Grupiranje podataka

Habijanić, Ana

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike i računarstva

Ana Habijanić

Grupiranje podataka

Diplomski rad

Osijek, 2020.

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike i računarstva

Ana Habijanić

Grupiranje podataka

Diplomski rad

Mentorica: izv.prof.dr.sc. Darija Marković

Osijek, 2020.

Sadržaj

1 Uvod

Grupiranje podataka je jedna od najraširenijih tehnika analize podataka. U svim disciplinama, od društvenih znanosti preko biologije do računarstva, ljudi stvaraju prvu ideju o svojim podatcima tako što promatraju smislene grupe unutar njih. Na primjer, biolozi grupiraju gene na temelju sličnosti njihovih ponašanja u različitim eksperimentima; trgovci grupiraju kupce na temelju njihovih profila u svrhu ciljanog marketinga; dok astronomi grupiraju zvijezde na temelju njihove udaljenosti.

Prva stvar koju prirodno trebamo riješiti jest što je to grupiranje? Intuitivno, grupiranje je razvrstavanje elemenata skupa objekata tako da slični objekti završe u istoj grupi dok će različiti objekti biti odvojeni u druge grupe. Naravno, ovaj opis nije dovoljno precizan, stoga ćemo u radu definirati vrste grupiranja i glavne algoritme pomoću kojih ih provodimo, da pobliže objasnimo bit grupiranja. Vidjet ćemo kako izabrati najbolju metodu i dobiti optimalno rješenje.

Kako bismo se upoznali sa svijetom grupiranja, na početku moramo dobro upoznati ključne pojmove koje ćemo spominjati u radu. Sljedeće definicije preuzete su iz [10] i [11]:

Definicija 1. *Neka je* $X = \{x_i \in \mathbb{R}^n : i = 1, \ldots, n\}$ *skup koji sadrži* $n \geq 2$ *elemenata. Rastav skupa* X *na* $1 \leq K \leq n$ *disjunktnih nepraznih podskupova* $C_1, C_2, \ldots, C_K$ *takvih da je*

$$\bigcup_{k=1}^{K} C_k = X, \qquad C_r \cap C_s = \emptyset, \quad r \neq s, \qquad |C_k| \geq 1, \quad k = 1, \ldots, K, \tag{1.1}$$

zovemo K*–particija skupa* X *i označavamo s* $C = \{C_1, C_2, \ldots, C_K\}$*. Elemente particije zovemo klasteri, a skup svih particija skupa* X *sastavljenih od* K *klastera koje zadovoljavaju* (1.1) *označavamo s* $P(X; K)$*.*

Definicija 2. *Funkcija udaljenosti ili metrika je funkcija* $d : X \times X \to \mathbb{R}$ *za koju vrijedi:*

1. $d(x, y) \geq 0, \quad \forall x, y \in X, \quad$ *(nenegativnost),*
2. $d(x, y) = 0$ *ako i samo ako* $x = y, \quad \forall x, y \in X \quad$ *(strogost),*
3. $d(x, y) = d(y, x), \quad \forall x, y \in X \quad$ *(simetričnost),*
4. $d(x, y) + d(y, z) \geq d(x, z), \quad \forall x, y, z \in X \quad$ *(nejednakost trokuta).*

Svojstva 1 i 2 zajedno se nazivaju pozitivna definitnost. Najčešće korištena je Minkowskijeva udaljenost:

$$d(x, y) = \Big(\sum_{j=1}^{n} (x_j - y_j)^p\Big)^{\frac{1}{p}}.$$

Za $p = 1$ *dobivamo L1–udaljenost, za* $p = 2$ *euklidsku udaljenost.*

Definicija 3. *Funkciju* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$*, koja ima svojstvo pozitivne definitnosti*

$$\forall x, y \in \mathbb{R}^n \; d(x, y) \geq 0 \quad i \quad d(x, y) = 0 \iff x = y,$$

zovemo kvazimetrička funkcija.

Dvije najčešće korištene kvazimetričke funkcije na $\mathbb{R}^n$ su kvazimetrička funkcija najmanjih kvadrata (*eng. least squares distance like function*) i L1-metrička funkcija koja se često naziva Manhattan metrička funkcija:

$$d_{LS}(x,y) = ||x-y||_2^2 = (x-y)^T(x-y) = \sum_{i=1}^{n}(x_i-y_i)^2 \quad \text{least squares (LS) kvazimetrička funkcija,}$$

$$d_1(x,y) = ||x-y||_1 = \sum_{i=1}^{n}|x_i-y_i| \quad \text{L1–metrička funkcija (Manhattan metrika).}$$

Općenitiji pojam od udaljenosti je mjera sličnosti, odnosno njoj komplementarna mjera različitosti. Udaljenost možemo tumačiti kao geometrijsku interpretaciju sličnosti ili različitosti. Razlika je u tome što mjera sličnosti (kao i mjera različitosti) nije metrika.

Definicija 4. *Mjera sličnosti definira se kao funkcija* $s : X \times X \to [0,1]$ *za koju vrijedi:*

1. $s(x,x) = 1, \quad \forall x \in X,$
2. $0 \leq s(x,y) \leq 1, \quad \forall x,y \in X,$
3. $s(x,y) = s(y,x), \quad \forall x,y \in X.$

Ako usporedimo svojstva mjere udaljenosti i mjere sličnosti, uočavamo da one nisu suprotne u smislu komplementa, već je jednu moguće preslikati u drugu korištenjem monotono padajuće funkcije. Funkciju udaljenosti d možemo preslikati u sličnost s tako da je:

$$s(x,y) = \frac{1}{1+d(x,y)}.$$

Pretvorba sličnosti u udaljenost je malo teža zbog uvjeta nejednakosti trokuta. Većinom se ove dvije funkcije koriste jednako, ali mjera sličnosti može pružiti dodatnu fleksibilnost.

2 Definicija grupiranja

Grupiranje, također poznato kao *nenadzirano klasificiranje*, je poznata metoda koja se koristi u prepoznavanju uzoraka i rudarenju podataka. Ono ima široki spektar primjena u mnogo područja. Zadani podatci su uglavnom vektori u višedimenzionalnom prostoru. Osnovni cilj grupiranja je razdjeljivanje podataka prema zadanom kriteriju sličnosti uz postizanje velike sličnosti između podataka unutar iste grupe, a male sličnosti između podataka koji pripadaju različitim grupama. Matematički, grupiranje dijeli početni prostor na K dijelova ovisno o nekoj mjeri sličnosti, gdje vrijednost broja K možemo i ne moramo znati unaprijed. Dani skup X tada rastavljamo na njegovu particiju $C = \{C_1, C_2, \ldots, C_K\}$. Stvara se matrica particija $U(X)$ za dani skup X, koji se sastoji od n podataka, $X = \{x_1, x_2, \ldots, x_n\}$, takvih da je

$$\sum_{j=1}^{n} u_{kj} \geq 1, \quad \text{za } k = 1, \ldots, K, \tag{2.1}$$

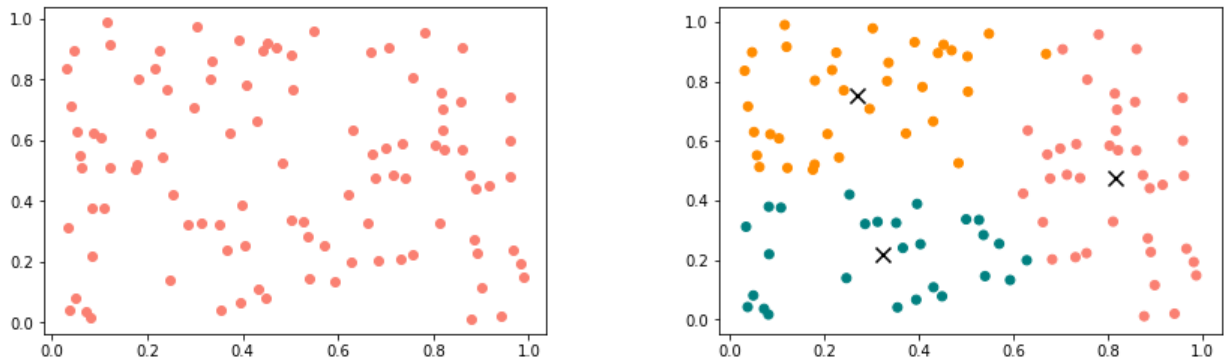
$$\sum_{k=1}^{K} u_{kj} = 1, \quad \text{za } j = 1, \ldots, n, \quad \text{i} \tag{2.2}$$

$$\sum_{k=1}^{K} \sum_{j=1}^{n} u_{kj} = n.$$

Prema [10], elemente matrice particije $U \in \{0, 1\}^{K \times n}$ možemo zapisati kao

$$u_{kj} = \begin{cases} 1, & \text{ako podatak } x_j \text{ pripada klasteru } C_k, \\ 0, & \text{ako podatak } x_j \text{ ne pripada klasteru } C_k. \end{cases}$$

Uvjet (2.1) osigurava da će svaki klaster sadržavati barem jedan podatak, dok uvjet (2.2) osigurava da će svaki podatak x_i pripasti točno jednom klasteru.



(a) Podatci prije grupiranja

(b) Podatci nakon grupiranja

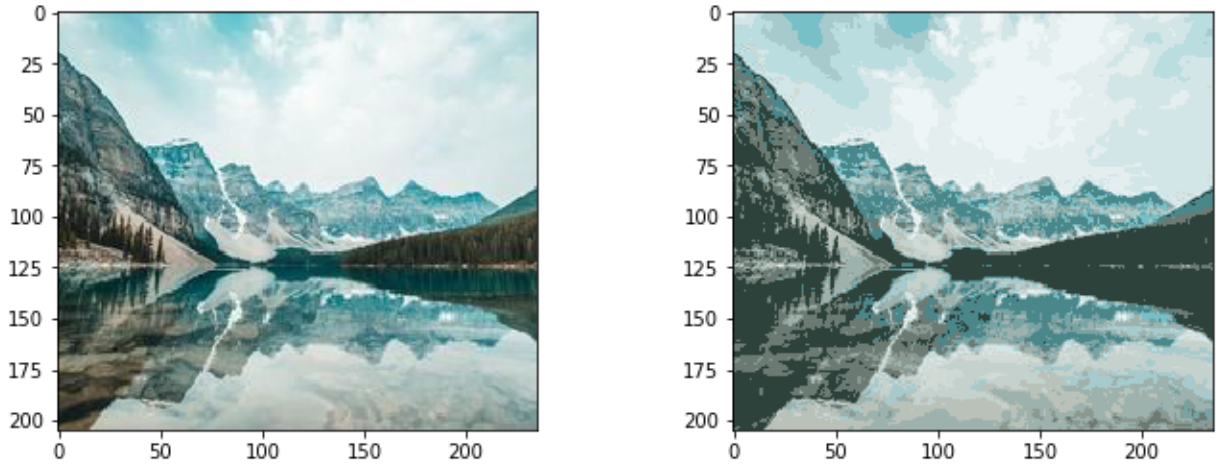
Slika 1: Primjer grupiranja dvodimenzionalnih podataka

3 Primjene grupiranja

Kao što smo na početku rekli, grupiranje ima širok spektar primjena pa ćemo spomenuti neke od najvažnijih (vidi [11]).

1. Istraživanje podataka
   Grupiranje se vrlo često koristi kod istraživanja podataka kako bismo pronašli skrivene strukture u podatcima. Kada podatke grupiramo, možemo ručno označiti klastere, centoridi klastera se mogu tumačiti kao prototipni predstavnici klastera, a za svaki klaster utvrđujemo tipične raspone vrijednosti značajki. Tako možemo podatke opisati na jednostavniji način, omogućava nam uočavanje pravilnosti i sličnosti u podatcima te otkrivanje odnosa među klasterima.

2. Kompresija podataka
   Kod kompresije podataka, grupiranje se koristi za preslikavanje kontinuiranih vrijednosti u diskretne vrijednosti. Na primjer, 24-bitne digitalne slike grupiraju se u 256 klastera te se svaka boja može predstaviti centroidom klastera. Na taj način ostvarujemo kompresiju s 24 bita na 8 po slikovnom elementu. Ovaj postupak naziva se kvantizacija vektora.



(a) Slika planine prije kompresije

(b) Slika planine nakon kompresije

Slika 2: Kompresija slike

3. Predobrada
   Grupiranje se može koristiti kao tehnika predobrade kod nadziranog učenja čiji je cilj smanjenje dimenzionalnosti prostora, tj. smanjenje broja značajki. Smanjenjem dimenzionalnosti uspješno štedimo prostor i vrijeme izvođenja te smanjujemo utjecaj šumova.

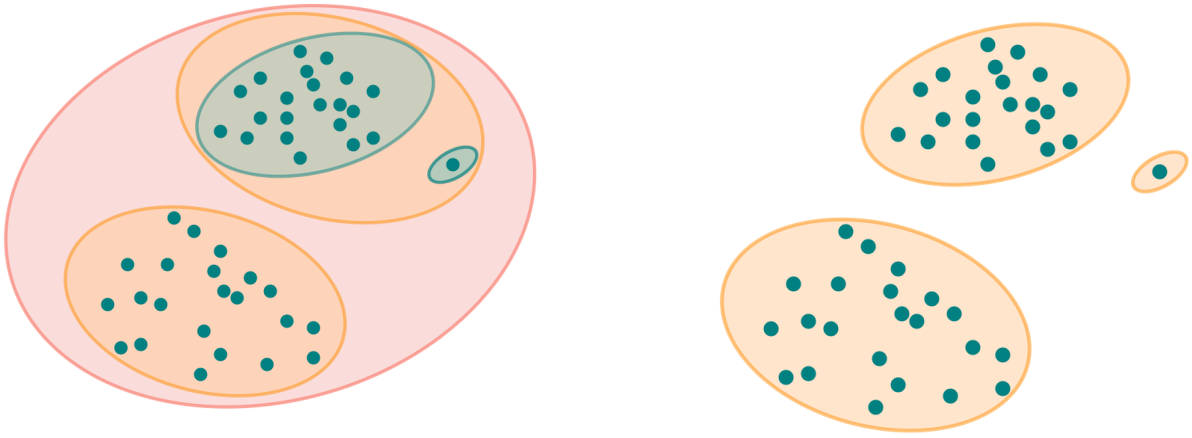
4. Grupiraj i označi
   Kada imamo skup podataka za učenje u kojem je samo mali dio označen, grupiranje možemo koristiti u kombinaciji s nadziranim učenjem kako bismo označili sve podatke u skupu. Način na koji se to izvodi je grupiranje svih podataka, a zatim neoznačene podatke unutar svakog klastera označiti prema već označenim podatcima u tom klasteru. Kao oznaka klastera odabire se ona koja se najčešće pojavljuje u tom klasteru. Grupiraj i označi tehnika je polunadziranog učenja.

4 Vrste grupiranja

Danas postoje mnoge vrste grupiranja. Klasična podjela algoritma za grupiranje je u 4 klase (detaljnije vidi u [6]): hijerarhijsko, particijsko, grupiranje temeljeno na gustoći podataka i grupiranje temeljeno na mreži podataka. Navest ćemo neke od metoda za svaku klasu. Algoritam grupiranja jednostruke povezanosti, algoritam grupiranja prosječnom povezanosti te algoritam grupiranja potpunom povezanošću su metode za hijerarhijsko grupiranje. Algoritam K–srednjih vrijednosti je primjer particijskog grupiranja, DBSCAN predstavlja grupiranje temeljeno na gustoći podataka, dok je STING algoritam metoda grupiranja temeljena na mreži podataka. O svakoj od navedenih metoda ćemo kasnije reći nešto više.

Grupiranje, s druge strane, možemo podijeliti prema čvrstoći granica među klasterima. Postoji tvrdo grupiranje u kojem svaki podatak može pripadati točno jednom klasteru te meko grupiranje u kojem jedan podatak može pripadati u više klastera odjednom, ali s različitim stupnjem ili različitom vjerojatnosti pripadanja.

Također, možemo promatrati različite pristupe s obzirom na kriterij grupiranja. To može biti minimizacija kriterijske funkcije (koristi se kod algoritma K–srednjih vrijednosti), maksimizacija izglednosti (EM algoritam) ili možemo grupirati prema zadanoj mjeri sličnosti ili funkciji udaljenosti (koristi se kod hijerarhijskog grupiranja).



(a) Hijerarhijsko grupiranje

(b) Particijsko grupiranje

Slika 3: Razlika između hijerarhijskog i particijskog grupiranja

4.1 Particijsko grupiranje

Particijsko ili neugniježdeno grupiranje je podjela skupa podataka u klastere koji sadrže slične podatke. Klasteri mogu biti disjunktni (kod tvrdog grupiranja), gdje podatak može pripadati točno jednom klasteru, dok kod mekog grupiranja podatak može pripadati u više klastera s određenom težinom. Klastere uglavnom predstavlja centar klastera ili centroid, odnosno podatak koji sažima opis svih podataka u tom klasteru. Definicija centroida klastera ovisi o vrsti podataka koje proučavamo; na primjer ako imamo zadan skup realnih brojeva, tada će njihov centroid biti aritmetička sredina danog skupa. U slučaju grupiranja dokumenata, centroid može biti lista riječi koja se pojavljuje u nekom minimalnom broju dokumenata unutar klastera. Ako je broj klastera velik, centroidi mogu biti dalje klasterirani kako bi stvorili hijerarhiju unutar skupa. U nastavku ćemo govoriti o popularnim algorit-

mima za particijsko grupiranje, kao što su K–srednjih vrijednosti, K–medoida, neizrazitih C–srednjih vrijednosti i algoritam maksimizacije očekivanja.

4.1.1 Algoritam K–srednjih vrijednosti

Grupiranje metodom K–srednjih vrijednosti (*eng. K-means*) predstavio je 1967. godine James MacQueen. Zbog dobrih rezultata i jednostavnosti, algoritam se i danas često koristi i jedan je od najkorištenijih algoritama grupiranja. On pripada nenadziranom učenju i koristi se u obradi podataka koji nemaju definirane kategorije. To je iterativan algoritam čiji je cilj podjela podataka u K klastera, a kao rezultat osim K klastera, daje i njihove predstavnike takve da je veličina

$$J(C) = \sum_{j=1}^{n} \sum_{k=1}^{K} u_{kj} \times ||x_j - c_k||^2 \tag{4.1}$$

minimalna. Funkcija J naziva se funkcija cilja, c_k je centroid klastera k, a x_j je j-ti podatak. Početni centroidi klastera su nasumično izabrani podatci iz danog skupa X, a početna particija se formira korištenjem principa minimalnih udaljenosti. U sljedećim koracima algoritma centroidi klastera se ažuriraju i postaju aritmetičke sredine elemenata klastera. Postupak particioniranja (taj korak se često naziva *pridruživanje*) i ažuririranja centroida (tzv. *korekcija*) ponavljamo sve dok se ne dogodi jedan od sljedećih događaja:

(a) centroidi klastera se ne mijenjaju kroz iteracije,

(b) vrijednost funkcije cilja J postane manja od zadane tolerancije,

(c) izvršen je maksimalan predviđen broj iteracija.

U nastavku slijedi algoritam K–srednjih vrijednosti.

Algoritam 1 K–SREDNJIH VRIJEDNOSTI

Korak 1: *(Inicijalizacija)* Nasumično izaberi K centroida klastera $c_1, c_2, \ldots, c_K$ od n podataka $x_1, x_2, \ldots, x_n$.

Korak 2: *(Pridruživanje)* Dodijeli podatak x_i, $i = 1, 2, \ldots, n$ klasteru C_k, $k \in \{1, 2, \ldots, K\}$ ako i samo ako

$$||x_i - c_k|| < ||x_i - c_p||, \quad p = 1, 2, \ldots, K, \ k \neq p.$$

Korak 3: *(Korekcija)* Odredi nove centroide klastera $c_1^*, c_2^*, \ldots, c_K^*$ tako da vrijedi:

$$c_k^* = \frac{\sum\limits_{x_i \in C_k} x_i}{n_k}, \quad k = 1, 2, \ldots, K,$$

gdje je n_k broj podataka u klasteru C_k.

Korak 4: Zaustavi algoritam ako je ispunjen neki od zaustavnih kriterija. Inače, $c_k = c_k^*$, $k = 1, 2, \ldots, K$ i vrati se na korak 2.

Općenito, ako algoritam ne pronađe optimalno rješenje u četvrtom koraku, on će završiti zbog izvedenog maksimalnog broja iteracija. Pravila za ažuriranje centroida klastera su dobivena diferenciranjem funkcije cilja J s obzirom na centroide i izjednačavanjem diferencijala

s nulom. Svrha analize u nastavku je minimizacija funkcije J.

$$\frac{\partial J}{\partial c_k} = 2\sum_{j=1}^{n} u_{kj}(x_j - c_k)(-1) = 0, \quad k = 1, 2, \ldots, K, \tag{4.2}$$

$$\sum_{j=1}^{n} u_{kj}x_j - c_k \sum_{j=1}^{n} u_{kj} = 0, \tag{4.3}$$

$$c_k = \frac{\sum_{j=1}^{n} u_{kj}x_j}{\sum_{j=1}^{n} u_{kj}}. \tag{4.4}$$

Primjetimo da je broj $\sum_{j=1}^{n} u_{kj}$ zapravo broj elemenata klastera k, tj. n_k. Stoga, centroide možemo zapisati kao:

$$c_k^* = \frac{\sum_{x_i \in C_k} x_i}{n_k}. \tag{4.5}$$

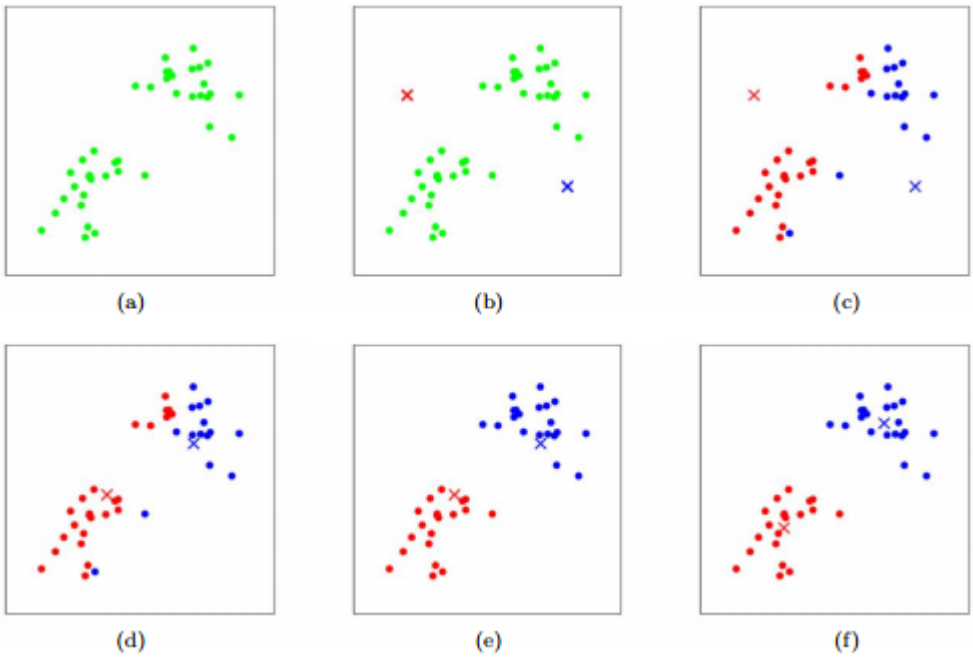
Želimo dobiti minimum funkcije J, stoga nam je potrebno da druga derivacija $\frac{\partial^2 J}{\partial c_k^2}$ bude veća od nule:

$$\frac{\partial^2 J}{\partial c_k^2} = 2\sum_{j=1}^{n} u_{kj}. \tag{4.6}$$

Uočimo kako je desna strana jednakosti pozitivna, što implicira da novi izbori centroida zaista dovode do minimalne vrijednosti funkcije cilja.

Napomena. *Algoritam K–srednjih vrijednosti najčešće koristi LS–kvazimetričku funkciju, uz koju je centroid skupa podataka aritmetička sredina klastera. Ponekad, može se koristiti i L1–kvazimetrička funkcija, uz koju za centroid skupa podataka dobivamo medijan klastera.*

Uočimo kako algoritam, osim izbora za početne centriode, ima dodatan izvor nedeterminističnosti, a to je određivanje kojem će klasteru pripasti podatak koji je jednako udaljen od dva centroida. To se uglavnom rješava dogovorno. Zbog toga treba paziti kod implementacije da se ovaj problem riješi na proizvoljan, ali konzistentan način kako ne bi došlo do beskonačne petlje u algoritmu.



Slika 4: Algoritam K–srednjih vrijednosti kroz iteracije

Odabir centroida

Algoritam K–srednjih vrijednosti pretražuje prostor veličine broja različitih particija od n podataka u K skupova. Postavljaju se pitanja hoće li algoritam konvergirati i je li grupiranje koje je algoritam pronašao optimalno u smislu funkcije cilja J. Može se dokazati da će algoritam sigurno konvergirati. Broj mogućih particija je K^n i konačan je, pa je konačan i broj kombinacija u kojima se c_k nalazi u središtu svojeg klastera. U svakoj iteraciji algoritma se pogreška smanjuje te algoritam pronalazi novo rješenje. S obzirom da je takvih konačno mnogo, algoritam će stati u konačnom broju koraka. Drugo pitanje također ima jednostavan odgovor; lako se pokaže da algoritam daje optimalno rješenje. On je pohlepan algoritam[1] koji će pronaći lokalno optimalno rješenje. O izboru centroida c_k ovisi hoće li algoritam dati globalno optimalno rješenje. Neki od načina izbora početnih centroida (vidi više u [11]):

- Slučajnim odabirom izabrati K podatka kao centroide c_k. Na ovaj način izbjegavamo da centroidi budu smješteni na mjesta u prostoru gdje nema podataka, ali ne rješavamo problem globalne optimizacije. Kod ovakvog načina biranja centroida, problem stvaraju stršeće vrijednosti koje mogu završiti u posebnim klasterima. Iako se čini da je dobro rješenje da stršeće vrijednosti budu u posebnim klasterima, problem je u tome što je broj klastera K ograničen te se one trebaju poklopiti s prirodnim (većinskim) klasterima koji postoje u podatcima.

- Izračunati centroid c cijelog skupa X te mu dodavati manje slučajne vektore kako bismo dobili K centroida c_k. Ovim načinom riješili smo problem stršećih vrijednosti, ali smo ostali na lokalnoj optimizaciji.

- Izračunati prvu glavnu komponentu skupa podataka metodom PCA[2], razdijeliti raspon

[1] Pohlepni pristupi promatra što mu je u danom trenutku najbolje ("lokalno optimalno") učiniti. Pohlepni algoritmi vrlo često ne daju optimalno rješenje, ali mogu dati solidnu aproksimaciju.

[2] PCA (*eng. Principal Component Analysis*) ili analiza glavnih komponenti je statistički postupak za reduciranje dimenzije podataka.

na K jednakih intervala te podatke u K grupa i uzeti centroide tih grupa kao vrijednosti za c_k.

- Nasumično odabrati jedan početan centroid c_k, a svaki sljedeći odrediti tako da je što dalje od ostalih centroida. Ovakav pristup koristi se kod K–means++ algoritma u kojem je vjerojatnost da je podatak x_i novi centroid c_i proporcionalna kvadratu udaljenosti tog podatka od njemu najbližeg, već odabranog centroida c_k.

$$P(c_i = x_i|X) = \frac{||x_i - c_k||^2}{\sum_j ||x_j - c_k||^2}.$$

  Na ovaj način stršeće vrijednosti imaju veću vjerojatnost da budu izabrane za centroide, ali njih je u pravilu manje pa je veća vjerojatnost izbora prosječnog podatka. Dokazano je kako ovakav izbor početnih centoida ubrzava konvergenciju algoritma te znatno smanjuje pogrešku grupiranja.

Kada se početni centroidi određuju nedeterministički, preporučljivo je algoritam pokrenuti više puta kako bi se dobilo rješenje sa što manjom pogreškom. Osim izbora centroida, rješenje algoritma ovisi i o odabranom broju klastera. Taj problem promotrit ćemo kasnije jer je on zajednički svim algoritmima grupiranja.

4.1.2 Algoritam K–medoida

Kako smo vidjeli u prošlom potpoglavlju, algoritam K–srednjih vrijednosti u funkciji cilja koristi euklidsku udaljenost za računanje udaljenosti između podataka. To utječe na izvedivost algoritma jer tada postaje osjetljiv na stršeće vrijednosti te je ograničen na podatke koji se mogu prikazati u vektorskom prostoru. Ponekad nemamo takve podatke, već imamo informaciju o međusobnoj sličnosti parova podataka. Na primjer, zadan može biti skup riječi koje želimo grupirati na temelju sličnosti znakovnih nizova i dobiti klastere grafijski sličnih riječi ili grupirati ljude na temelju jakosti poznanstava pa dobiti grupe ljudi koji se međusobno dobro poznaju. Tada raspolažemo mjerom sličnosti, odnosno mjerom različitosti koja se računa između parova podataka.
Algoritam K–medoida je još jedan particijski algoritam za grupiranje. Medoid je reprezentativni objekt skupa podataka čija je prosječna različitost od svih objekata u klasteru minimalna. K–medoid je poopćenje algoritma K–srednjih vrijednosti u kojem funkciju cilja definiramo pomoću mjere različitosti $\nu(x, x')$ između dva podataka:

$$\hat{J}(C) = \sum_{j=1}^{n} \sum_{k=1}^{K} u_{kj} \nu(x_j, c_k).$$

Mjera različitosti ν (kao i njoj komplementarna mjera sličnosti) općenitija je od euklidske mjere udaljenosti i od bilo koje druge mjere udaljenosti te je time algoritam otporniji na stršeće vrijednosti.

Tehnika grupiranjem K–medoidom grupira skup od n podataka u K klastera, pri čemu je vrijednost broja K poznata.

Koraci algoritma jednaki su koracima algoritma K–srednjih vrijednosti, ali se grupiranje odvija po drugačijem kriteriju. Također, zaustavni kriteriji podudaraju se zaustavnim kriterijima kod algoritma K–srednjih vrijednosti.

Algoritam 2 K–MEDOIDA

Korak 1: *(Inicijalizacija)* Nasumično izaberi K podataka iz skupa od n podataka i postavi ih kao K medoida.

Korak 2: *(Pridruživanje)* Svaki podatak x_i iz skupa dodijeli se najbližem medoidu (najbližem u smislu izabrane mjere sličnosti, ovisno o potrebama i tipu podataka).

Korak 3: *(Korekcija)* Odredi nove medoide klastera i izračunaj pogrešku za novo grupiranje.

Korak 4: Zaustavi algoritam ako je ispunjen neki od zaustavnih kriterija. Inače, ponavljaj korake 2 i 3.

Najčešće korišteni algoritam u ovoj tehnici grupiranja zove se Particioniranje oko medoida (*eng. Partitioning around medoids*, PAM).

(a) K–srednjih vrijednosti (b) K–medoida

Slika 5: Usporedba K–srednjih vrijednosti na slici 5a) i K–medoida na slici 5b)

4.1.3 Algoritam neizrazitih C–srednjih vrijednosti

Tehniku neizrazitih C–srednjih vrijednosti (*eng. fuzzy C–means*) koristimo kada prirodno neki od podataka mogu djelomično pripadati u dva ili više klastera. Neizrazito grupiranje primjenjuje se kod analize slike i signala, medicinske dijagnostike, tomografije, astronomije, kod prepoznavanja govora, u znanosti o okolišu, itd. (detaljnije se može proučiti u [10]). Algoritam neizrazitih C–srednjih vrijednosti pripada mekom grupiranju i to je algoritam kod kojeg element skupa pripada svakom klasteru s nekom određenom težinom. Algoritam je razvio Joe Dunn 1973. godine, a Jim Bezdek ga je unaprijedio 1981. godine. Najviše se koristi u prepoznavanju uzoraka. Zasniva se na minimiziranju funkcije cilja:

$$J(C) = \sum_{j=1}^{c} \sum_{k=1}^{n} (u_{ik})^{\mu} ||c_i - x_k||, \tag{4.7}$$

gdje je n kardinalnost skupa podataka, $\mu \geq 1$ parametar zamućenosti, $u_{ik} \in [0, 1]$ težina koja opisuje u kolikoj mjeri x_k pripada i-tom klasteru, a c_i je centroid i-tog klastera. Za u_{ik} vrijedi

$$\sum_{k=1}^{n} u_{ik} = 1.$$

Iako je rezultat neizrazitog grupiranja djelomično grupiranje, možemo iskoristiti informacije dobivene u matrici particije. Rješenje ovog algoritma, kao i algoritma K–srednjih vrijednosti, može ostati lokalno optimalno ovisno o izboru početnih vrijednosti i potrebno je unaprijed znati broj klastera.

Algoritam 3 NEIZRAZITIH C–SREDNJIH VRIJEDNOSTI

Korak 1: Pomoću matrice particije U izračunaj centroide klastera te spremi u vektor $C = [c_i]$ tako da je:

$$c_i = \frac{\sum_{k=1}^{n} (u_{ik})^{\mu} x_k}{\sum_{k=1}^{n} (u_{ik})^{\mu}}, \quad i = 1, \ldots, c. \tag{4.8}$$

Korak 2: Ažuriraj matricu U tako da je:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{||x_k - c_i||}{||x_k - c_j||} \right)^{\frac{2}{\mu - 1}}}, \quad i = 1, \ldots, c, \ k = 1, \ldots, n. \tag{4.9}$$

Korak 3: Ako je izvršen maksimalni broj iteracija ili postignut kriterij zaustavljanja, zaustavi algoritam. Inače se vrati na korak 1.

4.1.4 Grupiranje temeljeno na distribuciji

Ova kategorija algoritama za grupiranje pretpostavlja da klasteri prate neku specifičnu distribuciju. Algoritam maksimizacije očekivanja (*eng. expectation–maximization algorithm*, EM) je istaknuti primjer ove kategorije.

Tehnika grupiranja maksimizacijom očekivanja je temeljena na miješanom modelu[3]. To je iterativan proces čiji je cilj odrediti parametre distribucije klastera. Osnovna ideja algoritma je podjela n podataka iz danog skupa u K klastera tako da je ukupna vjerojatnost pojave svih klastera maksimalna. Ulazni podatci algoritma su dani skup X, broj klastera K, greška konvergencije E i maksimalan broj iteracija. Izvršavaju se dva koraka u svakoj iteraciji. Prvi je E-korak (*eng. expectation*) u kojem računamo vjerojatnost pripadnosti svake točke svakom klasteru. Drugi korak je M-korak (*eng. maximization*) koji ponovno procjenjuje vektor parametara svake klase nakon ponovne podjele. Algoritam završava nakon izvršenja maksimalnog broja koraka ili kada parametri distribucije konvergiraju. Ova statistička tehnika grupiranja pretpostavlja da su dani klasteri u Gaussovoj (normalnoj) distribuciji, stoga, ako oni to nisu, algoritam neće dati uspješan rezultat grupiranja.

Algoritam 4 EM ALGORITAM

Korak 1: S obzirom na dan skup podataka, odredi početni skup parametara.

Korak 2: *(E–korak)* Koristeći informacije o podatcima, izračunaj vjerojatnosti pripadnosti svakog podatka svakom klasteru.

Korak 3: *(M–korak)* Iskoristi upotpunjene podatke generirane nakon E–koraka za ažuriranje parametara.

Korak 4: Korake 2 i 3 ponavljaj sve do konvergencije.

[3]Miješani model M koji ima K klastera C_k, $k = 1, \ldots, K$ pridružuje vjerojatnost svakom podatku x: $P(x|M) = \sum_{k=1}^{K} W_k \cdot P(x|C_k, M)$, gdje su W_k težine mješavine.

Inicijalizacija algoritma

Neka je zadan broj klastera K. Svaki klaster k je predstavljen s vektorom parametra θ, sastavljenim od centroida c_k, kovarijancom Σ_k i matrice kovarijacije P_k. Ovi parametri opisuju Gaussovu distribuciju: $\theta^{(t)} = (c_k^{(t)}, \Sigma_k^{(t)}, P_k^{(t)}), \quad k = 1, \ldots, K$.

Na početku, za $t = 0$, EM algoritam nasumično generira početne centroide c_k, kovarijancu Σ_k i kovarijacijsku matricu P_k. Nadalje, uzastopnim iteracijama cilj algoritma je procijeniti vektor parametara θ prave distribucije. Drugi način kako započeti EM algoritam bio bi korištenjem klastera dobivenih tehnikom hijerarhijskog grupiranja.

E–korak

U ovom koraku računamo vjerojatnost pripadnosti svakog podatka svakom klasteru $P(C_k|x_i)$. Svaki podatak iz skupa predstavljen je vektorom značajki $x_i,\ i = 1, \ldots, n$. Pripadnost podatka klasteru računa se kao vjerojatnost svake značajke tog podatka u usporedbi sa značajkama ostalih podataka u klasteru C_k.

$$P(C_k|x) = \frac{|\Sigma_k^{(t)}|^{-\frac{1}{2}} e^{n_k} P_k^{(t)}}{\sum_{i=1}^{n} |\Sigma_k^{(t)}|^{-\frac{1}{2}} e^{n_k} P_i^{(t)}}$$

M–korak

Ovaj korak računa parametre distribucije svakog klastera za sljedeći korak. Na početku, aritmetička sredina c_k klastera k računa se kao aritmetička sredina svih podataka iz skupa ovisno o stupnju relevantnosti svakog podatka.

$$c_k^{(t+1)} = \frac{\sum_{i=1}^{n} P(C_k|x_i) x_i}{\sum_{i=1}^{n} P(C_k|x_i)}.$$

Nadalje, koristimo Bayesov teorem kako bismo izračunali kovarijacijsku matricu za sljedeću iteraciju. Slijedi da je $P(A|B) = P(B|A) \cdot P(A) \cdot P(B)$. Sada, zbog uvjetne vjerojatnosti pojave klastera vrijedi da je:

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{n} P(C_k|x_i)(x_i - c_k^{(t)})(x_i - c_k^{(t)})^T}{\sum_{i=1}^{n} P(C_k|x_i)}.$$

Vjerojatnost pojave svakog klastera sada se računa pomoću aritmetičke sredine vjerojatnost C_k ovisno o stupnju relevantnosti svakog podatka iz klastera.

$$P^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} P(C_k|x_i).$$

Ove karakteristike opisuju vektor parametara θ koji predstavlja distribuciju svakog klastera. Ovaj vektor θ koristimo u sljedećoj iteraciji.

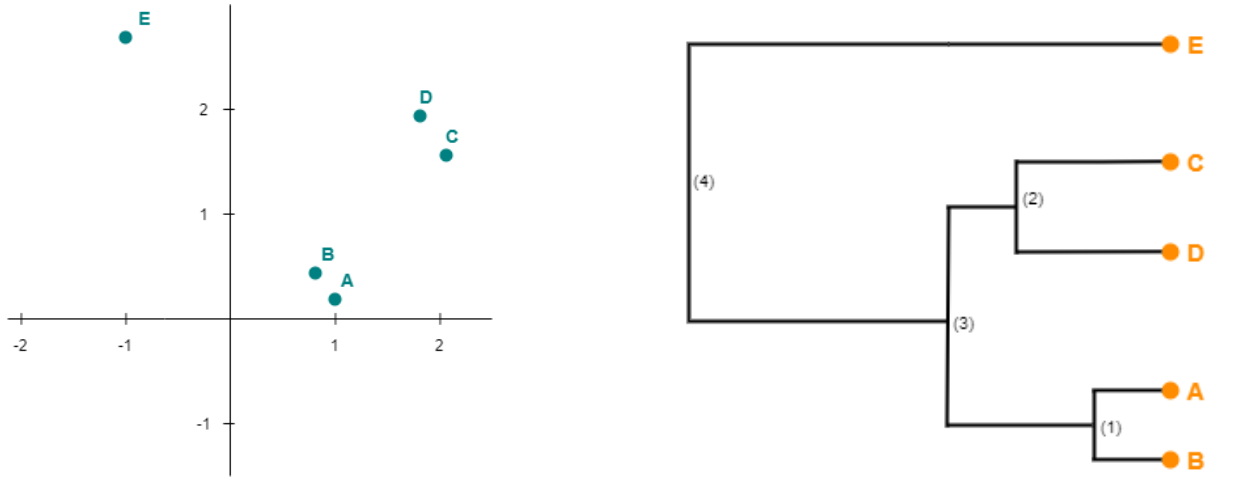
4.2 Hijerarhijsko grupiranje

Iz [11] slijedi da za razliku od particijskog grupiranja, hijerarhijsko grupiranje rezultira hijerarhijom klastera. Hijerarhija klastera prikazuje se dendrogramom. Dendrogram je stablo[4] u kojem listovi odgovaraju podatcima, a vodoravne linije odgovaraju povezivanjima na određenoj udaljenosti. Ovakav prikaz grupiranja je zanimljiv jer se može presjeći na bilo kojoj udaljenosti i dobiti klastere koje bismo dobili particijskim grupiranjem na toj udaljenosti.

Postoje dvije vrste hijerarhijskog grupiranja: aglomerativno i divizno. Aglomerativno grupiranje je "bottom-up" pristup u kojem se kreće od jednog podatka u svakom klasteru pa se postepeno spajaju parovi klastera dok svi podatci ne budu u istom klasteru. Suprotno, divizno grupiranje je "top-down" pristup koji pretpostavlja da svi podatci pripadaju istom klasteru, pa se postepeno razdvajaju po slojevima hijerarhije. Ovakvo spajanje i razdvajanje klastera određeno je pohlepnim načinom.

Za razliku od algoritma K–srednjih vrijednosti i EM-algoritma, hijerarhijsko grupiranje nema teorijsku osnovu te je heuristički postupak.

Hijerarhijsko grupiranje provodi se pomoću funkcije udaljenosti ili mjere sličnosti, s ciljem pronalaska klastera podataka koji su najsličniji jedni drugima.



(a) Skup podataka

(b) Dendrogram skupa podataka sa slike 6(a)

Slika 6: Skup podataka i dendrogram tog skupa

4.2.1 Hijerarhijsko aglomerativno grupiranje

Najčešće korišten algoritam hijerarhijskog grupiranja je algoritam hijerarhijskog aglomerativnog grupiranja (HAC). Kao što smo spomenuli, on je "bottom-up" algoritam. Počinje tako da je svaki podatak u svojem klasteru, a potom se spajaju po dva najbliža klastera sve dok ne dođemo do zadanog broja klastera K (saznaj više u [11]).

Algoritam 5 HAC

Inicijalizacija: Postavi svaki podatak $x_i \in X, i = 1, \ldots, n$ u zasebni klaster.
Korak 1: Spoji dva najbliža klastera.
Korak 1 se ponavlja sve dok ne dođemo do danog broja klastera K.

[4]Stablo je povezan graf bez ciklusa.

Kada je $K = 1$ rezultat algoritma je potpuni dendrogram koji se može presjeći na bilo kojoj udaljenosti. U koraku koji se ponavlja, algoritam pronalazi par najbližih klastera. Potrebno je definirati udaljenost klastera. Općenito, udaljenost skupova A i B može se definirati na više načina:

$$D_c(A, B) = d(c_A, c_B) \quad \text{udaljenost centroida } c_A, c_B \text{ skupova,}$$

$$D_{min}(A, B) = \min_{a \in A, b \in B} d(a, b) \quad \text{minimalna udaljenost,}$$

$$D_{max}(A, B) = \max_{a \in A, b \in B} d(a, b) \quad \text{maksimalna udaljenost,}$$

$$D_{avg}(A, B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b) \quad \text{prosječna udaljenost,}$$

gdje $|A|$ i $|B|$ predstavljaju broj elemenata skupa A, odnosno B.

Udaljenost D_{min} između dva klastera je zapravo najmanja udaljenost između podataka u tim klasterima. Tada dobivamo grupiranje temeljem jednostruke povezanosti. Za razliku od D_{min}, definira se D_{max} kao najveća udaljenost podataka klastera. Takvo grupiranje naziva se potpuno povezano. Ove dvije udaljenosti dat će slične rezultate ako su dani kompaktni i dobro odvojeni klasteri. Inače može doći do značajnijih razlika; rezultat jednostrukog povezivanja bit će dugi, ulančani klasteri dok će se manji i zbijeni klasteri dobiti potpunim povezivanjem.

Rezultate jednostrukog i potpunog povezivanja možemo povezati s teorijom grafova. Naime, spajanje dva klastera C_i i C_j odgovara uvođenju brida između odgovarajućih podataka u tim klasterima. Ako koristimo jednostruko povezivanje, novi brid bit će između dva najbliža podatka iz ta dva klastera. Budući da se bridovi stvaraju između podataka različitih klastera, a nikad među podatcima iz istog klastera, rezultat će biti stablo. U slučaju kada je $K = 1$, algoritam HAC generira minimalno razapinjuće stablo[5]. Obratno, potpuno povezan graf dobit ćemo kod potpunog povezivanja jer spajanje dva klastera odgovara uvođenju bridova između svih parova podataka.

Jednostruko i potpuno povezivanje dva su rubna slučaja izračuna udaljenosti između klastera i osjetljiviji su na šum. Bolje rješenje dobije se koristeći prosječnu udaljenost D_{avg}. Slično, postoji i D_c, mjera udaljenosti centroida klastera. Ona je računalno najjednostavnija mjera, ali je ograničena na udaljenosti definirane u vektorskom prostoru. Ako imamo podatke koji nisu prilagođeni za računanje u vektorskom prostoru, prednost ima mjera D_{avg} koju možemo primijeniti na bilo koju mjeru sličnosti.

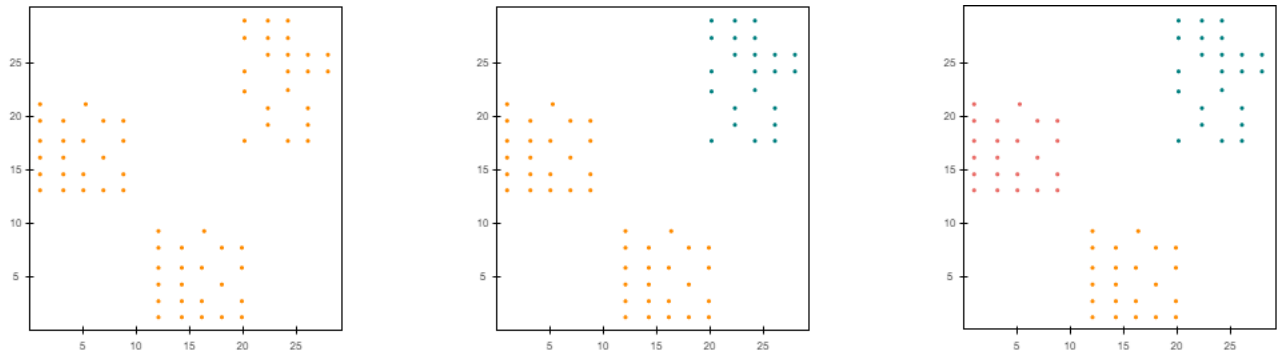
4.2.2 Hijerarhijsko divizno grupiranje

Kako je već objašnjeno u radu, divizno grupiranje koristi "top-down" pristup koji je suprotan aglomerativnom grupiranju.

Algoritam 6 DIVIZNO GRUPIRANJE

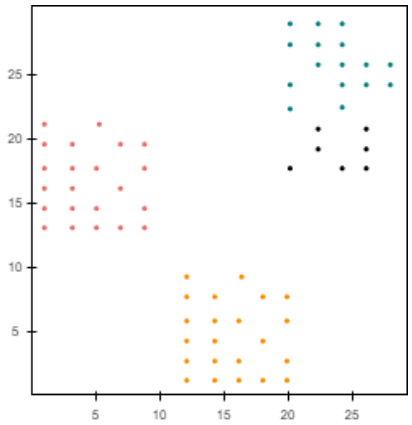
Korak 1: Na početku, svi podatci pripadaju istom klasteru.
Korak 2: Podijeli klaster na dva najmanje slična klastera.
Korak 3: Nastavi rekurzivno stvarati nove klastere sve dok se ne postigne željeni broj klastera.

[5]Minimalno razapinjuće stablo je stablo koje povezuje sve vrhove nekog (težinskog) grafa, pri čemu je ukupna suma težina svih bridova minimalna.



Slika 7: Prikaz postupka diviznog grupiranja u tri klastera

Slika 7 prikazuje tri vidno međusobno udaljena skupa podataka. Zato smo zaustavili algoritam nakon dobivanja tri klastera. Međutim, slika 8 prikazuje što bi se dogodilo kada bismo nastavili dijeliti klastere.

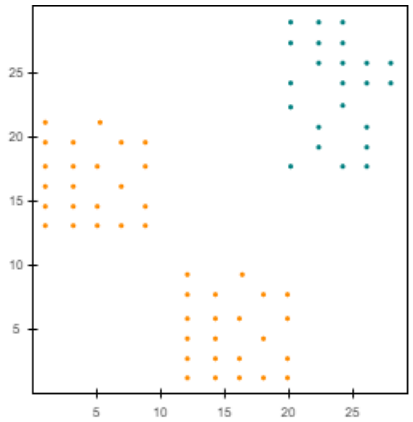


Slika 8: Prikaz diviznog grupiranja u četiri klastera

Odabir klastera za dijeljenje

Prvi problem ovog algoritma je kako ćemo odabrati klaster koji ćemo sljedeći podijeliti. Prvo ćemo izračunati sumu kvadratnih grešaka svakog klastera te odabrati onaj koji ima najveću vrijednost.

U našem primjeru (slika 9), trenutno su podatci podijeljeni u dva klastera. Kako bismo podijelili na tri klastera, moramo pronaći sumu kvadratnih grešaka za svaku točku u narančastom i zelenom klasteru.



Slika 9: Prikaz diviznog grupiranja u dva klastera

Klaster s najvećom kvadratnom greškom se razdvaja u dva klastera, tako stvarajući novi klaster. Na slici 9 vidljivo je kako narančasti klaster ima najveću kvadratnu grešku te se on razdvaja u dva klastera stvarajući tako ukupno tri klastera. Detaljnije o diviznom grupiranju može se pronaći u [8].

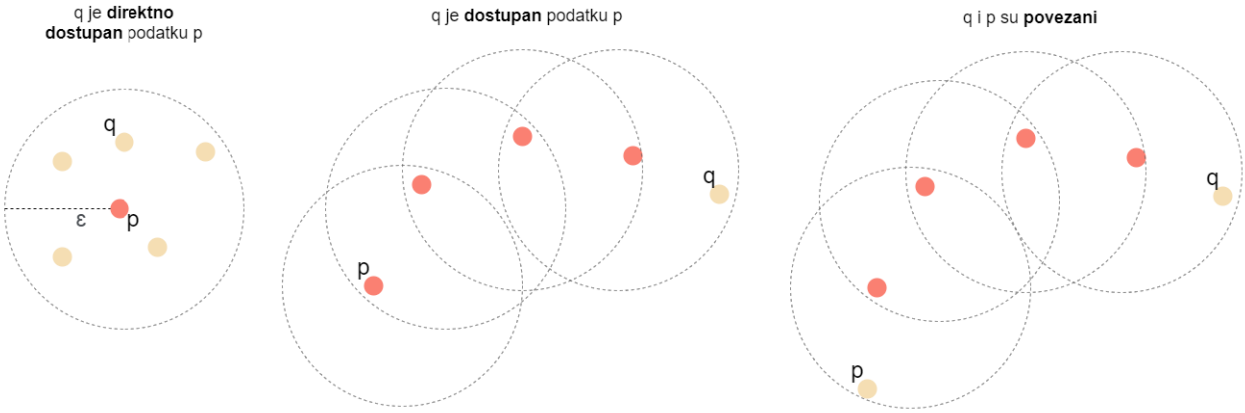
4.3 Grupiranje temeljeno na gustoći podataka

Algoritmi za grupiranje temeljeni na gustoći podataka su vrsta tehnike grupiranja koja u osnovi primjenjuje kriterij lokalnog klastera. Klastere promatramo kao regije u prostoru u kojem podatci tvore gusto područje i razdvojeni su regijama s malom gustoćom podataka. Guste regije podataka ponekad mogu stvoriti neke proizvoljne oblike i podatci mogu biti nasumično raspoređeni unutar te regije. Stoga, algoritmi temeljeni na gustoći podataka mogu lako prepoznati klastere bilo kakvih oblika, ali se oslanjaju na uvjetu da podatci unutar određenog klastera čine stisnutu regiju. Na primjer, za rudarenje podataka je traženje stršećih vrijednosti važnije od pronalaska običnih slučajeva. Više o ovoj vrsti grupiranja može se pronaći u [1], [5] i [12].

Postoje mnogi algoritmi temeljeni na gustoći podataka:

1. DBSCAN algoritam (*eng. Density-based spatial clustering of application with noise*)
   S obzirom na dani skup podataka, algoritam grupira podatke koji su bliski jedni drugima tako da stršeće vrijednosti ostanu u području s malom gustoćom podataka. Uspješno pronalazi proizvoljne oblike sve dok klasteri stvaraju gustu regiju. DBSCAN algoritam je temeljen na konceptu dostupnosti temeljenoj na gustoći, koju definiramo na sljedeći način:
   Podatak q je direktno dostupan (*eng. direct densitiy-reachable*, DDR) podatku p ako je p udaljen od q za najviše ϵ te ako postoji dovoljan broj podataka oko p tako da može biti stvoren klaster oko p i q. Uočimo da relacija direktne dostupnosti nije simetrična, tj. ako je p direktno dostupan podatku q, ne mora značiti da je q direktno dostupan podatku p.
   Podatak q je dostupan (*eng. density reachable*, DR) podatku p ako postoji niz podatka $p_1, \ldots, p_n$, $p_1 = p, p_n = q$, gdje je svaki p_{i+1} direktno dostupan (DDR) podatku p_i. Također, relacija nije simetrična jer podatak p može biti dostupan podatku q, ali q može ležati na rubu klastera i tada neće imati dovoljan broj susjeda da se broji kao pravi element klastera.
   Uz to, definiramo i povezanost baziranu na gustoći (*eng. density-connectedness*, DC)

kao: dva podatka p i q su povezana ako postoji niz podataka $o_1, o_2, \ldots, o_n$ takvi da su o_1 i p dostupni, o_2 i o_1 dostupni, o_3 i o_2 dostupni,..., te su o_n i q dostupni. Klaster



Slika 10: Direktna dostupnost, dostupnost i povezanost podataka

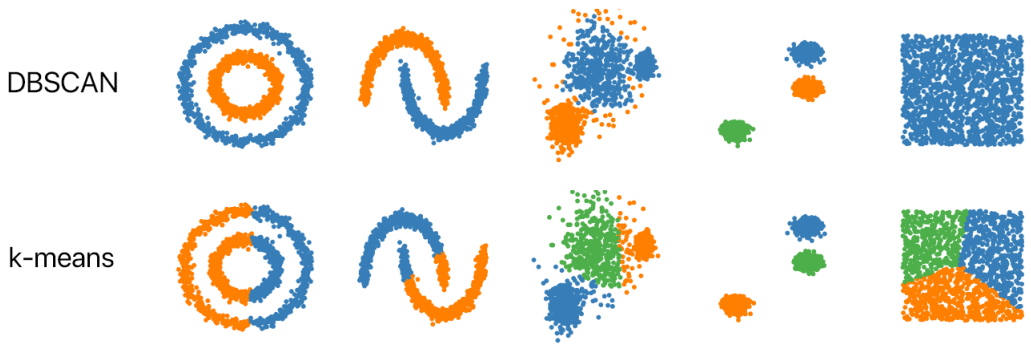
pronađen DBSCAN algoritmom mora zadovoljavati dva uvjeta: svi podatci unutar određenog klastera moraju biti međusobno povezani te ako je podatak povezan s drugim podatkom u klasteru, tada je on uključen u strukturu klastera.

U algoritmu imamo dva parametra:

ϵ udaljenost potrebna za računanje direktne dostupnosti,

$minPts$ minimalni broj podataka potrebnih za formiranje klastera.

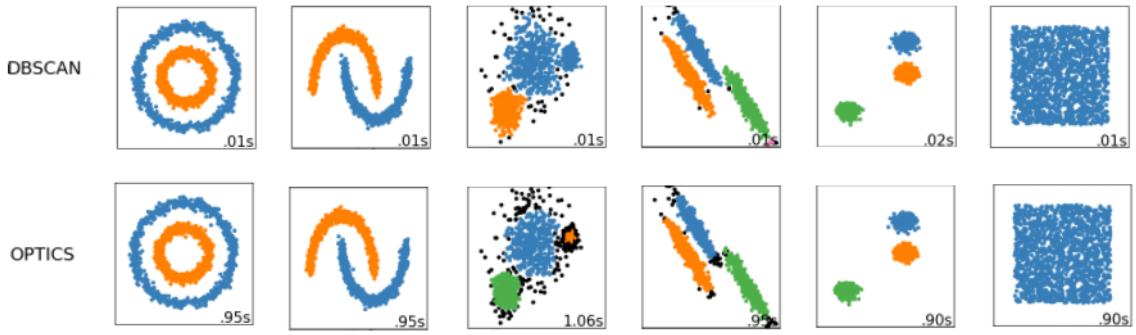
Početni (*eng. seed*) podatak je nasumično izabran podatak koji nije posjećen. DBSCAN algoritam počinje s takvim podatkom. Tada se formira ϵ-susjedstvo početne točke i provjera se njegova veličina (broj susjeda). Ako ima dovoljan broj podataka, stvara se klaster koji sadrži sve podatke iz tog susjedstva. Ako susjedstvo nije dovoljno veliko, podatak označavamo kao stršeću vrijednost. Taj podatak može kasnije biti pronađen u drugoj ϵ-okolini s dovoljnim brojem susjeda te na taj način biti smješten u klaster. Kada podatak uključimo u klaster, njegovo ϵ-susjedstvo se također uključuje u klaster pomoću koncepta povezanosti. Ovaj postupak ponavljamo sve dok nema više podataka koji se mogu uključiti. Tada algoritam počinje s novim početnim podatkom i završava kada takvih podataka više nema.



Slika 11: Razlika DBSCAN i algoritma K–srednjih vrijednosti

2. GDBSCAN algoritam (*eng. Generalized density-based spatial clustering of application with noise*)
   GDBSCAN je generalizirana verzija DBSCAN algoritma koja proširuje definiciju gustoće podataka. Stoga može biti primijenjena na klastere koji imaju različite oblike, kao i dvodimenzionalne poligone.

3. OPTICS algoritam (*eng. Ordering points to identify the clustering structure*)
   Ovo je algoritam koji koristi hijerarhijsko grupiranje temeljeno na gustoći. Za razliku od DBSCAN algoritma koji otkriva klastere koji imaju gustoću koju je korisnik definirao, OPTICS proizvodi hijerarhijsku strukturu danog skupa ovisnu o gustoći. Graf koji nastaje ovim algoritmom prikazuje klastere različite gustoće kao i hijerarhijske klastere.



Slika 12: Razlika DBSCAN i OPTICS algoritma

4.4 Grupiranje temeljeno na mreži podataka

Slično kao grupiranje temeljeno na gustoći, grupiranje temeljeno na mreži podataka često se koristi za određivanje klastera u velikim višedimenzionalnim prostorima. Ponovno klastere promatramo kao guste regije. Detaljnije se može proučiti u [1] i [5].

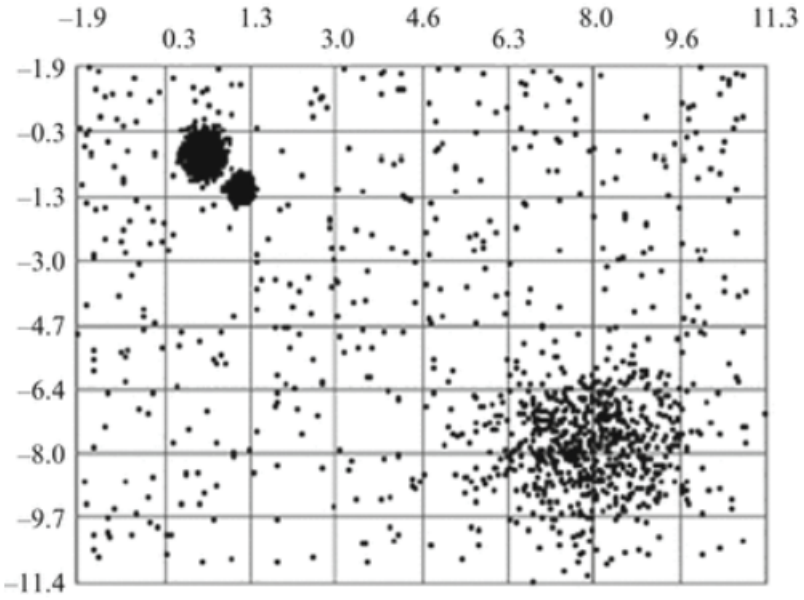
Vremenska složenost većinskih tehnika grupiranja linearno ovisi o veličini danog skupa. Glavna prednost grupiranja temeljenog na mreži je sposobnost rada s velikim skupovima podataka. Osnovna razlika grupiranja temeljenog na mreži od onog temeljenog na gustoći podataka je što ovi algoritmi ne rade sa skupovnim podatcima već s okolnim prostorom. Općenito, ove korake koriste tipični algoritmi temeljeni na mreži podataka:

Algoritam 7 Grupiranje temeljeno na mreži podataka

Korak 1: Generiraj strukture mreže. Ovo se može postići dijeljenjem prostora podataka na konačan broj mreža.
Korak 2: Izračunaj gustoću mreže kao ukupan broj podataka unutar te mreže.
Korak 3: Sortiraj mreže po njihovim gustoćama.
Korak 4: Izračunaj centroide klastera.
Korak 5: Prijeđi na susjednu mrežu.

Često korišten algoritam u ovoj tehnici grupiranja je STING (*eng. Statistical information grid*). On se uglavnom koristi za grupiranje prostornih baza podataka. Prostor podataka prvo podijelimo na mrežu. Ove mreže predstavljene su hijerarhijskom strukturom. Korijen hijerarhije je na nivou 1 i njegova djeca su na sljedećim nivoima. Ćelija na nivou i sadrži

uniju područja sve njegove djece na nivou $i + 1$. U slučaju algoritma STING, svaka ćelija ima 4 djece. Dakle, svako dijete predstavlja četvrtinu područja roditeljske ćelije. STING se jedino može koristiti u slučaju dvodimenzionalnog prostora.



Slika 13: Primjer mreže podataka

4.5 Provjera klastera

Kod svih je metoda grupiranja broj klastera, odnosno parametar K, potrebno odrediti unaprijed. U slučaju hijerarhijskog grupiranja, možemo odabrati $K = 1$ i izgraditi čitav dendrogram, ali se problem očituje u kojem ćemo koraku napraviti presijecanje. Jedan od glavnih problema grupiranja je odabir broja klastera. Idealno, broj klastera odgovara broju prirodnih klastera u skupu podataka, ali taj podatak je uglavnom nepoznat.

Parametar K nazivamo hiperparametar: to je parametar složenosti modela koji se ne ugađa učenjem. Njega ne možemo optimizirati tako da minimiziramo funkciju cilja. Funkcija cilja monotono opada s porastom K te doseže minimum za $K = n$, ali tako smo došli do prenaučenosti modela. Odabir optimalnog broja K znači odabrati optimalnu složenost modela, tj. složenost za koju je sposobnost generalizacije najveća.

Moramo napomenuti kako problem traženja optimalnog broja klastera spada u NP-teške probleme. Najčešće rješenje dobiva se ispitivanjem različitih pokazatelja koje nazivamo indeksi. U nekim jednostavnijim slučajevima je broj klastera prirodno određen, kao što je grupiranje studenata u $K = 5$ klastera ovisno o uspjehu na studiju ili kao što je kvantizacija boja. Ako to nije slučaj, broj klastera na koji grupiramo skup X nije poznat te tada moramo potražiti particiju koja se sastoji od klastera koji su interno što kompaktniji, a eksterno što bolje međusobno razdvojeni. Takva particija ima najprikladniji broj klastera.

Znamo da vrijednost funkcije cilja ne raste povećanjem broja klastera, zato je moguće tražiti takvu optimalnu particiju za koju vrijednost funkcije cilja naglo opada. To nije egzaktan kriterij, ali s drugim kriterijima nas može dovesti do optimalnog rješenja.

Navest ćemo nekoliko najpoznatijih indeksa.

4.5.1 Calinski-Harbasz (CH) indeks

CH indeks definira se tako da interno kompaktnija particija čiji su klasteri međusobno dobro razdvojeni ima veću CH vrijednost.

Funkciju cilja J zapisat ćemo kao:

$$J(C) = \sum_{k=1}^{K} \sum_{x \in C_k} ||c_k - x||_2^2.$$

Vrijednost funkcije J na optimalnoj particiji pokazuje ukupno rasipanje elemenata klastera $C_1, \ldots, C_K$ te particije do njihovih centroida $c_1, \ldots, c_K$. Kako smo spomenuli, smanjenjem vrijednosti funkcije J smanjuje se i rasipanje, pa su klasteri interno kompaktniji.

Iz tog razloga je CH indeks optimalne particije C^* obrnuto proporcionalan funkciji cilja $J(C^*)$. Za određivanje CH indeksa definirat ćemo još jednu funkciju G:

$$G(C) = \sum_{k=1}^{K} m_k ||c_k - c||_2^2,$$

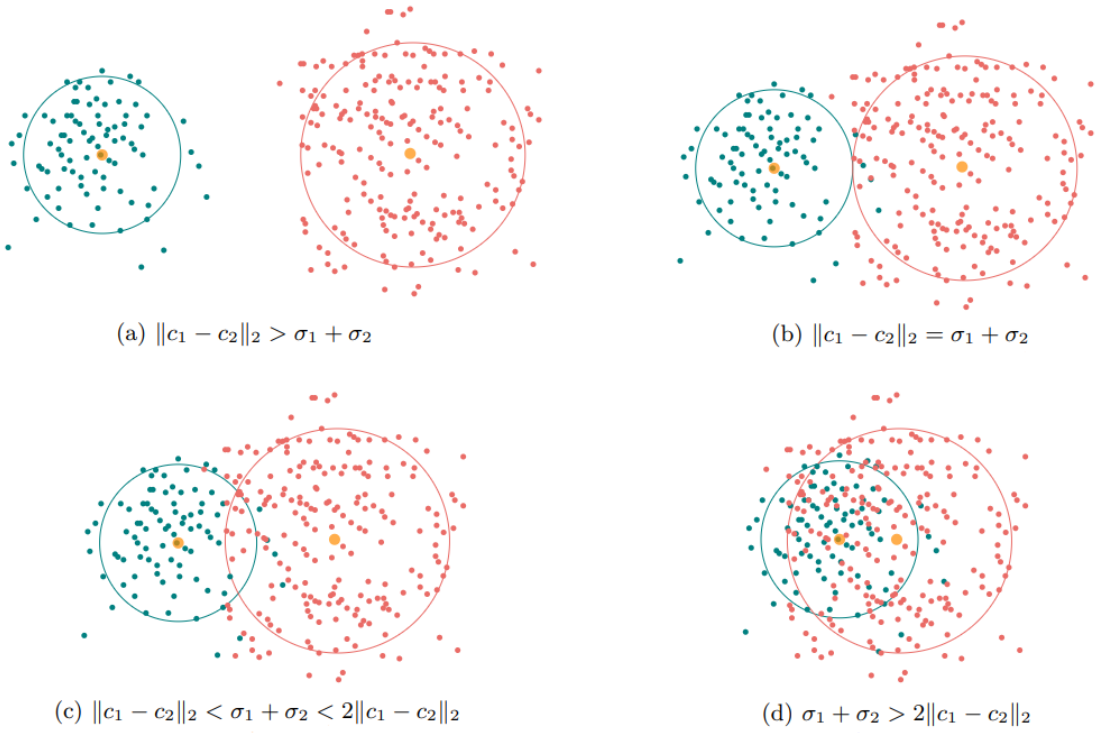
gdje je $m_k = |C_k|$ broj elemenata particije C_k, a $c = \frac{1}{n} \sum_{i=1}^{n} x_i$ centroid skupa X. Vrijednost funkcije G na particiji C govori o ukupnoj težinskoj razdvojenosti centroida $c_1, \ldots, c_K$ klastera $C_1, \ldots, C_K$. Povećanjem vrijednosti funkcije G, povećava se udaljenost centroida c_k do centroida cijelog skupa c. Tada su i centroidi c_k međusobno maksimalno udaljeni. Zato je CH indeks optimalne particije C^* proporcionalan vrijednosti funkcije $G(C^*)$. Stoga, CH indeks particije C^* definiramo kao:

$$CH(K) = \frac{G(C^*)/(K-1)}{J(C^*)/(n-K)}.$$

4.5.2 Davies-Bouldin (DB) indeks

DB indeks definiramo tako da interno kompaktnije particije čiji su klasteri međusobno bolje razdvojeni imaju manju DB vrijednost.

Neka je zadana točka $c \in \mathbb{R}^2$ u ravnini oko koje primjenom Gaussove normalne distribucije s varijancom σ^2 se generira n slučajnih točaka x_i. Ovakav skup točaka naziva se sferičan skup podataka i označit ćemo ga s X. Iz statistike znamo da se u krugu $K(c, \sigma)$ sa središtem u točki c i radijusom σ (standardna devijacija) nalazi oko 68% točaka skupa X. Takav krug naziva se glavni krug skupa podataka X.



Slika 14: Odnos dva sferična skupa podataka

Neka su za dvije različite točke $c_1, c_2 \in \mathbb{R}^2$ i dvije različite varijance σ_1^2, σ_2^2 generirana dva sferična skupa podataka X_1, X_2 i neka su $K_1(c_1, \sigma_1), K_2(c_2, \sigma_2)$ njihovi odgovarajući glavni krugovi. Na slici 14 prikazani su mogući odnosi skupova X_1 i X_2 s obzirom na međusobni položaj njihovih glavnih krugova $K_1(c_1, \sigma_1)$ i $K_2(c_2, \sigma_2)$. Na slici 14a) vidimo skupove X_1, X_2 čiji se glavni krugovi ne sijeku i za njih vrijedi $||c_1 - c_2||_2 > \sigma_1 + \sigma_2$, a na slici 14b) prikazani su skupovi čiji se krugovi dodiruju i vrijedi $||c_1 - c_2||_2 = \sigma_1 + \sigma_2$. Stoga, možemo reći da se glavni krugovi $K_1(c_1, \sigma_1)$ i $K_2(c_2, \sigma_2)$ skupova X_1 i X_2 presijecaju ako vrijedi

$$||c_1 - c_2||_2 \leq \sigma_1 + \sigma_2,$$

odnosno da su glavni krugovi razdvojeni ako vrijedi

$$\frac{\sigma_1 + \sigma_2}{||c_1 - c_2||_2} < 1.$$

Promotrimo sada optimalnu particiju C skupa X s klasterima $C_1, \ldots, C_K$ i njihovim centroidima $c_1, \ldots, c_K$. Pogledajmo jedan klaster c_k i njegov odnos prema ostalim klasterima. Veličinom

$$D_k = \max_{s \neq k} \frac{\sigma_k + \sigma_s}{||c_k - c_s||_2} \tag{4.10}$$

zadano je najveće moguće preklapanje klastera C_k s nekim drugim klasterom. Pri tome su

$$\sigma_k^2 := \frac{1}{|C_k|} \sum_{x \in C_k} ||c_k - x||_2^2, \quad k = 1, \ldots, K.$$

Broj

$$\frac{1}{K}(D_1 + \cdots + D_K) \tag{4.11}$$

prosjek je brojeva (4.10) i predstavlja mjeru interne kompaktnosti i eksterne razdvojenosti klastera u particiji. Što je broj (4.11) manji, klasteri su bolje razdvojeni i kompaktniji. Stoga, DB indeks optimalne partcije C skupa X s klasterima $C_1, \ldots, C_K$ i njihovim centroidima $c_1, \ldots, c_K$ definiramo kao

$$DB(K) = \frac{1}{K} \sum_{k=1}^{K} \max_{s \neq k} \frac{\sigma_k + \sigma_s}{||c_k - c_s||_2}, \tag{4.12}$$

gdje je $\sigma_k^2 := \dfrac{1}{|C_k|} \displaystyle\sum_{x \in C_k} ||c_k - x||_2^2$.

4.5.3 Kriterij širine Silhouette (SWC)

Kriterij širine Silhouette veoma je popularan u klaster analizi i njegovim primjenama. Za optimalnu K–particiju s klasterima $C_1, \ldots, C_K$, SWC definiramo kao: za svaki $x_i \in X \cap C_r$ računamo brojeve

$$\alpha_{ir} = \frac{1}{|C_r| \sum_{a \in C_r} d(x_i, a)}, \quad \beta_{is} = \min_{s \neq r} \frac{1}{|C_s|} \sum_{b \in C_s} d(x_i, b), \tag{4.13}$$

a SWC indeks je definiran s

$$SWC(K) = \frac{1}{n} \sum_{i=1}^{n} \frac{\beta_{is} - \alpha_{ir}}{\max\{\beta_{is}, \alpha_{ir}\}}.$$

Veći SWC broj dat će klasteri koji su kompaktniji i bolje separirani.

Računanje SWC indeksa ima dugotrajnu numeričku proceduru, pa se često upotrebljava pojednostavljeni kriterij širine Silhouette (SSC). On koristi udaljenost od podataka $x_i \in X \cap C_r$ do centroida $c_q, \ldots, c_K$ umjesto prosječne vrijednosti (4.13)

$$\alpha_{ir} = d(x_i, c_r), \quad \beta_{is} = \min_{s \neq r} d(x_i, c_s), \quad SSC(K) = \frac{1}{n} \sum_{i=1}^{n} \frac{\beta_{is} - \alpha_{ir}}{\max\{\beta_{is}, \alpha_{ir}\}}.$$

Literatura

[1] S. Bandyopadhyay, S. Saha, *Unsupervised Classification*, Springer-Verlag, Berlin Heidelberg, 2013.

[2] R. B. Bapat, *Graphs and Matrices*, Springer-Verlag, London, 2010.

[3] L. Buitinck i dr., *API design for machine learning software: experiences from the scikit-learn project*, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, str. 108–122.

[4] K. Devčić, I. Tonković Pražić, Ž. Župan, *Klaster analiza: primjena u marketinškim istraživanjima*, 3 (2012), Zbornik radova Međimurskog veleučilišta u Čakovcu, Preuzeto s: `https://hrcak.srce.hr/83433`, str. 15–22.

[5] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, 2012, str. 443–495.

[6] A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., USA, 1988.

[7] G.J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, Wiley Series in Probability i Statistics, 2007.

[8] A. Patnaik, P. Bhuyan, K. V. Rao, *Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets*, AEJ - Alexandria Engineering Journal 55 (2016), str. 407–418.

[9] F. Pedregosa i dr., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research 12 (2011), str. 2825–2830.

[10] R. Scitovski, K. Sabo, *Klaster analiza i prepoznavanje geometrijskih objekata*, Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku, Osijek, 2020.

[11] J. Snajder, B. Dalbelo Bašić, *Strojno učenje*, Skripta, dostupna na: `https://www.fer.unizg.hr/_download/repository/StrojnoUcenje.pdf`.

[12] P.N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison-Wesley Longman Publishing Co., Inc., USA, 2005.

[13] U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and computing 17.4 (2007), str. 395–416.

Sažetak

Cilj ovog rada je upoznavanje s tehnikama grupiranja podataka. Grupiranje podataka je proces u kojem od grupe različitih objekata stvaramo klastere sličnih objekata.

Postoje različite vrste grupiranja, ovisno o podatcima s kojima radimo. Četiri glavne vrste su: particijsko i hijerarhijsko grupiranje te grupiranje temeljeno na gustoći ili mreži podataka.

Particijsko grupiranje grupira podatke u klastere koje čine slični podatci. Najznačajniji algoritam ove skupine je algoritam K–srednjih vrijednosti. Nastavno na njega, nastale su i druge inače kao što su algoritmi K–medoida i neizrazitih C–srednjih vrijednosti.

Hijerarhijsko grupiranje, s druge strane, stvara hijerarhiju klastera koja se prikazuje dendrogramom. Postoje dva pristupa: "bottom-up" (aglomerativno) i "top-down" (divizno) grupiranje koji određuju kojim će se redoslijedom spajati, odnosno razdvajati klasteri.

Kod klaster analize, potrebno je odrediti početne centroide te izabrati broj K za broj klastera. Za svaki od ta dva problema postoje razne metode koje nas dovode do optimalnog rješenja.

Nadalje, grupiranja temeljena na gustoći, odnosno mreži podataka koriste informacije o gustoći, obliku te broju zadanih podataka. Uz njihove varijacije, najistaknutiji algoritmi su DBSCAN za gustoću podataka te STING algoritam za mrežu podataka.

Ključne riječi: grupiranje podataka, particijsko grupiranje, hijerarhijsko grupiranje, algoritam K–srednjih vrijednosti, klaster

Data clustering

Summary

The aim of this paper is to get acquainted with the data clustering techniques. clustering is the process of making a group of abstract objects into classes of similar objects.

There are different types of clustering, depending on the type of data. The four main types are: partitional, hierarchical, density-based, and grid-based clustering.

Partitional clustering clusters the dataset into clusters that contain similar data. The most significant algorithm is the K–means algorithm. Additional to this algorithm, other versions appeared, such as the K–medoids algorithm and fuzzy C–means algorithm.

Hierarchical clustering, on the other hand, creates a hierarchy of clusters which is represented with a dendrogram. There are two approaches: the "bottom-up" (agglomerative) and the "top-down" (divisive) clustering which determines the order of joining or splitting the clusters.

In the cluster analysis, we need to determine the initial centers and the number of clusters. For each of those problems, there are different methods that result in an optimal solution.

Moreover, the density-based and the grid-based clustering are using information about the density, the shape, and the number of data. The most significant algorithms are the DBSCAN for the density-based and the STING for the grid-based clustering.

Key words: data clustering, partitional clustering, hierarchical clustering, K–means algorithm, cluster

Životopis

Moje ime je Ana Habijanić, rođena sam 4.10.1996. godine u Virovitici.
Završila sam Osnovnu školu Petra Preradovića u Pitomači i Gimnaziju Petra Preradovića u Virovitici, prirodoslovno-matematički smjer. Upisala sam preddiplomski studij Matematike na Odjelu za matematiku 2015. godine te ga završila 2018. godine s temom završnog rada "Konveksnost u normiranom prostoru" kod mentorice doc. dr. sc. Suzane Miodragović. Iste godine upisujem diplomski studij Matematike i računarstva na Odjelu za matematiku. Na diplomskom studiju odradila sam stručnu praksu u firmi Prototyp.