SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 121

Usporedba metoda grupiranja primjenom programskog jezika Python

Tomislav Bjelčić

Zagreb, rujan 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 12. ožujka 2021.

ZAVRŠNI ZADATAK br. 121

Pristupnik:	**Tomislav Bjelčić (0036513877)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	izv. prof. dr. sc. Goran Delač

Zadatak: **Usporedba metoda grupiranja primjenom programskog jezika Python**

Opis zadatka:

Istražiti algoritme grupiranja ostvarene u Python programskoj knjižnici Scikit-Learn. Odabrati i opisati primjeren podskup algoritama i skup podataka za vrednovanje. Programski ostvariti i provesti postupak vrednovanja algoritama s naglaskom na mjere učinkovitosti, primjerice gustoću grupa, te s obzirom na računalnu učinkovitost. Navesti korištenu literaturu i primljenu pomoć.

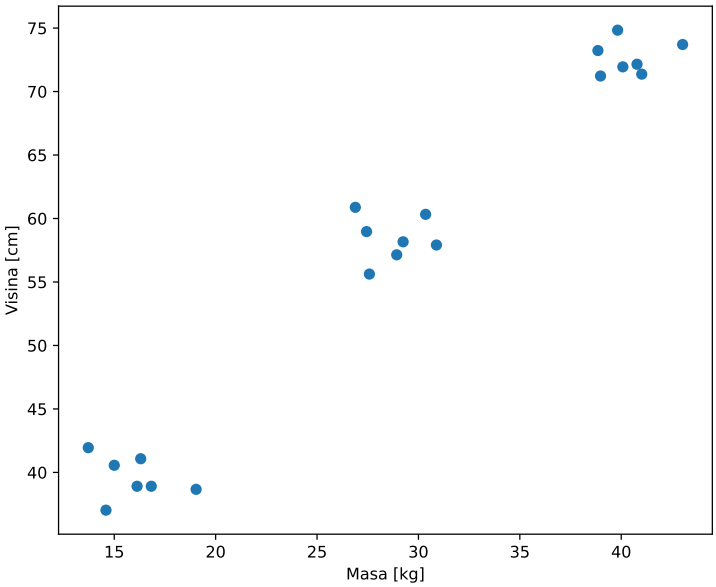
Rok za predaju rada: 11. lipnja 2021.

SADRŽAJ

1. Uvod

Grupiranje (engl. *clustering*) je postupak kojim se neki skup podataka razvrstava u skupine, odnosno grupe (engl. *clusters*), u kojima su podaci međusobno slični. Grupiranje je jedno od metoda **nenadziranog strojnog učenja** (engl. *unsupervised learning*), dakle ulazni podaci nisu označeni (engl. *unlabeled*), odnosno nemaju neku ciljnu vrijednost koja bi naznačila kojoj grupi pripada neki podatak. Algoritmi grupiranja iz takvih podataka onda moraju sami prepoznati grupe podataka, odrediti za svaki ulazni podatak kojoj grupi bi pripadao i, ukoliko je to moguće, odrediti **stršeće vrijednosti** (engl. *outliers*).

Primjerice, neka su na raspolaganju podaci o visini i masi odraslih pasa iz nekog skloništa za životinje. U ovom jednostavnom primjeru ulazni podaci imaju dvije značajke, odnosno dimenzije: visina i masa. Općenito, ulazni podaci mogu imati proizvoljno mnogo značajka. Ulazni podaci su na slici 1.1 prikazani kao točke na grafu gdje os apscisa predstavlja masu, a os ordinata predstavlja visinu pojedinog psa.



Slika 1.1: Podaci o psima u skloništu za životinje

Podaci se prirodno grupiraju u tri grupe koje odgovaraju različitim pasminama. Ulazni podaci nisu označeni, dakle na raspolaganju je samo masa i visina. Algoritmi grupiranja moraju prepoznati da se radi o tri grupe i slične točke pridruži istoj grupi. Općenito, poželjno je da algoritam grupiranja podatke grupira na onaj način koji odgovara prirodnom grupiranju. U navedenom primjeru prirodno grupiranje je vizualno jasno i ono odgovara pasminama, no to ne mora biti slučaj. Neće uvijek ni vizualnom metodom biti jednoznačno jasno kakvo je njihovo prirodno grupiranje. Ovdje bi mjera sličnosti mogla biti definirana koristeći udaljenost točaka (što su točke bliže, više su slične) s obzirom da se radi o brojevnim podacima, no općenito podaci mogu biti bilo kakve prirode, što će pojedini algoritmi grupiranja uzimati u obzir.

Cilj ovog završnog rada je opisati različite algoritme grupiranja, pokrenuti ih na odabranim skupovima podataka, različitim metodama ih vrednovati i na taj način usporediti. Uz to će biti objašnjeno zašto neki algoritmi daju bolje rezultate od drugih algoritama na određenim skupovima podataka.

Ostatak rada organiziran je na sljedeći način: u poglavlju 2 spominju se općeniti pojmovi vezani za algoritme grupiranja te su navedene osnovne podjele algoritama grupiranja. Poglavlje 3 opisuje četiri različita popularna modela i algoritma grupiranja. Osim opisa pojmova vezanih za svaki algoritam, spominje se njihova učinkovitost i kako ispravno odabrati parametre tih algoritama. Metode vrednovanja algoritama grupiranja dane su u poglavlju 4 gdje su opisane često korištene mjere koje ocjenjuju

rezultate grupiranja. U poglavlju 5 odabrano je nekoliko skupova podataka, pokrenuti su algoritmi grupiranja nad njima i rezultati su vrednovani mjerama opisanim u poglavlju 4. Konačno, zaključak je dan u poglavlju 6.

2. Općenito o algoritmima grupiranja

U kontekstu algoritama grupiranja[1] ulazni podaci (primjeri) su skup od N neoznačenih, višedimenzionalnih **točaka** (vektora)

$$\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$$

gdje svaka točka $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1}, x_n)$ ima n **značajki** (engl. *features*), odnosno dimenzija. Prostor podataka iz koje dolaze pojedine značajke, odnosno točke, mogu biti razne. Primjerice, može biti riječ o realnim brojevima (uvodni primjer), znakovima (primjerice grupiranje neoznačenih novinskih članaka), kategorijskim podacima poput vrijednosti istina/laž, itd. Neki algoritmi grupiranja imaju osnovne pretpostavke o tome iz kojeg prostora podataka dolaze značajke, što znači da nisu primjenjivi na one podatke koji nemaju ispunjene takve pretpostavke.

2.1. Udaljenost, sličnost i različitost točaka

U uvodnom poglavlju grupiranje je opisano kao postupak kojim se skup podataka razvrstava u grupe na takav način gdje su točke u istoj grupi međusobno manje-više "slične". Potrebno je definirati kako se određuje, odnosno kako se mjeri takva "sličnost" između dvije točke.

Neka je $\mathcal{V}$ prostor iz kojeg dolaze ulazni podaci čiji je $\mathcal{D}$ podskup, te neka je $\mathbb{R}$ skup realnih brojeva. **Udaljenost** (engl. *distance measure*) je funkcija

$$d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$$

koja za svaki $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$ zadovoljava tzv. svojstva **metrike**:

1. $d(\mathbf{x}, \mathbf{y}) = 0 \quad$ akko $\quad \mathbf{x} = \mathbf{y}$
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \qquad$ (simetričnost)

[1] Isto tako i u kontekstu nenadziranog strojnog učenja.

3. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (nejednakost trokuta)

Iz navedenih aksioma metrike slijedi da je $d(\mathbf{x}, \mathbf{y}) \geq 0$ za svaki $\mathbf{x}, \mathbf{y} \in \mathcal{V}$. Konceptualno, dvije točke koje se više "razlikuju" u svom prostoru imaju veću udaljenost. Ponekad su podaci takvi da su sve značajke realni brojevi, to jest $\mathcal{V} \subseteq \mathbb{R}^n$, tada je **Euklidova udaljenost** prikladna[2] mjera udaljenosti koja zadovoljava svojstva metrike:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.1}$$

Općenito, ako je $\mathcal{V}$ normirani vektorski prostor sa definiranom normom $\|\cdot\| : \mathcal{V} \to \mathbb{R}$ tada se udaljenost može definirati pomoću te norme:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\| \tag{2.2}$$

U slučaju realnih značajki i ako se koristi Euklidova norma, dobije se upravo 2.1.
Vrlo često se u praksi pojavljuju značajke koje nisu realni brojevi niti dolaze iz nekog drugog normiranog prostora. Tada se koriste neke druge mjere udaljenosti specijalizirane za specifične prostore $\mathcal{V}$. Primjerice, postoji

- **Hammingova udaljenost**: binarni nizovi duljine n
- **Jaccardova udaljenost**: skupovi i multiskupovi
- **Udaljenost uređivanja** (engl. *Edit distance*): znakovni nizovi

Sličnost (engl. *similarity measure*) je vrsta mjere koja, poput mjere udaljenosti, brojčano iskazuje koliko se razlikuju dvije točke. Za razliku od mjere udaljenosti, mjera sličnosti ne mora zadovoljavati sva svojstva metrike poput nejednakosti trokuta. Jednako vrijedi i za **mjeru različitosti** (engl. *dissimilarity measure*). Kako su dvije točke "sličnije" tako mjera sličnosti raste, a mjera različitosti pada. Detaljna definicija mjera sličnosti i različitosti nije navedena u ovom poglavlju iz razloga što se u algoritmima grupiranja (barem onim koji su opisani u ovom radu) koriste uglavnom mjere udaljenosti u svrhu kvantificiranja sličnosti ili različitosti točaka.

2.2. Vrste grupiranja

Ne postoji jedinstvena definicija grupe koja vrijedi za sve algoritme grupiranja. Svaki algoritam ima svoj model grupiranja u kojem je definirano što je to grupa, a sam algoritam pokušava točke grupirati na način koji to najviše odgovara za taj model. Različiti

[2]Euklidska udaljenost nije jedina mjera udaljenosti za takav prostor, ali je najkorištenija.

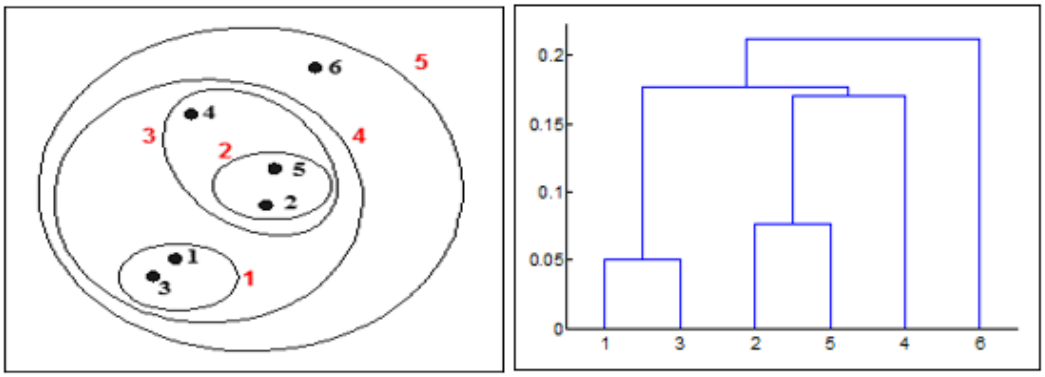
modeli grupiranja su detaljnije razmotreni u sljedećem poglavlju.
Postoje dvije općenite podjele algoritama grupiranja. S obzirom na koji način se oblikuju grupe, grupiranje može biti:

- **Hijerarhijsko grupiranje**
- **Particijsko grupiranje**

Kod hijerarhijskog grupiranja svaka grupa, počevši od grupe koja predstavlja cijeli skup točaka $\mathcal{D}$, ima podgrupe koje se tako rekurzivno dijele sve dok svaka točka nije svoja grupa. Na taj način se gradi hijerarhija grupa (od tud i naziv hijerarhijskog grupiranja) koji se može prikazati **dendrogramom**. Kod hijerarhijskog grupiranja postoje dva pristupa:

- **Aglomerativno grupiranje**: početno je svaka točka u svojoj grupi pa se spajaju u veće grupe;
- **Divizivno grupiranje**: početna grupa je $\mathcal{D}$ pa se ona dijeli u manje podgrupe.

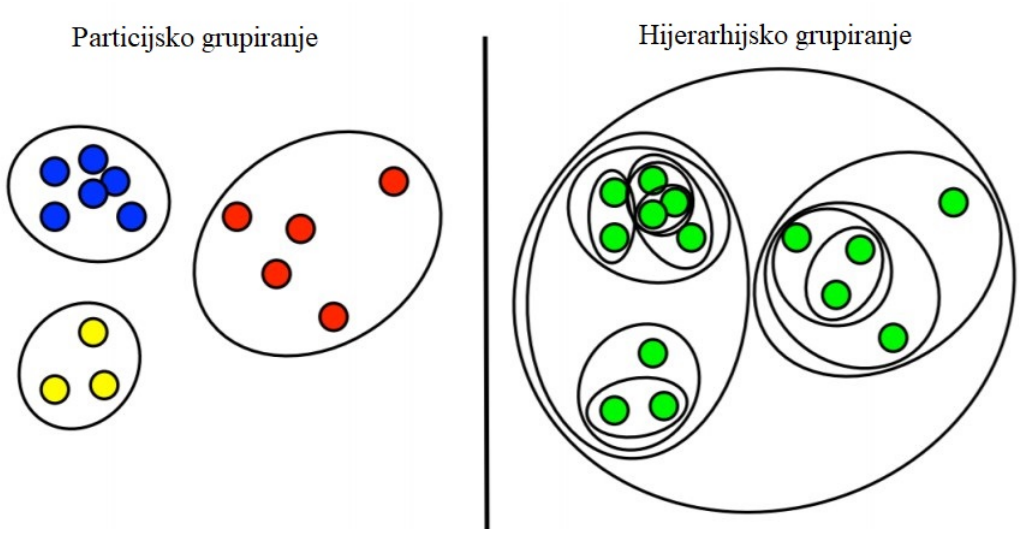
Na slici 2.1 prikazan je tijek nekog hijerarhijskog aglomerativnog grupiranja nekog ulaznog skupa točaka.



Slika 2.1: Tijek nekog hijerarhijskog aglomerativnog grupiranja [9]

Na lijevoj strani crnom bojom su označene i numerirane točke ulaznog skupa podataka, a crvenim brojevima je označen redoslijed kojim je postupak grupiranja spajao grupe u veće grupe. Na desnoj strani je postupak i rezultat grafički prikazan u obliku dendrograma. Na donjoj liniji su oznake točaka, a značenje brojeva na lijevoj liniji će biti objašnjeno u poglavlju koje detaljnije opisuje hijerarhijsko grupiranje. Ako se dendrogram promatra kao stablo čiji su čvorovi (pod)grupe, a listovi točke iz $\mathcal{D}$, tada aglomerativno grupiranje gradi dendrogram od listova prema korijenu, a divizivno grupiranje od korijena prema listovima.
Za razliku od hijerarhijskog grupiranja, particijsko grupiranje nema hijerarhiju grupa i

podgrupa. Rezultat grupiranja je neki fiksiran broj grupa bez neke unutarnje strukture osim prikladnih točaka u njima. Najpopularniji i najučinkovitiji algoritmi su upravo particijska grupiranja.
Primjer na slici 2.2 ilustrira razliku particijskog i hijerarhijskog grupiranja.



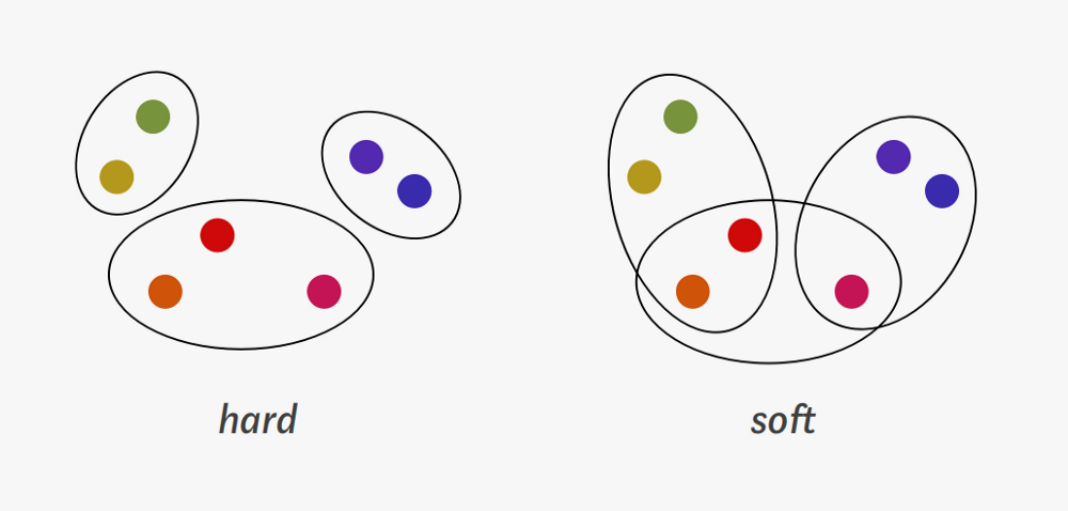
Slika 2.2: Usporedba particijskog i hijerarhijskog grupiranja [7]

Prirodne grupe točaka ponekad nije moguće jednoznačno odrediti. Grupiranja tada mogu, bilo grupiranje particijsko ili hijerarhijsko, neke točke svrstati ili u isključivo[3] jednu grupu ili u više grupa istovremeno. Prema tome se grupiranja dijele na:

- **Čvrsto grupiranje** (engl. *hard clustering*): jedna točka može pripadati isključivo jednoj grupi;
- **Meko grupiranje** (engl. *soft clustering*): jedna točka može pripadati više grupa sa nekom mjerom pripadnosti svakoj od tih grupa. Primjerice, mjera pripadnosti točke nekoj grupi se može prikazati kao vjerojatnost da ta točka pripada toj grupi.

Na slici 2.3 vizualno je prikazana razlika između čvrstog i mekog grupiranja nekog skupa točaka.

[3]Neki algoritmi grupiranja uključuju mogućnost stršećih vrijednosti i tada te točke nisu svrstane u nijednu grupu.



Slika 2.3: Primjer čvrstog i mekog grupiranja [5]

Primjerice crvenu točku u ovom primjeru je meko grupiranje svrstalo u dvije grupe istovremeno, a čvrsto grupiranje u isključivo jednu.

3. Algoritmi grupiranja

U daljnjim potpoglavljima opisan je podskup algoritama grupiranja implementiranih u Pythonovoj programskoj knjižnici **Scikit-learn**, unutar modula za grupiranje `sklearn.cluster`. Svako potpoglavlje sadrži najprije opis modela grupiranja kojeg taj algoritam koristi, a zatim opis samog algoritma te svojstva i složenost. Za neke algoritme i modele je dodatno opisano na koji način odrediti optimalne parametre algoritma.

3.1. Algoritam K-sredina

Algoritam **K-sredina** (engl. *K-means clustering*) je jednostavan, učinkovit i najpopularniji algoritam particijskog čvrstog grupiranja koji ulazni skup točaka $\mathcal{D}$ particionira u K grupa. K je broj grupa koje je potrebno proizvesti i on se zadaje unaprijed kao parametar algoritma. U poglavlju 3.1.6 su opisane neke metode određivanja optimalnog broja grupa K.

3.1.1. Model centroida

Algoritam K-sredina predstavlja svaku grupu sa jednom zamišljenom točkom koja označava središte te grupe: **centroidom**. Neka je $\mathcal{C} \subseteq \mathcal{D}$ grupa koja sadrži M točaka ($M = |\mathcal{C}|$, $|\mathcal{C}|$ predstavlja kardinalitet skupa $\mathcal{C}$), odnosno $\mathcal{C} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{M}$. Centroid te grupe $\boldsymbol{\mu}$ se definira kao:

$$\boldsymbol{\mu} = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{x} \in \mathcal{C}} \mathbf{x} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}^{(i)}$$

Centroid se računa na isti način kao i aritmetička sredina, odakle dolazi naziv K-sredina. Radi se o operacijama zbrajanja točaka te u konačnici dijeljenja sa nekim brojem. Dakle, da bi uopće bilo moguće računati centroide, nad točkama, odnosno ulaznim podacima, koje potječu iz prostora $\mathcal{V}$, moraju biti definirane operacije zbraja-

nja i množenja sa skalarom (u našem slučaju realnim brojem $\frac{1}{M}$). Ovo razmatranje dovodi do pretpostavke modela u kojem prostor podataka $\mathcal{V}$ mora biti vektorski prostor[1]. Standardni algoritam K-sredina je tada primjenjiv samo ako se radi o vektorskom prostoru. Najčešće je u pitanju podskup realnog koordinatnog prostora, odnosno $\mathcal{V} \subseteq \mathbb{R}^n$. Postoji i varijanta algoritma koja radi nad podacima bilo kakve prirode i koristi mjeru sličnosti (koja je općenitija, odnosno manje "zahtjevna" od mjere udaljenosti): **algoritam K-medoida**, no taj algoritam nije opisan u ovom radu.
Kao što pojam sugerira, centroid neke grupe konceptualno predstavlja centar, odnosno središte te grupe. Jasno je da se centroid grupe uopće ne mora nalaziti u grupi.

3.1.2. Kriterijska funkcija

Cilj algoritma K-sredina jest minimizirati **kriterijsku funkciju**. Neka je ulazni skup točaka $\mathcal{D}$ čvrsto grupiran u K grupa:

$$\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{C}_k, \qquad \forall\, (i \neq j) : \mathcal{C}_i \cap \mathcal{C}_j = \emptyset$$

Kriterijska funkcija mjeri raspršenje točaka unutar grupa tako da zbraja kvadratna odstupanja od centroida. Kako se radi o čvrstom grupiranju, grupe su međusobno disjunktne, odnosno ne može se dogoditi da neka točka iz $\mathcal{D}$ završi u više grupa. Neka je $\boldsymbol{\mu}_k$ centroid grupe $\mathcal{C}_k$. Kriterijska funkcija za dano grupiranje kod algoritma K-sredina se definira izrazom:

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_k} d\,(\mathbf{x}, \boldsymbol{\mu}_k)^2$$

Ako podaci dolaze iz podskupa realnog koordinatnog prostora, onda možemo koristiti Euklidsku udaljenost i Euklidsku normu, odnosno $d\,(\mathbf{x}, \boldsymbol{\mu}_k) = \|\mathbf{x} - \boldsymbol{\mu}_k\|$. Tada kriterijska funkcija glasi:

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2$$

Kriterijska funkcija zbraja kvadratna odstupanja točaka[2] od centroida grupe u kojima se nalaze. Što su točke bliže svojim centroidima to će iznos kriterijske funkcije biti manji, i to predstavlja bolje grupiranje kod algoritma K-sredina. Algoritam K-sredina nastoji minimizirati iznos kriterijske funkcije, odnosno particionirati (grupirati) skup

[1] Vektorski prostor je skup objekata, odnosno vektora, nad kojim su definirane operacije međusobnog zbrajanja i množenja sa skalarom, a te operacije zadovoljavaju 8 aksioma vektorskog prostora.

[2] Zbog toga se u engleskoj literaturi J navodi pod imenom *Within-cluster sum of squares*, ili skraćeno WCSS. Osim takvog naziva, drugi engleski naziv je *inertia*.

$\mathcal{D}$ na način koji će dati minimalan iznos kriterijske funkcije. Analitičkim postupcima se ne može pronaći minimum kriterijske funkcije, stoga se optimizacija provodi iterativno, a iterativni postupak optimizacije nudi algoritam K-sredina.

3.1.3. Opis algoritma

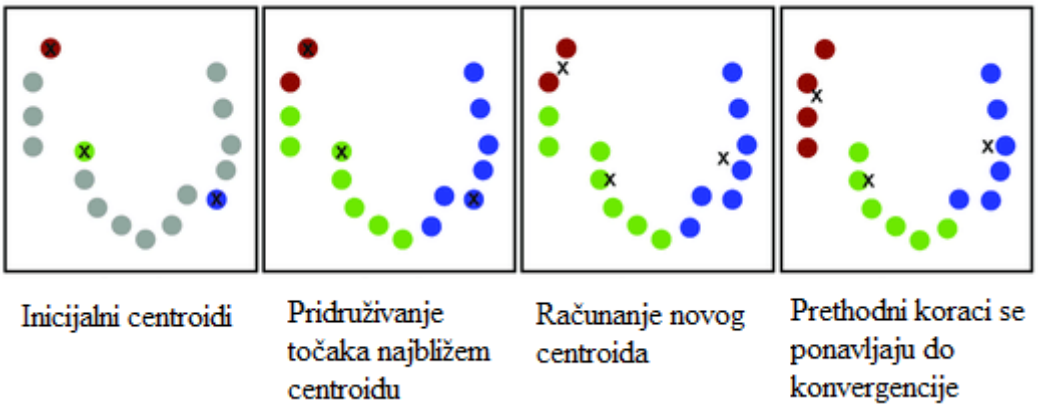
Najprije se inicijalizira početnih K centroida, te se u svakoj iteraciji sve točke iz $\mathcal{D}$ pridružuju najbližem centroidu. Zatim se centroidi svake grupe ponovno računaju na temelju točaka iz te grupe (pridruženi starom centroidu koji im je bio najbliži). Postupak se ponavlja do konvergencije, to jest dok dvije uzastopne iteracije nisu ništa promijenile u smislu da sve točke zadržavaju svoje grupe i ponovno računanje svih centroida ih ne mijenja.

Algoritam 1 Algoritam K-sredina

Parametri: broj grupa K
Ulaz: skup točaka $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$
Izlaz: čvrste disjunktne grupe $\mathcal{C}_k, \mathcal{D} = \bigcup_{k=1}^{K} \mathcal{C}_k$

inicijaliziraj centroide $\boldsymbol{\mu}_k,\ k \in \{1, \ldots, K\}$
ponavljaj
  $\mathcal{C}_k \leftarrow \emptyset,\ k \in \{1, \ldots, K\}$
  za svaki $\mathbf{x}^{(i)} \in \mathcal{D}$
    $k \leftarrow \operatorname{argmin}_{j \in \{1,\ldots,K\}} \left\|\mathbf{x}^{(i)} - \boldsymbol{\mu}_j\right\|$
    $\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \left\{\mathbf{x}^{(i)}\right\}$
  kraj za
  za svaki $k \in \{1, \ldots, K\}$
    $\boldsymbol{\mu}_k \leftarrow \frac{1}{|\mathcal{C}_k|} \sum_{\mathbf{x} \in \mathcal{C}_k} \mathbf{x}$
  kraj za
dok $\boldsymbol{\mu}_k$ ne konvergiraju

Na slici 3.1 je na jednostavnom primjeru prikazan princip rada algoritma sa zadanim brojem grupa $K = 3$. Križići predstavljaju centroide, a točke iste boje su pridružene istoj grupi.



Slika 3.1: Demonstracijski primjer grupiranja algoritmom K-sredina [4]

Potrebno je napomenuti da je algoritam kao početne centroide odabrao već postojeće točke, no to ne mora uvijek biti slučaj. Metode odabira početnih centroida su raspravljene u poglavlju 3.1.5.

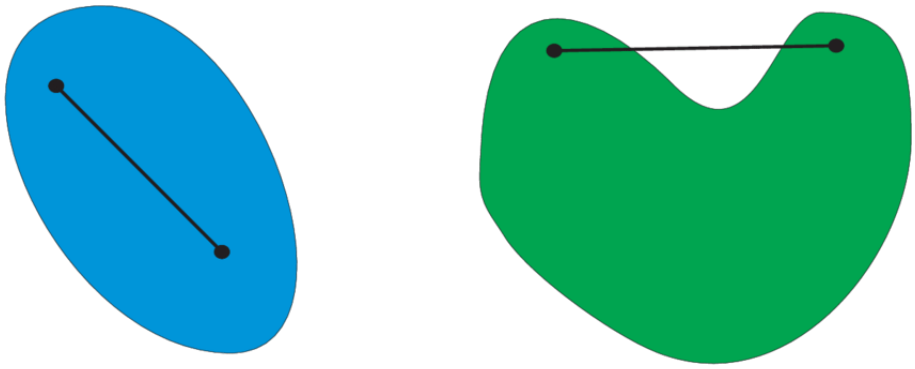
3.1.4. Svojstva i složenost algoritma

Kao što je spomenuto na početku poglavlja 3.1, cilj algoritma K-sredina jest minimizirati kriterijsku funkciju. Postavlja se pitanje, hoće li algoritam u tome uvijek i uspjeti? Odgovor je: neće. Algoritam pronalazi lokalni optimum, ali ne garantira da će to biti i globalni optimum. Velik utjecaj na konačni rezultat ima upravo prvi korak algoritma: inicijalizacija početnih K centroida. Kako bi se osigurao što bolji rezultat, odnosno što manji iznos kriterijske funkcije, jako je važno na pametan način odabrati početne centroide. Jednako tako je važno odabrati ispravan broj grupa: parametar K. Dodatno, algoritam se može pokrenuti više puta sa različitim izborima početnih centroida i tada se prati koji prolaz algoritma i koje grupiranje je dalo najmanji iznos kriterijske funkcije. Time se pomaže spriječiti problem "zaglavljivanja" u lokalnom optimumu.

Vremenska složenost algoritma K-sredina je $\mathcal{O}(TnKN)$, gdje je T broj iteracija algoritma do konvergencije, n broj značajki (dimenzija) točaka, a N broj točaka. Korak algoritma u kojem se točke pridružuju najbližem centroidu je složenosti $\mathcal{O}(nKN)$ jer je potrebno za svaku od N točaka ispitati koji od K trenutnih centroida je najbliži, dakle mora ispitati udaljenost od svakog centroida, a računanje udaljenosti je složenosti $\mathcal{O}(n)$. Korak algoritma u kojem se računaju centroidi je složenosti $\mathcal{O}(nN)$ jer iako se iterira po grupama, efektivno se iterira po svim ulaznim točkama i obavljaju se operacije zbrajanja i dijeljenja s nekim brojem (koje su također složenosti $\mathcal{O}(n)$).
Takva složenost je prihvatljiva i algoritam uz dobar odabir početnih centroida i dobar

odabir parametra K proizvodi dobre rezultate. Zbog povoljne složenosti, algoritam je učinkovit i za velike količine podatka (veliki N), a broj grupa i značajka ionako su najčešće[3] puno manji od N.

Što se tiče kvalitete grupiranja, postoje slučajevi u kojima algoritam neće ispravno razdvojiti grupe čak i uz odabir optimalnog K i kvalitetan odabir početnih centroida. Neizravna pretpostavka algoritma, zbog prirode računanja centroida i pridruživanja točaka najbližem centroidu, jest konveksan oblik grupe. Ukratko, skup u nekom prostoru je konveksan ako je segment između bilo koje dvije točke iz skupa također u skupu, u cjelosti. Na slici 3.2 su prikazana dva skupa, plavi je konveksan, a zeleni nije jer postoje dvije točke između kojih segment nije u cjelosti u skupu.



Slika 3.2: Konveksan i nekonvenksan skup [1]

Algoritam K-sredina neće dobro razdvojiti grupe koje nisu konveksnog oblika. Razlog tomu je mogućnost da se centroid takve grupe nalazi izvan oblika grupe pa ako se blizu tog centroida pojavljuje neka druga grupa, algoritam K-sredina ih neće dobro razdvojiti.
Postoji mogućnost da grupiranje neće biti kvalitetno čak i ako su sve prirodne grupe konveksne. Takvi slučajevi su primjerice bliske grupe nejednako velikih oblika ili grupe koje su "nakošenog" oblika. Demonstracija takvih neoptimalnih grupiranja dana je u poglavlju 5.

3.1.5. Odabir početnih K centroida

Način na koji se početni centroidi odabiru kod algoritma K-sredina nije jasno naznačen u algoritmu, a u poglavlju 3.1.4 je spomenuto da je to jako važno kako bi algoritam što

[3]Kod velikog broja značajki tradicionalni algoritmi grupiranja, ali i postupci obrade podataka u pravilu nisu učinkoviti. Tada se radi o problemu prokletstva visoke dimenzionalnosti (engl. *curse of dimensionality*)

više smanjio iznos kriterijske funkcije.

Postoji niz strategija za odabir početnih K centroida. Prva mogućnost jest kao početne centroide postaviti K slučajno izabranih točaka iz $\mathcal{D}$. Slučajan odabir nije baš najbolja ideja jer rezultati neće biti dobri ako se neki od početnih K centroida nalaze unutar istih prirodnih grupa.
Početni centroidi se mogu izabrati na način da samo prvi centroid bude slučajno odabrana točka iz $\mathcal{D}$, a daljnji centroidi se odabiru isto iz $\mathcal{D}$ (ali među onima koji nisu još odabrani kao centroidi) na način da su što udaljeniji od već odabranih centroida. Kao sljedeći uzastopni centroid odabire se točka koja maksimizira udaljenost do najbližeg, već odabranog centroida. Takva strategija odabira opisana je u nastavku.

Parametri: broj grupa K
Ulaz: skup točaka $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$
Izlaz: skup inicijalnih K centroida: $\mathcal{I} = \{\boldsymbol{\mu}_k\}_{k=1}^{K}$

$\mathcal{I} \leftarrow \emptyset$
$\widehat{\mathcal{D}} \leftarrow \mathcal{D}$
slučajno odaberi točku $\mathbf{x} \in \widehat{\mathcal{D}}$
$\mathcal{I} \leftarrow \mathcal{I} \cup \{\mathbf{x}\}$
$\widehat{\mathcal{D}} \leftarrow \widehat{\mathcal{D}} \setminus \{\mathbf{x}\}$
ponavljaj dokle god $|\mathcal{I}| < K$
  $\boldsymbol{\mu} \leftarrow \operatorname{argmax}_{\mathbf{x} \in \widehat{\mathcal{D}}} \left\| \mathbf{x} - \operatorname{argmin}_{\boldsymbol{\mu}' \in \mathcal{I}} \|\mathbf{x} - \boldsymbol{\mu}'\| \right\|$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{\mu}\}$
  $\widehat{\mathcal{D}} \leftarrow \widehat{\mathcal{D}} \setminus \{\boldsymbol{\mu}\}$
kraj ponavljaj
vrati $\mathcal{I}$

Ova strategija odabira početnih centroida je u pravilu dobra, no ima jedan nedostatak: odabirat će se stršeće vrijednosti, a kako su one relativno udaljene od prirodnih grupa, to bi moglo uzrokovati loše grupiranje. Postoji alternativa ovakvoj strategiji koja do neke mjere rješava problem odabira stršećih vrijednosti.

Umjesto odabira točke koja maksimizira udaljenost do najbližeg centroida, druga mogućnost je svakoj točki pridružiti vjerojatnost odabira koja je proporcionalna kvadratu udaljenosti do najbližeg centroida. Takva strategija odabira je poznata kao **K-means++**. Neka je u nekom trenutku već odabrano k centroida i one se nalaze u skupu

$\mathcal{I}$. Vjerojatnost odabira neke točke $\mathbf{x} \in \widehat{\mathcal{D}}$ kao sljedeći centroid glasi:

$$P\left(\boldsymbol{\mu}_{k+1} = \mathbf{x} \middle| \widehat{\mathcal{D}}, \mathcal{I}\right) = \frac{\min_{\boldsymbol{\mu}' \in \mathcal{I}} \|\mathbf{x} - \boldsymbol{\mu}'\|^2}{\sum_{\mathbf{x}' \in \widehat{\mathcal{D}}} \min_{\boldsymbol{\mu}' \in \mathcal{I}} \|\mathbf{x}' - \boldsymbol{\mu}'\|^2}$$

Nakon što se izračunaju vjerojatnosti za svaku od neodabranih točaka iz $\widehat{\mathcal{D}}$, točka se slučajno odabire prema navedenoj razdiobi vjerojatnosti. Stršeće vrijednosti i dalje imaju najveću vjerojatnost odabira, ali stršećih vrijednosti nema mnogo, pa je ipak vjerojatniji odabir neki od prirodno prosječnih primjera.
K-means++ u praksi daje bolje rezultate od varijante K-sredina u kojem se svi K početni centroidi odabiru slučajno. Iznosi kriterijskih funkcija ispadaju manji i algoritam prije konvergira, odnosno potreban je manji broj iteracija do stacionarnog stanja.

3.1.6. Odabir broja grupa K

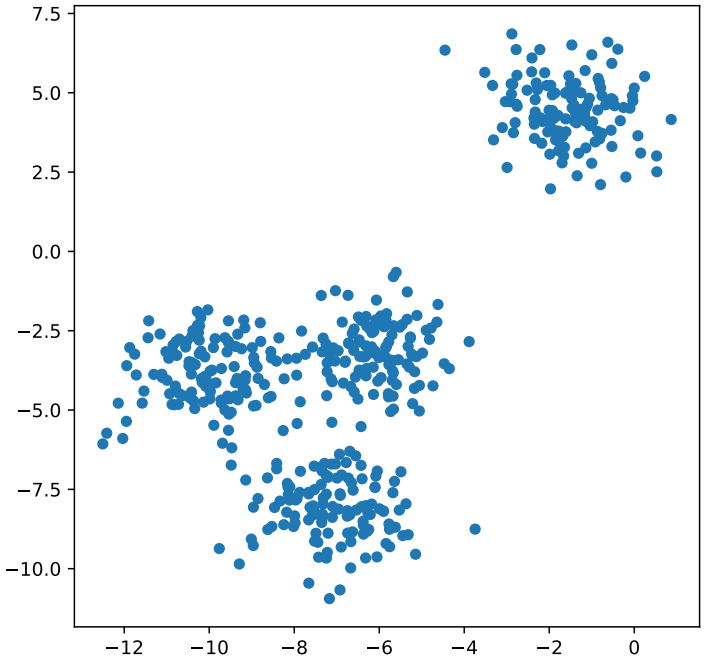
Kako bi se algoritam K-sredina mogao pokrenuti, potrebno je kao parametar K specificirati koliko grupa treba proizvesti. Idealno se postavlja K jednak broju prirodnih grupa, no to najčešće unaprijed nije poznato. Također postoji mogućnost da optimalan broj grupa K nije jednoznačno određen jer postoji više načina prirodnog grupiranja podataka.

Ovo poglavlje opisuje često korištenu metodu za određivanje optimalnog broja grupa kod algoritma K-sredina: **metodu koljena** (engl. *elbow method*). Još jedna metoda bit će opisana u poglavlju 4.1.2 nakon što se spomenu potrebni pojmovi.

Metoda koljena

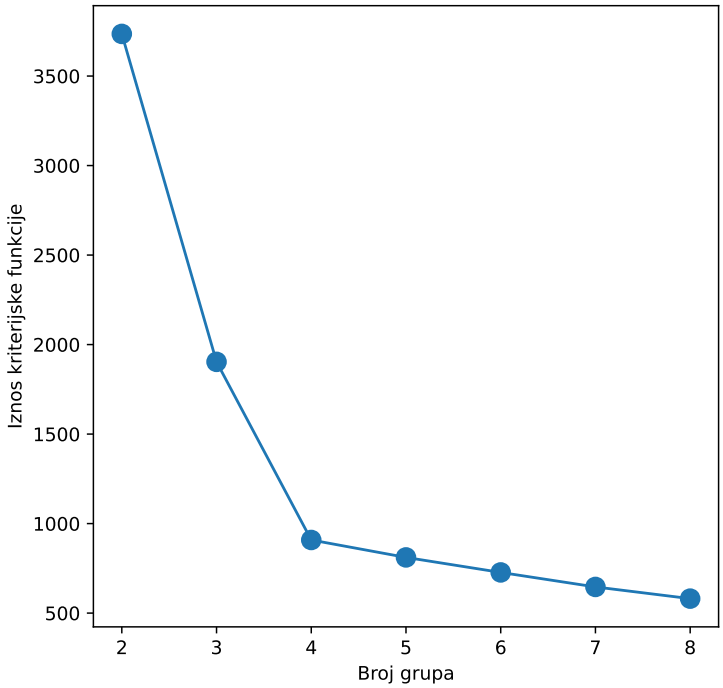
Metoda koljena je grafička metoda koja prati ovisnost kriterijske funkcije J (nakon što završi algoritam K-sredina) o broju grupa K. Jasno je da porast K smanjuje iznos J jer se grupe "smanjuju" povećanjem broja grupa u smislu da im se smanjuje raspršenost od centroida. U krajnjem slučaju, ukoliko se odabere broj grupa $K = N$, tada će svaka ulazna točka biti vlastita grupa i onda vrijedi $J = 0$.

Neka je na raspolaganju neki skup podataka s dvjema realnim značajkama. Grafički prikaz tih podataka u dvodimenzijskom koordinatnom sustavu prikazan je na slici 3.3.



Slika 3.3: Primjer skupa dvodimenzijskih podataka

Cilj je metodom koljena odrediti optimalan broj grupa za algoritam K-sredina. Neka su mogući kandidati $K \in \{2, \ldots, 8\}$. Za svaku od tih vrijednosti pokreće se algoritam K-sredina (uz strategiju K-means++ i Euklidovu udaljenost) nad ovim skupom podataka i računa konačan iznos kriterijske funkcije J. Rezultati su prikazani grafom na slici 3.4.



Slika 3.4: Ovisnost kriterijske funkcije o broju grupa

Porast broja grupa do 4 bilježi strmi pad iznosa kriterijske funkcije, a daljnim porastom se ne dobiva značajan pad iznosa kriterijske funkcije. Graf kod broja grupa 4 ima oblik koljena, otkud dolazi naziv metode koljena. Navedeno razmatranje je jaka naznaka da se radi o optimalnom broju grupa $K = 4$. Broj grupa manji od toga naglo povećava iznos kriterijske funkcije, a broj grupa veći od toga ne rezultira značajnim smanjenjima kriterijske funkcije. Zaista, vizualnim promatranjem na slici 3.3 jasno je da se u navedenom primjeru radi o 4 prirodne grupe.

3.2. Model Gaussove mješavine

Model Gaussove mješavine (engl. *Gaussian mixture model, GMM*) je najčešće korišten algoritam mekog probabilističkog grupiranja koji kao cilj ima svakoj točki iz $\mathcal{D}$ pridružiti vjerojatnosti da pripada različitim grupama: tzv. **odgovornost**. Model Gaussove mješavine pretpostavlja da su podaci "nastali" iz kombinacije (mješavine) K različitih vjerojatnosnih distribucija, odnosno gustoća. Svaka od K grupa je predstavljena svojom vjerojatnosnom gustoćom, a u slučaju Gaussovih mješavina, radi se o multivarijatnim n-dimenzijskim normalnim (Gaussovim) razdiobama. To znači da je

svaka grupa predstavljena svojom višedimenzijskom normalnom razdiobom: svojom lokacijom, kovarijacijskom matricom i vjerojatnosnom gustoćom.

3.2.1. Multivarijatna normalna razdioba

Slučajni n-dimenzijski stupčani vektor $\mathbf{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ ima multivarijatnu normalnu razdiobu, odnosno

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

s **lokacijom** $\boldsymbol{\mu} = (\mathbb{E}[X_1], \ldots, \mathbb{E}[X_n])^{\mathrm{T}}$ i **kovarijacijskom matricom** $\boldsymbol{\Sigma}$ kada svaka komponenta X_i ima univarijatnu normalnu razdiobu. Kovarijacijska matrica je simetrična, pozitivno semidefinitna matrica reda $n \times n$ čiji elementi predstavljaju kovarijancu između komponenata, odnosno

$$\boldsymbol{\Sigma}_{ij} = \operatorname{cov}[X_i, X_j] = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]$$

Funkcija gustoće takvog slučajnog vektora tada glasi:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \tag{3.1}$$

Izraz $(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ je kvadrat tzv. **Mahalanobisove udaljenosti** između vektora $\mathbf{x}$ i lokacije $\boldsymbol{\mu}$. To je višedimenzijska generalizacija ideje udaljenosti točke od srednje vrijednosti u broju standardnih devijacija kod jednodimenzijskih razdioba.

3.2.2. Model miješane gustoće

Model miješane gustoće, pa tako i model Gaussove mješavine, je **generativan model** podataka koji svaki primjer generira, odnosno uzorkuje, iz miješane gustoće: linearne kombinacije K gustoća svake grupe, odnosno K komponenata. Svaki podatak u tom modelu se uzorkuje na sljedeći način: prvo se prema kategoričkoj distribuciji grupa, gdje se grupe sada prikazuju apriorno nepoznatim oznakama y (jer se radi o nenadziranom strojnom učenju), odabere grupa k. Neka je $P(y = k)$ vjerojatnost da je odabrana grupa k. Nakon što se odabere grupa, primjer $\mathbf{x}$ se onda uzorkuje prema vjerojatnosnoj gustoći te grupe koju označavamo sa $p(\mathbf{x}|y = k)$, odnosno ako se označe parametri distribucije (gustoće) grupe k sa $\boldsymbol{\theta}_k$, onda se ta vjerojatnost označava sa $p(\mathbf{x}|\boldsymbol{\theta}_k)$. Apriorna gustoća $p(\mathbf{x})$ iz koje su uzorkovani ulazni primjeri iskaže se formulom potpune vjerojatnosti:

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(y = k)\, p(\mathbf{x}|y = k) = \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}|\boldsymbol{\theta}_k) \tag{3.2}$$

Uvedena je oznaka $\pi_k = P(y = k)$, a ta vjerojatnost se naziva **koeficijent mješavine** (engl. *mixture weight*). Jasno je da mora vrijediti

$$\sum_{k=1}^{K} \pi_k = 1 \tag{3.3}$$

U slučaju modela Gaussove mješavine, gustoća grupe $p(\mathbf{x}|\boldsymbol{\theta}_k)$ se računa prema 3.1, gdje su parametri grupe $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$.
Cilj mekog grupiranja kod ovakvog modela jest odrediti vjerojatnost da primjer $\mathbf{x}$ pripada nekoj grupi k, što se označava kao $P(y = k|\mathbf{x})$ i naziva se odgovornost. Za svaki podatak $\mathbf{x}^{(i)}$ iz skupa od N ulaznih primjera $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ računa se odgovornost: vjerojatnost da je $\mathbf{x}^{(i)}$ uzorkovan iz grupe k. Ta vjerojatnost se može izraziti Bayesovom formulom:

$$P\left(y = k\middle|\mathbf{x}^{(i)}\right) = \frac{P(y = k)\, p\left(\mathbf{x}^{(i)}\middle|y = k\right)}{p\left(\mathbf{x}^{(i)}\right)} = \frac{\pi_k\, p\left(\mathbf{x}^{(i)}\middle|\boldsymbol{\theta}_k\right)}{\sum_{j=1}^{K} \pi_j\, p\left(\mathbf{x}^{(i)}|\boldsymbol{\theta}_j\right)} \tag{3.4}$$

Kako bi se mogla računati ovakva vjerojatnost, potrebno je poznavati parametre modela miješane gustoće. To znači da je za svaku od K komponenata potrebno poznavati koeficijent mješavine i parametre gustoće svake komponente. U slučaju Gaussove mješavine, parametri svake komponente su lokacije i kovarijacijske matrice svake komponente. Neka su parametri modela predstavljeni oznakom

$$\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\theta}_k\}_{k=1}^{K} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Sljedeće poglavlje bavi se pitanjem na koji način i kojim kriterijem se određuju parametri modela $\boldsymbol{\theta}$.

3.2.3. Metoda najveće izglednosti

Parametri modela $\boldsymbol{\theta}$ se procjenjuju **metodom najveće izglednosti** (engl. *maximum likelihood estimation, MLE*). Potrebno je definirati funkciju izglednosti parametara $\boldsymbol{\theta}$ na uzorku $\mathcal{D}$. Pod pretpostavkom da su primjeri iz uzorka nezavisni, tada se funkcija izglednosti može definirati kao umnožak gustoća svakog primjera iz uzorka:

$$L(\boldsymbol{\theta}; \mathcal{D}) = \prod_{i=1}^{N} p\left(\mathbf{x}^{(i)}\right) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k\, p\left(\mathbf{x}^{(i)}\middle|\boldsymbol{\theta}_k\right)$$

Iz definicije i iz 3.2 je jasno da funkcija izglednosti ovisi o parametrima $\boldsymbol{\theta}$.

Cilj metode najveće izglednosti jest procijeniti parametre $\boldsymbol{\theta}$, koji će uz uzorak $\mathcal{D}$ maksimizirati funkciju izglednosti. Drugim riječima, potrebno je pronaći parametre

koje od svih mogućih parametara (tzv. prostor parametara $\boldsymbol{\Theta}$) maksimiziraju vjerojatnost pojavljivanja podataka iz uzorka $\mathcal{D}$. Formalno, traži se

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \, L\left(\boldsymbol{\theta}; \mathcal{D}\right)$$

Vrlo se često u praksi ne maksimizira funkcija izglednosti, nego **log-izglednosti**:

$$\ell\left(\boldsymbol{\theta}; \mathcal{D}\right) = \ln L\left(\boldsymbol{\theta}; \mathcal{D}\right) = \ln \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\theta}_k\right) = \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\theta}_k\right) \quad (3.5)$$

Razlog tomu je taj što se, primjerice, u ovom slučaju radi s produktima, a lakše je raditi sa sumama, a logaritmiranje produkte pretvara u sume. Zbog toga što je prirodni logaritam monotono rastuća funkcija, kad god se postiže maksimum funkcije izglednosti, tada i log-izglednost postiže svoj maksimum:

$$\widehat{\boldsymbol{\theta}}_{\text{mle}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \, L\left(\boldsymbol{\theta}; \mathcal{D}\right) = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \, \ell\left(\boldsymbol{\theta}; \mathcal{D}\right) = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\theta}_k\right)$$

Ovakav optimizacijski problem nema rješejne u zatvorenoj formi, stoga su potrebne iterativne optimizacijske metode koje će pokušati maksimizirati log-izglednost. Jedna od takvih metoda je **algoritam maksimizacije očekivanja**.

3.2.4. Algoritam maksimizacije očekivanja

Algoritam maksimizacije očekivanja (engl. *expectation-maximization algorithm, EM*) osniva se na proširenju generativnog modela opisanog u 3.2 sa tzv. **skrivenim (latentnim) varijablama** koje modeliraju vrijednosti koje se ne opažaju u podacima. Takvo proširenje modela u konačnici omogućuje pronalazak iterativnog postupka koji postupno povećava log-izglednost. Proširenje modela i matematičke formulacije iza njega nisu navedeni, nego je iskazan samo konačni algoritam.

EM algoritam ima vrlo sličnu strukturu kao i algoritam K-sredina. Najprije se na neki način inicijaliziraju parametri modela Gaussove mješavine $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ kao što je algoritam K-sredina inicijalizirao početnih K centroida. Broj komponenata Gaussove mješavine K je, kao i kod algoritma K-sredina, parametar algoritma (razlikovati od parametara modela) i on u smislu algoritma mora biti unaprijed poznat. Nakon što su inicijalizirani parametri modela, iterativni postupak počinje. Najprije se za sve ulazne primjere računa odgovornost za svaku komponentu, odnosno grupu, prema formuli 3.4. Taj korak kod EM algoritma se naziva **E-korak**. U usporedbi s algoritmom K-sredina, ovo je analogno pridruživanju točaka najbližim centroidima, što

je bilo "čvrsto" pridruživanje, a sada se radi o probabilističkom mekom pridruživanju. Nakon što se izračunaju odgovornosti, parametri modela se iznova računaju na temelju novih izračunatih odgovornosti, što je analogno ponovnom računanju centroida kod algoritma K-sredina. Taj korak se naziva **M-korak**. Koraci E i M se ponavljaju do konvergencije parametara ili log-izglednosti. Kao oznaka odgovornosti korištena je oznaka

$$h_k^{(i)} = P\left(y = k \middle| \mathbf{x}^{(i)}\right)$$

Algoritam 2 Algoritam maksimizacije očekivanja nad GMM

Parametri: broj grupa, odnosno komponenata K
Ulaz: skup ulaznih primjera $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$
Izlaz: parametri modela $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$

inicijaliziraj parametre modela $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$
ponavljaj
  E-korak:
  za svaki $\mathbf{x}^{(i)} \in \mathcal{D}$ i $k \in \{1, \ldots, K\}$
    $h_k^{(i)} \leftarrow \frac{\pi_k \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$
  kraj za

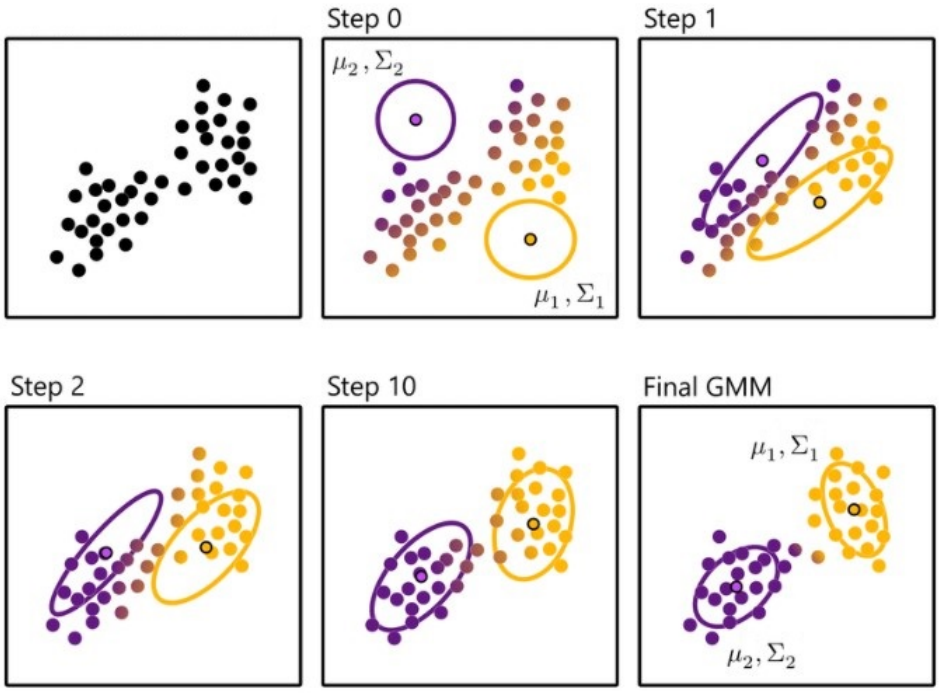
  M-korak:
  za svaki $k \in \{1, \ldots, K\}$
    $\boldsymbol{\mu}_k \leftarrow \frac{\sum_{i=1}^{N} h_k^{(i)} \mathbf{x}^{(i)}}{\sum_{i=1}^{N} h_k^{(i)}}$
    $\boldsymbol{\Sigma}_k \leftarrow \frac{\sum_{i=1}^{N} h_k^{(i)} \left(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\right)\left(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\right)^{\mathrm{T}}}{\sum_{i=1}^{N} h_k^{(i)}}$
    $\pi_k \leftarrow \frac{1}{N} \sum_{i=1}^{N} h_k^{(i)}$
  kraj za

  $\ell\left(\boldsymbol{\theta}; \mathcal{D}\right) \leftarrow \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$
dok ne konvergiraju parametri $\boldsymbol{\theta}$ ili log-izglednost $\ell\left(\boldsymbol{\theta}; \mathcal{D}\right)$
vrati $\boldsymbol{\theta}$

Iz "natreniranih" parametara modela se lako računaju odgovornosti prema formuli 3.4, što je rezultat mekog grupiranja:

$$P\left(y = k \middle| \mathbf{x}^{(i)}\right) = h_k^{(i)} = \frac{\pi_k \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \, p\left(\mathbf{x}^{(i)} \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

Na slici 3.5 prikazan je tijek rada algoritma nad dvodimenzijskim primjerima sa zadanim brojem komponenata $K = 2$.



Slika 3.5: Tijek rada algoritma maksimizacije očekivanja nad modelom Gaussovih mješavina [3]

Nijanse boja točaka označavaju u kojoj mjeri točka pripada pojedinim grupama. Nakon svake iteracije se lokacije (zamišljene točke s crnim obrubom) i kovarijacijske matrice (vizualizirane elipsama, sve točke na elipsi imaju istu Mahalanobisovu udaljenost od lokacije) mijenjaju dok se ne postigne stacionarno stanje: konvergencija log-izglednosti ili parametara modela.

3.2.5. Svojstva i složenost algoritma

Algoritam maksimizacije očekivanja nastoji maksimizirati log-izglednost opisanu formulom 3.5, ali u tome neće uvijek uspjeti. Pronaći će lokalni maksimum log-izglednosti, ali to ne mora biti globalni maksimum. Rezultat uvelike ovisi o kvalitetnoj inicijalizaciji parametara $\boldsymbol{\theta}$. Slučajna inicijalizacija je moguća, ali najčešće se parametri inicijaliziraju tako da se najprije nad skupom primjera provede algoritam K-sredina koji će kao rezultat dati K čvrstih grupa. Lokacije $\boldsymbol{\mu}_k$ svake komponente inicijaliziraju se centroidima dobivenih grupa, a koeficijenti mješavine π_k računaju se kao udjeli broja primjera u grupi k u odnosu na ukupan broj primjera N. Kovarijacijske matrice $\boldsymbol{\Sigma}_k$

mogu se inicijalizirati procjenama kovarijanci iz uzorka: primjera iz grupe k.

Što se tiče vremenske složenosti, algoritam je učinkovit jer je linearan s brojem primjera i brojem komponenata. Računanje Gaussove gustoće, uz prethodno izračunatu determinantu i inverz kovarijacijske matrice[4], je $\mathcal{O}(n^2)$. Prema tome, E-korak je vremenske složenosti $\mathcal{O}(NKn^2)$ i prostorne složenosti $\mathcal{O}(NK)$. Što se tiče M-koraka, za svaku komponentu, računanje nove lokacije je $\mathcal{O}(Nn)$, računanje nove kovarijacijske matrice je $\mathcal{O}(Nn^2)$, a računanje novog koeficijenta mješavine je $\mathcal{O}(N)$. U opisu algoritma nije eksplicitno navedeno, ali potrebno je i računati inverz i determinantu nove kovarijacijske matrice jer su potrebni za kasnije računanje gustoće, što je u općenitom slučaju $\mathcal{O}(n^3)$. Dakle, M-korak je složenosti $\mathcal{O}(K(Nn^2 + n^3))$. Računanje log-izglednosti je vremenske složenosti $\mathcal{O}(NKn^2)$. Najzahtjevniji od svih tih koraka je M-korak, stoga je uz T iteracija vremenska složenost algoritma $\mathcal{O}(TK(Nn^2 + n^3))$.

Složenost se može poboljšati ako se uvedu ograničenja na oblik kovarijacijske matrice, što pojednostavljuje model Gaussove mješavine i operacije poput matričnog množenja, invertiranja i računanje determinante postaju jednostavnije. Tako primjerice kovarijacijske matrice mogu biti

1. **Izotropne**: svaka komponenta ima kovarijacijsku matricu oblika $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$, gdje je $\mathbf{I}$ jedinična matrica, a σ_k^2 dijeljena (između značajki) varijanca svake komponente. Značajke svake komponente su tada nekorelirane, imaju jednaku varijancu i grupe su sferičnog oblika.

2. **Dijeljene**: svaka komponenta ima istu kovarijacijsku matricu.

3. **Dijagonalne**: značajke komponente su nekorelirane.

Algoritam je učinkovit i na velikim skupovima primjera zbog toga što je složenost u svakom slučaju linearan s brojem primjera.

Zbog prirode generativnog modela podataka, kao i kod algoritma K-sredina, postoji neizravna pretpostavka o konveksnom obliku grupa. EM algoritam najčešće neće moći kvalitetno razdvojiti dvije grupe koje su relativno blizu, a barem jedna od njih ima nekonveksan oblik. Za razliku od algoritma K-sredina, EM algoritam će ispravno razdvojiti konveksne nejednako velike ili ukošene grupe jer se svi takvi oblici mogu ispravno modelirati različitim oblicima kovarijacijske matrice. Kako svaka komponenta Gaussove mješavine predstavlja razdiobu normalnog slučajnog vektora, grupiranje ovim algoritmom moguće je samo ako su sve značajke realni brojevi, dakle model i algoritam nisu primjenjivi nad podacima koji nisu realni brojevi.

[4] tzv. matrica preciznosti ili samo preciznost.

3.2.6. Odabir broja komponenata K

Odabir optimalnog broja komponenata nužan je uvjet kvalitetnog grupiranja, isto kao što je to bio slučaj kod algoritma K-sredina. Zbog toga što je model Gaussove mješavine statistički model, prikladno je koristiti **Akaikeov informacijski kriterij** (engl. *Akaike information criterion, AIC*) za određivanje optimalnog broja komponenata. Akaikeov informacijski kriterij se računa izrazom

$$AIC = 2m - 2\ln L$$

gdje m predstavlja broj parametara modela, a L maksimiziranu izglednost. Dobra procjena optimalnog broja komponenata K kod modela Gaussovih mješavina je ona vrijednost K koja minimizira Akaikeov informacijski kriterij. Akaikeov informacijski kriterij pada povećanjem izglednosti, a raste povećanjem broja parametara.
U slučaju Gaussovih mješavina, parametri modela su lokacije, kovarijacijske matrice i koeficijenti mješavine. Svaka se lokacija sastoji od n parametara s obzirom da se radi o n-dimenzijskom vektoru, dakle za K lokacija radi se o Kn mnogo parametara. Koeficijenata mješavine ima K jer toliko ima komponenata, ali uvijek se zadnji može izračunati iz ostalih $K-1$ jer vrijedi jednakost 3.3, prema tome radi se o $K-1$ mnogo parametara. Što se tiče kovarijacijskih matrica, broj parametara ovisi o tome postoje li ograničenja na oblik matrice i ako postoje, o kojim ograničenjima se radi. U općenitom slučaju bez ograničenja, svaka kovarijacijska matrica sadrži n dijagonalnih elemenata i $\binom{n}{2}$ nedijagonalnih elemenata, što za svaku kovarijacijsku matricu daje ukupno $\frac{n(n+1)}{2}$ parametara, a za njih K mnogo to je ukupno $\frac{Kn(n+1)}{2}$ parametara. Ovaj broj je manji ako se uvedu ograničenja na kovarijacijsku matricu. Primjerice, ako su kovarijacijske matrice dijagonalne, tada se broje samo dijagonalni elementi, što daje broj parametara Kn. Iz navedenih razmatranja za model Gaussove mješavine vrijedi, u općenitom slučaju

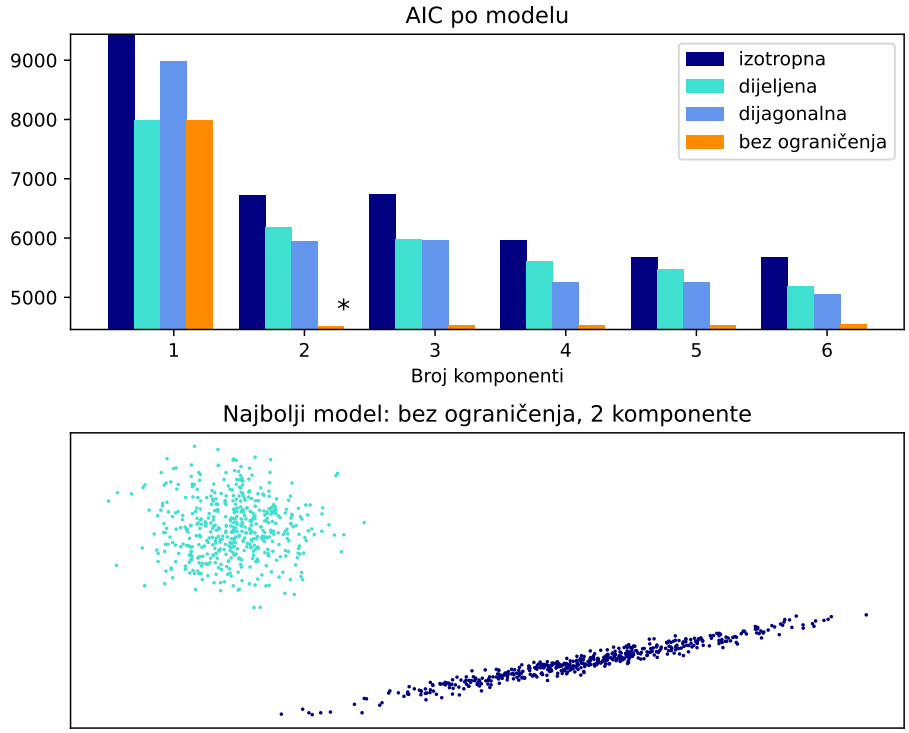
$$m = \frac{Kn\,(n+1)}{2} + Kn + K - 1$$

U svakom slučaju radi se o direktnoj ovisnosti o broju komponenata. Izglednost L također u konačnici ovisi o broju komponenata uz fiksan skup podataka. U tu svrhu ima smisla označiti broj parametara sa $m\,(K)$, a izglednost sa $L\,(K)$. Optimalan broj komponenata neka je K^* i koristeći Akaikeov informacijski kriterij on se procjenjuje kao

$$K^* = \underset{K\in\mathbb{N}}{\operatorname{argmin}}\,(2\,m\,(K) - 2\ln L\,(K))$$

Određivanje optimalnog modela

Osim variranja broja komponenata, mogu se varirati i ograničenja na kovarijacijske matrice modela Gaussove mješavine. Na slici 3.6 je prikazan primjer skupa dvodimenzijskih podataka i želi se pronaći najbolji model Gaussove mješavine s obzirom na broj komponenata i ograničenja kovarijacijske matrice.



Slika 3.6: Primjer odabira optimalnog modela Gaussovih mješavinae

Za svaki broj komponenata pokrenut je EM algoritam sa navedenim ograničenjima i zabilježena vrijednost Akaikeovog informacijskog kriterija na gornjem dijagramu. Iz dijagrama je vidljivo da za sve brojeve komponenata AIC ispada najmanji za modele bez ograničenja. Uz to, iznos AIC ispada najmanji za 2 komponente, stoga je takav model najbolji izbor. Zaista, promatranjem podataka vidljivo je da se radi o mješavini dvaju komponenata i općenitim kovarijacijskim matricama bez ograničenja.

3.3. Hijerarhijsko aglomerativno grupiranje

U poglavlju 2.2 navedeno je na koji način hijerarhijsko grupiranje gradi grupe te su spomenuta dva pristupa hijerarhijskog grupiranja: aglomerativno i divizivno. Ovo po-

glavlje bavi se standardnim algoritmom **hijerarhijskog aglomerativnog grupiranja** (engl. *hierarchical agglomerative clustering, HAC*). Također je u poglavlju 2.2 dan primjer jednog hijerarhijskog aglomerativnog grupiranja na slici 2.1. U tom primjeru prikazano je kako su se grupe međusobno spajale. Odluka koje dvije grupe spojiti nije slučajna, stoga je prije formalnog opisa algoritma potrebno prvo definirati pojmove vezane za spajanje grupa.

3.3.1. Spajanje grupa

Odluka koje dvije grupe spojiti osniva se na **kriteriju spajanja** (engl. *linkage criterion*). Kriterij spajanja mjeri koliko su dvije grupe slične, odnosno različite. On se osniva na nekoj od mjera sličnosti između dviju točaka, bila to udaljenost, sličnost ili različitost, i one su opisane u poglavlju 2.2. Najčešće se u svrhu definiranja kriterija spajanja koriste mjere udaljenosti, no mogu se koristiti i mjere sličnosti ili različitosti.

Kriterij spajanja u ovom radu nije detaljno opisan kao mjera udaljenosti, samo na površnoj razini. Neka su su $\mathcal{C}_i, \mathcal{C}_j \subset \mathcal{D}$ neprazni i disjunktni podskupovi skupa ulaznih točaka $\mathcal{D}$. Neka oznaka $D(\mathcal{C}_i, \mathcal{C}_j)$ predstavlja kriterij spajanja (bitno je ovu oznaku razlikovati od oznake udaljenosti između dvije točke, npr. $d(\mathbf{x}, \mathbf{y})$) i on mjeri koliko su grupe $\mathcal{C}_i$ i $\mathcal{C}_j$ udaljene. Postoji mnogo kriterija spajanja. U ovom se poglavlju navode one koje su implementirane u programskoj knjižnici Scikit-learn.

Jednostruka povezanost (engl. *single-linkage clustering*) udaljenost između grupa definira kao minimalnu udaljenost između točaka u tim grupama, gdje je jedna točka iz jedne, a druga točka iz druge grupe:

$$D_{\text{min}}(\mathcal{C}_i, \mathcal{C}_j) = \min_{(\mathbf{x},\mathbf{y}) \in \mathcal{C}_i \times \mathcal{C}_j} d(\mathbf{x}, \mathbf{y})$$

Potpuna povezanost (engl. *complete-linkage clustering*) udaljenost između grupa definira slično kao i jednostruka povezanost, ali ovaj se put gleda maksimalna udaljenost:

$$D_{\text{max}}(\mathcal{C}_i, \mathcal{C}_j) = \max_{(\mathbf{x},\mathbf{y}) \in \mathcal{C}_i \times \mathcal{C}_j} d(\mathbf{x}, \mathbf{y})$$

Prosječna povezanost (engl. *average-linkage clustering*) udaljenost između grupa definira kao prosječnu udaljenost svih parova grupa, opet gdje je jedna točka iz jedne, a druga točka iz druge grupe:

$$D_{\text{avg}}(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{|\mathcal{C}_i| \cdot |\mathcal{C}_j|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{C}_i \times \mathcal{C}_j} d(\mathbf{x}, \mathbf{y})$$

Wardova metoda, također poznata kao Wardova metoda minimalne varijance, prati zbroj kvadratnih pogrešaka (između točaka i centroida) u grupama prije i nakon spajanja. U osnovi Wardova metoda pretpostavlja da točke dolaze iz realnog koordinatnog

prostora, no postoje generalizacije na bilo kakve podatke koje koriste mjeru sličnosti. Prema tome, kao mjera pogreške koristi se Euklidova udaljenost između točke i centroida, odnosno koristi se Euklidova norma. Neka je $\mathrm{ESS}(\mathcal{C})$ zbroj kvadratnih pogreški (engl. *error sum of squares*) neke neprazne grupe $\mathcal{C}$ čiji je centroid $\boldsymbol{\mu}$:

$$\mathrm{ESS}(\mathcal{C}) = \sum_{\mathbf{x} \in \mathcal{C}} \|\mathbf{x} - \boldsymbol{\mu}\|^2$$

Wardova metoda udaljenost između grupa definira kao razliku sume kvadratnih pogrešaka prije i nakon spajanja grupa:

$$D_{\text{Ward}}(\mathcal{C}_i, \mathcal{C}_j) = \mathrm{ESS}(\mathcal{C}_i \cup \mathcal{C}_j) - (\mathrm{ESS}(\mathcal{C}_i) + \mathrm{ESS}(\mathcal{C}_j))$$

Gornja formula može se raspisivanjem svesti na jednostavniju formulu:

$$D_{\text{Ward}}(\mathcal{C}_i, \mathcal{C}_j) = \frac{|\mathcal{C}_i| \cdot |\mathcal{C}_j|}{|\mathcal{C}_i| + |\mathcal{C}_j|} \left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|^2$$

gdje su $\boldsymbol{\mu}_i$ i $\boldsymbol{\mu}_j$ redom centroidi grupa $\mathcal{C}_i$ i $\mathcal{C}_j$.

3.3.2. Opis algoritma

Hijerarhijsko aglomerativno grupiranje najprije stvara N grupa, odnosno na početku tretira svaku ulaznu točku iz $\mathcal{D}$ kao zasebnu grupu. Prilikom svake iteracije donosi odluku koje dvije grupe spojiti na temelju kriterija spajanja: spojiti one dvije grupe koje su najbliže prema kriteriju spajanja koji je unaprijed odabran kao parametar algoritma. Nakon spajanja grupa, broj ukupnih grupa smanjuje se za 1 i postupak se ponavlja.
Naravno, algoritam može ponavljati postupak dok ne ostane samo jedna grupa, ali to najčešće nije od koristi. Umjesto toga, postupak se može zaustaviti[5] kada se ispuni neki kriterij zaustavljanja, a to može biti jedna od sljedećih opcija:

- dostignut je željeni broj grupa;
- najmanja udaljenost između dvije grupe premašila je neku granicu.

Odluka kada će postupak završiti je također parametar algoritma. Ukoliko se želi zaustaviti postupak nakon dostignutih K grupa, broj K potrebno je predati kao parametar, a ako se ne želi premašiti najmanja udaljenost ϵ između grupa tijekom spajanja, tada je potrebno specificirati broj ϵ.

[5]Tada se na neki način radi o particijskom grupiranju jer kada se zaustavi postupak, onda se presiječe horizontalno dendrogram i prihvate grupe kakve jesu na toj razini, bez hijerarhijske strukture.

Algoritam 3 Hijerarhijsko aglomerativno grupiranje

Parametri: kriterij spajanja $D(\mathcal{C}_i, \mathcal{C}_j)$, uvjet zaustavljanja: broj grupa K ili najveća dopuštena minimalna udaljenost između grupa ϵ

Ulaz: skup točaka $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$

Izlaz: grupiranje: skup međusobno disjunktnih grupa $\Gamma = \{\mathcal{C}_i\}_{i=1}^{M}$ koji sačinjavaju skup točaka $\mathcal{D} = \bigcup_{i=1}^{M} \mathcal{C}_i$, gdje broj grupa M je ili nepoznat ili $M = K$ ovisno o uvjetu zaustavljanja

$\Gamma \leftarrow \emptyset$

za svaki $i \in \{1, \ldots, N\}$

  $\mathcal{C}_i \leftarrow \left\{\mathbf{x}^{(i)}\right\}$

  $\Gamma \leftarrow \Gamma \cup \{\mathcal{C}_i\}$

kraj za

ponavljaj

  prekini ako je zadan K i vrijedi $|\Gamma| \leq K$

  $(\mathcal{C}_i, \mathcal{C}_j) \leftarrow \operatorname{argmin}_{(\mathcal{C}_a, \mathcal{C}_b) \in \Gamma \times \Gamma} D(\mathcal{C}_a, \mathcal{C}_b)$

  prekini ako je zadan ϵ i vrijedi $D(\mathcal{C}_i, \mathcal{C}_j) > \epsilon$

  $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \mathcal{C}_j$

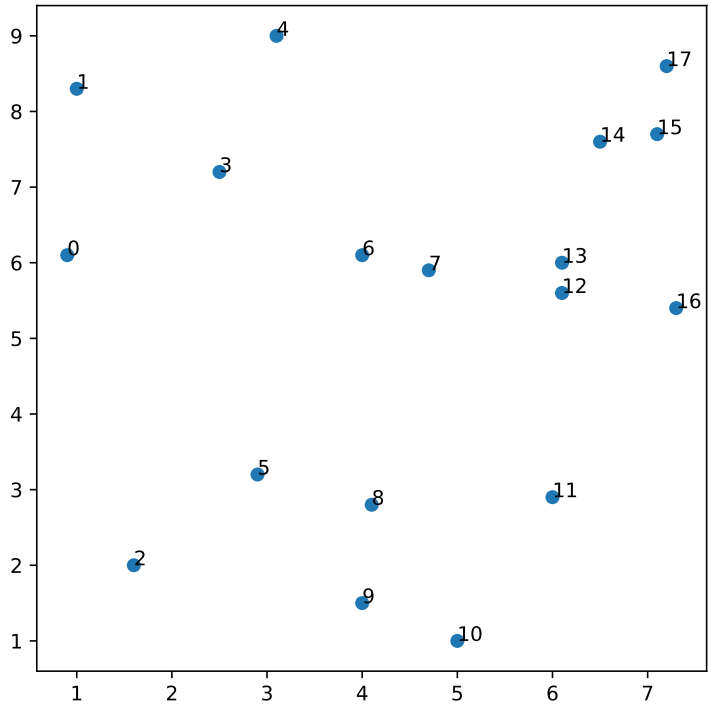
  $\Gamma \leftarrow \Gamma \setminus \{\mathcal{C}_j\}$

kraj ponavljaj

vrati Γ

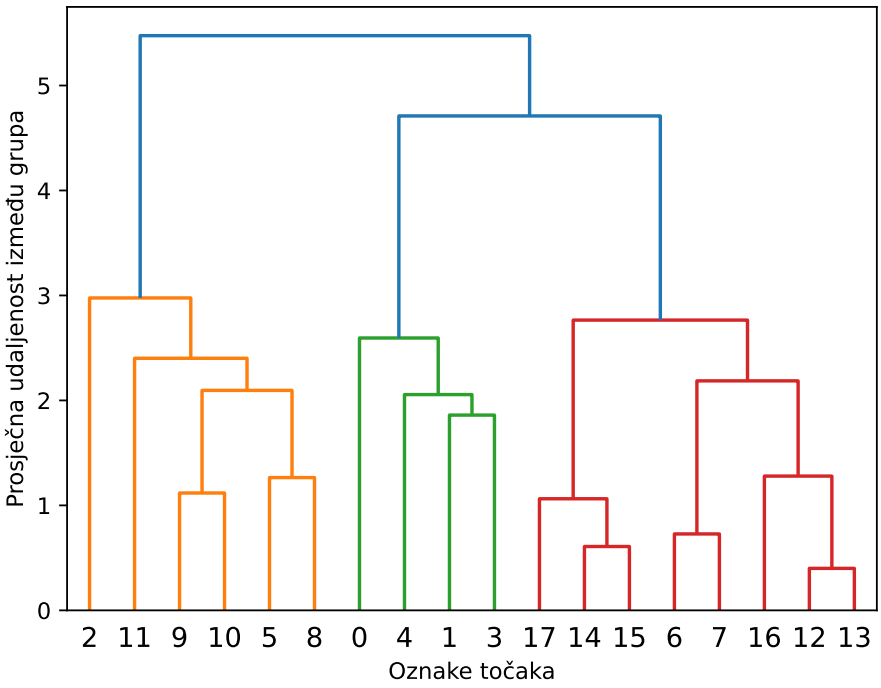
3.3.3. Primjer grupiranja

Neka su ulazni podaci točke s dvjema realnim značajkama prikazani na slici 3.7. Radi lakšeg snalaženja, sve su točke označene brojevima.



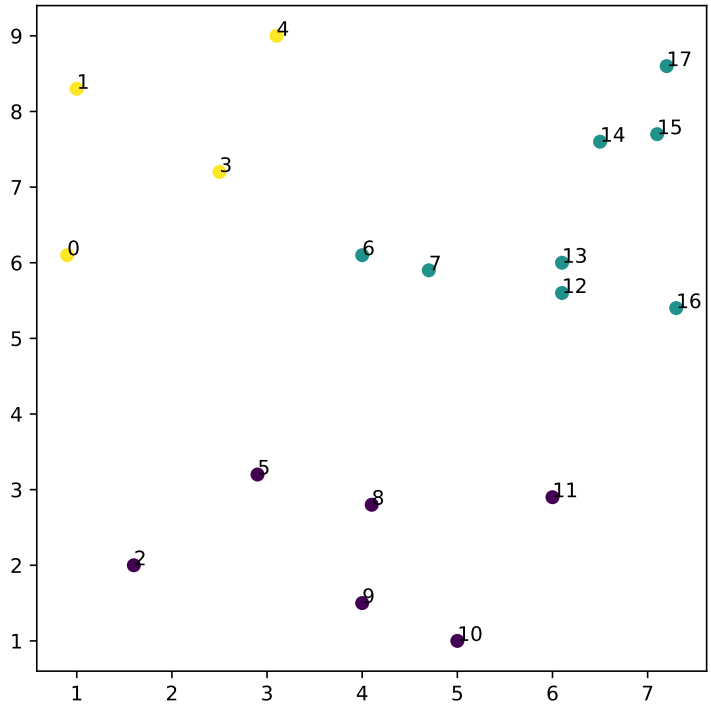
Slika 3.7: Ulazni podaci

Neka kriterij spajanja bude prosječna povezanost i neka algoritam spoji sve podgrupe, odnosno kao uvjet zaustavljanja zadan je broj grupa $K = 1$. Nad tim podacima pokrenut je algoritam hijerarhijskog aglomerativnog grupiranja. Rezultat grupiranja prikazan je dendrogramom na slici 3.8.



Slika 3.8: Dendrogram hijerarhijskog aglomerativnog grupiranja

Na donjoj liniji nalaze se oznake svake točke, a na lijevoj liniji nalazi se mjera koja iskazuje koja je bila prosječna udaljenost (jer koristimo prosječnu povezanost kao kriterij spajanja) između grupa u trenutku spajanja. Ukoliko se želi particijski grupirati primjere, to se odlučuje prema vlastitoj volji. Primjerice, ukoliko se ne žele spajati grupe čija prosječna udaljenost prelazi $\epsilon = 3.5$, tada se dendrogram presiječe horizontalno na toj razini i kao rezultat dobiju 3 grupe jer bi se dendrogram presjekao na 3 mjesta. Na sličan se način mogu i po volji grupirati točke u proizvoljan broj grupa K: dendrogram se presiječe na razini na kojoj se nalazi K grupa. Na slici 3.9 prikazan je rezultat grupiranja ako se specificira $\epsilon = 3.5$.



Slika 3.9: Hijerarhijsko aglomerativno grupiranje uz $\epsilon = 3.5$

3.3.4. Svojstva i složenost algoritma

Opisani algoritam daje dobre rezultate, no najveći nedostatak je vremenska i prostorna složenost. Neka se algoritam zaustavlja nakon što je preostalo K grupa. To znači da algoritam u glavnoj petlji radi $N - K$ koraka. Korak u kojem se odlučuje koje dvije grupe spojiti mora proći kroz sve kombinacije grupa, što je vremenske složenosti $\mathcal{O}(N^2)$. Prema tome, vremenska složenost algoritma je $\mathcal{O}(N^2(N-K))$, a kako je u praksi N puno veći od K, onda je vremenska složenost $\mathcal{O}(N^3)$. Iz tog razloga algoritam nije dobar izbor za velike skupove podataka. Informacija o udaljenosti između točaka jako se često koristi tijekom izvršenja algoritma, stoga implementacije u pravilu najprije konstruiraju tzv. **matricu udaljenosti** (engl. *distance matrix*). Matrica udaljenosti simetrična je matrica dimenzija $N \times N$ i sadrži izračunatu udaljenost između svih parova točaka. Računanje udaljenosti je tada vremenski brzo, ali problem je u prostornoj složenosti: $\mathcal{O}(N^2)$ (jer je u najmanju ruku potrebno pohraniti $\binom{N}{2}$ brojeva), i to za velike skupove podataka već predstavlja veliki problem. Umjesto matrice udaljenosti, analogno postoji matrica sličnosti.

Standardni "naivni" algoritam je vremenske složenosti $\mathcal{O}(N^3)$, no postoje bolje implementacije složenosti $\mathcal{O}(N^2 \log N)$. Specijalno, ako se koriste jednostruka ili potpuna povezanost, postoje algoritmi složenosti $\mathcal{O}(N^2)$ poznati kao SLINK i CLINK.

3.4. DBSCAN

DBSCAN (engl. *Density-based spatial clustering of applications with noise*) je algoritam čvrstog particijskog grupiranja koji se zasniva na **modelu gustoće točaka** (engl. *density model*) u prostoru. Kod takvog su modela grupe sačinjene od "gusto" skupljenih točaka, odnosno točaka koje oko sebe imaju puno bliskih susjednih točaka.

3.4.1. Model gustoće točaka u prostoru

Pojam "gusto" skupljenih točaka definira se u odnosu na svaku točku $\mathbf{x}$ u prostoru kao dovoljan broj ostalih točaka (uključujući i $\mathbf{x}$) čija je udaljenost do $\mathbf{x}$ manja od neke fiksne vrijednosti. Neka je parametar $\varepsilon \geq 0$ polumjer okoline u kojoj se traže susjedne točke, odnosno točke udaljene za ε. Neka $\mathcal{R}(\mathcal{D}, d, \mathbf{x}, \varepsilon)$ označava okolinu točke $\mathbf{x}$ polumjera ε: skup točaka iz $\mathcal{D}$ čija je udaljenost od $\mathbf{x}$ manja ili jednaka ε, s obzirom na mjeru udaljenosti d:

$$\mathcal{R}(\mathcal{D}, d, \mathbf{x}, \varepsilon) = \{\mathbf{y} \in \mathcal{D} \mid d(\mathbf{x}, \mathbf{y}) \leq \varepsilon\}$$

Točka $\mathbf{x}$ također pripada toj okolini.

Kako bi se upotpunio model gustoće točaka, potrebno je specificirati i koliko se najmanje točaka unutar okoline polumjera ε (oko neke točke) mora nalaziti da bi se područje smatralo "gustim", odnosno koliko najmanje točaka mora biti da bi se stvorila jedna grupa. Neka je taj parametar m.
Dakle, uz standardne podatke poput ulaznog skupa točaka i mjere udaljenosti, potrebno je specificirati i dodatne parametre ε i m. S obzirom na uvedene pojmove, model gustoće točaka sve točke iz ulaznog skupa primjera $\mathcal{D}$ dijeli na:

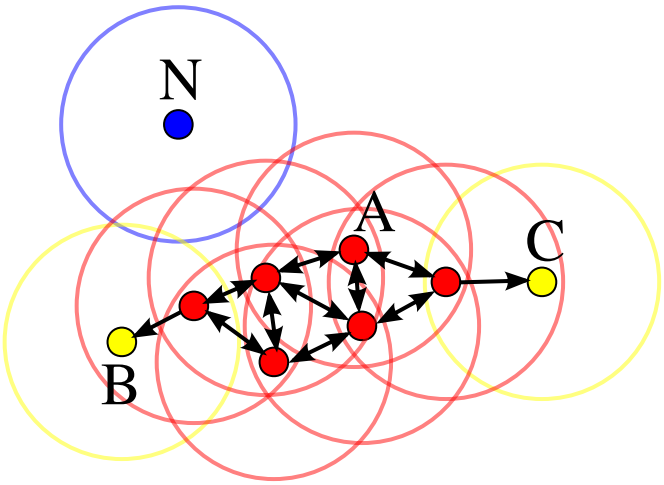
1. **Jezgrene točke** (engl. *core points*): točke koje u svojoj okolini polumjera ε imaju barem m točaka (uključujući i sebe). Točka $\mathbf{x}$ je jezgrena točka ako vrijedi $|\mathcal{R}(\mathcal{D}, d, \mathbf{x}, \varepsilon)| \geq m$.

2. **Granične točke** (engl. *border points*): točke koje nemaju dovoljno (m) točaka u svojoj okolini polumjera ε, ali su dio okoline barem jedne jezgrene točke. Točka $\mathbf{x}$ je granična točka ako vrijedi:
   $|\mathcal{R}(\mathcal{D}, d, \mathbf{x}, \varepsilon)| < m$, $\exists \mathbf{y} \in \mathcal{D} : \mathbf{x} \in \mathcal{R}(\mathcal{D}, d, \mathbf{y}, \varepsilon)$, $\mathbf{y}$ je jezgrena točka

3. **Šum** (engl. *noise points*): točke koje nemaju dovoljno točaka u svojoj okolini polumjera ε i nisu dio okoline nijedne jezgrene točke.

Kod takvog se modela grupom smatraju jezgrene točke i okoline oko njih. Ako se u okolini jezgrene točke nalazi neka druga jezgrena točka, oni zajedno sa svojim okolinama čine istu grupu. Na taj se način točke povezuju u istu grupu, odnosno čine gusto skupljene točke (gusto s obzirom na parametre ε i m). Točka šuma u svojoj okolini može imati samo ostale točke šuma i granične točke. Granične će se točke nalaziti na rubovima takvih grupa, a šum predstavljaju stršeće vrijednosti i takve točke nisu svrstane ni u jednu grupu. Ovakav model omogućava koncept stršećih vrijednosti, za razliku od prethodnih modela kod kojih su oni mogli predstavljati problem kod formiranja grupa jer se nalaze daleko od svih ostalih točaka.

Na slici 3.10 vizualiziran je primjer grupe kod takvog modela.



Slika 3.10: Primjer grupe kod modela gustoća točaka [11]

Okoline točaka polumjera ε vizualizirane su kružnicama oko točaka. U ovom primjeru $m = 4$. Crvene točke su jezgrene, žute su granične, a plava točka N je šum. Sve crvene točke oko sebe imaju 4 ili više točaka u svojoj okolini, što ih čini jezgrenima. Granične točke B i C imaju samo dvije točke u svojoj okolini, no nalaze se u barem jednoj okolini jezgrene točke, dok to nije slučaj sa točkom N. Dakle postoji jedna grupa koju čine sve crvene i žute točke, te postoji jedna točka šuma.

3.4.2. Opis algoritma

Cilj algoritma DBSCAN jest pronaći jezgrene točke i "proširivati" ih sve dok se ne nađu granične točke. Algoritam započinje tako da uzme proizvoljnu točku iz ulaznog

skupa i provjerava koliko točaka ima u svojoj okolini. Ako nema dovoljno, onda se na trenutak točka svrstava kao šum, a ako ima, onda je pronađena jezgrena točka i započinje se nova grupa. Točke iz okoline su također u toj grupi pa ako je neka od tih točaka opet jezgrena, onda je i njihova okolina dio grupe i na taj se način gradi grupa do graničnih točaka. Tada se uzima nova neobrađena točka i postupak se ponavlja.

Rezultat grupiranja u kontekstu algoritma DBSCAN su oznake svake točke: točke sa istim oznakama su u istoj grupi. Jedina su iznimka točke šuma: za njih mora postojati posebna oznaka, ali to ne znači da čine jednu grupu, nego se one smatraju nesvrstanima. Stoga će skup podataka nakon završetka algoritma biti skup uređenih parova

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(i)}, y^{(i)} \right) \right\}_{i=1}^{N}$$

gdje $y^{(i)}$ predstavlja oznaku točke $\mathbf{x}^{(i)}$. Prije izvršena algoritma, s obzirom da se radi o nenazdiranom strojnom učenju, oznake nisu poznate, odnosno nisu definirane. DBSCAN tijekom izvršavanja dodjeljuje oznake svakoj točki koju obradi, najjednostavnije u obliku cijelih brojeva. Oznaka šuma je posebna oznaka. Neka je $\mathrm{L}\left(\mathbf{x}\right)$ oznaka grupe za točku $\mathbf{x}$, dakle za svaki $\mathbf{x}^{(i)} \in \mathcal{D}$ vrijedi

$$\mathrm{L}\left(\mathbf{x}^{(i)}\right) = y^{(i)}$$

U nastavku je opisan algoritam DBSCAN koji dodjeljuje oznake svakoj ulaznoj točki iz $\mathcal{D}$.

Algoritam 4 DBSCAN

Parametri: mjera udaljenosti d, minimalan broj točaka grupe m, polumjer okoline ε

Ulaz: neoznačeni skup točaka $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$

Izlaz: označeni skup točaka $\mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ sa **definiranim** oznakama $y^{(i)}$

$k \leftarrow 0$

za svaki $\mathbf{x}^{(i)} \in \mathcal{D}$ (1)

vrati se na početak petlje (1) ako je oznaka $\mathrm{L}\left(\mathbf{x}^{(i)}\right)$ definirana

$\mathcal{S} \leftarrow \mathcal{R}\left(\mathcal{D}, d, \mathbf{x}^{(i)}, \varepsilon\right)$

ako $|\mathcal{S}| < m$ **onda**

$\mathrm{L}\left(\mathbf{x}^{(i)}\right) \leftarrow$ šum

vrati se na početak petlje (1)

kraj ako

$k \leftarrow k + 1$

$\mathrm{L}\left(\mathbf{x}^{(i)}\right) \leftarrow k$

$\mathcal{S} \leftarrow \mathcal{S} \setminus \left\{\mathbf{x}^{(i)}\right\}$

stvori red elemenata $\mathcal{Q} \leftarrow \mathcal{S}$

ponavljaj dok red $\mathcal{Q}$ nije prazan (2)

ukloni prvi element $\mathbf{q}$ iz reda $\mathcal{Q}$

ako $\mathrm{L}(\mathbf{q}) =$ šum **onda** $\mathrm{L}(\mathbf{q}) \leftarrow k$ {Granična točka}

vrati se na početak petlje (2) ako je oznaka $\mathrm{L}(\mathbf{q})$ definirana

$\mathrm{L}(\mathbf{q}) \leftarrow k$

$\widehat{\mathcal{S}} \leftarrow \mathcal{R}(\mathcal{D}, d, \mathbf{q}, \varepsilon)$

ako $\left|\widehat{\mathcal{S}}\right| \geq m$ **onda**

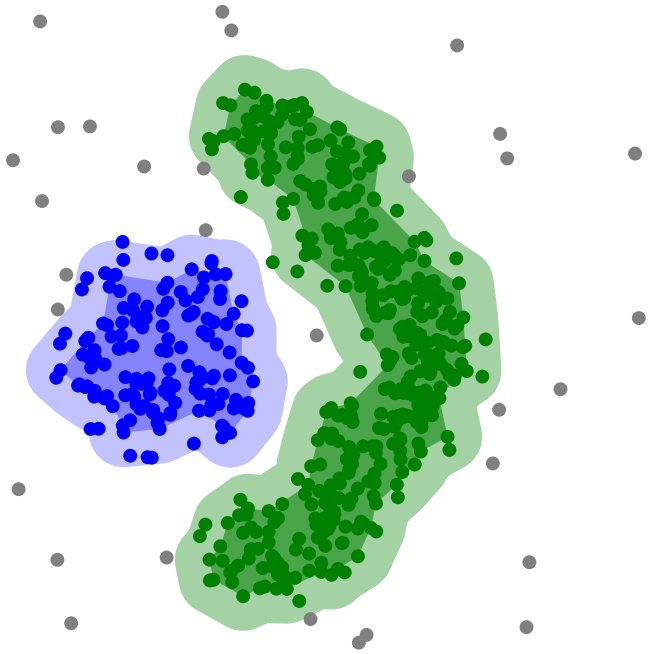
stavi sve elemente iz $\widehat{\mathcal{S}}$ na kraj reda $\mathcal{Q}$

kraj ako

kraj ponavljaj

kraj za

Na slici 3.11 je prikazan primjer grupiranja dvodimenzijskih podataka algoritmom DBSCAN.



Slika 3.11: Primjer grupiranja algoritmom DBSCAN [11]

Točke koje se nalaze unutar područja svjetlije nijanse su granične točke, a točke unutar područja tamnije nijanse su jezgrene točke. Sive točke su šum.

3.4.3. Svojstva i složenost algoritma

Najveća prednost modela gustoće točaka i algoritma DBSCAN jest mogućnost razdvajanja grupa nekonveksnih oblika, što nije slučaj kod algoritma K-sredina i EM algoritma. To je omogućeno samim modelom gustoće točaka jer se grupe grade prema tome koliko su gusto raspoređene u prostoru, a ne koliko su udaljene od centroida.

Nije potrebno unaprijed odrediti broj grupa koje algoritam treba proizvesti, što nije lako odrediti općenito. S druge strane, potrebno je specificirati dobre parametre m i ε, što može biti izazovno ukoliko se ne poznaje priroda i domena podataka. Ukoliko se odaberu premali m i preveliki ε, algoritam će spajati rijetka područja što može rezultirati spajanjem više prirodnih grupa u jednu zajedno sa šumom. S druge strane, ako se odaberu preveliki m i premali ε, algoritam će veliki dio točaka označiti kao šum. Odabir ispravnih parametara m i ε postaje još izazovniji ukoliko grupe variraju u gustoći.

Vremenska složenost algoritma DBSCAN ovisi na koji je način izvedeno traženje okoline točke $\mathbf{x}$ polumjera ε: $\mathcal{R}(\mathcal{D}, d, \mathbf{x}, \varepsilon)$. Naivan je pristup linearna pretraga svakog elementa iz $\mathcal{D}$ i računanje udaljenosti do $\mathbf{x}$. Takav pristup je vremenske složenosti $\mathcal{O}(Nn)$. Najčešće se u implementacijama za spremanje ulaznog skupa podataka $\mathcal{D}$

koriste strukture podataka osmišljene za spremanje prostornih podataka[6]. U tom se slučaju traženje $\mathcal{R}(\mathcal{D}, d, \mathbf{x}, \varepsilon)$ ubrzava i vremenske je složenosti $\mathcal{O}(\log N)$. Traženje okoline obavlja se za svaku točku, dakle vremenska složenost algoritma DBSCAN je $\mathcal{O}(N^2)$ kod naivnog pristupa, a $\mathcal{O}(N \log N)$ ako se koristi brza pretraga okoline, što je prihvatljivo za velike skupove podataka. Što se tiče prostorne složenosti, ona je $\mathcal{O}(N)$ jer je potrebna struktura podataka koja omogućava spremanje "neobrađenih" susjednih točaka i njihovo uklanjanje nakon obrade. Ako se koristi prethodno pripremljena matrica udaljenosti, tada se radi o prostornoj složenosti $\mathcal{O}(N^2)$, ali se zauzvrat ubrzava računanje udaljenosti.

[6]Primjerice K-D stablo, R-stablo. Ako se ulazni podaci spreme u neku bazu podataka, ta baza onda može stvoriti jednu od takvih struktura kao indeks.

4. Postupci vrednovanja algoritama grupiranja

Prethodno se poglavlje bavilo problemom grupiranja: na koji način podijeliti ulazni skup točaka u grupe sa međusobno sličnim točkama. Nakon što se to napravi na neki način, postavlja se pitanje na koji način ocijeniti rezultat grupiranja. Taj zadatak može biti jednako težak kao i sam problem grupiranja ukoliko nije dostupna neka vanjska informacija o tome kako bi podaci trebali stvarno biti grupirani. Naravno, kada bi na raspolaganju bile takve informacije, ne bi bilo potrebe za grupiranjem. Prema tome, vrednovanja grupiranja dijele se na **unutarnje** (engl. *internal evaluation*) i **vanjsko** (engl. *external evaluation*) vrednovanje. Kod vanjskog vrednovanja, osim ulaznih podataka dostupne su i njihove oznake, dok to nije slučaj kod unutarnjeg vrednovanja.

4.1. Unutarnje vrednovanje

Unutarnje vrednovanje nastoji nekom grupiranju dodijeliti ocjenu samo na temelju ulaznih podataka i oznaka koje je generiralo grupiranje. Kriteriji vrednovanja najčećše daju veće ocjene grupiranjima koja stvaraju grupe u kojima su točke međusobno slične, a različite u odnosu na točke iz ostalih grupa. Nedostatak takvih vrednovanja jest taj da će dati bolje rezultate algoritmima koji upravo nastoje povećati ocjenu tog specifičnog vrednovanja, a da takav algoritam nije u stvarnosti obavio grupiranje na optimalan način.

4.1.1. Davies-Bouldin indeks

Davies-Bouldin indeks promatra sve parove grupa i bilježi koliko su grupe udaljene, kao i koliko su grupe same po sebi raspršene. Neka je dano grupiranje $\Gamma = \{\mathcal{C}_k\}_{k=1}^{K}$.

Neka je S_i prosječna kvadratna udaljenost točaka iz grupe $\mathcal{C}_i$ od centroida $\boldsymbol{\mu}_i$ te grupe:

$$S_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{x} \in \mathcal{C}_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

S_i je mjera raspršenja unutar grupe $\mathcal{C}_i$. Naravno, kako bi se mogao prema ovakvoj definiciji računati S_i, mora se moći izračunati centroid, a to je moguće jedino ako podaci dolaze iz nekog vektorskog prostora. Ranije je spomenuto da se najčešće radi o realnom koordinatnom prostoru pa se koristi Euklidova udaljenost i Euklidova norma. Neka je M_{ij} udaljenost centroida grupa $\mathcal{C}_i$ i $\mathcal{C}_j$:

$$M_{ij} = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$$

M_{ij} mjeri koliko su dobro razdvojene grupe u smislu koliko su im udaljeni centroidi. Neka je R_{ij} definiran kao

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}$$

Ono mjeri koliko su dobro grupirane međusobno grupe $\mathcal{C}_i$ i $\mathcal{C}_j$: veća razdvojenost i manje raspršenje unutar grupa dat će manji iznos R_{ij}, što znači bolje grupiranje između te dvije grupe. R_{ij} se može shvatiti kao neka mjera sličnosti između dvije grupe. Neka R_i označava najveći iznos R_{ij} za grupu $\mathcal{C}_i$ i sve ostale grupe:

$$R_i = \max_{\substack{1 \leq j \leq K \\ j \neq i}} R_{ij}$$

Davies-Bouldin indeks DBI se tada definira kao

$$DBI = \frac{1}{K} \sum_{k=1}^{K} R_k$$

Nedostatak vrednovanja Davies-Bouldin indeksom je činjenica da se veći iznosi postižu ako su grupe konveksnog oblika, odnosno iznosi su veći kod grupiranja sa centroidnim modelima (algoritam K-sredina, EM algoritam i model Gaussovih mješavina) nego kod modela kojeg koristi primjerice DBSCAN. Uz to, Davies-Bouldin indeks može se računati samo nad grupiranjima kod kojih je moguće računati centroide i može se koristiti Euklidova udaljenost.

4.1.2. Vrijednost siluete

Vrijednost siluete (engl. *silhouette score*) mjeri koliko je svaka točka iz $\mathcal{D}$ slična svojoj grupi u odnosu na ostale grupe, dakle, svakoj se točki pridružuje vrijednost siluete. Metoda koja se zasniva na ovoj mjeri naziva se **metoda siluete**. Za svaku točku $\mathbf{x}^{(i)}$ iz

ulaznog skupa točaka $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ neka $\mathcal{C}\left(i\right)$ označava grupu kojoj pripada ta točka. Također je na raspolaganju grupiranje koje je generirao promatrani algoritam, odnosno skup od K grupa $\Gamma = \{\mathcal{C}_k\}_{k=1}^{K}$. Prije nego što se definira vrijednost siluete $s(i)$, neka $a(i)$ označava prosječnu udaljenost točke $\mathbf{x}^{(i)}$ do ostalih točaka u svojoj grupi:

$$a(i) = \frac{1}{\left|\mathcal{C}\left(i\right)\right| - 1} \sum_{\substack{\mathbf{y} \in \mathcal{C}(i) \\ \mathbf{y} \neq \mathbf{x}^{(i)}}} d\left(\mathbf{x}^{(i)}, \mathbf{y}\right)$$

Neka $b(i)$ označava prosječnu udaljenost točke $\mathbf{x}^{(i)}$ do točaka najbliže grupe (najbližu u smislu prosječne udaljenosti):

$$b(i) = \min_{\substack{\mathcal{C} \in \Gamma \\ \mathcal{C} \neq \mathcal{C}(i)}} \frac{1}{|\mathcal{C}|} \sum_{\mathbf{y} \in \mathcal{C}} d\left(\mathbf{x}^{(i)}, \mathbf{y}\right)$$

Vrijednost siluete se tada definira kao

$$s(i) = \frac{b(i) - a(i)}{\max\left\{a(i), b(i)\right\}}$$

a u slučaju da je točka $\mathbf{x}^{(i)}$ sama u svojoj grupi, odnosno $\left|\mathcal{C}\left(i\right)\right| = 1$, tada je $s(i) = 0$ jer vrijednost $a(i)$ nema smisla. Uzimajući u obzir međusobne odnose $a(i)$ i $b(i)$, definicija vrijednosti siluete može se izraziti na sljedeći način:

$$s(i) = \begin{cases} 0 & \text{ako } \left|\mathcal{C}\left(i\right)\right| = 1 \\ 1 - \frac{a(i)}{b(i)} & \text{ako } a(i) < b(i) \\ 0 & \text{ako } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{ako } a(i) > b(i) \end{cases}$$

Jasno je vidljivo iz ovakve definicije vrijednosti siluete da ona može samo poprimiti vrijednosti između -1 i 1, odnosno $-1 \leq s(i) \leq 1$. Kako $a(i)$ mjeri koliko je točka $\mathbf{x}^{(i)}$ blizu ostalim točkama iz svoje grupe, manja vrijedost $a(i)$ znači veću sličnost $\mathbf{x}^{(i)}$ sa ostalim točkama iz svoje grupe. S druge strane, $b(i)$ mjeri koliko je $\mathbf{x}^{(i)}$ blizu točkama iz najbliže susjedne grupe, stoga veća vrijednost $b(i)$ znači veću "razdvojenost", odnosno veću različitost od ostalih grupa. Prema tome, kada je $a(i)$ puno manji od $b(i)$, vrijednost siluete $s(i)$ će biti blizu 1, što označava dobro grupiranje konkretno za promatranu točku. Analogno, kada je $b(i)$ puno veći od $a(i)$, to znači da je točka bliža nekoj drugoj grupi nego vlastitoj. Tada će $s(i)$ biti blizu -1 i to označava loše grupiranje. Vrijednost siluete blizu 0 znači da se $a(i)$ i $b(i)$ malo razlikuju, što znači da je točka $\mathbf{x}^{(i)}$ na granici dvije grupe.

Osim što se može računati vrijednost siluete za svaku točku $\mathbf{x}^{(i)}$ iz ulaznog skupa točaka $\mathcal{D}$, može se računati vrijednost siluete za cijelo grupiranje kao aritmetička sredina silueta svih točaka:

$$s = \sum_{i=1}^{N} \frac{s(i)}{N}$$

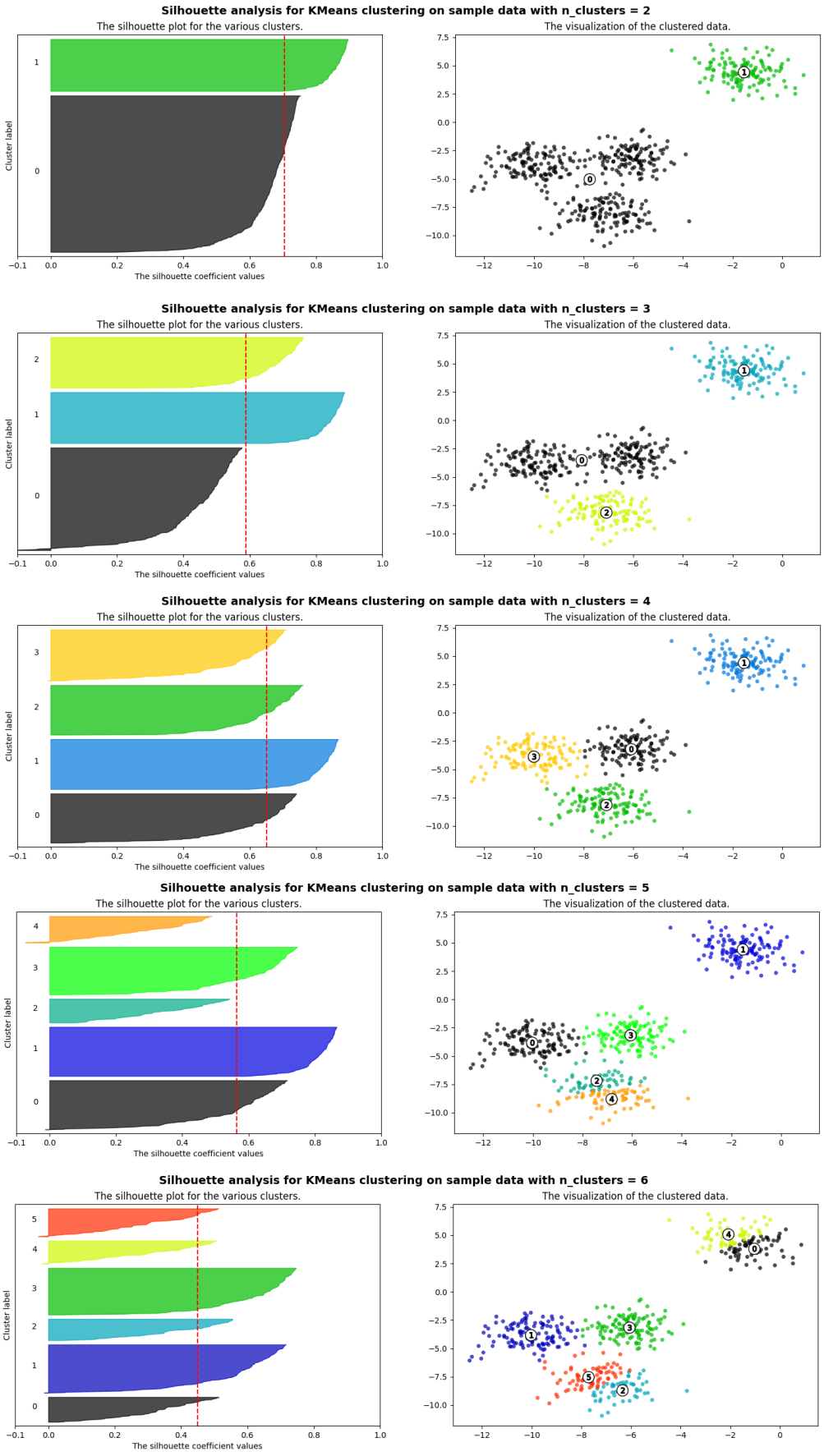
Kao i kod Davies-Bouldin indeksa, vrijednosti siluete veće su kod točaka grupa konveksnog oblika, što ga ne čini dobrim izborom unutarnjeg vrednovanja ako se grupiranje radi modelom koji ne pretpostavlja konveksan oblik grupa (primjerice model gustoće točaka kod algoritma DBSCAN).

Metoda siluete

Metoda siluete, odnosno analiza siluete, zasniva se na računanju vrijednosti siluete svake točke iz ulaznog skupa. Metoda siluete može se iskoristiti kako bi se odredio optimalan broj grupa kod algoritma K-sredina. U nastavku je primjer analize siluete nad istim ulaznim podacima kao na slici 3.3 gdje je bila demonstrirana metoda koljena.

Neka je dano grupiranje koje je proizvelo K-sredina. Nakon grupiranja, vrijednost siluete računa se za svaku točku iz ulaznog skupa i grafički se prikazuje na dijagramu uz zabilježenu informaciju u kojoj grupi se nalazi svaka točka. Također se prati koja je prosječna vrijednost siluete za sve točke te se grafički pokušava zaključiti koliko je kvalitetno grupiranje s obzirom na dobivene vrijednosti siluete u odnosu na prosječnu vrijednost. Analiza siluete provedena je za brojeve grupa $K \in \{2, 3, 4, 5, 6\}$ na slici 4.1. Na lijevoj strani je dijagram sa sortiranim vrijednostima siluete za svaki ulazni podatak u svakoj dobivenoj grupi. Crvena crtkana linija predstavlja prosječnu vrijednost siluete za cijeli skup podataka. Na desnoj strani je prikazan rezultat grupiranja za različite parametre K.

Ona grupiranja kod kojih postoje točke sa negativnim vrijednostima siluete su loša. U ovom primjeru, takva grupiranja su ona sa brojem grupa 3, 5, 6. Također su loša ona grupiranja kod kojih postoji grupa čija svaka točka ima vrijednost siluete manju od prosječne. To je slučaj kod grupiranja s brojem grupa 3 (crno obojana grupa) i 5 (tirkizno i narančasto obojane grupe). Dakle, grupiranja s brojem grupa 2 i 4 prihvatljiva su u kontekstu analize siluete.



Slika 4.1: Analiza siluete

4.2. Vanjsko vrednovanje

Vanjsko vrednovanje, uz ulazne podatke i grupe koje je promatrani algoritam generirao, na raspolaganju ima i oznake svake točke koje otkrivaju referentno grupiranje. Te oznake nisu poznate postupcima nenadziranog strojnog učenja, pa tako i algoritmima grupiranja. Do njih je ponekad moguće doći, primjerice ljudskom ocjenom ulaznog skupa primjera ili su mogli biti poznati ranije, no odluka je bila ne specificirati oznake metodama nenadziranog strojnog učenja (pa tako i grupiranja).

4.2.1. Randov indeks

Randov indeks mjeri preciznost grupiranja tako da promatra sve parove ulaznih primjera i uspoređuje dva grupiranja istovremeno. Jedno je grupiranje ono koje je generirao algoritam grupiranja, a drugo je grupiranje (referentno grupiranje) prema prethodno poznatim oznakama: dva su primjera u istoj grupi ako imaju istu oznaku. Za svaki mogući par primjera promatra se jesu li završili u istoj grupi i jesu li oni u istoj grupi kod referentnog grupiranja. Parovi ulaznih primjera se tada dijele na:

1. **Istinito pozitivne** (engl. *true positive*): primjeri se nalaze u istoj grupi u oba grupiranja. Neka je TP broj takvih parova.

2. **Istinito negativne** (engl. *true negative*): primjeri se nalaze u različitim grupama u oba grupiranja. Neka je TN broj takvih parova.

3. **Lažno pozitivne** (engl. *false positive*): primjeri se nalaze u istoj grupi u dobivenom grupiranju, a u referentnom grupiranju se nalaze u različitim grupama. Neka je FP broj takvih parova.

4. **Lažno negativne** (engl. *false negative*): primjeri se nalaze u različitim grupama u dobivenom grupiranju, a u referentnom grupiranju se nalaze u istoj grupi. Neka je FN broj takvih parova.

Parovi primjera su dobro grupirani ako situacija odgovara referentnom grupiranju, dakle promatraju se primjeri koji su istinito pozitivni i istinito negativni. Randov indeks RI se tada definira kao:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{\binom{N}{2}}$$

Jasno je da su u nazivniku pokriveni svi mogući parovi ulaznih primjera, dakle broj takvih parova mora biti $\binom{N}{2}$. Moguće vrijednosti su $RI \in [0, 1]$. Grupiranje koje je identično referentnom grupiranju ima $RI = 1$.

Primjerice, neka su ulazni primjeri skup od 5 prirodnih brojeva od 1 do 5. Neka su ulazni podaci grupirani na sljedeći način:

$$\{\{1,2,4\},\{3,5\}\}$$

Neka je referentno grupiranje

$$\{\{1,2\},\{3,4\},\{5\}\}$$

Za ovaj primjer je $TP = 1$ jer samo par 12 se nalazi u istim grupama u oba grupiranja, a $TN = 5$ i radi se o parovima 13, 15, 23, 25, 45. Prema tome $RI = \frac{1+5}{\binom{5}{2}} = \frac{6}{10} = 0.6$.

Nedostatak vrednovanja Randovim indeksom jest činjenica da jednako nagrađuje istinito pozitivne i istinito negativne parove primjera. U slučaju da se brojevi grupa razlikuju između dobivenog i referentnog grupiranja, to rezultira velikim brojem istinito negativnih primjera. Postoji mogućnost da Randov indeks slučajnog grupiranja bude broj puno veći od 0, kao i da Randov indeks relativno lošeg grupiranja bude broj blizu 1. Taj problem rješava **prilagođeni Randov indeks** (engl. *adjusted Rand index*) koji uzima u obzir očekivani Randov indeks za slučajna grupiranja. Prilagođeni Randov indeks slučajnim grupiranjima dodijeljuje vrijednosti bliže nuli, a lošim grupiranjima može dodijeliti negativnu vrijednost. Prilagođeni Randov indeks je češće korištena mjera vanjskog vrednovanja u odnosu na neprilagođeni. Način na koji se računa prilagođeni Randov indeks i matematičke formulacije iza njega nećemo navoditi.

4.2.2. Uzajamna informacija

Uzajamna informacija (engl. *mutual information*) je mjera iz teorije informacije koja mjeri kako se mijenja nesigurnost predviđanja jedne slučajne varijable ukoliko su poznate realizacije druge slučajne varijable. U kontekstu grupiranja radi se o oznakama grupa, (multi)skupovima koji u sebi sadrže oznake za svaki ulazni podatak.

Neka je X diskretna slučajna varijabla koja može poprimiti vrijednosti $x_1, \ldots, x_n$ sa vjerojatnostima $P(x_i) = P(X = x_i)$. **Entropija** slučajne varijable X se definira kao

$$H(X) = \sum_{i=1}^{n} P(x_i) \log P(x_i)$$

gdje izbor baze logaritma može varirati između primjena, no najčešće se uzima baza 2. Kod grupiranja, skup oznaka grupa se mogu smatrati kao realizacijama neke slučajne varijable. Neka skup U sadrži ukupno N oznaka ($|U| = N$) od kojih je n različitih oznaka $u_1, \ldots, u_n$, te neka je U_i podskup (konceptualno radi se o grupi s tom oznakom)

od U koji sadrži samo oznake u_i. Entropija takvog grupiranja $H(U)$ je tada

$$H(U) = -\sum_{i=1}^{n} P(u_i) \log P(u_i) = -\sum_{i=1}^{n} \frac{|U_i|}{N} \log \frac{|U_i|}{N}$$

Primjerice, entropija skupa $U = \{0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2\}$ (ovdje je $N = 11$ i $n = 3$ jer su prisutne 3 različite vrijednosti) je

$$H(U) = -\left(\frac{3}{11} \log \frac{3}{11} + \frac{2}{11} \log \frac{2}{11} + \frac{6}{11} \log \frac{6}{11}\right) = 1.435$$

te bi to odgovaralo entropiji grupiranja $\{\{a, b, c\}, \{d, e\}, \{f, g, h, i, j, k\}\}$ gdje mala slova predstavljaju svaki ulazni podatak.

Uzajamna informacija za dva grupiranja (dva skupa oznaka) U i V, gdje U ima n različitih oznaka, a V ima m različitih oznaka, definira se kao

$$\mathrm{MI}(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(u_i, v_j) \log \frac{P(u_i, v_j)}{P(u_i) P(v_j)} = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|U_i \cap V_j|}{N} \log \frac{N |U_i \cap V_j|}{|U_i| |V_j|}$$

Što su grupiranja sličnija, to će uzajamna informacija biti veća.

Osim uzajamne informacije, koristi se i **normalizirana uzajamna informacija** koja se definira kao

$$\mathrm{NMI}(U, V) = \frac{\mathrm{MI}(U, V)}{\frac{H(U) + H(V)}{2}}$$

i moguće vrijednosti su između 0 i 1. Obje mjere imaju isti problem kao i neprilagođeni Randov indeks: slučajna grupiranja imaju iznose koji su znatno veći od 0. Na isti se način uvodi i najčešće koristi **prilagođena uzajamna informacija** (engl. *adjusted mutual information*).

5. Programsko ostvarenje i rezultati

5.1. Programska knjižnica Scikit-learn

Scikit-learn besplatna je programska knjižnica otvorenog koda za programski jezik Python namijenjena za strojno učenje. Za potrebe ovog završnog rada posebno je zanimljiv modul za grupiranje `sklearn.cluster` gdje je implementirana većina[1] algoritama grupiranja. Mjere vrednovanja implementirane su u `sklearn.metrics`.

Svaki je algoritam grupiranja modeliran razredom, a u konstruktoru takvog razreda specificiraju se parametri i postavke algoritma. Primjerice, algoritam K-sredina je modeliran razredom `sklearn.cluster.KMeans` i u konstruktoru se može specificirati: broj grupa, maksimalan broj iteracija, strategiju inicijalizacije početnih centroida, kolika se promjena kriterijske funkcije između iteracija smatra konvergencijom, koliko puta ponoviti algoritam, žele li se prethodno izračunati udaljenosti između svih parova točaka, itd. Na sličan se način mogu podešavati i ostali algoritmi grupiranja, a kod onih čiji model to dopušta može se specificirati i proizvoljna mjera udaljenosti.

Pokretanje algoritma nad stvorenim objektom postiže se pozivom metode `fit` ili neke srodne metode (primjerice metoda `fit_predict`). Kao obavezan argument prima se matrica[2] dimenzija $N \times n$, gdje svaki redak te matrice predstavlja jedan ulazni n-dimenzionalni podatak. Metoda `fit` ne obavlja nužno grupiranje, već "namješta" model iz ulaznih podataka. U kontekstu strojnog učenja, radi se o treniranju modela. U slučaju algoritma K-sredina, računaju se završni centroidi. Kod EM algoritma Gaussovih mješavina, računaju se parametri $\boldsymbol{\theta}$. Kod hijerarhijskog aglomerativnog grupiranja, gradi se "presiječeno" (dakle ne potpuno) stablo hijerarhije, a algoritam DBSCAN traži jezgrene točke i usput obavlja grupiranje. Metoda `fit_predict` uz treniranje obavlja i grupiranje ulaznog skupa točaka. Pozivatelju se vraća polje duljine N koje sadrži cjelobrojne oznake čvrstih grupa počevši od 0 za svaki ulazni podatak (oznaka na indeksu i je oznaka grupe za ulazni podatak u i-tom retku). Algoritam DB-

[1] Osim EM algoritma Gaussovih mješavina koja je dio `sklearn.mixture` modula.

[2] Za modeliranje i manipulaciju matrica koristi se programska knjižnica NumPy

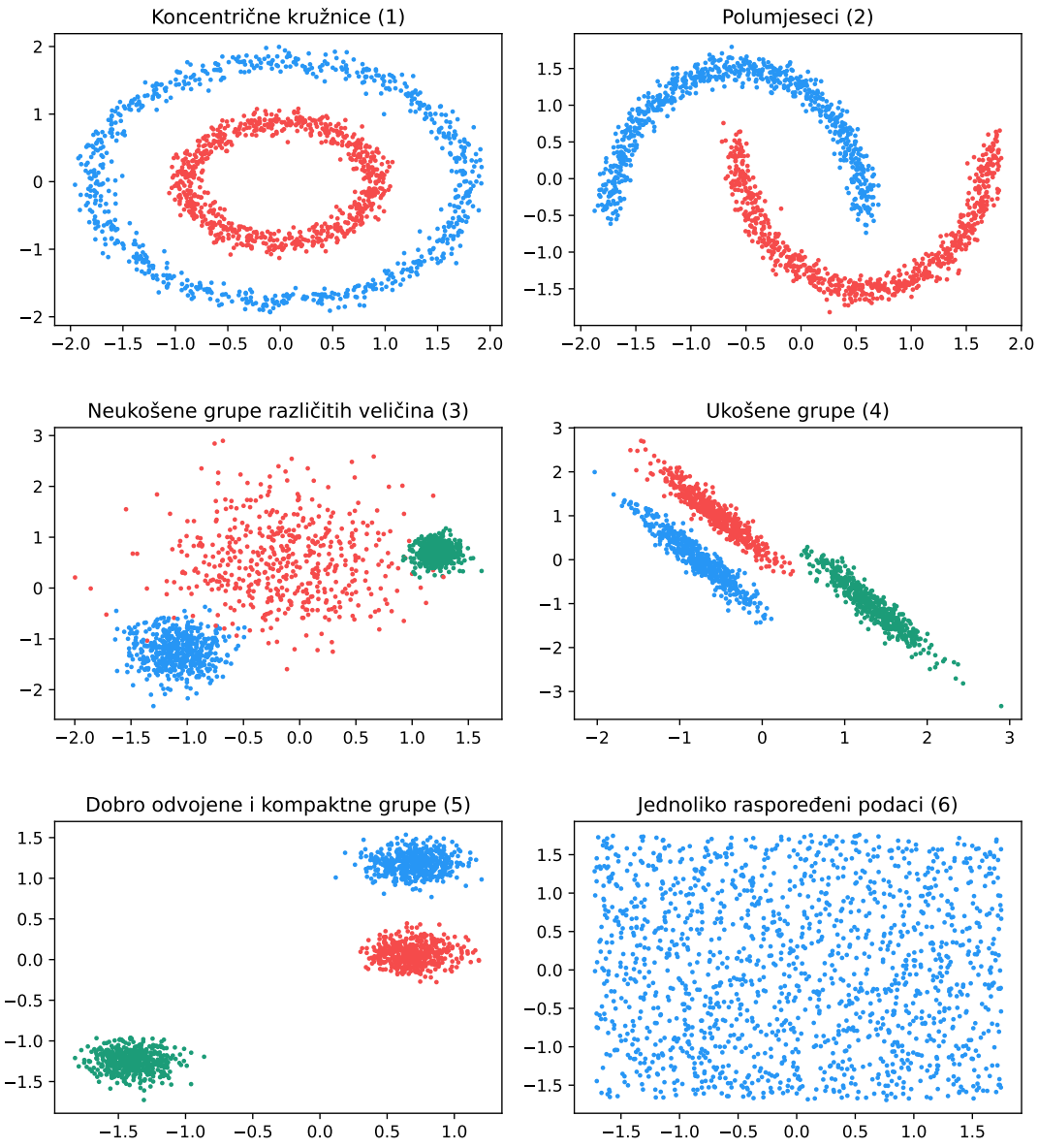
SCAN šum označuje sa -1. Ako postoji za konkretan razred, može se koristiti i metoda `predict` koja dodjeljuje oznake grupa podataka koji nisu nužno ulazni podaci nad kojima je treniran model. Kako se radi o nenadziranom strojnom učenju, mogu se i novi neviđeni podaci klasificirati na temelju starih (korištenih za treniranje modela). Od opisanih algoritma u poglavlju 3, tu mogućnost nude algoritam K-sredina i EM algoritam Gaussovih mješavina. Kod algoritma K-sredina, svaki podatak predan kao argument metode `predict` svrstava se u onu grupu čiji je centroid najbliži, a kod Gaussovih mješavina svrstava se u onu komponentu sa najvećom odgovornošću. Posebno za model Gaussovih mješavina, postoji i metoda `predict_proba` koja vraća odgovornosti (meko grupiranje) za svaki ulazni podatak i svaku komponentu u obliku matrice $N \times K$. Naglasak ovog rada je usporedba metoda grupiranja postojećih podataka, a ne klasifikacija novih neviđenih primjera, stoga je korištena samo metoda `fit_predict`.

5.2. Skupovi podataka

Za potrebe usporedbe algoritama grupiranja korišteno je 6 različitih skupova podataka. Radi se o umjetno stvorenim podacima koji služe za demonstraciju i usporedbu različitih algoritama grupiranja. Kako bi svi opisani algoritmi i modeli grupiranja bili primjenjivi, značajke svakog podatka realni su brojevi, a njihova je dimenzija 2 kako bi se lakše vizualno prikazali. Svaki skup ima 1500 ulaznih točaka kako trajanje izvršenja algoritama ne bi bilo predugo, a s druge strane to je sasvim dovoljno za demonstraciju. Prema tome

$$N = 1500, \qquad n = 2, \qquad \mathcal{V} \subseteq \mathbb{R}^2$$

Na slici 5.1 prikazane su točke svakog korištenog skupa podataka u realnom koordinatnom prostoru. Kako su podaci umjetno stvoreni, oznake referentnih grupa unaprijed su poznate kako bi se algoritmi mogli vrednovati mjerama vanjskog vrednovanja. Referentno grupiranje prikazano je bojanjem točaka. Praktični skupovi podataka nisu ovako jednostavni, niti su unaprijed poznate oznake grupa.



Slika 5.1: Korišteni skupovi podataka

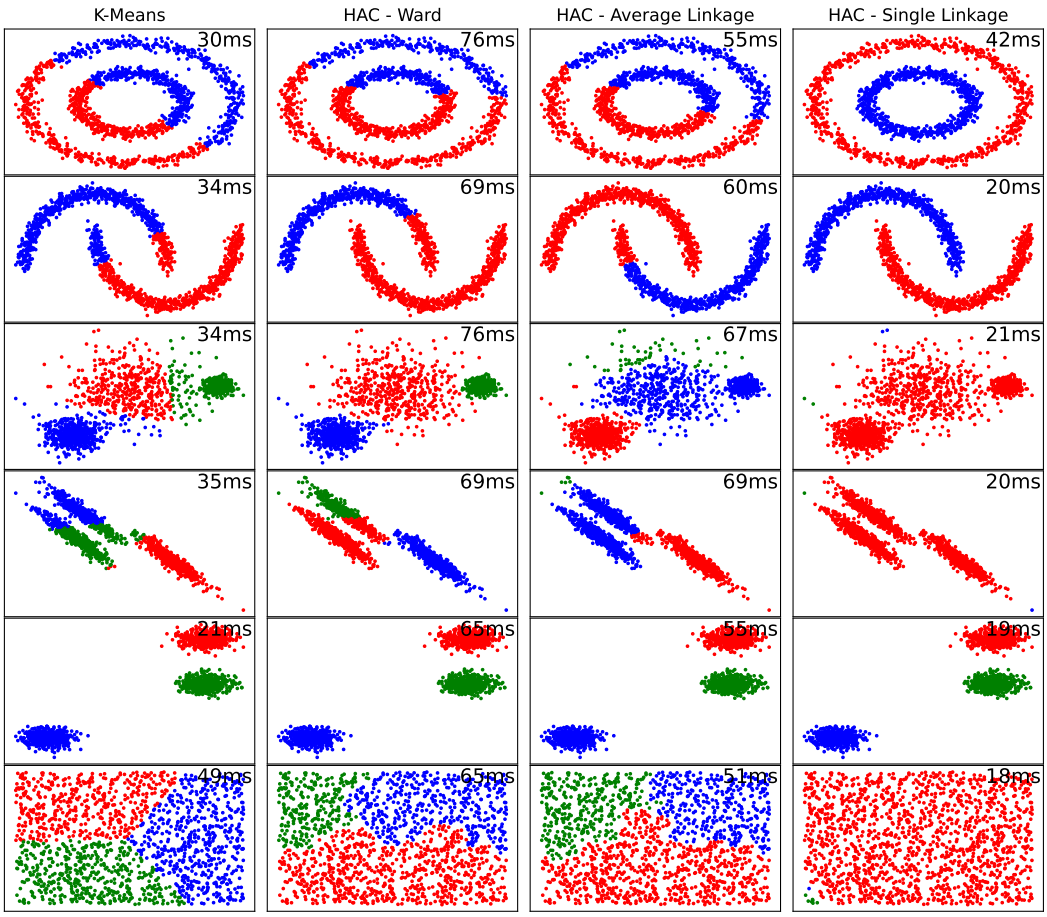
Skupovi podataka (1) i (2) primjeri su s grupama nekonveksnih oblika. Podaci iz skupova (3), (4), (5) uzorkovani su iz dvodimenzionalnih normalnih razdioba te se grupiraju prema tome iz koje od tri razdiobe su uzorkovane, a točke skupa (6) uzorkovane su iz dvodimenzionalne jednolike distribucije na nekom području.

5.3. Rezultati

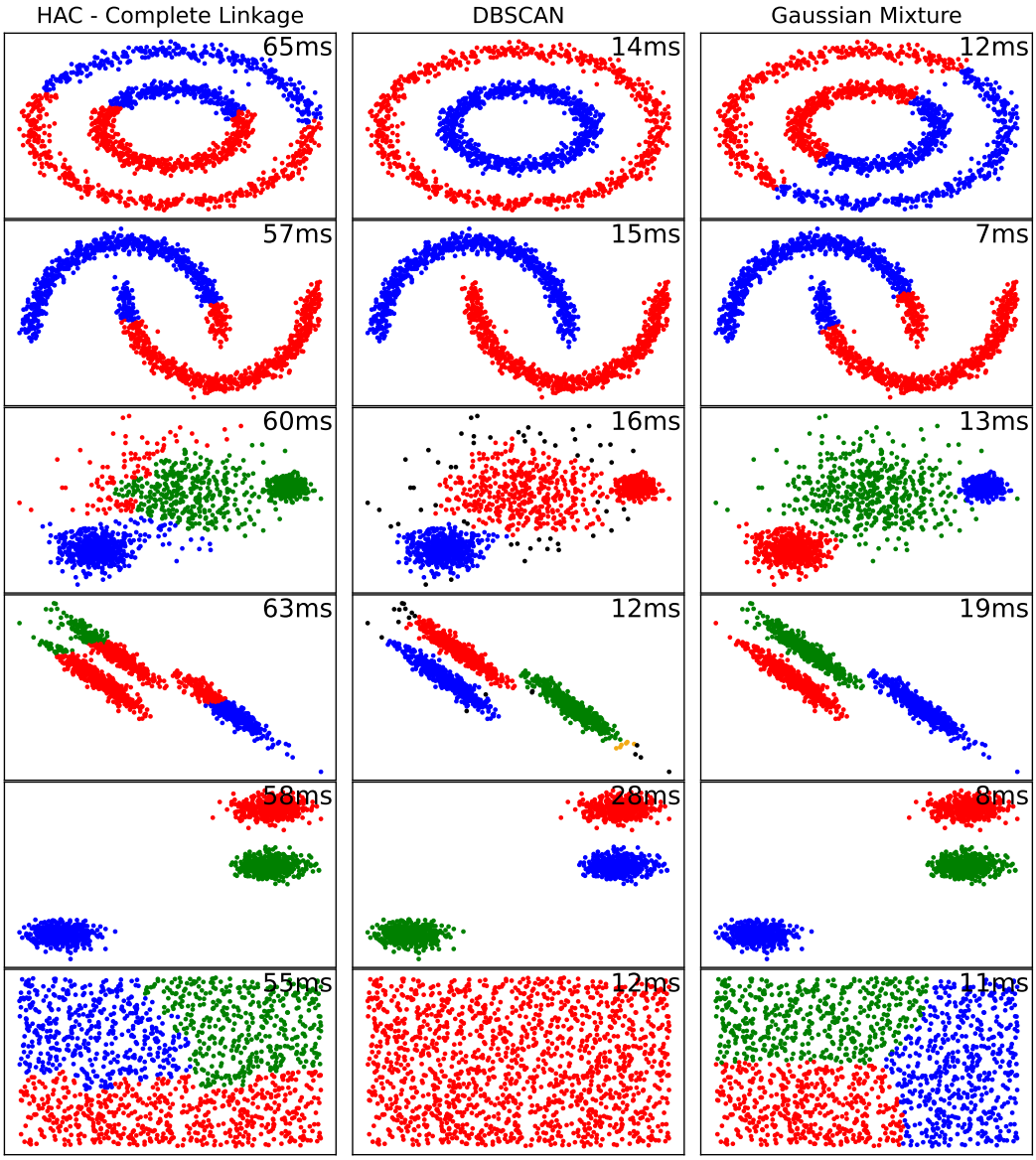
Nad spomenutim skupovima podataka pokrenuto je 7 algoritama grupiranja: K-sredina, DBSCAN, EM algoritam Gaussovih mješavina (bez ograničenja kovarijacijske ma-

trice) i Hijerarhijsko aglomerativno grupiranje sa svakim od 4 opisana kriterija spajanja. Za svaki algoritam i skup podataka specificirani su optimalni parametri, osim za skup (6). U slučaju algoritama kod kojih se može specificirati broj grupa (ili broj komponenata kod Gaussovih mješavina), uvijek je odabran onaj broj grupa koji odgovara prirodnom grupiranju. U slučaju algoritma DBSCAN, parametri m i ε odabrani su tako da daju najbolje moguće rezultate za taj model. Što se tiče skupa (6), odabrani su parametri identični kao za skup (5). U svim je algoritmima korištena Euklidova udaljenost.

Rezultati grupiranja prikazani su na slikama 5.2 i 5.3. Vrijeme u milisekundama potrebno za izvršenje nekog algoritma nad nekim skupom podataka zapisano je u gornjem desnom kutu. Kod algoritma DBSCAN, točke šuma obojane su crnom bojom.



Slika 5.2: Rezultati grupiranja



Slika 5.3: Rezultati grupiranja

Uz grupiranje, izračunate su mjere unutarnjeg i vanjskog vrednovanja za svako grupiranje (osim za skup podataka (6)) te su prikazane tablicom 5.1. Kako tablica ne bi bila prevelika, od mjera vanjskog vrednovanja uključeni su samo prilagođeni Randov indeks i prilagođena uzajamna informacija.

Tablica 5.1: Iznosi mjera unutarnjeg i vanjskog vrednovanja

Skup	Mjera	K-Means	HAC - Ward	HAC - Average	HAC - Single	HAC - Complete	DBSCAN	GMM/EM
(1)	ARI	-6.604×10^{-4}	2.409×10^{-4}	-6.322×10^{-4}	1.000	3.816×10^{-3}	1.000	6.035×10^{-4}
	AMI	-4.765×10^{-4}	1.920×10^{-4}	-4.623×10^{-4}	1.000	2.976×10^{-3}	1.000	-4.454×10^{-4}
	DBI	1.185	1.197	1.186	989.794	1.195	989.794	1.189
	s	0.354	0.323	0.349	0.114	0.330	0.114	0.352
(2)	ARI	0.486	0.559	0.730	1.000	0.600	1.000	0.501
	AMI	0.387	0.559	0.691	1.000	0.492	1.000	0.401
	DBI	0.804	0.838	0.871	1.023	0.814	1.023	0.804
	s	0.500	0.455	0.456	0.389	0.493	0.389	0.500
(3)	ARI	0.809	0.961	0.551	1.427×10^{-5}	0.538	0.550	0.966
	AMI	0.797	0.938	0.667	2.220×10^{-3}	0.627	0.664	0.942
	DBI	0.635	0.656	0.757	0.659	0.844	2.222	0.662
	s	0.626	0.604	0.512	0.130	0.542	0.532	0.596
(4)	ARI	0.608	0.685	0.544	-8.889×10^{-7}	0.212	0.975	1.000
	AMI	0.618	0.748	0.652	-8.037×10^{-7}	0.358	0.955	1.000
	DBI	0.700	0.687	0.510	0.394	0.677	3.798	0.850
	s	0.510	0.480	0.490	0.216	0.313	0.396	0.472
(5)	ARI	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	AMI	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	DBI	0.270	0.270	0.270	0.270	0.270	0.270	0.270
	s	0.810	0.810	0.810	0.810	0.810	0.810	0.810

5.4. Usporedba rezultata i učinkovitosti grupiranja

Algoritam K-sredina uspješno je razdvojio samo skup (5), a ostale nije jer nisu bile ispunjenje sve pretpostavke o oblicima grupa. Grupe skupova (1) i (2) nekonveksnog su oblika, a grupe skupova (3) i (4) su ili nejednakih veličina ili su ukošene, što predstavlja problem za model centroida. Neke je točke, iako pripadaju u stvarnosti jednoj grupi, zbog toga što su udaljenošću bliže ostalim grupama, algoritam K-sredina svrstao u te krive grupe.

Kriterij spajanja velikim dijelom utječe na rezultate kod hijerarhijskih aglomerativnih grupiranja. Jednostruka povezanost u potpunosti je ispravno razdvojila nekonveksne grupe skupova (1) i (2), ali je potpuno pogrešno grupirala skupove (3) i (4) jer kod tih skupova postoje međusobno vrlo bliske točke između prirodnih grupa, a kako jednostruka povezanost gleda najmanju udaljenost između parova točaka, onda ih je spojila u jednu grupu. Taj se efekt jednostruke povezanosti zatim proširio po gotovo cijelom skupu i u konačnici se ostale dvije grupe sastoje od jedne ili dvije točke. Wardova povezanost uspjela je većim dijelom razdvojiti grupe skupa (3) jer Wardova metoda prati promjenu varijance prije i nakon spajanja grupa, a kako grupe skupa (3) nisu ukošene i uzorkovane su iz normalnih razdioba, praćenje varijance je upravo ono što je potrebno napraviti za takve podatke. Osim slučaja sa Wardovom metodom i

skupa (3), kriteriji spajanja nisu uspjeli pravilno razdvojiti grupe skupova (osim (5)) jer nisu idealnog oblika poput grupa skupa (5).

Algoritam DBSCAN u potpunosti je uspio razdvojiti grupe nekonveksnih oblika skupova (1) i (2) jer to omogućava model gustoće točaka, a gusta područja nisu blizu jedna drugima u kojem bi se slučaju mogla svrstati u jedno gusto područje: jednu grupu. Taj je problem vidljiv kod grupiranja skupa (3), gdje je desna prirodna grupa spojena sa srednjom jer gusta područja nisu dovoljno odvojena da bi model gustoće točaka shvatio da se radi o dvije grupe. Grupiranje skupa (4) algoritmom DBSCAN je veoma dobro unatoč tome što su prepoznate 4 grupe, a ne prirodne 3. Također nema puno točaka šuma (svega dvadesetak) u odnosu na grupiranje skupa (3) gdje ih ima nezanemarivo mnogo.

EM algoritam Gaussovih mješavina postiže najbolje rezultate na skupovima (3) i (4) u odnosu na ostale algoritme. Razlog tome jest činjenica da podaci tih skupova odgovaraju generativnom modelu Gaussovih mješavina: točke su uzorkovane iz tri Gaussove distribucije. To je najviše izraženo kod skupa (4) gdje je Gaussova mješavina i EM algoritam savršeno grupirao cijeli skup, što je moguće zbog toga što nije bilo ograničenja na kovarijacijske matrice pa su uzeti u obzir i ukošeni oblici grupa. Grupiranja skupa (1) i (2) nisu dobra zbog neposredne pretpostavke o konveksnim oblicima grupa kod Gaussovih mješavina.

Svi su algoritmi savršeno grupirali skup (5) jer su grupe relativno kompaktne, dobro odvojene i konveksne, što je idealan slučaj grupa. Rezultati grupiranja skupa (6) zanimljivi su jer prikazuju kako se formiraju oblici grupa tijekom izvršenja algoritama i njihove konačne oblike.

Iznosi mjera za vanjsko vrednovanje su očekivani: što grupiranje više sliči prirodnom grupiranju, to su iznosi mjera veći. S druge strane, mjere unutarnjih vrednovanja ne oslikavaju najbolje kvalitetu grupiranja u većini slučajeva. Primjerice, srednja vrijednost siluete za grupiranje skupa (1) hijerarhijskim aglomerativnim grupiranjem s jednostrukom povezanošću niska je unatoč tomu što je grupiranje savršeno obavljeno. Osim toga, srednja vrijednost siluete niža je od grupiranja koja nisu uspjela razdvojiti nekonveksne grupe. Razlog tomu je činjenica da unutarnje mjere pridjeljuju veće iznose grupiranjima s grupama konveksnih oblika, neovisno o tome je li takvo grupiranje ispravno ili nije.

5.4.1. Usporedba učinkovitosti algoritama

Iz dobivenih vremena grupiranja vidljiv je nedostatak hijerarhijskog aglomerativnog grupiranja, a to je velika vremenska složenost što je uzrokovalo relativno velika vremena izvođenja. Iznimka je jednostruka povezanost. Algoritmi DBSCAN i EM algoritam Gaussovih mješavina grupiranje su obavili relativno brzo, što je očekivano jer je u odnosu na broj podataka N složenost manja od kvadratne. Možda neočekivan rezultat su vremena izvođenja algoritma K-sredina, ali postoje razlozi za to. Od svih opisanih algoritama, općenito najučinkovitiji je algoritam K-sredina jer je linearan po svim parametrima. Kod ovih skupova podataka to nije došlo do izražaja jer skupovi nisu jako veliki ($N = 1500$), a kada bi bili nekoliko redova veličine veći, vidjelo bi se kako je algoritam K-sredina brži od ostalih. Uz to je algoritam K-sredina odradio puno iteracija zbog problematičnih oblika grupa, zbog čega je vrijeme bilo malo više nego kod algoritama DBSCAN i EM algoritma.

6. Zaključak

Problem grupiranja može biti vrlo izazovan ako se ne zna puno o podacima. Izbor algoritma grupiranja ovisi o nekoliko stvari koje je potrebno uzeti u obzir: veličina skupa podataka, prostor podataka, odnosno priroda značajki i postoji li znanje o kojem modelu podataka bi mogla biti riječ. Osim odabira algoritma, odabir parametara također može biti izazovan.

Za jako velike skupove podataka i realne značajke algoritam K-sredina i Gaussove mješavine su općenito dobar izbor. Ako skup podataka nije pretjerano velik, vrijedi pokušati pokrenuti hijerarhijsko aglomerativno grupiranje s obzirom da nisu ograničeni na realne značajke (osim Wardove metode). Ako postoji sumnja da su podaci nepravilnih oblika, DBSCAN je dobar izbor.

U svakom slučaju, potrebno je postupcima vrednovanja uvjeriti se u ispravnost grupiranja odabranim algoritmom. Unutarnja vrednovanja mogu biti od koristi, ali potreban je oprez s obzirom da mogu navesti na krive zaključke.

LITERATURA

[1] Đorđe Baralić i Lazar Milenkovic. Surprising examples of manifolds in toric topology! 04 2017.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, i Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. U *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, stranice 108–122, 2013.

[3] M. Julia Carbajal. *Separation and acquisition of two languages in early childhood: A multidisciplinary approach*. Doktorska disertacija, 09 2018.

[4] Hee Im, Shenghua Zhong, i Justin Halberda. Grouping by proximity and the visual impression of approximate number in random dot arrays. *Vision research*, 126, 09 2015. doi: 10.1016/j.visres.2015.08.013.

[5] Korbinian Koch. A friendly introduction to text clustering, 26. Ožujak 2020. URL `https://towardsdatascience.com/a-friendly-introduction-to-text-clustering-fa996bcefd04`.

[6] Jure Leskovec, Anand Rajaraman, i Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd izdanju, 2014. ISBN 1107077230.

[7] Viktor Mysko. Clustering. URL `https://courses.cs.ut.ee/MTAT.08.042/2018_spring/uploads/Main/Lecture4a.pdf`.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, i E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Esha Rajpal. Sap hana text mining functions – part1, 18. Veljača 2018. URL https://blogs.sap.com/2018/02/18/sap-hana-text-mining-functions-part1/.

[10] Wikipedia contributors. Cluster analysis — Wikipedia, the free encyclopedia, 2021. URL https://en.wikipedia.org/wiki/Cluster_analysis. Datum pristupa: 7. 9. 2021.

[11] Wikipedia contributors. Dbscan — Wikipedia, the free encyclopedia, 2021. URL https://en.wikipedia.org/wiki/DBSCAN. Datum pristupa: 7. 9. 2021.

[12] Jan Šnajder. Strojno učenje: Grupiranje. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2021.

Usporedba metoda grupiranja primjenom programskog jezika Python

Sažetak

Grupiranje je postupak particioniranja skupa neoznačenih podatka na podskupove sa međusobno sličnim podacima. Ovaj rad opisuje osnovne pojmove problema grupiranja i uspoređuje četiri različita algoritama grupiranja na jednostavnim skupovima podataka.

Algoritam K-srednjih vrijednosti je najpoznatiji, najučinkovitiji algoritam grupiranja koji se zasniva na modelu centroida. Zahtjeva poznavanje broja grupa i može grupirati konveksne oblike grupa. Hijerarhijsko aglomerativno grupiranje zasniva se na kriteriju spajanja, gradi hijerarhiju grupa spajanjem i nije učinkovit kao K-sredina, ali je primjenjiv na svim vrstama podataka. DBSCAN se zasniva na modelu gustoće točaka koje prepoznaje i grupira gusto raspoređene podatke. DBSCAN je brz, poznaje koncept šuma i može grupirati proizvoljne oblike. EM algoritam Gaussovih mješavina nastoji procijeniti parametre modela Gaussovih mješavina koji podatke promatra kao realizacije više Gaussovih razdioba. EM algoritam je brz i može grupirati konveksne grupe raznih oblika.

Opisani algoritmi su pokrenuti, vrednovani i uspoređeni na 6 različitih, relativno jednostavnih skupova dvodimenzijskih podataka s realnim značajkama.

Ključne riječi: Grupiranje, algoritam K-sredina, model Gaussovih mješavina, DBSCAN, Scikit-learn

Comparison of Clustering Methods Using Python Programming Language

Abstract

Clustering is a process of partitioning a set of unlabeled data into subsets with similar data. This thesis describes basic concepts regarding cluster analysis and compares four different clustering algorithms on simple data sets.

K-means algorithm is the best known, most efficient clustering algorithm based on the centroid model. K-means requires the knowledge of the number of clusters and is able to group convex group shapes. Hierarchical agglomerative clustering is based on linkage criterion and it builds a hierarchy of clusters by merging them, and is not as efficient as K-means, but is applicable to all types of data. DBSCAN is based on the density model which discovers and clusters dense regions of data. DBSCAN is fast, has a notion of noise and is able to cluster arbitrarily shaped clusters. EM algorithm for Gaussian mixture model aims to estimate parameters of Gaussian mixture model that views the data as realizations of several Gaussian distributions. EM algorithm is fast and able to cluster convex groups of various shapes.

Described algorithms were executed, evaluated and compared on 6 different, relatively simple sets of two-dimensional data with real number features.

Keywords: Clustering, K-means, Gaussian mixture model, DBSCAN, Scikit-learn