Tampereen yliopisto

Valtteri Pinkkilä

Graafiteoreettisten klusterointimenetelmien soveltaminen sisätilapaikannuksessa

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Matematiikka
Toukokuu 2019

TIIVISTELMÄ

Valtteri Pinkkilä: Graafiteoreettisten klusterointimenetelmien soveltaminen
sisätilapaikannuksessa
Pro gradu -tutkielma
Tampereen yliopisto
Matematiikan ja tilastotieteen tutkinto-ohjelma
Toukokuu 2019

Tässä tutkielmassa käsitellään sisätilapaikannusta ja sovelletaan klusterointialgoritmeja WLAN-signaalien voimakkuuksista muodostuvien sormenjälkien klusterointiin. Työssä esitellään paikannusmenetelmä, jossa sormenjälkien välisiä etäisyyksiä arvioimalla pyritään luomaan tarkasteltavasta alueesta radiokartta. Sormenjälkien välisten etäisyyksien arviointi hankaloituu, kun etäisyydet kasvavat riittävän suuriksi. Näin ollen menetelmä edellyttää sormenjälkien klusterointia pienempiin ryhmiin.

Tutkielmassa käydään läpi klusteroinnin perusperiaatteet ja pääpaino on graafiteoreettisissa klusterointimenetelmissä. Työssä käydään läpi Markov-, k-means- ja affinity propagation -klusterointialgoritmin toimintaperiaatteet. Tämän lisäksi tarkastellaan laajemmin spektrisiä klusterointialgoritmeja. Algoritmeja testataan Tampereen yliopiston Hervannan kampuksen Sähkötalo-rakennuksesta kerätyillä aineistoilla eri samanlaisuus- ja erilaisuusfunktioita käyttäen. Testauksia suoritetaan kaksiulotteisilla aineistoilla, jolloin sormenjälkiä klusteroidaan kerroksittain. Tämän lisäksi algoritmeja testataan aineistoilla, joissa sormenjälkiä on kolmessa eri kerroksessa. Eri algoritmien palauttamia klusterointeja vertaillaan klustereiden yhtenäisyyden perusteella tutkielmassa esiteltävän arviointimenetelmän mukaisesti. Klusteroinnin lisäksi suoritetaan paikannusmenetelmää havainnollistava esimerkki.

Työssä saadut tulokset osoittavat, että yleisesti ottaen yhtenäisimmät RSS-sormenjäljistä muodostetut klusterit saadaan spektrisillä klusterointialgoritmeilla käytettäessä kosini-, Czekanowski- tai Wang-samanlaisuusfunktiota. Klusteroitaessa kaksiulotteista aineistoa saadaan tutkimuksissa kooltaan tasaisempia klustereita kuin kolmiulotteista aineistoa klusteroitaessa. Yhtenäisten klustereiden muodostaminen onnistuu paremmin tiheämmällä mittausaineistolla. Testaukset osoittavat myös, että sormenjälkien kerrosten väliseen erotteluun tarvitaan klusteroinnin lisäksi sensoreista saatavaa informaatiota.

Avainsanat: graafiteoria, klusterointi, sisätilapaikannus, WLAN
Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

Sisältö

Merkinnät

$\mathbb{1}$	vektori, jonka kaikki alkiot ovat ykkösiä
$\mathbb{1}_C$	indikaattori-vektori
$A \in \mathbb{R}^{m \times n}$	$m \times n$ -kokoinen reaalinen matriisi
A^{T}	matriisin A transpoosi
A^{-1}	matriisin A käänteismatriisi
C	klusteri
$\overline{C}$	graafin solmujoukon C komplementti
$\mathbb{C}$	kompleksilukujen joukko
$\deg(v)$	graafin solmun v aste
D	astematriisi
$\overline{e}$	virheiden keskiarvo
E	graafin särmäjoukko
$\Gamma(v)$	graafin solmun v naapurusto
G	graafi
G_A	graafista G johdettu graafi
η	vaimennuskerroin
I	identiteettimatriisi
λ	matriisin ominaisarvo
L	normalisoimaton Laplacen matriisi
L_{rw}	normalisoitu Laplacen matriisi
L_{sym}	normalisoitu Laplacen matriisi
$\mu(C)$	klusterin C keskus
$\mathbb{R}$	reaalilukujen joukko
$\mathbb{R}_+$	positiivisten reaalilukujen joukko
S	samanlaisuusmatriisi
τ_G	graafin G Markovin matriisi
$\mathrm{Tr}(A)$	matriisin A jälki

Ψ	inflaatio-operaatio
V	graafin solmujoukko
$\lvert X \rvert$	joukon X alkioiden lukumäärä
$w(u, v)$	solmujen u ja v välisen särmän paino
$w(G)$	graafin G särmien yhteenlaskettu paino
W	graafin vierusmatriisi
$W(V_1, V_2)$	graafin irrotukseen kuuluvien särmien yhteenlaskettu paino
$\mathbf{x} \in \mathbb{R}^n$	n-ulotteinen vektori
$\mathcal{X}$	datamatriisi

Lyhenteet

AP	Affinity Propagation -klusterointialgoritmi
GNSS	Maailmanlaajuisesti saatavilla oleva satelliittipaikannusjärjestelmä (Global Navigation Satellite System)
GPS	Yhdysvaltain puolustusministeriön ylläpitämä satelliittipaikannusjärjestelmä (Global Positioning System)
IoT	Esineiden internet (Internet of Things)
MAC	Verkkosovittimen ethernet-verkossa yksilöivä osoite (Media Access Control)
MCL	Markov-klusterointialgoritmi (Markov Cluster Algorithm)
RSS	Vastaanotetun signaalin voimakkuus (Received Signal Strength)
SLAM	Paikannusmenetelmä, jonka avulla pyritään muodostamaan samanaikaisesti myös tutkittavan alueen kartta (Simultaneous Localization and Mapping)
WAP	Langattoman tietoliikenneverkon tukiasema (Wireless Access Point)
WLAN	Langaton lähiverkko (Wireless Local Area Network)

1 Johdanto

Paikannuspohjaiset palvelut ovat nykypäivänä kysyttyjä. Jotta käyttäjän reaaliaikaiseen sijaintiin perustuvia sovelluksia voidaan tarjota, käyttäjän sijainti tulee tietää riittävällä tarkkuudella. Sisätiloissa paikannuspohjaisia palveluita voidaan hyödyntää esimerkiksi turvallisuuspalveluissa ja terveydenhuollon valvonnassa, ja niillä voidaan mahdollistaa muun muassa IoT-laitteiden paikantaminen. Siinä missä *Global Positioning System* (GPS) ja muut satelliittipaikannusmenetelmät (*Global Navigation Satellite Systems*, GNSS) kykenevät tarjoamaan riittävän paikannustarkkuuden ulkotiloissa, niin sisätiloissa riittävän tarkan paikannusmenetelmän kehittäminen on edelleen avoin ongelma. Satelliittipaikannuksella ei kyetä saamaan riittävän tarkkaa paikannustulosta sisätiloissa signaalien liiallisesta heikkenemisestä johtuen. [16, 29]

Sisätilapaikannukselle on esitelty useita erilaisia tekniikoita. Monet näistä menetelmistä edellyttävät ylimääräisten lähettimien ja vastaanottimien asentamista. *Wireless Local Area Network* (WLAN) on viime aikoina kasvattanut suosiotaan johtuen langattomien tukiasemien (WAP) kattavuudesta ja nykyisten mobiililaitteiden sekä langattomien vastaanottimien mahdollisuudesta mitata signaalien voimakkuuksia (RSS). Näin ollen kyseinen menetelmä ei edellytä uusia ylimääräisiä lisälaitteita. WLAN on kuitenkin kehitetty langattomaksi verkoksi eikä paikannustarkoitukseen, jolloin syntyy monia haasteita luotaessa sen avulla sisätilapaikannusmenetelmää. Erilaisten WLAN-paikannusmenetelmien joukossa WLAN-sormenjälkipaikannus on saanut suosiota johtuen menetelmän hyvistä tuloksista. [16]

Tässä työssä esitellään sisätilapaikannusmenetelmä, joka perustuu sormenjälkien välisten etäisyyksiä arvioimiseen. Etäisyyksien arvioiminen hankaloituu, kun sormenjälkien yhteisten tukiasemien lukumäärä pienenee. Näin ollen menetelmä vaatii sormenjälkien ryhmittelyä pienempiin osajoukkoihin, joiden sisällä etäisyyksien arviointi on mahdollista. Tämän tutkielman tarkoituksena on vertailla erilaisia klusterointimenetelmiä, joiden avulla sormenjälkiaineiston alkiot voidaan jakaa mahdollisimman yhtenäisiin osajoukkoihin. Tutkielmaa varten on mitattu kaksi eri RSS-sormenjäljistä koostuvaa aineistoa yhdestä kolmikerroksisesta rakennuksesta. Testaukset on jaettu kahteen eri kategoriaan. Ensimmäisessä vaiheessa aineistot jaetaan osiin kerroksittain ja klusterointialgoritmeja testataan kaksiulotteisilla mittausdatoilla. Toisessa vaiheessa klusteroidaan koko aineistoja, jolloin klusteroitava data sisältää sormenjälkiä useista kerroksista. Algoritmien testausten lisäksi työssä esitetään koko paikannusmenetelmää havainnollistava esimerkki. Tämän tarkemmin ei kuitenkaan syvennytä sormenjälkien välisten etäisyyksien arviointiin, vaan tutkielman pääpaino on klusterointialgoritmien vertailussa.

Tutkielman luvussa 2 käydään läpi työssä tarvittavat graafiteoreettiset termit ja määritelmät. Lisäksi luvussa määritellään spektrisessä graafiteoriassa tarvittavia lineaarialgebran käsitteitä. Klusteroinnin ja graafiteoreettisen klusteroinnin perusperiaatteisiin paneudutaan tutkielman luvussa 3. Luvun lopussa esitetään klusteroinnin kannalta tärkeitä samanlaisuus- ja etäisyysfunktioita ja tarkastellaan erilaisia samanlaisuusgraafeja.

Luvussa 4 esitellään neljä erilaista klusterointimenetelmää. Ensimmäisenä tarkastellaan satunnaiskulkua graafissa simuloivaa Markov-klusterointialgoritmia. Markov-algoritmin jälkeen tarkastellaan tunnettua k-means-klusterointialgoritmia. Kolmas työssä esiteltävä algoritmi on affinity propagation -klusterointialgoritmi. Affinity propagation -algoritmin jälkeen paneudutaan spektrisiin klusterointimenetelmiin hieman laajemmin lähtien liikkeelle graafien Laplacen matriisien määrittelystä, ja osoitetaan joitakin klusteroinnin kannalta merkittäviä Laplacen matriisien ominaisuuksia. Tämän jälkeen esitetään kolme eri spektristä klusterointialgoritmia. Luvun lopuksi todistetaan, kuinka spektrisillä algoritmeilla voidaan approksimoida graafien erilaisten irrotusten minimointia.

Tutkielman luvussa 5 käydään läpi tyypilliset WLAN-sisätilapaikannusmenetelmät ja tarkastellaan, kuinka RSS-sormenjälkien klusterointia on aiemmin hyödynnetty sisätilapaikannuksessa. Luvussa esitellään idea uudenlaisesta paikannusmenetelmästä ja klusteroinnin roolista tässä menetelmässä. Lisäksi käydään läpi haasteita, joita ilmenee sormenjälkiä klusteroitaessa. Tämän jälkeen tarkastellaan tutkimuksissa käytettyjä aineistoja ja esitellään menetelmä, jolla algoritmien tuottamia klusterointeja vertaillaan. Lisäksi käydään algoritmikohtaisesti läpi eri testauksissa käytetyt parametrit. Luvun 5 lopussa vertaillaan ja analysoidaan saatuja klusterointituloksia ja tarkastellaan paikannusmenetelmää havainnollistavaa esimerkkiä. Luvussa 6 tehdään yhteenveto ja pohditaan jatkotutkimuksen kannalta mielenkiintoisia seikkoja, kuten sensoreista saatavan informaation lisäämistä klusteroitaviin sormenjälkiin.

2 Graafiteoriaa

Tässä luvussa käydään lyhyesti läpi tutkielman kannalta tarpeelliset graafiteoreettiset määritelmät ja termit. Luvun lopussa esitellään myös luvussa 4 tarvittavia lineaarialgebran käsitteitä. Luvussa on käytetty graafiteorian osalta lähteitä [17, 23, 28] ja lineaarialgebran osalta kirjoja [2, 6, 26] .

Graafi G on pari (V, E), missä $V \neq \emptyset$ on äärellinen joukko alkioita, joita kutsutaan solmuiksi, ja E on äärellinen joukko alkioita, joita kutsutaan särmiksi. Solmujen lukumäärää graafissa $G = (V, E)$ merkitään $n = |V|$. Mikäli joukko E koostuu järjestämättömistä pareista $\{u, v\}$, missä $u, v \in V$, niin graafia G kutsutaan *suuntaamattomaksi graafiksi*. Mikäli joukko E koostuu järjestetyistä pareista (u, v), graafia G kutsutaan *suunnatuksi graafiksi*. *Painotettu graafi* on graafin yleistys, missä funktio $w : E \to \mathbb{R}$ määrittää jokaiselle graafin särmälle painon. Graafia sanotaan *täydelliseksi*, mikäli sen jokaisen solmuparin välillä on särmä.

Mikäli graafin kahden solmun välillä on särmä, näitä solmuja kutsutaan särmän *päätesolmuiksi*, ja ne ovat toistensa *vierussolmuja*. Jos solmusta on särmä solmuun itseensä, särmää kutsutaan *luupiksi*. Solmun u kaikkien vierussolmujen joukkoa kutsutaan solmun u *naapurustoksi*, ja sitä merkitään $\Gamma(u)$. Graafi $G = (V, E)$ on *kaksijakoinen graafi*, mikäli sen solmujoukko V, voidaan jakaa kahteen osajoukkoon V_1 ja V_2 siten, että särmäjoukon E jokaisen särmän toinen päätesolmu kuuluu joukkoon V_1 ja toinen joukkoon V_2.

Graafin $G = (V, E)$, jossa on n-solmua, *vierusmatriisi* W on $n \times n$-matriisi, jossa

$$w_{ij} = \begin{cases} 1, & \text{jos } \{v_i, v_j\} \in E, \\ 0, & \text{muuten.} \end{cases}$$

Mikäli G on painotettu graafi niin tällöin

$$w_{ij} = \begin{cases} w(v_i, v_j), & \text{jos } \{v_i, v_j\} \in E, \\ 0, & \text{muuten.} \end{cases}$$

Graafin *spektri* on sen vierusmatriisin ominaisarvojen joukko. Solmun v *aste*, $\deg(v)$ kertoo kuinka monen särmän päätesolmuna solmu v on. Tarkasteltaessa painotettua graafia $G = (V, E)$, solmun $u \in V$ aste määritellään niiden särmien painojen summana, joiden päätesolmuna solmu u on

$$\deg(u) = \sum_{i=1}^{n} w(u, v_i),$$

missä $v_i \in V$, kaikilla $i = 1, \ldots, n$ ja $w(u, v_i) = 0$, mikäli $\{u, v_i\} \notin E$.
Graafin $G = (V, E)$ diagonaalinen *astematriisi* on

$$D = \begin{pmatrix} \deg(v_1) & 0 & 0 & \cdots & 0 & 0 \\ 0 & \deg(v_2) & 0 & \cdots & 0 & 0 \\ 0 & 0 & \deg(v_3) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \deg(v_{n-1}) & 0 \\ 0 & 0 & 0 & \cdots & 0 & \deg(v_n) \end{pmatrix}.$$

Jaetaan graafin $G = (V, E)$ solmut kahteen eri luokkaan V_1 ja V_2. Tällöin niiden särmien joukko, joiden toinen päätesolmu kuuluu joukkoon V_1 ja toinen päätesolmu joukkoon V_2, muodostavat graafin G *irrotuksen*. Irrotukseen kuuluvien särmien lukumäärää merkitään merkinnällä

$W(V_1, V_2)$. Mikäli graafi on painotettu, merkinnällä tarkoitetaan särmien yhteenlaskettua painoa, jolloin

$$W(V_1, V_2) = \sum_{v_i \in V_1, v_j \in V_2} w(v_i, v_j).$$

Solmujoukon V osajoukon C *komplementista* $V \setminus C$ käytetään merkintää $\overline{C}$.

Polku graafissa $G = (V, E)$ on äärellinen solmusta alkava ja solmuun päättyvä vuorotteleva jono graafin solmuja ja särmiä $v_0, e_1, v_1, e_2, \ldots, v_{k-1}, e_k, v_k$, missä solmut v_{i-1} ja v_i ovat särmän e_i päätesolmut. Edellä kuvattua polkua kutsutaan solmujen v_0 ja v_k väliseksi poluksi. Polku on *yksinkertainen*, jos yksikään solmu ei esiinny siinä kahta kertaa. Graafin kaksi solmua ovat *yhdistetyt*, mikäli niiden välillä on polku. Graafin sanotaan olevan *yhtenäinen*, jos graafin kaikkien solmuparian välillä on polku. *Sykli* on yksinkertainen polku, joka alkaa ja päättyy samaan solmuun. Mikäli graafissa ei ole yhtään sykliä, se on syklitön. Syklitöntä graafia kutsutaan *metsäksi*, ja yhtenäistä syklitöntä graafia *puuksi*.

Graafi $G' = (V', E')$ on graafin $G = (V, E)$ *aligraafi*, mikäli $V' \subseteq V$ ja $E' \subseteq E$. Graafin *komponentti* on graafin maksimaalinen yhtenäinen aligraafi. Mikäli graafin G aligraafi G' on täydellinen graafi, niin se on graafin G *klikki*. Mikäli E' muodostuu niistä joukon E särmistä, joiden päätesolmut kuuluvat joukkoon V', niin tällöin G' on solmujoukon V' *(solmu)indusoima aligraafi*. Yhtenäinen syklitön aligraafi, joka sisältää graafin kaikki solmut, on graafin *virittävä puu*. Graafin virittävässä puussa on tasan $n - 1$ särmää [28, s. 33-36]. Mikäli graafi on painotettu, niin virittävää puuta, jonka särmien yhteenlaskettu paino on pienin mahdollinen, kutsutaan *pienimmäksi virittäväksi puuksi*.

Määritellään seuraavaksi luvussa 4.1 tarvittavia käsitteitä. Matriisia, jonka koko on $m \times n$ ja jonka alkiot ovat reaalilukuja, merkitään $A \in \mathbb{R}^{m \times n}$. Matriisia, joka kerrottuna itsellään on matriisi itse, kutsutaan *idempotenttiseksi matriisiksi*. Vektoria, jonka kaikki alkiot ovat suurempia tai yhtä suuria kuin nolla kutsutaan *homogeenikseksi*, mikäli kaikki sen nollasta poikkeavat alkiot ovat yhtä suuria. Matriisia, jonka kaikki alkiot ovat suurempia tai yhtä suuria kuin nolla kutsutaan *sarake-homogeeniseksi*, mikäli kaikki sen sarakkeet ovat homogeenisia. Sarake-homogeenista matriisia, joka on idempotenttinen kutsutaan *kaksois-idempotenttiseksi* matriisiksi.

Luvussa 4.4 käsitellään algebrallista graafiteoriaa, joten käydään vielä tämän luvun lopuksi läpi luvussa 4.4 tarvittavia lineaarialgebran käsitteitä. Neliömatriisi on *identiteettimatriisi*, kun sen kaikki diagonaalialkiot ovat ykkösiä ja diagonaalin ulkopuoliset alkiot ovat kaikki nollia. Identiteettimatriisista käytetään merkintää I. Matriisi on *symmetrinen matriisi*, kun se on itsensä transpoosi. Matriisin A jälki on sen diagonaalialkioiden summa, ja siitä käytetään merkintää $\mathrm{Tr}(A)$. Matriisi $A \in \mathbb{R}^{n \times n}$ on *positiivisesti semidefiniitti*, mikäli kaikilla vektoreilla $\mathbf{x} \in \mathbb{R}^n$ pätee $\mathbf{x}^{\mathrm{T}} A\mathbf{x} \geq 0$. Matriisi $Q \in \mathbb{C}^{n \times n}$ on *unitaarinen matriisi*, jos sen kompleksikonjugaatin transpoosi on matriisin käänteismatriisi. Vektorit $\mathbf{x}$ ja $\mathbf{y}$ ovat keskenään *ortogonaalisia*, mikäli niiden välinen pistetulo on nolla.

3 Klusterointianalyysi

Ryhmittelyn eli klusteroinnin tehtävä on jakaa aineiston alkiot homogeenisiin ryhmiin eli klustereihin. Tällöin kaksi datan mielivaltaista alkiota kuuluvat samaan klusteriin, mikäli ne ovat samanlaisempia keskenään kuin kaksi mielivaltaista eri klustereihin kuuluvaa alkiota. Jotta tämän kaltainen alkioiden ryhmittely olisi mahdollista, tulee löytää vastaus kahteen kysymykseen: 1) mikä on sopiva menetelmä alkioiden välisten samanlaisuuksien määrittämiseen ja 2) kuinka näitä alkioiden välisiä samanlaisuuksia tulisi käyttää klusteroinnin muodostamiseksi. Alkioiden välisten samanlaisuuksien määrittämisessä tulee ottaa tarkkaan huomioon tarkasteltava aineisto. Tämän tutkielman aliluvussa 3.3.1 tarkastellaan joitakin samanlaisuus- ja erilaisuusfunktioita. Klusterointia hyödynnetään useilla eri tieteenaloilla ja monenlaisiin eri ongelmiin. Tästä johtuen erilaisten klusterointialgoritmien kirjo on laaja. Sopivan algoritmin valinnassa tulee aineisto ja sen rakenne ottaa huomioon. Tässä tutkielmassa tarkennutaan syvemmin graafiteoreettisiin klusterointimenetelmiin. [31, s. 1-7]

Klusteroinnin tehtävänä on siis löytää aineistosta sen luonnollinen rakenne. Klusterointi eroaa luokittelusta siten, että luokittelussa luokat ovat ennalta määrätyt, kun taas klusteroinnissa luokkia ei ole annettu, vaan tehtävänä on löytää ne aineistosta. Ryhmittelyn lisäksi klusterointia voidaan käyttää esimerkiksi suuren datamäärän redusointiin etsimällä kullekin klusterille sopiva edustaja-alkio tai jotkin klusteria edustavat ominaispiirteet. [31, s. 9-10]

Mikäli tarkasteltava aineisto on täysin tasajakautunut, saadaan mielivaltaisia klusterointituloksia. Näin ollen ennen klusteroinnin aloittamista onkin tarpeellista määrittää selvät kriteerit sille, kuinka klustereiden homogeenisuutta ja/tai klustereiden välisiä keskinäisiä suhteita arvioidaan. Klusterointi ei itsessään ole yksinkertainen prosessi, ja se edellyttää monen eri seikan huomioimista ja tarkastelua. Seuraavassa on lueteltu esimerkkejä klusterointiin liittyvistä käytännön haasteista: sisältääkö aineisto ulkopuolisia arvoja ja miten niitä tulisi käsitellä, kuinka alkioiden väliset samanlaisuudet määritellään, mitä klusterointialgoritmia tulisi käyttää, kuinka moneen klusteriin data jakautuu ja onko saatu klusterointi perusteltu. [31, s. 2-5]

Formalisoidaan seuraavaksi klusterointiongelma, ja siirrytään sen jälkeen tarkastelemaan graafiteoreettista klusterointia yleisellä tasolla. Tämän jälkeen tarkastellaan alkioiden välisten samanlaisuuksien määrittämistä ja esitellään joitakin tutkielman kannalta oleellisia samanlaisuus- ja erilaisuusfunktioita. Luvun lopuksi tarkastellaan erilaisia samanlaisuusgraafeja.

3.1 Klusterointiongelman formalisointi

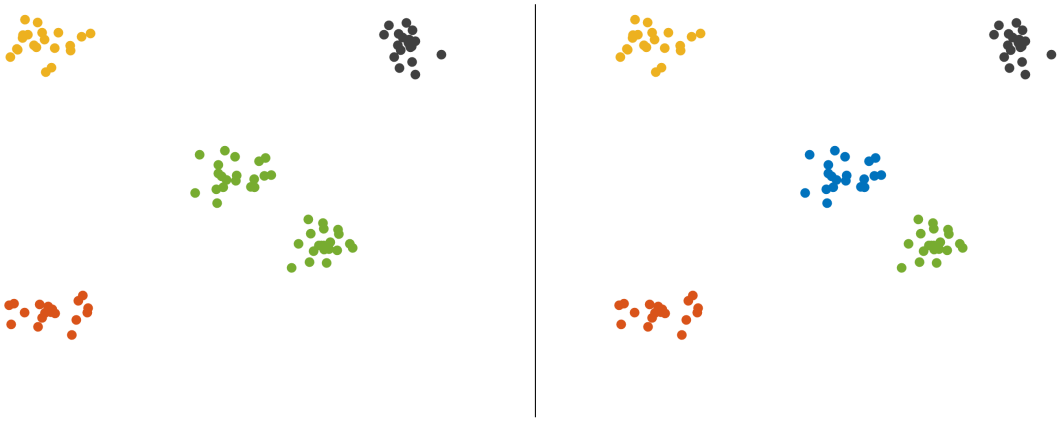
Tyypillisesti klusteroitava aineisto koostuu joukosta alkioita, joita on m kappaletta. Olkoon aineiston eri ominaisuuksien lukumäärä n. Tällöin jokainen datan alkio voidaan esittää n-ulotteisena vektorina $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^{\mathrm{T}}$, missä x_{ij} kuvastaa aineiston alkion i ominaisuutta j. Näitä vektoreita kutsutaan ominaisuusvektoreiksi. Yhdistämällä nämä ominaisuusvektorit matriisiksi aineisto voidaan esitää yhtenä datamatriisina $\mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)^{\mathrm{T}}$, missä matriisin rivi i vastaa alkion i ominaisuusvektoria ja matriisin sarakkeet vastaavat yhtä ominaisuutta. Määrittämällä menetelmä alkioiden välisten samanlaisuuksien/erilaisuuksien laskemiseksi voidaan aineisto esittää $m \times m$ -dimensioisena samanlaisuus-/erilaisuusmatriisina S. Tämän matriisin alkiot s_{ij} kuvaavat alkioparien i ja j välisiä pareittaisia samanlaisuuksia/erilaisuuksia. [31, s. 9-16]

Klusteroinnin päämääränä on jakaa havaintojen joukko X osajoukoiksi $C_1, \ldots, C_k$, joiden lukumäärä k on pienempi kuin havaintojen lukumäärä m. Joukkoja C_i kutsutaan klustereiksi. Tämän kaltaisen jaon tulee täyttää seuraavat kolme ehtoa:

i. Jokaisen klusterin tulisi sisältää vähintään yksi alkio, $C_j \neq \emptyset, j = 1, \ldots, k$.

ii. Jokaisen alkion tulisi kuulua johonkin tiettyyn klusteriin, $\bigcup_{j=1}^{k} C_j = X$.

iii. Jokaisen alkion tulisi kuulua vain yhteen klusteriin, $C_{j_1} \cap C_{j_2} = \emptyset$, missä $j_1 \neq j_2$.



Kuva 3.1: Esimerkki klusteroitavasta aineistosta. Aineisto voidaan klusteroida, joka neljään tai viiteen eri klusteriin riippuen siitä, miten samanlaisuus alkioiden välillä määritellään.

Edelliset kolme ehtoa vastaavat myös joukon X *osituksen* määritelmää [20, s. 63]. Erityisesti silloin, kun $k = m$, jokainen klusteri sisältää vain tasan yhden alkion tarkasteltavasta joukosta. Tämä jako on triviaali, ja näin ollen tulisikin keskittyä sellaisiin jakoihin, missä k on huomattavasti pienempi kuin m. On olemassa myös klusterointialgoritmeja, jotka eivät noudata kaikkia edellä mainittuja kriteereitä. Sumeat klusterointialgoritmit sallivat alkion kuuluvan useampaan kuin vain yhteen klusteriin. Tällöin alkioilla saattaa olla eri kuuluvuusasteita useammassa eri klusterissa. Jotkut klusterointialgoritmit saattavat tunnistaa datasta yksittäisiä etäisiä alkioita, joita algoritmi ei sijoita mihinkään klusteriin, vaan merkitsee alkiot ulkopisteiksi. Näin ollen edellä kuvatut kriteerit eivät aina ole välttämättömiä ehtoja, vaan erilaisia kriteereitä tulee tarkastella tapauskohtaisesti. Kuvassa 3.1 nähdään esimerkkitilanne tyypillisestä klusterointiongelmasta, jossa datan alkiot ovat jakautuneet erillisiin ryhmiin. Yksi klusteri on vasemmalla yläkulmassa, yksi vasemmalla alakulmassa ja yksi ryhmä oikealla yläkulmassa. Tämän lisäksi kuvan keskelle jää alkioita, joista voidaan muodostaa joko yksi tai kaksi klusteria riippuen siitä, miten samanlaisuus alkioiden välillä määritellään. [31, s. 9-16]

3.2 Graafien klusterointi

Monet nykypäivän aineistot ja ongelmat ovat mallinnettavissa graafien avulla [24, s. 319]. Kun klusterointi perustuu alkioiden välisiin samanlaisuuksiin, niin aineisto on luontevaa esittää graafina. Tällöin graafin solmuina toimivat aineiston alkiot ja solmujen välillä kulkee särmä, mikäli solmut ovat keskenään riittävän samanlaisia. Joskus aineistoa on luontevampaa kuvata painotetulla graafilla, jolloin särmien painot kuvaavat alkioiden välisiä samanlaisuuksia tai erilaisuuksia. Aliluvussa 3.3.2 tarkastellaan erilaisia tapoja muodostaa samanlaisuusgraafi. Graafeja klusteroitaessa tavoitteena on jakaa graafien solmut klustereihin niin, että klusterit sisältävät mahdollisimman samanlaisia solmuja ja että eri klustereissa olevat solmut ovat keskenään mahdollisimman erilaisia. [23]

Kuten aiemmin jo mainittiin, ei klustereille voida määrittää tiettyjä kriteereitä, jotka pitäisi aina olla voimassa, vaan niitä tulee tarkastella tapauskohtaisesti. Sama pätee myös graafien

klustereille. Voidaan kuitenkin määritellä joitakin yleisiä graafien klustereille suotuisia ominaisuuksia. Käydään seuraavassa läpi joitakin tällaisista ominaisuuksista. Jokaisen graafin klusterin tulisi olla yhtenäinen eli klusterin sisällä tulisi jokaisen solmuparin välillä olla vähintään yksi tai mahdollisesti useampi polku. Mikäli solmuparin välillä ei ole polkua, näiden solmujen tulisi kuulua eri klustereihin. Lisäksi klusterin solmuja yhdistävien polkujen tulisi kulkea ainoastaan klustereiden sisällä. Toisin sanoen jonkin solmujoukon muodostaman klusterin C tulisi olla itsessään yhtenäinen. Solmujen asteiden avulla voidaan tutkia, kuinka hyvin ne soveltuvat klusterin alkioiksi. Klusterin C yksittäiseen solmuun $v \in C$ liittyvät särmät voidaan jakaa kahteen eri ryhmään: sisäiset särmät ovat särmiä, joiden toisena päätesolmuna on samaan klusteriin C kuuluvia solmuja ja ulkoiset särmät ovat särmiä, jotka yhdistävät solmun v solmuihin, jotka eivät kuulu klusteriin C

$$\begin{aligned} \deg_{\mathrm{s}}(v, C) &= |\Gamma(v) \cap C|, \\ \deg_{\mathrm{u}}(v, C) &= |\Gamma(v) \cap (V \setminus C)|, \\ \deg(v) &= \deg_{\mathrm{s}}(v, C) + \deg_{\mathrm{u}}(v, C). \end{aligned}$$

Mikäli $\deg_u(v, C) = 0$, niin solmu v kuuluu selvästi klusteriin C. Jos taas $\deg_s(v, C) = 0$, niin solmun ei tulisi kuulua klusteriin C, koska sillä ei ole ollenkaan yhteyttä klusterin muihin solmuihin. Klustereiden välisiä irrotuksia tutkimalla voidaan tarkastella, kuinka muusta graafista erotettuja klusterit ovat. Mitä paremmin klusteri eroaa muusta graafista, sitä pienempi/suurempi on irrotuksen painon $W(C, V \setminus C)$ arvo. [23]

Yksi historian ensimmäisistä graafien irrotuksiin perustuvista klusterointimenetelmistä on Zahnin esittelemä menetelmä [35], joka perustuu kahteen eri vaiheeseen. Ensiksi aineistosta muodostetulle graafille, jossa särmät kuvaavat alkioiden välisiä samanlaisuuksia, konstruoidaan maksimaalinen virittävä puu. Toisessa vaiheessa tästä puusta poistetaan särmiä pienimmän painon mukaan. Näin toimimalla saadaan muodostettua joukko yhdistettyjä komponentteja. Vastaavanlaista ideaa käytetään myös tämän tutkielman sovellusosiossa klusterointialgoritmien tulosten vertailuun, mutta klusterointi tarkoituksessa Zahnin kehittämä menetelmä on toimiva vain silloin, kun klusterit ovat selvästi erottuneet toisistaan. Jos graafin solmujen tiheys vaihtelee, menetelmän suorituskyky heikkenee. Lisäksi menetelmä edellyttää toimiakseen, että klustereiden rakenne on etukäteen tiedossa. Klusterointi tarkoituksiin on olemassa kehittyneempiä menetelmiä kuten luvussa 4 esiteltävä spektrinen klusterointimenetelmä. [31, s. 49]

Kaksijakoiset graafit ovat luonnollinen tapa mallintaa tilanteita, mikäli solmut voidaan jakaa kahteen eri luokkaan. Olkoon $G = (A \cup B, E)$ kaksijakoinen graafi, jossa särmät kulkevat ainoastaan joukkojen A ja B välillä. Graafi G voidaan nyt muuntaa kahdeksi graafiksi G_A ja G_B. Tutkitaan kahta solmua u ja v graafissa A. Graafi on kaksijakoinen, jolloin $\Gamma(u) \subseteq B$ ja $\Gamma(v) \subseteq B$. Mitä enemmän joukon A kahdella eri solmulla on yhteisiä naapureita, sitä samanlaisempia solmut ovat. Näin ollen voidaan luoda graafi $G_A = (A, E_A)$ siten, että

$$\{u, v\} \in E_A, \text{ jos ja vain jos } (\Gamma(u) \cap \Gamma(v)) \neq \emptyset.$$

Samoin voidaan muodostaa graafi G_B. Painotettu versio graafille G_A saadaan asettamalla

$$w(u, v) = |\Gamma(u) \cap \Gamma(v)|.$$

Mikäli tarkasteltava kaksijakoinen graafi on itsessään painotettu, voidaan johdetuille graafeille määrittää särmien painot myös samanlaisuus- tai erilaisuusfunktioiden avulla. Klusterointi voidaan suorittaa joko alkuperäiselle graafille G tai johdetuille graafeille G_A tai G_B. [23]

3.3 Mittoja klustereiden löytämiseksi

3.3.1 Etäisyys- ja samanlaisuusfunktiot

Jotta voitaisiin määritellä datan pisteiden välisiä yhteyksiä, siihen tarvitaan samanlaisuuden $s\colon X \times X \mapsto \mathbb{R}$ tai erilaisuuden $d\colon X \times X \mapsto \mathbb{R}$ mitta. Huomattavin ero samanlaisuus- ja erilaisuusfunktioiden välillä on niiden kasvusuunnalla. Käytettäessä erilaisuusfunktiota arvot ovat sitä pienempiä, mitä samanlaisempia arvot ovat. Vastaavasti taas samanlaisuusfunktiota käytettäessä samanlaisuusarvo on sitä suurempi, mitä samanlaisempia tutkittavat alkiot ovat [24, sivu 304]. Silloin kun graafin solmuja vertaillaan samanlaisuusfunktiolla, niin klusterointialgoritmin tulisi sijoittaa samaan klusteriin solmuja, joiden samanlaisuusarvo on suuri. Vertailtaessa solmuja erilaisuusfunktiolla samaan klusteriin tulee alkioita, joiden erilaisuusarvo on mahdollisimman pieni. Erityinen esimerkki erilaisuusmitasta on etäisyys (metriikka) [31, s. 16].

Määritelmä 3.1. Olkoon X joukko. Joukon X metriikka on etäisyysfunktio

$$d\colon X \times X \mapsto \mathbb{R}_+ \cup \{0\},$$

missä kaikille $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$ pätee

1. $d(\mathbf{x}, \mathbf{y}) = 0$, jos ja vain jos $\mathbf{x} = \mathbf{y}$,
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (symmetrisyys),
3. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ (kolmioepäyhtälö).

Samanlaisuusmitta voidaan muuttaa erilaisuusmitaksi ja päinvastoin. Esimerkiksi, jos $d_{\max}$ on kahden pisteen välinen maksimierilaisuus joukossa X, niin tällöin erilaisuus voidaan muuttaa samanlaisuudeksi $s(\mathbf{x}, \mathbf{y}) = d_{\max} - d(\mathbf{x}, \mathbf{y})$. Näin saatu samanlaisuusmitta saavuttaa maksimiarvonsa kun $\mathbf{x} = \mathbf{y}$ (alkio on identtinen itsensä kanssa, $d_{\max} - d(\mathbf{x}, \mathbf{y}) = d_{\max} - 0 = d_{\max}$), ja mitä pienempi on tämän mitan arvo, sitä vähemmän samanlaiset (enemmän erilaiset) vertailtavat pisteet ovat keskenään [31, s. 17]. Kirjallisuudessa käytettyjen erilaisten etäisyys- ja samanlaisuusfunktioiden lukumäärä on erittäin suuri [23]. Esimerkiksi artikkelista [4] löytyy perusteellinen katsaus erilaisiin samanlaisuus- ja erilaisuusfunktioihin. Sopivan etäisyys- tai samanlaisuusfunktion määrittäminen tai löytäminen riippuu tehtävän luonteesta. Joskus sopivan funktion löytäminen saattaa osoittautua itse klusterointiakin haastavammaksi ongelmaksi [23]. Käydään seuraavaksi läpi joitakin, tutkielman sovellusosion kannalta oleellisia, samanlaisuus- ja erilaisuusfunktioita.

Kun kaikki ominaisuudet, joilla datajoukon X pisteitä kuvataan, ovat kvantitatiivisia, niin tällöin jokainen datan piste voidaan esittää n-dimensioisena vektorina $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T$. Tunnetuin erilaisuusmitta on Euklidinen etäisyys

$$d_{\text{Euklidinen}}(\mathbf{x}_i, \mathbf{x}_j) = \left\|\mathbf{x}_i - \mathbf{x}_j\right\| = \sqrt{\sum_{l=1}^{n} \left|x_{il} - x_{jl}\right|^2}.$$

Euklidinen etäisyys kuuluu laajempaan etäisyysfunktioiden perheeseen, joka tunnetaan L_p metriikkana tai normina

$$d_{L_p}(\mathbf{x}_i, \mathbf{x}_j) = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_p = \sqrt[p]{\sum_{l=1}^{n} \left|x_{il} - x_{jl}\right|^p}.$$

Euklidisesta etäisyysfunktiosta saadaan yleisesti käytetty samanlaisuusfunktio, negatiivinen Euklidinen etäisyys

$$s_{\text{Euklidinen}}(\mathbf{x}_i, \mathbf{x}_j) = -\left\|\mathbf{x}_i - \mathbf{x}_j\right\| = -\sqrt{\sum_{l=1}^{n} \left|x_{il} - x_{jl}\right|^2}.$$

Samanlaisuusfunktiona voidaan käyttää myös negatiivista neliöityä Euklidista etäisyyttä. Esitellään seuraavaksi logaritminen Gaussinen samanlaisuusfunktio

$$s_{\text{Gaussinen}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_{il} - x_{jl})^2}{2\sigma^2}\right)\right).$$

Funktion palauttamia arvoja voidaan säädellä parametrin σ avulla. Yksi yleinen menetelmä laskea samanlaisuuksia on laskea vektoreiden välisen kulman kosini. Tästä saadaan kosini-samanlaisuusfunktio

$$s_{\text{kosini}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{l=1}^{n} x_{il} x_{jl}}{\|\mathbf{x}_i\| \left\|\mathbf{x}_j\right\|}.$$

Kosini-samanlaisuusfunktio voidaan helposti muuntaa yleisesti käytössä olevaksi kosini-erilaisuusfunktioksi

$$d_{\text{kosini}}(\mathbf{x}_i, \mathbf{x}_j) = 1 - s_{\text{kosini}}(\mathbf{x}_i, \mathbf{x}_j).$$

Edellä saatu erilaisuusfunktio ei kuitenkaan ole metriikka, sillä funktio palauttaa negatiivisia arvoja, eikä se myöskään toteuta kolmioepäyhtälöä. Samanlaisuuksien laskemiseen voidaan käyttää myös Czekanowski-samanlaisuusfunktiota

$$s_{\text{Czekanowski}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{2\sum_{l=1}^{n} \min(x_{il}, x_{jl})}{\sum_{l=1}^{n} (x_{il} + x_{jl})}.$$

Artikkelissa [30] on esitelty samanlaisuusfunktio, joka painottaa suuria lähekkäisiä arvoja voimakkaammin kuin pieniä. Tässä tutkielmassa funktiosta käytetään Wang-nimeä artikkelin kirjoittajan nimen mukaan.

$$s_{\text{Wang}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{|R|} \sum_{\forall l \in R} \frac{\min(x_{il}, x_{jl})}{\max(x_{il}, x_{jl})},$$

missä $x_{il} \in P$, jos $x_{il} \neq 0$, $x_{jl} \in Q$, jos $x_{jl} \neq 0$ ja $R = P \cup Q$. Mikäli $x_{il} \notin P$, niin $x_{il} = 0$ ja vastaavasti mikäli $x_{jl} \notin Q$, niin $x_{jl} = 0$. [4, 5, 30]

3.3.2 Samanlaisuusgraafit

Kun sopiva funktio datapisteiden välisten suhteiden arvioimiseen on löydetty, aineisto voidaan kuvata samanlaisuusgraafin avulla. Samanlaisuusgraafissa aineiston alkiot toimivat solmuina ja solmuja yhdistävät särmät kuvastavat päätesolmujen välisiä samanlaisuuksia. Näin muodostetun samanlaisuusgraafin vierusmatriisi W vastaa aineiston samanlaisuusmatriisia S, missä alkio w_{ij} kuvastaa solmujen v_i ja v_j välistä samanlaisuutta. Samanlaisuusgraafin tavoitteena on mallintaa paikallista naapurustosuhdetta datapisteiden välillä. Kun samanlaisuusgraafit on muodostettu käyttäen valittua samanlaisuusfunktiota, niitä voidaan edelleen muokata ennen klusteroinnin aloittamista. Seuraavassa esitellään kolme eri samanlaisuusgraafi-tyyppiä. [19]

Samanlaisuusgraafia, jossa yhdistetään kaikki solmut, joiden pareittainen samanlaisuus on suurempaa kuin ε, kutsutaan *ε-naapurusto graafiksi*. [1, s. 179-180][19]

Käytettäessä *k-lähimmän naapurin graafia* solmu v_i yhdistetään solmuun v_j, mikäli solmu v_j on solmun v_i k:n lähimmän naapurin joukossa. Tästä muodostuva samanlaisuusgraafi ei yleensä ole symmetrinen, sillä jos v_j kuuluu solmun v_i k:n lähimmän naapurin joukkoon, ei se tarkoita, että v_i olisi solmun v_j k:n lähimmän naapurin joukossa. On olemassa kaksi eri menetelmää, joilla muodostunut graafi voidaan muokata symmetriseksi. Ensimmäinen vaihtoehto on poistaa särmiltä suunnat. Näin ollen, mikäli v_j on solmun v_i k:n lähimmän naapurin joukossa, tai päinvastoin, lisätään suuntaamaton särmä solmujen v_i ja v_j välille. Tästä syntynyttä graafia kutsutaan *k-lähimmän naapurin graafiksi*. Toinen vaihtoehto on yhdistää solmut v_i ja v_j suuntaamattomalla särmällä, jos ja vain jos molemmat solmut kuuluvat toistensa k:n lähimmän naapurin joukkoon. Näin syntynyttä graafia kutsutaan *molemminpuoliseksi k-lähimmän naapurin graafiksi*. Käytettäessä kumpaa tahansa menetelmää särmien painot kuvastavat särmien päätesolmujen välisiä samanlaisuuksia. [1, s. 179-180][19]

Täydellisessä graafissa jokaisella solmuparilla on paino. Graafin tulisi kuvastaa paikallista naapurustosuhdetta, joten täydellinen graafi on hyödyllinen ainoastaan niissä tapauksissa, joissa samanlaisuusfunktio itsessään mallintaa tällaista suhdetta. Esimerkki tällaisesta samanlaisuusfunktiosta on Gaussin kernel-funktio

$$s(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right),$$

missä parametrin σ avulla säädellään naapuruston leveyttä. Parametrilla on vastaavanlainen rooli, joka parametrilla ε on ε-naapurusto graafissa. [31, s. 186][19]

4 Klusterointialgoritmeja

Tässä luvussa esitellään neljä erilaista klusterointimenetelmää. Enimmäiseksi esitellään van Dongenin [6] kehittämä Markov-klusterointialgoritmi, joka perustuu satunnaiskulkuun graafissa. Tämän jälkeen esitellään yleisesti tunnetuin klusterointialgoritmi k-means. Sen jälkeen käydään läpi Freyn ja Dueckin kehittämä [12] affinity propagation -klusterointialgoritmi. Lopuksi tarkastellaan hieman laajemmin spektrisiä klusterointimenetelmiä ja esitetään sekä normalisoimaton, että kaksi normalisoitua spektristä klusterointialgoritmia. Lisäksi tarkastellaan spektristä klusterointia graafin irrotuksen näkökulmasta.

4.1 Markov-klusterointialgoritmi

Markov-klusterointialgoritmi (MCL) on nopea tutkittavassa graafissa satunnaiskulkua simuloiva klusterointimenetelmä. Graafien luonnollisissa klustereissa särmien lukumäärän klustereiden sisällä tulisi olla suurempi kuin särmien lukumäärän eri klustereiden välillä. Tarkastellaan seuraavaksi graafia, joka omaa luonnollisen klusterirakenteen. Valitaan graafista mielivaltainen solmu s. Kun satunnaisesti siirrytään solmusta s johonkin sen vierussolmuun, on todennäköisempää päätyä saman klusterin solmuun, kuin päätyä jonkin toisen klusterin solmuun. Markov-klusterointialgoritmi perustuu tähän ideaan. Algoritmi laskee satunnaiskulkujen todennäköisyyksiä graafissa käyttäen tähän vuorotellen kahta eri operaatiota, jotka ovat ekspansio ja inflaatio. [6, 7]

Markov-klusterointialgoritmi saa parametrinaan samanlaisuusgraafin. Mikäli tarkasteltava aineisto on esitetty datamatriisin muodossa samanlaisuusgraafi voidaan muodostaa luvussa 3 esitellyillä menetelmillä. Samanlaisuusgraafin särmien painojen tulee olla positiivisia ja kaikkien solmujen aste tulee olla nollaa suurempi. Aluksi algoritmi lisää graafin jokaiselle solmulle luupin. Oletusarvoisesti kunkin solmun luupin painoksi voidaan asetetaan solmun painavimman särmän paino [33]. Tämän jälkeen matriisi muutetaan stokastiseksi Markovin matriisiksi. [7]

Määritelmä 4.1. Olkoon $G = (V, E)$ graafi ja olkoon $|V| = n$. Olkoon graafin G vierusmatriisi $W \geq 0$, jonka alkiot ovat suurempia tai yhtä suuria kuin nolla. Oletetaan lisäksi, että graafin jokaisen solmun aste on nollaa suurempi. Graafiin G liittyvä *Markovin matriisi* τ_G on matriisi, joka koostuu matriisin W normalisoiduista sarakevektoreista. Olkoon D diagonaalimatriisi, jonka diagonaalialkiot vastaavat matriisin W sarakkeiden painojen summia. Siispä $d_{kk} = \sum_{i=1}^{n} W_{ik}$ ja $d_{ij} = 0, i \neq j$. Tällöin

$$\tau_G = WD^{-1}. \tag{4.1}$$

Markovin matriisia τ_G vastaa graafi G', jota kutsutaan graafia G *vastaavaksi Markovin graafiksi*.

Markovin matriisi kuvastaa siirtymätodennäköisyyksiä kaikkien solmuparien välillä. Matriisista voidaan laskea n:n mittaisen satunnaiskulun todennäköisyys minkä tahansa kahden solmuparin välillä korottamalla Markovin matriisi potenssiin n. Tätä operaatiota kutsutaan *ekspansioksi*. Matriisin potenssiin korottamisen jälkeen todennäköisyydet samaan klusteriin kuuluvien solmuparien välillä ovat suurempia, sillä pidemmän mittaiset polut ovat todennäköisempiä saman klusterin sisällä. Tätä ominaisuutta voimistaakseen algoritmi hyödyntää sen toista operaatiota eli inflaatiota. Potenssiin korottamisen jälkeen suoritetaan matriisin Hadamardin kertolasku, jossa matriisin alkiot korotetaan alkioittain potenssiin. Tämän jälkeen matriisin jokainen sarake skaalataan uudelleen niin, että matriisi vastaa jälleen stokastista matriisia. [7]

Määritelmä 4.2. Olkoon $M \in \mathbb{R}^{n\times m}$, $M \geq 0$ matriisi, jonka alkiot ovat suurempia tai yhtä suuria kuin nolla ja $r \in \mathbb{R}_+$ positiivinen reaaliluku. Matriisia, joka saadaan uudelleen skaalaamalla jokainen matriisin M potenssiin r korotettu sarake, merkitään $\Psi_r M$, ja operaatiota Ψ_r kutsutaan *inflaatio-operaatioksi* eksponentti kertoimella r. Siispä $\Psi_r \colon \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$, ja matriisin $\Psi_r M$ alkioille pätee

$$(\Psi_r M)_{ij} = \frac{(M_{ij})^r}{\sum_{k=1}^{n}(M_{kj})^r}. \tag{4.2}$$

Ekspansiota ja inflaatiota jatketaan vuorottelemalla niin kauan, kunnes algoritmi saavuttaa tasapainotilan. Lähes poikkeuksetta matriisi suppenee kaksois-idempotentti matriisiksi [6]. Tämän jälkeen klusterit voidaan muodostaa matriisin riveiltä. Markov-algoritmin pseudokoodi on esitetty algoritmissa 1. Algoritmi saa parametrinaan samanlaisuusgraafin G, graafiin lisättävien luuppien painot Δ, ekspansio-kertoimen e sekä inflaatio-kertoimen r. [7]

Algoritmi 1 Markov-klusterointialgoritmi(G, Δ, e, r) [6]

1: Lisää graafiin luupit $G = G + \Delta$
2: Muodosta matriisi $T_1 = \tau_G$, (4.1)
3: **do**
4: $\quad T_2 = (T_1)^e$
5: $\quad T_1 = \Psi_r T_2$, (4.2)
6: **while** T_1 ei ole idempotentti matriisi
7: Muodosta klusterit matriisista T_1

4.2 k-means-klusterointialgoritmi

k-means-klusterointialgoritmi on yksi tunnetuimmista klusterointimenetelmistä. Algoritmi saa parametrinaan klustereiden lukumäärän k ja sen toiminta alkaa sillä, että se valitsee k klusterikeskusta datapisteiden joukosta. Yksinkertaisimmillaan tämä valinta voidaan tehdä valitsemalla datasta satunnaisesti k datapistettä. Tämän jälkeen algoritmi laskee valitun erilaisuusfunktion avulla jokaiselle datan pisteelle lähimmän klusterikeskuksen ja sijoittaa pisteen keskuksen edustamaan klusteriin. Kun datan jokainen piste on määrätty johonkin klusteriin, lasketaan uudet klusterikeskukset. Kunkin klusterin uudeksi keskukseksi asetetaan klusteriin kuuluvien pisteiden keskiarvo. Tätä proseduuria jatketaan, kunnes klusterit pysyvät riittävän vakaina eli esimerkiksi siihen asti, että kaikki klusterit säilyvät samoina koko iterointikierroksen ajan. k-means-algoritmin pseudokoodi on esitetty algoritmissa 2. Algoritmi saa klustereiden lukumäärän k lisäksi parametrinaan datamatriisin $\mathcal{X} \in \mathbb{R}^{m\times n}$. [1, s. 89-91][13, s. 161-162][24, s. 330-331]

Algoritmi 2 k-means ($\mathcal{X} \in \mathbb{R}^{m\times n}, k$) [1, s. 89]

1: Valitse k pistettä klusterikeskuksiksi
2: **do**
3: $\quad$ Muodosta k klusteria sijoittamalla kukin piste sen lähimpään keskukseen
4: $\quad$ Laske ja määritä jokaiselle klusterille uusi keskus
5: **while** Klusterit pysyvät muuttumattomina

k-means-algoritmin kanssa voidaan käyttää useita eri erilaisuusmittoja. Erilaisuusmitan valinta vaikuttaa siihen, mihin keskuksiin datapisteet päätyvät, ja tätä kautta se vaikuttaa lopullisen

klusterointituloksen laatuun. Yleisesti ottaen suosituin erilaisuusmitta k-means-algoritmia käytettäessä on Euklidinen etäisyys. Klusteroinnin tulosta voidaan arvioida objektifunktion arvolla. Algoritmin tavoite on minimoida tämän funktion arvo. Algoritmin objektifunktio määritellään seuraavalla tavalla. Olkoon X datajoukko, joka koostuu alkioista joita on n kappaletta, ja olkoot $C_1, C_2 \ldots, C_k$, joukon X alkioista muodostettuja erillisiä klustereita. Tällöin objektifunktio on

$$E = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mu(C_i)),$$

missä $\mu(C_i)$ on klusterin C_i keskus. Etäisyysfunktion valinnan lisäksi klustereiden lukumäärän k valinta sekä klusterikeskusten alustukset vaikuttavat saatuun klusterointitulokseen. Saatua klusterointitulosta voidaan yrittää parantaa suorittamalla algoritmi useaan kertaan samoilla parametreilla, mutta vaihtamalla klusterikeskusten alkuvalintaa. Jokaisella suorituskerralla tallennetaan objektifunktion palauttama arvo ja valitaan klusteroinneista se, jolla on saatu pienin virhearvo. [1, s. 89-91][13, s. 161-162][24, s. 330-331]

4.3 Affinity propagation -klusterointialgoritmi

Edellä tarkastellun k-means-klusterointialgoritmin toiminta perustuu siihen, että tarkasteltavasta datajoukosta valitaan edustaja-alkiot, jonka jälkeen klustereiden keskukset pyritään löytämään minimoimalla objektifunktion arvo. Tämän menetelmän heikkoutena on se, että lopullinen klusterointitulos riippuu edustaja-alkioiden valinnasta. Tästä syystä algoritmi suoritetaan usein useampaan kertaan ja hyvä lopputulos saavutetaan vain, jos vähintään yhdessä alkuvalinnassa valitut edustajat ovat riittävän lähellä optimaalisia klusterikeskuksia. Affinity propagation -klusterointialgoritmi (AP) tarjoaa tähän ongelmaan toisenlaisen lähestymistavan, ja siinä kaikki datapisteet ovat potentiaalisia klustereiden edustaja-alkioita. Kun klustereiden keskukset on valittu todellisista datan alkioista, niitä kutsutaan klustereiden edustajiksi. [12]

Affinity propagation -algoritmi saa parametrinaan datapisteiden väliset samanlaisuudet samanlaisuusmatriisissa S. Matriisin arvo $s(i, k)$, missä $i \neq k$ kuvastaa sitä, kuinka hyvin piste k sopii datapisteen i edustajaksi. Affinity propagation -algoritmi ei saa parametrinaan klustereiden lukumäärää, vaan klustereiden lukumäärään voidaan vaikuttaa samanlaisuusmatriisin diagonaalialkioilla. Matriisin diagonaalialkiot $s(k, k)$ kuvastavat sitä, kuinka todennäköisesti kukin datapiste k valikoituu edustaja-alkioksi. Näitä arvoja kutsutaan preferensseiksi. Mitä suurempi preferenssiarvo datapisteellä on, sitä todennäköisemmin se valikoituu edustajaksi. Mikäli kaikkia datan pisteitä voidaan pitää yhtä todennäköisinä vaihtoehtoina klustereiden edustajiksi, kaikki preferenssiarvot asetetaan yhtä suuriksi. Tällöin yhteistä preferenssiarvoa säätelemällä voidaan vaikuttaa muodostuvien klustereiden lukumäärään. [9, 12]

Tarkastellaan datan pisteitä graafin solmuina. Affinity propagation -algoritmin toiminta perustuu kahdenlaisten viestien vaihtoon solmujen välillä, ja nämä viestit kulkeutuvat graafin särmiä pitkin. Jokaisen alkion i ja jokaisen mahdollisen edustajaehdokkaan k välille algoritmi laskee vastuuarvon $r(i, k)$, joka kuvastaa sitä kuinka hyvin piste k toimii edustajana solmulle i. Jokaisen edustajaehdokkaan ja datapisteen välille algoritmi taas laskee saatavuusarvon $a(i, k)$, joka kuvastaa kertynyttä näyttöä siitä, että solmun i tulisi valita edustajakseen solmu k. Algoritmin alussa kaikki saatavuusarvot asetetaan nolliksi $a(i, k) = 0$. Tämän jälkeen vastuut lasketaan käyttäen seuraavaa sääntöä

$$r(i, k) = s(i, k) - \max_{j\,:\,j \neq k} \{a(i, j) + s(i, j)\}. \tag{4.3}$$

Koska algoritmi alustaa saatavuusarvot nolliksi, algoritmin ensimmäisellä iterointikierroksella vastuuarvot $r(i,k)$ riippuvat ainoastaan algoritmin parametrinaan saamista samanlaisuusarvoista. Algoritmin myöhemmillä iterointikierroksilla päivityssääntö (4.3) ottaa huomioon myös päivitetyt saatavuusarvot. Diagonaalialkiot $r(k,k)$ kuvastavat kertynyttä näyttöä sille, että solmun k tulisi toimia edustajana. Alkioiden arvoihin vaikuttavat solmujen preferenssiarvot sekä se, kuinka vastahakoisesti solmut asettuvat joidenkin toisten edustajien klustereihin. [12]

Siinä missä päivityssääntö (4.3) sallii kaikkien edustajaehdokkaiden kilpailla datapisteistä, niin seuraava saatavuuspäivitys kerää näyttöä sille, mikä edustajaehdokas olisi paras valinta kullekin datapisteelle

$$a(i,k) = \min\left\{0, r(k,k) + \sum_{j\,:\,j\neq i, j\neq k} \max\{0, r(j,k)\}\right\}. \tag{4.4}$$

Päivityssääntö ottaa huomioon ainoastaan positiiviset vastuut. Näin ollen sillä ei ole merkitystä, kuinka huonosti jokin edustaja joitakin solmuja edustaa, vaan tärkeämpää on tietää kuinka hyvin se edustaa joitakin datapisteitä. Saatavuus arvot $a(k,k)$ määritellään eritavalla

$$a(k,k) = \sum_{j\,:\,j\neq k} \max\{0, r(j,k)\}. \tag{4.5}$$

Arvo $a(k,k)$ kuvastaa kertynyttä näyttöä siitä, että piste k on edustaja. Arvon suuruuteen vaikuttaa sen saamat positiiviset vastuut muilta datan pisteiltä. [12]

Affinity propagation -algoritmin päivityssääntöjä toistetaan vuorotellen niin kauan, kunnes asetettu päättymisehto toteutuu. Iterointi voidaan lopettaa esimerkiksi silloin, kun jokin ennalta asetettu määrä iterointikierroksia on suoritettu tai kun muutokset viesteissä laskevat jonkin tason alapuolelle. Iteroinnin jälkeen jokainen solmu i sijoitetaan solmun k klusteriin, joka maksimoi summan $a(i,k) + r(i,k)$. Mikäli $k = i$, tällöin solmu i on itse klusterin edustaja-alkio. Joissakin tilanteissa syntyvän numeerisen oskilloinnin välttämiseksi viestejä päivitettäessä tulee käyttää vaimennuskerrointa. Vaimennuskertoimen η arvo tulee valita väliltä $(0, 1)$. Viestejä päivitettäessä uuden viestin arvoksi asetetaan η kertaa edellisen iterointikierroksen arvo, johon lisätään $(1-\eta)$ kertaa kyseisellä iterointikierroksella saatu arvo. [12]

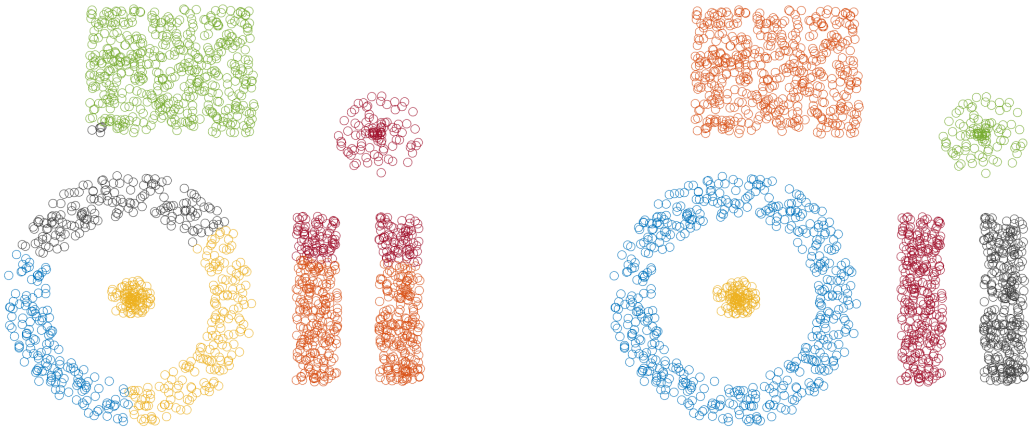
Algoritmi 3 Affinity propagation ($S \in \mathbb{R}^{n\times n}$) [8, 12] [31, s.130]

1: $r(i,k) = 0$, $a(i,k) = 0$, kaikilla $i, k = 1 \ldots n$
2: **while** Päättymisehto ei toteudu **do**
3: $\quad r'(i,k) = s(i,k) - \max_{j\,:\,j\neq k}\{a(i,j) + s(i,j)\}$, (4.3)
4: $\quad r(i,k) = \eta r(i,k) + (1-\eta) r'(i,k)$
5: $\quad a'(k,k) = \sum_{j\,:\,j\neq k} \max\{0, r(j,k)\}$, (4.5)
6: $\quad a'(i,k) = \min\left\{0, r(k,k) + \sum_{j\,:\,j\neq i, j\neq k} \max\{0, r(j,k)\}\right\}$, (4.4)
7: $\quad a(i,k) = \eta a(i,k) + (1-\eta) a'(i,k)$
8: $c_i = \arg\max_k r(i,k) + a(i,k)$, missä c_i merkitsee solmun i klusteria.

Affinity propagation -klusterointialgoritmi saa parametrinaan samanlaisuusmatriisin S. Mikäli tarkasteltava aineisto on esitetty datamatriisina, voidaan samanlaisuusfunktion avulla muodostaa aineiston samanlaisuusmatriisi. Klusterointialgoritmin pseudokoodi on esitetty algoritmissa 3.

4.4 Spektrinen klusterointi

Spektrinen klusterointi on noussut vuosien saatossa suosituksi klusterointimenetelmäksi. Kun sitä verrataan perinteisiin klusterointialgoritmeihin kuten k-means-algoritmiin, niin monissa tilanteissa se tuottaa paremman klusterointituloksen. Spektriset klusterointialgoritmit eivät tee oletuksia klustereiden muodosta, ja näin ollen ne soveltuvat muodoiltaan haasteellisempien aineistojen klusterointiin. Kuvassa 4.1 on esimerkki tilanteesta, jossa spektrinen algoritmi tuottaa halutun klusteroinnin, toisin kuin k-means-algoritmi. Tämän lisäksi spektriset klusterointialgoritmit on helppo toteuttaa, ja ne toimivat tehokkaasti. Spektrisen klusteroinnin idea perustuu spektriseen graafiteoriaan. Spektriset klusterointialgoritmit jakavat samanlaisuusgraafit klustereihin perustuen graafien irrotuksiin. Optimaalinen klusterointi saavutetaan minimoimalla graafien irrotuksiin perustuvan objektifunktion arvo. Ongelmana kuitenkin on, että pääsääntöisesti objektifunktion optimaalisen arvon löytäminen on NP-täydellinen ongelma. Spektrisillä menetelmillä alkuperäinen diskreetti ongelma voidaan kuitenkin relaksoida suoritettavaksi polynomisessa ajassa. [15, 19]



(a) k-means

(b) Spektrinen klusterointi

Kuva 4.1: Esimerkkitapaus tilanteesta, jossa spektrinen algoritmi tuottaa halutun klusteroinnin, toisin kuin k-means-algoritmi.

Tässä luvussa tarkasteltava graafi $G = (V, E)$ on suuntaamaton painotettu graafi, jonka vierusmatriisi on W. Graafin särmien painot ovat ei-negatiivisia, jolloin $w_{ij} = w_{ji} \geq 0$. Tarkasteltaessa ominaisarvoja ei oleteta niiden olevan normalisoituja, vaan esimerkiksi vakiovektori $\mathbb{1}$ ja vektori $c\mathbb{1}$, jollakin $c \neq 0$, mielletään samaksi ominaisvektoriksi. Ominaisarvot järjestetään aina kasvavaan järjestykseen ja puhuttaessa k:sta ensimmäisestä ominaisvektorista tarkoitetaan ominaisvektoreita, jotka vastaavat k:ta pienintä ominaisarvoa. Merkinnällä $w(C)$, missä $C \subseteq V$, tarkoitetaan solmujoukon C indusoiman aligraafin särmien yhteenlaskettua painoa.

4.4.1 Graafien Laplacen matriisit

Spektrisen klusteroinnin pohjana on graafien Laplacen matriisit. Graafille voidaan määritellä sekä normalisoimaton, että normalisoidut Laplacen matriisit. Lähdetään liikkeelle graafin normalisoimattoman Laplacen matriisin määrittelystä. [1, s. 180][19][31, s. 188]

Määritelmä 4.3. Graafin G normalisoimaton *Laplacen matriisi* on

$$L = D - W,$$

missä D on graafin G astematriisi ja W vierusmatriisi.

Graafin normalisoimattomalla Laplacen matriisilla on monia hyödyllisiä ominaisuuksia. Seuraavassa esitellään näistä spektrisen klusteroinnin kannalta merkityksellisimpiä. [1, s. 180-181][19]

Lause 4.4. *Olkoon G suuntaamaton painotettu graafi ja L tämän graafin normalisoimaton Laplacen matriisi. Tällöin matriisi L toteuttaa seuraavat ehdot:*

1. *Kaikille vektoreille $\mathbf{f} \in \mathbb{R}^n$ pätee*

$$\mathbf{f}^T L\mathbf{f} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

2. *L on symmetrinen ja positiivisesti semidefiniitti.*

3. *Matriisin L pienin ominaisarvo on $\lambda = 0$, ja sitä vastaava ominaisvektori on vakiovektori $\mathbf{y} = \mathbb{1}$.*

4. *Matriisilla L on n ei-negatiivista, reaalilukuarvoista ominaisarvoa $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Todistus. 1.

$$\begin{aligned}
\mathbf{f}^{\mathrm{T}} L\mathbf{f} &= \mathbf{f}^{\mathrm{T}}(D - W)\mathbf{f} \\
&= \mathbf{f}^{\mathrm{T}} D\mathbf{f} - \mathbf{f}^{\mathrm{T}} W\mathbf{f} \\
&= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij} \\
&= \frac{1}{2}\left(2\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij}\right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 + \sum_{j=1}^{n} d_j f_j^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij}\right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} f_i^2 \sum_{j=1}^{n} w_{ij} + \sum_{j=1}^{n} f_j^2 \sum_{i=1}^{n} w_{ij} - 2\sum_{i,j=1}^{n} f_i f_j w_{ij}\right) \\
&= \frac{1}{2}\left(\sum_{i,j=1}^{n} w_{ij} f_i^2 + \sum_{i,j=1}^{n} w_{ij} f_j^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j\right) \\
&= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.
\end{aligned}$$

2. Matriisin L symmetrisyys seuraa suoraan matriisien W ja D symmetrisyydestä. Matriisin L positiivinen semidefiniittisyys seuraa kohdasta yksi, sillä $\mathbf{f}^{\mathrm{T}} L\mathbf{f} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \geq 0$, pätee kaikilla $\mathbf{f} \in \mathbb{R}^n$.

3. Kohdan 2 nojalla matriisi L on positiivisesti semidefiniitti, joten pienin mahdollinen ominaisarvo on nolla. Matriisille L pätee $L\mathbb{1} = 0$, joten sen pienin ominaisarvo on nolla, ja tätä vastaa ominaisvektori $\mathbb{1}$.

4. Koska matriisi L on symmetrinen ja sen alkiot ovat reaalilukuja, niin tästä seuraa, että sen ominaisarvot ovat myös reaalisia [1, s. 181]. Näin ollen väite seuraa kohdasta 3.

□

Graafin normalisoimattoman Laplacen matriisin ja sen ominaisarvojen ja ominaisvektoreiden avulla voidaan kuvata monia graafin ominaisuuksia. Käsitellään seuraavaksi yhtä tällaistä ominaisuutta, joka on spektrisen klusteroinnin kannalta merkittävä. [1, s. 181][19]

Määritelmä 4.5. Olkoon $G = (V, E)$ graafi, ja olkoon $C \subset V$ graafin G solmujen osajoukko. Tällöin *indikaattori-vektori* $\mathbb{1}_C = (f_1, \ldots, f_n)^{\mathrm{T}} \in \mathbb{R}^n$ on vektori, missä

$$f_i = \begin{cases} 1, & \text{jos } v_i \in C, \\ 0, & \text{muuten.} \end{cases}$$

Lause 4.6. *Olkoon G suuntaamaton painotettu graafi ja L tämän graafin normalisoimaton Laplacen matriisi. Tällöin matriisin L ominaisarvon $\lambda = 0$ kertaluku k vastaavaa graafin yhdistettyjen komponenttien $C_1, C_2, \ldots, C_k$ lukumäärää graafissa G. Ominaisarvon $\lambda = 0$ ominaisavaruus on viritetty komponenttien indikaattori-vektoreilla $\mathbb{1}_{C_1}, \mathbb{1}_{C_2}, \ldots, \mathbb{1}_{C_k}$.*

Todistus. Tarkastellaan aluksi tilannetta, jossa graafi G on yhdistetty, jolloin siis $k = 1$. Olkoon $\mathbf{f}$ matriisin L ominaisarvoa $\lambda = 0$ vastaava ominaisvektori. Tällöin pätee

$$0 = \mathbf{f}^{\mathrm{T}} L \mathbf{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$

Koska kaikki painot w_{ij} ovat ei-negatiivisia, niin tarkasteltava summa voi saada arvon nolla ainoastaan silloin, kun kaikki summattavat $w_{ij}(f_i - f_j)^2$ saavat arvon nolla. Näin ollen, kun graafin kaksi solmua v_i ja v_j ovat yhdistetyt ($w_{ij} > 0$), niin on oltava $f_i = f_j$. Tästä seuraa, että vektorin $\mathbf{f}$ arvojen tulee olla vakio kaikille niille solmuille, jotka voidaan yhdistää toisiinsa polulla. Graafi muodostuu nyt vain yhdestä komponentista, jolloin sen kaikki solmut voidaan yhdistää toisiinsa polulla. Näin ollen ominaisvektoriksi $\mathbf{f}$ saadaan vakiovektori $\mathbf{y} = \mathbb{1}$, jota vastaa ominaisarvo 0. Tämä ominaisvektori on graafin ainoan komponentin indikaattori-vektori. [19]

Tutkitaan sitten tilannetta, jossa graafi koostuu useammasta yhdistetystä komponentista k. Yleisyyttä menettämättä voidaan olettaa, että graafin solmut on järjestetty sen mukaan mihin komponenttiin solmut kuuluvat. Nyt graafin G vierusmatriisi W on lohkodiagonaalinen matriisi, ja sama pätee myös graafin Laplacen matriisille L

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}.$$

Huomataan, että jokainen matriisin L lohko L_i on itsessään Laplacen matriisi, sillä jokainen matriisi L_i vastaa graafin G aligraafia, joka on sen i:nnes yhdistetty komponentti. Matriisin L lohkodiagonaalisuudesta seuraa, että sen spektri on sen lohkojen ominaisarvojen yhdiste. Ominaisarvoja vastaavat matriisin L ominaisvektorit ovat lohkojen ominaisvektorit, jotka on

täytetty arvolla 0 muiden lohkojen kohdalta. Koska jokainen L_i on yhtenäisen graafin Laplacen matriisi, niin tiedetään, että jokaisella matriisilla L_i on ominaisarvo $\lambda = 0$ yhden kerran, ja ominaisarvoa vastaava ominaisvektori on i:nnen komponentin vakio yksikkövektori. Näin ollen matriisilla L on yhtä monta komponenttia, kuin sillä on ominaisarvoja 0, ja näitä vastaavat ominaisvektorit ovat yhdistettyjen komponenttien indikaattori-vektoreita. [1, s. 181] [19] □

Yllä tarkasteltiin graafin normalisoimatonta Laplacen matriisia. Toinen vaihtoehto on tarkastella graafin normalisoituja Laplacen matriiseja. Kirjallisuudessa on kaksi eri matriisia, joista käytetään nimitystä normalisoitu Laplacen matriisi. [15, 19][31, s. 188]

Määritelmä 4.7. Graafin symmetrinen normalisoitu Laplacen matriisi on

$$L_{\text{sym}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

Satunnaiskulkuun liittyvä graafin Laplacen matriisi on

$$L_{\text{rw}} = D^{-1} L = I - D^{-1} W.$$

Samoin kuin graafin normalisoimattomalle Laplacen matriisille, niin myös graafin normalisoiduille Laplacen matriiseille, voidaan määritellä useita eri ominaisuuksia. Tarkastellaan joitakin näistä ominaisuuksista seuraavaksi. [1, s. 182-183][19]

Lause 4.8. *Olkoon G suuntaamaton painotettu graafi, ja olkoot L_{sym} ja L_{rw} tämän graafin normalisoituja Laplacen matriiseja. Tällöin Laplacen matriisit toteuttavat seuraavat ehdot*

1. *Kaikille vektoreille $\mathbf{f} \in \mathbb{R}^n$ pätee*

$$\mathbf{f}^T L_{sym} \mathbf{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 .$$

2. *λ on matriisin L_{rw} ominaisarvo ja $\mathbf{u}$ tätä vastaava ominaisvektori, jos ja vain jos λ on matriisin L_{sym} ominaisarvo, jota vastaa ominaisvektori $\mathbf{w} = D^{\frac{1}{2}}\mathbf{u}$.*

3. *λ on matriisin L_{rw} ominaisarvo ja $\mathbf{u}$ tätä vastaava ominaisvektori, jos ja vain jos vektori $\mathbf{u}$ ratkaisee yleistetyn ominaisarvoyhtälön $L\mathbf{u} = \lambda D\mathbf{u}$.*

4. *Matriisit L_{sym} ja L_{rw} ovat positiivisesti semidefiniittejä*

5. *Matriisin L_{rw} pienin ominaisarvo on $\lambda = 0$, jota vastaa ominaisvektori, joka on vakiovektori $\mathbf{u} = \mathbb{1}$. Matriisin L_{sym} pienin ominaisarvo on $\lambda = 0$, jota vastaa ominaisvektori $\mathbf{u} = D^{\frac{1}{2}}\mathbb{1}$.*

6. *Matriiseilla L_{sym} ja L_{rw} on n ei-negatiivista, reaalilukuarvoista ominaisarvoa $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Todistus. 1.

$$\begin{aligned}
\mathbf{f}^{\mathrm{T}} L_{\mathrm{sym}} \mathbf{f} &= \mathbf{f}^{\mathrm{T}} (I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) \mathbf{f} \\
&= \mathbf{f}^{\mathrm{T}} I \mathbf{f} - \mathbf{f}^{\mathrm{T}} D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \mathbf{f} \\
&= \sum_{i=1}^{n} f_i^2 - \sum_{i,j=1}^{n} f_i f_j \frac{w_{ij}}{\sqrt{d_i d_j}} \\
&= \frac{1}{2} \left(2 \sum_{i=1}^{n} f_i^2 - 2 \sum_{i,j=1}^{n} f_i f_j \frac{w_{ij}}{\sqrt{d_i d_j}} \right) \\
&= \frac{1}{2} \left(\sum_{i=1}^{n} f_i^2 + \sum_{j=1}^{n} f_j^2 - 2 \sum_{i,j=1}^{n} f_i f_j \frac{w_{ij}}{\sqrt{d_i d_j}} \right) \\
&= \frac{1}{2} \left(\sum_{i=1}^{n} f_i^2 \sum_{j=1}^{n} \frac{w_{ij}}{d_i} + \sum_{j=1}^{n} f_j^2 \sum_{i=1}^{n} \frac{w_{ij}}{d_j} - 2 \sum_{i,j=1}^{n} f_i f_j \frac{w_{ij}}{\sqrt{d_i d_j}} \right) \\
&= \frac{1}{2} \left(\sum_{i,j=1}^{n} w_{ij} \frac{f_i^2}{d_i} + \sum_{i,j=1}^{n} w_{ij} \frac{f_j^2}{d_j} - 2 \sum_{i,j=1}^{n} w_{ij} \frac{f_i f_j}{\sqrt{d_i d_j}} \right) \\
&= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 .
\end{aligned}$$

2.

$$\begin{aligned}
& L_{\mathrm{sym}} \mathbf{w} = \lambda \mathbf{w} \\
\Leftrightarrow \quad & D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \mathbf{w} = \lambda \mathbf{w} \\
\Leftrightarrow \quad & D^{-\frac{1}{2}} D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \mathbf{w} = D^{-\frac{1}{2}} \lambda \mathbf{w} \\
\Leftrightarrow \quad & D^{-1} L D^{-\frac{1}{2}} D^{\frac{1}{2}} \mathbf{u} = \lambda D^{-\frac{1}{2}} D^{\frac{1}{2}} \mathbf{u} \\
\Leftrightarrow \quad & D^{-1} L \mathbf{u} = \lambda \mathbf{u} \\
\Leftrightarrow \quad & L_{\mathrm{rw}} \mathbf{u} = \lambda \mathbf{u}.
\end{aligned}$$

3.

$$\begin{aligned}
& L_{\mathrm{rw}} \mathbf{u} = \lambda \mathbf{u} \\
\Leftrightarrow \quad & D^{-1} L \mathbf{u} = \lambda \mathbf{u} \\
\Leftrightarrow \quad & D D^{-1} L \mathbf{u} = D \lambda \mathbf{u} \\
\Leftrightarrow \quad & L \mathbf{u} = \lambda D \mathbf{u}.
\end{aligned}$$

4. Matriisille L_{sym} väite seuraa kohdasta 1, sillä

$$\mathbf{f}^{\mathrm{T}} L_{sym} \mathbf{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \geq 0$$

ja matriisille L_{rw} väite seuraa tämän jälkeen kohdasta 2.

5. Ensimmäinen väite on selvä, sillä $L_{\text{rw}}\mathbb{1} = 0$. Toinen väite seuraa kohdasta 2.

6. Koska matriisi L_{sym} on symmetrinen ja sen alkiot ovat reaalilukuja, niin tästä seuraa, että sen ominaisarvot ovat myös reaalisia [1, s. 183]. Näin ollen kohdan 5 nojalla väite pätee matriisille L_{sym}. Tämän jälkeen matriisille L_{rw} väite seuraa kohdasta 2.

□

Graafin normalisoiduille Laplacen matriiseille voidaan osoittaa samalla tavalla kuin normalisoimattomallekin Laplacen matriisille, että ominaisarvon $\lambda = 0$ kertaluku vastaa graafin yhdistettyjen komponenttien lukumäärää. [1, s. 184][19]

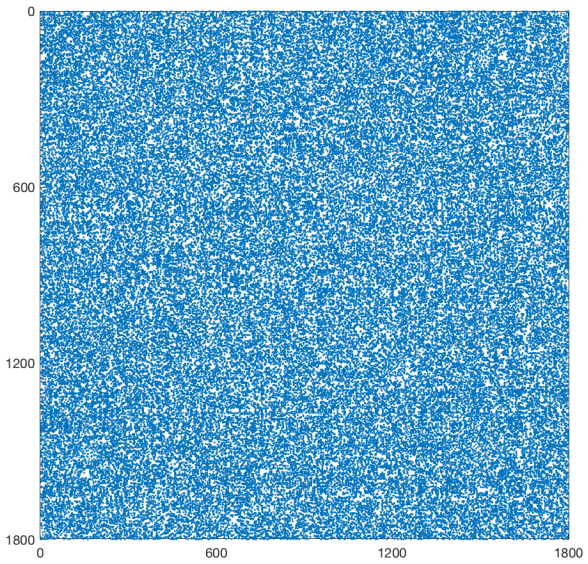
Lause 4.9. *Olkoon G suuntaamaton painotettu graafi ja L_{sym} sekä L_{rw} tämän graafin normalisoituja Laplacen matriiseja. Tällöin molemmille matriiseille L_{sym} ja L_{rw} pätee, että ominaisarvon $\lambda = 0$ kertaluku k vastaa graafin yhdistettyjen komponenttien $C_1, C_2, \ldots, C_k$ lukumäärää graafissa G. Ominaisarvon $\lambda = 0$ ominaisavaruus on viritetty komponenttien indikaattori-vektoreilla $\mathbb{1}_{C_i}$, kun Laplacen matriisina on L_{rw}. Kun Laplacen matriisina on L_{sym} ominaisarvon $\lambda = 0$ ominaisavaruus on viritetty vektoreilla $D^{\frac{1}{2}}\mathbb{1}_{C_i}$*

Todistus. Matriisille L_{sym} väite todistetaan vastaavalla tavalla kuin lause 4.6 käyttäen apuna lauseen 4.8 kohtaa 1. Tämän jälkeen väite voidaan todistaa matriisille L_{rw} käyttäen apuna lauseen 4.8 kohtaa 2. □

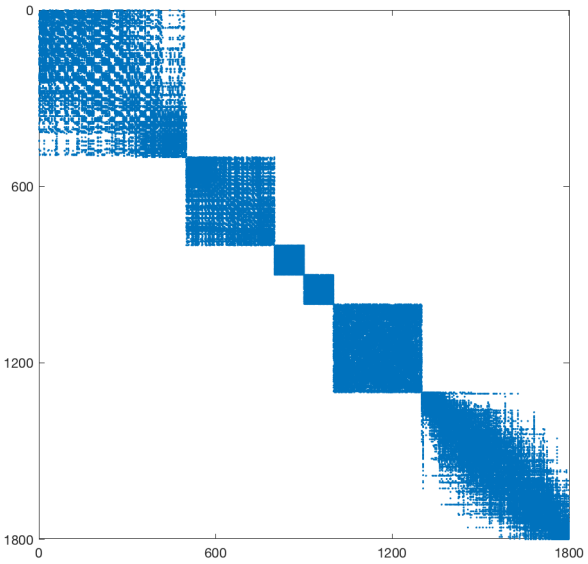
4.4.2 Spektriset klusterointialgoritmit

Pohditaan seuraavaksi ideaalitilannetta, jossa graafi muodostuu k:sta erillisestä klikistä. Tällöin graafi siis rakentuu erillisistä komponenteista ja sen Laplacen matriisi on lohkodiagonaalinen ja sillä on ominaisarvo $\lambda = 0$, jonka kertaluku on k. Näitä ominaisarvoja vastaa ominaisvektorit, jotka toimivat indikaattori-vektoreina kullekin komponentille siten, että klikkiin kuuluvilla solmuilla on eri arvo vektorissa kuin solmuilla, jotka ovat klikin ulkopuolella. Muokataan tätä graafia niin, että lisätään särmiä klikkien välille ja vastaavasti poistetaan joitakin särmiä klikkien sisältä. Tällaisessa, klusteroinnin kannalta lähes ideaalissa tapauksessa, jossa graafi omaa selvän klusterirakenteen, mutta klustereiden välillä kulkee särmiä, ominaisarvot, jotka olivat nolla, kasvavat hieman. Samoin myös vastaavat ominaisvektorit muuttuvat. Tästä huolimatta Laplacen matriisin ominaisvektoreiden avulla graafin perimmäinen rakenne voidaan havaita, vaikka tällaisia muutoksia tehtäisiinkin. Tämä idea toimii pohjana spektriselle klusteroinnille, jossa Laplacen matriisin ominaisvektoreiden kombinaatiota käytetään graafin lopullisen klusteroinnin laskemiseen. Kuvassa 4.2 on esitetty kuvassa 4.1b näkyvän aineiston 5-lähimmän naapurin samanlaisuusgraafin vierusmatriisi kahdessa eri järjestyksessä. Kuvassa 4.2a matriisin alkiot ovat mielivaltaisessa järjestyksessä, ja kuvassa 4.2b alkiot on järjestetty Laplacen matriisin ominaisvektoreiden arvojen mukaan, jolloin havaitaan kuinka matriisista selvästi erottuu kuusi erillistä lohkoa. [19, 23]

Pääsääntöisesti spektriset klusterointialgoritmit rakentuvat kolmesta eri vaiheesta, jotka ovat aineiston esikäsittely, aineiston muuttaminen spektriseen muotoon ja lopullinen klusterointi. Aluksi muodostetaan samanlaisuusgraafi ja tämän vierusmatriisi. Tämän jälkeen muodostetaan graafin Laplacen matriisi ja lasketaan tämän ominaisarvot ja ominaisvektorit. Näiden ominaisvektoreiden avulla jokainen datan alkio kuvataan spektriseen esitysmuotoon perustuen yhteen tai useampaan ominaisvektoriin. Lopuksi solmut sijoitetaan yhteen tai useampaan klusteriin perustuen tähän uuteen esitysmuotoon. Kun aineisto halutaan jakaa k:hon ($k > 2$) klusteriin, voidaan tähän käyttää kahta eri menetelmää. Ensimmäinen vaihtoehto on graafin rekursiivinen



(a) Samanlaisuusgraafin vierusmatriisi, kun alkiot ovat mielivaltaisessa järjestyksessä.



(b) Ominaisvektoreiden arvojen mukaan järjestetty samanlaisuusgraafin vierusmatriisi.

Kuva 4.2: Samanlaisuusgraafin vierusmatriisi.

kahtiajako, jossa graafi jaetaan ensiksi kahteen osaan perustuen Laplacen matriisin toiseksi suurimpaan ominaisvektoriin, jota kutsutaan myös *Fielder*-vektoriksi. Tämän jälkeen samaa menetelmää jatketaan rekursiivisesti uusille aligraafeille niin kauan kunnes halutta määrä klustereita on saavutettu. Tämä menetelmä on kuitenkin epävakaa, ja koska se hyödyntää ainoastaan yhtä ominaisvektoria, menetelmä menettää klusteroinnin kannalta hyödyllistä informaatiota muiden ominaisvektoreiden mukana. Toinen vaihtoehto lopullisen klusteroinnin toteuttamiselle onkin hyödyntää useampaa Laplacen matriisin ominaisvektoria ja käyttää näiden vektoreiden arvoja kuvaamaan alkuperäiset datapisteet spektriseen avaruuteen. Kun datan alkiot on esitetty uudessa muodossa, voidaan lopullinen klusterointi suorittaa esimerkiksi k-means-algoritmilla. [15][31, s. 183-218]

Esitellään seuraavaksi yleisimmät spektriset klusterointialgoritmit. Oletuksena on, että aineisto koostuu havainnoista, joita on n kappaletta. Kaikkien havaintojen välille lasketaan samanlaisuusarvo $s_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$ jollakin samanlaisuusfunktiolla, joka on symmetrinen ja ei-negatiivinen. Muodostuneesta samanlaisuusmatriisista käytetään merkintää S. Algoritmissa 4 on esitetty normalisoimattoman spektrisen klusterointialgoritmin pseudokoodi.

Algoritmi 4 Normalisoimaton spektrinen klusterointi ($S \in \mathbb{R}^{n \times n}$, k) [19]

1: Muodosta samanlaisuusgraafi jollakin aliluvussa 3.3.2 esitetyllä tavalla. Merkitään samanlaisuusgraafin vierusmatriisia kirjaimella W ja astematriisia kirjaimella D.
2: Laske $L = D - W$.
3: Laske matriisin L k ensimmäistä ominaisvektoria $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$.
4: Muodosta matriisi $U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k)$, $U \in \mathbb{R}^{n \times k}$.
5: C = k-MEANS(U)

Normalisoidusta spektrisestä klusterointialgoritmista on olemassa kaksi eri versiota, joissa toisessa käytetään Laplacen matriisia L_{rw} ja toisessa matriisia L_{sym}. Algoritmissa 5 käytetään yleistetyn ominaisarvoyhtälön $L\mathbf{u} = \lambda D\mathbf{u}$ ominaisvektoreita, jotka lauseen 4.8 nojalla vastaavat

matriisin L_{rw} ominaisvektoreita. Matriisin L_{rw} laskeminen edellyttää kertolaskua matriisilla D^{-1}, joka saattaa sisältää hyvin pieniä arvoja. Näin ollen yleistetyn ominaisarvoyhtälön ratkaiseminen on numeerisesti vakaampi menetelmä ominaisvektoreiden laskemiseen. [19][31, s. 197]

Algoritmi 5 Normalisoitu spektrinen klusterointi (Shi ja Malik) ($S \in \mathbb{R}^{n \times n}$, k) [19]

1: Muodosta samanlaisuusgraafi jollakin aliluvussa 3.3.2 esitetyllä tavalla. Merkitään samanlaisuusgraafin vierusmatriisia kirjaimella W ja astematriisia kirjaimella D.
2: Laske $L = D - W$.
3: Laske yleistetyn ominaisarvoyhtälön $L\mathbf{u} = \lambda D\mathbf{u}$ k ensimmäistä yleistettyä ominaisvektoria $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$.
4: Muodosta matriisi $U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k)$, $U \in \mathbb{R}^{n \times k}$.
5: C = k-MEANS(U)

Algoritmissa 6 esiteltävä normalisoitu spektrinen klusterointialgoritmi puolestaan käyttää Laplacen matriisia L_{sym}. Algoritmi eroaa algoritmeista 4 ja 5 siinä, että ominaisvektoreiden laskemisen jälkeen se suorittaa rivien normalisoinnin. Tämä on tarpeen, koska ominaisarvon $\lambda = 0$ ominaisavaruus on viritetty ominaisvektoreilla $D^{-\frac{1}{2}}\mathbb{1}_{C_i}$ eikä vektoreilla $\mathbb{1}_{C_i}$. [1, s. 184][19]

Algoritmi 6 Normalisoitu spektrinen klusterointi (Ng, Jordan ja Weiss) ($S \in \mathbb{R}^{n \times n}$, k) [19]

1: Muodosta samanlaisuusgraafi jollakin aliluvussa 3.3.2 esitetyllä tavalla. Merkitään samanlaisuusgraafin vierusmatriisia kirjaimella W ja astematriisia kirjaimella D.
2: Laske $L_{\text{sym}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.
3: Laske matriisin L_{sym} k ensimmäistä ominaisvektoria $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$.
4: Muodosta matriisi $U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k)$, $U \in \mathbb{R}^{n \times k}$.
5: Muodosta matriisi T matriisista U normalisoimalla rivit normiin 1, eli $t_{ij} = u_{ij} / \sqrt{\sum_{j=1}^{k} u_{ij}^2}$.
6: C = k-MEANS(T)

Edellä esitetyt algoritmit ovat kaikki idealtaan samanlaisia, ja ne eroavat toisistaan niissä käytettävien Laplacen matriisien osalta. Graafien Laplacen matriisien ominaisuuksien ansiosta algoritmien datapisteille tekemä muunnos on klusteroinnin kannalta edullinen ja muunnoksen jälkeen k-means-klusterointialgoritmi toimii tehokkaasti [19]. Algoritmien lopullisen klusteroinnin suorittamiseen voidaan käyttää myös muuta menetelmää kuin k-means-algoritmia. Näitä menetelmiä on listattuna esimerkiksi kirjassa [31, s. 218-219]. Seuraavaksi tarkastellaan, kuinka spektrisillä klusterointimenetelmillä voidaan approksimoida erilaisia graafin irrotuksia.

4.4.3 Spektrinen klusterointi irrotuksen näkökulmasta

Kun klusteroitava aineisto on esitetty samanlaisuusgraafina, voidaan klusteroinnin tavoite esittää seuraavanlaisessa muodossa. Etsitään graafin jako siten, että särmien painot eri ryhmien välillä ovat pieniä ja särmien painot ryhmien sisällä ovat suuria. Seuraavassa tarkastellaan graafien klusterointia irrotuksen näkökulmasta ja nähdään kuinka spektrinen klusterointi voidaan johtaa approksimaatioksi tällaisille graafien jaoille. Tässä aliluvussa on käytetty lähteenä kirjoja [1, s. 185-188][31, s. 189-224] sekä artikkelia [19].

Olkoon G samanlaisuusgraafi ja olkoon W tämän graafin vierusmatriisi. Kaikkein suorin ja yksinkertaisin tapa löytää graafin jako on ratkaista pienimmän irrotuksen ongelma. Olkoon k haluttujen osajoukkojen lukumäärä. Tällöin pienimmän irrotuksen ongelmassa etsitään graafille

jako $C_1, \ldots, C_k$, joka minimoi ehdon

$$\text{cut}(C_1, \ldots, C_k) = \frac{1}{2} \sum_{i=1}^{k} W(C_i, \overline{C_i}).$$

Erityisesti tapauksessa $k = 2$ pienimmän irrotuksen ongelma on helppo ratkaista [25]. Ongelmana on, että näin muodostunut graafin jako ei kuitenkaan usein tuota klusteroinnin kannalta mielekästä jakoa, vaan erottaa yhden solmun muusta graafista. Irrotuksen minimoimisen lisäksi halutaan, että klusterit muodostuvat järjellisen suurista ryhmistä solmuja. Pienimmän irrotuksen objektifunktiota hieman muokkaamalla voidaan kuitenkin kiertää edellä kuvattu ongelma. Yleisimmät funktiot ovat RatioCut ja normalisoituirrotus Ncut. Nämä objektifunktiot eroavat toisistaan siinä, että RatioCutissa graafin osajoukon C koko määritellään sen solmujen lukumäärän $|C|$ perusteella, kun taas Ncutissa koko määritellään sen särmien painojen perusteella $w(C)$. Määritelmät ovat

$$\text{RatioCut}(C_1, \ldots, C_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{|C_i|} = \sum_{i=1}^{k} \frac{cut(C_i, \overline{C_i})}{|C_i|},$$

$$\text{Ncut}(C_1, \ldots, C_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{w(C_i)} = \sum_{i=1}^{k} \frac{cut(C_i, \overline{C_i})}{w(C_i)}.$$

Erityisesti summan $\sum_{i=1}^{k}(1/|C_i|)$ minimi saavutetaan, kun kaikkien osajoukkojen C_i koko on sama, ja vastaavasti summan $\sum_{i=1}^{k}(1/w(C_i))$ minimi saavutetaan kun kaikkien osajoukkojen C_i paino on sama. Molemmat objektifunktiot pyrkivät siis löytämään klusterit siten, että ne ovat tasapainossa kokonsa tai vastaavasti painonsa suhteen. Ongelmana on, että tasapainoehdon lisääminen tekee pienimmän irrotuksen ongelmasta laskennallisesti NP-täydellisen ongelman. Spektriset klusterointialgoritmit kuitenkin ratkaisevat relaksoidut versiot näistä ongelmista. Seuraavassa nähdään kuinka relaksoimalla RatioCut päädytään normalisoimattomaan spektriseen klusterointiin. Vastaavasti Ncutin relaksointi johtaa normalisoituun spektriseen klusterointiin. Ennen kuin siirrytään tarkastelemaan optimointiongelmia, esitetään ongelmien kannalta kaksi tärkeää lausetta [31, s. 200-202].

Lause 4.10. *Olkoon $M \in \mathbb{R}^{m \times m}$ symmetrinen matriisi, jonka ominaisarvot ovat $\lambda_1 \leq \ldots \leq \lambda_m$. Olkoon $\mathcal{V}_k$ ominaisvektoreiden $\mathbf{v}_1, \ldots, \mathbf{v}_k$ virittämä avaruus ja sovitaan, että $\mathcal{V}_0 = \{\mathbf{0}\}$. Tällöin*

$$\lambda_k = \min_{\substack{\mathbf{u} \neq \mathbf{0} \\ \mathbf{u} \perp \mathcal{V}_{k-1}}} \frac{\mathbf{u}^T M \mathbf{u}}{\mathbf{u}^T \mathbf{u}},$$

missä $\mathbf{u} \perp \mathcal{V}_k$ tarkoittaa, että vektori $\mathbf{u}$ on ortogonaalinen kaikkien vektoreiden $\mathbf{v} \in \mathcal{V}_k$ kanssa.

Lause 4.11. *Olkoon $M \in \mathbb{R}^{m \times m}$ symmetrinen matriisi, jonka ominaisarvot ovat $0 \leq \lambda_1 \leq \ldots \leq \lambda_m \in \mathbb{R}$ ja niitä vastaavat ominaisvektorit $\mathbf{v}_1, \ldots, \mathbf{v}_m$. Olkoon $U = \mathbb{R}^{m \times k}$, $1 \leq k \leq m$ unitaarinen matriisi ja olkoon I, $k \times k$ -identiteettimatriisi. Tällöin ratkaisu ongelmaan*

$$Y = \arg\min_{U^T U = I_k} \text{Tr}(U^T M U),$$

on matriisi $Y = (\mathbf{v}_1, \ldots, \mathbf{v}_k)Q$, missä $Q \in \mathbb{C}^{k \times k}$ on unitaarinen matriisi.

Lähdetään sitten tarkastelemaan RatioCutia, ja tarkastellaan aluksi tapausta $k = 2$, jolloin siis graafin solmut jaetaan kahteen osaan. Tavoitteena on ratkaista seuraavanlainen optimointiongelma

$$\min_{C \subset V} \text{RatioCut}(C, \overline{C}). \tag{4.6}$$

Muokataan tätä ongelmaa hieman käytännöllisempään muotoon. Määritellään vektori $\mathbf{f} = (f_1, \ldots, f_n)^{\mathrm{T}} \in \mathbb{R}^n$ osajoukon $C \subset V$ avulla siten, että

$$f_i = \begin{cases} \sqrt{|\overline{C}|/|C|}, & \text{jos } v_i \in C, \\ -\sqrt{|C|/|\overline{C}|}, & \text{jos } v_i \in \overline{C}. \end{cases} \tag{4.7}$$

Nyt optimointiongelma (4.6) voidaan kirjoittaa uudessa muodossa käyttäen graafin normalisoimatonta Laplacen matriisia. Käyttämällä apuna lauseen 4.4 kohtaa 1 saadaan

$$\begin{aligned}
\mathbf{f}^{\mathrm{T}} L \mathbf{f} &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 \\
&= \frac{1}{2} \sum_{v_i \in C, v_j \in \overline{C}} w_{ij} \left(\sqrt{\frac{|\overline{C}|}{|C|}} + \sqrt{\frac{|C|}{|\overline{C}|}} \right)^2 + \frac{1}{2} \sum_{v_i \in \overline{C}, v_j \in C} w_{ij} \left(-\sqrt{\frac{|C|}{|\overline{C}|}} - \sqrt{\frac{|\overline{C}|}{|C|}} \right)^2 \\
&= \frac{1}{2} \sum_{v_i \in C, v_j \in \overline{C}} w_{ij} \left(\frac{|\overline{C}|}{|C|} + \frac{|C|}{|\overline{C}|} + 2 \right) + \frac{1}{2} \sum_{v_i \in \overline{C}, v_j \in C} w_{ij} \left(\frac{|\overline{C}|}{|C|} + \frac{|C|}{|\overline{C}|} + 2 \right) \\
&= \left(\frac{1}{2} \sum_{v_i \in C, v_j \in \overline{C}} w_{ij} + \frac{1}{2} \sum_{v_i \in \overline{C}, v_j \in C} w_{ij} \right) \left(\frac{|\overline{C}|}{|C|} + \frac{|C|}{|\overline{C}|} + 2 \right) \\
&= \left(\frac{1}{2} \text{cut}(C, \overline{C}) + \frac{1}{2} \text{cut}(\overline{C}, C) \right) \left(\frac{|\overline{C}|}{|C|} + \frac{|C|}{|\overline{C}|} + \frac{|C|}{|C|} + \frac{|\overline{C}|}{|\overline{C}|} \right) \\
&= \text{cut}(C, \overline{C}) \left(\frac{|C| + |\overline{C}|}{|C|} + \frac{|C| + |\overline{C}|}{|\overline{C}|} \right) \\
&= \text{cut}(C, \overline{C}) \left(\frac{|V|}{|C|} + \frac{|V|}{|\overline{C}|} \right) \\
&= |V| \left(\frac{\text{cut}(C, \overline{C})}{|C|} + \frac{\text{cut}(C, \overline{C})}{|\overline{C}|} \right) \\
&= |V| \cdot \text{RatioCut}(C, \overline{C}).
\end{aligned}$$

Lisäksi saadaan, että

$$\mathbf{f}\mathbb{1} = \sum_{i=1}^{n} f_i = \sum_{v_i \in C} \sqrt{\frac{|\overline{C}|}{|C|}} - \sum_{v_i \in \overline{C}} \sqrt{\frac{|C|}{|\overline{C}|}} = |C| \sqrt{\frac{|\overline{C}|}{|C|}} - |\overline{C}| \sqrt{\frac{|C|}{|\overline{C}|}} = 0.$$

Toisin sanoen, kun vektori $\mathbf{f}$ määritellään kuten yhtälössä (4.7), se on ortogonaalinen vakiovektorin $\mathbb{1}$ kanssa. Lopuksi vielä huomataan, että vektorille $\mathbf{f}$ pätee

$$\mathbf{f}^{\mathrm{T}} \mathbf{f} = \sum_{i=1}^{n} f_i^2 = |C| \frac{|\overline{C}|}{|C|} + |\overline{C}| \frac{|C|}{|\overline{C}|} = |\overline{C}| + |C| = n.$$

Nyt ehdon (4.6) minimointiongelma voidaan yhtäpitävästi esittää muodossa

$$\min_{C \subset V} \frac{\mathbf{f}^{\mathrm{T}} L \mathbf{f}}{\mathbf{f}^{\mathrm{T}} \mathbf{f}}, \text{ missä } \mathbf{f} \perp \mathbb{1} \text{ ja } f_i \text{ on yhtälön (4.7) muotoa.} \tag{4.8}$$

Ehdon (4.8) minimointiongelma on diskreetti optimointiongelma, sillä ratkaisuvektorin $\mathbf{f}$ alkioiksi f_i sallitaan vain kaksi tiettyä arvoa. Tämä optimointiongelma on NP-täydellinen, ja tässä tapauksessa selvin tapa relaksoida se on hylätä ehdon (4.8) diskreettisyysehto ja sallia vektorin $\mathbf{f}$ alkioille f_i mielivaltaisia reaalilukuarvoja. Tästä saadaan seuraavanlainen relaksoitu optimointiongelma,

$$\min_{\mathbf{f} \in \mathbb{R}^n} \frac{\mathbf{f}^{\mathrm{T}} L \mathbf{f}}{\mathbf{f}^{\mathrm{T}} \mathbf{f}}, \text{ missä } \mathbf{f} \perp \mathbb{1} \text{ ja } \mathbf{f}^{\mathrm{T}} \mathbf{f} = n.$$

Lauseen 4.4 kohdan kolme sekä lauseen 4.10 nojalla nähdään, että tämän optimointiongelman ratkaisu on vektori $\mathbf{f}$, joka saa arvokseen matriisin L toiseksi pienintä ominaisarvoa vastaavan ominaisvektorin arvon. Näin ollen siis RatioCutin minimointia voidaan arvioida Laplacen matriisin L toisella ominaisvektorilla. Jotta nyt saataisiin muodostettua graafin jako, vektori $\mathbf{f}$ tulee muuntaa relaksoidun ongelman reaalisesta ratkaisuvektorista takaisin diskreetiksi indikaattorivektoriksi. Yksinkertaisin tapa tämän tekemiseen on käyttää vektorin $\mathbf{f}$ merkkiä indikaattorifunktiona, jolloin saadaan

$$\begin{cases} v_i \in C, & \text{jos } f_i \geq 0, \\ v_i \in \overline{C}, & \text{jos } f_i < 0. \end{cases}$$

Erityisesti tapauksessa $k > 2$, jota seuraavaksi käsitellään, tämän kaltainen menettely on liian yksinkertainen. Sen sijaan monet spektriset klusterointialgoritmit jakavat vektorin $\mathbf{f}$ alkiot f_i kahteen ryhmään A ja $\overline{A}$ k-means-klusterointialgoritmin avulla. Tällöin tutkittaville datapisteille saadaan ratkaisuna seuraavanlainen klusterointi

$$\begin{cases} v_i \in C, & \text{jos } f_i \in A, \\ v_i \in \overline{C}, & \text{jos } f_i \in \overline{A}. \end{cases}$$

Tämä vastaa tarkalleen normalisoimatonta spektristä klusterointia tapauksessa $k = 2$.

Edellinen vastasi siis tapausta, jossa tutkittava datajoukko jaettiin kahteen osaan $k = 2$. Tarkastellaan seuraavaksi relaksoitua RatioCutin minimointiongelmaa yleisessä tapauksessa. Olkoon joukon V jako k:hon erilliseen joukkoon $C_1, \ldots, C_k$. Määritellään k indikaattori-vektoria $\mathbf{h}_j = (h_{1j}, \ldots, h_{nj})^{\mathrm{T}}$ siten, että

$$h_{ij} = \begin{cases} 1/\sqrt{|C_j|}, & \text{jos } v_i \in C_j, \\ 0, & \text{muuten,} \end{cases} \tag{4.9}$$

missä $i = 1, \ldots, n$ ja $j = 1, \ldots, k$. Olkoon $H \in \mathbb{R}^{n \times k}$ matriisi, joka sisältää nämä k indikaattorivektoria sarakkeinaan. Huomataan, että $\mathbf{h}_i^{\mathrm{T}} \mathbf{h}_i = 1$ ja $\mathbf{h}_i^{\mathrm{T}} \mathbf{h}_j = 0$, jos $i \neq j$, joten matriisin H sarakkeet ovat keskenään ortogonaalisia, ja matriisille H pätee $H^{\mathrm{T}} H = I$. Samoin kuin

tapauksessa $k = 2$, saadaan

$$\begin{aligned}
\mathbf{h}_i^{\mathrm{T}} L \mathbf{h}_i &= \frac{1}{2} \sum_{j,l=1}^{n} w_{jl}(h_{ij} - h_{il})^2 \\
&= \frac{1}{2} \sum_{v_j \in C_i, v_l \in \overline{C_i}} w_{jl} \left(\frac{1}{\sqrt{|C_i|}} + 0 \right)^2 + \frac{1}{2} \sum_{v_j \in \overline{C_i}, v_l \in C_i} w_{jl} \left(0 - \frac{1}{\sqrt{|C_i|}} \right)^2 \\
&= \frac{1}{2} \sum_{v_j \in C_i, v_l \in \overline{C_i}} w_{jl} \frac{1}{|C_i|} + \frac{1}{2} \sum_{v_j \in \overline{C_i}, v_l \in C_i} w_{jl} \frac{1}{|C_i|} \\
&= \frac{1}{2} \frac{\mathrm{cut}(C_i, \overline{C_i})}{|C_i|} + \frac{1}{2} \frac{\mathrm{cut}(C_i, \overline{C_i})}{|C_i|} \\
&= \frac{\mathrm{cut}(C_i, \overline{C_i})}{|C_i|}.
\end{aligned}$$

Tämän lisäksi nähdään, että

$$\mathbf{h}_i^{\mathrm{T}} L \mathbf{h}_i = (H^{\mathrm{T}} L H)_{ii}.$$

Nämä yhdistämällä saadaan

$$\begin{aligned}
\mathrm{RatioCut}(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{\mathrm{cut}(C_i, \overline{C_i})}{|C_i|} &= \sum_{i=1}^{k} \mathbf{h}_i^{\mathrm{T}} L \mathbf{h}_i \\
&= \sum_{i=1}^{k} (H^{\mathrm{T}} L H)_{ii} \\
&= \mathrm{Tr}(H^{\mathrm{T}} L H).
\end{aligned}$$

Näin ollen $\mathrm{RatioCut}(C_1, \ldots, C_k)$ minimointiongelma voidaan kirjoittaa uudelleen muodossa

$$\min_{C_1, \ldots, C_k} \mathrm{Tr}(H^{\mathrm{T}} L H), \text{ missä } H^{\mathrm{T}} H = I \text{ ja } H \text{ on määritelty kuten ehdossa (4.9).} \tag{4.10}$$

Kuten tapauksessa $k = 2$, relaksoidaan ehdon (4.10) ongelma sallimalla matriisin H alkioiksi mielivaltaisia reaalilukuja. Tällöin relaksoiduksi ongelmaksi saadaan

$$\min_{H \in \mathbb{R}^{n \times k}} \mathrm{Tr}(H^{\mathrm{T}} L H), \text{ missä } H^{\mathrm{T}} H = I. \tag{4.11}$$

Lauseen 4.11 nojalla ratkaisu saadaan valitsemalla matriisiksi H sellainen matriisi, joka sisältää matriisin L k ensimmäistä ominaisvektoria sarakkeinaan. Lauseen unitaarisen matriisin voidaan olettaa olevan identiteettimatriisi. Kun tarkastellaan normalisoimattoman spektrisen klusterointialgoritmin pseudokoodia, huomataan, että matriisia H vastaa nyt algoritmin matriisi U. Kuten tapauksessa $k = 2$, tulee reaaliarvoinen ratkaisumatriisi palauttaa takaisin diskreetiksi jaoksi. Yleinen tapa on tehdä se käyttäen k-means-klusterointialgoritmia matriisin U riveille. Tämä siis johtaa suoraan yleiseen normalisoimattomaan spektriseen klusterointialgoritmiin.

Johdetaan seuraavaksi normalisoitu spektrinen klusterointi relaksoimalla Ncutin minimointi. Tämä tapahtuu hyvin samanlaisella tavalla kuin edellä RatioCutin tapauksessa. Myös Ncutin tapauksessa voidaan tarkastella aluksi tapausta, jossa $k = 2$ ja graafi jaetaan kahteen osaan [19]. Siirrytään nyt kuitenkin suoraan tarkastelemaan yleistä tapausta, jossa $k > 2$. Määritellään k indikaattori-vektoria $\mathbf{h}_j = (h_{1j}, \ldots, h_{nj})^{\mathrm{T}}$ siten, että

$$h_{ij} = \begin{cases} 1/\sqrt{w(C_j)}, & \text{jos } v_i \in C_j, \\ 0, & \text{muuten,} \end{cases} \tag{4.12}$$

missä $i = 1, \ldots, n$ ja $j = 1, \ldots, k$. Olkoon $H \in \mathbb{R}^{n \times k}$ matriisi, joka sisältää nämä k indikaattorivektoria sarakkeinaan. Yhtälön (4.12) nojalla pätee

$$\mathbf{h}_i^{\mathrm{T}} D \mathbf{h}_i = \sum_{j=1}^{n} d_j h_{ij}^2 = \frac{\sum_{v_j \in C_i} d_j}{w(C_i)} = \frac{w(C_i)}{w(C_i)} = 1.$$

Kaikki matriisin H sarakkeet ovat keskenään ortogonaalisia, joten $\mathbf{h}_i^{\mathrm{T}} D \mathbf{h}_j = 0$, jos $i \neq j$. Näin ollen $H^{\mathrm{T}} D H = I$. Vastaavalla tavalla kuin RatioCutin tapauksessa voidaan laskea, että

$$\mathbf{h}_i^{\mathrm{T}} L \mathbf{h}_i = \frac{cut(C_i, \overline{C_i})}{w(C_i)}.$$

Tämän seurauksena saadaan

$$\begin{aligned} \mathrm{NCut}(C_1, \ldots, C_k) &= \sum_{i=1}^{k} \frac{\mathrm{cut}(C_i, \overline{C_i})}{w(C_i)} \\ &= \sum_{i=1}^{k} \mathbf{h}_i^{\mathrm{T}} L \mathbf{h}_i \\ &= \mathrm{Tr}(H^{\mathrm{T}} L H). \end{aligned}$$

Näin ollen $\mathrm{NCut}(C_1, \ldots, C_k)$ minimointiongelma voidaan kirjoittaa uudelleen muodossa

$$\min_{C_1, \ldots, C_k} \mathrm{Tr}(H^{\mathrm{T}} L H), \text{ missä } H^{\mathrm{T}} D H = I \text{ ja } H \text{ on määritelty kuten ehdossa (4.12).}$$

Relaksoimalla diskreettisyysehto ja sijoittamalla $Q = D^{\frac{1}{2}} H$, saadaan relaksoiduksi ongelmaksi

$$\min_{\substack{Q \in \mathbb{R}^{n \times k} \\ Q^{\mathrm{T}} Q = I}} \mathrm{Tr}[Q^{\mathrm{T}} (D^{-\frac{1}{2}} L D^{-\frac{1}{2}}) Q] = \min_{\substack{Q \in \mathbb{R}^{n \times k} \\ Q^{\mathrm{T}} Q = I}} \mathrm{Tr}(Q^{\mathrm{T}} L_{\mathrm{sym}} Q),$$

joka eroaa yhtälöstä (4.11) vain sillä, että normalisoimattoman Laplacen matriisin sijaan siinä on normalisoitu Laplacen matriisi L_{sym}. Jälleen käyttämällä lausetta 4.11 ratkaisu saadaan valitsemalla matriisiksi Q sellainen matriisi, joka sisältää matriisin L_{sym} k ensimmäistä ominaisvektoria sarakkeinaan. Näin ollen päädytään algoritmin 6 mukaiseen menettelyyn. Uudelleen sijoittamalla $H = D^{\frac{1}{2}} Q$ ja käyttämällä lauseen 4.8 kohtaa 2 nähdään, että matriisi H koostuu matriisin L_{rw} k:sta ensimmäisestä ominaisvektorista. Tämä johtaa algoritmin 5 mukaiseen spektriseen klusterointiin.

5 Menetelmien soveltaminen sisätilapaikannuksessa

Tämän luvun alussa käydään läpi tyypillinen WLAN-sormenjälkipaikannuksen radiokartta rakenne ja tarkastellaan miten klusterointia on aiemmin käytetty WLAN-sormenjälkipaikannuksessa. Tämän jälkeen esitellään idea uudenlaiselle paikannusmenetelmälle, joka perustuu sormenjälkien välisten etäisyyksien arvioimiseen signaalien voimakkuuksien perusteella. Menetelmä edellyttää sormenjälkidatan klusterointia, johon luvussa tarkennutaan syvemmin. Luvussa vertaillaan viittä eri klusterointialgoritmia käyttäen erilaisia samanlaisuus- ja erilaisuusfunktioita. Vertailuissa käytetään tutkielmaa varten mitattuja aineistoja. Lisäksi esitellään uutta menetelmää havainnollistava esimerkki. Luvun lopuksi esitetään saadut tulokset ja analysoidaan näitä tuloksia.

5.1 Sisätilapaikannuksesta ja aiempi klusterointitutkimus

Vuosien saatossa sisätilapaikannuksen ongelmiin on kehitetty suuri määrä erilaisia tekniikoita. Monissa tekniikoissa on ongelmana kuitenkin se, että ne edellyttävät erilaisten vastaanottamien ja lähettimien asentamista, ja näin ollen menetelmät aiheuttavat suuria taloudellisia kustannuksia ja ovat infrastruktuurisesti hankalia toteuttaa. WLAN on noussut suosituksi menetelmäksi ratkoa sisätilapaikannuksen ongelmia, sillä langattomat tukiasemat kattavat suurelta osin nykypäivän rakennukset ja nykyaikaisiin kännyköihin ja muihin langattomiin vastaanottimiin on upotettuna antennit, joiden avulla voidaan mitata signaalien voimakkuusarvoja langattomista tukiasemista [32, 34]. WLAN-tekniikkaa ei kuitenkaan ole alun perin suunniteltu sisätilapaikannukseen, joten paikantaminen WLAN-signaalien avulla kohtaa monia eri haasteita. [16]

Yleisesti ottaen RSS-mittauksia hyödyntävät menetelmät voidaan jakaa kahteen eri kategoriaan; mallipohjaisiin (model-based, path-loss) ja mallivapaisiin (model-free, radiokartta) menetelmiin. Mallipohjaiset menetelmät käyttävät kerättyjä RSS-sormenjälkimittauksia optimoimaan ennalta määrätyn vaimenemismallin parametrien arvot. Radiokarttapohjaiset (Radio Map, RM) tekniikat, joita voidaan kutsua myös sormenjälkimenetelmiksi, käyttävät hyväkseen tukiasemien runsasta lukumäärää nykypäivän toimisto- ja kerrostalorakennuksissa. WLAN-paikannuksessa RSS-sormenjälkien ominaisuudet riippuvat suurelta osin mittauspisteen ympäristön ominaisuuksista ja alueella olevista tukiasemista. Näin ollen tällaisilla alueilla tukiasemista saatujen signaalien voimakkuusarvojen joukot muodostavat pääsääntöisesti yksilöllisiä sormenjälkiä eri sijainneissa. [16]

Tyypillisesti WLAN-sormenjälkimenetelmät koostuvat offline- ja online-aikaisista menetelmistä. Offline-vaiheessa lähdetään liikkeelle radiokartan muodostamisesta. Ensimmäiseksi tarkasteltava alue jaetaan pohjapiirustusta apuna käyttäen soluihin. Tämän jälkeen mitataan signaalien voimakkuudet, jolloin jokaisen solun sisällä suoritetaan mittaus referenssipisteestä tietyn aikavälin ajan. Mittauksissa tallennetaan tukiasemista välittyvien radiosignaalien voimakkuusarvot kussakin referenssipisteessä tiettynä ajanhetkenä. Näin mitatuista sormenjäljistä muodostuva tietokanta muodostaa radiokartan koko tarkasteltavasta alueesta. Tietokanta sisältää siis kunkin solun referenssipisteen koordinaattiarvot, havaitut signaalien voimakkuusarvot eri tukiasemista sekä aikaleiman kullekin suoritetulle mittaukselle. Online-vaiheessa paikannettavan käyttäjän kussakin sijainnissa havaitsemia signaalien voimakkuusarvoja verrataan algoritmien avulla radiokartan eri referenssiarvoihin ja pyritään näin löytämään radiokartasta samanlaisimmat sormenjäljet. Tämän jälkeen samanlaisimpien referenssipisteiden avulla pyritään estimoimaan

käyttäjän sen hetkinen sijainti. [14, 16]

Käyttäjän sijainnin arviointi voidaan suorittaa joko verkko- tai laitekeskisesti. Verkkokeskisessä toteutustavassa käyttäjän mittaamat signaalien voimakkuusarvot lähetetään verkon kautta palvelimelle, jossa sormenjälkitietokannan perusteella lasketaan käyttäjän sijainnille estimaatti. Tämän jälkeen sijainnin estimaatti lähetetään takaisin käyttäjälle. Menetelmän etuna on se, että se ei vaadi suurta tiedonsiirtoa laitteen ja verkon välillä. Laitekeskisessä menetelmässä osa sormenjälkitietokannasta lähetetään palvelinpuolelta laitteeseen ja käyttäjän sijainnin estimaatti lasketaan käyttäjän laitteella. Menetelmä edellyttää offline-vaiheessa suuremman datamäärän siirtämistä verkon ja laitteen välillä. Laitekeskinen menetelmä kuitenkin tarjoaa käyttäjälle enemmän yksityisyyttä, sillä toisin kuin verkkokeskisessä menetelmässä, siinä ei siirretä käyttäjän sijainnin estimaattia verkon ja käyttäjän välillä. Lisäksi laitekeskinen menetelmä ei edellytä jatkuvaa Internet-yhteyttä, ja se toimii pienemmällä viiveellä, sillä sijainnin laskeminen suoritetaan käyttäjän laitteessa. [5]

Paikannukseen tarvittavan datansiirron minimoimiseksi laitteen ja verkon välillä tulee siirrettävän sormenjälkitietokannan kokoa pienentää. Lisäksi tavoiteltaessa matala kompleksista ja nopeaa paikannustulosta vertailtavan tietokannan koon tulee olla riittävän pieni. Datan koon pienentämiseen voidaan käyttää klusterointia. Radiokartan klusterointi voidaan suorittaa joko koordinaatti tai signaalin voimakkuus keskisesti. Artikkelissa [5] on vertailtu näitä kahta menetelmää keskenään. Käytetyimmät klusterointialgoritmit signaalin voimakkuus keskisessä menetelmässä perustuvat k-means-algoritmiin ja sen eri variaatioihin sekä affinity propagation -klusterointialgoritmiin [5]. Artikkelissa [16] on listattu eri klusterointialgoritmeja sormenjälkitietokannan koon pienentämiseksi. Tyypillisesti radiokartan klusterointi alkaa sopivan samanlaisuus-/erilaisuusmitan valinnalla. Valitun funktion avulla määritetään kaikkien sormenjälkien välille jokin samanlaisuus- tai erilaisuusarvo. Tämän jälkeen sormenjäljet klusteroidaan valitulla algoritmilla käyttäen näitä arvoja. Klusteroinnin seurauksena jokainen sormenjälki kuuluu johonkin tiettyyn klusteriin. Tämän lisäksi klusterointi määrittää jokaiselle klusterille jonkin edustaja-alkion. Nyt sen sijaan, että käyttäjän saamia signaalien voimakkuusarvoja verrattaisiin referenssi tietokannan kaikkiin eri alkioihin, saatuja arvoja verrataan klustereiden edustaja-alkioihin ja valitaan klustereista se, jonka arvot ovat lähimpänä käyttäjän saamia signaalien voimakkuuksia. Näin säästetään runsaasti laskenta aikaa, sillä vertailtavien alkioiden lukumäärä pienenee huomattavasti. Tämän karkean paikannuksen jälkeen voidaan suorittaa käyttäjän paikan tarkempi estimointi esimerkiksi KNN- tai WKNN-algoritmeilla. [5, 16]

Affinity propagation -klusterointialgoritmia on sovellettu RSS-sormenjälkien klusterointiin esimerkiksi artikkeleissa [5, 11]. Artikkelissa [22] on sormenjälkiä klusteroitu k-means-algoritmilla estimoitaessa käyttäjän sen hetkistä kerrosta. k-means-algoritmia on sovellettu myös esimerkiksi artikkeleissa [27, 37]. Spektristä klusterointialgoritmia sormenjälkien klusterointiin on puolestaan käytetty muun muassa artikkelissa [21].

Klusteroinnin ja sen jälkeisen tarkemman paikannuksen lopputulokseen saattaa merkittävästi vaikuttaa valitun samanlaisuus-/erilaisuusfunktion valinta. Kirjallisuudessa on tutkittu laajasti eri funktioita RSS-sormenjälkien välisten samanlaisuuksien ja erilaisuuksien määrittämiseksi. Artikkelissa [29] vertaillaan 51:tä eri samanlaisuus- ja erilaisuusfunktiota. Sormenjälkien välisten samanlaisuuksien ja erilaisuuksien laskemiseen on myös kehitetty omia funktioita. Esimerkiksi artikkelissa [5] esitellään muunnelma logaritmisesta Gaussisesta samanlaisuusfunktiosta ja artikkelissa [36] esitellään erilaisuusmitta, joka ottaa huomioon havaittujen tukiasemien vaihtelevan määrän eri mittauspisteissä. Wang ja muut käyttävät artikkelissaan [30] sormenjälkien samanlaisuuksien laskemiseen samanlaisuusfunktiota, joka painottaa suuria lähekkäisiä arvoja voimakkaammin kuin pieniä. Tätä funktioita käytetään myös tämän tutkielman testauksissa.

5.2 Tutkimusongelman esittely

Edellä kuvatun WLAN-sisätilapaikannusmenetelmän yhtenä heikkoutena on se, että radiokartan muodostaminen on aikaa vievä prosessi. Tämä johtuu siitä, että radiokartan mittaajan tulee kantaa dataa tallentavaa laitetta jokaiseen referenssipisteeseen ja tallentaa signaalien voimakkuusmittauksia tietyn aikaa aina kussakin referenssipisteessä. Lisäksi tukiasemien paikat saattavat ajan saatossa vaihtua, ja osa tukiasemista voidaan poistaa kokonaan käytöstä sekä uusia tukiasemia voidaan lisätä. Näin ollen mitattujen radiokarttojen arvot saattavat muuttua ajan myötä, ja niiden päivittäminen luo omat haasteensa. [34]

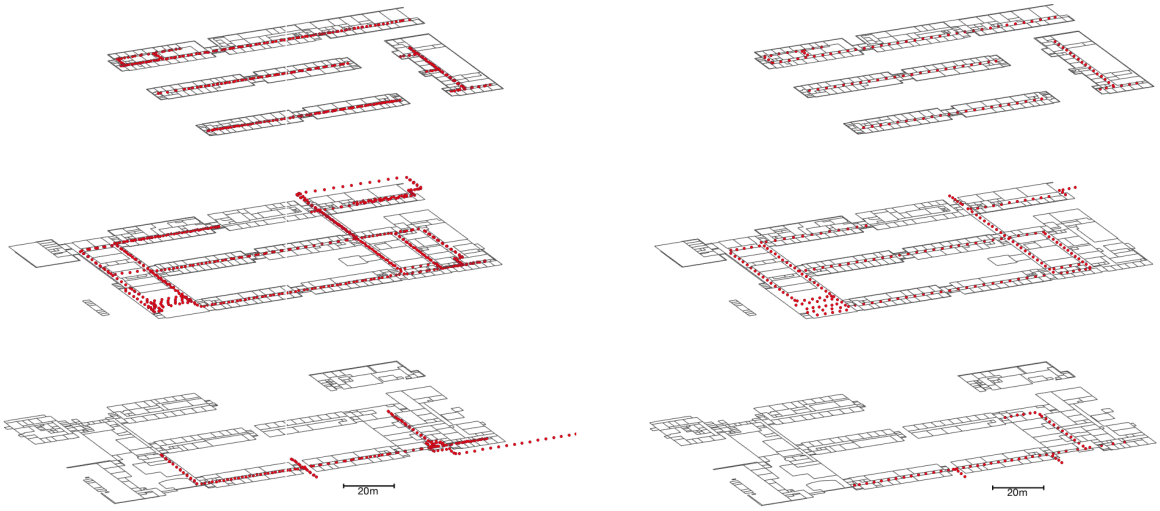
Mikäli mitattujen RSS-sormenjälkihavaintojen keskinäisiä etäisyyksiä pystyttäisiin estimoimaan riittävällä tarkkuudella, tämä mahdollistaisi yhdessä ajoittaisten GPS-havaintojen avulla sen, että radiokartta voitaisiin luoda pelkän signaalin voimakkuusdatan perusteella. Käyttäjän paikan estimoinnin lisäksi kerätyn datan perusteella kyettäisiin luomaan kuljetusta alueesta karttarakenne. Paikannusmenetelmää, jossa paikannuksen lisäksi pyritään muodostamaan alueesta karttarakenne, kutsutaan SLAM-menetelmäksi (*Simultaneous Location and Mapping, SLAM*) [10]. Edellä kuvattu paikannusmenetelmä sisältää monia eri haasteita. Suurin haasteista on signaalien voimakkuusarvojen muuttaminen sormenjälkien välisiksi etäisyyksiksi. Kun mittauspisteiden etäisyydet kasvavat, niiden yhteisten tukiasemien lukumäärä pienenee ja etäisyyksien arviointi hankaloituu. Tämä ongelma voidaan ratkaista sillä, että ennen kuin mittauspisteiden välisiä etäisyyksiä lähdetään arvioimaan, klusteroidaan mittauspisteet niissä havaittujen signaalien voimakkuusarvojen perusteella. Tämän jälkeen sormenjälkien välisiä etäisyyksiä voidaan arvioida klustereittain. Tässä tutkielmassa vertaillaan eri klusterointimenetelmiä, jotka jakavat tarkasteltavat mittauspisteet pienempiin osajoukkoihin. Tässä tutkielmassa ei syvennytä sormenjälkien välisten etäisyyksien laskemiseen. Klusteroinnin ja etäisyyksien estimoinnin jälkeen klusterit tulee lisäksi yhdistää. Tämä luo myös omat haasteensa, eikä tässä tutkielmassa paneuduta tähän ongelmaan sen tarkemmin.

RSS-sormenjälkiä klusteroitaessa kohdataan monia eri haasteita. Signaalien voimakkuusarvot heikkenevät seinien, huonekalujen, ihmisten ja muiden rakennuksissa olevien objektien johdosta, mikä aiheuttaa epäjohdonmukaisuutta mittaustuloksiin [3]. Lisäksi mittaukset saattavat sisältää puuttuvia arvoja. Erityisesti kuuluvuus alueen rajalla signaali saatetaan havaita heikkona tai yhtä hyvin se saatetaan olla havaitsematta. Käsiteltävä aineisto sisältää näin ollen paljon kohinaa, mikä tuo omat haasteensa klusterointiin. Tukiasemien lukumäärät vaihtelevat alueittain, mikä saattaa aiheuttaa alueellisia eroja mittauspisteiden välisiin samanlaisuuksiin. Näihin haasteisiin voidaan vaikuttaa klusterointialgoritmin valinnalla, kuin myös sopivan samanlaisuusfunktion valinnalla. Eri ongelmat heijastuvat myös kunkin klusterointialgoritmin parametrien valintaan. Tutkimuksen testejä tehdessä osoittautui, että on haastavaa löytää yleisiä parametreja jollekin klusterointialgoritmille niin, että algoritmi toimisi halutulla tavalla kaikissa olosuhteissa ja kaikilla aineistoilla. Klustereiden kokoa voidaan tarkastella sekä klustereihin kuuluvien sormenjälkien lukumäärän, että niiden käsittämän pinta-alan perusteella. Näistä jälkimmäinen on tutkielman tavoitteiden kannalta mielenkiintoisempi. Tässä tutkielmassa ei ole syvennytty optimaalisen klusterikoon etsimiseen, sillä siihen vaikuttaa myös esimerkiksi sormenjälkien etäisyyksien arviointiin liittyvät tekijät.

5.3 Tutkimusaineisto

Työssä esiteltyjä algoritmeja ja samanlaisuusmittoja testataan Tampereen yliopiston Hervannan kampuksen Sähkötalo-rakennuksesta kerätyillä aineistoilla. WLAN-signaalien voimakkuusarvoista muodostuvia sormenjälkiä kerättiin kolmesta eri kerroksesta HERE Indoor Radio Map-

per -sovelluksella, joka oli asennettuna Nexus-puhelimeen. Ennen mittausten aloittamista sovellukseen ladattiin rakennuksen pohjapiirustus kustakin kerroksesta. Sovellus toimii siten, että käyttäjä merkitsee sijaintinsa kartalle ja aloittaa signaalien voimakkuusarvojen mittaamisen. Samanaikaisesti käyttäjän tulee lähteä liikkeelle ja kulkea suoraa reittiä tasaisella nopeudella. Kun käyttäjä haluaa pysähtyä, hän lopettaa tallentamisen ja merkitsee kartalle sen hetkisen sijaintinsa. Tämän jälkeen sovellus interpoloi reitin aikana tehdyt signaalien voimakkuusmittaukset lähtö- ja loppupisteen välille. Kuvassa 5.1 näkyy tässä tutkimuksessa käytetyt aineistot.



(a) Aineisto 1

(b) Aineisto 2

Kuva 5.1: Tutkimuksissa käytettyjen aineistojen mittauspisteet kartalla.

Kerättyjen sormenjälkihavaintojen lukumäärät kerroksittain on taulukoituna taulukossa 5.1. Mittausteknisistä syistä sormenjälkiä ei ole kerätty kerrosten väliltä, kuten esimerkiksi portaista.

Taulukko 5.1: Sormenjälkien lukumäärät aineistoissa.

	Kerros 1	Kerros 2	Kerros 3	Yhteensä
Aineisto 1	194	642	380	1216
Aineisto 2	64	201	116	381

Testauksissa käytetyt aineistot eroavat rakenteeltaan toisistaan, mikä heijastuu myös klusterointiin. Ensimmäisissä kerroksissa ulkoa mitattuja sormenjälkiä on enemmän aineistossa 1 kuin aineistossa 2. Vaikka ulkoa mitatut sormenjäljet onkin kerätty rakennuksen välittömästä läheisyydestä, niin silti näissä sormenjäljissä havaittujen tukiasemien lukumäärät ovat pienempiä, ja signaalien voimakkuusarvot ovat heikompia kuin sisätiloista mitatuissa sormenjäljissä. Tämä saattaa joissakin tilanteissa johtaa siihen, että vaikka kahden ulkoa mitatun mittauspisteen välinen etäisyys on suuri, niin niiden välinen samanlaisuusarvo on verrattain korkea. Rakennuksen ulkopuolelta mitattujen mittauspisteiden mahdollisia GPS-havaintoja voitaisiin hyödyntää estämään epäyhtenäisten klustereiden muodostuminen. Rakennuksen kolmas kerros jakaantuu selvästi irrallisiin osiin, mikä helpottaa hyvän klusteroinnin löytämistä. Suurin ero aineistojen 1 ja 2 välillä on mittauspisteiden tiheydellä. Aineistossa 1 yksittäiseltä käytävältä on pääsääntöi-

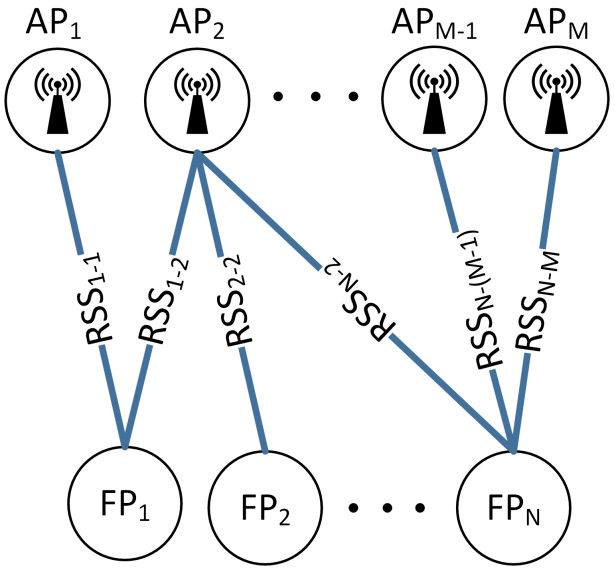
sesti suoritettu mittauksia useammin kuin kerran, kun taas aineistossa 2 yksittäiseltä käytävältä on kerätty mittauksia pääsääntöisesti vain kerran.

Tutkimuksessa kerätyistä mittauspisteistä on tallennettu kunkin pisteen sijainti kartalla ja tämän lisäksi kaikkien pisteestä havaittujen tukiasemien yksilöidyt MAC-osoitteet ja signaalien voimakkuusarvot. Yksittäinen RSS-arvo kuvastaa tietyn vastaanotetun WLAN-radiosignaalin voimakkuutta ja se ilmaistaan desibeleinä (dBm). Signaalien voimakkuusarvot ovat negatiivisia kokonaislukuja. Mitä suurempi havaittu signaalin voimakkuusarvo on, sitä voimakkaampana signaali havaitaan. Voimakkain signaalin voimakkuusarvo aineistossa 1 on −23dBm ja aineistossa 2 voimakkain arvo on −24dBm. Aineiston analysoinnin helpottamiseksi arvot on muutettu positiivisiksi seuraavalla tavalla

$$RSS_{pos}(x) = \begin{cases} RSS_i - (\min -1), & \text{jos WAP}_i \text{ esiintyy mittauspisteessä } x, \\ 0, & \text{muuten,} \end{cases}$$

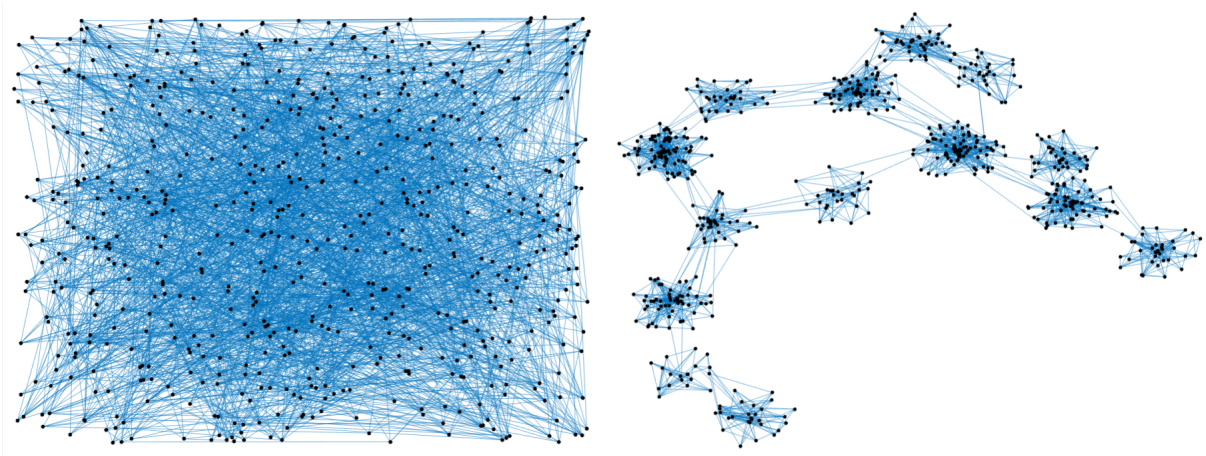
missä i on tukiaseman identifioiva indeksi ja min arvo on aineiston heikoin havaittu signaalin voimakkuusarvo. Tukiasemat yksilöivät MAC-osoitteet mahdollistavat signaalien voimakkuusarvojen vertailemisen. Kuultujen eri MAC-osoitteiden lukumäärä aineistossa 1 on 1057 ja aineistossa 2 määrä on 1006.

Tarkastellaan seuraavassa kerättyä aineistoa graafiteoreettisessa mielessä. Kerätystä sormenjälkidatasta voidaan muodostaa graafi, jonka solmut koostuvat joukosta mittauspisteitä ja joukosta tukiasemia. Särmän paino solmujen välillä ilmaisee mittauspisteessä tukiasemasta saadun signaalin voimakkuuden arvon, mikäli mittauspisteestä on havainto. Mikäli mittauspisteessä ei ole havaittu tukiaseman signaalia, ei solmujen välillä ole särmää. Tukiasemasta ei voi olla särmää toiseen tukiasemaan, ja vastaavasti mittauspisteestä toiseen mittauspisteeseen ei voi olla särmää. Näin ollen graafi on kaksijakoinen. Tilannetta on havainnollistettu kuvassa 5.2. Klusterointitehtävänä on luokitella mittauspisteet, joilla on samanlaisia havaintoja eri tukiasemista samaan klusteriin. Merkitään mittauspisteiden joukkoa kirjaimella A ja vastaavasti tukiasemien joukkoa kirjaimella B. Nyt tarkoituksena on klusteroida mittauspisteiden joukkoa, joten klusterointi suoritetaan johdetulle graafille G_A. Käyttämällä valittua samanlaisuus-/etäisyysfunktiota voidaan nyt kaksijakoisesta graafista muodostaa johdettu graafi, jossa solmujen (sormenjälkien) väliset



Kuva 5.2: Kaksijakoinen graafi, joka koostuu tukiasema-solmuista ja sormenjälki-solmuista ja niitä yhdistävistä signaalien voimakkuusarvoista.

särmät kuvaavat niiden välisiä samanlaisuuksia/erilaisuuksia. Tämän prosessin avulla mitatusta datasta saadaan muodostettua samanlaisuusgraafi. Näin syntynyttä samanlaisuusgraafia voidaan muokata edelleen aliluvussa 3.3.2 esitellyillä menetelmillä. Kuvassa 5.3 on esitetty tutkielman aineiston 1 toisesta kerroksesta mitattujen mittauspisteiden muodostama 5-lähimmän naapurin samanlaisuusgraafi. Kuvassa 5.3a solmujen sijainti on satunnainen, kun taas kuvassa 5.3b graafin solmut on ryhmitelty klusteroinnin jälkeen niin, että aina saman klusterin solmut sijaitset lähellä toisiaan. Kuvasta havaitaan, että k-lähimmän naapurin graafissa särmät ovat keskittyneet klustereiden sisälle ja klustereiden välillä särmien lukumäärä on selvästi pienempi.



(a) k-lähimmän naapurin graafi

(b) Ryhmitelty k-lähimmän naapurin graafi

Kuva 5.3: Ryhmittelemätön ja ryhmitelty 5-lähimmän naapurin samanlaisuusgraafi.

5.4 Tutkimusmenetelmät

Työssä tehtävät testaukset voidaan jakaa kahteen eri kategoriaan. Ensiksi aineisto jaetaan kerroksittain ja klusterointi suoritetaan kunkin kerroksen aineistolle erikseen. Tässä vaiheessa klusteroidaan siis ainoastaan sellaisia sormenjälkiä, jotka ovat kaikki samassa kerroksessa. Tästä vaiheesta käytetään nimitystä 2D-klusterointi. Toisessa kategoriassa klusterointialgoritmeja käytetään koko data-aineistolle, jolloin sormenjälkiä on kolmessa eri kerroksessa. Tästä vaiheesta käytetään nimitystä 3D-klusterointi. Klusterointia testataan viidellä eri klusterointialgoritmilla, jotka ovat normalisoimaton spektrinen, normalisoitu spektrinen (Shi ja Malik), Markov, k-means ja affinity propagation. Algoritmien toimintaperiaatteet on esitelty luvussa 4. Kaikkia algoritmeja testataan viidellä eri samanlaisuus-/erilaisuusfunktiolla, lukuun ottamatta k-means-algoritmia, jolla suoritetaan testaukset ainoastaan kahdella eri erilaisuusfunktiolla. Testauksissa käytetyt eri samanlaisuus- ja erilaisuusfunktiot on eritelty taulukossa 5.2. Logaritmista Gaussista samanlaisuusfunktiota käytettäessä varianssin σ arvoksi on asetettu kaikissa testauksissa 5dBm. Arvoa hienosäätämällä voidaan vaikuttaa samanlaisuusarvoihin ja lopulliseen klusterointitulokseen [5]. Seuraavassa käydään algoritmikohtaisesti läpi eri testauksissa käytetyt asetukset sekä 2D, että 3D tapauksissa.

Käydään ensin läpi 2D-testauksissa käytetyt parametrit. Sekä normalisoimattomassa että normalisoidussa spektrisessä klusteroinnissa on käytetty samoja parametreja. Samanlaisuusgraafina on käytetty kaikilla metriikoilla 10-lähimmän naapurin painotonta samanlaisuusgraafia.

Taulukko 5.2: Testauksissa käytetyt samanlaisuus-/erilaisuusfunkiot.

	Euklidinen	Gaussinen	Kosini	Czekanowski	Wang
Spektrinen*	$s_{\text{Euklidinen}}$	$s_{\text{Gaussinen}}$	s_{kosini}	$s_{\text{Czekanowski}}$	s_{Wang}
MCL*	$s_{\text{Euklidinen}}$	$s_{\text{Gaussinen}}$	s_{kosini}	$s_{\text{Czekanowski}}$	s_{Wang}
k-means	$d^2_{\text{Euklidinen}}$	-	d_{kosini}	-	-
AP	$s_{\text{Euklidinen}}$	$s_{\text{Gaussinen}}$	$-1 + s_{\text{kosini}}$	$-1 + s_{\text{Czekanowski}}$	$-1 + s_{\text{Wang}}$

* Samanlaisuusgraafia on muokattu ennen klusteroinnin suorittamista.

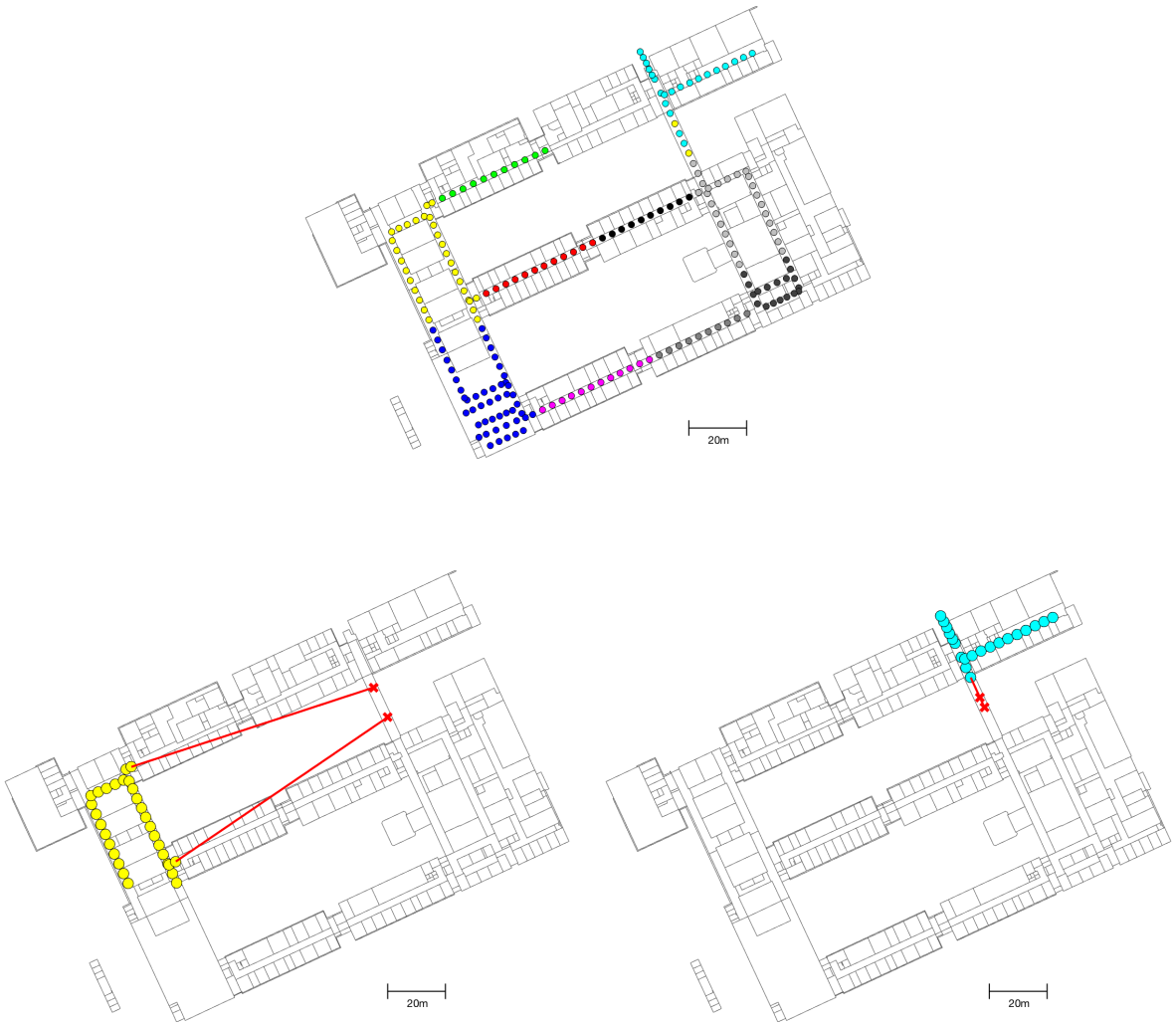
Graafi on muodostettu siten, että aluksi alkioiden väliset samanlaisuudet on laskettu samanlaisuusfunktiolla, minkä jälkeen on muodostettu 10-lähimmän naapurin samanlaisuusgraafi. Tämä graafi on lopuksi muutettu painottomaksi graafiksi poistamalla jäljelle jääneistä särmistä painot. Särmien painojen poistamisella voidaan tasoittaa samanlaisuuksissa ilmeneviä alueellisia eroja. Lopullisen klusteroinnin suorittamiseen algoritmi hyödyntää Matlab-ohjelman valmista k-means-klusterointialgoritmia käyttäen etäisyysmetriikkana neliöityä Euklidista etäisyyttä $d^2_{\text{Euklidinen}}$. Lukuun ottamatta klustereiden lukumäärän määräävää parametria, algoritmeissa on käytetty samoja parametreja sekä aineistossa 1, että aineistossa 2. Markov-algoritmissa samanlaisuusgraafina on myös käytetty 10-lähimmän naapurin samanlaisuusgraafia, ja se on muutettu painottomaksi graafiksi vastaavalla tavalla kuin spektrisen klusteroinnin tapauksessa. Algoritmin parametreiksi on valittu aineistossa 1 arvot $e = 5$ ja $r = 2$ ja aineistossa 2 arvot $e = 3$ ja $r = 2$. k-means-algoritmi on suoritettu Matlab-ohjelman valmiilla toteutuksella ja iterointikierroksien lukumääräksi on asetettu 50. Affinity propagation -algoritmissa vaimennuskertoimeksi on valittu 0.9 ja jokaiselle mittauspisteelle on asetettu sama preferenssiarvo, joka on samanlaisuusfunktiosta riippuen, aineistossa 1 joko 7 tai 8 ja aineistossa 2 joko 3 tai 4 kertaa kaikkien samanlaisuusarvojen mediaani. Algoritmia iteroidaan kaikissa tapauksissa 200 kierrosta. Algoritmien palauttamien klustereiden lukumääriin vaikuttavat tekijät on koottu taulukkoon 5.3. Spektrisille algoritmeille ja k-means-algoritmille klustereiden lukumäärä annetaan parametrina, mutta Markov- ja affinity propagation -algoritmeissa klustereiden lukumäärää ei valita, vaan sitä voidaan säädellä algoritmien parametrien avulla.

Taulukko 5.3: Klustereiden lukumäärien säätely eri algoritmeissa.

	Klustereiden lukumääriin vaikuttavat tekijät
Spektrinen	määrätään parametrilla k
MCL	säädellään parametreilla e ja r
k-means	määrätään parametrilla k
AP	säädellään preferenssiarvoilla p

Klusteroitaessa koko aineistoa kerralla käytetään testauksissa lähes samoja parametreja mitä 2D-klusteroinnissa. Spektrisillä algoritmeilla parametrit ovat muuten samat, mutta klustereiden lukumäärän määrääväksi parametriksi on asetettu aineistossa 1 $k = 30$ ja aineistossa 2 $k = 24$. MCL algoritmilla klustereiden lukumäärää on säädelty parametreilla e ja r asettaen aineistossa 1 arvot $e = 5$ ja $r = 2$ ja aineistossa 2 arvot $e = 2$ ja $r = 2$. Muuten algoritmin asetukset ovat samat kuin 2D-vaiheessa. k-means-algoritmilla klustereiden lukumäärän määrääväksi parametriksi on annettu aineistossa 1 $k = 30$ ja aineistossa 2 $k = 24$. Myös affinity propagation -algoritmilla asetukset ovat muuten samat kuin 2D-vaiheessa, mutta aineistossa 1 preferenssiarvot ovat 4 tai 5

kertaa kaikkien samanlaisuusarvojen mediaani, samanlaisuusfunktiosta riippuen ja aineistossa 2 preferenssiarvot ovat 2 kertaa kaikkien samanlaisuusarvojen mediaani.

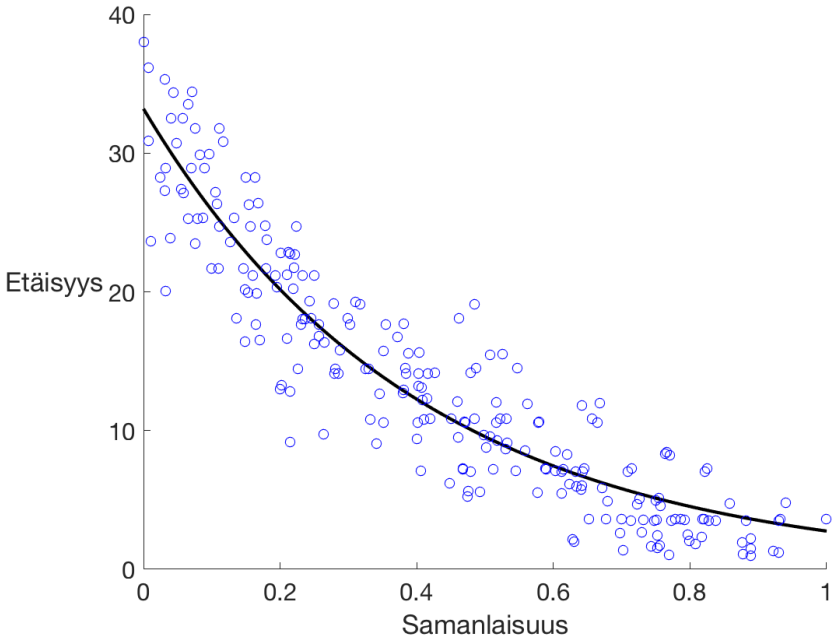


Kuva 5.4: Ylläolevan kuvan klusteroinnissa kahdessa eri klusterissa havaitaan poikkeavia solmuja. Alemmissa kuvissa näiden kahden klusterin poikkeavat solmut on merkitty punaisilla risteillä ja ne on yhdistetty niiden lähimpiin ydin-solmuihin punaisilla viivoilla.

Saatuja tuloksia 2D-klusteroinnissa vertaillaan seuraavanlaisella menetelmällä. Kaikista muodostuneista klustereista muodostetaan oma täydellinen graafi, joissa solmuina toimivat mittauspisteet ja särmien painot vastaavat mittauspisteiden välisiä keskinäisiä todellisia etäisyyksiä metreinä. Kunkin klusterin muodostamaa omaa graafia käsitellään kutakin erikseen. Muodostetuille graafeille määritetään pienin virittävä puu *Primin algoritmilla* [28, s. 326]. Näistä virittävistä puista poistetaan kaikki särmät, joiden paino ylittää ennalta määrätyn raja-arvon. Aineistossa 1 arvoksi on asetettu 4 metriä ja aineistossa 2 rajana on 5.5 metriä. Aineistoon 1 on valittu pienempi raja-arvo johtuen siitä, että aineistossa on sormenjälkihavaintoja tiheämmin. Särmien poiston jälkeen muodostuneista metsistä valitaan solmujen lukumäärältään suurimmat komponentit. Näitä komponentteja nimitetään klustereiden ytimiksi. Ytimien ulkopuolelle jääviä solmuja kutsutaan poikkeaviksi solmuiksi. Tämän jälkeen kullekin poikkeavalle solmulle lasketaan lyhin etäisyys klusterin mihin tahansa solmuun, joka kuuluu klusterin ytimeen. Kullekin klusterille

lasketaan ytimiin kuuluvien alkioiden lukumäärä ja poikkeavien alkioiden etäisyydet ytimistä. 2D-vaiheessa tulokset kootaan kerroksittain ja lasketaan klustereiden ytimiin kuuluvien alkioiden prosentuaalinen osuus sekä keskiarvo poikkeavien alkioiden etäisyyksistä ytimiin. Kuvassa 5.4 on havainnollistettu tätä virheiden laskentamenetelmää. Samaa menetelmää käytetään myös vertailtaessa 3D-klusteroinnin tuloksia. Erona 2D-klusterointiin, 3D-klusteroinnissa yksi klusteri saattaa sisältää sormenjälkihavaintoja useammasta kuin yhdestä kerroksesta. Laskettaessa sormenjälkien välisiä todellisia etäisyyksiä 3D-vaiheessa ei kuitenkaan oteta huomioon pystysuunnassa ilmenevää etäisyyttä, vaan tarkastellaan ainoastaan eroja pituus- ja leveyssuunnassa. Tähän menettelyyn on päädytty, sillä testaukset osoittivat, että sormenjälkien klusterointi eri kerroksiin on erityisen haasteellista pelkkien WLAN-signaalien voimakkuusarvojen perusteella. Tämä ongelma korostuu erityisesti avoimemmissa tiloissa. Tarkemman kerrosinformaation saamiseksi apuna voidaan käyttää esimerkiksi barometrisensorista saatavaa informaatiota [18].

Edellä kuvattujen vertailujen lisäksi suoritetaan myös menetelmän kokonaiskuvaa havainnollistava esimerkki. Esimerkki suoritetaan aineiston 1 mittausdatalla ja klustereiden muodostamiseen käytetään 2D-klusterointia. Menetelmässä määritetään kullekin muodostuneelle klusterille epälineaarisella pienimmän neliösumman menetelmällä funktio $f(x) = ae^{bx}$, joka kuvaa kahden sormenjälkihavainnon välisen samanlaisuuden sormenjälkien väliseksi etäisyydeksi metreinä. Tässä samanlaisuusfunktiona käytetään kaikissa tilanteissa Wang-samanlaisuusfunktiota. Kuvassa 5.5 on esimerkki yhden klusterin samanlaisuus-etäisyys -aineistoon sovitetusta käyrästä. Optimoitujen funktioiden avulla voidaan samanlaisuusarvot muuntaa etäisyyksiksi metreinä ja



Kuva 5.5: Esimerkki funktion sovituksesta klusterin sormenjälkien välisten samanlaisuuksien ja etäisyyksien väliseen aineistoon.

saada kussakin klusterissa kaikkien klusteriin kuuluvien sormenjälkien väliset parittaiset etäisyysarviot metreinä. Kun sormenjälkien välisille etäisyyksille on saatu arviot, voidaan käyttää moniulotteista skaalausta (Multidimensional scaling, MDS) muodostamaan kunkin klusterin muoto. Lopuksi kullekin klusterille tulee suorittaa vielä Euklidinen muunnos, joka tehdään Matlab-ohjelman valmiilla procrustes-funktiolla, jolle annetaan parametrina kaikkien sormenjälkien todelliset sijainnit sekä moniulotteisesta skaalauksesta saadut arvot. Moniulotteisen skaa-

lauksen arvoille funktio laskee optimaalisen rotaation, peilauksen, siirron ja skaalauksen, kun ne suhteutetaan sormenjälkien todellisiin sijainteihin. Lopputuloksena saadaan arvio kunkin sormenjäljen todellisesta sijainnista. Muunnoksen jälkeen kunkin algoritmin tuottamille arvoille lasketaan virheiden keskiarvo. Menetelmässä käytetään sormenjälkien välisiä todellisia etäisyyksiä eikä sen näin ollen ole tarkoitus toimia täysin realistisena esimerkkinä, vaan antaa kokonaiskuva paikannusideasta.

5.5 Tulokset ja analysointi

Käydään seuraavaksi läpi testauksissa saadut tulokset ja analysoidaan niitä. Ensimmäiseksi käydään läpi 2D-klusteroinnin tulokset, ja sen jälkeen 3D-klusteroinnin tulokset. Tämän jälkeen vertaillaan 2D-klusteroinnin ja 3D- klusteroinnin tuloksia keskenään. Lopuksi käydään vielä läpi menetelmää havainnollistavasta esimerkistä saatuja arvoja. Testauksissa saadut tulokset on taulukoitu aineisto- ja menetelmäkohtaisesti. Taulukoissa merkintä spektrinen 1 viittaa normalisoimattomaan spektriseen klusterointiin ja merkintä spektrinen 2 viittaa normalisoituun spektriseen klusterointiin. 2D-klusteroinnin tapauksessa tulokset on esitelty kerroksittain ja 3D-testauksissa koko aineiston osalta. Taulukoihin on koottu klustereiden ytimiin kuuluvien sormenjälkien prosentuaalinen osuus sekä poikkeavien sormenjälkien etäisyyksien keskiarvot lähimpiin ydinalkioihin metreinä. Klusterointivirheestä puhuttaessa tarkoitetaan sillä seuraavassa ytimien ulkopuolelle jäävien alkioiden suhteellista osuutta sekä niiden etäisyyksien keskiarvojen suuruutta metreinä.

Koska tutkielman klusterointitulosten vertailu perustuu menetelmään, joka arvioi klustereiden yhtenäisyyttä, on tarpeen tarkastella minkä kokoisia saadut klusterit ovat. Klusterit saattavat olla täysin yhtenäisiä, mutta niiden alkioiden lukumäärät voivat vaihdella merkittävästi eri klustereiden välillä. Paikannusmenetelmän kannalta olisi toivottavaa, että klusterit kattaisivat pinta-alaltaan mahdollisimman samansuuruisia alueita. Klustereiden kokoja tarkastellaan niistä kerättyjen tunnuslukujen avulla ja ne on taulukoitu tapauskohtaisesti. Myös klustereiden lukumäärät eri tapauksissa on taulukoitu, ja ne löytyvät liitteiden taulukoista 6.1, 6.2, 6.3 ja 6.4.

Tarkastellaan ensimmäiseksi aineistolla 1 saatuja klusterointituloksia. Saadut tulokset on koottu taulukkoon 5.4. Ensimmäisestä kerroksesta kerätyn datan klusterointia hankaloittaa se, että osa sormenjäljistä on mitattu rakennuksen ulkopuolelta useammasta eri sijainnista. Näin ollen, vaikka mittauspisteiden etäisyydet ovatkin suuria, niin sormenjälkien samanlaisuudet ei-

Taulukko 5.4: Virheet aineistossa 1, 2D-klusterointi.

		Euklidinen		Gaussinen		Kosini		Czekanowski		Wang	
		ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}(m)$	ydin(%)	$\bar{e}$ (m)
	Spektrinen 1	88.14	72.96	86.08	74.13	96.91	19.91	99.48	42.48	98.97	23.52
	Spektrinen 2	88.66	71.49	89.69	68.43	96.91	19.91	98.97	23.52	98.97	23.52
Kerros 1	MCL	80.41	86.64	78.35	82.59	97.42	22.19	99.48	42.48	99.48	42.48
	k-means	90.72	62.67	NA	NA	91.24	65.3	NA	NA	NA	NA
	AP	79.9	80.39	70.1	66.25	97.42	22.19	98.45	30.7	88.66	27.39
	Spektrinen 1	99.22	28.33	99.69	62.84	99.22	26.96	100.0	-	100.0	-
	Spektrinen 2	99.22	28.33	99.69	62.84	99.53	42.81	100.0	-	97.66	28.76
Kerros 2	MCL	93.46	23.3	97.2	21.78	99.38	38.58	98.91	16.89	99.84	15.5
	k-means	99.69	8.078	NA	NA	97.82	15.23	NA	NA	NA	NA
	AP	92.37	40.54	90.97	53.99	92.99	24.82	92.52	23.08	91.12	30.99
	Spektrinen 1	100.00	-	100.00	-	99.74	4.84	100.00	-	100.00	-
	Spektrinen 2	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-
Kerros 3	MCL	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-
	k-means	100.00	-	NA	NA	100.00	-	NA	NA	NA	NA
	AP	100.00	-	100.00	-	100.00	-	100.00	-	99.74	9.03

Taulukko 5.5: Klusterikokojen tunnuslukuja aineistossa 1, 2D-klusterointi.

	Euklidinen				Gaussinen				Kosini				Czekanowski				Wang			
	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max
Spektrinen 1	40.53	20.44	16	98	40.53	22.96	15	101	40.53	18.03	15	92	40.53	17.94	16	90	40.53	20.26	19	95
Spektrinen 2	40.53	22.30	17	98	40.53	20.67	19	101	40.53	19.92	15	94	40.53	17.33	21	91	40.53	18.24	19	95
MCL	43.43	22.19	18	93	41.93	19.39	19	105	39.23	21.21	12	95	39.23	16.1	13	76	39.23	17.94	13	82
k-means	40.53	22.35	18	123	-	-	-	-	40.53	21.58	19	123	-	-	-	-	-	-	-	-
AP	41.93	24.54	18	140	46.77	33.22	21	184	41.93	18.5	15	89	40.53	17.04	19	89	43.43	17.15	23	89

Taulukko 5.6: Virheet aineistossa 2, 2D-klusterointi.

		Euklidinen		Gaussinen		Kosini		Czekanowski		Wang	
		ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)
	Spektrinen 1	89.06	51.74	87.50	50.08	100.00	-	100.00	-	100.00	-
	Spektrinen 2	87.50	50.40	85.94	51.13	100.00	-	100.00	-	100.00	-
Kerros 1	MCL	96.88	34.58	96.88	34.58	100.00	-	100.00	-	98.44	9.84
	k-means	87.50	18.94	NA	NA	87.50	18.94	NA	NA	NA	NA
	AP	96.88	34.58	96.88	34.85	100.00	-	100.00	-	100.00	-
	Spektrinen 1	99.00	10.70	98.51	37.09	98.01	33.84	98.01	49.00	98.01	49.00
	Spektrinen 2	100.00	-	92.54	76.13	98.01	33.84	98.01	49.00	98.01	49.00
Kerros 2	MCL	95.52	25.64	95.02	32.06	98.01	33.84	98.01	49.00	98.01	49.00
	k-means	93.03	50.33	NA	NA	89.05	36.25	NA	NA	NA	NA
	AP	95.52	37.34	88.56	60.50	98.01	33.84	98.01	37.52	98.01	33.65
	Spektrinen 1	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-
	Spektrinen 2	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-
Kerros 3	MCL	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-
	k-means	100.00	-	NA	NA	100.00	-	NA	NA	NA	NA
	AP	100.00	-	100.00	-	100.00	-	100.00	-	100.00	-

vät välttämättä merkittävästi poikkea toisistaan, sillä ulkona kuultujen tukiasemien lukumäärä on pienempi ja signaalien voimakkuudet ovat heikompia. Erityisesti käytettäessä Euklidista ja Gaussista samanlaisuusfunktiota aineiston 1 ensimmäisessä kerroksessa ulkoa mitatut mittauspisteet asettuvat samoihin klustereihin ja tämä näkyy poikkeuksellisen suurina virhearvoina. Aineiston kolmannessa kerroksessa virheitä ei juurikaan synny, sillä rakennus jakautuu selvästi erillisiin osiin helpottaen klusteroinnin onnistumista. Pienimmät virheet aineistossa 1 saavutetaan spektrisillä algoritmeilla ja Markov-algoritmilla käytettäessä kosini-, Czekanowski- tai Wang-samanlaisuusfunktiota. Kerroksissa 2 ja 3 spektrisillä algoritmeilla saadaan pieniä virhearvoja myös Euklidista ja Gaussista samanlaisuusfunktiota käytettäessä. Huonoin kokonaisvaltainen klusterointitulos aineistossa 1 saadaan affinity propagation -algoritmilla ja Gaussisella samanlaisuusfunktiolla.

Tarkasteltaessa klusterikokojen tunnuslukuja taulukosta 5.5 havaitaan, että jokaisella algoritmilla saadaan melko samansuuruisia arvoja muutamaa poikkeusta lukuun ottamatta. Pienimmän ja suurimman klusterin välinen kokoero on kaikilla menetelmillä huomattava, ja tätä voidaan selittää mittauspisteiden lukumäärien alueellisilla eroilla. Jotkut klusterit sisältävät ainoastaan käytäviltä mitattuja mittauspisteistä, kun taas jotkut klusterit rakentuvat avoimen tilan mittauspisteistä. Näin ollen, kun tarkastellaan suurimpia mittauspisteiden välisiä etäisyyksiä suurissa ja pienissä klustereissa, ei arvoissa välttämättä ole merkittävää eroa.

Tarkastellaan sitten aineistolla 2 saatuja klusterointituloksia, jotka on koottu taulukkoon 5.6. Samoin kuten aineistossa 1, niin myös aineistossa 2, parhaat klusterointitulokset saadaan spektrisillä algoritmeilla ja Markov-algoritmilla käytettäessä kosini-, Czekanowski- tai Wang-samanlaisuusfunktiota. Tämän lisäksi aineistossa 2 saadaan yhtä hyviä tuloksia myös affinity propagation -algoritmilla edellä mainituilla samanlaisuusfunktioilla. Aineiston 2 ensimmäisessä kerroksessa ulkoa mitattujen pisteiden lukumäärä on pienempi, mikä näkyy yleisesti ottaen

Taulukko 5.7: Klusterikokojen tunnuslukuja aineistossa 2, 2D-klusterointi.

	Euklidinen				Gaussinen				Kosini				Czekanowski				Wang			
	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max
Spektrinen 1	15.88	8.76	10	48	15.88	8.67	10	47	15.88	8.62	8	47	15.88	8.98	6	47	15.88	9.05	5	47
Spektrinen 2	15.88	8.60	10	48	15.88	10.73	10	61	15.88	8.53	8	47	15.88	8.95	7	47	15.88	9.02	6	47
MCL	18.14	8.70	10	48	18.14	8.72	10	48	18.14	8.63	10	47	18.14	9.00	10	47	18.14	9.11	10	48
k-means	15.88	9.73	5	47	-	-	-	-	15.88	7.64	5	39	-	-	-	-	-	-	-	-
AP	15.88	7.85	10	46	16.57	9.13	9	45	19.05	8.35	11	47	17.32	8.37	10	47	18.14	9.27	10	48

Taulukko 5.8: Virheet aineistossa 1, 3D-klusterointi.

	Euklidinen		Gaussinen		Kosini		Czekanowski		Wang	
	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)
Spektrinen 1	95.48	74.35	89.39	43.99	95.56	63.05	100.0	-	99.67	17.99
Spektrinen 2	91.78	41.01	89.97	59.15	97.29	17.99	98.77	72.87	99.84	23.47
MCL	95.81	64.81	95.72	67.29	97.86	19.64	99.51	17.12	99.1	12.94
k-means	95.97	58.68	NA	NA	95.15	50.96	NA	NA	NA	NA
AP	88.98	45.69	91.28	48.33	96.55	19.15	96.63	18.53	96.38	15.22

hieman pienempinä virheinä, kun arvoja verrataan aineiston 1 vastaaviin tuloksiin. Vertailtaessa aineiston 1 ja 2 tuloksia rakennuksen toisessa kerroksessa havaitaan, että aineistolla 1 saadaan keskimäärin pienempiä virheitä, mitä voidaan selittää sillä, että klusterointi onnistuu paremmin tapauksissa, joissa mittauspisteitä on tiheämmin. Kerroksen 3 klusteroinneissa ei ilmene juuri ollenkaan virheitä kummassakaan aineistossa. Aineiston 2, 2D-klusteroinnin klusterikokojen tunnusluvut löytyvät taulukosta 5.7. Koska aineistossa 2 on sormenjälkiä huomattavasti harvemmassa mitä aineistossa 1, niin klustereihin kuuluvien sormenjälkien lukumäärät ovat myös selvästi pienempiä. Kun aineistossa 1 yksi klusteri sisältää keskimäärin noin 40 sormenjälkeä, niin aineistossa 2 vastaava luku on noin 16 sormenjälkeä.

Siirrytään sitten tarkastelemaan koko aineistolla tehtyjä testauksia. Aineiston 1 tulokset ovat taulukossa 5.8. Yleisesti ottaen parhaat tulokset aineistolla 1 saavutetaan jälleen kosini-, Czekanowski- ja Wang-samanlaisuusfunktioilla. Normalisoimaton spektrinen klusterointialgoritmi Czekanowski-samanlaisuusfunktiolla tuottaa ainoastaan yhtenäisiä klustereita tarkasteltaessa klustereita leveys- ja pituussuunnassa. Myös normalisoidulla spektrisellä algoritmilla ja Markov-algoritmilla päästään lähes täysin yhtenäisiin klustereihin. Kuvassa 5.6 nähdään esimerkki 3D-klusteroinnista. Kuvassa mittauspisteet on merkitty kartalle oikeisiin sijainteihinsa ja mittauspisteen väri kuvastaa klusteria, johon se kuuluu. Kuvan klusterointi on saatu normalisoimattomalla spektrisellä klusterointialgoritmilla ja Czekanowski-samanlaisuusfunktiolla. Vertailtaessa aineiston 1 3D-klusteroinnin klusterikokojen tunnuslukuja (taulukko 5.9) vastaaviin 2D-klusteroinnin lukuihin huomataan, että keskimääräinen klusterikoko pysyy keskimäärin lähes samana. Klusterikokojen hajonnat sen sijaan kasvavat ja suurimman ja pienemmän klusterin kokoero kasvaa. Voidaan siis päätellä, että 3D-klusterointi ei yleisellä tasolla tuota kooltaan yhtä tasaisia klustereita mitä 2D-klusterointi. Tämä sama ilmiö toistuu aineistolla 2.

Taulukko 5.9: Klusterikokojen tunnuslukuja aineistossa 1, 3D-klusterointi.

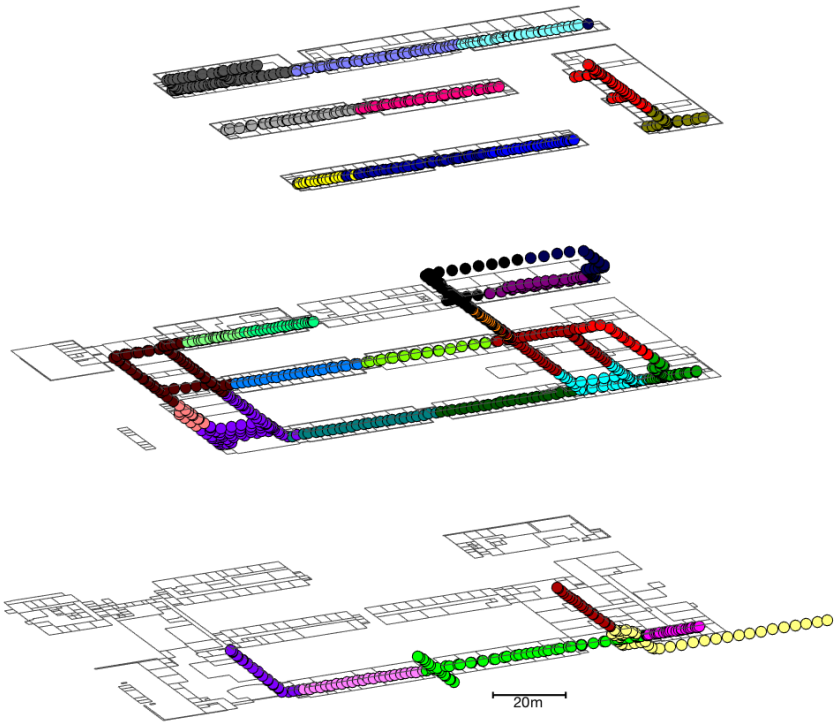
	Euklidinen				Gaussinen				Kosini				Czekanowski				Wang			
	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max
Spektrinen 1	40.53	26.79	18	118	40.53	28.54	18	152	40.53	22.65	15	104	40.53	21.68	13	107	40.53	22.87	13	107
Spektrinen 2	40.53	26.18	15	126	40.53	35.39	18	201	40.53	22.22	15	120	40.53	21.11	18	110	40.53	22.17	18	121
MCL	48.64	31.57	18	131	48.64	32.46	19	153	46.77	26.57	18	117	45.04	22.61	14	104	36.85	18.98	14	104
k-means	40.53	27.91	5	163	-	-	-	-	40.53	27.04	16	153	-	-	-	-	-	-	-	-
AP	36.85	22.75	15	136	45.04	37.88	15	215	43.43	21.45	16	113	40.53	15.07	18	78	39.23	15.01	15	71

Aineistolla 2 saadut 3D-klusteroinnin tulokset löytyvät taulukosta 5.10. Parhaat tulokset saavutetaan spektrisillä algoritmeilla ja Markov-algoritmilla käytettäessä Wang-samanlaisuus-

Taulukko 5.10: Virheet aineistossa 2, 3D-klusterointi.

	Euklidinen		Gaussinen		Kosini		Czekanowski		Wang	
	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)	ydin(%)	$\bar{e}$ (m)
Spektrinen 1	88.98	26.18	88.98	34.05	94.23	16.89	94.49	17.26	96.06	19.86
Spektrinen 2	88.45	36.94	91.6	33.93	93.44	16.06	91.86	59.69	96.06	32.29
MCL	94.49	30.99	95.01	35.22	93.96	16.75	96.33	20.96	96.59	22.67
k-means	92.65	30.78	NA	NA	93.18	34.21	NA	NA	NA	NA
AP	91.86	30.29	88.98	42.05	93.44	17.75	96.06	25.93	93.44	19.45

funktiota sekä affinity propagation -algoritmilla käytettäessä Czekanowski-samanlaisuusfunktiota. Käytettäessä 3D-klusterointia ei yleisesti ottaen aineistossa 2 päästä yhtä hyviin klusterointituloksiin kuin aineistossa 1. Tästä voidaan jälleen päätellä, että klusteroitaessa sormenjälkiä saavutetaan yhtenäisempiä klustereita, kun datapisteitä on kerätty tiheästi eri alueilta. Tarkasteltaessa aineiston 2 3D-klusteroinnin klusterikokojen tunnuslukuja taulukosta 5.11 huomataan, että erityisesti spektrisillä algoritmeilla saadaan tasaisia klusterikokoja myös harvalla mittausaineistolla 3D-klusterointia käytettäessä.



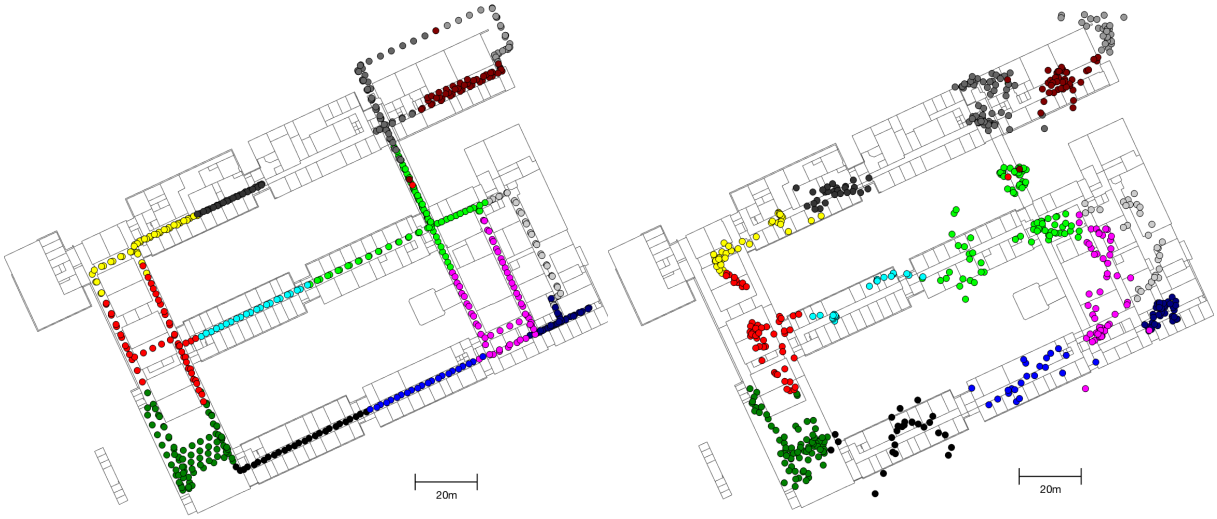
Kuva 5.6: Esimerkki 3D-klusteroinnin tuloksesta.

Vertailtaessa 2D-klusteroinnista saatuja tuloksia 3D-klusteroinnista saatuihin tuloksiin voidaan todeta, että 2D-klusterointi tuottaa yleisesti ottaen yhtenäisempiä klustereita kuin 3D-klusterointi. Lisäksi 2D-klusteroinnilla saadut klusterit ovat kooltaan tasaisempia kuin 3D-klusteroinnilla saadut. Parhaimmillaan 3D-klusterointi kuitenkin tuottaa tiheällä mittausdatalla hyvin yhtenäisiä klustereita, eikä näin ollen datan klusterointi edellytä mittausdatan jaottelua kerroksiin ennen klusteroinnin suorittamista. Kuten jo aiemmin todettiin sormenjälkien klusteroiminen yhtenäisiin klustereihin niin, että klusterit sisältävät mittauspisteitä ainoastaan yhden kerroksen sisältä tuottaa ongelmia. Näin ollen kerrosinformaation selvittämiseksi tarvitaan apua sensoreista saatavasta datasta. Mikäli aineisto voidaan sensoridatan avulla jakaa kerroksiin,

Taulukko 5.11: Klusterikokojen tunnuslukuja aineistossa 2, 3D-klusterointi.

	Euklidinen				Gaussinen				Kosini				Czekanowski				Wang			
	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max	ka	σ	min	max
Spektrinen 1	15.88	6.55	10	36	15.88	7.74	9	42	15.88	5.74	10	32	15.88	5.16	10	30	15.88	5.18	10	27
Spektrinen 2	15.88	6.87	10	39	15.88	6.12	10	32	15.88	5.50	10	30	15.88	5.77	10	29	15.88	7.98	6	46
MCL	15.24	7.04	9	39	15.88	7.83	9	39	15.24	5.66	10	33	15.24	8.73	1	47	15.24	8.57	2	48
k-means	15.88	8.46	5	33	-	-	-	-	15.88	7.01	5	30	-	-	-	-	-	-	-	-
AP	14.11	7.01	5	35	16.57	11.13	8	55	15.24	7.02	9	40	15.24	7.09	8	41	14.11	6.05	9	40

voidaan klusterointi suorittaa 2D-klusterointina kerroksittain. Toinen vaihtoehto on suorittaa 3D-klusterointi ja tämän jälkeen tarkastella löytyykö klustereiden sisällä sormenjälkien sensoriinformaatiosta sellaisia poikkeamia, että voidaan olettaa sormenjälkien olevan eri kerroksissa, mutta kuitenkin lähellä toisiaan pituus- ja leveyssuunnassa.



(a) Klusterointi

(b) Sormenjäljet estimoiduissa sijainneissaan

Kuva 5.7: Klusterointi ja sormenjälkien sijaintien estimointi.

Tarkastellaan vielä lopuksi kokonaiskuvaa havainnollistavaa esimerkkiä ja siitä saatuja tuloksia. Kuvassa 5.7 on esimerkki menetelmästä kerroksessa 2, kun klusterointialgoritmina on käytetty normalisoimatonta spektristä klusterointialgoritmia ja samanlaisuuksien laskemiseen on käytetty kosini-samanlaisuusfunktiota. Kuvassa 5.7a näkyy sormenjäljet kartalla oikeilla mittauspaikoillaan ja värikoodi kuvastaa klusteria johon kukin sormenjälki kuuluu. Kuvassa 5.7b näkyy sormenjälkien estimoidut sijainnit, kun klustereiden sormenjäljille on suoritettu etäisyyksien arviointi, moniulotteinen skaalaus ja Euklidinen muunnos. Taulukkoon 5.12 on koottu eri

Taulukko 5.12: Estimointivirheiden keskiarvoja paikannusmenetelmää havainnollistavassa esimerkissä.

	Euklidinen	Gaussinen	Kosini	Czekanowski	Wang
Spektrinen 1	7.02	7.17	6.44	6.17	6.14
Spektrinen 2	7.18	6.75	6.37	6.31	6.07
MCL	7.24	7.52	6.21	5.96	5.96
k-means	6.99	NA	6.94	NA	NA
AP	9.1	10.58	6.83	6.81	7.41

algoritmeilla ja samanlaisuusfunktioilla saatuja estimointivirheitä. Lasketut virheet eivät kuitenkaan ole täysin vertailukelpoisia, sillä klustereiden lukumäärät eivät ole kaikilla algoritmeilla ja samanlaisuusfunktioilla täysin samoja, mikä vaikuttaa oleellisesti virheiden suuruuteen. Kokonaisvaltaisesti voidaan kuitenkin todeta, että myös tässä esimerkissä parhaimmat tulokset saavutetaan kosini-, Czekanowski- sekä Wang-samanlaisuusfunktioilla.

6 Yhteenveto

Tässä työssä testattiin eri klusterointialgoritmeja WLAN-signaalien voimakkuuksista muodostuvien sormenjälkien klusterointiin. Työn alussa esiteltiin klusteroinnin ja graafiteoreettisen klusteroinnin perusperiaatteita. Tämän jälkeen käsiteltiin klusteroinnin onnistumisen kannalta tärkeässä roolissa olevia samanlaisuus- ja erilaisuusfunktioita ja esiteltiin erilaisia samanlaisuusgraafityyppejä. Työssä esitettiin Markov-, k-means- ja affinity propagation -klusterointialgortimin toimintaperiaatteet. Lisäksi spektrisiin klusterointialgoritmeihin paneuduttiin tarkemmin ja käytiin läpi niiden graafiteoreettinen toimintaidea. Ennen työssä suoritettuja testauksia käytiin läpi tyypillisen RSS-sormenjäljistä muodostuvan radiokartan muodostamisprosessi ja tarkasteltiin miten klusterointia on aiemmin hyödynnetty RSS-sormenjälkiin perustuvassa paikannuksessa.

Algoritmeja testattiin tutkielmaa varten mitatuilla aineistoilla, jotka erosivat toisistaan mittauspisteiden tiheydellä. Testaukset suoritettiin sekä 2-ulotteisella, että 3-ulotteisella aineistolla käyttäen eri samanlaisuus- ja erilaisuusfunktioita. Algoritmien tuottamien klustereiden vertailu perustui siihen, kuinka yhtenäisiä muodostuneet klusterit olivat. Tämän lisäksi tarkasteltiin klusterikokojen tunnuslukuja. Kokonaisvaltaisesti tarkastellen yhtenäisimmät klusterit saavutettiin spektrisillä algoritmeilla sekä Markov-klusterointialgoritmilla, kun käytettiin kosini-, Czekanowski- tai Wang-samanlaisuusfunktiota. Tilanteissa, joissa aineisto oli selvästi jakautunut erillisiin käytäviin, saavutettiin hyviä klusterointituloksia kaikilla testatuilla menetelmillä. Ulkoa mitatut sormenjäljet tuottivat klusteroinnin suhteen haasteita erityisesti Euklidista ja Gaussista funktiota käytettäessä. Vertailtaessa klusterikokoja 2D-klusteroinnin ja 3D-klusteroinnin välillä havaittiin, että 2D-klusteroinnilla saadut klusterit olivat kooltaan tasaisempia kuin 3D-klusteroinnilla saadut. Klustereiden yhtenäisyyttä 3D-klusteroinnissa tarkasteltiin ainoastaan pituus- ja leveyssuunnassa. Sormenjälkien jaottelu eri kerroksiin osoittautui testauksissa haasteelliseksi. Lisäksi havaittiin, että tiheämpää mittausaineistoa klusteroitaessa saadaan helpommin muodostettua yhtenäisempiä klustereita. Algoritmien vertailujen lisäksi suoritettiin uudenlaista paikannusideaa havainnollistava esimerkki. Myös esimerkki suoritettiin eri algoritmeilla, mutta saadut tulokset eivät olleet täysin vertailukelpoisia.

Työssä käytettiin sormenjälkien klusterointiin ainoastaan WLAN-tukiasemista saatuja signaalien voimakkuusarvoja. Jatkotutkimuksena olisi mielenkiintoista lisätä sormenjälkiin enemmän informaatiota. Esimerkiksi barometrisensorista saatavan informaation avulla voitaisiin klusterointia kehittää myös kerrosten väliseen erotteluun. Myös mahdollisten GPS-havaintojen avulla voitaisiin parantaa klusteroinnista saatavia tuloksia. Työssä esitelty uudentyyppinen paikannusidea pohjautuu sormenjälkien välisten etäisyyksien arvioimiseen. Riittävän tarkan etäisyyksien arviointimenetelmän ja klusteroinnin yhteen sovittaminen olisi jatkotutkimuksen kannalta oleellista. Lisäksi malliin tuo oman haasteensa klustereiden yhdistäminen. Uusi tutkimisen haara voisikin olla soveltaa sumeita klusterointialgoritmeja sormenjälkien ryhmittelyyn. RSS-sormenjäljistä muodostuvien klustereiden klusterirajat ovat häilyviä, mikä puoltaisi sumeiden algoritmien käyttöä.

Kirjallisuutta

[1] C. C. Aggarwal, C. K. Reddy. *Data clustering Algorithms and applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series (2013).

[2] D. S. Bernstein. *Matrix Mathematics Theory, Facts, and Formulas*. Princeton University Press (2009).

[3] R. S. Campos, L. Lovisolo, M. L. R. de Campos. *Wi-Fi multi-floor indoor positioning considering architectural aspects and controlled computational complexity*. Expert Systems with Applications, 41, 14, s. 6211-6223 (2014).
https://www.sciencedirect.com/science/article/pii/S0957417414002073

[4] S.-H. Cha. *Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions*. International journal of mathematical models and methods in applied sciences, 1, 4, s. 300-307 (2007).
http://users.uom.gr/~kouiruki/sung.pdf

[5] A. Cramariuc, H.Huttunen, E. S. Lohan. *Clustering benefits in mobile-centric WiFi positioning in multi-floor buildings*. International Conference on Localization and GNSS, IEEE (2016).
https://ieeexplore.ieee.org/document/7533846

[6] S. van Dongen. *Graph clustering by flow simulation*. Väitöskirja (2000).
https://micans.org/mcl/index.html?sec_thesisetc

[7] S. van Dongen. *A cluster algorithm for graphs*. INS-R0010 (2000).
https://micans.org/mcl/index.html?sec_thesisetc

[8] D. Dueck, B. J. Frey. *Non-metric affinity propagation for unsupervised image categorization*. IEEE 11th International Conference on Computer Vision, IEEE (2007).
https://ieeexplore.ieee.org/document/4408853

[9] D. Dueck. *Affinity propagation: Clustering data by passing messages*. Väitöskirja (2009).
http://www.cs.columbia.edu/~delbert/docs/DDueck-thesis_small.pdf

[10] H. Durrant-Whyte, T. Bailey. *Simultaneous localization and mapping: part I*. IEEE Robotics & Automation Magazine, 13, 2, s. 99-110 (2006).
https://ieeexplore.ieee.org/document/1638022

[11] C. Feng, W. S. A. Au, S. Valaee, Z. Tan. *Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing*. IEEE Transactions on Mobile Computing, 11, 12, s. 1983-1993 (2012).
https://ieeexplore.ieee.org/document/6042868

[12] B. J. Frey, D. Dueck. *Clustering by Passing Messages Between Data Points*. Science, 315, 5814, s. 972-976 (2007).
http://science.sciencemag.org/content/315/5814/972

[13] G. Gan, C. Ma, J. Wu. *Data Clustering Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability (2007).

[14] V. Honkavirta, T. Perälä, S. Ali-Löytty, R. Piché. *A Comparative Survey of WLAN Location Fingerprinting Methods*. 6th Workshop on Positioning, Navigation and Communication, IEEE (2009).
https://ieeexplore.ieee.org/document/4907834

[15] H. Jia, S. Ding, X. Xu, R. Nie. *The latest research progress on spectral clustering*. Neural Computing and Applications, 24, 7-8, s. 1477-1486 (2014).
https://link.springer.com/article/10.1007%2Fs00521-013-1439-2

[16] A. Khalajmehrabadi, N. Gatsis, D. Akopian. *Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges*. IEEE Communications Surveys & Tutorials, 19, 3, s. 1974-2002 (2017).
https://ieeexplore.ieee.org/abstract/document/7874080

[17] P. Koivisto, R. Niemistö. *Graafiteoriaa*. Tampereen yliopisto, Informaatiotieteiden yksikön raportteja (2018).
http://tampub.uta.fi/handle/10024/102835

[18] B. Li, B. Harvey, T. Gallagher. *Using Barometers to Determine the Height for Indoor Positioning*. International Conference on Indoor Positioning and Indoor Navigation, IEEE (2013).
https://ieeexplore.ieee.org/document/6817923

[19] U. von Luxburg. *A Tutorial on Spectral Clustering*. Statistics and Computing, 17, 4, s. 395-416 (2007).
https://link.springer.com/article/10.1007/s11222-007-9033-z

[20] J. Merikoski, A. Virtanen, P. Koivisto. *Johdatus diskreettiin matematiikkaan*. WSOY (2004).

[21] L. Peng, J. Liu, M. Sheng, Y. Zhang, D. Hou, Y. Zheng, J. Li. *3D Indoor Localization based on Spectral Clustering and Weighted Backpropagation Neural Networks*. International Conference on Communications, IEEE (2017).
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8330353

[22] A. Razavi, M. Valkama, E.-S. Lohan. *K-Means Fingerprint Clustering for Low-Complexity Floor Estimation in Indoor Mobile Localization*. IEEE Globecom Workshops, IEEE (2015).
https://ieeexplore.ieee.org/abstract/document/7414026

[23] S. E. Schaeffer. *Graph clustering survey*. Computer Science Review, 1, 1, s. 27-64 (2007).
https://www.sciencedirect.com/science/article/pii/S1574013707000020

[24] S. S. Skiena. *The Data Science Design Manual*. Springer (2017).

[25] M. Stoer, M. Wagner. *A Simple Min-Cut Algorithm*. Journal of the ACM, 44, 4, s. 585-591 (1997).
https://dl.acm.org/citation.cfm?id=263872

[26] G. Strang. *Introduction to Linear Algebra, 5th Edition*. Wellesley - Cambridge Press (2016).

[27] N. Swangmuang, P. Krishnamurthy. *On Clustering RSS Fingerprints for Improving Scalability of Performance Prediction of Indoor Positioning Systems*. Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, s. 61-66 (2008).
https://dl.acm.org/citation.cfm?id=1410027

[28] K. Thulasiraman, M. N. S. Swamy. *Graphs: Theory and algorithms*. John Wiley & Sons, Inc. (1992).

[29] J. Torres-Sospedra, R. Montoliu, S. Trilles, O. Belmonte, J. Huerta. *Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprint indoor positioning systems*. Expert Systems with Applications, 42, 23, s. 9263-9278 (2015).
https://www.sciencedirect.com/science/article/pii/S0957417415005527

[30] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, R. R. Choudhury. *No Need to War-Drive: Unsupervised Indoor Localization*. Proceedings of the 10th international conference on Mobile systems, applications, and services, s. 197-210 (2012).
https://dl.acm.org/citation.cfm?id=2307655

[31] S. Wierzchonn, M. Klopotek. *Modern Algorithms of Cluster Analysis*. Springer International Publishing AG (2018).
https://link.springer.com/book/10.1007/978-3-319-69308-8

[32] L. Wirola, L. Wirola, R. Piché. *Bandwidth and Storage Reduction of Radio Maps for Offline WLAN Positioning*. International Conference on Indoor Positioning and Indoor Navigation, IEEE (2013).
https://ieeexplore.ieee.org/document/6817885

[33] S. J. Wodak, J. Vlasblom. *Markov clustering versus affinity propagation for the partitioning of protein interaction graphs*. BMC Bioinformatics (2009).
https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-99

[34] L. Xiao, A. Behboodi, R. Mathar. *Learning the Localization Function: Machine Learning Approach to Fingerprint Localization* (2018).
https://arxiv.org/abs/1803.08153

[35] C. T. Zahn. *Graph-theoretical methods for detecting and describing gestalt clusters*. IEEE Transactions on Computers, C-20, 1, s. 68-86 (1971).
https://ieeexplore.ieee.org/abstract/document/1671676

[36] C. Zhou, A. Wieser. *CDM: Compound dissimilarity measure and an application to fingerprinting-based positioning*. International Conference on Indoor Positioning and Indoor Navigation (2018).
https://arxiv.org/abs/1805.06208

[37] G. Zou, L. Ma, Z. Zhang, Y. Mo. *An indoor positioning algorithm using joint information entropy based on WLAN fingerprint*. Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), IEEE (2014).
https://ieeexplore.ieee.org/document/6963033

Liite

Taulukko 6.1: Klustereiden lukumäärät aineistossa 1, 2D-klusterointi.

		Euklidinen	Gaussinen	Kosini	Czekanowski	Wang
Kerros 1	Spektrinen 1	7	7	7	7	7
	Spektrinen 2	7	7	7	7	7
	MCL	5	5	6	6	6
	k-means	7	-	7	-	-
	AP	6	5	6	7	6
Kerros 2	Spektrinen 1	14	14	14	14	14
	Spektrinen 2	14	14	14	14	14
	MCL	13	15	15	15	15
	k-means	14	-	14	-	-
	AP	14	12	14	14	13
Kerros 3	Spektrinen 1	9	9	9	9	9
	Spektrinen 2	9	9	9	9	9
	MCL	10	9	10	10	10
	k-means	9	-	9	-	-
	AP	9	9	9	9	9

Taulukko 6.2: Klustereiden lukumäärät aineistossa 2, 2D-klusterointi.

		Euklidinen	Gaussinen	Kosini	Czekanowski	Wang
Kerros 1	Spektrinen 1	5	5	5	5	5
	Spektrinen 2	5	5	5	5	5
	MCL	4	4	4	4	4
	k-means	5	-	5	-	-
	AP	4	4	4	4	3
Kerros 2	Spektrinen 1	10	10	10	10	10
	Spektrinen 2	10	10	10	10	10
	MCL	10	10	10	10	10
	k-means	10	-	10	-	-
	AP	12	11	9	10	10
Kerros 3	Spektrinen 1	9	9	9	9	9
	Spektrinen 2	9	9	9	9	9
	MCL	7	7	7	7	7
	k-means	9	-	9	-	-
	AP	8	8	7	8	8

Taulukko 6.3: Klustereiden lukumäärät aineistossa 1, 3D-klusterointi.

	Euklidinen	Gaussinen	Kosini	Czekanowski	Wang
Spektrinen 1	30	30	30	30	30
Spektrinen 2	30	30	30	30	30
MCL	25	25	26	27	33
k-means	30	-	-	30	-
AP	33	27	28	30	31

Taulukko 6.4: Klustereiden lukumäärät aineistossa 2, 3D-klusterointi.

	Euklidinen	Gaussinen	Kosini	Czekanowski	Wang
Spektrinen 1	24	24	24	24	24
Spektrinen 2	24	24	24	24	24
MCL	25	24	25	25	25
k-means	24	-	24	-	-
AP	27	23	25	25	27