SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO-MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

AK. GOD. 2019./2020.

SEMINARSKI RAD

ALTERNATIVE ZA K-MEANS

ALGORITAM KOJE PRONALAZE BOLJA GRUPIRANJA

KOLEGIJ: Uvod u složeno pretraživanje podataka

MENTOR: Prof. dr. sc. Zlatko Drmač

AUTORI: Mia Matijašević, Maja Piskač, Mia Tadić

ZAGREB, 2019. god.

SADRŽAJ

SAŽETAK

Proučavamo ponašanje standardnog k-means algoritma i nekoliko njegovih alternativa poput k-harmonic means algoritma, Gaussian expectation-maximization algoritma, fuzzy k-means algoritma te dvije nove varijante k-harmonic means algoritma (Hybrid 1 i Hybrid 2).

Naš je cilj pronaći svojstva ovih algoritama koja pridonose pronalasku što boljih (optimalnijih) grupiranja. Svaki algoritam opisat ćemo u okviru pripadne funkcije članstva i težinske funkcije te potom pokazati kako se gore navedeni algoritmi različito ponašaju na niskodimenzionalnim skupovima podataka (točnije, promatrat ćemo za dimenziju 2).

Vidjet ćemo da različite inicijalizacije utječu na rezultate, ocjene te usporedbe algoritama, te da posjedovanje slabe funkcije članstva (*soft membership*) i nekonstantne težinske funkcije pridonosi boljim rezultatima. Prosječna ponašanja algoritama pokazuju da k-harmonic means ipak ima prednost nad ostalim algoritmima.

UVOD

Data clustering (tj. metoda pronalaženja prirodnih grupiranja podataka) je bitna u strojnom učenju i prepoznavanju uzoraka. Tipično pri grupiranju, ne postoji optimalno rješenje problema. No, algoritmi iz ovog projekta pokazuju da dobro minimiziraju određene matematičke kriterije (koji variraju od algoritma do algoritma). Algoritmi poput k-meansa (KM) pronalaze lokalne, a ne globalne minimume, stoga oni koji bolje minimiziraju matematičke kriterije, smatramo boljima (tj. kvalitetnijim grupiranjima).

Algoritmi poput k-meansa spadaju u skupinu *center-based* algoritama. To su algoritmi koji koriste broj centara za reprezentaciju i/ili particiju ulaznih podataka. Svaki centar definira klaster (grupu) sa središnjom točkom. Takvi algoritmi započinju s pokušajem pogotka rješenja i onda profinjuju pozicije centara (središta) sve dok ne dođu do lokalnog optimuma. Ove metode mogu dovesti i do konvergencije prema lošem lokalnom minimumu čemu je razlog osjetljivost na inicijalizaciju. Zbog toga smo usmjereni na poboljšanje algoritama u svrhu manje osjetljivosti na inicijalizaciju kako bi davali bolje rezultate.

K-harmonic means algoritam (KHM), za razliku od k-means algoritma, proizašao je iz kriterija optimizacije baziranog na harmonijskoj sredini. KHM obećava brzo pronalaženje dobrih rješenja grupiranja te se pokazao boljim u usporedbi s klasičnim k-means algoritmom u mnogim testovima.

Svaki od algoritama ima svoju funkciju cilja koja definira koliko je dobro rješenje, a cilj svakog algoritma je tu funkciju minimizirati. S obzirom na to da te funkcije ne mogu biti minimizirane direktno, koristimo iterativne algoritme koji naposljetku konvergiraju prema lokalnom minimumu.

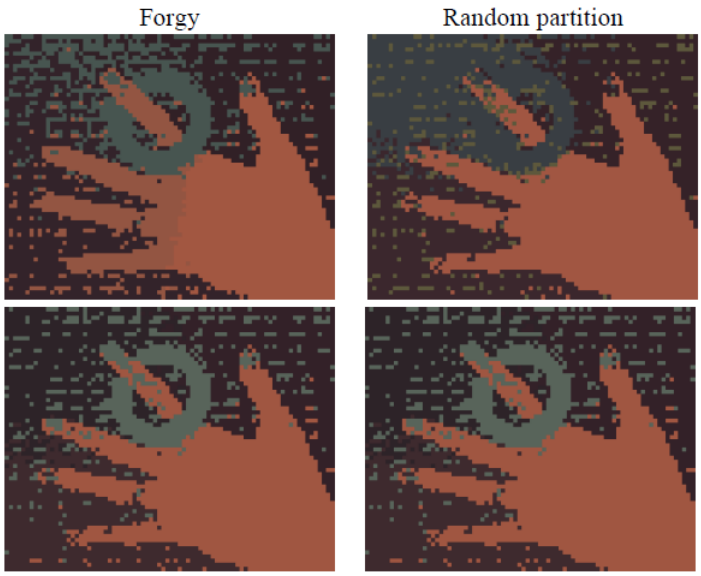
U ovom radu pokazat ćemo ujedinjeni model za promatranje *center-based* algoritama kao što su k-means, k-harmonic means i fuzzy k-means algoritmi. Zatim ćemo izvesti dva nova algoritma temeljena na svojstvima KM i KHM algoritama, s nazivima Hybrid 1 (H1) i Hybrid 2 (H2).

MOTIVACIJA

Motivirani smo ka traženju najkvalitetnijeg algoritma za klasteriranje (grupiranje) primjerom segmentacije slike. Zadatak segmentacije slike je grupiranje piksela slike na temelju boje, teksture i lokacije. Postupak se sastoji od dva dijela: prvi dio odnosi se na definiranje korisnih atributa na pikselima koji se koriste kao ulaz za klasteriranje, dok drugi dio vezan je za prosljeđivanje podataka algoritmu te klasteriranje istih da bismo dobili segmentaciju.

Koristi se slika ruke koja se konvertira u skup podataka od 5 dimenzija. Prve dvije dimenzije su koordinate piksela (x, y) dok su zadnje tri vezane uz boju (l,u,v) iz LUV prostora boja (standard u računalnoj grafici za reprezentiranje boja). Normalizira se svaka dimenzija te se klasteriraju podaci koristeći dva algoritma (KM i KHM) i dvije različite inicijalizacije (Forgy i Random Partition) u 5 klastera. Rezultati su pokazali da je KHM dao bolje rezultate te manje ovisi o tipu inicijalizacije.

Na sljedećim slikama uočavamo da KM algoritam lošije segmentira sliku jer podijeli ruku na dva dijela s Forgy inicijalizacijom, s druge strane KHM oštrije odvaja ruku od pozadine te segmentacija ne ovisi o inicijalizaciji.



OPĆI MODEL ITERATIVNOG ALGORITMA

Definirajmo d-dimenzionalan skup od n podataka $X = \{x_1, \dots, x_n\}$ kao skup za grupiranje (klasteriranje). Definirajmo d-dimenzionalan skup od k centara/središta $C = \{c_1, \dots, c_k\}$ kao rješenje tj. grupiranje koje algoritam treba poboljšati. Funkcija članstva $m(c_j|x_i)$ definira omjer/udio svake točke x_i koja pripada centru c_j sa svojstvima $m(c_j|x_i) \geq 0$ i $\sum_{j=1}^{k} m(c_j|x_i) = 1$. Neki algoritmi koriste jaku funkciju članstva(*hard membership function*), tj. $m(c_j|x_i) \in \{0,1\}$, dok neki koriste slabu (*soft membership function*) tj. $0 \leq m(c_j|x_i) \leq 1$. Težinska funkcija $w(x_i)$ sa svojstvom $w(x_i) \geq 0$ definira koliko utjecaja podatak x_i ima u ponovnom određivanju parametara centara u sljedećoj iteraciji.

Koraci općenitog modela iterativnog algoritma:

1. Inicijalizirati algoritam sa „pogođenim“ centrima C.
2. Za svaki podatak x_i, izračunati „članstvo“ $m(c_j|x_i)$ u svakom centru c_j i svakoj težini $w(x_i)$.
3. Za svaki centar c_j, ponovno mu odrediti lokaciju iz svih podataka x_i s obzirom na njihove funkcije članstva i težine:

$$c_j = \frac{\sum_{i=1}^{n} m(c_j|x_i) w(x_i) x_i}{\sum_{i=1}^{n} m(c_j|x_i) w(x_i)}$$

4. Ponavljati korake 2. i 3. dok se ne dođe do konvergencije.

Sada možemo uspoređivati algoritme prema njihovima funkcijama članstva i težinskim funkcijama.

K-MEANS

K-means algoritam particionira podatke u k skupova. Rješenje je skup od k centara/središta, od kojih je svaki smješten u tzv. *centroid* grupe (središte svih onih podataka kojima je taj centar najbliži). Funkcija cilja koju KM algoritam optimizira je:

$$\mathrm{KM}(\mathrm{X},\mathrm{C}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \min_{\mathrm{j}\in\{1,\ldots,\mathrm{k}\}} \left\|\mathrm{x_i} - \mathrm{c_j}\right\|^2$$

Ova funkcija cilja daje algoritam koji minimizira varijancu (udaljenost) unutar pojedinog klastera, tj. grupe (kvadrirana udaljenost između centara i njemu dodijeljenih točaka). Funkcija članstva i težinska funkcija dane su sa:

$$\mathrm{m_{KM}}(\mathrm{c_l}|\mathrm{x_i}) = \begin{cases} 1, & \text{za } \mathrm{l} = \arg\min_{\mathrm{j}} \left\|\mathrm{x_i} - \mathrm{c_j}\right\|^2 \\ 0, & \text{inače} \end{cases}$$

$$\mathrm{w_{KM}}(\mathrm{x_i}) = 1$$

KM ima jaku funkciju članstva i konstantnu težinsku funkciju koja svim točkama daje jednaku važnost.

GAUSSIAN EXPECTATION-MAXIMIZATION

Gaussian expectation-maximization (GEM) algoritam za klasteriranje koristi linearnu kombinaciju d-dimenzionalnih Gaussovih distribucija kao centara. On minimizira sljedeću funkciju cilja

$$GEM(X,C) = -\sum_{i=1}^{n} log\left(\sum_{j=1}^{k} p(x_i|c_j)p(c_j)\right)$$

gdje je $p(x_i|c_j)$ vjerojatnost od x_i generiranog Gaussovom distribucijom s centrom c_j, a $p(c_j)$ je prethodna vjerojatnost centra c_j. Funkcija članstva i težinska funkcija dane su sa:

$$m_{GEM}(\mathrm{c_j}|\mathrm{x_i}) = \frac{p(x_i|c_j)p(c_j)}{p(x_i)}$$

$$w_{GEM}(x_i) = 1$$

Primijetimo da se koristi Bayesovo pravilo za izračunavanje mekog članstva te je m_{GEM} također vjerojatnost. Kao i kod KM algoritma, konstantna težinska funkcija daje svim točkama jednaku važnost.

FUZZY K-MEANS

Fuzzy k-means algoritam (FKM) je prilagodba KM algoritma koja koristi slabu funkciju članstva. Za razliku od KM, FKM dopušta da podatak djelomično pripada svim centrima.

$$FKM(X,C) = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{r}\left\|x_i - c_j\right\|^2$$

Parametar u_{ij} predstavlja udio podatka x_i koji je dodijeljen centru c_j sa svojstvom $\sum_{j=1}^{k} u_{ij} = 1$ za svaki i i $u_{ij} \geq 0$, dok za r vrijedi $r \geq 1$.

Funkcija članstva i težinska funkcija su:

$$m_{FKM}(c_j|x_i) = \frac{\|x_i - c_j\|^{-2/(r-1)}}{\sum_{j=1}^{k}\|x_i - c_j\|^{-2/(r-1)}}$$

$$w_{FKM}(x_i) = 1$$

FKM ima slabu funkciju članstva i konstantnu težinsku funkciju. Kako r teži u 1 odozgo, algoritam se sve više ponaša kao standardni KM te centri dijele manje podataka.

K-HARMONIC MEANS

Funkcija cilja K-harmonic means algoritma koristi harmonijsku sredinu udaljenosti pojedinog podatka od svih centara.

$$KHM(X,C) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\left\|x_i - c_j\right\|^p}}$$

Input parametar p ima svojstvo $p \geq 2$.

Funkcija članstva i težinska funkcija dane su sa:

$$m_{KHM}(c_j|x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}}$$

$$w_{KHM}(x_i) = \frac{\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}}{(\sum_{j=1}^{k}\|x_i - c_j\|^{-p})^2}$$

KHM koristi slabu funkciju članstva, ali i varirajuću težinsku funkciju. Ova težinska funkcija daje veću težinu točkama čije su udaljenosti od centara velike, što pomaže centrima u širenju kako bi pokrili što veći broj podataka. Dakle, harmonijski red daje bolju (manju) ocjenu onim točkama koje su blizu barem jednom centru. To je inače svojstvo harmonijskog reda - sličan je funkciji cilja KM algoritma, ali je ovo glatka diferencijabilna funkcija.

Implementacija KHM algoritma dodatno zahtijeva rješavanje slučaja $x_i = c_j$. U tom slučaju koristimo Zhangovo rješenje $\max(\|x_i - c_j\|, \varepsilon)$, gdje je ε mala pozitivna vrijednost.

NOVI ALGORITMI GRUPIRANJA

Kao što smo već rekli, KHM ima slabu funkciju članstva i varirajuću težinsku funkciju te se pokazao najmanje osjetljiv na inicijalizaciju, stoga ćemo analizirati njegova svojstva te definirati dva nova algoritma nazvana Hybrid 1 i Hybrid 2. Samo ime nastalo je iz činjenice da su oni doista hibridni algoritmi koji kombiniraju svojstva algoritama KM i KHM. Svrha nastanka tih dvaju algoritama je da se otkriju koja svojstva i učinke imaju funkcija članstva i težinska funkcija algoritma KHM.

HYBRID 1

Hybrid 1 (H1) koristi jaku funkciju članstva iz KM algoritma. Svaka točka pripada točno jednom i to najbližem centru. H1 koristi težinsku funkciju iz KHM algoritma koja daje veću težinu onim točkama čije su udaljenosti od centara velike. Očekujemo da će ovaj algoritam konvergirati brže od algoritma KM zbog težinske funkcije, no i dalje će imati problema zbog jake funkcije članstva. Kao što smo naveli, imamo sljedeću funkciju članstva i težinsku funkciju:

$$m_{H1}(c_l|x_i) = \begin{cases} 1, & \text{za } l = \arg\min_j \|x_i - c_j\|^2 \\ 0, & \text{inače} \end{cases}$$

$$w_{H1}(x_i) = \frac{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{k} \|x_i - c_j\|^{-p}\right)^2}$$

HYBRID 2

Hybrid 2 (H2) koristi slabu funkciju članstva iz KHM algoritma te konstantnu težinsku funkciju KM algoritma. Funkcije su sljedeće:

$$m_{H2}(c_j|x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}$$

$$w_{H2}(x_i) = 1$$

Primijetimo da H2 nalikuje na FKM, tj. za određene vrijednosti r i p oni su matematički ekvivalentni. No svejedno ćemo ih testirati odvojeno iz razloga da otkrijemo kako funkcija članstva i težinska funkcija iz KHM utječu na grupiranje.

TESTIRANJE

Želimo odgovoriti na nekoliko pitanja: kako različite inicijalizacije utječu na pojedini algoritam, kakav je utjecaj slabog, odnosno jakog članstva i koje su koristi od varirajućih, odnosno konstantnih težina.

Iako svaki algoritam minimizira drugačiju funkciju cilja, mi ćemo za svaki algoritam računati:

$$\sqrt{\mathrm{KM(X, C)}}$$

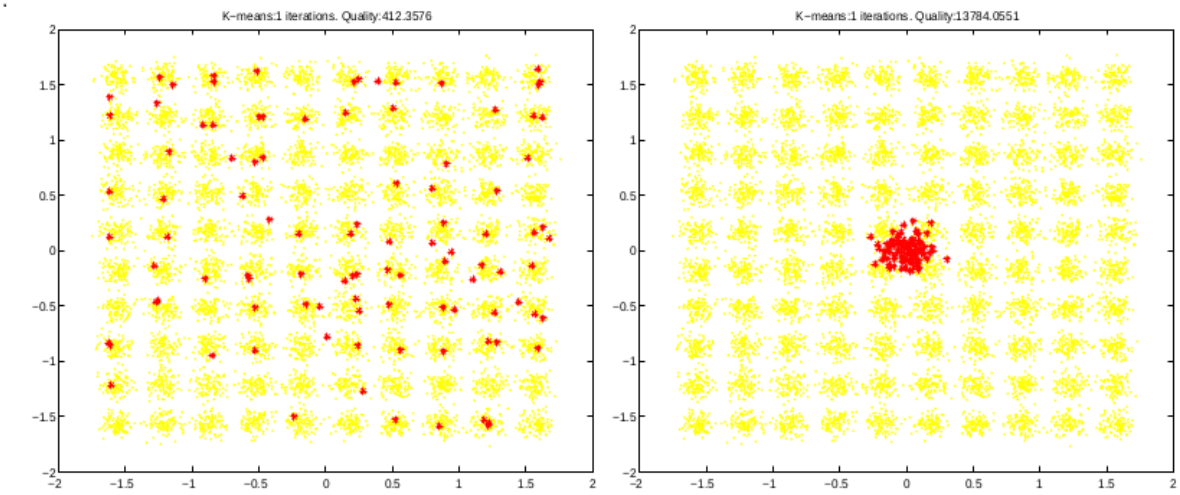
To je razumna metrika za prosuditi kvalitetu klastera, a uz to, koristeći istu metriku možemo usporediti različite algoritme. Koristimo korijen jer kvadratna funkcija može „preuveličati" ozbiljnost loših problema. Također dodatno koristimo ocjenu klasteriranja

$$R_{d,i} = \sqrt{\frac{KM(X_{d,i}, C_{d,i})}{KM(X_{d,i}, O_{d,i})}}$$

gdje je d u našem slučaju jednak 2 (dimenzija), i je broj podataka, a $O_{d,i}$ nam predstavlja optimalno klasteriranje.

Napravili smo testove nad dvodimenzionalnim skupom podataka, zbog jasnijeg prikaza na grafovima te time i lakšeg uočavanja razlika među algoritmima. Kod je pisan u Python-u. Skup podataka kojeg smo koristili u testovima generiran je pomoću funkcije *make_blobs* sa standardnom devijacijom od 0.8 (funkcija se nalazi u biblioteci *sklearn*). Ta funkcija generira n nasumičnih točaka (gdje je n argument koji se prosljeđuje funkciji) koje su već raspoređene u k prirodnih grupa/klastera (k je također proslijeđen funkciji). Važno je napomenuti da su dobiveni podaci normalizirani Gaussovom, tj. normalnom, razdiobom.

Svaki algoritam pokreće se dvaput – jednom sa Forgy inicijalizacijom centara, drugi put sa Random partition inicijalizacijom centara. Forgy metoda nasumično odabire k točaka/podataka iz skupa od n zadanih točaka i koristi ih kao početne centre. Random partition metoda dodjeljuje svaku točku nasumično odabranoj grupi i potom kao inicijalne centre postavi središta točaka dodijeljenih određenom centru. Razlika između ove dvije inicijalizacije je što Forgy raširi sve centre po skupu podataka, dok Random Partition sve centre smjesti u malo područje oko središta svih točaka/podataka.

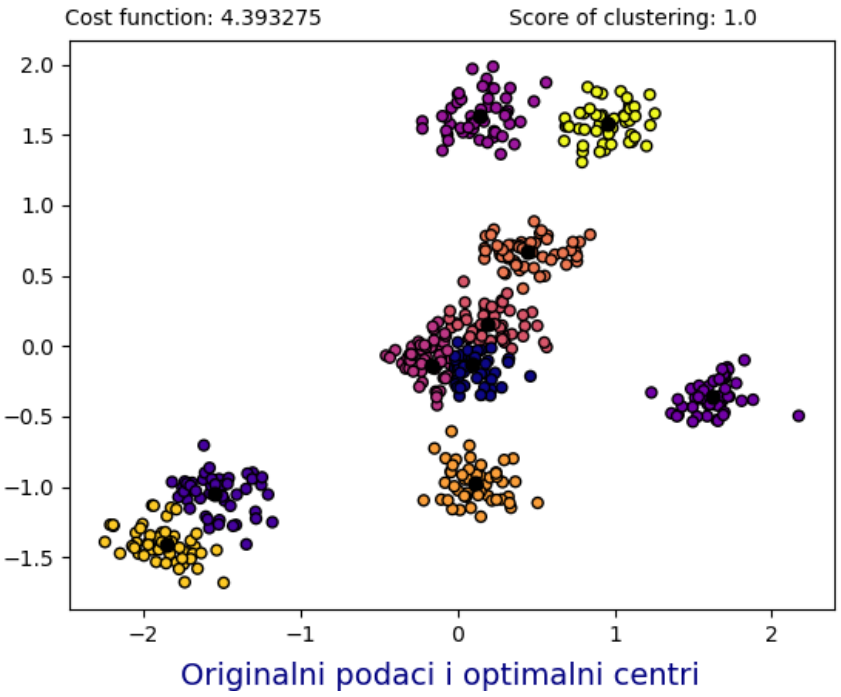


Slika 1. **Usporedba Forgy i Random partition inicijalizacije**

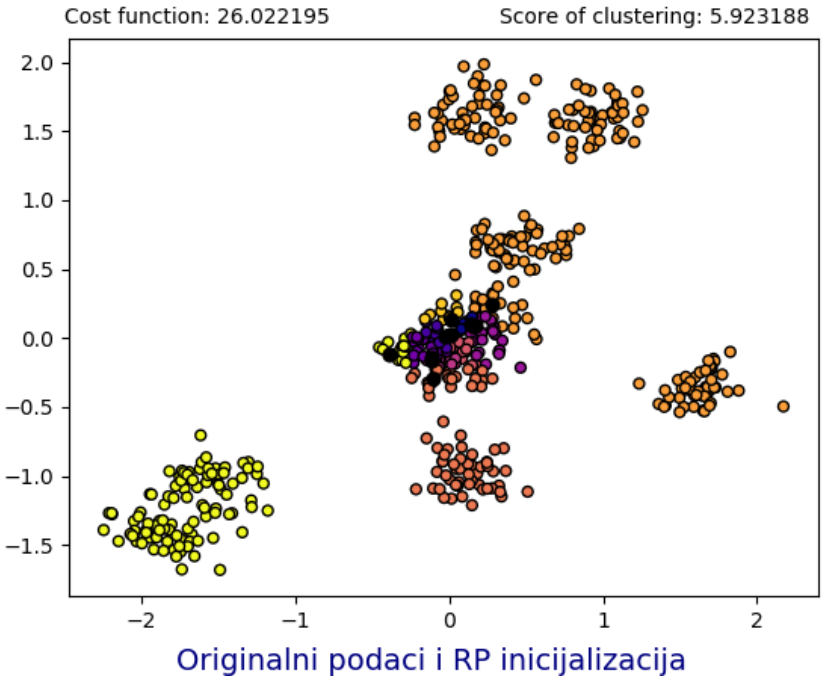
Vrijednost varijable p smo postavili na 3.5 jer je to najbolja vrijednost parametara po Zhangu, a r smo postavili na 1.3 što se pokazalo najboljim odabirom prilikom početnog testiranja te prema članku.

Testove smo proveli na 200 podataka i 8 klastera.

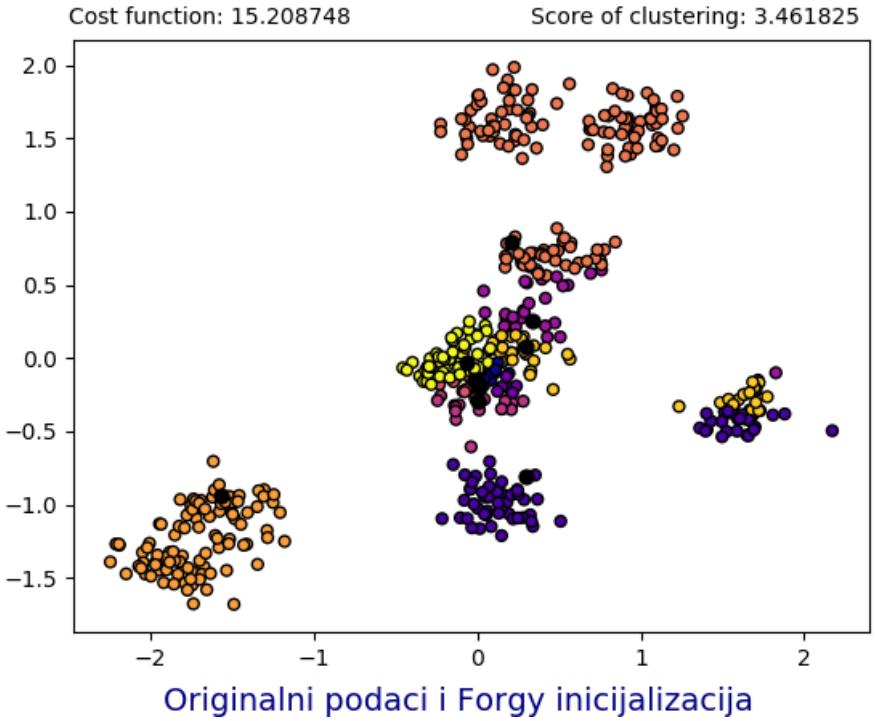
Na sljedećim slikama nalazi se prikaz rezultata jednog provedenog testa. Na svakoj slici je naveden iznos funkcije cilja po kojoj ćemo uspoređivati efikasnosti algoritama.



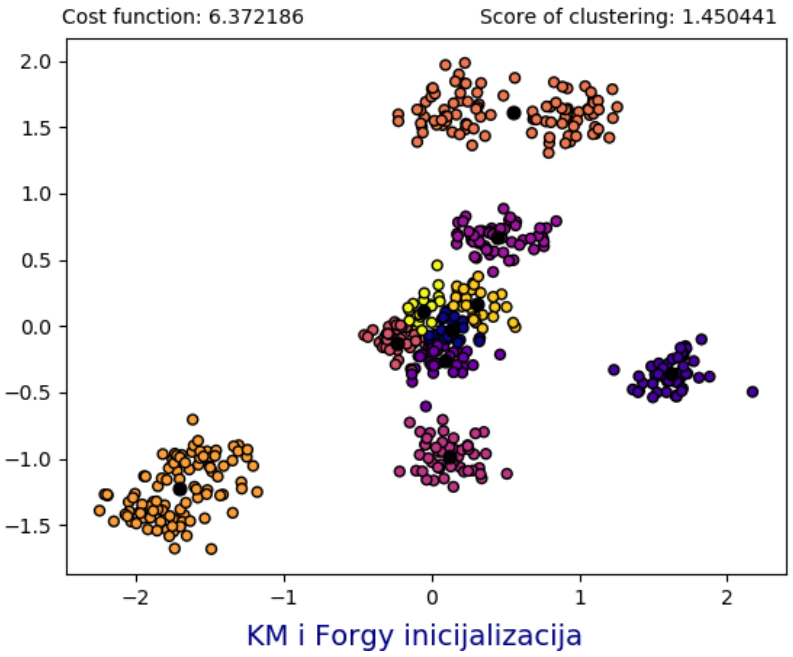
Slika 2. **Generirani podaci grupirani u klastere funkcijom *make_blobs* i pripadni optimalni centri**



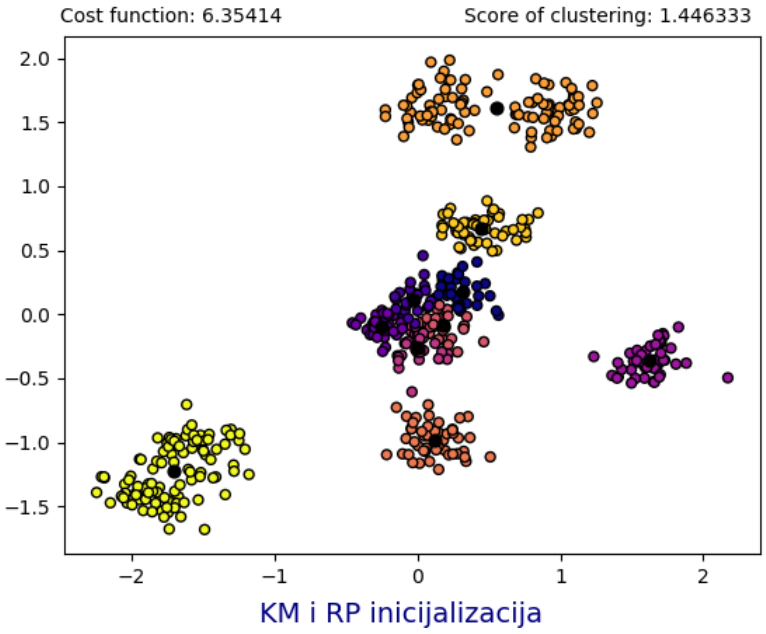
Slika 3. **Originalni podaci podijeljeni u klastere nastale Random partition inicijalizacijom**



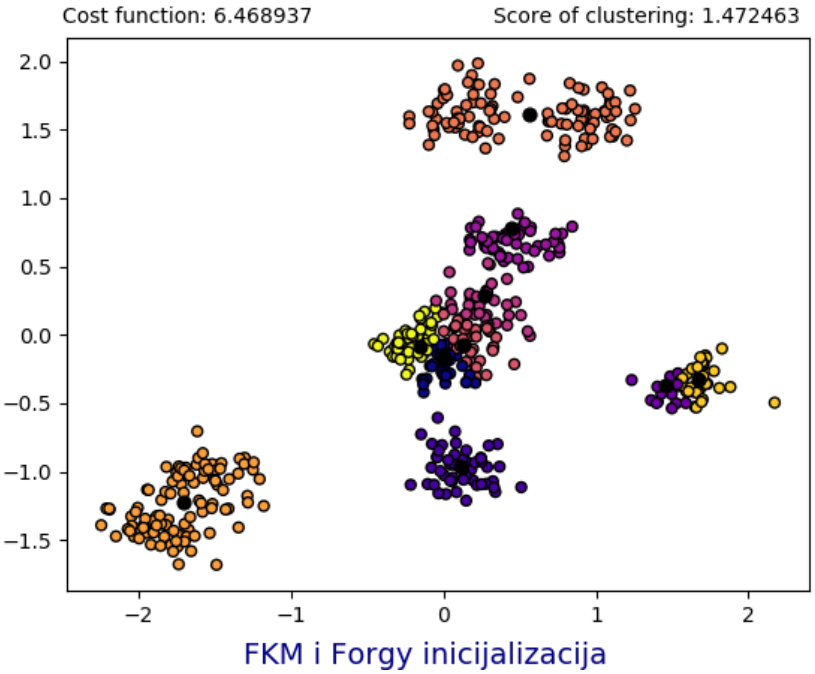
Slika 4. **Originalni podaci podijeljeni u klastere nastale Forgy inicijalizacijom**



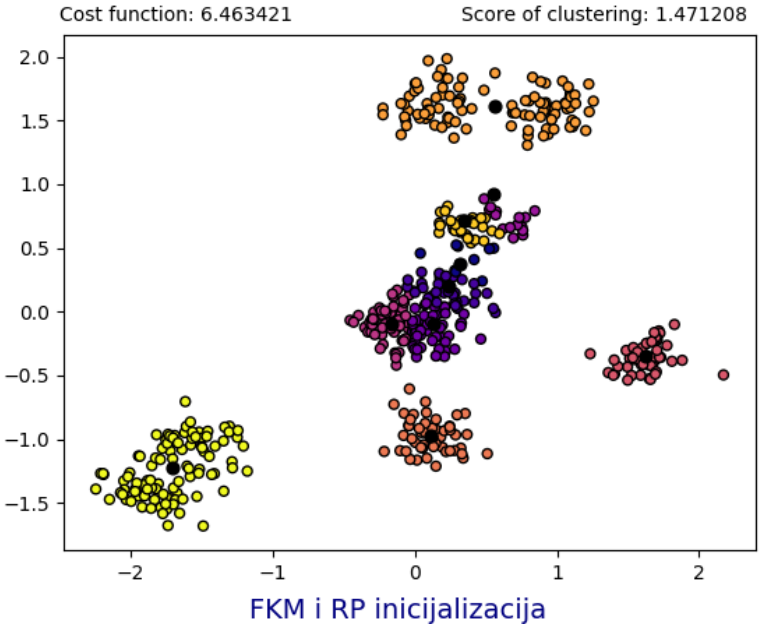
Slika 5. **Rezultati k-means algoritma sa Forgy inicijalizacijom**



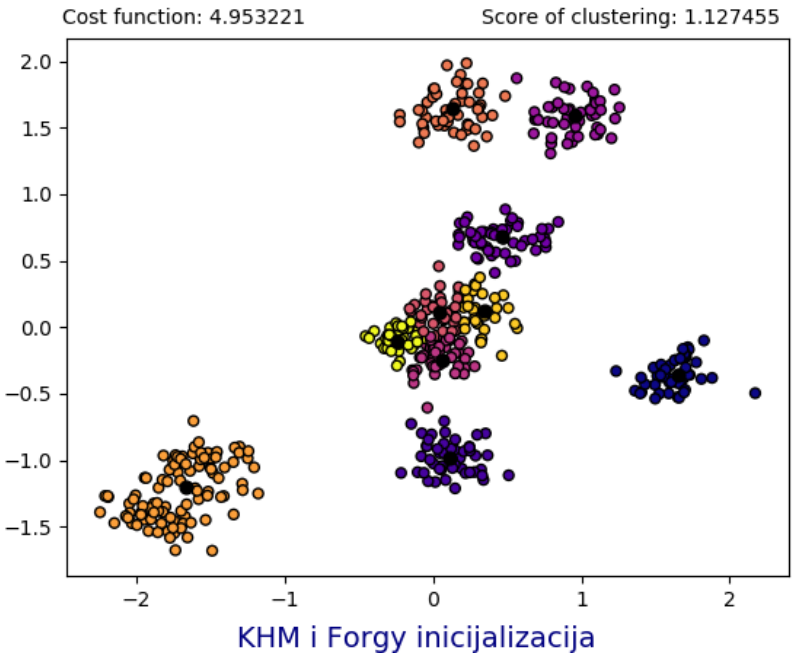
Slika 6. **Rezultati k-means algoritma sa Random partition inicijalizacijom**



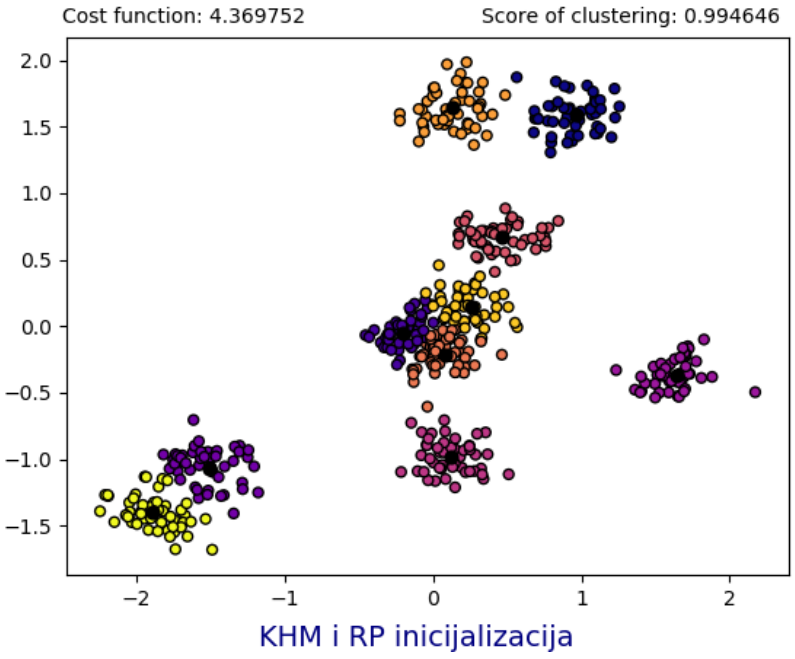
Slika 7. Rezultati Fuzzy k-means algoritma sa Forgy inicijalizacijom



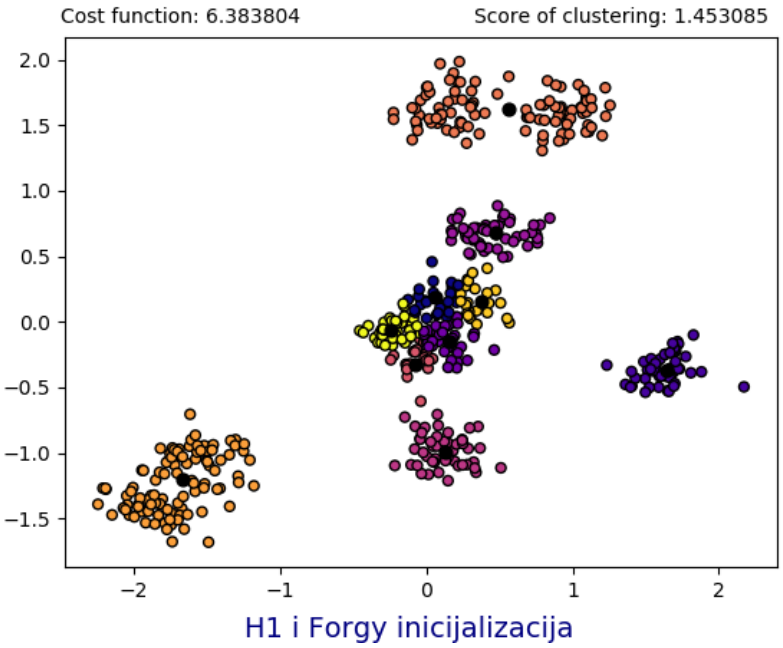
Slika 8. Rezultati Fuzzy k-means algoritma sa Random partition inicijalizacijom



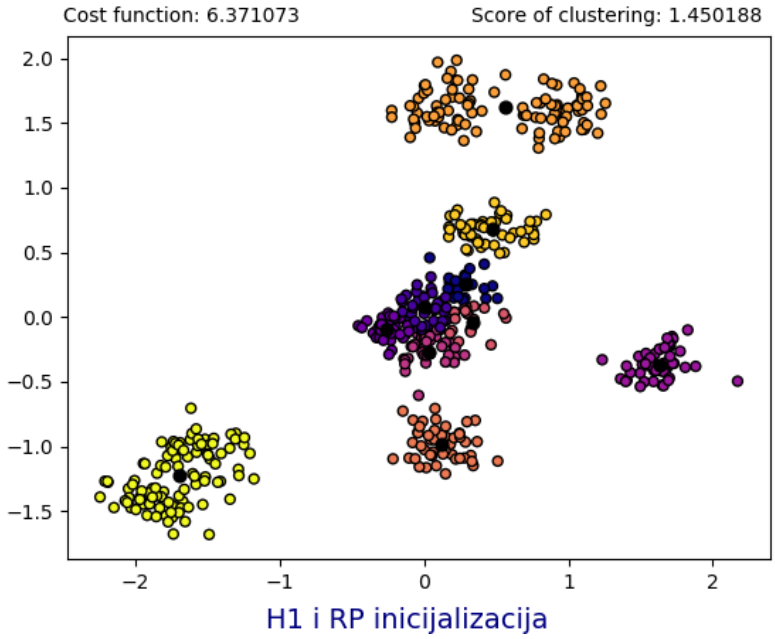
Slika 9. Rezultati k-harmonic means algoritma sa Forgy inicijalizacijom



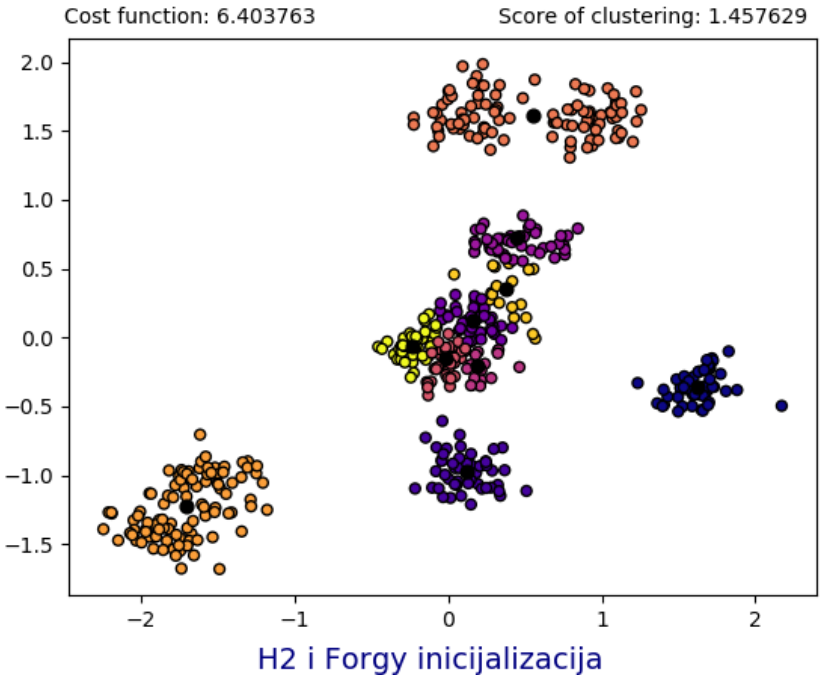
Slika 10. Rezultati k-harmonic means algoritma sa Random partition inicijalizacijom



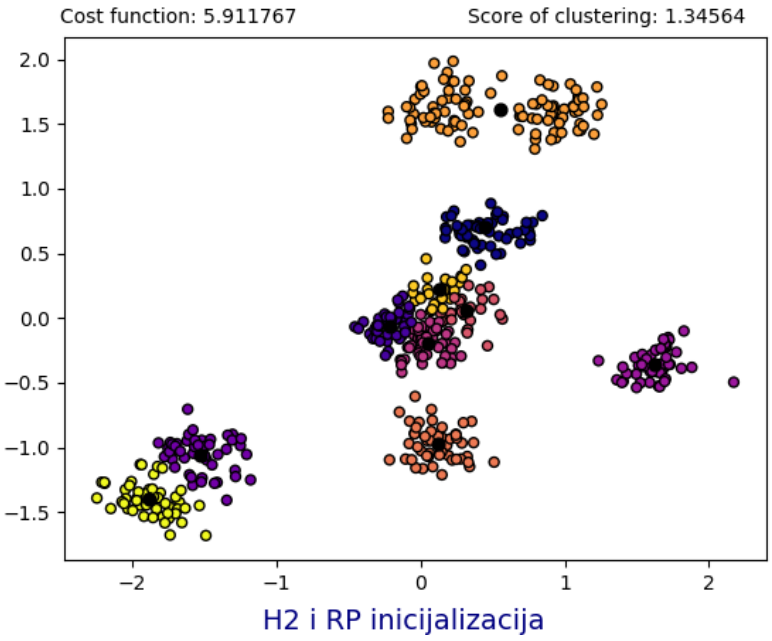
Slika 11. Rezultati Hybrid 1 algoritma sa Forgy inicijalizacijom



Slika 12. Rezultati Hybrid 1 algoritma sa Random partition inicijalizacijom



Slika 13. **Rezultati Hybrid 2 algoritma sa Forgy inicijalizacijom**



Slika 14. **Rezultati Hybrid 2 algoritma sa Random partition inicijalizacijom**

ZAKLJUČAK

Navedene slike prikazuju prosječne rezultate naših provedenih testova. Na temelju provedenih eksperimenata došli smo do zaključka da k-harmonic means algoritam nalazi optimalnije rješenje, tj. bolje minimizira funkciju cilja, nego ostali algoritmi.

K-harmonic means algoritam ne ovisi uvelike o inicijalizaciji, no kada dođe do značajnije razlike, onda su rezultati bolji s Random partition inicijalizacijom. Čak i u tom slučaju kad su različiti, svejedno oba daju najbolje rezultate u odnosu na ostale algoritme i inicijalizacije.

K-means algoritam daje varirajuće rezultate ovisno o ulaznim podacima. Također ovisi o inicijalizaciji - s Random partition inicijalizacijom u prosjeku daje bolje rezultate.

Hybrid 1 i hybrid 2 algoritmi se ne ističu ni kao najgori, a ni kao najbolji, ali ako je jedna inicijalizacija pogodna za jedan od njih, onda je u našem testiranju pogodna i za drugi.

Fuzzy k-means algoritam uglavnom daje lošije rezultate u odnosu na druge algoritme. Od svih kombinacija algoritama i inicijalizacija u prosjeku je najgori Fuzzy k-means u kombinaciji s Forgy inicijalizacijom.

LITERATURA

[1] Alternatives to the K-means Algorithm that Find Better Clusterings: Greg Hamerly, Charles Elkan (Department of Computer Science and Engineering, University of California)

[2] www.learnpython.org

[3] www.stackoverflow.com