Hierarkkinen klusterointi

Kari Lehmussaari
kari.lehmussaari@helsinki.fi

Klusterointimenetelmät-seminaari
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C-2002-54, s. 76-85, marraskuu 2002

Tiivistelmä

Tässä raportissa selitetään käsitteellisellä tasolla hierarkkisen klusteroinnin periaatteet, kuten kokoava (agglomerative) eli alhaalta 'lehdistä' ylöspäin tapahtuva ja jakava (divisive) eli ylhäältä 'juuresta' alaspäin tapahtuva klusterointi. Lisäksi esitellään kaikkiin hierarkkisiin klusterointimenetelmiin keskeisesti liittyvä etäisyysmatriisi, jonka sisältämien etäisyyksien avulla kokoavassa menetelmässä valitaan jokaisella tasolla yhdistettävät klusterit ja jakavassa menetelmässä päätetään mikä klustereista jaetaan osiin tietyllä tasolla.

Etäisyysmatriisin laskentaan on olemassa monia erilaisia kaavoja, joista tässä raportissa keskitytään erityisesti tutkimaan lähin naapuri (Single-Linkage) laskentamenetelmää, joka perustuu siihen että etäisyys klustereiden välillä määritellään klustereissa lähimpänä toisiaan sijaitsevien pisteiden perusteella ja naapurikeskiarvo (Average-Linkage) menetelmää, jossa etäisyys klustereiden välillä lasketaan kaikkien klustereihin kuuluvien pisteiden keskiarvoista.

Käsitteiden ja etäisyysmatriisin lisäksi raportissa esitellään myös muutamia tunnettuja hierarkkisiin klusterointimenetelmiin ja erityisesti etäisyysmatriisin laskentaan liittyviä ongelmia ja rajoituksia, kuten klusteroinnin staattisuutta ja aikavaativuutta, häiriöpisteiden vaikutusta klusteroinnin tulokseen ja kyvyttömyyttä löytää sisäkkäisiä klustereita. Ongelmien tutkimisen kautta mietitään myös hierarkkisen klusteroinnin ja eri etäisyysmatriisin laskentamenetelmien soveltuvuutta tietyn tyyppisen tiedon klusterointiin. Lopuksi esitetään vielä yhteenveto hierarkkisen klusteroinnin hyvistä ja huonoista puolista.

1 Johdanto

Tämän seminaariraportin tarkoituksena on yksityiskohtaisesti esitellä hierarkkisen klusteroinnin periaatteet, siihen liittyvät ja yleisesti käytössä olevat menetelmät, sekä käsitteet kuten kokoava (agglomerative) klusterointi, jakava (divisive) klusterointi ja etäisyysmatriisi. Lisäksi raportissa pyritään useiden esimerkkien kautta havainnollistamaan hierarkkisten klusterointimenetelmien toimintaa käytännössä ja esittelemään ja analysoimaan jo saatuja koetuloksia, sekä hierarkkisen klusteroinnin hyviä ja huonoja puolia.

Luvussa 2 ensin käydään lävitse itse hierarkkinen klusterointi ja siihen liittyvä käsitteistö kuten kokoava ja jakava hierarkkinen klusterointi ja esitellään tarkemmin kokoavien klusterointimenetelmien toimintaa yleisellä tasolla.

Luvussa 3 esitellään lyhyesti joukko kokoavan klusteroinnin yhteydessä sovellettavia etäisyysmatriisin laskentamenetelmiä, kuten lähin naapuri (Single-Linkage), naapurikeskiarvo (Average-Linkage), kaukaisin naapuri (Complete-Linkage), keskipiste analyysi (Centroid Analysis) ja Wardin -menetelmä (Ward’s Method) [JMF99, SLH01]. Lisäksi esitellään

kokoavien hierarkkisten klusterointimenetelmien toimintaa käytännön esimerkkien avulla. Erikseen käsiteltävinä ovat lähin naapuri ja naapurikeskiarvo -menetelmät. Esimerkkien avulla pyritään simuloimaan näitä kahta yleistä tapaa etäisyysmatriisin laskemiseksi ja näiden matriisien pohjalta luotavien hierarkkisten klustereiden muodostamista, sekä vertailemaan saatuja tuloksia keskenään. Lisäksi selvitetään hierarkkisen klusterointi algoritmien soveltuvuutta erilasten tietotyyppien ja rakenteiden klusterointiin [JMF99].

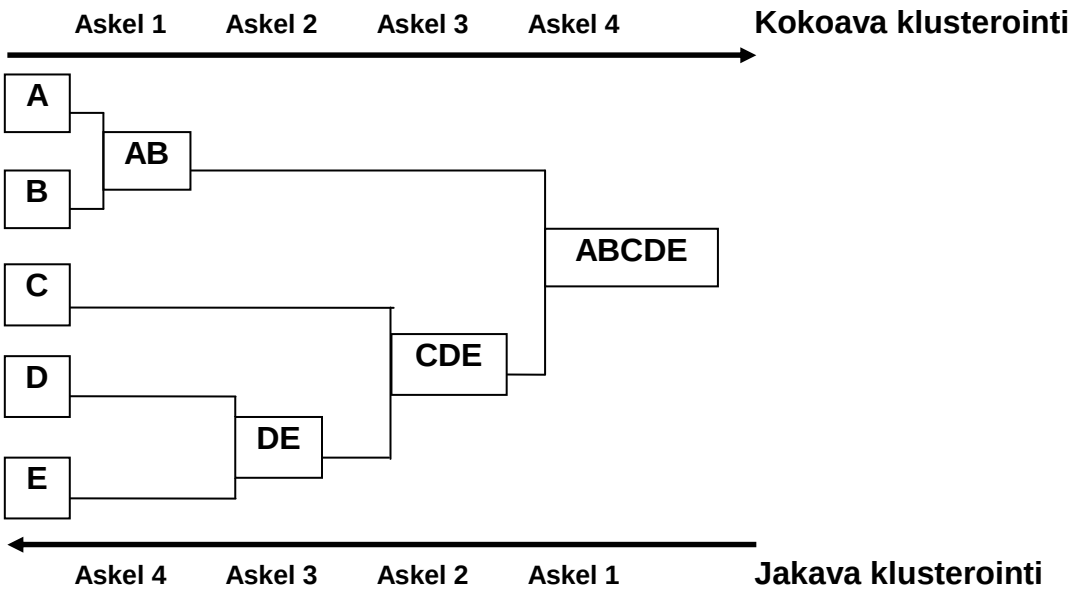
Lopuksi luvussa 4 esitetään edellisissä luvuissa tehtyjen vertailuiden ja esimerkkien pohjalta vedettävissä olevia johtopäätöksiä ja tuodaan tiivistelmänomaisesti esiin hierarkkisten klusterointimenetelmien vahvuudet ja heikkoudet [JMF99].

2 Hierarkkinen Klusterointi

2.1 Kokoava ja jakava klusterointi

Hierarkkiset klusterointimenetelmät ovat joko kokoavia (agglomerative) tai jakavia (divisive). Kokoavissa menetelmissä klusterointi etenee sarjana, jossa *n* kappaletta klusteroitavana olevia olioita yhdistettään ryhmiksi etäisyysmatriisiin laskettujen etäisyyksien perusteella. Vastaavasti jakavissa menetelmissä klusterointi etenee sarjana tapahtumia, jossa *n* kappaletta klusteroitavana olevia olioita jaetaan useampaan klusteriin ja klusteroinnin alkaessa kaikki alkiot kuuluvat samaan klusteriin. Koska kokoavat menetelmät ovat näistä kahdesta huomattavasti yksinkertaisempia toteuttaa, ovat ne myös huomattavasti yleisemmin käytettyjä kuin jakavat menetelmät [KHK99].

Klusteroinnin hierarkiaan tapahtuneet muutokset ja muodostetut klusterit ovat luonteeltaan pysyviä, siten että mahdollisia virheitä klustereiden muodostamisessa on mahdotonta korjata niiden synnyttyä [HKK99]. Seuraava kuva (kuva 1) havainnollistaa kokoavan ja jakavan klusteroinnin eroja, kun klusteroitavana on viidestä pisteestä {A, B, C, D, E} koostuva joukko.



Kuva 1: Kokoava (agglomerative) ja jakava (divisive) klusterointi

Jakavat menetelmät voidaan jakaa monoteistisiin (monothetic) ja polyteistisiin (polythetic) menetelmiin sen mukaan miten ne suhtautuvat klusteroitavien olioiden ominaisuuksiin. Monoteistiset menetelmät perustuvat erotteluun joka tehdään yhden tarkkaan

määrätyn ominaisuuden perusteella ja polyteistiset puolestaan perustuvat kaikkien ominaisuuksien arvojen perusteella tehtävään jakoon.

Polyteistisistä jakavista menetelmistä mainittakoon Mac-Naughton-Smithe et al. esittämä menetelmä, jossa tietystä klusterista yksinkertaisesti erotetaan aina se kokonaisuus, jonka eroavaisuus suhteessa kyseiseen jäljellä olevaan klusteriin nähden on kaikkein suurin. Monoteistisiä menetelmiä puolestaan käytetään yleensä kun klusteroitavana on binääri-muotoista tietoa, jolloin jako voidaan yleisesti suorittaa niihin alkioihin joilla joko on tai vastaavasti ei ole tiettyä ominaisuutta [He99].

2.2 Kokoavien hierarkkisten algoritmien toiminta

Yleisellä tasolla kaikki hierarkkiset kokoavat algoritmit toimivat samalla tasolla ja eroavat toisistaan lähinnä siinä miten klusteroitavan tiedon samankaltaisuus on määritelty ja millä kaavoilla pisteiden ja vastaavasti klustereiden etäisyys toistaan lasketaan [SLH01]. Yleisellä tasolla kaikkien kokoavien hierarkkisten klusterointi menetelmien toimintaa voidaankin kuvata seuraavasti:

- Oletetaan että on olemassa *N* datapistettä joukossa R^T.
- Määritellään joukon R^T pisteille samankaltaisuuden/eroavaisuuden määrittelevät ehdot, joiden avulla etäisyysmatriisi voidaan laskea ja valitaan etäisyysfunktio, jonka avulla pisteiden etäisyys toisistaan voidaan laskea (esimerkiksi lähin naapuri tai kaukaisin naapuri). Tässä pisteiden etäisyys toisistaan voi olla yksinkertaisesti euklidinen etäisyys tai mikä hyvänsä muu monimutkaisempi etäisyys. Joskus etäisyysfunktiota ei ole tarpeen määritellä ja pisteiden etäisyydet voidaan antaa suoraan *NxN* matriisissa.
- Edetään kokoavan mallin mukaisesti alhaalta ylöspäin siten että aluksi on olemassa *N* klusteria joista jokainen sisältää yhden datapisteen ja lopulta yksi klusteri joka sisältää *N* datapistettä.
- Aloitetaan N klusterista joista jokainen koostuu yhdestä datapisteestä. Jokaisella iteraatiokierroksella toimitaan seuraavasti:

1. Käyttäen senhetkistä etäisyysmatriisia, valitse kaksi toisiaan lähinnä olevaa klusteria.
2. Päivitä klustereiden määrä ja sisältö yhdistämällä kaksi lähintä klusteria.
3. Päivitä etäisyysmatriisi laskemalla uudet etäisyydet muodostettuun klusteriin.

- Toistetaan kunnes kaikki datapisteet on yhdistetty yhdeksi klusteriksi tai ennalta sovittu lopetusehto on saavutettu.
- Jos kaksi datapistettä on toisistaan yhtä etäällä, valitaan mielivaltaisesti toinen niistä. Tämä valinta vaikuttaa myös syntyvän hierarkian rakenteeseen.

3 Klusterointi menetelmiä

3.1 Etäisyysmatriisin laskenta

Kaikkein tärkein kohta edellisessä kappaleessa esitetyssä yleisessä kuvauksessa kokoavien hierarkkisten algoritmien toiminnasta on etäisyysmatriisin laskeminen, jonka perusteella valitetaan klusteriin yhdistettävät alkiot. Etäisyysmatriisin laskemiseen on olemassa monia kaavoja, mutta ja niistä yleisimmät ovat lähin naapuri (Single-Linkage), kaukaisin naapuri (Complete-Linkage) ja naapurikeskiarvo (Average-Linkage) kaavat [JMF99].

Lähin naapuri kaavassa ryhmien eli klustereiden välinen etäisyys on määritelty kunkin klusterin lähimpien jäsenten väliseksi etäisyydeksi ja etäisyysmatriisi *D*, voidaan laskea kaavalla:

$$D(C_1,C_2) = \min_{x1 \in C1,\ x2 \in C2} \{ d(x_1,x_2) \},$$

jossa C_1,C_2 ovat klustereita ja $d(x_1,x_2)$ on kyseisiin klustereihin kuuluvien pisteiden välinen etäisyys.

Toinen yleinen kaava on kaukaisin naapuri, joka puolestaan määrittelee etäisyyden kahden klustereihin kuuluvan pisteen suurimpana etäisyytenä ja jonka etäisyysmatriisi D saadaan kaavalla:

$$D(C_1,C_2) = \max_{x1 \in C1,\ x2 \in C2} \{ d(x_1,x_2) \},$$

jossa C_1,C_2 ovat klustereita ja $d(x_1,x_2)$ on kyseisiin klustereihin kuuluvien pisteiden välinen etäisyys. Ainoaksi eroksi siis jää, että kun pisteiden välisistä etäisyyksistä lähin naapuri kaavassa valittiin lyhin niin kaukaisin naapuri kaava etsii kaikkein pisintä etäisyyttä.

Kolmantena kaavana on naapurikeskiarvo, joka määrittelee etäisyydeksi klustereihin kuuluvien piste-parien etäisyyksien keskiarvon ja jossa etäisyys matriisi D voidaan laskea kaavalla:

$$D(C_1,C_2) = 1/\#(C_1)\#(C_2) \sum_{x1 \in C1,\ x2 \in C2} d(x_1,x_2),$$

jossa $\#(C_1)\#(C_2)$ on kyseisiin klustereihin kuuluvien olioiden lukumäärien tulo.

Muita hyvin tunnettuja kaavoja ovat vielä lisäksi mm. Wardin -menetelmä ja klusterin keskipisteen laskemiseen perustuva, keskipiste analyysi, jossa etäisyys klustereiden välillä on määritelty klustereiden keskipisteiden väliseksi etäisyydeksi. Tämän menetelmän huonona puolena voidaan kuitenkin todeta, että jos yhdistettävät klusterit ovat hyvin eri kokoiset niin uuden klusterin keskipiste on hyvin lähellä suuremman klusterin keskipistettä, jolloin pienemmän klusterin ominaisuudet käytännössä hukkuvat kokonaan.

Wardin -menetelmässä etäisyys puolestaan lasketaan klusteriin kuuluvien pisteiden hajonnasta, kyseisen klusterin keskipisteeseen nähden. Itse hajonta puolestaan lasketaan laskemalla yhteen klusteriin kuuluvien pisteiden neliöön korotettu etäisyys toisistaan ja klusterin keskipisteestä [He99].

3.2 Lähin naapuri -menetelmä

Kaikkein tunnetuin ja yksinkertaisin hierarkkisista klusterointimenetelmistä on lähin naapuri kaavaan perustuva kokoava klusterointimenetelmä. Sen toimintaa on havainnollistettu seuraavan esimerkin avulla, jossa aluksi on määritelty etäisyysmatriisi D, viidelle klusteroitavalle pisteelle {A,B,C,D,E}. Oletetaan että alkiot eli tässä esimerkissä pisteet, sijaitsevat kaksiulotteisessa avaruudessa ja niiden etäisyys toisistaan on määritelty yksinkertaisesti euklidisena etäisyytenä.

	A	B	C	D	E
A	0.0	2.0	6.0	10.0	9.0
B	2.0	0.0	5.0	9.0	8.0
C	6.0	5.0	0.0	7.0	11.0
D	10.0	9.0	7.0	0.0	6.0
E	9.0	8.0	11.0	6.0	0.0

Kuva 2: Etäisyysmatriisi D={A,B,C,D,E}

Matriisista (kuva 2) nähdään että D_{AB} on etäisyydeltään kaikkein pienin, joten ensimmäinen klusteri muodostetaan pisteistä A ja B. Tämän jälkeen lasketaan uusi etäisyysmatriisi, jonka ensimmäinen rivi ja sarake on varattu ensimmäiselle klusterille (A,B) ja siis koostuu pisteistä {(A,B), C, D, E}. Seuraavat etäisyydet pitää siis laskea uudelleen.

$$D_{(A,B)C} = min\ [D_{AC},\ D_{BC}] = D_{BC} = 5.0,$$
$$D_{(A,B)D} = min\ [D_{AD},\ D_{BD}] = D_{BD} = 9.0,$$
$$D_{(A,B)E} = min\ [D_{AE},\ D_{BE}] = D_{BE} = 8.0$$

Tämän jälkeen voidaan muodostaa uusi matriisi, joka näyttää seuraavalta:

	A,B	C	D	E
A,B	0.0	5.0	9.0	8.0
C	5.0	0.0	7.0	11.0
D	9.0	7.0	0.0	6.0
E	8.0	11.0	6.0	0.0

Kuva 3: Etäisyysmatriisi D={(A, B), C, D, E}

Uudesta matriisista (kuva 3) nähdään että etäisyys $D_{(AB)C}$ on kaikkein pienin, joten seuraava klusteri muodostuu klusterista (A,B) ja pisteestä C. Tämän jälkeen joudutaan jälleen määrittelemään uudet etäisyydet klusterin (A,B,C) ja pisteiden D ja E välille ja laskemalla saadaan seuraavat etäisyydet:

$$D_{(A,B,C)D} = min\ [D_{AD},\ D_{BD},\ D_{CD}] = D_{CD} = 7.0,$$
$$D_{(A,B,C)E} = min\ [D_{AE},\ D_{BE},\ D_{CE}] = D_{BE} = 8.0,$$

Täten uusi etäisyysmatriisi muodostuu pisteistä {(A,B,C), D, E}, jossa ensimmäinen rivi ja sarake on varattu klusterille (A,B,C) ja näyttää siis seuraavalta:

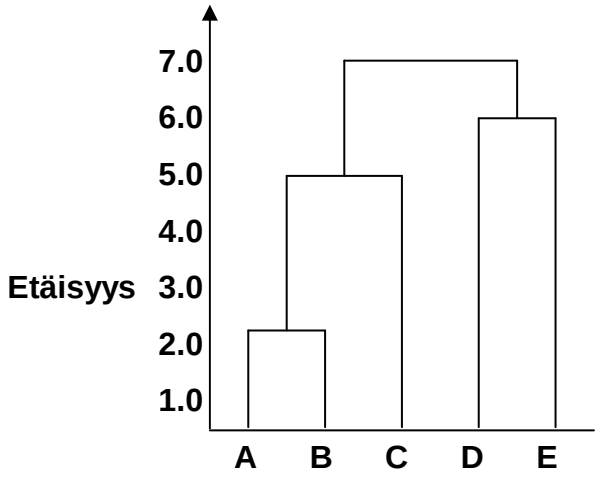
	A,B,C	D	E
A,B,C	0.0	7.0	8.0
D	7.0	0.0	6.0
E	8.0	6.0	0.0

Kuva 4: Etäisyysmatriisi D={(A,B,C), D, E}

Lyhin etäisyys tässä matriisissa (kuva 4) on nyt D_{DE}, joten seuraavaksi yhdistetään piste D ja E yhdeksi klusteriksi. Lopuksi yhdistetään vielä klusterit (A,B,C) ja (D,E), jonka jälkeen klusterointi on valmis. Kerraten klusteroinnin eri vaiheet, klusterointi järjestys oli siis seuraava:

{A, B, C, D, E},
{(A, B), C, D, E},
{(A, B, C), D, E},
{(A, B, C), (D, E)},
{(A, B, C, D, E)}

Varsinaisen hierarkian havainnollistamiseksi kyseinen klusterointi järjestys voidaan myös esittää hierarkiaa kuvaavana dendrogrammina (kuva 5), jossa pystysuoralla akselilla on määritelty klusterien välinen etäisyys kun ne yhdistettiin ja vaakasuoralla akselilla pisteet, joista klusterit on muodostettu [JMF99].



Kuva 5: Hierarkiaa kuvaava dendrogrammi

3.3 Naapurikeskiarvo -menetelmä

Toinen yleisesti käytössä oleva yksinkertainen hierarkkinen klusterointi algoritmi on naapurikeskiarvo kaavalla saatua etäisyysmatriisia käyttävä kokoava menetelmä. Havainnollistetaan algoritmin toimintaa käyttämällä samaa viiden klusteroitavan pisteen joukkoa kuin lähin naapuri klusterointi menetelmässä. Alkutilanteessa, kun kaikki pisteet vielä kuuluvat omiin klustereihinsa, on etäisyysmatriisi siis täsmälleen samanlainen kuin lähin naapuri menetelmässä. Jälleen lyhin etäisyys on D_{AB}, joten pisteet A ja B yhdistetään ja ne muodostavat ensimmäisen klusterin. Tämän jälkeen lasketaan uusi pisteille uudet etäisyydet ja käyttäen yllä esitetty kaavaa saadaan:

$$D_{(A,B)C} = 1/2(D_{AC} + D_{BC}) = 5.5,$$
$$D_{(A,B)D} = 1/2(D_{AD} + D_{BD}) = 9.5,$$
$$D_{(A,B)E} = 1/2\ (D_{AE} + D_{BE}) = 8.5$$

Tämän jälkeen voidaan muodostaa uusi etäisyysmatriisi (kuva 6), joka näyttää seuraavalta:

	A,B	C	D	E
A,B	0.0	5.5	9.5	8.5
C	5.5	0.0	7.0	11.0
D	9.5	7.0	0.0	6.0
E	8.5	11.0	6.0	0.0

Kuva 6: Etäisyysmatriisi D={(A,B), C, D, E}

Tälläkin kertaa lyhin etäisyys on $D_{(AB)C}$, joten yhdistetään piste C klusteriin (A,B) ja lasketaan uusi etäisyysmatriisi klusterin (A,B,C) ja pisteiden D ja E välille.

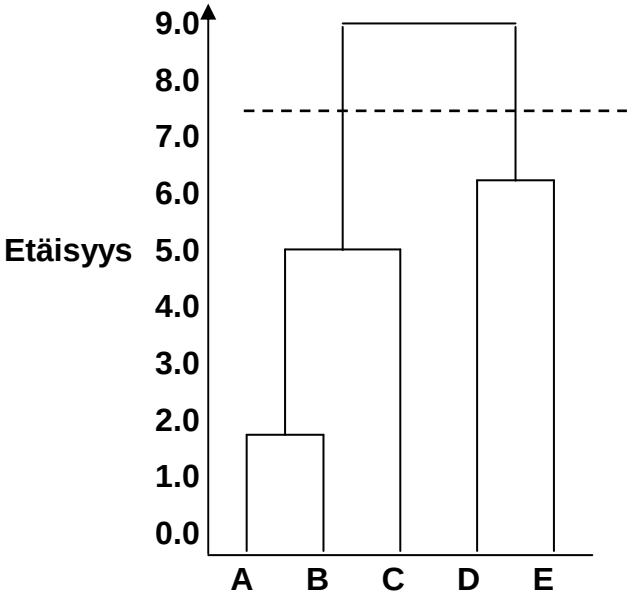
$$D_{(A,B,C)D} = 1/3(D_{AD} + D_{BD} + D_{CD}) = 1/3(10+9+7) = 9.33,$$
$$D_{(A,B,C)E} = 1/3(D_{AE} + D_{BE} + D_{CE}) = 1/3(9+8+11) = 9.33$$

Ja nyt uusi etäisyysmatriisi (kuva 7) näyttää seuraavalta:

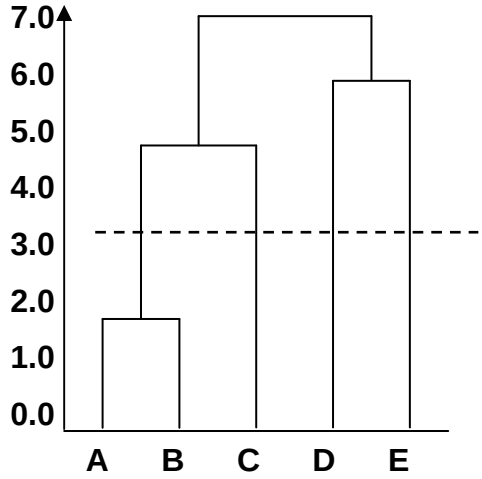
	A,B,C	D	E
A,B,C	0.0	9.33	9.33
D	9.33	0.0	6.0
E	9.33	6.0	0.0

Kuva 7: Etäisyysmatriisi D={(A,B,C),D,E}

Huomataan että etäisyys D_{DE} on kaikkein lyhin, joten D ja E muodostavat seuraavan klusterin ja kun lopuksi vielä yhdistetään klusterit (A,B,C) ja (D,E), havaitaan että tuotettu hierarkia on täsmälleen sama kuin minkä lähin naapuri klusterointimenetelmä tuotti. Toisaalta menetelmät toimivat eri tavoin, joten tietyissä tilanteissa on täysin mahdollista että samasta joukosta klusteroimalla eri etäisyysmatriisin laskenta kaavoja käyttäen, saadaan tuotettua erilainen hierarkia. Saadusta hierarkiasta piirretty dendrogrammi on esitetty vasemman puoleisessa kuvassa (kuva 8) ja vertailun vuoksi oikealla puolella on esitetty vielä lähin naapuri -menetelmällä saatu hierarkia (kuva 9).



Kuva 8: Dendrogrammi naapurikeskiarvo



Kuva 9: Dendrogrammi lähin naapuri

Vertailemalla luotuja dedrogrammeja (kuvat 8 ja 9) huomataan helposti että vaikka kummankin menetelmän tuottama klusterointi on pinnallisesti tarkasteltuna samanlainen, on hierarkiaa esittävissä dendrogrammeissa silti selviä eroja. Koska hierarkkinen klusterointi ei itse asiassa tuota valmiita klustereita, vaan yhden ainoan klusterin, joka sisältää kaikki alkiot, voivat juuri nämä eroavaisuudet olla merkittäviä kun hierarkiasta etsitään todellisia klustereita.

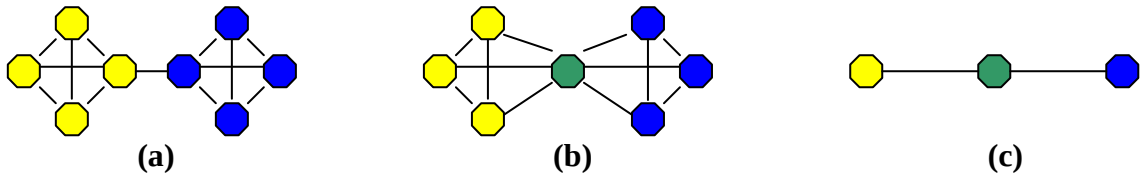
Yleisin tapa erottaa klusterit hierarkiasta on luonnollisesti etäisyyden perusteella, mutta muitakin tapoja voi olla [KHK99]. Etäisyyden perusteella toimittaessa tutkitaan niiden välien kokoja, jotka esiintyvät puussa liitettyjen alkioiden välillä. Jos väli on tarpeeksi suuri, voidaan puun oksa katkaista tämän välin kohdalta omaksi klusterikseen. Dendrogrammeja esittäviin

kuviin piirretyt katkoviivat edustavat tällaisia mahdollisia katkaisukohtia. Huomataan että vaikka klusterointi tapahtuu samassa järjestyksessä ja puut ovat muodoltaan samanlaiset, eivät lopulliset klusteroinnit kuitenkaan ole ekvivalentteja keskenään. Tämä on yleistä kaikkien erilaisten hierarkkisten menetelmien välillä [JMF99].

3.3 Tunnettuja Ongelmia

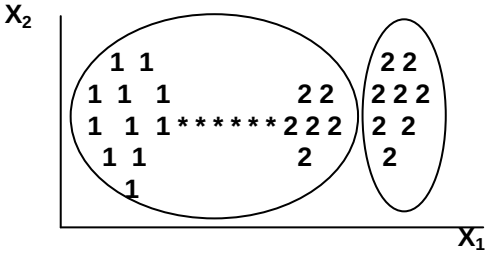
Kaikissa hierarkkisissa klusterointi menetelmissä ongelmaksi muodostuu menetelmän ahneus ja luotavan hierarkian staattisuus, eli kertaalleen tiettyyn klusteriin määrättyjä alkioita ei myöhemmin voi enää siirtää toisiin klustereihin vaan kertaalleen suoritettu klusterointi on peruuttamaton.

Esimerkki tästä löytyy kuvasta seuraavasta kuvasta (kuva 10), jossa kohdassa (a) ollaan alku tilanteessa. Silmämääräisesti voidaan helposti huomata että alkiot muodostavat 2 klusteria. Oletetaan kuitenkin että etäisyys keskimmäisten alkioiden välillä on lyhyempi kuin muiden alkioiden välillä, jolloin kokoava hierarkkinen klusterointi menetelmä yhdistää ahneesti keskimmäiset alkiot samaan klusteriin kohdan (b) mukaisesti. Lopulta päädytään kohdan (c) mukaiseen optimaaliseen klusterointiin, jossa yhdistettyjen alkioiden etäisyydet toisistaan ovat mahdollisimman pienet, mutta joka kuitenkin on väärä. Edellä esitettyä ongelmaa on pyritty ratkaisemaan monella eri tavalla, esimerkiksi soveltaen hierarkkisen algoritmin tuottamaan klusterointiin jälkeenpäin erilaisia jalostusmenetelmiä, jotka potentiaalisesti pystyisivät löytämään ja siirtämään virheellisesti klustereihin liitetyt alkiot oikeisiin klustereihinsa. Tässä tapauksessa siis halkaisemaan kohdan (c) keskimmäisen klusterin ja sijoittamaan alkiot uudelleen. Suoraan hierarkkiseen klusterointi algoritmeihin liittyen kyseinen ongelma on kuitenkin yhä ratkaisematta [HKK99].

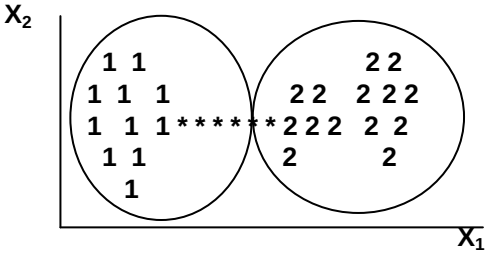


Kuva 10: Esimerkki ahneesta hierarkkisesta klusteroinnista

Monissa kokoavissa hierarkkisissa klusterointimenetelmissä, kuten lähin naapuri, ongelmana on menetelmien taipumus liittää yhteen klustereita ulkoisten klustereihin kuulumattomien häiriöpisteiden (noise points) kautta, jotka sijaitsevat klustereiden välillä. Tätä ilmiötä kutsutaan ketjutukseksi [JMF99]. Seurauksena tietyt menetelmät eivät välttämättä pysty löytämään todellisia klustereita kaikissa tilanteissa. Seuraavat kuvat, jotka esittävät lähin naapuri (kuva 11) ja kaukaisin naapuri (kuva 12) -menetelmillä löydettyjä klustereita samasta datasta havainnollistavat tätä ongelmaa.



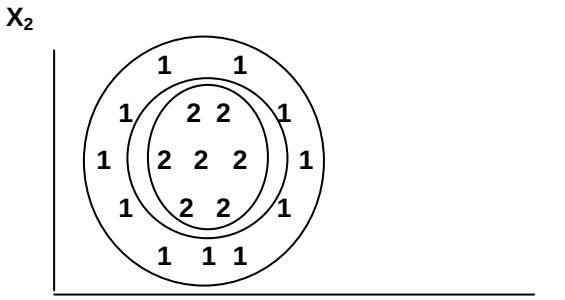
Kuva 11: Lähin naapuri (single-linkage)



Kuva 12: Kaukaisin naapuri (complete-linkage)

Edellisissä kuvissa (kuvat 11 ja 12) '*' edustaa niin sanottuja häiriöpisteitä, jotka eivät varsinaisesti kuulu kumpaankaan klusteriin. Kuitenkin vasemmanpuoleisessa kuvassa, joka kuvaa lähin naapuri -menetelmän toimintaa klusteroitaessa kyseisiä alkioita, yhdistää menetelmä virheellisesti todellisiin klustereihin 1 ja 2 kuuluvia alkioita samaan klusteriin, juuri ketjutukseksi nimetyn ilmiön vuoksi. Samassa tilanteessa kaukaisin naapuri -menetelmä kuitenkin sijoittaa alkiot oikeisiin klustereihin, vaikka häiriö pisteet tulevatkin mukaan klustereihin.

Verrattuna kaukaisin naapuri menetelmään, on lähin naapuri kuitenkin monipuolisempi. Esimerkiksi jos klusteroitavan on kaksi sisäkkäin sijoittunutta klusteria, pystyy lähin naapuri menetelmä löytämään nämä klusterit, mutta kaukaisin naapuri -menetelmä puolestaan ei [JMF99]. Ongelma on esitetty seuraavassa kuvassa (kuva 13), joka esittää lähin naapuri -menetelmällä löydettyjä sisäkkäisiä klustereita, joita kaukaisin naapuri -menetelmällä ei pystyttäisi löytämään.



Kuva 13: Lähin naapuri (Single-Linkage)

Jokainen menetelmä sisältää omat hyvät ja huonot puolensa, mutta erikoisen tärkeäksi valittaessa sopivaa hierarkkista klusterointi menetelmää ja erityisesti kaavaa etäisyysmatriisin laskentaan nousee tieto siitä, minkä tyyppistä tietoa ollaan klusteroimassa.

4 Johtopäätöksiä

4.1 Hyvät ja huonot puolet

Hierarkkisella klusteroinnilla on monia hyviä puolia, kuten varsinkin kokoavien menetelmien yksinkertaisuus. Muina hyvinä puolina kannatta huomioida että klusteroinnin hierarkia pystytään kuvaamaan helposti visuaalisessa muodossa ja että menetelmän edetessä suoritettavien operaatioiden määrä pienenee jatkuvasti. Kuitenkin etäisyysmatriisin laskeminen, varsinkin käytettäessä hiemankin monimutkaisempia kaavoja sen laskentaan, saattaa olla hyvinkin raskasta. Tämä puolestaan heikentää hierarkkisten menetelmien skaalautuvuutta suurille tietojoukoille [SLH01, KHK99].

Hierarkkiset klusterointi menetelmät ovat myös luonteeltaan ahneita siten, että vaikka ne tuottavatkin klusterien välisten ja sisäisten etäisyyksien perusteella mitattuna optimaalisen klusteroinnin, se ei välttämättä ole oikea klusterointi kyseisille alkioille. Tähän liittyen hyvänä puolena mainittakoon että hierarkkisen klusteroinnin teoria puoli on hyvin tunnettu ja menetelmien ongelmat ja soveltuvuudet hyvin tiedossa, joten sopivan menetelmän valitseminen oikeanlaisen datan klusterointiin saattaa hyvinkin tuottaa parhaan mahdollisen klusteroinnin [JMF99].

Viitteet

[JMF99] A.K. Jain, M.N. Murty, P.J. Flynn. *Data Clustering: A Review*. ACM Computing Surveys, Vol. 31, No. 3, September 1999. [http://www.isi.uu.nl/TGV/jain.pdf]

[SLH01] A. Szymkowiak, J. Larsen and L.K. Hansen. *Hierarchical Clustering for Datamining*, Proc. 5th Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies KES'2001, Osaka and Nara, Japan, 6--8 Sept., 2001. [http://citeseer.nj.nec.com/szymkowiak01hierarchical.html]

[KHK99] G. Karypis, Eui-Hong Han, V. Kumar. *CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modelling*, IEEE Computer, Vol. 32, No. 8, Aug. 1999. [http://www-users.itlabs.umn.edu/~karypis/publications/Papers/PDF/chameleon.pdf]

[HKK99] E.H. Han, G. Karypis and V. Kumar. *Multilevel refinement for hierarchical clustering*, Technical Report TR-99-020, Department of Computer Science, University of Minnesota, Minneapolis, 1999. [http://www-users.cs.umn.edu/~karypis/publications/Papers/PDF/clrefine.pdf]

[He99] Qin He. *A Review of Clustering Algorithms as Applied in IR*, UIUCLIS--1999/6+IRG [http://alexia.lis.uiuc.edu/research/irg/uiuclis--1999-6+irg.pdf]