Spatiaalinen klusterointi

Miika Nurminen (minurmin@cc.jyu.fi)

9. joulukuuta 2003

Tiivistelmä

Seminaarityössä käsitellään spatiaalisia klusterointimenetelmiä osana tiedonlouhintaa. Menetelmien päätyypit esitellään ja kustakin käydään tarkemmin läpi tyypillinen esimerkkialgoritmi. Käsiteltävät algoritmit ovat CLARANS, AMOEBA, DBSCAN ja STING.

1 Johdanto

Suurten tietomassojen käsittely ja analysointi on tietokantojen, WWW:n ja satelliittijärjestelmien takia yhä haastavampaa. Miljoonia tietueita sisältävien tietovarastojen hahmotus ei ole manuaalisin keinoin käytännöllistä tai edes mahdollista. Myös paikkatiedon määrä ja saatavuus on lisääntynyt muun tiedon mukana mm. satelliittipaikannuksen, karttapalvelujen ja muiden paikkatietojärjestelmien ansiosta.

Tiedonlouhinta (tiedonrikastus) on suurten tietojoukkojen (tietokantojen, mittaustiedon, paikkatiedon...) analyysia ja mallien muodostusta. Klusterointi on havainnollinen tapa tiivistää tietoa ja etsiä havaintojoukosta keskenään samanlaisia alkioita. Seminaarityössä tutkitaan paikkatiedon louhintaan soveltuvia klusterointimenetelmiä.

Seminaarityö jakautuu seuraavasti: luvussa 2 käsitellään tiedon louhintaa yleisesti ja spatiaalista klusterointia tarkemmin sovelluksineen. Luvussa 3 esitellään ja arvioidaan esimerkkialgotitmit. 4. luku on yhteenveto.

2 Klusterointi tiedonlouhinnan osana

Tiedonlouhinta (*Data Mining*) on suurten, tiettyä tarkoitusta varten kerättyjen tietojoukkojen analyysia, jonka tarkoituksena on löytää odottamattomia suhteita ja tiivistää dataa uusilla tavoilla, jotka ovat sekä ymmärrettäviä että käyttökelpoisia

[7, sivut 1-4]. Tiedonlouhinta on tieteidenvälistä toimintaa, joka käsittää joukon erilaisia menetelmiä ja algoritmeja. Tiedonlouhinnan lähitieteitä ovat tilastotiede, tietokannat, koneoppiminen, hahmontunnistus, tekoäly ja visualisointi. Muista tieteistä tiedonlouhinnan erottaa keskittyminen suurten tietomassojen käsittelyyn ja tavoite luoda datasta tietämystä.

Klusterointi on tiedonlouhinnan perusmenetelmä, joka ryhmittelee havaintoaineiston lähellä toisiaan olevista alkioista koostuviin klustereihin (ryppäisiin). Tiedonlouhinnassa klustereiden ajatellaan kuvaavan tietojoukossa piilossa olevia hahmoja, yleisemmin kyse on tiheysjakauman estimoinnista. Koneoppimisen näkökulmasta klusterointi kuuluu ohjaamattoman (*unsupervised*) oppimisen menetelmiin, tilastotieteen kannalta se on monimuuttujamenetelmiin kuuluvaa ryhmittelyanalyysia. [6]

Intuitiivisesti klusterin käsite on selkeä, mutta sen formaali määrittely on hankalaa. Havaintoaineistosta muodostetut klusterit eivät ole välttämättä yksikäsitteisiä, vaan voivat riippua esim. datapisteiden käsittelyjärjestyksestä. Lisäksi eri algoritmeilla muodostetut klusterit poikkeavat toisistaan muodoltaan, määrältään ja tarkkuudellaan. Estivill-Castron [4] mukaan klusterointialgoritmien moninaisuus johtuu niiden taustalla olevista erilaisista matemaattisista malleista. Lisäksi vaatimukset klustereille vaihtelevat sovellusaloittain. Yleensä hyvin klusteroidussa datajoukossa yksittäisessä klusterissa olevat havainnot ovat keskenään samanlaisia, mutta eri klusterit poikkeavat toisistaan mahdollisimman paljon [6].

2.1 Paikkatiedon louhinta

Paikkatiedon louhinta on implisiittisen tiedon, paikkasuhteiden ja muiden piilossa olevien hahmojen erottelua paikkatietokannasta [9]. Tässä tietokanta tarkoittaa mitä tahansa tietojoukkoa, joka sisältää sijaintitietoa. Paikkatiedossa olioihin liittyy tavanomaisen ominaisuustiedon lisäksi tietoa paikasta, topologiasta ja etäisyyksistä. Spatiaalinen riippuvuus aiheuttaa lisävaatimuksia laskennalle: lähellä toisiaan olevat havainnot vaikuttavat toisiinsa, joten tutkittaessa tiettyä alkiota on myös sen naapurit otettava huomioon [2].

Klusterointi soveltuu paikkatiedon louhintaan, koska taustatietoa datasta ei tarvita. Tämä on etu verrattuna vanhempiin paikkatiedon analyysimenetelmiin, joissa käyttäjän täytyi määritellä datajoukosta riippuva spatiaalisia suhteita kuvaava käsitehierarkia ennen käsittelyä. Klusteroinnin kohteena voi olla sijaintitieto [5] tai ominaisuustieto [6]. Myös lähestymistapojen yhdistäminen on mahdollista. Paikallisuutta painottava (*SD, spatially dominant*) menetelmä on hakea aluksi tietokannasta joukko (ominaisuustiedon perusteella) kiinnostavia havaintopisteitä ja sitten klusteroida ne koordinaattien perusteella. [10]

Klusterointia hyödynnetään esim. maankäytöltään tai sääolosuhteiltaan samanlaisten alueiden etsinnässä [6], maanjäristysten jäljityksessä [2] tai rikosalueiden

kartoituksessa [5]. Klusteroinnin tuloksena on tiivis esitys datasta, jota voidaan käyttää esim. teemakartan luonnissa. Klustereiden dataa voidaan edelleen jatkokäsitellä muilla tiedonlouhinnan menetelmillä, kuten assosiaatiosääntöjen etsinnällä.

2.2 Klusterointimenetelmien päätyypit

Kaikkiin klusterointimenetelmiin (mahdollisesti joitakin ruudustopohjaisia menetelmiä lukuunottamatta) liittyy tavalla tai toisella tietoalkioiden vertaamiseen käytetty mitta, jolla alkioiden samanlaisuus määritellään. Yleisiä mittoja ovat esimerkiksi L_p-etäisyydet

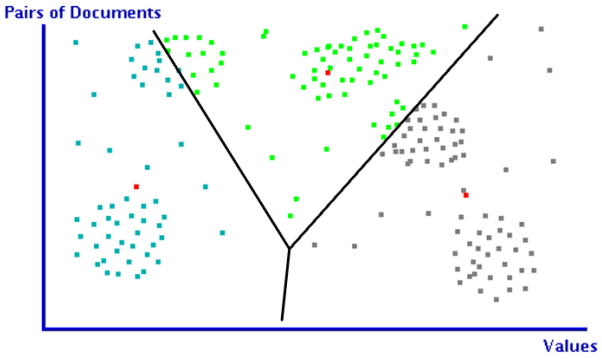
$$d(x, y) = \| x - y \|_p = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p},$$

missä x ja y ovat verrattavien alkioiden piirrevektoreita. Erikoistapauksena L_2-etäisyys on tuttu euklidinen etäisyys. Muista mitoista mainittakoon esim. piirrevektorien välinen kulma samanlaisuuden määrityksessä. Numeeristen vektoreiden lisäksi mittoja on kehitetty myös muille tietotyypeille (esim. luokiteltu muuttuja), mutta tässä rajoitutaan käsittelemään numeerisia muuttujia.

Klusterointialgoritmeja voidaan luokitella monilla eri tavoilla. Berkhin [1], Han *et al.* [6] ja Kolatch [8] esittävät kukin omat jaottelunsa, joista tässä on esitetty kooste spatiaalisen klusteroinnin kannalta oleellisten menetelmien osalta. Menetelmien päätyypit ovat seuraavat:

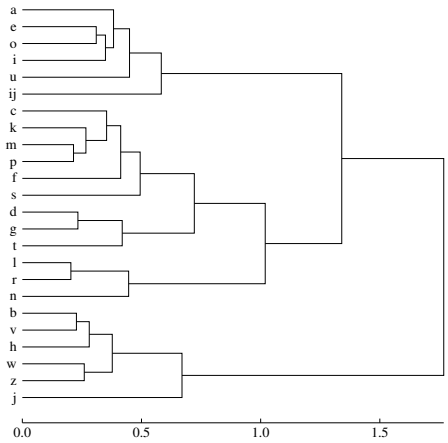
- **Osittavat** menetelmät perustuvat havaintoalkioiden iteratiiviseen jakamiseen. Klusterien paikat muuttuvat algoritmin edetessä, mutta klusterien määrä on ylensä sidottu. Tyypillinen esimerkki osittavasta algoritmista on K-means -algoritmi. Aluksi käyttäjä määrittelee klusterien määrän, joille arvotaan alustavat paikat. Jokainen havainto sijoitetaan kuuluvaksi lähimpään klusteriin, jonka paikkaa korjataan arvojensa keskiarvon perusteella. Kuvassa 1[1] on esitetty klusteroinnin tulos esimerkkidatasta K-means -algoritmilla. Kuvasta nähdään, että klusterointi ei ole onnistunut kunnolla: esimerkiksi vasemman yläkulman ”luontainen” klusteri jakautuu kahden klusterin kesken.
- **Hierarkkiset** menetelmät jakautuvat kokoaviin ja jakaviin menetelmiin. Datasta muodostetaan puurakenne, jonka solmut edustavat klustereita tietyllä tarkkuustasolla. Kokoavissa menetelmissä puu muodostetaan aloittamalla yksittäisistä havaintoalkioista ja yhdistelemällä lähimpiä alkioita, jakavissa

[1] Kuva on luotu Chihoon Leen klusterointiappletilla, URL http://www.cs.sfu.ca/~cleee/personal/clustering/applet.html



Kuva 1: Esimerkki K-means -algoritmin tuloksesta kolmella klusterilla.

menetelmisssä koko havaintoaineisto tulkitaan alussa yhdeksi klusteriksi, jota jaetaan. Hierarkkisten menetelmien etuna on mahdollisuus tarkastella havaintoaineistoa monella tarkkuustasolla, toisaalta klusterien ”luonnollisen” määrän selvittäminen voi olla hankalaa. Hierarkkisen klusteroinnin tuloksia havainnollistetaan usein dendrogrammilla, josta on esimerkki kuvassa 2[2].



Kuva 2: Dendrogrammi.

- **Tiheyteen** (*density*) perustuvat menetelmät perustuvat ajatukseen, että klusteroituneiden pisteiden tiheys on ympäröivien pisteiden tiheyttä suurempi. Tällöin lähellä toisiaan olevat pisteet luokitellaan samaan klusteriin. Tiheyteen perustuvat algoritmit soveltuvat mielivaltaisen muotoisten klustereiden etsintään ja kestävät hyvin kohinaa (poikkeamia). Klusterit voidaan muodostaa yksittäisten pisteiden paikallisten ominaisuuksien perusteella.

[2]Kuva on luotu Peter Kleinwegin *den*-sovelluksella, URL http://odur.let.rug.nl/~kleiweg/clustering/clustering.html

- **Ruudustoon** (*grid*) perustuvissa menetelmissä on päinvastainen lähestymistapa hierarkkisiin ja osittaviin menetelmiin verrattuna. Kun edellisissä menetelmissä keskityttin klustereiden muodostamiseen havaintopisteiden perusteella, ruudustoon perustuvissa menetelmissä lähdetään havaintoavaruudesta ja sen osittamisesta. Klusterit määritetään tarkimman havaintoavaruuden osituksen ja ”valittujen ruutujen” perusteella. Menetelmät eivät tällöin ole riippuvaisia havaintoalkioiden valinnan järjestyksestä, mutta klusteroinnin tarkkuus riippuu osituksen tarkkuudesta.

Edellä kuvattu jako ei ole kattava eikä täysin täsmällinen. Monet uudemmat menetelmät yhdistelevät eri perustyyppejä (esim. aluksi havaintoavaruuden ositus ruudustomenetelmällä ja klusterointi tiheysmenetelmällä). Tilastollisten menetelmien lisäksi klusterointiin on sovellettu myös laskennallisesti älykkäitä tekniikoita (geneettiset algoritmit, itseorganisoitavat kartat) ja signaalinkäsittelytekniikoita (aallokeanalyysi). Laajojen tietokantojen, erilaisten tietotyyppien ja korkeadimensioisen datan käsittelyyn on kehitetty erikoismenetelmiä. Paikkatiedon käsittelyn kannalta tärkeä ryhmä ovat rajoitepohjaiset menetelmät, joista esimerkkinä on esteiden määrittely havaintoalueelle (esim. järvet ja joet klusteroitessa maalla olevaa dataa). Kaikenkaikkiaan klusterointialgoritmien kirjo on hyvin laaja. [1]

3 Spatiaalisia klusterointimenetelmiä

Luvussa käydään läpi eri algoritmeja, joista kukin edustaa tiettyä klusteroinnin päätyyppiä. Suunnittelijoidensa mukaan menetelmät soveltuvat erityisesti paikkatiedon klusterointiin. CLARANS on osittava menetelmä, AMOEBA hierarkkinen, DBSCAN perustuu alkioiden tiheyteen, STING on ruudustomenetelmä. Algoritmit on valittu selkeyttä ja havainnollisuutta (ei tehokkuutta tai parasta mahdollista tarkkuutta) silmälläpitäen. Klusterointialgoritmien laatua voidaan arvioida seuraavien yleisten vaatimusten [8] pohjalta:

1. **Tehokkuus** ja **skaalautuvuus**. Tiedonlouhinnassa käsitellään suuria tietomassoja (esim. miljoonia tietueita), joten tämä on ilmeinen vaatimus.
2. **Klustereiden muoto**. Paikkatietoaineiston klusterit voivat olla mielivaltaisen muotoisia tai sisäkkäisiä.
3. **Robustius**. Havaintoaineisto voi sisältää poikkeamia (*outliers*), jotka vääristävät klustereita.
4. **Havaintojen järjestys**. Algoritmin tulos ei saisi riippua syötealkioiden järjestyksestä.
5. **Riippumattomuus taustatiedosta**. Käyttäjältä ei pidä vaatia sovellusaluekohtaista tietoa (esim. klustereiden määrää tai etäisyystietoa).

6. **Moniulotteisuuden käsittely**. Ominaisuustietoa klusteroidessa ulottuvuuksien (muuttujien) määrä voi kasvaa suureksi (esim. kymmeniä ulottuvuuksia), mikä vaikuttaa klusterointialgoritmin tehoon.

Useimmat klusterointialgoritmit eivät täytä kaikkia laatuvaatimuksia. Esimerkiksi osittavien algoritmien tulokset riippuvat yleensä alkioiden käsittelyjärjestyksestä. Useimmat algoritmit vaativat taustatietoja sovellusalueesta parametrisoinnin muodossa. Erityisesti vanhemmissa menetelmissä ei ole otettu huomioon moniulotteisuuden vaatimuksia.

Koska ei ole olemassa kaikille sovellusalueille soveltuvaa yleismenetelmää, algoritmin valinnassa on otettava huomioon sovellusalue ja datan luonne. Esimerkiksi yleisesti tehottomana pidetty K-means saattaa soveltua hyvin tilanteeseen, jossa kauppaketju suunnittelee kauppojen sijoittelua väestöjakauman mukaan [6]. K-meansin käyttöä tulee se, että klusterien (perustettavien supermarketien) määrä on etukäteen tiedossa ja havaintoalkioiden (asiakkaiden) tulee olla mahdollisimman lähellä klusterien keskuksia.

3.1 CLARANS

CLARANS (*Clustering Large Applications based on RANdomized Search*) oli ensimmäisiä erityisesti paikkatiedon klusterointiin kehitettyjä algoritmeja. Algoritmi muistuttaa K-meansia siinä mielessä, että jokaista klusteria edustaa keskuspiste. K-meansista poiketen keskuspiste kuuluu havaintoaineistoon ja valitaan siten, että piste on lähellä klusteriehdokkaan mediaania. Lähestymistapaa kutsutaan medoidimenetelmäksi ja sen etuna K-meansiin on robustius. Poikkeamapisteet vaikuttavat mediaaniin huomattavasti keskiarvoa vähemmän.

CLARANS tukee laajojen tietomassojen käyttöä näytteistämisellä. Havaintojoukosta poimitaan määrätyn kokoinen näyte, jonka perusteella klusterit muodostetaan. Klusteroinnin tarkkuuden parantamiseksi näytejoukkoa päivitetään jokaisella iteraatiokierroksella. Algoritmin parametrit ovat klustereiden määrä ja näytekoko. [10]

Algoritmin pahin puute on sen tehottomuus. Medoidin laskennan takia algoritmin kompleksisuus on kertaluokkaa $\Theta(n^2)$, jonka takia algoritki soveltuu huonosti suurille datajoukoille. Algoritmin löytämät klusterit ovat muodoltaan aina konvekseja. Hyviä puolia ovat riippumattomuus datapisteiden järjestyksestä ja melko hyvä poikkeamien sieto [8]. Huomautettakoon, että CLARANSin jälkeen on kehitetty tarkempia ja tehokkaampia osittamiseen perustavia algoritmeja, mutta tämä otettiin käsittelyyn sen tunnettuvuuden vuoksi.

3.2 AMOEBA

AMOEBA on hierarkkinen menetelmä spatiaaliseen klusterointiin. Vanhemmista hierarkkisista menetelmistä poiketen havaintojoukon jako ei perustu pelkästään pisteiden välisiin etäisyyksiin, vaan niiden Delaunay-kolmiointiin. Lähestymistapa poikkeaa useimmista muista klusterointimenetelmistä, mikä tekee algoritmista kiinnostavan. Algoritmin tulos on kuitenkin tyypilliseen hierarkkisten menetelmien tapaan puu, jonka solmuja klusterit ovat.

Algoritmi etenee seuraavasti: aluksi havaintoaineisto muunnetaan Delaunay-graafiksi, tämän jälkeen algoritmi etenee karsimalla graafista liian pitkät kaaret. Solmut, joille ei jää yhtään kaarta poistetaan graafista poikkeamina, muut yhtenäiset aligraafit merkitään klustereiksi, ja algoritmia sovelletaan niihin rekursiivisesti. Kynnysarvo solmua p sivuavien kaarten pituudelle määritetään seuraavalla kaavalla: [5]

$$\mu_g + \frac{\sigma}{\mu(p)/\mu_g},$$

missä μ_g on graafin kaarien pituuden keskiarvo, σ kaarien pituuden keskihajonta ja $\mu(p)$ solmua p sivuavien kaarten pituuden keskiarvo.

AMOEBA-algoritmilla on monia toivottavia ominaisuuksia: Delaunay-kolmioinnin muodostamiseen menevän ajan (ja samalla algoritmin) laskennallinen vaativuus on kertaluokkaa $\Theta(n\ log(n))$. Algoritmi löytää lähes mielivaltaisen muotoisia klustereita, ei vaadi parametreja, on riippumaton alkioiden käsittelyjärjestyksestä ja kestää hyvin poikkeamia. Algoritmin merkittävin puute on, että se on Delaunay-kolmioinnista johtuen suunniteltu 2-ulotteista dataa silmälläpitäen. [8]

3.3 DBSCAN

DBSCAN-algoritmi (*Density Based Spatial Clustering of Applications with Noise*) käy kertaalleen kaikki havaintopisteet läpi ja luokittelee ne ydinpisteisiin, reunapisteisiin tai kohinaksi (poikkeama). Klusterit muodostuvat ydin- ja reunapisteistä. Käsiteltävälle pisteelle haetaan ϵ-parametrin säteellä oleva naapurusto, jota kasvatetaan iteratiivisesti niin kauan, kuin ympäristöstä löytyy $minp$-parametrin verran lähipisteitä. Ydinpisteillä on aina vähintään $minp$ lähipistettä, reunapisteet kuuluvat jonkin ydinpisteen naapurustoon, mutta niillä on vähemmän reunapisteitä. Loput pisteet ovat kohinaa. Naapurien haut voidaan toteuttaa käyttämällä R*-puuta, joka on spatiaalinen indeksointirakenne. [3]

DBSCAN oli varhaisimpia havaintoaineiston tiheyteen perustuvia menetelmiä. Algoritmin merkittävin parannus aikaisempiin on läheisyyden käyttö klusteroinnissa: piste voidaan luokitella kuuluvaksi klusteriin naapureidensa perusteella, mikä mahdollistaa havaintojoukon klusteroinnin yhdellä datan läpikäynnillä, siis

$\Theta(n)$-kompleksisuuden. (R*-puusta tehtävien hakujen takia DBSCAN-algoritmin kompleksisuus on tosin kertaluokkaa $\Theta(n\ log(n))$, mutta sittemmin on kehitetty tehokkaampia algoritmeja).

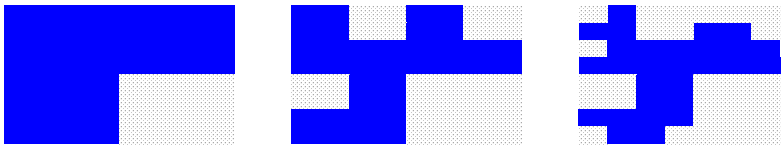
Algoritmin vahvuuksia ovat kohinan sieto, kohtuullinen suorituskyky, riippumattomuus havaintojen järjestyksestä ja mielivaltaisen muotoisten klusterien löytäminen. Heikkoutena algoritmi vaatii käyttäjältä kaksi parametria, joilla voi olla merkittävä vaikutus klusterointiin. Tällöin algoritmia voi joutua ajamaan monta kertaa sopivimpien parametrien löytämiseksi. Monimutkaisella datajoukolla ongelmia voi tulla myös siitä, että parametrit ovat globaaleja. Tällöin tiheydeltään vaihtelevien mutta muotonsa puolesta klusteroituneiden alueiden (esim. hierarkkiset rakenteet) löytäminen voi olla hankalaa. [8]

3.4 STING

STING (*STatistical INformation Grid*) on esimerkki ruudustopohjaisesta menetelmästä puhtaimmillaan. Se ei ole pelkästään klusterointimenetelmä, vaan tavallaan kysely-ympäristö spatiaaliselle datalle. Edellä käsiteltyjen menetelmien tyypillinen käyttötilanne muodostaa klusterit jokaisen kyselyn jälkeen (katso luku 2.1). STING-algoritmin ideana on muodostaa yhdellä käsittelyllä ruudusto, jossa on itsessään ominaisuustiedon koosteita. Tällöin ruudustoa voi käyttää erilaisiin kyselyihin muita klusterointialgoritmeja nopeammin. Suurin osa tarvittavista tiedoista on jo valmiina ruudustossa.

Algoritmin perustana on monitasoinen ruudusto, joka jakaa havaintoavaruuden säännöllisiin alueisiin monella eri tarkkuustasolla. Ylin taso kuvaa aluetta yhtenä kokonaisuutena, alimman tason tarkkuus riippuu havaintoarvojen tiheydestä. Jokaiseen ruuduston soluun liitetään lisätietoina ruudun alueella olevien pisteiden määrä. Lisäksi jokaista sovellusalueen (numeerista) attribuuttia kohden lasketaan keskiarvo, keskihajonta, minimi, maksimi sekä jakauma. Jakaumatyyppi päätellään ennalta määrätystä joukosta (esim. normaalijakauma, eksponenttijakauma, tasajakauma tai tuntematon) χ^2-testin avulla. Tarpeen mukaan myös muita tilastollisia tunnuslukuja voidaan lisätä. Ruudusto muodostetaan alhaalta ylös: aluksi lasketaan tarkat arvot, joiden pohjalta muodostetaan ylemmän tason arvoja.

Algoritmin käyttö perustuu SQL-kieltä muistuttaviin kyselyihin, joilla määritetään haettava muuttuja ja ehtoja, jotka haun pitää täyttää. Ruudustoa käydään rekursiivisesti läpi alkaen ylimmältä tasolta. Soluihin tallennettujen tunnuslukuja jakaumatietojen avulla päätellään, onko solu relevantti kyselyn kannalta. Kysely toistuu ruudukon seuraavalla tasolla relevanteiksi havaituille soluille, loput jätetään tuloksesta pois. Klusterit muodostetaan DBSCAN-algoritmin tapaan yhdistämällä lähekkäin olevat solut. Ruudukon läpikäyntiä ja relevanttien solujen hakua on havainnollistettu kuvassa 3. [11]



Kuva 3: STING-algoritmin aluehaun eteneminen.

STING-algoritmin hyviä puolia ovat riippumattomuus havaintojen käsittelyjärjestyksestä, kohinan sieto (kuten DBSCAN-algoritmilla) ja erityisesti suorituskyky: sekä ruduston muodostus- että hakuvaiheen kompleksisuus on vain $\Theta(n)$. Algoritmi pystyy käsittelemään tehokkasti suuriakin datamääriä, sillä kyselyvaiheessa suorituskyky riippuu vain ruuduston koosta - ei datan. Algoritmin nopeus saavutetaan kuitenkin tarkkuuden kustannuksella. Muodostetut klusterit ovat muodoltaan kulmikkaita ja luonteeltaan approksimaatioita DBSCAN-algoritmin tuloksista. [8]

4 Yhteenveto

Tiedonlouhinta on suurten tietojoukkojen analysointia ja mallien muodostamista. Tiedonlouhintaa voi soveltaa periaatteessa mihin tahansa digitaaliseen tietoon, josta tässä on keskitytty paikkatiedon louhintaan. Paikkatiedossa olioihin liittyy tavanomaisen ominaisuustiedon lisäksi tietoa paikasta, topologiasta ja etäisyyksistä. Spatiaalinen riippuvuus aiheuttaa lisävaatimuksia laskennalle.

Klusterointi on tiedonlouhinnan perusmenetelmä, jonka tarkoitus on ryhmitellä havaintoaineisto lähellä toisiaan olevista alkioista koostuviin klustereihin. Klusterointi soveltuu paikkatiedon louhintaan, koska taustatietoa datasta ei tarvita. Klusteroinnin tuloksena on tiivis esitys datasta, jota voidaan käyttää esim. teemakartan luonnissa.

Menetelmien päätyypit ovat osittavat, hierarkkiset, tiheyteen ja ruudustoon perustuvat menetelmät. Erilaisia klusterointialgoritmeja on tutkittu runsaasti ja uudemmat algoritmit yhdistelevät eri menetelmätyyppejä. Ideoita on haettu myös läheisiltä tutkimusaloilta, kuten laskennallisesti älykkäistä järjestelmistä tai signaalinkäsittelystä. Algoritmeja vertailemalla osoittautuu, että ei ole olemassa yhtä ylivertaista klusterointialgoritmia, vaan sovellusalue ja datan luonne määräävät valittavan algoritmin.

Lähteet

[1] P. Berkhin. Survey of clustering data mining techniques. Tekninen raportti, Accrue Software, 2002. URL: http://citeseer.nj.nec.com/berkhin02survey.html.

[2] M. Ester, H.-P. Kriegel ja J. Sander. Algorithms and applications for spatial data mining. Kirjassa H. J. Miller ja J. Han, toim., *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS*, ss. 160–187. Taylor and Francis, 2001. URL: http://www.dbs.informatik.uni-muenchen.de/Publikationen/Papers/Chapter7.revised.pdf.

[3] M. Ester, H.-P. Kriegel, J. Sander ja X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Kirjassa E. Simoudis, J. Han ja U. Fayyad, toim., *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, ss. 226–231. AAAI Press, 1996. URL: http://www.aaai.org/Press/Proceedings/KDD/1996/kdd96.html.

[4] V. Estivill-Castro. Why so many clustering algorithms: a position paper. *SIGKDD Explor. Newsl.*, 4(1):65–75, 2002. URL: http://doi.acm.org/10.1145/568574.568575.

[5] V. Estivill-Castro ja I. Lee. Amoeba: Hierarchical clustering based on spatial proximity using Delaunay triangulation. Tekninen raportti 99-05, School of Electrical Engineering and Computer Science, The University of Newcastle, Australia, 1999. URL: http://www.cs.newcastle.edu.au/Dept/techrep.html.

[6] J. Han, M. Kamber ja A. K. H. Tung. Spatial clustering methods in data mining: A survey. Kirjassa H. J. Miller ja J. Han, toim., *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS*. Taylor and Francis, 2001. URL: http://www-faculty.cs.uiuc.edu/~hanj/pdf/gkdbk01.pdf.

[7] D. Hand, H. Mannila ja P. Smyth. *Principles of Data Mining*. MIT Press, 2001.

[8] E. Kolatch. Clustering algorithms for spatial databases: A survey. University of Maryland, Department of Computer Science, 2001. URL: http://citeseer.nj.nec.com/436843.html.

[9] K. Koperski, J. Adhikary ja J. Han. Spatial data mining: Progress and challenges. Kirjassa *Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, 1996*, 1996. URL: http://citeseer.nj.nec.com/koperski96spatial.html.

[10] R. T. Ng ja J. Han. Efficient and effective clustering methods for spatial data mining. Kirjassa J. Bocca, M. Jarke ja C. Zaniolo, toim., *20th International*

Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile proceedings, ss. 144–155. Morgan Kaufmann Publishers, 1994. URL: http://citeseer.nj.nec.com/ng94efficient.html.

[11] W. Wang, J. Yang ja R. R. Muntz. Sting: A statistical information grid approach to spatial data mining. Kirjassa M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos ja M. A. Jeusfeld, toim., *Twenty-Third International Conference on Very Large Data Bases*, ss. 186–195. Morgan Kaufmann, 1997. URL: http://citeseer.nj.nec.com/wang97sting.html.