

1 / 40

- 3 / 40

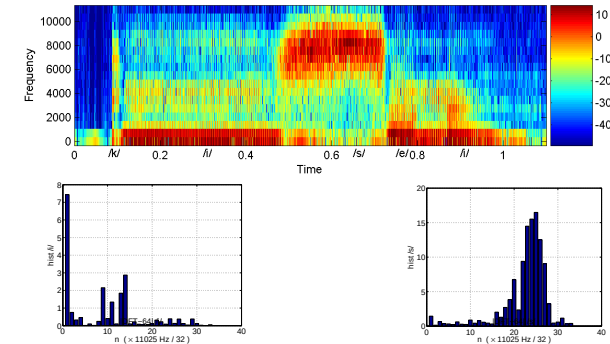
- 2 / 40

4 / 40

Datamatriisi (3)

- Spektrogrammissa siis matriisi $\mathbf{X} = [\mathbf{x}(1) \ \mathbf{x}(2) \ \dots \ \mathbf{x}(n)]$, jossa $\mathbf{x}(i)$:t ovat peräkkäisissä aikaikkunoissa (esim. 20 ms välein) laskettuja diskreettejä Fourier-muunnoksia (katso kalvo 17)
- Toinen esimerkki spektrogrammista: ihmisen puhe (seuraava kuva)

Datamatriisi (4)



Kuva: Ylhäällä spektrogrammi sanasta "kiisseli". Sanan Fourier-muunnoksen eli spektrin histogrammi kahdessa kohdassa sanaa: vasemmalla /i/, oikealla /s/. Vastaavan ajan mukana "tanssivan" patsasrivin näet "missä tahansa" audiosoittimessa.

Datamatriisi (5)

- Kaikkea dataa ei suinkaan voi esittää vektorina mutta tämä kattaa hyvin paljon tapauksia. Vrt. myös relaatiotietokannat joissa tiedot on talletettu binäärivektoreina
- Merkitään seuraavassa d -ulotteista vektoria merkinnällä $\mathbf{x}$ ja sen alkioita $x_1, x_2, \dots, x_d$.
- Oletan että vektorien yhteenlasku $\mathbf{x} + \mathbf{y}$, sisätulo $\mathbf{x}^T \mathbf{y}$ ja normi $\|\mathbf{x}\|$ ovat tunnettuja, samoin matriisilla kertominen $\mathbf{Ax}$, matriisien yhteenlasku $\mathbf{A} + \mathbf{B}$ ja tulo $\mathbf{AB}$ sekä matriisin ominaisarvot ja -vektorit $\mathbf{Ax} = \lambda \mathbf{x}$.

Datan piirreirrotus ja vektorointi

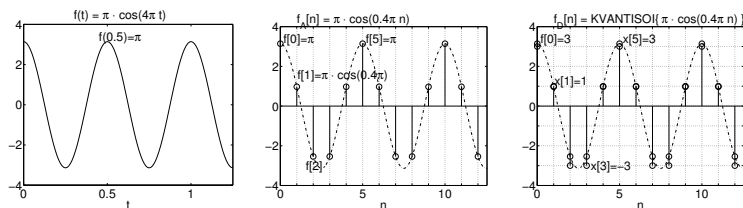
Analogisen datan digitointi

- Usein data on jo vektorimuodossa: siis järjestetty joukko lukuja (jatkuva-arvoisia, binäärisiä ...).
- Jotta jatkuva (analoginen) data saataisiin tähän muotoon, se täytyy diskretoida (digitoida).
- Analoginen *signaali*, esim. ääni, kuvataan ajan mukana vaihtuvana jatkuvana funktiona $f(t)$

Datan piirreirrotus ja vektorointi (2)

Analogisen datan digitointi

- Sekä aika t että amplitudi f täytyy diskretoida jotta ne voi esittää digitaalisessa muodossa

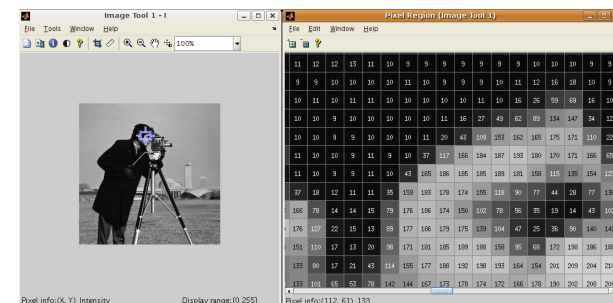


Kuva: (a) analoginen signaali $f(t)$; (b) ajan suhteen digitoitu lukujono $f_A[n] \in \mathbb{R}$, $n \in \mathbb{Z}$; (c) ajan ja amplitudin suhteen digitoitu $f_D[n] \in \mathbb{Z}$. Äänisignaaliissa tyypillinen digitointi 8 bitillä antaa $2^8 = 256$ tasoa y-akselille välillä $-1 \dots +1$.

Datan piirreirrotus ja vektorointi (3)

Analogisen datan digitointi

- Esim. digikamera, tietokoneen äänikorttiin kytketty mikrofoni



Kuva: Matlabin "kameramies" ja yksityiskohta silmän alueelta. Resoluutio 256×256 , tyyppi `uint8` eli $[0, 255] \in \mathbb{Z}$, jossa 0=musta ja 255=valkea.

Datan piirreirrotus ja vektorointi (4)

Analogisen datan digitointi

- Käytännössä mittalaite tekee sen analogia-digitaali (A/D) -muuntimensa avulla
- Tähän liittyvä teoreettinen kysymys on *näytteistys* (*sampling*): kuinka tiheään näytteitä tulee ottaa, ettei hävitetä tietoa?
- Yksiulotteinen signaali $f(t)$ muutetaan jonoksi $f_m = f(mT)$ missä T on näytteistysväli
- Jos näytteistystaajuus $1/T$ on riittävän suuri verrattuna $f(t)$:n suurimpaan taajuuteen, voidaan itse asiassa koko $f(t)$ palauttaa näytteistään (näytteistyslause, sampling theorem; ks. kurssi T-61.3015 Digitaalinen signaalinkäsittely ja suodatus)
- Demo näyteenotosta <http://www.jhu.edu/~signals/sampling/>

Datan piirreirrotus ja vektorointi

Taajuusspektri

- Mitä tarkoittaa $f(t)$:n taajuus?
- Funktiolla $f(t)$ on *taajuusspektri* joka määritellään *Fourier-muunnoksen* avulla

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

missä ω on (kulma)taajuus ja $i = \sqrt{-1}$

- Tämän kompleksifunktion itseisarvo voidaan piirtää käyränä, ja se on ns. amplitudispektri (magnitudispektri)
- Perusteluja: taajuus on jaksollisen funktion ominaisuus (montako kertaa sekunnissa jakso toistuu), ja sinikäyrä $f(t) = \sin(\alpha t + \theta)$ on kaikkein tyypillisin jaksollinen funktio. α on sen (kulma)taajuus, θ on vaihe

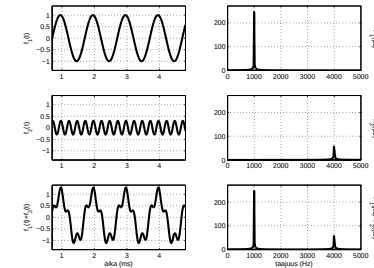
Datan piirreirrotus ja vektorointi (2)

Taajuusspektri

- Sinikäyrän Fourier-muunnos on nolla paitsi kulmataajuuden arvoilla $\omega = \pm\alpha$ (miksi?), joten F-muunnos paljastaa sinin taajuuden.
- Jos $f(t)$ on summa monesta sinikäyrästä, niiden kaikkien taajuudet näkyvät piikkeinä *taajuusalueessa* eli kompleksitasossa.

Datan piirreirrotus ja vektorointi (3)

Taajuusspektri



Kuva: Kaksi sinisignaalia ja niiden summa aika- ja taajuustasossa.

Datan piirreirrotus ja vektorointi (4)

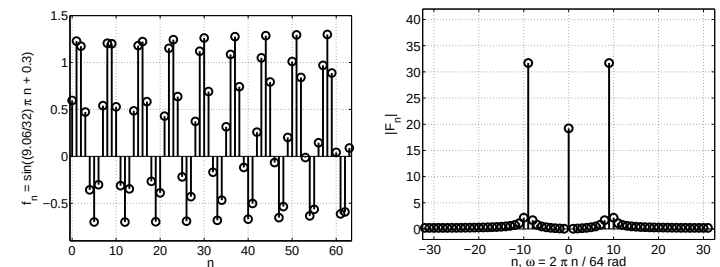
Taajuusspektri

- Diskreetti Fourier-muunnos (DFT) määritellään digitoidulle signaalille f_m , $m = 0, \dots, T - 1$:

$$F_n = \sum_{m=0}^{T-1} f_m e^{-2\pi i m n / T}$$

Datan piirreirrotus ja vektorointi (5)

Taajuusspektri



Kuva: 64-pituinen lukujono f_m , jossa keskiarvo ja yksi sini. Oikealla sen DFT-64 F_n , jossa korkeimmat piikit $n = 0$ ja $n = \pm 9$.

- Demo Fourier-sarjalla approksimoinnista

► <http://www.jhu.edu/~signals/fourier2/>

Datan piirreirrotus ja vektorointi

Taajuussuodatus

- Usein aikasignaalista halutaan poistaa tiettyjä taajuuksia, esim. korkeita taajuuksia jotka vastaavat kohinaa tai verkkovirran taajuus 50 Hz.
- Tässä käytetään hyväksi lineaarisia suotimia, jotka laskennallisesti toteutetaan ns. *konvoluutiona*:

$$g_k = \sum_{m=-\infty}^{\infty} f_m s_{k-m}$$

missä f_m on suodatettava digitaalinen signaali ja s_j on vastaavasti suodin (äärellinen lukujono).



17 / 40

Datan piirreirrotus ja vektorointi (2)

Taajuussuodatus

- On helppo todistaa että konvoluution Fourier-muunnokselle pätee

$$G_n = F_n \cdot S_n$$

eli taajuustasossa spektrit kerrotaan keskenään

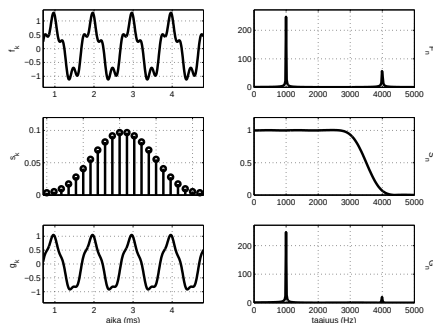
- Jos siis halutaan estää / korostaa tiettyjä taajuuksia, valitaan suodin s_k siten että sen F-muunnoksessa S_n nämä taajuudet ovat heikkoja / vahvoja.
- Esimerkki suodattimesta aika- ja taajuustasossa. Tässä tehdään alipäästösuodatus (keskiarvoistaminen) aikatasossa $g_k = \sum_m f_m s_{k-m}$ vasemmassa sarakkeessa tai vastaavasti taajuustasossa $G_n = F_n \cdot S_n$. Vertaa ekvalisaattorin toimintaan kotistereoissa tai musiikkisioittimissa.



18 / 40

Datan piirreirrotus ja vektorointi (3)

Taajuussuodatus



Kuva: Signaalin suodatus. Vasemmalla aikatason digitaaliset lukujonot (signaalit): sisääntulo f_k , suodin s_k ja suodatuksen lopputulos g_k . Oikealla vastaavat Fourier-muunnetyt spektrit $f_k \leftrightarrow F_n$. Korkeataajuisen komponentti vaimenee suodatuksessa.



19 / 40

Datan piirreirrotus ja vektorointi (4)

Taajuussuodatus

- Suodattimen suunnittelu käytännössä on oma taiteenlajinsa, ks. T-61.3015 Digitaalisen signaalinkäsittelyn ja suodatuksen kurssi.
- Joko aika-alueen digitaalinen signaali f_m tai taajuusalueen digitaalinen signaali $|F_n|$ (amplitudispektri, mahdollisesti täydennettynä vaiheella) voivat sellaisenaan olla vektoreita joita jatkossa analysoidaan. Esimerkkinä puheen spektrogrammi.
- Kuville on olemassa 2-ulotteinen DFT jota voi vastaavalla tavalla käyttää kuvien suodatukseen; vrt. digikameroiden kuvankäsittelyoptiot.



20 / 40

Datan piirreirrotus ja vektorointi (5)

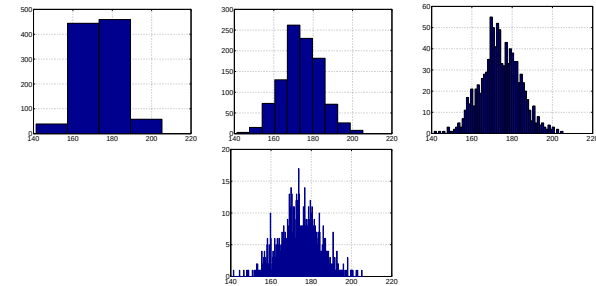
Taajuussuodatus

- Spektriä voi myös kokonaisuudessaan tai osina käyttää tehokkaana piirrevektorina kuville. Se paljastaa esim. kuvassa olevien viivojen yleisimmät suunnat. Enemmän kurssissa T-61.5100 Digitaalinen kuvankäsittely.

Muita tyypillisiä piirteitä eri datatyypeille

Histogrammi

- Hyvin yleinen ja tehokas piirretyyppi on *histogrammi*, jossa suureen arvot "lokeroidaan"



Kuva: Normaalijakautuneita satunnaislukuja $N = 1000$ kpl. Sama data, mutta pylväiden lukumäärä 4, 10, 60 ja 300.

Muita tyypillisiä piirteitä eri datatyypeille (2)

Histogrammi

Histogrammissa pylvään korkeus kertoo, kuinka monta arvoa osui lokeroon.

- DFT on esimerkki tästä eli taajuushistogrammi: paljonko signaaliissa tai kuvassa on tiettyä taajuutta
- Kuvan värihistogrammi kertoo, montako pikseliä kuvassa on kutakin väriä (dimensio 16.7 miljoonaa tiivistämättömänä, kun kolmeväriäinen 8-bittinen esitys)
- Tekstidokumentin sanahistogrammi kertoo montako kertaa kukin sana esiintyy dokumentissa (dimensio 50.000 - 100.000 tiivistämättömänä)

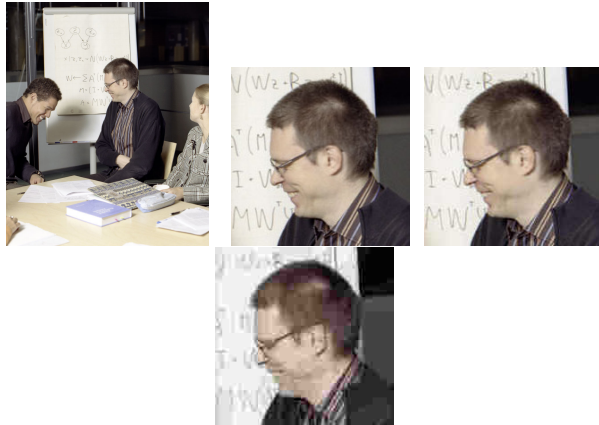
Muita tyypillisiä piirteitä eri datatyypeille

Kuva ja ääni

- Hyvä piirteiden lähde ovat standardit, etenkin JPEG (joint photographic experts group) ja MPEG (moving pictures expert group); esim. MPEG-1 Audio Layer 3 eli MP3 äänelle ja musiikille

Muita tyypillisiä piirteitä eri datatyypeille (2)

Kuva ja ääni



Muita tyypillisiä piirteitä eri datatyypeille (3)

Kuva ja ääni

Kuva: Alkuperäinen kuva, josta kolme eri JPEG-tallennusta. Yksityiskohtakuvien tavukoot 42 kT, 12 kT, 6 kT.

- Ne on kehitetty äänen, kuvien ja videon pakkaamiseen, mutta pakattua muotoa voi käyttää myös piirteinä automaattisessa analyysissä

Muita tyypillisiä piirteitä eri datatyypeille

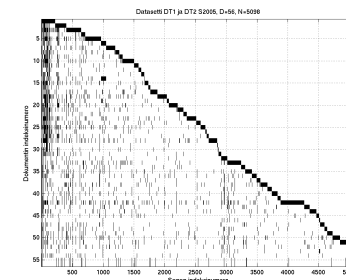
Teksti vektorina

- Edellä mainittiin jo sanahistogrammit
- Jos yksittäisten dokumenttien sanahistogrammit ovat datamatriisin **X** sarakkeina, sitä kutsutaan termi-dokumentti-matriisiksi.
- Kuvan esimerkissä termi-dokumenttimatriisi, jossa kukin rivi vastaa yhtä dokumenttia ja kukin sarake vastaa yhtä sanaa. Sanojen lukumäärät ≥ 1 on korvattu mustalla pisteellä. Sarakkeet on järjestetty sanojen esiintymisjärjestykseen. Ensimmäisessä dokumentissa yli 200 sanaa, toisessa myös noin 200 sanaa, joista noin 150 uusia verrattuna ensimmäiseen, jne. Lähde: 56 harjoitustyötä DT-kurssilla 2005. HUOM! Dataan on

Muita tyypillisiä piirteitä eri datatyypeille (2)

Teksti vektorina

tahallisesti lisätty yksi “lähes” kopiodokumentti. Mikä dokumentti?



Kuva: Termi-dokumentti-matriisi, jossa kukin rivi ($D = 56$) vastaa yhtä dokumenttia ja kukin sarake ($N = 5098$) vastaa yhtä sanaa.

Muita tyypillisiä piirteitä eri datatyypeille (3)

Teksti vektorina

- Käytännössä matriisia useimmiten muokataan seuraavasti:
 - Tavallisimmat sanat (ja, siis, ...) poistetaan; 2. sanoja (termejä) painotetaan niiden tärkeyden mukaan, usein käyttäen ns. *käänteistä dokumenttitaajuutta*

$$w(sana) = \log\left(\frac{L}{df(sana)}\right)$$

missä $w(sana)$ on painokerroin, $df(sana)$ on sanan *dokumenttifrekvenssi* (kuinka monessa kokoelman dokumentissa tuo sana esiintyy), ja L on kokelman dokumenttien lukumäärä

- Nyt kunkin dokumentin sanahistogrammissa arvot kerrotaan painoilla w .



29 / 40

Muita tyypillisiä piirteitä eri datatyypeille (4)

Teksti vektorina

- Siten esim. sana "ja" joka luultavasti esiintyy jokaisessa dokumentissa (suomenkielisessä kokoelmassa) saa histogrammiarvon 0.
- Hankalampi tapaus ovat synonyymit ja sanojen taivutusmuodot. Näitä pohditaan kurssissa T-61.5020 Statistical Natural Language Processing (Luonnollisen kielen tilastollinen mallinnus)
- Samaa vektorointimuotoa voi käyttää mille tahansa *merkkijonoille*; esim. geneettisessä koodissa on 4 kirjainta A,C,G,T jotka ovat lyhenteitä tietyille nukleotideille. "Sana" voi olla esim. 3 merkin pituinen osajono: AAA, AAC, TTT joita on vain $4^3 = 64$ erilaista ja siten "termi-dokumenttimatriisin" sarakkeet 64-dimensioisia.



30 / 40

Muita tyypillisiä piirteitä eri datatyypeille (5)

Teksti vektorina

- Samantapainen idea yleistyy *puille* ja niitäkin yleisemmille tietorakenteille.



31 / 40

Dimensionaalisuuden kirous

Otoksen koko suhteessa avaruuden dimensioon

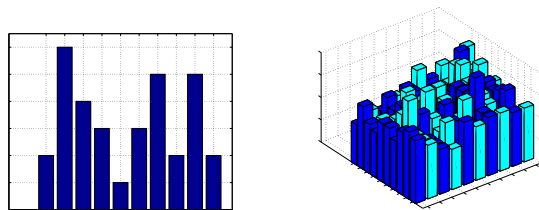
- Ajatellaan histogrammia moniulotteisessa avaruudessa. Katsotaan oheista 1-ulotteista histogrammia, jossa 10 lokeroa. Jos halutaan tasossa (2-ulotteinen avaruus) 10 lokeroa / dimensio, tulee jo $10 \times 10 = 100$ lokeroa (pienä suorakaidetta). Riittääkö data täyttämään ne?



32 / 40

Dimensioaalisuuden kirous (2)

Otoksen koko suhteessa avaruuden dimensioon



Kuva: Datahistogrammi (a) 1D; (b) 2D

- d -ulotteisessa avaruudessa 10^d ; esim. jos $d = 79$, niin 10^{79} , joka on maailmankaikkeuden atomien arvioitu lukumäärä. Näin jo 79-ulotteisessa vektoriavaruudessa, jossa on 10 lokeroa kullakin koordinaattiakselilla, tulisi vain 1 atomi kuhunkin lokeroon



33 / 40

Dimensioaalisuuden kirous (3)

Otoksen koko suhteessa avaruuden dimensioon

- Mikä tahansa vektorijoukko korkeaulotteisessa avaruudessa on tavattoman harva
- Tämä on *dimensioaalisuuden kirous*: tarvitaan valtavasti dataa minkäänlaisen mallin muodostamiseksi korkeaulotteiselle datalle



34 / 40

Dimensioaalisuuden kirous

Korkeaulotteisten vektoriavaruuksien omituisuuksia

- Ajatellaan neliötä, kuutiota, hyperkuutiota jonka sivun pituus on 1 ja dimensio d . Ajatellaan datajoukkoa (datamatriisin X sarakkeet) joka on jakautunut tasaisesti hyperkuutioon. (Kussakin kahdessa osajoukossa joilla on sama tilavuus on keskimäärin sama määrä vektoreita). Kun d on suuri, törmätään joihinkin yllättäviin ilmiöihin.

- 1. Melkein koko hyperkuutio tarvitaan kattamaan pienikin osuus vektoreista.

Mikä on sellaisen pienemmän hyperkuution sivun pituus, joka sisältää osuuden p kaikista vektoreista? (Esim. kymmenesosan, $p = 0.1$). Merkitään sivun pituutta $s(p)$

Neliö: $s(0.1) = \sqrt{0.1} \approx 0.32$, $s(p) = \sqrt{p}$

Kuutio: $s(0.1) = (0.1)^{1/3} \approx 0.46$, $s(p) = p^{1/3}$



35 / 40

Dimensioaalisuuden kirous (2)

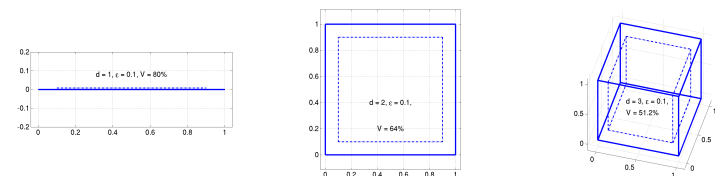
Korkeaulotteisten vektoriavaruuksien omituisuuksia

10-ulotteinen: $s(0.1) = (0.1)^{1/10} \approx 0.80$

d -ulotteinen hyperkuutio: $s(p) = p^{1/d}$

Mille tahansa osuudelle p pätee $s(p) \rightarrow 1$ kun $d \rightarrow \infty$.

Lähes koko hyperkuutio tarvitaan kattamaan pienikin osuus datasta.



Kuva: Sisäpisteiden suhteellinen määrä pienenee dimension kasvaessa.



36 / 40

Korkeaulotteisten vektoriavaruuksien omituisuuksia

- Voidaan osoittaa että pisteiden keskimääräinen etäisyys on

$$D(d, n) = \frac{1}{2} \left(\frac{1}{n} \right)^{1/d}.$$

Nähdään että olipa n mikä tahansa, $D(d, n) \rightarrow 0.5$ kun $d \rightarrow \infty$.

Korkeaulotteisten vektoriavaruuksien omituisuuksia

- Jos tehdään tasavälinen histogrammi, niin melkein kaikki lokerot ovat hyperkuution pinnalla. Olkoon lokeroiden lukumäärä dimensioita kohti b , kaikkiaan silloin lokeroita b^d . Niistä osuus $(\frac{b-2}{b})^d$ on kuution sisäosassa. Mutta tämä menee nollaan kun $d \rightarrow \infty$.
- Kaikki yhdessä osoittavat että korkeaulotteinen hyperkuutio ei vastaa käsitystämme kuutiosta, vaan on pikemminkin “piikikäs”.
- Opetus: data-analyysissä olisi tärkeää saada dimensio jotenkin pudotettua mahdollisimman pieneksi, että vältetään yllä olevat ilmiöt.

Esimerkki piirreirrotuksesta: PicSOM

► <http://www.cis.hut.fi/picsom/>

Yhteenveto

- Tässä luvussa esitettiin datamatriisi X yhtenä tapana esittää havaintoja.
- Datamatriisista voidaan irrottaa piirteitä, jotka kuvaavat tehokkaammin dataa. Piirteitä voidaan käyttää paremmin esimerkiksi luokittelussa tai ryhmittelyssä.
- Laskenta tulee monella tapaa ongelmalliseksi, jos mitattavien suureiden määrä eli datamatriisin dimensio kasvaa suureksi. Seuraavassa luvussa esitellään yksi tapa pudottaa tehokkaasti datan dimensiota.