UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA

Milan Ž. Vukićević

RAZVOJ I PROJEKTOVANJE ALGORITAMA ZA KLASTEROVANJE EKSPRESIJA GENA

DOKTORSKA DISERTACIJA

Beograd, 2014.

UNIVERSITY OF BELGRADE FACULTY OF
ORGANIZATIONAL SCIENCES

Milan Ž. Vukićević

DEVELOPMENT AND DESIGN OF ALGORITHMS FOR CLUSTERING GENE EXPRESSION DATA

DOKTORAL DISSERTATION

Beograd, 2014.

Mentor:

Prof. dr Milija Suknović, redovni profesor
Univerzitet u Beogradu, Fakultet organizacionih nauka

Članovi komisije:

1. Prof. dr Boris Delibašić, vanredni profesor
Univerzitet u Beogradu, Fakultet organizacionih nauka

2. dr Dragan Radojević, naučni savetnik, Institut Mihajlo Pupin

Datum odbrane: _____ 2014. godine

Razvoj i projektovanje algoritama za klasterovanje ekspresija gena

Apstrakt

Grupisanje podataka (klasterovanje) ekspresija gena zauzima važnu ulogu u biomedicinskim istraživanjima. Osnovna karakteristika podataka o genskim ekspresijama je visoka dimenzionalnost podataka (preko 1000 atributa). Sa druge strane, troškovi uzorkovanja su jako visoki i u najvećem broju istraživanja ne postoji više od 200 ispitanika. Ovakav disbalans između broja atributa i veličine uzorka, predstavlja veliki izazov pri projektovanju i primeni algoritama klasterovanja, koji mogu dati dobre performanse. Autori u ovoj oblasti često naglašavaju važnost postojanja što većeg broja algoritama klasterovanja, kako bi analitičari imali mogućnost izbora algoritma koji najviše odgovara podacima koje analiziraju. Sa druge strane izbor najboljeg algoritma za konkretne podatke, je jedan od najizazovnijih zadataka u procesu klasterovanja.

Kao potencijalno rešenje upotrebljen je komponentni pristup za razvoj i primenu algoritama za otkrivanje zakonitosti u podacima (OZP). Ovaj pristup omogućava korisniku da manuelno ili automatski dizajnira veliki broj algoritama, kombinujući komponente iz postojećih algoritama iz literature. Ovaj pristup takođe omogućava korisniku uvid i uticaj na strukturu samog algoritma (a ne samo na parametre), kao i njeno jednostavno proširenje.

U ovoj disertaciji predstavljeno je proširenje postojeće arhitekture za dizajn algoritama klasterovanja čime je omogućen dizajn više od hiljadu različitih algoritama klasterovanja. Međutim, proširenjem prostora mogućih algoritama, izbor najboljeg metoda klasterovanja za posmatrani problem postaje još izazovniji zadatak.

Da bi se rešio ovaj problem, sprovedena je detaljna evaluacija razvijenih komponetnih algoritama klasterovanja nad više od 30 skupova podataka (preko 30000 eksperimenata). Na osnovu formirane eksperimentalne baze podataka, predložena su dva pristupa za selekciju i rangiranje algoritma za klasterovanje ekspresija gena.

Prvi pristup, koristi OZP modele (klasifikacija, asocijacija i regresija) kako bi automatski identifikovao algoritme koji daju dobre perfomanse modela nad novim podacima. Drugi pristup proširuje nedavno predloženi sistem meta-učenja za rangiranje i izbor algoritma za klasterovanje podataka o ekspresiji gena. Predloženi sistem proširuje postojeći prostor algoritama i koristi komponente kao meta-atribute za deskripciju algoritama.

Oba pristupa su implementirana i evaluirana na podacima o genskim ekspresijama koji su prikupljeni iz medicinskih baza podataka, a koji su korišćeni u prethodnim istraživanjima u oblasti klasterovanja. Eksperimentalna evaluacija je pokazala obećavajuće rezultate pri predviđanju performansi algoritama klasterovanja, njihovom rangiranju i selekciji, kao i potencijal za integraciju ova dva pristupa.

Ključne reči: klasterovanje, genske ekspresije, razvoj zasnovan na komponentama
Naučna oblast: Organizacione nauke
Uža naučna oblast: Modeliranje poslovnih sistema i poslovno odlučivanje
UDK broj: 519.816:004.6

Development and design of algorithms for clustering gene expression data

Abstract

Clustering of gene expression data plays an important role in biomedical research. High dimensionallty (over 1000 attributes) is the main characteristic of this type of data. On the other side, sampling costs are very high and in most research there are no more than 200 samples. Such disbalanse between number of attributes and number of samples makes a great challenge for design and application of good performing clustering algorithms. Authors in this area, often emphesise the importance of existance of large number of algorithms in order to provide analysts a possibility to use best suited algorithm for specific data at hand. On the other side, algorithm selection is one of the most challenging tasks in clustering process.

As a potential solution to this problem, component based approach for development and application of data mining algorithms is used for algorithm design and selection. This approach enables manual or automatic design of large number of algorithms, using components from the existing algorithms from the literature. Additionally, this approach enables influence on algorithm structure (not only parameters), as well as simple extension of this structure.

In this dissertation, extension of existing architecture for component based algorithm design is presented. Extended architecture enables design of more than a 1000 clustering algorithms. Nevertheless, existence of such a large space of possible algorithms makes the selection of the right algorithm for data at hand even more challanging task.

In order to tackle this problem detailed evaluation of the developed component based clustering algorithms is conducted on more than 30 datasets (over 30000 experiments). Using experimental database formed from the experimental results, two approaches for selection and ranking of clustering algorithms are proposed.

First approach is using data mining models (classification, association and regression) in order to automatically identify good performing clustering algorithms for new data. Second approach extends the existing meta-learning system for ranking and selection of algorithms for clustering gene expression data. Proposed system extends the algorithm space and uses algorithm components as meta-attributes (algorithm descriptions).

Both approaches are implemented and evaluated on on the gene expression data that are gathered from medical databases, and are used in previous researches in the clustering area. Experimental evaluation showed promising results in prediction of clustering algorithm performance, their ranking and selection, as well as a potention for integration of these two approaches.

Keywords: clustering, gene expression data, component based approach
Scientific field: Organizational sciences
Field of scientific expertise: Business system modeling and business decision making
UDK code: 519.816:004.6

Sadržaj

1 Uvod

Grupisanje podataka (klasterovanje) je jedan od fundamentalnih problema računarskih nauka kao što su: mašinsko učenje, otkrivanje zakonitosti u podacima - OZP (eng. data mining) ili otkrivanje paterna (Ene et al., 2011). Klasterovanje je korišćeno u mnogim oblastima primene kao što su: bio-informatika (Ayady et al., 2012; Baralis et al., 2011), OZP na web-u (Wan et al., 2011), grupisanje dokumenata (Ayady et al., 2012; Kalogeratos and Likas, 2011; Chen and Tseng, 2010), grupisanje tokova podataka (eng. *data streams*) (Da Silva et al., 2011), geologija (Grujić et al., 2012), Edukativni OZP (Jovanović et al., 2012; Romero and Ventura, 2011) itd.

U okviru primene klaster algoritama u bioinformatici, posebno mesto zauzima klasterovanje podataka o ekspresiji gena koji se beleže u mikronizovima (Andreopoulos et al., 2009). Ova oblast je bazirana na kvantitavnoj analizi podataka koji omogućavaju razumevanje funkcija gena, regulacije gena, funkcije ćelija, pod-tipova ćelija itd. Geni koji imaju sličnu ekspresiju mogu da se grupišu jer poseduju slične ćelijske funkcije. Ovim putem se omogućava dodatno razumevanje funkcija mnogih gena za koje informacije dosad nisu bile dostupne, a neophodne su za razumevanje i dijagnostiku određenih bolesti.

Skupovi podataka o ekspresiji gena sadrže merenja smanjenja ili povećanja ekspresije gena po vremenskim tačkama, uzorcima tkiva ili pacijentima. Oni su predstavljeni kao matrica numerickih vrednosti. Tehnologije DNK mikročipova omogućavaju simultano praćenje velikog broja karakteristika genskih ekspresija sa malom količinom grešaka. Takođe, zamenom standardnih dijagnostičkih alata dolazi se, prvenstveno zbog zamene ekspertskog znanja, do smanjenja troškova, smanjenja vremena obrade, kao i mogućnost obrade bez domenskog znanja.

Imajući u vidu gore navedeno, primenom adekvatnih algoritama klasterovanja, bila bi omogućena pravovremena identifikacija i delovanje na različite bolesti uz smanjenje troškova. Međutim, jedan od najvećih problema kod ove analize je izbor pravog algoritma klasterovanja za konkretan skup podataka (Iam-on et al., 2010). Ovaj

problem je uzrokovan činjenicom da performanse algoritama klasterovanja zavise od konkretnih osobina skupa podataka (Giancarlo et al., 2010). Zbog toga (Quackenbush, 2001) tvrdi da je izbor odgovarajućeg algoritma ključni element eksperimentalnog dizajna u analizi ovih podataka.

Dodatni problem pravi činjenica da su tipični mikronizovi koji predstavljaju ekspresiju gena, specifični zbog toga što imaju mali broj slučajeva (instanci), dok je broj polja koja predstavljaju gene (atribute) veoma veliki (Piatetsky-Shapiro and Tamayo, 2003). Zbog toga je analiza ovih podataka veoma kompleksan problem (Shao et al., 2011).

Različiti tipovi algoritama klasterovanja su korišćeni u analizi podataka o ekspresiji gena (Andreopoulos et al., 2009): hijerarhijski, algoritmi bazirani na mrežama, gustini, grafovima itd. dok posebno mesto zauzimaju partitivni algoritmi, koji mogu da se koriste u originalnom obliku (Ayady et al., 2012; Geraci et al., 2009), kao unapređeni algoritmi (Geraci et al., 2009) ili kao integralni delovi konsenzus šema algoritama (Giancarlo and Utro, 2011; Monti et al., 2003; Xu and Wunsch, 2010; Punera and Ghosh, 2008). Postojanje velikog broja algoritama u svakoj od pomenutih klasa unosi dodatni problem pri izboru algoritma za konkretan skup podataka. Iako postoje neke preporuke za izbor algoritma za klasterovanje bioloških podataka (Andreopoulos et al., 2009), ne postoji kosenzus o najboljem algoritmu za ovako težak zadatak grupisanja.

Kao potencijalno rešenje za ovaj problem, komponentni pristup u dizajnu algoritama klasterovanja mogao bi dati obećavajući pravac istraživanja. Komponentni pristup koji je predložen od strane (Delibašić et al; 2009, Suknović et al., 2012) omogućava kolaborativni dizajn velikog broja (hiljade) particionih algoritama klasterovanja koji su sastavljeni od različitih delova i poboljšanja originalnih algoritama iz ove klase. Postojanje ovako velikog broja algoritama povećava šanse za pronalaženje algoritama koji daju dobre performanse na specifičnim skupovima podataka kao što su podaci o ekspresijama gena.

Sa druge strane, postojanje ovako velikog broja algoritama analitičarima dodatno otežava proces izbora adekvatnog algoritma (Iam-on et al., 2010) i zbog toga je veoma

važno kreirati metodologiju za automatsku selekciju algoritama klasterovanja koji imaju dobre performanse pri radu sa podacima o ekspresijama gena.

Dakle **generalni cilj** istraživanja predstavljenog u ovoj disertaciji je *razvoj originalne metodologije za projektovanje, evaluaciju i selekciju algoritama za klasterovanje podataka o ekspresiji gena.*

Ovaj generalni cilj će biti ostvaren kroz nekoliko **posebnih ciljeva**:

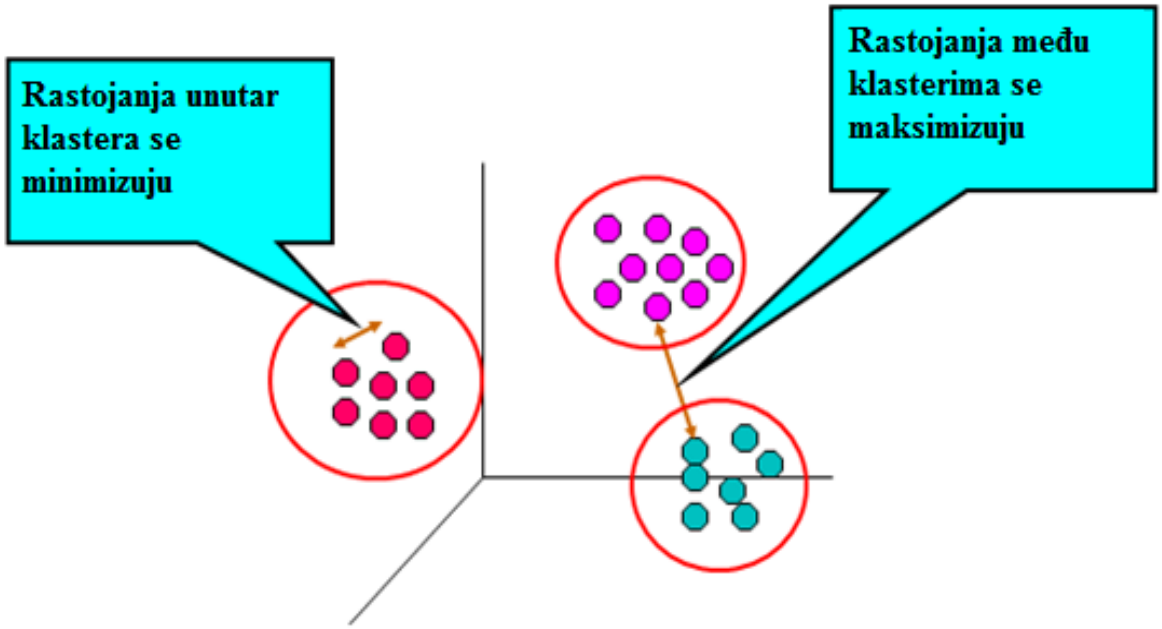
- Predlog arhitekture za jednostavan dizajn komponentnih algoritama klasterovanja;
- Evaluacija velikog broja algoritama klasterovanja nad velikim brojem skupova podataka o ekspresiji gena;
- Identifikacija adekvatnih mera evaluacije;
- Predlog automatizovanih sistema za selekciju i rangiranje algoritama za konkretan skup podataka uz pomoć metoda OZP (Vukićević et al., 2012a) kao i pristupa meta-učenja (De Souto et al., 2008, Nascimento et al., 2010, Radovanović et al., 2013).

2 Klasterovanje (grupisanje podataka)

Klasterovanje podrazumeva grupisanje slučajeva (objekata, instanci) u grupe, tako da su objekti unutar grupe slični među sobom, a između grupa različiti (Hartigan, 1975). Klaster predstavlja skup, tj. domen, u okviru koga se nalaze elementi sa zajedničkim osobinama. Grupe dobijene procesom klasterovanja uvek treba da zadovolje dva osnovna kriterijuma:

- Svaka grupa predstavlja homogen skup: objekti koji pripadaju istoj grupi su međusobno slučni.
- Svaka se grupa razlikuje u odnosu na ostale, tj. objekti koji pripadaju određenoj grupi značajno se razlikuju od onih koji pripadaju ostalim grupama.

Kod procesa klasterovanja teži se smanjenju rastojanja između objekata unutar jedne grupe a povećanju rastojanja između objekata u dve različite grupe što je prikazano na slici 2.1.



Slika 2.1. Definicija klastera

Klasterovanje predstavlja nenadgledani zadatak OZP, što znači da klasteri nemaju predefinisane klase objekata odnosno da algoritmi klasterovanja ne mogu da uče da kreiraju modele na osnovu "istinitih modela" (kao u slučaju klasifikacije) već se

objekti grupišu prema njihovoj međusobnoj sličnosti (geometrijskim osobinama), a korisnik treba da uvidi da li formirani klasteri imaju značaj za svoj slučaj primene. Jedna od najpopularnijih formalnih definicija klasterovanja je sledeća (Jain et al., 1999):

Klasterovanje je nenadgledana OZP tehnika koja deli skup podataka u K regiona na osnovu njihove sličnosti/različitosti koja se određuje na osnovu neke metrike, gde vrednost K može i ne mora da bude apriori poznata. Glavni cilj svake tehinke klasterovanja je da napravi matricu U(X), dimenzija K x N , za dati skup podataka X, koji se sastoji od n paterna, $X = x_1, x_2,..., x_n$, izdeljeni segmenti se mogu predstaviti kao $U=[u_{k,j}]$, k=1..K i j=1..n gde je $u_{k,j}$ član x_j paterna k-tog klastera.

Formirane grupe mogu biti definisane na različite načine (Jain et al., 1999):

- Identifikovane grupe mogu biti ekskluzivne i tada svaki objekat može pripadati samo jednoj grupi,
- Grupe se mogu preklapati i tada svaki objekat može pripadati većem broju grupa,
- Grupe mogu biti probablističke: svaki objekat može pripadati većem broju grupa,
- Grupe mogu posedovati hijerarhijsku strukturu: postoji gruba podela objekata na najvišem nivou, a finije strukturiranje se postiže na nižim nivoima.

2.1 Klasterovanje ekspresija gena

Mnoga biološka istraživanja se zasnivaju na analizi dezoksiribonukleinska kiselina (DNK). DNK skladišti sve genetske informacije potrebne za razvoj i funkcionisanje svih živih bića. DNK funkcioniše na sledeći način: deo DNK, koji se naziva gen, se prepisuje u molekule, koji se nazivaju mRNK, čiji je zadatak da prenese informacije do ribozoma koji, zatim, pretvara informacije u protein. Protein obavlja najviše funkcija ćelije od kojih su najbitnije regulisanje, prevođenje i transkripcija DNK. Nakon izgradnje proteina, kaže se da je gen izražen, tj. poseduje ekspresiju. Stoga se može zaključiti da je gen osnovni element DNK. Ekspresija gena se najčešće posmatra kao rezultat procesa sinteze proteina, što je veoma važno za utvrđivanje njegove biološke

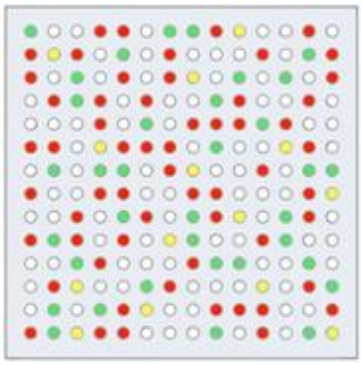
funkcije. Generalno gledano, ekspresije gena delimo na one koji se čuvaju u DNK i one koji se čuvaju u tkivima. One ekspresije koje se čuvaju u DNK su odgovorne za održavanje vitalnosti svih tipova ćelija u organizmu, tj. oni sprovode biološke funkcije koje su zajedničke za sve tipove ćelija. S druge strane, ekspresije gena koje su vezane za tkiva generišu proteine neophodne za funkcionisanje tog specifičnog tkiva. Jasno je uočljivo da mala promena (mutacija) u ekspresiji gena može prouzrokovati prenošenje pogrešne informacije do mRNK, koji će zatim tu pogrešnu informaciju pretvoriti u protein, koji će loše izvršavati svoju biološku funkciju. Rezultat ovoga mogu biti bolesti, a najčešće kancer.

Do kraja dvadesetog veka podatke o genskim ekspresijama domenski eksperti su dobijali isključivo posredstvom skupih aparata koji su imali visok nivo greške. Pored toga, rezultate su tumačili sami, bez računarske pomoći. Tumačenje je najviše zavisilo od iskustva eksperta, što je moglo da dovede do još veće greške. Bitno je napomenuti da je broj sondi, koji je merio vrednosti ekspresije gena, bio relativno mali. Ukupno oko 200 sondi. Iz navedenih razloga razvila se nova tehnologija za prikupljanje podataka. Reč je o DNK mikročipovima. DNK mikročipovi omogućavaju merenje više gena istovremeno. Međutim, merenje se obavlja nad, uslovno rečeno, samo nekoliko gena istovremeno, od ukupnog broja gena. Razlog tome je prvenstveno količina informacija. Naime, merenjem i samo nekoliko genskih ekspresija dolazi se do problema koji statističari nazivaju „p veće od N", gde je p broj atributa a N broj uzorka. Najnoviji mikročipovi uspevaju da koriste preko 50000 sondi, tj. broj atributa je preko 50000. Imajući u vidu da se ispituje najviše 200 ljudi, može se zaključiti da standardne statističke metode ne mogu da obrade ovoliki broj podataka. Drugi problem koji se javlja jeste da, čak i da postoji izlazna informacija za uzorak, algoritmi klasifikacije se ne snalaze sa ovim disbalansom u količini atributa i uzorka. Osnovni problem je u tome kako odrediti koji deo uzeti za treniranje, a koji za testiranje. Stoga se više napora ulaže u polje učenja bez nadgledanja, tj. klasterovanje. Klasterovanjem se omogućava identifikacija grupa koregulisanih gena, identifikacija prostornih i/ili vremenskih paterna u genskih ekspresijama, a najčešće utvrđivanje reda u ekspresijama koje su u najboljem slučaju ne potpuno nasumične. Trenutno postoje dve tehnologije za merenje

genskih ekspresija, tj. dva tipa mikročip tehnologije. To su cDNA i Affymetrix. Obe obavljaju tri osnovne procedure:

- Proizvodnja: Mikročip je mali čip sastavljen od stakla, najlonske membrane ili silikona na koje se mogu staviti desetine hiljada DNK molekula, tj. sondi u polja koja su postavljana u obliku rešetki.
- Priprema, obeležavanje i hibridizacija: Dva uzorka (jedan za testiranje i drugi za kontrolu) se prepisuju na čip, obeležavaju se fluorescentnom bojom ili radioaktivnim izotopom, i zatim hibridizuju na sondu na površini čipa.
- Skeniranje čipa: Skeniranjem čipa se čita jačina signala koji se emituje sa obeležene i hibridizovane površine.

Sami podaci genskih izraza se ne predstavljaju u obliku pogodnom za analitičku obradu. Kao što se vidi na slici 2.2 podaci se prikupljaju u matričnom obliku, međutim vrednosti nisu numeričke. Najčešće su redovi geni, a kolone čipovi (uzorak). Vrednost se predstavlja bojom. Stoga je prvi korak prevesti podatke u numeričke vrednosti. Najčešće se vrednost zelene boje prevodi u veću vrednost (pozitivnu), crvena u nižu vrednost (negativnu), dok se bela boja prevodi u vrednost blisku nuli.



Slika 2.2. Podaci ekspresije gena[1]

Uzmimo za primer matricu podataka koja je data na slici 2.3. U redovima se nalaze geni, a u kolonama čipovi (uzorak). Tako će se u polju $x_{1,1}$ nalaziti vrednost, tj. genska ekspresija gena 1 uzorka 1. U opštem slučaju, notacija $x_{i,j}$ odgovara ekspresiji gena i u uzorku j.

[1] Izvor: http://www.nature.com/scitable/content/microarray-chip-6603382

$$X = \begin{array}{c|cccc} \textit{Gene} & \textit{Chip1} & \textit{Chip2} & \cdots & \textit{Chip20} \\ \hline 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,20} \\ 2 & x_{2,1} & x_{2,2} & \cdots & x_{2,20} \\ 3 & x_{3,1} & x_{3,2} & \cdots & x_{3,20} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 12{,}000 & x_{12000,1} & x_{12000,2} & \cdots & x_{12000,20} \end{array}$$

Slika 2.3. Matrica ekspresije gena

Klasterovanjem ovakve matrice vrši se klasterovanje gena. Međutim, češće to nije zadatak klasterovanja. Češće je potrebno klasterovati uzorak u zavisnosti od gena. Takva matrica se dobija transponovanjem postojeće matrice. Ovim potezom redovi postaju kolone, a kolone redovi. Ukoliko pogledamo sliku 2.4 videćemo da imamo 12000 atributa i uzorak od dvadeset ispitanika. Vrednost u matrici X^T koja se obeležava sa $y_{i,j}$ predstavlja vrednost ekspresije za dati gen.

$$X^T = \begin{array}{c|cccccc} \textit{Chip} & \textit{Gene 1} & \textit{Gene 2} & \textit{Gene 3} & \cdots & \textit{Gene 12000} \\ \hline 1 & y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,12000} \\ 2 & y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,12000} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 20 & y_{20,1} & y_{20,2} & y_{20,3} & \cdots & y_{20,12000} \end{array}$$

Slika 2.4. Transponovana matrica ekspresije gena

Sve gore navedeno odgovara drugom koraku u CRISP-DM (Wirth and Hipp, 2000) metodologiji, razumevanju podataka. Nakon razumevanja podataka potrebno je transformisati podatke tako da oni budu u odgovarajućem obliku za obradu. Ispravna strategija za obradu, koja ne samo da uklanja što je moguće više šuma u podacima, već i obezbeđuje osnovu za poređenje genskih ekspresija je zaista od suštinskog značaja za efikasnu klaster analizu, i sastoji se iz sledećih pet koraka (Fridlyand, 2012):

- Normalizacija;
- Nelinearna transformacija;

- Zamena nedostajućih vrednosti;
- Filtriranje podataka;
- Standardizacija podataka.

Normalizacija predstavlja prvi korak u procesu pripreme podataka za obradu. Razlog tome je što se podaci za različite gene mere različitim skalama. Zbog toga vrednosti nisu uporedive, što može dovesti da algoritam bude naklonjeniji prema nekim genima u odnosu na druge. Svođenjem na istu meru, ispravlja se naklonjenost. Normalizacija se najčešće vrši uz pomoć L_2 ili L_∞ norme.

Razlog zašto se vrše nelinearnetransformacije je taj što se genski izrazi predstavljaju u dva kanala, test i kontrolni izrazi. Kako nije moguće svaki testni izraz iskontrolisati dešava se da izrazi koji nemaju odgovarajući kontrolni izraz (neregulisani izrazi) imaju vrednosti između jedan i beskonačno, dok testni izrazi koji imaju više odgovarajućih kontrolnih izraza imaju vrednost između nula i jedan. Ovakav šum u podacima se smanjuje logaritmovanjem izraza. Zbog ovog šuma linearne transformacije nisu moguće. Dodatni izvor šuma u podacima mogu biti i fotoreceptori koji beleže vrednosti ekspresije gena. Log-transformacijama se obezbeđuje vlasništvo nad celim opsegom signala.

Zamena nedostajućih vrednosti je veoma bitna u oblasti biomedicine i bioinformatike. Kako se, zbog simultanog prikupljanja podataka, dešava da se podaci ne snime na čip potrebno je te nedostajuće vrednosti popuniti. Imputacija aritmetičke sredine se pokazala kao dobra alternativa. Razlog tome je ogromna dimenzionalnost problema. Naime, razvijanje modela koji bi dodavao nedostajuću vrednost bi predugo trajao, te se stoga koriste jednostavnije metode.

Mnoge vrednosti ekspresija gena ne nose značajnu informaciju za posmatranu biološku funkciju ili se teško menjaju tokom vremena (pa nisu interesantni) ili su vrednosti za svaki uzorak konstantne. Ovakvi geni imaju beznačajan udeo u problemu te ih treba izbaciti iz analize. Pokazano je da linearne redukcija atributa (metoda glavnih komponenti) ne donosi dobar odnos u smanjenju dimenzije problema i opisane

varijanse. Stoga se češće primenjuju nelinearne transformacije. Najčešće se koriste Isomape, Laplasove sopstvene mape, nelinearna metoda principijelnih komponenti i slično.

Kako su biolozi najčešće zainteresovani za grupisanje gena koji imaju isto regulativno ponašanje potrebno je pronaći gene koje imaju slično relativno ponašanje, ali koje divergiraju. Stoga se često, u ovom poslednjem koraku, vrednosti standardizuju kako bi imale aritmetičku sredinu nula i standardu devijaciju jedan.

Međutim, nije dovoljno ispoštovati navedene korake kako bi se dobili dobri rezultati klasterovanja. Ključni koraci, prilikom primene klaster algoritma, prema (Fridlyand, 2012) su: izbor mere odstojanja i izbor algoritma klasterovanja. Merenje odstojanja se pokazuje kao krucijalan faktor u klasterovanju genskih ekspresija. Najčešće se koriste Euklisko i Menhetn odstojanje, ali često se koristi i korelaciono odstojanje. Euklidsko i Menhetn odstojanje mere apsolutno odstojanje između dva vektora (uzorka), i smatra se da je Menhetn otporniji na izuzetke. U eksperimentalnom delu ovog rada će poseban fokus biti na izboru adekvatne mere odstojanja za klasterovanje ekspresije gena. Međutim, ako je potrebno meriti trend, tj. relativno odstojanje onda je bolje koristiti korelaciono odstojanje. Algoritmi klasterovanja koji se koriste u ovoj oblasti obično su jednostavni i njihova kompleksnost je niska zbog velike dimenzionalnosti problema i zbog toga se u ovom radu posebno analiziraju algoritmi bazirani na K-means algoritmu (Lloyd, 1982).

2.2 Algoritmi za klasterovanje ekspresija gena

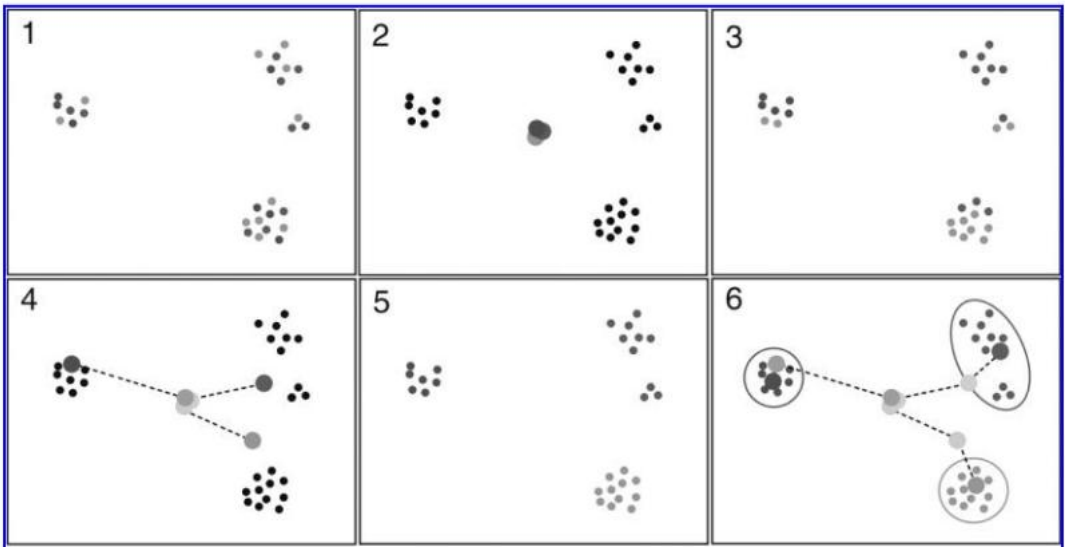
U ovoj sekciji dat je pregled metoda za klasterovanje podataka o ekspresiji gena, koji su usko vezani sa komponentnim pristupom prikazanim u ovom radu sa posebnim naglaskom na algoritme bazirane na predstavnicima i hijerarhijske algoritme. Iscrpni pregledi literature koja je generalno vezana za klasterovanje u ovoj oblasti mogu se pronaći u (Xu and Wunsch, 2010; Belacel et al., 2006). (Xu and Wunsch, 2010) su dali temeljni pregled upotrebe i evaluacije algoritama klasterovanja u biomedicinskim istraživanjima. Oni naglašavaju važnost pronalaženja adekvatnih algoritama za različite

biomedicinske primene. (Belacel et al., 2006) su dali pregled tehnika klasterovanja za primenu nad podacima o ekspresiji gena i u ovoj sekciji će detaljnije biti analizirani algoritmi identifikovani u (Belacel et al., 2006).

2.2.1 Klastering baziran na predstavnicima

U modelima baziranim na predstavnicima[2], klasteri se predstavljaju kao jedan vektor, koji ne mora da bude neki od slučaja u tom skupu podataka. Ako pretpostavimo da treba da formiramo K klastera, problem bi se mogao opisati kao "pronađi K centara klastera i dodeli slučaje najbližem klasteru, tako da kvadratno rastojanje od klastera (njegovog centra) bude minimalno". Ovaj problem je np-težak i zbog toga su razvijeni različiti algoritmi (heuristike) koji aproksimiraju optimalno rešenje ovog problema u prihvatljvom vremenu. U ovu grupu spadaju algoritmi kao što su: K-Means (Hartigan and Wong, 1979), K-Means++ (Arthur and Vassilvitskii, 2007) ili G-Means (Hamerly and Elkan, 2003).

K-means je intuivnan i jedan najčešće primenjivanih algoritama iz ove klase. Prvi korak u procesu klasterovanja je slučajna selekcija K objekata gde svaki izabrani objekat predstavlja inicijalni klaster, odnosno inicijalnog predstavnika klastera.



Slika 2.5. Postupak klasterovanja K-means algoritmom (Belacel et al., 2006)

[2] Često se u literaturi ova klasa algoritama naziva klastering baziran na centroidima, međutim centroidi su samo jedan od načina izračunavanja predstavnika klastera (npr. moguće je računati medoide), što će biti detaljnije objašnjeno u ovoj sekciji.

Nakon toga se objekti dodeljuju svojim najbližim predstavnicima. Kada su objekti dodeljeni predstavnicima, pristupa se procesu preračunavanja novih predstavnika za svaki klaster i objekti se ponovo dodeljuju. Ovaj iterativni postupak se zaustavlja kada prestane promena predstavnika. Ovaj postupak je ilustrovan na slici 2.5.

Jedan od najvećih problema pri korišćenju K-means algoritma je da on najčešće nalazi samo lokalno optimalno rešenje koje je jako zavisno od izbora inicijalnih predstavnika. Zbog toga se ovaj algoritam pokreće više puta sa različitim inicijalnim predstavnicima i odabira se najbolje rešenje. Praksa je da se koristi ograničenje da inicijalni centroidi pripadaju skupu podataka koji testiramo, čime dobijamo bolja rešenja. Najveća mana ovog tipa algoritma je ta što zahteva od korisnika da na početku definiše broj klastera. Takođe, algoritam funkcioniše tako da uglavnom pravi klastere približno slične veličine, čime dolazi do grešaka na ivicama klastera, što proizilazi iz toga da algoritam optimizuje centre klastera, a ne njihove granice.

K-means pati od mnogih mana od kojih su najčešće kritikovane:

- modeli su zavisni od izbora početnih centroida,
- centroidi se često nalaze u zamci lokalnog optimuma,
- prisistvo ekstremnih vrednosti jako utiče na rezultate klasterovanja,
- pretpostavlja podjednaku važnost svih atributa,
- broj klastera zavisi od korisnika itd.

I pored svih navedenih i često kritikovanih problema koje ovaj algoritam ima, on je "daleko najpopularniji algoritam u naučnim i privrednim primenama" (Berkhin, 2006) i svrstan je u "top 10 algoritama" OZP (Wu et al., 2007). Razlog tome je što ovaj algoritam poseduje sledeće kvalitete:

- jednostavan je za upotrebu,
- intuitivno je razumljiv,
- računski je efikasan i relativno skalabilan kada se procesuiraju veliki skupovi podataka,
- konvergira lokalnom optimumu u malom broju iteracija,
- implementiran je i dostupan u većini OZP okruženja.

Iz ovih razloga K-means je često korišćen za klasterovanje podataka o ekspresiji gena i vrlo često su njegove mane otklanjane tako što su razvijani algoritmi koji kao osnovu koriste K-means.

U daljem tekstu biće dat pregled algoritama koji unapređuju K-means kao i njihove primene kod klasterovanja podataka o ekspresiji gena, a još detaljniji pregledi se mogu naći u (Kumar and Wasan, 2010) koji su dali komparativnu analizu algoritama baziranih na K-means algoritmu u primeni nad podacima o ekspresiji gena, kao i u (Dhiraj and Rath, 2009) koji su detaljno ispitali ponašanje K-means algoritma nad ovim podacima.

K-means algoritam je proširivan i unapređivan na puno različitih načina (npr. uključivanje donje i gornje granice za broj klastera, spajanje i razdvajanje klastera pri određivanju optimalnog broja klastera ili poboljšanje način inicijalizacije klastera). Sistematski pregled većine ovih algoritama dali su (Bock, 2007) and (Jain, 1999). U ovoj sekciji će biti dat pregled popularnih i skoro predloženih poboljšanja algoritama K-means tipa i ukratko nekoliko često korišćenih platformi za OZP kao i njihove raspoložive fleksibilnosti za klaster algoritme. Globalni K-means kao i njegove modifikacije (Lai, 2010; Bagirov, 2008; Hansen et al., 2005; Likas et al., 2003) predlažu rešenja za izbegavanje zamke lokalnog optimuma. (Bouras et al., 2010) koriste K-means sa kosinusnom merom sličnosti za klasterovanje dokumenata. (Tang et al., 2010) su razvili novi fuzzy-c-means algoritam baziran na pristupu otežanih atributa. (Mumtaz and Duraiswamy, 2010) su predložili DBK-means algoritam kao hibrid između DBScan i K-means algoritma. (Žalik, 2008) je predložila K-means algoritam u kome korisnik ne mora sam da određuje tačan broj klastera. Ovo je postignuto optimizacijom funkcije cilja koja minimizuje originalnu srednju kvadratnu grešku. PCA analiza je korišćena za redukciju dimenzija i poboljšanje inicijalizacije u (Tajunisha and Saravanan, 2010). (Fahim et al., 2009, Yedla et al., 2010) su razvili i evaluirali tehnike za identifikaciju poboljšanih inicijalnih centroida kod K-means-a.

Kako bi se poboljšala robusnost na šum i ekstremne vrednosti, predstavljen je K-medoid algoritam (Mercer and College, 2003). Medoid je reprezentativna tačka u klasteru, izabrana od strane algoritma (K medoid predstavlja K klastere). Korišćenje medoida

ima dve prednosti: prva, nema ograničenja kod tipova atributa; i druga, medoidi su postojeće tačke u klasteru i stoga su, za razliku od centroida, generisani bez računanja - računa se samo aritmetička sredina (Berkhin, 2006). Samim tim, K- medoid algoritam je manje osetljiv na ekstremne vrednosti. Najpopularniji K-medoid algoritam je PAM (eng. *Partitioning Around Medoids*) (Kaufman and Rousseeuw, 1990), i njegovo unapređenje PAMSIL (van der Laan et al., 2003). PAMSIL zamenjuje funkciju cilja korišćenu u PAM sa prosečnom siluetnom vrednošću (eng. *average sihlouette*). Particionisanje u ovom slučaju zavisi ne samo od toga u kojoj meri objekat pripada datom klasteru, već i od toga u kojoj meri pripada sledećem najbližem klasteru. Eksperiment na simuliranim podacima mikroniza je pokazao da PAMSIL može da pronađe male homogene klastere (van der Laan et al., 2003).

Kao većina metoda koje koriste particiono klasterovanje, pri korišćenju K-means-a je neophodno znati tačan broj klastera. Nekoliko istraživanja je bilo usmereno ka razvijanju metode koja može da odredi broj klastera. (Yeung et al., 2001) su koristili probabilističke modele kako bi rešili problem optimalnog broja klastera. (Hruschka et al., 2006) su predstavili EAC (eng. *evolutionary algorithm for clustering*). EAC uključuje K-means algoritam kao lokalnu proceduru pretrage, koristi funkciju cilja baziranu na centroidima, eliminiše "cross-over" operator i dodaje sofisticirane operatore mutacije. EAC proširuje genetski algoritam klastera i može automatski da otkrije optimalni broj klastera.

Uobičajena kritika K-means algoritma je da on konvergira samo ka lokalnom optimumu. Neke tehnike optimizacije, kao što su tabu pretraživanje, simulirano kaljenje i genetski algoritmi su korišćeni kako bi se postigla globalna optimizacija. Međutim, ovi algoritmi su skupi, kako vremenski, tako i računski. U pokušaju da smanje troškove, (Krishna and Narasimha Murty, 1999) su predstavili nov metod klasterovanja GKA (*genetic K-means*). GKA hibridizuje genetski algoritam sa algoritmom najbržeg pada i K-means algoritmom. Rafinisanjem mutacije bazirane na odstojanju, umesto skupog "cross-over" operatora, GKA konvergira globalnom optimumu brže nego drugi evolutivni algoritmi. (Lu et al., 2004b) su predstavili brzi genetski K-means algoritam

(FGKA) koji uključuje efikasnu evaluaciju funkcije cilja (varijacije unutar klastera) i simplifikaciju mutacija.

(Lu et al., 2004a) su dizajnirali inkrementalni genetski K-means algoritam (IGKA) koji ostvaruje bolje vremenske performanse sa smanjenjem verovatnoće mutacije. (Shai et al., 2003) su koristili K-means analizu klastera kako bi identifikovali molekularne subtipove glioma. (Bayá and Granitto, 2011) su analizirali uticaj različitih mera odstojanja na performanse PAM i hijerarhijskih algoritama. Oni su takođe predložili novu meru odstojanja koja je bazirana na algoritmu najbližih suseda. (Baralis et al., 2011) su predložili novu meru sličnosti gena, baziranu na sposobnosti da odvaja slučajeve koji pripadaju različitim klasama. (Giancarlo et al., 2010) su analizirali uticaj različitih mera odstojanja na performanse hijerarhijskog i K-means algoritma klasterovanja i predložili su Pirsonovo (*Pearson*), Kosinusno i Euklidsko odstojanje kao adekvatna za klasterovanje podataka o ekspresiji gena. (Cheung, 2003) je predložio generalizovani K-means algoritam koji je nazvan K*-means. On je baziran na kompetitivnom učenju kažnjavanja rivala (*eng. rival penalized competitive learning - RPCL*), a generalizacija je postignuta u smislu da korisnici ne moraju da odrede tačan broj klastera, već samo gornju granicu mogućeg broja klastera.

Raspoloživi algoritmi klasterovanja se takođe mogu analizirati sa aspekta OZP softvera. Zbog toga je urađena analiza softvera otvorenog koda, kao i nekih komercijalnih softvera (IBM SPSS Modeler, Matlab). Dublja analiza pomenutog softvera bi bila van okvira ovog rada, pa se ovde daju samo generalni zaključci. Svi analizirani softveri (Tabela 2.1) imaju veliki broj algoritama baziranih na predstavnicima. JAVA-ML ima nekoliko algoritama koji su slični K-means-u (npr. originalni K-means, K-medoids), i nudi fleksibilnost u izboru mera odstojanja/sličnosti kao i različite mere evaluacije. Orange takođe omogućava izbor nekoliko mera sličnosti/odstojanja kod K-mkeans algoritma. Tanagra implementira fleksibilnosti korišćenjem različitih metoda za "Ažuriranje predstavnika". Weka takođe nudi fleksibilnost pri izboru različitih mera odstojanja/sličnosti. Ona omogućava samo kombinovanje euklidskog odstojanja sa ažuriranjem predstavnika na osnovu aritmetičke sredine (MEAN). Za sva ostala odstojanja koristi medoide (predstavnici koji su najmanje udaljeni od svih ostalih

objekata u klasteru). RapidMiner ne obezbeđuje mnogo fleksibilnosti za K-means, ali umesto toga obezbeđuje puno mera koje mogu raditi nad numeričkim, kategoričkim i mešanim (numerički i kategorički) tipovima atributa. Autori RapidMiner i WEKA softvera nisu dozvolili kombinovanje originalnog K-means algoritma sa ostalim merama odstojanja, pošto su verovatno bili svesni da aritmetička sredina optimizuje unutar-klaster odstojanje, samo u Euklidskom prostoru. Ipak kosinusno odstojanje pokazuje dobre performanse u kombinaciji sa aritmetičkom sredinom pri klasterovanju dokumenata. CLUTO softver takođe omogućava fleksibilnost izbora mera sličnosti/odstojanja, pri čemu je kosinusno odstojanje postavljeno kao podrazumevano. Ovaj softver takođe nudi hijerarhijsko-divizionu strategiju binarnog particionisanjam, kao i različite mere evaluacije. R softverski paket nudi četitri različite verzije K-means-a (Lloyd, MacQueen, Forgy, Hartigan and Wong).

Tabela 2.1. OZP softveri otvorenog koda

Softver	***Referenca***
RapidMiner	(Mierswa et al., 2006)
Weka	(Hall et al., 2009)
Tanagra	(Rakotomalala, 2005)
Orange	(Demsar et al., 2004)
R	(R Development Core Team, 2008)
KNIME	(Berthold et al., 2006)
JAVA-ML	(Abeel et al., 2009)
CLUTO	(Karypis, 2002)

Matlab daje nekoliko mogućnosti za inicijalizaciju predstavnika kao i mogućnost rešavanja problema praznih klastera u K-means algoritmu. Takođe Matlab koristi "batch" i "online" klastering ("batch" klastering preračunava predstavnike tek nakon što su svi objekti dodeljeni klasteru, dok "online" klastering, preračunava predstavnike nakon svakog dodavanja objekta klasteru), ali ih koristi sekvencijalno npr. prvo izvrši "batch" K-means a zatim izvrši "online". Sa druge strane, IBM SPSS Modeler ne nudi fleksibilnosti u okviru K-means algoritma. Moguće je menjati samo broj iteracija izvršavanja.

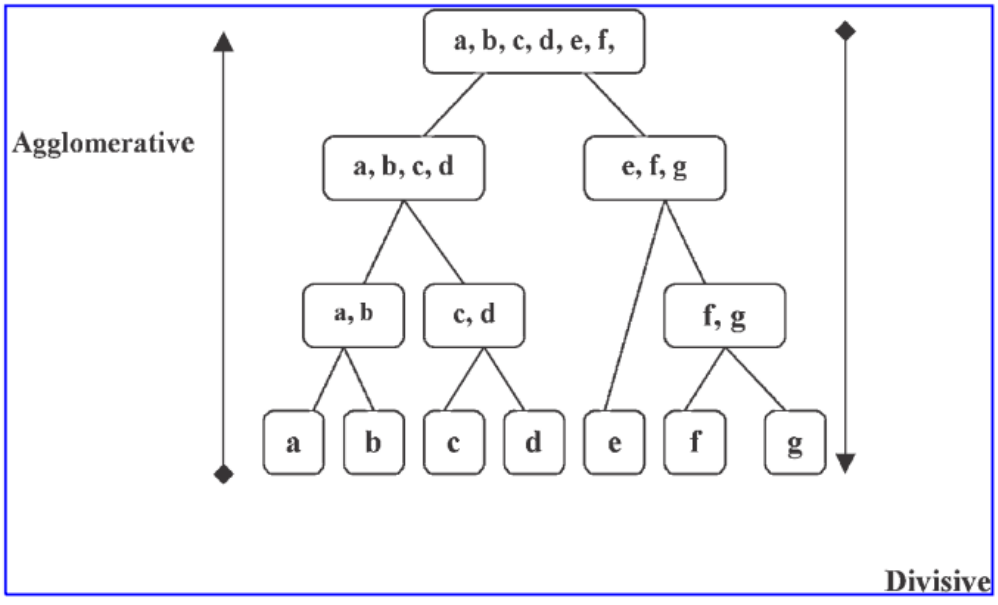
Generalni zaključak je da u softverima klaster algoritmi omogućavaju neke fleksibilnosti, posebno kada su u pitanju mere sličnosti/odstojanja. Fleksibilnosti u

algoritmima nisu standardizovane i ne postoji softver koji omogućava integraciju svih fleksibilnosti koje su identifikovane u (Delibašić et al., 2009).

Iz prethodnog pregleda, može se zaključiti da su za većinu nedostataka K-means algoritma predložena unapređenja u literaturi ili implementirana u OZP okruženjima. Problem je u tome što su najčešće ta unapređena parcijalna i što kod klasičnog pristupa u razvoju algoritama nije moguće kombinovati unapređenja u okviru jednog algoritma. Ovaj problem je jedna od glavnih motivacija ovog rada i biće detaljno razmatran u sekciji 3 i sekciji 4.

2.2.2 Hijerarhijski klastering

Algoritmi iz ove klase grupišu objekte u stablo klastera po aglomerativnom ili divizionom principu. Aglomerativno klasterovanje predstavlja pristup grupisanja od dna ka vrhu (eng. *bottom-up*) gde se na početku procesa svaki objekat posmatra kao poseban klaster. Zatim se najbliži parovi klastera spajaju, sve dok se svi objekti ne nađu u istom klasteru ili dok se ne ispuni neki od kriterijuma zaustavljanja. Diviziona strategija klasterovanja primenjuje strategiju od vrha ka dnu (eng. *top-down*). Svi objekti se na početku posmatraju kao jedan klaster, a zatim se klasteri dele, dok svaki objekat ne postane poseban klaster ili se ne ispuni neki od kriterijuma zaustavljanja (Slika 2.6).



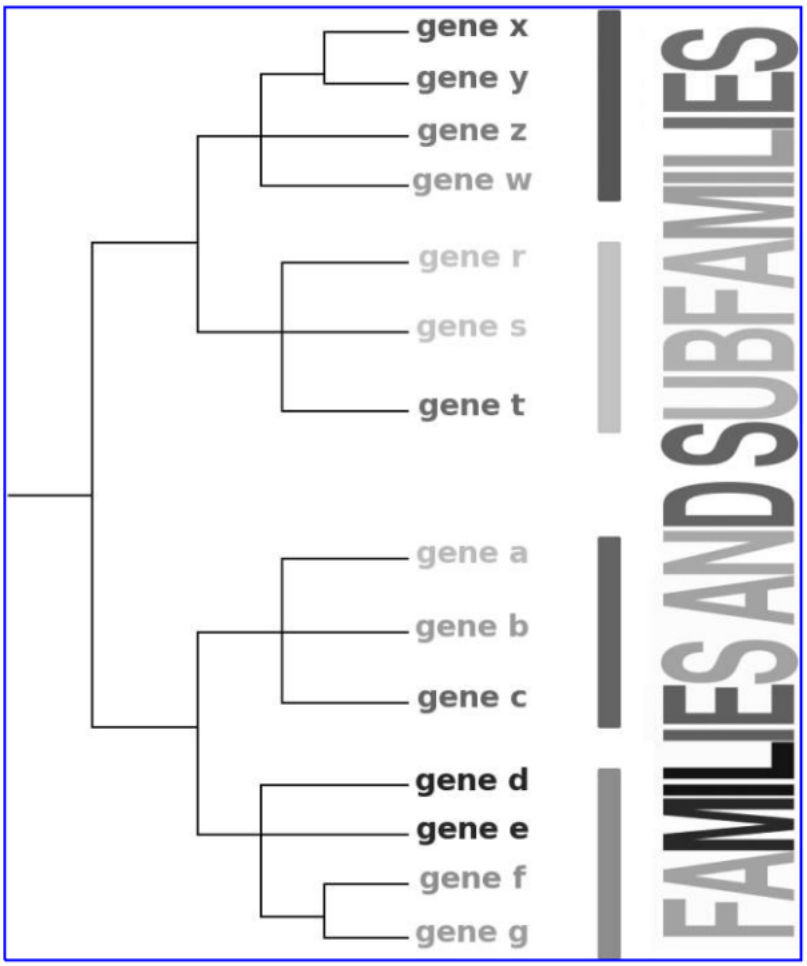
Slika 2.6. Diviziono i aglomerativno hijerarhijsko klasterovanje (Belacel et al., 2006)

Prema načinu merenja udaljenosti između klastera, algoritmi iz ove klase se mogu podeliti na sledeće tipove:

- Algoritmi sa jednostrukim povezivanjem (*single linkage clustering*) - rastojanje između dva klastera se računa kao rastojanje između njihovih najbližih objekata. Svaki objekat klasterovan ovom metodom je bliži makar jednom objektu unutar svog klastera nego bilo kom drugom objektu koji pripada drugom klasteru.
- Algoritmi sa kompletnim povezivanjem (*complete linkage clustering*) - rastojanje između dva klastera se računa kao rastojanje između njihovih najdaljih objekata. Ova metoda kreira klastere sa objektima koji su najudaljeniji jedni od drugih u okviru različitih klastera, odnosno proizvodi klastere čiji objekti leže unutar maksimalnog odstojanja između slučajeva u skupu podataka.
- Algoritmi sa prosečnim povezivanjem (*average linkage clustering*) - rastojanje između dva klastera se računa kao rastojanje između njihovih centroida (prosečnih vrednosti svih elemenata klastera).

Kod divizionog klasteringa za klaster sa n objekata postoji $(2^{n-1}-1)$ mogućih načina deljenja objekata u dve podgrupe. Određivanje svih mogućih divizija je zbog toga vremenski i računski skupo, posebno za klasterovanje ekspresija gena (Xu and Wunsch, 2005) i zbog toga ovaj tip klasterovanja nije često korišćen u praksi za ovaj problem.

Izlaz iz hijerarhijskog algoritma klasterovanja je dvodimenzionalni dendrogram, a grane dendrograma prikazuju grupisanje podataka. Dužina horizontalnih grana prikazuje sličnosti između klastera (slika 2.7).



Slika 2.7. Dendrogram hijerhijskog klasterovanja (Belacel et al., 2006)

Metode hijerarhijskog klasterovanja su detaljno analizirane kod primene na ekspresije gena, kao i kod drugih tipova mikronizova (npr. CGH nizova, proteinskih nizova). Kod primene na analizu kancera hijerarhijsko klasterovanje je korišćeno za identifikaciju tipova kancera (Nielsen et al., 2002; Ramaswamy et al., 2003), za otkrivanje novih podtipova kancera (Alizadeh et al., 2000), i za istraživanje mehanizama geneze tumora (Welcsh et al., 2002) iz podataka o ekspresiji gena. (Au et al., 2004) su istraživali hijerarhijsko klasterovanje za identifikaciju različitih podgrupa kod ćelija karcinoma pluća. (Makretsov et al., 2004) su primenili hijerarhijsko klasterovanje na višestruko obeležene mikronizove za identifikaciju specifičnih proteina (eng. *multiple marker microarray immunostaining data*) koji su rezultovali poboljšanom prognozom kod pacijentkinja sa invazivnim kancerom dojke. Hijerarhijsko klasterovanje je takođe korišćeno (Mougeot et al., 2006) za profilisanje ekspresija gena tkiva jajnika i pokazali

su da se klasterovanjem može utvrditi razlika između niskog potencijala malignosti/rane faze kancera i mogućih pred-kancerogenih stanja. Grafička reprezentacija hijerarhijskog klasterovanja omogućava korisnicima da vizualizuju globalne paterne u podacima o ekspresiji gena. Zbog toga je ova metoda jedna od najčešće primenjivanih u biološkim istraživanjima. Ipak, nekoliko ključnih problema kod hijerarhijskog klasterovanje i dalje nisu rešeni. Najozbiljniji problem ovog metoda (Jiang et al., 2004) jeste nedostatak robusnosti na šum, visoku dimenzionalnost i ekstremne vrednosti (eng. *outliers*). Ovi algoritmi su takođe skupi, u smislu i vremenske i računske složenosti (Xu and Wunsch, 2005) i zbog toga je primena ovog algoritma na velike skupove podataka ograničena. Takođe, oba pristupa (aglomerativni i divizioni) koriste takozvanu (eng. *greedy*) pohlepnu strategiju koja sprečava rafinisanje klastera. To znači da kada se jednom donese odluka da se spoje (ili razdvoje) klasteri oni se dalje ne razmatraju i ne optimizuju. Iterativno spajanje klastera nije određeno globalnim kriterijumom, već se iterativno u svakom koraku određuje lokalni optimum uz pomoć odstojanja parova objekata (Tan et al., 2005).

Nekoliko alternativnih pristupa je predloženo za rešavanje problema standardnog hijerarhijskog klasterovanja. Neki od ovih pristupa koriste particione algoritme (bazirane na predstavnicima), kao sto je K-means da bi generisali inicijalne klastere, a zatim, primenili hijerarhijsko klasterovanje koje koristi te klastere kao inicijalne tačke. Da bi se rešili problemi primene na velikim skupovima podataka predložena su neka unapređenja hijerarhijskog klasterovanja. Jedno od njih je CURE koje ima za cilj da bude robusno na ekstremne vrednosti (eng. *outliers*) i da identifikuje klastere sa nesferičnim oblicima i klastere različitih veličina (Guha et al., 1998). U ovom algoritmu svaki klaster je predstavljen sa fiksnim brojem dobro raspršenih (eng. *scattered*) tačaka, a korišćenje više predstavnika jednog klastera omogućava CURE algoritmu da identifikuje ne-standardne oblike klastera. CURE je takođe manje osetljiv na ekstemne vrednosti pošto se njihov uticaj umanjuje tokom procesa skupljanja rasejanih tačaka prema predstavniku. CURE koristi slučajno uzorkovanje i particionisanje kako bi skalirao veliki skup podataka bez gubitka kvaliteta klastera. Kompleksnost ovog pristupa raste linearno sa povećanjem veličine skupa podataka, dok njegova vremenska kompleksnost nije lošija od tradicionalnih hijerarhijskih algoritama. Još jedan značajan

algoritam je BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (Zhang et al., 1996). BIRCH kreira posebnu strukturu koja se zove stablo atributa klastera (eng. *clustering feature (CF) tree).* BIRCH algoritam skenira bazu podataka, a rezultati se čuvaju u memoriji u formi CF stabla. Tokom faze pred-procesiranja, nagomilane tačke su grupisane u podklastere, dok su rasute tačke otklonjene kao ekstremne vrednosti. Algoritam dalje koristi hijerarhijsko klasterovanje bazirano na predstavnicima kako bi primenio globalno klasterovanje, inkrementalno eliminisao ekstremne vrednosti i rafinisao klastere. BIRCH može da radi sa bilo kojom dodeljenom količinom memorije i predstavlja jedan od reprezentativnih algoritama za klasterovanje velikih skupova podataka.

2.3 Ostali algoritmi za klasterovanje ekspresija gena

Pored dobro poznatog K-means algoritma i sličnih algoritama iz ove klase, kod klasterovanja podataka o ekspresiji gena, koriste se i mnoge druge metodologije. (Thalamuthu et al., 2006) su prikazali jasnu evaluaciju šest različitih algoritama klasterovanja primenjenih na podatke o ekspresiji gena. (Nascimento et al., 2009) su koristili GRASP (*eng. Greedy Randomized Adaptive Search Procedure*) algoritam za pronalaženje particija u podacima o ekspresiji gena. Takođe su ispitivali uticaj različitih mera odstojanja na performanse GRASP algoritma. (Iam-on et al. 2010) su razvili takozvani ansambl (*eng. ensamble*) metod baziran na linkovima (*eng. link-based*) za primenu u ovoj oblasti. (Ayadi et al., 2012) su predložili BicFinder algoritam baziran na metodologiji bi-klasterovanja. (Moise et al., 2009) su prikazali sistematsku evaluaciju pod-prostornih (*eng. sub-space*) i projektovanih (*eng. projected*) tehnika klasterovanja, pri velikom broju različitih eksperimentalnih postavki, sa fokusom na podatke sa velikim brojem atributa (*eng. high dimensional data*). (Wu et al., 2002) su prikazali važnost primene višestrukih klaster algoritama za otkrivanje relevantnih bioloških paterna. (Monti et al., 2003) su predložili okruženje (*eng. framework*) za klastering podataka o ekspresiji gena koje koristi sheme ponovnog uzorkovanja (*eng. re-sampling*) i pojedinačne klastering algoritme za kreiranje višestrukih modela klasterovanja. Oni su pokazali da konsenzus algoritmi daju stabilnija rešenja i kvalitetnije particije nego pojedinačni algoritmi. Ovo istraživanje je prošireno u (Yu et al., 2007) koji su predložili unapređeni konsenzus klastering algoritam baziran na grafovima, (Giancarlo and Utro,

2011) su unapredili ovaj metod kako bi ubrzali proces konsenzus klasteringa. (Pirim et al., 2011) su koristili pristup kosenzus klasteringa za dobijanje konsenzus particija tako što su spajali različite particije dobijene iz pojedinačnih algoritama. I ovaj pristup je testiran na podacima o ekspresiji gena i pokazao je bolje rezultate od individualnih algoritama koji su korišćeni za dobijanje konsenzus particija. Još jedan od zanimljivih pristupa je ispitivanje alternativnih modela klasterovanja. Kod klasterovanja podataka o ekspresiji gena, grupisanje gena bazirano na njihovim funkcijama ili strukturi može biti podjednako korisno. Zbog toga je cilj alternativnog klasterovanja da generiše različite klaster modele, tako da se rezultati mogu analizirati iz različitih perspektiva i na osnovu njih ispitivati nove hipoteze (Dang and Bailey, 2010). Ko-klasterovanje je takođe korišćeno u analizi ekspresije gena (Shaham et al., 2011; Yan et al., 2011; Bonchi et al., 2011). Ono omogućava simultano klasterovanje redova i kolona skupa podataka, čime se omogućava identifikacija pod-skupova redova koji imaju slične osobine na različitim pod-skupovima kolona i obrnuto. Algoritmi bazirani na raspodeli definišu klastere kao slučajeve koji su najpribližniji određenom modelu distribucije. Za razliku od teorije, gde je ovaj tip modela dobro utemeljen, u praksi se ovi modeli nisu pokazali dobro. Razlog tome je preveliko prilagođavanje modela podacima (eng *over-fitting*), odnosno, model se nekad fokusira na greške i šumove, a ne na većinu podataka, čime se gubi suština. Ovaj problem se rešava postavljanjem ograničenja na kompleksnost samog modela tj. raspodele kojom se opisuje model. Najrasprostranjeniji model iz ove grupe je maksimizovanje očekivanja (expectation-mazimization) algoritam. Za ovaj algoritam se pre njegovog pokretanja određuje fiksni broj gausovih funckija raspodele (da bi se izbeglo preveliko prilagođavanje podacima), koje se inicijalizuju sa slučajnim parametrima i onda se u toku iteracija se otpimizuju. Takođe je moguće utvrditi i povezanost i uticaj između pojedinih atributa. Kao problem se nameće to, što korisnik mora da zna da odabere odgovarajući model, a takođe treba imati u vidu da za pojedine skupove podataka ne postoji odgovarajući statistički model koji bi mogao da opiše te podatke.

Algoritmi bazirani na gustini, opisuju klastere kao delove prostora podataka gde je veća gustina slučajeva nego u drugim delovima skupa podataka. Klasteri se formiraju na osnovu povezivanja tačaka koje poseduju odgovarajuću gustinu, ali samo onih koje se

ne smatraju za šum tj. onih slučajeva koji ispunjavaju kriterijum gustine (eng. *density criterium*). Slučajevi koji se nalaze u prostoru između klastera se smatraju za šum. Ova klasa algoritama rešava jedan od najvećih problema koji ima K-means: identifikaciju klastera proizvoljnih oblika (ne- sferičnih).

DBScan (Sander et al., 1998) i OPTICS (Ankerst et al., 1999) su među najpopularnijima u ovoj klasi. Ipak algoritmi zasnovani na gustini nisu pogodni za analizu podataka velikih dimenzija, jer kako dimenzionalnost problema raste, tako rastu i relativna odstojanja između slučajeva i time se otežava analiza gustine (Andreopoulos et al., 2009). Pored toga, DBScan identifikuje klaster šuma (*eng. noise cluster*) dok algoritmi bazirani na centroidima to ne rade i zbog toga je teško porediti rezultate (Raczynski et al., 2010).

Složenost DBScan-a, tipičnog predstavnika ove grupe algoritama je niska i to predstavlja veliku prednost pri klasterovanju visokodimenzionih podataka. Osnovni nedostatak ovih algoritama je taj što oni granice klastera definišu na osnovu pada gustine, što znači da kod skupova podataka koji imaju Gausovu raspodelu, klasteri će izgledati proizvoljno (ne-sferično) jer se gustina konstantno smanjuje.

Sa druge strane Fuzzy C-means (Tang et al., 2010; Bazdek, 1981) je uspešno primenjen na podatke o ekspresiji gena (Dembélé and Kastner, 2010) i ima kompatibilnu strukturu sa generičkim algoritmima prikazanim u ovom radu i zbog toga je i ovaj algoritam uključen u evaluaciju.

2.4 Evaluacija algoritama klasterovanja

Evaluacija modela klasterovanja se bavi procenom koliko su kvalitetni rezultati klasterovanja (Maulik, 2002). Evaluacija je jedan od najvećih problema i jedan od osnovnih uslova za uspešnu primenu klasterovanja (Jain, 1988). Ovo proističe iz činjenice da je klasterovanje po prirodi nenadgledan proces, odnosno da algoritmi koji kreiraju modele klasterovanja nisu vođeni "istinitim" klasterima, već grupišu podatke na osnovu neke mere sličnosti (geometrijske osobine). Najveći problem je u tome što

korisnici obično ne znaju unapred koja mera sličnosti adekvatno oslikava strukturu konkretnih podataka.

Sa druge strane, postoje situacije kada se algoritmi klasterovanja koriste kao zamena za algoritme klasifikacije (nadgledane algoritme), kao u slučaju podataka o ekspresiji gena. Ovo se radi iz razloga što ovi podaci imaju problem velikog disbalansa između broja atributa i broja slučajeva, gde broj atributa neretko prelazi broj od 10000 dok je broj slučajeva obično manji od 200. U ovakvim situacijama algoritmi za klasifikaciju obično ne uspevaju da kreiraju validne modele i zbog toga se primenjuju algoritmi klasterovanja. U situacije kada se algoritmi klasterovanja koriste kao zamena za klasifikaciju, "prave" strukture (klasteri) su unapred poznate i evaluacija se vrši poređenjem poklapanja između "pravih" klastera i klastera dobijenih uz pomoć algoritama. Shodno tome da li je poznata prava struktura klasterovanja, mere evaluacije se mogu podeliti na:

- Interne mere i
- Eksterne mere.

Interne mere evaluiraju klastere na osnovu nekih poželjnih geometrijskih osobina (npr. blizina objekata unutar klastera i udaljenost objekata između klastera) i ne zahtevaju postojanje izlaznog atributa. Ove mere se često koriste nakon procesa klasterovanja, ali često i tokom procesa pri čemu algoritam zapravo optimizuje zadatu meru. Problem je u tome što korisnici obično ne znaju šta je odgovarajuća mera za njihov konkretan problem (skup podataka).

Sa druge strane, eksterne mere obezbeđuju objektivnu informaciju o tome koliko je dobar klaster model u odnosu na prave klase (tzv. zlatni standard). Problem kod ovih mera je što prave klase moraju biti unapred poznate. Iako je klastering po prirodi nenadgledani proces (klase nisu poznate unapred, već se kreiraju uz pomoć algoritama), ukoliko su poznate klase iz prethodnih sličnih problema, moguće je identifikovati dobre algoritme i koristiti ih kod budućih problema.

Što se tiče eksternih mera evaluacije, postignut je konsenzus o tome koje je najbolje koristiti (Vinh, 2010), dok kod internih mera evaluacije to nije moguće. Zbog toga će u narednom tekstu biti opisan veći broj internih mera evaluacije kao i one eksterne mere koje su identifikovane kao pogodne za ocenu modela klasterovanja (Vinh et al., 2010). U sekciji 4 će biti i predložena metodologija za rešavanje ovih problema u oblasti klasterovanja ekspresija gena.

2.4.1 Interne mere evaluacije

Interne mere evaluacije se baziraju na ova dva kriterijuma (Tan et al., 2005):

- kompaktnost (eng. *compactness*) i
- razdvojenost (eng *separation*).

Kompaktnost nam pokazuje koliko su slučajevi u samom klasteru slični (eng. *related*) Određena grupa mera kompaktnost računa preko varijanse. Niža varijansa znači veću kompaktnost. Pored varijanse određeni broj algoritama kompaktnost utvrđuje na osnovu razdaljine, maksimalna prosečna razdaljina dva slučaja u klasteru ili maksimalna/prosečna razdaljina slučaja od centroida klastera.

Razdvojenost nam pokazuje koliko su klasteri različiti (dobro razdvojeni). Razdvojenost između klastera se određuje na osnovu razdaljine između centroida ili najmanje razdaljine između dva slučaja u različitim klasterima. Takođe se u pojedinim slučajevima koristi mera gustine. U daljem tekstu će biti opisane mere evaluacije.

Kompaktnost (*Compactness* ili *Overall Deviation*) klastera OD (Handle et al., 2006) se meri ukupnom devijacijom i ona predstavlja zbir suma udaljenosti svih slučaja svakog klastera od svojih centroida. Niska vrednost znači da su klasteri dobro formirani tj. da su svi slučajevi klastera relativno blizu svom centru. Formula izgleda ovako:

$$OD(C) = \sum_{j=1}^{C} \sum_{i=1}^{m_j} d(x_i^j, \mu_j)$$

Gde je

C - skup svih klastera

μ_j - centroid klastera C_j

x_i^j - i-ti slučaj u j-tom klasteru

d - funckija rastojanja x_i^j slučaja od centroida μ_j

Mera sveukupne devijacije je našla svoju upotrebu u velikom broju algoritama i dobro se pokazala kod sfernih klastera ali nedostatke je ispoljila u klasterima nepravilnog oblika. Niska vrednost OD nam govori da su klasteri dobro formirani tj. da su slučajevi blizu centroida.

Povezanost (*Connectivity*) CO (Handle et al., 2007) pokazuje da li su okolni slučajevi u istom klasteru. Formula izgleda ovako:

$$Conn(C) = \sum_{i=1}^{N} \sum_{j=1}^{L} f(x_i, x_j)$$

$$f\left(x_i, x_j\right) = \begin{cases} \frac{1}{j}, & \text{ako } x_i \text{ i } x_j \text{ nisu u istom klasteru} \\ 0, & \text{ako jesu u istom klasteru} \end{cases}$$

Gde je:

N - broj svih slučajeva u skupu podataka

L - parametar za koliko najbližih suseda proveravamo pripadnost klasteru

x_i - slučaj za koji trenutno računamo povezanost

x_j - j -ti najbliži slučaj slučaju x_i

Niska vrednost CO nam govori da su klasteri dobro struktuirani. Ova mera se dobro pokazala kod proizvoljno oblikovanih klastera, a loše se pokazala kod klastera između kojih je malo rastojanje.

Xie-Beni index (Xie et al., 1991), XB indeks, predstavlja količnik između ukupne devijacije OD i rastojanja između centara klastera $sep(C)$. XB indeks se računa na sledeći način:

$$XB = \frac{OD(C)}{sep(C)}.$$

Formulu za OD je opisana ranije:

$$OD(C) = \sum_{j=1}^{C} \sum_{i=1}^{m_j} d(x_i^j, \mu_j)$$

a formula za rastojanje klastera je :

$$sep(C) = \min_{i \neq j} \left\{ \left\| \mu_i - \mu_j \right\| \right\}$$

gde su:

C - skup svih klastera

μ_i - centroid klastera C_i

μ_j - centroid klastera C_j

Separacija klastera označava minimalnu vrednost razlike između dva centra klastera. XB indeks se pokazao kao dobra mera za hiper-sferične klastere. Niža vrednost XB indeksa znači da su klasteri dobro oformljeni.

Globalni siluet indeks GS (*global silhouette*) (Rousseeuw, 1987) se izračunava iz siluetnih vrednosti (*silhouette values*) svih slučajeva u skupu podataka na sledeći način:

$$S = \frac{1}{C} \sum_{j=1}^{C} \left[\frac{1}{m_j} \sum_{i=1}^{m_j} \frac{b(x_i^j) - a(x_i^j)}{\max(b(x_i^j), a(x_i^j))} \right]$$

m_j - broj slučajeva u klasteru C_j

x_i^j - i-ti slučaj u j-tom klasteru

C - ukupan broj klastera

Gde su: $a(x_i^j) = \frac{1}{m_j - 1} \sum_{k=1}^{m_j} d(x_i^j, x_k^j)$

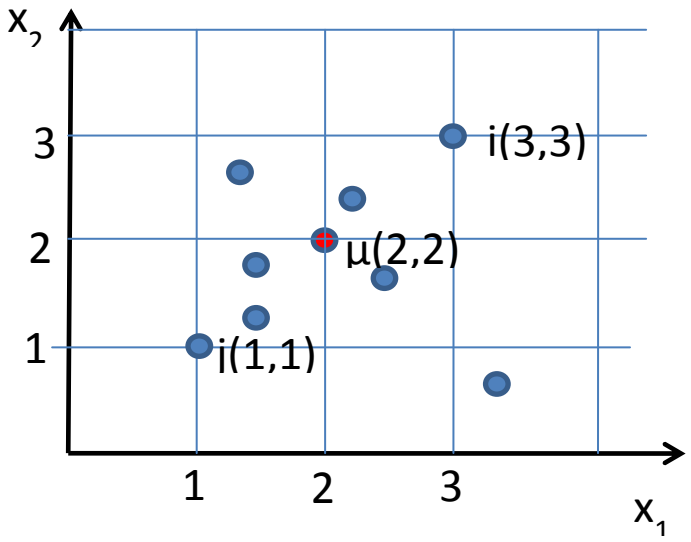
$$b(x_i^j) = \min_{j, j \neq i} \left[\frac{1}{m_j} \sum_{k=1}^{m_j} d(x_i^j, x_k^j) \right]$$

GS može da poredi kvalitet klasterovanja za različit broj klastera, uzima vrednost od [-1, 1], gde viša vrednost označava da je klasterovanje bolje obavljeno.

Index simetrije (*Symmetry index*) *Sym* (Saha et al., 2007) se bazirana na simetričnosti objekata u određenom geometrijskom prostoru. Tačka simetrije (*Point symmetry distance*) je definisana na sledeći način: za N slučajeva $j = 1...N$ i za referentni vektor (centroid) μ, tačka simterije za tačku x_k i vektor μ je

$$d_s(x_j, \mu_i) = \min \frac{\| d(x_j^i - \mu_i) + d(x_k^i - \mu_i) \|}{\| d(x_j^i - \mu_i) \| + \| d(x_k^i - \mu_i) \|}$$

Cilj je naći tačku simetrije u Euklidskom prostoru: $\| d(x_j^i - \mu_i) \|$ i $| d(x_k^i - \mu_i) \|$. Ova funkcija je minimizovana kada je ispunjen uslov simetrije $2*\mu - i$ za K najbližih čvorova. $2*\mu - i$. Slika 2.8 pokazuje slučaj kada je navedeni uslov ispunjen gde je čvor *i*(3,3) i njemu simetričan čvor *j*(1,1) u odnosu na centar $\mu(2,2)$.



Slika 2.8. Simterični slučajevi u odnosu na centar

Iznos simetrije za klaster I je suma simetrije tačaka za sve njegove slučajeve

$$E_i = \sum_{j=1}^{m_i} d_s(x_j^i, \mu_i)$$

Iznos simetrije za ceo skup podataka je zbir simetrija klastera:

$$S_k = \sum_{i=1}^{C} E_i$$

Sym-index se koristi za merenje kvaliteta klasterovanja i računa se preko formule:

$$Sym(C) = \frac{1}{C} \times \frac{1}{S_k} \times D_k$$

C - broj klastera

D_k - maksimalno Euklidsko rastojanje između dva centra klastera u svim mogucim kombinacijama

Sym-index bi trebalo da bude maksimalan da bi se odredio tačan broj klastera

$$D_K = \max_{i,j=1}^{K} \left\| \bar{c}_i - \bar{c}_j \right\|$$

Dejvis-Boldin indeks (*Davies-Bouldin index*) *DB* (Davies et al., 1979) istovremeno posmatra sličnost unutar klastera i razlike među klasterima:

$$DB = \frac{1}{C} \sum_{j=1}^{C} \max_{p, p \neq j} \left[\frac{S_j - S_p}{d(\mu_j, \mu_p)} \right]$$

μ_j - predstavnik j -tog klastera

C - broj klastera

S_j -mera rasejanosti j -tog klastera

d - funckija koja računa rastojanje dva centroida

Formula za rasejanost klastera izgleda ovako:

$$S_j = \frac{1}{m_j} \sum_{i=1}^{mj} d(x_i^j, \mu_j)$$

m_j - broj slučajeva u i -tom klasteru

x_i^j - i -ti slučaj u j -tom klasteru

μ_j - centroid j -tog klastera

Postoji više različitih varijacija *DB* koje nastaju usled različitih metrika (npr. Euklidska, kosinusna, Čebišljeva itd.) za merenje bliskosti čvorova. Niža vrednost *DB* govori da postoji veća bliskost slučajeva u klasteru i veća razdvojenost samih klastera čime je klasterovanje kvalitetnije.

Danov indeks (*Dunn index*) *DI* (Dunn, 1974), određuje kvalitet klastera na osnovu prečnika klastera. Ima više različitih načina na koje je moguće utvrditi prečnik klastera. Jedan od načina je preko maksimalne razdaljine slučajeva unutar klastera:

$$\Delta_i = \max_{x,y \in C_i} d(x, y)$$

Δ_i -prečnik i -tog klastera

x - slučaj koji pripada i -tom klasteru

y - slučaj koji pripada i -tom klasteru

ili preko srednje vrednosti razdaljine svih parova

$$\Delta_i = \frac{1}{|m_i||m_i - 1|}\sum_{j=1}^{m_i}\sum_{p=1}^{m_i} d(x_j^i, x_p^i)$$

x_i^j - j -ti slučaj koji pripada i -tom klasteru

x_p^j - p -ti slučaj koji pripada i -tom klasteru

m_i -broj slučajeva u i -tom klasteru

Ili preko razdaljine svih tačaka od centroida klastera

$$\Delta_i = \frac{\sum_{j=1}^{m_i} d(x_j^i, \mu_i)}{|m_i|}, \mu_i = \frac{\sum_{j=1}^{m_i} x_j^i}{|m_i|}$$

C_i -broj slučaja u i -tom klasteru

x_j^i - j -ti slučaj koji pripada i-tom klasteru

μ_i -centroid i -tog klastera

Slično kao i razdaljine unutar klastera, računaju se i razdaljine između klastera $d(C_i, C_j)$, ovo je razdaljina između klastera C_i i C_j.

Za C klastera Danov index DI bi izgledao ovako:

$$DI = \min_{1 \le i \le C}\left\{\min_{1 \le j \le C, j \ne i}\left\{\frac{d(C_i, C_j)}{\max_{1 \le k \le C}\Delta_k}\right\}\right\}$$

DI zavisi od broja klastera i od odabrane metrike, problem DI je taj što je on dobar koliko je i njegov najlošiji klaster dobar, čime dolazimo do zaključka da ako je samo jedan klaster loš i DI rezultat će biti loš. Niža vrednost Danovog indeksa označava kvalitetnije klasterovanje.

Akaike informcioni kriterijum (*Akaike information criterion*) AIC (Akaike, 1974*)* je mera za izbor odgovarajućeg statističkog modela za odgovarajući skup podataka. AIC se zasniva na konceptu entropije informacija i nudi meru gubitka informacija kada se određeni model koristi.

$$AIC(C) = 2p_j - 2\sum_{j=1}^{K} \xi_j$$

C -broj klastera

ξ_j -logaritamska vrednost podataka za j -ti klaster

p_j -broj parametara u modelu sa C klastera

$$p_c = K[2P^{kon} + \sum_{k=1}^{P^{kat}} (L_{kat} - 1)]$$

P^{kat} -broj kategoričkih promenljivih

P^{kont} -broj kontinualnih promenljivih

L_{kat} -broj različitih kategorija u okviru k -te kategoričke promenljive

$$\xi_j = -m_j \left[\sum_{k=1}^{p^{kont}} \frac{1}{2} \log(\sigma_k{}^2 + \sigma_{jk}{}^2) + \sum_{k=1}^{p^{kat}} E_{jk} \right]$$

m_j -broj slučajeva u j -tom klasteru

$\sigma_k{}^2$ -varijansa k -te kontinualne varijable na celom skupu

$\sigma_{jk}{}^2$ -varijansa k -te kontinualne varijable u j -tom klasteru

E_{jk} -deo logaritamske verodostojnosti za j-ti klaster koji se odnosi na k -tu kategoričku promenljivu

$$E_{jk} = -\sum_{i=1}^{L_{kat}} \frac{m_{jki}}{m_j} \log \frac{m_{jki}}{m_j}$$

m_{jki} -broj slučajeva u j -tom klasteru koji za k -tu varijablu imaju i-tu vrednost

m_j -broj slučajeva u j -tom klasteru

Niža vrednost AIC mere označava kvalitetnije klasterovanje.

Bajesov informacioni kriterijum (*Bayesian information criterion*) BIC, poznat i kao Švarcov kriterijum (*schwarz criterion*) (Schwarz, 1978) SBC ili $SBIC$ je kriterijum za izbor odgovarajućeg modela u grupi sa konačnim brojem modela. Bazira se na funkciji sličnosti (likelihood function) i usko je povezan AIC. BIC ima sledeći oblik :

$$BIC(C) = -2\sum_{j=1}^{K} \xi_j + p_j \log(N)$$

C -broj klastera

ξ_j -logaritamska vrednost podataka za j -ti klaster

p_j -broj parametara u modelu sa J klastera

N -ukupan broj slučajeva u svim klasterima

$$p_j = K[2P^{kon} + \sum_{k=1}^{P^{kat}} (L_{kat} - 1)]$$

P^{kat} -broj kategoričkih promenljivih

P^{kont} -broj kontinualnih promenljivih

L_{kat} -broj različitih kategorija u okviru k-te kategoričke promenljive

$$\xi_j = -m_j \left[\sum_{k=1}^{p^{kont}} \frac{1}{2} \log({\sigma_k}^2 + {\sigma_{jk}}^2) + \sum_{k=1}^{p^{kat}} E_{jk} \right]$$

m_j -broj slučajeva u j -tom klasteru

${\sigma_k}^2$ -varijansa k -te kontinualne varijable na celom skupu

${\sigma_{jk}}^2$ -varijansa k -te kontinualne varijable u j -tom klasteru

E_{jk} -deo logaritamske verodostojnosti za j -ti klaster koji se odnosi na k tu kategoričku promenljivu

$$E_{jk} = -\sum_{i=1}^{L_{kat}} \frac{m_{jki}}{m_j} \log \frac{m_{jki}}{m_j}$$

m_{jki} -broj slučajeva u j -tom klasteru koji za k -tu varijablu imaju i -tu vrednost

m_j -broj slučajeva u j -tom klasteru

Homogenost (*homogenity*) (Shamir et al., 2001) predstavlja prosečnu razdaljinu između svakog slučaja u klasteru i predstavnika klastera:

$$H = \frac{1}{N}\sum_{j=1}^{C}\sum_{i=1}^{m_j} d(x_i^j, \mu_j)$$

Gde su:

N -broj slučajeva u celom skupu podataka

C -broj klastera

m_j -broj slučajeva u j -tom klasteru

d -funkcija razdaljine

x_i^j - i -ti slučaj u j -tom klasteru

μ_j -centroid j -tog klastera

Razdvojenost (*Separation*) (Shamir et al., 2001) predstavlja ponderisanu funkciju homogenosti i formula izgleda ovako:

$$S = \frac{\sum_{i=1}^{C}\sum_{j=1}^{C} m_i m_j d(\mu_i \mu_j)}{\sum_{i=1}^{C}\sum_{j=1}^{C} m_i m_j}$$

Gde su:

C -broj klastera

m_j -broj slučajeva u j-tom klasteru

d -funkcija razdaljine

x_i^j - i -ti slučaj u j-tom klasteru

μ_j -centroid j-tog klastera

Smanjivanje vrednosti funkcije H i povećavanje vrednosti funkcije S ukazuje na poboljšavanje kvaliteta klastera. Treba naglasiti da ove dve funkcije nisu linearno zavisne tj. da je moguće da se jedna funckija ponaša drugačije od druge. Ove dve mere se često koriste u paru tj. od njih se pravi količnik, čime se dobija bolji uvid u kvalitet obavljenog klasterovanja.

Calinski-Harabasz index CH (Calinski et al., 1974) se računa na ovaj način:

$$CH = \frac{(N-C)\sum_{j=1}^{C} m_j \| \mu_j - \mu \|}{(C-1)\sum_{j=1}^{C}\sum_{i=1}^{m_j} \| x_i^j - \mu_j \|}$$

N -broj slučajeva u celom skupu podataka

C -broj klastera

m_j -broj slučajeva u klasteru j-tom klasteru

μ_j -centroid j-tog klastera

μ -centroid celog skupa podataka

x_i^j - i-ti slučaj u j-tom klasteru

Veća vrednost ove funckije ukazuje da je klasterovanje bolje obavljeno.

Koren srednje kvadratne devijacije (*Root-mean-square standard deviation*) $RMSSTD$ (Sharma, 1996) predstavlja kvadratni koren varijanse svih elemenata u skupu podataka. Ova mera se jedino bavi procenom koliko je sam klaster homogen, ne uzima u obzir koliko je ovaj klaster različit od drugih. Formula izgleda ovako:

$$RMSSTD = \left[\frac{\sum_{j=1}^{C}\sum_{i=1}^{m_j} \| x_i^j - \mu_j \|}{P\sum_{j=1}^{C} m_j} \right]^2$$

Gde su:

C -broj klastera

m_j -broj slučajeva u klasteru j -tom klasteru

x_i^j - i -ti slučaj u j -tom klasteru

μ_j -centroid j -tog klastera

P - broj atributa koji opisuju svaki slučaj

R-kvadrat (*R-squared*) RS (Sharma, 1996) predstavlja količnik razlike kvadrata rastojanja svih slučajeva u skupu podataka i centroida celog skupa podataka i kvadrata rastojanja svih slučajeva u svakom klasteru i centroida svakog klastera i kvadrata rastojanja svih slučajeva u skupu podataka i centroida celog skupa podataka. Formula izgleda ovako:

$$RS = \frac{\sum_{i=1}^{N} \| x_i - \mu \|^2 - \sum_{j=1}^{C} \sum_{i=1}^{m_j} \| x_i^j - \mu_j \|^2}{\sum_{i=1}^{N} \| x_i - \mu \|^2}$$

Gde su:

N -broj slučajeva u celom skupu podataka

μ -centroid celog skupa podataka

x_i - i -ti slučaj u skupu podataka

C - broj klastera

m_j - broj slučajeva u klasteru j -tom klasteru

x_i^j - i -ti slučaj u j -tom klasteru

μ_j - centroid j-tog klastera

m_j - broj slučajeva u j -tom klateru

RS predstavlja meru međusobne različitosti klastera i ne uzima u obzir međusobnu sličnost unutar klastera.

Modifikovana Hubertova G statistika (*Modified Hubert G statistics*) (Hubert, 1985) procenjuje različitost između klastera računajući različitost slučajeva u klasterima tj. njihovu razdaljinu (veća vrednost ove statistike označava kvalitetnije klasterovanje). Formula izgleda ovako:

$$\Gamma = \frac{2}{N(N-1)} \sum_{i=1}^{C} \sum_{j=1}^{C} \sum_{p=1}^{m_i} \sum_{q=1}^{m_j} d(x_p^i, x_q^j) d(\mu_i, \mu_j)$$

Gde su:

N -broj slučajeva u celom skupu podataka

C -broj klastera

m_j -broj slučajeva u klasteru j -tom klasteru

x_p^i - p -ti slučaj u i -tom skupu podataka

μ_i - centroid i -tog klastera

d - funckija koja računa razdaljinu između dva slučaja

I Indeks I (Maulik, 2002) se računa na osnovu razdvojenosti klastera tj. na osnovu maksimalne razdaljine između predstavnika dva klastera i na osnovu kompaktnosti koja se računa na osnovu udaljenosti svih slučajeva i predstavnika celog skupa podataka i razdvojenosti od centara klastera. Formula je:

$$I = \left(\frac{1}{C} \frac{\sum_{i=1}^{N} d(x_i, \mu)}{\sum_{j=1}^{C} \sum_{i=1}^{m_j} d(x_i^j, \mu_j)} MAX \right)^P$$

Gde su:

N -broj slučajeva u celom skupu podataka

C -broj klastera

μ -centroid celog skupa podataka

m_j -broj slčajeva u klasteru j -tom klasteru

x_i^j - i -ti slučaj u j -tom klasteru

d - funckija koja računa distancu između dva slučaja

Pri čemu je:

$$MAX = \max_{i,j} d(\mu_i, \mu_j)$$

Visoka vrednost I nam govori da je klasterovanje dobro obavljeno.

SD indeks SD (Halkidi, 2000) se bazira na prosečnoj rasejanosti slučajeva iz skupa podataka i ukupnoj razdvojenosti klastera. Rasejanost se računa na osnovu varijanse

slučajeva unutar klastera, dok se razdvojenost klastera, računa na osnovu razdaljine između centara klastera. Niža vrednost SD indeksa nam govori da je klasterovanje dobro obavljeno. Formula izgleda ovako:

$$SD(C) = a * Scat(C) + DIS(C)$$

gde je:

a - težinski faktor

$$Scat(C) = \frac{1}{C}\sum_{i=1}^{C}\frac{\sigma(C_i)}{\sigma(D)}$$

gde je:

$\sigma(C_i)$ -varijansa i-tog klastera

$\sigma(D)$ -varijansa celog skupa podataka

$$DIS(C) = \frac{\max_{i,j}(d(\mu_i,\mu_j))}{\min_{i,j,i\neq j}(d(\mu_i,\mu_j))}\sum_{i=1}^{C}\left(\sum_{j=1}^{C}d(\mu_i,\mu_j)\right)^{-1}$$

C -broj klastera

μ_i -centroid i -tog klastera

max- funckija koja nalazi maksimalnu vrednost od svih slučajeva

min- funckija koja nalazi minimalnu vrednost od svih slučajeva

d -funkcija koja računa razdaljinu između dva slučaja

SDbw indeks *S_Dbw* (Halkidi, 2001) dodaje gustinu kao faktor kod računanja razdvojenosti klastera. Ideja je da za svaki par centroida klastera važi da bar kod jednog gustina bude veća od gustine srednje tačke (između centroida). Kompaktnost unutar klastera se računa na isti način kao kod SD. Maksimalna vrednost S_Dbw indeksa znači da je klasterovanje optimalno urađeno.

Formula za računanje ovog indeksa je sledeća:

$$S\_Dbw = Scat(C) + Dens\_bw(C)$$

gde je:

$$Scat(C) = \frac{1}{C}\sum_{i=1}^{C}\frac{\sigma(C_i)}{\sigma(D)}$$

i gde su:

$\sigma(C_i)$ -varijansa i-tog klastera

$\sigma(D)$ -varijansa celog skupa podataka

$$Dens\_bw(C) = \frac{1}{C(C-1)}\sum_{i=1}^{C}\left(\sum_{j=1, i\neq j}^{C}\frac{\sum_{p=1}^{m_i+m_j} d(x_p, u_{i,j})}{\max\left(\sum_{p=1}^{m_i} d(x_p.\mu_p), \sum_{p=1}^{m_j} d(x_p.\mu_p)\right)}\right)$$

C -broj klastera

m_j -broj slučajeva u klasteru j-tom klasteru

m_i -broj slučajeva u klasteru i-tom klasteru

μ_p -centroid p-tog klastera

x_p - p-ti slučaj u klasteru

2.4.2 Eksterne mere evaluacije

Kao što je rečeno u Sekciji 2.4 eksterne mere evaluacije ocenjuju stepen preklapanja klastera dobijenih algoritmom u odnosu na "pravi" klaster model (objekti u koji su u istom klasteru u jednom modelu treba da budu u istom klasteru i u drugom modelu). Eksterni indeksi za evaluaciju klastera u skorije vreme dobijaju sve veći interes od strane naučne zajednice (Balachandran and Khemani, 2011). Pokazano je da pored osobina metrike, eksterni indeksi treba takođe da budu normalizovani i prilagođeni slučajnosti (eng. *adjusted for chance*). Ovo je bitno zato što neprilagođeni indeksi daju bolje rezultate sa porastom broja klastera (čak i kada je broj kreiranih klastera veći od pravog broja klastera). U skorašnjem istraživanju, (Vinh, 2010) je sproveo iscrpno poređenje među brojnim merama evaluacije i adjusted mutual information (AMI) indeks

je preporučen kao generalno dobra mera za evaluaciju i poređenje klastera kao i za dizajn algoritama. Pokazano je da AMI (Vinh, 2010) zadovoljava osobine metrike i normalizacije i da identifikuje tačan broj klastera bolje nego ostale mere (ovo je dokazano na brojnim simuliranim i realnim skupovima podataka). Štaviše, AMI je pokazao vrhunske performanse na 8 skupova podataka o ekspresiji gena koji su korišćeni za eksperimentalnu evalaciju i u ovom istraživanju. Zbog svega navedenog, za evaluaciju kvaliteta modela klasterovanja, u istraživanju predstavljenom u ovoj disertaciji, korišćen je AMI indeks. Pored AMI indeksa koristi se i ARI (Milligan and Cooper, 1987) indeks za komparaciju komponentnih algoritama sa drugim algoritmima iz literature (pošto u trenutku istraživanja nije bilo radova koji su koristili AMI indeks za evaluaciju klaster algoritama na podacima o ekspresiji gena).

Oba indeksa porede dva klaster modela. Ukoliko su rezultujuće particije identične, vrednost oba indeksa je 1, a ukoliko se particije potpuno razlikuju, vrednost oba indeksa je 0. Ovde će biti opisana oba indeksa, korišćenjem notacije iz (Vinh, 2010).

Ukoliko S predstavlja skup u kome se sadrži N objekata (instanci, slučajeva), onda klasterovanje U na S predstavlja način particionisanja objekata u ne-preklapajuće pod-skupove $\{U_1,U_2,...,U_R\}$. Informacija o preklapanju između dva klasterovanja $U=\{U_1,U_2,...,U_R\}$ i $V=\{V_1,V_2,...,V_R\}$ može se opisati uz pomoć tabele kontigencije (Tabela 2.2).

Tabela 2.2. Matrica kontigencije za računanje ARI indeksa

U/V	V_1	V_2	…	V_C	Sums
U1	n_{11}	n_{12}	…	n_{1C}	a_1
U2	n_{21}	n_{22}	…	n_{2C}	a_1
⋮	⋮	⋮		⋮	⋮
UR	n_{R1}	n_{R2}	…	n_{RC}	a_R
Sums	b_1	b_2	…	b_C	

ARI predstavlja meru koja je bazirana na prebrojavanju parova objekata po kojima se dva klasterovanja slažu ili ne slažu. Parovi objekata u S mogu biti klasifikovani u jjednu od 4 grupe - N_{11}: broj parova koji se nalaze u istom klasteru i u U i u V; N_{00}: broj parova

koji se nalaze u različitim klasterima i u U i u V; N_{01}: Broj parova koji se nalaze u istom klasteru u U, ali se nalaze u različitim klasterima u V; N_{10}: broj parova koji se nalaze u različitim klasterima u U, a u istim klasterima u V. Kada je matrica kontigencije definisana, ARI se računa kao:

$$ARI(U,V) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00}+N_{01})(N_{01}+N_{11}) + (N_{00}+N_{10})(N_{10}+N_{11})}$$

AMI je informaciono-teoretska mera (Vinh, 2010). Ukoliko su data dva klasterovanja, njihove entropije, njihova zajednička entropija, uslovne entropije i mera međusobne informacije (*mutual information*) su date kao marginalne i zajedničke distribucije objekata u U i V respektivno, kao:

$$H(U) = -\sum_{i=1}^{R} \frac{a_i}{N} \log \frac{a_i}{N}$$

$$H(U,V) = -\sum_{i=1}^{R}\sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N}$$

$$H(V,U) = -\sum_{i=1}^{R}\sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N}$$

$$I(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{n_{ij}}{N} \log \frac{n_{ij}/N}{a_i b_j / N^2}$$

Konačno, AMI se računa kao:

$$AMI = \frac{I(U,V) - E\{I(U,V)\}}{\max\{H(U),H(V)\} - E\{I(U,V)\}\}}$$

3 Komponentni pristup u razvoju algoritama klasterovanja

Komponentni pristup ili pristup belih kutija (Delibašić et al., 2009; Delibašić et al., 2012, Suknović et al., 2012, Vukićević et al., 2012a, Vukićević et al., 2012b, Jovanović et al., 2014), u izgradnji OZP algoritama je nastao kada je identifikovano da neke familije algoritama (npr. stabla odlučivanja ili algoritmi klasterovanja zasnovani na predstavnicima), imaju jako sličnu strukturu i da se algoritmi koji se razvijaju na principu "crnih kutija" uglavnom unapređuju inkrementalno (Sonnenburg et al., 2007). To znači da se postojeći algoritmi unapređuju parcijalno (npr. K-means++ algoritam se razlikuje od originalnog K-means samo u načinu inicijalizacije, K-medoids u načinu preračunavanja predstavnika itd.). Algoritmi implementirani kao "crne kutije" olakšavaju rad analitičarima tako što sakrivaju detalje funkcionisanja algoritma i kontrolisani su samo pomoću ulaznih parametara. Sa druge strane, ovakav način razvoja algoritama stvara niz drugih problema:

- da bi se razvilo parcijalno unapređenje, neophodno je implementirati kompletan algoritam. Ovo često vodi do različitih implementacija istog algoritma (ili dela algoritma) i samim tim nekorektne komparacije performansi sa originalnim ili drugim algoritmima;
- nemogućnost kombinovanja prednosti postojećih algoritama;
- algoritmi često ostaju "zarobljeni" u test okruženjima i razvoj algoritama u literauri ne prati njihova implementacija u softverima, a samim tim ni šira primena.

Kao potencijalno rešenje ovih problema predložen je razvoj i primena algoritama na principu "belih kutija" (komponentni pristup). Kod ovog pristupa, određena klasa ili podskup klase algoritama (npr. stabla odlučivanja ili algoritmi klasterovanja zasnovani na predstavnicima) se raščlanjavaju na pod-probleme koji su zajednički svim algoritmima te klase (npr. pod-problem inicijalizacije se može rešiti slučajnim izborom početnih predstavnika klastera - kao kod K-means ili odabirom najudaljenih tačaka kao kod K-means++). U tom slučaju korisniku je omogućeno da bira željena rešenja pod-problema (komponentu), bez obzira na funkcionisanje ostatka algoritma i time da sam kreira različite algoritme (kombinacije komponenata). Intuitivno je jasno da se ovakvim

pristupom otvara mogućnost izgradnje velikog broja "hibridnih" algoritama i da se uvećava šansa za pronalaženje najboljeg algoritma za konkretne podatke. Ovakva mogućnost je posebno bitna, s obzirom na raznolikost korisničkih zahteva i skupova podataka i da ne postoji nijedan algoritam koji je dominanto bolji od ostalih. Jaku podršku ovoj tvrdnji daju tzv. „No Free Lunch" teoreme (Wolpert, 1996), gde autori dokazuju da je uvek moguće kreirati najbolji algoritam za određeni problem.

Takođe se naglašava da bi takav pristup ubrzao razvoj i primenu novih algoritama jer bi omogućio (Sonnenburg et al., 2007):

- kombinovanje prednosti različitih algoritama,
- reprodukciju naučnih rezultata,
- detaljnije poređenje algoritama,
- rad sa postojećim resursima uz manje ponavljanje implementacija,
- brže usvajanje u drugim oblastima i privredi i
- kolaborativno stvaranje standarda.

Da bi se ovakav pristup realizovao neophodno je da postoji:

- generička struktura (pod-problemi sa standardizovanim ulazima i izlazima),
- repozitorijum komponenti (rešenja pod-problema),
- generički algoritam, koji upravlja izvršenjem komponenti kada je definisana struktura (konkretan algoritam).

3.1 Pod-problemi i ponovo upotrebljive komponente

Tokom celokupnog procesa klasterovanja počevši od pred-procesiranja, pa sve do evaluacije klaster modela, korisnik mora da donese nekoliko odluka (Milligan and Cooper, 1987). Neke od njih su: odluka o izboru mere sličnosti/odstojanja ili način inicijalizacije centroida. Prateći ideje iz (Delibašić et al., 2009), zajedničke fleksibilnosti algoritama (pod-problemi) predstavljaju tačke u algoritmu gde je neophodno odlučiti o izboru rešenja (komponente). Generalno, za svaki pod-problem postoji više rešenja.

Dakle u predloženom pristupu je identifikovano pet pod-problema, koji su sumarno prikazani u tabeli 3.1.

Tabela 3.1. Ulazno-izlazna struktura pod-problema (Delibašić et al., 2012)

Pod-problem	*Ulaz*	*Izlaz*
Inicijalizacija predstavnika (IP)	Skup podataka	Predstavnici
Mera odstojanja (MO)	Par objekata	Odstojanje
Ažuriranje (Kreiranje) predstavnika (AP)	Skup podataka, predstavnici	Predstavnici
Evaluacija klastera (EK)	Klaster model (skup podataka, propadnost), predstavnici	Evaluacija
Stop kriterijum (SK)	Stanje algoritma	Da/Ne

Inicijalizacija predstavnika (IP) koristi skup podataka kao ulaz i vraća skup predstavnika klastera na izlazu. Ovaj pod-problem omogućava i korišćenje metoda inicijalizacije koji su ekstrahovane iz algoritama koji nisu originalno bazirani na predstavnicima (npr. DIANA, PCA itd.). Originalni Diana algoritam vraća pripadnost objekata klasteru i zbog toga se ova inicijalizacija ne može koristiti u izvornom obliku, već se moraju preračunati predstavnici. Da bi generički algoritam ostao konzistentan, koristi izabranu komponentu za ažuriranje/kreiranje predstavnika kako bi izračunao početne predstavnike uz pomoć komponenti ekstrahovanih iz algoritama koji originalno nisu bili bazirani na predstavnicima. (Milligan, 1980) je predložio ovakvu hibridizaciju algoritama sa Ward-ovom hijerarhijskom inicijalizacijom (Ward, 1963), a u (Delibašić et al., 2012) je ovaj pristup proširen sa sledećim komponentama za inicijalizaciju klaster modela: DIANA, GMEANS, PCA, and XMEANS.

Mera odstojanja (MO) na ulazu uzima dva objekta i vraća odstojanje (ili sličnost) između njih. Komponente iz ovog pod-problema su korišćene i kod drugih pod-problema (npr. inicijalizacija).

Ažuriranje (kreiranje) predstavnika (AP) na ulazu prima predstavnike klastera i klaster model (objekte i pripadnosti klasterima), a kao izlaz, vraća ažurirane (ili nove u slučaju inicijalizacije). U slučaju da kod pod-problema inicijalizacije izabrana komponenta nije odredila predstavnike (npr. DIANA ili PCA) već samo odredila pripadnosti objekata klasterima, funkcija *create()* se poziva da odredi predstavnike.

Tabela 3.2. Repozitorijum komponenti za dizajn generičkih algoritama (Delibašić et al., 2012)

Pod-problem	**Rešenja (komponente) i abrevijacije**	**Reference**
Inocijalizacija predstavnika		
	na slučajan načim (RANDOM)*	(Lloyd, 1982)
	kao u SPSS implementeaciji k-means algoritma (SPSS)	(Hartigan, 1975)
	Korišćenjem hijerarhijsko divizionog algoritma (DIANA)	(Kaufman and Rousseeuw, 1990)
	korišćenjem hijerarhijskog klasterovanja sa binarnom divizijom (XMEANS)	(Pelleg and Moore, 2000)
	korišćenjem hijerarhijskog klasterovanja sa binarnom divizijom baziranom na hijerarhijskom klasterovanju i sopstvenim vektorima podataka (GMEANS)	(Hammerly and Elkan, 2003)
	korišćenjem hijerarhijskog klasterovanja sa binarnom divizijom baziranom na glavnim komponentama podataka(PCA)	(Ding and He, 2004)
	Sa probabilističkom distribucijom baziranom na odstojanju (KMEANS++)*	(Arthur and Vassilvitskii, 2007)
Mera odstojanja		
	sa Euklidskim odstojanjm (EUCLID)*	npr. (Xu&Wunsch, 2010)
	sa Čebišljevim odstojanjem (CHEBY)	npr. (Xu&Wunsch, 2010)
	sa "city block" odstojanjem (CITY)	npr. (Xu&Wunsch, 2010)
	Sa korelacijom (CORREL)	npr. (Xu&Wunsch, 2010)
	sa kosinusnim odstojanjem (COSINE)*	npr. (Xu&Wunsch, 2010)
Ažuriranje predstavnika		
	sa aritmetičkom sredinom (MEAN)*	(Hartigan and Wong, 1979)
	sa medijanom (MEDIAN)	(Kaufman and Rousseeuw, 1990)
	sa "online" procesiranjem sa stopom učenja zavisnom od iteracija (KOHONEN)*	(Kohonen, 2001)
	sa "online" procesiranjem sa stopom učenja zavisnom od objekata koji pripadaju klasteru (BOTTOU)	(Bottou and Bengio, 1995)
	sa "online" procesiranjem sa penalizovanjem rivala (KSTAR)	(Cheung, 2003)
Evaluacija klastera		
	sa siluet indeksom (SILHOU)	(Rousseeuw, 1987)
	sa kompaktnošću (COMPACT)*	(Kaufman and Rousseeuw, 1990)
	sa XB indeksom (XB)	(Xie and Beny, 1991)
	sa povezanošću (CONN)	(Ester et al., 1996)
	sa min-max presekom (MMC)	(Ding et al., 2001)
	sa BIC kriterijumom (BIC)	(Schwarz G, 1978)
	sa AIC kriterijumom (AIC)	(Akaike, 1974)
Stop kriterijum		
	Stabilnost predstavnika (REPSTAB)	(Hartigan and Wong, 1979)
	Broj iteracija (NITER)*	(Bottou and Bengio, 1995)
	Granica kvaliteta klastera (CETHR)	(Pelleg and Moore, 2000)
	Vreme (TIME)	(Kohonen, 2001)
	Stabilnost pripadnosti (MEMBERS)*	(Cheung, 2003)

Evaluacija klastera (EK) na ulazu prima klaster model, a vraća ocenu kvaliteta klastera. Komponente ovog pod-problema zapravo predstavljaju implementacije internih mera za evaluaciju klastera. U predloženom generičkom algoritmu klasterovanja (Slika 3.1), pod-problem evaluacije klastera, se koristi da bi se donela odluka o izboru najboljeg modela od onih koji su generisani u različitim re-startovanjima i iteracijama algoritma. Na ovaj način, se odabira najbolji model prema željenoj meri evaluacije. U sekciji 3.3.2 će biti objašnjeno da selektovane komponente za ovaj pod-problem mogu da se koriste i kao komponente u drugim pod-problemima. Npr. kod inicijalizacije, sve hijerarhijske komponente koriste evaluaciju klastera (DIANA, PCA, XMEANS i GMEANS koriste ove komponente kako bi odredile da li će se klasteri deliti). Deljenje komponenata između pod-problema dodatno olakšava i ubrzava impleentaciju novih algoritama i smanjuje potrebu za re-implementacijom.

Stop kriterijum prati trenutno stanje algoritma (trenutna iteracija, stabilnost predstavnika, vreme izvršenja itd.) i vraća signal za zaustavljanje izvršenja generičkog algoritma**.**

Ideja iz (Delibašić et al., 2009) je u (Delibašić et al., 2012) proširena tako što su identifikovane i implementirane komponente za sve pod-probleme iz tabele 3.2.

Pored rešenja prikazanih u tabeli 3.2 postoji još puno rešenja u literaturi za svaki pod-problem. Na primer za pod-problem "Inicijalizacija predstavnika", (Astrahan, 1970) je predložio traženje particija sa dobro razdvojenim objektima koji imaju puno suseda. (Bradley and Fayyad, 1998) su koristili "bootstrapping" tehniku kako bi pronašli stabilne particije. (Faber, 1994) je definisao metod za izbor predstavnika klastera iz regiona sa velikom gustinom. (Steinley, 2003) pripisuje objekte klasterima na osnovu jednakih verovatnoća, (Mirkin, 2005) primenjuje Max-Min proceduru, a (Belal and Daoud, 2005) predlažu korišćenje medijana atributa sa najvećom varijansom, (Maitra, 2009) generiše veliki broj lokalnih pod-klastera i selektuju predstavnike iz najudaljenijih klastera itd. (Tajunisha and Saravanan, 2010) su predložili model za poboljšanje originalnog K-means algoritma sa fleksibilnošću izbora različitih tehnika inicijalizacije i različitih mera odstojanja. (Lühr and Lazarescu, 2009) opisuju

inkrementalno klasterovanje dinamičkih tokova podataka koristeći mere sličnosti bazirane na povezanosti objekata (eng. connectivity based). Pristup baziran na komponentama za automatsku selekciju atributa predložili su (Mierswa and Morik, 2005). Sistematsku evaluaciju metoda inicijalizacije za K-means algoritam je predstavljena u (Peterson et al., 2010). (Xiong et al., 2009) su sproveli detaljnu analizu mera za evaluaciju klastera u odnosu na raspodelu podataka. Takođe, postoji nekoliko studija koje testiraju različite tipove K-means algoritma koji se razlikuju u jednom ili više delova (npr. Altun et al., 2006). Prateći proces klasterovanja predložen od strane (Milligan and Cooper, 1987), (Walesiak and Dudek, 2007), su razvili "clusterSim" paket u programskom jeziku "R", da bi utvrdili optimalnu proceduru klasterovanja za konkretne podatke, tako što su variralli sve kombinacije mera odstojanja, normalizacije podataka i različitih algoritama. (Achtert et al., 2008) su predložili ELKI okruženje koje ima fleksibilnosti rada sa različitim tipovima podataka i koristi nekoliko mera sličnosti/odstojanja za svaki od tipova podataka. Ipak sve pomenute fleksibilnosti su dizajnirane i evaluirane samo na nekoliko pod-problema (npr. inicijalizacija predstavnika ili mera odstojanja/sličnosti), dok su ostali delovi algoritma bili fiksirani. Okruženje predloženo u ovom radu, omogućava jednostavnu integraciju pomenutih poboljšanja, kroz arhitekturu koja će biti predstavljena u Sekciji 3.3.

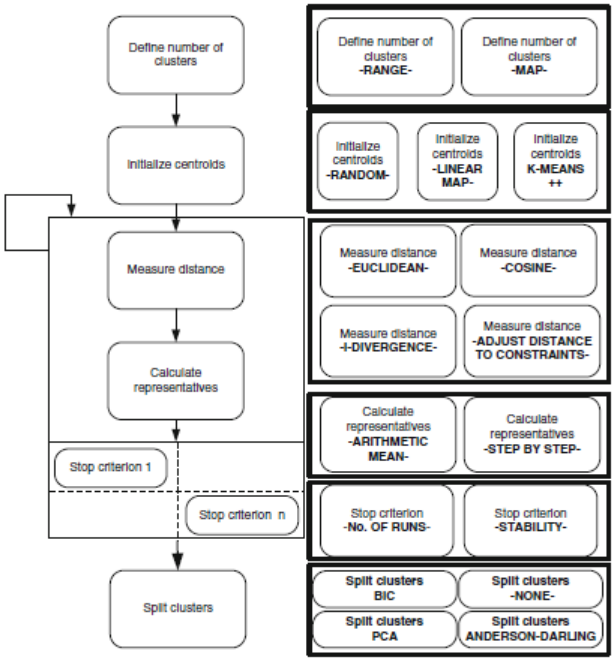
3.2 Generički algoritam za klasterovanje

U pristupu razvoja algoritama klasterovanja zasnovanom na komponenatama, tok algoritma se može definisati generičkim klastering algoritmom (Slika 3.1). Generički algoritmi predloženi u (Delibašić et al., 2009; Delibašić et al., 2012) koriste kao osnovu particioni K-means algoritam, pri čemu omogućavaju izbor različitih poboljšanja (komponenata) u odnosu na originalni algoritam. Pored toga u ove algoritme su integrisane ideje hijerarhijskog klasterovanja (divizionog) čime se rešava problem određivanja broja klastera od strane korisnika. Pseudo kod originalnog K-means algoritma je prikazan u algoritmu 1.

Algoritam 1. Originalni K-means algoritam
Input:
Dataset
Output:
Cluster model (Dataset with memberships) and cluster representatives
Parameters:
minK: initial number of clusters
maxK: desired final number of clusters through hierarchical division
refinePartitions (local/global): refinement method in Hierarchical division
GCAlgorithm(Dataset, K)

```
1.      // Initialization
2.      Randomly "Initialize representatives" to initialize K representatives
3.      // Refinement
4.      repeat
5.              for each instance from dataset
6.                      for each representative
7.                              "Measure distance" (instance, representative)
8.                      end
9       .               Assign instance to nearest representative by calculating mean
12.             end
13.     Until "Stop criterion"
```

U istraživanju (Delibašić et al., 2009) je razmatrana algoritam koji definiše raspon željenog broja klastera koji se definiše uz pomoć donje i gornje granice broja klastera ("Define number of clusters - RANGE" na slici 1). Određivanje broja klastera uz pomoć samo-organizujućih mapa (Kohonen, 2001; "Define number of clusters - MAP" na slici 3.1) je van okvira ovog istraživanja.



Slika 3.1. Generički algoritam klasterovanja, pod-problemi i komponente (Delibašić et al., 2009)

Generički algoritam definiše dva parametra: donju i gornju granicu broja klastera (minK i maxK). Prvi korak (pod-problem) je inicijalizacija predstavnika. Koristeći komponente ovog pod-problema kreiraju se početni centroidi klastera. Nakon toga, prelazi se na iterativni postupak, u kome se prvo mere udaljenosti/sličnosti objekata sa svakim predstavnikom, a zatim se svaki objekat dodeljuje najbližem/najsličnijem predstavniku (klasteru). Konačno, preračunava se novi predstavnik novoformiranog klastera. Ovaj postupak se ponavlja dok se ne ispuni neki od kriterijuma zaustavljanja (npr. broj iteracija, vreme itd.). Nakon završetka ovog iterativnog postupka, dobijen je minimalni broj klastera (minK). Klasteri se zatim evaluiraju nekom od internih mera evaluacije (pod-problem „Evaluacija klastera"). Nakon toga algoritam binarno deli postojeće klastere i ponovo ih evaluira. Ukoliko je novi klaster model bolji po internoj meri evaluacije, novi klasteri se zadržavaju, u suprotnom ne. Ovaj postupak se ponavlja dok se ne dostigne maksimalni broj klastera ili dok nema poboljšanja po meri evaluacije.

U (Delibašić et al., 2012), opisani algoritam je formalizovan i sastoji se iz 3 faze (pseudo kod je prikazan u algoritmima 1 i 2):

- Inicijalizacija (*Initialization*) gde je inicijalnih (minK) predstavnika generisano
- Rafinisanje (*Refinement*), gde se objekti iterativno dodeljuju predstavnicima i predstavnici se preračunavaju i
- Hijerarhijska divizija (*Hierarchical division*), gde se klasteri binarno dele, dok se željeni broj klastera (maxK) ne kreira ili se algoritam ne zaustavi.

Generički klaster algoritam ima tri globalna parametra:

- minK: broj klastera koji se generiše u fazi inicijalizacije;
- maxK: maksimalni broj klastera koji se može kreirati u procesu hijerarhijske divizije i
- refinePartitions: definiše kako će klasteri biti redefinisani nakon hijerarhijske divizije (posle binarnog deljenja klastera). Postoje dve mogućnosti:
    - local: redefinisati klastere "decu" bazirane na podacima iz "roditeljskog" klastera i
    - global: redefinisati sve klastere, ovoga puta sa dva nova predstavnika (kao u Hammerly and Elkan, 2003).

Algoritam 2. Generički algoritam baziran na predstavnicima (Delibašić et al., 2012)

Input:

Dataset

Output:

Cluster model (Dataset with memberships) and cluster representatives

Parameters:

minK: initial number of clusters

maxK: desired final number of clusters through hierarchical division

refinePartitions (local/global): refinement method in Hierarchical division

```
GCAlgorithm(Dataset, minK, maxK, refinePartitions)
// Initialization
Use "Initialize representatives" to initialize min_K representatives
// Refinement
repeat
    for each randomly (without replacement) sampled instance from dataset
        for each representative
            "Measure distance" (instance, representative)
        end
    . Assign instance to nearest representative
        If "Update representatives".isOnline() then
            "Update representatives".update()
    end
    If not "Update representatives".isOnline() then
        "Update representatives".update()
until "Stop criterion"
// Hierarchical division
If maxK > minK
    Split each cluster binary using "Initialize representatives" (minK=2)
    repeat
        Do "Evaluate clusters" on child clusters and parent clusters
        Choose best difference between child and parent clusters evaluation
        If difference is positive
            If refinePartitions is local
                Do Refinement (Parent cluster dataset, child centroids)
            If refinePartitions is global
                Do Refinement (Whole dataset, all centroids)
    until number of clusters = maxK or no splitting in last loop
```

Predloženi algoritam koristi "refinement" faze iz "online" i "batch" algoritama koji su bazirani na K-means-u. Ovo je postignuto tako što se na slučajan način uzorkuje skup podataka (što je obavezno kod "online" algoritama) a nema uticaja na "batch" algoritme. Takođe je važno definisati za svaku komponentu za ažuriranje predstavnika da li će biti korišćen "online" ili "batch" metod. Ovo se postiže tako što se stavlja obeležje na svaku komponentu (binarni atribut koji označava način ažuriranja predstavnika) i na taj način generički algoritam "zna" kako da je koristi.

Hijerarhijska diviziona faza se često koristi za određivanje "pravog" broja klastera (npr. X-means (Pelleg and Moore, 2000) koristi ovu strategiju). Ona se koristi kada korisnik nije siguran koliko tačno treba njegov model da definiše klastera ali može da da okvirni raspon (minimalni i maksimalni broj klastera). U (Delibašić et al., 2012), transformisano je i prilagođeno generičkom algoritmu nekoliko komponenti koje su u originalnim algoritmima mogle da inicijalizuju samo dva klastera. Transformisane su tako da mogu da inicijalizuju K klastera (u zavisnosti od minK parametra koji je korisnički definisan), tako što primenjaju hijerarhijsku, binarnu divizionu proceduru. Ovakva inicijalizacija je korišćena je kod DIANA, PCA, XMEANS i GMEANS komponenti i predstavljena je u Algoritmu 3.

Algoritam 3. Inicijalizacija predstavnika uz pomoć hijerarhijsko-divizionog algoritma (Delibašić et al., 2012).

Input:
    Dataset
Output:
    Representatives
Parameter:
    K: number of clusters
Hierarchical division (Dataset, K)
1. Binary split (Dataset)
2. **repeat**
3. **for** each cluster
4. parentEC = "Evaluate cluster" (current cluster)
5. childCluster =Binary split (current cluster)
6. children EC = "Evaluate cluster" (childCluster)
7. **end**
8. Choose best improvement between parent and child clusters evaluation
9. Replace parent cluster with child clusters
10. **until** K clusters

3.2.1 Kompleksnost generičkog algoritma

Kako bi se odredila skalabilnost generičkog algoritma, kompleksnost je određena na nivou pod-problema i komponenti, pošto kompleksnost zavisi od selektovanih komponenti za izvršenje algoritma. Kompleksnost većine komponenata je opisana u radovima koji su referencirani u tabeli 3.2. Particioni deo algoritma (redovi 1-15 algoritma 2) sekvencijalno izvršava faze inicijalizacije i rafinisanja. Inicijalizacija se izvršava samo jednom, dok se rafinisanje (dodeljivanje objekata predstavnicima i

preračunavanje predstavnika) izvršava dok se ne ispuni neki od kriterijuma zaustavljanja. Posle ove faze minimalni broj klastera (minK) je određen. Hijerarhijsko-divizioni deo (redovi 16-27 u algoritmu 2) ponavlja prethodno opisanu proceduru, na svakom klasteru (sa fiksiranim K=2) dok se ne kreira maksimalni broj klastera (maxK) ili dok više nema poboljšanja u modelu (mereno odabranom internom merom evaluacije). U svakom koraku, divizioni deo algoritma deli svaki klaster na dve particije. Klasteri "deca" koji najviše poboljšavaju ukupni kvalitet modela se zadržavaju i rafinišu (Posle svekog divizionisanja, broj klastera raste za 1). Prema tome, ukupno vreme izvršavanja algoritma je:

$$time\ (Init + Refine) + \frac{maxK - minK}{2} * (maxK + minK) * time(init) + (maxK - minK) * time(refine)$$

Važno je primetiti da kompleksnost generičkog algoritma na nivou pod-problema ne prelazi kompleksnost X-means ili G-means algoritama, koji imaju slične divizione strategije. U zavisnosti od broja objekata, kompleksnost inicijalizacije i rafinisanja je linearna ili kvadratna, kada se koriste komponente koje su prethodno opisane. Zbog toga, ukupna kompleksnost generičkog algoritma nikada ne prelazi kvadratnu kompleksnost i često ima linearnu kompleksnost. Sa druge strane, postoji mnogo napora da se obezbedi skalabilnost postojećih algoritama (npr. Kumar et al., 2010; Ene et al., 2011), pa se i ova poboljšanja mogu integrisati u generičko okruženje za klasterovanje.

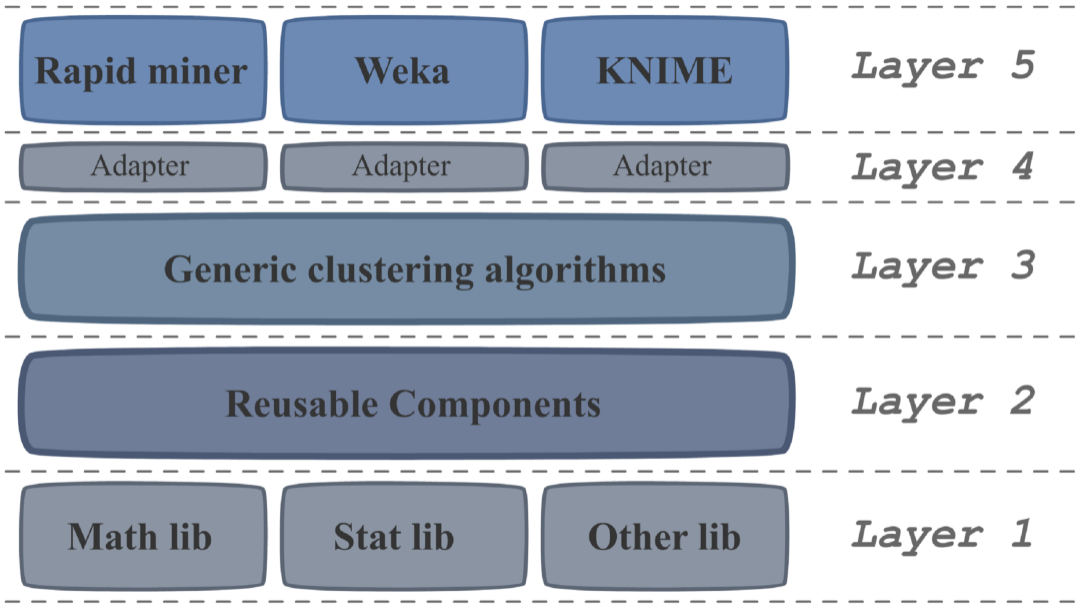
3.3 Softverska arhitektura za kolaborativni dizajn algoritama

Kako bi se obezbedio kolaborativni razvoj algoritama (kao što je predloženo u Sonnenburg et al., 2007), kao i jednostavno korišćenje algoritama baziranih na komponentama, neophodno je da arhitektura bude lako proširiva sa novim komponentama, pod-problemima i algoritmima. Takva arhitektura je već predložena za algoritme stabala odlučivanja (Vukićević et al., 2012c) i implementirana je u okviru WhiBo okruženja (ekstenzija RapidMiner-a). Više o WhiBo ekstenziji se može pronaći na web stranici projekta: www.whibo.fon.bg.ac.rs. Integracija ove arhitekture u WhiBo okruženje će biti predmet budućeg rada. Ova integracija treba da omogući jednostavan

dizajn algoritma uz pomoć GUI-ja i pontencijalno može integrisati napore RapidMiner i WhiBo korisnika u razvoju algoritama.

3.3.1 Generička arhitektura bazirana na komponentama

Predloženo okruženje za integraciju generičkih algoritama klasterovanja (u RapidMiner i druga okruženja) je prikazano na 3.2 uz pomoć slojevitog dijagrama. Slojevi su nezavisni i formirani su na različitim nivoima apstrakcije. Ovo omogućava jednostavno proširenje postojećeg okruženja.

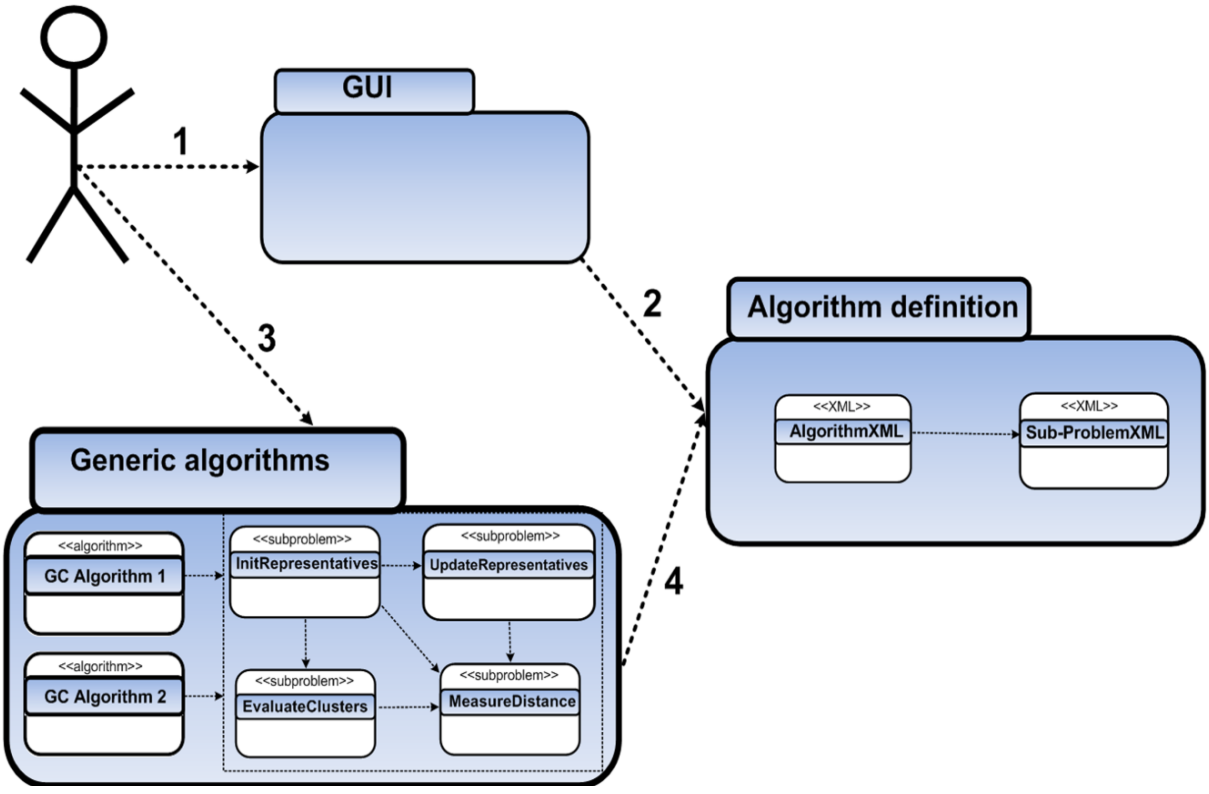


Slika 3.2. Dijagram slojeva generalne arhitekture za razvoj i primenu algoritama zasnovanih na komponentama (Vukićević et al., 2012c, Delibašić et al., 2012)

Sloj 1 predstavlja repozitorijum standardnih matematičkih, statističkih i drugih metoda i operacija (npr. operacije sa matricama, statistički testovi itd.). Ponovo upotrebljive komponente su sačuvane u Sloju 2.

Ponovo upotrebljive komponente mogu koristiti metode iz Sloja 1 i oblikovane su tako da odgovaraju U/I strukturi iz tabele 3.1. Sloj 3 čuva generičke algoritme koji komponente mogu sastavljati u konkretne algoritme. Generički algoritmi obezbeđuju modele koji se mogu koristiti u različitim softverskim rešenjima. Sloj 4 uključuje adaptere koji omogućavaju korišćenje generičkih algoritama u različitim softverima (Sloj 5). Slika 3.2 prikazuje strukturu softverskih paketa koja podržava nezavisnost

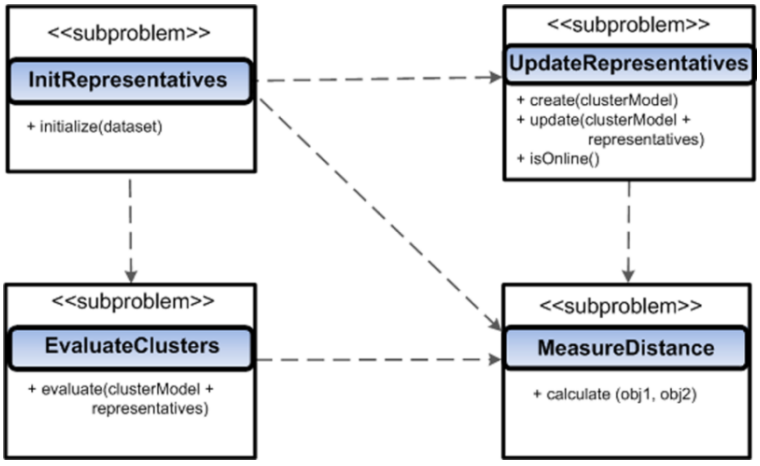
između repozitorijuma komponenti, generičkih algoritama i korisničkog interfejsa. "Generic algorithms" paket sadrži repozitorijum komponenti i generičke algoritme koji su konstruisani da bi mogli da prave specifične algoritme klasterovanja. "Algorithm definition" paket treba da omogući kompletno opisivanje specifičnih algoritama, uključujući izabrane komponente i njihove parametre. Korisnik definiše algoritam uz pomoć grafičkog interfejsa, tako što dodeljuje komponentu/komponente za svaki pod-problem. Nakon što je algoritam definisan čuva se kao perzistentan objekat (npr. XML datoteka). Ovakva arhitektura obezbeđuje separaciju algoritamske logike (programskog koda koji implementira generički algoirtam) i grafičkog interfejsa (slika 3.3). Ova separacija omogućava jednostavno dodavanje novih pod-problema i komponenti (kao i njihovu evaluaciju i komparaciju) bez potrebe za promenom grafičkog interfejsa.



Slika 3.3. Nezavisnost grafičkog interfejsa od domenske logike (Delibašić et al., 2012)

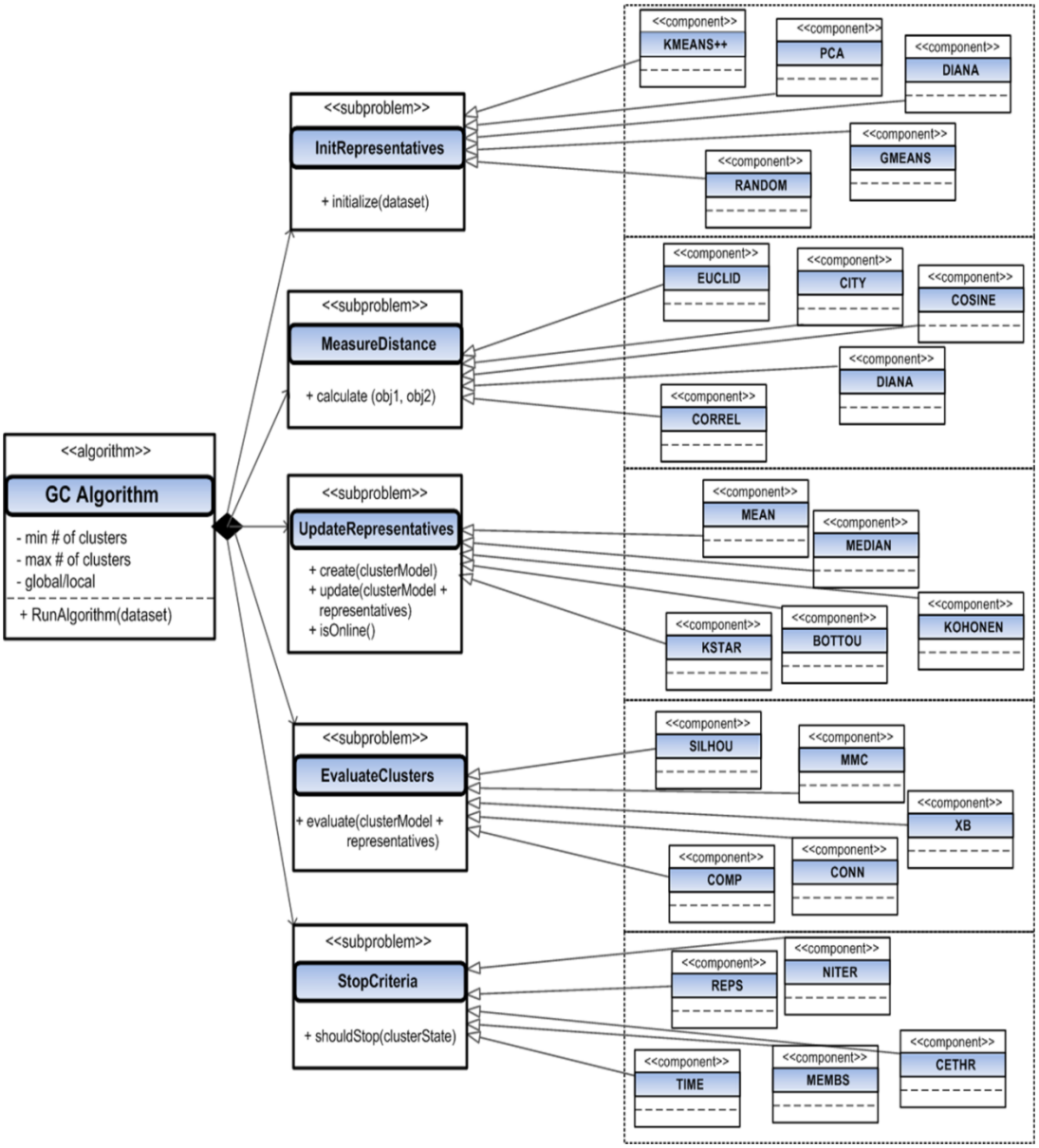
3.4 Struktura generičkog algoritma klasterovanja

Kao što je pomenuto u prethodnoj sekciji, komponente iz različitih pod-problema mogu da koriste jedna drugu. Interakcija između pod-problema, omogućava konzistentno korišćenje komponenata kroz ceo algoritam (npr. ako je komponenta COSINE izabrana za merenje odstojanja, ona će takođe biti korišćena i pri inicijalizaciji, ažuriranju predstavnika i evaluaciji klastera). U analiziranom softveru otvorenog koda, često se može pronaći da su mere odstojanja nekonzistentno korišćene u okviru algoritma.



Slika 3.4. Interakcije između pod-problema (Delibašić et al., 2012)

Dijagram klasa generičkog algoritma klasterovanja Sloj 3 na slici 3.2) je prikazan na slici 3.4. Generički klaster algoritam uključuje kompoziciju interfejsa za pod-probleme. Svaki pod-problem (u objektno orijentisanom dizajnu se naziva interfejs) je implementiran sa različitim rešenjima (komponentama). Standardizovana U/I struktura (tebela 3.2) dozvoljava implementaciju i ponovno korišćenje komponenti bez potrebe za promenama u generičkom algoritmu.



Slika 3.5. Dijagram klasa generičkog algoritma klasterovanja (Delibašić et al., 2012)

Arhitektura i generički algoritmi prikazani u ovom poglavlju će biti korišćeni za dizajn i evaluaciju velikog broja algoritama klasterovanja nad podacima o ekspresijama gena. Nakon evaluacije, u sekcijama 5 i 6, biće predstavljene metode za automatsku identifikaciju najboljih algoritama za konkretne podatke.

4 Primena, rangiranje i selekcija komponentnih algoritama za klasterovanje ekspresija gena

U ovoj sekciji biće prikazana primena i evaluacija komponentnih algoritama klasterovanje podataka o ekspresiji gena. Generalni cilj istraživanja je razvoj originalne metodologije za projektovanje i evaluaciju algoritama za klasterovanje podataka o ekspresiji gena. Ovaj generalni cilj treba da bude ostvaren kroz ostvarenje dva posebna cilja:

- pokazati da su komponentni algoritmi bazirani na predstavnicima adekvatni za klasterovanje podataka o ekspresiji gena i
- predložiti metodologiju za dizajn i izbor i rangiranje komponentnih algoritama za klasterovanje ekspresija gena.

Ovi ciljevi će biti ostvareni kroz sledeći postupak:

- Evaluacija performansi komponentnih algoritama, dizajniranih generičkim partitivnim algoritmom (Algoritam 2) baziranom na predstavnicima (bez mogućnosti automatske identifikacije tačnog broja klastera).
- Evaluacija generičkog partitivno-hijerarhijskog algoritma (Algoritam 2 koji uključuje Algoritam 3), kao i njegova mogućnost identifikacije tačnog broja klastera.
- Ispitivanje postojanja korelacije između internih i eksternih mera evaluacije klastera (pošto interne mere mogu biti direktno optimizovane za vreme rada algoritma).
- Konačno, u Sekciji 5 predlog i evaluacija proširenog sistema meta-učenja za selekciju i rangiranje algoritama klasterovanja koji uključuje komponentne algoritme kao i interne mere evaluacije.

Evaluacijom navedenog postupka, proveriće se ispravnost sledećih hipoteza:

Opšta hipoteza: Moguće je projektovati algoritme klasterovanja ekspresije gena, koji će biti bolji od postojećih rešenja i moguće je identifikovati najbolji algoritam za konkretan skup podataka.

Posebna hipoteza 1: Razmenom ideja i delova postojećih algoritama moguće je projektovati nove algoritme koji će davati kvalitetnije modele od originalnih algoritama.

Posebna hipoteza 2: Razmenom ideja i delova postojećih algoritama baziranim na komponentama moguće je projektovati nove algoritme koji će davati kvalitetnije modele od drugih tipova algoritama klasterovanja.

Posebna hipoteza 3: Moguće je identifikovati podskupove (kombinacije) komponenata koje daju kvalitetnije rezultate klasterovanja u odnosu na druge podskupove komponenata.

Posebna hipoteza 4: Postoji korelacija između nekih internih i eksternih mera evaluacije algoritama za klasterovanje ekspresija gena.

Posebna hipoteza 5: Korišćenjem partitivno hijerarhijskog algoritma baziranog na komponentama moguće je automatski odrediti tačan broj klastera za konkretan skup podataka.

Posebna hipoteza 6: Ukrštanjem metapodataka ekspresije gena, metapodataka strukture algoritma i performansi algoritma moguće je donositi odluke za projektovanje algoritama klasterovanja ekspresije gena.

4.1 Primena partitivnog algoritma klasterovanja baziranog na komponentama

Osnovna ideja ovog istraživanja je bila da se pruži dokaz o tome da je razmenom komponenata između algoritama klasterovanja, moguće dobiti kvalitetnije algoritme za klasterovanje podataka o ekspresiji gena u odnosu na originalne algoritme kao i na druge klase algoritama.

4.1.1 Eksperimentalna postavka

Za potrebe ovog eksperimenta dizajnirano je 432 algoritma baziranih na komponentama opisanih u prethodnoj sekciji (6*4*3*6). Ovi algoritmi su dizajnirani kombinacijom komponenata iz četiri pod-problema:

- Inicijalizacija - RANDOM, SPSS, XMEANS, GMEANS, PCA, KMEANS++,
- Mera odstojanja - EUCLIDEAN, CITY, CORREL, COSINE,
- Ažuriranje predstavnika - MEAN, MEDIAN, ONLINE,
- Evaluacija klastera - COMPACT, XB, CONN, SILHOU, AIC, BIC.

Pošto skupovi podataka već poseduju informaciju o tačnom broju klasa, postavili smo parametar K (broj klastera) na tačan broj, kao što je to uobičajeno u literaturi (Ahmad and Dey, 2007; Forestier et al., 2010) za probleme otkrivanja klasa (eng. class retrieval). Dodatno, parametar K je menjan u intervalu [2, 2K], ali su mere evaluacije pokazale najbolje rezultate kada je postavljeno tačno K. Svi algoritmi su puštani po 10 puta i prikazani su prosečni rezultati. Pored ispitivanja glavne hipoteze: da li novi algoritmi dizajnirani uz pomoć komponenti mogu dovesti do boljih rezultata na konkretnom skupu podataka u odnosu na postojeće algoritme, takođe je ispitivan uticaj jedne ili kombinacije komponenata na svakom skupu podataka i prosečno na svim skupovima podataka. Za evaluaciju predloženog pristupa, meren je kvalitet klastera sa eksternim merama evaluacije: Adjusted mutual index (AMI) (Vinh, 2010) and Adjusted Rand Index (ARI) (Milligan and Cooper, 1987). Kao što je objašnjeno u sekciji 2.4.2, AMI indeks je korišćen zato što je nedavno identifikovan kao najadekvatnija mera za evaluaciju klaster modela, dok je ARI indeks korišćen kako bi rezultati ovog istraživanja mogli biti upoređeni sa rezultatima iz literature.

4.1.2 Podaci

Eksperimenti prikazani u ovom radu su sprovedeni na 5 simuliranih i 17 realnih skupova podataka o ekspresiji gena (Tabela 4.1).

Tabela 4.1. Skupovi podataka za evaluaciju algoritama

Referenca	*Skup podataka*	*Broj klasa*	*Broj atributa*	*Broj instanci*
(Monti et al., 2003; Yu et al., 2007)				
	Gaussian3	3	600	60
	Gaussian 5 delta2	5	2	500
	Gaussian 5 delta3	4	2	400
	Gaussian 4	6	600	60
	Simulated6	3	600	60
(Monti et al., 2003)				
	Novartis	4	1000	103
	Normal	13	1277	90
(Monti et al., 2003; Yu et al., 2007)				
	Leukemia	3	999	38
	Lung cancer	4+	1000	197
	StJude	6	985	248
	CNSTumors	5	1000	48
(Nascimento et al., 2010)				
	BreastA	3	1213	98
	BreastB	4	1213	49
	DBLCLA	3	661	141
	DBLCLB	3	661	180
	MultiA	4	5565	103
(Giancarlo et al., 2010)				
	CNSRat	6	17	112
	Leukemia (small)	3	100	38
	Lymphoma	3	100	80
	NCI60	8	200	57
	PBM	18	139	2329
	Yeast_Cell	5	72	698

Ovi skupovi podataka su preuzeti iz istraživanja (Giancarlo et al., 2010; Monti et al., 2003; Nascimento et al., 2010; Yu et al., 2007) jer su oni korišćeni za validaciju performansi klaster algoritama i zbog toga je bilo moguće poređenje rezultata. Skupovi podataka, kao i njihovi kratki opisi, su predstavljeni u tabeli 4.1. Zajednička karakteristika svih korišćenih skupova podataka je da imaju veliki broj atributa i jako mali broj slučajeva (instanci). Ova karakteristika je često isticana u literaturi kao glavni problem kod klasterovanja podataka o ekspresiji gena pošto su rezultati klasterovanja često osetljivi na šumove (eng. noise) i podložni su prevelikom prilagođavanju (*eng.*

over-fitting) (Monti et al., 2003). Originalni izvori i detaljniji opisi skupova podataka se mogu pronaći u referenciranim radovima.

4.1.3 Poređenje sa dobro poznatim algoritmima

U prvom eksperimentu kvalitet komponentnih algoritama klasterovanja je upoređen sa kvalitetom originalnih algoritama iz kojih komponente potiču. Originalni algoritmi su bili: K-means (RANDOM-EUCLIDEAN-MEAN-COMPACT), K-medians (RANDOM-EUCLIDEAN-MEDIAN-COMPACT), K-means++ (KMEANS++EUCLIDEAN-MEAN-COMPACT) i G-means (GMEANS-EUCLIDEAN-MEAN-COMPACT). Na ovaj način je kreirano okruženje za "fer" testiranje algoritama na istoj platformi gde su sve komponente imale istu implementaciju. Ovaj način testiranja je predložen u (Sonnenburg et al., 2007). Algoritmi bazirani na komponentama su evaluirani kako bi se proverilo da li algoritmi dizajnirani razmenom komponenata mogu dati bolje rezultate nego algoritmi iz kojih su komponente ekstrahovane. Poslednje dve kolone u Tabeli 4.2 prikazuju AMI vrednosti za najbolji i najlošiji algoritam, respektivno.

Tabela 4.2. Poređenje komponentnih i originalnih algoritama - AMI vrednosti (Delibašić et al., 2012)

Skup podataka	*K-means*	*K-medians*	*K-means++*	*G-means*	*FCM*	*Najbolji*	*Najgori*
Gaussian3	0,662	0,755	0,720	0.565	**1**	**1**	0,192
Gaussian4	0,867	0,773	0,869	0,864	0,857	**0,883**	0,400
Gaussian5 delta2	0,665	0,673	0,678	0,681	0,671	**0,700**	0,456
Gaussian5 delta3	0,926	0,916	0,924	**0,928**	0,922	**0,928**	0,523
Simulated6	0,743	0,787	0,779	0.215	**1**	**1**	0.148

Najbolji algoritmi na svakom skupu podataka su prikazani u Tabeli 4.3. Na skupovima "Gaussian 3" i "Simulated 6" FCM i dosta drugih algoritama je imalo savršen rezultat (svi su pronašli tačne particije), ali na svim ostalim skupovima različite kombinacije komponenti su dale algoritam sa najboljim performansama u smislu eksterne mere evaluacije. Ovi rezultati su dali podstrek za dalje istraživanje performansi algoritama baziranih na komponentama na realnim podacima.

Tabela 4.3. Najbolji komponentni algoritmi na veštačkim skupovima podataka (Vukićević et al., 2012b)

Skup podataka	***Algoritmi***
Gaussian3	166 algorithms
Gaussian4	GMEANS-EUCLIDEAN-MEAN-XB, PCA-EUCLIDEAN-MEAN-XB
Gaussian5 delta2	GMEANS-CITY-MEAN-CONN
Gaussian5 delta3	PCA-EUCLIDEAN-MEAN-COMPACT
Simulated6	26 algorithms

Na osnovu rezultata prikazanim u Tabeli 2 i Tabeli 3 može se zaključiti da su na svakom skupu podataka najbolje performanse dobijene kombinacijom komponenata koja je različita od kombinacije u originalnim algoritmima čime se potvrđuje naučna zasnovanost hipoteze:

- Razmenom ideja i delova postojećih algoritama moguće je projektovati nove algoritme koji će davati kvalitetnije modele od originalnih algoritama.

4.1.4 Poređenje sa rezultatima iz literature

Drugi eksperiment želi da pokaže da su komponentni algoritmi bazirani na centroidima, kompetitivni sa drugim algoritmima koji su u skorije vreme primenjeni na podatke o ekspresiji gena. Rezultati su upoređeni sa različitim tipovima klaster algoritama: hijerarhijskim, baziranim na konsenzusu i baziranim na meta heuristikama. Prvo su upoređeni rezultati sa K-means i hijerarhijskim algoritmima (prosečnog povezivanja - ALink i kompletnog povezivanja - CLink) koji su prikazani u (Giancarlo et al., 2010). U svakom algoritmu (osim FCM) varirano je 9 mera odstojanja. Tabela 4.4 pokazuje vrednosti ARI indeksa (maksimalne po svih 9 mera) za svaki algoritam, a poslednja kolona prikazuje vrednosti najboljeg algoritma baziranog na komponentama.

Tabela 4.4. Poređenje sa rezultatima iz (Giancarlo et al., 2010) (ARI values)

Skup podataka	*K-Means (9 distances)*	*ALink (9 distances)*	*CLink (9 distances)*	*FCM*	*Best RC based algorithm*
CNSRat	0,266	0,233	0,221	0,223	**0,279**
Leukemia (small)	0,919	0,919	0,919	0,910	**1**
Lymphoma	0,591	**0,678**	0,603	0,601	0,591
NCI60	0,442	0,498	0,451	**0,577**	**0,5138**
PBM	0,444	0,578	**0,589**	0,422	0,449
Yeast_Cell	0,496	**0,558**	0,424	0,424	0,541

Rezultati pokazuju da su algoritmi bazirani na komponentama imali najbolji rezultat na 2, ALink takođe na 2, a CLink i FCM na po jednom skupu podataka. Na "Lymphoma" skupu "ALink" je dao najbolji rezultat. Detaljnije ispitivanje rezultata iz (Giancarlo et al., 2010) je pokazalo da je ovaj rezultat dobijen korišćenjem Mahalanobis-ovog (d3) odstojanja. Ista mera odstojanja je dala najbolji rezultat sa CLink algoritmom na PBM skupu. Ova mera odstojanja nije korišćena u 432 algoritma koji su dizajnirani uz pomoć komponenata. Ovaj rezultat govori u prilog tome da dodavanje novih komponenata može biti korisno za dalje unapređivanje algoritama klasterovanja na podacima o ekspresiji gena.

Algoritmi bazirani na komponentama su takođe upoređeni sa GRASP algoritmom koji je testiran sa 4 različite mere odstojanja i čiji su rezultati prikazani u (Nascimento et al., 2010). Rezultati u Tabeli 4.5. pokazuju vrednosti ARI indeksa za GRASP algoritam sa svakom metrikom odstojanja, zatim za FCM i na kraju za najbolji algoritam baziran na komponentama.

Generički algoritam klasterovanja je uspeo da pronađe najbolji rešenje na 4 od 6 skupova podataka. Interesantno je primetiti da na "DBLCLA" skupu, najbolji rezulat GRASP algoritma (0.8) je postignut sa City Block odstojanjem, dok je kod generičkog algoritma postignuto najbolje particionisanje (0.96) sa SPSS-CORREL-ONLINE-AIC kombincijom komponenti.

Na "Novartis" skupu GRASP daje slične rezultate za sve metrike odstojanja (0.92) a KMEANS++-EUCLIDEAN-ONLINE-COMPACT algoritam je imao 0.96. Na DBLCLB skupu,

Tabela 4.5. Poređenje sa rezultatima iz (Nascimento et al., 2010) (ARI values)

Skup podataka	*GRASP (Euclidian)*	*GRASP (City Block)*	*GRASP (Cosine)*	*GRASP (Pearson)*	*FCM*	*Najbolji komponentni algoritam*
BreastA	0,682	0,682	0,686	0,692	0,628	**0,773**
BreastB	0,626	0,228	0,626	**0,694**	0,384	0,483
DBLCLA	0,408	0,800	0,605	0,585	0,664	**0,958**
DBLCLB	0,481	0,700	0,502	0,527	0,389	**0,794**
MultiA	0,874	0,899	0,805	0,828	**0,901**	0,724
Novartis	0,92	0,921	0,920	0,920	0,876	**0,966**

GRASP je pokazao 0.7, a PCA-CORREL-MEAN-XB 0.79. Na "BreastA" i "MultiA" najbolji rezultat je dao FCM i GRASP sa "Pearson" and "City Block" merama odstojanja.

Ansambl algoritmi klasterovanja su veoma popularan metod za analizu podataka o ekspresiji gena i pokazano je da daju bolje rezultate kreirajući konsenzus algoritama nego kada se koriste pojedinačni algoritmi (Monti et al., 2003; Yu et al., 2007). Komponentni algoritmi su upoređeni sa konsenzus algoritmom klasterovanja baziranom na grafovima (graph based consensus clustering - GCC) koji koristi Klastering korelacije i K-means (GCCcorr and GCCkmeans) (Yu et al., 2007) kao i sa konsenzus klastering koji koristi SOM hijerarhijsko klasterovanje (CCsom and CCkmeans) (Monti et al., 2003) u Tabeli 4.6.

Rezultati iz Tabele 4.6 pokazuju da su algoritmi dizajnirani od komponenata dali najbolje rezultate na svim skupovima podataka (neki od skupova, nisu testirana u oba rada. Skupovi koji su testirani u Monti et al., 2003 ali ne i u Yu et al., 2007 su obeleženi sa "x"). Rezultati prikazani u Tabeli 4.4. i Tabeli 4.5 pokazuju da komponentni algoritmi mogu da imaju bolje performanse od drugih tipova algoritama, što ukazuje na

potrebu za dubljom analizom sinergetskog efekta komponenti iz različitih pod-problema. Ova analiza potvrđuje naučnu zasnovanost hipoteze:

- Razmenom ideja i delova postojećih algoritama baziranim na komponentama moguće je projektovati nove algoritme koji će davati kvalitetnije modele od drugih tipova algoritama klasterovanja.

Tabela 4.6. Poređenje sa rezultatima iz (Monti et al., 2003; Yu et al., 2007).

Skup podataka	*GCCcorr*	*GCCKmeans*	*CChc*	*CCsom*	*FCM*	*Najbolji komponentni algoritam*
CNSTumors	0,658	0,718	0,549	0,429	0,452	**0,733**
Leukemia	0,831	0,831	**1**	0,721	0,910	**1**
Lung cancer	0,544	0,562	0,31	0,233	0,651	**0,920**
Novartis	x	x	0,921	0,897	0,877	**0,958**
Normal	x	x	0,572	0,487	0,415	**0,617**
St.Jude	0,873	0,86	0,948	0,825	0,952	**0,955**

Ipak, važno je naglasiti da potvrda naučne zasnovanosti ove hipoteze, ne znači da druge tipove algoritama treba isključiti iz izbora najboljeg algoritma u ovoj oblasti, već samo da treba uključiti i komponentne algoritme bazirane na predstavnicima. Dodatno, klastering baziran na konsenzusu koristi pojedinačne algoritme (često bazirane na centroidima) i tehnike ponovnog uzorkovanja (*eng. re-sampling*) ili više različitih algoritama koji kreiraju konsenzus particije i identifikuju "pravi" broj klastera, ovaj eksperiment pokazuje da uključivanje komponentnih algoritama u konsenzus okruženja bi moglo da vodi ka još boljim rezultatima. U eksperimentima u sledećoj sekciji, vrši se identifikacija dobrih komponenti (i algoritama) koji bi mogli biti korišćeni u konsenzus okruženjima za klasterovanje podataka o ekspresiji gena.

4.1.5 Identifikacija dobrih komponenata za klasterovanje podataka o ekspresiji gena

Da bi se ispitalo koliko dobro određena komponenta rešava određeni pod-problem, korišćena je sledeća procedura. Prvo je izabran pod-problem (npr. "Measure distance") i kreirane su grupe algoritama koje se razlikuju samo u jednoj komponenti. Zatim je testirano, da li su razlike između grupa značajne, korišćenjem Wilcoxon-ovog testa za

poređenje po parovima sa 95% poverenja. Ovaj test je prikladan pošto je moguće upariti algoritme iz različitih grupa koji se razlikuju samo u komponentama koje se testiraju, dok ostale komponente algoritma ostaju iste. Zatim su komponente podeljene u dve grupe: "dobre" i "loše", slično kao u (Wijaya et al., 2010). Sve komponente su u grupi "dobre", osim ukoliko se na testu nisu pokazale značajno lošije od najbolje komponente. Komponente u grupi "dobre" se preporučuju za korišćenje u algoritmu, koji bi trebao da reši konkretan skup podataka na dobar način. Ove komponente su prikazane u Tabeli 4.7. Iz Tabele 4.7 se može videti da je PCA komponenta jedina "dobra" na "Leukemia" i "CNSTumors" skupovima. PCA i GMEANS su dve "dobre" komponente na "Leukemia (small)", "Lymphoma", "DBLCLB" i "MultiA".

Tabela 4.7. Komponente sa najboljim performansama na svakom skupu podataka (Vukićević et al., 2012b)

Skup podataka	*Inicijalizacija predstavnika (najbolji)*	*Mera odstojanja (najbolji)*	*Ažuriranje predstavnika (najbolji)*	*Evaluacija klastera (najbolji)*
Leukemia (small)	GMEANS, PCA	CITY, CORREL	MEAN, MEDIAN	AIC, BIC, SILHOU
Lymphoma	GMEANS, PCA	CITY, CORREL, COSINE	ONLINE	AIC, BIC, COMPACT, XB
NCI60	RANDOM, SPSS, XMEANS, GMEANS, PCA KMEANS++	CITY, CORREL	MEAN, MEDIAN, ONLINE	AIC, BIC, SILHOU
Yeast_Cell	XMEANS, GMEANS, PCA	CORREL	MEAN	AIC, BIC
CNSTumors	PCA	CORREL, COSINE	MEAN, MEDIAN	AIC, BIC
DBLCLA	XMEANS, GMEANS, PCA	CORREL	MEAN	AIC, BIC, SILHOU
Leukemia	PCA	CORREL	MEAN, ONLINE	AIC, BIC, CONN, SILHOU
Normal	RANDOM , SPSS, KMEANS++	CORREL	MEAN, MEDIAN	AIC, BIC
Novartis	RANDOM , XMEANS, PCA, KMEANS++,	CORREL, COSINE	MEAN	AIC, BIC, COMPACT, SILHOU
St.Jude	RANDOM, XMEANS, GMEANS, PCA, KMEANS++	CORREL,COSINE	MEAN	AIC, BIC
DBLCLB	GMEANS, PCA	CORREL	MEAN	AIC, BIC, COMPACT, SILHOU
BreastA	XMEANS, GMEANS, PCA	CORREL,COSINE	MEAN, ONLINE	COMPACT, CONN, SILHOU, XB
BreastB	XMEANS	CORREL,COSINE	MEDIAN, ONLINE	AIC, BIC, SILHOU, CONN, XB
CNS Rat	RANDOM, XMEANS, GMEANS, PCA, KMEANS++	EUCLIDEAN, CITY, CORREL, COSINE	MEAN	AIC, BIC, SILHOU
MultiA	XMEANS, GMEANS, PCA	CORREL, COSINE	MEAN	SILHOU, XB
PBM	RANDOM, SPSS, XMEANS, GMEANS, PCA KMEANS++	CORREL	ONLINE	AIC, BIC, COMPACT, SILHOU, XB
Lung Cancer	RANDOM, SPSS, XMEANS, PCA, KMEANS++	EUCLIDEAN, CITY	MEAN	AIC, BIC, COMPACT, CONN, SILHOU, XB

XMEANS je bio često u "dobroj" grupi, ali u većini slučajeva nije bio značajno bolji od PCA i GMEANS (osim u slučaju "BreastB"). KMEANS++, RANDOM i SPSS su bile u "dobroj" grupi, kada performanse algoritma klasterovanja nisu zavisile od inicijalizacije predstavnika (sve ili skoro sve komponente su bile u "dobroj" grupi). Jedini izuzetak je bio skup podataka "Normal" gde su samo ove 3 komponente bile u "dobroj" grupi. Iz prethodne analize rezultata, može se zaključiti da su PCA i GMEANS komponente najbolji kandidati za izgradnju algoritama za klasterovanje podataka o ekspresiji gena.

(Giancarlo et al., 2010) su identifikovali "Pearson", "Cosine" i "Euclidean" odstojanja kao najrobusnije za klasterovanje podataka o ekspresiji gena. Rezultati iz Tabele 4.7 pokazuju da na svim skupovima podataka (osim "Lung Cancer"), CORREL ("Pearson") komponenta je bila među "dobrim". Takođe, ona je bila jedina "dobra" na 7 skupova. COSINE komponenta je bila u "dobroj" grupi 8 puta, ali uvek sa CORREL. CITY je bila u "dobroj" grupi 4 puta ali nikada kao jedinstvena "dobra" komponenta. Za razliku od (Giancarlo et al., 2010) rezultati ovog istraživanja pokazuju da EUCLIDEAN nije komponenta sa dobrim performansama (samo na "Lung Cancer" skupu je svrstana među "dobre").

Za pod-problem ažuriranja predstavnika MEAN je bila 14 puta među "dobrim" komponentama (8 puta je bila jedinstvena). ONLINE je 6 puta bila među "dobrim" komponentama, a dva puta je bila jedinstvena. MEDIAN je 5 puta bila među "dobrim" komponentama, ali nikada jedinstvena.

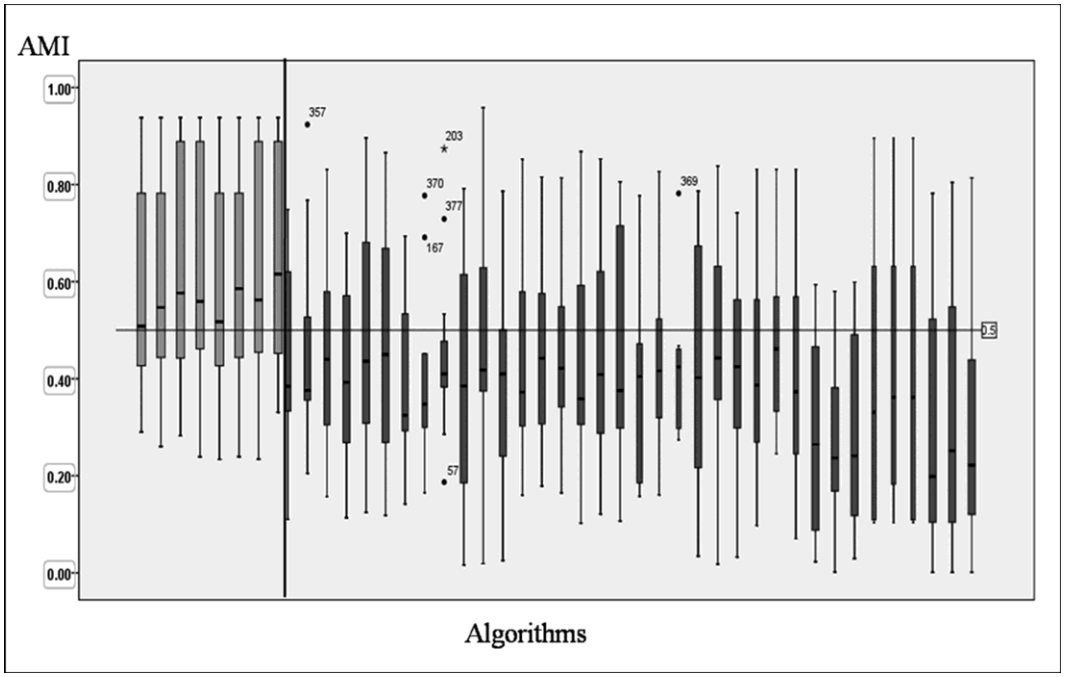
AIC i BIC su bile ne-dominirane komponente (za "evaluate clusters" pod-problem) na skoro svim skupovima podataka (osim na "BreastA" i "MultiA"), a često kao jedinstvena "dobra". SILHOU je takođe često bila u "dobroj" grupi, u većini slučajeva zajedno sa AIC i BIC. COMPACT je u većini slučajeva bila dominirana komponenta. Ovo je jako važno zapažanje, zbog toga što je ova komponenta jako često korišćena za izbor modela u originalnim algoritmima a pokazalo se da može dovesti do pogrešnih rezultata.
Iz ove analize identifikovani su dobri kandidati za dizajn algoritama za klasterovanje podataka o ekspresiji gena:

- Inicijalizacija predstavnika - PCA i GMEANS,
- Merenje odstojanja - CORREL i COSINE,
- Ažuriranje predstavnika - MEAN,
- Evaluacija klastera - AIC, BIC i SILHOU.

Posle statističke analize performansi pojedinačnih komponenata na svakom skupu podataka, testirano je da li komponente identifikovane kao "dobre", zaista kreiraju algoritme koji daju bolje rezultate od algoritama kreiranih od "loših" komponenata. Algoritmi su podeljeni u dve grupe: "dobri" sastavljeni od "dobrih" komponenata i "ostali" sastavljeni od loših komponenata (BIC komponenta je iskljucena iz analize zato što je pokazala jako slične performanse kao i AIC kada je učestvovala u dizajnu algoritama).

Ova podela komponenata po grupama je redukovala prostor algoritama (komponente nisu mešane između grupa pri dizajnu algoritama) i omogućila je lakšu identifikaciju algoritama sa dobrim performansama.

U sledećem eksperimentu razmatra se izbor algroritama iz prostora algoritama koji definišu sve dostupne komponente. Slika 4.1 pokazuje boks graf (eng. *box-plot*) AMI vrednosti (y-osa) za svaki algoritam (x-osa). Može se videti sa slike da su algoritmi sastavljeni od "dobrih" komponenti (prvih osam stubaca, levo od vertikalne referentne linije) zaista imali bolje performanse od ostalih. Dalje, korišćen je t-test za poređenje nezavisnih uzoraka (algoritama iz "dobre" i "loše" grupe) koji je pokazao značajnu razliku u performansama na nivou značajnosti ispod 0.001. Slika 4.1 i statistički t-test su pokazali da su komponente identifikovane kao dobre u prethodnom eksperimentu zaista dobri kandidati za dizajn algoritama za klasterovanje podataka o ekspresiji gena i da trebaju biti razmatrani pri dizajnu budućih algoritama. Za izvođenje još generalnijih zaključaka, trebalo bi izvršiti dodatno testiranje koje bi uključilo više komponenti i koje bi uključivalo zavisnost performansi algoritama sa karakteristikama skupova podataka.



Slika 4.1 Boks dijagram performansi algoritama na svim skupovima podataka (Vukićević et al., 2012b)

Dodatno je ispitano da li neke komponente koje nisu identifikovane kao "dobre" mogu da grade algoritme sa dobrim performansama. Regresiono stablo je korišćeno da bi se pronašla pravila prema kojima treba kombinovati komponente da bi se dobili algoritmi koji daju dobre AMI vrednosti. U tu svrhu su korišćeni rezultati svih algoritama nad svakim skupom podataka (432x17 eksperimenata ukupno). Stablo odlučivanja je primenjeno nad rezultatima i pokazalo je da postoji velika razlika u performansama algoritama nad različitim skupovima podataka. Nad skupovima 'DBLCLA', 'Leukemia', 'Leukemia (small)' , 'Novartis' i 'St.Jude', algoritmi su pokazali slične performanse (predviđena vrednost 0.730) i generalno bolje rezultate nego nad drugim skupovima (predviđena vrednost 0.397). Detaljnije ispitivanje karakteristika skupova podataka nije pokazalo da su grupe skupova podataka identifikovane stablom odlučivanja slične. Ovo upućuje na potrebu za detaljnijim istraživanjem karakteristika podataka o ekspresiji gena kao što je predloženo u (De Souto MCP et al., 2008).

Iz celog stabla odlučivanja, ekstrahovana su najzanimljivija pravila i upoređene predviđene AMI vrednosti. Iako su CITY i EUCLIDEAN imali najgore performanse (gore nego prosečne predviđene vrednosti u korenu stabla), one su imale najviše predviđene AMI vrednosti u nekim kombinacijama komponenata:

```
IF "measuring distance" IN {CITY, EUCLIDEAN}
AND "recalculate representatives"=MEAN
     AND "evaluating clusters" IN {AIC, BIC, SILHOU}
      AND "initialize representatives" IN {GMEANS, PCA}
THEN predicted AMI = 0,904
```

Druga najbolja kombinacija komponenata je predstavljena sledećim pravilom:

```
IF "measuring distance" = CORREL
  AND "evaluating clusters" IN {AIC, BIC, SILHOU}
    AND "initializing representatives" IN {GMEANS, PCA,XMEANS}
      AND "updating representatives" IN {MEAN, MEDIAN}
THEN predicted AMI = 0.893
```

Treće najbolje pravilo:

```
IF "measuring distance" = COSINE
  AND "evaluating clusters" IN {AIC, BIC, SILHOU, XB}
    AND "initialize representatives" IN {GMEANS, PCA}
      AND "updating representatives" IN {MEAN, MEDIAN}
THEN predicted AMI value = 0.874
```

Ovi rezultati potvrđuju rezultate iz Tabele 4.7 i Slika 4.1, ali sa druge strane pokazuju da dobra kombinacija komponenti (kao ona iz prvog pravila) može proizvesti dobar rezultat čak iako se te komponente ne ponašaju generalno dobro na ovim skupovima. Pravila prikupljena iz stabla odlučivanja bi trebalo koristiti kao smernice za dizajn i evaluaciju komponentnih algoritama za klasterovanje podataka o ekspresiji gena. Ovo je dosta važno pošto pristup baziran na komponentama za dizajn algoritama omogućava dizajn veoma velikog broja algoritama (u ovoj sekciji su analizirana 432) i vremenski je skupo evaluirati svaki. Prema tome, ukoliko algoritmi dobijeni iz prvog pravila (12 algoritama) daju zadovoljavajuće rezultate, tada nema potrebe za daljom evaluacijom. Tri pravila koja su opisana pokazuju koje komponente treba kombinovati da bi se dobili klasteri dobrog kvaliteta. Iz prvog pravila može se videti da CITY i EUCLIDEAN mere odstojanja ne bi trebalo koristiti sa XMEANS inicijalizacijom, MEDIAN preračunavanjem predstavnika ili XB evaluacijom klastera. Ove predložene komponente treba koristiti kao početnu tačku u potrazi za odgovarajućim algoritmom, kako bi se uštedelo vreme.

Na osnovu svega navedenog, može se potvrditi naučna zasnovanost hipoteze:

- Moguće je identifikovati podskupove (kombinacije) komponenata koje daju kvalitetnije rezultate klasterovanja u odnosu na druge podskupove komponenata.

4.2 Metod za dizajn algoritama i optimizaciju broja klastera kod klasterovanja podataka o ekspresiji gena

U prethodnoj sekciji je prikazana primena generičkog algoritma klasterovanja kao i metod za identifikaciju "dobrih" komponenata za klasterovanje podataka o ekspresiji gena. Međutim, kod primene ovog algoritma korisnik je morao sam da odredi tačan broj klastera. Kao potencijalno rešenje ovog problema primenjen je partitivno-hijerarhijski generički algoritam (Algoritmi 2 i 3 iz Sekcije 3.2) koji omogućava korisniku da definiše raspon broja klastera.

4.2.1 Podaci

Za potrebe ovog eksperimenta korišćena su četiri sintetička i šest realnih skupova podataka koji su preuzeti iz (Monti et al., 2003) i koji su korišćeni za evaluaciju algoritama klasterovanja nad podacima o ekspresiji gena. Ovi skupovi podataka su reprezentativni (Monti et al., 2003) i često su korišćeni za testiranje algoritama klasterovanja (569 citata do 22.12.2013. godine). Oni su reprezentativni jer predstavljaju prave klastere (identifikovane od strane eksperata ili generisani sintetički). Mere evaluacije su iste kao i u prethodnom eksperimentu.

Tabela 4.8. Sintetički skupovi podataka: (Monti et al., 2003)

Sintetički skupovi	*Broj klastera*	*Broj slučajeva*	*Broj atributa*
Gaussian 3	3	60	600
Gaussian4	4	400	2
Gaussian5delta3	5	500	2
Simulated 6	6	60	600

Tabela 4.9. Realni skupovi podataka: (Monti et al., 2003)

Realni skupovi	*Broj klastera*	*Broj slučajeva*	*Broj atributa*
Leukemia	3	38	999
Novartis	4	103	1000
Lung cancer	4+	197	1000
CNS Tumors	5	48	1000
St. Jude	6	248	985
Normal	13	90	1277

4.2.2 Eksperimentalna postavka

Kao što je opisano u Sekciji 3.2, hijerarhijsko-divizioni generički algoritam je omogućio dizajn još većeg broja algoritama u odnosu na prethodne eksperimente. Uz pomoć ovog algoritma i repozitorijuma komponenti izgrađeno je 1008 algoritama. Dodatno, u obzir su uzete 4 različita tipa normalizacije podataka.

U ovom eksperimentu anlaizirano je da li komponenti algoritmi mogu da pronađu tačan broj klastera u podacima o ekspresiji gena. Kao i u prethodnim eksperimentima, za evaluaciju su korišćene eksterne mere: AMI i ARI.

U Tabeli 4.10, prikazano je da za svaki skup podataka postoje neki algoritmi koji pronalaze tačan broj klastera i koji su identifikovali dobre strukture klastera (visoke vrednosti eksternih mera evaluacije). Ipak, na "CNS Tumors" i "Normal" skupovima podataka algoritmi nisu uspeli da pronađu "zlatni standard" u skrivenim podacima iako je bilo algoritama koji su pronašli tačan broj klastera. Za skup podataka "Normal", ovo je bilo i očekivano pošto ovaj skup ima daleko veći broj klastera u odnosu na ostale (13), sa jako malo objekata (90), tako da i drugi tipovi algoritama nisu uspeli da odrede prve strukture klastera (Monti et al., 2003; Yu et al., 2007). "CNS Tumors" nema ovaj problem i zbog toga, se očekuje da bi drugi tipovi algoritama mogli da budu bolje prilagođeni distribuciji ovih podataka.

Tabela 4.10. Broj algoritama na svakom skupu podataka koji su našli tačan broj K. Maksimalne i srednje vrednosti za ARI i AMI su takođe prikazane (Vukićević et al., 2012a)

Sintetički skupovi	***# Alg. sa pravim K***	***% Alg. sa pravim K***	***ARI (max)***	***ARI (avg)***	***AMI (max)***	***AMI (avg)***
Gaussian 3	462	22.92%	1.00	0.55	1.00	0.55
Gaussian4	240	11.90%	0.90	0.74	0.87	0.73
Gaussian5delta3	132	6.55%	0.94	0.84	0.92	0.84
Simulated 6	96	4.76%	1.00	0.22	1.00	0.26

Tabela 4.11. Broj algoritama na svakom skupu podataka koji su našli tačan broj K. Maksimalne i srednje vrednosti za ARI i AMI su takođe prikazane (Vukićević et al., 2012a)

Realni skupovi	*# Alg. sa pravim K*	*% Alg. sa pravim K*	*ARI (max)*	*ARI (avg)*	*AMI (max)*	*AMI (avg)*
Leukemia	23	1.14%	0.45	0.28	0.47	0.33
Novartis	491	24.36%	1.00	0.43	1.00	0.44
Lung cancer	128	6.35%	0.92	0.50	0.87	0.54
CNS Tumors	7	0.35%	0.28	0.17	0.40	0.26
St. Jude	194	9.62%	0.96	0.68	0.96	0.71
Normal	73	3.62%	0.95	0.40	0.94	0.44

Takođe se može primetiti velika razlika između prosečnih i maksimalnih AMI i ARI vrednosti. Zbog toga je pokušana generalizacija rezultata na algoritmima kod kojih je određen tačan broj klastera i gde su AMI vrednosti bile u 10% najboljih. Npr. na "Leukemia" skupu podataka (max AMI = 1) svi algoritmi koji su ostvarili AMI u rasponu [0.9 i 1] su bili analizirani.

Performanse komponentnih algoritama su takođe upoređeni sa drugim tipovima algoritama koji su objavljeni u literaturi. Rezultati su upoređeni sa (Monti et al., 2003; Yu et al., 2007)

Tabela 4.12. Poređenje sa rezultatima iz (Monti et al., 2003; Yu et al., 2007)

Realni skupovi	*GCCcorr*	*GCC Kmeans*	*CChc*	*CCsom*	*Najbolji komponentni algoritam*
CNSTumors	0,658	**0,718**	0,549	0,429	0.597
Leukemia	0,831	0,831	**1**	0,721	**1**
Lung cancer	0,544	0,562	0,31	0,233	**0,921**
Novartis	x	x	0,921	0,897	**0,960**
Normal	x	x	**0,572**	0,487	**0,572**
St.Jude	0,873	0,86	0,948	0,825	**0.955**

Rezultati pokazuju da su algoritmi dizajnirani uz pomoć razmene komponenti dali bolje ili podjednako dobre rezultate kao i ostali algoritmi na pet od 6 skupova podataka (skupovi koji su testirani u (Monti et al., 2003), ali ne i u (Yu et al., 2007) su obeleženi sa "x"). Pošto algoritmi na bazi konsenzusa koriste obične algoritme (često bazirane na predstavnicima) i tehnike višestrukog uzorkovanja ili nekoliko običnih algoritama za

kreiranje particija na bazi konsenzusa i identifikaciju pravog broja klastera, ovaj algoritam ukazuje da bi uključivanje komponentnih algoritama u konsenzus okruženja, moglo da vodi ka još boljim rezultatima.

Rezultati prikazani u Tabeli 4.12. obezbeđuju dokaz da su algoritmi zasnovani na komponentama uporedivi sa uspešno primenjivanim algoritmima za klasterovanje podataka o ekspresiji gena. Ipak, struktura najboljih komponentnih algoritama (Tabela 4.13) pokazuje da su na različitim skupovima podataka, različiti algoritmi (kombinacije komponenata) imali najbolje rezultate.

Tabela 4.13. Struktura najboljih komponentnih algoritama (Vukićević et al., 2012a)

	Normal	***IR***	***MD***	***UR***	***EC***	***glob/loc***
CNSTumors	MEANSTD	PCA	COSINE	ONLINE	COMPACT	glob
Leukemia	MAXMIN	GMEANS	COSINE	ONLINE	SILHOU	glob
Lung cancer	MEANSTD	DIANA	EUCLID	ONLINE	AIC	glob
Novartis	L2	XMEANS	CITY	MEAN	SILHOU	glob
Normal	MEANSTD	GMEANS	CORREL	ONLINE	SILHOU	glob
St.Jude	L2	PCA	CITY	ONLINE	AIC	glob

Može se primetiti da su AIC i SILHOU jedine komponente za pod-problem evaluacije klastera, a ONLINE je bila najčešća komponenta kod ažuriranja predstavnika u sastavu najboljih algoritama. Ipak, detaljnija analiza rezultata je pokazala da postoji puno algoritama koji su pokazali minimalno odstupanje performansi od najboljih, a imali su potpuno drugačiju strukturu. Zbog toga, kao i zbog činjenice da je evaluiran veliki broj algoritama, primenjene su tehnike otkrivanja zakonitosti u podacima, kako bi se generalizovali rezultati, identifikacijom pravila za dizajn algoritama sa dobrim performansama za klasterovanje podataka o ekspresiji gena. Tehnike bazirane na algoritmima OZP, za generalizaciju zaključaka iz ove sekcije, će biti predstavljene u sekciji 4.2.3.

4.2.3 Primena OZP tehnika za identifikaciju algoritma prilagođenim podacima

Kao što je već rečeno, donošenje generalnih zaključaka baziranim na eksperimentalnim rezultatima je veoma teško iz razloga što ne postoji jedan algoritam koji je najbolji za sve skupove podataka (čak ni skupove podataka u određenoj oblasti primene) i algoritmi

su obično dizajnirani za jedan ili nekoliko skupova podataka. Ipak, delimična generalizacija može biti uočena. Kao što je rečeno u prethodnoj sekciji, eksperimenti su pokazali veliku razliku između srednjih i maksimalnih AMI vrednosti algoritama koji su tačno odredili broj klastera. Zbog toga je fokus samo na najboljim algoritmima (algoritmi čije su AMI vrednosti u 10% najboljih). Visoke vrednosti AMI indeksa znače da većina objekata koji treba da budu u istom klasteru, jesu u istom klasteru. Sa druge strane, ovo ne znači da su svi algoritmi sa visokom vrednošću AMI mere prepoznali tačan broj klastera (iako je velika verovatnoća da visoka AMI vrednost označava i tačan broj klastera). U ovom slučaju, posle selekcije algoritama sa najboljim AMI vrednostima, 333 algoritma je odredilo tačan broj klastera, dok 342 nije. Zbog toga je korišćen čuveni *A priori* algoritam, za otkrivanje asocijativnih pravila, koji bi mogao da identifikuje pravila (kombinacije komponenti) koje su vodile do otkrivanja tačnog broja klastera.

Asocijativna pravila su predstavljena u formi: *AKO pretpostavka ONDA posledica* i kao ulazne promenljive (AKO deo pravila - *pretpostavke*) korišćene su sve komponente, svih pod-problema, dok je kao izlazna promenljiva (ONDA deo pravila - posledica) bila binarni indikator ("true ili false") tačnog broja klastera.

Tabela 4.14. Asocijativna pravila za generalizaciju rezultata (Vukićević et al., 2012a)

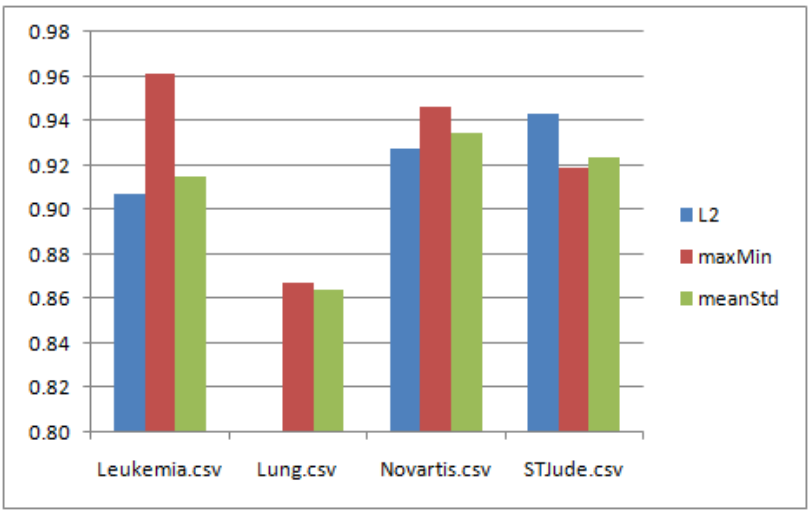
IF	***THEN True K***	***Support***	***Confidence***
ND =MEANSTD UR=MEAN MD =CORREL	FALSE	12.44	91.67
ND =MEANSTD UR=MEDIAN MD =CORREL	FALSE	12.44	91.67
ND = MAXMIN UR=ONLINE	TRUE	15.41	80.01
ND = MAXMIN UR=ONLINE G/L = LOCAL	TRUE	8.44	82.46

Kvalitet asocijativnih pravila je meren uz pomoć podrške (eng. *support*) i sigurnosti (eng. *confidence*). Podrška predstavlja proporciju slučajeva za koje važi celo pravilo (AKO i ONDA deo) u ukupnom broju slučajeva. Sigurnost pokazuje odnos podrške pretpostavke (AKO dela) i ukupne podrške (AKO i ONDA dela delova). Nakon

izbacivanja pravila sa niskom podrškom i sigurnošću, kao i redudantnih pravila, ostala su zanimljiva pravila koja su prikazana u Tabeli 4.14:

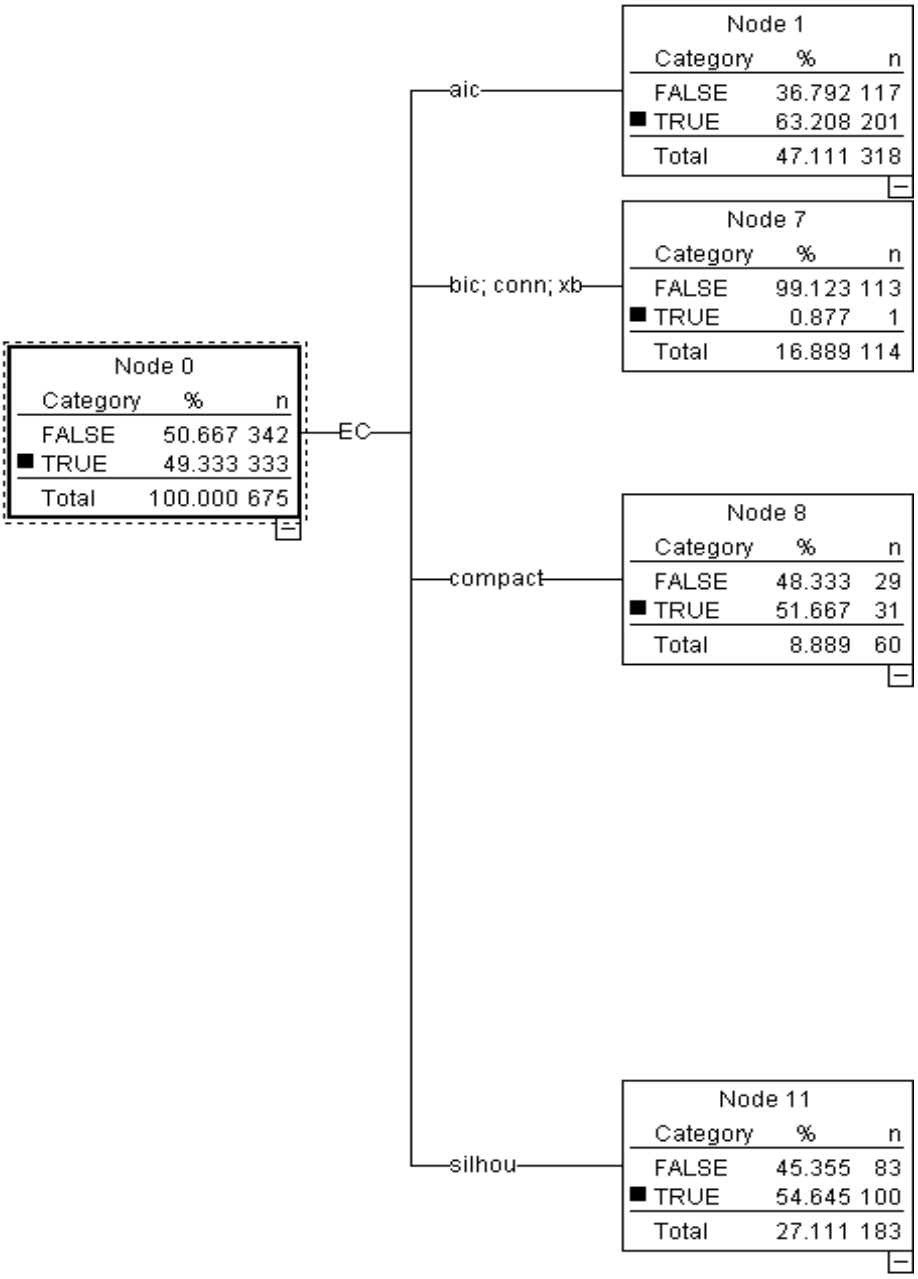
Iz Tabele 4.14 iznad, može se zaključiti da MEAN i MEDIAN komponente iz pod-problema ažuriranja predstavnika ne bi trebalo koristiti za dizajn algoritama za klasterovanje podataka o ekspresiji gena. Sa druge strane, tačan broj klastera je jako često određivala komponenta ONLINE kada je kombinovana sa MAXMIN tehnikom normalizacije i lokalnom divizionom strategijom.

Iz prethodnog razmatranja može se zaključiti da normalizacija skupa podataka ima uticaj na kvalitet algoritama klasterovanja. Sa slike 4.2, može se videti da se L1 normalizacija nikada ne pojavljuje u algoritmima koji mogu da otkriju pravu strukturu klastera.



Slika 4.2. AMI vrednosti za različite normalizacije na realnim skupovima podataka (Vukicević et al., 2012a)

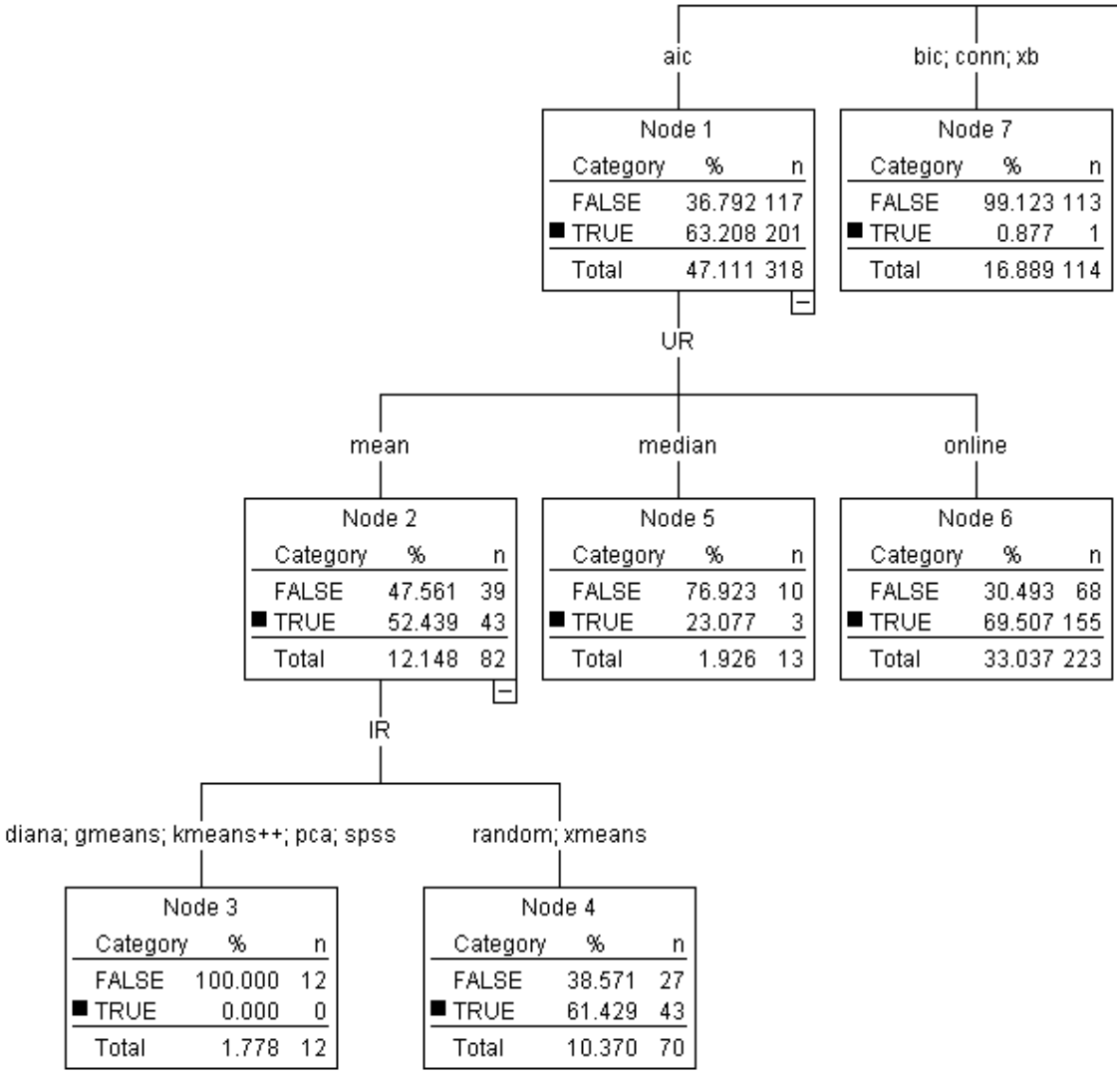
Dalje ispitivanje uticaja različitih kombinacija komponenti na mogućnost algoritama da pronađu tačan broj klastera je vršeno uz pomoć stabla odlučivanja C4.5. Ulazne i izlazne promenljive su ostale iste kao i za opisani *A priori* algoritam. Stablo odlučivanja je identifikovalo pod-problem evaluacije klastera (selekcije modela) kao najznačajniji, pošto je prvi nivo stabla odlučivanja kreiran na osnovu komponenti iz ovog pod-problema.



Slika 4.3. Grananje stabla odlučivanja po različitim komponentama za evaluaciju klastera (Vukicević et al., 2012a)

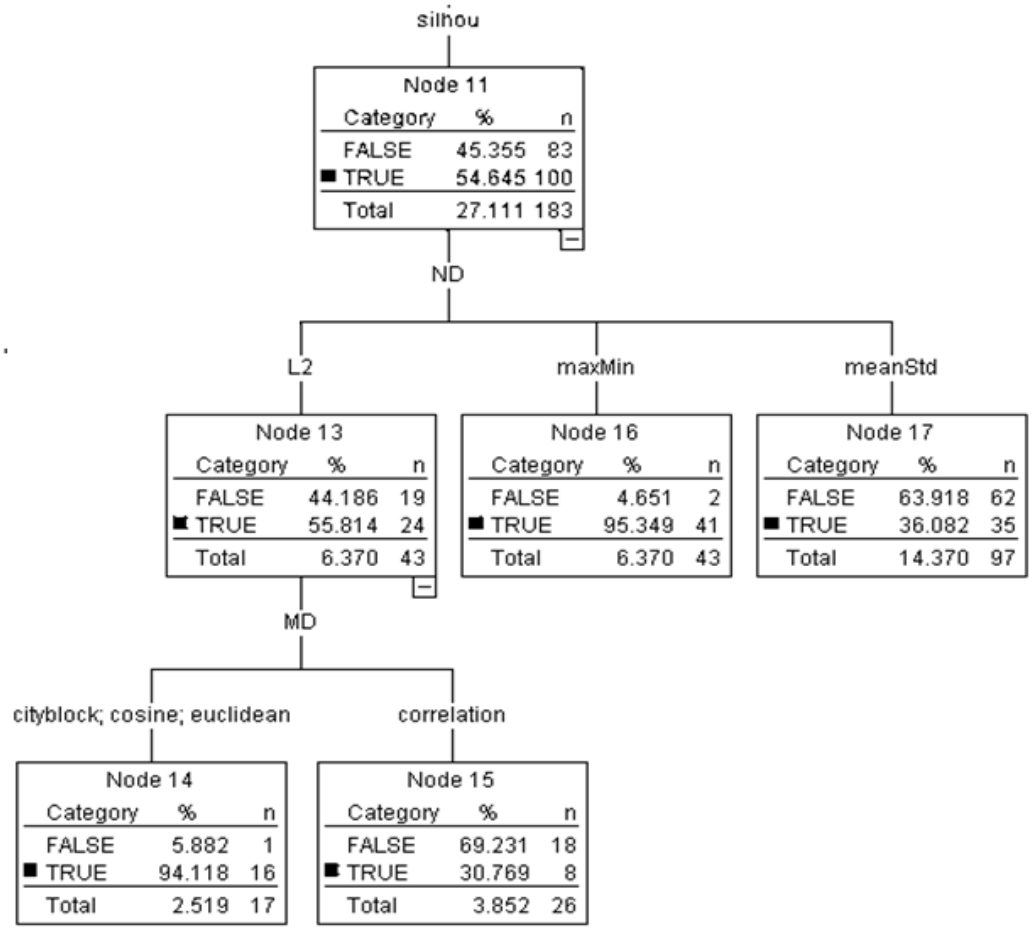
Sa Slike 4.3. se može jasno videti da se BIC, CONN i XB komponente ne bi trebalo koristiti za dizajn algoritama klasterovanja u ovoj oblasti zato što su u 113 od 114 slučajeva, ove komponente uzrokovale identifikaciju pogrešnog broja klastera. Algoritmi koji su koristili AIC su najčešće uspevali da identifikuju tačan broj klastera, ali ne sa svim kombinacijama komponenata (201 od 318 algoritama sa AIC komponentom je pronašlo tačan broj klastera). Detaljnija analiza stabla odlučivanja je pokazala da je AIC sa ONLINE komponentom najčešće otkrivala tačan broj klastera. U

kombinaciji sa MEAN komponentom za ažuriranje predstavnika dobro su se pokazale RANDOM i XMEANS komponente za inicijalizaciju (Slika 4.4).



Slika 4.4. Pravila o algoritmima koji uključuju AIC komponentu (Vukicević et al., 2012a)

Model stabla odlučivanja je takođe otkrio interesantna pravila o algoritmima koji koriste SILHOU komponentu za evaluaciju modela (Slika 4.5.) U kombinaciji sa MAXMIN normalizacijom daje dobre rezultate. Ukoliko se koristi sa L2 normalizacijom trebalo bi koristiti CITY, COSINE ili EUCLID meru odstojanja.



Slika 4.5. Pravila o algoritmima koji uključuju SILHOU komponentu RC (Vukicević et al., 2012a)

Ovi rezultati potvrđuju naučnu zasnovanost hipoteze:

- Korišćenjem partitivno hijerarhijskog algoritma baziranog na komponentama moguće je automatski odrediti tačan broj klastera za konkretan skup podataka,

međutim, bitno je napomenuti da bi za donošenje generalnih zaključaka neophodno ponoviti ove eksperimente nad većim brojem skupova podataka.

4.3 Identifikacija korelacije između internih i eksternih mera evaluacije algoritama klasterovanja

Jedan od najvažnijih problema klasterovanja je evaluacija klaster modela (Halkidi et al., 2001). U skorije vreme, postoji mnogo napora da se identifikuju adekvatne metodologije za korišćenje internih i eksternih mera za evaluaciju klaster modela. Ipak ove metodologije su obično razvijene posebno za interne i posebno za eksterne mere. Kao što je već rečeno, interne mere za evaluaciju klaster modela, se obično koriste nakon procesa klasterovanja (evaluacija konačnih modela). Sa druge strane, postoje algoritmi koji optimizuju interne mere za vreme izvršenja tih algoritama (npr. K-means optimizuje ukupno odstojanje unutar klastera). Primeri ovakvih algoritama se mogu pronaći u (npr. Zhao and Karypis, 2004; Cagnina et al., 2008) i pokazali su veoma obećavajuće rezultate. Ipak se često dešava da se evaluacija konačnih modela evaluira sa nekom drugom internom merom (npr. Silhouette indeks) i o čemu je diskutovano u (Famili et al., 2004). Kao alternativa ovom pristupu, predloženo je da se ista interna mera koristi i za optimizaciju funkcije cilja kao i za evaluaciju finalnog modela (Dom, 2001). Primećeno je da je ovakav pristup moguć samo ukoliko određena interna mera evaluacije dobro oslikava poželjne osobine u određenom slučaju primene, kao i da postoji algoritam koji efikasno optimizuje tu meru. Takvi algoritmi mogu biti dizajnirani, ali je problem što korisnici najčešće ne mogu unapred da odrede koja je to geometrijska osobina (interna mera) poželjna u njihovom slučaju primene.

Sa druge strane, eksterne mere evaluacije, mogu da obezbede objektivnu informaciju o tome koliko je dobar klaster model u odnosu na postojeće klase u podacima, ukoliko postoje ("zlatni standard"). Problem je u tome, što eksterne mere ne mogu biti optimizovane u toku izvršenja algoritama, zbog "nenadgledane" prirode procesa klasterovanja (prave klase nisu poznate unapred, kao kod klasifikacionih problema). Kao korak prema rešenju ovog problema, (Vukićević et al., 2011) su predložili metodu za detekciju internih mera koje su visoko korelisane sa eksternim merama, a za identifikaciju ovakvih mera je korišćen komponentni pristup za izgradnju algoritama klasterovanja i ova metoda će biti predstavljena u ovoj sekciji.

Kako bi se generalizovali zaključci o korelaciji između internih i eksternih mera, 432 (kao što je opisano u sekciji 4.1.1) algoritma je dizajnirano uz pomoć komponentnig pristupa. Analiza tako velikog broja algoritama, omogućilo je proučavanje korelacija između internih i eksternih mera na čitavoj klasi algoritama baziranih na predstavnicima a ne samo na pojedinačnim algoritmima (npr. K-means).

Sprovedena su tri eksperimenta, sa ciljem da se:

- identifikuju interne mere koje su najviše korelisane sa eksternom merom AMI,
- unaprede performanse algoritama (merene AMI indeksom) optimizacijom najbolje korelisane komponente selekciju modela (evaluaciju klastera) analizira korelacija internih i drugih eksternih mera (pored AMI indeksa).

4.3.1 Pregled sličnih pristupa

Sličan pristup je predložen od strane (Ingaramo et al., 2008), gde je K-star algoritam korišćen da bi se istražila korelacija između nekoliko internih mera i F-mere (eksterne) i zaključili su da neke od njih mogu poboljšati performanse algoritama za klasterovanje dokumenata. Takođe, u (Errecalde et al., 2010) je pokazano da interne mere mogu biti korišćene u različitim etapama klastering algoritma baziranog na AntTree modelu, kako bi se popravile performanse. (Gurrutxaga et al., 2011) su predložili da bi interne mere trebalo validirati na osnovu eksternih. Oni su koristili K-means da generišu različite particije i poredili particije po najboljim internim merama u odnosu na najbolje eksterne mere. (Ben-Hur et al., 2002) su predložili metod koji istražuje mere stabilnosti klaster algoritma na osnovu preturbacije (različitog uzorkovanja) objekata iz skupa podataka. Ova metoda se može koristiti kod bilo kog algoritma klasterovanja. Može se koristiti za određivanje tačnog broja klastera kao i nedostatak strukture u podacima.

Predlog metodologije (Vukićević et al., 2011) i koja će biti predstavljena u ovoj sekciji, do neke mere prati rad (Ingaramo et al., 2008; Gurrutxaga et al., 2011). Ovde se istražuje korelacija između nekoliko internih mera sa preporučenim eksternim merama za evaluaciju algoritama klasterovanja (Vinh et al., 2010; Wu et al., 2009). Dodatni cilj

je bio generalizacija rezultata na klasu klastering algoritama baziranih na predstavnicima uz pomoć dizajna baziranog na komponentama.

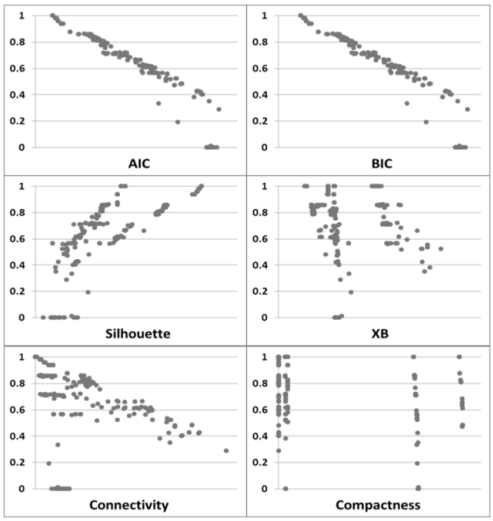
4.3.2 Korelacija internih mera sa AMI indeksom

Kao što je prethodno diskutovano, korišćenje eksternih (nadgledanih) mera pri izvršenju algoritama klasterovanja, bi poremetilo nenadgledanu prirodu ovih algoritama. Ipak bi bilo zgodno odrediti koje interne mere (ukoliko uopšte postoje) bi mogle biti korišćene u toku izvršenja algoritama za donošenje odluka o izboru klaster modela koji će biti najbliži pravim klasama (zlatnom standardu).

U ovim eksperimentima, relacija između internih i eksternih mera je analizirana na osnovu statističke korelacije. Korelacija je računata nad 432 algoritma koji su opisani u sekciji 4.2. Korišćenje velikog broja algoritama umesto samo jednog (kao u većini drugih istraživanja), značajno povećava generalnost zaključaka, pošto je korelacija merena na široj klasi algoritama baziranih na predstavnicima.

Da bi se izabrala odgovarajuća interna mera (surogat) za eksternu meru, ispitana je njihova korelacija. Analiza korelacije između eksternog indeksa AMI i 6 internih mera na skupu podataka *Gaussian3* je prikazana na slici ispod. Ovaj rezultat pokazuje da su među analiziranim merama, AIC i BIC jako korelisani sa AMI merom. Ipak uočene su još neke pravilnosti.

.



Slika 4.6. Distribucija AMI vrednosti za šest internih meta na Gaussian 3 skupu podataka

Npr. može se videti da *compactness* ima nekoliko grupa dobro razdvojenih vrednosti. Nakon detaljnije analize otkriveno je da je izbor mere odstojanja i načina ažuriranja predstavnika uticao na ostvarene vrednosti *compactness* mere. Ovo znači da algoritmi koji se razlikuju u ovim komponentama ostvaruju vrednosti *compactness* mere koje nisu uporedive, čime je *compactness* neupotrebljiva za poređenje algoritama koji se čak i minimalno razlikuju.

AIC i BIC su jako korelisani sa AMI merom i imali su identične vrednosti zbog njihove slične formule izračunavanja. Jedina razlika je u tome da BIC zavisi od veličine skupa podataka i zbog toga (vrednosti su poređene na istim skupovima podataka) su dobijene identične vrednosti korelacije sa AMI. Zbog toga se daljim eksperimentima prikazuje rezultati korelacija između AIC i AMI.

Tabela 4.15. Korelacija između AMI indeksa i internih mera na sintetičkim skupovima podataka

	Compactness	***XB***	***Silhouette***	***Connectivity***	***AIC/BIC***
Gaussian3	0.258	0.254	0.771	0.535	**0.965**
Gaussian4	0.276	0.512	0.391	0.539	**0.988**
Gaussian5	-0.565	0.152	-0.388	-0.548	**0.926**
Simulated6	0.263	0.421	0.667	0.4	**0.9**

Tabela 4.16. Korelacija između AMI indeksa i internih mera na realnim skupovima podataka

	Compactness	***XB***	***Silhouette***	***Connectivity***	***AIC/BIC***
Leukemia	0.254	0.599	0.711	0.884	**0.916**
Novartis	0.099	0.73	0.722	0.732	**0.911**
Lung cancer	-0.320	-0.160	0.333	**0.806**	0.627
Normal	0.569	0.116	0.632	0.557	**0.736**
St. Jude	0.179	0.579	0.705	0.553	**0.949**
CNS Tumors	0.366	0.333	0.205	0.248	**0.414**

Tabela 4.15 i Tabela 4.16 prikazuju da je korelacija između AMI mere i internih mera bila konzistentna na svim skupovima podataka. Za sve interne mere osim *Silhouette*, niže vrednosti su bolje. Za lakšu interpretaciju ovih rezultata, promenjen je znak svih mera koje je trebalo minimizovati. Prema tome, negativna korelacija znači da se AMI mera pogoršava (opada) dok se vrednost interne mere poboljšava (raste), čime takva mera postaje neupotrebljiva za ovakav zadatak. Može se videti da velika korelacija postoji između AIC/BIC i AMI mere. Jedini izuzetak je skup podataka *lung cancer*, gde mera *connectivity* ima veću korelaciju sa AMI.

Ipak posle detaljnije analize, je identifikovano da je za većinu algoritama na ovom skupu podataka, AMI index bio ispod 0.5, što ukazuje na na to da algoritmi nisu uspeli da identifikuju "pravu" strukturu klastera. Moguće je da je uzrok tome činjenica da za taj skup podataka i dalje nije utvrđen tačan broj klastera. Kada su selektovani samo algoritmi koji ostvaruju AMI preko 0.5 na ovom skupu podataka, korelacija sa AIC merom je bila 0.82, a sa *connectivity* 0.64.

4.3.3 Selekcija modela bazirana na internim merama

Eksperimenti prikazani u prethodnoj sekciji su ukazali na to da su AIC i BIC najprikladnije interne mere za procenu AMI u odsustvu znanja o "pravim klasterima" kod podataka o ekspresiji gena i na osnovu rezultata i diskusije može se potvrditi naučna zasnovanost hipoteze:

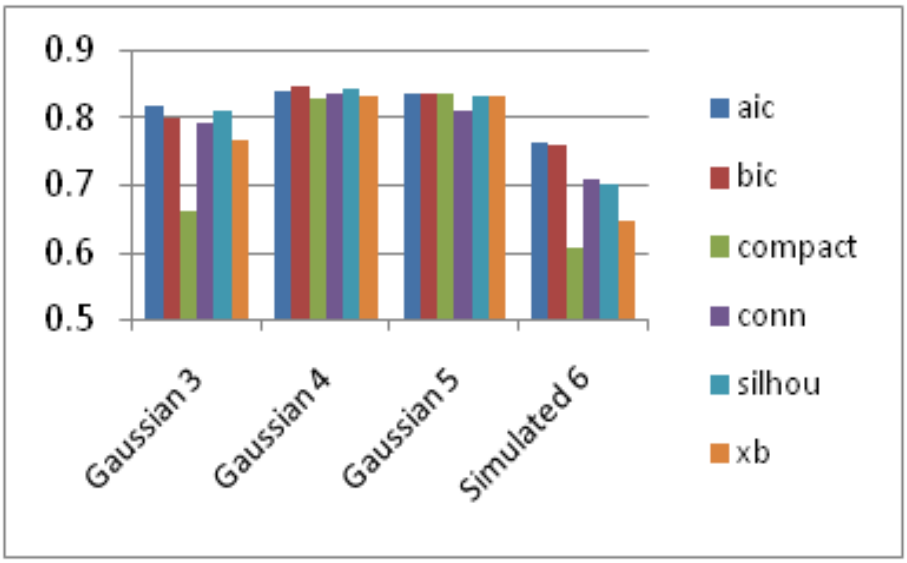
- Postoji korelacija između nekih internih i eksternih mera evaluacije algoritama za klasterovanje ekspresija gena.

Kako su AIC i BIC pokazali identične performanse dalje je detaljnije analizirana AIC mera i postavljena je dodatna hipoteza:

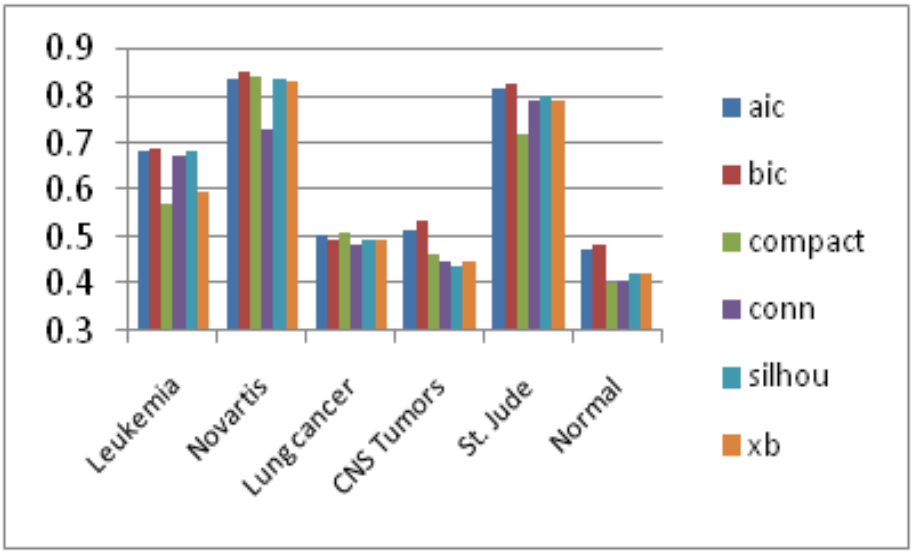
Dodatna hipoteza 1: AIC interna mera za evaluaciju algoritama klasterovanja može biti korišćena kao odrednica za selekciju modela klasterovanja i da će algoritmi koji koriste ovu meru imati značajno bolje rezultate (AMI mera) u odnosu na ostale.

Kao što je rečeno, u ovom istraživanju je korišćeno 432 algoritma koji su koristili 6 različitih interne mere za vreme svog izvršavanja. Ovi algoritmi su podeljeni u 6 grupa od kojih se svaka sastoji od 72 algoritma: u svakoj grupi je korišćena različita (ali fiksirana u grupi) interna mera za selekciju modela, a ostale komponente su varirane. Zatim je testirana hipoteza da algoritmi koji sadrže AIC internu meru daju značajno bolje rezultate nego algoritmi sa drugim internim merama (iz drugih grupa). Poređene su prosečne AMI vrednosti po grupama. Pošto performanse algoritama najviše zavise od karakteristika konkretnog skupa podataka, algoritmi su evaluirani na svakom skupu posebno.

Sa Slike 4.7 i Slike 4.8 se može videti da su najbolje ili skoro najbolje prosečne AMI vrednosti dobijene u grupama gde su AIC i BIC korišćeni kao mere za selekciju modela. Na skupovima podataka gde ove dve mere nisu dale najbolje rezultate, AMI vrednosti su bile veoma slične za sve algoritme.



Slika 4.7. Prosečne AMI vrednosti algoritama sa različitim komponentama za selekciju modela na sintetičkim skupovima podataka



Slika 4.8. Prosečne AMI vrednosti algoritama sa različitim komponentama za selekciju modela na realnim skupovima podataka

Da bi ustanovili da su razlike u prosečnim AMI vrednostima prikazane na Slici 4.7 i Slici 4.8. značajne korišćen je Vilkoksonov (Wilcoxon) upareni test rangova. Algoritmi su poređeni po parovima gde su komponente u svakom paru algoritama iste po svim

pod-problemima, osim u evaluaciji klastera (npr. RANDOM-MEAN-AIC je poređen sa RANDOM-MEAN-COMPACT). Na ovaj način je izolovan uticaj komponenata iz pod-problema evaluacije klastera na performanse algoritama. Tabela 4.17 i Tabela 4.18 pokazuju da su u većini slučajeva gde su algoritmi koristili AIC, dobijeni značajno bolji rezultati nego pri korišćenju drugih komponenti (ovde, * označava nivo značajnosti od 0.05, **0.01, i ***0.001).

Tabela 4.17. Nivoi značajnosti razlika u performansama algoritama koji koriste AIC i algoritama koji koriste druge mere evaluacije na sintetičkim skupovima podataka

	SILHOU	***CONN***	***XB***	***COMPACT***
Gaussian 3	0.970	0.153	0.006**	0.000***
Gaussian 4	0.601	0.788	0.758	0.586
Gaussian 5	0.761	0.000**	0.002*	0.737
Simulated 6	0.018*	0.030*	0.000	0.000***

Tabela 4.18. Nivoi značajnosti razlika u performansama algoritama koji koriste AIC i algoritama koji koriste druge mere evaluacije na realnim skupovima podataka

	SILHOU	***CONN***	***XB***	***COMPACT***
Leukemia	0.791	0.699	0.006**	0.001**
Novartis	0.645	0.01**	0.871	0.529
Lung cancer	0.732	0.219	0.585	0.459
Normal	0.000***	0.000***	0.003**	0.001**
St. Jude	0.168	0.050*	0.040*	0.000***
CNS Tumors	0.000***	0.000***	0.002**	0.000***

Iz prethodnih rezultata može se zaključiti da su AIC i BIC komponente konzistentno davale najbolje ili skoro najbolje rezultate. Ipak korišćenje nekih drugih mera za izbor najboljeg modela je ponekad davalo dobre rezultate (Slika 4.7 i Slika 4.8) iako u prethodnim eksperimentima nisu pokazale jaku korelaciju sa AMI merom. Ipak, u realnoj primeni (bez prethodnog znanja o pravim klasterima) nedostatak korelacije može dovesti do nekonzistentnih odluka. Korisnik tada ne može biti siguran, da li će klaster model koji je izabran korišćenjem mere koja je slabo korelisana sa AMI, proizvesti tačne strukture klastera (dobre vrednosti eksterne mere). Jako korelisane interne mere, kao što je AIC, mogu nam dati određenu sigurnost u ostvarenju dobrih vrednosti eksternih mera, zato što su se ponašale konzistentno na analiziranim skupovima podataka čime se potvrđuje i dodatna hipoteza definisana u ovoj sekciji.

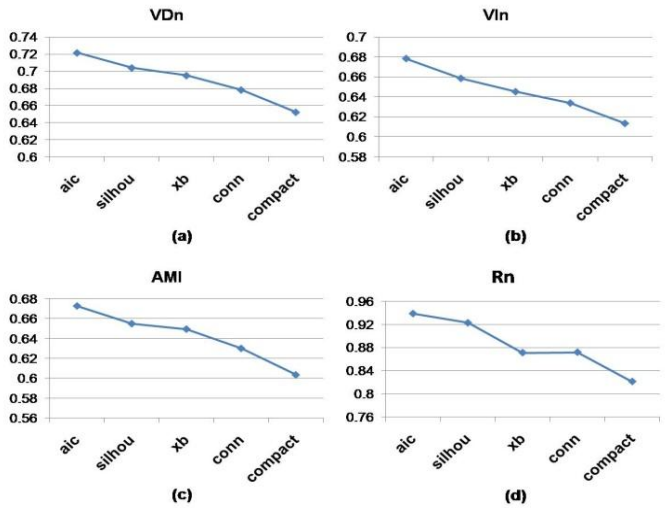
Ipak, kao i kod zaključaka u sekciji 4.2, i ovaj zaključak treba uzeti sa rezervom pošto su eksperimenti rađeni na malom broju skupova podataka. Za generalizaciju ovih

zaključaka na problem klasterovanja podataka o ekspresiji gena bio bi potreban daleko veći broj reprezentativnih skupova podataka, a takođe bi trebalo uzeti u obzir i njihove osobine. Korak ka ovoj generalizaciji biće prikazan u Sekciji 5.

4.3.4 Kvalitet selektora modela baziran na internim merama evaluacije

U ovoj sekciji je prikazan ponovljeni eksperiment iz prethodne sekcije, ali sa drugim eksternim merama koje su preporučene u literaturi za evaluaciju kvaliteta modela koji su dobijeni algoritmima baziranim na predstavnicima. Rezultati su bili konzistentni sa rezultatima dobijenim sa AMI merom. Obzirom da su rezultati slični ovde neće biti prikazani detaljni rezultati sa svim merama, a umesto toga biće prikazane srednje vrednosti performansi na svim skupovima podataka. Eksterne mere prikazane na slici ispod imaju različite apsolutne vrednosti, pošto imaju različite gornje i donje granice (Vinh et al., 2010; Wu et al., 2009) i zbog toga nisu uporedive među sobom. Za AMI i Rn, vrednosti bliže 1 označavaju bolje rezultate klasterovanja, dok su vrednosti bliže 0 bolje za VIn i VDn. Zbog jasnije prezentacije vrednosti VIn i VDn su invertovane (1-*vrednost*).

Na Slici 4.9, prikazane su prosečne performanse 432 algoritma sa različitim komponentama za selekciju modela na svih 10 skupova podataka. Može se videti da se sve eksterne mere slažu oko ranga mera za selekciju modela i da je AIC uvek najbolji.



Slika 4.9. Uticaj 5 internih mera na performanse algoritama meren sa 4 eksterna indeksa

Ponovo je testirana statistička značajnost razlika između AIC i drugih mera, sa Vilkoksonovim uparenim testom rangova. U Tabeli 4.19, nivo značajnosti razlika je analiziran po svakoj eksternoj meri. Može se videti da AIC pokazuje značajno bolje performanse u većini slučajeva, osim za SILHOU, gde je značajna razlika evidentirana samo kod VIn.

Tabela 4.19. Značajnost razlika prosečnih performansi između AIC i ostalih internih mera

	SILHOU	**CONN**	**XB**	**COMPACT**
AMI	0.095	0.000	0.000	0.000
VIn	0.018	0.000	0.000	0.000
VDn	0.141	0.000	0.000	0.000
Rn	0.272	0.000	0.000	0.000

5 Meta - učenje za algoritme klasterovanja za ekspresiju gena

Kao što je pominjano u uvodnoj sekciji ovog rada, eksponencijalni rast podataka, kao i veliki broj računski i vremenski zahtevnih algoritama su doveli do jednog od najvećih problema modernog OZP: pronaći najbolji algoritam za konkretne podatke (Iam-on et al., 2010). Ovo je veliki problem pošto većina analitičara obično nema vremena i/ili resursa za kreiranje i evaluaciju svih modela uz pomoć svih raspoloživih algoritama. Jedan od obećavajućih pristupa za rešavanje ovog problema je pristup meta-učenja (Smith-Miles, 2008; Smith-Miles, 2009).

5.1 Sistemi Meta-učenja

Meta-učenje je metodologija koja rešava izbor algoritama kod različitih zadataka OZP na osnovu prethodnog znanja (istorijskih podataka o eksperimentima). Osnovna ideja je skladištenje eksperimentalnih rezultata. Pod eksperimentalnim rezultatima se podrazumevaju podaci o:

- algoritamima koji su evaluirani,
- meta-podaci o skupovima podataka (statističke karakteristike skupova podataka),
- performanse algoritama na svakom skupu podataka (npr. za tačnost klasifikacije kod problema klasifikacije ili AMI kod problema klasterovanja).

U pristupu meta-učenja, kreira se meta-model (klasifikacioni ili regresioni), koji predviđa rang algoritma ili vrednost performanse algoritma na novom skupu podataka. Prednost ovakvog pristupa je što meta-model može da predvidi performanse algoritama na podacima koji još nisu viđeni, odnosno nema potrebe za testiranjem svih raspoloživih algoritama.

Sistem meta-učenja se može opisati preko sledećih elemenata (Smith-Miles, 2008; Smith-Miles, 2009):

- Prostor problema, P, koji predstavlja skup instanci klase datog problema (najčešće problemi klasifikacije i regresije);
- Prostor meta-atributa, M, koji sadrži karakteristike koje opisuju dati problem (između ostalog veličina uzorka za trening, korelacija između atributa);
- Prostor algoritama, A, koji predstavlja skup algoritama kandidata za rešavanje problema u prostoru P;
- Metrike performansi, Y, koji meri performanse algoritma na datom problemu (npr. tačnost klasifikacije (za klasifikacione probleme) ili koren srednje kvadratne greške (za probleme regresije)).

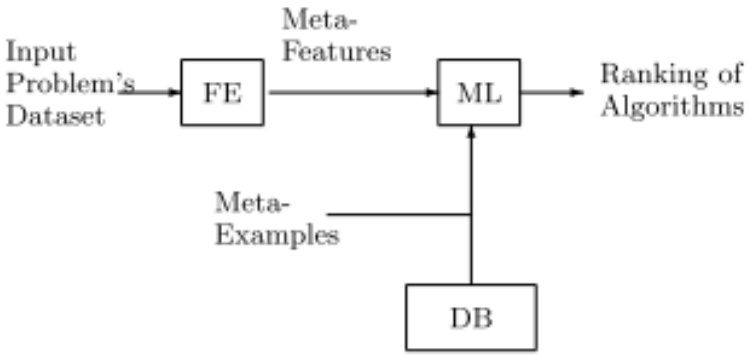
Generalno, postupak kreiranja i primene sistema meta-učenja je sledeći: prvo se skupovi podataka iz prostora problema P evaluiraju korišćenjem algoritama iz prostora algoritama A. Pored toga, meta-karakteristike skupova podataka se dovode u vezu sa performansama algoritma (formira se baza meta-primera). Nakon toga, kreiraju se (i ocenjuju metrike performanse) regresioni ili klasifikacioni modeli kako bi se odredili paterni odnosa između meta-atributa i performansi algoritma. Primena modela se vrši, tako što se za novi skup podataka (problem), izdvajaju meta-atributi i vrši se predviđanje performansi. Ovaj postupak je veoma važan kada je reč o velikim skupovima podataka zbog toga što se drastično smanjuje broj algoritama koje je potrebno evaluirati nad originalnim skupom podataka.

Važno je napomenuti da je meta-učenje uobičajena tehnika za izbor i rangiranje algoritama koje uče sa nadgledanjem (klasifikacija ili regresija), ali kako klasterovanje predstavlja tehniku učenja bez nadgledanja i kako, prema (Iam-on et al., 2010), ne postoji konsenzus o najboljoj meri evaluacije koja određuje željene karakteristike skupa podataka (unutrašnje mere evaluacije), primena meta-učenja za probleme klasterovanja je izučavana samo u poslednjih nekoliko godina (Nascimento et al., 2009).

Glavni problem leži u tome što korisnici uglavnom ne znaju tačno koja je karakteristika grupa podataka (interna mera evaluacije) adekvatna za njihovu oblast primene. S druge strane, spoljne mere za procenu klastera mogu obezbediti objektivne informacije o tome

koliko je klaster dobar u odnosu na prave klase u podacima (eng. „ground truth") i postojeći sistemi meta-učenja u oblasti klasterovanja su bazirani baš na ovim merama.

Specifičan sistem meta-učenja za klasterovanje genskih ekspresija (koji je proširen u ovom istraživanju) predložili su (Prudencio et al., 2011). Procena kvaliteta algoritma i stvaranje baze podataka meta-uzoraka je zasnovano na korigovanom Rand indeksu, koji predstavlja eksternu meru evaluacije. Mehanizam zaključivanja se zasniva na klasifikacionim algoritmima, dok se kao izlazni atribut koristi rang evaluiranih algoritama klasterovanja. Sistem je prikazan na slici 5.1.



Slika 5.1 – Sistem meta-učenja za klasterovanje ekspresija gena (Prudencio et al., 2011)

Još jedna specifičnost ovog sistema su meta-atributi koje se koriste za opis podataka genskih ekspresija. Ovde ćemo ukratko opisati meta-atribute korišćene u pomenutom radu:

1. LgE: $\log_{10}$ uzorka. Predstavlja sirov pokazatelj uzorka za obuku modela;
2. LgREA: $\log_{10}$ odnosa uzorka i broja atributa. Predstavlja sirov pokazatelj koliko je raspoloživo uzoraka po atributu;
3. PMV: procenat nedostajućih vrednosti. Pokazatelj kvaliteta podataka;
4. MN: Multivarijantna normalnost. Predstavlja grubi pokazatelj usklađenosti distribucije podataka sa distribucijom normalne raspodele;
5. SK: Statistički pokazatelj asimetričnosti podataka;

6. Chip: tip tehnologije čipa koji je korišćen za uzimanje uzorka (cDNA ili Affymetrix);
7. PFA: procenat atributa koji se koristi nakon primene filtera za izbor atributa;
8. PO: procenat izuzetaka. Predstavlja statistički pokazatelj kvaliteta podataka;
9. NRE: normalizovana relativna entropija. Predstavlja pokazatelj koliko je odstupanje stvarne distibucije među klasama u odnosu na uniformnu raspodelu;
10. SC10: "mali" klasteri. Mera koja predstavlja broj klasa sa kardinalnošću, odnosno brojem uzoraka, manjom od praga $\theta = 10$;
11. SC15: slična mera kao i prethodna. Jedina razlika je u pragu, koji iznosi $\theta = 15$;
12. BC: "veliki" klasteri. Mera koja predstavlja broj klasa sa kardinalnošću većom od praga $\theta = 50$;
13. k-NN outliers: greška klasifikacije dobijena primenom metode *k* najbližih suseda ($k = 3$). Još jedan pokazatelj kvaliteta podataka.

Pored meta-atributa, u ovom istraživanju, korišćene su i komponente (uključujući i tip normalizacije) i interne mere evaluacije kao meta-atributi koji će služiti za rangiranje i izbor algoritma za klasterovanje genskih ekspresija.

Ovaj sistem meta-učenja se sastoji iz dve faze. Prva faza je treniranje, dok je druga faza primena. U fazi treniranja, algoritam "meta-učenja" (ML) izvlači znanje iz skupa podataka meta-primera sačuvanih u bazi podataka (DB). Takvo znanje se odnosi na karakteristike podataka i performanse algoritama kandidata. Pored meta-atributa, koji su opisani u prethodnom odeljku, korišćeni su meta-atributi algoritma u DB modulu. Novi meta-atributi su RC od četiri podproblema opisanih u tabeli 4.1, i jedan dodani atribut koji opisuje tehniku normalizacije korišćenu pre klasterovanja. Pored ovih atributa, dodati su i atributi koji predstavljaju interne mere evaluacije algoritma. Time se sistem proširuje na ukupno 24 meta-atributa.

U fazi korišćenja, za novi skup podataka, FE određuje vrednosti meta-atributa koji opisuje podatke i skladište algoritama (AL) koji, zatim, pruža opise dostupnih

algoritama. Na osnovu tih rezultata ML modul izbacuje rang dostupnih algoritama. Da bi se to omogućilo, ML modul koristi znanje dobijeno kao rezultat trening faze.

Podaci nad kojima će se sprovesti ovo istraživanje će biti 30 skupova podataka genskih ekspresija, korišćenih u (Nascimento et al., 2009; Prudencio et al., 2011). Osnovne karakteristike skupova podataka su date u tabeli 5.1.

Tabela 5.1. Osnovne karakteristike skupova podataka koji su korišćeni pri razvoju sistema meta-učenja

Skup podataka	Tip čipa	Tkivo	Uzorak	#Atributa	#Klasa
Armstrong-2002-v1	Affymetrix	Krv	72	12582	2
Armstrong-2002-v2	Affymetrix	Krv	72	12582	3
Bhattacharjee-2001	Affymetrix	Pluća	203	12600	5
Dyrskjot-2003	Affymetrix	Bešika	40	7129	3
Golub-1999-v1	Affymetrix	Koštana srž	72	7129	2
Gordon-2002	Affymetrix	Pluća	181	12533	2
Laiho-2007	Affymetrix	Debelo crevo	37	22883	2
Nutt-2003-v1	Affymetrix	Mozak	50	12625	4
Nutt-2003-v2	Affymetrix	Mozak	28	12625	2
Pomeroy-2002-v1	Affymetrix	Mozak	34	7129	2
Pomeroy-2002-v2	Affymetrix	Mozak	42	7129	5
Ramaswamy-2001	Affymetrix	Više tkiva	190	16063	14
Shipp-2002-v1	Affymetrix	Krv	77	7129	2
Singh-2002	Affymetrix	Prostata	102	12600	2
Su-2001	Affymetrix	Više tkiva	174	12533	10
West-2001	Affymetrix	Pluća	49	7129	2
Yeoh-2002-v1	Affymetrix	Koštana srž	248	12625	2
Yeoh-2002-v2	Affymetrix	Koštana srž	248	12625	6
Alizadeh-2000-v1	D. Channel	Krv	42	4022	2
Alizadeh-2000-v2	D. Channel	Krv	62	4022	3
Alizadeh-2000-v3	D. Channel	Krv	62	4022	4
Bittner-2000	D. Channel	Koža	38	8067	2
Bredel-2005	D. Channel	Mozak	50	41472	3
Garber-2001	D. Channel	Grudi	66	24192	4
Lapointe-2004-v1	D. Channel	Prostata	69	42640	3
Lapointe-2004-v2	D. Channel	Prostata	110	42640	4
Liang-2005	D. Channel	Mozak	37	24192	3
Risinger-2003	D. Channel	Endometrijum	42	8872	4
Tomlins-2006-v1	D. Channel	Prostata	104	20000	5
Tomlins-2006-v2	D. Channel	Prostata	92	20000	4

6 Prošireni model meta-učenja za klasterovanje ekspresija gena

Najvažniji problem kod rada sistema meta učenja jeste veličina prostora problema kao i prostora algoritama (Smith-Miles 2008; Smith-Miles 2009). Na osnovu rezultata iz Sekcije 3 može se zaključiti da komponentni dizajn algoritama za klasterovanje ekspresija gena značajno proširuje prostor postojećih algoritama, ali takođe i da je jako teško identifikovati najbolji algoritam za konkretan skup podataka. Kao što je diskutovano ranije, jedan od uzroka za to je taj, što korisnik ne zna unapred koja interna mera evaluacije (koja se implicitno optimizuje tokom rada algoritma) je adekvatna za konkretan skup podataka. Na osnovu ovog razmatranja, prirodno je zaključiti da bi uključivanje komponentnih algoritama u prostor algoritama meta učenja moglo da poboljša performanse samih sistema meta-učenja (ne samo kod klasterovanja ekspresija gena, već kod svih OZP problema).

Sa druge strane, u Sekciji 4 je pokazano da je moguće da postoji korelacija između internih i eksternih mera evaluacije. To znači da se optimizacijom nekih internih mera evaluacije dobijaju kvalitetniji rezultati klasterovanja u odnosu na prave klastere ("zlatni standard"). Ovo razmatranje dovodi do zaključka da bi dodavanje internih mera izmerenih tokom rada algoritama klasterovanja u prostor meta-atributa takođe moglo da poboljša performanse sistema meta-učenja.

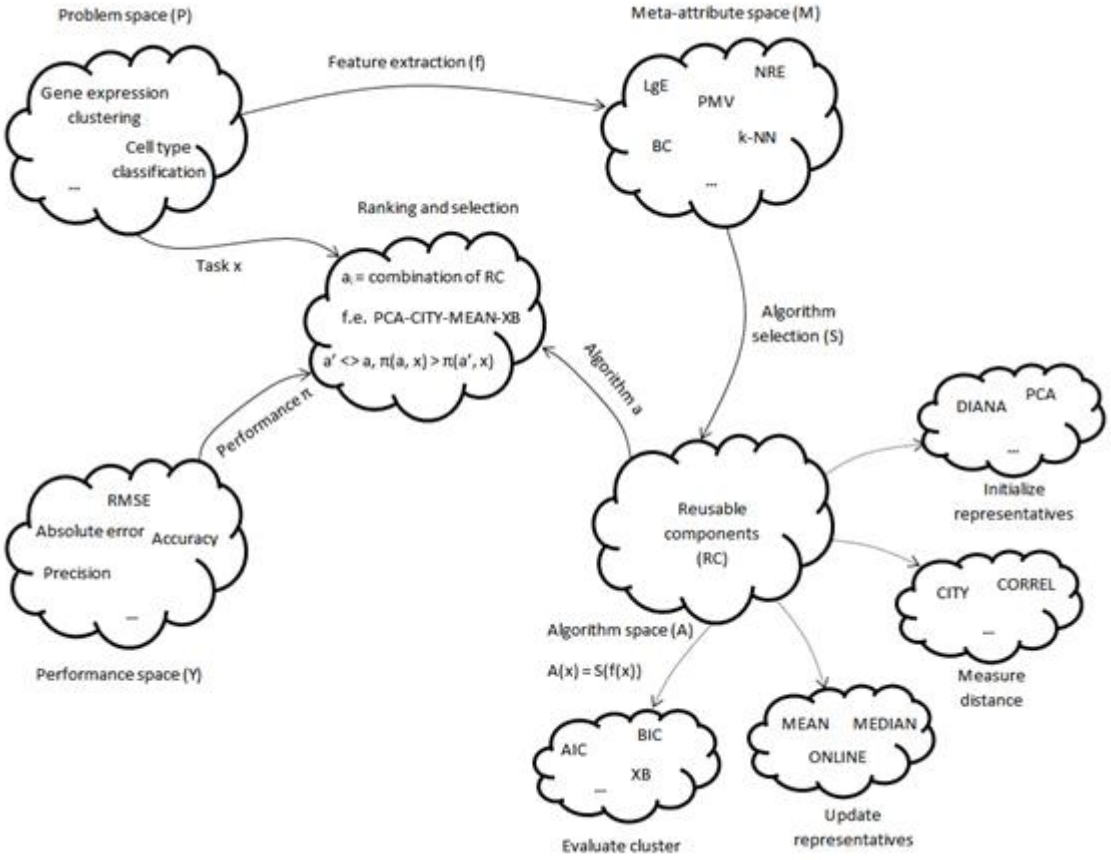
Na osnovu pomenutih zapažanja u (Radovanović 2013; Vukićević 2014) predložen je prošireni sistem meta učenja. Razlike proširenog sistema u odnosu na (Prudencio et al., 2011) su:

- Prostor algoritama je proširen sa komponentnim algoritmima klasterovanja.
- Prostor meta-atributa je proširen sa meta-atributima o algoritmima (komponentama). Na ovaj način je moguće identifikovati koje su to komponente ili pod-problemi (ne samo celi algoritmi) uticali na performanse algoritama (u Sekciji 4 je prikazano da neki pod-problemi daleko manje utiču na performanse). Na ovaj način je moguće i identifikovati koje to komponente treba bolje prilagoditi klasterovanju ekspresija gena.

- Prostor meta-atributa je proširen sa internim merama evaluacije.
- AMI eksterna mera je korišćena za predviđanje kvaliteta algoritama. Kao što je rečeno ova mera je pokazala najbolje performanse pri merenju kvaliteta klasterovanja, pogotovu u oblasti klasterovanja ekspresija gena (Vinh, 2010).
- Meta-model je baziran na algoritmima regresije. Time je izbegnut problem malih razlika u performansama, npr. ukoliko se performanse dva algoritma razlikuju tek na trećoj decimali, a koriste se rangovi (klasifikacioni meta-model), ovi algoritmi će zauzeti različite rangove iako je takva razlika zanemarljiva kada je u pitanju AMI indeks.
- Predložen je specifičan proces za kreiranje meta-modela, koji omogućava automatsku selekciju meta-atributa, kao i algoritama (implementacija ovog meta modela će detaljno biti opisana u sledećoj sekciji).

Dakle prošireni sistem meta-učenja koristi sve meta-atribute kao i (Prudencio et al., 2011), a pored njih dodaje i nove koje su opisani u prethodnom tekstu. Pored pomenutih specifičnosti, sistem funkcionisanja se malo razlikuje u odnosu na (Prudencio et al., 2011): u trening fazi se kreirara meta-model za predviđanje performansi algoritama, dok se u fazi primene ekstrahuju meta-podaci novog slučaja (skupa podataka) i na osnovu proširene baze meta-primera predviđaju performanse svakog od algoritama u bazi. Kao krajnji ishod dobija se lista algoritama koji su sortirani u opadajućem nizu prema AMI indeksu, a na analitičaru je da odredi koje algoritme i koliko njih će testirati nad realnim podacima. Predloženi prošireni sistem koji je definisan po uzoru na (Matijaš et al., 2013) prikazan je na slici 5.2.

Može se primetiti da je prostor problema na slici 5.2 (gornji levi oblak) podeljen na više tipova problema. To znači da se ovaj sistem jednostavno može proširiti i na druge bio-medicinske probleme samo dodavanjem specifičnih meta-atributa (kao što je kod klasterovanja ekspresija gena tip čipa).

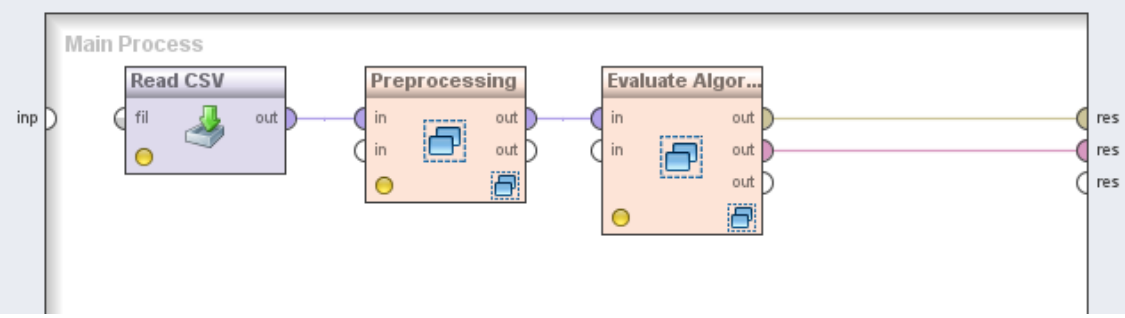


Slika 5.2. Proširenи sistem meta-učenja za klasterovanje bio-medicinskih podataka (Vukićević et al., 2014)

6.1.1 Osnovni proces evaluacije i selekcije meta-modela

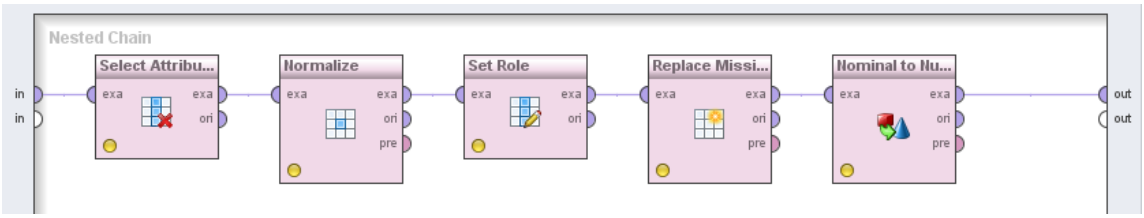
Kao što je navedeno u prethodnoj sekciji, prošireni model sistema-meta učenja ima integrisan proces za automatsku selekciju najboljeg algoritma regresije kao i selekciju atributa koji daju najbolje performanse meta-modela. Proces je implementiran u RapidMiner (Mierswa et al., 2006) softveru otvorenog koda za OZP i omogućava automatsko ažuriranje meta-modela i atributa pri dodavanju novih eksperimenata.

Podaci se čitaju sa diska uz pomoć *Read CSV* operatora, a nakon toga se vrši pred-procesiranje podataka, kao i evaluacija algoritama (Slika 5.3).



Slika 6.1. Glavni proces

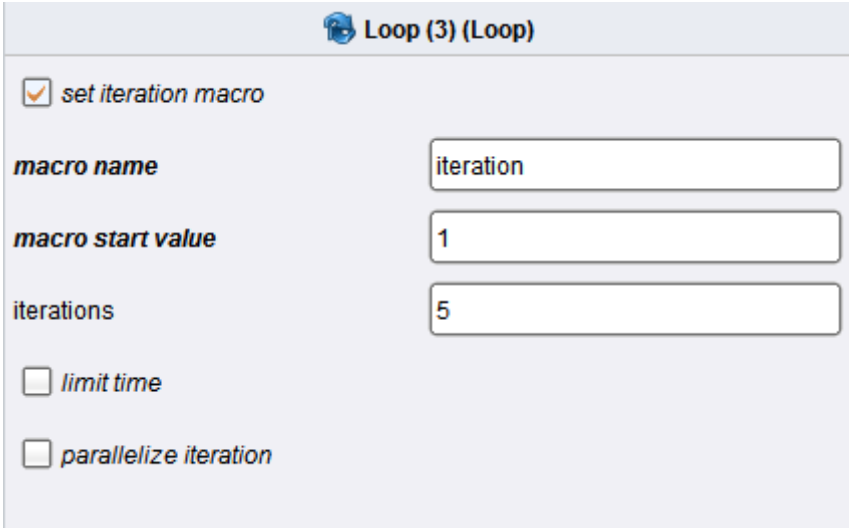
Unutar pod-procesa *Preprocessing* operatora, nalazi se niz operatora, koji pripremaju podatke za evaluaciju algoritama (Slika 5.4). Prvi operator je izbor atributa (eng. *Select Attributes*) čime se omogućava poređenje performansi algoritama sa različitim pod-skupovima meta-atributa. Drugi korak je normalizacija podataka. Normalizacija u ovoj fazi je neophodna zato što različite interne mere evaluacije su merene na različitim mernim skalama. Ovde se koristi max-min normalizacija. Uz pomoć trećeg (*Set role*) operatora AMI se obeležava kao izlazna (ciljna) promenljiva. Zatim se popunjavaju nedostajuće vrednosti (*Replace Missing Values*). Nedostajuće vrednosti se u ovom slučaju zamenjuju aritmetičkom sredinom. Kako većina algoritama regresije mogu da rade samo sa numeričkim prediktorima (ulaznim promenljivim), primenom tzv. d*ummy coding*-a vrši se prebacivanje kategoričke promenjive u numeričku. D*ummy coding* svaki kategorički atribut pretvara u onoliko binarnih atributa (koji uzimaju vrednost 0 ili 1) koliko ima različitih kategorija u tom atributu.



Slika 6.2. Pod-proces pred-procesiranja

Nakon završetka predprocesiranja podataka mogu se pokrenuti algoritmi koji će naučiti da predviđaju vrednosti AMI-ja na osnovu meta-atributa. Unutar podprocesa za evaluaciju algoritama nalazi se operator *Loop* koji omogućava automatsku iteraciju i evaluaciju nekoliko algoritama samo jednim pokretanjem procesa. Kako se u ovom eksperimentu evaluira 5 algoritama operator *Loop* se parametrizuje na sledeći način:

dodaje se makro koji označava redni broj iteracije *iteration* i postavlja se njegova početna i krajnja vrednost Slika (5.5).



Slika 6.3. Petlja u RapidMiner-u

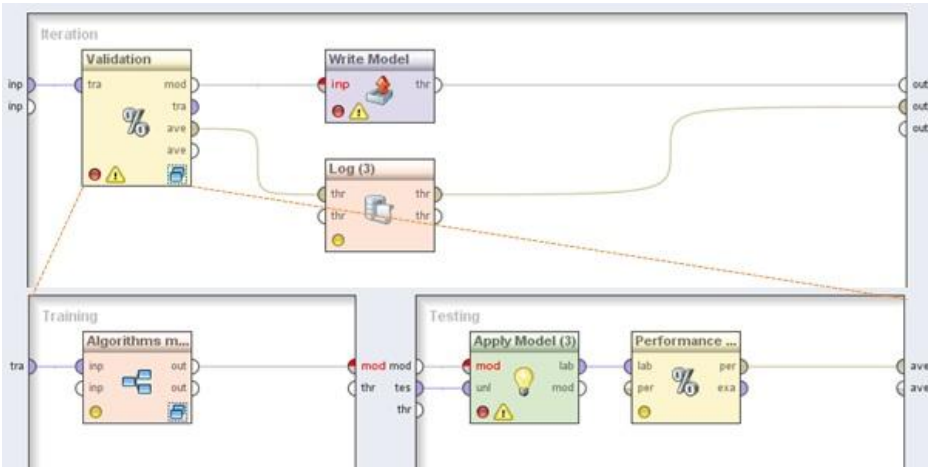
Unutar petlje, kao što se vidi na slici 5.6, se nalazi validacija. Validacija modela koja se vrši je kros-validacija, koja se izvršava deset puta. Kros-validacija predstavlja takav vid validacije gde se skup podataka deli na deset podskupova, tako da se treniranje modela vrši na devet podskupova, a testira na preostalom. Unutar validacije se nalaze dva polja. Jedno polje se odnosi na trening modela, a drugo na primenu modela i merenje performansi. Za merenje performansi koriste se dve mere. To su koren srednje kvadratne greške (RMSE) i apsolutna greška (AE). Koren srednje kvadratne greške predstavlja najčešće korišćenu meru evaluacije regresionih modela i računa se preko formule:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}}$$

gde je y_t vrednost koji je dobio model, a $\hat{y}_t$ stavrna vrednost.

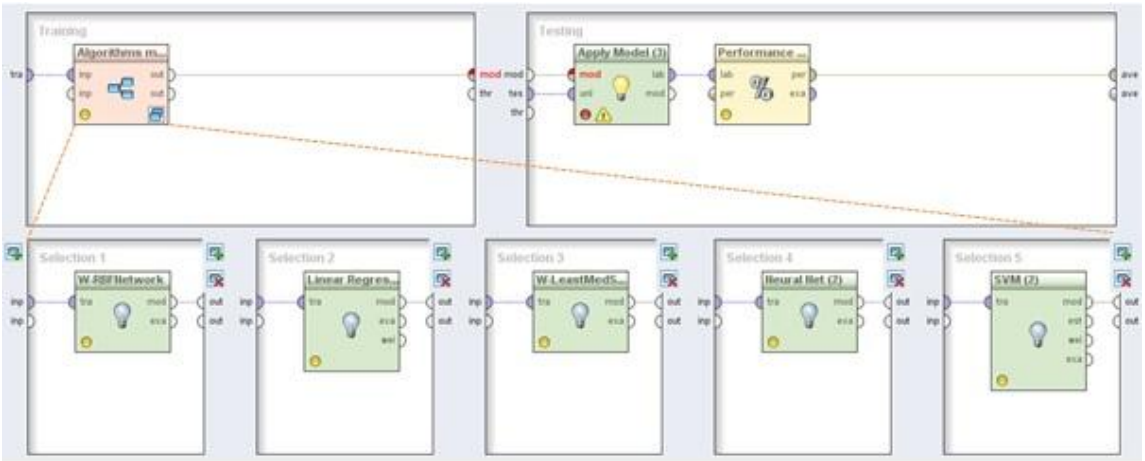
Apsolutna greška predstavlja apsolutnu vrednost razlike predviđene i stvarne vrednosti i dobija se preko formule:

$$AE = \sum_{t=1}^{n} |y_t - \hat{y}_t|$$



Slika 6.4. Proces unutar petlje

Nakon treniranja modela i izračunavanja vrednosti korena srednje kvadratne greške i apsolutne greške vrši se beleženje modela, pomoću operatora *Write Model*, i beleženje rezultata u datoteku na disku, pomoću operatora *Log*.



Slika 6.5. Postupak treniranja regresionih modela

Za postupak treniranja izabrani su sledeći algoritmi:

- Normalizovana Gausova mreža radijalne osnove – Korišćenjem k-means algoritma vrši klasterovanje, kojim se obezbeđuju funkcije. Nad tim funkcijama se vrši logistička regresija, u slučaju klasifikacije, odnosno linearna regresija u slučaju regresije.
- Linearna regresija – Predstavlja statistički algoritam koji modeluje odnos između zavisne promenjive (AMI) i objašnjavajućih promenjivih (meta-

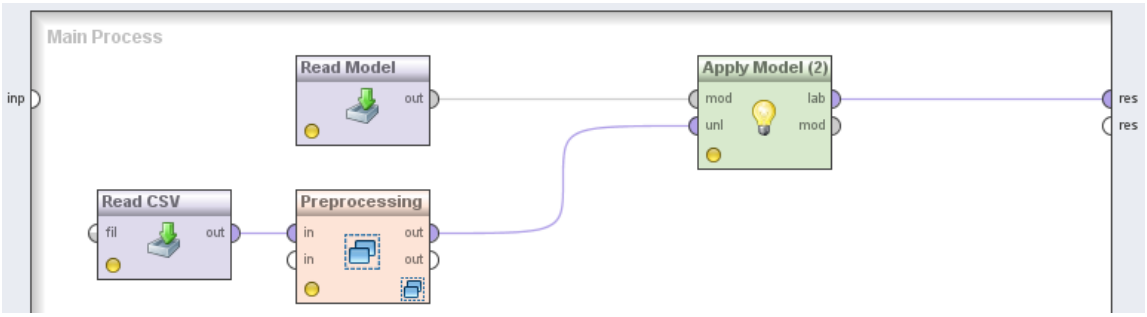
atributi), tako što postavlja liniju kroz oblast korišćenjem metode najmanjih kvadrata.

- Linearna regresija koja koristi metodu najmanje kvadratne medijane – Ovaj algoritam predstavlja nadogradnju linearne regresije. Jedina razlika je u načinu evaluacije greške. Umesto metode najmanjih kvadrata koristi metodu najmanje kvadratne medijane.
- Neuronska mreža – Predstavlja robustan algoritam mašinskog učenja, koji je sposoban za rešavanje najtežih problema, kako klasifikacije, tako i regresije. Koristi se višeslojni perceptron sa brojem neurona u skrivenom sloju jednakom:

$$\frac{(broj\ atributa) + (broj\ klasa)}{2} + 1$$

- Mašine sa vektorima podrške –Takođe, robustan algoritam mašinskog učenja koji optimizuje razdaljinu linije između različitih klasa posmatranog problema tako da razdaljina bude što šira. Nelinearne zadatke rešava korišćenjem kernela koji mapira ulazne atribute u višedimenzioni prostor. Može se koristiti i za probleme klasifikacije i za probleme regresije.

Sledeći korak predstavlja primenu modela. Za taj problem napravljen je novi RapidMiner proces (slika 5.8). Proces se sastoji iz nekoliko opeartora. Postupak učitavanja i predprocesiranja podataka je identičan. Drugi deo je čitanje modela. Kako smo u prethodnom koraku sačuvali model na disku, sada čitamo taj model koristeći operator *Read Model*. Nakon toga, podaci i model se kombinuju i operatoru *Apply Model*. Kao rezultat dobijaju se vrednosti AMI i predviđena vrednost za AMI.



Slika 6.6. Primena modela

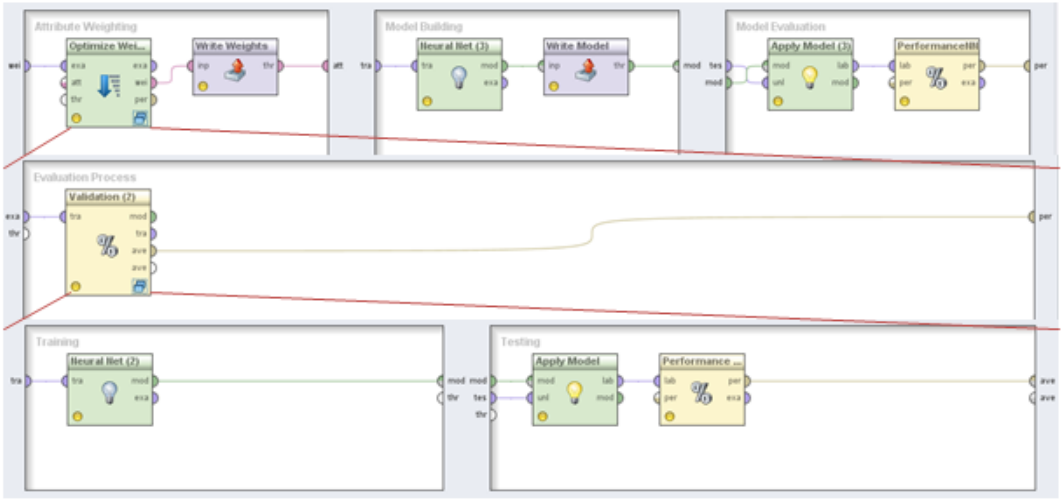
6.1.2 Proces za automatsku selekciju meta-atributa

Osnovni proces prikazan u prethodnoj sekciji je moguće dodatno unaprediti. Naime, prilikom treniranja modela uzeti su u obzir svi atributi. Međutim, ne doprinose svi atributi podjednako rešenju. Neki od njih koji su međusobno apsolutno visoko korelisani donose istu informaciju i mogu da pogoršaju performanse algoritma. Stoga, razvijen je drugi RapidMiner proces koji za cilj ima optimizaciju atributa.

Optimizacija atributa se može raditi na nekoliko načina. Prvi način je optimizacija unapred. Ovim tehnikom se kreće od jednog atributa koji ima najveću korelaciju sa ciljnim atributom. Nakon toga se iterativno dodaju atributi, tako da na početku imaju nisku težinu i, opet, iterativno se težina povećava. Optimizacija se prekida kada se dogodi korak gde nema optimizacije funkcije cilja ili ako je ona manja od zadate. Drugi način je optimizacija unazad. Postupak optimizacije kreće sa celim skupom ulaznih atributa. Iterativno se izbacuju atributi koji najmanje doprinose izlaznom atributu. Onog trenutka kada se vrednost modela smanji optimizacija se prekida. Ova dva načina predstavljaju model primene sirove snage (eng. b*rute force*) kao optimizacije. Zbog te osobine mogu da traju poprilično dugo. Sledeća dva načina predstavljaju primenu metaheuristika u optimizaciji. Metaheuristike su poznate po tome da daju dobra, ali ne optimalna rešenja, za kraće vreme. Upravo zbog toga biće korišćeni u ovom procesu. Prvi predstavlja korišćenje genetskih algoritama. Na taj način se kreira familija rešenja koja se naziva populacija. Rešenja se ukrštaju po principu Darvinove teorije evolucije, čime se dolazi do boljih rešenja. Kriterijum zaustavljanja je broj iteracija. Druga metaheuristička tehnika je zasnovana na roju čestica. Kao takva, vrlo je slična

genetskom algoritmu. Jedina razlika je u tome što imitiraju način leta ptica selica ili grupisanje riba kada ih napadne grabljivac.

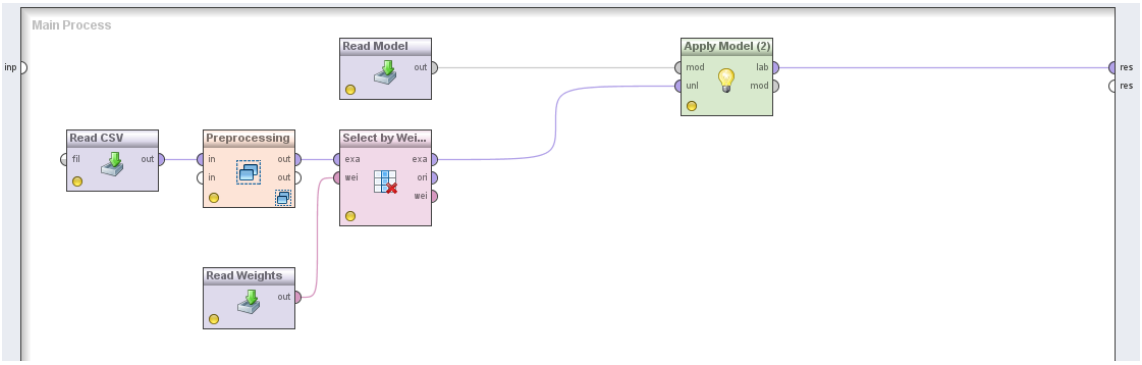
Kako ova optimizacija oduzima dosta vremena eksperiment je sproveden samo sa neuronskom mrežom. Proces kreće isto kao i prost. Prvo se učitavaju podaci, zatim se vrši predprocesiranje podataka. Unutar predprocesiranja se vrši izbor atributa, normalizacija, postavljanje ciljnog atributa, zamena nedostajućih vrednosti i pretvaranje kategoričkih promenjivih u numeričke. Postupak evaluacije algortima je naizgled isti, ali postoje razlike. Naime, umesto kros-validacije koristi se jednostavna validacija gde se 90% uzorka koristi za treniranje, a 10% za testiranje modela.



Slika 6.9. Evaluacija modela sa optimizacijom atributa

Na slici 5.10 se nalazi RapidMiner proces koji koristi optimizaciju atributa. Pored dela za treniranje i testiranje modela, postoji i deo za optimizaciju težina atributa. U tom delu korišćen je operator *Optimize Weight* koji se razlikuje za tip optimizacije, tj. da li se koristi genetski algoritam ili algoritam roja čestica. Unutar operatora optimizacije nalazi se unutršnja validacija. Kao i spoljašnja, i unutrašnja validacija je prosta validacia koja koristi 70% uzorka za treniranja i preostalih 30% za testiranje. Unutar validacije se nalaze dva polja. Jedno je za treniranje, i u njemu se nalazi neuronska mreža. Dok se u drugom delu nalaze operatori koji primenjuju trenirani model i mere njegove performanse. Kada se optimizacija težina atributa odradi prelazi se na treniranje modela. Za opisani proces neophodno je prilagoditi proces za primenu modela. Naime pored učitavanja i predprocesiranja podataka i učitavanja modela potrebno je učitati i težine

atributa. Međutim, nije dovoljno samo učitati težine, već je potrebno i dodati operator koji će te težine primeniti. Taj operator se zove *Select by Weight*. On kao ulazne parametre prima skup podataka i težine atributa. Nakon otežavanja atributa postupak je identičan prethodnom.



Slika 6.10. Proces za primenu modela sa optimizacijom težina atributa

6.1.3 Eksperimentalni rezultati

Kao što je opisano u Sekciji 5.2, u ovom radu se predlaže prošireni model meta-učenja, koji uključuje komponente algoritama i interne mere evaluacije, kao dodatne meta-atribute. U narednom tekstu, biće opisani eksperimenti koji pokazuju da dodavanje ovih meta atributa može uticati na performanse meta-modela u sistemu meta-učenja.

Ovaj eksperiment će se sprovesti u tri koraka, svaki sa sedam kombinacija meta-atributa:

- jednostavan proces (bez optimizacije bazirane na selekciji atributa).
- optimizacija performansi meta-modela, uz pomoć određivanja težina meta-atributa korišćenjem genetskog algoritma,
- optimizacija performansi meta-modela, uz pomoć određivanja težina meta-atributa korišćenjem.

Kombinacije meta-atributa koje će biti korišćene u svakom koraku su sledeće:

- Meta-atributi o skupovima podataka
- Interne mere;
- Komponente;

- Meta-atributi o skupovima podataka i interne mere;
- Meta-atributi o skupovima podataka i komponente;
- Interne mere i komponente;
- Meta-atributi o skupovima podataka, interne mere i komponente.

Rezultati dobijeni nakon sprovođenja prvog eksperimenta (jednostavan proces nad meta-atributima o skupovima podataka) se mogu videti u tabeli 5.1. Jasno se vidi da rezultati dobijeni korišćenjem neuronske mreže postižu najbolje rezultate. Koren srednje kvadratne greške (RMSE) iznosi 0.1185, odnosno apsolutna greška je 0.0866 što znači da su sve greške kad se saberu, zanemarujući znak greške, iznosi 0.0866. Pored neuronskih mreža, dobre rezultate dala je i linearna regresija i njena srednja kvadratna greška iznosi 0.1320.

Tabela 6.1 – Rezultati regresionih algoritama (nad meta-atributima o skupovima podataka)

Algoritam	**RMSE**	**AE**
Normalizovana Gausova mreža radijalne osnove	0.1571	0.1181
Linearna regresija	0.1320	0.0989
Linearna regresija (najmanja kvadratna medijana)	0.1671	0.1025
Neuronska mreža	**0.1185**	**0.0866**
Mašine sa vektorima podrške	0.1801	0.1359

Kao što je već napomenuto, zbog kompleksnosti modela korišćena je samo neuronska mreža. Bitno je napomenuti da se težine atributa i sam model čuvaju na disku. Nakon završetka spoljne validacije, na disku se beleže rezultati.

Težine atributa, dobijenih sprovođenjem opisanog procesa se nalaze u tabeli 5.2. Zanimljivo je da težine deluju potpuno nezavisne. Ukoliko pogledamo meta-atribut SC_10, genetski algoritam ga je otežao sa 0.9078, dok ga je optimizacija roja čestica izbacila iz analize. Takođe, meta-atribut LgREA je kod genetskog algoritma otežan sa 0.0637, a optimizacija roja čestica ga smatra izuzetno važnim (otežao je taj atribut na 1.9768).

Tabela 6.2. Težine atributa

Atribut	Težina (Genetski algoritam)	Težina (Roj čestica)
chip = 0	0.6463	0.8424
chip = 1	0.7936	0.9299
LgE	0.7089	0.3163
LgREA	0.0637	1.9768
PMV	0.7649	1
MN	1	0.6572
SK	0.5080	1
PFA	0.9716	1
PO	0.1314	0.1056
NRE	0.8746	0.7670
SC_10	0.9078	0
SC_15	0.7499	0.5833
BC	0	0.6752
kNN	0	0.6572

Međutim, postavlja se pitanje da li su se performanse algoritma popravile, i ako jesu za koliko su se popravile. Rezultati su prikazani u tabeli 5.3. Kako je koren srednje kvadratne greške bio 0.1185, a apsolutna greška 0.0866, očigledno je došlo do poboljšanja. Malo bolje rezultate prikazuje optimizacija težina atributa pomoću genetskog algoritma.

Tabela 6.3 – Rezultati optimizacije težina atributa

Tip optimizacije	RMSE	AE
Genetski algoritam	0.1095	0.0759
Optimizacija roja čestica	0.1112	0.0789

Zbog vremenske i računske zahtevnosti kompleksnijih procesa, dalji eksperimenti su vršeni sa tri regresiona algoritma koja su se pokazala najbolje u prethodnom eksperimentu: linearna regresija, neuronske mreže i mašine sa vektorima podrške.

Nakon sprovođenja prostog procesa, nad internim merama, dobijeni su rezultati prikazani u tabeli 5.4. Rezultati prikazuju da interne mere imaju slabiju predikcionu moć od meta-atributa.

Tabela 6.4. Rezultati prostog procesa za interne mere evaluacije

Algoritam	RMSE	AE
Linearna regresija	0.1521	0.1142
Neuronske mreže	0.1535	0.1270
Mašine sa vektorima podrške	0.1574	0.1190

U slučaju optimizacije težina atributa, kao i u prethodnom delu, korišćena je samo neuronska mreža. Rezultati su prikazani u tabeli 5.5. Primenom optimizacije težina se došlo do gorih rešenja, čime se pokazuje da interne mere same nemaju dovoljnu predikcionu moć za posmatrani problem.

Tabela 6.5. Rezultati procesa sa optimizacijom težina atributa za interne mere evaluacije

Optimizacija	RMSE	AE
Genetski algoritam	0.1594	0.1073
Optimizacija roja čestica	0.1583	0.1031

Sledeća kombinacija je korišćenje samo komponenti za predviđanje kvaliteta klasterovanja. U poglavlju 5.1. pokazano je da postoji razlika između komponenti i tipa normalizacije u odnosu na AMI. Rezultati su prikazani u tabeli 5.6. Kao i kod internih mera, rezultati su gori nego kod meta-atributa. Najbolji rezultat prikazala je linearna regresija, a njen rezultat je gotovo identičan (razlikuje se na četvrtoj decimali za RMSE) u odnosu na model dobijen za interne mere.

Tabela 6.6. Rezultati prostog procesa za komponente

Algoritam	RMSE	AE
Linearna regresija	0.1527	0.1133
Neuronske mreže	0.1743	0.1534
Mašine sa vektorima podrške	0.1943	0.1747

U slučaju optimizacije težina atributa, korišćena je neuronska mreža. Rezultati su prikazani u tabeli 5.7. Nakon optimizacije rezultati su se popravili na drugoj decimali. Međutim, greška je i dalje velika, što dovodi do zaključka da ni upotreba samo komponenti nije dovoljno moćna za predviđanje kvaliteta klasterovanja.

Tabela 6.7. Rezultati procesa sa optimizacijom težina atributa za komponente

Optimizacija	RMSE	AE
Genetski algoritam	0.1583	0.1321
Optimizacija roja čestica	0.1536	0.1087

Kako ni interne mere ni komponente (izolovane među sobom i od ostalih meta-atributa) ne mogu dovoljno dobro predviđaju kvalitet klasterovanja, pretpostavka je da će kombinacija doprineti boljim rezultatima. Prvo su testirani meta-atributi skupova podataka i internih mera. Nakon sprovođenja eksperimenta dobijeni su rezultati koji su prikazani u tabeli 5.8. Kao što je i pretpostavljeno, rezultati su se popravili. Koren srednje kvadratne greške i apsolutna greška su niže na nivou druge decimale, što znači

da se dodavanjem novih atributa dobija na predikcionoj moći. U ovom slučaju najmanju grešku ima neuronska mreža, nižu nego neuronska mreža koja koristi samo meta-atribute uz optimizaciju težina atributa.

Tabela 6.8. Rezultati prostog procesa za meta-atribute i interne mere

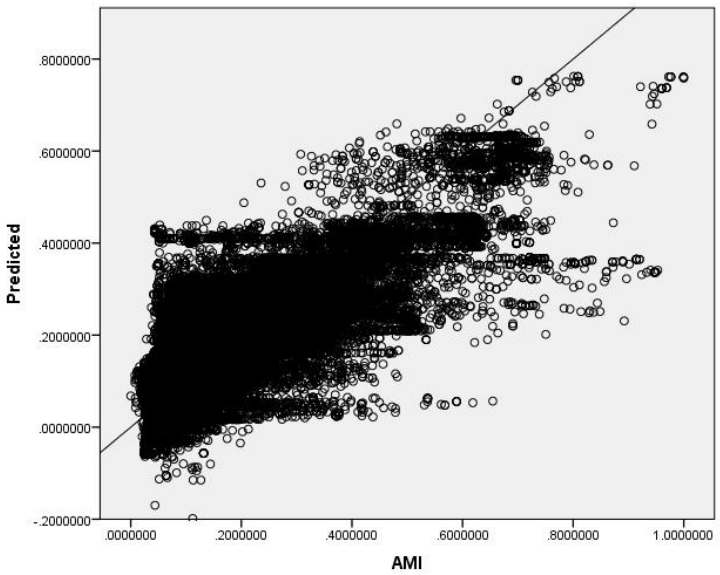
Algoritam	RMSE	AE
Linearna regresija	0.1266	0.0937
Neuronske mreže	0.1047	0.0691
Mašine sa vektorima podrške	0.1573	0.1190

Korišćenjem optimizacije težina atributa se dobijeni rezultati još više unapređuju, što se i vidi u tabeli 5.9. Koren srednje kvadratne greške neuronske mreže nakon optimizacije težina atributa pada ispod 0.1, a apsolutna greška ispod 0.06 što znači da greši, bez obzira na stranu greške samo 0.06.

Tabela 6.9. Rezultati procesa sa optimizacijom težina atributa za meta-atribute i interne mere

Optimizacija	RMSE	AE
Genetski algoritam	0.0899	0.0616
Optimizacija roja čestica	0.0882	0.0599

Dakle, može se zaključiti da se dodavanjem novih atributa dobija na prediktivnoj moći. Rezultati su vidljivi i na odnosu stvarne i predviđene vrednosti za AMI kod procesa sa optimizacijom težina atributa korišćenjem roja čestica za meta-atribute i interne mere (slika 5.12).



Slika 6.7. Odnos stvarne i predviđene vrednosti za AMI kod procesa sa optimizacijom težina atributa za meta-atribute i interne mere

Kako je pokazano da proširivanje skupa meta-atributa sa internim merama evaluacije utiče na performanse meta modela, u sledećem eksperimentu se proverava da li to važi i za komponente algoritama. Rezultati su prikazani u tabeli 5.10. Iako su mašine sa vektorima podrške pokazale lošije performanse nego u prethodnim eksperimentima, neuronske mreže su pokazale, najbolje rezultate u odnosu na sve prethodne eksperimente.

Tabela 6.10. Rezultati prostog procesa za meta-atribute i komponente

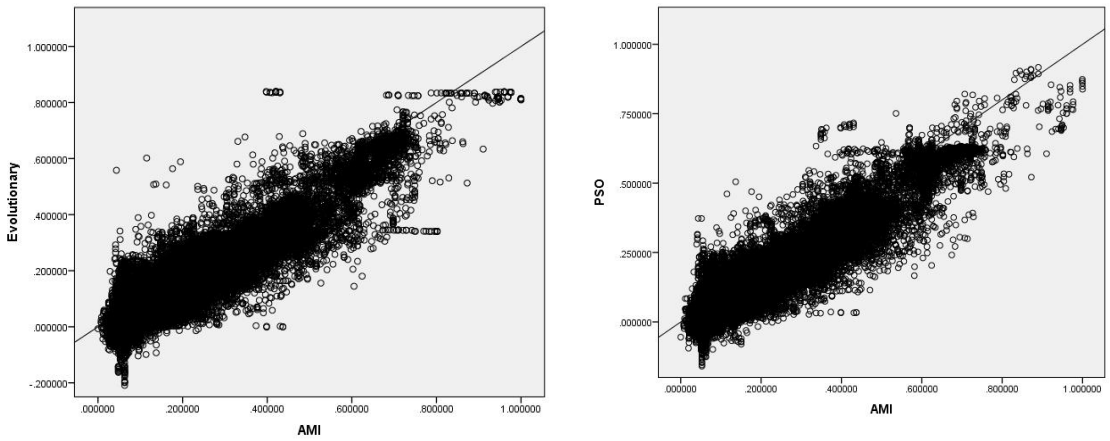
Algoritam	RMSE	AE
Linearna regresija	0.1256	0.0935
Neuronske mreže	0.0779	0.0541
Mašine sa vektorima podrške	0.2559	0.2365

Nakon optimizacije težina atributa rezultati (prikazani u tabeli 5.11) su još bolji. Korišćenjem optimizacije roja čestica apsolutna greška iznosi tačno 0.05.

Tabela 6.11. Rezultati procesa sa optimizacijom težina atributa za meta-atribute i komponente

Optimizacija	RMSE	AE
Genetski algoritam	0.0741	0.0579
Optimizacija roja čestica	0.0680	0.0500

Međutim, interesantniji su podaci prikazani na slici 5.13. Sa leve strane se nalazi odnos stvarne i predviđene vrednosti za AMI kod procesa sa optimizacijom težina atributa za meta-atribute i komponente kada se koristio genetski algoritam, dok se sa desne strane nalazi odnos stvarne i predviđene vrednosti kada se koristila optimizacija roja čestica. Na x osi se nalazi stvarna vrednost, dok se na y osi nalazi predviđena vrednost. Jasno se vidi da se gotovo svaki uzorak nalazi na dijagonali, tj. da se vrednosti skoro preklapaju.



Slika 6.8. Odnos stvarne i predviđene vrednosti za AMI kod procesa sa optimizacijom težina atributa za meta-atribute i komponente

Sledeća kombinacija koju treba isprobati je korišćenja atributa internih mera i komponenti. Pretpostavka je da će ova kombinacija imati najnižu predikcionu moć, jer komponente i interne mere same po sebi nemaju moć (što je i pokazano u ovom poglavlju). Rezultati, dobijeni sprovođenjem eksperimenta, su prikazani u tabeli 5.12.

Tabela 6.12. Rezultati prostog procesa za interne mere i komponente

Algoritam	RMSE	AE
Linearna regresija	0.1482	0.1107
Neuronske mreže	0.1732	0.1197
Mašine sa vektorima podrške	0.1574	0.1190

Pretpostavka je bila tačna. Iako se najbolji rezultat popravio, tj. linearna regresija je pokazala bolje rezultate nego kod internih mera i komponenti pojedinačno i mašine sa vektorima podrške pokazale identičan rezultat kao i kad se koriste samo interne mere, rezultati nisu na zadovoljavajućem nivou. Čak ni nakon optimizacije težina neuronske mreže, iako boljeg rezultata, ne pokazuju dovoljno dobre rezultate.

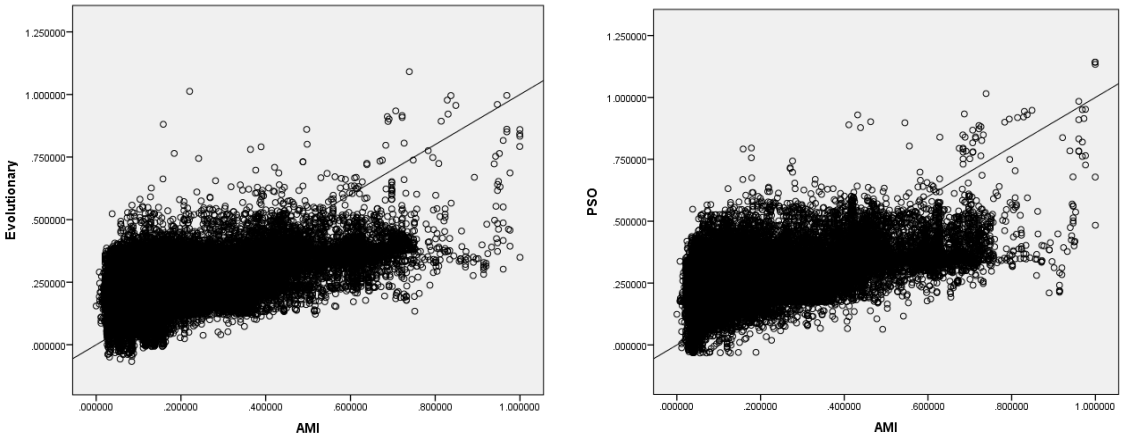
Tabela 6.13. Rezultati procesa sa optimizacijom težina atributa za interne mere i komponente

Optimizacija	RMSE	AE
Genetski algoritam	0.1516	0.1268
Optimizacija roja čestica	0.1711	0.1476

Posmatrajući sliku 5.14, koja predstavlja odnos stvarne i predviđene vrednosti za AMI kod procesa sa primenom genetskog algoritma zaotežavanje atributa za meta-atribute i komponente sa leve strane i primenu optimizacije težina atributa sa rojevima čestica sa desne strane, vidimo da je greška velika. Međutim, predviđene vrednosti gotovo da

pripadaju intervalu [0, 0.5], gotovo da ne postoje vrednosti iznad 0.5. Takođe, postoje slučajevi gde je predviđena vrednost AMI-ja iznad jedan.

Bez obzira na grešku, interne mere i komponente niti same, niti zajedno ne mogu biti dobri prilikom procene kvaliteta klasterovanja. Međutim, ukoliko se ukombinuju sa meta-atributima rezultati se značajno poboljšavaju. Stoga, sledeći korak u eksperimentu je pravljenje modela koji će koristiti i meta-atribute i interne mere klastera i komponente.



Slika 6.9. Odnos stvarne i predviđene vrednosti za AMI kod procesa sa optimizacijom težina atributa za interne mere i komponente

Konačno, u tabeli 5.14 su prikazani rezultati meta-modela dobijenog nad svim raspoloživim meta-podacima (meta-atributima o skupovima podataka, internih mera evaluacije i komponenti).

Tabela 6.14. Rezultati prostog procesa za meta-atribute, interne mere i komponente

Algoritam	RMSE	AE
Linearna regresija	0.1211	0.0897
Neuronske mreže	0.0681	0.0484
Mašine sa vektorima podrške	0.1574	0.1190

Očekivano, algoritmi su prikazali bolje ili iste performanse. Od ispitanih algoritama najbolje rezultate ima neuronska mreža, kod koje koren srednje kvadratne greške iznosi 0.0681, a apsolutna greška 0.0484. Nakon optimizacije težina atributa (pogledati tabelu 5.15) greške su još niže. Poboljšanja su na trećoj decimali. Međutim, optimizacijom roja čestica dobijena je takva kombinacija težina atributa da je apsolutna greška 0.0411, tj.

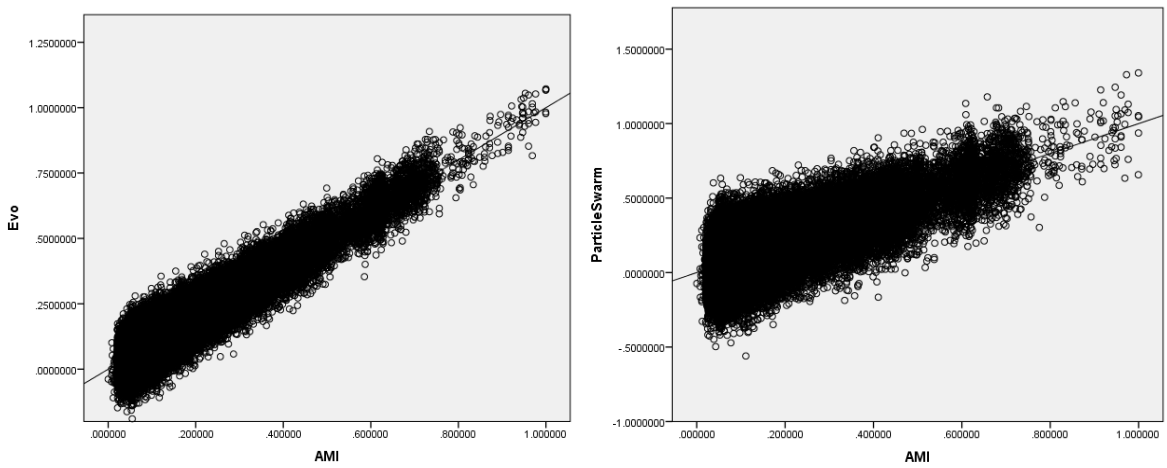
greška iznosi 4.11%. Sa druge strane, genetskim algoritmom dobijena je kombinacija težina atributa koja je dovela do apsolutne greške od 0.0356, tj. greške od 3.56%.

Tabela 6.15. Rezultati procesa sa optimizacijom težina atributa za meta-atribute, interne mere i komponente

Optimizacija	RMSE	AE
Genetski algoritam	0.0517	0.0356
Optimizacija roja čestica	0.0565	0.0411

Gledajući rezultate dobijene nakon optimizacije težina atributa, prikazane na slici 5.15 može se zaključiti da dodavanjem atributa u problem klasterovanja genskih ekspresija dolazi se do preciznijeg rešenja. Sa leve strane se nalaze rezultati nakon primene optimizacije težina atributa preko genetskog algoritma, dok se sa desne strane nalaze rezultati nakon primene optimizacije težina atributa nakon primene optimizacije rojeva čestica.

Primenom optimizacije rojeva čestica izgleda da je problem previše dobro istreniran, te greška na skupu podataka deluje veća nego što je prvobitno prikazano. Sa druge strane optimizacija preko genetskog algoritma izgleda da je dobro uhvatila šablone u problemu i uspešno ih primenila.



Slika 6.10. Odnos stvarne i predviđene vrednosti za AMI kod procesa sa optimizacijom težina atributa za meta-atribute, interne mere i komponente

Rezultati evaluacije proširenog modela meta-učenja potvrđuju naučnu zasnovanost i opšte hipoteze ove disertacije:

Opšta hipoteza: Moguće je projektovati algoritme klasterovanja ekspresije gena, koji će biti bolji od postojećih rešenja i moguće je identifikovati najbolji algoritam za konkretan skup podataka.

7 Zaključak

Primena algoritama klasterovanja na podatke o ekspresijama gena potencijalno ima veliki uticaj na razumevanje genetskog koda, identifikaciju pravilnosti u genskim ekspresijama, unapređenje dijagnostičkih sistema i smanjenje troškova dijagnostike i lečenja. Specifičnost ovih podataka u smislu njihove velike dimenzionalnosti i malog uzorka većini algoritama klasterovanja onemogućava kreiranje kvalitetnih modela. Osnovna ideja ovog rada je bila da se predloži metodologija koja će potencijalno dati rešenje za ovaj problem tako što će ponuditi okvir za brzi i kolaborativni razvoj algoritma klasterovanja, kao i različite modele za selekciju algoritama koji su najbolje prilagođeni podacima o ekspresiji gena.

7.1 Ostvareni doprinos

Kontinuirano istraživanje u oblasti razvoja komponentnih algoritama klasterovanja i njihove primene na podacima o ekspresijama gena. Najznačajniji doprinos je predlog originalne metodologije za projektovanje, evaluaciju i selekciju algoritama za klasterovanje podataka o ekspresiji gena koja je zasnovana na proširenom sistemu meta-učenja, koji integriše osobine algoritama (komponente)i interne mere evaluacije kao meta-atribute. Ovaj sistem takođe uključuje proces za automatsku selekciju meta-modela baziran na automatskoj selekciji meta-atributa čime omogućava jednostavno ažuriranje i prilagođavanje novim skupovima podataka, algoritmima i meta podacima.

Pored toga, naučni doprinos rada ogleda se i u sledećem:

- Pregled savremenih modela koji mogu da se koriste za klasterovanje podataka o ekspresijama gena.
- Predlog arhitekture za kolaborativni dizajn komponentnih algoritama klasterovanja
- Predlog generičkih algoritama klasterovanja (divizioni i hijerarhijsko-divizioni)

- Primena i detaljna analiza komponentnih algoritama klasterovanja na realnim podacima kao i poređenje sa drugim algoritmima iz literature.
- Predlog metoda za identifikaciju komponenata koje učestvuju u dizajnu algoritama sa dobrim performansama i koje bi trebalo razmatrati pri dizajnu novih algoritama.
- Predlog modela meta-učenja prilagođenog algoritmima za klasterovanje podataka o ekspresiji gena.

7.2 Pravci daljeg istraživanja

Kako je preduslov za dizajn i implementaciju sistema meta-učenja sa dobrim performansama (Smith-Miles, 2008), veliki prostor raspoloživih algoritama, kao jedan od pravaca budućeg rada biće proširenje repozitorijuma komponenti kao i dodavanje novih pod-problema u postojeći generički algoritam klasterovanja.

U sekciji 4, pokazano je da komponentni algoritmi mogu da daju bolje rezultate u odnosu na konsenzus algoritme. Obzirom da su (Monti et al., 2003; Yu et al., 2007) pokazali da konsenzus algoritmi generalno daju stabilnija i kvalitetnija rešenja u odnosu na pojedinačne algoritme, uključivanje komponentnih algoritama u konsenzus šeme bi moglo da da još bolje performanse.

Još jedan pravac daljeg razvoja je implementacija meta-heurističkih algoritama za pretragu prostora mogućih komponentnih algoritama klasterovanja. Ovakav pristup je već uspešno primenjen u oblasti komponentnih stabala odlučivanja (Jovanović et al., 2011; Jovanović et al., 2014). Ovakav pristup omogućava automatsku selekciju najboljeg algoritma za konkretne podatke, ali za razliku od pristupa meta-učenja, vrši se evaluacija mnogo više algoritama klasterovanja. Dakle, meta-učenje daje procenu performansi i ranga algoritama na osnovu meta-podataka, dok meta-heuristički pristup određuje tačne performanse na novom skupu podataka, ali zbog toga je vremenski daleko zahtevniji pa je takav pristup preporučljivo koristiti kod skupova podataka manjih dimenzija. Iz ovog razmatranja može se zaključiti da bi kombinacija ova dva

pristupa mogla da da dobre rezultate klasterovanja uz smanjenje vremena evaluacija. Integracija ova dva pristupa će takođe biti predmet budućeg istraživanja.

8 Literatura

1 Abeel T., de Peer Y.V., Saeys Y. (2009) Java-ML: A Machine Learning Library, *Journal of Machine Learning Research* 10, pp.931-934.

2 Achtert E., Kriegel H., Zimek A. (2008) ELKI: A Software System for Evaluation of Subspace Clustering Algorithms, Proc. of *20th International Conference on Scientific and Statistical Database Management*, pp.580-585.

3 Ahmad A, Dey L (2007) A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*. doi:10.1016/j.datak.2007.03.016

4 Akaike H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control*. doi: 10.1109/TAC.1974.1100705

5 Alizadeh A. A., Eisen M. B., Davis R. E., Ma, C., Lossos, I. S., Rosenwald, A et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403, pp.503–511.

6 Altun O., Dursunoglu N., Amasyali M., Clustering Application Benchmark (2006), *IEEE International Symposium on Workload Characterization*, pp.178-181.

7 Andreopoulos B, An A, Wang X et al. (2009) A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics* 10(3) pp.297-314

8 Ankerst M, Breunig M, Kriegel H, et al. (1999) OPTICS: Ordering points to identify the clustering structure. In: Proceedings of the *ACM SIGMOD'99 Int. Conf. on Management of Data*. Philadelphia, USA, pp.49–60.

9 Arthur D, Vassilvitskii S (2007) K-means ++: The Advantages of Careful Seeding. In Proceedings of the *18th annual ACM-SIAM symposium on Discrete algorithms (SODA '07). Society for Industrial and Applied Mathematics*, Philadelphia, PA, USA, 1027-1035

10 Au N.H., Cheaug M., Huntsman D.G., Yorida A., Coldman A., and Elliott W.M. (2004). Evaluation of immunohistochemical markers in non-small cell lung cancer by unsupervised hierarchical clustering analysis: a tissue microarray study of 284 cases and 18 markers. *J Pathol* 204, pp.101–109.

11 M. M. Astrahan, Speech analysis by clustering, or the hyperphome method, Stanford Artificial Intelligence Project Memorandum AIM-124, Stanford, CA: Stanford University, 1970.

12 Ayadi W, Elloumi M, Hao J.K (2012) BicFinder: a biclustering algorithm for microarray data analysis. *Knowledge and Information Systems* 30, 341-358. doi: 10.1007/s10115-011-0383-7

13 Bagirov A. (2008) Modified global kK-means algorithm for minimum sum-of-squares clustering problems, Pattern Recognition 41(10), pp.3192-3199.

14 Balachandran, V., P, D., Khemani, D. (2011) Interpretable and reconfigurable clustering of document datasets by deriving word-based rules. *Knowledge and Information Systems*, doi: 10.1007/s10115-011-0446-9

15 Baralis E., Bruno G., Flori, A. (2011) Measuring gene similarity by means of the classification distance. *Knowledge and Information Systems* 29, pp.81-101. doi:10.1007/s10115-010-0374-0

16 Baya AE and Granitto PM (2011) Clustering gene expression data with a penalized graph-based metric. *BMC bioinformatics* 12, pp.1-18

17 Bezdek,J.C. (1981) Pattern Recognition With Fuzzy Objective Function Algorithms. *Plenum Press, New York*.

18 Belacel N, Wang Q, Cuperlovic-Culf M (2006) Clustering methods for microarray gene expression data. *OMICS: A Journal of Integrative Biology* 10(4) pp.507-531. doi:10.1089/omi.2006.10.507

19 Belal M. and Daoud A. (2005) A new algorithm for cluster initialization, In Proc. *World Academy of Science, Engineering and Technology*, pp.74–76.

20 Ben-Hur A., Elisseeff A. and Guyon I. (2002) Stability based method for discovering structure in clustered data. *Pac Symp Biocomputing* 7, pp.6-17.

21 Berthold M., Cebron N., Dill F., Di Fatta G., Gabriel T., Georg F., Meinl T., Ohl P., Sieb C., and Wiswedel B. (2006) KNIME: The Konstanz Information Miner, Proc. of the Workshop on Multi-Agent Systems and Simulation MAS&S, *4th Annual Industrial Simulation Conference* (ISC), Palermo, Italy, Jun. 2006, pp.58–61.

22 Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pp.25-71. Springer Berlin Heidelberg.

23 Bock H. H. (2007) Clustering methods: A history of the K-means algorithm, in: P. Brito et al. (eds.): Selected contributions in data analysis and classification, Springer Verlag, Heidelberg, 2007, pp.161-172.

24 Bonchi, F., Gionis, A., Ukkonen, A. (2011) Overlapping correlation clustering. Proc. of *11th IEEE International Conference on Data Mining* (ICDM), pp.51-60, doi:10.1109/ICDM.2011.114

25 Bottou L, Bengio Y (1995) Convergence properties of the k-means algorithms. *Advances in Neural Information Processing Systems* 7, G. Tesauro and D. Touretzky, eds., pp.585-592. MIT Press, 1995.

26 Bouras C., Tsogkas V. (2010) Advanced Knowledge - based Systems, Invited Session of the *14th International Conference on Knowledge – based and Intelligent Information & Engineering Systems*, Cardiff Wales, UK, 2010, pp.379 – 388.

27 Bradley P. S. and Fayyad U. M. (1998) Refining initial points for K-Means clustering, in Proc. *15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp.91–99.

28 Cagnina L., Errecalde M., Ingaramo D. and Rosso P. (2008) A discrete particle swarm optimizer for clustering short-text corpora. Proc. of the *3rd International Conference on Bioinspired Optimization Methods and their Applications* 13-14 October 2008, Ljubljana, Slovenia

29 R.B. Calinski and J. Harabasz (1974), A Dendrite Method for Cluster Analysis, *Comm. in Statistics*, vol. 3, pp.1-27.

30 Chen C-L and Tseng F.S.C. (2010) An integration of WordNet and fuzzy association rule mining for multi-label document clustering, *Data & Knowledge Engineering* 69(11). doi:j.datak.2010.08.003

31 Cheung Y (2003) k*-Means: A new generalized k-means clustering algorithm. *Pattern Recognition Letters* 24(15) pp.2883-2893. doi:10.1016/S0167-8655(03)00146-6

32 Da Silva A., Chiky R., Hébrail G. (2011) A clustering approach for sampling data streams in sensor networks. *Knowledge and Information Systems*, doi: 10.1007/s10115-011-0448-7

33 Dang H-X., Bailey J. (2010) A Hierarchical Information Theoretic Technique for the Discovery of Non Linear Alternative Clusterings. Proc. of the *16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* 2010, 573-582

34 Davies D. and Bouldin D. (1979) A cluster separation measure, *IEEE PAMI*, 1(2), pp.224–227.

35 De Souto MCP, Prudencio RBC, Soares RGF et al. (2008) Ranking and selecting clustering algorithms using a meta-learning approach. In: Proceeding of the *IEEE International Joint Conference on Neural Networks*, pp.3729-3735. doi: 10.1109/IJCNN.2008.4634333

36 Delibašić B, Kirchner K, Ruhland J et al. (2009) Reusable components for partitioning clustering algorithms. *Artif Intell Rev* 32 pp.59–75. doi: 10.1007/s10462-009-9133-6.

37 Delibašić, B., Vukićević, M., Jovanović, M., Kirchner, K., Ruhland, J., & Suknović, M. (2012). An architecture for component-based design of representative-based clustering algorithms. *Data & Knowledge Engineering*, 75 pp.78-98. doi:10.1016/j.datak.2012.03.005

38 Dembélé D, Kastner P (2003) Fuzzy C-means method for clustering microarray data. *Bioinformatics* 19 pp.973-980.

39 Demšar J., Zupan B., Leban G., Curk T. (2004) Orange: From Experimental Machine Learning to Interactive Data Mining, In *Knowledge Discovery in Databases: PKDD 2004*, pp.537-539.

40 Dhiraj K, Rath SK (2009) Gene Expression Analysis Using Clustering. In: Proceeding of *3rd International Conference on Bioinformatics and Biomedical Engineering*, pp.154-163.

41 Ding C, He X (2004) Principal component analysis and effective k-means clustering. In: Proceeding of the *SIAM International Conference on Data Mining*, pp.497-502.

42 Ding C., He X., Zha H., Gu M., and Simon H. (2001) A min-max cut algorithm for graph partitioning and data clustering, In Proc. *IEEE Int'l Conf. Data Mining*, 2001.

43 Dom B. (2001). An information theoretic external cluster validity measure. *IBM Research Report RJ 10219*. IBM's Almaden Research Center, San Jose, CA, 2001.

44 Dunn J., Well separated clusters and optimal fuzzy partitions (1974) *J. Cybern.*, 4(1), pp.95–104.

45 Ene A, Im, S., Moseley B. (2011) Fast Clustering using MapReduce. Proc. of the 17th ACM SIGKDD International Conference on *Knowledge Discovery and Data Mining* (KDD) 2011, pp. 681-689.

46 Errecalde M., Ingaramo D., and Rosso P (2010) A new AntTree-based algorithm for clustering short-text corpora. *Intelligence* 10(1), pp.1-7.

47 Ester M, Kriegel H, Sander J et al. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceeding of the *2nd International Conference on Knowledge Discovery and Data Mining*, pp.226-231.

48 Faber V. Clustering and the continuous K-means algorithm, *Los Alamos Science 22* (1994) pp. 138-144.

49 Fahim A., Salem A, Torkey F., Saake G., Ramadan M., An Efficient K-means with good initial startingpoints, *Computer Science and Telecommunications* 2 (19) (2009) 47-57.

50 Famili A., Liu G. and Liu Z (2004). Evaluation and optimization of clustering in gene expression data analysis. *Bioinformatics* 20(10), pp.1535-1545.

51 Forestier G, Gançarski P, Wemmert C (2010) Collaborative clustering with background knowledge. *Data & Knowledge Engineering* 69(2) pp.211-228. doi:10.1016/j.datak.2009.10.004

52 Fridlyand. (2012). Microarray Data Analysis. In T. Speed, *Selected Works of Terry Speed* pp.585-604. Berlin: Springer Berlin Heidelberg.

53 Geraci F, Leoncini M, Montangero M et al. (2009) K-Boost: a scalable algorithm for high-quality clustering of microarray gene expression data. *Journal of computational biology a journal of computational molecular cell biology* 16(6) pp.859-873. doi:10.1089/cmb.2008.0201

54 Giancarlo R, Utro F (2011) Speeding up the Consensus Clustering methodology for microarray data analysis. *Algorithms for molecular biology: AMB* 6.

55 Giancarlo R, Lo Bosco G, Pinello L (2010) Distance Functions, Clustering Algorithms and Microarray Data Analysis. In: Blum C., Battiti R. (eds.) *Learning and Intelligent Optimization* 6073, pp.125-138.

56 Guha S., Rastogi R., and Shim K. (1998) CURE: An efficient clustering algorithm for large databases. Proceedings of *ACM SIGMOD International Conference of Management of Data*.

57 Grujic M, Andrejiová M, Marasová D et al. (2012) Using principal components analysis and clustering analysis to assess the similarity between conveyor belts. Technics Technologies Education Management – TTEM 7(1):4-10.

58 Gurrutxaga I., Muguerza J., Arbelaitz O., Pérez JM. and Martín JI (2011) Towards a standard methodology to evaluate internal cluster validity indices. *Pattern Recognition Letters* 32(3), pp.505-515.

59 Halkidi M., Vazirgiannis M., and Batistakis Y (2000), Quality scheme assessment in the clustering process, in *PKDD*, London, UK, pp.265–276.

60 Halkidi M., Batistakis Y. and Vazirgiannis M (2001). On clustering validation techniques, *Journal of Intelligent Information Systems* 17, pp.107-145.

61 Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. (2009) The WEKA Data Mining Software: An Update, *SIGKDD Explor. Newsl.* 11(1) pp.10-18. doi:10.1145/1656274.1656278

62 Hamerly G, Elkan C (2003) Learning the k in k-means. In: Proceeding of the *Neural Information Processing Systems* vol.17 pp. 281–288.

63 Handl, J., & Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *Evolutionary Computation, IEEE Transactions on*, 11(1), 56-76.

64 Hansen P., Ngai E., Cheung B.K., Mladenovic N., Analysis of global K-means, an incremental heuristic for minimum sum-of-squares clustering, *Journal of Classification* 22 (2) (2005) pp.287–310.

65 Hartigan JA (1975) Clustering Algorithms. Probability & Mathematical Statistics, *John Wiley & Sons Inc*.

66 Hartigan J.A, Wong M.A (1979) A K-Means Clustering Algorithm. *Applied Statistics* 28, pp.100-108.

67 Hruschka, E.R., Campello, R., and de Castro, L.N. (2006). Evolving clusters in gene-expression data. *Inf Sci*, 176, pp.1898–1927.

68 Hubert L. and Arabie P. (1985), *Comparing partitions, Journal of Classification*, 2(1), pp.193–218.

69 Iam-on N, Boongoen T, Garrett S (2010) LCE: a link-based cluster ensemble method for improved gene expression data analysis. *Bioinformatics* 26, pp.1513-1519.

70 Ingaramo D., Pinto D., Rosso P. and Errecalde M (2008) Evaluation of internal validity measures in short-text corpora. *Computational Linguistics and Intelligent Text Processing*, pp.555–567.

71 Jain, A.K., and Dubes, R.C. (1988). Algorithms for Clustering Data, *Prentice-Hall, Englewood, NJ*.

72 Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3), pp. 264-323.

73 Jiang D., Tang C., and Zhang A. (2004) Cluster analysis for gene expression data: a survey. *IEEE Trans Knowledge and Data Eng* 16, pp.1370–1386.

74 Jovanović M, Delibašić B, Vukićević M, et al. (2011) Optimizing performance of decision tree component-based algorithms using evolutionary algorithms in Rapid Miner, In proc. of the *2nd RapidMiner Community Meeting and Conference*, Dublin.

75 Jovanovic, M., Vukicevic, M., Milovanovic, M., & Minovic, M. (2012). Using data mining on student behavior and cognitive style data for improving e-learning systems: a case study. International Journal of Computational Intelligence Systems, 5(3), 597-610.

76 Jovanović, M., Delibašić, B., Vukićević, M., Suknović, M., & Martić, M. (2014). Evolutionary approach for automated component-based decision tree algorithm design. *Intelligent Data Analysis*, 18(1), pp.63-77.

77 Karypis G. (2002) CLUTO a clustering toolkit, *Technical Report 02-017*, Dept. of Computer Science, University of Minnesota, 2002.

78 Kalogeratos A and Likas A (2011) Document clustering using synthetic cluster prototypes, *Data & Knowledge Engineering* 70(3), pp.284-306. doi:j.datak.2010.12.002

79 Kaufman L, Rousseeuw P (1990) Finding Groups in Data: An Introduction to Cluster Analysis. *Wiley*.

80 Kohonen T. (2001) Self-OrganizingMaps. *Springer Verlag, Berlin, third edition*.

81 Krishna, K., and Narasimha Murty, M. (1999). Genetic K-means algorithm. *IEEE Trans Syst Man Cybern, Part B.* 29, pp.433–439.

82 Kumar P, Wasan SK (2010) Comparative Analysis of k-mean Based Algorithms. *International Journal of Computer Science and Network Security* 10(4) pp.314-318.

83 Lai J. (2010) Fast global K-means clustering using cluster membership and inequality, *Pattern Recognition* 43 (5) pp.1954-1963.

84 Likas A., Vlassis M., Verbeek J. (2003) The global K-means clustering algorithm, *Pattern Recognition* 36 (2), pp.451-461.

85 Lloyd S (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28(2), pp.129-137. doi: 10.1109/TIT.1982.1056489

86 Lu Y., Lu S., Deng Y. and Brown, S. J. (2004a). Incremental genetic K-means algorithm and its application in gene expression data analysis. *BMC Bioinformatics* 5, pp.172–182.

87 Lu Y., Lu S., Fotouhi F., Deng Y. and Brown, S.J. (2004b). FGKA: a fast genetic K-means clustering algorithm. Proceedings of the 2004 *ACM symposium on Applied computing* (SAC), Nicosia, Cyprus, March 2004.

88 Lühr S. and Lazarescu M. (2009) Incremental clustering of dynamic data streams using connectivity based representative points, *Data & Knowledge Engineering* 68, pp.1-27.

89 Maitra R. (2009), Initializing partition-optimization algorithms, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 6 (1), pp.144–157.

90 Makretsov, N.A., Huntsman, D.G., Nielsen, T.O., Yorida, E., Peacock, M., Cheang, M.C.U., et al. (2004). Hierarchical clustering analysis of tissue microarray immunostaining data identifies prognostically significant groups of breast carcinoma. *Clin Cancer Res* 10, pp.6143–6151.

91 Matijaš, M., Suykens, J. A., & Krajcar, S. (2013). Load forecasting using a multivariate meta-learning system. *Expert systems with applications*, 40(11) pp. 4427-4437.

92 Maulik U. and Bandyopadhyay S. (2002), Performance evaluation of some clustering algorithms and validity indices, IEEE PAMI, vol. 24, pp.1650–1654, 2002.

93 Mercer D.P. and College L. (2003). Clustering large datasets. Available at http://www.stats.ox.ac.uk/~mercer/documents/Transfer.pdf. Accessed May 11, 2006.

94 I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler (2006) YALE: Rapid Prototyping for Complex Data Mining Tasks, In Proc. *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD-06), 2006.

95 Mierswa I, Morik K. (2005) Automatic Feature Extraction for Classifying Audio Data. *Machine Learning* 58 (2-3) pp.127-149.

96 Milligan GW, Cooper MC (1987) Methodology Review: Clustering Methods. *Applied Psychological Measurement* 11(4) pp.329-354. doi: 10.1177/014662168701100401.

97 Milligan G (1980). An Examination Of The Effect Of Six Types Of Error Perturbation On Fifteen Clustering Algorithms, *Psychometrika* 45 pp.325-342.

98 Mirkin B. (2005), Clustering for data mining: A data recovery approach, London, Chapman and Hall, 2005.

99 Moise G, Zimek A, Kröger P et al. (2009) Subspace and projected clustering: experimental evaluation and analysis. *Knowledge and Information Systems*, 21(3) pp.299-326. doi: 10.1007/s10115-009-0226-y

100 Monti S, Tamayo P, Mesirov J et al. (2003) Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*. 52 pp.91-118. doi: 10.1023/A:1023949509487

101 Mougeot, J.L., Bahrani-Mostafavi, Z., Vachris, J.C., McKinney, K.Q., Gurlov, S., Zhang, J., et al. (2006). Gene expression profiling of ovarian tissues for determination of molecular pathways reflective of tumorigenesis. *J Mol Biol* 358, pp.310–329.

102 Mumtaz K., Duraiswamy K. (2010) A Novel Density based improved K-means Clustering Algorithm–Dbkmeans, *International Journal on Computer Science and Engineering* 2(2) pp. 213 – 218.

103 Nascimento A, Prudencio R, de Souto M, et al. (2009) Mining Rules for the Automatic Selection Process of Clustering Methods Applied to Cancer Gene Expression Data. In: Proceedings of the *19th International Conference on Artificial Neural Networks: Part II*, Springer-Verlag Berlin, Heidelberg.

104 Nascimento MCV, Toledo FMB, Carvalho A (2010) Investigation of a new GRASP-based clustering algorithm applied to biological data. *Computers & Operations Research.* 37(8) pp. 1381-1388. doi:10.1016/j.cor.2009.02.014

105 Nielsen, T.O., West, R.B., Linn, S.C., Alter, O., Knowling, M.A., O'Connel, J.X., et al. (2002). Molecular characterization of soft tissue tumours: a gene expression study. Lancet 359, 1301–1307.

106 Pelleg D, Moore A (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters. Proc. of the *Seventeenth International Conference on Machine Learning*. Vol. 17. Morgan Kaufmann, pp.727-734.

107 Peterson A., Ghosh A., Maitra R. (2010) A systematic evaluation of different methods for initializing the K-means clustering algorithm, *IEEE Transaction on Knowledge and Data Engineering, 2010.In press*.

108 Piatetsky-Shapiro G., Tamayo P. (2003) Microarray Data Mining: Facing the Challenges. *ACM SIGKDD Explorations Newsletter*. 5(2) pp.1-5. doi:10.1145/980972.980974

109 Punera K., Ghosh J. (2008) Consensus-Based ensembles of soft clusterings. *Applied Artificial Intelligence*. 22, pp.780-810.

110 Pirim H., Gautam D., Bhowmik T. (2011) Performance of an ensemble clustering on biological datasets. *Mathematical and Computational Applications* 16(1), pp.87-96

111 Prudencio, R., de Souto, M., & Ludemir, T. (2011). Selecting machine learning algorithms using the ranking meta-learning approach. *Meta-Learning in Computational Intelligence* 358, pp.225-243. doi:10.1007/978-3-642-20980-2_7

112 Quackenbush J (2001) Computational analysis of microarray data. Nature reviews. *Genetics* 2, pp.418-427

113 R Development Core Team, R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria, 2008, ISBN 3-900051-07-0, URL http://www.R-project.org.

114 Raczynski L, Wozniak K, Rubel T, Zaremba K (2010) Application of Density Based Clustering to Microarray Data Analysis. *Intl Journal of Electronics and Telecommunications* 56(3) pp.281-286.

115 Radovanovic S., Vukićević M., Jovanović M., Delibašić B., Suknović M., Meta-learning system for clustering gene expression microarray data. In proc. of the *4th RapidMiner Community Meeting and Conference*, August 27- 29, Porto, Portugal, www.rcomm2013.org

116 Radovanović S. (2013), Rešavanje problema klasterovanja genskih ekspresija primenom sistema meta-učenja zasnovanog na komponentama, Master rad, Univerzitet u Beogradu, Fakultet Organizacionih Nauka.

117 R. Rakotomalala, TANAGRA: un logicielgratuit pour l'enseignement et la recherche, in *Actes de EGC'2005*, RNTI-E-3, vol. 2, 2005, pp.697-702.

118 Ramaswamy S., Ross K.N., Lander E.S., and Golub T.R. (2003). A molecular signature of metastasis in primary solid tumors. *Nat Genet*. 33, pp.49–54.

119 Romero C. And Ventura S. (2011) Educational data mining: a review of the state-of-the-art, *IEEE Trans. Syst. Man Cybernet. C Appl. Rev*., 40(6) pp.601–618.

120 Rousseeuw P.J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20 pp.53-65. doi:10.1016/0377-0427(87)90125-7

121 Saha S. & Bandyopadhyay S. (2007) A validity index based on cluster symmetry. In *Systems, Man and Cybernetics,IEEE International Conference on* (pp.462-467).

122 Sander J, Ester M, Kriegel H, et al. (1998) Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. *Data Mining Knowl Disc* (2) pp.169–194.

123 Schwarz G (1978) Estimating the dimension of a model. *Annals of Statistics* 6(2) pp.461–464.

124 Shai R., Shi T., Kremen T.J., Horvath S., Liau L.M., Cloughesy T.F., et al. (2003). Gene expression profiling identifies molecular subtypes of gliomas. *Oncogene* 22, pp.4918–4923.

125 Shamir R. and Sharan R. (2001) Algorithmic approaches to clustering gene expression data. *Current Topics in Computational Biology*, MIT Press, Boston, MA.

126 Shaham E., Sarne D., Ben-Moshe B. (2011) Sleeved Co-Clustering of lagged data. *Knowledge and Information Systems*. doi: 10.1007/s10115-011-0420-6

127 Shao J., Plant C., Yang Q., Böhm C. (2011) Detection of Arbitrarily Oriented Synchronized Clusters in High-dimensional Data. Proc. of *11th IEEE International Conference on Data Mining* (ICDM), 607-616, doi:10.1109/ICDM.2011.50

128 Sharma S. (1996) Applied multivariate techniques. NewYork, NY, USA: *John Wiley & Sons, Inc.*, 1996.

129 Smith-Miles K. (2008) Towards insightful algorithm selection for optimization using meta-learning concepts. In: Proceedings of the *IEEE International Joint Conference on Neural Networks* pp.4118-4124.

130 Smith-Miles K. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1), 6. doi:10.1145/1456650.1456656

131 Sonnenburg S, Braun M, Ong CS, et al. (2007) The Need for Open Source Software in Machine Learning. *J Mach Learn Res* 8 pp.2443-2466.

132 Steinley D. (2003) Local optima in K-means clustering: what you don't know may hurt you, *Psychological Methods* 8 (3) pp.294–304.

133 Suknović M., Delibašić B., Jovanović M., Vukićević M., Becejski-Vujaklija D., & Obradović, Z. (2012). Reusable components in decision tree induction algorithms. *Computational Statistics*, 27(1), pp.127-148.

134 Tajunisha N., Saravanan V. (2010) An Increased Performance of Clustering High Dimensional Data Using Principal Component Analysis, *First International Conference on Integrated Intelligent Computing*, 2010, pp.17-21.

135 Tan P.-N., Steinbach M. and Kumar V. (2005), Introduction to DataMining. USA: *Addison-Wesley Longman, Inc*.

136 Tang C., Wang S., Xu W. (2010) New fuzzy c-means clustering model based on the data weighted approach, *Data & Knowledge Engineering* 69 (9) pp.881-998.

137 Thalamuthu A, Mukhopadhyay I, Zheng X et al. (2006) Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics* 22, pp.2405-12.

138 van der Laan, M., Pollard, K.S., and Bryan, J. (2003). A new partitioning around medoids algorithm. *J Stat Comput Simul* 73, pp.575–584.

139 Vinh NX (2010) Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J Mach Learn Res* 11 pp.2837-2854.

140 Vinh N. X., Epps J., Bailey J. (2010) Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res* 11, pp.2837-2854.

141 Vukićević M, Delibašić B, Jovanović M, Suknović M, Obradović Z (2011) Internal Evaluation Measures as Proxies for External Indices in Clustering Gene Expression Data, In: Proceeding of the 2011 *IEEE International Conference on Bioinformatics and Biomedicine* (BIBM11), Atlanta, Georgia, USA, Nov. 12-15

142 Vukićević, M., Delibašić, B., Obradović, Z., Jovanović, M., Suknović, M. (2012a) " A Method for Design of Data-tailored Partitioning Algorithms for Optimizing the Number of Clusters in Microarray Analysis," Proc. 2012 *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, San Diego, CA, May 2012.

143 Vukićević M, Kirchner K, Delibašić B, Jovanović M, Ruhland J, Suknović M (2012b) Finding best algorithmic components for clustering microarray data, *Knowledge and Information Systems*, http://dx.doi.org/10.1007/s10115-012-0542-5

144 Vukićević, M., Jovanović, M., Delibašić, B., Išljamović, S., & Suknović, M. (2012c). Reusable component-based architecture for decision tree algorithm design. International Journal on Artificial Intelligence Tools, 21(05).

145 Vukićević M., Radovanović S., Milovanović M., and Minović M. (2014) Cloud Based Meta-learning System for Predictive Modeling of Biomedical Data, *Scientific World Journal, In press.*

146 Wan M., Jönsson A., Wang C., Li L., Yang Y. (2011) Web user clustering and Web prefetching using Random Indexing with weight functions. *Knowledge and Information Systems*. doi: 10.1007/s10115-011-0453-x

147 Walesiak M., Dudek A. (2007) Symulacyjna optymalizacja wyboruprocedury klasyfikacyjne jdladanegoty pudanych - charakterystykaproblemu, *Zeszyty Naukowe Uniwersytetu Szczecinskiego* 450, pp.635-646.

148 Ward J. H. (1963), Hierarchical grouping to optimize an objective function, *Journal of American Statistical Association* 58 (301), pp.236–244.

149 Welcsh, P.L., Lee, M.K., Gonzalez-Hernandez, R.M., Black, D.J., Mahadevappa, M., Swisher, E.M., and et al. (2002). BRCA1 transcriptionally regulates genes involved in breast tumorigenesis. *Proc Natl Acad Sci* USA, May 99, pp.7560–7565.

150 Wijaya A, Kalousis M, Hilario M (2010) Predicting Classifier Performance Using Data Set Descriptors and Data Mining Ontology. In: *Proceeding of the 3rd Planning to Learn Workshop*.

151 Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the *4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (pp.29-39).

152 D.H. Wolpert (1996), The lack of a priori distinctions between learning algorithms, *Neural computation* 8 (7), pp.1341-1390.

153 Wu L.F, Hughes T.R, Davierwala A.P (2002) Large-scale prediction of Saccharomyces cerevisiae gene function using overlapping transcriptional clusters. *Nature genetics* 31, pp.255-265.

154 Wu X, Kumar V, Quinlan J.R, et al. (2007) Top 10 algorithms in data mining, Knowledge and Information Systems 14(1), pp.1-37. doi: 10.1007/s10115-007-0114-2.

155 Wu J., Xiong H., Chen J. (2009) Adapting the Right Measures for K-means Clustering. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp.877-886.

156 Xie X.L, Beni G (1991) A Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence 13(8). pp.841-847.

157 Xiong H., Wu J., Chen J. (2009) K-means clustering versus validation measures: a data-distribution perspective, IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society 39 (2), pp.318-331.

158 Xu R., and Wunsch D. (2005). Survey of clustering algorithms. IEEE Trans Neural Networks 16, pp.645–678.

159 Xu R, Wunsch DC (2010) Clustering Algorithms in Biomedical Research: A Review. IEEE Reviews in Biomedical Engineering 3:120-154. doi:10.1109/RBME.2010.2083647

160 Yan Y., Chen L., Tjhi W-C. (2011) Semi-supervised fuzzy co-clustering algorithm for document classification. Knowledge and Information Systems, doi: 10.1007/s10115-011-0454-9.

161 Yedla M., Pathakota S.R., Srinivasa T.M. (2010) Enhancing K-means clustering algorithm with improved initial centers, International Journal of Computer Science and Information Technologies 1 (2) pp.121-125.

162 Yeung K.Y., Fraley A., Murua A.E., Raftery A.E., and Ruzzo W.L. (2001). Model-based clustering and data transformations for gene expression data. Bioinformatics 17, 977–987.

163 Yu Z., Wong H-S. and Wang H. (2007) Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics 23, pp.2888-2896.

164 Zhang T., Ramakrishnan R., and Livny M. (1996). BIRCH: An efficient data clustering method for very large database. ACM SIGMOD Conference, pp.103–114.

165 Zhao Y., Karypis G. (2004) Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. Machine Learning 55(3), pp.311-331.

166 Žalik K., An efficient k′-means clustering algorithm. Pattern Recognition Letters, 29 (9) (2008) 1385-1391.

9 Biografija autora

Milan Vukićević je rođen 12.06.1983. godine u Beogradu, Republika Srbija. Od tada živi u Beogradu, gde je završio osnovno obrazovanje. Srednje obrazovanje završava takođe u Beogradu, u XIII beogradskoj gimnaziji 2002. godine.

Fakultet Organizacionih Nauka, Univerziteta u Beogradu, upisuje 2002. godine. Diplomira na smeru za informacione sisteme 2007. Diplomske (Master) studije na odseku za Informacione sisteme i tehnologije, na Fakultetu organizacionih nauka, Univerziteta u Beogradu završio je 2008. godine sa prosečnom ocenom 10, odbranom diplomskog rada pod nazivom „Projektovanje skladišta podataka studentske službe FON-a“ čiji se praktični rezultati aktivno koriste pri analizi nastavnog procesa Fakulteta Organizacionih nauka. Iste godine zapošljava se kao Analitičar podataka studentske službe FON-a i kao demonstrator učestvuje u nastavi iz predmeta Sistemi za podršku odlučivanju i Ekspertni sistemi, kod prof. dr Milije Suknovića. 2008. godine držao kurseve i pripremio materijale (skripte i testove) “Skladišta podataka” i “OLAP” u okviru obuke zaposlenih u Republičkom zavodu za statistiku Srbije za firmu Belit. Iste godine radi na razvoju Aplikacija za elektronsku oglasnu tablu za oglašavanje javnih nabavki“, za potrebe Uprave za javne nabavke. Za svoj nastavni rad kontinuirano je ocenjivan od strane studenata prosečnom ocenom preko 4,40 na skali do 5.

Zaposlenje
2007 - 2008: Projektant informacionih sistema, BELIT - Belgrade Information Technologies.
2008 - 2010: Saradnik u nastavi, Univerzitet u Beogradu, Fakultet organizacionih nauka, Katedra za Organizaciju poslovnih sistema.
2010 - 2014: Asistent, Univerzitet u Beogradu, Fakultet organizacionih nauka, Katedra za Organizaciju poslovnih sistema.

Kontinuirana edukacija
2009: Pohađao kurs "Akademske veštine" koje je organizovalo Ministarstvo Nauke republike Srbije a sproveo dr Steve A. Quarrie
2011: Certified RapidMiner "Analyst"
2012: Certified RapidMiner "Expert"

Izjava o autorstvu

Potpisani: Milan Vukićević
Broj indeksa: 37/2008

Izjavljujem

da je doktorska disertacija pod naslovom

Razvoj i projektovanje algoritama za klasterovanje ekspresija gena

- rezultat sopstvenog istraživačkog rada,
- da predložena disertacija u celini ni u delovima nije bila predložena za dobijanje bilo koje diplome prema studijskim programima drugih visokoškolskih ustanova,
- da su rezultati korektno navedeni i
- da nisam kršio/la autorska prava i koristio intelektualnu svojinu drugih lica.

U Beogradu, 12.03.2014

Potpis doktoranda

Milan Vukićević

Izjava o istovetnosti štampane i elektronske verzije doktorskog rada

Ime i prezime autora: Milan Vukićević
Broj indeksa: 37/2008
Studijski program: Operaciona istraživanja
Naslov rada: **Razvoj i projektovanje algoritama za klasterovanje ekspresija gena**
Mentor prof. dr Milija Suknović

Potpisani Milan Vukićević

Izjavljujem da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predao/la za objavljivanje na portalu **Digitalnog repozitorijuma Univerziteta u Beogradu**.

Dozvoljavam da se objave moji lični podaci vezani za dobijanje akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada.

Ovi lični podaci mogu se objaviti na mrežnim stranicama digitalne biblioteke, u elektronskom katalogu i u publikacijama Univerziteta u Beogradu.

U Beogradu, 12.03.2015

Potpis doktoranda

Milan Vukićević

Izjava o korišćenju

Ovlašćujem Univerzitetsku biblioteku „Svetozar Marković“ da u Digitalni repozitorijum Univerziteta u Beogradu unese moju doktorsku disertaciju pod naslovom:

Razvoj i projektovanje algoritama za klasterovanje ekspresija gena

koja je moje autorsko delo.

Disertaciju sa svim prilozima predao/la sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalni repozitorijum Univerziteta u Beogradu mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio/la.

1. Autorstvo
2. Autorstvo - nekomercijalno
3. Autorstvo – nekomercijalno – bez prerade
4. Autorstvo – nekomercijalno – deliti pod istim uslovima
5. Autorstvo – bez prerade
6. Autorstvo – deliti pod istim uslovima

(Molimo da zaokružite samo jednu od šest ponuđenih licenci, kratak opis licenci dat je na poleđini lista).

U Beogradu, 17.03.2014

Potpis doktoranda

Milan Mihajlović