УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Емилија Стошић

ОДРЕЂИВАЊЕ ПОГОДНОСТИ ПОДАТАКА ЗА КЛАСТЕРОВАЊЕ

мастер рад

Београд, 2026.

Ментор:

др Ненад Митић, редован професор
Универзитет у Београду, Математички факултет

Чланови комисије:

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

др Мирјана Маљковић Ружичић, доцент
Универзитет у Београду, Математички факултет

Датум одбране:

Наслов мастер рада: Одређивање погодности података за кластеровање

Резиме: У савременим условима интензивног генерисања података расте потреба за методама које омогућавају откривање скривених образаца у великим скуповима података. Кластеровање, као основна техника ненадгледаног учења, омогућава груписање сличних објеката, али његова успешност у великој мери зависи од карактеристика самих података, због чега је процена њихове погодности од посебног значаја.

У раду се разматра проблем процене погодности података за кластеровање, са освртом на идентификацију кластерске структуре. Приказане су визуелне методе, као и Хопкинсова статистика као значајна квантитативна мера структуираности података. У раду је реализована њена имплементација у програмском језику C.

Експериментални резултати показују да предложено решење поуздано разликује насумичне и кластерски организоване скупове података. Хопкинсова статистика се показала као ефикасан и практично применљив алат за иницијалну анализу података, који доприноси правилном избору метода кластеровања и унапређењу квалитета резултата.

Кључне речи: кластеровање, кластерска тенденција, Хопкинсова статистика, визуелне методе, мере сличности, растојање Минковског, случајне тачке

Садржај

Глава 1

Увод

1.1 Предмет и циљ рада

Живимо у времену у коме се подаци генеришу брже него икада раније. Сваког тренутка настају милиони нових записа – од трансакција у банкарским системима, преко медицинских резултата, до података које остављамо на друштвеним мрежама "паметним" (IoT) уређајима и индустријским сензорима. Према извештајима компанија као што су *IBM* и *Google*, количина података у свету мери се зета (10^{21}) и јотабајтима (10^{24}), а њихов раст има експоненцијални карактер [11]. Међутим, сама чињеница да података има много не значи да они аутоматски носе знање. Између „податка" и „информације" налази се процес анализе, селекције, структурисања и интерпретације [4].

У таквом окружењу, кластер анализа представља једну од значајних техника за откривање и анализу образаца у подацима [8]. Кластер анализа обухвата поступке који се могу примењивати како пре, тако и након самог процеса кластеровања. Пре кластеровања, анализа се може користити за испитивање да ли у подацима уопште постоји тенденција ка формирању група, док се након формирања кластера може користити за њихово описивање, тумачење и вредновање. На тај начин, кластер анализа не представља само поступак проналажења кластера, већ шири процес испитивања и разумевања структуре података.

За разлику од надгледаних метода машинског учења, где постоје унапред дефинисане класе, кластеровање настоји да открије унутрашњу структуру података без претходних ознака. Оно одговара на питање: да ли се објекти

у скупу природно групишу и, ако да, на који начин? Управо ту почиње суштински проблем теме „Одређивање погодности података за кластеровање“.

У пракси, није сваки скуп података погодан за кластер анализу. Често се сусрећемо са ситуацијама у којима подаци немају изражену структуру, већ су распоређени насумично или су снажно оптерећени шумом. Примена алгоритама кластеровања у таквим условима може довести до привидних, вештачких група које немају стварно значење. Другим речима, алгоритам ће „пронаћи“ кластере чак и када они у суштини не постоје. То може довести до погрешних закључака, лоших пословних одлука или научно неутемељених резултата.

Додатну сложеност уноси и сама природа савремених података. Данашњи скупови су често:

- високодимензиони (велики број атрибута),
- хетерогени (комбинација нумеричких, категоричких и текстуалних података),
- непотпуни (са недостајућим вредностима),
- оптерећени шумом и аномалијама,
- великог обима (хиљаде, милиони или милијарде слогова).

У високодимензионим просторима јавља се феномен „проклетства димензионалности“, где класичне мере растојања губе дискриминативну моћ (способност неког показатеља, методе или модела да јасно разликује, раздвоји различите групе, класе или појаве). Објекти постају „подједнако удаљени“, што отежава идентификацију природних група. У таквим условима, пре примене било ког алгоритма кластеровања, неопходно је поставити фундаментално питање: да ли подаци уопште поседују тенденцију ка груписању?

Одређивање погодности података за кластеровање представља управо тај почетни, али често занемарени корак анализе. Уместо да се одмах приступи избору алгоритма (нпр. к-средина, хијерархијско кластеровање, *DBSCAN*), неопходно је проценити да ли структура података оправдава примену ових метода. Ова процена може бити заснована на статистичким мерама, визуелним техникама или комбинацији оба приступа.

Један од статистичких приступа јесте Хопкинсова статистика, која омогућава квантитативну процену степена кластеризације у скупу података [6]. Њена суштина је у поређењу стварне структуре података са насумично генерисаним узорком у истом простору. Ако су подаци блиски насумичној

расподели, кластеровање нема јасно оправдање. Са друге стране, вредности које указују на одступање од случајности сигнализирају присуство унутрашње структуре.

Поред статистичких мера, значајну улогу имају и визуелне технике, као што су *VAT (Visual Assessment of cluster Tendency)* и *iVAT*, које омогућавају интуитиван увид у постојање потенцијалних кластера кроз реорганизацију матрице растојања. Ове методе омогућавају истраживачу да „види“ структуру података пре формалне примене алгоритма, што је посебно значајно у истраживачким и експлораторним анализама.

У савременом контексту обраде великих количина података, где се одлуке често доносе аутоматизовано и у реалном времену, овај корак провере погодности постаје још важнији. Погрешно кластеровање може утицати на сегментацију клијената, дијагностичке системе у медицини, препоруке у е-трговини или анализу ризика у финансијама. Због тога је неопходно развити методологију која је:

- теоријски заснована,
- статистички исправна,
- рачунарски ефикасна,
- скалабилна на велике скупове података,
- независна од броја слогова и атрибута.

Проблем одређивања погодности података за кластеровање значајан је у савременој анализи података, посебно имајући у виду све већу количину и сложеност података који се обрађују. Он повезује теоријске концепте из области статистике, геометрије и машинског учења са практичном применом метода над скуповима различитих величина и димензија.

1.2 Мотивација и значај истраживања

У овом раду су приказане технике за одређивање погодности података за кластеровање, као и њихова практична примена кроз имплементацију Хопкинсове статистике и визуелне провере тенденција кластеровања. Посебан акценат стављен је на методе које омогућавају процену постојања кластерске

структуре независно од избора конкретног алгоритма кластеровања и броја очекиваних кластера.

У оквиру рада су анализиране како квантитативне, тако и визуелне технике процене кластерибилности података. Хопкинсова статистика користи се као нумеричка мера која омогућава објективну процену степена груписања унутар скупа података, док визуелне методе, попут *VAT* приступа, пружају интуитиван увид у постојање и распоред потенцијалних кластера.

Практични део рада обухвата имплементацију наведених метода на начин који обезбеђује добијање поузданих резултата без обзира на величину улазног скупа података, како у погледу броја објеката, тако и у погледу броја атрибута. Посебна пажња посвећена је скалабилности решења, употреби узорковања и независности имплементације од димензионалности података, чиме се омогућава примена предложеног приступа и на велике и сложене скупове података.

1.3 Структура рада

Рад је организован у више поглавља која систематично обрађују проблематику кластер анализе и процене погодности података за кластеровање.

У првом поглављу дат је увод у тему истраживања, при чему су представљени предмет и циљ рада, као и мотивација и значај истраживања у области анализе података.

Друго поглавље посвећено је основним појмовима кластеровања. У оквиру овог поглавља објашњен је појам и циљеви кластер анализе, представљене су теоријске основе кластеровања и различите парадигме које се користе у овој области. Поред тога, разматране су и најважније препреке за успешно кластеровање.

У трећем поглављу анализира се погодност података за кластер анализу. Посебна пажња посвећена је појму кластерске тенденције, као и карактеристикама различитих типова података, укључујући структуриране, неструктуриране и насумичне податке. Такође се разматра значај процене погодности података пре примене алгоритама кластеровања.

У четвртом поглављу представљене су методе за процену погодности података за кластеровање. Посебно су описане визуелне методе и Хопкинсова

статистика, које омогућавају процену постојања кластерске структуре у подацима.

У петом поглављу приказана је практична имплементација метода за процену погодности података за кластеровање. У оквиру овог поглавља извршена је имплементација Хопкинсовеове статистике и визуелна анализа кластерске тенденције на одабраном скупу података, као и анализа добијених резултата.

На крају рада дат је закључак у коме су сумирани главни резултати истраживања и указано је на могуће правце даљег развоја и примене метода за процену кластерске структуре података.

Глава 2

Основни појмови кластеровања

2.1 Појам и циљеви кластеровања

Кластеровање представља метод ненадгледаног учења чији је основни задатак идентификација структуре у скупу података без унапред познатих класа или ознака. За разлику од класификације, где је циљ репродукција већ дефинисане поделе, код кластеровања се структура података открива искључиво на основу међусобних односа између објеката [3, 4].

Основна идеја кластеровања јесте да се објекти који се међусобно слични, по одређеним атрибутима, сврстају у исту групу, док се објекти који се разликују, распоређују у различите групе, при чему се сличност најчешће квантификује мерама растојања или сличности. Кластеровање се посматра као проблем моделовања латентне структуре података, где се претпоставља постојање скривених група које генеришу опажене податке. Ова карактеристика чини кластеровање изузетно значајним у ситуацијма када не постоји претходно знање о организацији података, али истовремено намеће питање - да ли подаци уопште поседују унутрашњу структуру која оправдава њихово груписање?

Уколико су подаци распоређени приближно равномерно или случајно у простору обележја, сваки алгоритам ће произвести одређену поделу, али таква подела неће имати реално аналитичко значење. Управо из тог разлога појам и циљ кластеровања морају се посматрати у непосредној вези са питањем погодности података за ову врсту анализе.

2.1.1 Циљ кластеровања и његова ограничења

Основни циљ кластеровања јесте максимизација унутаркластерске хомогености и међукластерске хетерогености [3]. Ипак, постизање овог циља не представља довољан услов да би се добијени резултати сматрали значајним. Већина алгоритама за кластеровање заснива се на оптимизацији одређене функције циља, која је способна да формира кластере чак и кад у подацима не постоји јасно изражена структура. Због тога се резултати кластеровања не смеју тумачити искључиво на основу вредности оптимизационог критеријума, већ је неопходно проценити и да ли добијени кластери заиста представљају природне групе у подацима.

Ово ограничење једно је од кључних разлога због којих су у пракси поред самог алгоритма кластеровања примењују раличите методе за процену квалитета кластера и испитивање постојања кластерске структуре у подацима.

2.1.2 Утицај карактеристика података на кластеровање

Погодност података за кластеровање односи се на степен у којем подаци испољавају кластерску тенденцију, односно природну склоност ка формирању група. Уколико су растојања између објеката приближно једнака, разлика између унутаркластерских и међукластерских дистанци постаје статистички занемарљива [4, 12].

Са аспекта циља кластеровања, подаци су погодни за ову анализу уколико испуњавају следеће услове:

- Постоји довољна варијабилност која омогућава раздвајање група;
- Ниво шума не надвладава сигнал у подацима;
- Димензионалност простора не доводи до концентрације мера растојања;
- Изабрана метрика сличности адекватно одражава стварне односе међу објектима.

У високо-димензионалним просторима често долази до феномена концентрације растојања, при чему разлике између најближих и најудаљенијих тачака постају релативно мале [5]. У таквим условима, чак и алгоритми који формално минимизују функцију циља могу произвести нестабилне и тешко интерпретабилне кластере.

Због тога се у савременој анализи података наглашава потреба претходне процене кластерске тенденције пре саме примене алгоритама. Та процена може обухватити статистичке мере, визуелизацију, редукцију димензионалности и анализу расподеле растојања [4, 12].[4, 13]

Кластери би требало да представљају смислене групе које доприносе разумевању посматраног феномена. Уколико подаци нису погодни за кластеровање, резултати могу довести до погрешних закључака и неадекватних одлука. Стога се у оквиру истраживања погодности података за кластеровање појам кластеровања проширује: оно није само поступак груписања, већ инструмент откривања структуре чија успешност директно зависи од својстава података. Анализа погодности података представља неопходан корак који логички претходи примени алгоритама кластеровања.

2.2 Теоријске основе кластер анализе

Кластер анализа обухвата различите статистичке и алгоритамске приступе који се користе за идентификацију и испитивање структуре података.

Нека је дат скуп података

$$X = \{x_1, x_2, ..., x_n\}, x \in R^d$$

где сваки објекат представља вектор обележја у d-димензионалном простору. Задатак кластеровања јесте партиционисање скупа X на k дисјунктних подскупова $C_1, C_2, ..., C_k$, тако да су објекти унутар истог кластера међусобно слични, док су објекти из различитих кластера што је могуће више различити [12], односно важи:

1. $C_i \neq \emptyset$,
2. $C_i \cap C_j = \emptyset$ за $i \neq j$,
3. $\cup_{i=1}^{k} C_i = X$.

Код партитивних метода проблем се често формулише као оптимизациони задатак минимизације функције циља. Типичан пример је минимизација суме квадрата одступања објеката од центроида кластера:

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \tag{2.1}$$

где μ_i представља центроид кластера C_i.

Овакав приступ карактеристичан је за алгоритам к-средина. Овај модел имплицитно претпоставља да су кластери приближно сферног облика и сличне величине, што представља једно од важних теоријских ограничења методе.

Основ кластер анализе чини дефинисање адекватне мере сличности или растојања. У нумеричким просторима најчешће се користи Еуклидско растојање, док се код података високе димензионалности или текстуалних репрезентација често примењује косинусна сличност. Избор метрике директно утиче на геометријску структуру кластера и може значајно променити резултат анализе [12].

Важан теоријски аспект јесте да изабрана мера задовољава особине метрике: ненегативност, симетричност и неједнакост троугла. У супротном, могу настати аномалије у хијерархијским методама или потешкоће у конвергенцији алгоритама. У високо-димензионалним просторима додатни проблем представља феномен познат као „проклетство димензионалности“, где растојања између тачака постају приближно једнака, чиме се отежава јасно раздвајање кластера [4].

2.2.1 Парадигме кластеровања

Постоји више различитих теоријских приступа формирању и идентификовању кластера, који се разликују према претпоставкама о структури, облику и расподели података. Најзначајније парадигме кластеровања су:

Партициона парадигма - заснива се на директној оптимизацији функције циља и подразумева унапред задат број кластера. Циљ је минимизација варијансе унутар кластера или максимизација растојања између њихових центара [4]. Овде је кластер геометријски дефинисан (често као „облак тачака“ око центроида).

Хијерархијска парадигма - формира структуру налик стаблу (дендрограм) сукцесивним спајањем или раздвајањем кластера. Начин одређивања растојања између кластера дефинише се критеријумима повезивања, као што

су *Single linkage* (метода најближег суседа), *Complete linkage* (метода најдаљег суседа) и *Average linkage* (метода просечног повезивања) [12]. Кластер се посматра као део структуре стабла (дендрограма). Нема једне фиксне партиције, већ се анализира структура на више нивоа грануларности.

Парадигме засноване на густинама - кластери се дефинишу као региони повећане густине података раздвојени областима ниске густине. Овакав приступ омогућава проналажење кластера неправилног облика. Овде не полазимо од геометријске компактности, већ од локалне густине.

Модел-заснована (вероватносна) парадигма - полази од претпоставке да су подаци генерисани из мешавине вероватносних расподела. У том контексту, кластер представља латентну променљиву, а параметри модела процењују се методом максималне вероватноће или *EM* алгоритмом [3]. За разлику од класичног, тврдог *(hard)* кластеровања, у оквиру ове парадигме објекат не мора бити искључиво придружен једном кластеру, већ се његова припадност различитим кластерима може изразити преко вероватноћа. Овај приступ даје чврсту статистичку основу и омогућава процену вероватноће припадности сваког објекта одређеном кластеру.

2.3 Препреке за добро кластеровање

Кластер анализа представља једну од најзначајнијих техника ненадгледаног учења у областима рударења података *(Data Mining)* и машинског учења *(Machine Learning)*. Иако овај концепт делује интуитивно и једноставно, практична примена кластер анализе често је праћена бројним изазовима који могу значајно утицати на квалитет добијених резултата.

Препреке за успешно кластеровање могу потицати из различитих извора: карактеристика самих података, ограничења алгоритама кластеровања, избора параметара метода, као и недовољне припреме података. Разумевање ових проблема представља важан корак у правилној примени кластер анализе и интерпретацији добијених резултата.

Због наведених изазова, у пракси је веома важно пре примене алгоритама кластеровања проценити да ли посматрани скуп података уопште поседује структуру погодну за груписање. У ту сврху развијене су различите статистичке и аналитичке методе које омогућавају процену погодности података за кластер анализу. Неке од најчешће коришћених метода биће представљене

у наредном поглављу.

2.3.1 Висока димензионалност података

Један од најзначајнијих проблема у кластер анализи представља висока димензионалност података. У савременим системима за прикупљање и складиштење података често се анализирају скупови који садрже велики број атрибута. Последица овог јесте да традиционалне мере сличности, као што је Еуклидско растојање, губе дискриминативну моћ. То отежава идентификацију природних група у подацима и може довести до формирања кластера који не одражавају стварну структуру података. Због тога се у пракси често користе технике редукције димензионалности или селекције атрибута како би се задржале само најрелевантније карактеристике података [4].

2.3.2 Присуство шума и аномалија

Реални скупови података ретко су идеално структурирани. Често садрже шум и аномалије које могу нарушити структуру података. Шум представља случајне варијације или нетачности у подацима, док аномалије представљају објекте који значајно одступају од већине података.

Присуство оваквих елемената може имати значајан утицај на резултате кластер анализе. Алгоритми који се заснивају на израчунавању центроида кластера, као што су методе типа К-средина, посебно су осетљиви на аномалије. Један екстреман објекат може значајно померити центроид кластера и тиме нарушити структуру група. Из тог разлога је у многим случајевима неопходно применити технике откривања и уклањања аномалија пре примене алгоритама кластеровања [3].

2.3.3 Различите размере атрибута

У пракси се подаци често прикупљају из различитих извора и могу бити изражени у различитим јединицама мере. На пример, један атрибут може представљати старост особе изражену у годинама, док други може представљати годишњи приход у хиљадама јединица валуте. Уколико се такви подаци непосредно користе у алгоритмима кластеровања, атрибути са већим нумеричким опсегом могу доминирати у израчунавању растојања.

Овај проблем може довести до тога да поједини атрибути имају непропорционално велики утицај на резултате кластер анализе. Да би се то избегло, уобичајена пракса је примена техника нормализације или стандардизације података, чиме се вредности атрибута доводе у упоредив опсег [4].

Међутим, примена ових техника не мора увек довести до бољег резултата. Уколико одређени атрибут има већу стварну важност за разликовање објеката, његово изједначавање са осталим атрибутима може неоправдано смањити његов утицај на резултат кластеровања. На тај начин нормализација или стандардизација може потиснути значај информативнијих атрибута и довести до структуре кластера која мање одговара стварним односима у подацима. Због тога избор начина скалирања не би требало да буде аутоматски поступак, већ треба да узме у обзир природу података, мерне јединице и релативни значај појединачних атрибута.

2.3.4 Избор мере сличности

Избор одговарајуће мере сличности или растојања представља један од кључних фактора у кластер анализи. Различите мере могу довести до различитих резултата кластеровања, чак и када се примењују на исти скуп података. Најчешће коришћене мере укључују Еуклидско растојање, Менхетн растојање, косинусну сличност, растојање Минковског и Жакардов индекс сличности. Међутим, ниједна од ових метрика није универзално погодна за све типове података. На пример, Еуклидско растојање добро функционише код континуалних нумеричких података, али може бити мање погодно за податке високе димензионалности или категоријалне атрибуте. Погрешан избор мере сличности може довести до формирања кластера који не одражавају стварне односе између објеката [12].

2.3.5 Одређивање броја кластера

У многим алгоритмима кластеровања неопходно је унапред дефинисати број кластера који треба пронаћи у подацима. Међутим, у пракси тај број често није познат. Неправилан избор броја кластера може довести до погрешне интерпретације структуре података.

Уколико је број кластера превелик, могу се добити групе које су сувише мале и не представљају значајне обрасце у подацима. Са друге стране, ако

је број кластера сувише мали, различите групе података могу бити спојене у један кластер. Због тога се у анализи података користе различити индекси валидности који квантитативно процењују добијене кластерске структуре на основу компактности и међусобне раздвојености кластера и статистичке методе за процену оптималног броја група [7].

2.3.6 Различита густина и облик кластера

Још један значајан изазов у кластер анализи представља чињеница да кластери у реалним подацима могу имати различите облике, величине и густине. Неки алгоритми кластеровања подразумевају да су кластери сферног облика и приближно једнаке величине. Међутим, у пракси се често јављају кластери неправилних облика или различите густине.

У таквим ситуацијама алгоритам може погрешно интерпретирати структуру података и формирати кластере који не одговарају стварним групама. Овај проблем посебно долази до изражаја код метода заснованих на репрезентативним елементима, док алгоритми засновани на густини могу бити погоднији за откривање сложенијих структура у подацима [3].

2.3.7 Скалабилност алгоритама

Савремени информациони системи генеришу огромне количине података, што поставља нове захтеве пред алгоритме кластеровања. Алгоритам мора бити способан да обради велике скупове података у разумном времену и уз прихватљиву потрошњу ресурса.

Неки алгоритми кластеровања имају високу рачунарску сложеност и постају непрактични када се примењују на велике скупове података. Због тога је развој скалабилних алгоритама један од важних праваца истраживања у области анализе података [12].

2.3.8 Интерпретабилност резултата

Чак и када је кластеровање успешно изведено са техничке стране, може се јавити проблем интерпретације добијених резултата. Кластери представљају групе објеката са сличним карактеристикама, али њихово значење често није одмах очигледно.

Истраживач мора анализирати карактеристике сваког кластера како би утврдио шта та група представља у контексту посматраног проблема. У неким случајевима, кластери могу бити тешко интерпретабилни или не могу бити директно повезани са конкретним појавама у реалном систему [7].

2.3.9 Квалитет и припрема података

Квалитет података је један од кључних фактора који одређује успешност процеса кластеровања и других метода истраживања података. Под квалитетом података подразумевају се тачност, потпуност, конзистентност и релевантност података за конкретну анализу. Недостатак квалитетних података може довести до ефекта *„Garbage in, Garbage out“* (Лоши подаци – лоши резултати), где чак и најсофистицираније алгоритми дају непоуздане резултате [3, 4].

Пре примене кластровања, подаци се често морају очистити и припремити. Основне фазе припреме података укључују:

1. **Чишћење података** *(Data Cleaning)*: уклањање поновљених записа, обрада недостајућих вредности, исправљање недоследности и отклањање очигледних грешака насталих приликом уноса или прикупљања података [9].

2. **Трансформацију података** *(Data Transformation)*: нормализација или стандардизација вредности како би се обезбедила компаративност међу различитим атрибутима, што је критично у алгоритмима заснованим на растојању, као што су K-средине и хијерархијско кластеровање. [4, 12].

3. **Одабир атрибута** *(Feature Selection)*: идентификација релевантних атрибута који утичу на структуру кластера, чиме се смањује утицај ирелевантних и редундантних информација и повећава ефикасност алгоритама [3, 13].

4. **Руковање недостајућим вредностима**: уколико подаци садрже недостајуће вредности, начин њихове обраде зависи од природе података и захтева примењеног алгоритма. Један од могућих приступа је импутација (*imputation*), односно замена недостајуће вредности процењеном вредношћу применом одговарајућих метода. У одређеним случајевима могуће је искључити записе који садрже недостајуће вредности, нарочито када је њихов број мали. Међутим, недостајуће вредности се могу и оставити непромењене уколико примењени алгоритам подржава рад са таквим подацима. У том случају неки алгоритми могу једноставно занемарити атрибут са недостајућом вред-

ношћу и користити остале доступне атрибуте, док други могу самостално одредити или обрадити недостајућу вредност током процеса анализе. На тај начин није неопходно аутоматски уклањати непотпуне записе или претходно попуњавати све недостајуће вредности, већ се начин њиховог третирања прилагођава конкретном алгоритму [12].

Висок квалитет и правилна припрема података омогућавају боље дефинисане и интерпретабилне кластере, смањујући ризик од погрешних закључака и повећавајући поузданост аналитичких резултата. Практичне студије показују да чак и мали пропусти у припреми података могу значајно нарушити кластерску структуру, што наглашава важност систематског приступа у фази припреме података [3, 4, 9, 12].

Глава 3

Погодности података за кластеровање

Проблематика погодности података за кластеровање обухвата више аспеката. Пре свега, неопходно је утврдити да ли у подацима постоје природне групе објеката, односно да ли подаци испољавају такозвану кластерску тенденцију. Поред тога, важно је проценити у којој мери је структура података изражена и да ли је довољно стабилна да омогући поуздано кластеровање. Ова питања су посебно значајна у савременим системима анализе података, где се обрађују велики и комплексни скупови података различитог порекла [4].

Уколико се кластер анализа примени на податке који немају јасно изражену структуру, резултати могу бити погрешно интерпретирани. Алгоритам ће у већини случајева ипак формирати одређени број кластера, јер је његов задатак управо груписање објеката. Међутим, ти кластери могу представљати само случајне варијације у подацима, а не стварне групе објеката. Због тога је у анализи података неопходно применити методе које омогућавају процену склоности података ка формирању кластера [7].

У наставку поглавља биће размотрени основни аспекти који утичу на погодност података за кластеровање, укључујући појам кластерске тенденције, карактеристике структурираних и неструктурираних података, као и проблем насумичних података и значај процене погодности података пре примене алгоритама кластер анализе.

3.1 Тенденција ка постојању кластера

Један од основних концепата у анализи погодности података за кластеровање јесте појам кластерске тенденције. Кластерска тенденција представља својство скупа података које указује на постојање природних група објеката унутар посматраног простора података. Другим речима, она описује степен у којем подаци показују склоност ка формирању кластера, односно да ли су објекти распоређени тако да формирају компактне групе које су међусобно довољно раздвојене. Уколико је кластерска тенденција изражена, може се очекивати да ће алгоритми кластер анализе успети да идентификују смислене и интерпретабилне кластере. Насупрот томе, уколико подаци немају изражену структуру груписања, резултати кластеровања могу бити случајни или тешко интерпретабилни [7].

Кластерска тенденција може се посматрати као индикатор структуре података. У скуповима података са јасно израженом кластерском структуром, објекти који припадају истом кластеру имају мање међусобно растојање у односу на објекте из других кластера. Овај принцип представља основу већине алгоритама кластеровања, који настоје да минимизују унутрашњу варијабилност кластера и истовремено максимизују разлике између различитих кластера. Међутим, у пракси структура података често може бити сложена, са преклапањем кластера, различитим густинама или неправилним облицима група, што додатно отежава процену кластерске тенденције [12].

Због тога је у анализи података развијен низ приступа који омогућавају процену кластерске тенденције пре примене алгоритама кластеровања. Ови приступи могу бити визуелни, статистички или засновани на анализи растојања између објеката. Њихова основна сврха је да пруже увид у структуру података и помогну истраживачима да процене да ли је примена кластер анализе оправдана. У наредним поглављима биће детаљније размотрене карактеристике различитих типова података, као и проблем насумичних података који представљају један од најчешћих изазова у анализи кластерске структуре.

3.2 Структурирани и неструктурирани подаци

У зависности од степена организације и начина представљања, подаци се најчешће деле на структуриране и неструктуриране [4]. Структурирани подаци организовани су у унапред дефинисаној форми, док неструктурирани подаци немају јасно дефинисану шему организације и могу се појављивати у различитим облицима.

Разлика између ова два типа података значајна је са аспекта њихове припреме и примене метода анализе, јер начин представљања података непосредно утиче на могућности њихове даље обраде. Основне карактеристике структурираних и неструктурираних података, као и специфичности њихове примене у кластеровању, приказане су у наредним одељцима.

3.2.1 Структурирани подаци

Структурирани подаци су најчешће организовани у облику табела које садрже редове и колоне. Сваки ред у табели представља један објекат или запис, док колоне представљају атрибуте који описују карактеристике тог објекта. Оваква организација података омогућава њихово једноставно складиштење, претраживање и обраду у оквиру система за управљање базама података [4].

Код структурираних података сваки атрибут има јасно дефинисан тип вредности, као што су нумеричке вредности, категоричке ознаке или логичке вредности. Управо овај степен организације омогућава примену различитих алгоритама анализе података, укључујући статистичке методе, технике машинског учења и алгоритме кластеровања. У областима као што су анализа тржишта, финансијска анализа или медицинска истраживања, већина података је управо структурирана и чува се у релационим базама података [12].

Једна од основних предности структурираних података јесте могућност лаког дефинисања мере сличности између објеката. Пошто су вредности атрибута унапред дефинисане и најчешће нумеричке или категоричке природе, могуће је применити различите мере растојања као што су Еуклидско растојање, Менхетн растојање или косинусна сличност. Ове мере представљају основу за рад многих алгоритама кластер анализе [4].

Међутим, иако су структурирани подаци погодни за анализу, њихова примена у кластер анализи праћена је одређеним изазовима. Један од најчешћих проблема јесте различита скала атрибута, услед које поједини атрибути могу имати непропорционално велики утицај на резултате кластеровања. Због тога се у процесу припреме података често примењују технике нормализације или стандардизације вредности атрибута [3]. Нормализација и стандардизација представљају технике скалирања података које се примењују како би сви атрибути били упоредиви без обзира на њихове мерне јединице или опсег вредности. Нормализацијом се вредности пресликавају у унапред дефинисан интервал, док се стандардизацијом подаци трансформишу тако да имају средњу вредност једнаку нули и стандардну девијацију јединицу. Ове технике су посебно значајне код алгоритама кластеровања који се заснивају на мерама растојања, јер спречавају да атрибути са већим нумеричким опсегом непропорционално утичу на формирање кластера.

Упркос овим изазовима, структурирани подаци представљају најпогоднији тип података за примену класичних метода анализе података и кластер анализе. Управо због тога су многи алгоритми у области анализе података развијени имајући у виду управо овакву структуру података.

3.2.2 Неструктурирани подаци

Неструктурирани подаци се појављују у различитим облицима као што су текстуални документи, слике, аудио записи, видео материјали и подаци генерисани на друштвеним мрежама [12].

Савремени информациони системи генеришу огромне количине неструктурираних података. Процењује се да значајан део података који се данас ствара у дигиталном окружењу припада управо овој категорији. Због тога је анализа неструктурираних података постала важна област истраживања у оквиру дисциплина као што су рударење података *(Data Mining)* и машинско учење *(Machine Learning)* [3].

Један од главних изазова у раду са неструктурираним подацима јесте чињеница да они не садрже експлицитно дефинисане атрибуте који би се могли директно користити. Због тога је често неопходно применити технике екстракције карактеристика, којима се из неструктурираних података издвајају релевантне информације и трансформишу у нумерички облик погодан за даљу анализу.

На пример, код текстуалних података често се користе методе које омогућавају представљање текста у облику вектора карактеристика. Ови вектори се затим могу користити у различитим алгоритмима кластеровања како би се идентификовале групе сличних докумената. Сличан приступ примењује се и код анализе слика или аудио сигнала, где се из оригиналних података издвајају карактеристике које описују њихову структуру [4].

Иако је анализа неструктурираних података сложенија у односу на анализу структурираних података, њен значај у савременим системима анализе података непрестано расте. Развој нових метода обраде и представљања података омогућио је примену техника кластер анализе и у областима у којима је то раније било тешко изводљиво.

3.3 Насумични подаци

Насумични подаци представљају скупове у којима није присутна јасно изражена структура која би указивала на природно груписање објеката. Објекти су у таквим скуповима распоређени приближно случајно у простору, без изражених група или других образаца који би указивали на постојање кластерске тенденције.

Уколико се алгоритам кластеровања примени на насумичне податке, он ће ипак формирати одређени број кластера, јер је то његова основна функција. Међутим, ови кластери неће представљати стварне групе објеката, већ ће бити резултат случајних варијација у подацима.

Овај проблем може довести до погрешних закључака у анализи података. На пример, истраживач може закључити да постоје различите групе објеката у подацима, иако се у стварности ради о потпуно случајној расподели. Због тога је важно применити методе које омогућавају разликовање између структурираних и насумичних података.

Један од приступа у овој области подразумева поређење посматраног скупа података са симулираним случајним подацима. Уколико се показује да су карактеристике оба скупа сличне, може се закључити да посматрани подаци немају изражену кластерску структуру [6].

Глава 4

Методе процене погодности

У литератури се наводи више приступа процени погодности података за кластеровање. Ови приступи се могу грубо поделити у неколико група у зависности од начина на који анализирају структуру података. Једну групу чине методе које се ослањају на визуелну анализу података и матрица растојања, док другу групу чине статистичке методе које покушавају да квантитативно измере степен кластерске структуре у подацима. Поред тога, постоје и хибридни приступи који комбинују визуелне и статистичке технике како би се добила поузданија процена структуре података [7].

Избор одговарајућег приступа зависи од карактеристика скупа података, укључујући број објеката, број атрибута, тип података и меру сличности, односно растојања која се користи. Посебно је важно нагласити да процена кластерске тенденције није независна од начина на који се дефинише растојање између објеката. Различите метрике могу различито описати сличност објеката и, самим тим, довести до различитих закључака о постојању кластерске структуре.

У наставку су детаљније описане две групе метода које су од значаја за даљу анализу: визуелне методе, са посебним осврtом на методу *Visual Assessment of Clustering Tendency (VAT)*, и статистички приступ заснован на Хопкинсовој статистици.

4.1 Визуелне методе

Визуелна анализа података представља један од основних корака у процесу истраживачке анализе података. Пре примене сложених алгоритама

кластер анализе, корисно је извршити визуелну инспекцију података како би се стекла интуитивна представа о њиховој структури. Визуелне методе омогућавају истраживачу да уочи потенцијалне групе објеката, неправилности у расподели података, присуство шума или аномалија, као и могуће проблеме у припреми података. Овај приступ је посебно значајан у раним фазама анализе, јер омогућава брзо и интуитивно разумевање карактеристика скупа података [4].

Основна идеја визуелних метода заснива се на графичком представљању података у простору атрибута или у облику матрица које описују међусобну сличност објеката. Уколико у подацима постоји кластерска структура, она се често може препознати као груписање тачака у одређеним регионима простора или као блокови сличних вредности у матрици растојања. На тај начин визуелизација омогућава истраживачу да стекне почетни увид у потенцијалну структуру података пре примене формалних метода кластеровања [12].

Једна од најједноставнијих визуелних техника јесте графичко приказивање података у облику дијаграма расејања *(scatter plot)*. У оваквом приказу сваки објекат из скупа података представља се као тачка у координатном систему чије осе одговарају посматраним атрибутима. Уколико подаци имају изражену кластерску структуру, групе објеката могу се уочити као компактни скупови тачака који су међусобно раздвојени. Дијаграми расејања су посебно корисни код података са малим бројем димензија, јер омогућавају директан увид у просторну расподелу објеката [7].

Међутим, овај приступ има ограничења када је број атрибута већи од два или три. У таквим ситуацијама директна визуелизација постаје отежана, па се често примењују технике редукције димензионалности које омогућавају пројекцију података у простор мање димензије. Иако овакве технике могу олакшати визуелну анализу, треба имати у виду да редукција димензионалности може довести до губитка дела информација, што може утицати на исправност интерпретације структуре података [13].

Поред директног приказа података у простору атрибута, значајну улогу у визуелној процени кластерске структуре имају и методе засноване на анализи матрица растојања. У овом приступу најпре се израчунава матрица која садржи међусобна растојања свих парова објеката у скупу података. Ова матрица затим се приказује у облику графичке мапе, где боја или интензитет пиксела одговара вредности растојања између објеката. Уколико у подацима

постоје кластери, у матрици растојања се често могу уочити карактеристични блокови који одговарају групама међусобно сличних објеката [2]. За скуп података који садржи n објеката, матрица растојања може се представити у следећем облику:

$$D = \begin{bmatrix} 0 & d(x_1, x_2) & \dots & d(x_1, x_n) \\ d(x_2, x_1) & 0 & \dots & d(x_2, n) \\ \vdots & \vdots & \ddots & \vdots \\ d(x_n, x_1) & d(x_n, x_2) & \cdots & 0 \end{bmatrix} \tag{4.1}$$

где је: $d(x_i, x_j)$ растојање између објеката x_i и x_j. У анализи података најчешће се користи Еуклидско растојање, које се дефинише као

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \tag{4.2}$$

где је: m број атрибута, док x_{ik} и x_{jk} означавају вредности k-тог атрибута за објекте x_i и x_j. Добијена матрица растојања представља основу за даљу визуелну анализу структуре података у оквиру *VAT* методе.

Поступак примене *VAT* методе може се описати кроз следеће кораке:

1. **Формирање матрице растојања** - За посматрани скуп података израчунава се матрица растојања D која садржи растојања између свих парова објеката.

2. **Избор почетног објекта** - Као почетни објекат бира се онај који има највећу удаљеност од осталих објеката у скупу података.

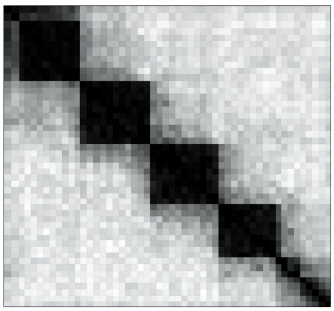
3. **Итеративно додавање објеката** - У сваком кораку алгоритам бира објекат који је најближи већ изабраном скупу објеката и додаје га у листу већ изабраних објеката, чиме се постепено формира нови редослед објеката.

4. **Примена добијеног редоследа на матрицу растојања** - Добијена пермутација објеката користи се за истовремено преуређивање редова и колона матрице растојања, чиме се формира матрица погодна за визуелну анализу.

5. **Визуелизација матрице** - Тако добијена матрица се приказује у облику слике, где интензитет боје представља величину растојања између објеката.

6. **Интерпретација резултата** - Анализом добијене визуелизације могуће је проценити да ли у подацима постоји кластерска структура, која се уочава у облику тамних блокова дуж главне дијагонале матрице.

Матрица се реорганизује тако да се објекти који су међусобно слични налазе једни поред других. У овој матрици редови и колоне представљају појединачне објекте из скупа података, док сваки елемент матрице означава растојање између одговарајућег пара објеката. Тако преуређена матрица се затим приказује у облику визуелне представе, односно слике, где интензитет боје представља величину растојања између посматраних објеката [2]. Резултат примене описаног поступка приказује се у облику VAT визуелизације матрице растојања, као што је илустровано на Слици 4.1.



Слика 4.1: Пример *VAT* визуелизације матрице растојања. Тамни квадратни блокови дуж главне дијагонале указују на постојање кластера у скупу података.

Као што је приказано на Слици 4.1, уколико у подацима постоји изражена кластерска структура, она се у *VAT* визуелизацији манифестује у облику тамних квадратних блокова дуж главне дијагонале матрице. Сваки од ових блокова представља групу објеката који су међусобно слични, односно имају мала међусобна растојања. Главна дијагонала матрице је увек тамна, јер представља растојање објекта од самог себе, које је једнако нули. Насупрот томе, уколико су подаци насумично распоређени и не садрже јасну структуру, *VAT* визуелизација неће показати јасно дефинисане блокове, већ ће матрица имати неправилан и расут образац боја. На тај начин, ова метода омогућава истраживачу да већ у раној фази анализе стекне увид у потенцијално постојање кластера у подацима [12].

Предност *VAT* методе је у томе што омогућава релативно једноставно уочавање потенцијалних кластера чак и у сложенијим скуповима података.

Поред тога, ова метода може пружити индикацију о приближном броју кластера у подацима на основу броја уочених блокова у матрици. Због тога се *VAT* често користи као прелиминарни корак у анализи података пре примене конкретних алгоритама кластеровања [2].

Ипак, као и друге визуелне технике, *VAT* метода има одређена ограничења. Интерпретација добијеног приказа у великој мери зависи од искуства истраживача, а код веома великих скупова података визуелна анализа може постати отежана. Поред тога, визуелне методе не пружају формалну статистичку меру кластерске тенденције, већ пре свега служе као интуитивни алат за иницијалну анализу структуре података [2].

Код веома великих скупова података визуелизација може постати тешка за тумачење, јер се појављује велики број елемената у матрици. Због наведених ограничења, визуелне методе се у пракси често комбинују са статистичким техникама које омогућавају квантитативну процену кластерске структуре. На тај начин се добија поузданија процена погодности података за кластеровање, јер се визуелни увид допуњује формалним статистичким показатељима. Једна од најчешће коришћених статистичких метода за процену кластерске тенденције јесте Хопкинсова статистика, која ће бити детаљније разматрана у наредном одељку.

4.2 Хопкинсова статистика

Једна од најпознатијих статистичких метода за процену кластерске тенденције података је Хопкинсова статистика (Hopkins statistic). Ова метода представља квантитативни приступ који омогућава процену да ли је посматрани скуп података значајно структуриран или се понаша као насумично распоређен скуп тачака у посматраном простору. За разлику од визуелних метода, које се у великој мери ослањају на субјективну интерпретацију резултата, Хопкинсова статистика даје нумеричку меру која омогућава објективнију процену погодности података за кластеровање [1].

Основна идеја ове методе заснива се на поређењу растојања између реалних података и растојања која би се добила у случају да су подаци равномерно насумично распоређени у посматраном простору. Уколико су подаци заиста насумични, њихова просторна структура ће бити слична структури случајно генерисаних тачака. Са друге стране, уколико у подацима постоји изражена

кластерска структура, та растојања ће показати значајна одступања од случајног распоређивања [10].

Нека је дат скуп података

$$X = \{x_1, x_2, ..., x_n\}$$

који се налази у m-димензионалном простору. Нека је k број насумично изабраних објеката из скупа података, при чему важи $k \ll n$. За сваки изабрани објекат x_i израчунава се растојање до његовог најближег суседа у скупу података. Ова растојања означавају се са

$$w_i = \min_{j \neq i} d(x_i, x_j)$$

где функција $d(\cdot)$ представља меру растојања (најчешће Еуклидско растојање).

Истовремено се генерише k случајних тачака

$$y_1, y_2, ..., y_k$$

у оквиру истог простора. За сваку од ових тачака одређује се растојање до најближег објекта из скупа података:

$$u_i = \min_{j} d(y_i, x_j)$$

Након тога, Хопкинсова статистика се израчунава према следећој формули:

$$H = \frac{\sum_{i=1}^{k} u_i}{\sum_{i=1}^{k} u_i + \sum_{i=1}^{k} w_i} \tag{4.3}$$

У појединим радовима и имплементацијама Хопкинсове статистике користи се и варијанта формуле у којој се посматрају квадрати растојања. У том случају вредност Хопкинсове статистике може се изразити у следећем облику:

$$H = \frac{\sum_{i=1}^{k} u_i^2}{\sum_{i=1}^{k} u_i^2 + \sum_{i=1}^{k} w_i^2} \tag{4.4}$$

где је:

u_i^2 - квадрат растојања од случајно генерисане тачке до најближег објекта

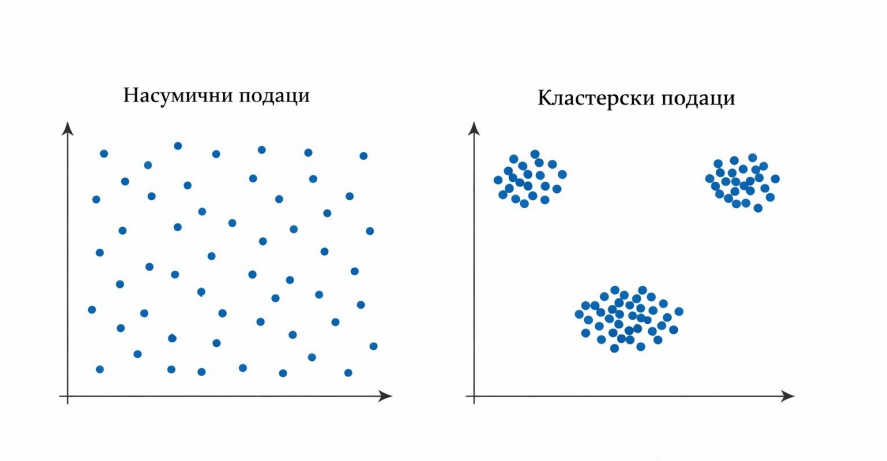
w_i^2 - квадрат растојања од изабраног објекта до његовог најближег суседа

k - број узоркованих тачака.

Коришћење квадрата растојања има за циљ наглашавање већих просторних разлика између тачака у m-димензионалном простору, што у одређеним случајевима може допринети стабилнијој процени кластерске тенденције. Коришћењем квадрата растојања, избегавамо израчунавање квадратног корена за Еуклидско растојање и поједностављујмо рачунску процедуру. Поред тога, употреба квадрата растојања не утиче на релативни однос вредности у имениоцу и бројиоцу формуле, па се добија еквивалентна мера кластерске тенденције података [6].

Вредност Хопкинсове статистике налази се у интервалу $0 \leq H \leq 1$. Ради прегледније интерпретације, вредност $H \approx 0.5$ указује на то да су подаци приближно насумично распоређени, без јасно изражене кластерске структуре. У овом случају примена алгоритама кластеровања најчешће не даје значајне резултате. Вредности $H \to 1$ указују на постојање изражене кластерске тенденције, при чему су објекти груписани у компактне регионе простора, што указује на погодност података за примену метода кластер анализе. Насупрот томе, вредности $H \to 0$ указују на то да су подаци регуларно, односно приближно равномерно распоређени у простору. Објекти су тада приближно једнако удаљени једни од других, те не постоји природна кластерска структура.

На слици 4.2 приказана је разлика између насумично распоређених и кластерски организованих података.



Слика 4.2: Насумична и кластерска структура података

Као што је приказано на левој стани Слике 4.2, приказан је скуп података који не испољава јасну просторну структуру, већ карактеристике насумичне расподеле, што је у складу са вредностима Хопкинсове статистике које теже ка 0.5. Насупрот томе, десни део слике приказује податке са израженом кластерском структуром, где су објекти груписани у компактне и добро раздвојене регионе простора, што резултира вредностима Хопкинсове статистике блиским јединици. Овај визуелни приказ наглашава суштину методе, која се заснива на разликовању просторне организације података кроз анализу растојања између објеката и случајно генерисаних тачака.

Једна од важних предности Хопкинсове статистике је њена релативна једноставност и могућност примене на различите типове података. Поред тога, ова метода не захтева претходно познавање броја кластера, што је чини погодном за почетну анализу података. Због тога се Хопкинсова статистика често користи као први корак у процесу анализе података пре примене конкретних алгоритама кластеровања [4].

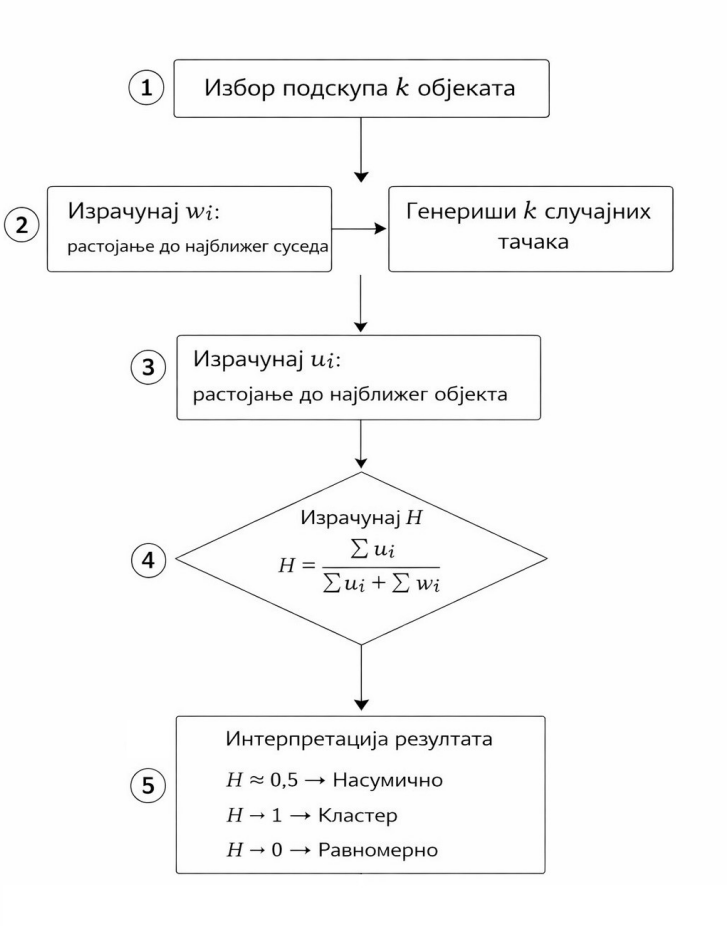
Међутим, као и свака статистичка метода, Хопкинсова статистика има и одређена ограничења. Пре свега, резултати могу зависити од избора параметра k, односно броја насумично изабраних објеката. Уколико је овај број сувише мали, процена може бити нестабилна. Са друге стране, веома велики број изабраних објеката може повећати рачунску сложеност методе, нарочито код великих скупова података.

Поред тога, важно је нагласити да Хопкинсова статистика не одређује директно број кластера у подацима, већ само процењује да ли у подацима постоји кластерска структура. Због тога се у пракси ова метода најчешће комбинује са другим техникама анализе података, као што су визуелне методе или алгоритми за одређивање оптималног броја кластера.

Упркос наведеним ограничењима, Хопкинсова статистика представља један од најчешће коришћених приступа за процену кластерске тенденције података. Њена примена омогућава истраживачима да у раној фази анализе података процене да ли је примена кластер анализе оправдана, што може значајно допринети ефикасности и поузданости даље анализе. Израчунавање Хопкинсове статистике подразумева више узастопних корака који обухватају избор подскупа објеката, генерисање случајних тачака и израчунавање растојања до најближих суседа.

Ради лакшег разумевања поступка израчунавања Хопкинсове статистике,

алгоритам се може представити и у облику дијаграма тока. Овакав приказ омогућава јасније сагледавање редоследа корака, као и односа између појединачних фаза алгоритма. Посебно је значајан у контексту практичне имплементације, јер пружа интуитиван увид у структуру поступка и олакшава његову реализацију у програмском језику.



Слика 4.3: Дијаграм тока поступка израчунавања Хопкинсове статистике

Дијаграм тока (слика 4.3) приказује основне кораке алгоритма за израчунавање Хопкинсове статистике. Процес почиње избором подскупа објеката из оригиналног скупа података, након чега следи израчунавање растојања до најближих суседа. Паралелно са тим, генеришу се случајне тачке у посматраном простору и за њих се одређују растојања до најближих објеката. На основу добијених вредности израчунава се Хопкинсова статистика, чија интерпретација омогућава процену кластерске тенденције података.

На основу изложене теоријске основе, може се закључити да Хопкинсова статистика представља ефикасну и поуздану методу за процену кластерске тенденције података. Приказани математички модел, алгоритамски кораци, као и визуелна интерпретација, омогућавају свеобухватно разумевање начина на који се ова мера примењује у анализи података.

Ипак, пуни значај ове методе долази до изражаја тек кроз њену практичну примену. Имплементација алгоритма омогућава анализу реалних и синтетичких скупова података, као и проверу добијених теоријских закључака у конкретним условима.

У наредном поглављу биће приказана практична имплементација Хопкинсове статистике у програмском језику C, уз анализу начина на који се поједини кораци алгоритма реализују у програмском окружењу, као и приказ резултата добијених применом на различите скупове података.

Глава 5

Практична имплементација

Након теоријског разматрања метода за процену погодности података за кластеровање, са посебним освртом на Хопкинсову статистику, природан наставак истраживања представља њена практична имплементација. Док теоријски модели пружају основу за разумевање понашања података, тек кроз реализацију алгоритма у програмском окружењу могуће је у потпуности сагледати његову применљивост, ефикасност и ограничења у раду са конкретним скуповима података.

У овом поглављу је приказан поступак имплементације Хопкинсове статистике у програмском језику C, са циљем да се омогући анализа података независно од величине скупа и броја атрибута. Посебна пажња биће посвећена организацији података, начину генерисања случајних тачака, као и ефикасном израчунавању растојања између објеката у m-димензионалном простору. Ови кораци представљају кључне елементе алгоритма и директно утичу на тачност и перформансе добијених резултата.

Поред саме имплементације, биће размотрени и практични аспекти примене, укључујући избор параметра k, утицај димензионалности података, као и потенцијалне оптимизације алгоритма. На тај начин омогућава се не само репродукција теоријских резултата, већ и њихова верификација у реалним условима, што представља важан корак у анализи кластерске структуре података. У наставку поглавља биће најпре описана структура програмског решења и основне функционалности система, након чега следи детаљан приказ алгоритма и анализа резултата добијених на тестним скуповима података. На овај начин се заокружује целина рада, повезујући теоријске концепте са њиховом практичном применом.

5.1 Опис програмског решења

У оквиру практичног дела рада развијено је програмско решење чији је основни циљ израчунавање Хопкинсове статистике и процена погодности задатог скупа података за кластеровање. Имплементација је реализована у програмском језику C, који омогућава ефикасну обраду података и директну контролу над меморијским ресурсима, што је од посебног значаја при раду са већим скуповима података.

Програм је осмишљен тако да прихвата улазне податке у облику табеларно организованог скупа, где сваки ред представља један објекат, а свака колона одговара одређеном атрибуту. На основу овако дефинисаног улаза, програм врши анализу просторне структуре података применом Хопкинсове статистике.

Основна функционалност програма обухвата неколико кључних корака. Најпре се врши учитавање података и њихово смештање у одговарајућу структуру у меморији. Након тога, одређују се границе посматраног простора, односно минималне и максималне вредности сваког атрибута, што је неопходно за генерисање случајних тачака. У наставку се спроводи поступак избора подскупа објеката, израчунавање растојања до најближих суседа, као и генерисање и анализа случајних тачака у складу са алгоритмом описаним у претходном поглављу.

Као резултат извршавања програма добија се вредност Хопкинсове статистике, која служи као квантитативна мера кластерске тенденције посматраног скупа података. На основу те вредности могуће је донети закључак о томе да ли подаци поседују структуру погодну за примену метода кластер анализе.

Посебна пажња приликом пројектовања решења посвећена је његовој општости и применљивости на различите скупове података. Програм је конципиран тако да омогућава рад са произвољним бројем објеката и атрибута, чиме се обезбеђује флексибилност и применљивост у различитим сценаријима анализе података.

У наставку поглавља биће детаљније описана структура улазних података, као и начин на који су поједини кораци алгоритма реализовани у програмском коду.

5.2 Структура података и улази формат

Ефикасна имплементација Хопкинсове статистике у великој мери зависи од начина организације података и њихове структуре у меморији. У оквиру овог рада, подаци се посматрају као скуп објеката представљених у m - димензионалном простору, где сваки објекат има m нумеричких атрибута. Такав приступ омогућава примену стандардних мера растојања, као што је Еуклидско растојање, које се користи у даљој анализи.

За потребе имплементације, подаци се учитавају из текстуалне датотеке, где су вредности атрибута раздвојене зарезима или размацима. Овакав формат је широко прихваћен и омогућава једноставну припрему и коришћење различитих скупова података.

У оквиру коришћеног улазног формата, први ред текстуалне датотеке садржи информацију о броју димензија, односно броју атрибута посматраних објеката (вредност m). Након овог податка следе редови који представљају појединачне објекте, при чему сваки ред садржи m нумеричких вредности. На овај начин програм аутоматски одређује димензионалност улазног простора и прилагођава даљу обраду броју атрибута без потребе за изменом изворног кода.

Приликом учитавања података, вредности се смештају у дводимензионални низ у меморији, где редови одговарају објектима, а колоне атрибутима. На овај начин омогућен је директан и ефикасан приступ појединачним елементима, што је од кључног значаја за израчунавање растојања између објеката. Поред самих података, програм чува и додатне информације као што су укупан број објеката n и број атрибута m.

За потребе генерисања случајних тачака, неопходно је одредити границе простора у коме се подаци налазе. То подразумева израчунавање минималне и максималне вредности за сваки атрибут на основу улазног скупа података. Добијене вредности се користе за генерисање случајних координата које прате равномерну расподелу унутар дефинисаног простора.

Посебан изазов представља обрада скупова података различитих величина. У том смислу, имплементација је прилагођена тако да омогућава рад са већим бројем објеката и атрибута, уз ефикасно коришћење меморијских ресурса. Ово се постиже коришћењем динамичке алокације меморије, чиме се избегавају ограничења фиксних структура података.

У наведеном примеру сваки ред представља тачку у дводимензионалном простору, при чему прва колона одговара координати x, а друга координати y. Оваква структура омогућава једноставно проширење на већи број димензија. Пре саме обраде, подаци се учитавају у одговарајућу структуру у меморији, где се представљају као матрица реалних бројева димензија $n * m$, при чему је n број објеката, а m број димензија. Правилно дефинисана структура података и флексибилан улазни формат представљају основу за даљу имплементацију алгоритма Хопкинсове статистике.

5.2.1 Учитавање података

Учитавање података реализовано је читањем из текстуалне датотеке, при чему се сваки ред интерпретира као један објекат у m - димензионалном простору. Приликом учитавања врши се парсирање вредности и њихово смештање у динамички алоцирану структуру података. У оквиру имплементације дефинисана је функција `load_data()`, која је задужена за отварање датотеке, читање података, алоцирање меморије, проверу исправности улазног формата и ослобађање меморије. Функција такође одређује број објеката и димензија, што је неопходно за даљу обраду.

```
/* Ucitavanje podataka iz datoteke */
int load_data(const char *filename, double ***data, int *n, int *m) {
    FILE* file = fopen(filename, "r");
    if (!file) return NULL;

    // ovde ide logika citanja
    // ...

    fclose(file);
    return 0;
}
```

5.3 Имплементација Хопкинсове статистике

Након учитавања података помоћу функције `load_data()`, приступа се израчунавању Хопкинсове статистике. Имплементација Хопкинсове стати-

стике у програмском језику C обухвата све кораке описане у претходним теоријским разматрањима, прилагођене директној обради података у меморији. Основни циљ је добијање вредности H која квантитативно описује кластерску тенденцију скупа података, уз могућност примене на различите величине и димензионалности података.

5.3.1 Структура програма

Програм је организован у неколико функционалних целина:

1. **Учитавање података** - Подаци се учитавају из датотеке и смештају у динамички алоцирани дводимензионални низ. Осим самих података, чувају се параметри n (број објеката) и m (број атрибута), као и минималне и максималне вредности сваког атрибута, потребне за генерисање случајних тачака.

2. **Избор подскупа објеката** - Насумично се бира k објеката из скупа података, где је $k \ll n$. Овај подскуп представља репрезентативни узорак за израчунавање растојања унутар реалних података.

3. **Израчунавање растојања до најближег суседа** - За сваки објекат x_i изабраног подскупа израчунава се растојање до његовог најближег суседа у оригиналном скупу података. Та растојања се чувају у низу $w_1, w_2, ..., w_k$.

4. **Генерисање случајних тачака** - Унутар дефинисаних граница простора генерише се k случајних тачака које прате равномерну расподелу. Ово обезбеђује поређење реалних података са насумичним распоредом.

5. **Израчунавање растојања случајних тачака** - За сваку генерисану тачку y_i израчунава се растојање до најближег објекта у оригиналном скупу података, чиме се добија низ $u_1, u_2, ..., u_k$.

6. **Израчунавање Хопкинсове статистике** - На основу добијених растојања израчунава се вредност H према формули:

$$H = \frac{\sum_{i=1}^{k} u_i}{\sum_{i=1}^{k} u_i + \sum_{i=1}^{k} w_i} \tag{5.1}$$

Ова вредност се затим користи за процену кластерске тенденције, при чему вредности приближне 0.5 указују на насумичну структуру података, док вредности ближе 1 указују на изражену кластерску структуру.

5.3.2 Организација кода у функције

Имплементација Хопкинсове статистике организована је кроз више логичких целина које одговарају појединим фазама алгоритма. Иако програм није у потпуности модуларизован у смислу издвајања свих корака у засебне функције, његова структура је јасно подељена према функционалности.

Основну целину представља функција за учитавање података, која је задужена за читање улазног скупа из датотеке и смештање вредности у одговарајућу структуру података у меморији. Ова функција такође врши основну проверу исправности улазних података.

Након учитавања, у програму се врши одређивање граница посматраног простора, односно минималних и максималних вредности за сваки атрибут. Ови параметри се касније користе за генерисање случајних тачака које прате равномерну расподелу унутар дефинисаног простора.

Избор случајних објеката из постојећег скупа података није реализован као посебна функција, већ је интегрисан у главни део алгоритма. За избор појединачног објекта користи се генератор случајних бројева, при чему се индекс објекта добија изразом `rand()` % n. На овај начин бира се један од n објеката из скупа података. Избор се врши без враћања, што значи да се након избора објекат више не разматра као кандидат за наредне изборе. Поступак се понавља k пута, при чему се за сваки наредни избор обезбеђује да претходно изабрани објекти не могу бити поново одабрани. На тај начин се добија k међусобно различитих случајно изабраних објеката који учествују у израчунавању Хопкинсове статистике.

Поред тога, у оквиру програма реализована је функција за израчунавање растојања између тачака, која представља кључни део алгоритма. На основу ове функције врши се одређивање најближег суседа како за реалне, тако и за случајно генерисане тачке.

Главна функција програма обједињује све наведене кораке. У њој се иницијализују параметри, позивају одговарајуће функције, врши избор узорака, генерисање случајних тачака, израчунавање растојања и на крају добија вредност Хопкинсове статистике.

```
/* Izracunavanje Hopkinsove statistike */
double compute_hopkins(double **data, int n, int m, int k, int r) {
    double *min = malloc(m * sizeof(double));
    double *max = malloc(m * sizeof(double));
```

```
if(min == NULL || max == NULL)
{
    printf("Greska pri alokaciji memorije.\n");

    free(min);
    free(max);

    return -1;
}

compute_bounds(data, n, m, min, max);

double sum_u = 0.0;
double sum_w = 0.0;

int *selected = calloc(n, sizeof(int));
if(selected == NULL)
{
    printf("Greska pri alokaciji memorije.\n");

    free(min);
    free(max);

    return -1;
}

for (int i = 0; i < k; i++) {
    /* Nasumicna tacka */
    double *rand_point = malloc(m * sizeof(double));
    if(rand_point == NULL)
    {
        printf("Greska pri alokaciji memorije.\n");
        free(min);
        free(max);
        return -1;
    }
    generate_random_point(rand_point, m, min, max);
    sum_u += nearest_neighbor(rand_point, data, n, m, r, -1);

    free(rand_point);
```

```
        int index;

        do {
            index = rand() % n;
        } while (selected[index]);

        selected[index] = 1;

        sum_w += nearest_neighbor(data[index], data, n, m, r, index);
    }

    free(selected);
    free(min);
    free(max);
    return sum_u / (sum_u + sum_w);
}
```

Оваква организација кода омогућава јасно праћење тока алгоритма, као и једноставно проширење и оптимизацију појединих делова програма.

5.3.3 Флексибилност и примена

Имплементација Хопкинсове статистике у оквиру овог рада пројектована је тако да буде довољно флексибилна за примену над различитим скуповима података. Програм подржава рад са подацима различитих димензија, при чему број атрибута није фиксно ограничен већ се одређује на основу улазног скупа.

Посебна пажња посвећена је начину учитавања података, при чему је омогућено читање вредности раздвојених произвољним бројем размака, табулатора или зареза, као и њиховим различитим комбинацијама. На овај начин обезбеђена је флексибилност при коришћењу скупова података у различитим текстуалним форматима.

Флексибилност имплементације огледа се и у избору параметра k, који представља број узорака за израчунавање Хопкинсове статистике. Вредност овог параметра може се прилагодити величини скупа података, чиме се постиже баланс између тачности процене и временске сложености алгоритма.

Поред тога, програм је конципиран тако да омогућава лако проширење, на пример увођењем различитих мера растојања или оптимизационих техника

за брже проналажење најближег суседа. Ово је посебно значајно код великих скупова података, где перформансе алгоритма долазе до изражаја.

Практична примена овакве имплементације огледа се у могућности брзе процене погодности података за кластеровање пре примене конкретних алгоритама кластер анализе. На тај начин се избегава непотребна примена сложених метода на скуповима који не поседују кластерску структуру.

Оваква флексибилност чини предложено решење погодним како за академска истраживања, тако и за практичне примене у областима као што су анализа података, машинско учење и системи за подршку одлучивању.

5.4 Генерисање случајних тачака

Генерисање случајних тачака представља кључни корак у имплементацији Хопкинсове статистике, јер омогућава поређење структуре реалних података са насумично распоређеним тачкама у истом простору. Основна идеја је да се унутар граница дефинисаних оригиналним скупом података генеришу тачке које прате равномерну расподелу, чиме се обезбеђује референтни модел за процену кластерске тенденције.

5.4.1 Одређивање граница простора

Да би се генерисале случајне тачке, неопходно је прво одредити опсег вредности за сваки атрибут. За сваки од m атрибута израчунавају се минимална и максимална вредност:

$$min_j = \min_{i=1,...,n} x_{ij}, \quad max_j = \max_{i=1,...,n} x_{ij}, \quad j = 1, 2, ..., m$$

Ове вредности дефинишу хиперправоугаони простор (bounding box) унутар кога ће се генерисати случајне тачке.

5.4.2 Генерисање равномерно распоређених тачака

За сваку случајну тачку y_i, њене координате се генеришу независно по свакој димензији према формули:

$$y_{ij} = min_j + r_{ij} \cdot (max_j - min_j)$$

где је $r_{ij} \in [0, 1]$ случајан број генерисан из равномерне расподеле. На овај начин свака координата тачке равномерно покрива дефинисани интервал, што обезбеђује да су генерисане тачке равномерно распоређене у целом простору.

5.4.3 Имплементација у програмском језику C

У програмском језику C, генерисање случајних бројева се најчешће реализује коришћењем функције `rand()`, која враћа цео број у интервалу `[0,RAND_MAX]`. Да би се добила реална вредност у интервалу `[0,1]`, користи се нормализација:

$$r = \frac{\texttt{rand()}}{\texttt{RAND\_MAX}}$$

Практична имплементација генерисања једне случајне тачке може се описати следећим корацима:

1. За сваки атрибут j, генерисати случајан број r_j

2. Израчунавати координату y_j према формули

3. Поновити поступак за свих m димензија

Поступак се затим понавља k пута како би се добио скуп случајних тачака потребан за даљу анализу.

Генерисање случајних тачака реализовано је функцијом која користи равномерну расподелу унутар дефинисаних граница:

```
/* Generisanje slucajne tacke */
void generate_random_point(double *point, int m,
                        double *min, double *max) {
    for (int j = 0; j < m; j++) {
        double r = (double)rand() / RAND_MAX;
        point[j] = min[j] + r * (max[j] - min[j]);
    }
}
```

На овај начин свака координата тачке равномерно покрива задати интервал, што је у складу са теоријским моделом.

5.4.4 Практични аспекти и ограничења

Иако је равномерна расподела најчешћи избор, важно је напоменути да она подразумева да су све области простора једнако вероватне. У реалним скуповима података, међутим, објекти су често концентрисани у одређеним регионима, што може довести до разлика између генерисаних и стварних структура.

Поред тога, квалитет генератора случајних бројева може утицати на резултат. Због тога је препоручљиво иницијализовати генератор функцијом `srand()` како би се избегло понављање истих секвенци:

```
srand(time(NULL));
```

Такође, у случају великих димензија (m велико), равномерно генерисане тачке могу бити „ретке“ у односу на стварне податке, што је познато као проблем проклетства димензионалности. Ипак, за потребе Хопкинсове статистике, овај приступ остаје стандард и даје поуздане резултате у пракси.

5.5 Израчунавање растојања и оптимизација

Израчунавање растојања између објеката представља један од кључних и рачунски најзахтевнијих корака у имплементацији Хопкинсове статистике. Овај корак се понавља велики број пута, како за изабране објекте из скупа података, тако и за случајно генерисане тачке, због чега има значајан утицај на укупне перформансе алгоритма.

У оквиру имплементације омогућен је избор различитих метрика растојања, при чему се користи растојање Минковског као општи облик који обухвата и Менхетн и Еуклидско растојање. На тај начин се различите метрике могу изабрати променом вредности параметра r, без потребе за посебним функцијама за сваку метрику.

5.5.1 Растојање Минковског

За две тачке $x = (x_1, x_2, ..., x_m)$ и $y = (y_1, y_2, ..., y_m)$, у m-димензионалном простору, растојање Минковског дефинише се као::

$$d(x,y) = (\sum_{j=1}^{m} \|x_j - y_j\|^r)^{\frac{1}{r}} \tag{5.2}$$

где параметар r одређује врсту растојања.

У практичној имплементацији није неопходно израчунавати корен r-тог степена приликом проналажења најближег суседа. Пошто израчунавање корена не утиче на поредак вредности растојања, приликом проналажења најближег суседа могуће је користити само суму степенованих апсолутних разлика, без израчунавања корена. Због тога се за поређење растојања користи:

$$d^r(x,y) = \sum_{j=1}^{m} \|x_j - y_j\|^r \tag{5.3}$$

чиме се избегава рачунање корена и смањује рачунска сложеност израчунавања.

У имплементацији је израчунавање реализовано једном функцијом која као параметар прима вредност r:

```
double minkowski_distanceEarlyStop(double *a, double *b, int m, int r, double
    max_dist)
{
   double sum = 0.0;

   for (int i = 0; i < m; i++) {

       double diff = fabs(a[i] - b[i]);

       sum += pow(diff, r);


       if (sum >= max_dist)
           return sum;
   }

   return sum;
}
```

Функција прима два вектора a и b, број димензија m, параметар r и тренутно најмање познато растојање max_dist. У свакој димензији израчунава

се апсолутна разлика координата, која се затим подиже на степен r и додаје тренутној суми.

5.5.2 Избор метрике растојања

Како би се омогућило поређење различитих начина мерења удаљености, имплементација подржава избор метрике путем параметра r, који се задаје као аргумент командне линије приликом покретања програма. Вредност параметра одређује врсту растојања која ће се користити: за *r=1* користи се Менхетн растојање, за *r=2* Еуклидско растојање, док се за *r≥3* користи растојање Минковског са одговарајућом вредношћу параметра r. При тестирању са великим бројем редова (око 50.000) није уочено значајно успорење за вредности параметра r≤60. За веће вредности параметра r долазило је до прекорачења, услед чега се као резултат израчунавања растојања враћа вредност -nan (not-a-number). Изабрана вредност се прослеђује функцији за израчунавање растојања приликом проналажења најближег суседа.

5.5.3 Претрага најближег суседа

За сваки објекат x_i и сваку случајну тачку y_i потребно је пронаћи најближег суседа у скупу података. У основној имплементацији, овај поступак се реализује линеарном претрагом, при чему се растојање рачуна до свих осталих објеката:

- за w_i: претрага по целом скупу података (осим самог објекта)
- за u_i: претрага по целом скупу података

Овај приступ има временску сложеност $O(n)$ по једном упиту, што доводи до укупне сложености $O(kn)$ за цео алгоритам. Претрага најближег суседа реализована је линеарним проласком кроз скуп података:

```
/* Najblizi sused */
double nearest_neighbor(double *point, double **data, int n, int m, int r, int
    index)
{
    int first = (index == 0) ? 1 : 0;
```

```
    double min_dist = minkowski_distanceEarlyStop(
        point, data[first], m, r, DBL_MAX
    );

    for (int i = 0; i < n; i++) {

        if (i == index)
            continue;

        double d = minkowski_distanceEarlyStop(
            point, data[i], m, r, min_dist
        );

        if (d < min_dist)
            min_dist = d;
    }

    return min_dist;
}
```

За сваки објекат одређује се најмање растојање, што представља основу за израчунавање вредности u_i и w_i

5.5.4 Оптимизација израчунавања растојања

С обзиром на велики број операција, могуће је применити више оптимизација:

1. **Коришћење квадрата растојања** - Избегавање израчунавања квадратног корена значајно убрзава алгоритам без утицаја на резултат.

2. **Рано прекидање рачунања** - Током израчунавања растојања, ако тренутна сума пређе већ пронађено минимално растојање, даљи прорачун се може прекинути.

3. **Локалност података** - Чување података у континуалним меморијским структурама (нпр. низови) побољшава перформансе због бољег коришћења кеш меморије.

4. **Смањење броја узорака** k - Избор адекватне вредности параметра k омогућава баланс између тачности и времена извршавања.

5.5.5 Напредне технике оптимизације

За веће скупове података могуће је применити напредније методе за претрагу најближег суседа, као што су:

- **kd-стабла (k-d tree)**
- **ball tree структуре**
- **просторни индекси**

Ове методе могу смањити сложеност претраге, али њихова имплементација је сложенија и није неопходна за основну реализацију Хопкинсове статистике.

5.5.6 Практични аспекти

У оквиру ове имплементације, примењен је једноставан и ефикасан приступ заснован на линеарној претрази, уз оптимизацију коришћењем квадрата растојања. Овај приступ је довољан за већину практичних примена и омогућава јасну и разумљиву реализацију алгоритма.

Важно је напоменути да се са повећањем броја објеката n и броја атрибута m значајно повећава време извршавања, што представља ограничење ове методе. Ипак, за потребе анализе кластерске тенденције, добијени резултати су поуздани и у складу са теоријским очекивањима.

5.6 Експериментални резултати

У циљу провере исправности и ефикасности имплементације Хопкинсове статистике, извршени су експерименти над више различитих скупова података. Тестирање је обухватило како синтетичке скупове података, тако и различите типове расподела, са циљем да се испита понашање алгоритма у контролисаним условима.

5.6.1 Опис тест скупова података

За тестирање је коришћен реални скуп података Iris, преузет са интернета, док су остали скупови података вештачки генерисани за потребе експерименталне анализе. Улазни подаци смештени су у датотеке *data_iris, data_large,*

data_large_with_outlier, *data_random_large* и *data_uniform_large*. Осим скупа Iris, који садржи 150 објеката са четири нумеричка атрибута, остали скупови садрже по 100 објеката са два атрибута. На овај начин омогућено је испитивање понашања имплементације над скуповима података различитих карактеристика.

- *data_iris* – реални скуп података Iris, који садржи 150 објеката описаних помоћу четири нумеричка атрибута.
- *data_large* – вештачки генерисан скуп од 100 објеката са два атрибута, намењен испитивању рада алгоритма над већим бројем објеката.
- *data_large_with_outlier* – вештачки генерисан скуп од 100 објеката са два атрибута, који садржи и издвојене вредности (outlier-е), ради испитивања њиховог утицаја на резултат.
- *data_random_large* – вештачки генерисан скуп од 100 објеката са два атрибута, са насумично распоређеним вредностима.
- *data_uniform_large* – вештачки генерисан скуп од 100 објеката са два атрибута, чије су вредности равномерно распоређене у задатом интервалу.

5.6.2 Резултати експеримента

На основу извршавања програма добијене су следеће вредности Хопкинсове статистике:

Тип података	**Вредност** H
Насумични подаци	0.48
Кластерски подаци	0.97
подаци са шумом	0.8
Равномерно распоређени подаци	0.28

5.6.3 Анализа резултата

Добијени резултати су у складу са теоријским очекивањима. За насумично распоређене податке, вредност Хопкинсове статистике је приближна $H = 0.5$,

што указује на одсуство кластерске структуре. Ово потврђује да алгоритам исправно препознаје насумичну природу података.

Са друге стране, за кластерски распоређене податке добијена је значајно већа вредност ($H = 0.97$), што указује на јасно изражену кластерску тенденцију. Овај резултат показује да имплементација успешно детектује присуство кластера у подацима. Када у подацима са кластерском структуром имамо шумове и елементе ван граница, вредност Хопкинсове статистике је *(H = 0.8)*, што и даље указује на постојање кластера.

За равномерно распоређене податке, вреднсот Хопкинсове статистике је приближна $H = 0$. Ово такође потврђује да алгоритам добро ради и да за равномерно распоређене податке не налази кластере.

5.6.4 Тестирање имплементације на скупу података Iris

За проверу исправности имплементиране Хопкинсове статистике коришћен је Iris скуп података, који представља један од најчешће коришћених скупова у области анализе података и машинског учења. Скуп садржи 150 објеката, при чему је сваки објекат описан са четири нумеричка атрибута: дужина чашичног листића *(Sepal Length)*, ширина чашичног листића *(Sepal Width)*, дужина круничног листића *(Petal Length)* и ширина круничног листића *(Petal Width)*.

Оригинални скуп података садржи и категоријски атрибут Species, који означава припадност једној од три врсте цвета: setosa, versicolor и virginica. Пошто се Хопкинсова статистика заснива искључиво на просторном распореду објеката, овај атрибут није коришћен у израчунавању, већ су у обзир узете само нумеричке карактеристике.

За потребе тестирања, подаци су прилагођени улазном формату који користи имплементирани програм. На почетку текстуалне датотеке наведен је број димензија посматраног простора. Након тога сваки ред датотеке садржи четири нумеричке вредности које представљају један објекат у четвородимензионалном простору. Категоријска ознака врсте је уклоњена јер није релевантна за израчунавање растојања. Након припреме података, извршено је тестирање имплементације коришћењем различитих метрика растојања, при чему је избор метрике извршен путем аргумента командне линије.

Добијене вредности Хопкинсове статистике биле су блиске вредности 1. Такав резултат указује да посматрани скуп података није равномерно распо-

рeђен у простору, већ да постоји изражена тенденција ка формирању кластера. Ово је у складу са познатом структуром Iris скупа, где се различите врсте цветова разликују по својим мереним карактеристикама, посебно по димензијама круничног листића. Вредност Хопкинсове статистике добијена експериментом потврђује да имплементирани алгоритам правилно идентификује постојање кластерске структуре у подацима.

5.6.5 Утицај параметра k

Током експеримената уочено је да вредност параметра k (параметар k представља број узорака који се користе за процену Хопкинсове статистике, односно, број реалних и случајних тачака које учествују у рачунању) утиче на стабилност резултата. Мање вредности k доводе до већих осцилација у вредности статистике, док веће вредности обезбеђују стабилније резултате, али уз повећање времена извршавања.

5.6.6 Дискусија

Резултати експеримента показују да имплементирана Хопкинсова статистика пружа поуздану процену погодности података за кластеровање. Алгоритам успешно разликује насумичне и структуриране скупове података, што потврђује његову практичну применљивост. Иако су експерименти изведени на релативно малим скуповима података, добијени резултати указују да се исти принципи могу применити и на веће скупове, уз одговарајуће оптимизације описане у претходном поглављу.

5.7 Покретање програма

Имплементирани програм омогућава израчунавање Хопкинсове статистике уз избор различитих метрика растојања. Програм је реализован тако да се параметри неопходни за извршавање задају путем аргумената командне линије, чиме се омогућава једноставна промена улазног скупа података и начина израчунавања без измене изворног кода. Програм је превођен и покретан у *Windows* окружењу, коришћењем *GCC* компајлера. Пре покретања програма потребно је извршити компајлирање изворног кода. Приликом превођења коришћен је ниво оптимизације -O3, који омогућава компајлеру да примени на-

предне оптимизације са циљем побољшања перформанси извршног програма. Програм се може превести следећом командом:

```
gcc Hopkins_statist.c -O3 -o hopkins.exe -lm
```

Опција -O3 укључује виши ниво оптимизације кода, што може допринети смањењу времена извршавања програма, посебно код операција које се велики број пута понављају током израчунавања растојања. Опција -lm служи за повезивање математичке библиотеке која се користи за функције из заглавља math.h.

Након успешног компајлирања, програм се покреће задавањем назива улазне датотеке и вредности параметра r. Општи облик покретања програма је:

```
hopkins.exe [naziv_datoteke] [r]
```

где naziv_datoteke представља назив датотеке која садржи улазни скуп података, док параметар r одређује метрику растојања. Вредност параметра може бити:

r=1 – Менхетн растојање;

r=2 – Еуклидско растојање;

r≥3 – Растојање Минковског са одговарајућом вредношћу параметра r.

На овај начин могуће је променити улазни скуп података или метрику растојања једноставном изменом аргумената командне линије, без потребе за изменом изворног кода.

Глава 6

Закључак

У овом раду анализиран је проблем процене погодности података за примену метода кластеровања, са посебним осврtом на идентификацију постојања кластерске структуре у скупу података. Полазећи од теоријских основа кластер анализе, истакнуто је да успешност кластеровања у великој мери зависи од карактеристика самих података, као и од њихове припреме и избора одговарајућих мера сличности. Посебно су наглашени изазови као што су висока димензионалност, присуство шума и аномалија, као и различите размере атрибута, који могу значајно утицати на квалитет резултата.

Кроз разматрање појма кластерске тенденције показано је да није сваки скуп података погодан за примену кластеровања, те да је неопходно спровести иницијалну процену пре избора конкретног алгоритма. У том контексту, поред визуелних метода, посебно је истакнута Хопкинсова статистика као ефикасна и широко прихваћена квантитативна мера за процену степена структуираности података.

У раду је детаљно описан алгоритам израчунавања Хопкинсове статистике, укључујући избор подскупа података, генерисање случајних тачака и израчунавање растојања до најближих суседа. Показано је да вредност ове статистике пружа јасну интерпретацију природе података: вредности приближне 0.5 указују на насумичност, вредности ближе 1 на постојање кластерске структуре, док вредности ближе 0 могу указивати на равномерну расподелу.

Практични део рада обухвата имплементацију Хопкинсове статистике у програмском језику C, при чему је посебна пажња посвећена организацији кода и оптимизацији алгоритма. За израчунавање растојања користи се Минковскијево растојање, које омогућава примену различитих метрика у зависно-

сти од вредности параметра r.

Резултати експеримената потврдили су теоријске претпоставке. Имплементирана метода успешно разликује насумичне и кластерски организоване податке, а анализа утицаја параметра k указује на потребу пажљивог избора величине узорка ради постизања стабилних и поузданих резултата.

На основу свега изложеног може се закључити да Хопкинсова статистика представља једноставан, али изузетно користан алат за процену погодности података за кластеровање. Њена примена омогућава боље разумевање структуре података и доношење информисаних одлука о даљој анализи.

Као правци будућег рада могу се издвојити унапређење перформанси кроз примену напреднијих структура података за претрагу најближег суседа, као и проширење примене на високодимензионалне и реалне скупове података.

За крај, правилна процена погодности података не представља само уводни корак, већ суштински предуслов за успешно кластеровање, јер управо она одређује да ли ће резултати анализе бити смислени, поуздани и применљиви у реалним условима.

Додатак А

Изворни код програма

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>


void free_data(double **data, int n)
{
    if(data == NULL)
        return;

    for(int i = 0; i < n; i++)
    {
        free(data[i]);
    }

    free(data);
}

int load_data(const char *filename, double ***data, int *n, int *m)
{
    FILE *f = fopen(filename, "r");

    if (f == NULL) {
        printf("Greska pri otvaranju datoteke.\n");
        return 1;
    }
```

```
// Ucitavanje broja atributa
if (fscanf(f, "%d", m) != 1) {
    printf("Greska pri ucitavanju broja atributa.\n");
    fclose(f);
    return 1;
}

int capacity = 100;

*data = malloc(capacity * sizeof(double *));

if (*data == NULL) {
    printf("Greska pri alokaciji memorije.\n");
    fclose(f);
    return 1;
}

*n = 0;

while (1) {

    // Ako nema vise prostora, povecaj niz pokazivaca
    if (*n >= capacity) {

        capacity *= 2;

        double **temp = realloc(*data,
                                capacity * sizeof(double *));

        if (temp == NULL) {

            printf("Greska pri realokaciji memorije.\n");

            for(int i = 0; i < *n; i++)
                free((*data)[i]);

            free(*data);

            fclose(f);

            return 1;
        }
```

```
        *data = temp;
    }


    // Privremeni red podataka
    double *row = malloc((*m) * sizeof(double));

    if (row == NULL) {
        printf("Greska pri alokaciji reda.\n");

        for(int i = 0; i < *n; i++)
            free((*data)[i]);

        free(*data);

        fclose(f);
        return 1;
    }

    // Pokusaj citanja prvog atributa
    if (fscanf(f, "%lf", &row[0]) != 1) {
        free(row);
        break;
    }

    // Citanje ostalih atributa
    for (int j = 1; j < *m; j++) {

        int c;

        do {
            c = fgetc(f);
        } while (c == ' ' || c == '\t' || c == ',');

        if (c != EOF)
            ungetc(c, f);

        if (fscanf(f, "%lf", &row[j]) != 1) {

            free(row);
```

```
            for (int i = 0; i < *n; i++)
                free((*data)[i]);

            free(*data);
            fclose(f);

            return 1;
        }
    }

    // Dodavanje reda u matricu
    (*data)[*n] = row;
    (*n)++;
}

fclose(f);
return 0;
}

double minkowski_distanceEarlyStop(double *a, double *b, int m, int r, double
    max_dist)
{
    double sum = 0.0;

    for (int i = 0; i < m; i++) {

        double diff = fabs(a[i] - b[i]);

        sum += pow(diff, r);


        if (sum >= max_dist)
            return sum;
    }

    return sum;
}

double nearest_neighbor(double *point, double **data, int n, int m, int r, int
    index)
{
    int first = (index == 0) ? 1 : 0;
```

```
    double min_dist = minkowski_distanceEarlyStop(
        point, data[first], m, r, DBL_MAX
    );

    for (int i = 0; i < n; i++) {

        if (i == index)
            continue;

        double d = minkowski_distanceEarlyStop(
            point, data[i], m, r, min_dist
        );

        if (d < min_dist)
            min_dist = d;
    }

    return min_dist;
}

/* Odredjivanje granica prostora */
void compute_bounds(double **data, int n, int m, double *min, double *max) {
    for (int j = 0; j < m; j++) {
        min[j] = data[0][j];
        max[j] = data[0][j];

        for (int i = 1; i < n; i++) {
            if (data[i][j] < min[j]) min[j] = data[i][j];
            if (data[i][j] > max[j]) max[j] = data[i][j];
        }
    }
}


/* Generisanje slucajne tacke */
void generate_random_point(double *point, int m, double *min, double *max) {
    for (int j = 0; j < m; j++) {
        double r = (double)rand() / RAND_MAX;
        point[j] = min[j] + r * (max[j] - min[j]);
    }
}
```

```
/* Izracunavanje Hopkinsove statistike */
double compute_hopkins(double **data, int n, int m, int k, int r) {
    double *min = malloc(m * sizeof(double));
    double *max = malloc(m * sizeof(double));

    if(min == NULL || max == NULL)
    {
        printf("Greska pri alokaciji memorije.\n");

        free(min);
        free(max);

        return -1;
    }

    compute_bounds(data, n, m, min, max);

    double sum_u = 0.0;
    double sum_w = 0.0;

    int *selected = calloc(n, sizeof(int));
    if(selected == NULL)
    {
        printf("Greska pri alokaciji memorije.\n");

        free(min);
        free(max);

        return -1;
    }

    for (int i = 0; i < k; i++) {
        /* Nasumicna tacka */
        double *rand_point = malloc(m * sizeof(double));
        if(rand_point == NULL)
        {
            printf("Greska pri alokaciji memorije.\n");
            free(min);
            free(max);
            return -1;
        }
```

```
        generate_random_point(rand_point, m, min, max);
        sum_u += nearest_neighbor(rand_point, data, n, m, r, -1);

        free(rand_point);

        int index;

        do {
            index = rand() % n;
        } while (selected[index]);

        selected[index] = 1;

        sum_w += nearest_neighbor(data[index], data, n, m, r, index);
    }

    free(selected);
    free(min);
    free(max);
    return sum_u / (sum_u + sum_w);
}

/* Glavni program */
int main(int argc, char *argv[]) {
    /*double data[MAX_N][MAX_M];*/
    double **data;
    int n;
    int m;
    char metric;

    if (argc != 3) {

        printf("Upotreba: %s naziv_datoteke r\n", argv[0]);
        printf("1 - Menhetn rastojanje\n");
        printf("2 - Euklidsko rastojanje\n");
        printf("<=3 - Rastojanje Minkovskog\n");

        return 1;
    }

        int r = atoi(argv[2]);
```

```
    if (r < 1)
    {
        printf("Parametar r mora biti najmanje 1.\n");
        return 1;
    }

    srand(time(NULL));
    char *filename = argv[1];

    if(load_data(filename, &data, &n, &m) != 0)
    {
        return 1;
    }

    if(n == 0)
    {
        printf("Datoteka nema podatke.\n");
        free_data(data,n);
        return 1;
    }

    /*int k = n / 2;*/
    int k = (int)sqrt(n);
    if (k < 10) k = 10;
    if (k > n) k = n / 2;
    if (k > 20) k = 20;
    double H = compute_hopkins(data, n, m, k, r);

    printf("Hopkins statistika - H = %lf\n", H);

    free_data(data, n);
    return 0;
}
```

Библиографија

[1] Arindam Banerjee and Joydeep Ghosh. Validating clusters using the hopkins statistic. *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 149–153, 2004.

[2] James C. Bezdek and Richard J. Hathaway. Vat: A tool for visual assessment of cluster tendency. *Proceedings of the International Joint Conference on Neural Networks*, 2002.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Waltham, MA, USA, 3 edition, 2012.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[6] Brian Hopkins and John G. Skellam. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18(2):213–227, 1954.

[7] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[8] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[9] S. B. Kotsiantis. Data preprocessing for supervised learning. *International Journal of Computer Science*, 1(2):111–117, 2006.

[10] A. B. Lawson and D. G. Jurs. New index for clustering tendency and its application to chemical problems. *Journal of Chemical Information and Computer Sciences*, 30(1):36–41, 1990.

[11] Viktor Mayer-Schönberger and Kenneth Cukier. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt, Boston, MA, USA, 2013.

[12] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to Data Mining*. Pearson, 2nd edition, 2019.

[13] Rui Xu and Donald Wunsch. *Clustering*. Wiley-IEEE Press, Hoboken, 2009.