Klusterointi ensimmäisen kertaluvun logiikalla

Jaana Heino
Jaana.Heino@iki.fi

Klusterointimenetelmät-seminaari
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C–2002–54, s. 96–106, marraskuu 2002

Tiivistelmä

Tässä raportissa käsitellään klusterointia ensimmäisen kertaluvun logiikalla. Tällä tarkoitetaan sellaista klusterointia, jossa klusteroitavat objektit on kuvailtu 1. kertaluvun logiikalla, tai jossa menetelmä sinänsä hyödyntää logiikkaohjelmointia, tai jossa klusteroinnin tulokset ovat 1. kertaluvun logiikalla kuvailtuja, tai näiden yhdistelmää.

Raportissa esitellään kaksi menetelmää: Gilles Bissonin käsitteellinen bottom-up -klusterointi 1. kertaluvun objekteille ja Luc De Raedtin *et al.* esittämä päätöspuuoppimiseen perustuva menetelmä.

Eräs ensimmäisen kertaluvun klusterointiin liittyvä ongelma on se, että klusterointi jo määritelmällisesti sisältää käsityksen siitä, että "samankaltaisia" tai "toisiaan lähellä olevia" objekteja kootaan ryhmiksi. Etäisyysmittojen määrittely ensimmäisen kertaluvun logiikalla kuvatuille objekteille ei kuitenkaan ole yksinkertaista. Kun objektit eivät välttämättä koostu samasta määrästä attribuutteja ja niiden arvoja, normaalit arvovektoreille määritellyt euklidiset etäisyysmitat eivät sellaisenaan sovellu käyttöön.

Tämä ongelma tulee esiin myös kuvatuissa menetelmissä. Kumpikaan niistä ei esitä varsinaista etäisyysmittaa, vaan molemmat kiertävät ongelman (Bisson käyttämällä *ad hoc* samankaltaisuuskriteeriä, joka ei täytä etäisyysmitalle asetettavia vaatimuksia, De Raedt *et al* muotoilemalla ratkaisunsa niin, että se on käytetystä etäisyysmitasta irrallinen). Raportin lopuksi käsitellään vielä lyhyesti tätä ongelmaa ja eräitä sen mahdollisia ratkaisuja.

1 Johdanto

Useissa klusterointimenetelmissä oletetaan, että klusteroitavat objektit ovat kuvailtavissa arvovektoreina; toisin sanoen siten, että jokaisella objektilla on samat attribuutit ja ne eroavat vain näiden attribuuttien arvojen perusteella.

On kuitenkin useita sellaisia ongelmakenttiä, joissa tämä oletus ei pidä paikkaansa ja joissa objektien muuntaminen tällaiseen muotoon johtaa hankalaan esitystapaan ja/tai piirreavaruuden ulottuvuuksien kasvuun liian suureksi. Tällöin objektien sisäinen rakenne on erilainen; ne saattavat esimerkiksi koostua toisista objekteista, joiden määrää ei ole etukäteen määrätty. Esimerkiksi perheet koostuvat perheenjäsenistä, jotka voivat olla eri asemassa perheessä, ja joilla voi kullakin olla omia ominaisuuksia.

Ensimmäisen kertaluvun logiikka soveltuu tällaisten objektien kuvailuun [Bis92, BRJD99]. Sen käyttäminen tuo kuitenkin mukanaan useita ongelmia, sillä objektien kuvauskielen ilmaisuvoiman kasvaessa myös sen monimutkaisuus kasvaa, ja siten kasvavat myös algoritmien monimutkaisuus ja usein aikavaatimukset [BRJD99].

Erityisenä ongelmana on etäisyyskäsitteen määritteleminen rakenteisille objekteille; kun objekteilla ei välttämättä ole samoja tai samaa määrää ulottuvuuksia, tavanomaisesti käytetyt etäisyysmitat kuten euklidinen tai Manhattan-etäisyys eivät sellaisenaan ole mielekkäitä käsitteitä.

Tässä raportissa käsitellään erilaisia menetelmiä ensimmäisen kertaluvun objektien klusterointiin. Luvussa 2 palautetaan mieliin joitakin peruskäsitteitä liittyen logiikkaan, etäisyyksiin ja päätöspuihin ja todetaan päätöspuuoppimisen ja klusteroinnin välinen yhteys.

Luvussa 3 esitellään Gilles Bissonin esittämä KBG-algoritmi, bottom-up-tyyppinen algoritmi käsitehierarkioiden oppimiseen ensimmäisen kertaluvun objekteille ja luvussa 4 De Raedtin *et al.* esittämä myös klusterointiin soveltuva loogisia päätöspuita oppiva "puolen kertaluvun" järjestelmä C0.5. Luvussa 5 käsitellään etäisyysmittojen määrittelyä ensimmäisen kertaluvun objekteille.

2 Peruskäsitteitä

2.1 Ensimmäisen kertaluvun logiikka

Ensimmäisen kertaluvun logiikan *termit* muodostetaan numeroituvasta joukosta V muuttujasymboleita, joukosta C vakioita ja joukosta F 1- tai useampipaikkaisia funktioita, jotka palauttavat jonkin vakion. Termi on joko muuttuja, vakio tai muotoa $f(t_1, ..., t_n)$, missä $f \in F$ on n-paikkainen funktio ja t_i:t ovat termejä. [RB98, Vää87]

Olkoon P joukko 0- tai useampipaikkaisia predikaatteja (totuusarvon tosi tai epätosi palauttavia "funktioita") annetulle joukolle C. *Atomikaavat* ovat vakioita tai muotoa $p(t_1, ..., t_m)$ missä $p \in P$ on m-paikkainen predikaatti ja t_i:t termejä. [RB98, Vää87]

Kaavojen joukko $A_{V,C,F,P}$ muodostetaan nyt seuraavasti

1. Kaikki atomikaavat kuuluvat A:han.
2. jos $\mathbf{a} \in A$, niin $\neg\mathbf{a} \in A$
3. jos $\mathbf{a}, \mathbf{b} \in A$, niin $\mathbf{a} \wedge \mathbf{b} \in A$
4. jos $\mathbf{a}, \mathbf{b} \in A$, niin $\mathbf{a} \vee \mathbf{b} \in A$
5. jos $\mathbf{a} \in A$ ja $\mathbf{v_i} \in V$, niin $\exists\mathbf{v_i} : \mathbf{a} \in A$
6. jos $\mathbf{a} \in A$ ja $\mathbf{v_i} \in V$, niin $\forall\mathbf{v_i} : \mathbf{a} \in A$

[Vää87]

Näille oletetaan ilmeinen tulkinta (tarkemmin ottaen Tarskin totuusmääritelmän mukainen tulkinta, ks. esim. [Vää87]).

Tässä raportissa käsitellyissä klusterointijärjestelmissä käytetään sekä rajoitteita että laajennuksia tälle tunnetulle määritelmälle. Ensinnäkin oletetaan, että käytössä on lisäksi keino ottaa huomioon *kvantitatiivista* informaatiota numeeristen arvojen muodossa; tällöin predikaattien ja funktioiden sallitaan operoida myös näillä arvoilla.

Sallittuja operaatioita voidaan myös rajoittaa. Esimerkiksi Bissonin KBG-järjestelmä [Bis92] ei salli negaatioita eikä funktiosymboleja. Yleisesti on järkevää myös rajoittaa objektien kuvailua siten, että ne muodostuvat kokoelmista atomikaavoja (ie, vakioita, predikaatteja ja niiden negaatioita, siten että kaavojen välille oletetaan implisiittinen konjunktio ja muuttujat ovat eksistentiaalisesti kvantifioituja; tämä vastaa jotakuinkin logiikan teorian "mallin" käsitettä). Itse klusterointimenetelmä tai sen tulosten kuvailu voi silti hyödyntää monimutkaisempiakin lauseita.

Yllä mainittiin perheet esimerkkinä objekteista, joita voisi olla luontevaa kuvailla 1. kertaluvun logiikalla. Esimerkiksi kirjoittajan perhettä voitaisiin kuvailla lyhyesti seuraavasti:

$\{A, B, C, D, lapsi(C), lapsi(D),$

$\forall lapsi(v_1) : isa(A, v_1) \wedge aiti(B, v_1),$

$nuorempi(D, C), nuorempi(B, A)\}$

tai, jos rajoitetaan kuvailukieltä atomikaavoihin:

$\{A, B, C, D, lapsi(C), lapsi(D),$

$isa(A, C), isa(A, D), aiti(B, C), aiti(B, D),$

$nuorempi(D, C), nuorempi(B, A)\}$

2.2 Etäisyysmitta

Etäisyysmitta eli *metriikka* joukolle E on funktio $d : E \times E \to \mathbb{R}$ siten että kaikille $x, y, z \in E$:

1. $d(x, y) \geq 0$,
2. $d(x, y) = 0 \Leftrightarrow x = y$,
3. $d(x, y) = d(y, x)$,
4. $d(x, z) \leq d(x, y) + d(y, z)$

Tunnetusti, viimeistä ehtoa sanotaan kolmioepäyhtälöksi, ja se sanoo, että kahden pisteen välinen suora etäisyys ei voi olla pitempi kuin näiden pisteiden välinen matka kuljettaessa jonkin toisen pisteen kautta. Tämä on klusteroinnissa huomattavan tärkeä ominaisuus; se mm. takaa sen, että mikäli tietyn klusterin kaikkien objektien etäisyys tietystä pisteestä on korkeintaan n, kahden klusteriin kuuluvan objektin välinen etäisyys on myös rajattu (on korkeintaan $2n$).

Tulemme näkemään, että ensimmäisen kertaluvun logiikalla kuvailtujen objektien välille ei ole yksinkertaista määritellä samankaltaisuutta *järkevällä tavalla kuvaavaa* funktiota, joka täyttäisi etäisyysmitan ehdot, erityisesti kolmioepäyhtälön. (Paitsi aito etäisyysmitta, samankaltaisuuskriteerin pitäisi tietysti olla mielekäs siinä suhteessa, että se kuvaa jotain todellista ja mielenkiintoista ilmiötä objektien suhteissa toisiinsa.)

2.3 Prototyypit ja yleistykset

Prototyyppi tai *yleistys* (engl. prototype, generalization) on käsite, joka vastaa käyttötarkoituksiltaan attribuutti-arvo-pohjaisten klusterointimenetelmien käsitettä "klusterin keskipiste" tai "klusterin edustaja". Sen tarkka määrittely riippuu lähteestä - kuten usein klusterin edustajankin määrittely.

Olkoon O kaikkien mahdollisten objektien joukko.

Bisson määrittelee binäärisen yleistyksen funktioksi $g : O \times O \to O$ siten, että sen palauttama looginen objekti "kattaa" (*subsumes*) molemmat arvot [Bis92]. Esimerkiksi jos $o_1 = \{a, p(a)\}$, $o_2 = \{b, p(b)\}$, niin yksi mahdollinen yleistys olisi $\exists v_1 : p(v_1)$, missä a ja b ovat vakioita ja v_1 on muuttuja.

Ramon ja Bruynooghe [RB98] puolestaan määrittelevät prototyypin jonkin etäisyysmitan suhteen seuraavasti: olkoon $S = \{o_1, ..., o_n\} \subset O$ ja d jokin etäisyysmitta O:ssa. Tällöin $o \in O$ on *euklidinen prototyyppi* S:lle jos ja vain jos $\sum_{i=1}^{n} d(o_i, o)^2 = \min_{p \in O} \sum_{i=1}^{n} d(o_i, p)^2$. Toisin sanoen, o on euklidinen prototyyppi S:lle, jos ja vain jos o on se avaruuden objekti, johon kaikkien S:n objektien yhteenlaskettujen etäisyyksien neliö on pienempi tai yhtä suuri kuin mihinkään muuhun objektiin.

Euklidisen prototyypin käsite vastaa täysin euklidisen keskipisteen käsitettä ja on sikäli intuitiivinen; ollakseen mielekäs se vaatii kiutenkin sekä aidon etäisyysmitan että keinon konstruoida looginen objekti, joka *kaikkien mahdollisten* objektien joukossa toteuttaa annetun ehdon. Tämä kaikkien mahdollisten objektien joukko on yleensä kuitenkin ääretön, eikä konstruointi ole välttämättä mahdollista.

Valitsemalla etäisyyden minimoiva objekti kaikkien kuviteltavissa olevien objektien O sijaan joukosta S (etsimällä se objekti jolle $\sum_{i=1}^{n} d(o_i, o)^2 = \min_{p \in S} \sum_{i=1}^{n} d(o_i, p)^2$) päästään *euklidisen medoidin* käsitteeseen [KW00]. Tämän objektin laskenta vaatii "vain" mielekkään etäisyysmitan olemassaolon, mutta toisaalta valittu objekti ei välttämättä missään mielessä edusta objektijoukon S keskustaa käytetyn etäisyysmitan mielessä (esimerkiksi asettuvat kaksiulotteisessa avaruudessa tasaisesti ympyrän kehälle, ei mikään niistä ole euklidisessa mieelssä lähempänä keskustaa kuin joku toinen).

2.4 Päätöspuut ja klusterointi

Päätöspuut ovat alunperin luokitteluongelmiin kehitetty rakenne. Niiden perusidea on seuraava. Kussakin sisäsolmussa on testi, ja jokaista testin mahdollista tulosta vastaa yksi solmun lapsista. Luokiteltaessa uutta objektia edetään puun juuresta, ja suoritetaan testejä yksi kerrallaan siirtyen aina siihen lapseen, joka vastaa testin tulosta. Puun lehdessä on tätä objektia vastaava luokka (tai todennäköisyysjakauma joka kuvaa sitä, mihin luokkaan objekti luultavasti kuuluu).

Osittavan päätöspuuoppimisen (top-down learning of decision trees) perusidea on myös yksinkertainen. Aloitetaan niin, että kaikki opetusesimerkit (eli klusteroitavat objektit) ovat puun juuressa (toistaiseksi ainoassa solmussa). Etsitään jollain kriteerillä yksittäinen testi, joka parhaiten jakaa esimerkit luokkiin. Solmuun laitetaan tämä testi, ja esimerkit jaetaan solmun lapsille sen mukaan. Toimintaa jatketaan rekursiivisesti, kunnes joku lopetusehto saavutetaan - esimerkiksi solmun esimerkeistä kaikki tai tietty prosentti kuuluvat samaan luokkaan, ja/tai solmussa on vähemmän kuin tietty määrä esimerkkejä. (Tämän jälkeen puuta voidaan modifioida turhien haarojen karsimiseksi jne.)

On helppo nähdä, että syntyvä päätöspuu on itseasiassa hierarkkinen klusterointi käytetyille opetusesimerkeille; kuhunkin lehteen päätyvät objektit muodostavat yhden klusterin. Kun lisäksi huomataan, että testinvalintakriteerinä voidaan käyttää luokkainformaation sijaan jotain klusteroinnista tuttua kriteeriä - esimerkiksi sitä, että kuhunkin lapseen joutuvien objektiryhmien välinen etäisyys on mahdollisimman suuri - on selvää, että tällainen päätöspuuoppiminen on eräs klusterointimenetelmä.

3 KBG - kokoava menetelmä ja yritys etäisyysmitaksi

Tässä luvussa käsitellään yksityiskohtaisesti eräs yritys määritellä etäisyysmittaa ensimmäisen kertaluvun logiikalle. Käsittely on suhteellisen yksityiskohtainen, eikä esitetty metriikka ole kaikilta osin tyydyttävä; se toimii lähinnä esimerkkinä toisaalta siitä, mitä kaikkea voitaisiin ja täytyisi ottaa huomioon ja toisaalta siitä, mikä voi mennä pieleen. On suositeltavaa, että lukija ei tuhlaa aikaa menetelmän yksityiskohtien ymmärtämiseen, jos ne eivät helposti aukea.

3.1 Menetelmän kuvaus

Gilles Bissonin vuonna 1992 julkaisema menetelmä KBG oppii käsitehierarkioita objekteille, jotka on kuvailtu eräällä ensimmäisen kertaluvun logiikan variaatiolla. Objekteja kuvailtaessa ei käytetä negaatioita eikä funktiosymboleja; toisaalta kieltä on laajennettu käsittelemään kvantitatiivista informaatiota. [Bis92]

Itse järjestelmä on alunperin kehitetty oppimaan käsitehierarkioita; jätän tässä käsittelemättä siitä suurimman osan ja kuvailen vain osat jotka suoraan ovat tekemisissä klusteroinnin ja etäisyyden määritelmän kanssa.

Bissonin *atomi* on 1-tai useampipaikkainen predikaatti, jonka argumentit ovat tyypitettyjä. Argumentit voivat olla entiteettejä, joiden merkitys riippuu ainoastaan siitä, mitä predikaatit niistä sanovat, tai ne voivat olla arvoja, joilla on semantiikka itsessään. Bisson esittää itse esimerkkinä atomista mm. ilmauksen $age(John, 58)$. Tässä John on entiteetti ja 58 on arvo.

Kaikki arvot ovat tyypitettyjä. Käyttäjä voi vapaasti määritellä uusia tyyppejä, mutta kullekin tyypille on määriteltävä kaksi funktiota. Ensimmäinen näistä, similariteettifunktio, $VSIM_T : T \times T \rightarrow [0...1]$ määrittelee ”samankaltaisuuden” asteen kahden eri arvon välillä. Toinen, yleistysfunktio, $VGEN_T : T \times T \rightarrow T$ palauttaa arvon joka kattaa molemmat arvot.

Tällä tasolla, kuvailtaessa yksittäisiä arvoja, on näiden funktioiden määrittely vielä suhteellisen helppoa, sillä voidaan käyttää esimerkiksi erotuksia, loogisia operaattoreita, keskiarvoja tai skaaloja.

> Esimerkiksi tyypin ”ikä vuosina” parittainen similariteetti voitaisiin määritellä $VSIM_{age}(V1, V2) = (99 - |V1 - V2|)/99$ ja sukupuolen similariteetti vaikkapa $VSIM_{sex} = 1$, jos $V1 = V2$, muuten 0.

Klusteroitavat objektit koostuvat kukin (epätyhjästä) joukosta entiteettejä ja joukosta atomeja jotka kuvailevat näitä entiteettejä tai niiden suhteita.

> Esimerkiksi kahta perhettä kuvailevat objektit voisivat olla:
> $E1 = father(paul, yves), sex(yves, male), age(yves, 13), age(paul, 33), french(paul)$
> $E2 = father(john, ann), sex(ann, female), age(ann, 28), age(john, 58), american(john)$

Itse menetelmä on suhteellisen yksinkertainen: siinä iteratiivisesti etsitään objektien joukosta ne kaksi, jotka ovat eniten keskenään samanlaiset. Ne liitetään yhteen, ja niistä muodostetaan yleistys. Objektit poistetaan klusteroitavien joukosta ja korvataan yleistyksellä.

Näin jatketaan kunnes jäljellä on enää yksi yleistys koko joukosta. Tuloksena on binääripuu, jossa kussakin lehdessä on yksi alkuperäinen objekti ja jonka sisäsolmut vastaavat eri karkeustasojen klustereita objekteille.

(Bisson esittää toisessa artikkelissa myös monimutkaisemman kokoamismenetelmän, jonka tuloksena on suunnattu verkko puun sijaan. Oleellinen osuus tästä menetelmästä voidaan kuitenkin selostaa tämän yksinkertaisen algoritmin avulla, joten monimutkainen versio jätetään käsittelemättä.)

Mielenkiintoinen osuus Bissonin menetelmää on se, miten päästään arvoille määritellyistä similariteeteista ja yleistyksistä vastaaviin suureisiin *objekteille.*

Tiivistettynä, samankaltaisuuden laskenta KBG:ssä lähtee liikkeelle siitä, että verrataan kaikkia objektissa E1 esiintyviä entiteettejä kaikkiin objektissa E2 esiintyviin entiteetteihin pareittain. Näiden samankaltaisuus lasketaan sen perusteella, kuinka paljon entiteeteissä esiintyy niitä koskevia predikaatteja, ja kuinka samankaltaisia nämä predikaatit ovat. Tämän jälkeen etsitään sellainen paritus E1:n ja E2:n entiteeteille, että pareiksi joutuneiden entiteettien välinen samankaltaisuus on mahdollisimman korkea. Näiden samankaltaisuuksien keskiarvo otetaan objektien väliseksi samankaltaisuudeksi.

Tarkemmin sanottuna etäisyys kahden eri objektissa esiintyvän entiteetin välillä lasketaan seuraavasti. Olkoot X ja Y kaksi objektissa $E1$ ja $E2$ esiintyvää entiteettiä ja T_k jokin predikaatti, joka esiintyy molemmissa objekteissa. Olkoon n tämän predikaatin entiteettiargumenttien määrä ja m sen arvoargumenttien määrä. Merkitään $T_k(A_1, ..., A_n, V_1, ..., V_m)$ sitä predikaatin T_k instanssia, joka esiintyy objektissa E1, ja vastaavasti $T_k'(B_1, ..., B_n, U_1, ..., U_m)$ sitä instanssia joka esiintyy objektissa E2. Tässä A_i, B_i ovat predikaatin entiteettiargumentit ja V_i, U_i sen arvoargumentit.

Nyt

$$TSIM(T_k, T_k') = (\frac{1}{n}\sum_{i=1}^{n} s(A_i, B_i) * \frac{1}{m}\sum_{j=1}^{m} s(U_j, V_j))$$

siten että

$s(A_i, B_i) = 1$, jos $A_i = X$ ja $B_i = Y$,

$s(U_j, V_j) = VSIM_T(U_j, V_j)$, ja

$s(A_i, B_i) = SIM(A_i, B_i)$ jos $A_i \neq X$ tai $B_i \neq Y$

(ja tulkiten, että jos entiteetti- tai arvoargumentteja ei ole, vastaava osa kaavasta jätetään kokonaan pois).

Toisin sanoen, laskettaessa objekteissa esiintyvien *predikaattien* samankaltaisuutta näiden kahden *entiteetin* suhteen, verrataan predikaatissa samoissa paikoissa esiintyviä argumentteja toisiinsa. Jos argumentit ovat nämä entiteetit, ne saavat maksimisamankaltaisuuden. Jos entiteettiargumenteista jompikumpi tai molemmat ovat jotain muita entiteettejä, käytetään niiden välistä samankaltaisuutta. Arvoargumenttien suhteen samankaltaisuus lasketaan niille määritellyllä samankaltaisuusfuntkiolla. Tulos jaetaan argumenttien määrällä.

> Ylläesitetyssä objekteissa predikaatissa *father* on kaksi entiteettiargumenttia. Näistä toisessa entiteetit $Paul$ ja $John$ esiintyvät samassa paikassa (samankaltaisuus on 1). Toisessa esiintyvät $Yves$ ja Ann. Predikaatin *father* samankaltaisuudeksi Paulille ja Johnille saadaan siis $\frac{1}{2}(1 + SIM(Yves, Anne))$. Vastaavasti *age*-predikaatille saadaan samankaltaisuudeksi $VSIM_{age}(33, 58) = 0,75$. Muita yhteisiä predikaatteja Johnilla ja Paulilla ei olekaan.

Kahden entiteetin samankaltaisuus kahden *objektin* puitteissa määritellään seuraavasti:

$$SIM(X,Y)\frac{\sum_{T_i \in CO} TSIM(T_i, T_i')}{max(Occ(X), Occ(Y))}$$

missä CO on niiden predikaattien joukko, joissa E1:ssä esiintyy X ja E2:ssa Y, ja $Occ(X)$ on niiden predikaattien määrä, joissa X esiintyy E1:ssä.

> Kaikki perhe-esimerkissä esiintyvät entiteetit ovat (E1:ssä) $Paul$ ja $Yves$ sekä (E2:ssa) $John$ ja Ann. Meidän on siis laskettava neljä samankaltaisuutta entitettien välille: $Paul - John$, $Yves - John$, $Paul - Ann$ ja $Yves - Ann$. $Occ(Paul) = 3$, sillä Paul esiintyy E1:ssä 3:ssa predikaatissa. Vastaavasti $Occ(John)$ on myös 3. Johnin ja Paulin keskinäiseksi samankaltaisuudeksi saadaan siis $((1/2(1 + SIM(Yves, Anne) + 0,75)/3$.

Nyt huomataan, että jotta voidaan määritellä kahden entiteetin välinen samankaltaisuus näiden objektien puitteissa, saatetaan tarvitan kahden muun entiteetin välistä samankaltaisuutta. Näistä lausekkeista muodostuu yhtälöryhmä, joka ratkaisemalla saadaan samankaltaisuudet.

Tämän jälkeen, kuten mainittiin, etsitään E1:n ja E2:n entiteettien välille sellainen paritus, että samankaltaisuuksien summa maksimoituu. E1:n ja E2:n väliseksi samankaltaisuudeksi tulee näiden samankaltaisuuksien keskiarvo. (Tarkkaan ottaen Bissonin malli antaa vielä mahdollisuuden painottaa predikaatteja eri painoilla, mutta selkeyden vuoksi se on jätetty tästä esityksestä pois.)

> Perhe-esimerkin yhtälöt ratkaisemalla saadaan $SIM(John, Paul) = 50,5\%$, $SIM(Yves, Anne) = 53,4\%$, $SIM(Paul, Anne) = 31,7\%$ ja $SIM(Yves, John) = 18,2\%$. Tästä nähdään, että paras paritus on $Yves - Anne$ ja $John - Paul$, mistä saadaan $SIM(E1, E2) = (53,4 + 50,5)\%/2 = 52\%$.

Sama laskenta toistetaan jokaiselle objektiparille ja tulokset kootaan matriisiin. Tämän jälkeen oppiminen voidaan suorittaa kokoamalla näitä objekteja puiksi, kuten edellä on kuvattu.

Yleistysaskel on periaatteessa hyvin yksinkertainen: objektit yhdistetään aiemmin lasketun parhaan parituksen mukaisesti korvaten yhteisissä predikaateissa ilmenevät parit yhdellä muuttujalla; ne predikaatit joille ei löydy paria jätetään pois uudesta yleistetystä objektista. (Esimerkiksi, jos X ja Y ovat eräs pari parhaassa entiteettien parituksessa, ja toisessa objektissa esiintyy predikaatti $age(X, 30)$, mutta toisessa ei esiinny predikaattia $age(Y, n)$ jätetään age-predikaatti kokonaan pois yleistetystä objektista. Tämä on tulkittava tarkoittamaan, että X:ää ja Y:tä vastaavan entiteetin ikä voi olla mikä hyvänsä).

Yleistyksen jälkeen muotoillaan uusi täsmälleen yleistyksen täyttävä objekti täysin uusin entiteetein, ja yhdistetyt objektit korvataan sillä. Tämä tehdään siksi, että samankaltaisuuskriteeri on määritelty entiteeteille, ei muuttujille, joten muuttujia sisältävän yleistyksen käyttö iteraation seuraavalla askeleella olisi mahdotonta. Yleistys voidaan kuitenkin säilyttää klusterin kuvauksena.

3.2 Kommentteja menetelmästä

Bissonin menetelmä on sikäli merkittävä, että siinä on pyritty määrittelemään jonkinlaista "etäisyys" mittaa ensimmäisen kertaluvun objekteille.

Esitetyn samankaltaisuuskriteerin eräs huomionarvoinen piirre on sen monitasoisuus: ensin määritellään samankaltaisuus arvoille, sen perusteella entiteeteille ja predikaaterille tietyissä objekteissa, ja tämän jälkeen edellisen askeleen perusteella objekteille itselleen. Tämä moniportaisuus esiintyy myös muissa vastaavissa samankaltaisuuskriteerien määrittelyissä [RB98], ja kuvastaa ajatusta siitä, että rakenteisten objektien välisen samankaltaisuuden määritelmän pitäisi perustua niiden sisältämien samankaltaisten osien suhteisiin, ennemmin kuin esimerkiksi osien määrään tai yksittäisten osien ominaisuuksiin.

Bissonin kriteerin ongelma on kuitenkin se, että se ei ole etäisyysmitta. Jo Bissonin artikkelissa esiintyvästä esimerkistä ([Bis92], s. 461) on helppo havaita, että se ei mm. täytä kolmioepäyhtälöä. Siinä esimerkiksi E2:n ja E4:n välinen samankaltaisuus on 14 %, mutta E2:n ja E1:n välinen 52 % ja E1:n ja E4:n 28 %, eli samankaltaisuus E2:n ja E4:n välillä ”E1:n kautta” on huomattavasti korkeampi kuin ”suoraan”. (Huom: Bissonin mitta on samankaltaisuusmitta, eli mitä suurempi luku, sitä *lähempänä* objektit ovat - koska mitta on normalisoitu välille 0-100 %, vastaa etäisyyttä 0 samankaltaisuus 100 %.)

Bissonin järjestelmällä on ilmeisesti saavutettu suhteellisen hyviä käsitteellisen oppimisen tuloksia. Varsinaisia tuloksia klusteroinnille ei kuitenkaan ole esitetty. Missään tapauksessa esitettyä samankaltaisuuskriteeriä ei kuitenkaan voi yleistää käytettäväksi muissa klusterointimenetelmissä, johtuen sen puutteista etäisyysmittana.

4 C0.5 - päätöspuuklusterointia ”puolen kertaluvun” logiikalla

4.1 Menetelmän kuvaus

Luc de Raedt ja Hendrik Blockeel esittävät artikkelissaan [RB97] ja laajemmassa raportissa muiden kirjoittajien kanssa [BRJD99] menetelmän, joka hyödyntää 1. kertaluvun logiikkaa päätöspuuoppimisessa.

Heidän kontekstissaan jokainen objekti on kokoelma faktoja, jotka ilmaistaan kokoelmina predikaatteja. Esimerkkinä he esittävät mm. Bongard-ongelmina tunnettujen hahmontunnistuskuvioiden kuvailun. Nämä kuviot koostuvat geometrisista tasokuvioista, jotka voivat sijaita eri päin ja toistensa sisällä. Eräs tällainen kuvio voitaisiin kuvailla esim. [BRJD99]

$\{circle(c_1), triangle(c_2), points(c_2, up), inside(c_2, c_1)\}$

missä c_1 ja c_2 ovat vakioita. Tässä kuvailukielessä ei sallita negaatioita tai kvanttoreita.

De Raedtin *et al.* klusterointi perustuu *ensimmäisen kertaluvun loogisten päätöspuiden* oppimiseen. Ensimmäisen kertaluvun päätöspuu on sellainen binääripäätöspuu, jossa kussakin solmussa sijaitsee jokin looginen kysely, eli lause, joka joko on tosi tai ei ole tosi annetuille faktajoukoille [RB97, BRJD99]. Esimerkiksi kysely $\exists X : triangle(X)$ on tosi ylläkuvatulle objektille. Kyselyt saavat myös jakaa muuttujia puussa ylempänä olevien solmujen kanssa [RB97, BRJD99]. Esimerkiksi jos edellä mainittua kyselyä seuraisi solmu, jossa kyselynä olisi $points(X, down)$, tämä viittaisi nimenomaan siihen, että ylemmässä solmussa kiinnitetylle X:lle pätee tämä väite (jonkun toisen kuvion sijaan).

Esitettyä menetelmää ei ole kiinnitetty etäisyysmitan ja prototyyppifunktion suhteen, vaan pidetään annettuna, että on käytettävissä *jokin* etäisyysmitta $d(e_1, e_2)$, missä e_1 ja e_2 ovat objekteja, ja *jokin* prototyyppifunktio $p(E)$, joka palauttaa prototyypin annetulle joukolle E objekteja. Kahden klusterin välinen etäisyys määritellään silloin etäisyydeksi $d(p(E_1), p(E_2))$. [RB97]

On huomattava, että vaikka objektit on kuvattu logiikalla, ja käytetty päätöspuuoppiminen sinänsä hyödyntää 1. kertaluvun logiikkaa, ei ole mitenkään välttämätöntä, että nämä etäisyys- ja prototyyppimitat määritellään 1. kertaluvun logiikan objekteille. Olisi täysin mahdollista esimerkiksi määritellä etäisyys euklidisena etäisyytenä joillekin sellaisille numeerisille faktoille, jotka jokaisella objektilla väistämättä on, ja prototyyppi vastaavasti näiden euklidisten pisteiden keskiarvona. Tähän tapaan esitetty menetelmä itse asiassa toimiikin; vaikka olisikin teoriassa mahdollista käyttää menetelmässä mitä tahansa etäisyysmittaa, myös 1. kertaluvun objekteille suoraan määriteltyä sellaista, sitä ei ole tehty. Siksi tekijät itsekin kutsuvat menetelmäänsä "puolen kertaluvun" logiikaksi. [RB97]

Kuhunkin solmuun liittyvä kysely on konjunktio niistä kyselyistä, jotka sijaitsevat polulla juuresta kyseiseen solmuun. Käyttäjä määrittelee käytettävän kielen - mahdollisten faktojen joukon ja sen, mitä kyselyjä kuhunkin solmuun voidaan liittää. Esimerkiksi sääntö $8 : p(+X, -Y, Z), q(Z)$ määrittelee, että konjunktio $p(X, Y, Z), q(Z)$ voi esiintyä solmuun liittyvässä kyselyssä maksimissaan 8 kertaa, ja että jaettassa lehtisolmua uusiksi lapsiksi on X tulkittava joksikin jo kyselyssä esiintyväksi muuttujaksi, Y voidaan tulkita sellaiseksi, ja Z on uusi muuttuja. [RB97]

Käytetään edelleen samaa esimerkkiä geometrisista kyselyistä kuin yllä; jos järjestelmällä olisi tällainen sääntö mallia $inside(+Y, Z)$, voitaisiin puun seuraavan tason solmuun liittää jakosääntö $inside(X, Y)$ (joka olisi tosi yllä annetulle objektille, kiinnitetyn X:n sisällähän on jokin toinen objekti). Mainittakoon vielä, että ellei toisin mainita, ovat kaikki solmujen lauseet eksistentiaalikvanttorilla varustettuja: edelläoleva on siis tulkittava $\exists Y : inside(X, Y)$, kun X on jo aiemmin kiinnitetty, mutta Y:tä ei.

Opittaessa päätöspuuta generoidaan jokaisessa solmussa kaikki annetun sääntöjoukon puitteissa mahdolliset kyselyt. Kukin kysely jakaa solmun objektit kahteen osaan: niille joille kysely on tosi ja niihin joille se on epätosi. Kyselyistä solmun jakavaksi kyselyksi valitaan se, joka maksimoi lapsiin syntyvän interklusterietäisyyden $(d(p(E_{tosi}), p(E_{epatosi}))$. [RB97, BRJD99]

Kuten päätöspuuoppimisessa yleensä, solmujen jakamista jatketaan iteratiivisesti, kunnes jokin pysähtymiskriteeri täyttyy. Selvästikin jakoa ei enää tehdä, jos lehdessä on vain yksi objekti, tai jos mahdollisia sääntöjä ei voida enää generoida. Yleensä kuitenkin halutaan pysähtyä jo aikaisemmin, siinä vaiheessa, kun yhdessä lehdessä olevat objektit ovat "riittävän samankaltaisia". Yksi mahdollinen kriteeri tälle on

$$w \times \sum_{e \in E} d(e, p(E)) \leq \sum_{e \in E_{tosi}} d(e, p(E_{tosi})) + \sum_{e \in E_{epatosi}} d(e, p(E_{epatosi}))$$

eli toisin sanoen solmua ei enää jaeta, jos pisteiden keskinäinen etäisyys syntyvien klusterien sisällä ei enää pienene tarpeeksi (painokertoimella $w \in (0, 1]$ voidaan vaikuttaa siihen, kuinka paljon etäisyyden halutaan putoavan). [RB97]

4.2 Kommentteja menetelmästä

Esitetty päätöspuuoppimisalgoritmi tuntuu järkevältä tavalta klusteroida objekteja, joita voidaan kuvailla 1. kertaluvun logiikalla. Sen hyvänä puolena ovat mm. se, miten jaettaessa klustereita subklustereihin voidaan hyödyntää suhteita, joita on aikaisemmin luotu objektien entiteettien välille (solmut voivat jakaa muuttujia), mikä lisää merkittävästi päätöspuiden ilmaisuvoimaa. Siitä huolimatta oppiminen on *lokaalisti* kussakin solmussa suhteellisen yksinkertaista. Toinen hyvä puoli on se, että syntyvä rakenne suoraan antaa 1. kertaluvun kuvailun klustereille: kuhunkin solmuun johtavaan polkuun liittyvä kysely on samalla kuvaus niistä objekteista jotka voivat kuulua ko. solmun klusteriin [RB97].

Toisaalta tuntuu rajoitteelta, että käyttäjän on etukäteen määriteltävä tavat, joilla solmujen kyselyjä voidaan laajentaa. Vaikka osa tällaisista laajennostavoista onkin epäilemättä itsestäänselvästi havaittavissa tarkastelemalla objekteja joita ollaan klusteroimassa, sääntöjä voidaan muotoilla niin monella tavalla, että tämän vaikutuksen tuloksiin täytyy olla huomattava.

De Raedt *et al.* eivät myöskään käsittele mahdollisesti kaikkein hankalinta osa-aluetta 1. kertaluvun logiikalla klusteroinnissa lainkaan; nimittäin sitä, miten etäisyysmitta olisi luontevinta heidän kontekstissaan määritellä.

Artikkelin [RB97] lopussa esitetään joitakin kokeellisia tuloksia menetelmälle. Ne vaikuttavat lupaavilta, mutta esityksestä ei täysin selviä, mitkä olivat klusteroinnissa käytetyt etäisyysmitat ja miten säännöt kyselyjen laajentamiselle oli määritelty. Täten on vaikea arvioida tuloksia muuten kuin toteamalla, että menetelmä on ilmeisesti vähintään vertailukelpoinen muiden menetelmien kanssa, mutta sen mahdollisesta paremmuudesta ei voida sanoa tarkempaa.

5 Vielä etäisyysmitoista

Edellä esitettyihin menetelmiin perehtyessä on selvinnyt, että etäisyysmittojen määritteleminen 1. kertaluvun objektien välille ei ole yksinkertaista, mutta että siitä riippuu suuresti koko klusteroinnin onnistuminen. Mainittakoon, että myös yrityksiä yleisten tällaisten mittojen generoimiseksi on tehty. Valitettavasti nämä mitat (ja erityisesti todistukset sille, että ne todella *ovat* etäisyysmittoja ja vieläpä järkeviä sellaisia) ovat sen verran monimutkaisia, että niitä ei ole mahdollista syvällisesti käsitellä tässä raportissa.

Esimerkiksi Ramon *et al* ovat kehittäneet mittaa, jossa kolmiportaisesti määritellään aito etäisyysmitta 1. kertaluvun objekteille [RBL98, RB98]. Hyvin lyhyesti, niissäkin ensin määritellään etäisyysmitta atomeille, perustuen siihen, kuinka paljon ne sisältävät samoja termejä. Sen jälkeen tämän etäisyyden perusteella määritellään etäisyys atomijoukoille (konjunktioille) perustuen etäisyyden minimoivaan paritukseen näiden välillä, ja vielä yleistetään tätä mittaa niin, että erilaiset muuttujien nimennät eri lauseissa eivät aiheuta tarpeetonta etäisyyden lisäystä. [RB98]

Lähestymistapa on sama kuin Bissonilla, mutta Ramonin *et al* mitta on aito etäisyysmitta. Monimutkaisuudestaan huolimatta se on myös laskettavissa polynomiaalisessa ajassa konjunktioiden (kokojen summan suhteen). Alustavat kokeilut tämän etäisyysmitan käyttämisestä ylläkuvatun päätöspuuoppimisen kanssa ovat antaneet lupaavia tuloksia. [RB98]

Jos meillä olisi hyvä etäisyysmitta ja hyvä prototyyppifunktio 1. kertaluvun objekteille, olisi mahdollista soveltaa monia tavallisia klusterointialgoritmeja näille objekteille sellaisenaan. Mathias Kirsten ja Stefan Wrobel [KW00] ovatkin todella ottaneet tällaisen mitan ja muuntaneet *k-means*-algoritmin *k-prototypes*-algoritmiksi. He saavuttivat hyviä tuloksia, kun valitsivat prototyypin ko. klusterin pisteiden joukosta (ns. *k-medoids*), mutta eivät onnistuneet kehittämään sopivaa yleistä prototyyppifunktiota. [KW00]

6 Yhteenveto

Tässä raportissa esiteltiin mahdollisuuksia hyödyntää ensimmäisen kertaluvun logiikkaa klusteroitavien objektien kuvailussa ja klusteroinnissa. Esiteltiin tarkemmin kaksi menetelmää, Bissonin kokoavan menetelmän ja De Raedtin *et al.* osittavan menetelmän.

Havaittiin, että vaikka ensimmäisen kertaluvun logiikasta on hyötyä objektien ja klusterien kuvailussa ja klusterien muodostamisessa, on suurena ongelmana näissä menetelmissä mielekkään etäisyysmitan ja ”klusterin keskipisteen” generointi ensimmäisen kertaluvun objekteille.

Lopussa todettiin, että lupaavia yrityksiä etäisyysmittojen ja prototyyppifuntkioiden määrittelemiseksi on kuitenkin tehty; on syytä odottaa alalla edistystä.

Viitteet

[Bis92] G. Bisson. Conceptual clustering in a first-order logic representation. In Neumann B, editor, *ECAI92*, pages 458–462. JWS, 1992.

[BRJD99] H. Blockeel, L. De Raedt, N. Jacobs, and B. Demoen. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, 3(1):59–93, 1999.

[KW00] M. Kirsten and S. Wrobel. Extending k-means clustering to first-order representations. In Frisch A Cussens J, editor, *ILP00*, volume 1866 of *LNAI*, pages 112–129. SV, 2000.

[RB97] L. De Raedt and H. Blockeel. Using logical decision trees for clustering. In Džeroski S and Lavrač N, editors, *ILP97*, volume 1297 of *LNAI*, pages 133–140. SV, 1997.

[RB98] J. Ramon and M. Bruynooghe. A framework for defining distances between first-order logic objects. In D. Page, editor, *ILP98*, volume 1446 of *LNAI*, pages 271–280. SV, 1998.

[RBL98] J. Ramon, M. Bruynooghe, and W. Van Laer. Distance measures between atoms. In *CompulogNet Area Meeting on Computational Logic and Machine Learing*, pages 35–41. University of Manchester, UK, May 1998.

[Vää87] J. Väänänen. *Matemaattinen logiikka*. Gaudeamus, Helsinki, 1987.