Fuzzy C-means klasterovanje članaka uz optimizaciju rojem čestica

Seminarski rad u okviru kursa
Računarska inteligencija
Matematički fakultet

Katarina Nikolić
katarina.b.nikolic@gmail.com

Septembar 2022

Sažetak

Fazi klasterovanje odnosi se na klasterovanje zasnovano na teoriji fazi skupova gde se jedna tačka prostora koji se klasteruje može pripadati različitim klasterima istovremeno, pri čemu se ova pripadnost izražava u stepenima između 0 i 1. **Fuzzy c-means** je jedan od najkorišćenijih i najjednostavnijih algoritama (source) fazi klasterovanja. U ovom radu korišćena je tehnika *optimizacije rojem čestica (particle swarm optimization)* u čistom obliku, kao i hibridna verzija gde se inicijalna populacija bira genetskim algoritmom, da bi se ispitala njihova primenljivost na ovaj problem. Klasterovanje je vršeno na skupu od 2000 prečišćenih članaka popularne medicine, transformisanih u tf-idf matricu čije su dimenzije smanjene tehnikom *single value decompositon*.

Sadržaj

1 Uvod

1.1 Motivacija

Uopšteno govoreći, cilj klasterovanja tekstualnih podataka je da se u velikoj količini dokumenata pronađu korisni obrasci (radi filtriranja lažnih vesti, analize sentimenata, otkrivanja tema) [1] U slučaju konkretnog skupa članaka koji je obrađen u ovom radu, može se izvući podatak o tome koje su (bile) neke od najzastupljenijih tema u medicinskom novinarstvu u trenutku objavljivanja, a stepeni pripadnosti mogu nam pružiti i informaciju o tome koje teme se često javljaju zajedno. Uz to, fazi klasterovanje (konkretno fuzzy c-means) se koristi i u praksi za obradu medicinskih tekstualnih podataka [4] [5].

1.2 Algoritam FCM

Klasterovanje se može posmatrati kao optimizacioni problem, gde je cilj tačke prostora particionisati tako da je vrednost korisnički definisane funkcije koja opisuje kvalitet klasterovanja što veća. *Meko* klasterovanje je dobar izbor u slučajevima gde ne postoji jasna, čista podela između klastera, i gde nam je potrebna nekakva težina w_ij da izrazi pripadnost n-dimenzione tačke x_i klasteru C_j. Fazi klasterovanje je tip mekog klasterovanja zasnovan na teoriji fazi skupova Lofija Zadeha, koje nudi prirodan i ekspresivan (ali ne probabilistički) model za meko klasterovanje. Fuzzy c-means pripada grupi algoritama za zasnovanih na prototipovima, što znači da je pripadnost tačke klasteru određena njenom blizinom "prototipu"(nekako odabranom predstavniku) tog klastera. Štaviše, može se smatrati "fazifikovanom"varijantom popularnijeg algoritma K-Means (K sredina), gde, umesto da se tačka dodeli klasteru čiji joj je centroid najbliži, tački dodeli težina pripadnosti svakome od klastera, proporcionalno udaljenosti od njegovog centroida. Da bi skup od k centroida$C_1, C_2, ...C_k$ formirao tzv. **fazi pseudo-particionisanje** (tj. validno fazi klasterovanje), potrebno je da važe sledeća pravila[6]:

1. Suma težina pridruženih svakoj čestici je u zbiru jednaka 1

$$\sum_{j=0}^{k} w_{ij} = 1$$

2. Svaki klaster sadrži barem jednu tačku sa barem nekom nenula težinom, ali nijedan ne sadrži svaku tačku sa težinom 1

$$0 < \sum_{i=0}^{m} w_{ij} < m$$

1.2.1 Dodela težina i funkcija sličnosti

Težina koja predstavlja stepene pripadnosti elementa skupa svakom klasteru (gde je klaster j definisan preko svog centroida Cj) definisana je kao[6]:

$$w_{ij} = (\frac{1}{d(\mathbf{x}_i, C_j)})^{\frac{1}{p-1}} / \sum_{q=1}^{k} \frac{1}{d(\mathbf{x}_i, C_q)}^{\frac{1}{p-1}}$$

Kao funkcije sličnosti razmatrane su funkcije koje se u literaturi spominju kao pogodne za rad sa tekstom - kosinusno rastojanje, euklidsko rastojanje, i rastojanje Čebiševa. Pri testiranju se kosinusno rastojanje pokazalo kao najefikasnije, pa je se u daljem tekstu ono podrazumeva kao funkcija sličnosti, sem ako nije drugačije naznačeno.

2 Priprema i odabir parametara

2.1 Priprema podataka

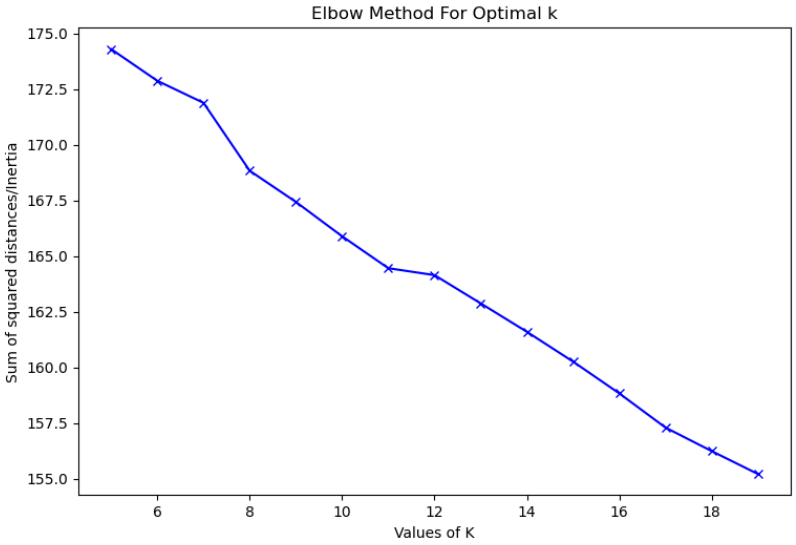
Skup podataka korišćen u projektu dostupan je besplatno na web stranici Kaggle i sastoji se od teksta nasumičnih 2000 članaka sa portala Medical News Today. Za pretprocesiranje članaka korišćena je biblioteka za python NLTK, koja nudi funkcionalnosti brisanja stop reči, svođenja reči na njihov koren i sl. U ovom radu korišćena je generička lista stop reči za engleski jezik koje biblioteka nudi. U nekom od budućih radova ostaje da se ispitaju performanse ako bi se koristio spisak stop reči prilagođen konkretnom domenu.
Tako transformisanu listu reči potom formira se tf-idf matrica, a zatim se na nju - zbog ogromnog broja atributa (reči) - primenjuje tehnika za smanjivanje dimenzionalnosti single value decomposition - SVD. Ovo je čest korak u obradi prirodnog jezika i pokazano je da dovodi do znatnog poboljšanja performansi algoritama koji rade nad tekstom. (izvor) SVD zahteva da broj atributa tabele ne prelazi broj instanci, pa sam se opredelila za to da broj atributa bude približno 90% broja instanci, za šta se eksperimentalno pokazalo da predstavlja neki balans između očuvanja varijanse skupa (obično oko 91%) i efikasnosti obrade.

2.2 Odabir parametara

2.2.1 Broj klastera

Ne postoji savršena metoda za odabir broja klastera, pogotovo kada vizuelizacija podataka nije moguća usled velikog broja dimenzija. Ipak, određene metode mogu dati približnu ideju. Metode korišćene u ovom radu su metod "lakta", koeficijent siluete i gap statistika.
Metoda lakta podrazumeva da se klasterovanje izvrši sa nekoliko različitih parametara k, i da se svako od tih klasterovanja evaluira nekom funkcijom kvaliteta klasterovanja (u ovom slučaju korišćena je SSE, odnosno suma kvadratnih grešaka), i ovi rezultati se grafički prikažu. Ideja metode da će se na grafiku u funkciji koja predstavlja meru kvaliteta kroz promenu parametra k videti prevoj ("lakat") za vrednost k koja predstavlja "tačan"ili lokalno optimalan broj klastera. U ovom radu korišćen je bibliotečki algortam `KMeans` iz biblioteke `scikit-learn`, kao gruba aproksimacija fazi klasterovanja. Klasterovanje je vršeno na podskupu od 200 nasumično odabranih članaka iz skupa, a za moguć broj klastera odabran je raspon od 5 do 20. Ovaj metod nije dao najjasnije rezultate, s obzirom na to da se kao mera kvaliteta korišćena SSE, koja svakako opada sa brojem klastera, a prevoji u funkciji nisu dovoljno upadljivi da bi se izvuklo jednoznačno rešenje.
Koeficijent siluete je mera koja određuje koliko su članovi jednog klastera bliski članovima drugih klastera. Moguće vrednosti su joj u rasponu $[-1, 1]$. Vrednosti koeficijenta siluete blizu 1 ukazuju na veliku udaljenost članova klastera od članova ostalih klastera, vrednosti oko 0 ukazuju na

Slika 1: Grafik za metodu lakta



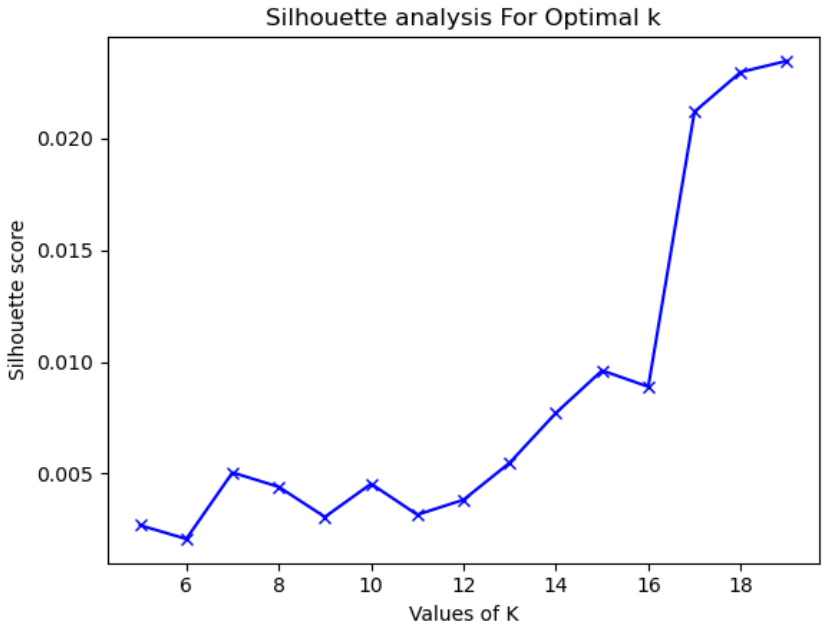
bliskost članova različitih klastera, a negativne vrednosti ukazuju da je član nekog klastera možda dodeljen pogrešnom klasteru. Merene su prosečne vrednosti koeficijenta siluete za svaki od predloženih brojeva klastera i rezultati su prikazani grafički. Ideja je odabrati broj klastera za koji je prosek najveći.
Gap statistika predstavljena je 2001. kao metod za određivanje optimalnog broja klastera, primenljiv na bilo koji algoritam klasterovanja [3]. Tačan opis metode može se naći u navedenom radu, a za potrebe ovog rada korišćena je gotova biblioteka za python `gap_statistic` (GitHub repozitorijum biblioteke).
Višestrukim primenjivanjem svih ovih metoda, zaključak je bio da optimalan broj klastera leži između 14 i 17. U ostatku rada podrazumeva se da je odabrano 15 klastera.

2.2.2 Parametri za PSO

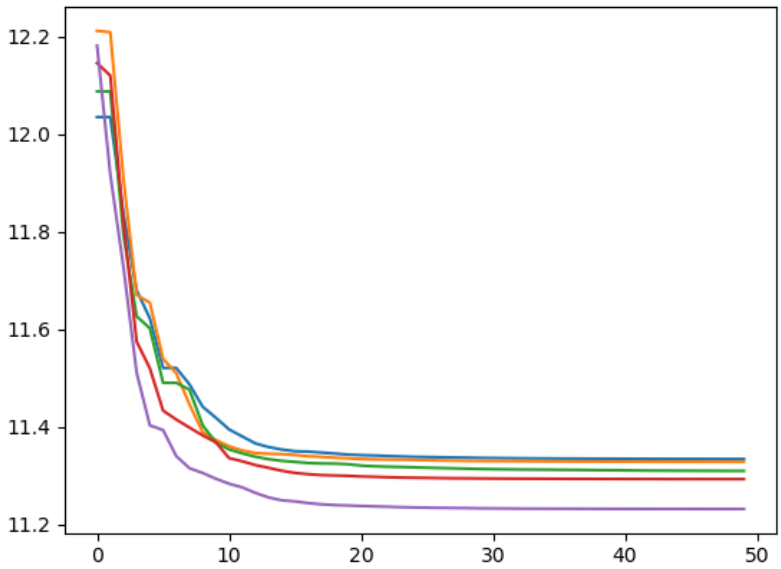
Isprva u testiranju su vrednosti konstanti inercije, $c1$ (konstanta uz kognitivnu komponentu) i $c2$ (konstanta uz socijalnu komponentu) postavljene na 0.5. Algoritam je pokretan 5 puta sa ovim vrednostima konstanti, i pritom je postavljeno da se traži 15 klastera uz pomoć 30 čestica u roju, sa 50 iteracija, na podskupu od 200 nasumično odabranih članaka.
Za manji skup se i ove vrednosti dobro pokazuju - algoritam se na oko 20 iteracija stabilizuje oko lokalnog ekstremuma (Slika 3). Prosečno vreme izvršavanja algoritma u ovom slučaju bilo je 21.4 sekunde.
U nekim radovima [2] predlaže se progresivno prilagođavanje inercije, tako da se sa kasnijim iteracijama smanjuje (kako se rešenje približava nekom lokalnom optimumu) (izvor). U sledećem pokretanju vrednost inercije se progresivno smanjuje od 0.9 do 0.4, formulom

$$w(t) = (w(0) - w(n_t)) * \frac{n_t - t}{n_t} + w(n_t)$$

Slika 2: Prosečne vrednosti koeficijenta siluete

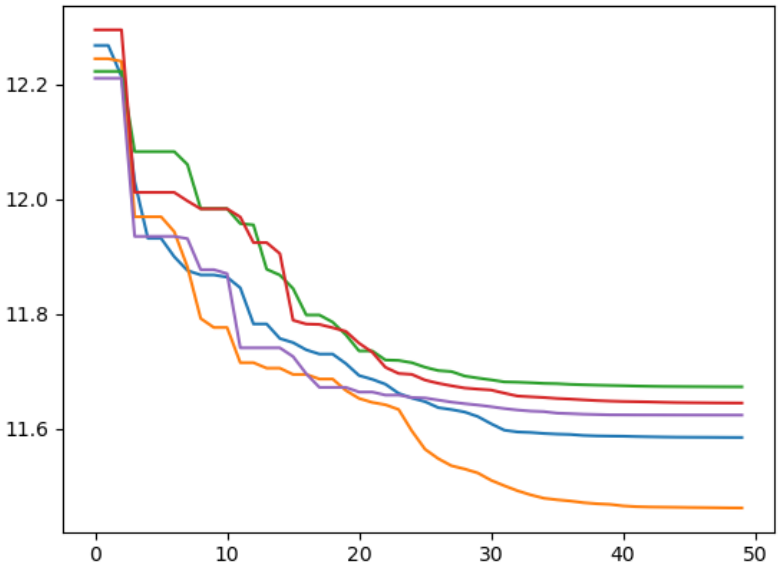


Slika 3: Vrednosti SSE po broju iteracija



gde je n_t broj iteracija, t trenutna iteracija, $w(0)$ početna inercija a $w(n_t)$ krajnja inercija. Ovaj pristup nije se pokazao dobrim u slučaju manjeg skupa, ali ima smisla isprobati ga na celom skupu i sa većim brojem iteracija (Slika 4). Zbog dodatnih operacija pri podešavanju inercije, izvršavanje je bilo malo sporije - u proseku 22.2 sekunde. Kognitivna komponenta

Slika 4: Vrednosti SSE po broju iteracija sa prilagođavanjem inercije - pri većoj inerciji vidi se sporija konvergencija



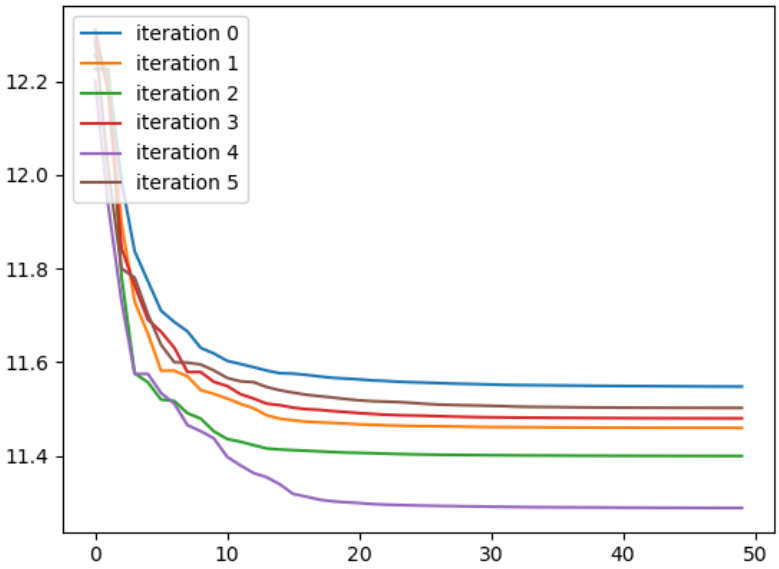
brzine čestice definiše koliko se ona oslanja na šopstvena saznanja", odnosno, za veću kognitivnu komponentu svaka čestica više naginje svom lokalnom optimumu *lbest*. Algoritam je testiran za vrednosti inercije i socijalne komponente 0.5, a kognitivna komponenta se kreće od 0.9 do 0.4.

Najbolje performanse postignute su kada je kognitivna komponenta imala vrednost 0.8. Prosečno vreme izvršavanja je kao i u početnom testu - 21.4 sekunde (Slika 5)

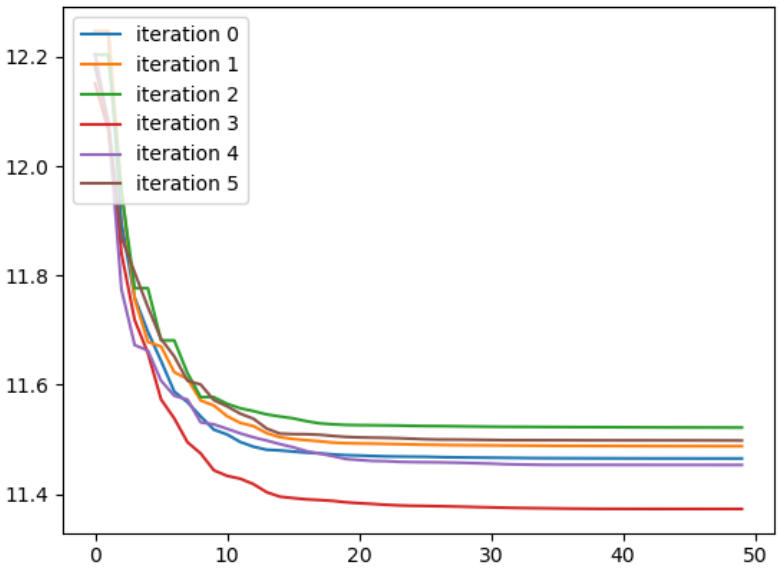
Isti test isproban je sa socijalnom komponentom. Što je socijalna komponenta veća, svaka čestica će više težiti rešenju *gbest*. Sa fiksiranim vrednostima inercije i kognitivne komponente na 0.5, najbolje ponašanje dobija se za vrednost socijalne komponente 0.6. (Slika 6) Prosečno vreme izvršavanja je malo više - 21.8 sekundi.

Iz ovoga se može zaključiti da je za ovaj skup bolje dati blagu prednost kognitivnoj komponenti. Međutim, dobre performanse se dobijaju i u slučaju kada su one izbalansirane. Ubuduće će se koristiti vrednost 0.7 za kognitivnu, a 0.6 za socijalnu komponentu. Takođe, bilo bi možda od interesa ispitati da li se pri većem skupu i broju iteracija vidi korist od inercije koja progresivno opada. Mana ovakvog testiranja je što nisu testirani ovi faktori u kombinaciji jedni s drugima (npr. visoka socijalna komponenta uz nisku kognitivnu). Ovo ostaje da se istraži u budućim radovima.

Slika 5: Vrednosti SSE sa različitim vrednostima kongitivne komponente



Slika 6: Vrednosti SSE sa različitim vrednostima socijalne komponente



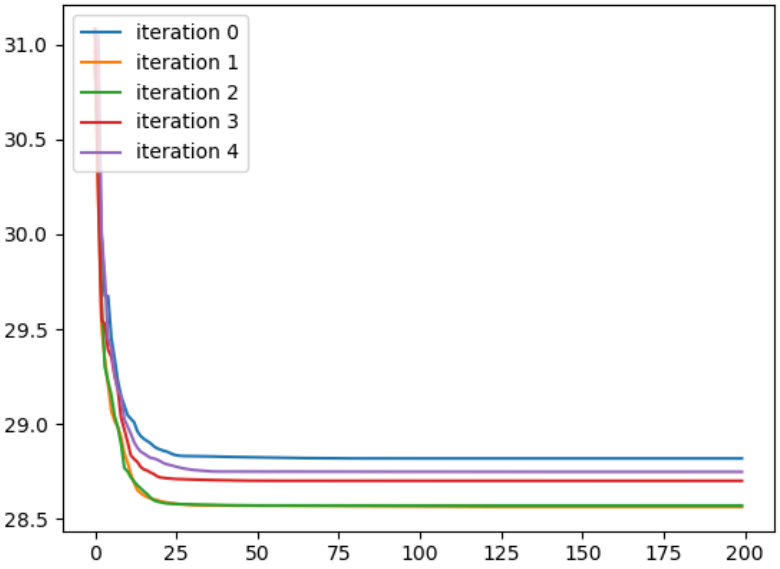
2.2.3 Ostali parametri

Parametri o kojima se nije diskutovalo ovde su p u funkciji koja dodeljuje težine tačkama, kao i parametri genetskog algoritma (veličina turnira pri turnirskoj selekciji, verovatnoća mutacije itd.). Pored toga, veličina roja i broj iteracija su u diskusiji gore ostali fiksni. Razlog tome je što su se vrednosti koje su inicijalno, nagađanjem, odabrane za ove parametre, pokazale dovoljno dobrim. U narednom radu bi moglo da se posmatra kako se ove vrednosti mogu dalje optimizovati.

3 Rezultati pokretanja algortima

3.1 Čist PSO

Veličina populacije je sa inicijalnog testiranja povećana na 50, broj iteracija na 200, i algoritmu su dati svih 2000 članaka. Parametri za PSO su postavljeni na vrednosti prodiskutovane u sekciji o odabiru parametara.



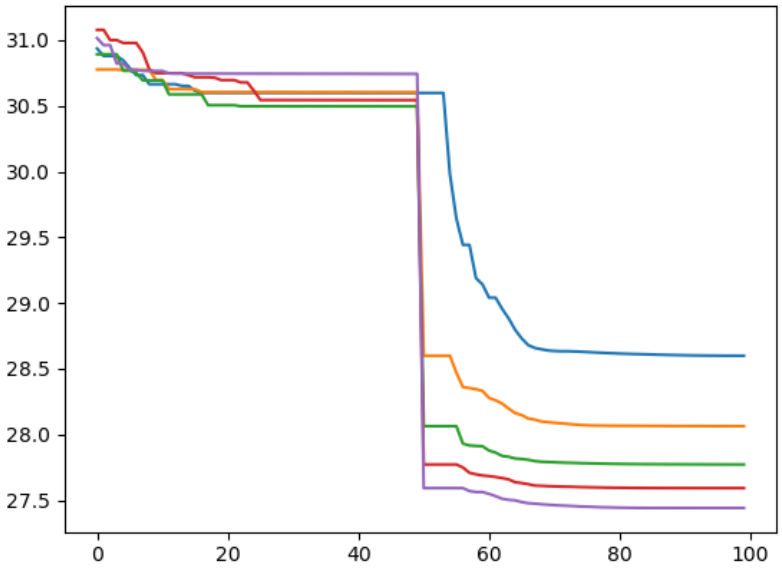
Slika 7: "čist"PSO

Prosečna dužina izvršavanja jednog klasterovanja je oko 30 minuta, što je veoma teško primeniti u realnim uslovima. S druge strane, drugi kriterijum zaustavljanja - npr. odsustvo promene u najboljoj vrednosti - može drastično da poboljša ove performanse, sa prekidom algoritma već oko pedesete iteracije.

3.2 PSO sa inicijalizacijom genetskim algoritmom

Jedna od glavnih otežavajućih okolnosti za rad bilo kog algoritma klasterovanja zasnovanog na prototipovima je što je veoma teško "opora-

viti se"od loše inicijalizovanih nasumičnih klastera. Ideja je ukupni broj predviđenih iteracija podeliti na dva dela - prvu polovinu iteracija koristi genetski algoritam, pa se njegov kranji rezultat koristi kao početni centroidi za PSO.[7]



Slika 8: hibridni PSO

4 Zaključak

Jedno od primarnih poboljšanja na ovom radu bi bilo optimizovanje implementacije. Implementirana na ovaj način, funkcija za dodeljivanje težina je ekstremno vremenski i prostorno zahtevna, a poziva se izuzetno često za vreme rada algoritma. Očekivano je da će kompleksnost izvršavanja biti veća nego pri običnom K-means algoritmu, i dimenzije obrađenog skupa dokumenata mogu da porastu do jako velikih proporcija, i zato je dobro što bolje i optimizovanije implementirati ključne delove koda.

Literatura

[1] Mehdi Allahyari i dr. "A brief survey of text mining: Classification, clustering and extraction techniques". U: *arXiv preprint arXiv:1707.02919* (2017.).

[2] Andries P. Engelbrecht. *Computational Intelligence. An Introduction*. 2. izdanje. Wiley, 2007. ISBN: 9780470035610,0470035617.

[3] Robert Tibshirani; Guenther Walther; Trevor Hastie. "Estimating the number of clusters in a data set via the gap statistic". U: *Journal Of The Royal Statistical Society* 63 (2 2001.), str. 411–423. ISSN: 0952-8385,1467-9868. DOI: 10.1111/1467-9868.00293.

[4] Amir Karami i dr. "Flatm: A fuzzy logic approach topic model for medical documents". U: *2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)*. IEEE. 2015., str. 1–6.

[5] Jiamin Li i Harold W Lewis. "Fuzzy clustering algorithms—review of the applications". U: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE. 2016., str. 282–288.

[6] Vipin Kumar Pang-Ning Tan Michael Steinbach. *Introduction to Data Mining*. Pearson, 2013. ISBN: 978-1-292-02615-2.

[7] K Premalatha i AM Natarajan. "Hybrid PSO and GA for global maximization". U: *Int. J. Open Problems Compt. Math* 2.4 (2009.), str. 597–608.