Klusterointi korkeaulotteisissa avaruuksissa

Jukka Kohonen
Jukka.Kohonen@cs.helsinki.fi

Klusterointimenetelmät-seminaari
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C–2002–34, s. 21–31, marraskuu 2002

Tiivistelmä

Monissa data-analyysitehtävissä joudutaan nykyisin tekemisiin korkeaulotteisen datan kanssa. Tällöin tavanomaiset klusterointimenetelmät eivät välttämättä johda hyviin tuloksiin tai ole edes laskennallisesti mahdollisia. Ongelmat liittyvät klusterointiin käytettävän etäisyysmitan jakaumaominaisuuksiin.

Tässä esitelmässä kuvaillaan, millaisiin hankaluuksiin datan korkea dimensio johtaa ja miksi, sekä joitakin menetelmiä näistä hankaluuksista selviämiseeen. Erään ratkaisuyrityksen tarjoavat perinteiset dimensionvähennysmenetelmät kuten pääkomponenttianalyysi, mutta luonteeltaan globaaleina ne eivät ratkaise ongelmaa täysin.

Aliavaruusklusterointi puolestaan on nimitys menetelmille, joissa data projisoidaan klusterointia varten useisiin sopiviin aliavaruuksiin, jopa kutakin klusteria varten erikseen. Perusoletus on, että datalla on mielekästä vaihtelua vain muutamissa avaruuden lukuisista suunnista ja että projisointi näiden suuntien muodostamaan mataladimensioiseen aliavaruuteen ei siis hukkaa mitään tärkeää. Perusongelmana onkin löytää tällaisia mielekkäitä suuntia.

1 Johdanto

Monilla klusteroinnin sovellusaloilla joudutaan nykyisin tekemisiin hyvin korkeaulotteisessa avaruudessa esitetyn — datan kanssa. Tyypillisiä esimerkkejä korkeaulotteisina vektoreina esitetystä datasta ovat: rasterikuvat[1], aikasarjat, genomit, tekstidokumentit[2], ja yleensä reaalimaailman olioiden (kuten asiakkaiden) lukuisia yksittäisiä ominaisuuksia luettelevat vektorit[3]. Näissä data on helposti yli 20-ulotteista, eikä yli 1000-ulotteinenkaan data ole tavatonta.

Data voidaan projisoida korkeaulotteisesta avaruudesta (esim. $\mathbb{R}^{100}$) erilaisiin *aliavaruuksiin* siinä tarkoituksessa, että näin saadusta alempiulotteisesta datasta löytyy helpommin ja parempia klustereita. Tässä tarvittavia lineaarialgebran käsitteitä palautetaan mieliin luvussa 2 ja projisoinnin vaikutusta klusterointiin tarkastellaan luvussa 3.

Luvussa 4 perehdytään syvemmin siihen, miten klusteroinnin kannalta tärkeät *etäisyysmitat* käyttäytyvät korkeaulotteisissa avaruuksissa.

[1]Rasterikuva, jossa on $m \times n$ pikseliä, voidaan tulkita mn-ulotteiseksi vektoriksi, jossa kunkin pikselin tummuusarvo vastaa yhtä ulottuvuutta. Kolmivärikuvalla vektorista tulee peräti $3mn$-ulotteinen.

[2]Tekstidokumentteja kuvataan usein sanojen frekvensseistä koostuvilla vektoreilla.

[3]Tällöin kukin ominaisuus eli attribuutti muodostaa oman ulottuvuutensa.

Jos projisointi on täysin irrallinen esikäsittelyvaihe, kyseessä on *dimensionvähennys* (dimension[ality] reduction). Tällaisista menetelmistä tunnetuimpia ovat pääkomponenttianalyysi (PCA), singulaariarvohajotelmat (SVD) ja aallokkeet (wavelets). Pääkomponenttianalyysiä esitellään luvussa 5.

Aliavaruusklusteroinnissa sopivan aliavaruuden etsintä, siihen projisointi ja siellä klusterointi ovat tiiviissä vuorovaikutuksessa. Pyritään valitsemaan nimenomaan sellaisia aliavaruuksia, jotka ovat klusteroinnin kannalta hyviä. Valintaa tehdään klusteroinnin aikana jopa klusterikohtaisesti: eri aliavaruuksissa löydetään eri klustereita. Aliavaruusklusterointia tarkastellaan luvussa 6. Tässä esitelmässä ei kuitenkaan paneuduta näiden menetelmien yksityiskohtiin lähemmin.

2 Hiukan lineaarialgebraa

Tarkastellaan konkreettisuuden vuoksi *reaali*kertoimista[4] D-ulotteista vektoriavaruutta $\mathbb{R}^D$. Avaruuden pisteitä ovat siis D-pituiset lukujonot eli vektorit $\mathbf{x} = (x_1, x_2, \ldots, x_D)$. Avaruudessa on määritelty vektorien yhteenlasku $\mathbf{x} + \mathbf{y}$ ja skalaarilla kertominen $t\mathbf{x}$. Lisäksi oletamme, että käytössä on jokin etäisyysmitta $dist(\mathbf{x}, \mathbf{y})$. Eräs tyypillinen etäisyysmitta on euklidinen metriikka L_2, mutta on syytä huomata, että käsiteltävälle datalle saattaa jokin muu metriikka olla mielekkäämpi (vrt. luku 4).

Aliavaruus on tason käsitteen yleistys. Formaalisti voidaan määritellä, että aliavaruus on sellainen avaruuden osajoukko, joka itsekin on vektoriavaruus. Käytännössä aliavaruus määritellään yleensä ns. kantavektorien $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_d}$ avulla: niiden virittämään (origon kautta kulkevaan) aliavaruuteen S kuuluvat kaikki pisteet, jotka ovat muotoa $\mathbf{x} = \sum_{i=1}^{d} t_i \mathbf{e_i}$, missä t_i ovat reaalilukuja. Kantavektorien lukumäärä d on samalla aliavaruuden dimensio. Jos $d = D - 1$, aliavaruutta kutsutaan hypertasoksi.

Puhutaan myös aliavaruuksista, jotka eivät kulje origon kautta. Pisteen $\mathbf{x_0}$ kautta kulkeva, vektorien $\mathbf{e_i}$ virittämä aliavaruus koostuu pisteistä, jotka ovat muotoa $\mathbf{x} = \mathbf{x_0} + \sum_{i=1}^{d} t_i \mathbf{e_i}$.

Pisteen $\mathbf{x}$ *projektio* aliavaruuteen S voidaan määritellä pisteeksi $\mathbf{x}' \in S$, joka minimoi etäisyysmitan $dist(\mathbf{x}, \mathbf{x}')$. Jos $dist = L_2$, projektion voi helposti laskea sisätulon avulla.

Kutsumme aliavaruutta

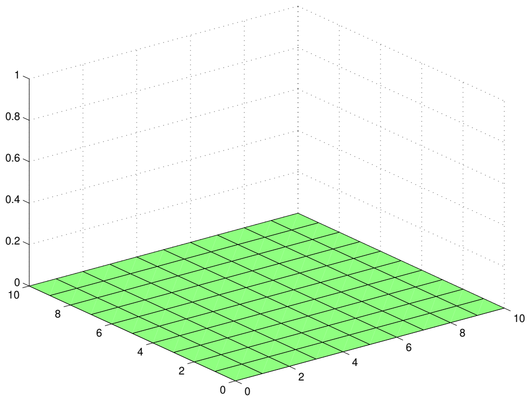
- *akselinsuuntaiseksi*, jos sillä on koordinaattiakselien suuntaiset kantavektorit, ja
- *vapaasuuntaiseksi*, jos näin ei ole.

Tämä on merkittävä erottelu, sillä kuten luvussa 6 tullaan huomaamaan, jotkin menetelmät etsivät ainoastaan akselinsuuntaisia aliavaruuksia. Silloin ollaan itse asiassa tekemässä *attribuuttien valintaa*: kun kutakin datan attribuuttia vastaa yksi avaruuden ulottuvuus, niin kutakin d:n attribuutin joukkoa vastaa d-ulotteinen akselinsuuntainen aliavaruus.

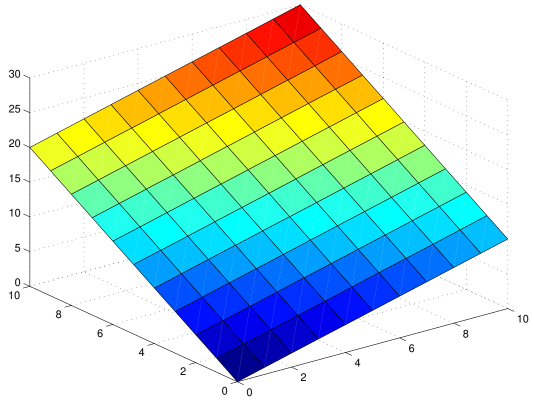
Lisäksi puhutaan yleisemmästä käsitteestä *alimonisto*. Alimonisto on pinnan käsitteen yleistys: toisin kuin aliavaruus, alimonisto voi olla kaareva. *Lokaalisti* eli mielivaltaisen pisteen pienessä ympäristössä alimonisto kuitenkin muistuttaa aliavaruutta.

Erilaisia alimonistoja on havainnollistettu kuvassa 1.

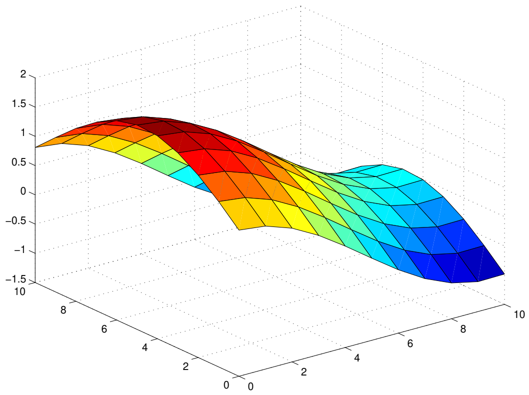
[4] Reaaliavaruuden sijasta voitaisiin tarkastella muitakin vektoriavaruuksia, esimerkiksi D-pituisten *bitti*vektorien muodostamaa avaruutta $\{0, 1\}^D$. Esitetyt peruskäsitteet sinänsä yleistyvät niihin helposti, mutta eivät ole välttämättä laskennallisesti yhtä mielekkäitä.



(a) Akselinsuuntainen aliavaruus



(b) Vapaasuuntainen aliavaruus



(c) Yleinen alimonisto

Kuva 1: Erilaisia alimonistoja ($D = 3$, $d = 2$)

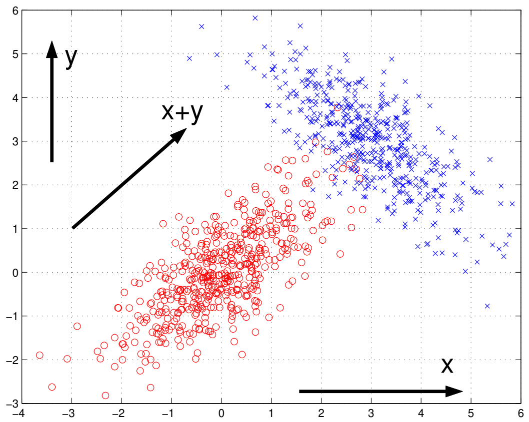
3 Klusterointi aliavaruudessa

Klusterointi perustuu pisteiden välisiin etäisyyksiin. Mitä tapahtuu, jos pisteet on ensin projisoitu johonkin aliavaruuteen?

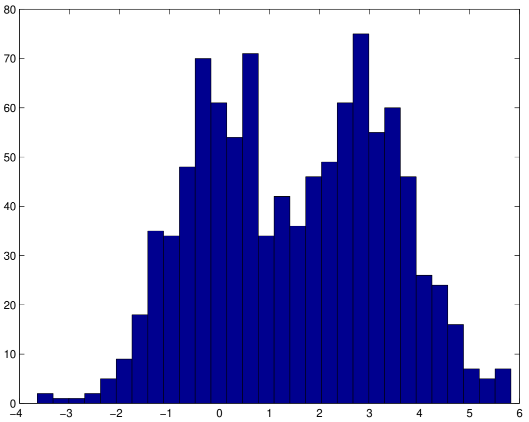
Tämä riippuu olennaisesti siitä, mihin aliavaruuteen projisoidaan ja miten etäisyydet muuttuvat tässä projektiossa. Ihanteellista olisi, jos klustereiden sisäiset etäisyydet vähenisivät mahdollisimman paljon, mutta klustereiden väliset etäisyydet eivät lainkaan.

Kuva 2 havainnollistaa projektion vaikutusta. Alkuperäinen data on 2-ulotteista ja koostuu kahdesta normaalijakautuneesta klusterista, jotka ovat eri suuntiin pitkulaisia. Akselinsuuntaisia 1-ulotteisia aliavaruuksia on kaksi, x-akseli ja y-akseli. Kumpikaan ei säilytä klusterien välistä etäisyyttä riittävästi, ja klusterit sekoittuvat toisiinsa. Eräs vapaasuuntainen aliavaruus, "$(x + y)$-akseli", sen sijaan tuottaa selkeästi kaksihuippuisen jakauman.

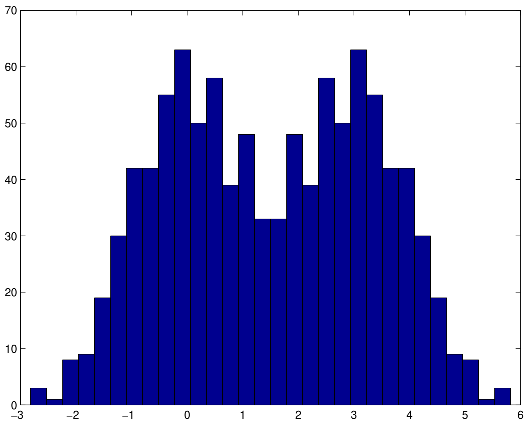
Datan korkea dimensio saattaa olla näennäistä, "vain" esitysmuodosta johtuvaa. Data saattaa todellisuudessa tyystin sijaita jollain matalaulotteisella alimonistolla. Sanotaan, että datan *implisiittinen dimensio* on alhainen. Vaihtelu muihin suuntiin on vähäistä ja niitä voisi kutsua *nollasuunniksi*. Toisaalta vaihtelua saattaa joissakin suunnissa olla paljonkin, mutta sellaista, ettei siitä ole hyötyä klusteroinnin kannalta. Näitä suuntia voisi kutsua *irrelevanteiksi* tai *kohinasuunniksi*. Suuntia, joissa mielekästä vaihtelua esiintyy, kutsumme *mielekkäiksi suunniksi*. Mielekkyys voidaan siis määritellä suorastaan pragmaattisesti: mie-



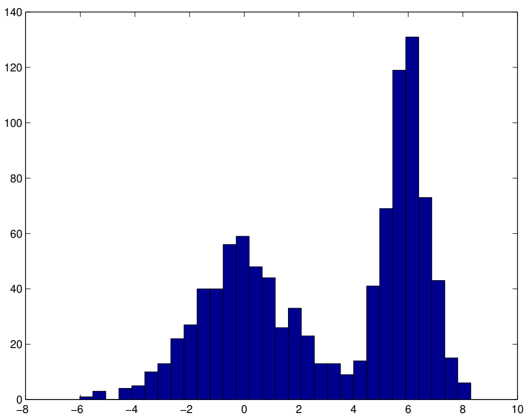
(a) Data alkuperäisessä avaruudessa ($D = 2$)



(b) Jakauma projisoituna x-akselille, klusterit sekoittuvat



(c) Jakauma projisoituna y-akselille, klusterit sekoittuvat



(d) Jakauma projisoituna $(x+y)$-akselille, klusterit selvästi erilliset

Kuva 2: Esimerkki projektion vaikutuksesta klusterointiin

lekkäitä ovat ne suunnat, joissa klusterointi onnistuu hyvin.

Miten näitä mielekkäitä suuntia löydetään? Jos käytettävissä on jokin objektiivinen mitta klusteroinnin onnistumiselle, ilmeinen ajatus olisi kokeilla läpi kaikki eri aliavaruudet, suorittaa klusterointi niissä, ja valita paras.

Tähän hienoon ajatukseen liittyy kiusallinen käytännön seikka. Pelkästään akselinsuuntaisia aliavaruuksia on D-ulotteisessa avaruudessa 2^D erilaista (D-alkioisen joukon osajoukot!). Kokeilujen määrä kasvaa hyvin äkkiä mahdottomaksi. Vapaasuuntaisia aliavaruuksia puolestaan on ääretön määrä!

Voidaan tietysti lähteä liikkeelle hyvin matalaulotteisista aliavaruuksista. Niitähän on luonnollisesti paljon vähemmän: 1-ulotteisia akselinsuuntaisia on D kpl, 2-ulotteisia $\binom{D}{2} \approx D^2/2$ jne. Kääntöpuolena on, että projisointi hyvin matalaulotteiseen aliavaruuteen hävittää paljon etäisyysinformaatiota, joten klusterit eivät ehkä silloin erotu.

On kuitenkin olemassa myös suorempia, esimerkiksi informaatioteoreettisia (entropia, MDL) ja tilastollisia (varianssi, maksimi- ja minimiarvon suhde, moodien lukumäärä) mielekkyyden kriteerejä. Nämä liittyvät läheisesti seuraavassa luvussa tarkasteltavaan (lokaalin projisoidun) etäisyysjakauman käsitteeseen. Jos tällaisia laskennallisia aliavaruuden laatukriteerejä on käytettävissä, klusterointia ei välttämättä tarvitse *suorittaa* jokaisessa tutkittavassa aliavaruudessa.

4 Etäisyysmitan ominaisuuksista

4.1 ”Dimensiokirous”?

Ihmisen geometrinen intuitio toimii hyvin 1...3-ulotteisissa avaruuksissa. Kun avaruuden dimensio on suurempi, tulee vastaan yllättäviä, epäintuitiivisiakin ilmiöitä. Näin voi käydä jo 10...15-ulotteisissa avaruuksissa [BGRS99]. Kun monissa sovelluksissa joudutaan käsittelemään 50...100-ulotteisia tai korkeampiakin avaruuksia, on kyseessä jo selvästi *korkeaulotteinen avaruus* ja intuitio johtaa helposti harhaan.

Korkeaulotteisiin avaruuksiin liitetään usein *dimensiokirouksen* käsite (curse of dimensionality). Tällä viitataan yleensä siihen, että jotkin asiat ovat korkeaulotteisissa avaruuksissa hankalia. Tässä olisi helppo sortua sellaiseen kehäajatteluun, että data-analyysin hankaluus korkeaulotteisissa avaruuksissa selitettäisiin dimensiokirouksen *aiheuttamaksi*, ja dimensiokirous määriteltäisiin *tarkoittamaan* sitä, että korkeaulotteisissa avaruuksissa analyysi on hankalaa. Mutta mitkä asiat itse asiassa ovat hankalia ja miksi?

Ilmeinen hankaluus on, että korkeaulotteiset avaruudet ovat tilavuudeltaan suuria, joten koko avaruuden läpikäyminen esim. tasavälisellä ruudukolla muodostuu käytännössä mahdottomaksi. Sitäpaitsi valtaosa ruuduista on tyhjiä tai lähes tyhjiä, joten korkeatiheyksisiä alueita ei oikein löydy. Näiden ongelmien odottaisi kuitenkin vaivaavan vain ruudukkoon perustuvia menetelmiä. Dimensiokirouksessa täytyy piillä jotain muutakin.

4.2 Kaikki on kaukana

Yleisempi hankaluus liittyy etäisyysmitan käsitteeseen. Karkeasti sanottuna: kun avaruuden dimensio D on suuri, käy niin, että ”kaikki on kaukana”. Klusteroinnin perusajatushan on, että kunkin pisteen $\mathbf{x}$ kanssa samaan klusteriin tulevat ne muut pisteet $\mathbf{y}$, jotka ovat *lähellä* $\mathbf{x}$:ää, jonkin etäisyysmitan mielessä. Jotta tämä on mielekästä, olisi etäisyysmitan oltava sellainen, että se todella erottelee pisteitä lähellä ja kaukana oleviin. Molemmanlaisia pisteitä pitäisi siis *olla* mielekkäitä määriä.

Tämä johtaa luonnollisesti *etäisyysjakauman* [BGRS99, HAK00] käsitteeseen. Tarkastellaan avaruutta ja dataa jostakin kiinnitetystä pisteestä $\mathbf{x}$ käsin. Etäisyysjakauma tarkoittaa kaikkien datapisteiden $\mathbf{y}$ etäisyyksien $dist(\mathbf{x},\mathbf{y})$ jakaumaa.[5] Jakauman kertymäfunktio $N(r)$ kertoo, montako datapistettä sisältyy $\mathbf{x}$-keskiseen r-säteiseen palloon

$$B(\mathbf{x}, r) = \{\mathbf{y} \mid dist(\mathbf{x},\mathbf{y}) \le r\}.$$

Tiheysfunktio $n(r)$ kertoo, paljonko pisteitä on juuri etäisyydellä r (tai kertoisi, jos jakauma olisi jatkuva).

Ihanteellinen tilanne klusteroinnin kannalta olisi, että jakaumassa olisi (ainakin) kaksi huippua, joiden kummankin kohdalla merkittävä määrä pisteitä. Tällöin lähemmän huipun pisteet voitaisiin tulkita pisteen $\mathbf{x}$ kanssa samaan klusteriin kuuluviksi ja kauemmat sen ulkopuolisiksi.

Esimerkiksi euklidinen etäisyys

$$L_2(\mathbf{x},\mathbf{y}) = \Big(\sum_{i=1}^{D} |x_i - y_i|^2\Big)^{1/2}$$

[5] Voitaisiin tarkastella myös globaalisti kaikkien parien $\mathbf{x},\mathbf{y}$ etäisyyksien jakaumaa.

on geometrisen intuition perusteella luonteva etäisyysmitta. Sillä on myös mukavia analyyttisiä ominaisuuksia: esimerkiksi se, että sen *neliösumma* annetun pistejoukon pisteisiin saadaan minimoitua yksinkertaisesti laskemalla pisteiden koordinaattien aritmeettinen keskiarvo kussakin ulottuvuudessa. Mutta onko se hyvä etäisyysmitta klusterointiin? Tutkitaan sen etäisyysjakaumaa.

Pallon sisältämien pisteiden määrään $N(r)$ vaikuttaa kaksi asiaa: pallon tilavuus $V(r)$ ja pisteiden tiheys pallon alueella. Euklidisen pallon tilavuudesta tiedetään, että

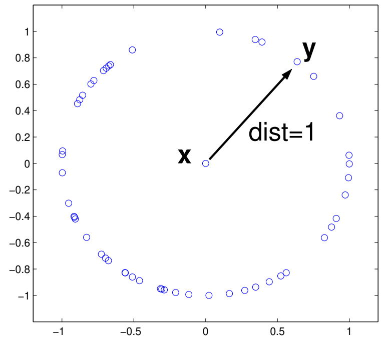
$$V(r) \propto r^D.$$

Jos pistejoukko on rajoitettu (kuten yleensä on), on olemassa jokin r_{max}, jota suuremmilla säteillä ei enää löydy lisää pisteitä, eli sen jälkeen pisteiden tiheys on nolla. Jos pisteiden tiheys pienemmillä r:n arvoillakaan ei vaihtele "kovin paljon", etäisyysjakaumaa dominoi nimenomaan pallon tilavuuden kasvunopeus. Jos pisteiden jakuma olisi tasainen, tämä olisi ilmeistä. Mutta itse asiassa osoittautuu [BGRS99], että näin käy huomattavasti epätasaisemmillakin jakaumilla. Klassinen esimerkki on riippumattomien ja samoin jakautuneiden muuttujien multinormaalijakauma, jossa tiheysfunktio on korkeimmillaan origossa, mutta häviävän pieni osa jakauman massasta on lähelläkään origoa.

Koska pallon kasvunopeus (tilavuuden derivaatta säteen suhteen) puolestaan on verrannollinen suureeseen r^{D-1}, on se suurimmillaan lähellä säteen maksimiarvoa r_{max}. Suurilla D:n arvoilla lähes *kaikki* pallon tilavuus on lähellä pallon kuorta, joten etäisyysjakauman tiheys on lähes kaikkialla nolla ja pallon kuoren kohdalla on kapea piikki. Tällöin euklidisen etäisyysmitan erottelevuus on siis heikko. Itse asiassa hyvin suurilla D:n arvoilla sen etäisyysjakauma muistuttaa klusteroinnin kannalta "pahinta mahdollista" eli *diskreettiä metriikkaa*

$$dist(\mathbf{x}, \mathbf{y}) = \begin{cases} 0 & \text{jos } \mathbf{x} = \mathbf{y}, \\ 1 & \text{jos } \mathbf{x} \neq \mathbf{y} \end{cases}$$

jossa kaikki avaruuden pisteet ovat yhtä kaukana toisistaan (kuva 3)!



Kuva 3: Diskreetti metriikka (pisteestä $\mathbf{x}$ katsottuna)

Euklidinen metriikka voidaan yleistää p-metriikaksi

$$L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{D} |x_i - y_i|^p \right)^{1/p}$$

ja tutkia etäisyysjakaumien käyttäytymistä p:n eri arvoilla ($p \in \mathbb{R}_+$). Charu Aggarwal [Agg01] on tutkinut tätä ja ehdottaa, että korkeaulotteisissa avaruuksissa tulisi L_2:n sijasta käyttää ns. Manhattan-metriikkaa

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} |x_i - y_i|$$

tai jopa ”murtometriikoita” L_p, $p < 1$. Pienet p:n arvot nimittäin johtavat tasaisempaan etäisyysjakaumaan ja sitä kautta parempaan erottelevuuteen.

Havainnossa on luultavasti perää, mutta ei vaikuta uskottavalta, että pelkkä p:n pienentäminen ratkaisisi kaikki korkeaulotteisten avaruuksien ongelmat. Kyseessähän on vain globaali säätöparametri eikä se mitenkään huomioi merkityksellisten aliavaruuksien *suuntia*. On kuitenkin kiintoisaa havaita, että luontevalta tuntuva mitta L_2 on eräiltä ominaisuuksiltaan verraten huono.

4.3 Datan implisiittinen dimensio

On hyödyllistä erottaa käsitteet *esitysmuodon* dimensio ja datan *implisiittinen* dimensio. Edellinen kuvaa sitä avaruutta, jossa datapisteet on esitetty, esim. $\mathbb{R}^{100}$. Jälkimmäinen kuvaa itse datan ominaisuuksia — toivottavasti melko lailla riippumatta esitysmuodosta.

Implisiittisellä dimensiolla d tarkoitetaan, että data todella sijaitsee avaruuden jollakin d-ulotteisella alimonistolla tai ainakin sen lähistöllä.

Implisiittinen dimensio voi olla paljon pienempi kuin esitysmuodon dimensio. D. Scott esittää jopa, että todellisessa tilastollisessa aineistossa on harvoin mielekästä vaihtelua useammassa kuin *viidessä* ulottuvuudessa ([Sco92], sivu 3). Tämä ehkä pitää parhaiten paikkansa ”perinteisissä” tilastollisissa aineistoissa. Jos myös kohinaa on vähän, on datan implisiittinen dimensio tällöin enintään 5.

Jos käytettävissä on etäisyysmitta *dist*, datan implisiittistä dimensiota voidaan mitata samaan tapaan kuin edellisessä aliluvussa, etäisyysjakaumaa eli r-säteisten pallojen sisältämien datapisteiden määriä $N(\mathbf{x}, r)$ tarkastelemalla.

Määritelmä 1 *Datan implisiittinen dimensio on d, jos*

$$N(\mathbf{x}, r) \propto r^d$$

eli jos N riippuu säteen d:nnestä potenssista.

Huomataan, että näin määritelty luku d ei ole välttämättä globaali suure. Eri pisteiden ympärillä dimensio voi olla erisuuri. Yhdenkin pisteen ympärillä dimensio voi olla eri skaaloissa erisuuri. Implisiittinen dimensio riippuu itse datasta, tarkasteltavasta paikasta ja skaalasta, sekä käytetystä etäisyysmitasta. Sen sijaan se *ei* suoraan riipu avaruuden ominaisuuksista. Mikään ei myöskään määrää, että d olisi kokonaisluku.

Näin määritelty ”implisiittinen dimensio” on varsin helposti empiirisesti laskettavissa mielivaltaisessa avaruuden pisteessä: lasketaan etäisyysjakauman kertymäfunktio riittävän monella r:n eri arvolla ja tutkitaan, mikä potenssifunktio sopii havaintoihin parhaiten.

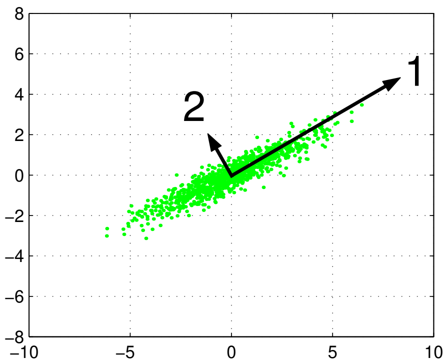
4.4 Irrelevantit ulottuvuudet ja kohina

Edellä puhuttiin siitä, että jokin ulottuvuus on merkityksetön, jos sen suunnassa ei ole varianssia.

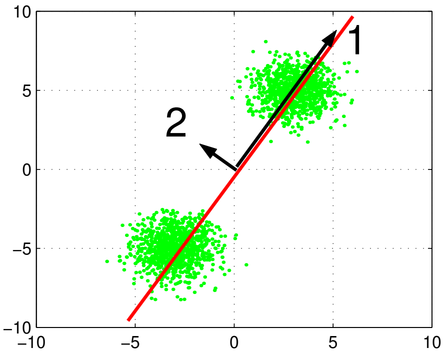
Vielä suurempi ongelma on, että varianssia on jossain suunnassa paljonkin – vaan ei merkityksellistä, klusteroituvaa varianssia. Jos data on jossain suunnassa on täysin tasajakautunut eikä lainkaan korreloi muiden suuntien kanssa, ei tästä suuresta varianssista ole apua klusteroinnissa, joten suuntaa voisi nimittää ”kohinasuunnaksi”. Esimerkiksi pääkomponenttianalyysi, johon tutustutaan seuraavassa luvussa, pitäisi tällaista suuntaa *erityisen* tärkeänä suuntana.

5 Pääkomponenttianalyysi

Pääkomponenttianalyysi on tunnetuin dimensionvähennysmenetelmä. Sillä voidaan etsiä ne avaruuden keskenään kohtisuorat suunnat, joissa datan *varianssi* on suurin (kuva 4). Sitä voidaan käyttää klusteroinnin esikäsittelyvaiheena: projisoidaan data muutaman suurimman pääkomponentin virittämään aliavaruuteen, ja klusteroidaan sitten. Tämä menetelmä on suosittu, sillä se on yksinkertainen ja tehokas toteuttaa, eikä se ole rajoittunut akselinsuuntaisiin aliavaruuksiin.



Kuva 4: Pääkomponentit (1=suurin)



Kuva 5: $D = 2$, $k = 2$, 1-ulotteinen aliavaruus

Jotta tässä on järkeä, täytyy pääkomponenttianalyysin löytämien suurivarianssisten suuntien olla myös klusteroinnin kannalta mielekkäitä. Näin usein onkin. Tämä selittyy osapuilleen seuraavasti: Jos data koostuu k:sta klusterista, ja $k \leq D$, on ilmeistä, että täsmälleen klusterien keskipisteiden kautta kulkee jokin $(k-1)$-ulotteinen aliavaruus (ks. kuva 5). Jos lisäksi klusterien keskipisteet ovat kaukana toisistaan, niin varianssi on tämän aliavaruuden suunnassa suhteellisen suuri.[6] On siis odotettavissa, että jos valitaan $d = k - 1$ pääkomponenttia, saadaan juuri tämä klusterien keskipisteiden kautta kulkeva aliavaruus ja klusterointi onnistuu hyvin, koska kaikki klusterien välinen etäisyys on tallella.

Ei ole kuitenkaan millään tavoin taattua, että näin käy. Jos klustereilla on paljon sisäistä varianssia klusteroinnin kannalta irrelevanttiin suuntaan (esim. kuvassa 5 suuntaan 2), pääkomponenttianalyysi voikin valita nämä suunnat, vaikka ne ovat erityisen *huonoja* suuntia klusteroinnin kannalta, pelkkiä ”kohinasuuntia”. Jotkin irrelevantit datan attribuutit saattavat olla jopa *tasajakautuneita* laajalle välille, jolloin niiden varianssi varmasti on suuri. Sellainen esimerkki esitetään artikkelissa [AGGR98].

Toiseksi varianssit eri suunnissa riippuvat mitta-asteikosta. Jos eri suuntien asteikot eivät ole vertailukelpoisia, ei niiden varianssienkaan vertailussa ole mieltä [Agg01].

Nämä ongelmat liittyvät siihen, että pääkomponenttianalyysi perustuu varianssiin, joka ei ole erityisen informatiivinen kriteeri klusteroinnin kannalta.

Jos klusterien lukumäärä on suuri $(k > D)$, ei mikään alle D-ulotteinen aliavaruus todennäköisesti kulje kaikkien klusterien keskipisteiden kautta, joten osa klusterien välisestä etäisyydestä väkisin katoaa projektiossa. Pääkomponenttianalyysi on luonteeltaan globaali ja lineaarinen, mikä ei johda hyviin tuloksiin, jos data todellisuudessa sijaitsee kaarevalla alimonistolla.

[6] Näissä suunnissa nimittäin varianssia kasvattaa klusterien välinen etäisyys, kun taas kohtisuoriin suuntiin varianssi syntyy vain klusterien sisäisestä varianssista.

6 Aliavaruusklusterointi

6.1 CLIQUE-algoritmi

Aliavaruusklusterointialgoritmeista ehkä tunnetuin on Rakesh Agrawalin *et al.* kehittämä CLIQUE [AGGR98]. Se on ruudukkoon perustuva, liittävä (agglomerative) algoritmi, joka käsittelee vain akselinsuuntaisia aliavaruuksia. Klusterin määritelmä perustuu yksikköruutujen tiheyteen ja alueen yhtenäisyyteen.

6.1.1 Algoritmin kuvaus

Syötteenä algoritmille annetaan

- itse datapisteet $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}$,
- ruudukon jako-osien määrä ξ, esim. 10, ja
- tiheyskynnys τ, esim 5 %.

Perusideana on, että kokeillaan erilaisia d-ulotteisia aliavaruuksia, pienistä d:n arvoista alkaen.[7] Kun d-ulotteista akselinsuuntaista aliavaruutta kokeillaan, kukin noista d:stä akselista jaetaan ξ osaan tasavälein. Näin saadaan ξ^d samankokoista ruutua. Ruutuja ei suorastaan käydä läpi yksitellen, mutta sopivalla algoritmilla selvitetään, mitkä yksittäiset ruudut ovat *tiheitä* eli sisältävät vähintään osuuden τ kaikista datapisteistä.

Kun tiheitä ruutuja on löydetty, toisiaan koskettavat tiheät ruudut liitetään sitten yhteen klustereiksi. Tuloksena on siis joukko *tiheiden ruutujen yhtenäisiä rykelmiä* eli klustereita.

Vaikka jostain aliavaruudesta on löydetty klusteri tai useita, jatketaan edelleen muiden aliavaruuksien kokeilemista ja tulostetaan myös niistä mahdollisesti löytyvät klusterit.

Tulokset esitetään havainnollisesti disjunktiivisessa normaalimuodossa (DNF). Kun klusterit on löydetty, algoritmissa on jälkikäsittelyvaihe, joka etsii kullekin klusterille mahdollisimman yksinkertaisen DNF-esityksen, eli esityksen akselinsuuntaisten laatikoiden yhdisteenä. Jos on vaikkapa valittu 2-ulotteinen aliavaruus, joka koostuu attribuuteista *age* ja *salary*, voi jonkin klusterin DNF-esitys olla esimerkiksi seuraavanlainen kolmen laatikon yhdiste:

$$\begin{aligned} & ((30 \le age < 50) \wedge (4 \le salary < 8)) \\ \vee \quad & ((40 \le age < 60) \wedge (2 \le salary < 6)) \\ \vee \quad & ((45 \le age < 70) \wedge (1 \le salary < 6)). \end{aligned}$$

Tiheät yksikköruudut, joista klusteri koostuu, voivat olla selvästi pienempiä kuin klusterin esityksessä käytetyt laatikot. Yksi ruutu voi olla esimerkiksi

$$(30 \le age < 35) \wedge (4 \le salary < 5),$$

jos attribuutin *age* jakoväli on 5 ja attribuutin *salary* jakoväli 1.

[7]Kokeilua nopeutetaan karsimalla tutkittavia ruutuja ja tutkittavia aliavaruuksia MDL-tyyppisellä kriteerillä. Tämä on kuitenkin luonteeltaan nopeusoptimointi, ei algoritmin perusominaisuus.

6.1.2 CLIQUE:n kritiikkiä

Algoritmi on idealtaan selkeä. Klusterin määritelmä on havainnollinen, mutta jotenkin karkea. Määritelmä perustuu globaaliin tiheyskynnykseen. Todellisuudessahan jotkin klusterit voivat olla huomattavasti harvempia kuin jotkin toiset.

Algoritmi kykenee löytämään klustereita, jotka ovat pitkulaisia eri suuntiin (eri aliavaruuksissa), myös sellaisissa tilanteissa, joissa esimerkiksi pääkomponenttianalyysistä ei ole apua. Koska algoritmi on liittävä, pystyy se löytämään myös epäsäännöllisen muotoisia klustereita. Jos käytetään pieniä laatikoita (suuri ξ), on klusterin muoto hyvin vapaa, mutta tällöin voi olla vaikea löytää tarpeeksi paljon tiheitä yksikköruutuja, että ne muodostavat mielekkäitä yhtenäisiä alueita (klustereita).

On huomattava, että CLIQUE-algoritmi ei tuota avaruuden *ositusta*, vaan vain joukon erilaisia, mahdollisesti toisiaan leikkaavia avaruuden alueita, jotka kukin ovat yhtenäisiä ja riittävän tiheitä. Se ei siis sellaisenaan sovi luokittelutehtäviin, mutta tiedon louhintaan se on omiaan: käyttäjälle kyetään esittämään havainnollinen tiivistelmä siitä, mitkä avaruuden alueet ovat erityisen tiheitä ja miten näitä alueita voi yksinkertaisesti luonnehtia muutamalla epäyhtälöllä.

Rajoittuminen akselinsuuntaisiin aliavaruuksiin on luultavasti tehty osittain teknisistä syistä. Tekijät esittävät tosin pääasialliseksi syyksi sen, että datan alkuperäisiä attribuutteja (esim. *age*, *salary*) koskevat epäyhtälöt ovat käyttäjän helposti ymmärtämiä, kun taas esimerkiksi niiden lineaarikombinaatiot eivät. Tämäkin pitänee paikkansa ainakin niissä sovelluksissa, joihin algoritmi on tarkoitettu.

Klusterin määritelmän karkeuden, osituksen puuttumisen ja akselinsuuntaisuusrajoituksen takia CLIQUE sopii mielestäni vain joihinkin korkeaulotteisen datan analysointitehtäviin.

6.2 Muita algoritmeja

CLIQUE-algoritmista on kehitetty muunnelmia: ENCLUS, jossa aliavaruuden valintakriteeri on vaihdettu entropiaan perustuvaksi, ja MAFIA, jossa tasavälinen ruudukko on korvattu adaptiivisella.

On myös vapaasuuntaisia aliavaruuksia käyttäviä menetelmiä. Charu Aggarwalin[8] ja Philip Yun ORCLUS [AY00] etsii datalle $(k+1)$-osituksen seuraavasti. Klustereita on k kpl ja kuhunkin klusteriin liittyy jokin joukko vapaasuuntaisia vektoreita, joiden virittämässä aliavaruudessa kyseisen klusterin pisteet erottuvat selvästi klusterin ulkopuolisista pisteistä. Tässä käytetään eräänlaista energian minimointikriteeriä. Klustereihin sopimattomat pisteet muodostavat oman $(k+1)$:nnen joukkonsa (outliers).

Hinneburgin ja Keimin OPTIGRID [HK99] etsii vapaasuuntaisia leikkaavia hypertasoja (cutting planes), jotka eivät osu korkeatiheyksisiin alueisiin ja jotka osittavat datan mahdollisimman hyvin. Hypertasolla eli $D-1$-ulotteisella aliavaruudella leikkaaminen vastaa itse asiassa sitä, että data projisoidaan kyseistä hypertasoa vastaan kohtisuoraan 1-ulotteiseen aliavaruuteen ja näin saatu 1-ulotteinen data yritetään jakaa kahteen osaan yhdellä kynnysarvolla.

Hyvä yleiskuvaus nykyisistä aliavaruusklusteroinnin algoritmeista löytyy Berkhinin katsauksesta [Ber02].

[8] Aggarwalia ei tule sekoittaa edellisessä aliluvussa mainittuun Rakesh Agrawaliin.

7 Yhteenveto

Jouduttaessa työskentelemään korkeaulotteisen datan kanssa havaitaan, etteivät tavanomaiset etäisyyden, lähinaapurin ja klusterin käsitteet ole enää täysin mielekkäitä. Tämä liittyy olennaisesti korkeaulotteisten metristen avaruuksien erikoisiin ominaisuuksiin.

Tyypilliselle datalle on kuitenkin löydettävissä mielekkäitä *aliavaruuksia*, joiden suunnassa etäisyys, lähinaapuri ja klusterointi edelleen ovat järkeviä. Mielekäs aliavaruus on usein lokaali käsite: eri datapisteiden ympäristöissä eri suunnat ovat mielekkäitä.

Jos merkityksellinen aliavaruus on globaali eikä tästä aliavaruudesta poikkeavaa kohinaa ole liikaa, voidaan hyviä tuloksia saada perinteisillä yksinkertaisilla dimensionvähennysmenetelmillä, kuten pääkomponenttianalyysillä.

Luotettavammat menetelmät perustuvat kuitenkin kehittyneempiin dimensionvalintakriteereihin. Aliavaruusklusterointi tarjoaa käyttökelpoisia menetelmiä näiden lokaalisti mielekkäiden suuntien löytämiseen ja korkeaulotteisenkin datan klusterointiin.

Viitteet

[Agg01] Charu C. Aggarwal. On the effects of dimensionality reduction on high dimensional similarity search. In *Symposium on Principles of Database Systems*, 2001.

[AGGR98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD International Conference on Management of Data*, 1998.

[AY00] Charu C. Aggarwal and Philip S. Yu. Finding generalized projected clusters in high dimensional spaces. In *SIGMOD-00, 2000*, pages 70–81, 2000.

[Ber02] Pavel Berkhin. Survey of clustering data mining techniques, 2002.

[BGRS99] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? *Lecture Notes in Computer Science*, 1540:217–235, 1999.

[HAK00] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *The VLDB Journal*, pages 506–515, 2000.

[HK99] Alexander Hinneburg and Daniel A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *The VLDB Journal*, pages 506–517, 1999.

[Sco92] D. W. Scott. *Multivariate Density Estimation*. Wiley, New York, 1992.