VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Kursinis darbas

Dokomentų klasterizacija

(Document clustering)

Atliko: 4 kurso 1 grupės studentas
Dominykas Ablingis (parašas)

Darbo vadovas:
lekt. Rimantas Kybartas (parašas)

Vilnius
2018

Turinys

Įvadas

Šiais laikais sparčiai vystantis informacinėms technologijoms ir gausėjant informacijos kiekiams, vis aktualesnė tampa kokybiška ir greita reikiamos informacijos paieška, jos organizavimas ir naujų įžvalgų išgavimas. Darbą su dideliais tekstinės informacijos kiekiais galėtų pagreitinti ir palengvinti įprastai skaitmeninių duomenų analizei naudojami klasterizavimo metodai.

Šio darbo tikslas yra išnagrinėti ir aprašyti klasterizavimo metodus, tekstinių duomenų parengimą darbui su klasterizavimo algoritmais bei gautų klasterių kokybės įvertinimo kriterijus.

Duomenų analizės eiga

Klasterizavimas tėra vienas iš žingsnių tekstinių duomenų analizės procese. Tekstinių duomenų analizės procesą galime suskirstyti į keletą žingsnių[FPS96]:

1. Probleminės srities nustatymas (angl. *problem domain*).
2. Duomenų surinkimas (angl. *data collection*).
3. Duomenų tvarkymas ir apdorojimas (angl. *cleaning and preprocessing*)[1].
4. Duomenų supaprastinimas ir transformavimas (angl. *reduction and projection*).
5. Duomenų analizavimas (angl. *data analysis*). Klasterizavimo atveju tai būtų tinkamo klasterizavimo algoritmo parinkimas ir taikymas. Kadangi egzistuoja didelė klasterizavimo algoritmų įvairovė, jie skiriasi ne tik veikimo principu, bet ir gautų rezultatų forma, todėl norint išsirinkti tinkamą algoritmą, reikia įvertinti tiek turimus duomenis, tiek norimus gauti rezultatus.
   - 5.1. Panašumo matas (angl. *proximity measure*) – tai kiekybinis matas, kuris įvertina kiek du požymių vektoriai yra „panašūs“ ar „nepanašūs“ tarpusavyje. Reikėtų pasitikrinti, ar visi požymiai yra lygiaverčiai ir tarp jų nėra dominuojančių.
   - 5.2. Klasterizavimo kriterijai / parametrai (angl. *clustering criterion*) – turėtų nurodyti, kokius klasterius tikimasi išskirti duomenų aibėje, taip pat duomenų jungimo į vieną klasterį ir atskyrimo į kelis klasterius kriterijus.
6. Rezultatų validavimas / įvertinimas (angl. *validation of results*) – klasterizavimo proceso rezultatų formalus teisingumo įvertinimas. Tam yra naudojami specifiniai metodai.
7. Rezultatų interpretavimas (angl. *interpretacion of results*) – nors gauti galutiniai klasteriai gali atitikti matematinius reikalavimus, bet gauti rezultatai gali prieštarauti sveikam protui.

[1] 3 ir 4-tas žingsniai gali būti apibrėžti kaip požymių pasirinkimas (angl. *feature selection*). Svarbu atrinkti požymius, kurie yra esminiai atskiriant objektus ir juose būtų konkreti informacija užduočiai atlikti, tuo pačiu stengiantis palikti kuo mažiau perteklinės informacijos ir neprarasti svarbios informacijos.

Visgi rezultatas tėra galimas duomenų suskirstymas į grupes, todėl dažniausiai gautus rezultatus reikia lyginti su iš anksto sudaryta hipoteze.

8. Rezultatų panaudojimas. Tai gali būti vienas iš didesnės analizės žingsnių arba konkretus rezultatas padedantis priimti sprendimą.

Šiame ir projektiniame darbe apžvelgsiu 2 – 6-tą žingsnius, daugiausia dėmesio skirdamas 5-tam žingsniui.

1. Tekstinių dokumentų parengimas klasterizavimui

Ši proceso dalis atsakinga už tai, kad tekstai iš žmogui patogios formos būtų paversti į formą patogią duomenų analizei, šiuo atveju, klasterizavimui. Šis procesas atliekamas tokiais žingsniais:

- **Informacijos išgavimas** – šiame žingsnyje įvairių formų (pvz., nuotraukos, audio įrašai) ir formatų (pvz., HTML, PDF, EPUB) dokumentus paverčiame į patogius analizei tekstus. Pašaliname formatavimą, paveiksliukus, išdėstymą ir kitą informaciją, kurią sunku paversti į vertingą tekstą.

- **Teksto filtravimas** – šiame žingsnyje panaikiname analizei nereikalingą tekstinę informaciją, supaprastiname ir suvienodiname žodžius[MPP].

  - **Leksikos analizė** – suskaidome tekstą į atskirus žodžius ir pašaliname simbolius, skyrybos ženklus ir skaičius.
  - **Nereikšmingų žodžių pašalinimas** – pašaliname žodžius, kurie nėra vertingi analizei. Tai gali būti mažai reikšmės turintys žodžiai, kurie labiau reikalingi „suklijuoti“ tekstą patogesniam skaitymui. Taip pat pašalinami praktiškai kiekviename dokumente sutinkami žodžiai, kurie nesvarbūs klasterizavimui.
  - **Sinonimiškumas ir daugiareikšmiškumas** – bandome rasti sinonimiškiems žodžiams vieną bendrą žodį ar daugiareikšmius žodžius paversti į konkretesnius.
  - **Morfologinis analizavimas** – skirtingas žodžio formas suvienodiname į vieną bendrinę formą. Tai atliekama išgaunant žodžio šaknį arba bandant paversti jį į bendrinę formą, dar žinomą kaip lema.

- **Požymių išskyrimas** – turimus filtruotus tekstinius duomenis reikia paversti į skaitinius, nes su tokiais dirba dauguma klasterizavimo algoritmų[ATL13]. Tai atliekame dokumentus paversdami į vektorius, kuriuose kiekvienas elementas atitinka visame dokumentų rinkinyje sutinkamus žodžius. Elementų reikšmės nurodo arba žodžio buvimą (binarinė reikšmė) tekste, arba kaip dažnai jie sutinkami (skaliarinė reikšmė).

2. Klasterizavimas

Klasteris – tai panašių objektų grupė[Tan+07]. Klasterinė analizė tai yra matematinių metodų visuma, kurios pagalba galima objektų arba reikšmių aibes pagal jų panašumus, suskirstyti į prasmingas grupes – klasterius. Tai atliekama be jokios papildomos informacijos apie tas grupes (jų dydį, kiekį, grupavimo požymius). Taigi, klasterių analizė yra iš anksto nežinomų struktūrų paieška. Iš to kyla pagrindinis klasterizavimo iššūkis ir privalumas – sugebėjimas atrasti sudėtingas struktūras be išankstinės informacijos apie jas. Kitaip tariant, pagrindinis klasterizavimo tikslas – maksimizuoti objektų, esančių toje pačioje grupėje, tarpusavio panašumą ir minimizuoti objektų, esančių skirtingose grupėse, panašumą.

Dokumentų klasterizavimas yra labai dažnai painiojamas su dokumentų klasifikavimu (dar vadinamas kategorizavimu). Klasifikuojant duomenis iš anksto apibrėžiamos kategorijos ir priklausomai nuo duomenų turinio, jie yra priskiriami kuriai nors iš tų kategorijų (dažnai vadinamų klasėmis), todėl duomenų klasifikacija yra priskiriama prižiūrimo mokymosi (angl. *supervised learning*) užduotims. Tuo tarpu klasterizuojant duomenis, jokios iš anksto apibrėžtų kategorijų aibės nėra, todėl klasterizavimas yra priskiriamas neprižiūrimo mokymosi (angl. *unservised learning*) užduotims.

2.1. Panašumo funkcijos / įverčiai

Nors turime klasterio apibrėžimą, bet vis dar neaišku kaip jį panaudoti tekstiniams duomenims, kaip apibrėžti dviejų dokumentų panašumą (ar skirtingumą). Ankstesniame skyriuje apibūdinome kaip tekstinį dokumentą paversti į skaitinius duomenis, sekantis žingsnis būtų dokumentų panašumo nustatymas, pasinaudojus panašumo ir metrikinėmis funkcijomis[2] Aptarsime dalį populiariausių klasterizavimo srityje naudojimų metrikų ir panašumo matavimo matų[Hua08].

Euklido atstumas tarp taško x ir taško y – tai trumpiausias atstumas tarp šių taškų. Plokštumoje arba trimatėje erdvėje – tai taškus jungianti tiesė. Bendru n-matės erdvės atveju šis atstumas apskaičiuojamas pagal formulę:

$$\mathrm{d}_{\text{Euklido}}(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Tai pati populiariausia metrika[JD88], bet ji turi problemų, kai matmenys turi skirtingą svarbą, taip pat kai du dokumentai yra su identišku turiniu, bet skirtingais ilgiais, bus laikomi skirtingais.

Manheteno atstumas (dar žinomas kaip miesto kvartalų atstumas) – absoliučių skirtumų tarp visų porų reikšmių suma ir skaičiuojamas pagal formulę:

[2]Metrika yra matematinis terminas, apibūdinantis atstumą matuojančias funkcijas. Todėl ne bet kurią funkcija galima vadinti metrika, ji turi atitikti reikalavimus[OCo04].

$$\mathrm{d}_{\text{Manhateno}}(q,p) = \sum_{i=1}^{n} |q_i - p_i|^2$$

Čebyševo atstumas – didžiausias absoliutus skirtumas tarp visų porų. Jis skaičiuojamas taip:

$$\mathrm{d}_{\text{Čebyševo}}(q,p) = \max_i |q_i - p_i|$$

(a) **(b)**

1 pav. a – Manhateno (mėlina linija) ir eukliko (žalia linija) atstumų palyginimas reliame pasaulyje.
Šaltinis: `https://arc.cs.rutgers.edu/courses/f17/lecture.08.extra.pdf`
b – Čebyševo atstumas atitinka karaliaus galimus ėjimus šakmatų žaidime.
Šaltinis: `http://www.ieee.ma/uaesb/pdf/distances-in-classification.pdf`

Minkovskio atstumas – apibendrina skirtingų atstumų metrikas. Parinkus skirtingas p reikšmes, galima gauti šiuos atstumus: p = 1 Manheteno, p = 2 Euklido, p $\to \infty$ Čebyševo:

$$\mathrm{d}_{\text{Minkowski}}(q,p) = \left(\sum_{i=1}^{n} |q_i - p_i|^2 \right)^{1/p}$$

Kosinuso koeficientas – kampas tarp dvejų vektorių. Mūsų atveju, vektoriai negali būti neigiami, tai šios funkcijos reikšmių sritis yra $[0, 1]$. 0 reiškia, kad abu dokumentai neturi bendrų terminų, 1 – abu dokumentai turi identiškus terminus (bet nebūtinai identiškus kiekius).

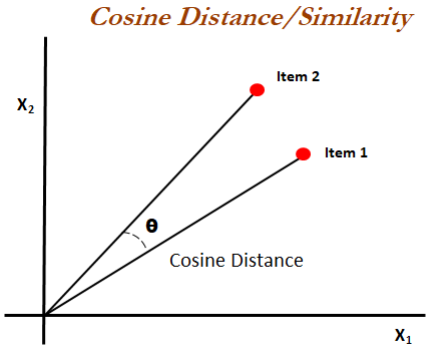
Jei dirbame su žodynais be svorių, galime naudoti aibėms skirtą versiją:

$$\mathrm{d}_{\text{Cosinuso}}(q,p) = \frac{|Q \cap P|}{\sqrt{|Q| \times |P|}}$$

Jei dirbame su svoriais:

$$s_{\text{Cosinuso}}(q,p) = \frac{\mathbf{Q}\cdot\mathbf{P}}{\|\mathbf{Q}\|\|\mathbf{P}\|} = \frac{\sum_{i=1}^{n} Q_i P_i}{\sqrt{\sum_{i=1}^{n} Q_i^2}\sqrt{\sum_{i=1}^{n} P_i^2}}$$

Kosinuso koeficientas yra vienas populiariausių panašumo matų[LA99]. Pagrindinis jo privalumas, kad jis identiškais laiko skirtingo dydžio dokumentus su vienodomis struktūromis, todėl išvengia didelio matmenų kiekio problemos.



2 pav. Kosinuso panašumas tarp dvejų taškų.
Šaltinis: `https://www.safaribooksonline.com/library/view/statistics-for-machine/9781788295758/eb9cd609-e44a-40a2-9c3a-f16fc4f5289a.xhtml`

2.2. Klasterizavimo algoritmai

Egzistuoja plati klasterizavimo algoritmų įvairovė (pvz.: 12 pav.), todėl atsiranda poreikis kaip nors juos suskirstyti. Vienas iš būdų – pagal sudarytų klasterių savybes:

- **Griežti** (angl. *exclusive, hard*) – kai klasteriai neturi bendrų narių. Kartais gali atsirasti atvejai, kai elementas „matematiškai“ gali priklausyti ne vienai grupei, bet toks atvejis yra retas. **Negriežti** (angl. *non-exclusive, soft, fuzzy*) – kai klasteriai tarpusavyje gali turėti bendrų narių. Tokiu atveju galime nagrinėti, kaip konkretus elementas priklauso skirtingoms grupėms ir kaip gerai jas atitinka.

- **Plokšti** (angl. *flat*) – kai elementai suskirstomi į grupes, kurios viena kitai yra lygios. **Hierarchiški** (angl. *hierarchical, taxonomic*) – kai grupė gali būti sudaryta iš kelių „konkretesnių“ grupių. Pvz., retryveris → šuo → žinduolis → gyvūnas.

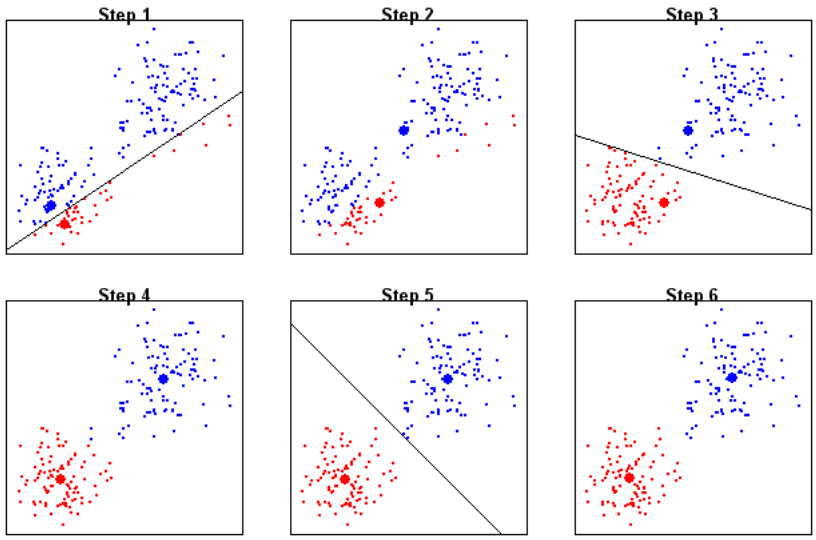
Kitas būdas – skirstyti algoritmus į grupes pagal veikimo principą[KCA14]. Toliau šiame darbe bus aprašomi populiariausi keturių tipų algoritmai.

2.3. K-vidurkiai

K-vidurkių (angl. *k-means*) algoritmas priklauso skirstant (angl. *partitional clustering*) arba centroidais[3] paremtu klasterizavimu grupėmis[Mac+67]. Dažniausiai šis algoritmas pradeda nuo atsitiktinių klasterių ir juos gerina vis keisdamas centroidų padėtį.

Veikimo principas:

1. Sukuriami k centroidai. („Inicijavimo metodai“ poskyryje detaliau apie tai).
2. Objektai priskiriami artimiausiam (pagal matavimo matą) centroidui. Taip sudaromi klasteriai.
3. Išsaugome sumą atstumų tarp centroidų ir jiems priklausančių objektų (toliau $Cdis$).
4. Apskaičiuojame klasterio naują centrą pagal jam priskirtus objektus. Jeigu centroido pozicija pasikeitė, einam į 2.-ą žingsnį, jeigu ne – sustojam.



3 pav. 1,3,5 žingsniai parodo dalijimo procesą; 2,4,6 – perskaičiavimo
Šaltinis: `https://medium.com/@dilekamadushan/introduction-to-k-means-clustering-7c0ebc997e00`

K-vidurkių metodas dažniausiai randa lokalų maksimumą, o globalaus maximumo suradimas yra NP – sudėtinga problema. Ji dažniausiai sprendžiama kelis kartus kartojant algoritmą su skirtingomis inicijavimo reikšmėmis ir parenkant tą, kuri grąžina mažiausią $Cdis$.

Dėl palygint greito konvergavimo ir paprasto veikimo, šis algoritmas tapo vienas populiariausių klasterizavimo algoritmų [WKQ+08]. Tačiau ne vieną kartą pastebėta, jog taikant K-vidurkių metodą, algoritmo iteracijų skaičius yra kur kas mažesnis nei klasterizuojamų objektų skaičius. Norint gauti kokybiškus klasterius, svarbios geros k reikšmės ir pradinės centroidų padėties parinkimas.

[3] Taškai apibūdinantys klasterį, bet nebūtinai jam priklausantys.

2.3.1. K parinkimas

Nors kai kurioms problemoms spręsti mes iš anksto galime žinoti k, bet daugumai problemų šis sprendimas gali atrodyti atsitiktinis. Vis didinat k reikšmę $Cdis$ sumažėtų ir mūsų modelis būtų vis „tikslesnis“, kol galiausiai kiekvienas objektas turėtų atskirą klasterį.

Vienas iš sprendimų yra „bausti“ (angl. *penalize*) sudėtingumą. Sum = $Cdis$ + $Complexity$. Nes didinat k, $Cdis$ mažėja vis lėčiau, galiausiai didėjantis $Complexity$ pradeda didinti $Suma$ reikšmę ir tada sustojame didinti k. Yra keletas skirtingų $Complexity$ apskaičiavimo funkcijų. Viena žinomiausių yra X-Means[PM+00].

2.3.2. Centroidų inicijavimo metodai:

1. Atsitiktinis – dažniausiai parenkama atsitiktinio objekto padėtis. Tai garantuoja, kad centroidai bus šalia duomenų, bet tuo pačiu padidėja šansas, kad bus parinkti šalimais esantys elementai.

2. Atstumu paremti (angl. *distance-based*) – pirmą centroidą parenkame atsitiktinai, tada surandame tolimiausią tašką nuo jo ir jį parenkame kitu centroidu, tada parenkame tolimiausią nuo dviejų esamų centroidų ir taip toliau. Nors tai išsprendžia problemą su šalimais esančiais klasteriais, bet šiuo atveju didelė tikimybė, kad bus parinkti atsiskyrėliai taškai (angl. *outlier*).

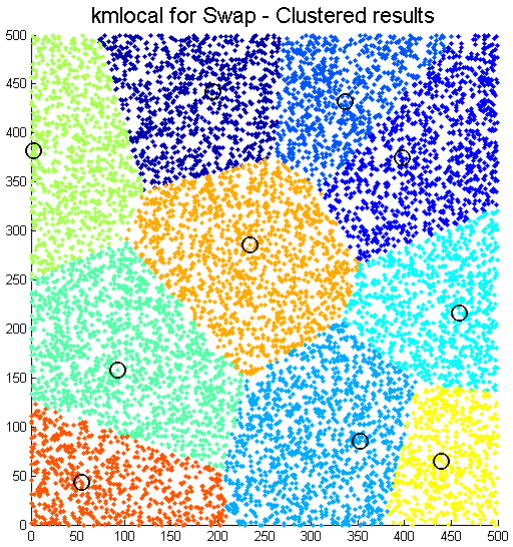
3. k-means++ (atsitiktinis + atstumo)[AV07] – parenka pirmą centroidą atsitiktinai, tada antrą vėl parenka atsitiktinai, bet šį kartą papildomai pritaikę svorį, proporcingą atstumui iki artimiausio centroido, pakeltą kvadratu ir kartojame iš naujo. Šis metodas parenka centroidus, kurie yra tolimai nuo kitų, bet kartu yra tankiose vietose.

2.3.3. K-vidurkio algoritmo savybės:

- Sudėtingumas.
- Sugeneruoja griežtus, plokščius klasterius.
- Sudaryti klasteriai yra sferinės formos, atitinka voronoi diagramas4.
- Privalome iš anksto nurodyti k pradinę reikšmę ir centroidų pradinę padėtį[4].
- Dėl palyginti spartaus veikimo ir paprastos realizacijos šis algoritmas rekomenduojamas kaip pirmas klasterizavimo metodas su naujais duomenų komplektais, nes rezultatai dažnai yra pakankami geri[AV06].
- Rezultatai gali būti smarkiai paveikti taškų atsiskyrėlių.

[4]Tai iš dalies prieštarauja klasterizavimo principui, kad klasterizavimas turi būti atliktas be papildomos informacijos.

- Labai tikėtina, kad du šalimais esantys taškai bus skirtingose grupėse.



4 pav. K-vidurkių sugeneruoti klasteriai atitinka Voronoi diagramas
Šaltinis: `https://summerofhpc.prace-ri.eu/quizz-clustered-data-using-k-means/`

2.3.4. K-vidurkio algoritmo alternatyvos

Dėl palygint paprasto veikimo principo ir populiarumo k-vidurkių algoritmas susilaukė daug modifikuotų versijų. Keletas jų:

- *K-Medians* – naudoja medianas vietoj vidurkių[JD88]. Šis metodas mažiau jautrus taškams atsiskyrėliams, bet yra lėtesnis.
- *Fuzzy C-means* – negriežta k-vidurkių versija[Dun73].
- *Bisecting k-means* – hierarchinė k-vidurkių versijak-means[SKK+00] .

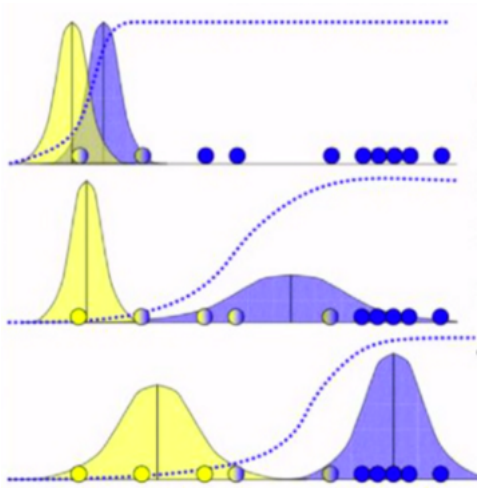
2.4. Lūkesčių-maksimizavimo

Lūkesčių-maksimizavimo (angl. *expectation–maximization*), (toliau EM) priklauso pasiskirstymo modelių (angl. *distribution models*) grupei ir dar yra vadinamas tikimybiniu klasterizavimo algoritmu. Šis metodas remiasi statistiniais pasiskirstymais, todėl tinka dirbtiniams duomenų rinkiniams. EM yra vienas garsiausių šios grupės algoritmų[WKQ+08] ir naudoja Gauso mišinių modelį (angl. *Gaussian mixture*). EM atveju duomenų rinkinys yra modeliuojamas su nustatytu Gauso pasiskirstymų (angl. *Gauss distribution*) skaičiumi, kuris yra atsitiktinai inicijuojamas, o jo parametrai yra iteratyviai optimizuoti, kad geriau atitiktų duomenų rinkinį.

Šis metodas yra laikomas bendresniu k-vidurkių metodo variantu, todėl turi daug panašumų: abu metodai randa lokalų maksimumą, todėl gali prireikti paleisti kelis kartus su skirtingai inicijuotais parametrais. Taip pat EM turi du skirtingus parametrus k ir klasterių pradines padėtis. EM sugeneruoti klasteriai yra negriežti ir kiekvienas taškas su atitinkama tikimybe gali priklausyti visiems klasteriams. Jei norime paversti griežtu klasteriu, tereikia kiekvieną tašką priskirti klasteriui, kuriam jis priklauso, su didžiausia tikimybe.

Veikimo principas:

1. Parenkame k Gauso pasiskirstymų pradines reikšmes.
2. Suskaičiuojame tikimybę, su kuria objektai priklauso kiekvienam iš pasiskirstymų (Lūkesčių (angl. *expectation*) žingsnis).
3. Pagal tikimybes pakoreguojame pasiskirstymus (Maksimizavimo (angl. *maximization*) žingsnis).
4. Kartojame 2. ir 3. žingsnius kol konverguoja (klasteriai nesikeičia tarp žingsnių).



5 pav. EM pavizdys su vienmačiais duomenimis.
Šaltinis: `https://www.youtube.com/watch?v=iQoXFmbXRJA`

2.4.1. Parametrų parinkimas

Naudojant EM iškyla ta pati problema kaip su k-vidurkių metodu – turime iš anksto parinkti k, pradines klasterių padėtis ir parametrus. Sprendimai iš principo yra labai panašūs į k-vidurkių, tik yra sudėtingesni, nes EM klasteriai yra Gauso pasiskirstymai, kurie turi daugiau parametrų nei centroidai.

2.4.2. Lūkesčių-maksimizavimo algoritmo savybės:

- Sugeneruoja negriežtus, plokščius klasterius.
- Teoretiškai konvergavimas gali užtrukti amžinybę, todėl reikia nurodyti nuo kokio minimalaus pasikeitimo nustoti ieškoti geresnio varianto.
- Negarantuoja, kad konverguos globaliam maksimume.
- Lengvai galime paversti į griežtus klasterius.
- Galime turėti taškų, kurie pagal tikimybes tikėtina vienodai priklauso keliems klasteriams. Jei mūsų užduočiai tinka, galime šiuos taškus apibrėžti „tarpiniais“ ir nepriskirti jų nei vienam klasteriui.
- Ne toks spartus kaip k-vidurkių metodas.
- Rezultatai sunkiau interpretuojami nei taikant griežtus metodus.

2.5. Hierarchinis

Hierarchiniai (angl. *hierarchical*) klasterizavimo algoritmai yra viena iš populiariausių dokumentų klasterizavimo algoritmų grupių, nes nereikalauja iš anksto nurodyti klasterių kiekio ar slenksčio (angl. *threshold*). Todėl jie grąžina bendrą visų klasterių tarpusavio priklausomybių struktūrą ir tai leidžia nuspręsti koks klasterių skaičius yra optimalus.
Hierarchiniai klasterizavimo algoritmai skirstomai į skaidymo ir jungimo.

2.5.1. Skaidymo algoritmai

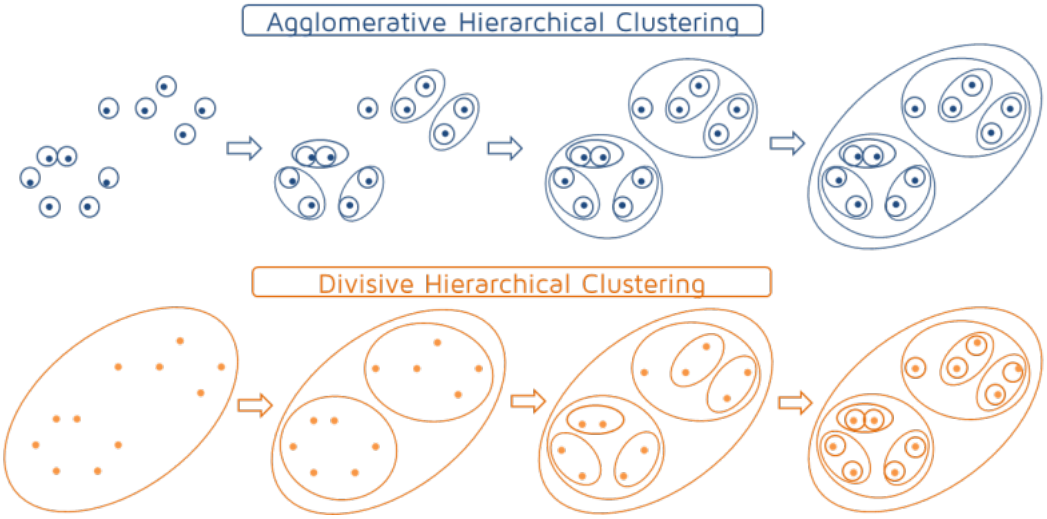
Skaidymo (angl. *divisive*) algoritmai – objektų priskyrimą pradeda iš viršaus į apačią (angl. *top-down*). Pradžioje visus objektus priskirdami vienam klasteriui, tada nuosekliai dalindami į smulkesnius klasterius, kol galiausiai kiekvienas objektas turi po atskirą klasterį.

2.5.2. Jungimo algoritmai

Jungimo (angl. *agglomerative*) algoritmai – objektus priskiria priešingai nei skaidymo – iš apačios į viršų (angl. *bottom-up*) . Iš pradžių kiekvienas objektas priskiriamas atskiram klasteriui, tada panašiausi klasteriai sujungiami, tai kartojama kol galiausiai turime vieną klasterį.

2.5.3. Atstumo matavimas / jungimo metodai

Naudojant hierarchinio klasterizavimo metodą iškyla problema: kaip išmatuoti atstumą tarp klasterių. Tam yra naudojami klasterių jungimo metodai, kurie kai kuriuose šaltiniuose yra laikomi atskirais klasterizavimo algoritmais. Keletą populiariausių jungimo metodų yra: [CS08]



6 pav. Grafinis pavizdys kuo skiresi jungimo nuo skaidymo algoritmai.
Šaltinis: `https://quantdare.com/hierarchical-clustering/`

Artimiausio kaimyno (angl. *nearest neighbor, single link*) – atstumas tarp artimiausių objektų atskiroje poroje klasterių Naudojant šį jungimo metodą, sudaromi klasteriai įgauna ilgų objektų grandinių formą.

Tolimiausio kaimyno (angl. *furthest neighbor, complete link*) – atstumas tarp tolimiausių objektų atskiroje poroje klasterių. Naudojant šį jungimo metodą, sudaromi klasteriai įgauna sferinę formą.

Vidutinių atstumų (angl. *average link*) – vidutinis atstumas tarp visų įmanomų objektų atskiroje poroje klasterių. Tai lėtai veikiantis, bet mažiau įtakojamas taškų atsiskyrėlių, metodas.

Centroidų (angl. *centroids*) – kaip ir k-vidurkių algoritme surandame visų klasterio objektų centroidą ir tada pamatuojame atstumą iki kito klasterio centroido.

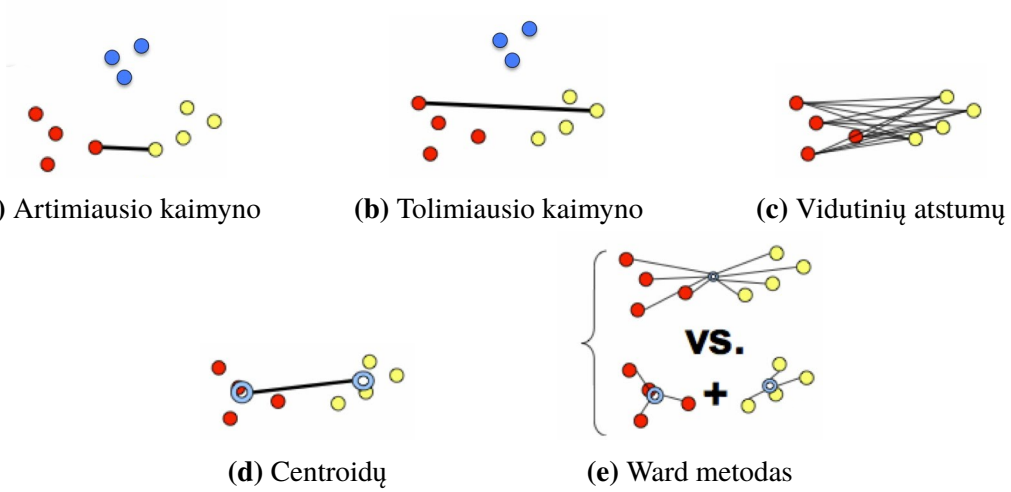
Ward metodas[5]

1. Apskaičiuojame kiekvieno klasterio centroidus ir $Cdis$ (taip kaip k-vidurkių metodas(2.3)).
2. Išmatuojame kiekvienos įmanomos klasterių poros naujus centroidus ir $Cdis$.
3. Jungiame tą porą klasterių, kurių nauja $Cdis$ reikšmė po sujungimo mažiausiai pasikeis.

2.5.4. Hierarchinių algoritmų savybės:

- Skaidymo sudėtingumas – $\mathcal{O}(2^n)$, bet praktikoje naudojami heuristiniai metodai (kaip k-vidurkių) skaidymą pagreitina.
  Jungimo sudėtingumas – bendru atveju $\mathcal{O}(n^3)$, tačiau panaudojus krūvos (angl. *heap*) duomenų struktūrą, sudėtingumą galime sumažinti iki $\mathcal{O}(n^2 logn)$, jei naudojame artimiausio[Sib73] ar tolimiausio[Def77] kaimyno atstumus, galima optimizuoti iki $\mathcal{O}(n^2)$.

[5]Čia aprašyta minimalaus pakitimo kriterijaus (angl. *Minimum variance criterion*) versija



(a) Artimiausio kaimyno **(b)** Tolimiausio kaimyno **(c)** Vidutinių atstumų

(d) Centroidų **(e)** Ward metodas

7 pav. Atstumo matavimas / jungimo metodai.
Šaltinis: `https://www.youtube.com/watch?v=vg1w5ZUF5lA`

- Sugeneruoja griežtus, hierarchinius klasterius.
- Rezultatas – dendograma, kuri vaizduojama taip: x-ašyje išdėstyti dokumentai, y-ašies atstumas parodo klasterių skirtingumą: artimiausi klasteriai jungsis žemiausiai, o tolimiausi – pačiam viršuje. Norint gauti konkretų klasterių skaičių, „nupjauname“ dendrogramą pasirinktame lygyje.
- Nereikia iš anksto nurodyti norimų klasterių kiekio ar slenksčio (angl. *threshold*).
- Reikia nurodyti papildomą atstumo matą.
- Suteikia daugiau informacijos nei plokščiasis klasterizavimas.
- Mažiau našus nei plokščiasis klasterizavimas.

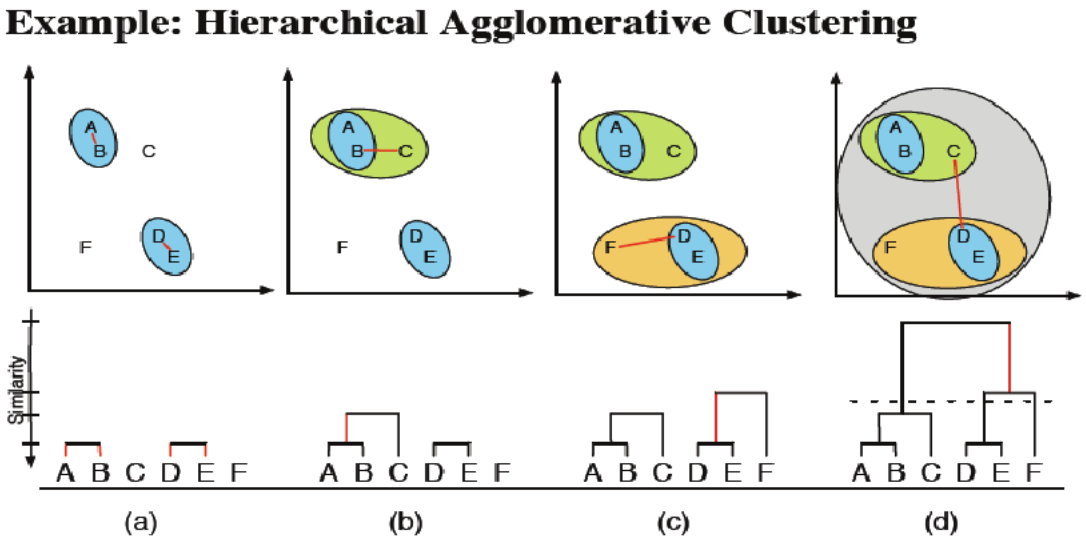
2.6. DBSCAN

DBSCAN (angl. *density-based spatial clustering of applications with noise*)[EKS+96] priklauso objektų tankiu pagrįsto klasterizavimo (angl. *density-based clustering*) metodų grupei. Taikant šį metodą duomenys sugrupuojami į klasterius remiantis tuo, kad duomenys yra pakankamai arti vieni kitų (tankūs). Tuo tarpu retai išsidėsčiusius duomenis laikome triukšmu (angl. *noise*) ir nepriskiriame jokiam klasteriui. DBSCAN tankį apibrėžia kaip taško aplinkoje (pagal parenkamą spindulį) esančių taškų minimalų kiekį (parenkamas parametras).

DBSCAN reikalauja dvejų parametrų: - maksimalus atstumas iki kaimyno ir $MinPts$ – minimalus kaimynų kiekis.

DBSCAN taškai gali būti trijų rūšių: **šerdiniai** taškai (angl. *core points*) ir **ribiniai** taškai (angl. *border points*), kurie kartu sudaro klasterius, o **atsiskyrėliai** taškai (angl. *utliers*) laikomi triukšmu.

Taisykės, pagal kurias taškai yra suskirstomi:

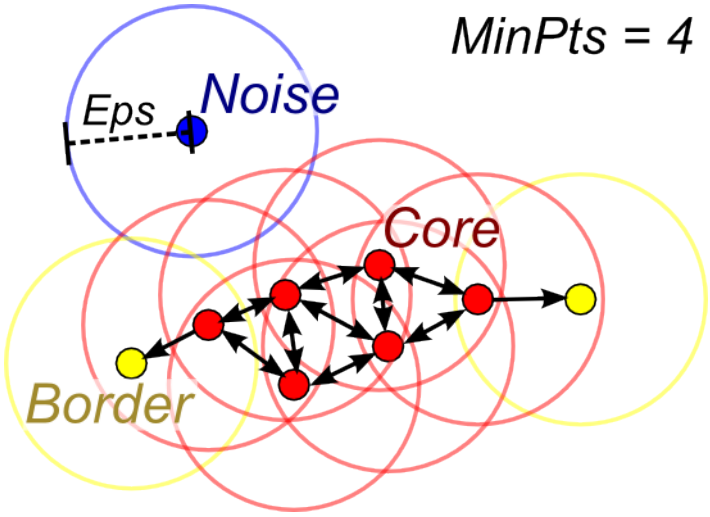


8 pav. Hierarchnio jungiamojo klasterizavimo pavizdys:
a) – Žingsnis rodo kaip sujungiami pavieniai objektai į klasterius.
b) – Kaip prie klasterių prijungiami objektai.
d) – Kaip galutinis rezultatas yra paverčiams į tris atskirus klasterius „atpjaunant“ dendogramą.
Šaltinis: [JWL12]

- Du taškai yra laikomi kaimynais, jeigu atstumas tarp jų yra mažesnis arba lygus .
- Taškas yra laikomas **šerdiniu**, jeigu turi nors$minPts$ kaimynų (įskaitant patį tašką).
- Taškas yra laikomas **ribiniu**, jei turi mažiau nei $minPts$ kaimynų, bet vienas iš jų yra šerdinis taškas.
- Visi taškai nepasiekiami per bet kokį kitą tašką yra laikomi **atsiskyrėliais**.

Veikimo principas:

1. Atsitiktinai parenkame neaplankytą tašką p. Jei nebėra neaplankytų taškų stojame.
2. Jeigu p turi pakankamai kaimynų, jis tampa **šerdiniu** tašku ir sudaro naują klasterį:
   2.1. Visi kaimyniniai taškai priskiriami klasteriui.
   2.2. Atliekame paiešką į gylį tarp neaplankytų klasterio taškų.
      2.2.1. Jei aplankytas taškas turi pakankamai kaimynų, pažymime kaip **šerdinį** ir jo kaimynus pridedame prie klasterio. Grįžtame į 2.2. žingsnį.
      2.2.2. Jei aplankytas taškas neturi pakankamai kaimynų, pažymime kaip **ribinį**. Grįžtame į 2.2. žingsnį.
      2.2.3. Jei visi klasterio taškai aplankyti, grįžtame į 1.-ą žingsnį.
3. Jei p neturi pakankamai kaimynų jis tampa **atsiskyrėliu** tašku (jei vėliau paaiškės, kad vienas iš kaimyninių taškų yra šerdinis, p taps **ribiniu** tašku). Grįžtame į 1.-ą žingsnį.



9 pav. DBSCAN metodo taškų rūšys. Mėlynas – pašalinis, geltonas – pasienio, raudonas – centrini
Šaltinis: `https://stats.stackexchange.com/questions/194734/dbscan-what-is-a-core-point`

2.6.1. Parametrų parinkimas

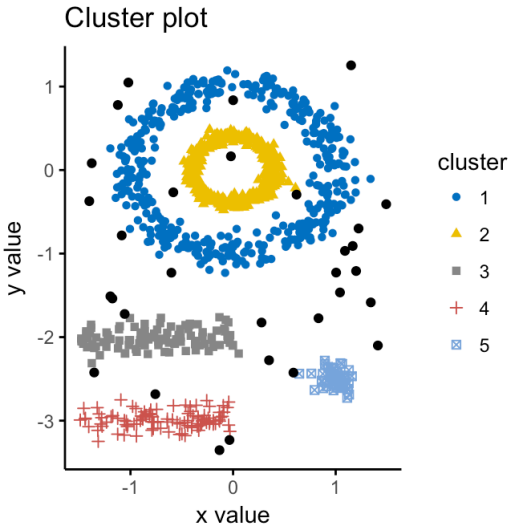
Idealioje situacijoje, jei dirbtume su fizinį pasaulį atitinkančiais duomenimis, reikštų fizinį atstumą tarp objektų, o MinPts laikytume mažiausiu norimų klasterių dydžiu. Deja, tokie atvejai reti, todėl yra kelios taisyklės kaip parinkti šiuos parametrus:

- $MinPts$ – minimalus kaimynų kiekis. $MinPts = 1$ netinkama, nes tada kiekvienas atsiskyrėlis taškas turėtų atskirą klasterį. $MinPts = 2$ sugeneruotų tokį pat rezultatą kaip hierarchinio klaserizavimo metodas su sinlge link metrika ir dendrograma atkirpta ties aukščiu. Taigi $MinPts$ reikšmė turėtų būti mažiausiai 3. Kaip apytikslė taisyklė $MinPts = 2 \cdot dim$ [SEK+98].
- - maksimalus atstumas iki kaimyno. reikšmę galime pasirinkit naudojant k-artimiausių kaimynų grafą (angl. *k-nearest neighbor graph*) (k-NNG). Taip pat alternatyviai DBSCAN versijai OPTICS (angl. *ordering points to identify the clustering structure*) nereikia parametro, bet dėl to grąžinami klasteriai yra hierarchiniai.

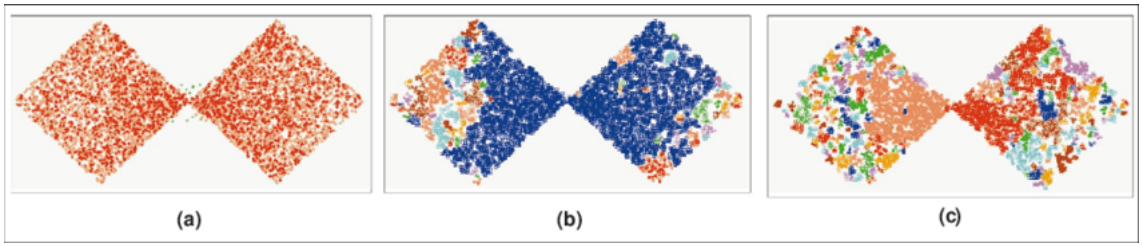
2.6.2. DBSCAN algoritmo savybės:

- Bendru atveju, sudėtingumas yra $\mathcal{O}(n^2)$. Jeigu duomenys patalpinti erdviniame indekse (angl. *spatial index*), sudėtingumas bus $\mathcal{O}(n \log n)$.
- Sugeneruoja griežtus, plokščius klasterius.
- Duomenims pakanka vienos iteracijos.
- Nereikia iš anksto nurodyti klasterių skaičiaus.

- Dirbant su tam tikrais duomenimis, parametrų parinkimas gali būti intuityvus, todėl atliekamas konkrečios srities eksperto.
- Reikia iš anksto nustatyti 2 skirtingus jautrius parametrus, todėl net dėl mažo jų pasikeitimo, rezultatai gali labai skirtis (pvz.: 11 paveikslelis). Pavyzdžiui, per didelė $minPts$ reikšmė reikštų, kad maži klasteriai bus laikomi triukšmu. Esant per mažai reikšmei objektai bus sujungti į vieną klasterį.
- Sudaryti klasteriai gali būti sudėtingų formų, vienas klasteris gali būti apsuptas kito klasterio (pvz.: 10 paveikslelis).
- Labai gerai susitvarko su taškais atsiskyrėliais.
- Nesusitvarko su skirtingo tankio klaseriais.
- Nėra visiškai deterministinis, nes priklausomai kokia eilės tvarka buvo parenkami ribiniai taškai, jie gali priklausyti skirtingiems klasteriams. Šią problemą išsprendžia DBSCAN*[CMZ+15] algoritmas, kuris ribinius taškus laiko triukšmu.



10 pav. DBSCAN sugeneruotų klasterių formų įvairovė.
Šaltinis: `http://www.sthda.com/english/wiki/wiki.php?id_contents=7940`



11 pav. Skirtingos ε reikšmės.
Šaltinis: [KHK99]

3. Kokybės vertinimas

Visi klasterizavimo metodai turi bendrą silpnybę – jų paskirtis atrasti duomenų struktūras, tačiau jie gali atrasti jas ir tais atvejais, kai duomenyse nėra jokių struktūrų [TK03]. Todėl klasterizavimo kokybės įvertinimas (angl. *evaluation*) yra vienas svarbiausių klasterizavimo proceso etapų. Jo metu gauti rezultatai parodo ar objektai (duomenys) buvo teisingai sugrupuoti į klasterius be išankstinės informacjos apie grupes. Egzistuoja 4 kriterijai klasterizavimo rezultatų kokybei įvertinti[FS+07]:

1. **Vidiniai** (angl. *internal*) kriterijai kokybę vertina lygindami objektų vienoduose klasteriuose panašumą ir objektų skirtumą skirtinguose klasteriuose. Deja, šio tipo kriterijai nėra universalūs, skirtingiems klasterizavimo metodams reikia parinkti skirtingus vidinius kriterijus.

2. **Išoriniai** (angl. *external*) kriterijai kokybę vertina lygindami gautus klasterius su jau iš anksto žinomomis duomenų klasėmis. Taigi, šiuo atveju vertiname neprižiūrimo mokymosi metodus su prižiūrimo mokymosi problemoms parengtais duomenimis. Nors labai tikėtina, kad neprižiūrimo mokymosi metodu sugeneruoti rezultatai bus blogesni, bet tai vis tiek labai vertingas vertinimo metodas. Tačiau svarbu atkreipti dėmesį, kad duomenis dažnai galima sugrupuoti keliais skirtingais būdais ir su duomenimis atėjusios etiketės (angl. *labels*) nebūtinai yra vienintelis galimas variantas.

3. **Rankiniai** (angl. *manual*) kriterijai, kai kokybė yra vertinama žmogaus. Praktikoje tokiu būdu visų sudarytų klasterių vertinimas užimtų labai daug laiko. Todėl dažniausiai vertintojui duodama pora objektų ir klausiama ar jie turėtų būti kartu, ar atskirai. Surinkę pakankamai rezultatų iš vertintojų, palyginame su rezultatais, gautais taikant klasterizavimo algoritmą. Taip pat šiuo atveju galima taikyti duomenų vizualizaciją, deja tai tampa ypač sudėtinga su didelės apimties duomenimis (tekstiniais dokumentais).

4. **Netiesioginiai** (angl. *indirect*) kriterijai įvertina ar klasterizavimas yra vertingas žingsnis, didesnės problemos sprendimui (pvz., klasterizavimas naudojamas vaizdų atpažinimui kaip tarpinis žingsnis matmenų kiekiui sumažinti). Todėl galime stebėti didesnės problemos sprendimo rezultatus su skirtingais klasterizavimo metodais (ar jų parametrais) ir parinkti tinkamiausią metodą.

Išvados

Kursiniame darbe buvo pasiektas užsibrėžtas tikslas – išnagrinėti ir aprašyti klasterizavimo metodai, jų veikimas ir savybės. Taip pat apžvelgtos panašumo funkcijos, tekstinių duomenų išgavimo ir jų parengimo darbui su klasterizavimo algoritmais būdai bei kokybės įvertinimo kriterijai.

Kursinio darbo metu surinkta ir išnagrinėta informacija apie dokumentų klasterizavimą bus panaudota projektiniame darbe atliekant eksperimentinį tyrimą. Taip pat projektiniame darbe planuoju išsamiau aprašyti tekstų parengimą, klasterių vertinimą, bei įrankius reikalingus šioms užduotims atlikti.

Literatūra

[ATL13] Salem Alelyani, Jiliang Tang ir Huan Liu. Feature selection for clustering: a review. *Data clustering: algorithms and applications*, 29:110–121, 2013.

[AV06] David Arthur ir Sergei Vassilvitskii. How slow is the k-means method? *Proceedings of the twenty-second annual symposium on computational geometry*. ACM, 2006, p.p. 144–153.

[AV07] David Arthur ir Sergei Vassilvitskii. K-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms*. Society for Industrial ir Applied Mathematics, 2007, p.p. 1027–1035.

[CMZ+15] Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek ir Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *Acm transactions on knowledge discovery from data (tkdd)*, 10(1):5, 2015.

[CS08] Prabhakar Raghavan Christopher D. Manning ir Hinrich Schütze. Definition and examples of metric spaces. 2008. URL: `https://nlp.stanford.edu/IR-book/html/htmledition/single-link-and-complete-link-clustering-1.html`.

[Def77] Daniel Defays. An efficient algorithm for a complete link method. *The computer journal*, 20(4):364–366, 1977.

[Dun73] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, 1973.

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu ir k.t. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*. Tom. 96. (34), 1996, p.p. 226–231.

[FPS96] Usama Fayyad, Gregory Piatetsky-Shapiro ir Padhraic Smyth. From data mining to knowledge discovery in databases. *Ai magazine*, 17(3):37, 1996.

[FS+07] Ronen Feldman, James Sanger ir k.t. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.

[Hua08] Anna Huang. Similarity measures for text document clustering. *Proceedings of the sixth new zealand computer science research student conference (nzcsrsc2008), christchurch, new zealand*, 2008, p.p. 49–56.

[JD88] Anil K Jain ir Richard C Dubes. Algorithms for clustering data, 1988.

[JWL12] Peter Janssen, Carsten Walther ir Matthias Lüdeke. *Cluster analysis to understand socio-ecological systems: a guideline*. Potsdam-Institut für Klimafolgenforschung, 2012.

[KCA14] Ammar Ismael Kadhim, Yu-N Cheah ir Nurul Hashimah Ahamed. Text document preprocessing and dimension reduction techniques for text document clustering. *Artificial intelligence with applications in engineering and technology (icaiet), 2014 4th international conference on*. IEEE, 2014, p.p. 69–73.

[KHK99] George Karypis, Eui-Hong Han ir Vipin Kumar. Chameleon: hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[LA99] Bjornar Larsen ir Chinatsu Aone. Fast and effective text mining using linear-time document clustering. *Proceedings of the fifth acm sigkdd international conference on knowledge discovery and data mining*. ACM, 1999, p.p. 16–22.

[Mac+67] James MacQueen ir k.t. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth berkeley symposium on mathematical statistics and probability*. Tom. 1. (14). Oakland, CA, USA, 1967, p.p. 281–297.

[MPP] K Mugunthadevi, SC Punitha ir M Punithavalli. Survey on feature selection in document clustering.

[OCo04] John O'Connor. Definition and examples of metric spaces. 2004. URL: `http://www-history.mcs.st-and.ac.uk/~john/MT4522/Lectures/L5.html`.

[PM+00] Dan Pelleg, Andrew W Moore ir k.t. X-means: extending k-means with efficient estimation of the number of clusters. *Icml*. Tom. 1, 2000, p.p. 727–734.

[SEK+98] Jörg Sander, Martin Ester, Hans-Peter Kriegel ir Xiaowei Xu. Density-based clustering in spatial databases: the algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194, 1998.

[Sib73] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973.

[SKK+00] Michael Steinbach, George Karypis, Vipin Kumar ir k.t. A comparison of document clustering techniques. *Kdd workshop on text mining*. Tom. 400. (1). Boston, 2000, p.p. 525–526.

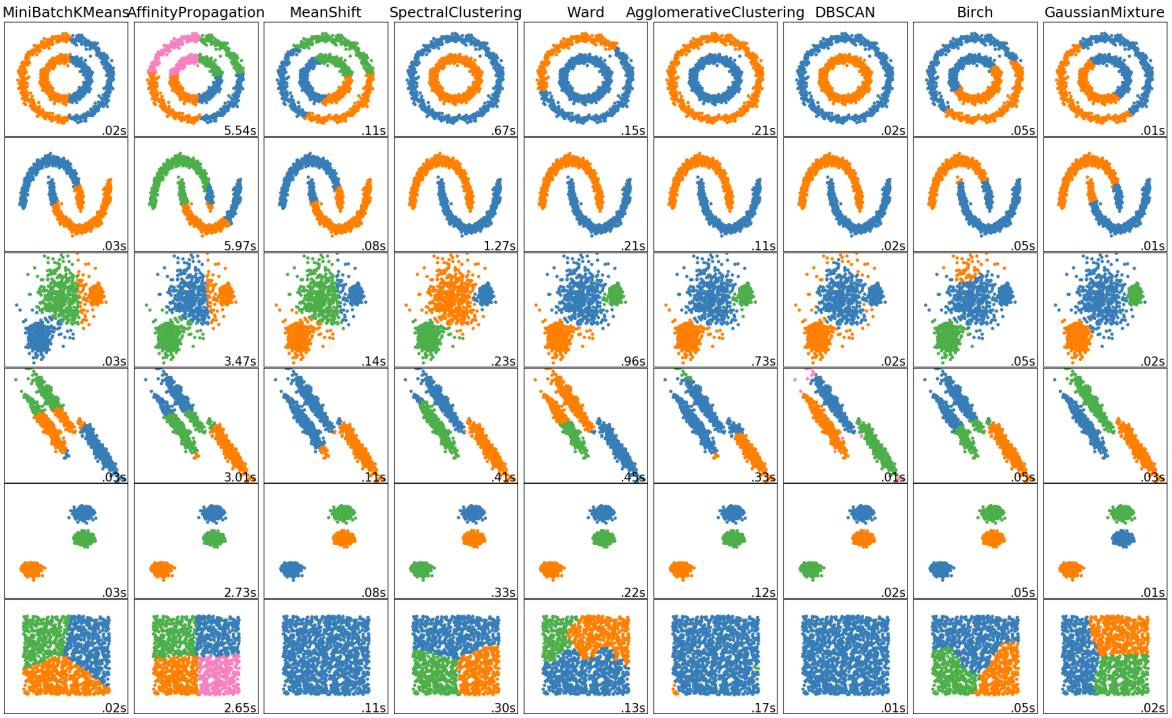
[Tan+07] Pang-Ning Tan ir k.t. *Introduction to data mining*. Pearson Education India, 2007.

[TK03] Sergios Theodoridis ir Konstantinos Koutroumbas. Feature selection. *Pattern recognition*, 5:261–322, 2003.

[WKQ+08] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh ir k.t. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

Priedas Nr. 1

Klasterizavimo algoritmų vizualizacija



12 pav. Klasterizavimo metodų palyginimas: MiniBatchKMeans – k-vidurkių;
Ward, AgglomerativeClustering – jungiamasis hierarchinis; GaussianMixture – Lūkesčių-Maksimizavimo
Šaltinis: `http://scikit-learn.org/stable/modules/clustering.html`