbf{u}_2, \mathbf{u}_3, \dots, \mathbf{u}_m]$

ja $[\mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_n]$, että matriisit $U = [\mathbf{u}_1 \mathbf{u}_2 \mathbf{u}_3 \dots \mathbf{u}_m]$ ja $V = [\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3 \dots \mathbf{v}_n]$ ovat unitaarisia. Tällöin

$$A_1 = U_1^* A V_1 = \begin{bmatrix} \mathbf{u}_1^* \\ \mathbf{u}_2^* \\ \vdots \\ \mathbf{u}_m^* \end{bmatrix} [A \mathbf{v} \ A \mathbf{v}_2 \ \dots \ A \mathbf{v}_n] = \begin{bmatrix} \mathbf{u}_1^* \\ \mathbf{u}_2^* \\ \vdots \\ \mathbf{u}_m^* \end{bmatrix} [\sigma_1 \mathbf{v} \ A \mathbf{v}_2 \ \dots \ A \mathbf{v}_n]. \quad (2.1)$$

Vektorit voidaan yhdistää matriisiksi niin, että $A_2 \in M^{(m-1) \times (n-1)}$ ja saadaan

$$\begin{bmatrix} \sigma_1 & \mathbf{u}^* A \mathbf{v}_2 & \dots & \mathbf{u}^* A \mathbf{v}_n \\ 0 & & & \\ \vdots & & A_2 & \\ 0 & & & \end{bmatrix} = \begin{bmatrix} \sigma_1 & \mathbf{z}^* \\ 0 & A_2 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{u}^* A \mathbf{v}_2 \\ \mathbf{u}^* A \mathbf{v}_3 \\ \vdots \\ \mathbf{u}^* A \mathbf{v}_n \end{bmatrix}. \quad (2.2)$$

Muodostetaan nyt yksikkövektori $\mathbf{c} = [\sigma_1 \mathbf{z}^*] / \sqrt{\sigma_1^2 + \mathbf{z}^* \mathbf{z}}$ ja lasketaan $\|A(V_1 \mathbf{c})\|^2$. Unitaarisella matriisilla kertominen ei muuta vektorin normia, joten $\|A(V_1 \mathbf{c})\|^2 = \|(U_1^* A V_1) \mathbf{c}\|^2$. Kertominen unitaarisella matriisilla U_1 tehtiin, sillä nyt voidaan sieventää lauseketta ja saadaan, että $\|(U_1^* A V_1) \mathbf{c}\|^2 = \|A_1 \mathbf{c}\|^2$. Sijoitetaan nyt yksikkövektori \mathbf{c} sievennettyyn lausekkeeseen

$$\|A_1 \mathbf{c}\|^2 = [(\sigma_1^2 + \mathbf{z}^* \mathbf{z})^2 + \|A_2 \mathbf{z}\|^2] / (\sigma_1 + \mathbf{z}^* \mathbf{z}) \geq \sigma_1^2 + \mathbf{z}^* \mathbf{z}. \quad (2.3)$$

Mikäli nyt $\mathbf{z} \neq 0$, niin saatu lauseke on suurempi kuin σ_1 mikä on ristiriidassa σ_1 :n määrittelyn kanssa. Päädytään siihen, että vektorin \mathbf{z} on pakko olla nollavektori ja näin ollen

$$A_1 = U_1^* A V_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & A_2 \end{bmatrix}. \quad (2.4)$$

Samaa jatketaan, kunnes jokainen singulaariarvo on käyty läpi. Tällöin ollaan saatu muodostettua matriisi Σ ja hajotelma on valmis. \square

2.2 Singulaariarvohajotelman ominaisuuksia

Singulaariarvohajotelman todistuksesta on hyvä huomata, että matriisilla A on r kappaletta positiivisia singulaariarvoja, joita on yhtä monta kuin matriisitulolla $A^* A$ on positiivisia ominaisarvoja. Matriisilla A on yhtä monta positiivisia singulaariarvoa kuin matriisilla AA^* on nollaa suurempia ominaisarvoja. Matriisin AA^* asteikin on yhtä suuri kuin nollaa suurempien singulaariarvojen määrä, joka on samalla

matriisin A aste, sillä $\text{rank}(A) = \text{rank}(AA^*)$ [4, s. 205]. Matriisin aste on siis helposti luettavissa singulaariarvohajotelmasta. Yleensä, kuten todistuksessakin, singulaariarvot esitetään suuruusjärjestyksessä. Huomioitavaa on myös se, että yleensä diagonaalimatriisi Σ ei ole symmetrinen paitsi ainoastaan silloin, kun alkuperäinen matriisi A on neliömatriisi.

2.3 Singulaariarvohajotelman laskeminen käsin

Otetaan laskuesimerkiksi alla näkyvä matriisi A ja lasketaan valmiiksi kyseisen matriisin konjugaattitranspoosi sekä matriisitulo A^*A . Saadaan

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -2 & 2 \end{bmatrix}, A^* = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 2 \end{bmatrix} \text{ ja } A^*A = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}. \quad (2.5)$$

Ensiksi lasketaan matriisin singulaariarvot, jotka saadaan matriisitulon A^*A ominaisarvoista. Ominaisarvot voidaan laskea yhtälöstä $A^*A\mathbf{x} = \lambda\mathbf{x}$, josta muokkaamalla saadaan $(A^*A - \lambda I)\mathbf{x} = 0$. Tällä yhtälöllä on ei-triviaaleja ratkaisuja, kun saatu matriisi on kääntövä eli sen determinantti on nolla. Saadaan

$$\begin{bmatrix} 5 - \lambda & -4 \\ -4 & 5 - \lambda \end{bmatrix} \mathbf{x} = 0 \Rightarrow (5 - \lambda)^2 - 16 = 0 \Rightarrow \lambda_1 = 9 \vee \lambda_2 = 1. \quad (2.6)$$

Ominaisarvojen avulla voidaan nyt etsiä matriisin A^*A ominaisvektorit, joiden avulla saadaan muodostettua matriisi V :

$$\lambda_1 = 9 : \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix} \mathbf{x}_1 = 0 \Rightarrow \mathbf{v}_1 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}, \quad (2.7)$$

$$\lambda_2 = 1 : \begin{bmatrix} 4 & -4 \\ -4 & 4 \end{bmatrix} \mathbf{x}_1 = 0 \Rightarrow \mathbf{v}_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}. \quad (2.8)$$

Seuraavaksi muodostetaan matriisia U varten vektorit \mathbf{u}_1 ja \mathbf{u}_2 :

$$\mathbf{u}_1 = \frac{1}{\sigma_1} A\mathbf{v}_1 = \frac{1}{3} \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ -4/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{18} \\ -1/\sqrt{18} \\ -4/\sqrt{18} \end{bmatrix}, \mathbf{u}_2 = \frac{1}{\sigma_2} A\mathbf{v}_2 = \frac{1}{1} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \quad (2.9)$$

Vektorin \mathbf{u}_3 muodostamiseksi hyödynnetään Gram-Schmidt menetelmää [4, s. 386], sillä vektorien \mathbf{u}_1 ja \mathbf{u}_2 muodostama kanta täytyy täydentää kolmannella vektorilla. Käytetään kolmannen vektorin muodostamisessa lähtökohtana yksikkövektoria \mathbf{e}_3 , sillä se ei selvästikään kuulu kahden edellisen vektorin virittämään aliavaruuteen.

Gram-Schmidt menetelmällä ei saada suoraan yksikkövektoria, vaan saatu vektori täytyy lopuksi normeerata. Näin ollen merkataan aluksi normeeramatonta vektoria \mathbf{u}'_3 , josta lopuksi normeeramalla saadaan yksikkövektori \mathbf{u}_3

$$\begin{aligned}
 \mathbf{u}'_3 &= \mathbf{e}_3 - \text{proj}_{\mathbf{u}_1}(\mathbf{e}_3) - \text{proj}_{\mathbf{u}_2}(\mathbf{e}_3) \\
 &= \mathbf{e}_3 - \frac{\mathbf{u}_1 \mathbf{e}_3}{\mathbf{u}_1 \mathbf{u}_1} - \frac{\mathbf{u}_2 \mathbf{e}_3}{\mathbf{u}_2 \mathbf{u}_2} \\
 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} - \begin{bmatrix} 1/\sqrt{18} \\ 1/\sqrt{18} \\ -4/\sqrt{18} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{18} \\ 1/\sqrt{18} \\ -4/\sqrt{18} \end{bmatrix} \\
 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 4/18 \\ -4/18 \\ -16/18 \end{bmatrix} \\
 &= \begin{bmatrix} -4/18 \\ 4/18 \\ 2/18 \end{bmatrix} \qquad \text{(Normeeraus } \mathbf{u}_3 = \frac{1}{\|\mathbf{u}'_3\|} \mathbf{u}'_3 \text{)} \\
 &= \begin{bmatrix} -2/3 \\ 2/3 \\ 1/3 \end{bmatrix}
 \end{aligned}$$

Nyt koossa on kaikki tarvittava matriisiin A lausumiseksi singulaariarvohajotelman avulla, voidaan kirjoittaa

$$A = \begin{bmatrix} 1/\sqrt{18} & 1/\sqrt{2} & -2/3 \\ -1/\sqrt{18} & 1/\sqrt{2} & 2/3 \\ -4/\sqrt{18} & 0 & 1/3 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}. \quad (2.10)$$

3. SOVELLUSKOHTEITA

Tietokoneelle tehokkain tapa muodostaa singulaariarvohajotelma on eri kuin aiemmin lasketussa esimerkissä eli laskea ensin AA^* -ominaisarvot. Jo vuonna 1970 esiteltiin tapa muodostaa singulaariarvohajotelma käyttäen hyväksi QR-hajotelmaa [1]. Johtuen pääosin siitä, että QR-hajotelman muodostaminen on tehokasta ja vakaa-ta. Kyseinen tapa muodostaa singulaariarvohajotelma oli niin hyvä, että siihen on vain lisätty tarkennuksia koskien pieniä singulaariarvoja. Muilta osin se on kuitenkin vieläkin yleisin tekniikka singulaariarvohajotelman muodostamiseksi. Singulaariarvohajotelmalle on monia sovelluskohteita, joista seuraavissa luvuissa esitellään pari niistä.

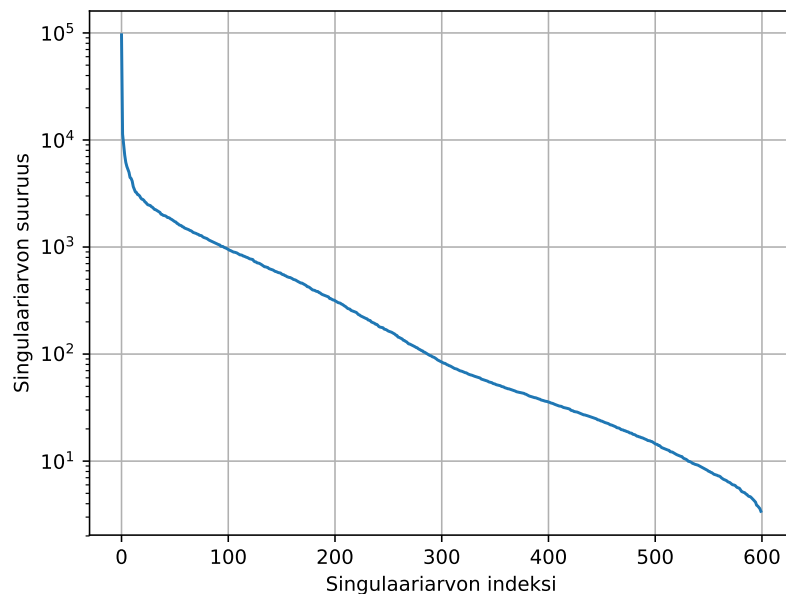
3.1 Datan pakkaaminen

Kuvan tunnistamisessa algoritmit osaavat tunnistaa kuvista asioita niin hyvin, että kuvan tarkkuutta voidaan vähentää pienentämättä merkittävästi tunnistamisen onnistumisprosenttia [5, s. 99]. Tällöin voidaan säästää muistia tallentamalla alku-peräisen kuvan sijasta approksimaatio. Liitteestä 1 löytyy Pythonilla koodattu koodiesimerkki siitä, miten kuvaa voi muokata singulaariarvohajotelman avulla. Kuvasta 3.1 pystyy näkemään, että kuvan laatu paranee aluksi nopeasti, kun kasvatetaan käytettyjen singulaariarvojen määrää.



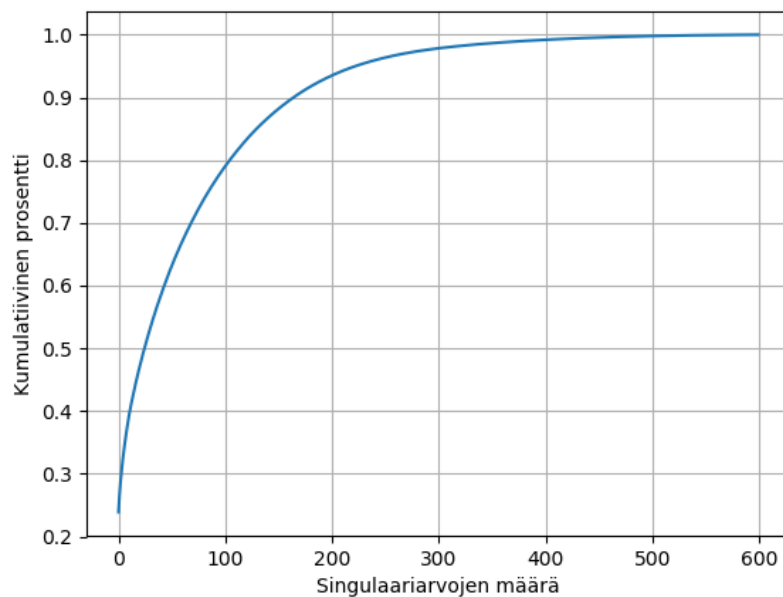
Kuva 3.1 Alkuperäinen kuva ja siitä tehdyt 5 eri approksimaatiota

Laadun parantuminen johtuu siitä, että useimmiten ensimmäiset singulaariarvot ovat selkeästi suurempia kuin viimeisimmät singulaariarvot. Tästä syystä ensimmäiset arvot muodostavat nopeasti alkuperäistä kuvaa muistuttavan kuvan ja loput arvot vain tarkentavat sitä. Esimerkkinä käytetyssä kuvassa 3.1 käy myös samoin, mikä näkyy kuvassa 3.2.



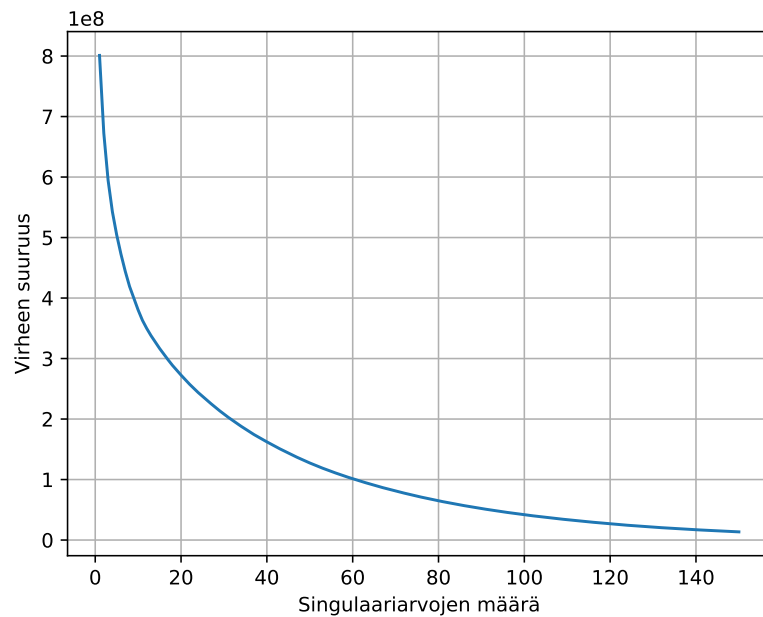
Kuva 3.2 Singulaariarvon suuruuden ja sen indeksin välinen riippuvuus

Kannattaa huomata, että y -akseli on skaalattu logaritmiseksi eli ensimmäisten ja viimeisten arvojen välillä on hyvin suuri ero. Kuvan muodostamisessa tärkeämpää yksittäisen singulaariarvon sijaan on se, kuinka paljon käytetyt singulaariarvot sisältävät tietoa eli kuinka iso käytettyjen singulaariarvojen summa on suhteessa niiden kaikkien summaan. Singulaariarvojen summaa voidaan pitää tiedon määränä, sillä singulaariarvot määräävät singulaarihajotelmassa, kuinka paljon milläkin matriisin U sarakevektorilla ja matriisin V rivivektorilla on vaikutusta lopputulokseen. Kuvassa 3.3 on havainnollistettu tätä singulaariarvojen sisältämän tiedon määrää.



Kuva 3.3 Singulaariarvojen kumulatiivinen summa jaettuna kokonaissummalla eli singulaariarvojen kumulatiivinen prosentti.

Kuten aiemmin todettiin, tiedon määrä kasvaa nopeasti singulaariarvojen määrän kasvaessa. Esimerkiksi jo 100 ensimmäistä arvoa sisältävät noin 80 % tiedosta, mutta riittääkö 100 arvoa aina vai tarvitaanko joskus enemmän. Vastaus tietenkin riippuu tapauksesta, mutta silti voidaan arvioida virheen määrää. Käytetään nyt yhtä yleisintä tekniikkaa virheen arvioimiseksi eli pienimmän neliösumman menetelmää. Otetaan alkuperäisen matriisin ja approksimaation välinen erotus ja sen jälkeen korotetaan saadun matriisin jokainen alkio toiseen potenssiin. Nyt tuloksena on matriisi, jonka jokainen alkio kertoo, kuinka paljon kyseisellä pikselillä on ero neliöllisesti verrattuna alkuperäisen matriisiin. Voidaan vielä summata kyseisen matriisin alkiot yhteet ja näin saadaan yksi luku, joka kertoo kuvan kokonaisvirheen määräästä. Kuvaan 3.4 on laskettu 150 ensimmäisellä alkiolla tehtyjen approksimaatioiden kokonaisvirheitä.



Kuva 3.4 Pienimmän neliösumman menetelmällä laskettu virhe approksimaatiolle.

Virheen määrä lähestyy kuvaajassa lähelle nollaa, vaikka singulaariarvoista on käytetty vasta neljäsosa. Siihen, kuinka montaa arvoa käytetään, kannattaa kiinnittää huomiota, sillä jokainen pois jätetty arvo vähentää myös aina yhden sarakkeen matriisista U ja yhden rivin matriisista V .

3.2 Pienimmän neliösumman menetelmä

Pienimmän neliösumman menetelmällä etsitään parasta ratkaisua yhtälölle $Ax = b$, jolle ei ole olemassa käypää ratkaisua. Pienimmän neliösumman menetelmä tarkoittaa sitä, että pyritään etsimään ratkaisu, jonka virhe neliöllisesti on mahdollisimman pieni. Matriisien tapauksessa yhtälöllä ei välttämättä ole ratkaisua, jos A ei ole neliömatriisi tai se ei ole kääntyvä, jolloin yhtälön ratkaisua $\mathbf{x} = A^{-1}\mathbf{b}$ ei voida laskea. Päädytään minimoimaan $\|Ax - b\|$, jonka ratkaisu pienimmän neliösumman menetelmällä saadaan pseudoinverssin avulla.

Määritelmä 4. Olkoon matriisi $A \in \mathbb{C}^{m \times n}$, jonka singulaariarvohajotelma on $U\Sigma V^*$. Tällöin matriisin A pseudoinverssi $A^\dagger = V\Sigma^\dagger U^*$, missä

$$\Sigma^\dagger = \text{diag}(1/\sigma_1, 1/\sigma_2, 1/\sigma_r, 0 \dots 0).$$

Pseudoinverssi voidaan myös määritellä toisella tavalla ilman singulaariarvohajotelmaa $A^\dagger = (A^*A)^{-1}A^*$. Käytetään kuitenkin tässä singulaariarvohajotelman avulla

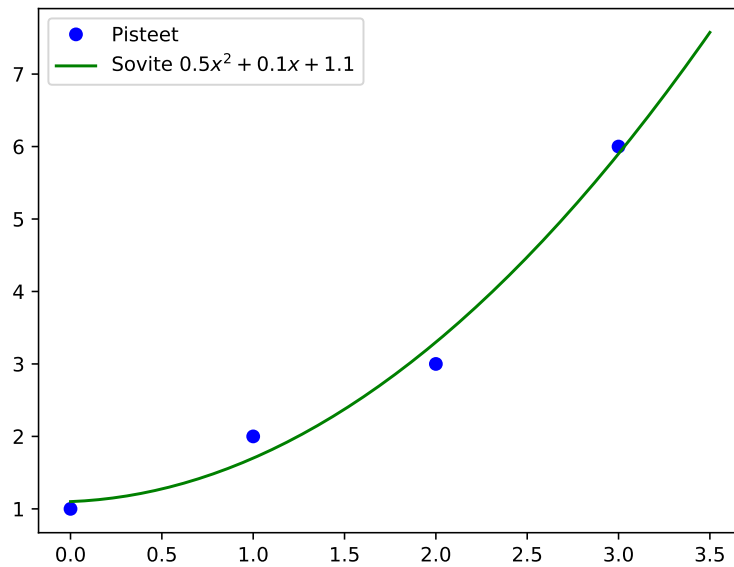
muodostettua pseudoinverssiä. Pseudoinverssin avulla voidaan nyt ratkaista alkuperäinen pienimmän neliösumman ongelma toteamalla, että yhtälölle $A\mathbf{x} = \mathbf{b}$ on olemassa yksikäsitteinen paras approksimaatio $\mathbf{x} = A^\dagger \mathbf{b}$ [4, s. 613].

Otetaan laskuesimerkiksi paraabelin sovittaminen pistejoukkoon $(0,1), (1,2), (2,3)$ ja $(3,6)$. Sijoitetaan pisteiden koordinaatit paraabelin yhtälöön ($y = ax^2 + bx + c$), jolloin saadaan alla oleva matriisi A .

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \end{bmatrix}. \quad (3.1)$$

Vektori \mathbf{b} muodostuu pistejoukon y -arvoista. Yhtälön $A\mathbf{x} = \mathbf{b}$ ratkaisemiseksi tarvitaan matriisin A singulaariarvohajotelmaa, jonka voi laskea koneella esimerkiksi matlabissa komennolla `svd(A)` tai pythonissa komennolla `numpy.linalg.svd(A)`. Molemmat funktiot palauttavat tuloksena kolme matriisiä, joiden avulla on helppo muodostaa pseudoinverssi $A^\dagger = V\Sigma U^*$. Kannattaa kuitenkin huomata pseudoinverssiä muodostaessa, että funktiot palauttavat matriisin V^* eikä matriisiä V . Ratkaisuksi saadaan nyt

$$A^\dagger \mathbf{b} = \begin{bmatrix} 0.5 \\ 0.1 \\ 1.1 \end{bmatrix}. \quad (3.2)$$



Kuva 3.5 Pisteet ja parhaiten pistejoukkoon sopiva paraabeli

Parhaiten pistejoukkoon sopii siis paraabeli $y = 0.5x^2 + 0.1x + 1.1$, jota on havainnollistettu kuvassa 3.5

3.3 Kohinainen data

Kohinaisella datalla tarkoitetaan dataa, jossa on mukana hieman virhettä, mikä esimerkiksi kuvassa tarkoittaisi sumeutta ja epätarkkuutta. Kohinaista dataa syntyy paljon eri tyyppisissä tilanteissa ja sen puhdistamiseen on kehitetty paljon algoritmeja. Alkutilanne on kuitenkin aina sama eli käsissä on kohinainen data, josta halutaan ulos alkuperäinen data. Matemaattisesti palataan edellisen kappaleen yhtälöön $Ax = b$, jossa nyt ajatellaan vektorin \mathbf{x} olevan alkuperäinen data ja matriisin A aiheuttavan virhettä dataan. Selvittäessä alkuperäistä dataa singulaariarvohajotelman avulla päädytään yleensä tekemisiin pseudoinverssin kanssa. Tällöin joudutaan tekemisiin matriisin häiriöalttiuden kanssa.

Määritelmä 5. Olkoon $A \in \mathbb{C}^{m \times n}$, missä σ_1 on matriisin A suurin ja σ_n vastaavasti sen pienin singulaariarvo. Matriisin A häiriöalttius on luku $\kappa = \sigma_1/\sigma_n$.

Mitä suurempi häiriöalttius κ on sitä alttiimpi matriisi on häiriölle. Mikäli κ on ääretön, niin matriisi on singulaarinen. Häiriötä voidaan pienentää jättämällä pois

pienimpiä singulaariarvoja, jolloin häiriöalttiuden suuruus vähenee. Tämä on loogista, sillä pseudoinverssissä keskimmaisessa matriisissa on singulaariarvojen käänteislukuja. Pienimmistä arvoista tulee siis hyvin suuria ja näin pienikin virhe arvoissa moninkertaistuu. Mitä vähemmän kohina on riippuvainen alkuperäisestä datasta, sitä paremmin pienien arvojen poisto tehoaa. Täytyy kuitenkin muistaa, että kohinan poistossa singulaariarvohajotelma on vain yksi osa ja hyvän lopputuloksen saavuttamiseksi tarvitaan muitakin algoritmeja.

4. YHTEENVETO

Työssä tutkittiin singulaariarvohajotelmaa ja käytiin läpi sen tärkeimmät ominaisuudet. Lisäksi esiteltiin myös pari sovelluskohdetta singulaariarvohajotelmalle. Yleisesti hajotelmalle on paljon sovelluskehteitä, mikä johtuu suurimmaksi osaksi hajotelman yksinkertaisuudesta ja informatiivisuudesta. Kolme matriisia sisältävät jokainen paljon hyödyllistä tietoa alkuperäisestä matriisista.

Osittain hajotelman hyödyllisyyden ansiosta singulaariarvohajotelman laskemisen algoritmeja on hiottu hyvin tehokkaiksi. Siitä huolimatta singulaariarvohajotelma on edelleen melko raskas laskea suurille matriiseille. Tämän takia osassa sovelluksissa, joissa singulaariarvohajotelmaa voisi hyödyntää, se on korvattu kevyemmällä algoritmilla. Itse asiassa esimerkiksi pienimmän neliösumman laskemiseksi on olemassa tehokkaampia tapoja kuin singulaariarvohajotelma. Singulaariarvohajotelma on kuitenkin silti yhä tärkeimpiä matriisihajotelmia ja todennäköisesti monipuolisuutensa ansiosta sille myös jatkossa löytyy lisää sovelluskohteita.

LÄHTEET

- [1] G. H. Golub ja C. Reinsch, Singular value decomposition and least squares solutions, *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
- [2] R. A. Horn ja C. R. Johnson, *Topics in Matrix Analysis*, 1991.
- [3] —, *Matrix Analysis 2nd Edition*, 2012.
- [4] D. Poole, *Linear Algebra : a Modern Introduction*, 2nd ed., 2006.
- [5] O. Vergara, R. Pinto-Elías, ja V. Sánchez, Singular value decomposition image compression system for automatic object recognition, vol. 5, no. 12, pp. 95–100, 2006.

LIITE 1: KOODI KUVAN PAKKAAMISEKSI SINGULAARIARVOHAJOTELMAN AVULLA

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
4
5 def rgb2gray(rgb):
6     return np.dot(rgb[...,:3],[0.21,0.72,0.07])
7
8 def main():
9     image = mpimg.imread("dog.jpg")
10    # Muutetaan kuva mustavalkoiseksi kaksiulotteiseksi matriisiksi
11    gray = rgb2gray(image)
12    #Kaksiulotteiselle matriisille voidaan muodostaa
13    #singulaariarvohajotelma. Huomautus: Numpy ei
14    #anna singulaariarvoja diagonaalimatriisissa vaan
15    #vektorina (1D array), jonka alkioina ovat singulaariarvot
16    U,S,V = np.linalg.svd(gray, full_matrices=True)
17
18    #Lasketaan virhe pienimmän neliösumman menetelmällä
19    mse = np.zeros(150)
20    values = np.arange(1,151)
21    for i in range(0,150):
22        uu = U[:, :values[i]]
23        ss = np.diag(S[:values[i]])
24        vv = V[:values[i], :]
25        approx = uu.dot(ss).dot(vv)
26        mse[i] = np.sum(np.sum(np.square((gray - approx))))
27
28    #Tehdään approksimaatioita alkuperäisestä kuvasta
29    ks = [10, 20, 50, 100, 200]
30    data = []
31    for k in ks:
32        u = U[:, :k]
33        s = np.diag(S[:k])
34        v = V[:k, :]
35        approx = u.dot(s).dot(v)
36        data.append((k, u, s, v, approx))
37
38    #Näytetään alkuperäinen kuva.
39    fig = plt.figure()
40    plt.suptitle("Alkuperäinen kuva ja approksimaatiot")
41    fig.add_subplot(2, 3, 1)
42    plt.imshow(gray, cmap = 'gray')
```

```
43 plt.title("Alkuperäinen")
44
45 # Sen jälkeen näytetään approksimaatiot
46 for j, (k, u, s, v, approx) in enumerate(data):
47     fig.add_subplot(2, 3, 2 + j)
48     plt.imshow(approx, cmap='gray')
49     ur, uc = np.shape(u)
50     vr, vc = np.shape(v)
51     information = ur * uc + np.shape(s)[0] + vr * vc
52     title = "{:d} singular values. Info used: {:f}".format(k,
information / (height * weight))
53     plt.title(title)
54 #Ensimmäinen kuvaaja näyttää singulaariarvot käyränä
55 plt.figure()
56 xx = S
57 plt.plot(xx)
58 plt.xlabel("Singulaariarvon indeksi")
59 plt.ylabel("Suuruus")
60 plt.yscale('log')
61 plt.grid()
62 plt.title("Singulaariarvot")
63
64 #Toinen kuvaaja näyttää kuinka paljon n ensimmäistä
65 #singulaariarvoa sisältää prosentteina verrattuna kaikkien
singulaariarvojen summaan
66 sum_of_singular_values = np.sum(S)
67 plt.figure()
68 plt.plot(np.cumsum(xx)/sum_of_singular_values)
69 plt.xlabel("Singulaariarvojen määrä")
70 plt.ylabel("Kumulatiivinen prosentti")
71 plt.grid()
72 plt.title("Singulaariarvojen kumulatiivinen prosentti")
73
74 #Viimeinen kuvaaja havainnollistaa virheen määrää eri
approksimaatioissa.
75 plt.figure()
76 plt.plot(values, mse)
77 plt.xlabel("Singulaariarvojen määrä")
78 plt.ylabel("Virheen suuruus")
79 plt.grid()
80 plt.title("Virhe approksimaatioiden ja alkuperäisen kuvan välillä")
81
82 plt.show()
83
84 main()
```