Klusterointi itseorganisoituvalla kartalla

Petri Hellemaa
Petri.Hellemaa@cs.helsinki.fi

Klusterointimenetelmät-seminaari
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C–2002–54, s. 1–11, marraskuu 2002

Tiivistelmä

Itseorganisoituva kartta on epälineaarinen kuvaus moniulotteisesta syöteavaruudesta pienempään ulottuvuuteen säännöllisen muotoiselle hilalle. Hila koostuu soluista, joista jokaiseen on liitetty vektori, joka vastaa jotain syöteavaruuden pistettä. Nämä mallivektorit opetetaan iteratiivisesti sellaisiksi, että ne kuvaavat syöteaineiston mahdollisimman kattavasti ja hilalla vierekkäisten solujen vektorit sijaitsevat syöteavaruudessa lähekkäin. Näyte kuvautuu aina siihen soluun, jonka mallivektoriin on pienin etäisyys, joten samankaltaiset syötteet kuvautuvat yleensä joko samoihin tai vierekkäisiin soluihin. Usein kohdeavaruus valitaan kaksiulotteiseksi, jolloin hila on helppo piirtää ja tuloksena on näkymä moniulotteiseen aineistoon.

Itseorganisoituvan kartan mallivektoreista muodostuu kattava joukko tyypillisiä syöteaineiston tapauksia, joten ne ovat tulkittavissa klustereiksi. Koska kartan opettaminen kuitenkin edellyttää, että hilan rakenne ja koko valitaan etukäteen, suositeltavampaa on tulkita klustereiksi hilan alueita, joissa vektorit ovat samankaltaisia. Tällöin ei myöskään rajoituta konvekseihin klustereihin, mutta niiden rajat jäävät tulkinnanvaraisiksi. Monen solun klusterit erottuvat väritettäessä hila niin, että sävytys paljastaa vierekkäisten solujen samankaltaisuuden.

Haluttaessa yksiselitteinen klusterointi mallivektorit voidaan syöttää esimerkiksi hierarkkiselle klusterointialgoritmille, jolloin hyödytään kartan tuottamasta tiivistyksestä, koska kartan opetusalgoritmin aikavaativuus ei riipu niin voimakkaasti aineiston koosta ja kohinaakin suodattuu pois.

1 Johdanto

Vuonna 1952 Turing havaitsi, että paikallisista vuorovaikutuksista voi syntyä globaali järjestys [Tur52]. Ilmiötä kutsutaan nykyisin *itseorganisoitumiseksi.* Tunnetuin itseorganisoitumiseen perustuva laskennallinen malli lienee Kohosen *itseorganisoituva kartta* [Koh82, Koh90, Koh95]. Tässä raportissa esitellään itseorganisoituvan kartan malli ja selvitetään erityisesti sen käyttömahdollisuuksia klusterointiin ja ylipäänsä tietämyksen muodostamiseen, johon se perusominaisuuksiltaan vaikuttaakin soveltuvan mainiosti.

Luvussa 2 itseorganisoituva kartta esitellään opetusalgoritmeineen vektorikoodausmenetelmänä. Opetusalgoritmia analysoitaessa todetaan yhteys k:n keskiarvon menetelmään. Luku 3 käsittelee opetetun kartan käyttöä moniulotteisen datan visualisointiin ja aineiston klusterointiin. Menetelmä ei tuota yksikäsitteistä klusterointia, joskin käyttäjä voi havaita selvän ryhmittelyn, mikäli kartta esimerkiksi väritetään sopivalla tekniikalla. Luvussa 4 mainitaan joitakin tärkeitä parannuksia ja laajennuksia menetelmään ja pyritään tarkastelemaan itseorganisoituvia karttoja yleisemmällä tasolla.

2 Itseorganisoituva kartta

Itseorganisoituvan kartan (engl. self-organizing map, SOM) kehitti Kohonen vuonna 1982 pyrkiessään luomaan biologisesti uskottavan assosiatiivimuistimallin [Koh82, Koh89, Koh90, Koh95]. Menetelmä on löytänyt tiensä monenlaisille sovellusalueille, muun muassa teollisuusprosessien valvontaan, puheentunnistukseen ja robotiikkaan. Tällaiset sovellukset perustuvat Kohosen mallin kykyyn mukautua sopivasti moniulotteiseen näyteavaruuteen.

Itseorganisoituva kartta on miellettävissä kaksiulotteiseksi neuronihilaksi, jonka eri alueet opetetaan reagoimaan erityyppisiin syötteisiin [Koh90]. Kyse on ohjaamattomasta kilpailuoppimisesta. Perinteisesti itseorganisoituva kartta esitelläänkin neuraalilaskennan mallina, mutta tässä yhteydessä se lienee pikemminkin tulkittavissa eräänlaiseksi *vektorikoodausalgoritmiksi*: se sijoittaa optimaalisesti rajallisen määrän vektoreita (koodeja) suurempiulotteiseen syöteavaruuteen ja näin ollen tiivistää dataa. “Optimaalisuus” ilmenee siten, että opetusvaiheen jälkeen nämä *mallivektorit* eli *mallit* ovat tulkittavissa edustavaksi joukoksi tyypillisiä syöteaineiston tapauksia.

Kartan keskeisin ominaisuus on sen kyky ulottuvuuden pienennykseen. Kuvaus pienempään ulottuvuuteen on epälineaarinen, minkä perusteella Ritter [Rit95] on luonnehtinut menetelmää pääkomponenttianalyysin yleistykseksi. (Pääkomponenttianalyysi kykenee vain lineaarisiin kuvauksiin.) Kartta(hila) muodostetaan yleensä yksi- tai kaksiulotteiseksi, sillä moniulotteisempaa avaruutta on vaikeampi visualisoida ja tulkita. Menetelmä sallii kyllä suuremmankin ulottuvuuden käytön. Opetetulla kartalla samankaltaiset mallit sijaitsevat lähekkäin muodostaen ryhmiä, kun taas toisistaan selvästi poikkeavat mallit päätyvät toisistaan etäälle. Kartta muodostetaan iteratiivisella opetusalgoritmilla, joka esitetään seuraavaksi luvussa 2.1. Luvussa 2.2 algoritmia tarkastellaan optimointiongelmana ja todetaan yhteys k:n keskiarvon menetelmään.

2.1 Kartan muodostaminen

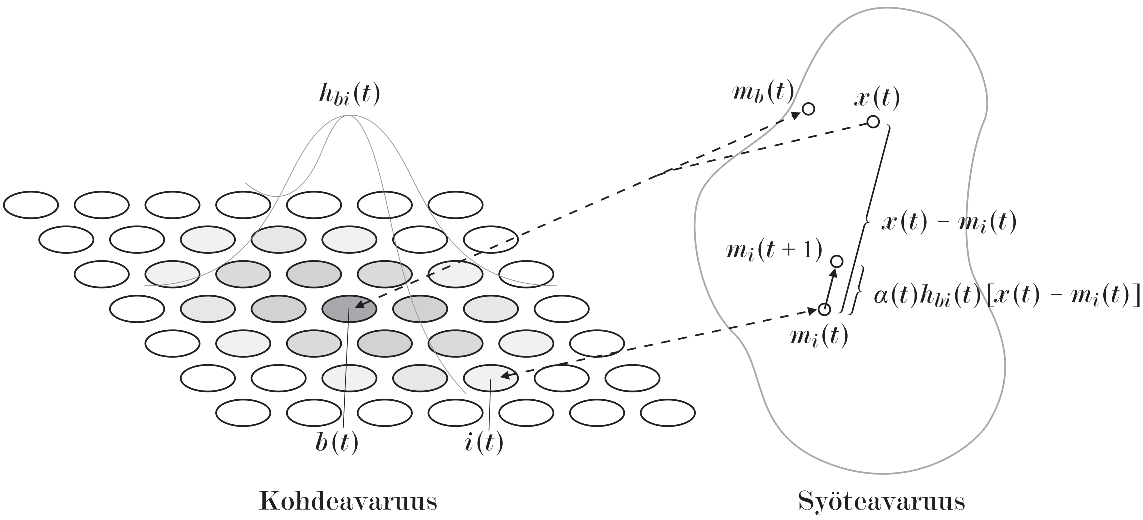
Itseorganisoituvan kartan muodostamiseksi syöteaineiston näytteet tulee kuvata piirrevektoreina. Mielekäs lopputulos edellyttää, että piirteitä painotetaan järkevästi. Yleensä syöteaineiston näytteiden kaikki vektorikomponentit ovat numeerisia ja ne skaalataan välille $[0, 1]$.

Karttahila on rakenteeltaan säännöllinen ja koostuu karttayksiköistä eli soluista. Solut ovat yleensä neli- tai kuusikulmaisia. Jokaisella kartan solulla i on syöteavaruuteen osoittava vektori $m_i = [m_{i1}, \cdots, m_{id}]$ missä d vastaa syöteavaruuden ulottuvuuksien lukumäärää. Syöteaineiston näyte kuvautuu aina sitä parhaiten vastaavaan malliin eli siihen soluun, johon liitettyyn vektoriin on pienin etäisyys syöteavaruudessa. Useimmiten käytetään euklidisia etäisyyksiä.

Kohosen kartan *itseorganisoituvuudella* tarkoitetaan sitä, että karttaa muodostettaessa mallivektorit järjestyvät hilalla. Tämä saadaan aikaan

1. keskittämällä kartan toiminta avaruudellisesti sen yksikön ympärille, joka on parhaiten virittynyt reagoimaan kulloiseenkiin syötteeseen, ja

2. herkistämällä paitsi voimakkaimmin reagoivaa yksikköä myös sen lähinaapureita entisestään nykyiselle näytteelle [Koh90].



Kuva 1: Itseorganisoituva kartta on kuvaus pienempään ulottuvuuteen. Kukin syöteavaruuden näyte kuvautuu sitä lähimpänä olevaan malliin m_b. Kuvaus muodostetaan iteratiivisella opetusalgoritmilla, jonka opetusaskelessa t satunnaisesti valittua näytettä $x(t)$ lähimpänä olevaa mallia $m_b(t)$ ja sen naapureita päivitetään hieman lähemmäksi näytettä. Naapuruuden aste on sumea ja sen määrää funktio $h_{bi}(t)$. Muutoksen suuruuteen vaikuttavat etäisyys näytteeseen, yleinen muutoskerroin $\alpha(t)$ ja naapuruuden aste $h_{bi}(t)$.

Näillä kahdella periaatteella voidaan karkeasti luonnehtia kartan opetusalgoritmin toimintaa yksittäisessä opetusaskelessa (vertaa allaolevaan).

Aluksi mallivektorit alustetaan antamalla jokaiselle vektorikomponentille esimerkiksi satunnaisia opetusjoukon (syöteaineiston) arvoja. Sitten karttaa aletaan opettaa [Koh95] iteratiivisesti niin, että jokaisessa opetusaskelessa t:

1. Valitaan syöteaineistosta satunnainen näytevektori x. Jokaisen mallivektorin m_i etäisyys näytevektoriin x lasketaan. Näytettä parhaiten vastaava yksikkö b, *voittaja*,[1] on se kartan yksikkö, jonka etäisyys näytteeseen on pienin ja jolle siis pätee

$$\|x - m_b\| = \min_i \{\|x - m_i\|\}. \tag{1}$$

2. Lähimmän yksikön b ja sen naapurien vektoreita siirretään lähemmäksi näytevektoria. Kutakin yksikköä i päivitetään kaavan

$$m_i(t+1) = m_i(t) + \alpha(t)h_{bi}(t)[x - m_i(t)] \tag{2}$$

mukaan, missä t on aika (opetusaskel), $\alpha(t)$ on yleinen muutoskerroin ja $h_{bi}(t)$ on naapuruusfunktio, joka saavuttaa maksimin voittajayksikön kohdalla.

Kuva 1 valaisee asiaa. Alussa muutosten pitäisi vaikuttaa kartan jokaiseen soluun, joten naapuruusfunktion säteen tulisi vastata kartan läpimittaa. Laskennan edetessä muutoskerrointa $\alpha(t)$ ja naapuruusfunktion vaikutussädettä pienennetään monotonisesti [Koh95].

Olkoon $\sigma(t)$ naapuruusfunktion leveys, jolloin luonnollinen valinta naapuruusfunktioksi on gaussilaisen pinnan tuottava funktio

$$h_{bi}(t) = \exp\left(-\frac{\|r_b - r_i\|^2}{2\sigma^2(t)}\right) \tag{3}$$

[1] Algoritmi perustuu kilpailuoppimiseen: mallit (neuronit) kilpailevat keskenään, ja syötettä lähimpänä oleva malli ”voittaa”.

missä r_b ja r_i ilmaisevat solujen b ja i sijainnit hilalla. Se saavuttaa maksiminsa voittajayksikön b kohdalla ja pienenee monotonisesti etäisyyden kasvaessa. Koska gaussilainen funktio palauttaa positiivisia arvoja etäisimpienkin solujen kohdalla, sen mukaan kaikkia mallivektoreita pitäisi päivittää. Käytännössä päivitetään vektoreita vain tietyn säteen sisällä laskennan nopeuttamiseksi. Esimerkiksi Kohosen alkuperäisessä algoritmissa tietyn säteen sisällä kaikkia soluja päivitettiin samalla kertoimella mutta alueen ulkopuolisiin soluihin ei koskettu [Koh82]. Tämä kuitenkin hidasti kartan konvergoitumista, ja opetusaskelia vaadittiin enemmän [Koh95]. Yleensä käytetäänkin jotain gaussilaisen funktion approksimaatiota rajoitetulla vaikutussäteellä.

Opetus jakautuu kahteen vaiheeseen [Koh90]:

Karkean opetuksen vaiheessa naapuruusfunktio vaikuttaa karttaan suurilla alueilla, minkä ansiosta mallit saavuttavat hilalla keskinäisen järjestyksen niin, että samankaltaiset syötteet alkavat kuvautua enimmäkseen joko samoihin tai vierekkäisiin soluihin. Vaiheen aikana muutoskerroin $\alpha(t)$ pienennetään nopeasti suuruusluokkaa 1,0 olevasta arvosta suuruusluokkaa 0,01 olevaan arvoon. Naapuruusfunktion säde voidaan pienentää nopeasti vaiheen loppuun mennessä kattamaan vain voittajien lähimmät naapurit.

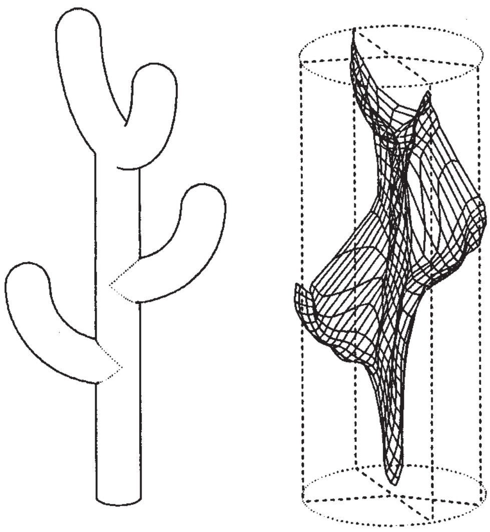
Hienosäätövaiheessa syöte pyritään jakamaan optimaalisesti vierekkäisten mallien kesken ja muutoskerroin $\alpha(t)$ pienennetään hitaasti nollaan. Viimeisillä iteraatioilla päivitetään vain voittajia. Hienosäätövaihe vaatii yleensä noin kymmenkertaisen määrän opetusaskelia verrattuna karkean opetuksen vaiheeseen.

Hilan rakenne ja koko, opetusaskelten määrä ja funktiot joudutaan määräämään etukäteen. Nyrkkisääntönä hyvän tilastollisen tarkkuuden saavuttamiseksi opetusaskelia tulisi olla vähintään 500 kertaa karttayksiköiden lukumäärä [Koh90]. Vaatimus ei riipu ulottuvuuksien määrästä. Kartan jäsentymisen kannalta on tärkeää, että solut eivät opi toisistaan riippumatta, vaan niitä pitää opettaa topologisina kokonaisuuksina. Aluksi, kun korjauskertoimet ovat suuria, mallit "vaeltavat" kartalla nopeasti ja samankaltaiset muodostavat ryhmiä. Opetuksen edetessä syntyy yhä paikallisempaa erittelyä.

Käsityksen kartan käyttäytymisestä opetusvaiheen aikana saa piirtämällä hilan kolmiulotteiseen syöteavaruuteen, jossa mallivektorit sijaitsevat. Hila, joka kohdeavaruudessa on muodoltaan säännöllinen, näyttääkin elastiselta verkolta, joka alkaa myötäillä aineiston tihentymiä (kuva 2). Elastinen pinta ei kuitenkaan kykene ottamaan huomioon kaikkia kolmiulotteisen avaruuden paikallisia tilastollisia piirteitä, joten on selvää, että epälineaarisessa kuvauksessa pienempään ulottuvuuteen esiintyy käytännössä epäjatkuvuuksia. Toisin sanoen se, että hilalla vierekkäiset mallit saadaan opettamalla samankaltaisiksi, ei takaa, että syöteavaruudessa kaksi hyvinkin lähekkäin sijaitsevaa pistettä kuvautuisivat samaan tai edes kahteen vierekkäiseen soluun. Käytännössä epäjatkuvuudet ovat kuitenkin seurausta siitä, että kartta on löytänyt jotain tilastollisesti merkittävämpää säännöllisyyttä.

2.2 Opetusalgoritmin ominaisuuksia

Opetusalgoritmi soveltuu suurillekin aineistoille, sillä sen aikavaativuus kasvaa lineaarisesti suhteessa opetusnäytteiden lukumäärään [Koh90] eikä se käytännössä edellytä muistitilaa kuin mallivektoreille ja kulloisellekin näytteelle. Toisaalta aikavaativuus kasvaa neljänteen potenssiin karttayksiköiden lukumäärän suhteen. Niinpä suurten karttojen opettaminen on hidasta, joskin menetelmää voidaan nopeuttaa monenlaisilla tekniikoilla. Esimerkiksi



Kuva 2: Kolmiulotteinen tasaisesti tiheä hahmo ja sitä epälineaarisesti approksimoiva kaksiulotteinen hila [Koh90].

WEBSOM-projektissa [Koh00] miljoonan yksikön kartta opetettiin 6,8 miljoonalla 500-ulotteisella näytevektorilla. (Tähän tosin kului aikaa yli 6 viikkoa kuuden prosessorin SGI O2000 -tietokoneella.)

Kun syöteaineisto on diskreetti ja naapuruusfunktio h_{bi} on kiinnitetty, itseorganisoituvan kartan opetusalgoritmi minimoi virhefunktiota

$$\sum_{i=1}^{N}\sum_{j=1}^{M} h_{bj} \|x_i - m_j\|^2 \tag{4}$$

missä N on opetusnäytteiden lukumäärä ja M on karttayksiköiden lukumäärä [Koh91]. Jos naapuruusfunktion arvo on 1 voittajayksikössä ja 0 muualla, opetusalgoritmi minimoi samaa virhefunktiota kuin tunnettu k:n keskiarvon menetelmä. k:n keskiarvon menetelmässähän minimoidaan

$$\sum_{k=1}^{C}\sum_{x \in Q_k} \|x - c_k\|^2 \tag{5}$$

missä C on muodostettavien klusterien lukumäärä, Q_k yksittäinen klusteri, x syötenäyte ja c_k klusterin k keskipiste. Muussa tapauksessa yhtälöstä (4) seuraa, että mallivektorit eivät ole *Voronoi-joukkojensa* keskipisteissä vaan ne ovat paikallisia, naapuruusfunktiolla *kaikkia* näytevektoreita painottaen saatuja keskipisteitä [Lam92]. Mallivektorin m Voronoi-joukolla tarkoitetaan kaikkia niitä näyteavaruuden pisteitä, joille m on lähin mallivektori ja jotka näin ollen kuvautuvat malliin m.

Todistukset sivuutetaan. Yhteys k:n keskiarvon menetelmään on kuitenkin selvä, kun tulkitaan karttahilan solut klustereiksi: k:n keskiarvon menetelmässä siirretään vain lähimpänä olevan klusterin keskipistettä kohti näytettä, kun taas itseorganisoituvalla kartalla mallivektorien (klusterien) keskipisteitä päivitetään sen mukaan, miten etäällä (h_{bi}) ne sijaitsevat voittajasolusta.

Monet klusterointialgoritmit käyttävät paikallisia kriteerejä, esimerkiksi klusterin Q sisällä saatetaan laskea kunkin siihen kuuluvan pisteen $x \in Q$ etäisyys lähimpään naapuriin $x' \in Q$ tai kahden klusterin Q, R välinen etäisyys etsimällä pisteet $x \in Q, y \in R$, jotka minimoivat klusterien Q ja R välisen etäisyyden. Kun klusteriin lisätään uusi näyte, joka on selvästi muista erillään, etäisyydet muuttuvat huomattavasti. Tämän herkkyyden välttämiseksi käytetään usein etäisyyksiä k lähimpään naapuriin. Koska itseorganisoituva kartta käyttää ryhmittelyssä implisiittisesti kaikkien etäisyyksien painotettua summaa, kyseistä ongelmaa ei esiinny.

3 Kartan klusterointi ja visualisointi

On kaksi tapaa nähdä kartta klusterointina:

1. tulkitaan jokainen solu klusteriksi tai

2. tulkitaan hilan *alueita* klustereiksi.

Jos jokainen solu tulkitaan klusteriksi, saavutetaan yksiselitteinen klusterointi, mutta klusterien sopiva lukumäärä pitää arvioida etukäteen. Yleensä klustereita halutaan pienehkö määrä, joten päädytään pieneen hilakokoon. Tällainen asetelma valitettavasti johtaa siihen, että jo hilan muoto vaikuttaa klusteroinnin laatuun ratkaisevasti. Lisäksi kaikista klustereista tulee väkisin konvekseja, koska näyte kuvautuu aina lähimpään malliin. Tämä tekniikka ei ole kovin suosittu.

Yleensä klusterien lukumäärää ei osata helposti arvioida eikä haluta rajoittua konvekseihin klustereihin. Tällöin voidaan valita suurehko hilakoko ja pitää klustereina sellaisia hilan alueita, joissa mallit ovat suhteellisen samankaltaisia. Hilakoon kasvattaminen todennäköisesti parantaa klusterointitarkkuutta. Klusterit muodostuvat useasta solusta yhdistäen niiden konveksit Voronoi-joukot suuremmiksi aliavaruuksiksi, joiden ei tarvitse olla konvekseja syöteavaruudessa. Klusterien havaitsemiseksi tarvitaan kuitenkin visualisointiin tarkoitettuja algoritmeja [Ult90, Kra95, Ves00].

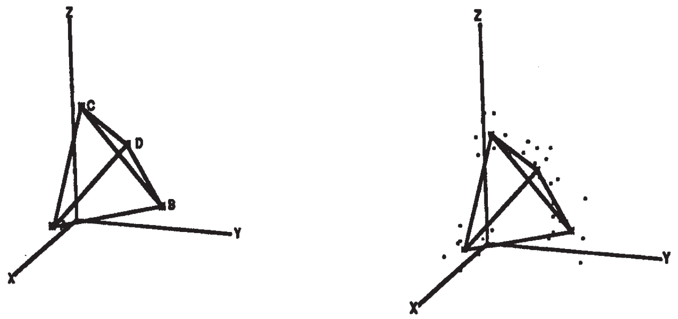
3.1 Klusterien visualisointi

On helppoa piirtää kaksiulotteinen säännöllinen hila, jolloin syntyy näkymä moniulotteiseen ja alunperin ehkä vaikeastikin tulkittavaan aineistoon, josta nyt saadaankin käsitys yhdellä silmäyksellä.

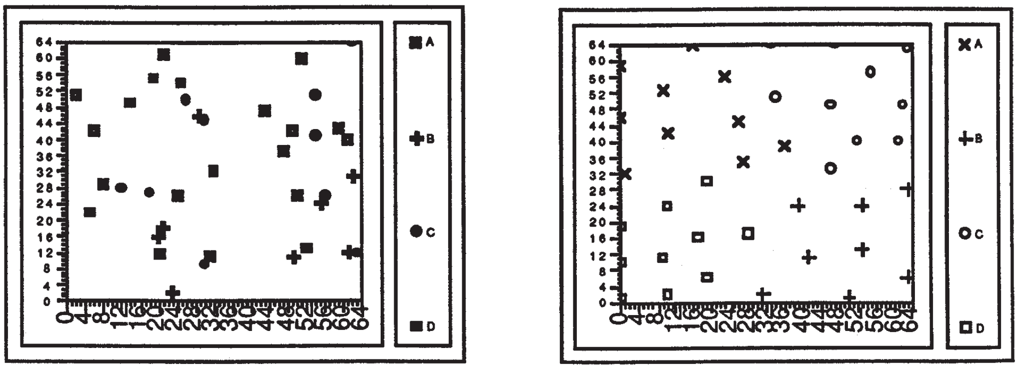
Mikäli kartta tuottaa topologiaa säilyttävän kuvauksen, niin oletettavasti myös syöteavaruuden klusterit eli samankaltaisten näytteiden joukot kuvautuvat hilalle niin, että samankaltaisuussuhteet säilyvät. Klusterien pitäisi siis näkyä myös hilalla. Ultsch ja Siemon [Ult90] havainnollistivat tilannetta tuottamalla seuraavanlaisen syötteen: neljä pistettä A, B, C, D muodostivat tetraedrin kolmiulotteisessa syöteavaruudessa, ja jokaisen pisteen ympäristöön tuotettiin 10 näytevektoria satunnaisen mutta rajoitetun etäisyyden päähän niin, että selvästi muodostui neljä klusteria. Kuva 3a esittää tetraedria ja kuva 3b klustereita. Klusterointialgoritmin pitäisi jakaa tällainen aineisto täsmälleen neljään osaan.

Kokeeksi 40 näytteestä muodostettiin 64×64 yksikön kartta. Kuva 4a esittää näytteiden sijoittumista satunnaisesti alustetulla hilalla ja kuva 4b tilannetta 50.000 opetusaskelen jälkeen [Ult90]. Mallivektorit järjestyvät niin, että klusterit sijoittuvat hilan eri kulmiin mahdollisimman kauas toisistaan. Näytteet kuitenkin kuvautuvat hilalle tasavälein, eikä mitään klusterointia olisi havaittavissa, ellei aineistoa olisi valmiiksi luokiteltu.

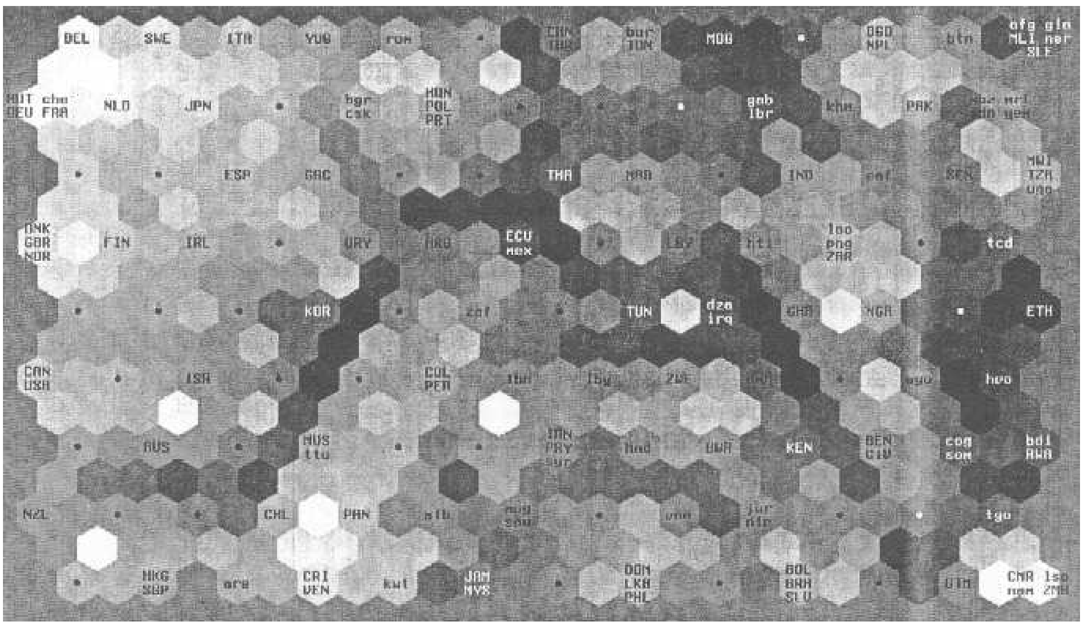
U-matriisi (engl. unified distance matrix) on Ultschin ja Siemonin ratkaisu ongelmaan ja suosittu tapa piirtää karttahila niin, että klusterit erottuvat. U-matriisi koostuu yksinkertaisesti karttayksiköiden ja niiden välittömien naapurien välisistä etäisyyksistä syöteavaruudessa mitattuna. Yleensä U-matriisi näytetään harmaasävykuvana tai kolmiulotteisena pintana [Ves99].



Kuva 3: (a) tetraedri ja (b) klusterit [Ult90].



Kuva 4: Kartta (a) ennen opetusta ja (b) 50.000 opetusaskelen jälkeen [Ult90].



Kuva 5: U-matriisi-menetelmällä tummennettu itseorganisoituva kartta [Koh95].

Kuvan 5 kartta käyttää U-matriisia [Koh95]. Se esittää eri maiden köyhyyttä tai oikeastaan köyhyystyyppejä, kun piirteiksi on valittu joukko elintasoon vaikuttavia mittareita. Jokaisen kuusikulmaisen solun tummuus vastaa ympäröivien etäisyyksien keskiarvoa. Musta väri ilmaisee suurinta vierekkäisten yksiköiden mallivektorien etäisyyttä ja valkoinen vastaavasti pienintä. Kartan vaaleat alueet ovat klustereita ja niitä erottavat tummat alueet klusterien välisiä rajoja. Näytteitä voidaan paikantaa hilalta tulostamalla kunkin solun kohdalle siihen kuvautuneisiin näytteisiin liitettyjä tekstejä.

Kraaijveld [Kra95] päätyi samanlaiseen menetelmään ja väritti neliön muotoisten solujen hilalla jokaisen solun niin, että solun tummuus riippui etäisyyksien maksimista neljään välittömään naapurisoluun. Lisäksi Kraaijveld näytti, että itseorganisoituvan kartan ryhmittely soveltuu luokittelualgoritmin syötteeksi paremmin kuin esimerkiksi perinteinen Sammonin kuvaus [Sam69] (joka kuvaa pisteitä pienempään ulottuvuuteen pyrkien säilyttämään kaikkien pisteparien etäisyyksiä). Ryhmittely sopii hieman paremmin lähimmän naapurin luokittimelle kuin lähimmän keskipisteen luokittimelle, koska kartan opetusalgoritmi tuottaa paljon paikallista erittelyä.

Kun uusi näyte kuvautuu sitä lähimpänä olevaan karttayksikköön, tämä voidaan tarvittaessa näyttää eri värillä. Kun karttayksiköitä on enemmän kuin opetusnäytteitä, hila interpoloi näytteiden välisiä alueita ja uusi näyte saattaa kuvautua ennestään tyhjään soluun. Suuri hilakoko mahdollistaa hyvän interpolointitarkkuuden.

3.2 Monivaiheinen klusterointi

U-matriisin paljastamat klusterit ovat kuitenkin vain ”katsojan silmässä”. Vaihtoehtona on syöttää opetetun kartan mallivektorit algoritmille, joka tekee varsinaisen yksiselitteisen klusteroinnin. Tällaisesta ratkaisusta on hyötyä, kun aineisto on hyvin laaja ja klusterointialgoritmi vaatii enemmän laskenta-aikaa kuin itseorganisoituvan kartan opetusalgoritmi. Esimerkiksi Vesanto ja Alhoniemi [Ves00] käyttivät karttaa hierarkkisten klusterointialgoritmien syötteenä, mikä johti tyydyttävään tarkkuuteen muutamassa minuutissa suurillakin aineistoilla, joiden suora klusterointi olisi muuten vaatinut aikaa useita tunteja.

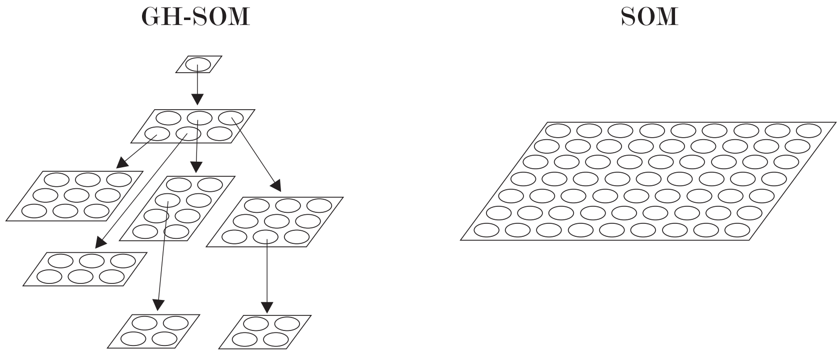
Joskus aineisto on niin laaja, että sitä täytyy tiivistää myös karttaa varten. Jos syöteaineisto on hyvin moniulotteinen, voidaan prosessointi jakaa useaan vaiheeseen aloittamalla esimerkiksi satunnaisprojektiosta [Kas98] ja etenemällä käyttäen yhä parempaan tarkkuuteen yltäviä tekniikoita. Käytettäessä karttaa muuhun kuin aineiston visualisointiin saattaa kannattaa muodostaa yli kaksiulotteinen kohdeavaruus, jossa aineisto on tarkemmin mallinnettavissa.

Kun itseorganisoituva kartta *kvantisoi* syöteavaruutta rajallisella joukolla vektoreita, se joutuu huomioimaan tihentymiä syöteavaruudessa harvojen alueiden kustannuksella. Tämän seurauksena se poistaa virheitä kohinaisesta signaalista ja soveltuu esiprosessointivaiheeksi esimerkiksi kohina-alttiille klusterointimenetelmille [Ves00].

4 Parannuksia ja muunnelmia

Vuosien varrella itseorganisoituvan kartan alkuperäisen mallin pohjalta on kehitetty lukuisia muunnelmia. Tässä yhteydessä tyydymme mainitsemaan niistä muutamia esimerkinomaisesti.

Tärkein parannus liittyy opetusvaiheen nopeuttamiseen. Opetuksen loppuvaiheessa mallivektorit ovat jo saavuttaneet keskinäisen järjestyksen hilalla, joten kun sama opetusnäyte syötetään lyhyen ajan jälkeen uudelleen, uusi voittaja löytyy todennäköisesti joko samasta solusta kuin edellisellä kerralla tai ainakin jostain edellisen voittajayksikön lähistöltä. Niinpä jokaiseen opetusnäytteeseen kannattaa liittää osoitin edelliseen voittajayksikköön, jolloin seuraava voittaja voidaan etsiä paikallisesti, aloittaen edellisen voittajayksikön sijainnista [Koh95]. Kun uusi voittaja on löydetty, osoitin päivitetään. Etsintä voidaan aloittaa valitsemalla osoitetun yksikön ja sen välittömien naapurien väliltä näytettä lähimpänä oleva malli. Etsintää jatketaan tämän yksikön ympäristöstä. Etsintä päättyy, kun mikään vie-



Kuva 6: Kasvava hierarkkinen itseorganisoituva kartta (GH-SOM) kasvaa monen karttahilan hierarkiaksi.

reisistä malleista ei ole enää lähempänä näytettä. Paikallinen haku ei välttämättä pääty globaaliin voittajaan, joten varmuuden vuoksi etsitään tietyin väliajoin globaali voittaja läpikäymällä kaikki kartan yksiköt.

Kartan konvergoituminen nopeutuu huomattavasti, jos mallivektorit saadaan alustettua valmiiksi jonkinlaiseen järjestykseen [Koh90]. Opetusalgoritmin aikavaativuudesta muistamme, että erityisesti suuren karttahilan opettaminen on hidasta. Eräs tapa nopeuttaa kartan muodostumista on opettaa yhä suurempia karttoja niin, että ensin opetetaan pieni hila ja interpoloidaan sen mallivektoreista hieman suuremman hilan alustus [Koh95]. Kutakin hilaa opetetaan, kunnes saavutetaan riittävä mallinnustarkkuus eli *vektorikvantisaatiovirhe* alittaa jonkin kynnysarvon.

Toiset parannukset tähtäävät parempaan mallinnustarkkuuteen. Edellä esitellyn kartan alkuperäisen opetusalgoritmin on muun muassa havaittu suosivan syöteavaruuden harvoja alueita tiheiden kustannuksella sikäli, että näytteiden todennäköisyysjakauma hilalla poikkeaa niiden jakaumasta syöteavaruudessa. DeSienon ”tietoisuusalgoritmi” (engl. conscience algorithm) [DeS88] pakottaa mallivektorien joukon noudattamaan syötteen todennäköisyysjakaumaa. Liian usein voittava vektori alkaa ”tuntea syyllisyyttä” ja vetäytyy kilpailusta.

Monet tutkijat ovat kokeilleet tapoja yhdistellä karttoja toisiinsa. Kartan mallivektoreita voidaan käyttää syötteenä toiselle kartalle, ja syötteeksi saatetaan valita vain hilan osia. Usein karttoja yhdistellään hierarkkisesti. Ensin saatetaan muodostaa pieni epätarkka kartta, jonka vektorit osittelevat syöteavaruuden. Mikäli jonkin karttayksikön sisällä esiintyy näytteissä suurta vaihtelua, muodostetaan siihen kuvautuneesta aineistosta uusi, tarkempi kartta.

Kasvavassa hierarkkisessa itseorganisoituvassa kartassa (GH-SOM) [Dit00] riittävää mallinnustarkkuutta haetaan kasvattamalla hilaa vähitellen opetusvaiheessa. Jos hilakoko lopulta ylittää jonkin kynnysarvon, sitä ei enää kasvateta, vaan sitten muodostetaan epätarkoista karttayksiköistä, joiden sisällä varianssi ylittää kynnysarvon, seuraavan tason karttoja ja aletaan opettaa ja kasvattaa niitä. Näin syntyy karttahilojen hierarkioita (kuva 6).

Karttojen hierarkia on verrattavissa hierarkkisen klusterointialgoritmin tuottamaan dendrogrammiin: jokaista dendrogrammin solmua vastaa kartta. Karttojen hierarkia kykenee pääsääntöisesti pienemmällä määrällä karttayksiköitä samaan mallinnustarkkuuteen kuin yksi kartta.

5 Yhteenveto

Itseorganisoituva kartta on topologiaa säilyttävä epälineaarinen kuvaus moniulotteisesta piirreavaruudesta pienempään ulottuvuuteen. Kuvaus muodostetaan iteratiivisella opetusalgoritmilla, jonka aikavaativuus riippuu enemmän kartan kuin syöteaineiston koosta. Ulottuvuuksien määrä ei vaikuta opetusaskelten määrään. Karttahilan soluihin on liitetty mallivektorit, jotka kartan opetuksen alkuvaiheessa saavuttavat keskinäisen järjestyksen niin, että lähekkäisissä soluissa mallit ovat samankaltaisia. Syöteavaruuden näyte kuvautuu sitä eniten muistuttavaan malliin, joten samankaltaiset näytteet kuvautuvat yleensä joko samoihin tai vierekkäisiin soluihin. Lisäksi mallivektorien joukko pyrkii edustamaan koko syöteaineistoa kattavasti. Opetusalgoritmi on eräänlainen toimintakuvaus saada tiedot järjestämään itse itsensä.

Usein käytetään kaksiulotteista hilaa, joka on helppo visualisoida värittämällä se esimerkiksi U-matriisin avulla. Visualisointiominaisuuksiensa ansiosta menetelmä soveltuukin erityisesti eksploratiiviseen data-analyysiin. Kartta toimii ikkunana moniulotteiseen syöteaineistoon ja paljastaa mahdollisia säännöllisyyksiä. Klusterit erottuvat esimerkiksi vaaleina alueina, joiden välissä on tummia juonteita. Klusterit ovat kuitenkin tulkinnanvaraisia, minkä vuoksi laadukas yksikäsitteinen klusterointi edellyttää jatkokäsittelyä.

Itseorganisoituva kartta kvantisoi mallivektoreillaan syöteavaruuden edustaviin osiin, minkä jälkeen mallivektorien joukko voidaan tulkita tiivistetyksi esitykseksi aineistosta ja se voidaan syöttää laskennallisesti vaativammille tai kohina-alttiille klusterointialgoritmeille, erityisesti sellaisille algoritmeille, joiden aikavaativuutta dominoi aineiston koko.

Käytännössä kartan alkuperäistä opetusalgoritmia käytetään harvoin sellaisenaan, koska on kehitetty monenlaisia tapoja muodostaa kartta tarkemmaksi ja nopeammin. Erityisesti opetuksen loppuvaiheessa kannattaa etsiä näytettä lähimpänä oleva malli paikallisesti siltä hilan alueelta, johon näyte viimeksi kuvautui.

Itseorganisoituva kartta on läheistä sukua k:n keskiarvon menetelmälle. Jos mallivektoreita ei päivitettäisi topologisina kokonaisuuksina, päädyttäisiin itse asiassa samaan optimointiongelmaan. Opetusalgoritmi käyttää implisiittisesti naapuruusfunktiolla painotettuja etäisyyksiä kaikkiin syöteaineiston näytteisiin, joten yksittäiset näytteet eivät juurikaan vaikuta kartan konvergoitumiseen.

Viitteet

[DeS88] Duane DeSieno. Adding a conscience to competitive learning. *Proc. ICANN'88, IEEE Int. Conf. on Neural Networks*, 1, 117–124, New York, 1988.

[Dit00] Michael Dittenbach, Dieter Merkl, Andreas Rauber. The growing hierarchical self-organizing map. *Proc. IJCNN 2000, Int. Joint Conf. on Neural Networks*, VI, 15–19, IEEE Computer Society Press, 2000.

[Kas98] Samuel Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering. *Proc. IJCNN'98, Int. Joint Conf. on Neural Networks*, 1, 413–418, IEEE Service Center, 1998.

[Koh82] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59–69, 1982.

[Koh89] Teuvo Kohonen. *Self-Organization and Associative Memory*, 3. painos, Springer-Verlag, 1989.

[Koh90] Teuvo Kohonen. The self-organizing map. *Proc. IEEE*, 78, 1464–1480, 1990.

[Koh91] Teuvo Kohonen. Self-organizing maps: optimization approaches. Teoksessa Teuvo Kohonen, Kai Mäkisara, Olli Simula, and Jari Kangas, toim., *Artificial Neural Networks*, 981–990, Elsevier, 1991.

[Koh95] Teuvo Kohonen. *Self-organizing maps.* Springer, 1995.

[Koh00] Teuvo Kohonen, et al. Self organization of a massive document collection. *IEEE Trans. Neural Networks*, 11, 574–585, 2000.

[Kra95] Martin Kraaijveld, Jianchang Mao. A nonlinear projection method based on Kohonen's topology preserving maps. *IEEE Trans. Neural Networks*, 6(3):548–559, 1995.

[Lam92] Jouko Lampinen J, Erkki Oja. Clustering properties of hierarchical self-organizing maps. *J. Mathematical Imaging and Vision*, 2(2–3):261–272, 1992.

[Rit95] Helge Ritter. Self-organizing feature maps: Kohonen maps. Teoksessa Michael Arbib, toim., *The Handbook of Brain Theory and Neural Networks*, 2–6, 846–851, The MIT Press, 1995.

[Sam69] John W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, C-18, 401–409, toukokuu 1969.

[Tur52] Alan Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society*, B, 237, 5–72, 1952.

[Ult90] Alfred Ultsch, H. Peter Siemon. Kohonen's self-organizing feature maps for exploratory data analysis. *Proc. INNC'90, Int. Neural Network Conference*, 305–308, Kluwer, 1990.

[Ves99] Juha Vesanto. SOM-based visualization methods. *Intelligent Data Analysis*, 3(2):111–126, Elsevier, 1999.

[Ves00] Juha Vesanto, Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Trans. Neural Networks*, 11, 586–600, 2000.