Severi Nättilä

Klusterointialgoritmien vertailu

Tietotekniikan kandidaatintutkielma

9. kesäkuuta 2021

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Severi Nättilä

Yhteystiedot: seanjana@student.jyu.fi

Ohjaaja: Leevi Annala

Työn nimi: Klusterointialgoritmien vertailu

Title in English: Comparison of clustering algorithms

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 23+0

Tiivistelmä: Tutkielmassa tutustutaan ryhmittelyn perusteisiin, todennäköisyysmallipohjaisen sekä ei-parametrisen datan klusterointiin ja menetelmiin. Klusterointimenetelmistä käydään läpi: EM-algoritmi, k-means, k-medoids, k-modes ja k-prototypes. Tutkitaan millaisen datan käsittelyyn menetelmät soveltuvat ja miksi juuri niitä hyödynnetään ryhmittelyssä.

Avainsanat: Algoritmi, Data, Klusterointi, GMM, EM-algoritmi, k-means, k-medoids, k-modes, k-prototypes, Ohjaamaton oppiminen Numeerinen data, Kategorinen data

Abstract: The point of this study is to focus on the basics of clustering, probability model-based approaches and non-parametric approaches. The clustering methods that this study focuses on are: EM-algorithm, k-means, k-medoids, k-modes and k-prototypes. This study also focuses on how these methods are applied in clustering of data and why they are used.

Keywords: Algorithm, Data, Clustering, GMM, k-means, k-medoids, k-modes, k-prototypes, Unsupervised learning, Numeric data, Categorical data

Jyväskylässä 9. kesäkuuta 2021

Nättilä Severi

Sisällys

1 Johdanto

Data-analyysin tarkoituksena on analysoida oikean maailman dataa, jota kertyy eri aloilta. Valtavia määriä dataa kertyy esimerkiksi lääketieteestä, biologiasta, sosiaalitieteistä ja taloudesta. Kaiken tämän datan tutkiminen on mahdotonta manuaalisesti. Koneoppiminen on keino automatisoida data-analyysiä. Yksi koneoppimisen menetelmistä on klusterianalyysi.

Klusterianalyysi toimii hyödyllisenä välineenä data-analyysissä. Klusterianalyysin tarkoituksena on muodostaa datasta ryhmiä eli klustereita etsimällä datasta samankaltaisuuksia sekä eroavaisuuksia. Datan samankaltaisuutta mitataan erilaisilla etäisyysmitoilla. Etäisyysmittoina voidaan käyttää esimerkiksi euklidista etäisyyttä.

Hierarkkinen ryhmittely oli varhaisin klusterointimenetelmä, jota biologit ja sosiaalitieteilijät käyttivät. Klusterianalyysistä taas tuli osa tilastollista usean muuttujan analyysiä (Yang, Lai ja Lin 2011, 1).

Vaikka klusterointi on tehokas tapa löytää suurista datamassoista uutta tietoa tai yhtenäisyyksiä, on klusterointi NP-kova ongelma (Mailinen 2015, 1). Tästä syystä tehokkaampien algoritmien kehittäminen erilaisille sovellusaloille on tärkeää. Klusterointimenetelmät voivat kuitenkin olla hankalia toteuttaa ja analysoida. Klusteroinnissa pienillä valinnoilla voi olla suuri merkitys. Klustereiden määrän k valinnalla on suoraan vaikutusta tuloksiin. Tuloksien oikeellisuutta voi olla vaikea tarkastella, sillä oikeaa klustereiden määrää harvoin tiedetään.

Tässä tutkielmassa keskitytään ohjaamattoman oppimisen menetelmään eli klusterointiin, käsitellään yleisessä käytössä olevia klusterointialgoritmeja ja tutkitaan niiden soveltuvuuksia datan analysointiin. Eli käsitellään millaisen datan kanssa mikäkin menetelmä toimii ja kuinka se toimii. Lisäksi käsitellään todennäköisyysmallipojaiselle lähestymistavalle Gaussin sekoitemallille käytettävää EM-algoritmia.

2 Ohjattu- ja ohjaamaton oppiminen

Koneoppiminen on tietojenkäsittelytieteissä käytetty joukko menetelmiä, jota hyödynnetään datan analysoinnissa. Koneoppimisongelmat jakautuvat yleisesti, joko ohjattuun oppimiseen tai ohjaamattomaan oppimiseen. Sekä ohjatussa, että ohjaamattomassa oppimisessa on erilaisia koneoppimisen menetelmiä, joita hyödyntämällä data-analyysiä pystytään automatisoimaan.

2.1 Ohjattu oppiminen

Ohjatussa oppimisessa käyttäjällä on dataa, joka sisältää pisteet x_i ja y_i, missä x_i on mahdollinen syöttö ja y_i on tästä saatava tulos. Ohjatun oppimisen tarkoituksena on pyrkiä opettamaan konetta kartoittamaan yhteyksiä datan lähtöjoukon x_i pisteiden sekä maalijoukon y_i pisteiden kanssa.

$$f(x_i) = y_i$$

kaikille i.

2.2 Ohjaamaton oppiminen

Ohjatussa oppimisessa tiedetään y_i, jonka avulla pystytään opettamaan konetta. Tällaisen datan kerääminen on kuitenkin aikaa sekä resursseja vievää. Jos datan maalijoukkoa y_i ei tunneta, pystytään hyödyntämään erilaisia ohjaamattoman oppimisen menetelmiä y_i :n selvittämiseksi sekä datan analysoimiseksi.

Ohjaamaton oppiminen on kuitenkin yleensä paljon haastavampaa kuin ohjattu oppiminen. Ohjaamattomassa oppimisessa ei myöskään ole mahdollista tarkistaa saatujen tuloksien oikeellisuutta, koska oikeaa vastausta y_i ei tiedetä.

Ohjaamattomien oppimisen menetelmien tarve kuitenkin kasvaa. Nykyään dataa kerätään paljon ja kaikelle datalle ei ole olemassa suoria vastauksia. Tällöin joudutaan turvautumaan

ohjaamattoman oppimisen työkaluihin, joista yksi on klusterointi. Klusteroinnissa dataa eritellään ryhmiin etsimällä datasta eroja sekä samanlaisuuksia.

3 Klusterointi

Klusterianalyysi on ohjaamatonta oppimista. Ohjaamattomassa oppimisessa käyttäjä antaa syötteitä esimerkiksi haluttujen klustereiden lukumäärän, mutta ei tulosteita. Vaikka tulosteita ei annettu pystytään datamassasta oppimaan klusteroinnin avulla löytämällä erilaisia homogeenisiä osaryhmiä joukoista. Samankaltaisuutta mitataan yleensä käyttäen erilaisia etäisyysmittoja, kuten eukilidsta etäisyyttä.

Ennen klusterointia on otettava huomioon datan muuttujien tyyppi sekä onko data todennäköisyysmallipohjaista vai ei-parametrista. Yleensä todennäköisyysmallipohjaiselle datalla pyritään käyttämään odotusarvon maksimointia EM-algoritmia ryhmittelemään data. Jos data on ei-parametrista tällöin joudutaan käyttämään muita menetelmiä. Tällaisia menetelmiä ovat muun muassa k-means, k-medoids, k-modes sekä k-prototypes.

Erilaiset muuttujat voidaan jakaa mitta-asteikon mukaisesti neljään eri luokkaan: Nominaaliasteikko, järjestysasteikko, välimatka-asteikko ja suhdeasteikko. Numeerisella datalla tarkoitetaan dataa, joka sisältää muuttujina välimatka- ja suhdeasteikon arvoja. Kategorisella datalla tarkoitetaan dataa, joka sisältää nominaali- ja järjestysasteikon arvoja. Eri ei-parametriset klusterointimenetelmät on kehitetty ryhmittelemään tietyntyyppistä dataa.

Ei-parametrisen datan klusterointia aloittaessa on tehtävä kaksi valintaa: kuinka mitata datapisteiden etäisyydet ja kuinka mitata klusteroinnin virhearvo. (Mailinen 2015, 2) Tyypillinen kriteeri sille, onko klusteri hyvä, on käyttää keskimääräistä neliövirhettä (mean squared error, MSE) sekä yhtenäistä neliövirhettä (total squared error, TSE) (Mailinen 2015, 1). Yksi klusteroinnissa käytetty etäisyysmitta on muun muassa Manhattanin etäisyys. Muita erilaisia etäisyyksien arviointiin käytettyjä mittoja on esimerkiksi Minkowskin etäisyys sekä euklidinen etäisyys, joita pystytään hyödyntämään esimerkiksi k-means klusteroinnissa. (Mailinen 2015, 2) K-means on tunnetuimpia klusterointialgoritmeja k-medoids algoritmin kanssa (Arora, Deepali ja Varshney 2016, 1).

Klusteroinnissa on aluksi otettava huomioon analysoitava data. Jos data on ainoastaan numeerista pystytään sen ryhmittelyyn hyödyntämään esimerkiksi edellä mainittua k-means algoritmia. Tällöin käyttäjän pitää tietää entuudestaan haluttujen ryhmien eli klustereiden

lukumäärä *k*. Tämä hankaloittaa datan analysointia sillä oikeaa klustereiden lukumäärää voi olla vaikea tietää.

Jos käytettävä on data on siistittyä ja muuttujien arvot vain numeerisia pystytään tällöin hyödyntämään k-means menetelmää. Näin ei kuitenkaan yleensä ole ja tällöin joudutaan turvautumaan eri klusterointi algoritmeihin, jotka eivät välttämättä ole yhtä yksinkertaisia, kuin k-means.

Jos halutaan klusteroida numeerista dataa, mutta klustereiden lukumäärän *k* arvon valitseminen tai laskeminen on haastavaa voidaan hyödyntää k-medoids menetelmää, jossa keskiarvon käyttämistä etäisyysmittana sijaan käytetään mittana mediaania, jonka selvittäminen voi olla helpompaa.

Myöskin dataa, joka ei ole numeerista voidaan klusteroida. Kategorisen datan klusterointiin voidaan hyödyntää esimerkiksi k-modes menetelmää, joka toimii samalla tavalla, kuin k-means, mutta muuttujina ovat kategoriset muuttujat.

Klusterointi on haastavaa tilanteissa, jossa data sisältää muuttujina sekä numeerisia, että kategorisia arvoja. Tähän käytetään algoritmina k-prototypes, joka on yhdistelmä k-means sekä k-modes menetelmiä.

4 Todennäköisyysmallipohjainen ja ei-parametrinen data

Dataa klusteroidessa on otettava huomioon datan muuttujien tyyppi. Eri klusterointialgoritmit toimivat erilaisille datatyypeille. Yleisesti klusteroinnissa käsiteltäviä datatyyppejä ovat joko kategorinen- tai numeerinen data. Kuitenkin käsiteltäessä isoja määriä dataa on muuttujien arvot harvoin ainoastaan kategorisia tai numeerisia. Yleensä näitä molempia ilmaantuu datassa. Tällöin voidaan dataa siistiä niin, että saadaan klusteroitua vain tietyn muuttujatyypin dataa tai käyttää klusterointimenetelmiä, jotka mahdollistavat sekä numeerisen, että kategorisen datan ryhmittelyn. Tilastotieteen näkökulmasta klusteronti jakautuu todennäköisyysmallipohjaisiin lähestymistapoihin ja ei-parametrisiin lähestymistapoihin (Yang, Lai ja Lin 2011, 1).

Todennäköisyysmallipohjaisessa lähestymistavassa pyritään käyttämään iteratiivista odotusarvon maksimointialgoritmia, EM-algoritmia. Ei-parametrista lähestymistapaa varten klusterointi menetelmät pohjautuvat erilaisten läheisyysmittareiden käyttämiseen. (Yang, Lai ja Lin 2011, 1) Yksi eniten käytetyistä ei-parametrisista klusterointi menetelmistä on k-means menetelmä, joka perustuu keskiarvon laskemiseen läheisyysmittana.

Kun käsitellään todennäköisyysmallipohjaista dataa pystytään hyödyntämään Gaussin sekoitemallia (GMM). Gaussin sekoitemallia käytetään jatkuvien mittausten parametrisena mallina.

4.1 Gaussin sekoitemalli

Gaussin sekoitemalli (GMM) on parametrinen todennäköisyystiheysfunktio, joka esitetään Gaussin komponenttitiheyksien painotettuna summana (Reynolds 2015, 1) Seuraava matemaattinen yhtälö antaa painotetun summan M-komponentin Gaussin tiheyksistä.

$$p(X|\lambda) = \sum_{i=1}^{M} w_i g(X|\mu_i, \sum_i)$$

Missä X on D ulotteinen jatkuva-arvoinen aineistovektori, $w_i = 1, 2, ..., M$ ovat sekoitepaino-

ja ja $g(X|\mu_i, \sum_i)$, $i = 1, 2, ..., M$ ovat komponentin Gaussin tiheydet (Reynolds 2015, 1)

Jokaisen komponentin tiheys on monimuuttuja Gaussin yhtälössä, joka on muotoa:

$$g(X|\mu_i, \sum_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sum_i|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu_i)' \sum_i^{-1}(x-\mu_i))$$

Jossa μ_i on keskiarvovektori ja $\sum_i$ on kovarianssimatriisi. Seospainot täyttävät seuraavan vaatimuksen PM$i = 1$ $w_i = 1$. Täydellisessä Gaussin sekoitemallissa on parametreina keskiarvovektorit, kovarianssimatriisit sekä sekoitepainot kaikkien komponenttien tiheyksistä (Reynolds 2015, 2)

Nämä parametrit esitetään yhdessä merkinnällä $\lambda = w_i, \mu_i, \sum_i i = 1, ..., M$. Tätä sekoitemallia käytetään jatkuvien mittausten todennäköisyysjakauman parametrisena mallina (Reynolds 2015, 2). Yksi tapa arvioida näitä parametreja on käyttää iteratiivista odotusarvon maksimointi algoritmia EM-algoritmia.

Todennäköisyysmallipohjainen lähestymistapa olettaa, että tietojoukko noudattaa todennäköisyysmallia siten, että sekoitemalleille soveliasta odotusarvon maksimointialgoritmia (EM-algoritmi) voidaan käyttää. (Yang, Lai ja Lin 2011, 1)

4.2 Odotusarvo ja maksimointialgoritmi

Sekoitemallien kanssa toimiessa käytetään usein odotusarvon maksimointialgoritmia (EM-algoritmi) (Yang, Lai ja Lin 2011, 1). Em-algoritmin kanssa pitää olla tarkkana, kun komponenttien määrä on annettava etukäteen, sillä Em-algoritmi on herkkä alkuarvoille (Yang, Lai ja Lin 2011, 1). EM-algoritmia voidaan käyttää, kun käsitellään tunnettua jakaumafunktiota.

Olkoon esimerkiksi aineisto $(x_1, x_2, ..., x_n)$ n kokoinen aineisto sekoitemallista, jossa vektorit x_j ovat d ulotteisia. Tällöin saadaan funktio:

$$f(X; \alpha, \theta) = \sum_{k=1}^{c} \alpha_k f(X; \theta_k)$$

missä $\alpha_k > 0$ on osuuksien sekoitukset rajoitukselle $\sum_{k=1}^{c} \alpha_k = 1$ ja $f(X; \theta)$ on x:n tiheys

luokasta k parametreilla θ_k (Yang, Lai ja Lin 2011, 2)

Olkoon $(z_1, z_2, ..., z_m)$ puuttuva data, missä z_c kuuluu $c = 1,2,3$. Jos $z_i = k$ se tarkoittaa, että i:nnes datapiste kuuluu siis K:nteen luokkaan. Tästä saadaan yhtenäinen data

$(x_1, x_2, ..., x_n, z_1, z_2, ..., z_m)$ tiheysfunktioksi:

$$f(x_1, ...x_n, z_1, ..., z_n; \alpha, \theta) = \prod_{i=1}^{n} \prod_{k=1}^{c} [\alpha_k f(x_i; \theta_k i)]^{Z_{ki}}$$

jossa:

$$z_{ki} = \begin{cases} 1 \text{ ,jos } Z_i = k \\ 0, \text{ ,jos } Z_i \neq k \end{cases}$$

Tämän jälkeen log-uskottavuusfunktio saadaan:

$$L(\alpha, \theta, x_1, x_2, ..., x_n, z_1, z_2, ..., z_n) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ki} ln(\alpha_k f(x_i, \theta_k))$$

EM algoritmin E askel käsitellään seuraavasti:

Koska muuttujia Z_{ki} ei tunneta, ehdollinen odotusarvo $E(Z_{ki}|x_i \alpha, \theta)$ korvaa ne. (Yang, Lai ja Lin 2011, 2)

Bayesin kaavalla saadaan:

$$Z_{ki} = E(Z_{ki}|x_i, \alpha, \theta) = \frac{\alpha_k f(x_i; \theta_k)}{\sum_{s=1}^{c} \alpha_s f(x_i; \theta_s)}$$

EM algoritmin M askel käsitellään seuraavasti:

Rajoituksella $\sum_{k=1}^{C} \alpha_k = 1$ maksimoidaan funktio

$$L(\alpha, \theta, x_1, x_2, ..., x_n, z_1, z_2, ..., z_n) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ki} ln(\alpha_k f(x_i, \theta_k))$$

Nyt voidaan päivittää yhtälö mittasuhteiden sekoittamiseksi käyttäen parametria

$$\alpha_k = \frac{\sum_{i=1}^{n} Z_{ki}}{m}.$$

4.3 K-Means

K-Means on yksi eniten käytetyistä ei-parametrisen datan klusterointi menetelmistä. K-means algoritmillä on vankka historia ja sitä sovelletaan useilla aloilla esimerkiksi signaalin käsittelyssä ja kuva-analyysissä.

K-means algoritmin avulla ryhmitellään n kappeletta m-ulotteisia datapisteitä/vektoreita $X_i = (x_i....,x_{im}), i = 1,...n, k$ kappaleeseen eri klustereita. K-means algoritmia käyttäessä aluksi määritellään k eli haluttu klustereiden lukumäärä. K-means menetelmä on niin sanottu prototyyppimenetelmä, jossa jokaisella klusterilla on oma edustajansa ja perusversiossa se on keskiarvo. Jotta klusterointi k-means menetelmällä olisi mahdollista, tulee datapisteiden olla numeerisia ja datapisteen X_i ja sentroidin C_j välinen etäisyys tulee voida laskea etäisyysfunktion $d(X_i, C_j)$ avulla. Yksi usein käytetty etäisyysfunktio k-means algoritmille on euklidinen etäisyysfunktio (Mailinen 2015, 2).

$$d(X_i, C_j) = \sqrt{\sum_{r=1}^{m} (x_{ir} - c_{jr})^2}$$

Missä X_i ja C_j ovat vektoreita

$$X_i = (x_1, x_2, ..., x_m) \text{ ja } C_j = (c_1, c_2, ..., c_m)$$

K-means klusteroinnissa aloitetaan alustamalla sentroidit C_j, joita on yhteensä k kappaletta. Yksi tapa alustaa sentroidit on valitsemalla satunnaiset datapisteet X_i alustaviksi sentroideiksi. (McKay 2004, 1)

$$P_j^{(t)} = X_i : ||X_i - C_j^{(t)}|| \leq ||X_i - C_j^{(t)}{}_*||$$

Tämän jälkeen lasketaan klustereiden keskiarvot:

$$C_j^{(t+1)} = \frac{1}{(P_j^{(t)})} \sum_{X_i \in P_j^{(t)}} X_i$$

Nämä askeleet toistetaan kunnes sentroidien paikat eivät enää muutu. (Mailinen 2015, 6)

K-means algoritmi on toteutukseltaan suhteellisen yksinkertainen ja sen käyttä on helposti toteutettavissa, käytetään sitä myös muiden monimutkaisempien algoritmien osana ositusvaiheessa.

K-means algoritmin tuloksena saatava klusterointi on paikallinen optimiratkaisu, mutta se ei välttämättä ole globaali optimi. (Mailinen 2015, 6) Tästä syystä k-means -algoritmin sijasta voidaan käyttää monimutkaisempia algoritmeja tai Gaussin sekoitemallia, jotka tarjoavat parempia tuloksia normaalin k-means algoritmiin verrattuna.

K-means algoritmi vaatii syötteenään ennakkotietona haluttujen klustereiden lukumäärän k. Eli, käyttäessä k-means klusterointia on tiedettävä valmiiksi oikea määrä klustereille ja jos käyttäjällä ei ole tietoa tästä voi olla parempi käyttää jotain muuta algoritmia. K:n arvo pystytään kuitenkin tarvittaessa selvittämään aloittamalla algoritmi tarpeeksi pienellä k:lla ja kasvattaa sitä tarvittaessa. K-means menetelmää kannattaakin käyttää silloin, kun data on siistiä ja kaikki klusteroitavat muuttujat ovat numeerisia. Näin ei kuitenkaan aina ole ja tällöin keskiarvojen laskemisen sijaan pystytään laskemaan esimerkiksi mediaaneja, jolloin voidaan käyttää K-medoids algoritmia.

Klusteroinnissa aikavaativuudet vaihtelevat $O(n)$ aina $O(n^5)$. K-means algoritmin aikavaativuus on

$$T(n) = O(I * k * n)$$

missä k on klustereiden lukumäärä ja I on tarvittavien iteraatioiden lukumäärä. (Mailinen 2015, 6).

4.4 K-Medoids

Kuten k-means, on myös K-medoids klusterointialgoritmi, mutta keskiarvojen laskemisen sijaan käytetään etäisyyksien laskemiseen mediaanin arvoa, jonka hyödyntäminen voi olla helpompaa kuin keskiarvojen. K-medoids menetelmässä hyödynnetään medoideja klusterin keskipisteenä. Medoidit valitaan satunnaisesti X_i datapisteistä, joista muodostetaan K_i klusteria. (Arora, Deepali ja Varshney 2016, 509)

Oletetaan, että n objektin, joilla kullakin on p muuttujaa, pitäisi olla ryhmitelty k $(k < n)$ klustereiksi, joissa k oletetaan olevan annettu. Määritetään objektin i j-muuttuja x_{ij} $(i = 1,...,n; j = 1,...,p)$ (Park ja Jun 2009, 2)

Euklidinen etäisyys objektin i ja objektin j antaa

$$d_{ij} = \sqrt{\sum_{r=1}^{m} (x_{ir} - c_{jr})^2}$$

Tämän jälkeen algoritmin toiminta perustuu kolmeen vaiheeseen, jotka käydään läpi.

1. Laske jokaisen parin välinen etäisyys kaikkien kohteiden valitun erilaisuuden perusteella.

2. Laske v_j objektille j seuraavasti:

$$V_j = \sum_{i=1}^{n} \frac{d_{ij}}{\sum_{l=1}^{n} d_i}$$

kun $j = 1,2,...,n$

3. Seuraavaksi lajittele v_j nousevassa järjestyksessä. Valitse k objektia joiden ensimmäiset k pienimmät arvot ovat alkuarvoja medoidit. Hanki alkuperäinen klusteritulos osoittamalla jokainen esine lähimpään medoidiin. Laske kaikkien kohteiden etäisyyksien summa heidän medoidilleen. (Park ja Jun 2009, 2)

4.5 K-Modes

K-means ja k-medoids menetelmät mahdollistaa klusteroinnin käyttäen ainoastaan numeerisia muuttujia. Tämä rajoittaa näiden menetelmien klusteroinnin hyödyntämistä kategoriseen

dataan. Jos kategorista dataa halutaan klusteroidan tällöin voidaan hyödyntää K-modes menetelmää, joka on toiminnaltaan samantapainen, kuin k-means, mutta mahdollistaa numeerisen datan sijaan klusteroinnin kategorisella datalla.

K-means algoritmi pystytään muokkaamaan pienillä säädöillä ryhmittelemään numeerisen datan sijasta kategorista dataa. Tämä saadaan toteutettua siten, että:

Olkoon X, Y kaksi kategorista objektia, joita kuvaavat m kategoriset attribuutit.

Etäisyysmittana K-modes menetelmälle voidaan käyttää:

$$d_1(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

missä

$$\delta(x_j, y_j) = \begin{cases} 0 \text{ ,jos } (x_j = y_j) \\ 1 \text{ ,jos } (x_j \neq y_j) \end{cases}$$

Olkoon X joukko kategorisia muuttujia $A_1, A_2, ..., A_m$

1. Joukon Moodi:

$X = (x_1, x_2, ..., x_n)$ -moodi on vektori $Q = (q_1, q_2, ..., q_m)$, joka minimoi

$$D(X,Q) = \sum_{i=1}^{n} d_1(X,Q)$$

Tässä Q ei välttämättä kuulu X. (Huang 1998, 289)

2. Joukon moodin etsintä:

Olkoon $n_{c_{kj}}$ niiden objektien lukumäärä joilla on k:nnes luokka $c_{k,j}$ atribuutissa A_j ja $f_r(A_j = c_{kj}|X) = \frac{n_{c_{kj}}}{n}$ luokan j suhteellinen frekvenssi $c_{k,j}$ X:ssä (Huang 1998, 289)

Funktio $D(X,Q)$ minimoidaan jos $f_r(A_j = q_j|X) >= f_r(A_j = c_{k,j}|X)$, kun $q_j \neq c_{k,j}$ kaikilla $j = 1, ..., m$) (Huang 1998, 289)

K-modes algoritmin aikavaativuus on

$$T(n) = O(T * k * n)$$

missä T on tarvittavien iteraatioidan määrä ja n on datan koko.

4.6 K-Prototypes

K-prototypes algoritmi mahdollistaa sekatyyppisten datapisteiden klusteroinnin. Toisin, kuin k-means tai k-modes menetelmät, mahdollistaa k-prototypes klusteroinnin sekä numeerisilla, että kategorisilla muuttujilla. K-prototypes on yksi tunnetuimmista menetelmistä sekatyyppisen datan ryhmittelyyn. Kuten k-means algoritmi, pyrkii k-prototypes klusteroimaan datapisteet X_i k -klusteriin, kun $(k < n)$.

Olkoon datajoukko $X = (x_1, x_2, ..., x_n)$, joka sisältää n datapistettä. K-prototypes algoritmin ideana on löytää k klusteria minimoimalla objektifunktio J:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(X_i, C_j),$$

missä $u_{ij} = 1$, jos $X_i \in U_j$, muuten $u_{ij} = 0$ (Huang 1998, 291)

K-prototypes eroaa k-means algoritmin toiminnasta siten, että numeeristen ja kategoristen muuttujien vaikutus datapisteen X_i ja sentroidin C_j väiseen etäisyyteen lasketaan erikseen. Tällöin käytetään etäisyysfunktiona

$$d(X_i, C_j) = d_{rea}(X_i^{(r)}, C_j^{(r)}) + \sum_{r=1}^{m_c} \delta(x_{ir}, c_{jr}),$$

kun

$$\sum_{r=1}^{m_c} \delta(x_{ir}, c_{jr}) = d_{cat}(X_i^{(c)}, C_j^{(c)})$$

jossa

$$\delta(x_{ir}, y_{jr}) = \begin{cases} 0 \text{ ,jos } (x_{ir} = y_{jr}) \\ 1 \text{ ,jos } (x_{ir} \neq y_{jr}) \end{cases}$$

missä $X_i^{(r)} = x_{i1}...,x_{im_r}$ ja $X_i^{(c)} = x_{i(mr+1)}...,x_{i(m_r+m_c)}$

reaaliarvojen vaikutuksen laskee etäisyysfunktio $d_{rea}(X_i^{(r)},C_j^{(r)})$, joka voi olla euklidinen etäisyys:

$$\sqrt{\sum_{r=1}^{m_r}(x_{ir}-c_{jr})^2}$$

(Huang 1998, 291).

5 Algoritmien vertailu

Klusterointia aloittaessa on otettava huomioon minkälaista dataa ollaan käsittelemässä. Jos käsiteltävä data on todennäköisyysmallipohjaista voidaan hyödyntaa Gaussin sekoitemenetelmiä sekä EM-algoritmia ryhmittelemään data. Kun käsiteltävä data on ei-parametrista joudutaan turvautumaan erilaisiin menetelmiin. Näitä menetelmiä on esimerkiksi k-means, k-medoids, k-modes sekä k-prototypes. Tällöin valitulla algoritmilla ja menetelmällä on suuresti väliä. Erilaisia algoritmeja on paljon ja ne ovat suunniteltu klusteroimaan tietyntyyppistä dataa.

Numeerista dataa ryhmiteltäessä pystytään käyttämään muun muassa k-means tai k-medoids menetelmiä. K-means algoritmia käyttäessä datan pitää olla numeerista ja mielellään siistittyä, jolloin k-means algoritmi ei käytä ryhmittelyyn maksimiaikaa. Lisäksi dataa klusteroidessa olisi ennestään tiedettävä tarvittavien klustereiden määrä k. K:n selvittäminen voi olla joskus haastavaa vaikkakin erilaisia tapoja tämän selvittämiseen on olemassa. Klusterointia käyttäessa on joskus tavoitteena saada tietyllä menetelmällä selvitettyä k :n arvo ja, jos se vaaditaan ennestään on klusterointi tällöin hyödytöntä tällaisella menetelmällä. Tällaisissa tapauksissa pystytään hyödyntämään esimerkiksi k-medoids algoritmia, jolloin saataisiin helpompia tapoja tarvittavien klustereiden lukumäärän selvittämiseen. Toisena huonona puolena k-means menetelmälle on sen rappeutuminen suuria datamassoja käsiteltäessä. Rappeutumisella tarkoitetaan, että klusterointi voi johtaa tyhjiin klustereihin (Natarajan Meghanathan 2012, 123). Tämä vie turhaa resursseja sillä tyhjillä klustereilla ei ole merkitystä luokittelussa. Tästä syystä k-medoids algoritmi on hyödyllisempi suuremmille dataseteille ja k-means pienemmille (Natarajan Meghanathan 2012, 498).

Jos x_i datapisteiden määrä on vähäinen, tällöin voi olla hyödyllisempää käyttää k-means algoritmia sen yksinkertaisuuden takia. Jos luokiteltavien datapisteiden x_i määrä on suuri käyttää k-means algoritmi suoritusajastaan maksimiajan ja on hyödyllisempää käyttää k-medoids algoritmia. Suurissa datajoukoissa k-medoids menetelmä on yleensä tehokkaampi, kuin k-means. K-medoids menetelmän huonona puolena on kuitenkin sen monimutkaisuus verrattuna k-means algoritmiin. (Arora, Deepali ja Varshney 2016, 6)

Aina käsiteltävä data ei kuitenkaan ole numeerista. Tällöin k-means ja k-medoids menetelmät eivät toimi. Jos käsitelvävä data on kategorista ja sitä halutaan klusteroida, voidaan käyttää menetelmänä k-modes algoritmia. Kategorista dataa löytyy käytännössä kaikkialta, joten sen käsittelyltä ei voida välttyä klusteroinnissa. Yleensä on totuttu ryhmittelemään numeerista dataa ja kategorisen datan ryhmittely on jäänyt vähemmälle huomiolle. (Zhang, Wang ja Song 2006, 355)

K-modes algoritmi on tarkoitettu klusteroimaan juurikin kategorista dataa. Pohjimmiltaan k-modes toiminta on samantapainen, kuin k-means algoritmit, mutta pienillä säädöillä se saadaan ryhmittelemään kategorista dataa. Kun k-means käyttää etäisyysmittana eukilidsta etäisyyttä, muuttamalla tätä pystytään saada algoritmi ryhmittelemään kategorista dataa. Keskiarvon (mean) sijaan käytetään moodia (mode) ja etäisyysfunktio muokataan käyttäen vertailuna epäyhteensopivuutta. Moodi saadaan laskemalla eri esiintymien lukumäärä ja valitsemalla useimmiten esiintyvä arvo. Moodi ei arvona kuitenkaan ole aina yksikäsitteinen. Tämä voi johtaa vääränlaiseen ryhmittelyyn. K-modes menetelmän jälkeen onkin kehitetty uusia menetelmiä kategorisen datan ryhmittelyyn muokkaamalla epäyhteensopivuusfunktiota. Muokkaamalla tätä funktiota saadaan

$$d_2(X_i, Q_l) = \sum_{j=1}^{m} \phi(x_{i,j}, q_{l,j})$$

missä

$$\phi(x_{i,j}, q_{l,j}) = \begin{cases} 1 - f_r(A_j = q_{l,j}|X_1) \text{ ,jos } (x_{i,j} = q_{l,j}) \\ 1 \text{ ,jos } (x_{i,j} \neq q_{l,j}) \end{cases}$$

Tämä pitää kuitenkin algoritmin aikavaativuuden samana $O(T * k * n)$. (He, Deng ja Xu 2005, 5) K-modes algoritmiin viitataan silti paljon ja sitä voidaan pitää kategorisen datan ryhmittelyn alkuperänä.

Suuria datamassoja käyttäessä data kuitenkaan harvoin on ainoastaan numeerista tai kategorista. Yleensä halutaan käsitellä satunnaisesti kerättyä sekalaista dataa. Tällöin numeeriseen tai kategoriseen dataan käytettävät menetelmät eivät toimi. Datasta pystytään siivoamaan

vain tietyt muuttujat, joita sitten klusteroidaan, joko numeerisen tai kategorisen datan menetelmillä. Tämä vie kuitenkin paljon aikaa eikä ole aina vaihtoehto. Tällöin on mahdollista käyttää hyväksi K-prototypes menetelmää. K-Prototypes menetelmä on yhdistelmä k-means sekä k-modes mentelmiä, joka mahdollistaa sekatyyppisen datan käsittelyn. Menetelmänä k-prototypesin käyttäminen voi olla haastavampaa kuin yksinkertaisen k-means menetelmän, mutta välttämätöntä sekamuuttujien tilanteessa. K-prototypes menetelmässä laskettiin datapisteiden X_i ja sentroidin C_i väliset etäisyydet erikseen. Tämä lisää algoritmin suoritusaikaa.

6 Yhteenveto

Tutkimuksessa huomataan, että erilaiset menetelmät on tarkoitettu juuri tietynlaisen datan ryhmittelyyn. Klusterointi on myös tieteellisesti vaikea käsite sillä se on NP-kova ongelma. (Mailinen 2015, 1)

Tutkimuksessa käsiteltiin todennäköisyysmallipohjaiselle datalle soveltuvaa EM-algoritmia ja ei-parametriselle datalla k-means, k-medoids, k-modes ja k-prototypes menetelmät. Numeerisen datan käsittelyssä k-means on tunnetuin ja yleisin käytössä oleva ryhmittelyalgoritmi. Silti on se paikoiten puutteellinen verrattuna k-medoids algoritmiin. K-means tarvitsee käyttäjän syöttämän klustereiden lukumäärän vaikka tämän selvittäminen voi olla haastavaa. Lisäksi ryhmiteltäessä suuria määriä dataa voi k-means algoritmi olla rappeutuva ja aiheuttaa tyhjiä klustereita. Tyhjät klusterit ovat merkityksettömiä luokittelussa ja tämä aiheuttaa ainoastaan turhaa resurssien kulutusta. K-medoids algoritmi on hyödyllisempi suurille datajoukoille, kuin k-means, koska keskiarvon sijaan se hyödyntää mediaania, joka on yleensä käyttäjän helpompi selvittää. Huonona puolena k-medoidille on, että se on paljon monimutkaisempi, kuin k-means.

Kategoriselle datalle yleisesti käytössä oleva k-modes on muokattu versio k-means algoritmista. K-modes menetelmästä on kehitetty jo uusia tehokkaampia menetelmiä, mutta on se silti yleisesti käytössä ja toimii kategorisen datan ryhmittelyn alkuperänä. K-modes algoritmiä on myös hyödynnetty sekatyyppisen datan ryhmittelyssä käytettävään k-prototypes algoritmiin. K-prototypes on yhdistelmä k-means ja k-modes menetelmiä.

Erilaisia ryhmittelymenetelmiä on paljon. Erilaiset menetelmät on kehitetty toimimaan tietynlaisen datan ja muuttujien kanssa. Tutkielmassa keskityttiin muutamaan tunnettuun ja yleisesti käytössä olevaan menetelmään. Uudemmat ja tehokkaammat menetelmät jäivät kuitenkin tutkimuksesta pois. Tulevaisuudessa aiheesta voisi tutkia lisää esimerkiksi vertailemalla useampia erilaisia algoritmeja. Lisäksi jotkut menetelmät saattavat jäädä käytöstä, kun parempia algoritmeja kehitellään. Tällöin olisi hyvä saada vertailutietoa uusien ja vanhempien algoritmien toiminnasta ja soveltuvuuksista.

Lähteet

Arora, Preeti, Deepali ja Shipra Varshney. 2016. “Analysis of K-Means and K-Medoids Algorithm For Big Data”. 1st International Conference on Information Security Privacy 2015, *Procedia Computer Science* 78:507–512. ISSN: 1877-0509. https://doi.org/https://doi.org/10.1016/j.procs.2016.02.095. https://www.sciencedirect.com/science/article/pii/S1877050916000971.

He, Zengyou, Shengchun Deng ja Xiaofei Xu. 2005. “Improving K-Modes Algorithm Considering Frequencies of Attribute Values in Mode”, 3801:157–162. Joulukuu. ISBN: 978-3-540-30818-8. https://doi.org/10.1007/11596448_23.

Huang, Zhexue. 1998. “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values”. *Data Mining and Knowledge Discovery* 2, numero 3 (syyskuu): 283–304. ISSN: 1573-756X. https://doi.org/10.1023/A:1009769707641. https://doi.org/10.1023/A:1009769707641.

Mailinen, Joonas. 2015. *New Alternatives for k-Means Clustering.* Kuopio: Forest / natural sciences 178. University of Eastern Finland.

McKay, David J.C. 2004. “Information Theory, Inference, and Learning Algorithms”. *IEEE Transactions on Information Theory* 50 (10): 2544–2545. https://doi.org/10.1109/TIT.2004.834752.

Natarajan Meghanathan, Nabendu Chaki, Dhinaharan Nagamalai. 2012. *Advances in Computing and Information Technology.* Springer-Verlag Berlin Heidelberg.

Park, Hae-Sang, ja Chi-Hyuck Jun. 2009. “A simple and fast algorithm for K-medoids clustering”. *Expert Systems with Applications* 36 (2, Part 2): 3336–3341. ISSN: 0957-4174. https://doi.org/https://doi.org/10.1016/j.eswa.2008.01.039. https://www.sciencedirect.com/science/article/pii/S095741740800081X.

Reynolds, Douglas. 2015. “Gaussian Mixture Models”. Teoksessa *Encyclopedia of Biometrics,* toimittanut Stan Z. Li ja Anil K. Jain, 827–832. Boston, MA: Springer US. ISBN: 978-1-4899-7488-4. https://doi.org/10.1007/978-1-4899-7488-4_196. https://doi.org/10.1007/978-1-4899-7488-4_196.

Yang, Miin-Shen, Chien-Yo Lai ja Chin-Ying Lin. 2011. *A robust EM clustering algorithm for Gaussian mixture models.* Department of Applied Mathematics, Chung Yuan Christian University.

Zhang, Peng, Xiaogang Wang ja Peter X.-K. Song. 2006. “Clustering Categorical Data Based on Distance Vectors”. *Journal of the American Statistical Association* 101 (473): 355–367. ISSN: 01621459. http://www.jstor.org/stable/30047463.