УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Радмила Нинковић

Кластеровање високодимензионалних података

Мастер рад

Београд 2022.

Ментор:

др Ненад МИТИЋ, редовни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Саша МАЛКОВ, ванредни професор
Универзитет у Београду, Математички факултет

др Јована КОВАЧЕВИЋ, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 21.09.2022.

Наслов мастер рада: Кластеровање високодимензионалних података

Резиме: Процес кластеровања представља поделу улазног скупа података објеката у групе (кластере) сличних објеката. Објекти се обично представљају као тачке у вишедимензионом простору, где је број димензија једнак броју атрибута који представљају карактеристике објекта. Напредак технологије је омогућио брже и једноставније добијање скупова објеката који су истовремено постали све сложенији са све већим бројем карактеристика. Број димензија таквих објеката се креће у распону од једноцифреног броја до неколико десетина хиљада, као у случају микронизова и различитих медицинских и биоинформатичких података. Вектори којима се објекти представљају обично садрже све већи број нула (ретки вектори) како се број димензија увећава. Јавља се проблем кластеровања таквих података при чему највећи број уобичајених техника није погодан због проблема са избором метрика при рачунању сличности. Додатно, технике димензионе редукције (нпр. PCA и SVD) нису одговарајуће уколико се различити кластери налазе у различитим потпросторима. Циљ рада је приказ техника које омогућују кластеровање скупова високодимензионих података. Поред карактеристика, предности и недостатака приказаних техника биће наведене и метрике за одређивање сличности које омогућују ефективно одређивање сличности таквих података. У раду ће бити приказано и поређење резултата добијених наведеним техникама кластеровањем скупа високодимензионалних биоинформатичких података са више од 10000 атрибута.

Кључне речи: кластеровање, високодимензионални подаци, кластеровање по потпросторима

Садржај

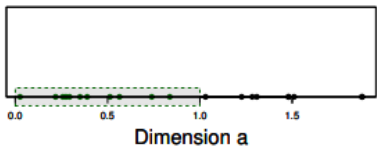
1. Увод

Кластер анализа има за циљ да открије групе или кластере сличних објеката. Објекти су обично представљени као тачка у вишедимензионалном простору. Сличност између објеката често се одређује коришћењем мера растојања у различитим димензијама у скупу података. Напредак технологије је олакшао и убрзао прикупљање података, али је довео до већих, сложенијих скупова података са много објеката и димензија. Како скупови података постају све већи и разноврснији, потребне су адаптације постојећих алгоритама како би се одржао квалитет и брзина кластеровања. Традиционални алгоритми кластеровања узимају у обзир све димензије улазног скупа података у покушају да сазнају што више о сваком описаном објекту. Међутим, у високодимензионалним подацима многе димензије су често ирелевантне. Такве димензије могу збунити алгоритме кластеровања скривањем кластера у подацима са шумом. Под високим димензијама уобичајно је да се сви објекти у скупу података налазе на скоро једнакој удаљености један од другог, што потпуно маскира кластере. Методе избора карактеристика су донекле успешно коришћене за побољшање квалитета кластера. Алгоритми који врше избор карактеристика проналазе подскуп димензија на којима ће се извршити кластеровање уклањањем небитних и сувишних димензија. За разлику од метода избора карактеристика које испитују скуп података у целини, алгоритми кластеровања потпростора локализују своју претрагу и способни су да открију кластере који постоје у више могуће преклапајућих потпростора.

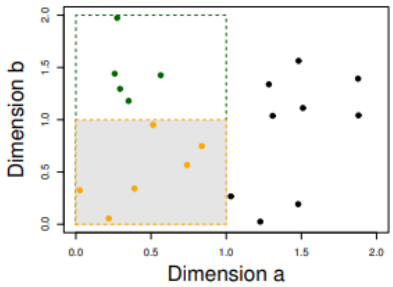
Још један разлог због којег се многи алгоритми кластеровања боре са високодимензионалним подацима је „проклетство димензионалности". Како се број димензија у скупу података повећава, мере сличности постају све бесмисленије. Додатне димензије шире тачке све док у врло високим димензијама нису готово једнако удаљене једна од друге. Слика 1 илуструје како додатне димензије шире тачке у узорку скупа података. Скуп података састоји се од 20 тачака насумично постављених на интервалу [0, 2] у свакој од три димензије. Слика 1а приказује податке пројектоване на једну осу. Тачке су релативно близу на бројчаној оси и захватају интервал јединичне величине. Слика 1б приказује исте податке представљене у две димензије. Сада се само око четвртине тачака налази у првом јединичном интервалу. На слици 1в је додата трећа димензија која додатно раздваја податке у простору. Јединични интервал сада садржи само око једне осмине првобитног броја тачака. [1]

Додавањем нових димензија, тачке ће се удаљавати све док не буду скоро подједнако међусобно удаљене и док удаљеност више не буде од значаја. Проблем постаје сложенији када су објекти повезани на различите начине у различитим подскуповима димензија. Алгоритми кластеровања управо имају за циљ да открију ову врсту

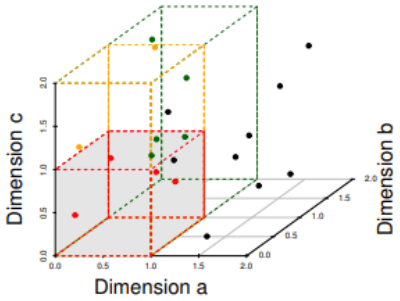
повезаности између објеката. Да би се пронашли такви кластери, небитне карактеристике морају бити уклоњене како би се алгоритам кластеровања фокусирао само на релевантне димензије. Кластери који се налазе у ниже димензионалном простору имају тенденцију да се лакше интерпретирају, што омогућава кориснику да боље усмери даље проучавање.



Слика 1а. Пројекција података у једној димензији



Слика 1б. Пројекција података у две димензије



Слика 1в. Пројекција података у три димензије

Потребни су алгоритми који могу да открију кластере формиране у различитим потпросторима великих скупова података високе димензионалности и представе их на лако разумљив и смислен начин. Овај сценарио постаје све чешћи јер се тежи истраживању података из различитих перспектива. Ако се документи групишу користећи речи као атрибуте, две групе докумената вероватно би биле повезане различитим

скуповима атрибута. Још један пример налазимо у биоинформатици са подацима о микронизовима ДНК. Једна популација ћелија у експерименту са микронизом може бити слична другој јер обе производе хлорофил, па се стога групишу на основу нивоа експресије одређеног скупа гена повезаних са хлорофилом. Међутим, популација ћелија у експерименту може бити слична другој и због тога што ћелије регулишу циркадијални сат[1] у организму. У овом случају, они би били груписани на другом скупу гена. Ако су ове карактеристике примарне при одређивању кластера, тада би се при кластеровању користили два различита подскупа гена. Овакви скупови података представљају нове изазове и циљеве за кластеровање. Алгоритми кластеровања потпростора су један од одговора на те изазове. Истичу се у ситуацијама попут горе описаних, где су објекти повезани на више различитих начина.

1.1. Проклетство димензионалности

Високодимензионални подаци тј. подаци описани великим бројем атрибута, представљају специфичне изазове за кластеровање. Познато је да такозвани термин „проклетство димензионалности“ првобитно настао да би се описало опште повећање сложености различитих рачунарских проблема које како димензионалност расте чини традиционалне алгоритме кластеровања неефикасним. Проклетство димензионалности, између осталог, значи да се са повећањем броја димензија са једне стране добија већа количина информација о подацима, а са друге се примећује губитак смислене разлике између сличних и различитих објеката. С обзиром на то да су високодимензионални објекти често слични, потребни су нови приступи кластеровања. Сходно томе, недавна истраживања усредсређена су на развој техника и алгоритама кластеровања посебно за високо-димензионалне податке. Постоји низ метода кластеровања које формирају различите моделе кластера и укључују различите алгоритамске приступе за откривање кластера. Заједничко свим приступима је чињеница да захтевају одређену основну процену сличности између инстанци података [2].

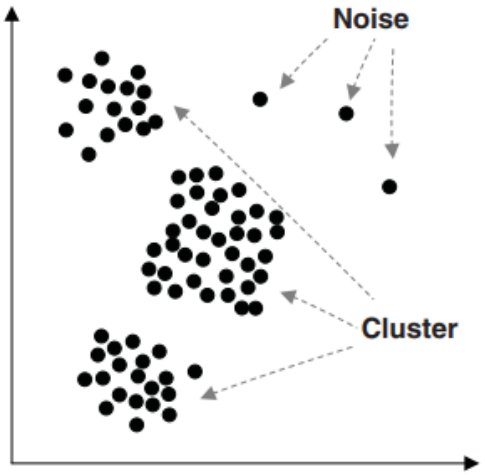
Опште запажање да многи алгоритамски проблеми показују знатно већу сложеност са повећањем броја димензија. Са друге стране, са повећањем димензионалности повећава се и број карактеристика на основу којих могу да се добију корисне информације. Ове супротности су биле разлог да термин који их описује добије назив "Проклетство димензионалности". У високодимензионалним просторима растојања међу објектима података постају све више бесмислена. То је од велике важности за сваки модел и алгоритам који се ослања на неку процену сличности објеката података. Традиционални

[1] Циркадијални сат тј. биолошки сат контролише биоритам свих живих бића. Утиче на ритмичност психофизичких збивања у човеку која су у вези са ритмичним збивањима у природи као што је смена годишњих доба, дана и ноћи, плиме и осеке итд.

алгоритми кластеровања за просторе ниских димензија, који се заснивају на процени сличности између парова или група објеката података као део процеса додељивања кластера, стога не успевају да пронађу значајне кластере у високодимензионалним подацима. Сходно томе, развијени су нови алгоритми за обраду високодимензионалних података.

У многим данашњим задацима истраживања података циљ је аутоматска анализа таквих података. Према томе, алгоритми морају бити способни за обраду великог броја атрибута који описују податке. Типичан проблем кластеровања велике количине података је анализа података о експресији гена за велике скупове података у биоинформатици.

У кластеровању сличност података није нужно дефинисана на основу удаљености објеката, већ на основу густине или повезаности објеката. За скуп података *D*, циљ кластеровања је одређивање *n* кластера $C_1,...,C_n$ тако да су инстанце из *D* део једног или више кластера. У приступима тврдог кластеровања објекти су елементи тачно једног кластера, док у приступима меког кластеровања објекти могу бити део неколико кластера (могуће са одређеном вероватноћом). Број кластера *n* је или улазни параметар, или се аутоматски одређује као део алгоритма кластеровања. Неки приступи формирају посебне кластере који садрже шум (енг. noise), тј. једну групу инстанци које нису део ни једног кластера, јер се разликују од готово свих осталих инстанци података. Уместо формирања посебног кластера ове инстанце се проглашавају елементима ван граница. Пример кластеровања у две димензије дат је на слици 2.



Слика 2. Приказ кластера и елемената шума

Скуп података подељен је на четири скупа - три кластера и део са шумом, одн. елементима ван граница. Алгоритми кластеровања формирају кластере тако да је

максимизован неки критеријум кластеровања, као што је сличност унутар истог кластера или различитост између различитих кластера.

2. Смањење димензионалности

Технике груписања података са високим димензијама укључују трансформацију карактеристика и технике избора особина. Технике трансформације карактеристика покушавају да сведу скуп података у мање димензије стварањем комбинација оригиналних атрибута. Ове технике су врло успешне у откривању латентне структуре у скуповима података. Међутим, пошто чувају релативну удаљеност између објеката, мање су ефикасни када постоји велики број небитних атрибута који скривају кластере. Такође, нове карактеристике су комбинације оригиналних и могу бити мање релевантне у контексту домена. Алгоритми избора особина издвајају само најрелевантније димензије из скупа података да би се открили инстанце које су сличне само у подскупу њихових атрибута. Иако су прилично успешни на многим скуповима података, алгоритми за избор карактеристика имају потешкоће када се кластери налазе у различитим потпросторима. Управо је ова врста података мотивисала развој алгоритама кластеровања потпростора. Ови алгоритми одабиром релевантних потпростора за сваки кластер посебно доприносе у избору карактеристика.

У приступима смањења димензионалности, идеја је пресликавање високо-димензионалних података у податке мање димензије. Губитак информација кроз овај процес у великој мери зависи од димензионалности података. Најраспрострањенији приступ је приступ анализе главних компонената (PCA *eng. Principal component analysis*). ПЦА одређује линеарну трансформацију која се обично користи за смањење димензионалности задржавањем само трансформисаних димензија највеће варијансе, а одбацивањем преосталих [4].

Настали губитак информација није увек прихватљив. Штавише, различите инстанце података могу имати различите релевантне димензије. Ове разлике се не бележе ако се користи глобална процена релевантности за све инстанце података.

2.1. Трансформација атрибута

Трансформације карактеристика (атрибута) обично се користе на високодимензионалним скуповима података. Ове методе укључују технике као што су анализа главних компоненти и декомпозиција појединачне вредности. Трансформације углавном задржавају изворну, релативну удаљеност између објеката. На тај начин оне сумирају скуп података стварањем линеарних комбинација атрибута у циљу откривања латентне

структуре. Трансформација карактеристика је чест корак у препроцесирању који омогућава алгоритму кластеровања да користи само неколико новонасталих карактеристика. Неколико метода кластеровања укључило је употребу таквих трансформација да би се идентификовале важне карактеристике и итеративно побољшало њихово кластеровање. Иако су често врло корисне, ове технике заправо не уклањају ниједан оригинални атрибут из разматрања. Дакле, информације из ирелевантних димензија се чувају, чинећи ове технике неефикасним у откривању кластера када постоји велики број ирелевантних атрибута који маскирају кластере. Још један недостатак употребе комбинације атрибута је што их је тешко протумачити, што често чини резултате кластеровања мање корисним. Због тога су трансформације карактеристика најприкладније за скупове података где је већина димензија релевантна за задатак кластеровања.

2.2. Избор карактеристика

Избором карактеристика покушавају се пронаћи атрибути скупа података који су најрелевантнији за истраживање података. То је често коришћена и моћна техника за проблем смањивања димензионалности. Избор карактеристика укључује претрагу различитих подскупова карактеристика и процену сваког од тих подскупова користећи неки критеријум. Најпопуларније стратегије претраживања су похлепна секвенцијална претраживања кроз простор карактеристика, било унапред или уназад. Критеријуми за оцењивање следе један од два основна модела, модел омотача и модел филтера.

Технике модела омотача процењују скуп података помоћу алгоритма за кластеровање података који ће на крају бити примењен. Дакле, они „обавијају“ поступак одабира око алгоритма за кластеровање података. Алгоритми засновани на моделу филтера испитују суштинска својства података како би проценили подскуп карактеристика пре кластеровања. Највећи део техника за избор карактеристика је везан за примену у класификацији, док је мањи део оријентисан ка потребама кластеровања. Главна разлика између избора карактеристика у класификацији и кластеровању је критеријум оцењивања. Приступи засновани на филтерима готово се увек ослањају на ознаке класа, најчешће процењујући корелацију између карактеристика и ознака класа. Код кластеровања не постоје универзално прихваћене мере тачности и ознаке класа [1].

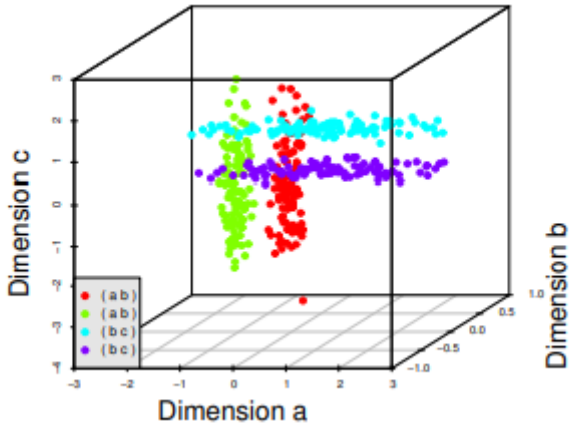
Поред коришћења различитих критеријума за оцењивање, коришћене су и разне методе претраживања у покушајима да се скупови података скалирају до великих, високо димензионалних скупова података. Са таквим скуповима података, насумично претраживање постаје одржива хеуристичка метода и користи се са многим наведеним критеријумима. Узорковање је још једна метода која се користи за побољшање скалабилности алгоритама.

3. Кластеровање по потпросторима

Кластеровање по потпросторима је проширење избора карактеристика. Циљ је пронаћи кластере у различитим потпросторима истог скупа података. Баш као и код избора карактеристика, кластеровање по потпросторима захтева метод претраге и критеријуме процене. Поред тога, кластеровање по потпросторима мора некако ограничити обим критеријума оцењивања како би се узели у обзир различити потпростори за сваки кластер.

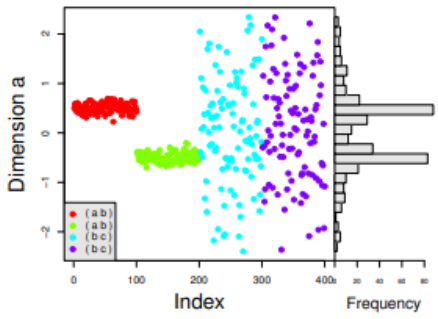
3.1. Мотивација

Можемо користити једноставан скуп података да бисмо илустровали потребу за кластеровањем по потпросторима. Направили смо узорак скупа података са 400 примерака у три димензије. Скуп података подељен је у четири кластера од по 100 примерака, од којих сваки постоји у само две од три димензије. Прва два кластера постоје у димензијама *a* и *b*. Подаци имају нормалну расподелу са очекивањем -0,5 и 0,5 у димензији *a* и 0,5 у димензији *b* и стандардним одступањима од 0,2. У димензији *c* ови кластери имају μ = 0 и σ = 1. Друга два кластера су у димензијама *b* и *c* и генерисана су на исти начин. Подаци се могу видети на слици 3. Кластеровањем ових података применом К-средина добијају се лоши резултати, јер се добијају кластери у ирелевантним димензијама за посматрани простор. Самим тим у високодимензионалним скуповима кластере постаје немогуће пронаћи [1].

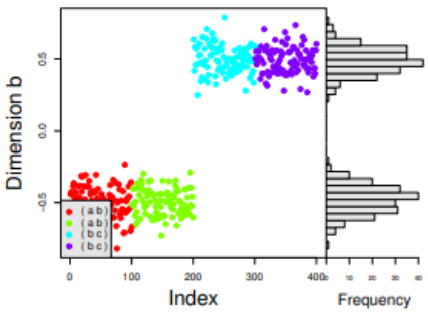


Слика 3. Приказ кластера из две димензије у тродимензионалном простору

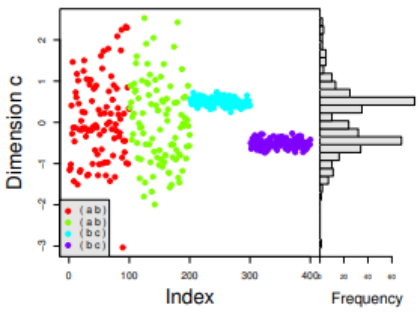
Употреба техника трансформације карактеристика као што је анализа главних компоненти у овом случају не даје боље резултате, јер су релативне удаљености сачуване као и ефекти ирелевантне димензије. Уместо тога, можемо покушати да употребимо алгоритам за одабир обележја да бисмо уклонили једну или две димензије. Слике 4а, 4б и 4в приказују податке пројектоване у једној димензији (са сортираним индексом на *x*-оси ради лакшег тумачења). Ниједна од ових пројекција података није довољна да у потпуности одвоји четири кластера. Ако се уклони само једна димензија, добију се графикони приказани на сликама 5а, 5б и 5в.



Слика 4а. Приказ података пројектованих у a димензији

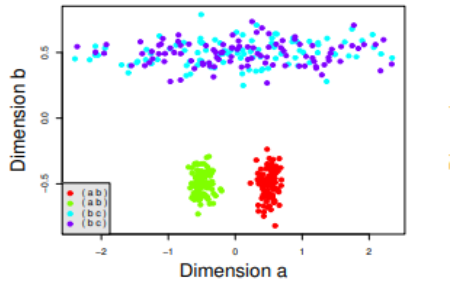


Слика 4б. Приказ података пројектованих у b димензији

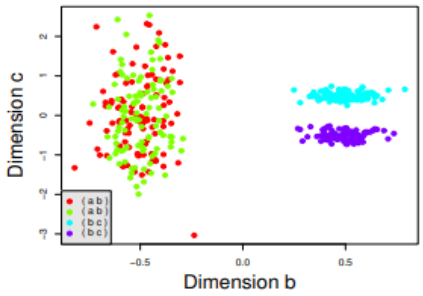


Слика 4в. Приказ података пројектованих у c димензији

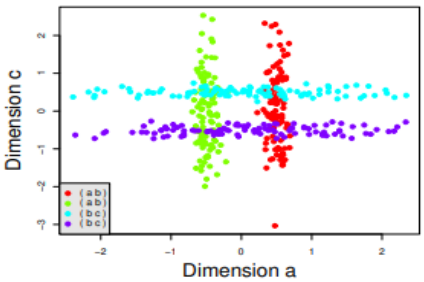
Лако се може уочити да се прва два кластера (црвени и зелени) лако раздвајају један од другог и од остатка података када се посматрају у димензијама *a* и *b* (слика 5а). Подаци из кластера су распрострањени само у димензијама *a* и *b*, а уклањањем димензије *c* уклања се шум из та два кластера. Преостала два скупа (плави и љубичасти) се у овом случају потпуно преклапају, јер су подаци из кластера распрострањени у димензијама *b* и *c*, а уклањање димензије *c* учинило је да се не разликују. Из тога следи да су највидљивији у димензијама *b* и *c* (слика 5б). Дакле, кључ за проналажење сваког од кластера у овом скупу података је тражење у одговарајућим потпросторима. У даљем тексту биће приказани различити циљеви и изазови кластеровања потпростора.



Слика 5а. Приказ података пројектованих у a и b димензијама



Слика 5б. Приказ података пројектованих у b и c димензијама



Слика 5в. Приказ података пројектованих у a и c димензијама

3.2. Циљеви и изазови

Главни циљ кластеровања је проналажење висококвалитетних кластера у разумном року. Међутим, различити приступи дефинишу кластере на различите начине, што онемогућава дефинисање универзалне методе кластеровања. Због тога алгоритми кластеровања морају давати разумне претпоставке, које су често засноване на улазним параметрима. Ипак, резултати се морају пажљиво анализирати како би се потврдила њихова поузданост.

Иако све методе кластеровања организују инстанце скупа података у групе, неке захтевају да кластери буду дискретне партиције, док неке формирају кластере који се преклапају омогућавајући инстанци да припада више од једног кластера. Алгоритми кластеровања потпростора такође морају дефинисати подскуп карактеристика релевантних за сваки кластер. Готово увек је дозвољено да се ови потпростори преклапају. Неке методе дозвољавају одступања која су дефинисана као тачке које не припадају ниједном кластеру. Алгоритми кластеровања такође се разликују по облику кластера које формирају.

Ниједна метода кластеровања није боља од друге. Будући да не постоји универзална метода кластеровања, не постоји ни универзална мера помоћу које би се могли упоредити резултати кластеровања. Одређене методе кластеровања су прикладније за одређене проблеме од других. Знање специфично за домен је често врло корисно у одређивању методе кластеровања која ће бити примењена [4].

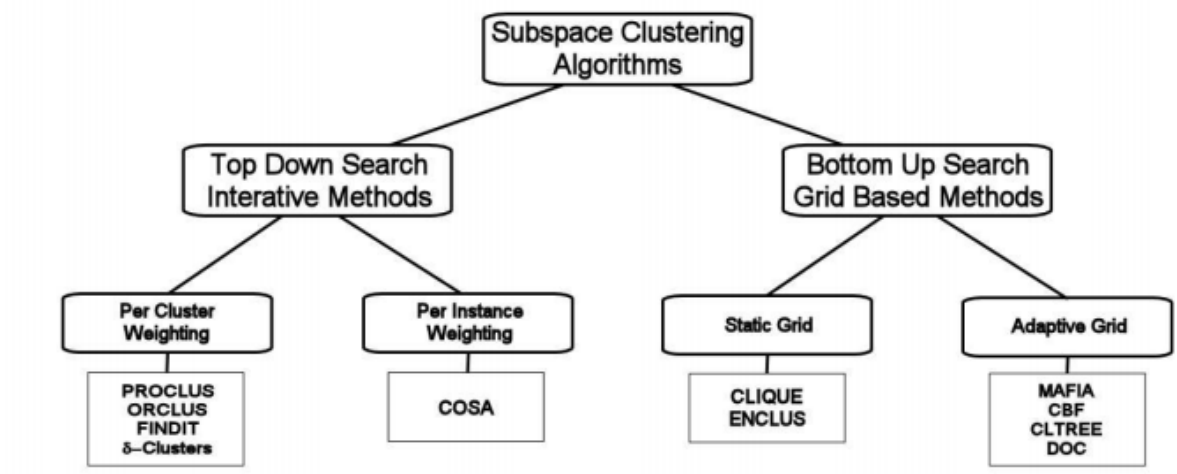
Сама природа кластеровања пружа кориснику мало информација о инстанцама које жели да открије. Због проблема одређивања квалитета кластера, често је тешко одредити број кластера у одређеном скупу података. Проблем алгоритама потпростора се огледа у томе што они такође морају одредити димензионалност потпростора. Многи алгоритми се ослањају на улазне параметре, међутим њихове тачне вредности често није могуће утврдити унапред, док са друге стране корисници често очекују да одговор на питање о оптималном броју кластера добију од самог алгоритма. Да би били најкориснији, алгоритми би требали тежити стабилности у низу вредности параметара. Међутим, процена квалитета кластера је од суштинске важности, јер ће сваки алгоритам кластеровања произвести одређене кластере за сваки скуп података, чак и ако су резултати бесмислени. Мере квалитета кластера не гарантују њихов значај, али формирани кластери треба да представљају значајан образац у подацима у контексту посматраног домена, што често захтева анализу од стране човека.

Да би верификовали резултате кластеровања, стручњаци из домена у коме се врши истраживање захтевају да кластери буду представљени на разумљив и смислен начин. Опет, потпросторни кластери не морају представљати само кластер, већ и потпростор у којем он постоји. Тумачење резултата кластеровања је такође важно, јер је кластеровање

често први корак који помаже усмеравању даљег проучавања. Поред стварања висококвалитетних, разумљивих кластера, алгоритми кластеровања такође се морају добро скалирати с обзиром на број инстанци и број димензија у великим скуповима података. Методе кластеровања потпростора такође морају бити скалабилне с обзиром на димензионалност потпростора у којима се налазе кластери. У високодимензионалним подацима, број могућих потпростора је огроман, што захтева ефикасне алгоритме претраживања.

4. Методе кластеровања по потпросторима

Овај одељак резимира главне алгоритме кластеровања по потпросторима. Две главне врсте алгоритама разликују се по начину претраживања потпростора. Наиван приступ може бити претрага кроз све могуће потпросторе и коришћење техника провере коректности кластера за одређивање потпростора са најбољим кластерима. Овај приступ није изводљив, јер је проблем формирања подскупова нерешив. Потребне су софистициране хеуристичке методе претраживања које одређују многе друге карактеристике алгоритма. Према томе, прва подела у хијерархији (слика 6) дели алгоритме кластеровања по потпросторима у две групе, методе претраживања одозго према доле и одоздо према горе. Слика 6 представља хијерархију алгоритама кластеровања по потпросторима организовану према употребљеној техници претраживања и мерама које се користе за одређивање локалитета.



Слика 6. Хијерархијска подела алгоритама кластера по потпросторима

Кластеровање по потпросторима врши процену карактеристике на нивоу подскупа података који представљају кластер. Наредна подела дели две категорије алгоритама на основу тога како одређују меру локалитета помоћу које ће се проценити потпростори.

Локално различита релевантност димензија (неке димензије су различите на неком подскупу, у неком локалу) је питање којим се баве два правца истраживања:

1. Потпростор или пројектовано кластеровање
2. Претраживање потпростора

Потпростор или пројектовано кластеровање претражују кластере у произвољним потпросторним пројекцијама високодимензионалног простора. Резултат кластеровања више није скуп кластера $C_1, ..., C_n$, већ потпросторних кластера (C_1, S_1), ..., (C_n, S_n) где је S_i подскуп димензија D који описују податке високе димензије скупа DS. Дакле, потпростор и пројектовано кластеровање одређују смањење димензионалности по кластеру [1].

Претраживање потпростора је првобитно уведено у CLIQUE алгоритму који простор карактеристика дискретизује у правилне интервале у свакој димензији. Густе ћелије које садрже више објеката од прага комбинују се са другим густим ћелијама у приступу одоздо према горе на основу димензионалности потпростора. Суседне густе ћелије се спајају како би се формирали кластери подспростора.

4.1. Методе претраживања потпростора одоздо према горе

Метода претраживања одоздо према горе користи предност својства затварања надоле да би смањила простор претраге користећи АПРИОРИ приступ. Алгоритми формирају хистограм за сваку димензију и бирају оне ћелије са густином изнад датог прага. Својство затварања надоле значи да ако постоје неке густе ћелије у к-димензионом простору, онда постоје густе ћелије у свим (к-1)-димензионим пројекцијама простора. Потпростори се тада могу формирати користећи само оне димензије које садрже густе ћелије, па се због тога простор претраге смањује. Затим се суседне густе ћелије комбинују и формирају кластере, што није увек интуитивно. У неким случајевима један кластер се може погрешно представити као два мања кластера. Алгоритам се зауставља када више не пронађе густе ћелије.

Природа приступа одоздо према горе доводи до преклапања кластера, где једна инстанца може бити у нула или више кластера. Добијање значајних резултата зависи од правилног постављања величине мреже и прага густине. То може бити посебно тешко, поготово јер се величина мреже и праг густине користе у свим димензијама скупа података. Популарна адаптација ове стратегије омогућава генерисање прилагодљиве мреже на основу података, да би стабилизовала резултате у широком дијапазону прагова густине.

Постоје два главне групе алгоритама. Прву групу чине приступи који се налазе у основи алгоритама CLIQUE и ENCLUS, који користе мрежу статичке величине за поделу сваке димензије у партиције. Друга група садржи приступе који су укључени у MAFIA, CBF, CLTree и DOC алгоритме. Ове методе се разликују по начину на који одређују тачке пресека на основу података за партиције сваке димензије. MAFIA и CBF користе хистограме за анализу густине података у свакој димензији. CLTree користи стратегију засновану на стаблу одлучивања како би пронашао најбољу тачку пресека. DOC користи

максималну ширину и минималан број инстанци по кластеру за спровођење насумичне претраге.

4.1.1. CLIQUE (Clustering In QUEst)

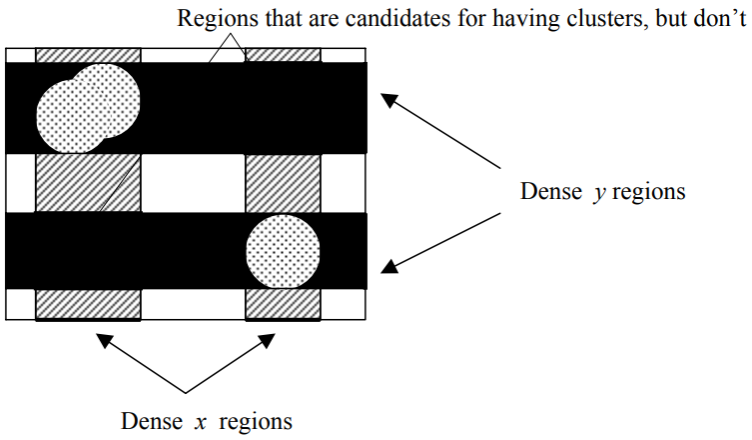
CLIQUE је један од првих алгоритама који је имао за циљ да пронађе кластере у потпросторима скупа података. Алгоритам комбинује кластеровање засновано на густини и мрежи и користи технику АПРИОРИ за проналажење потпростора погодних за кластеровање. Када се пронађу густи потпростори, они се сортирају према покривености, при чему је покривеност дефинисана као део скупа података који покривају густе ћелије у потпростору. Потпростори са највећом покривеношћу се задржавају, а остали се одбацују. Алгоритам затим проналази густе суседне ћелије у сваком од изабраних потпростора. Кластери настају комбиновањем таквих мрежних ћелија користећи похлепну стратегију. Алгоритам започиње са произвољном густином и похлепно формира групације ћелија у свакој димензији док унија тих групација не покрије читав кластер. Кластери чија је величина изнад жељене се смањују тако што се најмање групације ћелија одбацују док се више не може уклонити ниједна групација. ДНФ изрази се користе за представљање кластера и често су врло разумљиви. Кластери се такође могу преклапати, што значи да инстанце могу припадати више кластера. То је често корисно у кластеровању подспростора, јер кластери често постоје у различитим потпросторима и тако представљају различите релације [3].

CLIQUE је у стању да пронађе многе врсте и облике кластера и представља их на лако разумљиве начине. Такође, CLIQUE може да пронађе било који број кластера у било којем броју димензија и не постоји параметар који унапред одређује број кластера. Приступ генерисања кластера заснован на растућој густини омогућава CLIQUE-у да пронађе кластере произвољног облика. Одређивање параметара за неки скуп података може бити тешко. Величина мреже и праг густине су улазни параметри који у великој мери утичу на квалитет резултата кластеровања.

Чак и уз добро одабране параметре, фаза одсецања понекад може елиминисати мале, али важне групације из разматрања. Као и други алгоритми одоздо према горе, CLIQUE добро савладава велики број инстанци и димензија у скупу података, али не савладава добро велики број димензија у излазним кластерима. То најчешће није главни проблем, јер се кластеровање потпростора обично користи за проналажење нискодимензионалних кластера у високодимензионалним подацима.

На пример, ако испитамо расподелу *k* (хоризонталне) и *i* (вертикалне) координате тачака на слици 7, у једнодимензионалним расподелама видимо густа подручја која одражавају постојање дводимензионалних кластера. На слици 7, сиви хоризонтални стубови и искошени вертикални стубови указују на пројекције кластера на вертикалну и

хоризонталну осу. Слика 7 такође илуструје да велика густина у нижој димензији може само да сугерише могуће локације кластера у вишој димензији.



Слика 7. Приказ пројекције кластера на вертикалној и хоризонталној оси

4.1.2. ENCLUS

ENCLUS је метода кластеровања потпростора која се заснива на алгоритму CLIQUE. Међутим, ENCLUS не мери директно густину или покривеност, већ мери ентропију. Алгоритам се заснива на томе да потпростор са кластерима обично има нижу ентропију од потпростора без кластера. Могућност да се потпростор кластерује дефинисана је помоћу три критеријума: покривеност, густина и корелација. Ентропија се може користити за мерење сва три критеријума. Ентропија се смањује како се повећава густина ћелија. Под одређеним условима, ентропија ће се такође смањивати са повећањем покривености. Однос камата је мера корелације и дефинише се као разлика између збира мерења ентропије за скуп димензија и ентропије вишедимензионалне расподеле. Веће вредности указују на већу корелацију између димензија, а однос камата око нуле указује на независне димензије [1].

ENCLUS такође користи АПРИОРИ, приступ одоздо према горе као и CLIQUE за одређивање значајних потпростора. Одсецање се постиже користећи својство затварања наниже за ентропију (испод прага ω) и својство затварања према горе за корелацију да би се пронашли потпростори са минималном корелацијом. Ако је потпростор у великој корелацији (изнад прага ε), сви његови надпростори не смеју бити минимално повезани. Будући да би потпростори камата могли бити у минималној корелацији, ENCLUS претражује занимљиве потпросторе рачунањем добитка од камате и проналажењем потпростора чија ентропија прелази ω, а добитак од камате премашује ε. Једном када се пронађу занимљиви потпростори, кластери се могу идентификовати користећи исту методологију као CLIQUE или било који други постојећи алгоритам кластеровања. Као и

CLIQUE, ENCLUS захтева да корисник постави величину мреже, Δ. Такође, попут CLIQUE, алгоритам је веома осетљив на овај параметар.

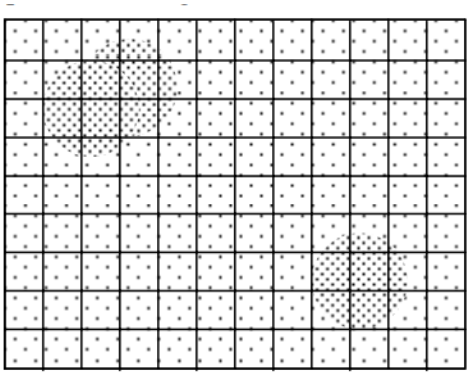
ENCLUS је у стању је да пронађе преклапајуће кластере произвољних облика у потпросторима различитих величина. ENCLUS има сличну временску сложеност као CLIQUE и нема лошу скалабилност у односу на димензије потпростора. Једно од побољшања које се предлаже је приступ одсецања, али заснован на ентропији. Ово би помогло да смањи број ирелевантних кластера у излазу, али би такође могло елиминисати мале, али смислене кластере. ENCLUS такође нуди велику флексибилност.

4.1.3. MAFIA

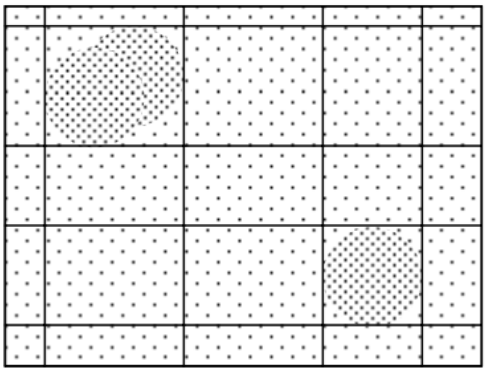
MAFIA је још једно проширење CLIQUE алгоритма које користи адаптивну мрежу засновану на дистрибуцији података како би се побољшала ефикасност и квалитет кластеровања. Алгоритам формира хистограм за одређивање минималног броја ћелија неке димензије, а затим комбинује суседне ћелије сличне густине да би се формирале веће ћелије. На овај начин, димензија је подељена на основу расподеле података и резултујућих граница ћелија које обухватају простор кластера прецизније него ћелије фиксне диманзије мреже. Када су ћелије дефинисане, MAFIA наставља слично као CLIQUE, користећи АПРИОРИ приступ генерише листу потпростора који се могу кластеровати посматрајући из једне димензије. MAFIA такође има за циљ да омогући паралелизацију процеса кластеровања [3].

Поред захтеваног параметра прага густине, алгоритам такође захтева од корисника да наведе критеријум за спајање суседних ћелија. Суседне ћелије у оквиру критеријума спајају се да би се формирале веће ћелије. Формирање адаптивне мреже зависи од ових ћелија. Такође, да би помогао алгоритму адаптивне мреже MAFIA узима подразумевану величину мреже као улаз за димензије где су подаци једнако распоређени. У таквим димензијама подаци су подељени на мали, фиксни број партиција. Упркос адаптивној природи мрежа, алгоритам је прилично осетљив на ове параметре.

Илустрације ради, CLIQUE би вероватније користио мрежу попут оне приказане на слици 8а, и на тај начин би разбио сваки од густих једнодимензионалних интервала у више подинтервала укључујући оне који су мање густине јер укључују део негустог региона. MAFIA започиње са великим бројем малих интервала за сваку димензију, а затим комбинује суседне интервале сличне густине да би на крају формирао мањи број већих интервала. Стога би мрежа MAFIA вероватно више личила на идеализовану мрежу приказану на слици 8б него на неоптималну мрежу са слике 8а.



Слика 8а. Приказ поделе кластера неоптималном мрежом



Слика 8б. Приказ поделе кластера идеализованом мрежом

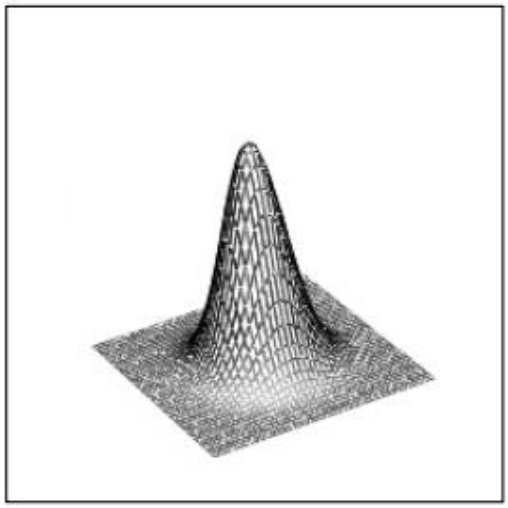
Као и сродне методе, MAFIA проналази произвољан број кластера произвољног облика у потпросторима различите величине. Такође представља кластере као ДНФ изразе. Због паралелизације и других побољшања, MAFIA се извршава брже од CLIQUE над сличним скуповима података. Линеарно се скалира бројем инстанци, а још боље бројем димензија у скупу података. Међутим, као и остали алгоритми у овој категорији, време извршавања експоненцијално расте са бројем димензија у кластерима.

4.1.4. DENCLUE

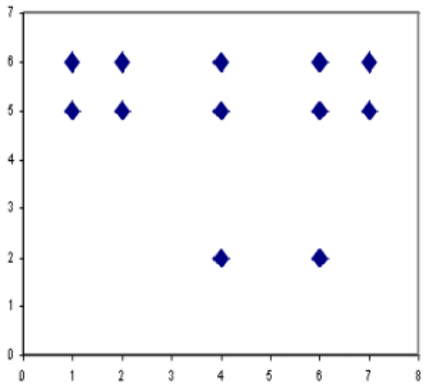
Овај алгоритам се може посматрати као уопштавање других алгоритама заснованих на густини. DENCLUE (DENsiti CLUstEring) је приступ кластеровању који укључује формалнији приступ заснован на густини, моделовањем укупне густине скупа тачака као збира функција повезаних са сваком тачком. Резултујућа функција укупне густине имаће локалне максимуме густине који се могу користити за директно дефинисање кластера. За сваку тачку података, растућа функција проналази њен најближи максимум, а скуп свих тачака података повезаних са одређеним максимумом постаје кластер. Међутим, ако је густина

на локалном максимуму мања од прага минималне густине, тада су тачке у придруженом кластеру класификоване као шум и оне се одбацују. Такође, ако се локални максимум може повезати са другим локалним максимумом путем тачака података, а густина у свакој тачки путање је изнад прага минималне густине, *x*, тада се кластери који су додељени овим локалним врховима спајају.

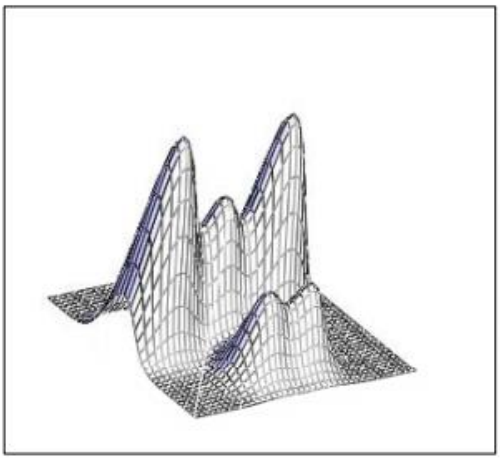
DENCLUE се заснива на добро развијеној области статистике и препознавању образаца познатије као „процена густине језгра“. Циљ процене густине језгра, као и многих других статистичких техника јесте да опише расподелу података помоћу функције. За процену густине језгра, допринос сваке тачке укупној функцији густине изражава се функцијом „утицаја“ (језгра). Укупна густина је тада збир функција утицаја сваке тачке. Функција утицаја и функција густине језгра су обично симетричне, и њихова вредност опада како се повећава удаљеност од тачке. На пример, за одређену тачку *к*, Гаусова функција, $\mathrm{K}(x) = \mathrm{e}^{\frac{-distance(x,y)^2}{2\sigma^2}}$ се често користи као функција језгра, где је *σ* параметар који одређује колико брзо опада утицај тачке. Слика 9а приказује како би Гаусова функција изгледала за једну дводимензионалну тачку, док слика 9в илуструје укупну функцију густине коју производе Гаусова функција примењена на скуп тачака које су приказане на слици 9б [3].



Слика 9а. Гаусова функција



Слика 9б. Скуп тачака



Слика 9в. Функција укупне густине

DENCLUE алгоритам има два корака, корак препроцесирања и корак кластеровања. У кораку препроцесирања, формира се мрежа за податке дељењем минималног граничног хиперправоугаоника у *d*-димензионалне хиперправоугаонике дужине ивице 2σ. Затим се одређују мрежне ћелије које садрже тачке. Ћелије мреже су нумерисане на једној ивици граничног хиперправоугаоника у односу на њихово порекло и ти кључеви се чувају у стаблу претраживања ради ефикаснијег приступа у каснијој обради. За сваку сачувану мрежну ћелију чува се број тачака, сума тачака у ћелији и везе са суседним ћелијама. За корак кластеровања DENCLUE узима у обзир само високонасељене мрежне ћелије и ћелије које су повезане са њима. За сваку тачку *k*, функција локалне густине израчунава се само узимајући у обзир оне тачке које су из мрежних ћелија које су у њеној близини. DENCLUE одбацује кластере чија је густина мања од ξ. Коначно, DENCLUE спаја густе ћелије које се могу спојити путем тачака које имају густину већу од ξ.

DENCLUE се може параметризовати тако да се понаша слично DBSCAN-у, али је много ефикаснији од DBSCAN-а. DENCLUE се такође може понашати као K-средина одабиром σ на одговарајући начин и изостављањем корака који спаја кластере дефинисане центром у кластере произвољног облика. Извођењем алгоритма за различите вредности σ, може се добити хијерархијско кластеровање.

4.1.5. OptiGrid

Упркос добрим карактеристикама DENCLUE-а у простору мале димензионалности, није ефикасан у случају повећане димензионалности или присуства шума. Исти истраживачи који су дефинисали DENCLUE дефинисали су и OptiGrid. Шум код високодиманзионалних података одговара једнолико распоређеним подацима који у мрежној ћелији постоје само као једна тачка. Више централизована расподела, попут Гаусове расподеле, доводи

до много више случајева када мрежна ћелија има више од једне тачке. Дакле, статистика о томе колико је ћелија вишеструко заузето може нам дати идеју о количини шума у подацима. Међусобне удаљености постају релативно уједначене са повећањем димензионалности, што значи да се максимална густина групе тачака може појавити у региону релативно празног простора, феномен познат као феномен празне тачке.

1. За сваку димензију
   i) Генерише се хистограм вредности података.
   ii) Одредити ниво шума. То се може постићи ручном претрагом хистограма. У супротном, преглед хистограма треба аутоматизовати.
   iii) Пронаћи крајњи леви и десни максимум, и максимуме к-1 између њих (к је број тражених кластера података).
   iv) Изабрати к минимума између максимума пронађених у претходном кораку. Те тачке представљају локације могућих резова тј. локације на којима би се могла поставити хиперраван за поделу података. Где је минимум даље десно од максимума са десне стране или даље лево од максимума са леве стране.
   v) Оценити сваки потенцијални рез, на пример, његовом густином.
2. Из свих димензија одабрати најбоље к резове, тј. резове најниже густине
3. Користећи ове резове, направити мрежу која дели податке
4. Пронаћи густе мрежне ћелије и додати их у листу кластера
5. Прецизирати листу кластера
6. Поновити кораке 1-5 за сваки кластер

OptiGrid тражи најбоље равни за сечење и формира мрежу која вероватно неће пресећи ниједан кластер, а потом проналази потенцијалне кластере у скупу мрежних ћелија и даље их дели ако је могуће. Са становишта ефикасности ово је много боље, али показало се да овај метод не функционише много боље од К-средина.
OptiGrid је сличан алгоритму MAFIA по томе што формира мрежу користећи партиционисање зависно од података. Међутим, за разлику од MAFIA и CLIQUE, не врши претрагу за најбољим потпростором. Стога је потребна додатна евалуација резултата овог алгоритма [3].

4.1.6. Cell-based Clustering Method (CBF)

Кластеровање засновано на ћелијама (CBF) има за циљ да реши проблеме скалабилности који се јављају код многих алгоритама претраживања који користе приступ одоздо према горе. Један од проблема алгоритама претраживања одоздо према горе је тај што се број формираних ћелија драстично повећава како се повећава број димензија. CBF алгоритам формира ћелије тако што формира оптималне партиције поновним испитивањем минималних и максималних вредности у датој димензији што резултује генерисањем мање количине ћелија [1].

CBF алгоритам је осетљив на два параметра. Праг секције одређује учесталост узорковања димензије. Време узорковања се смањује како се повећава гранична вредност јер је број објеката којима се приступа минимизован. Други параметар је праг ћелије који одређује минималну густину тачака података у ћелији. Групе са густином изнад овог прага одабране су као потенцијални кластери. Као и друге методе претраживања одоздо према горе, CBF је у стању да пронађе кластере различитих величина и облика. Такође се линеарно скалира са бројем инстанци у скупу података. Показало се да CBF има нешто нижу прецизност од CLIQUE, али је бржи у проналажењу и изградњи кластера.

4.1.7. CLTree

CLTree користи модификовани алгоритам стабла одлучивања за одабир најбољих равни одсецања за дати скуп података. Користи алгоритам стабла одлучивања за поделу сваке димензије у регионе, одвајајући области велике густине од области мале густине. CLTree следи стратегију одоздо према горе, процењујући сваку димензију посебно, а затим у следећим корацима користећи само оне димензије са областима велике густине. Пресеци стабла одлучивања одговарају границама области и бирају се на основу модификованих критеријума доделе који у суштини мере густину региона. Хипотетички, равномерно распоређени шум се додаје скупу података и стабло покушава да раздвоји прочитане податке од шума. Шум заправо не мора бити додат скупу података, већ се уместо тога може проценити густина за било који дати регион који се истражује.

Није потребно да се број и величина кластера задају као улазни параметри. Алгоритам захтева два улазна параметра.Први је *min_y* минимални број тачака које регион мора да садржи да би се током фазе одсецања сматрао значајним. Други је *min_rd* минимална релативна густина између два суседна региона пре него што се региони споје у већи кластер. Ови параметри се користе за одсецање стабла, а резултати су осетљиви на њихове промене. CLTree проналази кластере као хиперправоугаоне регионе у потпросторима различитих величина. Иако би изградња стабла могла бити временски сложена $O(n^2)$, алгоритам се линеарно скалира бројем инстанци и бројем димензија потпростора [1].

4.1.8. DOC (Density-based Optimal projective Clustering)

DOC је донекле хибридна метода која комбинује приступ заснован на мрежи који користи приступ одоздо према горе и метод итеративног побољшања приступа одозго према доле. DOC предлаже математичку дефиницију за појам оптималног пројективног кластера. Пројективни кластер је дефинисан као пар *(C, D)* где је *C* подскуп инстанци, а *D* подскуп димензија скупа података. Циљ је пронаћи пар где подскуп *C* има снажну тенденцију кластеровања у подскупу *D*. Да би пронашао ове оптималне парове, алгоритам случајним узорковањем формира мали подскуп *X*, који се назива дискриминишући скуп. У

идеалном случају, овај скуп се може користити за разликовање релевантних од небитних димензија кластера.

За дати пар кластера *(C, D)*, инстанце *p* из *C* и инстанце *q* из *X* требало би да важи следеће:

$$\forall i \in D, |q(i) - p(i)| \leq \omega$$

где је ω фиксна дужина странице потпросторног кластера, која се задаје унапред. *X* и *p* се добијају насумичним узорковањем и алгоритам се понавља са информацијом о најбољем резултату. Резултати у великој мери зависе од параметара. DOC такође захтева параметар α који одређује минималан број инстанци које могу формирати кластер, α заједно са ω одређују минималну густину кластера. Следећи параметар β је предвиђен за одређивање равнотеже између броја тачака и броја димензија у кластеру. Кластери су хиперправоугаоног облика и представљени су паром *(C, D)*, што значи да се скуп тачака података *C* пројектује на скуп димензија *D*. Време извршавања линеарно расте са бројем тачака података, али се експоненцијално повећава у зависности од број димензија у скупу података [1].

4.2. Итеративне методе претраживања потпростора одозго према доле

Приступ кластеровањем потпростора одозго према доле започиње проналажењем почетне апроксимације кластера у пуном простору карактеристика са димензијама једнаке тежине. Свакој следећој димензији се додељује тежина за сваки кластер. Ажуриране тежине се користе у следећој итерацији за поновно одређивање кластера. Овај приступ захтева вишеструке итерације скупих алгоритама кластеровања у пуном скупу димензија. Многе примене ове стратегије користе технику узорковања за побољшање перформанси. Алгоритми претраживања потпростора одозго према доле формирају кластере који су партиције скупа података, што значи да је свака инстанца додељена само једном кластеру. Многи алгоритми такође формирају кластер са елементима ван граница, елементима шума. Подешавање параметара је неопходно како би се добили значајни резултати. За овакве алгоритме најосетљивији параметри су број кластера и величина потпростора, које је често врло тешко утврдити унапред. Такође, с обзиром да је величина потпростора параметар, алгоритми претраживања потпростора одозго надоле теже проналазе кластере у истим или сличним потпросторима. За технике које користе узорковање, величина узорка је осетљив параметар и може играти велику улогу у квалитету коначних резултата.

Локалитет у методама одозго према доле одређује се приближном проценом груписања на основу тежина за до тада добијене димензије. PROCLUS, ORCLUS, FINDIT и δ-кластери одређују тежине инстанци за сваки кластер. COSA је јединствен по томе што користи к

најближих суседа за сваку инстанцу у скупу података како би одредио тежине за сваку димензију те инстанце.

4.2.1. PROCLUS

PROCLUS је био први алгоритам кластеровања потпростора одозго према доле. PROCLUS узоркује податке, затим бира скуп к медоида и итеративно побољшава кластеровање. Алгоритам користи приступ који се састоји од иницијализације, итерације и прочишћавања кластера. Иницијализација користи похлепни алгоритам за одабир скупа потенцијалних медоида који су удаљени један од другог. Циљ је да сваки кластер буде представљен бар једном инстанцом у изабраном скупу. Фаза итерације бира случајни скуп *k* медоида из овог редукованог скупа података, замењује лоше медоиде насумично одабраним новим медоидима и утврђује да ли се кластеровање побољшало. Квалитет кластера заснован је на просечној удаљености између инстанци и најближег медоида. За сваки медоид се бира скуп димензија чија су просечна растојања мала у поређењу са статистичким очекивањима. Укупан број димензија придружених медоидима мора бити *k***l*, где је *l* улазни параметар на основу којег се бира просечна димензионалност потпростора кластера. Једном када су изабрани потпростори за сваки медоид, користи се Менхетн растојање за додељивање тачака медоидима, формирајући кластере. Медоид кластера са најмањим бројем тачака се избацује заједно са било којим медоидима повезаним са мање од *(N/k)*minDeviation* тачака, где је *minDeviation* улазни параметар. Фаза пречишћавања израчунава нове димензије за сваки медоид на основу формираних кластера и додељује тачке медоидима, уклањајући тачке шума [1].

Као и многе методе претраживања потпростора одозго према доле, PROCLUS је пристрасан према кластерима хиперсферног облика. Такође, иако се кластери могу наћи у различитим потпросторима, потпростори морају бити сличних величина, јер се мора навести просечан број димензија за кластере. Кластери су представљени као скупови инстанци са повезаним медоидима и потпросторима и чине непреклапајуће партиције скупа података са могућим одступањима. Због употребе узорковања, PROCLUS је нешто бржи од CLIQUE на великим скуповима података. Међутим, коришћење малог броја репрезентативних тачака може проузроковати да PROCLUS у потпуности пропусти неке кластере. Квалитет кластера PROCLUS-а зависи од вредности улазних параметара које је често тешко одредити.

4.2.2. ORCLUS

ORCLUS је проширена верзија алгоритма PROCLUS који тражи потпросторе који нису осно паралелни. Овај алгоритам је произашао из запажања да многи скупови података садрже корелације међу атрибутима. Алгоритам се може поделити у три корака: додела кластера, одређивање потпростора и спајање.

Током фазе додељивања, алгоритам итеративно додељује тачке из скупа података најближим центрима кластера. Удаљеност између две тачке дефинисана је у потпростору *E*, где је *E* скуп ортонормалних (ортогоналних и нормализованих) вектора у неком *d*-димензионалном простору. Одређивање потпростора редефинише потпростор *E* повезан са сваким кластером израчунавајући коваријансне матрице за кластер и одабиром ортонормалних сопствених вектора са најмањим простирањем, односно са најмањим сопственим вредностима. Кластери који су близу један другог и имају сличне смерове ширења, спајају се током фазе спајања. Број кластера и величина димензионалности потпростора морају бити наведени. Одређује се општа шема за одабир одговарајуће вредности. Статистичка мера која се назива коефицијент прорећености кластера користи се након кластеровања како би се проценио квалитет избора димензионалности подпростора.

Алгоритам је рачунски захтеван углавном због израчунавања коваријансних матрица. ORCLUS користи насумично узорковање за побољшање брзине и скалабилности и као резултат може пропустити мање кластере. Крајње кластере дефинише центром кластера и придруженим партицијама скупа података, са могућим шумом.

4.2.3. FINDIT

FINDIT је по структури сличан PROCLUS-у и осталим алгоритмима претраживања потпростора одозго према доле, али користи јединствену меру удаљености која се назива димензионално растојање (енг. Dimension Oriented Distance, DOD). Алгоритам сабира број димензија на којима су две инстанце унутар граничне удаљености ε. Концепт се заснива на претпоставци да је у вишим димензијама смисленије да две инстанце буду блиске у неколико димензија. Алгоритам се обично састоји од три фазе, и то фазе узорковања, фазе формирања кластера и фазе додељивања података. Алгоритам започиње одабиром два мала скупа података генерисана насумичним узорковањем. Скупови се користе за одређивање почетних репрезентативних медоида кластера. У фази формирања кластера корелиране димензије се проналазе помоћу мере DOD за сваки медоид. FINDIT затим повећава вредност ε и понавља овај корак док се квалитет кластера не стабилизује. У завршној фази све инстанце су додељене медоидима на основу пронађених потпростора.

FINDIT захтева два улазна параметра, минимални број инстанци у кластеру и минималну удаљеност између два кластера. FINDIT је у могућности да пронађе кластере у потпросторима различите величине. Мера DOD зависи од ε прага који се одређује на итеративни начин. Итеративна фаза која одређује најбољу вредност ε праг значајно повећава време извршавања алгоритма. Због тога FINDIT користи технике узорковања као и други алгоритми претраживања потпростора одозго према доле. Узорковање помаже у побољшању перформанси, посебно код великих скупова података.

4.2.4. δ - Кластери

Алгоритам δ-кластера користи меру удаљености која покушава да одреди кохерентност коју приказује подскуп инстанци на подскупу атрибута. Кохерентне инстанце можда нису блиске у многим димензијама, али уместо тога прате сличан тренд, међусобну удаљеност. Једна кохерентна инстанца се може извести померањем друге инстанце за неко удаљење. Пиросонова корелација се користи за мерење кохерентности међу инстанцама. Алгоритам започиње иницијалним ситом тј. филтрирањем и итеративно побољшава укупан квалитет кластера насумичним мењањем атрибута и тачака података. Мера остатка се односи на смањење кохерентности које одређени атрибут или инстанца доноси кластеру. Алгоритам се завршава када се појединачна побољшања изједначе у сваком кластеру.

Алгоритам као параметре узима број кластера и појединачну величину кластера. Време извршавања је посебно осетљиво на параметар величине кластера. Ако се изабрана вредност много разликује од оптималне величине кластера, алгоритму може бити потребно знатно више времена да се заврши. Главна разлика између δ-кластера и других алгоритама је употреба кохеренције као мере сличности. Коришћење ове мере чини δ-кластере погодним за домене као што је анализа података микроталаса, проналазак сродних гена, тј. проналажење оних који слично реагују на услове околине.

4.2.5. COSA (Clustering On Subsets of Attributes)

COSA је итеративни алгоритам који додељује тежине свакој димензији за сваку инстанцу, а не за сваки кластер. Полазећи од тежински једнаких димензија, алгоритам испитује к најближих суседа (*knn*) сваке инстанце. Ова суседства се користе за израчунавање одговарајућих тежина димензија за сваку инстанцу. Веће тежине додељују се оним димензијама које имају мању дисперзију унутар *knn* групе. Те тежине се користе за израчунавање тежина димензија парова инстанци које се користе за ажурирање растојања коришћених у израчунавању *knn*. Поступак се понавља са новим растојањима док се тежине не уравнотеже. Околине сваке инстанце постају све богатије примерцима који припадају његовом кластеру. Тежине димензија су дефинисане као димензије кластера који има већу тежину. Као излаз се добија матрица удаљености заснована на тежинској инверзној експоненцијалној удаљености и погодна је као улаз за било који алгоритам кластеровања заснован на удаљености. Након кластеровања, упоређују се тежине сваке димензије инстанци кластера и израчунава се вредност укупног значаја за сваку димензију сваког кластера.

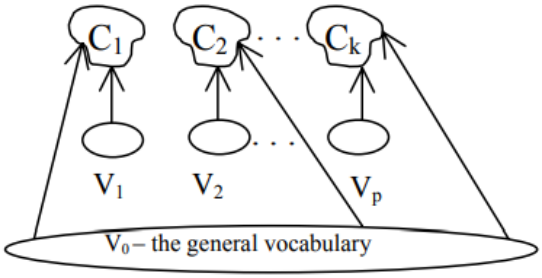
Број димензија у кластерима се не наводи директно, већ се број димензија множи параметром λ, који утиче на кластеровање на вишим димензијама. Сваки кластер може постојати у различитим потпросторима различитих величина, али имају тенденцију да буду сличне димензионалности.

4.3. Приступ „заснован на концепту” кластеровања података високе димензије

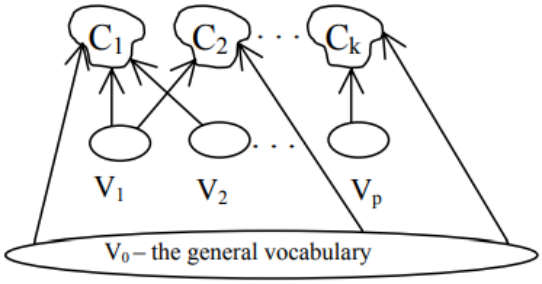
Кључна карактеристика неких високодимензионалних података је да два објекта могу бити веома слична, иако примењене мере растојања или сличности указују на то да су различите. Овај приступ дефинише сличност не у смислу заједничких атрибута, већ у смислу општег појма заједничких концепата.

Концепт простора представља скуп атрибута. На пример, за документе концепт би био скуп речи које карактеришу тему. Важност концепта је у томе што се за многе скупове података објекти у скупу података могу посматрати као генерисани из једног или више скупова концепата на пробабилистички начин. Дакле, концептуално оријентисан приступ докумената посматрао би сваки документ као да се састоји од речи које потичу из једног или више концепата, тј. скупа речи или речника, са вероватноћом да се свака реч одређује основним статистичким моделом. Скупове података са овом врстом структуре називамо концептним просторима, иако основни подаци могу бити представљени као тачке у векторском простору или у неком другом формату. Практична важност концептуалних простора је да се подаци који припадају концептуалним просторима морају третирати на различит начин у смислу како треба израчунати сличност између тачака и како објекте треба кластеровати.

Слика 10а приказује најједноставнији модел, који називамо моделом чистих концепата. У овом моделу, речи из документа *i*-те класе *Ci*, потичу или из општег речника V_0, или из тачно једног специјализованог речника V_1, V_2,..., V_p. Међутим, већа је вероватноћа да специјализована реч која се налази у документу потиче из специјализованог речника него из општег речника. За овај модел речници су само скупови речи и немају додатну структуру. У овом случају, као и у осталим размотреним случајевима, сви речници могу се преклапати. Слика 10б је врло слична слици 10а и приказује мало сложенији модел, који називамо модел вишеструких концепата. Једина разлика у односу на претходни модел је та што реч у документу из одређене класе може потицати из више специјализованих речника [3].



Слика 10а. Модел чистих концепата



Слика 10б. Модел вишеструких концепата

Статистички модел за горе представљене моделе засноване на концептима могао би бити следећи. Реч *v* у документу *d* из кластера C_i долази са једним или више речника са вероватноћом $P(v \mid C_i) = \sum P(v \mid V_j) * P(V_j \mid C_i)$. За чист концептуални модел, свака реч документа долази само из општег речника и једног од специјализованих речника. За модел вишеструких концепата, свака реч документа долази из једног или више специјализованих речника. Могући су и сложенији модели.

4.4. Алгоритам К-средина

Алгоритам К-средина је једноставан и заступљен алгоритам кластеровања. Основна идеја је пронаћи к кластера тако да су објекти унутар сваког кластера слични једни другима и различити од објеката у другим кластерима. К-средина је итеративан алгоритам у којем је дефинисан почетни скуп кластера. Кластери се у свакој итерацији ажурирају све док више није могуће побољшање или док број итерација не пређе задату границу [5].

4.4.1. Скалирање опсега поља

У већини скупова података постоји велика варијабилност у опсегу домена поља. На пример, ако се узме у обзир старост и број аутомобила по домаћинству, у зависности од популације која је од интереса, старост може имати вредности до 80 година или чак и више. Међутим, мало је вероватно да ће вредности броја аутомобила по домаћинству премашити три или четири у великој већини случајева. Ако се користе оба ова поља у њиховој природној скали као улаз за модел, пољу старости ће се вероватно дати много већа тежина у моделу него броју аутомобила по домаћинству, једноставно због опсега вредности (па самим тим и због разлика између објеката) јер је први много пута већи од другог.

Да би се надокнадио овај ефекат скале, поља домена се трансформишу тако да сва имају исту скалу. Домени поља се мењају тако да имају вредности између 0 и 1. Коришћена је

трансформација: $x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$, где је x_i' промењена вредност поља за унос x за објекат i, x_i је оригинална вредност x објекта i, x_{min} је минимална вредност x за све објекте, а x_{max} је највећа вредност x за све објекте.

4.4.2. Параметри модела

Примарни део у алгоритму к-средина је итеративни процес израчунавања центара кластера и додељивања објеката кластерима. Примарни кораци у поступку су:

1. Одабрати почетне центре кластера

2. Доделити сваки објекат најближем кластеру

3. Ажурирати центре кластера на основу објеката додељених сваком кластеру

Понављајти кораке 2 и 3 док у кораку 3 нема промена за центре кластера у односу на претходну итерацију, или уколико број итерација прелази параметар максималног броја итерација. Кластере дефинишу њихови центри. Центар кластера је вектор вредности за улазна поља. Векторске вредности су засноване на средњим вредностима за објекте додељене кластеру [6].

Почетни центри кластера бирају се помоћу maxmin алгоритма:

1.Иницијализовати први центар кластера као вредности улазних поља за први упис.

2. За сваки објекат података израчунати минималну удаљеност између објекта и сваког дефинисаног центра кластера.

3. Изабрати објекат са највећом минималном удаљеношћу од дефинисаних центара кластера. Додати нови центар кластера са вредностима улазних поља за изабрани објект.

4. Понављати кораке 2 и 3 док се k-ти центар кластера не дода у модел.

Након што су изабрани почетни центри кластера, алгоритам започиње итеративни процес додељивања тј. ажурирања.

У свакој итерацији алгоритма, сваки објекат је додељен оном кластеру чији центар му је најближи. За мере растојања се користи Еуклидско растојање

$$d_{ij} = ||X_i - C_i||^2 = \sum_{q=1}^{Q} \left(x_{qi} - c_{qj}\right)^2,$$

где је X_i вектор улазних поља за објекат i, C_j је вектор центра кластера за кластер j, Q је број улазних поља, x_{qi} је вредност q-тог улазног поља за i-ти објекат, а c_{qj} је вредност q-тог улазног поља за j-ти објекат. За сваки објекат израчунава се удаљеност између објекта и сваког центра кластера. Након што су објекти додељени најближим кластерима, центри

кластера се ажурирају. Центар кластера израчунава се као средњи вектор објеката који су додељени кластеру

$C_j = \underline{X_j}$,

где се вредности средњег вектора $\underline{X_j}$ израчунавају на уобичајен начин

$\overline{x_{qJ}} = \frac{\sum_{i=1}^{n_j} x_{qi(j)}}{n_j}$,

где је n_j број објеката у кластеру *j*, $x_{qi}(j)$ је *q*-та улазна вредност поља за објекат *i* која је додељена кластеру *j*.

4.4.3. Услови заустављања модела

Постоји неколико фактора који утичу на начин на који се израчунавају модели.

Maxmin итерације: параметар максималних итерација контролише колико дуго ће алгоритам тражити стабилно решење. Алгоритам ће поновити циклус ажурирања не више од наведеног броја пута. Ако и када се ово ограничење достигне, алгоритам се завршава и тренутни скуп кластера проглашава као коначни.

Толеранција грешке: параметар толеранције грешке пружа још један начин контроле колико дуго ће алгоритам тражити стабилно решење. Максимална промена средњих вредности кластера за итерацију *t* израчунава се као:

$\max_J \left| C_j(t) - C_j(t-1) \right| \vee$,

где је $C_j(t)$ вектор центра кластера за *j*-ти кластер итерације *t* и $C_j(t\text{-}1)$ је вектор центра кластера на претходној итерацији. Ако је максимална промена мања од наведене толеранције за тренутну итерацију, алгоритам се завршава и тренутни скуп кластера проглашава као коначни модел.

Вредност кодирања за параметре скупова контролише релативно одређивање постављених поља у алгоритму К-средина. Подразумевана вредност додељује једнак тежински коефицијент пољу опсега и постављеним пољима. Да би се јаче нагласила постављена поља, може се поставити вредност кодирања ближе 1; да би се више нагласила поља опсега, поставити вредност кодирања ближе 0.

Удаљеност до кластера израчунава се као еуклидско растојање између центара кластера:

$$d_{ij} = \left|\left| C_i - C_j \right|\right| = \sqrt{\sum_{q=1}^{Q} (c_{qi} - c_{qj})^2}$$

4.5. TwoStep алгоритам кластероваања

TwoStep је скалабилни алгоритам кластеровања дизајниран за руковање великим скуповима података. Може да обради непрекидне и категоричке атрибутима. Као што назив имплицира, алгоритам кластеровања TwoStep укључује два корака: прекластеровање-груписање објеката у више малих подкластера и кластеровање-груписање подкластера, који су резултат претходног корака, у жељени број кластера. Потребан је само један пренос података. Такође може аутоматски изабрати број кластера [6].

4.5.1. Прекластеровање

Корак прекластеровања користи приступ секвенцијалног груписања. Он разматра један по један објекат и одлучује да ли тренутни објекат треба спојити са претходно формираним кластерима или направити нови кластер на основу критеријума удаљености. Поступак се имплементира конструисањем стабла модификованих карактеристика кластера (ЦФ). ЦФ стабло се састоји од више нивоа чворова, а сваки чвор садржи бројне уносе. Унос тј. вредност која треба да се унесе у лист представља финални подкластер. Уноси чворова који нису листови се користе за брзо смештање новог објекта у исправан чвор листа. Сваки унос има особине ЦФ стабла које садржи број објеката уноса, средњих вредности и варијансе сваког опсега поља и рачуна се за сваку категорију сваког симболичког поља. Сваки објекат, почевши од коренског чвора, се спушта дуж ЦФ стабла, рекурзивно по нивоима кроз чворове до одговарајућег листа. Када стигне до листа, проналази најближи унос листа у чвору листа. Ако је објекат унутар граничне удаљености од најближег уноса листа, он се додаје и ЦФ тог уноса листа се ажурира. У супротном, започиње властити унос листа у чвору листа. Ако у чвору листа нема простора за формирање новог уноса листа, чвор листа се дели на два. Уноси у оригиналном чвору листа су подељени у две групе користећи најудаљенији пар као сито тј. филтер и прерасподељујући преостале уносе на основу критеријума удаљености.

Ако ЦФ стабло расте изнад дозвољене максималне величине, стабло се поново гради на основу постојећег стабла повећањем критеријума прага удаљености. Обновљено стабло је мање и стога има простора за нове објекте. Овај процес се наставља док се не обраде сви подаци. Сви објекти који спадају у исти унос могу се представити ЦФ уносом. Када се нови објекат додаје уносу, ново ЦФ стабло се може израчунати на основу новог објекта и старог ЦФ стабла без познавања појединачних објеката у уносу. Ова својства омогућавају чување само улаза ЦФ стабла, а не скупова појединачних објеката. Стога је ЦФ стабло много мање од оригиналних података и може се ефикасније складиштити у меморији. Структура изграђеног ЦФ стабла може зависити од редоследа објеката. Да би се смањио ефекат уређења, треба насумично изабрати објекте пре изградње модела.

4.5.2. Кластеровање

Корак кластеровања узима подкластере који су резултат корака предкластеровања као улаз, а затим их групише у жељени број кластера. Пошто је број подкластера много мањи од броја оригиналних података, традиционалне методе кластеровања се могу ефикасно користити. TwoStep користи сакупљајућу хијерархијску методу кластеровања, јер добро функционише са методом аутокластеровања. Хијерархијско кластеровање се односи на процес у коме се кластери рекурзивно спајају, све док на крају процеса не остане само један кластер који садржи све објекте. Процес започиње дефинисањем почетних кластера за сваки од подкластера добијених у кораку предкластеровања. Затим се упоређују сви кластери, а пар кластера са најмањом удаљеношћу се бира и спаја у један. Након спајања, нови скуп кластера се упоређује, најближи пар се спаја и процес се понавља све док се сви не споје. Пошто се кластери рекурзивно спајају на овај начин, лако је упоредити решења са различитим бројем кластера. Да би се добило решење са к кластера, стаје се са спајањем када преостане к кластера. Да би се добило решење са к-1 кластера, треба узети решење са к кластера и извршити још један корак спајања, итд.

4.5.3. Мера растојања

TwoStep користи меру растојања по лог-вероватноћи за прилагођавање симболичких поља и поља опсега. То растојање је засновано на вероватноћи. Спајање два кластера у један зависи од растојања израженог помоћу лог-вероватноће. При израчунавању лог-вероватноће, претпостављају се нормалне расподеле за поља опсега и мултиномијалне расподеле за симболичка поља. Такође се претпоставља да су поља независна једно од другог, па тако и објекти. Растојање између група *i* и *j* дефинисано је као

$$d(i,j) = \xi_i + \xi_j - \xi_{(i,j)},$$

где је

$$\xi_v = -N_v\left(\sum_{k=1}^{K^A} \frac{1}{2}\log(\sigma_k^2 + \sigma_{vk}^2) + \sum_{k=1}^{K^B} E_{vk}\right) \text{ и } E_{vk} = -\sum_{l=1}^{L_k} \frac{N_{vkl}}{N_v}\log\frac{N_{vkl}}{Nv}.$$

У овим изразима K^A је број поља за унос типа опсега, K^B је број поља за унос симболичког типа, L_k је број категорија за *k*-то симболично поље, N_v је број објеката у кластеру *v*, N_{vkl} је број објеката у кластеру *v* који припада *l*-тој категорији *k*-тог симболичког поља, σ_k^2 је процењена варијанса *k*-те непрекидне променљиве за све објекте, σ_{vk}^2 је процењена варијанса *k*-те непрекидне променљиве за објекте у *v*-том кластеру и <*i*,*j*> је индекс који представља кластер формиран комбиновањем кластера *i* и *j*.

Ако σ_k^2 се занемари у изразу за ξ_v, растојање између кластера *i* и *j* би било управо смањење лог-вероватноће када се та два кластера комбинују.

4.5.4. Одређивање броја кластера

TwoStep може користити хијерархијску методу кластеровања у другом кораку за процену решења више кластера и аутоматско одређивање оптималног броја кластера за улазне податке. Карактеристика хијерархијског кластеровања је да производи низ партиција у једном покретању. Насупрот томе, алгоритам к-средина би морао да се покрене више пута, по једном за сваки наведени број кластера, да би генерисао секвенцу. Да би аутоматски одредио број кластера, TwoStep користи процедуру из две фазе које добро функционишу са хијерархијском методом кластеровања.

У првој фази се израчунава Бајесов информациони критеријум (енг. Bayesian Information Criterion BIC) за сваки број кластера унутар наведеног опсега и користи се за проналажење почетне процене броја кластера. *BIC* се рачуна као:

$$BIC(J) = -2\sum_{j=1}^{J}\xi_j + m_J \log(N),$$

где је

$$m_J = J\left\{2K^A + \sum_{k=1}^{K^B}(L_K - 1)\right\}$$

Однос промене *BIC*-а при сваком узастопном спајању у односу на прво спајање одређује почетну процену. Нека је *dBIC(J)* разлика *BIC*-а између модела са *J* кластера и са *(J+1)* кластера, $dBIC(J) = BIC(J) - BIC(J+1)$. Тада је однос промене за модел *J* једнак $R_1(J) = \frac{dBIC(J)}{dBIC(1)}$.

Ако је $dBIC(1) < 0$, тада је број кластера једнак јединици, а друга фаза је изостављена. Иначе, почетна процена броја кластера k је најмањи број за који важи $R_1(J) < 0.04$.

У другој фази, почетна процена се побољшава проналажењем највећег релативног повећања удаљености између два најближа кластера у свакој фази хијерархијског кластеровања. Побољшање се врши на следећи начин:

- Полазећи од модела C_k означеног *BIC* критеријумом, узети однос минималне удаљености међу кластерима за тај модел и следећи већи модел C_{k+1}, односно претходни модел у поступку хијерархијског кластеровања,

  $$R_2(k) = \frac{d_{min}(C_k)}{d_{min}(C_{k+1})}$$

  где је C_k модел који садржи *k* кластера, а $d_{min}(C_k)$ је минимална удаљеност између кластера за модел *C*.
- Сада из модела C_{k+1} израчунати исти однос са следећим моделом C_k, као у претходном кораку. Понављати за сваки следећи модел док се не добије однос $R_2(2)$.
- Упоредити два највећа односа R_2, ако је највећи већи 1.15 пута од следећег највећег, онда изабрати модел са највећим односом R_2 као оптималан број

кластера. У супротном, из та два модела са највећим вредностима R_2, изаберите онај са већим бројем кластера као оптималан модел.

4.5.5. Руковање недостајућим и одступајућим вредностима

TwoStep не подржава празна поља. Објекти који садрже празнине, нуле или недостајуће вредности било које врсте искључени су из изградње модела.

Руковање одступајућим вредностима је опциони корак имплементиран у алгоритам у процесу изградње ЦФ стабла. Одступањима се сматрају објекти који се не уклапају добро у било који кластер. Сматра се да су објекти у уносу листа одступања ако је број објеката у уносу мањи од одређеног дела величине највећег уноса листа у стаблу ЦФ. Пре поновне изградње стабла ЦФ, проверава се да ли постоје потенцијалне одступајуће вредности и ако постоје остављају се по страни. Након поновне изградње ЦФ стабла, проверава се да ли се те одступајуће вредности могу уклопити без повећања величине стабла. На крају изградње ЦФ стабла, мали уноси који се не могу уклопити су одступајуће вредности.

4.5.6. Предвиђене вредности

Приликом бодовања објекта помоћу TwoStep алгоритма, објекат се додељује кластеру коме је најближи. Рачуна се растојање између објекта и сваког кластера, а кластер са најмањом удаљеношћу изабран је као најближи. Том кластеру се додељује предвиђена вредност објекта. Растојање се израчунава на сличан начин као при изградњи модела, имајући у виду да се објекат посматра као кластер са само једним објектом. Ако је руковање одступајућим вредностима модела било омогућено током изградње, растојање између објекта и најближег скупа се упоређује са прагом $C = log(V)$, где је

$$V = \prod_k R_k \prod_m L_m.$$

а R_k је опсег k-тог нумеричког поља и L_m је број категорија за m-то симболично поље. Ако је удаљеност од најближег кластера мања од C, доделити том кластеру предвиђену вредност. Ако је растојање веће од C, објекат прогласити као одступајућу вредност.

4.6. CLUTO

CLUTO је софтверски пакет за кластеровање скупова података малих и великих димензија и за анализу карактеристика различитих кластера [8].

CLUTO пружа три различите класе алгоритама кластеровања који раде или директно у простору карактеристика објекта или у простору сличности објекта. Ови алгоритми су засновани на парадигмама партиционисања, агломерације и партиционисања графа.

Кључна карактеристика већине CLUTO алгоритама за кластеровање јесте да они третирају проблем кластеровања као процес оптимизације који настоји да максимизује или минимизује одређену функцију критеријума кластеровања дефинисану било глобално или локално у целом простору решења. CLUTO пружа укупно седам различитих критеријумских функција (Табела 1.) које се могу користити за покретање и партиционих и агломеративних алгоритама за кластеровање. Показало се да већина ових функција критеријума производи висококвалитетна решења за кластеровање у скуповима података високе димензијоналности, посебно онима који настају у кластеровању докумената. Поред ових функција критеријума, CLUTO пружа неке од традиционалнијих локалних критеријума који се могу користити у контексту агломеративног кластеровања. Штавише, CLUTO пружа алгоритме кластеровања засноване на партиционисању графова који су веома погодни за проналажење кластера који формирају граничне регионе који обухватају различите димензије основног простора.

Важан аспект алгоритама кластеровања заснованих на партицији је метод који се користи за оптимизацију функције критеријума. CLUTO користи насумични инкрементални алгоритам оптимизације који је похлепан по природи, није рачунски захтеван, и показало се да производи висококвалитетна решења за кластеровање. CLUTO алгоритми за кластеровање засновани на партиционисању графова користе ефикасне алгоритме на више нивоа изведене из METIS и hMETIS, алгоритама за партиционисање графова и хиперграфова. CLUTO такође пружа алате за анализу пронађених кластера да би се разумели односи између објеката додељених сваком кластеру и релације између различитих кластера, као и алате за визуелизацију откривених кластера. CLUTO може да идентификује карактеристике које најбоље описују и/или праве разлику између кластера. Овај скуп карактеристика се може користити за боље разумевање скупа објеката додељених сваком кластеру и за пружање сажетих резимеа о садржају кластера.

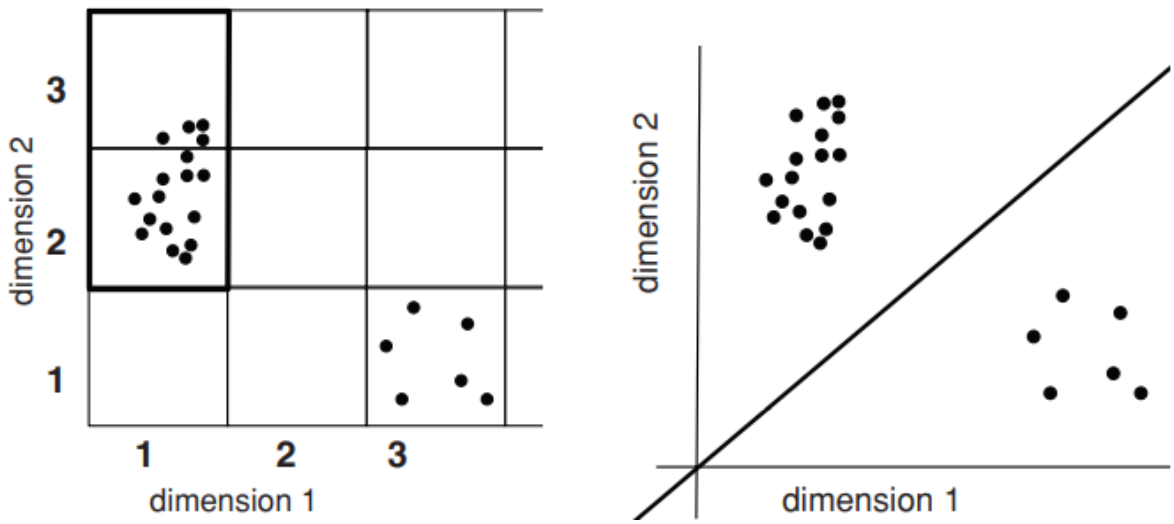
CLUTO-ови алгоритми су оптимизовани за рад на високодимензионалним скуповима података како у смислу броја објеката тако и броја димензија. Ови алгоритми могу брзо да групишу скупове података са неколико десетина хиљада објеката и неколико хиљада димензија. Штавише, пошто је већина високодимензионалних скупова података веома ретка, CLUTO директно узима у обзир ову особину и захтева меморију која је отприлике линеарна у односу на величину улаза. CLUTO алгоритам расподеле се састоји од самосталних програма (vcluster и scluster) за кластеровање и анализу ових кластера, као и од библиотеке преко које програм може директно приступити различитим алгоритмима за кластеровање и анализу имплементираних у CLUTO-у.

Criterion Function	Optimization Function	
$\mathcal{I}_1$	maximize $\sum_{i=1}^{k} \frac{1}{n_i} \left(\sum_{v,u \in S_i} \text{sim}(v,u) \right)$	(1)
$\mathcal{I}_2$	maximize $\sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} \text{sim}(v,u)}$	(2)
$\mathcal{E}_1$	minimize $\sum_{i=1}^{k} n_i \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sqrt{\sum_{v,u \in S_i} \text{sim}(v,u)}}$	(3)
$\mathcal{G}_1$	minimize $\sum_{i=1}^{k} \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sum_{v,u \in S_i} \text{sim}(v,u)}$	(4)
$\mathcal{G}'_1$	minimize $\sum_{i=1}^{k} n_i^2 \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sum_{v,u \in S_i} \text{sim}(v,u)}$	(5)
$\mathcal{H}_1$	maximize $\frac{\mathcal{I}_1}{\mathcal{E}_1}$	(6)
$\mathcal{H}_2$	maximize $\frac{\mathcal{I}_2}{\mathcal{E}_1}$	(7)

Табела 1: Математичка дефиниција CLUTO-ових функција критеријума кластеровања. Ознаке у овим једначинама су следеће: k је укупан број кластера, S је укупан број објеката који се групишу, Si је скуп објеката додељених i-ом кластеру, ni је број објеката у i-ом кластеру, v и u представљају два објекта, а sim(v, u) је сличност између два објекта.

5. Кластеровање у високодимензионалним просторима коришћењем мрежа и хиперравни

Неки приступи избегавају проблеме кластеровања заснованих на густини или кластеровања која користе најближе суседе коришћењем приступа заснованог на мрежи. Општа идеја је да се високодимензионални простор података подели на већи број ћелија и да се трага за кластерима на основу густине ћелија. Примери укључују WaveCluster, STING (статистичка информативна мрежа) и DENCLUE (груписање засновано на густини). Проблем већине ових приступа је број ћелија који експоненцијално расте са повећањем броја димензија. Чак и за алгоритме као што је DENCLUE који чува само густе ћелије, високодимензионални простори представљају проблеме у погледу откривања значајних својстава кластера као што је густина ћелија. Алгоритми као што је DENCLUE нису способни да поуздано открију кластере. У светлу проклетства димензионалности, простор података постаје све оскуднији, што онемогућава поуздано мерење густине у готово празним просторима. OptiGrid алгоритам проширује идеју DENCLUE за проблеме високодимензионалних простора. Идеја је да се изгради вишедимензионална мрежа дефинисана хиперравнима која пресеца простор релативно мале густине и која одваја објекте података у регионе. За једноставнију илустрацију опште идеје, дат је пример у две димензије на слици 11. [2]



Слика 11. Приказ одвајања потпростора помоћу мрезе и хеперравни

Традиционалне решетке одвајају сваку димензију у одређени број суседних интервала чији пресеци чине ћелије високе димензије. Потребно је материјализовати само насељене ћелије, а суседне ћелије попут ћелија (1, 2) и (1, 3) на слици лево могу се спојити. Коришћењем хиперравни, подаци се идеално одвајају користећи релативно мало

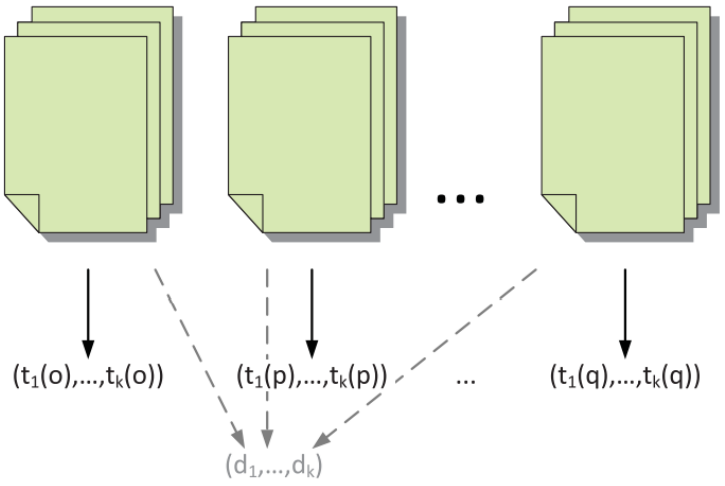
пресека кроз ретке области. Алгоритамски, то се постиже рекурзивним избором хиперравни ортогоналних барем на једну од пројекција које су укључене у процес кластеровања. Показано је да осно паралелне пројекције дају мању грешку за високодимензионалне просторе, што хиперравни чини корисним алатом за велике димензије. Кластеровање ортогоналних партиција проширује ову методу на рад са ограниченом главном меморијом и једним чувањем података. Алгоритам користи активно узорковање и осно паралелно партиционисање на основу статистичких тестова квалитета хиперравни.

6. Кластеровање високодимензионалних докумената

Документи могу бити представљени речима које садрже или подскупом препознатљивих речи. Фреквенција речи у сваком документу даје високу димензију вектору карактеристика. Најчешће се текстуални документи приликом кластеровања представљају коришћењем високодимензионалног вектора фреквенције појмова и инверзне фреквенције докумената (tf-idf). Ове вредности указују на то колико су појмови важни за одређени документ, колико су чести и колико је појам користан у разликовању докумената. Што мање докумената садржи тај термин, то је препознатљивији. Сваки објекат документа *O* пресликава се у tf-idf вектор

$$v(o) = \left(t_1(o) * log\left(\frac{n}{d_1}\right), \dots, t_k(o) * log\left(\frac{n}{d_k}\right) \right)$$

где *n* означава број докумената у скупу података, t_i је фреквенција *i*-тог члана у документу, а d_i фреквенција *i*-тог члана. На слици 12. је илустрована фреквенција појма која се израчунава по документу, док се обрнута фреквенција докумената израчунава у читавом скупу докумената. Како је број термина обично велик, посебно у различитим скуповима текстова, кластеровање докумената засновано на резултујућим векторима одвија се суштински у високодимензионалном окружењу [2].



Слика 12. Илустрација фреквеција појма по документу

Да би се решили проблеми кластеровања заснована директно на tf-idf векторима, због њихове велике димензионалности идеја је упоредити документе у пару, користећи косинусну сличност tf-idf вредности, и употребљавати ову сличност за одређивање веза између докумената. Косинусна сличност између два вектора v_1, v_2 дефинише се као

$cos(v1, v2) = \frac{v1 \bullet v2}{|v1| * v2V}$. Јачина везе одговара броју најближих суседа. У кластеровање су укључени само документи који имају минималну јачину везе, а они који имају јаке везе спојени су у исти кластер.

Приступи заједничког кластера идентификују кластере у подскуповима речи и подскуповима докумената истовремено, и стога су слични кластеровању потпростора или пројектованом кластеровању.

Због преклапања речи у различитим документима, проналажење доброг пресека може бити тежак задатак. Циљ алгоритама заснованих на језгру, као што је језгро к-средина, је пресликавање вектора карактеристика у високодимензионалном простору помоћу функције језгра (нпр. Гаусово језгро). У анализи докумената, косинусна функција је типичан избор. Алгоритми засновани на језгру се такође могу користити за иницијализацију тежина у методама спектралног кластеровања заснованих на графу.

7. Практични примери

Примена неких од претходно наведених алгоритама кластеровања биће илустрована на подацима који садрже експресију гена из хуманих ћелија периферне крви са једним једром (Peripheral Blood Mononuclear Cells, PBMC) ПБМЦ ћелије садрже пет главних типова ћелија [7]:

- дендритске ћелије (dendritic cells, DC)
- моноците (monocytes, MC)
- Б ћелије (B cells, BC)
- ћелије "природне убице" (natural killer cells, NK), и
- Т ћелије (T cells, TC)

При томе се ћелије неких од типова могу додатно груписати у подтипове. Коректно препознавање типова ћелија је од великог значаја у разумевању биолошких процеса у здравим ткивима и за дијагностиковање различитих медицинских стања. За илустрацију алгоритама кластеровања коришћен је материјал расположив на адреси http://projects.met-hilab.org/SCTdata/PBMC001. Све расположиве датотеке су спојене у једну уз претходну промену (маскирање ознака подтипова). Улазни подаци су вискодимензиони - сваки од 13299 редова садржи 30698 (целобројних) вредности које одговарају атрибутима (издвојеним генима) у посматраним ћелијама. Ови атрибути представљају ниво експресије гена. Изглед улазних података је

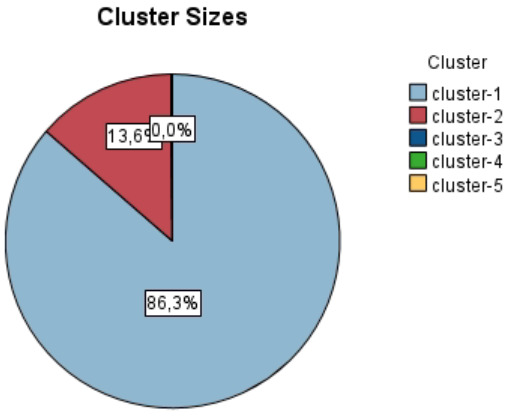
```
1       0       0       0       0       0       0       0       0       0       0       0
...     0       0       3       0       0       0       0       0       0       0       0
0       ...     0       0       0       0       0       0       0       0       0       0
0       0       ...     0       0       0       0       0       0       0       0       0
0       0       3       ...     0       0       0       0       0       0       0       0
0       0       0       0       ...     0       0       0       0       0       0       6
0       0       0       0       0       ...     0       0       0       0       0       0
0       0       0       0       0       0       ...     13      0       0       0       0
0       0       7       0       0       0       0       ...     0       0       0       0
0       0       0       0       0       4       0       0       ...     0       0       0
0       0       0       0       0       0       0       0       0       ...     0       0
0       0       0       0       0       0       0       0       0       0       ...     0
2       0       0       0       6       0       0       0       0       0       0       ...
3       0       0       0       0       0       0       0       0       0       0       0
...     0       0       0       0       0       0       0       0       0       0       0
0       ...
```

Поред вредности атрибута, подаци садрже (у посебној датотеци, овде нису наведени) идентификације типа ћелија које могу да послуже за спољашњу проверу квалитета кластеровања. У наредној табели су наведени (под) типови ћелија, ознаке за класу и њихов број у улазном скупу података. Ћелије су подељене у 15 класа, према подтиповима којима припадају.

Тип ћелије	**Ознака класе у материјалу**	**Број ћелија**
ambiguous	Klasa_1	116
aTreg	Klasa_2	921
DC	Klasa_3	142
M14	Klasa_4	1263
M16	Klasa_5	398
MemoryBcell	Klasa_6	491
NaiveBcell	Klasa_7	1169
NK	Klasa_8	1394
Non-classical T	Klasa_9	1431
rTreg	Klasa_10	1072
T-nonT	Klasa_11	426
T4eff.mem	Klasa_12	975
T4naive	Klasa_13	1134
T8eff.mem	Klasa_14	1031
T8naive	Klasa_15	1336

7.1. Примена К-средина и TwoStep алгоритма

На слици 13 (а и б) је приказан резултат примене К-средина за 5 кластера. Добијена је лоша подела јер је алгоритам формирао само два кластера.



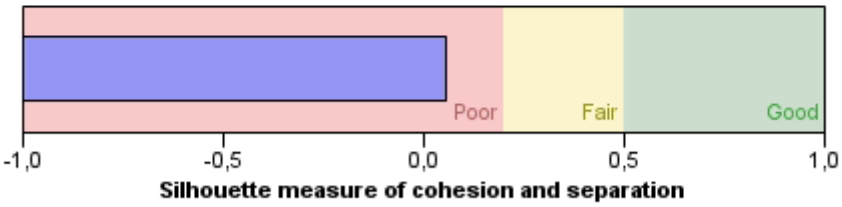
Size of Smallest Cluster	1 (0%)
Size of Largest Cluster	11476 (86,3%)
Ratio of Sizes: Largest Cluster to Smallest Cluster	11.476,00

Слика 13а. Резултати примене К-средина за добијање 5 кластера

Model Summary

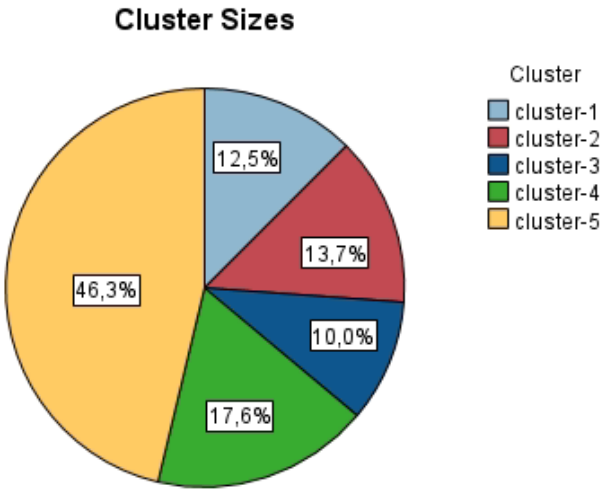
Algorithm	K-Means
Inputs	30698
Clusters	5

Cluster Quality



Слика 13б.

На слици 14 (а и б) је приказан резултат примене TwoStep алгоритма за 5 кластера. Овај алгоритам је дао бољи резултат, поделио је податке на 5 кластера и силуета је око 0 што значи да је подела релативно добра.



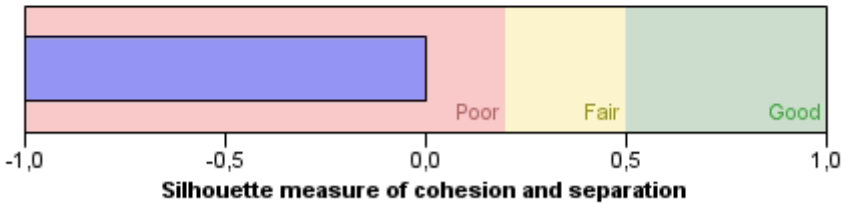
Size of Smallest Cluster	1324 (10%)
Size of Largest Cluster	6151 (46,3%)
Ratio of Sizes: Largest Cluster to Smallest Cluster	4,65

Слика 14а. Резултати примене TwoStep алгоритма за добијање 5 кластера

Model Summary

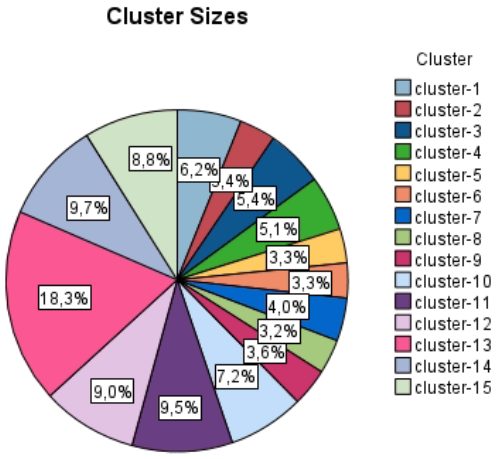
Algorithm	TwoStep
Inputs	20224
Clusters	5

Cluster Quality



Слика 14б.

На слици 15 (а и б) је приказан резултат примене TwoStep алгоритма за 15 до 30 кластера. Као резултат добијено је 15 кластера, такође са силуетом 0.



Size of Smallest Cluster	428 (3,2%)
Size of Largest Cluster	2429 (18,3%)
Ratio of Sizes: Largest Cluster to Smallest Cluster	5,68

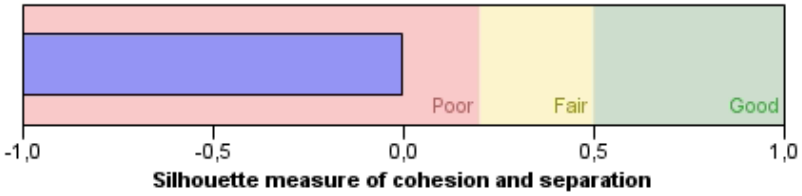
Слика 15а. Резултати примене TwoStep алгоритма за добијање 15 кластера

Последња два примера дају резултате који се даље могу користити за нека детаљнија испитивања. Такође могу се користити и као екстерни критеријуми квалитета кластеровања.

Model Summary

Algorithm	TwoStep
Inputs	20224
Clusters	15

Cluster Quality



Слика 15б.

7.2. Примена алгоритама у CLUTO алату

Резултати приказани на слици 16. добијени су применом CLUTO алгортима за 15 кластера, користећи $\mathcal{I}_1$ функцију из Табеле 1. Косинусна функција је коришћена као функција сличности. Алгоритам даје добре резултате за неке од класа у улазном скупу. За прву (CLUTO) класу (cid=0) видимо да је чистоћа максимална тј. једнака 1, а ентропија минимална што значи да је класа (*ambiguous* ћелије) добро одређена. По квалитету иза ње са чистоћом 0.827 је (CLUTO) класа 2 (cid=2), односно иницијална класа 3 која садржи дендритске ћелије (DC), па затим CLUTO класа 6 (cid=5) са подацима из иницијалних класа 4 и 5 која садржи сродне M14 и M16 ћелије, итд. Даљом анализом се види да су релативно лошији резултати добијени за иницијалне класе ћелија које су сродне (нпр. групе ћелија које припадају T надгрупи - rTreg, T-noT, T4eff.mem, T4.naive, T8eff.mem, T8.naive), што упућује на то да за коретно кластеровање ове групе ћелија треба издвојити у посебан улазни скуп и на њих поново применити алгоритам. При одређивању чистоће поједине класе коришћен је екстерни критеријум кластеровања на основу задатаих информација у припадности одређеној класи сваког појединачног слога у улазном скупу.

У даљим примерима су приказани резултати кластеровања истог скупа за различите функције растојања и различит број кластера. У случају кластеровања са 5 кластера (раздвајање по групама DC, MC, BC, NK, TC) из резултата се види да неке од функција

растојања дају квалитетније резултате и успевају да раздвоје неке од група ћелија (Слика 17), док употреба неких функција растојања не даје задовољавајуће резултате (Слике 18, 19), док у случају када је задат већи број кластера од (унапред познатог) броја група у улазном скупу (Слика 20) потребно је применити додатну анализу користећи резултате који су уписани у излазне датотеке.

Као илустрација, резултати примене CLUTO алгоритма за кластеровање у 5 и 15 кластера са различитим функцијама растојања као критеријума кластеровања су наведени у Додатку.

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=I1, SimFun=Cosine, #Clusters: 15
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=I1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------
15-way clustering: [I1=6.79e+03] [13299 of 13299], Entropy: 0.466, Purity: 0.510
------------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------------
  0    13 +0.523 +0.011 +0.236 +0.022 0.000 1.000 |   13    0    0    0    0    0    0    0    0    0    0    0    0    0    0
  1    92 +0.704 +0.036 +0.514 +0.023 0.272 0.674 |    3    0    0    0    0   27   62    0    0    0    0    0    0    0    0
  2   162 +0.615 +0.026 +0.500 +0.017 0.277 0.827 |    4    0  134    6    3    4    9    2    0    0    0    0    0    0    0
  3   373 +0.563 +0.025 +0.470 +0.021 0.220 0.732 |    0    0    0  273   99    0    1    0    0    0    0    0    0    0    0
  4   167 +0.570 +0.032 +0.495 +0.024 0.652 0.413 |    0   69    1    0    0    1    4    3   14    4    4   30    5   29    3
  5  1286 +0.480 +0.022 +0.439 +0.025 0.231 0.758 |    3    0    7  975  294    1    3    2    0    0    1    0    0    0    0
  6  1218 +0.508 +0.018 +0.468 +0.017 0.291 0.672 |   68    0    0    1    0  330  819    0    0    0    0    0    0    0    0
  7  1684 +0.535 +0.013 +0.496 +0.012 0.646 0.277 |    0  142    0    2    0    1    3    1  256  418    3   38  290   64  466
  8   400 +0.523 +0.020 +0.483 +0.020 0.319 0.645 |   17    4    0    0    0  118  258    1    0    0    2    0    0    0    0
  9  1815 +0.519 +0.014 +0.486 +0.014 0.631 0.320 |    0  153    0    1    0    0    0    0  125  423   13   58  580   77  385
 10  1355 +0.518 +0.017 +0.491 +0.017 0.580 0.390 |    1  351    0    4    0    4    3    0  269   45   10  528   37   85   18
 11   985 +0.511 +0.012 +0.486 +0.011 0.576 0.453 |    1   49    0    0    0    2    2    0   52  167    1   36  195   34  446
 12  1287 +0.496 +0.018 +0.480 +0.017 0.475 0.471 |    4    0    0    0    0    3    4   81  606    0  116   41    0  429    3
 13  1801 +0.473 +0.021 +0.458 +0.023 0.303 0.723 |    1    0    0    0    1    0    0 1303   17    0  267    0    0  212    0
 14   661 +0.495 +0.022 +0.482 +0.021 0.620 0.369 |    1  153    0    1    1    0    1    1   92   15    9  244   27  101   15
------------------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                            23.653 sec
   Clustering:                   1786.214 sec
   Reporting:                       3.748 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:         6539173888 bytes
   Current memory used:         3268800552 bytes
```

Слика 16. Резултати примене CLUTO алгоритма за добијање 15 кластера

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=H2, SimFun=Cosine, #Clusters: 5
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=H2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------
5-way clustering: [H2=7.89e-05] [13299 of 13299], Entropy: 0.529, Purity: 0.441
------------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------------
  0  1840 +0.489 +0.034 +0.447 +0.036 0.344 0.680 |   20    2  142 1252  395    6   13    8    0    0    2    0    0    0    0
  1  4628 +0.519 +0.015 +0.480 +0.014 0.654 0.281 |    1  363    0    5    1    4    8    1  477 1018   22  165 1065  196 1302
  2  1606 +0.507 +0.018 +0.470 +0.019 0.291 0.669 |   85    0    0    1    0  444 1075    0    0    0    1    0    0    0    0
  3  3257 +0.503 +0.020 +0.484 +0.019 0.682 0.285 |    7  556    0    5    0   34   71   73  928   54  112  810   69  505   33
  4  1968 +0.475 +0.021 +0.460 +0.024 0.349 0.667 |    3    0    0    0    2    3    2 1312   26    0  289    0    0  330    1
------------------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                                23.498 sec
   Clustering:                       3237.846 sec
   Reporting:                           3.701 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:             6539173888 bytes
   Current memory used:             3268800472 bytes
*******************************************************************************
```

Слика 17. Резултати примене CLUTO алгоритма за добијање 5 кластера користећи H_2 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=G1, SimFun=Cosine, #Clusters: 5
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=G1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------
5-way clustering: [G1=2.64e+01] [13299 of 13299], Entropy: 0.655, Purity: 0.322
------------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------------
  0  1772 +0.534 +0.014 +0.495 +0.013 0.641 0.256 |    0  166    0    1    0    0    0    0  291  454    2   41  299   64  454
  1  1805 +0.512 +0.012 +0.485 +0.011 0.579 0.391 |    0   90    0    0    1    0    0    0   80  356    4   58  445   65  706
  2  2669 +0.498 +0.026 +0.485 +0.025 0.842 0.207 |   18  283   92    5    1   31   67   70  553  214  130  307  345  395  158
  3  3412 +0.472 +0.023 +0.460 +0.029 0.555 0.368 |   96   30   50 1256  395  460 1102    5    6    0    3    4    1    3    1
  4  3641 +0.479 +0.022 +0.473 +0.027 0.655 0.362 |    2  352    0    1    1    0    0 1319  501   48  287  565   44  504   17
------------------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                                 23.502 sec
   Clustering:                        2018.879 sec
   Reporting:                            3.711 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:              6539173888 bytes
   Current memory used:              3268800472 bytes
*******************************************************************************
```

Слика 18. Резултати примене CLUTO алгоритма за добијање 5 кластера користећи G_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=G1, SimFun=Cosine, #Clusters: 15
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=G1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------
```

15-way clustering: [G1=2.13e+02] [13299 of 13299], Entropy: 0.489, Purity: 0.504

cid	Size	ISim	ISdev	ESim	ESdev	Entpy	Purty	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas
0	821	+0.529	+0.024	+0.464	+0.021	0.292	0.704	5	0	45	578	191	0	2	0	0	0	0	0	0	0	0
1	886	+0.535	+0.011	+0.497	+0.011	0.610	0.333	0	93	0	0	0	0	0	0	170	295	2	16	204	18	88
2	872	+0.470	+0.022	+0.431	+0.023	0.218	0.761	1	0	2	664	202	0	0	2	0	0	1	0	0	0	0
3	886	+0.537	+0.015	+0.499	+0.014	0.610	0.413	0	73	0	1	0	0	0	0	121	159	0	25	95	46	366
4	858	+0.509	+0.016	+0.471	+0.016	0.291	0.668	41	1	0	1	0	242	573	0	0	0	0	0	0	0	0
5	892	+0.519	+0.017	+0.486	+0.016	0.644	0.355	0	92	0	0	0	0	0	0	65	192	12	38	317	41	135
6	865	+0.517	+0.043	+0.485	+0.038	0.781	0.332	18	73	92	5	1	31	67	58	96	3	90	34	5	287	5
7	899	+0.519	+0.011	+0.490	+0.011	0.524	0.455	0	31	0	0	0	0	0	0	20	160	1	24	229	25	409
8	861	+0.503	+0.023	+0.475	+0.024	0.425	0.612	49	29	3	13	2	218	527	3	6	0	2	4	1	3	1
9	904	+0.514	+0.016	+0.489	+0.015	0.553	0.416	0	247	0	0	0	0	0	0	152	33	7	376	33	47	9
10	906	+0.508	+0.012	+0.484	+0.011	0.622	0.328	0	59	0	0	1	0	0	0	60	196	3	34	216	40	297
11	913	+0.504	+0.021	+0.487	+0.021	0.598	0.348	1	105	0	1	0	0	0	16	318	15	20	188	11	230	8
12	912	+0.502	+0.015	+0.485	+0.015	0.585	0.430	0	118	0	0	0	0	0	12	392	19	28	235	23	67	18
13	912	+0.482	+0.020	+0.466	+0.022	0.411	0.541	1	0	0	0	0	0	0	493	29	0	171	1	0	217	0
14	912	+0.467	+0.021	+0.452	+0.022	0.149	0.888	0	0	0	0	1	0	0	810	2	0	89	0	0	10	0

```
Timing Information ----------------------------------------------------------
   I/O:                               23.510 sec
   Clustering:                      2019.804 sec
   Reporting:                          3.738 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
```

Слика 19. Резултати примене CLUTO алгоритма за добијање 15 кластера користећи G_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=G1', SimFun=Cosine, #Clusters: 25
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=G1', NTrials=10, NIter=10

Solution --------------------------------------------------------------------
```

25-way clustering: [G1'=1.66e+08] [13299 of 13299], Entropy: 0.447, Purity: 0.535

cid	Size	ISim	ISdev	ESim	ESdev	Entpy	Purty	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas
0	14	+0.505	+0.029	+0.247	+0.046	0.095	0.929	13	0	0	0	1	0	0	0	0	0	0	0	0	0
1	123	+0.637	+0.067	+0.502	+0.036	0.468	0.569	5	7	0	0	0	30	70	7	1	0	2	0	1	0	0
2	192	+0.593	+0.035	+0.494	+0.022	0.369	0.714	5	1	137	28	13	3	4	0	0	0	1	0	0	0	0
3	400	+0.557	+0.027	+0.469	+0.022	0.215	0.745	0	0	0	298	101	0	1	0	0	0	0	0	0	0	0
4	258	+0.540	+0.032	+0.489	+0.024	0.652	0.368	2	95	0	0	0	2	5	0	32	8	3	60	11	34	6
5	542	+0.502	+0.020	+0.453	+0.020	0.245	0.745	2	0	3	404	128	0	2	2	0	0	1	0	0	0	0
6	1209	+0.509	+0.018	+0.468	+0.017	0.291	0.673	68	0	0	1	0	326	814	0	0	0	0	0	0	0	0
7	392	+0.522	+0.020	+0.482	+0.019	0.283	0.658	16	0	0	0	0	118	258	0	0	0	0	0	0	0	0
8	681	+0.468	+0.021	+0.429	+0.023	0.205	0.771	0	0	2	525	154	0	0	0	0	0	0	0	0	0	0
9	808	+0.539	+0.015	+0.500	+0.014	0.604	0.452	0	68	0	2	0	1	3	1	113	124	2	11	70	48	365
10	834	+0.533	+0.011	+0.496	+0.011	0.627	0.320	0	79	0	0	0	1	1	0	132	267	1	30	210	17	96
11	611	+0.520	+0.019	+0.486	+0.018	0.633	0.367	0	76	0	0	0	0	0	0	43	147	10	22	224	27	62
12	524	+0.525	+0.016	+0.491	+0.015	0.662	0.242	0	42	0	2	1	0	0	0	84	127	3	12	104	24	125
13	280	+0.485	+0.021	+0.454	+0.022	0.282	0.732	0	0	0	0	0	0	0	205	2	0	54	0	0	19	0
14	779	+0.520	+0.011	+0.490	+0.010	0.576	0.344	0	41	0	0	0	0	0	0	24	184	0	37	268	20	205
15	1343	+0.518	+0.017	+0.491	+0.016	0.566	0.419	1	357	0	3	0	4	3	1	230	36	12	563	30	82	21
16	1028	+0.511	+0.012	+0.486	+0.012	0.600	0.436	1	54	0	0	0	3	4	0	61	169	1	53	196	38	448
17	520	+0.510	+0.013	+0.485	+0.013	0.304	0.779	1	0	0	0	0	0	0	11	405	0	35	26	0	42	0
18	349	+0.503	+0.019	+0.483	+0.019	0.399	0.668	1	0	0	0	0	3	4	25	32	0	50	1	0	233	0
19	433	+0.491	+0.019	+0.471	+0.019	0.409	0.450	0	0	0	0	0	0	0	195	13	0	61	0	0	164	0
20	399	+0.477	+0.022	+0.458	+0.023	0.255	0.739	1	0	0	0	0	0	0	295	0	0	89	0	0	14	0
21	184	+0.484	+0.018	+0.465	+0.019	0.159	0.859	0	0	0	0	0	0	0	158	0	0	25	0	0	1	0
22	474	+0.467	+0.022	+0.449	+0.022	0.111	0.922	0	0	0	0	0	0	0	437	0	0	33	0	0	4	0
23	382	+0.495	+0.020	+0.479	+0.020	0.485	0.432	0	0	0	0	0	0	0	57	165	0	32	10	0	118	0
24	540	+0.496	+0.020	+0.482	+0.020	0.615	0.278	0	101	0	0	0	0	0	0	94	10	11	150	20	146	8

```
Timing Information ----------------------------------------------------------
   I/O:                              23.496 sec
   Clustering:                     2247.546 sec
   Reporting:                         3.715 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800552 bytes
```

Слика 20. Резултати примене CLUTO алгоритма за добијање 25 кластера користећи G_1 формулу

Закључак

Високодимензионални подаци су све чешћи у многим областима истраживања. Како се број димензија повећава, многи алгоритми кластеровања бивају неефикасни због проклетства димензионалности, деградирајући квалитет резултата. У великим димензијама подаци су веома ретки, а мере растојања све бесмисленије. Овај проблем је опширно проучаван и постоје различита решења, од којих је свако прикладно за различите типове високодимензионалних података и процедуре кластеровања података.

У овом раду су описани циљеви и изазови кластеровања високодимензионих података. Представљене су неке од најважнијих метода кластеровања високодимензионалних података. Описани су алгоритми кластеровања по потпросторима одозго према доле, одоздо према горе, алгоритми који користе мреже и хиперравни, као и К-средина и TwoStep и CLUTO алгоритми. На крају је илустрована могућност рада са једим скупом вишедимензионих података везаних за експресију гена из хуманих ћелија периферне крви са једним једром и приказани су резултати примене неких алгоритама на том скупу.

Додатак

Резултати примене CLUTO алгоритма:

```
Matrix Information ------------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options -----------------------------------------------------------------------
  CLMethod=Direct, CRfun=E1, SimFun=Cosine, #Clusters: 5
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=E1, NTrials=10, NIter=10

Solution ----------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------
5-way clustering: [E1=1.20e+08] [13299 of 13299], Entropy: 0.526, Purity: 0.443
------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------
  0  1851 +0.489 +0.034 +0.447 +0.036 0.351 0.678 |   20    4  142 1255  396    6   15    8    1    0    4    0    0    0    0
  1  4522 +0.519 +0.015 +0.480 +0.014 0.644 0.287 |    1  352    0    5    1    4    7    1  437 1016   17  141 1065  177 1298
  2  1602 +0.507 +0.018 +0.470 +0.019 0.292 0.669 |   85    1    0    1    0  444 1071    0    0    0    0    0    0    0    0
  3  3540 +0.503 +0.020 +0.483 +0.019 0.687 0.276 |    9  564    0    2    0   37   76   88  977   56  142  834   69  648   38
  4  1784 +0.472 +0.021 +0.458 +0.023 0.301 0.727 |    1    0    0    0    1    0    0 1297   16    0  263    0    0  206    0
------------------------------------------------------------------------------------------------------------------------------------

Timing Information ------------------------------------------------------------
   I/O:                                23.646 sec
   Clustering:                       3243.298 sec
   Reporting:                           3.697 sec
Memory Usage Information ------------------------------------------------------
   Maximum memory used:             6539173888 bytes
   Current memory used:             3268800472 bytes
*******************************************************************************
```

Слика 21. Резултати примене CLUTO алгоритма за добијање 5 кластера користећи E_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=H1, SimFun=Cosine, #Clusters: 5
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=H1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------
5-way clustering: [H1=5.60e-05] [13299 of 13299], Entropy: 0.525, Purity: 0.444
-------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
-------------------------------------------------------------------------------------------------------------------------------
  0  1843 +0.489 +0.034 +0.447 +0.036 0.345 0.680 |   20    3  142 1254  395    5   13    8    0    0    3    0    0    0    0
  1  4501 +0.520 +0.014 +0.480 +0.014 0.640 0.288 |    1  349    0    3    0    3    5    1  433 1017   17  134 1064  176 1298
  2  1614 +0.507 +0.018 +0.470 +0.019 0.296 0.667 |   85    3    0    1    0  447 1077    0    0    0    1    0    0    0    0
  3  3548 +0.502 +0.021 +0.483 +0.020 0.688 0.276 |    9  566    0    5    2   36   73   85  981   55  140  841   70  647   38
  4  1793 +0.473 +0.021 +0.458 +0.023 0.304 0.725 |    1    0    0    0    1    0    1 1300   17    0  265    0    0  208    0
-------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                              23.503 sec
   Clustering:                     3250.239 sec
   Reporting:                         3.713 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800472 bytes
*******************************************************************************
```

Слика 22. Резултати примене CLUTO алгоритма за добијање 5 кластера користећи H_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=I1, SimFun=Cosine, #Clusters: 5
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=I1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------
5-way clustering: [I1=6.69e+03] [13299 of 13299], Entropy: 0.524, Purity: 0.445
-------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
-------------------------------------------------------------------------------------------------------------------------------
  0  1821 +0.493 +0.028 +0.448 +0.031 0.324 0.689 |    7    0  141 1254  396    5   13    4    0    0    1    0    0    0    0
  1  4484 +0.520 +0.014 +0.480 +0.014 0.639 0.289 |    1  344    0    3    0    3    5    1  433 1008   17  132 1065  175 1297
  2  1618 +0.507 +0.018 +0.470 +0.019 0.300 0.666 |   85    4    0    1    0  448 1077    1    0    0    2    0    0    0    0
  3  3562 +0.502 +0.021 +0.483 +0.020 0.690 0.275 |    9  573    1    5    1   35   74   85  981   64  139  843   69  644   39
  4  1814 +0.470 +0.028 +0.457 +0.030 0.316 0.718 |   14    0    0    0    1    0    0 1303   17    0  267    0    0  212    0
-------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                              23.620 sec
   Clustering:                     1801.455 sec
   Reporting:                         3.811 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800472 bytes
*******************************************************************************
```

Слика 23. Резултати примене CLUTO алгоритма за добијање 5 кластера користећи I_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=I2, SimFun=Cosine, #Clusters: 5
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=I2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------
5-way clustering: [I2=9.43e+03] [13299 of 13299], Entropy: 0.525, Purity: 0.444
------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------
  0  1844 +0.489 +0.034 +0.447 +0.036 0.346 0.680 |   20    4  142 1254  395    5   13    8    0    0    3    0    0    0    0
  1  4501 +0.520 +0.014 +0.480 +0.014 0.641 0.288 |    1  346    0    4    0    3    5    1  435 1014   17  138 1064  175 1298
  2  1612 +0.507 +0.019 +0.470 +0.019 0.291 0.669 |   85    0    0    1    0  447 1078    0    0    0    1    0    0    0    0
  3  3565 +0.503 +0.021 +0.483 +0.019 0.690 0.275 |    9  571    0    4    2   36   73   93  980   58  140  837   70  654   38
  4  1777 +0.473 +0.021 +0.458 +0.023 0.300 0.727 |    1    0    0    0    1    0    0 1292   16    0  265    0    0  202    0
------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                                 23.515 sec
   Clustering:                        3329.005 sec
   Reporting:                            3.931 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:             6539173888 bytes
   Current memory used:             3268800472 bytes
*******************************************************************************
```

Слика 24. Резултати примене CLUTO алгоритма за добијање 5 кластера користећи I_2 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=E1, SimFun=Cosine, #Clusters: 15
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=E1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------
15-way clustering: [E1=1.19e+08] [13299 of 13299], Entropy: 0.470, Purity: 0.503
------------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------------
  0    15 +0.466 +0.069 +0.252 +0.048 0.179 0.867 |   13    0    0    0    1    0    1    0    0    0    0    0    0    0    0
  1   190 +0.598 +0.032 +0.497 +0.019 0.424 0.721 |    4    4  137   14    5    5   11    6    1    0    3    0    0    0    0
  2   393 +0.558 +0.027 +0.469 +0.022 0.218 0.738 |    0    0    0  290  102    0    1    0    0    0    0    0    0    0    0
  3   306 +0.572 +0.043 +0.496 +0.027 0.728 0.271 |    4   83    0    0    0   29   68    0   23    6    4   44    9   32    4
  4  1253 +0.480 +0.022 +0.438 +0.025 0.227 0.759 |    3    0    5  951  288    1    2    2    0    0    1    0    0    0    0
  5  1216 +0.508 +0.018 +0.468 +0.017 0.291 0.672 |   68    0    0    1    0  330  817    0    0    0    0    0    0    0    0
  6   386 +0.522 +0.020 +0.482 +0.019 0.291 0.658 |   17    1    0    0    0  114  254    0    0    0    0    0    0    0    0
  7  1652 +0.535 +0.013 +0.496 +0.012 0.650 0.279 |    0  148    0    2    0    2    4    1  243  403    3   37  283   65  461
  8  1869 +0.519 +0.015 +0.486 +0.014 0.637 0.315 |    0  154    0    3    1    0    0    0  138  443   13   66  588   75  388
  9  1371 +0.519 +0.017 +0.491 +0.016 0.575 0.408 |    1  354    0    2    0    5    4    1  252   40   11  559   36   86   20
 10  1001 +0.511 +0.011 +0.486 +0.011 0.583 0.449 |    1   50    0    0    0    2    3    0   56  170    1   38  194   37  449
 11   831 +0.482 +0.021 +0.465 +0.022 0.378 0.592 |    1    0    0    0    0    0    0  492   14    0  146    0    0  178    0
 12  1281 +0.498 +0.018 +0.481 +0.017 0.476 0.472 |    4    0    0    0    0    3    4   87  605    0  117   39    0  420    2
 13   953 +0.469 +0.021 +0.453 +0.022 0.193 0.845 |    0    0    0    0    1    0    0  805    2    0  117    0    0   28    0
 14   582 +0.496 +0.020 +0.482 +0.020 0.614 0.330 |    0  127    0    0    0    0    0    0   97   10   10  192   24  110   12
------------------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                                23.520 sec
   Clustering:                       3244.756 sec
   Reporting:                           3.722 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:             6539173888 bytes
   Current memory used:             3268800552 bytes
```

Слика 25. Резултати примене CLUTO алгоритма за добијање 15 кластера користећи E_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=H1, SimFun=Cosine, #Clusters: 15
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=H1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------
15-way clustering: [H1=5.71e-05] [13299 of 13299], Entropy: 0.465, Purity: 0.510
-------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
-------------------------------------------------------------------------------------------------------------------------------
  0    13 +0.523 +0.011 +0.236 +0.022 0.000 1.000 |   13    0    0    0    0    0    0    0    0    0    0    0    0    0    0
  1    92 +0.704 +0.036 +0.514 +0.023 0.272 0.674 |    3    0    0    0    0   27   62    0    0    0    0    0    0    0    0
  2   174 +0.609 +0.027 +0.500 +0.017 0.359 0.776 |    4    3  135    8    3    4    9    6    0    0    2    0    0    0    0
  3   375 +0.562 +0.026 +0.470 +0.022 0.220 0.733 |    0    0    0  275   99    0    1    0    0    0    0    0    0    0    0
  4   185 +0.560 +0.033 +0.492 +0.024 0.628 0.416 |    0   77    0    0    1    1    5    0   17    4    4   36    6   31    3
  5  1281 +0.480 +0.022 +0.439 +0.025 0.231 0.758 |    3    0    7  971  293    1    3    2    0    0    1    0    0    0    0
  6  1221 +0.508 +0.018 +0.468 +0.017 0.290 0.673 |   68    0    0    1    0  330  822    0    0    0    0    0    0    0    0
  7   393 +0.523 +0.020 +0.483 +0.020 0.306 0.649 |   17    3    0    0    0  117  255    0    0    0    1    0    0    0    0
  8  1721 +0.535 +0.013 +0.496 +0.012 0.646 0.275 |    0  150    0    2    0    1    3    1  264  430    3   36  291   67  473
  9  1802 +0.518 +0.014 +0.486 +0.014 0.631 0.324 |    0  150    0    1    0    0    0    0  118  420   13   64  583   75  378
 10  1370 +0.518 +0.017 +0.491 +0.016 0.576 0.403 |    1  350    0    4    0    5    3    0  261   39   11  552   38   88   18
 11   978 +0.511 +0.011 +0.486 +0.011 0.574 0.457 |    1   49    0    0    0    2    2    0   51  167    1   34  190   34  447
 12  1291 +0.496 +0.018 +0.480 +0.018 0.473 0.469 |    5    0    0    0    0    3    2   84  606    0  117   39    0  432    3
 13  1793 +0.473 +0.021 +0.458 +0.023 0.304 0.725 |    1    0    0    0    1    0    1 1300   17    0  265    0    0  208    0
 14   610 +0.496 +0.021 +0.482 +0.021 0.622 0.351 |    0  139    0    1    1    0    1    1   97   12    8  214   26   96   14
-------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                                23.513 sec
   Clustering:                       3241.290 sec
   Reporting:                           3.713 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:             6539173888 bytes
   Current memory used:             3268800552 bytes
```

Слика 26. Резултати примене CLUTO алгоритма за добијање 15 кластера користећи H_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=H2, SimFun=Cosine, #Clusters: 15
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=H2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------
15-way clustering: [H2=8.00e-05] [13299 of 13299], Entropy: 0.463, Purity: 0.509
------------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------------
  0    13 +0.523 +0.011 +0.236 +0.022 0.000 1.000 |   13    0    0    0    0    0    0    0    0    0    0    0    0    0    0
  1    96 +0.697 +0.040 +0.512 +0.024 0.314 0.656 |    3    3    0    0    0   27   63    0    0    0    0    0    0    0    0
  2   198 +0.596 +0.032 +0.496 +0.020 0.429 0.692 |    4    2  137   22   10    5   11    6    0    0    1    0    0    0    0
  3   398 +0.558 +0.026 +0.469 +0.021 0.215 0.744 |    0    0    0  296  101    0    1    0    0    0    0    0    0    0    0
  4   207 +0.552 +0.033 +0.491 +0.024 0.627 0.401 |    0   83    0    1    0    1    3    0   24    6    3   47    8   27    4
  5  1231 +0.479 +0.022 +0.438 +0.025 0.226 0.759 |    3    0    5  934  284    1    1    2    0    0    1    0    0    0    0
  6  1218 +0.508 +0.018 +0.468 +0.017 0.291 0.672 |   68    0    0    1    0  330  819    0    0    0    0    0    0    0    0
  7   388 +0.522 +0.022 +0.482 +0.021 0.290 0.660 |   17    0    0    0    0  114  256    0    0    0    1    0    0    0    0
  8  1932 +0.534 +0.013 +0.494 +0.012 0.663 0.261 |    0  169    0    2    0    2    4    1  307  464    6   69  328   76  504
  9   870 +0.519 +0.018 +0.486 +0.017 0.659 0.329 |    0  106    0    2    0    0    0    0   89  206   11   34  286   47   89
 10  1826 +0.512 +0.012 +0.485 +0.012 0.593 0.388 |    1   88    0    1    1    2    4    0   81  348    5   62  451   73  709
 11  1792 +0.511 +0.018 +0.488 +0.017 0.581 0.403 |    1  467    0    4    0    5    3    0  298   48   13  722   61  143   27
 12   729 +0.495 +0.020 +0.474 +0.020 0.450 0.417 |    2    0    0    0    1    3    2  270   25    0  121    0    0  304    1
 13  1162 +0.495 +0.018 +0.480 +0.017 0.463 0.522 |    3    3    0    0    0    1    2   73  606    0   96   41    0  335    2
 14  1239 +0.468 +0.021 +0.453 +0.022 0.190 0.841 |    1    0    0    0    1    0    0 1042    1    0  168    0    0   26    0
------------------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                              23.478 sec
   Clustering:                     3238.557 sec
   Reporting:                         3.731 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800552 bytes
```

Слика 27. Резултати примене CLUTO алгоритма за добијање 15 кластера користећи H_2 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=I2, SimFun=Cosine, #Clusters: 15
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=I2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------
15-way clustering: [I2=9.50e+03] [13299 of 13299], Entropy: 0.466, Purity: 0.509
-------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
-------------------------------------------------------------------------------------------------------------------------------
  0    13 +0.523 +0.011 +0.236 +0.022 0.000 1.000 |   13    0    0    0    0    0    0    0    0    0    0    0    0    0    0
  1    92 +0.704 +0.036 +0.514 +0.023 0.272 0.674 |    3    0    0    0    0   27   62    0    0    0    0    0    0    0    0
  2   176 +0.606 +0.029 +0.498 +0.018 0.370 0.767 |    4    4  135    9    3    4    9    6    0    0    2    0    0    0    0
  3   385 +0.560 +0.026 +0.469 +0.022 0.220 0.732 |    0    0    0  282  102    0    1    0    0    0    0    0    0    0    0
  4   185 +0.560 +0.033 +0.492 +0.024 0.628 0.416 |    0   77    0    0    1    1    5    0   17    4    4   36    6   31    3
  5  1270 +0.480 +0.022 +0.439 +0.026 0.231 0.758 |    3    0    7  963  290    1    3    2    0    0    1    0    0    0    0
  6  1211 +0.509 +0.018 +0.468 +0.017 0.291 0.674 |   68    0    0    1    0  326  816    0    0    0    0    0    0    0    0
  7   401 +0.521 +0.022 +0.482 +0.021 0.291 0.653 |   17    0    0    0    0  121  262    0    0    0    1    0    0    0    0
  8  1635 +0.534 +0.013 +0.495 +0.012 0.649 0.279 |    0  147    0    2    0    1    3    1  240  398    3   38  280   66  456
  9  1867 +0.519 +0.015 +0.486 +0.014 0.633 0.315 |    0  150    0    2    0    0    0    0  140  446   13   65  588   73  390
 10  1384 +0.518 +0.017 +0.491 +0.016 0.576 0.406 |    1  354    0    4    0    5    3    0  257   44   10  562   40   85   19
 11   999 +0.511 +0.012 +0.486 +0.011 0.577 0.452 |    1   49    0    0    0    2    2    0   55  170    1   35  196   36  452
 12  1286 +0.497 +0.019 +0.480 +0.018 0.478 0.471 |    5    0    0    0    0    3    3   93  606    0  114   39    0  420    3
 13  1777 +0.473 +0.021 +0.458 +0.023 0.300 0.727 |    1    0    0    0    1    0    0 1292   16    0  265    0    0  202    0
 14   618 +0.496 +0.019 +0.482 +0.019 0.618 0.324 |    0  140    0    0    1    0    0    0  100   10   12  200   24  118   13
-------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                                25.051 sec
   Clustering:                       3298.811 sec
   Reporting:                           3.780 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:             6539173888 bytes
   Current memory used:             3268800552 bytes
```

Слика 28. Резултати примене CLUTO алгоритма за добијање 15 кластера користећи I_2 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=E1, SimFun=Cosine, #Clusters: 25
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=E1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------
25-way clustering: [E1=1.18e+08] [13299 of 13299], Entropy: 0.446, Purity: 0.536
------------------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
------------------------------------------------------------------------------------------------------------------------------------------
  0    15 +0.466 +0.069 +0.252 +0.048 0.179 0.867 |   13    0    0    0    1    0    1    0    0    0    0    0    0    0    0    0
  1    94 +0.701 +0.038 +0.513 +0.024 0.290 0.670 |    3    1    0    0    0   27   63    0    0    0    0    0    0    0    0    0
  2   190 +0.598 +0.032 +0.497 +0.019 0.424 0.721 |    4    4  137   14    5    5   11    6    1    0    3    0    0    0    0    0
  3   393 +0.558 +0.027 +0.469 +0.022 0.218 0.738 |    0    0    0  290  102    0    1    0    0    0    0    0    0    0    0    0
  4   212 +0.549 +0.035 +0.490 +0.026 0.651 0.387 |    1   82    0    0    0    2    5    0   23    6    4   44    9   32    4
  5   503 +0.509 +0.016 +0.457 +0.018 0.237 0.751 |    2    0    3  378  117    1    2    0    0    0    0    0    0    0    0    0
  6  1216 +0.508 +0.018 +0.468 +0.017 0.291 0.672 |   68    0    0    1    0  330  817    0    0    0    0    0    0    0    0    0
  7   386 +0.522 +0.020 +0.482 +0.019 0.291 0.658 |   17    1    0    0    0  114  254    0    0    0    0    0    0    0    0    0
  8   798 +0.540 +0.015 +0.500 +0.014 0.600 0.459 |    0   68    0    2    0    1    3    1  103  126    2   10   69   47  366
  9   854 +0.533 +0.011 +0.496 +0.010 0.624 0.324 |    0   80    0    0    0    1    1    0  140  277    1   27  214   18   95
 10   750 +0.466 +0.021 +0.429 +0.023 0.219 0.764 |    1    0    2  573  171    0    0    2    0    0    1    0    0    0    0
 11   258 +0.492 +0.019 +0.457 +0.020 0.310 0.686 |    0    0    0    0    0    0    0  177    2    0   57    0    0   22    0
 12  1093 +0.520 +0.017 +0.487 +0.016 0.663 0.294 |    0  112    0    2    1    0    0    0  118  256   13   33  321   53  184
 13   723 +0.530 +0.017 +0.497 +0.016 0.538 0.368 |    0  189    0    0    0    2    0    0  189   14    5  266    8   44    6
 14   776 +0.520 +0.011 +0.490 +0.010 0.575 0.344 |    0   42    0    1    0    0    0    0   20  187    0   33  267   22  204
 15   530 +0.510 +0.013 +0.485 +0.013 0.306 0.777 |    1    0    0    0    0    0    0   12  412    0   35   27    0   43    0
 16  1001 +0.511 +0.011 +0.486 +0.011 0.583 0.449 |    1   50    0    0    0    2    3    0   56  170    1   38  194   37  449
 17   648 +0.510 +0.015 +0.488 +0.014 0.589 0.452 |    1  165    0    2    0    3    4    1   63   26    6  293   28   42   14
 18   341 +0.504 +0.020 +0.483 +0.020 0.411 0.657 |    3    0    0    0    0    3    4   22   32    0   52    1    0  224    0
 19   441 +0.493 +0.019 +0.472 +0.018 0.412 0.460 |    0    0    0    0    0    0    0  203   14    0   64    0    0  160    0
 20   390 +0.476 +0.022 +0.457 +0.022 0.261 0.741 |    1    0    0    0    0    0    0  289    0    0   82    0    0   18    0
 21   187 +0.484 +0.019 +0.465 +0.020 0.175 0.856 |    0    0    0    0    1    0    0  160    0    0   24    0    0    2    0
 22   508 +0.465 +0.022 +0.447 +0.022 0.111 0.921 |    0    0    0    0    0    0    0  468    0    0   36    0    0    4    0
 23   410 +0.492 +0.019 +0.478 +0.019 0.485 0.393 |    0    0    0    0    0    0    0   53  161    0   30   11    0  153    2
 24   582 +0.496 +0.020 +0.482 +0.020 0.614 0.330 |    0  127    0    0    0    0    0    0   97   10   10  192   24  110   12
------------------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                              23.508 sec
   Clustering:                     3244.881 sec
   Reporting:                         3.711 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800552 bytes
******************************************************************************
```

Слика 29. Резултати примене CLUTO алгоритма за добијање 25 кластера користећи E_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=H1, SimFun=Cosine, #Clusters: 25
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=H1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------
```

25-way clustering: [H1=5.76e-05] [13299 of 13299], Entropy: 0.444, Purity: 0.540

cid	Size	ISim	ISdev	ESim	ESdev	Entpy	Purty	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas
0	13	+0.523	+0.011	+0.236	+0.022	0.000	1.000	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	92	+0.704	+0.036	+0.514	+0.023	0.272	0.674	3	0	0	0	0	27	62	0	0	0	0	0	0	0	0
2	174	+0.609	+0.027	+0.500	+0.017	0.359	0.776	4	3	135	8	3	4	9	6	0	0	2	0	0	0	0
3	375	+0.562	+0.026	+0.470	+0.022	0.220	0.733	0	0	0	275	99	0	1	0	0	0	0	0	0	0	0
4	185	+0.560	+0.033	+0.492	+0.024	0.628	0.416	0	77	0	0	1	1	5	0	17	4	4	36	6	31	3
5	512	+0.511	+0.016	+0.458	+0.017	0.244	0.752	2	0	5	385	116	1	3	0	0	0	0	0	0	0	0
6	1221	+0.508	+0.018	+0.468	+0.017	0.290	0.673	68	0	0	1	0	330	822	0	0	0	0	0	0	0	0
7	393	+0.523	+0.020	+0.483	+0.020	0.306	0.649	17	3	0	0	0	117	255	0	0	0	1	0	0	0	0
8	795	+0.539	+0.015	+0.500	+0.014	0.594	0.467	0	58	0	2	0	1	2	1	99	130	2	12	70	47	371
9	226	+0.501	+0.015	+0.463	+0.016	0.354	0.642	0	0	0	0	0	0	1	145	4	0	51	0	0	25	0
10	926	+0.535	+0.011	+0.497	+0.010	0.620	0.324	0	92	0	0	0	0	1	0	165	300	1	24	221	20	102
11	769	+0.465	+0.021	+0.429	+0.023	0.219	0.762	1	0	2	586	177	0	0	2	0	0	1	0	0	0	0
12	1023	+0.520	+0.017	+0.486	+0.016	0.656	0.311	0	109	0	0	0	0	0	0	98	228	13	32	318	52	173
13	724	+0.530	+0.017	+0.497	+0.016	0.542	0.358	0	191	0	2	0	2	0	0	194	13	5	259	8	45	5
14	779	+0.521	+0.011	+0.490	+0.010	0.575	0.340	0	41	0	1	0	0	0	0	20	192	0	32	265	23	205
15	527	+0.510	+0.012	+0.485	+0.012	0.306	0.778	1	0	0	0	0	0	0	14	410	0	35	25	0	42	0
16	978	+0.511	+0.011	+0.486	+0.011	0.574	0.457	1	49	0	0	0	2	2	0	51	167	1	34	190	34	447
17	646	+0.510	+0.015	+0.487	+0.015	0.587	0.454	1	159	0	2	0	3	3	0	67	26	6	293	30	43	13
18	449	+0.495	+0.018	+0.473	+0.017	0.411	0.465	0	0	0	0	0	0	0	209	13	0	69	0	0	158	0
19	337	+0.501	+0.019	+0.482	+0.019	0.415	0.653	3	0	0	0	0	3	2	23	33	0	51	1	0	220	1
20	187	+0.483	+0.019	+0.465	+0.019	0.183	0.850	0	0	0	0	1	0	0	159	0	0	24	0	0	3	0
21	391	+0.474	+0.023	+0.456	+0.023	0.258	0.747	1	0	0	0	0	0	0	292	0	0	80	0	0	18	0
22	540	+0.464	+0.022	+0.447	+0.022	0.115	0.917	0	0	0	0	0	0	0	495	0	0	41	0	0	4	0
23	610	+0.496	+0.021	+0.482	+0.021	0.622	0.351	0	139	0	1	1	0	1	1	97	12	8	214	26	96	14
24	427	+0.488	+0.021	+0.476	+0.021	0.485	0.398	1	0	0	0	0	0	0	47	163	0	31	13	0	170	2

```
Timing Information ----------------------------------------------------------
   I/O:                              23.718 sec
   Clustering:                     3283.318 sec
   Reporting:                         3.734 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800552 bytes
*******************************************************************************
```

Слика 30. Резултати примене CLUTO алгоритма за добијање 25 кластера користећи H_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=H2, SimFun=Cosine, #Clusters: 25
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=H2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------
25-way clustering: [H2=8.03e-05] [13299 of 13299], Entropy: 0.452, Purity: 0.525
-------------------------------------------------------------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  Entpy Purty | Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas Klas
-------------------------------------------------------------------------------------------------------------------------------
  0    13 +0.523 +0.011 +0.236 +0.022 0.000 1.000 |   13    0    0    0    0    0    0    0    0    0    0    0    0    0    0
  1    96 +0.697 +0.040 +0.512 +0.024 0.314 0.656 |    3    3    0    0    0   27   63    0    0    0    0    0    0    0    0
  2   198 +0.596 +0.032 +0.496 +0.020 0.429 0.692 |    4    2  137   22   10    5   11    6    0    0    1    0    0    0    0
  3   398 +0.558 +0.026 +0.469 +0.021 0.215 0.744 |    0    0    0  296  101    0    1    0    0    0    0    0    0    0    0
  4   207 +0.552 +0.033 +0.491 +0.024 0.627 0.401 |    0   83    0    1    0    1    3    0   24    6    3   47    8   27    4
  5   529 +0.501 +0.020 +0.451 +0.021 0.235 0.745 |    2    0    3  394  128    1    1    0    0    0    0    0    0    0    0
  6  1218 +0.508 +0.018 +0.468 +0.017 0.291 0.672 |   68    0    0    1    0  330  819    0    0    0    0    0    0    0    0
  7   388 +0.522 +0.022 +0.482 +0.021 0.290 0.660 |   17    0    0    0    0  114  256    0    0    0    1    0    0    0    0
  8   693 +0.540 +0.013 +0.500 +0.012 0.632 0.359 |    0   65    0    1    0    0    0    1  126  138    2   23   61   27  249
  9   702 +0.468 +0.021 +0.431 +0.023 0.217 0.769 |    1    0    2  540  156    0    0    2    0    0    1    0    0    0    0
 10  1239 +0.532 +0.013 +0.496 +0.012 0.665 0.263 |    0  104    0    1    0    2    4    0  181  326    4   46  267   49  255
 11   870 +0.519 +0.018 +0.486 +0.017 0.659 0.329 |    0  106    0    2    0    0    0    0   89  206   11   34  286   47   89
 12   697 +0.529 +0.017 +0.496 +0.017 0.536 0.387 |    0  188    0    3    0    2    0    0  166   10    4  270    9   39    6
 13   283 +0.509 +0.019 +0.478 +0.020 0.466 0.519 |    2    0    0    0    1    3    2   59    9    0   59    0    0  147    1
 14   690 +0.520 +0.011 +0.491 +0.010 0.590 0.319 |    0   33    0    1    0    0    1    0   18  142    2   32  214   27  220
 15   644 +0.515 +0.012 +0.488 +0.012 0.498 0.579 |    1   17    0    0    0    2    3    0   30   83    0   11  102   22  373
 16   492 +0.510 +0.012 +0.484 +0.011 0.642 0.274 |    0   38    0    0    1    0    0    0   33  123    3   19  135   24  116
 17   550 +0.510 +0.013 +0.485 +0.013 0.362 0.722 |    1    0    0    0    0    1    2   18  397    0   50   24    0   57    0
 18   634 +0.509 +0.015 +0.487 +0.014 0.616 0.396 |    1  163    0    1    0    3    3    0   79   31    5  251   37   48   12
 19   623 +0.468 +0.021 +0.448 +0.021 0.130 0.905 |    0    0    0    0    0    0    0  564    1    0   52    0    0    6    0
 20   446 +0.492 +0.019 +0.472 +0.019 0.412 0.473 |    0    0    0    0    0    0    0  211   16    0   62    0    0  157    0
 21   202 +0.484 +0.018 +0.465 +0.019 0.188 0.847 |    0    0    0    0    1    0    0  171    0    0   26    0    0    4    0
 22   414 +0.474 +0.022 +0.456 +0.022 0.256 0.742 |    1    0    0    0    0    0    0  307    0    0   90    0    0   16    0
 23   461 +0.501 +0.020 +0.484 +0.019 0.556 0.436 |    0  116    0    0    0    0    0    0   53    7    4  201   15   56    9
 24   612 +0.487 +0.020 +0.477 +0.020 0.480 0.454 |    2    3    0    0    0    0    0   55  209    0   46   17    0  278    2
-------------------------------------------------------------------------------------------------------------------------------

Timing Information ----------------------------------------------------------
   I/O:                              23.476 sec
   Clustering:                     3297.892 sec
   Reporting:                         3.904 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:          6539173888 bytes
   Current memory used:          3268800552 bytes
*******************************************************************************
```

Слика 31. Резултати примене CLUTO алгоритма за добијање 25 кластера користећи H_2 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=I1, SimFun=Cosine, #Clusters: 25
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=I1, NTrials=10, NIter=10

Solution --------------------------------------------------------------------
```

25-way clustering: [I1=6.82e+03] [13299 of 13299], Entropy: 0.444, Purity: 0.538

cid	Size	ISim	ISdev	ESim	ESdev	Entpy	Purty	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas
0	13	+0.523	+0.011	+0.236	+0.022	0.000	1.000	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	92	+0.704	+0.036	+0.514	+0.023	0.272	0.674	3	0	0	0	0	27	62	0	0	0	0	0	0	0	0
2	162	+0.615	+0.026	+0.500	+0.017	0.277	0.827	4	0	134	6	3	4	9	2	0	0	0	0	0	0	0
3	373	+0.563	+0.025	+0.470	+0.021	0.220	0.732	0	0	0	273	99	0	1	0	0	0	0	0	0	0	0
4	167	+0.570	+0.032	+0.495	+0.024	0.652	0.413	0	69	1	0	0	1	4	3	14	4	4	30	5	29	3
5	492	+0.512	+0.016	+0.458	+0.017	0.247	0.746	2	0	5	367	114	1	3	0	0	0	0	0	0	0	0
6	1218	+0.508	+0.018	+0.468	+0.017	0.291	0.672	68	0	0	1	0	330	819	0	0	0	0	0	0	0	0
7	808	+0.540	+0.015	+0.500	+0.014	0.595	0.464	0	61	0	2	0	1	2	1	105	130	2	12	72	45	375
8	400	+0.523	+0.020	+0.483	+0.020	0.319	0.645	17	4	0	0	0	118	258	1	0	0	2	0	0	0	0
9	254	+0.501	+0.015	+0.462	+0.017	0.320	0.673	0	0	0	0	0	0	0	171	2	0	56	0	0	25	0
10	876	+0.534	+0.011	+0.496	+0.010	0.618	0.329	0	81	0	0	0	0	1	0	151	288	1	26	218	19	91
11	794	+0.466	+0.021	+0.430	+0.023	0.217	0.766	1	0	2	608	180	0	0	2	0	0	1	0	0	0	0
12	1049	+0.521	+0.016	+0.487	+0.016	0.657	0.299	0	112	0	0	0	0	0	0	105	240	13	29	314	54	182
13	730	+0.529	+0.017	+0.497	+0.016	0.544	0.349	0	195	0	1	0	2	0	0	198	16	5	255	8	45	5
14	766	+0.521	+0.011	+0.490	+0.010	0.573	0.347	0	41	0	1	0	0	0	0	20	183	0	29	266	23	203
15	521	+0.510	+0.012	+0.485	+0.012	0.308	0.774	1	0	0	0	0	0	0	11	403	0	35	26	0	45	0
16	985	+0.511	+0.012	+0.486	+0.011	0.576	0.453	1	49	0	0	0	2	2	0	52	167	1	36	195	34	446
17	625	+0.509	+0.015	+0.487	+0.015	0.596	0.437	1	156	0	3	0	2	3	0	71	29	5	273	29	40	13
18	451	+0.496	+0.017	+0.473	+0.017	0.413	0.461	0	0	0	0	0	0	0	208	15	0	65	0	0	163	0
19	329	+0.501	+0.019	+0.482	+0.019	0.431	0.638	3	0	0	0	0	3	4	22	35	0	50	1	0	210	1
20	187	+0.484	+0.019	+0.465	+0.019	0.175	0.856	0	0	0	0	1	0	0	160	0	0	24	0	0	2	0
21	393	+0.474	+0.022	+0.456	+0.023	0.264	0.735	1	0	0	0	0	0	0	289	0	0	85	0	0	18	0
22	516	+0.462	+0.022	+0.446	+0.022	0.112	0.921	0	0	0	0	0	0	0	475	0	0	37	0	0	4	0
23	661	+0.495	+0.022	+0.482	+0.021	0.620	0.369	1	153	0	1	1	0	1	1	92	15	9	244	27	101	15
24	437	+0.489	+0.020	+0.477	+0.020	0.480	0.398	0	0	0	0	0	0	0	48	168	0	31	14	0	174	2

```
Timing Information ----------------------------------------------------------
   I/O:                            23.787 sec
   Clustering:                   1780.390 sec
   Reporting:                       3.728 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:        6539173888 bytes
   Current memory used:        3268800552 bytes
*******************************************************************************
```

Слика 32. Резултати примене CLUTO алгоритма за добијање 25 кластера користећи I_1 формулу

```
Matrix Information ----------------------------------------------------------
  Name: cluto_cista_matrica.csv, #Rows: 13299, #Columns: 30698, #NonZeros: 408252702

Options ---------------------------------------------------------------------
  CLMethod=Direct, CRfun=I2, SimFun=Cosine, #Clusters: 25
  RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=30, AggloCRFun=I2, NTrials=10, NIter=10

Solution --------------------------------------------------------------------
```

25-way clustering: [I2=9.52e+03] [13299 of 13299], Entropy: 0.450, Purity: 0.524

cid	Size	ISim	ISdev	ESim	ESdev	Entpy	Purty	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas	Klas
0	13	+0.523	+0.011	+0.236	+0.022	0.000	1.000	13	0	0	0	0	0	0	0	0	0	0	0	0	0
1	92	+0.704	+0.036	+0.514	+0.023	0.272	0.674	3	0	0	0	0	27	62	0	0	0	0	0	0	0
2	176	+0.606	+0.029	+0.498	+0.018	0.370	0.767	4	4	135	9	3	4	9	6	0	0	2	0	0	0
3	385	+0.560	+0.026	+0.469	+0.022	0.220	0.732	0	0	0	282	102	0	1	0	0	0	0	0	0	0
4	185	+0.560	+0.033	+0.492	+0.024	0.628	0.416	0	77	0	0	1	1	5	0	17	4	4	36	6	31	3
5	484	+0.509	+0.018	+0.456	+0.020	0.248	0.746	2	0	5	361	112	1	3	0	0	0	0	0	0	0	0
6	1211	+0.509	+0.018	+0.468	+0.017	0.291	0.674	68	0	0	1	0	326	816	0	0	0	0	0	0	0	0
7	797	+0.539	+0.015	+0.500	+0.014	0.599	0.459	0	69	0	2	0	1	2	1	102	124	2	10	71	47	366
8	401	+0.521	+0.022	+0.482	+0.021	0.291	0.653	17	0	0	0	0	121	262	0	0	0	1	0	0	0	0
9	786	+0.468	+0.021	+0.431	+0.023	0.217	0.766	1	0	2	602	178	0	0	2	0	0	1	0	0	0	0
10	838	+0.533	+0.011	+0.496	+0.010	0.622	0.327	0	78	0	0	0	0	1	0	138	274	1	28	209	19	90
11	251	+0.495	+0.017	+0.459	+0.019	0.311	0.685	0	0	0	0	0	0	0	172	2	0	55	0	0	22	0
12	603	+0.522	+0.018	+0.486	+0.017	0.629	0.371	0	74	0	0	0	0	0	0	42	145	10	19	224	26	63
13	490	+0.526	+0.016	+0.492	+0.015	0.655	0.251	0	35	0	1	0	0	0	0	76	115	3	11	100	26	123
14	729	+0.529	+0.017	+0.497	+0.016	0.543	0.365	0	189	0	2	0	2	0	0	192	15	5	266	8	45	5
15	774	+0.521	+0.011	+0.490	+0.010	0.578	0.341	0	41	0	1	0	0	0	0	22	186	0	35	264	21	204
16	999	+0.511	+0.012	+0.486	+0.011	0.577	0.452	1	49	0	0	0	2	2	0	55	170	1	35	196	36	452
17	655	+0.510	+0.015	+0.487	+0.015	0.586	0.452	1	165	0	2	0	3	3	0	65	29	5	296	32	40	14
18	430	+0.494	+0.018	+0.472	+0.018	0.414	0.456	0	0	0	0	0	0	0	196	14	0	64	0	0	156	0
19	867	+0.504	+0.017	+0.483	+0.016	0.459	0.512	5	0	0	0	0	3	3	36	444	0	83	26	0	266	1
20	185	+0.484	+0.019	+0.465	+0.019	0.167	0.859	0	0	0	0	1	0	0	159	0	0	24	0	0	1	0
21	399	+0.476	+0.022	+0.457	+0.023	0.261	0.739	1	0	0	0	0	0	0	295	0	0	85	0	0	18	0
22	512	+0.465	+0.022	+0.448	+0.022	0.116	0.918	0	0	0	0	0	0	0	470	0	0	37	0	0	5	0
23	618	+0.496	+0.019	+0.482	+0.019	0.618	0.324	0	140	0	0	1	0	0	0	100	10	12	200	24	118	13
24	419	+0.490	+0.021	+0.477	+0.021	0.492	0.387	0	0	0	0	0	0	0	57	162	0	31	13	0	154	2

```
Timing Information ----------------------------------------------------------
   I/O:                                 23.660 sec
   Clustering:                        3236.256 sec
   Reporting:                            3.735 sec
Memory Usage Information ----------------------------------------------------
   Maximum memory used:              6539173888 bytes
   Current memory used:              3268800552 bytes
*******************************************************************************
```

Слика 33. Резултати примене CLUTO алгоритма за добијање 25 кластера користећи I_2 формулу

Литература

[1] Parsons, L., Haque, E. and Liu, H., 2004. Subspace clustering for high dimensional data: a review. *Acm sigkdd explorations newsletter,* 6(1), pp.90-105.

[2] Assent, I., 2012. Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(4), pp.340-350.

[3] Steinbach, M., Ertöz, L. and Kumar, V., 2004. The challenges of clustering high dimensional data. In *New directions in statistical physics* (pp. 273-309). Springer, Berlin, Heidelberg.

[4] Masulli, F. and Rovetta, S., 2012, May. Clustering high-dimensional data. In *International Workshop on Clustering High-Dimensional Data* (pp. 1-13). Springer, Berlin, Heidelberg

[5] Wu, X. and Kumar, V. eds., 2009. *The top ten algorithms in data mining*. CRC press.

[6] IBM SPSS Modeler 18.2 Algorithms Guide

[7] K. Verhoeckx, P. Cotter, I. López-Expósito, C. Kleiveland, T. Lea, A. Mackie, et al., The Impact of Food Bioactives on Health: in vitro and ex vivo models [Internet]. Cham (CH): Springer; 2015. Chapter 15. Расположиво на: https://www.ncbi.nlm.nih.gov/books/NBK500157/

[8] Karypis, G., 2002.CLUTO-a clustering toolkit. MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE.