KLAIPĖDOS UNIVERSITETAS
GAMTOS IR MATEMATIKOS MOKSLŲ FAKULTETAS
INFORMATIKOS KATEDRA

ALEKSAS NARŠČIUS
InfMag06 gr. studentas

DAUGIAMAČIŲ DUOMENŲ RINKINIŲ TYRIMAS TAIKANT MODIFIKUOTĄ RENYI-CLIPX KLASTERIZACIJOS ALGORITMĄ

Baigiamasis magistro darbas

Mokslinio darbo vadovė: dr. D. Baziukaitė

KLAIPĖDA, 2008

Klaipėdos universitetas
Gamtos ir matematikos mokslų fakultetas
Informatikos katedra

Baigiamasis magistro darbas

DAUGIAMAČIŲ DUOMENŲ RINKINIŲ TYRIMAS TAIKANT MODIFIKUOTĄ RENYI-CLIPX KLASTERIZACIJOS ALGORITMĄ

Aleksas Narščius

ANOTACIJA

Šiame baigiamajame magistro darbe didžiausias dėmesys skirtas daugiamačių duomenų rinkinių analizei, kuriai atlikti naudojami konceptualios klasterizacijos algoritmai. Tokia analizė susideda iš dviejų esminių dalių – tai klasterių duomenų rinkinyje radimo ir klasterius aprašančių taisyklių suformavimo. Analizuojami duomenų rinkiniai gali būti įvairūs. Šiuo atveju mus labiausiai domino realių iš virtualios mokymo(si) aplinkos gaunamų duomenų rinkinių, aprašančių besimokančiųjų pasiekimus, analizė. Algoritmu gautų rezultatų gerumas papildomai buvo patikrintas testuojant duomenų rinkinį su įvardintu klasės atributu. Tai leido įsitikinti gautų rezultatų patikimumu. Realizuoti algoritmai ir atlikto tyrimo rezultatai svarbūs adaptyvios intelektualios virtualios mokymo(si) aplinkos kūrimo etape bei bus naudinga algoritmus diegiant realioje VMA. Tai taip pat sukurs prielaidas kokybiškesnėms savarankiškoms studijoms, kai yra taikomos šiuolaikinės informacinės komunikacinės technologijos.

Darbe analizuotų Renyi entropijos, ITERATE, CLIP3 ir CLIP4 klasterizacijos algoritmų pagrindu, sudarytas modifikuotas Renyi-CLIPx klasterizacijos algoritmas, kuris pagerina sudaromų klasterių ir taisyklių kokybę virtualios mokymo(si) aplinkos bei kitokios prigimties duomenų rinkiniams.

Raktiniai žodžiai: klasterizacija, konceptuali klasterizacija, klasterizacijos taisyklės, Virtuali mokymo(si) aplinka, Renyi, ITERATE, CLIP3, CLIP4, modifikuotas Renyi-CLIPx algoritmas.

Klaipeda University
Faculty of Natural Sciences and Mathematics
Computer Science Department

Master's Thesis

MULTIDIMENSIONAL DATA ANALYSIS USING MODIFIED RENYI-CLIPX CONCEPTUAL CLUSTERING ALGORITHM

Aleksas Narščius

ABSTRACT

In this master's paper a great attention is paid for multidimensional data analysis by conceptual clustering algorithms. This analysis includes two aspects: discovering of clusters and clustering rules form various data sets. These rules help to decide to which cluster newly appearing object has to be assigned. We also are interested in analysis of virtual learning system's data sets which contains such information as grades of students. Clustering results are tested by using training data which real clusters are well known. Made results indicates about a very good clustering solution. Discovered algorithms and results of experiments are important for developing of virtual learning system. We suppose that it is useful for making better self-studies with modern communication technologies.

Modified Renyi-CLIPx clustering algorithm is proposed as result of analysis Renyi entropy, ITERATE, CLIP3 and CLIP4 clustering algorithms. It is used to discover better clusters and create rules for virtual learning and other data sets.

Key words: clustering, conceptual clustering, clustering rules, virtual learning system, Renyi entropy, ITERATE, CLIP3, CLIP4, modified Renyi-CLIPx algorithm.

TURINYS

LENTELIŲ RODYKLĖ

PAVEIKSLĖLIŲ RODYKLĖ

ĮVADAS

Duomenų gavyba (angl. *Data mining*) dažnai remiasi klasterių sudarymu ir taisyklių išvedimu. Klasterizacijos proceso metu iš duomenų rinkinio objektų formuojamas tam tikras klasterių skaičius. Siekiama užtikrinti, kad panašius požymius turintys duomenys yra priskirti tam pačiam klasteriui, o objektai, turintys esminių skirtumų, – skirtingiems klasteriams. Dažnai klasterizacijos algoritmai taikomi skaitiniams duomenims, bet pasitaiko rinkinių, kurių duomenys gali būti nominalūs arba skaitiniai-nominalūs tuo pačiu metu. Tokių duomenų klasterizacijai taikomi specifiniai algoritmai. Čia siūloma naudoti ne klasterizacijos, o konceptualios klasterizacijos sąvoką [6]. Konceptualios klasterizacijos algoritmai formuoja taisykles, pagal kurias naujus, į duomenų rinkinį įvestus, egzempliorius galima priskirti klasteriams. Tokių taisyklių sudarymas yra aktualus adaptyvios intelektualios virtualios mokymo(si) aplinkos (*AI VMA*) duomenų apie besimokančiuosius analizės uždavinys [5].

Virtuali mokymo(si) aplinka (*VMA*) – tai mokomosios medžiagos, užduočių, bendravimo ir vertinimo priemonių sistema, leidžianti el. erdvėje valdyti mokymo procesą. Tokias aplinkas naudoja daugelis universitetų, kolegijų ir įvairių kitų mokymo įstaigų visame pasaulyje. Pastaruoju metu *VMA* labai paplito ir Lietuvoje. Šis darbas yra sudedamoji *AI VMA* plėtojimo dalis [4]. Pagrindinis šio projekto tikslas – sukurti priemones, leidžiančias nuotolinių kursų kuratorius dalinai pakeisti intelektualiais agentais, kurie galėtų užtikrinti kokybiškesnes savarankiškas studijas. *AI VMA* – tai tokia mokymosi aplinka, kurioje kurso kuratoriaus vaidmenį pakeičia kompiuterizuota sistema, turinti programiniams agentams būdingų bruožų. Ši aplinka ypatinga ne tik tuo, kad beveik visus kuratoriaus vaidmenis atlieka agentas, bet ir tuo, kad jos pagrindinis tikslas – prisitaikyti prie besimokančiųjų poreikių, užtikrinti specialių kursų pateikimą ir intensyvumą pagal besimokančiųjų žinių lygį ir pasiruošimą. Ankstesniuose darbuose [6] suformuluotoms koncepcijom įgyvendinti reikia turėti tam tikrus programinius įrankius. Šie įrankiai turi:

1. Leisti agentui nuspręsti, kokius veiksmus atlikti.
2. Suskirstyti besimokančiuosius į tam tikras grupes – klasterius.

Realizuojant programinį agentą, būtina turėti informacijos apie besimokančiuosius. Pastaroji yra kaupiama *VMA* duomenų bazėse, kuriose saugomi besimokančiųjų žinių patikrinimo įvertinimai. Besimokančiųjų žinių lygis nusakomas duomenimis, gaunamais jų kontrolės (savikontrolės) metu. Taip pat reikia realizuoti algoritmus, leidžiančius suskirstyti besimokančiuosius į atskirus klasterius pagal šios probleminės srities duomenis. Atsižvelgdami į tai formuluojame tiriamojo darbo tikslą ir uždavinius.

Tikslas:

Pagerinti klasterizacijos algoritmu sudaromų klasterių ir juos aprašančių taisyklių kokybę probleminės srities duomenų rinkiniams.

Uždaviniai:

1. Aprašyti ir palyginti probleminės srities duomenų klasterizacijos algoritmus.
2. Realizuoti algoritmą, kuris probleminės srities duomenims sudaro klasterius ir generuoja objektų priskyrimo klasteriams taisykles.
3. Identifikuoti veiksmus, kuriuos reikia atlikti, kad realizuotas klasterizacijos algoritmas sudarytų gerai atskirtus klasterius ir nesudėtingas taisykles.
4. Patikrinti ir įvertinti realizuoto klasterizacijos algoritmo darbo efektyvumą testiniams ir realiems probleminės srities duomenimis.

Tolimesniame darbo skyriuje atliekama mokslinėse publikacijose dažnai minimų klasterizacijos algoritmų lyginamoji analizė. Antrajame, trečiajame ir ketvirtajame darbo skyriuose aptariamos duomenų rinkinių normalizacijos, diskretizacijos ir išskirčių nustatymo problemos. Penktajame skyriuje pristatomas tirtų algoritmų pagrindu sudarytas modifikuotas klasterizacijos algoritmas, kuris pagerina sudaromų klasterių ir taisyklių kokybę virtualios mokymo(si) aplinkos bei kitokios prigimties duomenų rinkiniams. Pagal gautus rezultatus suformuotos išvados pateiktos šeštajame skyriuje.

Šio magistrinio darbo aktualumas ir rezultatai autoriaus aprobuoti Informacinių technologijų 2007 (KTU) [7], Klaipėdos universiteto GMMF studentų mokslinės draugijos 10-ojoje [22], Klaipėdos universiteto GMMF studentų mokslinės draugijos 11-ojoje [21] ir 6-ojoje tarptautinėje „Technologijos mokslo darbai Vakarų Lietuvoje“ [23] konferencijose.

1. KLASTERIZACIJOS ALGORITMŲ LYGINAMOJI ANALIZĖ

Mokslinėse publikacijose dažnai minimi ir cituojami Renyi entropijos, CLIP3, CLIP4 ir ITERATE klasterizacijos algoritmai. Atsižvelgiant į tai siūloma geriau susipažinti su šių algoritmų veikimo principais ir juos palyginti. Tiriant pastaruosius algoritmus, būtina parinkti kriterijus, pagal kuriuos atliekama jų lyginamoji analizė. Įvedami šie keturi kriterijai [22]:

- klasterių iniciavimas,
- panašumo (skirtumo) tarp objektų nustatymas,
- optimalaus klasterių skaičiaus išrinkimas,
- algoritmo sudėtingumas.

Daugumai klasterizacijos algoritmų pradinių klasterių iniciavimo etapas svarbus, nes nuo sukurtų klasterių prototipų dažnai priklauso algoritmo sudaromų klasterių gerumas ir pastovumas. Klasterių pastovumas susijęs su panašumų bei skirtumų tarp duoto duomenų rinkinio objektų nustatymo, nes panašius esminius požymius turintys objektai turi būti priskirti tam pačiam klasteriui. Siekiant kad pateiktas duomenų rinkinys būtų suskirstytas į natūralias grupes, turi būti parinktas optimalus klasterių skaičius. Optimalaus klasterių skaičiaus parinkimas yra gana aktuali mokslinėse publikacijose minima problema [1, 12, 16]. Kai kuriais atvejais šį skaičių gali parinkti žmogus, bet norima, kad optimalų klasterių skaičių pasiūlytų pats algoritmas. Taigi pastarieji įvesti kriterijai yra svarbūs ir apima visą duoto duomenų rinkinio klasterizacijos procesą. Renyi entropijos, CLIP3, CLIP4 ir ITERATE konceptualios klasterizacijos algoritmai yra palyginti šių kriterijų požiūriu [7].

1.1. Renyi entropijos klasterizacijos algoritmas

Renyi entropijos klasterizacijos algoritmas generuoja pateikto duomenų rinkinio klasterius atsižvelgdamas į tarpklasterinę entropiją [16]. Užtikrinama, kad sukurti klasteriai yra atskirti didžiausios entropijos tarp klasterių požiūriu, kai tuo tarpu entropija tarp klasteryje esančių objektų yra mažiausia. Naujas duomenų rinkinio objektas priskiriamas tam klasteriui, kurio vidinės entropijos pokytis yra mažiausias [16]. Kadangi atliekami entropijos skaičiavimai, tai pateiktame duomenų rinkinyje gali būti tik skaitinės stebėjimų reikšmės. Renyi entropijos algoritmas nenagrinėja atvejų, kai pateiktame duomenų rinkinyje yra nežinomų stebėjimų reikšmių.

1.1.1. Klasterių iniciavimas

Renyi entropijos klasterizacijos algoritmo vykdymo pradžioje yra sukuriama K_{init} klasterių [16]. Tai atliekama atsitiktiniu būdu parenkant K_{init} objektų iš viso objektų rinkinio. Tokiu būdu kiekvienas parinktas taškas reiškia sudaromo klasterio užuomazgą arba prototipą. Po to objektas įtraukiamas į tą klasterį, šalia kurio nario jis randasi arčiausiai. Tai tęsiama tol, kol klasterių užuomazgoms yra priskiriamas nurodytas objektų skaičius N_{init} arba nebelieka nepriskirtų objektų. Tokiu būdu siekiama užtikrinti, kad sudarytas duomenų grupavimas būtų kuo jautresnis duomenų struktūrai [17].

Kiti objektai iš likusio pateikto duomenų rinkinio klasterizavimui gali būti parinkti keliais būdais [17]:

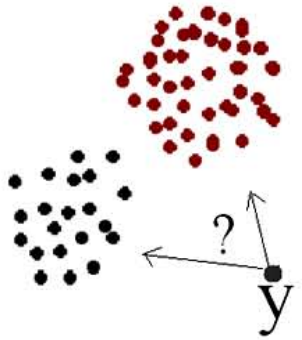
- Objektas parenkamas atsitiktiniu būdu. Šis būdas nerekomenduojamas, nes taip parinktas taškas, esantis toli nuo pradinių klasterių, gali sudaryti nepastovius klasterius. Todėl vykdant Renyi entropijos konceptualios klasterizacijos algoritmą kiekvieną kartą gali būti sugeneruoti skirtingi klasteriai.
- Parenkant objektą, esantį arčiausiai klasterio prototipo. Taip sudaromi klasteriai yra stabilesni. Klasterio prototipas gali būti jo vidurkis. Tuomet tiriamas objektas įtraukiamas į tą klasterį, kurio vidurkiui jis yra artimiausias. Kai duomenų rinkinio objektai nusakomi keliais požymiais (atributais), šis prototipo parinkimas nėra tikslingas, nes objekto atributų matavimo dydžiai gali nesutapti. Todėl reikia taikyti kitą objektų priskyrimo būdą, kuris remiasi entropijos pokyčio stebėjimu.

Patariama inicijuoti kuo didesnį pradinių klasterių skaičių K_{init} [16]. Pavyzdžiui, jei turime duomenų rinkinį, kurį sudaro T skirtingų objektų, tai pradinis klasterių skaičius K_{init} gali būti lygus $T \cdot 0{,}20$, o inicijuojamas objektų skaičius N_{init} kiekviename klasteryje lygus 3. Tuomet likusius 40% duomenų rinkinio objektų reikia priskirti klasteriams pagal artumą atitinkamo klasterio prototipui. Tokiu būdu priskyrus visus objektus klasteriams pradedami formuoti gana pastovūs klasteriai [16]. Sekančiuose algoritmo žingsniuose nustatomi blogai sudaryti klasteriai ir perskirstomi jų objektai kitiems klasteriams.

Pastaba *p1*: Renyi konceptualios klasterizacijos algoritmo pradinių klasterių inicijavimas atsitiktiniu būdu yra diskutuotinas veiksmas. Tai lemia klasterių pastovumą ir objektų priskyrimą klasteriams. Todėl kelis kartus vykdant Renyi entropijos konceptualios klasterizacijos algoritmą tam pačiam duomenų rinkiniui objektų priskyrimas klasteriams gali būti kitoks.

1.1.2. Panašumo (skirtumo) tarp objektų nustatymas

Panagrinėkime klasterių sudarymo problemą (1 pav.). Tegu atliekant klasterizacijos procesą duoto duomenų rinkinio X objektai priskiriami klasteriams C_1 ir C_2. Į duomenų rinkinį įtraukus naują objektą y iškyla problema: kuriam klasteriui priskirti šį objektą. Ši problema būdinga daugeliui klasterizacijos algoritmų. Ją galima spręsti naudojant panašumų (skirtumų) tarp nagrinėjamų objektų nustatymo metodus.



1 pav. ***Klasterių sudarymo problema***

Įtraukus naują objektą į klasterį, pasikeičia to klasterio entropija [8]. Čia entropija yra nagrinėjama informacijos teorijos kontekste ir apibrėžiama kaip neapibrėžtumo kiekis sistemoje. Jeigu objektas y klaidingai priskiriamas klasteriui C_1, tai klasterio C_1 entropija padidėja labiau, nei šį objektą teisingai priskyrus į klasterį C_2. Taigi laikomasi šios taisyklės: objektas y priskiriamas tam klasteriui, kurio entropijos pokytis po šio priskyrimo yra mažiausias. Apibendrinant sakoma, kad jei turime K skirtingų klasterių C_k, kur $k = 1,\ldots,K$, tai objektas y priskiriamas klasteriui C_i, jeigu yra teisinga

$$H(C_i + y) - H(C_i) < H(C_k + y) - H(C_k), \quad (1)$$

kiekvienam $k = 1,\ldots,K$, $k \neq i$ [15]. Čia $H(C_k)$ žymi klasterio C_k entropiją, o $H(C_k + y)$ – klasterio C_k entropiją, kai į jį įtraukiamas naujas objektas y. Taigi iš entropijos klasterio viduje galima spręsti, kuriam klasteriui priskirti objektą y. Tačiau iškyla kita problema – entropijos apskaičiavimas tiesiogiai iš pateikto duomenų rinkinio.

Entropiją galima tiesiogiai apskaičiuoti iš duomenų rinkinio pagal A. Renyi pasiūlytą principą [16]. Mokslinėse publikacijose šiuo pasiūlytu būdu skaičiuojama entropija vadinama Renyi entropija. 1960 m. A. Renyi entropiją stochastiniam kintamajam X, kurio tikimybinio tankio funkcija f_X, apibrėžia taip:

$$H_R(X) = \frac{1}{1-\alpha} \ln \int f_x^{\alpha} dx, \text{ kai } \alpha > 0 \text{ ir } \alpha \neq 1. \quad (2)$$

Norint tiesiogiai apskaičiuot Renyi entropiją iš pateikto duomenų rinkinio, reikia įvertinti tikimybinio tankio funkciją f_X. Vienas iš galimų sprendimų – Parzeno lango (angl. k. *Parzen Window*) metodas su Gauso branduolio (angl. k. *Gaussian kernel*) funkcija [16]. Šis metodas nagrinėja daugiamatę Gauso funkciją:

$$G(x)=\frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)|\Sigma|^{-1}(x-\mu)^T\right). \quad (3)$$

Čia: Σ – kovariacijos matrica, μ – duomenų rinkinio (klasterio) vidurkis, k – duomenų rinkinyje (klasteryje) esančių egzempliorių (objektų) skaičius. Galima daryti prielaidą, kad kovariacijos matrica yra lygi $\Sigma=\delta^2 I$, kur I – vienetinė matrica, δ – standartinis nuokrypis [17]. Tuomet Renyi entropija tiesiogiai apskaičiuojama iš duomenų.

Tegu klasterį C_k sudaro N_k diskrečių duomenų rinkinio objektų x_i, kur $i=1,\ldots,N_k$. Tuomet klasterio C_k tikimybinio tankio funkcija užrašoma tokiu pavidalu [16]:

$$f_X=\frac{1}{N_k}\sum_{i=1}^{N_k}G(x-x_i,\sigma^2 I). \quad (4)$$

Kai $\alpha=2$, tai iš (2) gauname lygtį, kuri vadinama Renyi kvadratine entropija [16]:

$$H_R(X)=-\ln\int f_X^2 dx. \quad (5)$$

Tuomet įstačius (4) į (5) gauname, kad

$$H_R(X)=-\ln\int\left(\frac{1}{N}\sum_{i=1}^{N}G(x-x_i,\delta^2)\right)\left(\frac{1}{N}\sum_{j=1}^{N}G(x-x_j,\delta^2)\right)dx=-\ln\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}G(x_i-x_j,2\delta^2). \quad (6)$$

Čia N – objektų skaičius nagrinėjamame duomenų rinkinyje (klasteryje).

Iš (6) matoma, kad panaudojant Parzeno langą su Gauso branduolio funkcija, pateikto duomenų rinkinio Renyi entropiją galima apskaičiuoti išvengiant lygčių integravimo.

Dydis $V(X)$ vadinamas *informacijos potencialu* (angl. k. *Information Potential*), nes jis įvertina skirtumą tarp egzempliorių x_i ir x_j panašiai taip, kaip tai daroma fizikoje skaičiuojant energijos potencialą tarp kelių kūnų [15]:

$$V(X)=\sum_{i=1}^{N}\sum_{j=1}^{N}G(x_i-x_j,2\delta^2). \quad (7)$$

Įstačius (7) į (6) ir pasinaudojus Gauso branduolio savybėmis, gaunama klasterio C_k entropija:

$$H(C_k)=-\log V(C_k). \quad (8)$$

Tokiu būdu apskaičiuota entropija vadinama *entropija klasterio viduje* (angl. k. *within-cluster entropy*). Taigi nuo σ pasirinkimo priklauso Parzeno lango artinio tikslumas. Teigiama, kad

turint normalizuotus duomenis, standartinį nuokrypį reikia parinkti iš intervalo $[0{,}06;\ 0{,}12]$ [17]. Tuomet Renyi konceptualios klasterizacijos algoritmas turėtų sudaryti gerai atskirtus klasterius. Kai kuriais atvejais kovariaciją galima tiesiogiai apskaičiuoti iš pateikto duomenų rinkinio.

Taigi pereinant visus klasterio C_k objektus galima apskaičiuoti entropiją klasterio viduje $H(C_k)$. Apskaičiuota entropijos reikšmė priklauso nuo standartinio nuokrypio pasirinkimo, tad gautus Renyi entropijos klasterizacijos rezultatus reikia stebėti esant skirtingoms σ reikšmėms. Siekiant išvengti σ pasirinkimo problemos, tiriamų duomenų kovariaciją Σ galima tiesiogiai apskaičiuoti iš duomenų.

1.1.3. Optimalaus klasterių skaičiaus išrinkimas

Dažnai klasterizacijos metu yra nežinomas tikras klasterių skaičius. Taip pat neaišku, ar sudaryti klasteriai atitinka tikrus klasterius. Šią problemą galima spręsti iš pradžių sukuriant daug pradinių klasterių. Po to ieškomi blogiausiai sudaryti klasteriai, kurie yra panaikinami, o jų objektai perskirstomi [17]. Kiekvieno perskirstymo metu klasterių skaičius mažinamas vienetu ir yra gaunami gerai atskirti klasteriai. Šis perskirstymas remiasi tarpklasterinės entropijos vertinimu ir tęsiasi tol, kol lieka 2 klasteriai.

Tarpklasterinės entropijos bendras lygties pavidalas:

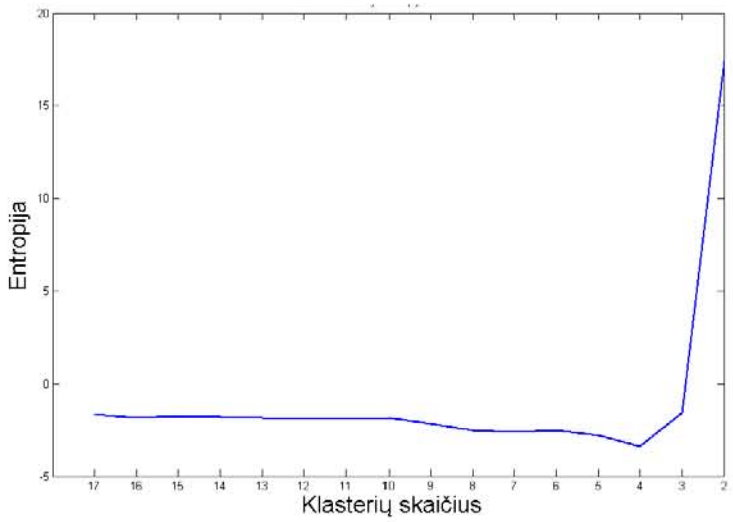
$$H(C_1,\ldots,C_K) = -\log V(C_1,\ldots,C_k), \quad (9)$$

kur

$$V(C_1,\ldots,C_K) = \frac{1}{2\prod_{k=1}^{K} N_k} \sum_{i=1}^{N}\sum_{j=1}^{N} M(x_{ij}) \cdot G(x_i - x_j, 2\sigma^2 I), \quad (10)$$

visiems klasteriams C_k, kai $k = 1,\ldots,K$ [17]. Čia $M(x_{ij})$ yra funkcija, įgyjanti reikšmę 1, jei x_i ir x_j nepriklauso tam pačiam klasteriui, arba 0, jei x_i ir x_j priklauso tam pačiam klasteriui. Esant gerai atskirtiems klasteriams, $V(C_1,\ldots,C_K)$ įgyja mažą, o tuo tarpu $H(C_1,\ldots,C_K)$ didelę reikšmę. Priešingu atveju, $V(C_1,\ldots,C_K)$ įgyja didelę, o $H(C_1,\ldots,C_K)$ mažą reikšmę. Pagal tai galima atpažinti blogai atskirtą klasterį ir jį perskirstyti. Čia blogai atskirtu klasteriu laikomas tas klasteris, kurį pašalinus tarpklasterinė entropija padidėja labiausiai, o tai reiškia, kad klasteriai geriau atskiriami vienas nuo kito [17]. Pavyzdžiui, jei turime tris klasterius C_1, C_2, C_3, tai skaičiuojant tarpklasterinę entropiją tarp klasterių C_1 ir C_2, į klasterį C_3 nėra atsižvelgiama. Šiuo atveju laikoma, kad klasterio C_3 narių funkcijos $M(x_{ij})$ yra lygios 0. Siekiant tiksliau nustatyti blogai atskirtą klasterį, galima fiktyviai perskirstyti tiriamo klasterio objektus kitiems klasteriams ir stebėti entropijos pokytį po šio perskirstymo. Tokiu būdu gali prireikti daugiau

skaičiavimo resursų, nes atliekamų operacijų skaičius padidėja, tačiau blogai atskirtas klasteris bus tiksliau identifikuotas.



2 pav. ***Renyi entropijos pokytis***

Palyginus visų klasterių porų tarpklasterinę entropiją, yra išrenkamas blogiausiai atskirtas klasteris, kurio elementai sekančiuose žingsniuose yra perskirstomi kitiems klasteriams. Atliekant blogai atskirtų klasterių šalinimą, kiekviename žingsnyje jų skaičius sumažėja vienetu. Prieš klasterių skaičiaus mažinimą ir jo narių perskirstymą reikia įsiminti visus klasterių narius. Tokiu būdu kiekviename žingsnyje yra išsaugoma klasterių aibė. Blogiausių klasterių šalinimo procesas baigiamas tuomet, kai lieka du klasteriai. Tuomet yra sudaryta priskyrimų klasteriams hierarchija. Tad užduodamas klausimas: kuriame hierarchijos lygmenyje yra galutiniai, atitinkantys tikruosius, klasteriai? Šią problemą padeda išspręsti tarpklasterinės entropijos reikšmės. Pastebėta, kad tarpklasterinės entropijos reikšmė drastiškai padidėja, kai klasterių skaičius pasidaro mažesnis, nei iš tikro yra klasterių [17]. Taigi stebinti tarpklasterinės entropijos pokytį prieš ir po jų narių perskirstymą galima parinkti galutinį klasterių skaičių. Iš 2 pav. matoma, kad sumažinus klasterių skaičių, entropijos pokytis staiga padidėja. Šiuo atveju tikėtina, kad tikrasis pateikto duomenų rinkinio klasterių skaičius yra lygus 3.

1.1.4. Algoritmo sudėtingumas

Kadangi pradiniai klasteriai yra inicijuojami atsitiktiniu būdu, tai klasterizacijos rezultatai gali skirtis esant tam pačiam duomenų rinkiniui. Būtina atsižvelgti į galutinių klasterių tarpklasterinę entropiją, nes ji padeda parinkti geriausią klasterių rinkinį. Renyi entropijos klasterizacijos algoritmo sudaryti klasteriai yra atskirti didžiausios entropijos tarp klasterių požiūriu, kai tuo tarpu entropija tarp klasteryje esančių objektų yra mažiausia. Atsižvelgiant į

visa tai nustatytas šio algoritmo sudėtingumo įvertinamas: $O(N^2)$, kur N – tai duotų objektų skaičius [17]. Siekiant užtikrinti didesnį tyrimo tikslumą, Renyi konceptualios klasterizacijos algoritmą tam pačiam duomenų rinkiniui reiktų įvykdyti bent kelis kartus ir atlikti gautų rezultatų analizę.

1.2. CLIP3 konceptualios klasterizacijos algoritmas

CLIP3 (angl. k. *Cover Learning using Integer Linear Programming*) yra hibridinis konceptualios klasterizacijos algoritmas [13]. Jis taikomas duomenų rinkiniams, kurie savyje turi tam tikrų netikslumų, neleidžiančių greitai suskirstyti šių duomenų į atitinkamus klasterius. Šį požymį turintys duomenų rinkiniai vadinami triukšmingais. Algoritmai, galintis sugeneruoti patikimas taisyklės esant triukšmams duomenyse, vadinami triukšmams atspariais algoritmais. Visas duomenų rinkinys skaidomas į teigiamą ir neigiamą duomenų rinkinį. Turint šiuos duomenų rinkinius, CLIP3 algoritmas dalina teigiamo rinkinio duomenis į poaibius, iš kurių yra pašalinamas triukšmas. Atlikus triukšmo šalinimo iteracinį procesą, gaunami duomenų poaibiai, kurie savyje neturi triukšmo. Toliau kiekvienam iš šių poaibių generuojamos klasterizacijos taisyklės. Šios taisyklės turi būti tokios, kad visi teigiamo duomenų rinkinio egzemplioriai patektų į vieną (teigiamų objektų) klasterį, o neigiamo duomenų rinkinio egzemplioriai – į kitą (neigiamų objektų) klasterį.

CLIP3 algoritmo pagrindas yra dvejetainių matricų, nusakančių triukšmo pasirodymą stebėjimuose, kūrimas ir jų skaidymas į poaibius. Tam naudojami sveikaskaitinio tiesinio programavimo uždavinio (angl. k. *Integer Liner Programming*, sutr. *IP*) sprendiniai [13]. Algoritmo vykdymo metu po kiekvieno teigiamų egzempliorių skaidymo į poaibius etapo pertekliniai poaibiai yra pašalinami. Pertekliniu poaibiu laikomas tas, kuris yra kito poaibio dalis arba identiškas kitam poaibiui. Perteklinių poaibių šalinimo operacija pagreitina taisyklių generavimo procesą. Taigi *IP* uždavinys sumažina galutinių poaibių, iš kurių yra generuojamos taisyklės, skaičių. Toliau išsamiau pateikiamos pagrindinės CLIP3 algoritmo fazės.

1.2.1. Klasterių iniciavimas

CLIP3 konceptualios klasterizacijos algoritmas vykdomas nuo teigiamų duomenų skaidymo į poaibius (mazgus) [13]. Algoritmo vykdymui turi būti paruošti teigiamų ir neigiamų egzempliorių duomenų rinkiniai. Šių rinkinių pavyzdžiai (*POS* ir *NEG* atitinkamai) pateikti 3 pav. *POS* duomenų rinkinį sudaro 5 egzemplioriai, o *NEG* – 3. *POS* ir *NEG* objektai turi būti

aprašomi vienodu požymių (atributų) skaičiumi. Šiuo atveju nagrinėjamų požymių skaičius yra 4, t.y. matricos *POS* ir *NEG* turi po 4 stulpelius.

Pastaba *p2*: Teigiamų stebėjimų duomenų rinkinys negali turėti egzempliorių iš neigiamų stebėjimų duomenų rinkinio. Rekomenduojama, kad pateiktuose duomenų rinkiniuose nebūtų pasikartojančių objektų [12].

$$POS = \begin{bmatrix} 1 & 2 & 3 & 2 \\ 1 & 3 & 1 & 2 \\ 2 & 3 & 2 & 5 \\ 3 & 3 & 3 & 2 \\ 1 & 1 & 1 & 3 \end{bmatrix}, \quad NEG = \begin{bmatrix} 3 & 1 & 2 & 5 \\ 1 & 2 & 2 & 4 \\ 2 & 1 & 2 & 3 \end{bmatrix}.$$

3 pav. ***POS ir NEG matricų pavyzdžiai***

Teigiamų egzempliorių duomenų rinkinio skaidymas atliekamas paieškos medžio principu. Kitaip nei kiti klasterizacijos algoritmai, CLIP3 neskaičiuoja skaidymo gerumo mato, pvz. entropijos. Tai leidžia naudoti duomenų rinkinį su įvairaus tipo požymiais. Šio proceso metu saugomi tik medžio lapai. Dėl to sudaromos paprastesnės taisyklės, mažesnis jų skaičius, ir algoritmo vykdymui reikia skirti mažiau atminties resursų. Algoritmo vykdymo metu generuojamos dvejetainės matricos. Dvejetainės matricos formuojamos pagal tokias taisykles [13]:

- Jeigu tam tikro teigiamo egzempliorio požymio reikšmė skiriasi nuo neigiamo egzempliorio atitinkamos požymio reikšmės, tai sakoma, kad tikėtina, jog šis požymis bus naudojamas taisyklėms generuoti, o į dvejetainę matricą reikia įrašyti 1.
- Jeigu tam tikro teigiamo egzempliorio požymio reikšmė sutampa su neigiamo egzempliorio atitinkamo požymio reikšme, tai sakoma, kad remiantis šiuo požymiu negalima generuoti taisyklės, o į dvejetainę matricą reikia įrašyti 0.

Tokios matricos generuojamos visiems mazgams pagal parinktą neigiamo stebėjimo egzempliorių. 4 pav. pateikta nagrinėjamo pavyzdžio *POS* duomenų rinkinio binarinė matrica *BIN* neigiamam stebėjimui *NEG1*. Kiekvienas sekantis mazgų lygmuo sudaromas nagrinėjant esamą lygmenį ir kitą neigiamų stebėjimų duomenų rinkinio objektą.

$$BIN = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}, \quad NEG1 = \begin{bmatrix} 3 & 1 & 2 & 5 \end{bmatrix}.$$

4 pav. ***POS duomenų rinkinio dvejetainė matrica BIN neigiamam stebėjimui NEG1***

Turint suformuotą dvejetainę matricą, yra konstruojamas ir sprendžiamas sveikaskaitis tiesinis programavimo uždavinys, kurio sprendiniai naudojami tolimesnių poaibių sudarymui. Duomenų skaidymas *IP* modelio pagalba yra daug greitesnis nei entropijos ar kitų dydžių skaičiavimas ir jų naudojimas [13]. *IP* uždavinio sprendiniai parodo svarbiausius egzempliorių požymius, kurie gali būti naudojami sudarant taisykles. Galimi keli *IP* uždavinio sprendimo būdai.

$$IP = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix},$$

$$branch1 = \begin{bmatrix} 1 & 2 & 3 & 2 \\ 1 & 3 & 1 & 2 \\ 2 & 3 & 2 & 5 \\ 1 & 1 & 1 & 3 \end{bmatrix}, \quad branch2 = \begin{bmatrix} 1 & 2 & 3 & 2 \\ 1 & 3 & 1 & 2 \\ 2 & 3 & 2 & 5 \\ 3 & 3 & 3 & 2 \end{bmatrix}.$$

5 pav. ***Nagrinėjamo pavyzdžio matricos BIN IP uždavinio galimas sprendinys ir iš jo suformuoti mazgai***

5 pav. pateiktas nagrinėjamo pavyzdžio dvejetainės matricos *BIN* (4 pav.) *IP* uždavinio galimas sprendinys. Kadangi pastarasis *IP* sprendinys turi du vienetus, tai sekančiame lygmenyje sugeneruojami du nauji mazgai (*branch1* ir *branch2*). Tokiu būdu yra taikomas paieškos medžio principas.

IP uždavinio sprendimo metu yra ieškomas binarinės matricos, kurios kiekviena eilutė yra atskiras objektas, o stulpelis – atitinkamas objekto požymis, minimalus sprendinys. Šis sprendinys – tai mažiausias skaičius požymių, kuris nusako daugelį egzempliorių. Sprendimo metu yra susumuojamas kiekvienas binarinės matricos stulpelis. Požymis, perdengiantis daugiausiai nagrinėjamų teigiamų egzempliorių (atitinkamas binarinės matricos stulpelis turi didžiausią vienetų skaičių), yra įtraukiamas į *IP* uždavinio sprendinį [13]. Objektai, kuriuos perdengia šis požymis, yra pašalinami iš nagrinėjamos teigiamų egzempliorių matricos. Šis procesas tęsiasi tol, kol nebelieka nei vieno neperdengto teigiamo objekto, arba tol, kol nėra sustabdoma *IP* uždavinio sprendinio paieška. Tai pačiai binarinei matricai galimi keli *IP* uždavinio sprendiniai. Kiekvienas *IP* uždavinio sprendinio įtrauktas požymis generuoja sekančio lygio mazgą. Šis mazgas sudaromas atsižvelgiant į nagrinėjamą egzempliorį – į naują mazgą įtraukiami tie nagrinėjami teigiami egzemplioriai, kurių atitinkamų požymių reikšmės nesutampa su tiriamo neigiamo egzempliorio atitinkamu požymiu iš *IP* uždavinio sprendinio. Tokiu būdu iš tam tikro mazgo galima sudaryti kelis sekančio lygio mazgus pagal *IP* uždavinio sprendimą konkrečiam neigiamam objektui. Siekiant padidinti skaičiavimų efektyvumą, būtina pašalinti perteklinius mazgus. Pertekliniu mazgu laikome tokį, kuris yra to paties lygmens mazgo poaibis arba dublikatas [12]. Turint suformuotą sekantį tiriamų mazgų lygį, yra parenkamas kitas

neigiamas objektas ir tokiu pačiu būdu sudaroma dvejetainė matrica. Vėliau sprendžiamas *IP* uždavinys pagal šį parinktą neigiamą objektą. Šis procesas tęsiamas tol, kol ištiriami visi neigiami egzemplioriai, arba tol, kol nepasiekiama įvesta algoritmo vykdymo sustojimo riba. Nuo neigiamų egzempliorių išdėstymo tvarkos priklauso sudaromi teigiamų egzempliorių mazgai bei galutinis algoritmo sprendinys. Ši problema būdinga daugeliui klasterizacijos algoritmų.

1.2.2. Panašumo (skirtumo) tarp objektų nustatymas

Šiame etape yra sumažinamas pirmoje CLIP3 algoritmo fazėje gautas galutinis mazgų skaičius. Kai kuriuos mazgus būtina pašalinti, nes jie apima tuos pačius duomenų rinkinio teigiamus egzempliorius. Siekiama turėti perdengtus visus pradinio duomenų rinkinio teigiamus egzempliorius su kuo mažesniu paskutinio lygio mazgų skaičiumi. Naudojant šiuos sugeneruotus mazgus, yra formuojama dvejetainė šabloninė matrica (angl. k. *Template Matrix*, sutr. *TM*). Šabloninės matricos kiekvienas stulpelis siejamas su konkrečiu mazgu, o eilutė atitinka duotos teigiamos matricos objektą. Laikomasi šių taisyklių [13]:

- Jeigu mazgas įtraukia konkretų teigiamos matricos egzempliorių, tai atitinkamoje šabloninės matricos stulpelio ir eilutės sankirtoje įrašomas 1.
- Jeigu mazgas neįtraukia konkretų teigiamos matricos egzempliorių, tai atitinkamoje šabloninės matricos stulpelio ir eilutės sankirtoje įrašomas 0.

Tokiu būdu yra suformuojama dvejetainė šabloninė matrica. Šios matricos *IP* modelio sprendinys randa minimalią mazgų aibę, kuri dengia visus duotus teigiamus objektus. Neįtraukti į šią aibę mazgai yra eliminuojami. Iš atrinktų mazgų yra formuojamos klasterizacijos taisyklės.

$$branch3 = \begin{bmatrix} 1 & 3 & 1 & 2 \\ 1 & 2 & 3 & 2 \\ 1 & 1 & 1 & 3 \end{bmatrix},\ branch4 = \begin{bmatrix} 3 & 3 & 3 & 2 \\ 2 & 3 & 2 & 5 \\ 1 & 3 & 1 & 2 \\ 1 & 2 & 3 & 2 \end{bmatrix},$$

$$TM = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$IP = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

6 pav. ***Nagrinėjamo pavyzdžio paskutinio lygmens mazgų branch3, branch4 TM matrica ir jos IP uždavinio sprendinys***

6 pav. pateikta nagrinėjamo pavyzdžio paskutinio lygmens mazgų *branch3* ir *branch4* šabloninė matrica *TM*. Kadangi *POS* duomenų rinkinys turi 5 objektus, tai matrica *TM* turi 5 eilutes. *TM* matricos stulpelių skaičius sutampa su esamo lygmens mazgų skaičiumi – šiuo atveju jis yra lygus 2. Šabloninės matricos *IP* sprendinys parodo, kad norint aprašyti visus teigiamus objektus yra būtini abu mazgai *branch3* ir *branch4*.

Turint suformuotus paskutinio lygmens mazgus, reikia nustatyti požymius ir šių požymių reikšmes, kurios lemia egzempliorių priskyrimą klasteriams. Tai atliekama lyginant paskutinio lygio mazgų egzempliorių požymius su neigiamų egzempliorių atitinkamais požymiais. Jei požymiai sutampa, tai neigiamų egzempliorių matricoje šis požymis pakeičiamas į 0. Taip kiekvienam mazgui sudaroma patikrintoji matrica (angl. k. *Checked Matrix*, sutr., *CM*), kuri turi tiek pat eilučių bei stulpelių kiek ir neigiamų egzempliorių matrica [13]. Iš patikrintosios matricos sudaroma dvejetainė matrica (čia nelygios 0 reikšmės pakeičiamos į 1) ir randamas jos *IP* uždavinio sprendinys.

$$CM3 = \begin{bmatrix} 3 & 0 & 2 & 5 \\ 0 & 0 & 2 & 4 \\ 2 & 0 & 2 & 0 \end{bmatrix}, \qquad CM4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 3 \end{bmatrix},$$

$$CM3BIN = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad CM4BIN = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$IP3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, \quad IP4 = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}.$$

7 pav. ***Nagrinėjamo pavyzdžio mazgų branch3 ir branch4 CM matrica, jos binarinis variantas ir IP uždavinio sprendinys***

Patikrintosios matricos *IP* sprendinys parodo, kokie požymiai yra naudojami sudarant taisykles. Iš 7 pav. matoma, kad taisyklėms sudaryti bus naudojami požymiai *P3* iš mazgo *branch3*, *P2* ir *P4* iš mazgo *branch4*. Kiekvienam likusiam *IP* uždavinio sprendinio požymiui taisyklė sudaroma atsižvelgiant į reikšmes, kurias įgyja pateiktosios teigiamų objektų matricos atitinkamo požymio reikšmės, prieš tai pašalinus reikšmes, kurias įgyja patikrintosios matricos atitinkamas požymis [13].

Taisyklė 1': **Jeigu** P3 = 1 **ARBA** P3 = 3, **tai** POS duomenų rinkinys
Taisyklė 2': **Jeigu** (P2 = 2 **ARBA** P2 = 3) **IR** (P4 = 2 **ARBA** P4 = 5), **tai** POS duomenų rinkinys

8 pav. ***Iš branch3 ir branch4 mazgų sudarytos taisyklės***

Taigi patikrintosios matricos dėka sudaromos taisyklės, nusakančios egzempliorius, kurie patenka į atitinkamus galutinio lygmens mazgus ir nepatenka į kitus galutinio lygmens mazgus.

Iš 8 pav. matoma, kad sudarant taisykles naudojami loginiai operatoriai IR, ARBA. Tarp to paties požymio reikšmių įterpiamas loginis operatorius ARBA, o tarp skirtingų – IR. Kai tenkinamas bendras taisyklės teisingumas, tai nagrinėjamas objektas priskiriamas atitinkamam klasteriui.

1.2.3. Optimalaus klasterių skaičiaus išrinkimas

Kadangi klasterizacijos taisyklės yra sudarytos iš galutinio lygmens mazgų, tai kiekviena taisyklė dengia tik dalį pateiktų teigiamų egzempliorių. Taip yra, nes kiekvienas galutinio lygmens mazgas yra poaibis pateikto teigiamų egzempliorių rinkinio. Tuomet turint sudarytą taisyklių sąrašą, yra atrenkama taisyklė, kuri aprašo daugiausiai duotų teigiamų egzempliorių ir mažiausiai neigiamų egzempliorių [12]. Suradus šią geriausią taisyklę, iš duoto teigiamo duomenų rinkinio yra pašalinami visi teigiami objektai, kuriuos aprašo ši taisyklė. Jei lieka objektų, kurių neaprašo pastaroji taisyklė, tai pereinama į pirmą algoritmo fazę ir algoritmas vykdomas dar kartą. Gaunamas rezultatas – tai taisyklių rinkinys, aprašantis tik teigiamus duotus duomenų rinkinio egzempliorius.

CLIP3 konceptualios klasterizacijos algoritmas vykdomas tol, kol nebelieka taisyklėmis nusakytų egzempliorių, arba pasiekiamas vienas iš įvestų algoritmo sustabdymo kriterijų.

Taisyklė 1: **Jeigu** P3 = 1 **arba** P3 = 3, **tai** POS duomenų rinkinys
Taisyklė 2: **Jeigu** P2 = 1, **tai** POS duomenų rinkinys

9 pav. ***Nagrinėjamo pavyzdžio sugeneruotos galutinio lygmens taisyklės***

Iš pateiktų teigiamų ir neigiamų egzempliorių duomenų rinkinių (3 pav.) pavyko sugeneruoti dvi taisykles (9 pav.). Pirmoji taisyklė aprašo 80%, antroji – 60% teigiamų egzempliorių. Abi taisyklės kartu aprašo 100% *POS* ir 0% *NEG* matricos egzempliorių.

Užbaigus CLIP3 algoritmo vykdymą, galima atlikti gautų taisyklių analizę. Gautos taisyklės turi nusakyti tik teigiamus objektus. Tačiau priklausomai nuo pateiktų teigiamų ir neigiamų egzempliorių, neigiamų egzempliorių užrašymo būdo, bei *IP* uždavinio sprendinio pasirinkimo, kai kurios taisyklės gali nusakyti ir neigiamus egzempliorius.

1.2.4. Algoritmo sudėtingumas

CLIP3 algoritmo trijų žingsnių apmokymo procesas skirtingais būdais dalija mokymosi egzempliorius, parenka geriausias taisyklės tam, kad būtų aprašyti teigiami mokymosi egzemplioriai. Sekančiame žingsnyje šitie teigiami mokymosi egzemplioriai yra pašalinami. Tai

leidžia atlikti kitą apmokymą naudojant mažesnį duomenų kiekį. CLIP3 klasterizacijos algoritmo sudėtingumo įvertinimas yra $O(N^2)$, kur N – duotų objektų skaičius [13]. Taigi CLIP3 apmokymo algoritmas yra geras duomenų apdorojimo įrankis, nes jis yra sąlyginai paprastas ir gana greitai generuoja taisyklės.

1.3. CLIP4 konceptualios klasterizacijos algoritmas

CLIP4 kaip ir CLIP3 yra hibridinis konceptualios klasterizacijos algoritmas [11]. Šio algoritmo privalumas tas, kad jis dirba su dideliu objektų ir atributų skaičiumi, bei duomenimis, kuriuose yra netikslumų. Atributai gali būti skaitiniai, nominalūs arba skaitiniai ir nominalūs tuo pačiu metu. Svarbu tai, kad CLIP4 dirba su egzemplioriais, kuriuose gali būti nežinomų atributų reikšmių. Nežinomų atributų reikšmių atsiranda dėl rezultatų užrašymo problemų ar kitų priežasčių. Esant mažam duomenų rinkiniui, šis algoritmas siūlo naudoti genetinius operatorius (angl. k. *Genetic operators*), kurie padeda sugeneruoti geresnes taisykles [11]. Kadangi daugelis CLIP4 skaičiavimo etapų sutampa su CLIP3 konceptualios klasterizacijos algoritmo etapais, tai toliau aptariami tik esminiai CLIP4 algoritmo niuansai.

1.3.1. Darbas su nežinomomis atributų reikšmėmis

Kai kurių egzempliorių atributų reikšmės nėra žinomos. Daugelis klasterizacijos algoritmų egzempliorius, turinčius nežinomų atributų reikšmių, pašalina iš duomenų rinkinio ir atlieka skaičiavimus su likusiais egzemplioriais. Kitaip nei tokie klasterizacijos algoritmai CLIP4 nepašalina objektų su nežinomomis atributų reikšmėmis [18].

Egzemplioriai su nežinomomis atributų reikšmėmis daro įtaką tik pirmoje ir antroje CLIP4 algoritmo fazėse. Nežinomos reikšmės yra ignoruojamos įrašant 0 į atitinkamas ląsteles sudaromose binarinėse matricoje. Pastarasis veiksmas atliekamas sudarant sekančio lygmens hierarchinio medžio mazgus (pirma algoritmo fazė) ir užpildant šabloninę matricą (antra algoritmo fazė) [18]. Taip CLIP4 algoritmas užtikrina, kad atributai su nežinomomis reikšmėmis nėra naudojami sudarant klasterizacijos taisykles, o egzempliorio, turinčio savyje tokių atributų, visi kiti atributai su žinomomis reikšmėmis yra taikomi sudarant objektų priskyrimo klasteriams taisykles.

CLIP4 konceptualios klasterizacijos algoritmo elgesys su nežinomomis atributų reikšmėmis leidžia sudaryti taisykles, kurioms įtakos nedaro šių reikšmių trūkumas. Kai kurie klasterizacijos algoritmai šią problemą sprendžia taikant statistinius metodus, ieškančius ryšių

tarp objektų, kurių atitinkamų atributų reikšmės yra žinomos, ir egzempliorių, kurių šios reikšmės nėra žinomos [19].

1.3.2. *IP* uždavinio sprendimas

Vienas iš pagrindinių skirtumų tarp CLIP3 ir CLIP4 konceptualios klasterizacijos algoritmų yra tas, kad taikomas kitas *IP* uždavinio sprendimo būdas. CLIP4 algoritmo *IP* uždavinio sprendimą sudaro šie žingsniai [18]:

1. Surandamos aktyvios eilutės, kurios turi mažiausią 1-tų skaičių (*min-eilutės*).
2. Iš *min-eilučių* išrenkami stulpeliai, kurie turi didžiausią 1-tų skaičių (*max-stulpeliai*).
3. Iš *max-stulepelių* išrenkami tie, kurie turi didžiausią 1-tų skaičių visose aktyviose eilutėse (*max-max-stulepiai*). Jeigu tokių stulpelių yra daugiau nei vienas, tai pereinama į 4-tą žingsnį; priešingu atveju – į 5-tą žingsnį.
4. Tarp *max-max-stulepių* randamas pirmasis stulpelis, kuris turi mažiausią 1-tų skaičių neaktyviose eilutėse.
5. Surastas stulpelis įtraukiamas į sprendinį.
6. Aktyvios eilutės, kurias dengia surastas stulpelis, pažymimos neaktyviomis eilutėmis. Jeigu visos eilutės yra neaktyvios, tai nutraukiamas ciklo vykdymas. Priešingu atveju pereinama į 1-mą žingsnį.

Pastaba *p3*: Čia aktyvios eilutės yra tos eilutės, kurios nėra padengtos dalinio sprendinio, o neaktyvios – tos, kurios yra padengtos dalinio sprendinio. Aktyvių ir neaktyvių eilučių skaičius turi sutapti su binarinės matricos eilučių skaičiumi.

1.3.3. Genetinis algoritmo modulis

Genetiniai algoritmai (sutr. *GA*) įvairiai taikomi mašininiame mokyme. Šių algoritmų analogų yra gyvoje aplinkoje. Genetinių algoritmų vykdymo metu nustatoma, kurie individų aibės objektai gali „susilaukti palikuonių“. Remiantis šiuo principu stipriausi individai „išgyvena ir prasiplečia“, o tuo tarpu silpniausi „išnyksta“.

CLIP4 konceptualios klasterizacijos algoritmo *GA* modulis naudoja šiuos sutartinius terminus [11]:

- Individais (angl. k. *Individual*) vadinami klasterizacijos medžio mazgai, kurie sudaromi algoritmo vykdymo metu. Kiekvienas iš jų aprašomas $POS_{i,j}$ matrica (*j*-tasis mazgas *i*-

tajame hierarchinio medžio lygmenyje) ir $SOL_{i,j}$ (*IP* problemos sprendinys $POS_{i,j}$ matricai).

- Populiacija (angl. k. *Population*) suprantama kaip aibė to paties hierarchinio medžio lygmens mazgų.

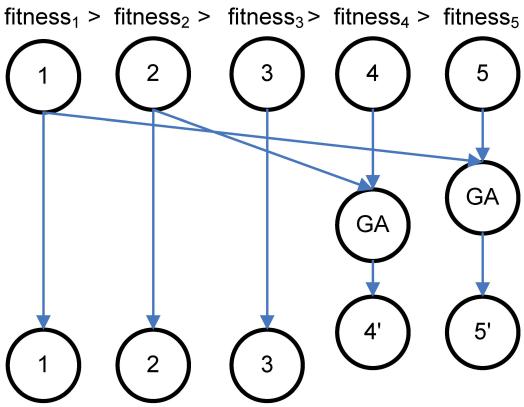
CLIP4 konceptualios klasterizacijos algoritmas gali taikyti *GA* modulį, jei tiriamo lygmens populiacija turi ne mažiau nei du skirtingus individus. Yra įvedama tinkamumo funkcija (angl. k. *Fitness function*), kuri naudojama parenkant tinkamiausius mazgus kitam hierarchinio medžio lygmeniui [11]. *i*-tojo hierarchinio medžio lygmens *j*-tojo mazgo tinkamumo funkcija ($fitness_{i,j}$) skaičiuojama tokiu būdu:

$$\text{fitness}_{i,j} = \frac{\text{POS}_{i,j}\ \textit{pavyzdžių skaičius}}{\textit{skaičius mazgų, kurie bus sugeneruoti}\ (i+1)\ \textit{lygmenyje iš}\ \text{POS}_{i,j}} \qquad (11)$$

Iš (11) matoma, kad tinkamumo funkcija yra apskaičiuojama padalinant matricos $POS_{i,j}$ eilučių skaičių iš $SOL_{i,j}$ vektoriaus vienetų skaičiaus. Vektoriaus $SOL_{i,j}$ vienetų skaičius parodo, kiek mazgų bus sugeneruota iš mazgo $POS_{i,j}$ sekančiame hierarchinio medžio lygmenyje. Tinkamumo funkcija įgyja didelę reikšmę tais atvejais, kai tiriami mazgai apima didelį egzempliorių skaičių, o sekančiame lygmenyje sugeneruojamas mažas mazgų skaičius. Naudojant tinkamumo funkciją galima pagerinti klasterizacijos algoritmo vykdymo spartą ir gauti geresnes taisykles [18].

Individai sekančio lygmens populiacijos formavimui parenkami pagal šią žingsnių seką [11]:

1. Nustatoma kiekvieno individo tinkamumo funkcija.
2. Pusė visų individų, kurie turi didžiausius tinkamumo funkcijos įvertinimus, parenkami sekančio hierarchinio medžio lygmens populiacijos formavimui, t.y. generuojami tolimesnio lygmens mazgai.
3. Kita likusi populiacijos pusė yra generuojama kombinuojant geriausius individus su blogiausiais (pats geriausias individas su pačiu blogiausiu, mažiau geresnis su mažiau blogesniu ir taip toliau) ir taikant genetinius operatorius.



10 pav. ***CLIP4 algoritmo GA modulio taikymo pavyzdys***

Taikant genetinio algoritmo modulio operatorius tam tikriems mazgams yra generuojami naują informaciją turintys sekančio lygmens mazgai (10 pav.).

Genetinio algoritmo modulis naudoja perdengimo (angl. k. *Crossover*) ir mutacijos (angl. k. *Mutation*) operatorius [2]. Šie operatoriai taikomi $SOL_{i,j}$ vektoriams. Tokiu būdu suformuotas vektorius $ChildSOL_{i,j}$ bei matrica $POS_{i,j}$, turinti didžiausią tinkamumo funkcijos reikšmę, generuoja naują individą.

GA modulio perdengimo operatorius apibrėžiamas tokiu būdu [11]:

$$ChildSOL_i = \max\left(Parent1SOL_i, Parent2SOL_i\right). \quad (12)$$

Čia $Parent1SOL_i$ ir $Parent2SOL_i$ yra *i*-tosios dviejų lyginamų tarpusavyje mazgų vektorių $SOL_{i,j}$ reikšmės.

Mutacijos operatoriai įterpia naują informaciją į sudarytą vektorių $ChildSOL_i$. Kiekvienas vektoriaus $ChildSOL_i$ 1-tas generuoja naują mazgą. Tačiau čia neatsižvelgiama į tuos 1-tus, kurie sudaromi iš to $SOL_{i,j}$ vektoriaus, turinčio didesnę *tinkamumo funkcijos* reikšmę. Šie 1-tai yra ignoruojami, kad jie negeneruotų pasikartojančių mazgų (juos ir taip generuoja mazgai, kurie antrame populiacijos formavimo žingsnyje įtraukiami kaip turintys didžiausias *tinkamumo funkcijos* reikšmes).

GA modulis dažnai taikomas tais atvejais, kai CLIP4 konceptualios klasterizacijos algoritmas generuoja taisykles duomenų rinkiniams su mažu egzempliorių skaičiumi [18]. Tokiu būdu sudarytos taisyklės gali aprašyti egzempliorius, kurie pieš tai nebuvo įtraukti į teigiamų stebėjimų duomenų rinkinį.

1.3.4. Vykdymo ribos

CLIP3 ir CLIP4 konceptualios klasterizacijos algoritmas gali turėti vykdymo ribas (angl. k. *Threshold*) [11], kurių paskirtis – kontroliuoti kuriamų taisyklių sudėtingumą ir nuspręsti, kada sustoti generuoti taisykles. Jų dėka pašalinami tie mazgai, kurie buvo sudaryti dėl neaiškumų pateiktuose pradiniuose duomenų rinkiniuose. Yra naudojamos šios ribos:

- *Triukšmo riba* (angl. k. *Noise Threshold*), nusakanti, kurios šakos, galinčios turėti triukšmingų teigiamų egzempliorių, remiantis *IP* sprendiniu turi būti pašalintos pirmoje algoritmo fazėje.
- *Geriausios taisyklės riba* (angl. k. *Best Rule Threshold*), sakanti, kad taisyklės, sugeneruotos antroje fazėje, taps geriausiomis taisyklėmis tik tuomet, jei jos aprašo nemažiau teigiamų arba likusių teigiamų egzempliorių nei tai nusako geriausios taisyklės riba.
- *Sustojimo riba* (angl. k. *Stop Threshold*), sustabdanti algoritmo veikimą, kai lieka mažiau nei sustojimo ribos taisyklėmis neaprašytų teigiamų egzempliorių.

CLIP4 klasterizacijos algoritmas papildomai įtraukia *skaidymo ribą* (angl. k. *Pruning Threshold*), kuri yra ekvivalenti tinkamumo funkcijai. Ši riba nustato šalinamas šakas, kurių tinkamumo funkcijos reikšmė mažesnė nei skaidymo riba.

Taigi geriausios taisyklės ir sustojimo ribos gali sustabdyti algoritmo veikimą. CLIP4 konceptualios klasterizacijos algoritmas vykdomas tol, kol nebelieka taisyklėmis nusakytų objektų, arba pasiekiamas įvestas algoritmo sustabdymo kriterijus.

1.3.5. Taisyklių generavimas

CLIP4 konceptualios klasterizacijos algoritmo sugeneruotos taisyklės užrašomos kitokiu būdu nei tai daro CLIP3 algoritmas. Čia taisyklės generuojamos tiesiogiai iš neigiamų stebėjimų duomenų rinkinio (*POS*) ir kiekvieno lygmens pasirinko mazgo egzempliorio [11]. Kiekvieno lygmens mazgai yra duoto teigiamo duomenų rinkinio poaibiai.

Taisyklė 1: **Jeigu** P1 $\neq$ 2 **ir** P1 $\neq$ 4 **ir** P3 $\neq$ 1, **tai** POS duomenų rinkinys
Taisyklė 2: **Jeigu** P2 $\neq$ 1 **ir** P2 $\neq$ 5, **tai** POS duomenų rinkinys

11 pav. ***CLIP4 klasterizacijos algoritmo sugeneruotų taisyklių pavyzdys***

11 pav. pavaizduotos pavyzdinės klasterizacijos taisyklės. Visur naudojamas loginis operatorius IR. Jei netenkinama bent viena sąlyga, tai nagrinėjamas objektas nėra įtraukiamas į

POS duomenų rinkinį. Toks taisyklių užrašymas yra kompaktiškesnis ir lengviau suprantamas nei CLIP3 algoritmo pateiktas būdas.

1.3.6. Algoritmo sudėtingumas

CLIP4 konceptualios klasterizacijos algoritmas skirtingai nuo CLIP3 naudoja atnaujintą *IP* uždavinio sprendimo būdą, taiko genetinio algoritmo modulį, sprendžia problemas, atsirandančias dėl egzempliorių, kurie turi atributų su nežinomomis reikšmėmis. Įvedus šiuos ir kitus pataisymus, algoritmas sudėtingumas nepakito – $O(N^2)$, kur N – egzempliorių skaičius [22]. Kai kuriais atvejais CLIP4 sugeneruoja geresnį taisyklių rinkinį nei CLIP3 [11].

1.4. ITERATE konceptualios klasterizacijos algoritmas

ITERATE yra hierarchinis konceptualios klasterizacijos algoritmas, kurio autoriai G. Biswas ir B. Weinberg. ITERATE algoritmas naudoja skaitinius ir nominalinius duomenis. Šis algoritmas sudarytas iš sekančių žingsnių [9]:

1. Naudojant didžiausių skirtumų rūšiavimo (*ADO*) algoritmą, sugrupuojami nuoseklūs duomenys.
2. Naudojant kategorizavimo įrankį, sudaromas klasifikavimo medis.
3. Iš klasifikavimo medžio išgaunami „geri“ pradinių duomenų skirsniai, kurie naudojami kaip starto taškai ieškant pageidaujamų klasterių.
4. Siekiant sudaryti maksimaliai atskirtus klasterius, perskirstomi objektai tarp sudarytų grupių.

4-tas žingsnis kartojamas tol, kol nebelieka objektų, kurie keičia savo klasę.

1.4.1. Klasterių iniciavimas

Klasterių iniciavimas yra antrasis ITERATE algoritmo žingsnis. Šiame etape naudojami pirmame žingsnyje surūšiuoti nuoseklūs duomenys, kur taikomas ADO rūšiavimo algoritmas [9]. Pagal šį algoritmą sekantis eilės objektas yra tas, kuris labiausiai padidina Manhattan'o atstumą tarp jo ir prieš tai esančių n objektų. Manhattan'o atstumas tarp dviejų objektų yra apibrėžiamas kaip jų atributų skirtumų skaičius. Paprastai dydį n, kuris turi atitikti tikėtiną klasių skaičių, apibrėžia vartotojas. Pirmasis objektas parenkamas iš tiriamo lygmens mazgo duomenų ir yra toks, kad labiausiai skirtųsi nuo prototipo [9].

Antrajame žingsnyje hierarchinis klasterizacijos metodas leidžia valdyti pradinių skirsnių kūrimą. Iš to paties duomenų rinkinio suformuotų pradinių skirstinių gali būti sudaryti skirtingi klasteriai. Tad „gero“ skirsnio pasirinkimas yra aktuali problema. ITERATE klasifikacijos medžio kūrimui naudojamo algoritmo bendrasis pavidalas [10]:

Inicializacija: Priskirti $L = N_1$
O_1 = klasterizuojamų objektų rinkinys
Kartoti kol L nėra tuščias
    Paimti pirmą elementą iš L, sakykime N_k
    Jei O_k turi daugiau nei vieną objektą
        Kartoti kiekvienam O_k objektui
            Jei pirmasis O_k objektas
                Sukurti naują mazgą N_{k+c} kaip N_k „vaiką“
                Padėti objektą į mazgą N_{k+c}
            Arba
                (i) Padėti objektą į visus N_k mazgus-„vaikus“ vieną po kito ir apskaičiuoti skirstinių vertes kiekvienam
                (ii) Padėti objektą kaip naują N_k mazgą „vaiką“ ir apskaičiuoti skirstinio vertę.
            Priskirti objektą mazgui, kurio skirstinio vertė yra didžiausia ir atnaujinti $A_i = V_{ij}$ mazgui.
        Baigti *Kartoti*
        Sudėti naujus „vaikus“ į L.
    End *Jei*
Baigti *Kartoti*

Čia N_i vaizduoja mazgą klasifikacijos medyje, o O_i apibrėžia duomenų objektų aibę mazge N_i. N_1 yra medžio šaknis, O_1 – visa aibė duomenų, kurie turi būti klasterizuoti. Klasifikacijos medžio kūrimas taiko paprastą skaidymo metodą, kuris nuo medžio šaknies skirsto duomenis į poaibius (angl. k. *subclasses*). Algoritmas naudoja skirsnio rezultatą (15) norėdamas nustatyti, ar pasirinktą objektą priskirti vienai iš esamų klasių, ar kurti naują klasę ir jai priskirti šį objektą [9]. Prieš pereinant į žemesnį lygmenį, pilnai suklasifikuojami esamo lygmens objektai. Taip „auginamas“ klasifikacijos medis.

1.4.2. Panašumo (skirtumo) tarp objektų nustatymas

Panašumo (skirtumo) tarp objektų nustatymą ITERATE algoritmas įvardina kaip ketvirtą žingsnį, tačiau jis labai susijęs su kitais žingsniais. Siekiant padidinti atskirų klasių koherentiškumą skirstinyje, yra naudojamas iteratyvus perskirstymo operatorius. Perskirstymo operatorius objektą *d* priskiria klasei *k*, jei jos atitikimas kategorijai (CM_{dk}) yra maksimalus [9]. Objektas lieka savo klasėje, jei yra kelios lygiavertės klasės ir jis yra vienoje iš jų. Perskirstymo iteracija susideda iš kiekvieno objekto priskyrimo nustatymo. Atitikimas tarp objekto *d* ir klasės *k* yra apibrėžtas tikimybiniu panašumo matu tokiu būdu:

$$CM_{dk} = P(C_k) \sum_{i,j \in \{A_i\}_d} \left(P(A_i = V_{ij} \mid C_k)^2 - P(A_i = V_{ij})^2 \right), \quad (13)$$

kur $P(A_i = \mathrm{V_{ij}})$ – tikimybė, kad savybė A_i bus lygi V_{ij}; $P(A_i = \mathrm{V_{ij}} \mid C_k)$ – sąlyginė tikimybė, kad savybė A_i bus lygi V_{ij} ir ji priklauso klasei C_k [9]. Kategorijos atitikimo matas CM_{dk} daro prielaidą, kad objektas turi tik vieną reikšmę kiekvienam atributui ($j \in \{A_i\}_d$).

1.4.3. Optimalaus klasterių skaičiaus išrinkimas

Tai trečiasis ITERATE klasterizacijos algoritmo žingsnis, kuris glaudžiai susijęs su antruoju. Tyrinėdami kognityviai pageidaujamus kategorizacijos lygius, A. Gluck ir E. Corter panaudojo tikimybės pritaikymo strategiją. Jie apibrėžė klasės C_k kategorijos naudingumą (CU_k) taip [10]:

$$CU_k = P(C_k) \sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2, \quad (14)$$

kur $P(A_i = \mathrm{V_{ij}})$ – tikimybė, kad savybė A_i bus lygi V_{ij}; $P(A_i = \mathrm{V_{ij}} \mid C_k)$ – sąlyginė tikimybė, kad savybė A_i bus lygi V_{ij} ir ji priklauso klasei C_k. Skirstinio rezultatu (angl. k. *Partition Score*, sutr. *PS*) vadinama skirstinio struktūra, susidedanti iš K klasių. *PS* naudingumas, yra apibrėžiamas kaip *CU* vidurkis K klasėms [10]:

$$PS = \frac{\sum_{k=1}^{K} CU_k}{K}. \quad (15)$$

A. Gluck ir E. Corter pademonstravo kategorijos naudos efektyvumą, nuspėjant pageidautiną kategorizacijos lygį, esant jau sudarytai klasifikacijos hierarchijai. D. Fisher pritaikė *PS* sąvoką COBWEB konceptualios klasterizacijos algoritme [10].

Pradinė skirsnių struktūra yra gaunama lyginant klasių arba mazgų *CU* reikšmę einant klasifikacijos medžiu. Einant bet kuria šaka nuo medžio šakninės viršūnės iki lapų, *CU* reikšmė

didėja, o po to staiga sumažėja. Klasės, kurių reikšmės toje šakoje yra mažesnės nei maksimali CU reikšmė, nepakankamai tiksliai apibrėžia duomenis ir todėl nėra naudingos klasterių formavimui [10]. Bendrasis algoritmas, generuojantis pradinius skirsnius, pateiktas žemiau [9].

Inicijavimas: Priskirti *mazgas* N = hierarchinio medžio šaknis (angl. k. *root*)
Priskirti *sąrašas* = $\{N\}$
Kartoti, kol *sąrašas* nėra tuščias
paimti „*vaikas(N)*“
Jeigu
$CU_N > CU_c \quad \forall c \in$ „*vaikas(N)*“,
Tuomet N nusako pradinio skirsnio grupę
Priešingu atveju
Jei $\forall c \in$ „*vaikas(N)*“ *IR* $CU_N \leq CU_c$
Tai įdėti c į *sąrašas*
Jei $\forall c \in$ „*vaikas(N)*“ *IR* $CU_N > CU_c$
Tai c nusako grupę pradiniame skirsnyje
Baigti Jeigu
Pašalinti N iš *sąrašas*
Priskirti N = pirmas elementas iš *sąrašas*
Baigti *Kartoti*

Iš pastarojo algoritmo šablono matoma, kad jei paieškos metu buvo parinktas koks nors mazgas, tai nei vienas kitas mazgas, esantis sekančiame šio mazgo lygmenyje, negali būti įtrauktas į pradinį skirsnį. Tai atliekama atsižvelgiant į kiekvieno mazgo kategorijos naudingumo dydį. Tokiu būdu sugeneruojami skirsniai, padedantys sudaryti pastovius klasterius.

1.4.4. Algoritmo sudėtingumas

Sudarant maksimaliai skirtingus ir surištus klasterius, ITERATE konceptualios klasterizacijos algoritmas apjungia hierarchinį ir duomenų rinkinio skaidymo metodus. Kadangi duotam duomenų rinkiniui yra taikomas didžiausių skirtumų rūšiavimo algoritmas ir gautų mazgų perskirstymas, tai padidėja bendras algoritmo sudėtingumas. ITERATE klasterizacijos algoritmo sudėtingumas įvertintas $O(N^3)$ reikšme, kur N – duotų objektų skaičius [22].

1.5. Klasterizacijos algoritmų lyginamosios analizės išvados

Algoritmų lyginamosios analizės metu aptarti dažnai mokslinėse publikacijose minimi Renyi entropijos, CLIP3, CLIP4 ir ITERATE klasterizacijos algoritmai. Esminiai jų skirtumai pateikti 1 lentelėje. Čia stulpeliai nagrinėja konkretų klasterizacijos algoritmą, o eilutės – tam tikras jų savybes. Šie algoritmai lyginami įsivestų kriterijų požiūriu: klasterių iniciavimu, panašumų tarp objektų nustatymu, optimalaus klasterių skaičiaus išrinkimu ir algoritmo sudėtingumu.

1 lentelė. ***Klasterizacijos algoritmų esminiai skirtumai***

	Renyi entropijos	**ITERATE**	**CLIP3 ir CLIP4**
Klasterių iniciavimas	Atsitiktiniu būdu parenkami klasterių centrai. Taškas, esantis arčiausiai bet kurio klasterio nario, yra įtraukiamas į šį klasterį.	Remiasi *ADO* rūšiavimo algoritmu ir hierarchiniu klasterizacijos metodu.	Duomenys skaidomi sprendimų medžio algoritmo principu.
Panašumų (skirtumų) tarp objektų nustatymas	Remiasi entropijos skaičiavimu.	Remiasi atitikimo kategorijai matu.	Remiasi sveikaskaitiniu tiesiniu programavimo uždaviniu (*IP*).
Optimalaus klasterių skaičiaus išrinkimas	Remiasi tarpklasterinės entropijos pokyčio stebėjimu.	Remiasi klasių kategorijos naudingumo pokyčio stebėjimu.	Remiasi triukšmo, geriausios taisyklės ir sustojimo ribomis.
Algoritmo sudėtingumas (N – objektų skaičius)	$O(N^2)$	$O(N^3)$	$O(N^2)$

Iš 1 lentelės matoma, kad pagal įvestus kriterijus lengviausiai realizuojamas Renyi entropijos klasterizacijos algoritmas. Pagal tuos pačius aspektus ITERATE algoritmas yra sunkiau įgyvendinamas, tačiau jis generuoja kokybiškesnius ir pastovesnius klasterius [10]. Siekiant alternatyvaus varianto, siūloma naudoti CLIP3 arba CLIP4 klasterizacijos algoritmus, kurie realizuojami lengviau nei ITERATE, o kuriamų klasterių požiūriu yra geresni nei Renyi entropijos algoritmas. Be to, CLIP3 ir CLIP4 konceptualios klasterizacijos algoritmai generuoja objektų priskyrimo klasteriams taisykles [13, 11].

2. STATISTINIS DUOMENŲ APDOROJIMAS

Statistinis duomenų apdorojimas apima daugelį veiksmų. Teigiama, kad jų taikymas gali pagerinti klasterizacijos rezultatus [16]. Šio darbo metu nagrinėjamos duomenų rinkinių normalizacijos, diskretizacijos ir išskirčių nustatymo problemos.

2.1. Duomenų normalizacija

Kai kurių autorių teigimu duomenis prieš klasterių sudarymą reikia normalizuoti taip, kad šių duomenų požymių vidurkis būtų lygus 0, o standartinis nuokrypis 1 [16]. Daugiamačio duomenų rinkinio normalizaciją pagal visus požymius galima atlikti taikant klasikinės statistikos metodus.

Atsitiktinio dydžio X normalizuota reikšmė ZX gaunama pagal šią formulę:

$$ZX = \frac{X - EX}{\sqrt{DX}}, \quad (16)$$

kur EX – atsitiktinio dydžio X vidurkis, DX – atsitiktinio dydžio X dispersija [14].

Atsitiktinio dydžio X dispersija:

$$DX = EX^2 - (EX)^2, \quad (17)$$

kur EX^2 – atsitiktinio dydžio kvadrato vidurkis, $(EX)^2$ – atsitiktinio dydžio X vidurkio kvadratas.

Tokiu būdu normalizavus duomenis, sudaromas duomenų rinkinys, kurio kiekvieno atributo vidurkis yra lygus arba labai artimas 0, o standartinis nuokrypis – 1.

2.2. Duomenų diskretizacija

Diskretizacijos proceso metu nuoseklūs duomenys yra suskaidomi į baigtinį intervalų skaičių, kurių kiekvienas apima visas reikšmes pakliūvančias į jį. Diskrečių intervalų skaičių dažniausiai parenka vartotojas, o jų pločius nustato diskretizacijos algoritmas. Intervalų ir jų pločių skaičius turi didelį efektą klasterizacijos algoritmų tikslumui ir veikimo spartai [7]. Kuo didesnis intervalų skaičiuos, tuo didesnė tikimybė, kad klasterizacijos algoritmas per daug prisiderina prie duomenų. Kai yra per mažai intervalų, informacijos tikslumas mažėja ir tai gali paslėpti ryšį tarp klasės ir intervalo kintamųjų. Tokiu atveju tam tikra informacija gali būti prarandama. Todėl bet koks diskretizacijos algoritmas, naudojamas kaip pradinio apdorojimo fazė klasterizacijos algoritmui, turi generuoti mažą diskrečių intervalų kiekį. Yra žinomi keli intervalų skaičiaus parinkimo būdai. Vienas iš jų – intervalų skaičius turi būti didesnis nei

norima aprašyti klasių, kitas – kiekvienam atributui F_i ($i = 1, \ldots,$) intervalų skaičių n_{Fi} galima apskaičiuoti pasinaudojant formule [7]:

$$n_{Fi} = \frac{M}{3C}, \quad (18)$$

kur M yra bendras diskretizuojamų narių skaičius, C – klasių skaičius.

Duomenų diskretizavimas gali būti atliekamas kaip klasterizacijos algoritmo sudedamoji dalis arba prieš klasterizacijos algoritmą [7].

2.2.1. Vienodo pločio diskretizacija

Vienodo pločio diskretizacijos algoritmas pirmiausiai randa mažiausią ir didžiausią reikšmes kiekvienam pateiktam atributui F_i. Po to jis padalina ištisinį atributų intervalą į vartotojo nurodytą vienodo pločio diskrečių intervalų skaičių n_{Fi}.

2.2.2. Vienodo dažnio diskretizacija

Vienodo dažnio diskretizacijos algoritmas pirmiausia surūšiuoja visus kiekvienos savybės duomenis didėjimo tvarka ir padalina juos į vartotojo nurodytą intervalų skaičių n_{Fi}. Tokiu būdu gaunami intervalai, kuriuose yra vienodas skaičius surūšiuotų elementų.

2.2.3. K-vidurkių klasterizacijos diskretizavimas

K-vidurkių metodas remiasi našumo indekso, apibrėžiamo kaip visų vektorių kvadratinis atstumas iki klasterio centro, minimizavimu. Jis gali būti nesunkiai modifikuotas vienam požymiui [7].

Tegu duomenų rinkinys susideda iš M objektų ir C klasių, ir yra nurodytas intervalų skaičius n_{Fi} atributui F_i. Tuomet k-vidurkių klasterizacijos diskretizavimo algoritmas atliekas taip [7]:

1. Kiekvienai klasei c_j , $j = 1, \ldots, C$ daryti

2. Pasirinkti $K = n_{Fi}$ kaip pradinius klasterių centrus. Pradžioje galima paimti pirmas K savybės reikšmes, kaip klasterių centrus.

3. Paskirstyti likusias savybės reikšmes tarp K klasterių centrų, kaip kriterijų naudojant minimalų atstumą. Dėl to savybės reikšmės susirenka apie atnaujintus K klasterių centrus.

4. Apskaičiuoti K naujų klasterių centrų, taip kad kvadratinio atstumo suma tarp naujo centro ir visų klasterio objektų būtų minimali.

5. Patikrinti ar atnaujinti K klasterių centrai sutampa su senaisiais centrais. Jei taip – tai eiti į pirmą žingsnį; jei ne – pereiti prie trečio žingsnio.

Rezultatas: galutinės savybių ribos susideda iš minimalios savybės reikšmės, vidutinės reikšmės tarp šalia esančių klasterių prototipų visoms klasėms ir didžiausios savybės reikšmės.

Jei informacijos apie klasę nėra, tada $j = 1$ kiekvienai savybei, ir gaunamas intervalų skaičius bus lygus n_{Fi}.

2.2.4. Diskretizacijos algoritmų lyginamosios analizės rezultatai

Atlikus vienodo pločio, vienodo dažnio ir k-vidurkių diskretizacijos algoritmų testavimą su realiais ir sugeneruotais nustatyta, kad įvestų kriterijų atžvilgiu parinktiems duomenų rinkiniams optimalus yra vienodo pločio diskretizacijos metodas [7]. Alternatyva jam gali būti vienodo dažnio diskretizacijos algoritmas, kuris turi daugelį vienodo pločio diskretizavimo algoritmo savybių, bet laiko kaštai augant objektų skaičiui didėja sparčiau. Galima daryti prielaidą, kad vienodo pločio algoritmas yra optimalus ir kitų sistemų sukauptiems duomenų rinkiniams.

2.3. Išskirčių nustatymas

Duomenų rinkiniuose gali būti objektų, kurių atributai turi per daug dideles arba mažas reikšmes lyginant su kitų objektų atitinkamų atributų reikšmėmis. Tokios reikšmės vadinamos išskirtimis. Atliekant klasterių sudarymą, kai duomenų rinkinyje yra išskirčių, gali būti sudaryti blogai atskirti klasteriai. Todėl reikia pašalinti objektus, turinčius savyje išskirčių. Išskirčių nustatymui taikoma statistinė duomenų analizė.

Sąlygine išskirtimi vadinamas duomuo, priklausantis intervalui [14]:

$$\left[\left(Q_1 - 3 \cdot \mathrm{IQR}, \mathrm{Q}_1 - 1{,}5 \cdot \mathrm{IQR}\right) arba \left(Q_3 + 1{,}5 \cdot \mathrm{IQR}, \mathrm{Q}_3 + 3 \cdot \mathrm{IQR}\right)\right]. \quad (19)$$

Išskirtimi vadinamas duomuo mažesnis už

$$Q_1 - 3 \cdot \mathrm{IQR}, \quad (20)$$

arba didesnis už

$$Q_3 + 3 \cdot \mathrm{IQR}. \quad (21)$$

Čia Q_1 ir Q_3 yra pirmasis ir trečiasis kvartiliai. Nėra bendros kvartilių skaičiavimų metodikos. Vienas iš būdų sako, kad kvartiliai dalina variacinę eilutę į keturias maždaug lygias dalis [14]. Taip randami Q_1, Q_2 ir Q_3 kvartiliai. Pirmasis kvartilis apima 25% pirmųjų variacinės eilutės

narių, o trečiasis – 75%. Antrasis kvartilis apima 50% pirmųjų variacinės eilutės narių – jis sutampa su variacinės eilutės mediana. Trečiojo ir pirmojo kvartilių skirtumas vadinamas kvartilių skirtumu ir žymimas *IQR*.

Taigi iš pateikto duomenų rinkinio sudarius variacinę eilutę ir pasinaudojus kvartilių skaičiavimu, yra nustatomos tiek sąlygines išskirtys, tiek pačios išskirtys. Nustatytas išskirtis galima pašalinti iš pateikto duomenų rinkinio ir iš atnaujinto duomenų rinkinio sudaryti klasterius. Jei iš duomenų rinkinio nėra pašalintos išskirtys, tai jos gali įtakoti objektų priskyrimą klasteriams ir pačių klasterių išsidėstymą vienas kito atžvilgiu.

Pastaba *p4*: Kaip reikia nustatyti išskirtis, jei pateikto duomenų rinkinio objektai turi daugiau nei vieną atributą? Vienas iš galimų šios problemos sprendimų būdų – nustatyti išskirtis kiekvienam duomenų rinkinio atributui ir iš duomenų rinkinio pašalinti tuos objektus, kurių nors vienas atributas turi nustatytą išskirtį.

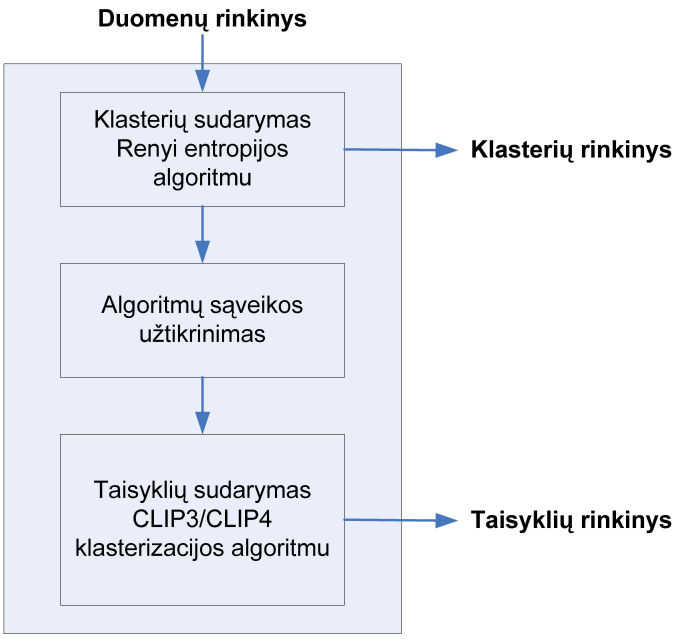
3. REZULTATAI IR JŲ APTARIMAS

Renyi entropijos klasterizacijos algoritmas ne tik sudaro klasterius, bet ir pagal tarpklasterinės entropijos pokytį nustato, koks yra tikrasis natūraliai suformuotas jų skaičius. Tačiau šis algoritmas negeneruoja klasterizacijos taisyklių, pagal kurias būtų galima spręsti, į kurį klasterį reikia įtraukti naujai pateiktą duomenų objektą. CLIP3 bei CLIP4 konceptualios klasterizacijos algoritmai sudaro objektų priskyrimo klasteriams taisykles. Šiems algoritmams turi būti pateikti teigiamų ir neigiamų egzempliorių (*POS* ir *NEG* atitinkamai) duomenų rinkiniai, pagal kuriuos generuojamos tokios taisyklės, kad teigiamų egzempliorių duomenų rinkinio objektai patenka į *POS* ir nepateka į *NEG* duomenų rinkinį. *NEG* duomenų rinkinio egzemplioriai negali patekti į *POS* rinkinį.

1 skyriuje atlikta Renyi entropijos, ITERATE, CLIP3 ir CLIP4 klasterizacijos algoritmų lyginamoji analizė. Pagal gautus rezultatus 3.1 skyriuje pristatomas konceptualios klasterizacijos algoritmas, kuris paremtas Renyi entropijos, CLIP3 bei CLIP4 algoritmų geriausiomis savybėmis. Šis algoritmas pavadintas modifikuotu Renyi-CLIPx konceptualios klasterizacijos algoritmu. Eksperimentų metu tiriama modifikuoto algoritmo rezultatų priklausomybė nuo standartinio nuokrypio (3.2.1 skyrius), inicijuojamų klasterių skaičiaus (3.2.2 skyrius) parinkimo. Yra stebima diskretizacijos (3.2.3 skyrius), genetinio algoritmo modulio (3.2.4 skyrius), neigiamo duomenų rinkinio užrašymo (3.2.5 skyrius) įtaka klasterizacijos taisyklėms. Šie eksperimentai atliekami siekiant išspręsti 1 skyriuje minėtas klasterizacijos algoritmų problemas. 3.2.6 skyriuje tiriami modifikuotu Renyi-CLIPx konceptualios klasterizacijos algoritmu sudaromi klasteriai ir generuojamos taisyklės virtualios mokymo(si) aplinkos duomenims. Visi eksperimentai atliekami kompiuteriu, kurio procesorius Intel®Core™2, taktinis dažnis 1,83 GHz, darbinės atminties kiekis 1024 Mb, įdiegta Microsoft®Windows Vista™ Ultimate SP1 operacijų sistema.

3.1. Modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas

Kadangi dalis klasterizacijos algoritmų formuoja klasterius, o kiti generuoja taisykles iš jau paruoštų klasterių, tai galima sudaryti *modifikuotą algoritmą*, kuris ne tik atskiria klasterius, bet ir generuoja klasterizacijos taisykles. Šis algoritmas gali būti realizuotas taikant Renyi entropijos, CLIP3 bei CLIP4 klasterizacijos algoritmus. 12 pav. pateikta modifikuoto Renyi-CLIPx konceptualios klasterizacijos algoritmo struktūros schema. Joje Renyi entropijos algoritmą galima pakeisti bet kokiu klasterius formuojančiu, o CLIP3 ar CLIP4 – taisykles generuojančiu algoritmu.



12 pav. ***Modifikuoto Renyi-CLIPx klasterizacijos algoritmo struktūros schema***

Renyi entropijos klasterizacijos algoritmas suskirsto duomenų rinkinio stebėjimus į atskirus klasterius, o CLIP3 arba CLIP4 algoritmas generuoja klasterizacijos taisykles (12 pav.).

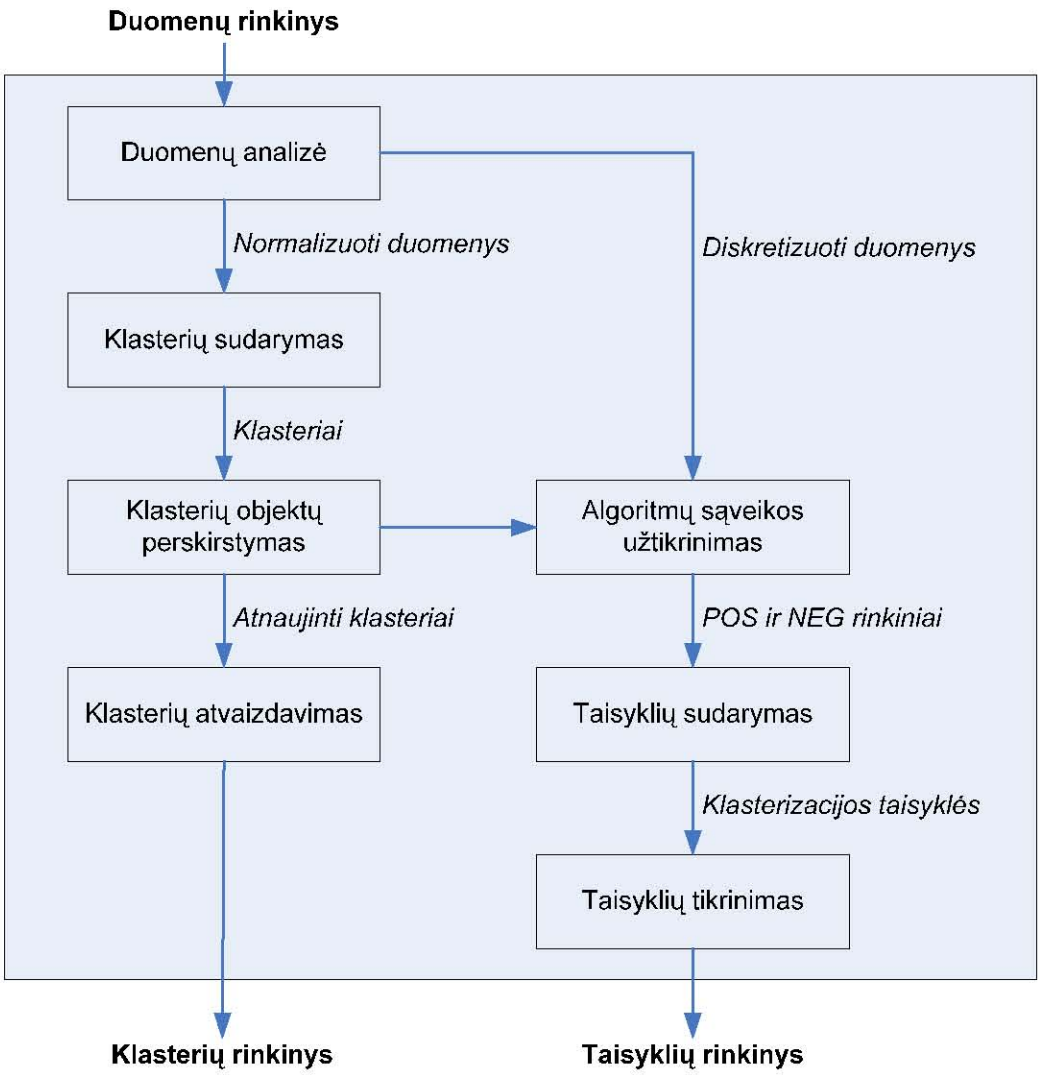
Pastaba *p5*: Kadangi CLIP3 bei CLIP4 algoritmams reikia pateikti po vieną teigiamų ir neigiamų stebėjimų duomenų rinkinį, tai įvedama ši taisyklė:

- Duomenų rinkinį *POS* sudaro elementai to klasterio, kurio egzemplioriams yra generuojamos taisyklės, o visų likusių klasterių egzemplioriai įtraukiami į *NEG* rinkinį.

Taigi generuojant klasterizacijos taisykles, kiekvieną kartą yra sudaromi *POS* ir *NEG* duomenų rinkiniai, o klasteriams paruošiamos unikalios taisyklės.

Sąveiką tarp Renyi ir CLIP3/CLIP4 klasterizacijos algoritmų realizuoja *algoritmų sąveikos modulis*. Pastarasis modulis užtikrina rezultatų transformavimą ir perdavimą tarp atskirų algoritmų – perduodama informacija apie Renyi entropijos algoritmo sudarytus klasterius CLIP3/CLIP4 algoritmui. Turint objektų priskyrimo klasteriams identifikatorius, galima sudaryti klasterizacijos taisykles.

Taigi modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas iš duomenų rinkinio nustato tikrą klasterių skaičių, sudaro šiuos klasterius ir kiekvienam jam generuoja unikalias klasterizacijos taisykles.



13 pav. ***Praplėsta modifikuoto Renyi-CLIPx klasterizacijos algoritmo struktūros schema***

13 pav. pateikta praplėsta modifikuoto Renyi-CLIPx konceptualios klasterizacijos algoritmo struktūros schema. Čia įvedami keturi moduliai:

- duomenų apdorojimo,
- klasterių objektų perskirstymo,
- klasterių atvaizdavimo,
- klasterizacijos taisyklių testavimo.

Kiekvienas šis modulis reikalingas sudarant gerai atskirtus klasterius ir patikimas klasterizacijos taisykles.

3.1.1. Duomenų apdorojimo modulis

Duomenų apdorojimo modulis skirtas padidinti klasterizacijos algoritmų vykdymo spartą, sudaryti kokybiškesnius klasterius bei geresnes taisykles. Šiame modulyje nagrinėjamas pateiktas duomenų rinkinys. Duomenų rinkinio analizės metu galima atlikti šiuos veiksmus:

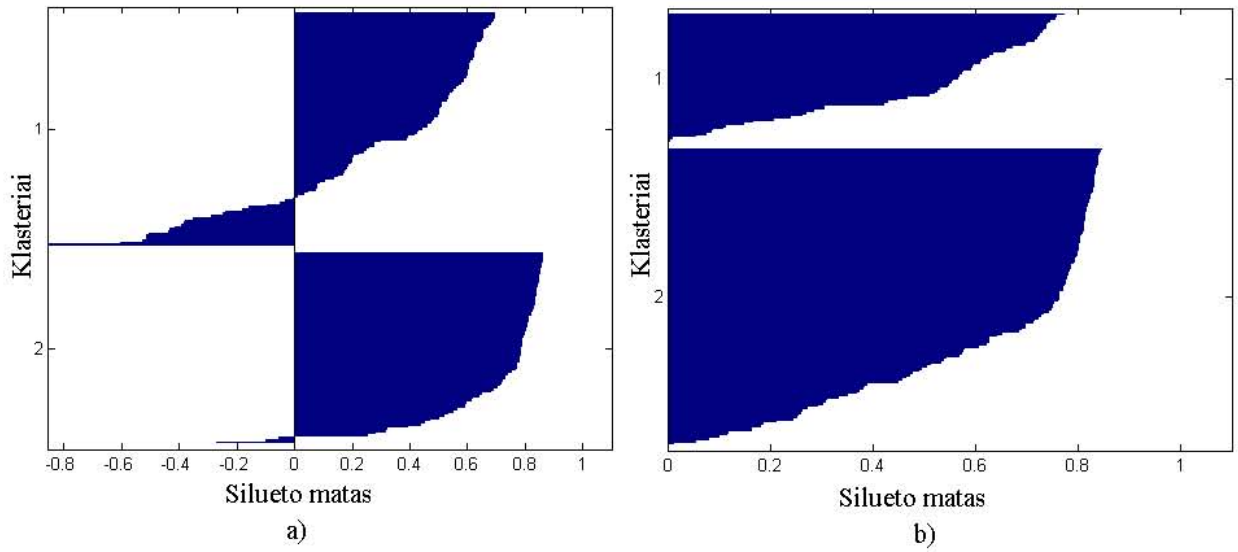
- nustatyti ir pašalinti sąlygines išskirtis ir išskirtis,
- pašalinti pasikartojančius objektus,
- diskretizuoti duomenis,
- normalizuoti duomenis.

Šie veiksmai atliekami atsižvelgiant į pateiktą duomenų rinkinį ir naudojamus klasterizacijos algoritmus.

Pastaba *p6*: Atlikus pateiktų duomenų normalizaciją arba diskretizaciją, būtina patikrinti, ar atnaujintame duomenų rinkinyje neatsirado pasikartojančių objektų. Jei iš pateiktų duomenų sudarome klasterius, o sekančiame etape diskretizuojame duomenis, tai gali būti taip, kad diskretizuoti objektai, turintys tokias pačias atributų reikšmes, yra įtraukti į skirtingus klasterius. Tokiu atveju neįmanoma sukurti konceptus aprašančių taisyklių. Siekiant to išvengi, reikia pašalinti pasikartojančius objektus, jei jie patenka į skirtingus klasterius, arba parinkti platesnį diskretizacijos intervalą.

3.1.2. Klasterių objektų perskirstymo modulis

Klasterių objektų perskirstymo modulis pagerina sudarytų klasterių kokybę. Sudarytų klasterių kokybei tirti gali būti taikomi keli būdai. Konkrečios priemonės pasirinkimas priklauso nuo nagrinėjamų duomenų savybių. Galima naudoti objektų siluetų (angl. k. *silhouette*) matų skaičiavimo būdą ir jų atvaizdavimą diagramomis. Siluetų matų diagrama vaizduoja, kaip arti kitų klasterių objektų yra tiriamo klasterio objektas. Šiame darbe vieno objekto nutolimas nuo kito skaičiuojamas pagal Euklido atstumą. Idealiu atveju (kai klasteriai yra maksimaliai atskirti vienas nuo kito ir objektas teisingai priskirtas klasteriui) pastarasis dydis yra lygus +1. Jeigu nėra aišku, kuriam klasteriui priklauso objektas, tai šis dydis yra lygus 0. Kai yra akivaizdu, kad objektas klaidingai įtrauktas į tiriamą klasterį, silueto mato dydis lygus -1. Taigi šis objektų priskyrimo klasteriams „gerumo“ matas įgyja reikšmes iš intervalo [-1; 1]. Siluetų diagramų pavyzdžiai pateikti 14 pav.

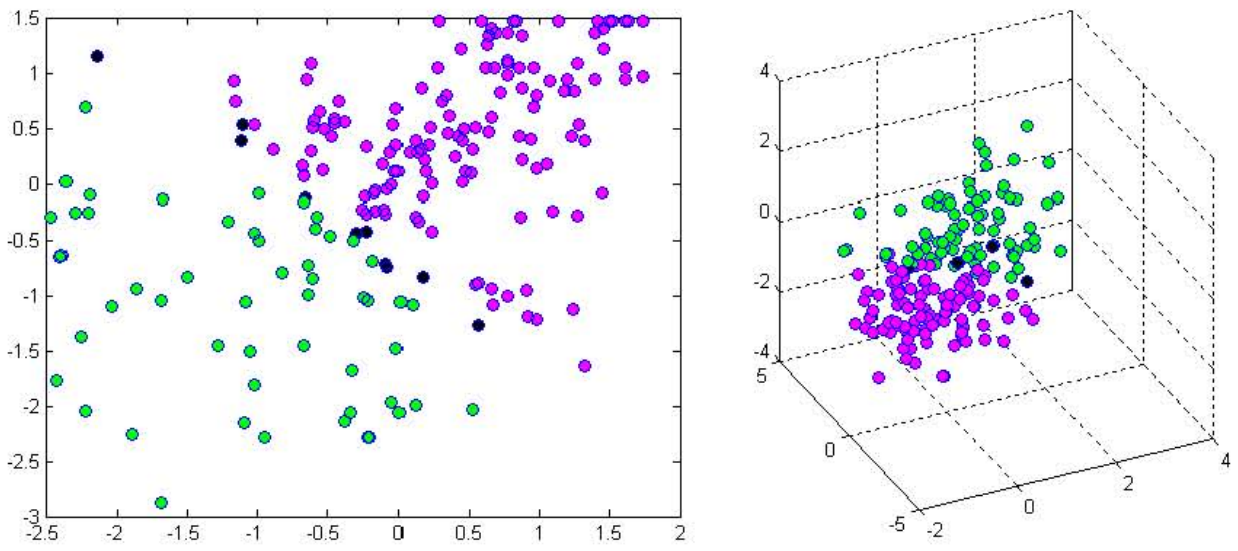


14 pav. ***Siluetų diagramų pavyzdžiai***

14 pav. pavaizduotos dviejų klasterių siluetų matų diagramos. Čia b) atveju visų sudarytų klasterių objektų siluetų matas yra teigiamas. Tai reiškia, kad objektų priskyrimas klasteriams yra teisingas. Kuo artimesnė vienetui šio mato reikšmė, tuo labiau tikėtina, kad objektas teisingai įtrauktas į klasterį. Turint klasterių objektų siluetų matus, pateiktam duomenų rinkiniui galima apskaičiuoti šių matų vidurkį. Pastarasis vidurkis padeda nuspręsti, koks yra tikrasis klasterių skaičius. Jei iš tų pačių duomenų yra sudaryti keli klasterių rinkiniai, tai laikoma, kad natūralus klasterių rinkinys yra tas, kurio siluetų matų vidurkis yra didžiausias. Atsižvelgiant į klasterių silueto mato savybes, šį matą galima pritaikyti klasterių objektų perskirstymo modulyje. Jei pagal šį santykinį matą yra nustatoma, kad objekto priskyrimas klasteriui yra abejotinas (14 pav. a), tai atliekamas šio objekto perskirstymas – objektas įtraukiamas į tą klasterį, kurio silueto matas padidėja daugiausiai. Tikimasi, kad po šio perskirstymo objektų priskyrimo klasterių įvertinimas bus geresnis (14 pav. b).

3.1.3. Klasterių atvaizdavimo modulis

Klasterių atvaizdavimo modulis vaizdžiai pateikia nagrinėjamų objektų priskyrimą klasteriams. Patogu klasterizacijos rezultatus analizuoti grafiniu pavidalu [19]. Kai kuriais atvejais šis rezultatų pateikimo būdas padeda nustatyti, ar duomenų rinkinys yra tinkamai suskirstytas į klasterius. Tarp gerai atskirtų klasterių yra riba, kuri skiria skirtingų klasterių objektus. Jei vieno klasterio objektai patenka į kito klasterio suformuotą sritį, tai galima daryti prielaidą, kad klasteriai nėra gerai atskirti.



15 pav. ***Klasterių pavaizdavimo atvejai***

15 pav. a) ir b) atvejais pateiktas duomenų, turinčių 2 ir 3 atributus atitinkamai, priskyrimas klasteriams. Čia matavimų skalės susietos su normalizuotais studentų įvertinimo balais. Sudarytų klasterių gerumą grafiškai galima atvaizduoti ir siluetų diagramomis. 14 pav. a) ir b) pateiktos siluetų matų diagramos prieš ir po objektų priskyrimo klasteriams perskirstymo.

3.1.4. Klasterizacijos taisyklių testavimo modulis

Klasterizacijos taisyklių testavimo modulis tikrina modifikuoto konceptualios klasterizacijos algoritmo sudaromų taisyklių gerumą. Klasterizacijos taisyklės turi būtų kuo paprastesnės ir aprašyti daugumą *POS* duomenų rinkinio egzempliorių. Taisyklių rinkinys laikomas geru, jei pagal jį visi teigiami egzemplioriai patenka tik į *POS* duomenų rinkinį, o neigiami tik į *NEG* rinkinį. Taikomi ir kiti kriterijai taisyklių gerumui užtikrinti. Pavyzdžiui, galima nurodyti, kiek mažiausiai kiekviena taisyklė turi aprašyti skirtingų atributų, egzempliorių, arba koks turi būti minimalus taisyklės sudėtingumas (loginių sąlygų skaičius). Modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas taisyklių gerumą užtikrina įvesdamas vykdymo ribas (triukšmo, geriausios taisyklės, skaidymo ir sustojimo ribos) bei genetinį modulį [11]. Taip siekiama, kad taisyklės nebūtų generuojamos tik vienam objektui, o tiktų didesniam objektų skaičiui. Kai kuriais atvejais pritaikius šiuos kriterijus sugeneruotas taisyklių rinkinys neaprašo viso *POS* duomenų rinkinio. Tuomet pasitelkiama ekspertų nuomonė.

Taigi įvestų keturių modulių pagalba modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas iš pateikto duomenų rinkinio ne tik sudaryto pakankamai gerai

atskirtus klasterius, generuota nesudėtingas klasterizacijos taisykles, bet ir aiškiai pateikia rezultatus.

3.2. Klasterių ir taisyklių sudarymas modifikuotu klasterizacijos algoritmu

Klasterių sudarymui ir klasterizacijos taisyklių išvedimui naudojami šie duomenų rinkiniai:

- Duomenų rinkinys *AP*. Šis duomenų rinkinys turi 202 objektus, kurių atributų visos reikšmės yra žinomos. Čia yra sukaupta informacija apie 202-jų studentų, klausiusių Asemblerinio programavimo ir valdymo kursą (kurso identifikatorius KU virtualioje mokymo(si) aplinkoje – 908009), atsiskaitymų rezultatai. Duomenų rinkinys *AP* turi 2 atributus.
- Duomenų rinkinys *ML*. Šis duomenų rinkinys turi 181 objektą su 3 atributais. Šie duomenys sukaupti KU virtualioje mokymo(si) aplinkoje apie 181 Matematinės logikos kurso (identifikatorius – 908005) studentų trijų atsiskaitymų rezultatus. Visų duomenų rinkinio objektų atributų reikšmės yra žinomos.
- Duomenų rinkinys *WINE*. Šis duomenų rinkinys turi 178 objektus ir 13 atributų. Jame sukaupta informacija apie trijų Italijos regione esančių vynuogynų derliaus cheminę analizę. Yra tiriama 13 cheminių elementų ir atlikta 178 stebėjimai. Visų stebimų atributų reikšmės yra žinomos. Pagal ekspertų nuomonę yra skiriamos trys pateikto duomenų rinkinio klasės. Pirmajai klasei priskiriami 59, antrajai – 71, trečiajai – 48 duomenų rinkinio objektai. Pateiktam duomenų rinkiniui yra nurodoma, kuriai klasei priklauso kiekvienas objektas. Duomenų rinkinys *WINE* svarbus tuo, kad remiantis pateikta informacija galima patikrinti, kiek gauti klasterizacijos rezultatai atitinka ekspertų nuomonę.

AP ir *ML* duomenų rinkiniai paimti iš KU virtualios mokymo(si) aplinkos duomenų bazės. Juose yra sukaupta informacija apie tam tikrų universitete dėstomų kursų studentų pasiekimus. Šie pasiekimai siejami su atliktų testų rezultatais ir yra matuojami skalėje nuo 0 iki 100. Galima naudoti ir kitus studentų pasiekimų duomenų rinkinius. *AP*, *ML* ir *WINE* duomenų rinkinius sekančiuose etapuose tiriame modifikuotu Renyi-CLIPx konceptualios klasterizacijos algoritmu.

3.2.1. Klasterizacijos rezultatų priklausomybė nuo standartinio nuokrypio reikšmės

Renyi entropijos klasterizacijos algoritmas paremtas entropijos nustatymu – skaičiuojama tarpklasterinė entropija ir entropija klasterių viduje. Renyi entropijos tikimybinio tankio funkcija gali būti tiesiogiai apskaičiuota iš duomenų rinkinio pagal Parzeno lango (angl. k. *Parzen window*) su Gauso branduolio funkcija atitikmenį. Daugiamatės Gauso branduolio funkcijos standartinio nuokrypio (σ) reikšmės parinkimas yra dažnai minima problema [3, 15]. Siūloma atlikti kelis eksperimentus su skirtingomis σ reikšmėmis ir nustatyti, kuriai standartinio nuokrypio reikšmei esant gaunami geriausi rezultatai [15]. Tokiu būdu parenkama pastovi standartinio nuokrypio reikšmė. Tai reiškia, kad skaičiuojant entropiją σ reikšmė visais atvejais yra vienoda, nors nagrinėjamų klasterių įtraukti duomenys gali labai skirtis. Įvardintą problemą autorius siūlo spręsti skaičiuojant standartinį nuokrypį tiesiogiai iš duomenų rinkinio.

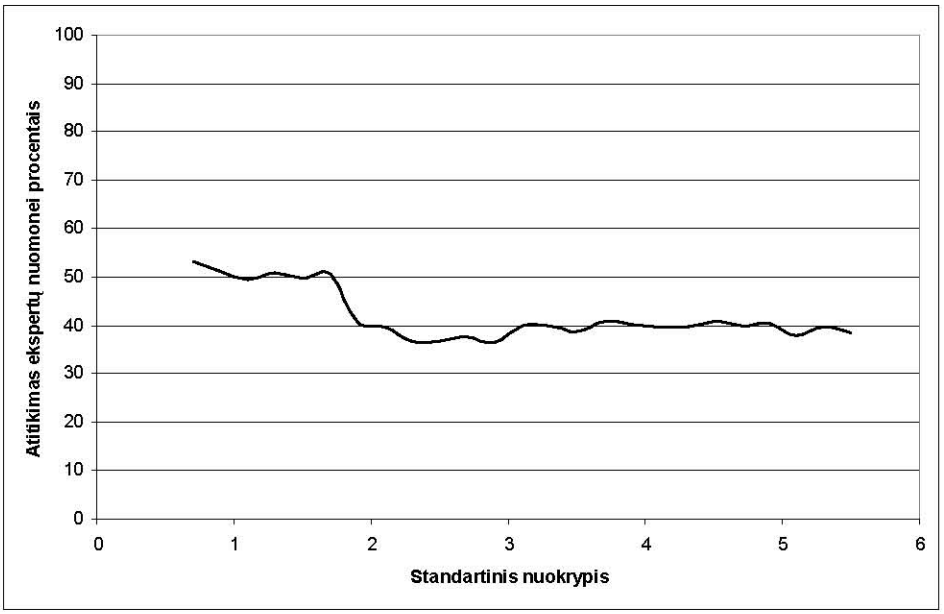
Toliau tiriama modifikuoto Renyi-CLIPx konceptualios klasterizacijos algoritmo sudarytų klasterių gerumo priklausomybė nuo standartinio nuokrypio reikšmės parinkimo. Eksperimentų metu nagrinėjami šie variantai:

- Nenormalizuotų duomenų klasterizacijos rezultatų priklausomybė nuo standartinio nuokrypio reikšmės.
- Normalizuotų duomenų klasterizacijos rezultatų priklausomybė nuo standartinio nuokrypio reikšmės.

Kadangi *WINE* duomenų rinkiniui yra pateikta ekspertų nuomonė, pasakanti kuriam klasteriui priklauso kiekvienas objektas, tai su šiuo rinkiniu atliekami sekantys tyrimai. Siekiant užtikrinti rezultatų patikimumą, kiekvienai parinktai standartinio nuokrypio reikšmei eksperimentas kartojamas 10 kartų, ir apskaičiuojamas gautų rezultatų vidurkis. Pradinis inicijuojamų klasterių skaičius K_{init} visais atvejais lygus 20. Keičiant standartinio nuokrypio reikšmę, stebimas klasterizacijos rezultatų atitikimas ekspertų nuomonei procentais ir bendras gerumas. Čia bendras gerumas suprantamas kaip gautų rezultatų siluetų matų vidurkis visiems sudarytiems klasteriams. Kuo klasterių silueto matų reikšmės artimesnės vienetui, tuo klasteriai geriau atskirti vienas nuo kito, ir duomenų rinkinio objektai tinkamiau priskirti jiems. Pagal šiuos įvestus kriterijus lyginami gauti ir perskirstyti klasterizacijos rezultatai. Objektų perskirstymas atliekamas Renyi-CLIPx konceptualios klasterizacijos algoritmo klasterių objektų perskirstymo modulio pagalba.

3.2.1.1. Nenormalizuotų duomenų klasterizacijos rezultatų priklausomybė nuo standartinio nuokrypio

16 pav. pateiktas modifikuotu Renyi-CLIPx konceptualios klasterizacijos algoritmu gautų klasterių atitikimas procentais ekspertų nuomonei. Nagrinėjamas *WINE* duomenų rinkinys nėra normalizuotas, o sudaromų klasterių skaičius yra lygus 3. Standartinio nuokrypio reikšmės imamos iš intervalo $[0{,}8;\ 5{,}5]$. Tyrimo metu σ reikšmės keitimo žingsnis yra 0,1. Rezultatai standartinio nuokrypio intervalui $(0;\ 0{,}8)$ nėra pateikti, nes Renyi entropijos nepavyko nustatyti pagal šias reikšmes – esant standartinio nuokrypio reikšmėms iš intervalo $(0;\ 0{,}8)$, *WINE* duomenų Gauso branduolio funkcija įgyja reikšmę lygią arba labai artimą 0, o logaritmas nuo 0 negalimas.

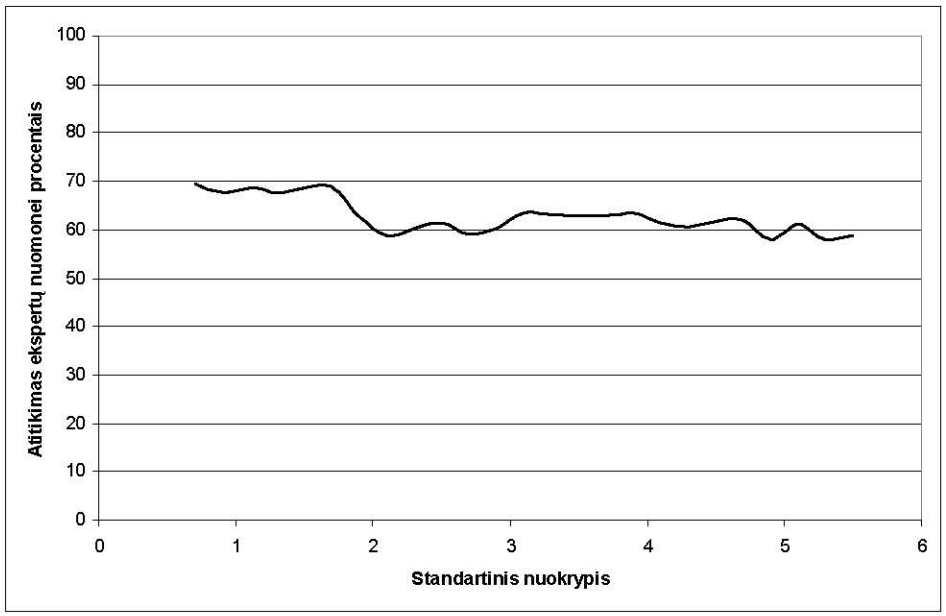


16 pav. ***WINE duomenų klasterių atitikimas ekspertų nuomonei pagal parinką standartinį nuokrypį***

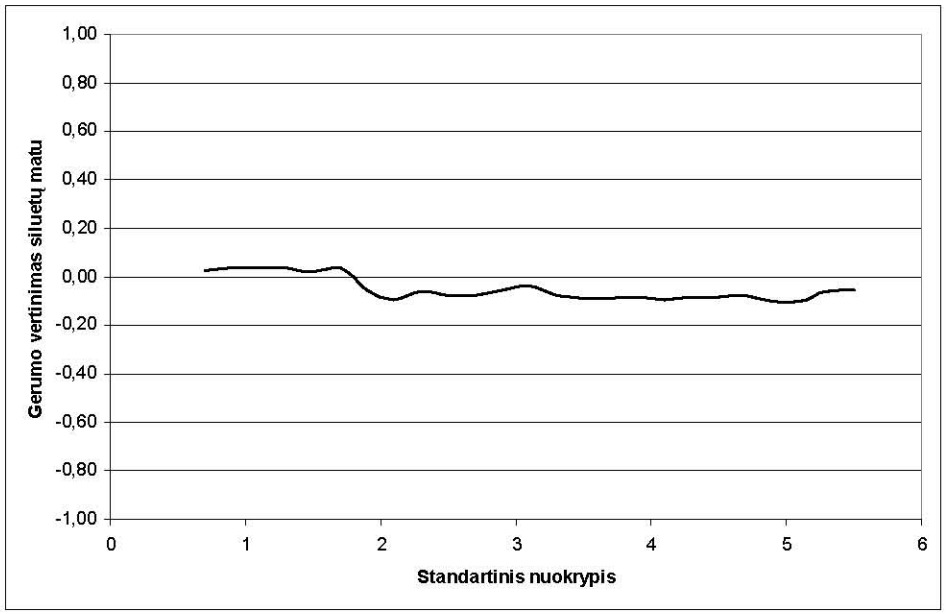
Didinant standartinio nuokrypio reikšmę, atitikimas ekspertų pateiktai nuomonei pasidaro beveik pastovus – 39,82%, o atitikimo vidurkis intervale $[0{,}8;\ 5{,}5]$ yra lygus 41,96% (16 pav.). Šis atitikimas didesnis, kai standartinio nuokrypio reikšmė yra iš intervalo $[0{,}8;1{,}7]$ – 50,83%. Taigi sudarytų klasterių atitikimas ekspertų nuomonei yra mažas, kai *WINE* duomenų rinkinys yra nenormalizuotas, o klasterizacijos rezultatai nėra perskirstyti.

Perskirstytų klasterizacijos algoritmo rezultatų atitikimas ekspertų nuomonei pateiktas 17 pav. Rezultatų perskirstymą atlieka klasterių objektų perskirstymo modulis. Vidutinis atitikimas ekspertų nuomonei yra lygus 62,82% o standartinio nuokrypio intervale $[0{,}8;1{,}7]$ – 68,98%.

Taigi vidutinis perskirstytų klasterizacijos rezultatų atitikimas pateiktai ekspertų nuomonei 23% didesnis nei neperskirstytų rezultatų. Tačiau šis įvertinimas yra gana mažas.

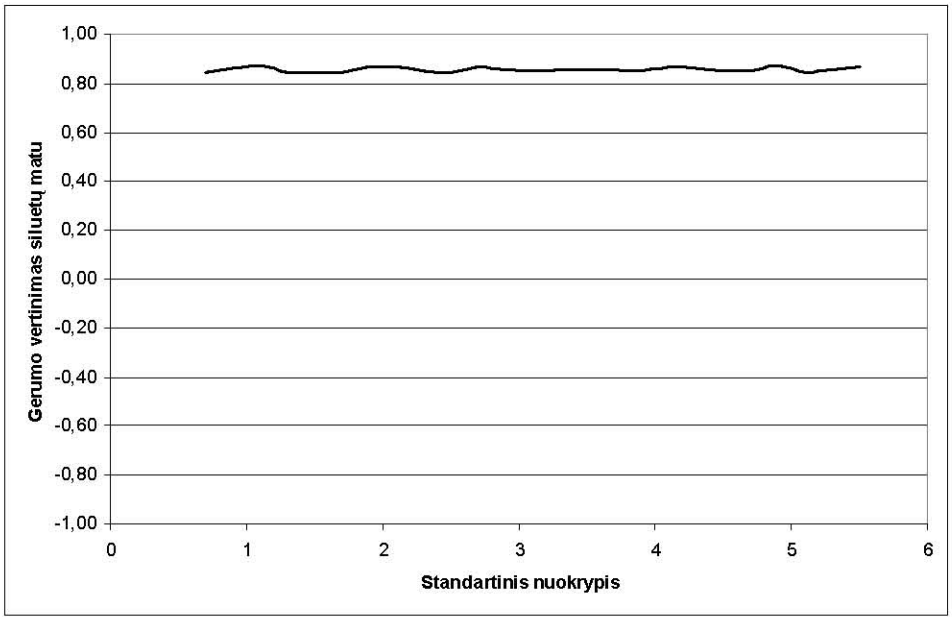


17 pav. ***WINE duomenų perskirstytų klasterių atitikimas ekspertų nuomonei pagal parinktą standartinį nuokrypį***



18 pav. ***WINE duomenų klasterių gerumo vertinimas pagal parinktą standartinį nuokrypį***

18 pav. pateiktas neperskirstytų klasterizacijos algoritmo rezultatų gerumas esant skirtingoms standartinio nuokrypio reikšmėms. Rezultatų gerumas paremtas siluetų mato skaičiavimu. Vidutinis gerumo įvertinimas visoms tiriamoms standartinio nuokrypio reikšmėms yra -0,05, kai tuo tarpu σ intervale $[0{,}8;\ 1{,}7]$ jis lygus 0,03. Kadangi maksimalus gerumo vertinimas gali būti 1, tai daroma prielaida, kad klasteriai nėra gerai atskirti.



19 pav. ***WINE duomenų perskirstytų klasterių gerumo vertinimas pagal parinktą standartinį nuokrypį***

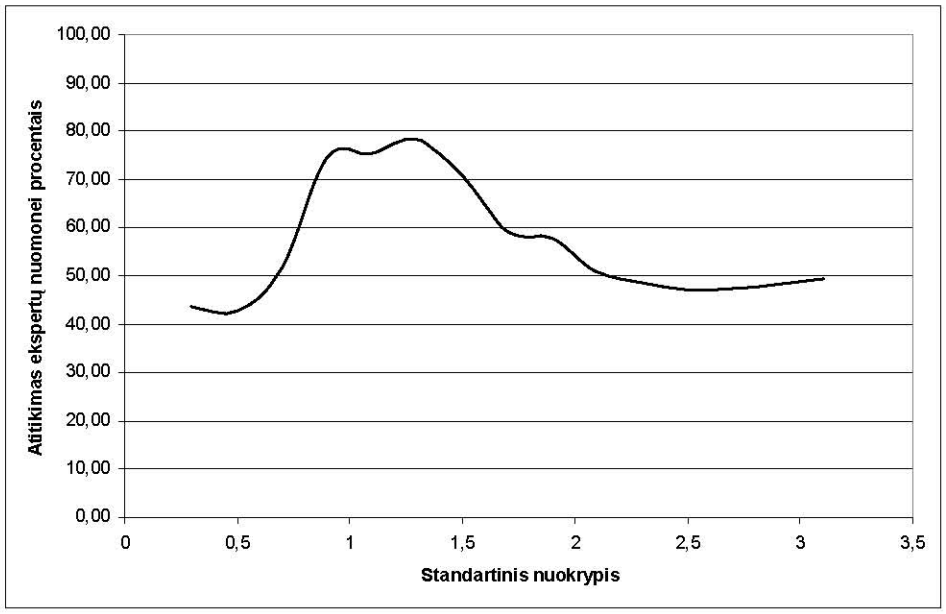
19 pav. pateiktas perskirstytų klasterizacijos algoritmo rezultatų gerumo vertinimas. Perskirstytų rezultatų gerumas didesnis, nei neperskirstytų (18 pav.). Čia vidutinis gerumas visoms tiriamoms standartinio nuokrypio reikšmėms yra lygus 0,86. Akivaizdu, kad klasterių objektų perskirstymo modulis siluetų mato požiūriu pagerina sudarytus klasterius.

Taigi perskirstytų klasterizacijos rezultatų vertinimas įvestų kriterijų atžvilgiu yra geresnis, nei neperskirstytų. Tačiau nenormalizuoto *WINE* duomenų rinkinio klasterizacijos rezultatų atitikimas pateiktai ekspertų nuomonei yra mažas. Parinkus standartinio nuokrypio reikšmes iš intervalo $[0{,}8; 1{,}7]$, rezultatų atitikimas ekspertų nuomonei ir gerumo vertinimas yra aukštesnis, nei esant kitoms σ reikšmėms.

3.2.1.2. Normalizuotų duomenų klasterizacijos rezultatų priklausomybė nuo standartinio nuokrypio

Duomenų rinkinio normalizavimo metu siekiama, kad kiekvieno požymio vidurkis būtų lygus 0, o standartinis nuokrypis 1. *WINE* duomenų normalizavimas atliekamas pagal (9) ir (10) formules. Šiame etape stebima klasterizacijos algoritmo rezultatų priklausomybė nuo standartinio nuokrypio reikšmės. Šiuo atveju sudaromi 3 klasteriai, kai *WINE* duomenų rinkinys yra normalizuotas. Nagrinėjamos standartinio nuokrypio reikšmės imamos iš intervalo $[0{,}3; 3{,}2]$. Tyrimo metu σ reikšmės keitimo žingsnis yra 0,1. Rezultatai standartinio nuokrypio intervalui

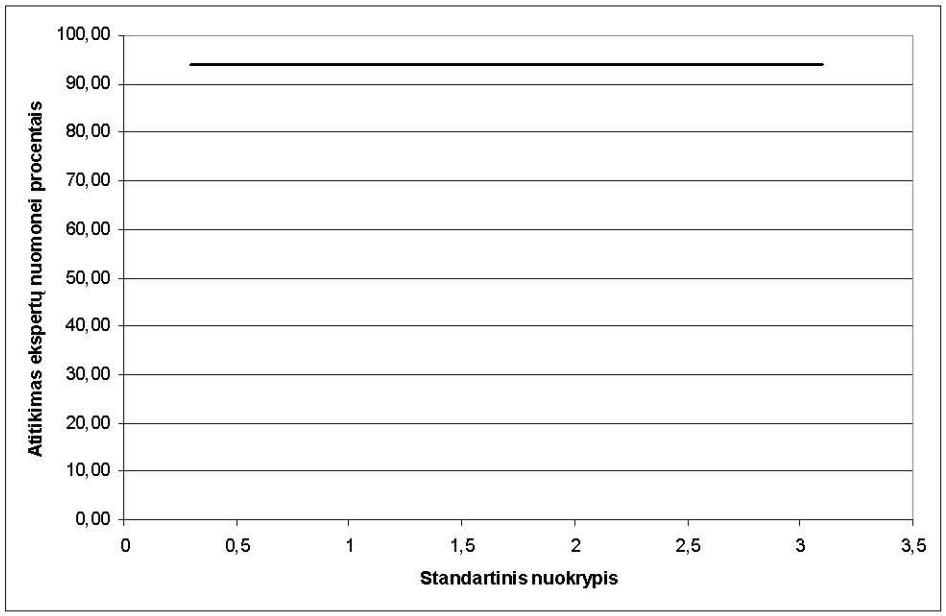
$(0;\ 0{,}3)$ nėra pateikti, nes šioms reikšmėms neina apskaičiuoti Renyi entropijos – Gauso branduolio funkcija įgyja reikšmę lygią arba labai artimą 0, o logaritmas nuo 0 negalimas.



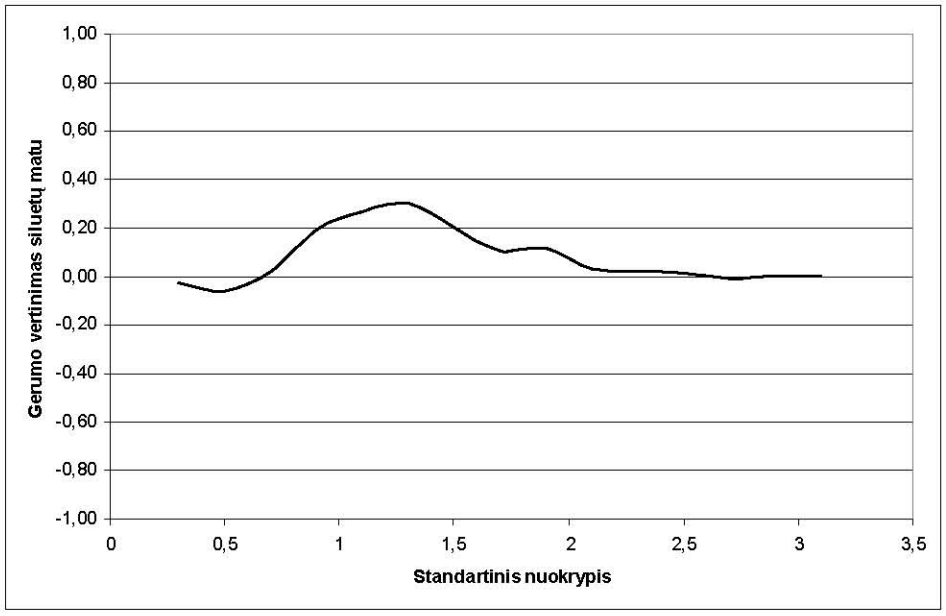
20 pav. ***WINE duomenų klasterių atitikimas ekspertų nuomonei pagal parinktą standartinį nuokrypį***

20 pav. pateiktas klasterizacijos algoritmu sudarytų klasterių atitikimas ekspertų nuomonei procentais. Šis vidutinis atitikimas visoms standartinio nuokrypio reikšmėms yra lygus 58,15%. Intervale $[0{,}9;\ 1{,}5]$ rezultatų įvertinimas yra didesnis – 74,79%. Gauti rezultatai geresni lyginant su nenormalizuotu *WINE* duomenų rinkiniu.

Iš 21 pav. matoma, kad normalizuoto *WINE* duomenų rinkinio perskirstytų klasterizacijos rezultatų atitikimas ekspertų pateiktai nuomonei esant skirtingoms standartinio nuokrypio reikšmėms yra pastovus ir lygus 93,82%. Šis atitikimo įvertinimas geresnis nei neperskirstytų rezultatų. Taigi siekiant suformuoti geriau atskirtus klasterius, reikia perskirstyti klasterizacijos rezultatus klasterių objektų perskirstymo modulio pagalba.

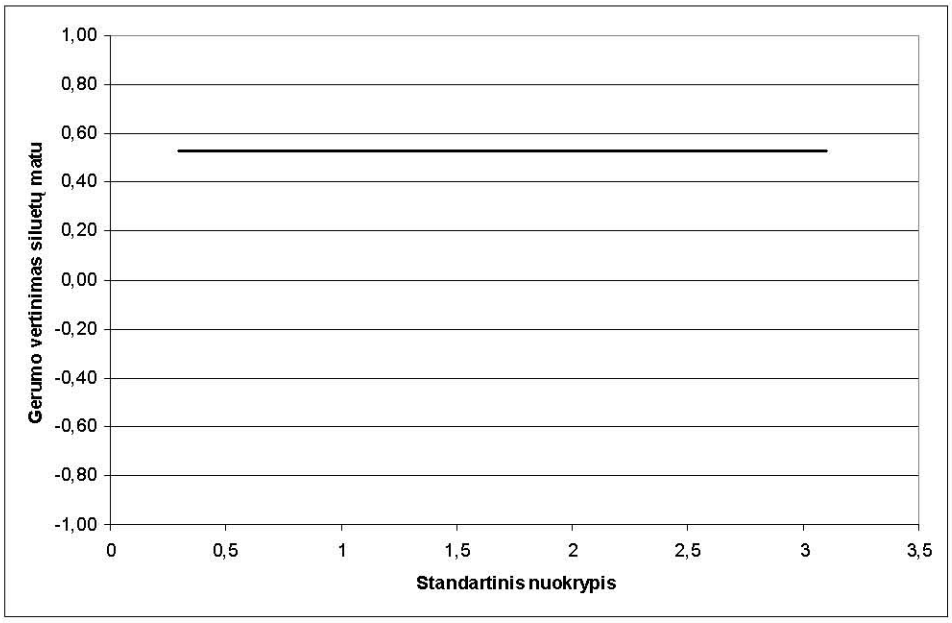


21 pav. ***WINE duomenų perskirstytų klasterių atitikimas ekspertų nuomonei pagal parinktą standartinį nuokrypį***



22 pav. ***WINE duomenų klasterių gerumo vertinimas pagal parinktą standartinį nuokrypį***

22 pav. pateiktas klasterizacijos algoritmo sudarytų klasterių gerumo vertinimas esant skirtingoms standartinio nuokrypio reikšmėms. Gerumo vertinimo vidutinė reikšmė yra lygi 0,08. Intervale $[0{,}9; 1{,}5]$ šis dydis įgyja geresnį įvertinimą – 0,24. Šie įvertinimai geresni lyginant su nenormalizuotų *WINE* duomenų įvertinimais, tačiau taip pat yra gana maži.



23 pav. ***WINE duomenų perskirstytų klasterių gerumo vertinimas pagal parinktą standartinį nuokrypį***

23 pav. pateiktas klasterizacijos algoritmu perskirstytų rezultatų gerumo vertinimas normalizuotam *WINE* duomenų rinkiniui. Sudarytų klasterių gerumo vertinimas yra pastovus – visame tirtame standartinio nuokrypio intervale įgyja reikšmę 0,53. Ši reikšmė mažesnė, nei tiriant klasterius, sudarytus iš nenormalizuotų *WINE* duomenų – tuomet gerumo vertinimas buvo lygus 0,86. Kadangi normalizuoto *WINE* duomenų rinkinio sudarytų klasterių atitikimas ekspertų pateiktai nuomonei yra aukštas, tai daroma prielaida, kad normalizavus *WINE* duomenų rinkinio objektų atributus, Euklido atstumas tarp skirtingų objektų taip pat sumažėja. Todėl silueto matų reikšmės yra mažesnės, nei turint nenormalizuotą duomenų rinkinį. Taigi normalizuoto *WINE* duomenų rinkinio siluetų mato reikšmė visiems sudarytiems klasteriams yra pastovi.

Normalizuoto *WINE* duomenų rinkinio klasterizacijos rezultatų atitikimas ekspertų nuomonei yra didesnis nei nenormalizuotų duomenų. Parinkus standartinio nuokrypio reikšmes iš intervalo $[0{,}9; 1{,}5]$, rezultatų atitikimas ekspertų nuomonei ir gerumo vertinimas yra aukštesnis. Perskirstytų klasterizacijos rezultatų vertinimas įvestų kriterijų atžvilgiu yra geresnis, nei neperskirstytų. Taigi siekiant gauti kokybiškesnius klasterius, būtina normalizuoti tiriamą duomenų rinkinį ir perskirstyti klasterizacijos rezultatus.

3.2.1.3. Klasterizacijos rezultatų priklausomybė nuo tiesiogiai iš duomenų skaičiuojamo standartinio nuokrypio

Standartinio nuokrypio reikšmę galima tiesiogiai apskaičiuoti iš duomenų. Tokiu būdu gauta σ reikšmė yra jautri nagrinėjamiems klasteriams. Atlikus 50 eksperimentų nustatyta, kad

skaičiuojant standartinio nuokrypio reikšmę iš duomenų rinkinio klasterizacijos rezultatų atitikimo vidurkis ekspertų pateiktai nuomonei yra lygus 61,62%, kai *WINE* duomenų rinkinys nėra normalizuotas ir gauti rezultatai neperskirstyti. Perskirsčius to paties duomenų rinkinio klasterizacijos rezultatus, atitikimas ekspertų nuomonei yra 68,60%. Tuo tarpu klasterių įvertinimas siluetų matų požiūriu vidutiniškai padidėja nuo 0,67 iki 0,84. Atlikus *WINE* duomenų rinkinio normalizaciją, gautų klasterių atitikimas ekspertų nuomonei yra lygus 91,08%, o perskirstytų klasterių atitikimas – 93,82%. Klasterių objektų perskirstymo modulis rezultatų įvertinimą siluetų matų požiūriu vidutiniškai padidina nuo 0,49 iki 0,53. Pastarasis matas mažesnis nei prieš tai nagrinėtu atveju dėl to, kad normalizuojant duomenis Euklido atstumas tarp atskirų klasterių objektų taip pat sumažėja.

Taigi iš normalizuotų *WINE* duomenų sudarytų klasterių atitikimas ekspertų nuomonei yra aukštas, kai gauti rezultatai perskirstomi klasterių objektų perskirstymo modulio pagalba, o standartinis nuokrypis tiesiogiai skaičiuojamas iš nagrinėjamų duomenų.

3.2.1.4. Standartinio nuokrypio įtakos klasterizacijos rezultatams apibendrinimas

2 ir 3 lentelėse pateikti apibendrinti modifikuoto Renyi-CLIPx klasterizacijos algoritmo rezultatai esant skirtingoms standartinio nuokrypio reikšmėms. Čia nagrinėjami atvejai, kai σ reikšmė parenkama iš tam tikro intervalo, ir kai ji apskaičiuojama tiesiogiai iš tiriamų objektų. Šiose lentelėse fiksuojamas vidutinis rezultatų atitikimas ekspertų pateiktai nuomonei bei gerumas pagal siluetų matą. Taip pat pateikiamas intervalas, iš kurio parenkant standartinio nuokrypio reikšmę gaunami rezultatai yra geresni nei jų nustatyti vidurkiai visame tirtame intervale.

2 lentelė. ***Klasterizacijos rezultatų vertinimas, kai standartinio nuokrypio reikšmė parenkama***

Duomenų normalizacija	Rezultatų perskirstymas	Įvertinimai
Ne	Ne	*Vidutinis atitikimas*: 41,96% *Vidutinis gerumas*: -0,05 *Siūlomas intervalas*: $[0,8; 1,7]$
Ne	Taip	*Vidutinis atitikimas*: 62,82% *Vidutinis gerumas*: 0,86 *Siūlomas intervalas*: $[0,8; 1,7]$
Taip	Ne	*Vidutinis atitikimas*: 58,15% *Vidutinis gerumas*: 0,08 *Siūlomas intervalas*: $[0,9; 1,5]$
Taip	Taip	*Vidutinis atitikimas*: 93,82%

		Vidutinis gerumas: 0,53 *Siūlomas intervalas*: >0

3 lentelė. ***Klasterizacijos rezultatų vertinimas, kai standartinis nuokrypis apskaičiuojamas***

Duomenų normalizacija	Rezultatų perskirstymas	Įvertinimai
Ne	Ne	*Vidutinis atitikimas*: 61,62% *Vidutinis gerumas*: 0,67 *Siūlomas intervalas*: >0
Ne	Taip	*Vidutinis atitikimas*: 68,60% *Vidutinis gerumas*: 0,84 *Siūlomas intervalas*: >0
Taip	Ne	*Vidutinis atitikimas*: 91,08% *Vidutinis gerumas*: 0,49 *Siūlomas intervalas*: >0
Taip	Taip	*Vidutinis atitikimas*: 93,82% *Vidutinis gerumas*: 0,53 *Siūlomas intervalas*: >0

Iš 2 bei 3 lentelių matoma, kad standartinio nuokrypio reikšmės daro skirtingą įtaką klasterizacijos algoritmo rezultatams esant tam pačiam duomenų rinkiniui. Nenormalizuoto *WINE* duomenų rinkinio gautų klasterizacijos rezultatų vidutinis atitikimas ekspertų nuomonei lygus 50,83%, kai standartinio nuokrypio reikšmė imama iš intervalo $[0{,}8; 1{,}7]$, o klasterizacijos rezultatai nėra perskirstyti. Jei klasterizacijos rezultatai perskirstomi, tai jų atitikimas ekspertų nuomonei yra 68,98%, kai standartinio nuokrypio reikšmė imama iš intervalo $[0{,}8; 1{,}7]$. Šiuose intervaluose gauti aukštesni sudarytų klasterių siluetų matų įvertinimai. Esant normalizuotam *WINE* duomenų rinkiniui, klasterizacijos rezultatų atitikimas ekspertų pateiktai nuomonei yra lygus 74,79%, kai σ reikšmė imama iš intervalo $[0{,}9; 1{,}5]$. Perskirstytų klasterizacijos rezultatų atitikimas ekspertų nuomonei nepriklauso nuo standartinio nuokrypio reikšmių – šis atitikimas lygus 93,82%. Pastoviu tampa ir klasterių vertinimas pagal siluetų matą – 0,53. Kai standartinio nuokrypio reikšmė tiesiogiai skaičiuojama iš nenormalizuoto duomenų rinkinio, tai vidutinis neperskirstytų rezultatų atitikimas ekspertų nuomonei yra 61,62%, o perskirstytų rezultatų – 68,60%. Normalizavus pateiktus duomenis, gaunamas atitikimas pasiskirstęs taip – 91,08% ir 93,82%.

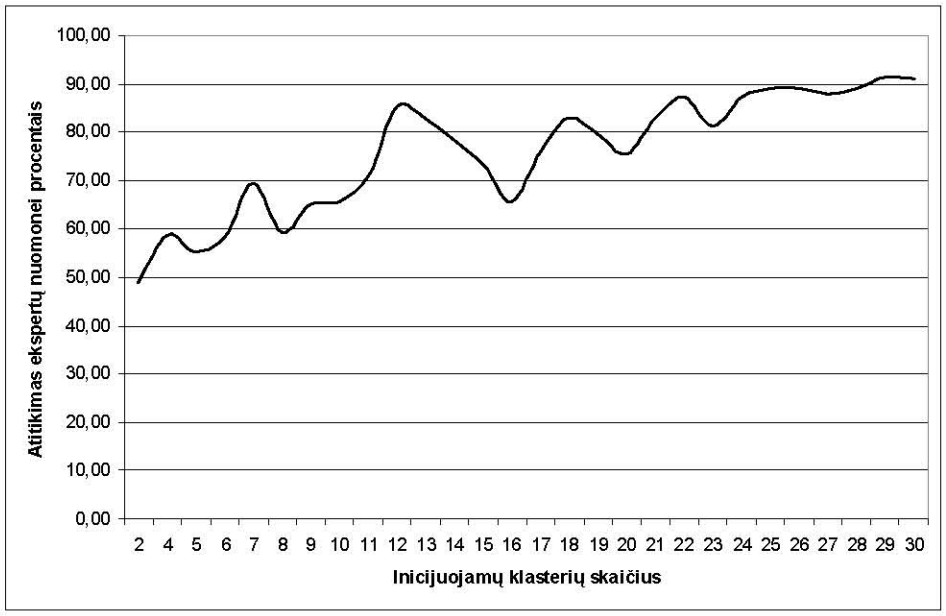
Taigi perskirsčius modifikuoto Renyi-CLIPx konceptualios klasterizacijos algoritmo rezultatus klasterių objektų perskirstymo modulio pagalba, gaunamas aukštesnis klasterių atitikimas pateiktai ekspertų nuomonei. Normalizavus tiriamą duomenų rinkinį, yra fiksuojami geresni rezultatai atitikimo ekspertų nuomonei ir siluetų mato kriterijų atžvilgiais.

3.2.2. Klasterizacijos rezultatų priklausomybė nuo inicijuojamų klasterių skaičiaus

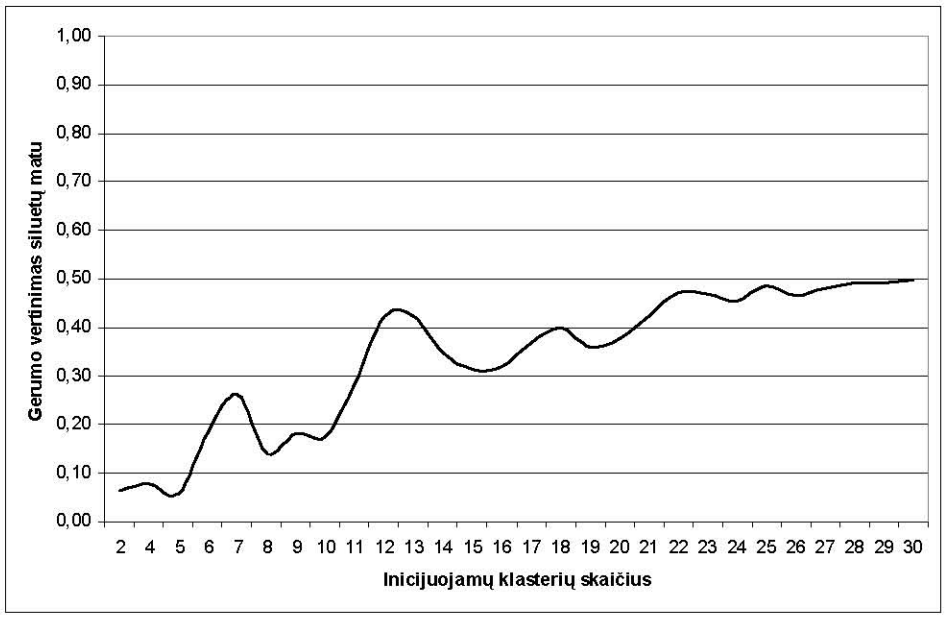
Modifikuoto Renyi-CLIPx konceptualios klasterizacijos algoritmo vykdymo pradžioje yra sukuriama K_{init} klasterių. Dydis K_{init} parenkamas vartotojo nuožiūra. Šalinant blogai atskirtus klasterius, jų skaičius nuosekliai mažinamas ir pagal gautus rezultatus yra nustatomas tikrasis išsidėstymas. Šiame etape tiriama inicijuojamų klasterių skaičiaus įtaka klasterizacijos rezultatams. Gauti rezultatai vertinami atitikimu ekspertų pateiktai nuomonei ir sudarytų klasterių siluetų matu. Kiekvienai K_{init} reikšmei eksperimentas kartojamas 10 kartų ir apskaičiuojamas gautų rezultatų vidurkis. Eksperimento vykdymo pradžioje inicijuojama 30 klasterių. Šis skaičius nuosekliai mažinamas, ir stebėjimas tęsiamas tol, kol lieka 2 klasteriai. Standartinis nuokrypis tiesiogiai apskaičiuojamas iš tiriamo duomenų rinkinio. Fiksuojamas laiko tarpas, per kurį yra įvykdomas klasterizacijos algoritmas esant skirtingoms K_{init} reikšmėms. Atskirai tiriami atvejai, kai *WINE* duomenų rinkinys yra normalizuotas ir nenormalizuotas.

3.2.2.1. Normalizuotų duomenų klasterizacijos rezultatų priklausomybė nuo inicijuojamų klasterių skaičiaus

Klasterizacijos algoritmo neperskirstytų rezultatų atitikimas ekspertų nuomonei procentais esant skirtingam inicijuojamų klasterių skaičiui pateiktas 24 pav. Aukštas atitikimas ekspertų nuomonei (89,22%) yra tuomet, kai K_{init} reikšmė imama iš intervalo $[22, 30]$. Vidutinis atitikimo vertinimas visame $[2, 30]$ intervale yra lygus 77,17%. Perskirsčius rezultatus gaunamas pastovus rezultatų atitikimas ekspertų nuomonei – 93,82%.



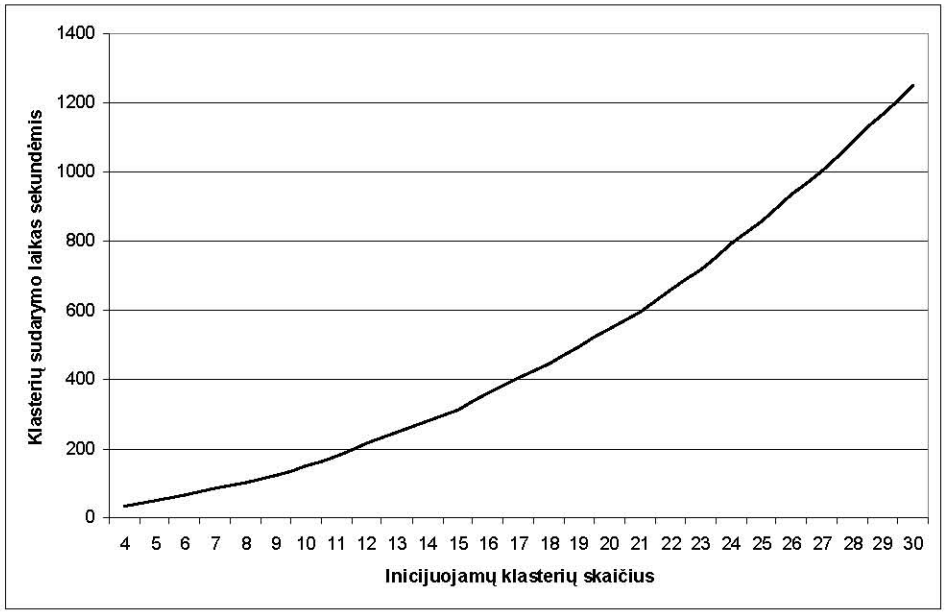
24 pav. ***WINE duomenų klasterizacijos rezultatų atitikimas ekspertų nuomonei pagal parinktą K_{init} reikšmę***



25 pav. ***WINE duomenų klasterizacijos rezultatų gerumo vertinimas pagal parinktą K_{init} reikšmę***

25 pav. pateiktas klasterizacijos algoritmo neperskirstytų rezultatų gerumo vertinimas siluetų mato požiūriu esant skirtingam inicijuojamų klasterių skaičiui. Vidutinis gerumo vertinimas K_{init} reikšmių intervale $[2, 30]$ yra lygus 0,35. Aukštesni vertinimo rezultatai gaunami, kai pradinis klasterių skaičius parenkamas iš intervalo $[22, 30]$ – 0,35. Tačiau algoritmo vykdymo laiko sąnaudos yra didesnės (26 pav.). Atlikus klasterizacijos rezultatų perskirstymą, gaunamas pastovus klasterių gerumo vertinimas siluetų matų požiūriu – 0,53.

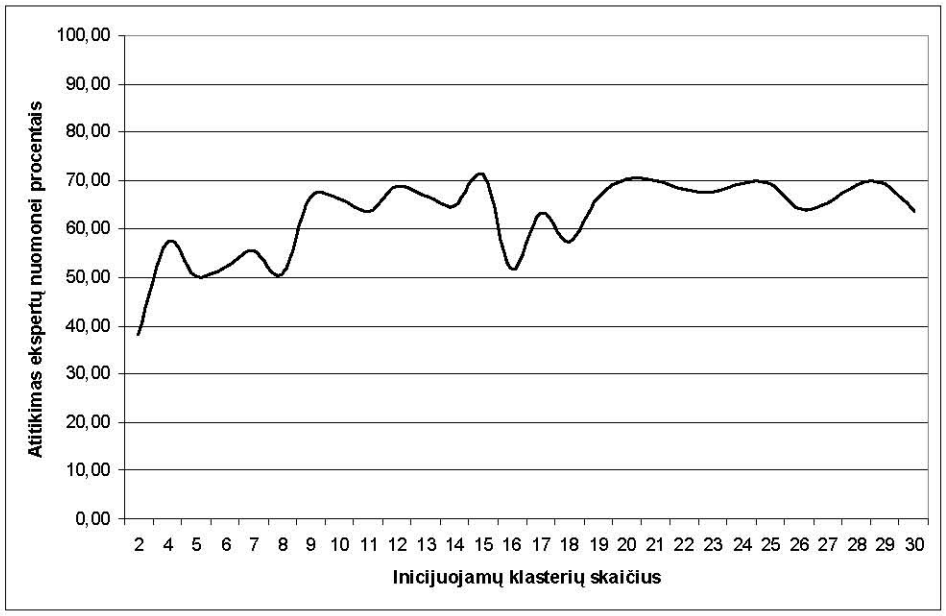
Iš 24, 25 ir 26 pav. galima teigti, kad siekiant optimalaus varianto, reikia inicijuoti apie 13 *WINE* duomenų rinkinio klasterių. Tuomet laiko sąnaudų, atitikimo ekspertų nuomonei ir klasterių gerumo požiūriais gauti rezultatai yra geriau vertinami.



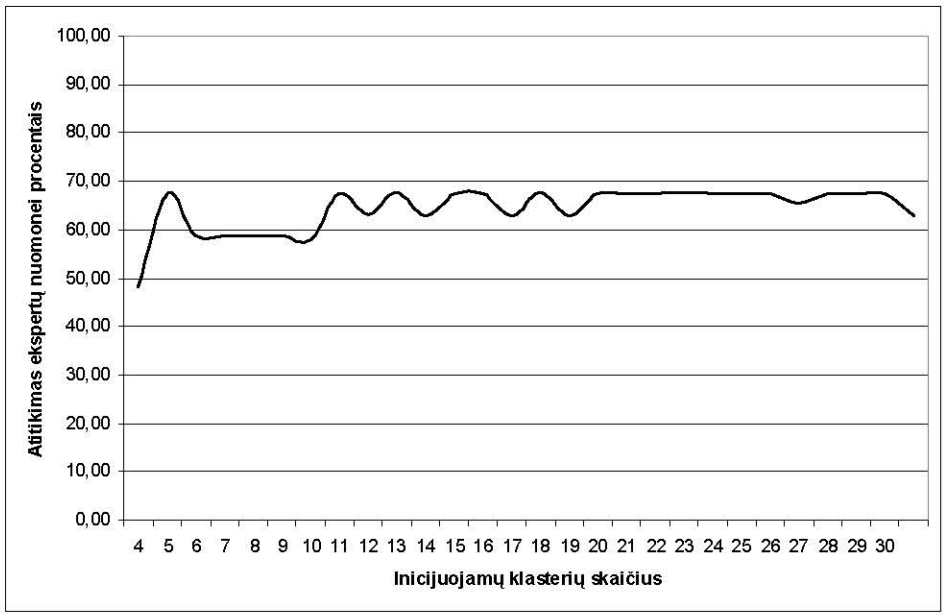
26 pav. ***WINE duomenų klasterių sudarymo laikas pagal parinktą K_{init} reikšmę***

3.2.2.2. Nenormalizuotų duomenų klasterizacijos rezultatų priklausomybė nuo inicijuojamų klasterių skaičiaus

Klasterizacijos algoritmo neperskirstytų rezultatų atitikimas ekspertų nuomonei esant skirtingam inicijuojamų klasterių skaičiui pateiktas 27 pav. Imant K_{init} reikšmes iš intervalo $[2, 30]$, atitikimas ekspertų nuomonei yra lygus 63,51%. Kai pradinis klasterių skaičius patenka į intervalą $[20, 25]$, tai rezultatų atitikimas yra didesnis – 69,19%. Mažinant K_{init} reikšmę, šis įvertinimas yra nepastovus ir mažesnis.

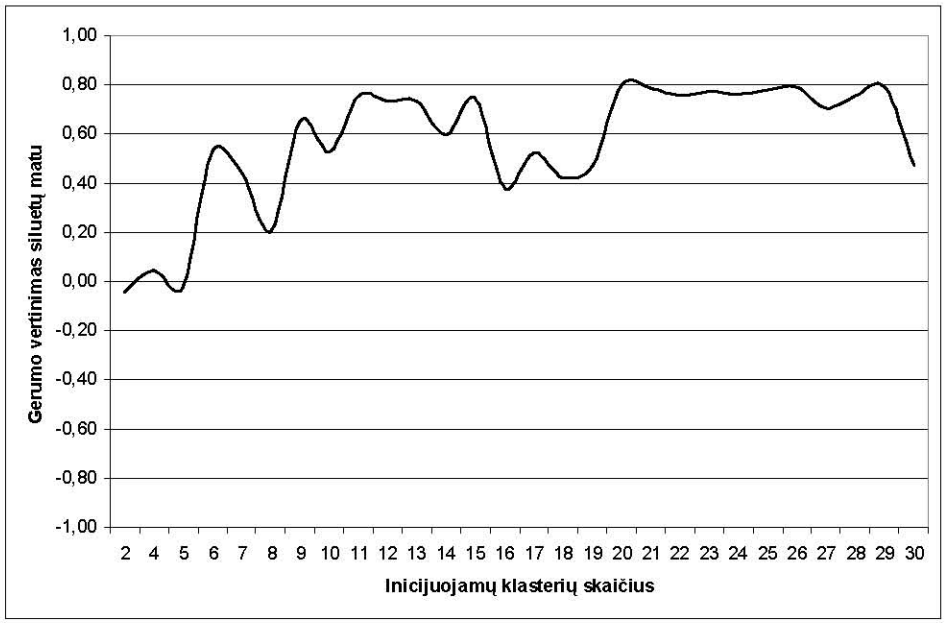


27 pav. ***WINE duomenų klasterizacijos rezultatų atitikimas ekspertų nuomonei pagal parinktą K_{init} reikšmę***



28 pav. ***WINE duomenų perskirstytų klasterizacijos rezultatų atitikimas ekspertų nuomonei pagal parinktą K_{init} reikšmę***

28 pav. pateiktas klasterizacijos algoritmo perskirstytų rezultatų atitikimas ekspertų nuomonei. Vidutinis rezultatų atitikimas esant skirtingoms K_{init} reikšmėms iš intervalo $[2, 30]$ yra lygus 64,66% – aukštesnis nei neperskirstytų rezultatų. Kai pradinis klasterių skaičius patenka į intervalą $[20, 25]$, tai rezultatų atitikimas yra didesnis – 67,45%. Mažinant inicijuojamų klasterių skaičių, gaunamas mažesnis atitikimo įvertinimas.



29 pav. ***WINE duomenų klasterizacijos rezultatų gerumo vertinimas pagal parinktą K_{init} reikšmę***

29 pav. pavaizduotas klasterizacijos algoritmo neperskirstytų rezultatų gerumo vertinimas siluetų mato požiūriu esant skirtingoms K_{init} reikšmėms. Bendras įvertinimas nagrinėjamame K_{init} reikšmių intervale $[2, 30]$ yra lygus 0,57. Aukštesni vertinimo rezultatai yra tuomet, kai pradinis klasterių skaičius parenkamas iš intervalo $[20, 25]$ – 0,78. Atlikus klasterizacijos rezultatų perskirstymą, gaunamas beveik pastovus klasterių gerumo vertinimas siluetų matų požiūriu, kurio vidurkis – 0,85.

Diagrama, vaizduojanti vidutines klasterizacijos algoritmo vykdymo laiko sąnaudas sekundėmis esant skirtingoms K_{init} reikšmėms, kai *WINE* duomenų rinkinys yra nenormalizuotas, yra panaši į diagramą, kai duomenys yra normalizuoti (26 pav.). Mažinant inicijuojamų klasterių skaičių, algoritmo vykdymo laikas taip pat sutrumpėja.

Taigi didinant inicijuojamų klasterių skaičių, sudarytų klasterių vertinimas pagal ekspertų nuomonę ir siluetų matą yra aukštesnis. Tuo pačiu didėja ir algoritmo vykdymo laiko sąnaudos. Perskirsčius rezultatus klasterių objektų perskirstymo modulio pagalba, gaunami pastovesni ir geriau vertinami rezultatai. Tačiau pastarieji įvertinimai mažesni, nei naudojant normalizuotą *WINE* duomenų rinkinį.

3.2.2.3. Inicijuojamų klasterių skaičiaus įtakos klasterizacijos rezultatams apibendrinimas

4 lentelėje pateiktas apibendrintas modifikuoto Renyi-CLIPx konceptualios klasterizacijos algoritmo rezultatų vertinimas esant skirtingam inicijuojamų klasterių skaičiui K_{init}. Čia fiksuojamas vidutinis rezultatų atitikimas ekspertų nuomonei, vidutinis gerumas pagal

siluetų matą. Pateikiamas intervalas, iš kurio parenkant K_{init} reikšmes gaunami geresni rezultatai nei jų nustatyti vidurkiai visame tirtame K_{init} intervale.

4 lentelė. ***Klasterizacijos rezultatų vertinimas esant skirtingoms K_{init} reikšmėms***

Duomenų normalizacija	Rezultatų perskirstymas	Įvertinimai
Taip	Ne	*Vidutinis atitikimas*: 77,17% *Vidutinis gerumas*: 0,35 *Siūlomas intervalas*: $[22, 30]$
Taip	Taip	*Vidutinis atitikimas*: 93,82% *Vidutinis gerumas*: 0,53 *Siūlomas intervalas*: >2
Ne	Ne	*Vidutinis atitikimas*: 63,51% *Vidutinis gerumas*: 0,57 *Siūlomas intervalas*: $[20, 25]$
Ne	Taip	*Vidutinis atitikimas*: 64,66% *Vidutinis gerumas*: 0,85 *Siūlomas intervalas*: $[20, 25]$

Norint per optimalų laiko tarpą sudaryti gerai atskirtus klasterius, reikia normalizuoti nagrinėjamą duomenų rinkinį ir perskirstyti klasterizacijos rezultatus. Rezultatų perskirstymas atliekamas klasterių objektų perskirstymo modulio pagalba. Kadangi algoritmo vykdymo pradžioje inicijuojamų klasterių centrai parenkami atsitiktiniu būdu, todėl klasterizacijos rezultatai gali skirtis.

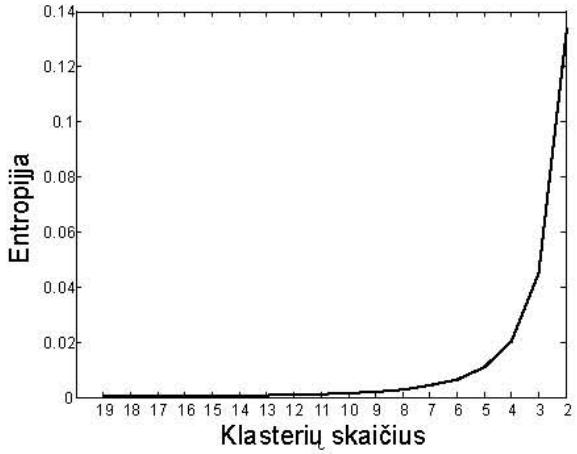
Pastaba *p7*: Iš 20 ir 22 pav., 24 ir 25 pav. matoma, kad klasterizacijos rezultatų atitikimo ekspertų nuomonei ir gerumo pagal siluetų matą vertinimų kreivės turi bendrą kitimo tendenciją, kai *WINE* duomenų rinkinys yra normalizuotas. Todėl daroma prielaida, kad tais atvejais, kai ekspertų nuomonė nėra žinoma, galima stebėti klasterių gerumo kitimą. Pagal šį kriterijų nustatoma, kokioms parametrų reikšmėms esant yra sudaromi kokybiški klasteriai ir kaip jie gerai vertinami.

3.2.3. Diskretizacijos įtaka klasterizacijos taisyklėms

Objektų priskyrimo klasteriams taisykles galima generuoti CLIP3, CLIP4 arba kitais konceptualios klasterizacijos algoritmais. Sugeneruotas taisyklių rinkinys laikomas geru, jei pagal jį visi teigiami egzemplioriai patenka tik į *POS* duomenų rinkinį, o neigiami – į *NEG* rinkinį. Jei tiriami daugiau nei du klasteriai, tai laikoma, kad *POS* rinkinį sudaro klasterio,

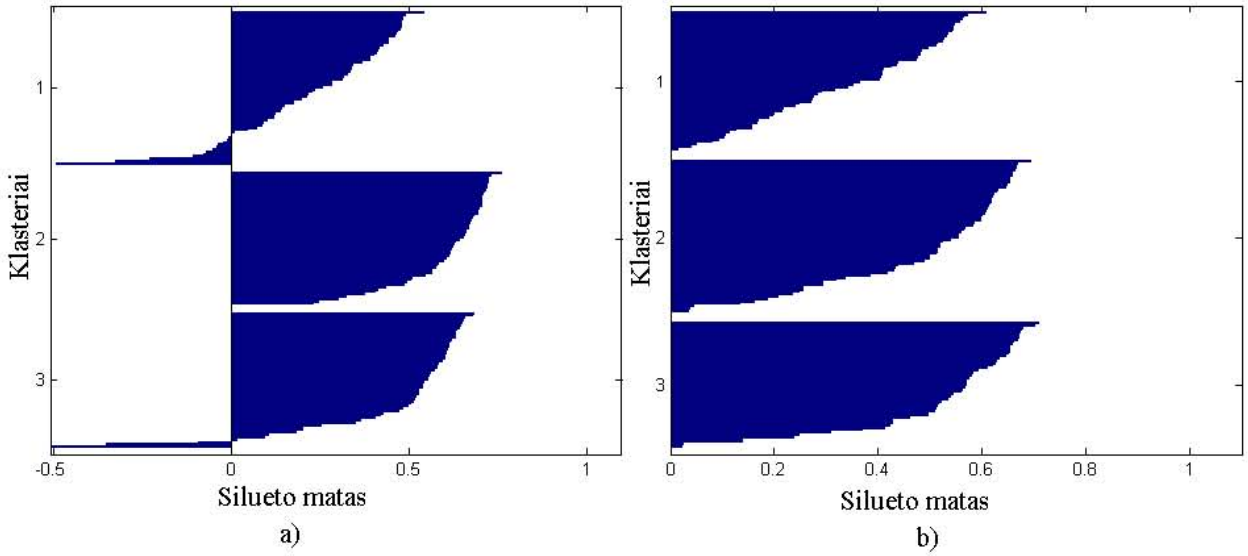
kuriam generuojamos taisyklės, egzemplioriai, o *NEG* – visų kitų klasterių egzemplioriai. Tokiu būdu galima sugeneruoti klasterizacijos taisykles kiekvienam rinkiniui.

Taisyklių generavimui naudojami klasterius, sudaryti iš *WINE* duomenų rinkinio. Šių klasterių sudarymo metu duomenų rinkinys yra normalizuotas. Inicijuojama 20 klasterių, standartinio nuokrypio reikšmė tiesiogiai nustatoma iš tiriamų rinkinių, o gauti rezultatai perskirstomi siluetų matų požiūriu. 30 pav. pateiktas tiriamo duomenų rinkinio tarpklasterinės entropijos pokytis.



*30 pav. **WINE duomenų tarpklasterinės entropijos pokytis***

Iš 30 pav. matoma, kad sumažinus klasterių skaičių iki 2 tarpklasterinės entropijos reikšmė drastiškai padidėja – tikėtina, kad tikrasis pateikto duomenų rinkinio klasterių skaičius yra 3. Atsižvelgiant į tai, kad ekspertų požiūriu *WINE* duomenų rinkinį sudaro 3 klasteriai, tai galima teigti, kad algoritmo rekomenduojamas klasterių skaičius yra tikslus.



*31 pav. **WINE duomenų sudarytų klasterių gerumo vertinimas: a) neperskirstytų, b) perskirstytų***

31 pav. pateiktas trijų sudarytų klasterių gerumo vertinimas siluetų matų požiūriu. Neperskirstytų klasterių atitikimas ekspertų nuomonei yra lygus 87,26%, o perskirstytų – 93,82%. Perskirsčius rezultatus pirmam klasteriui priskirta 60, antram klasteriui 65, o trečiam 53 egzemplioriai. Gautų rezultatų įvertinimai yra geri, todėl jiems generuojame klasterizacijos taisykles. Taisyklės generuojamos pagal sudarytus klasterius ir diskretizuotą duomenų rinkinį, kai diskrečių intervalų skaičius lygus 100 ir yra taikomas genetinio algoritmo modulis.

Klasterizacijos taisykles galima vertinti įvairiais kriterijais. Šiame darbe taisyklės sudėtingumas vertinamas pagal stebimų skirtingų požymių ir loginių sąlygų skaičių. Fiksuojamas skaičius klasterio objektų, kuriuos aprašo sugeneruotos taisyklės. Kadangi objektas įtraukiamas į tiriamą (*POS*) klasterį tik tuomet, kai yra tenkinamos gautos taisyklės, tai neturi būti tokių atvejų, kad pagal tas pačias taisykles į šį klasterį patenka objektai iš kitų (*NEG*) klasterių. Klasteriui sugeneruotas taisyklių rinkinys turi būti unikalus.

Atlikus klasterizacijos taisyklių generavimą modifikuotu Renyi-CLIPx algoritmu, kiekvienam klasteriui yra sudaryta viena taisyklė:

- Pirmajam klasteriui sugeneruota taisyklė nagrinėja 2 skirtingus požymius: 1-mą ir 13-tą. Taisyklės loginių sąlygų skaičius – 166.
- Antrajam klasteriui sugeneruota taisykle nagrinėja 3 skirtingus požymius: 4-tą, 9-tą ir 13-tą. Taisyklės loginių sąlygų skaičius – 143.
- Trečiajam klasteriui sugeneruota taisykle nagrinėja 2 skirtingus požymius: 7-tą ir 10-tą. Taisyklės loginių sąlygų skaičius – 175.

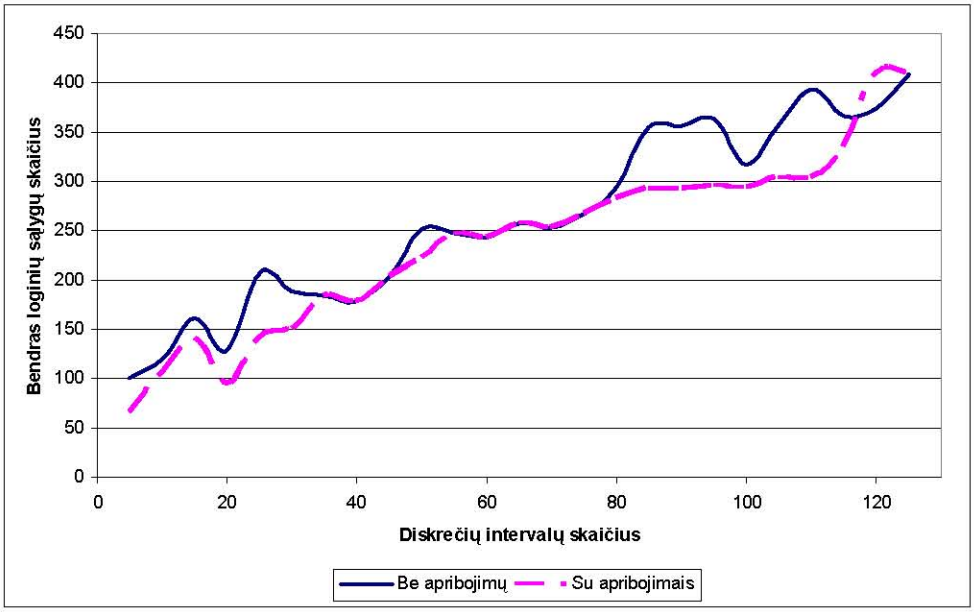
Sudarytų klasterių taisyklės tenkina visus atitinkamų klasterių objektus ir nei vienas objektas iš *NEG* duomenų rinkinio nepatenka į nagrinėjamą *POS* duomenų rinkinį. Visų taisyklių loginių sąlygų skaičius yra didelis (166 + 143 + 175 = 484), o visas taisyklių rinkinys nagrinėja 6 skirtingus požymius-atributus. Loginių sąlygų skaičius priklauso nuo nusakomų atributų reikšmių. Kuo daugiau nagrinėjama atributų reikšmių, tuo sudėtingesnė yra taisyklė. Skirtumas tarp to paties atributo reikšmių gali būti minimalus. Pavyzdžiui, reikšmės 5,64, 5,65 ir 5,68 yra nagrinėjamos atskirai. Sprendžiant šią problemą, yra būtina diskretizuoti visų atributų reikšmes.

Diskretizacijos proceso metu atributų reikšmės suskirstomos į nurodytą intervalų skaičių. Šiame etape tiriama *WINE* duomenims sudarytų taisyklių sudėtingumo, apimamų požymių ir klaidų skaičiaus priklausomybė nuo diskrečių intervalų pasirinkimo. Atskirai nagrinėjami atvejai, kai gautiems rezultatams yra netaikomi ir taikomi šie apribojimai:

- Triukšmo riba, kuri lygi 60%.
- Skaidymo riba, kuri lygi 2.
- Sustojimo riba, kuri lygi 2%.
- Mažiausias loginių sąlygų skaičius, kuris lygus 3.

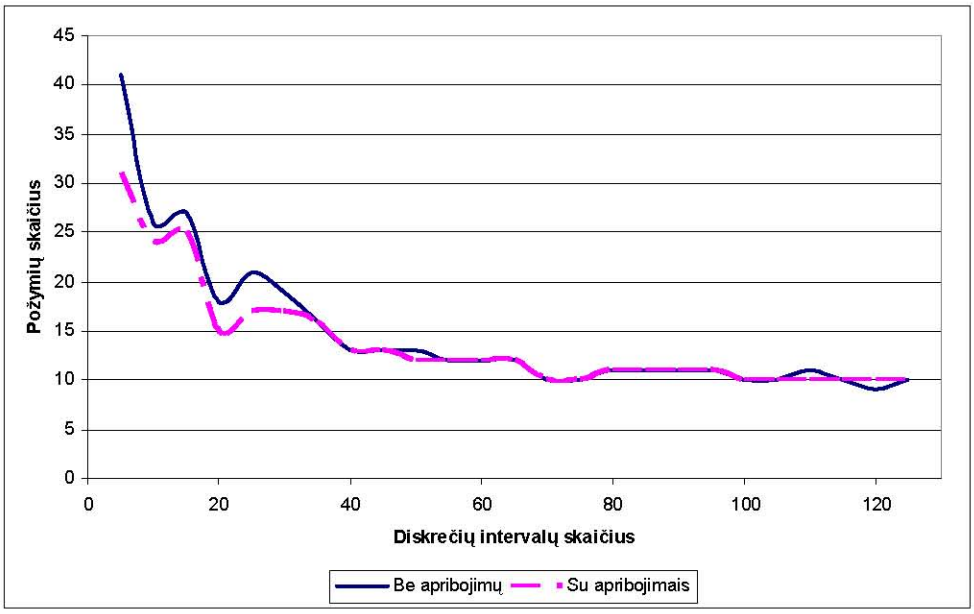
- Mažiausias taisyklės stebimas skirtingų požymių skaičius, kuris lygus 3.
- Minimalus egzempliorių, kuriuos aprašo taisyklė, skaičius. Šiam kriterijui priskirta reikšmė lygi 4.

Šių apribojimų įvedimu siekiama, kad taisyklės nebūtų generuojamos atsitiktiniu būdu mažam egzempliorių skaičiui.



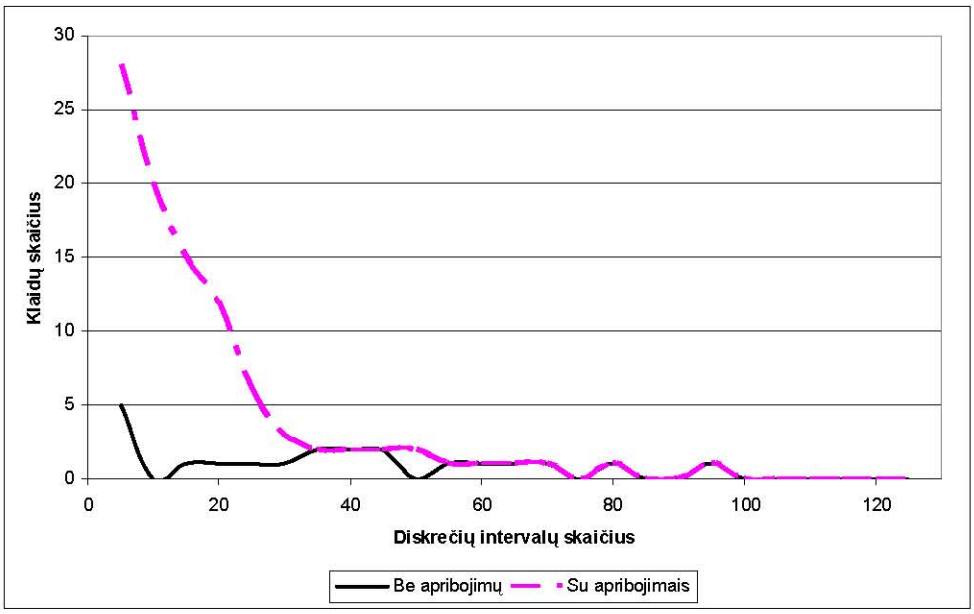
32 pav. ***WINE duomenų taisyklių sudėtingumas, kai rezultatai yra apriboti ir neapriboti***

32 pav. pateiktas bendras klasterizacijos taisyklių vertinimas pagal jų sudėtingumą esant skirtingam diskrečių intervalų skaičiui. Čia sudėtingumas siejamas su taisyklės loginių sąlygų skaičiumi. Didinant diskretizacijos intervalą yra sudaromos vis sudėtingesnės taisyklės. Kai diskrečių intervalų skaičius imamas iš $[90, 110]$, o taisyklių generavimo rezultatai yra apriboti, tai sudėtingumas yra beveik pastovus. Neapribotų rezultatų sudėtingumas daugeliu atvejų yra didesnis nei apribotų.



33 pav. ***WINE duomenų taisyklių bendras požymių skaičius, kai rezultatai yra apriboti ir neapriboti***

33 pav. pateiktas *WINE* duomenų klasterizacijos taisyklių vertinimas pagal nagrinėjamų požymių (atributų) skaičių. Čia požymių skaičius yra bendras visoms sudarytoms taisyklėms – t.y. jei tą patį požymį nagrinėja dvi skirtingos taisyklės, tai laikoma, kad bendras požymių skaičius yra lygus 2. Kuo mažesnį požymių skaičių nagrinėja taisyklė, tuo ji lengviau suprantama ir taikoma. Didinant diskretizacijos intervalų skaičių, nagrinėjamų požymių skaičius sumažėja ir pasidaro pastovus – 10. Pašalinus pasikartojančius požymius, skirtingų požymių skaičius tampa lygus 6. Tokiu būdu galima identifikuoti mažiausią atributų rinkinį, nusakantį kuriam klasteriui reikia priskirti nagrinėjamą objektą.



34 pav. ***WINE duomenų klasterizacijos taisyklių klaidų pasirodymas, kai rezultatai yra apriboti ir neapriboti***

WINE rinkinio objektų priskyrimo klasteriams taisyklių vertinimas pagal klaidų pasirodymą esant skirtingam diskretizacijos intervalui pateiktas 34 pav. Čia klaidomis laikomas skaičius *POS* rinkinio egzempliorių, kurių neaprašo sudarytos taisyklės, ir skaičius atvejų, kai klasterizacijos taisyklės *NEG* duomenų rinkinio objektus įtraukia į *POS* rinkinį. Didinant diskretizacijos intervalą, klaidų pasirodymų skaičius nusistovi ir pasidaro lygus 0. Didžiausias klaidų skaičius yra tuomet, kai klasterizacijos rezultatams taikomi apribojimai, ir parenkamas mažas diskretizacijos intervalas. Tuomet esant mažam diskrečių intervalų skaičiui, sudarytos taisyklės dažnai netenkina įvestų ribojimų.

Atlikus klasterizacijos taisyklių generavimą modifikuotu algoritmu, kai *WINE* duomenų rinkinio diskretizacijos intervalas lygus 100, kiekvienam klasteriui sudaryta viena taisyklė:

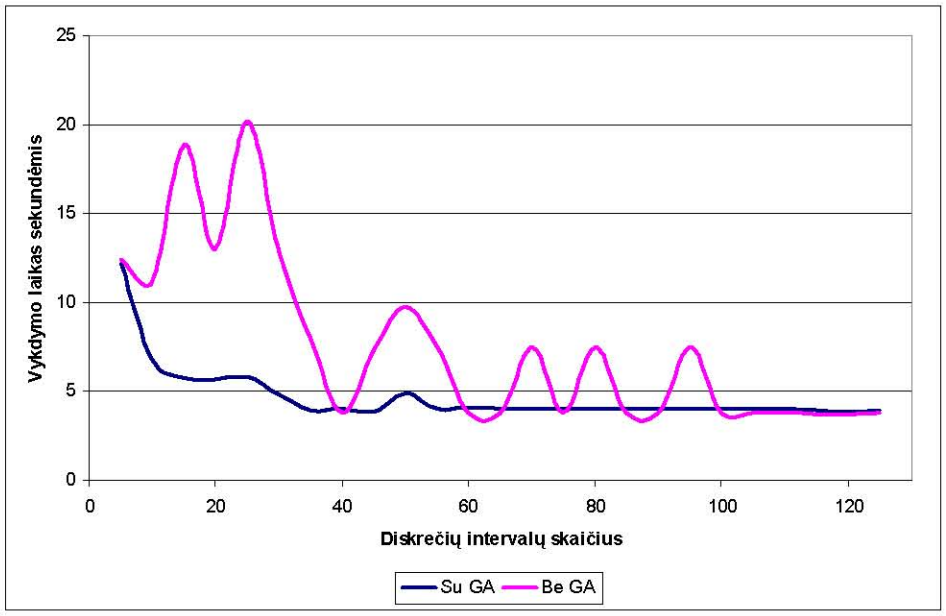
- Pirmajam klasteriui sugeneruota taisyklė nagrinėja 2 skirtingus požymius: 1-mą ir 13-tą. Taisyklės loginių sąlygų skaičius – 99.
- Antrajam klasteriui sugeneruota taisykle nagrinėja 3 skirtingus požymius: 4-tą, 9-tą ir 13-tą. Taisyklės loginių sąlygų skaičius – 93.
- Trečiajam klasteriui sugeneruota taisykle nagrinėja 2 skirtingus požymius: 7-tą ir 10-tą. Taisyklės loginių sąlygų skaičius – 93.

Visų taisyklių loginių sąlygų skaičius yra mažesnis (99 + 93 + 93 = 285), nei nediskretizuotų duomenų.

Taigi diskretizavus pradinius duomenis, sudaromų klasterizacijos taisyklių sudėtingumas vidutiniškai sumažėja nuo 500 iki 300. Taip pat identifikuojamas mažesnis bendras esminių atributų, nusakančių kuriam klasteriui reikia priskirti nagrinėjamą objektą, skaičius. Didinant diskretizacijos intervalą, sumažėja ir klaidų pasirodymų skaičius.

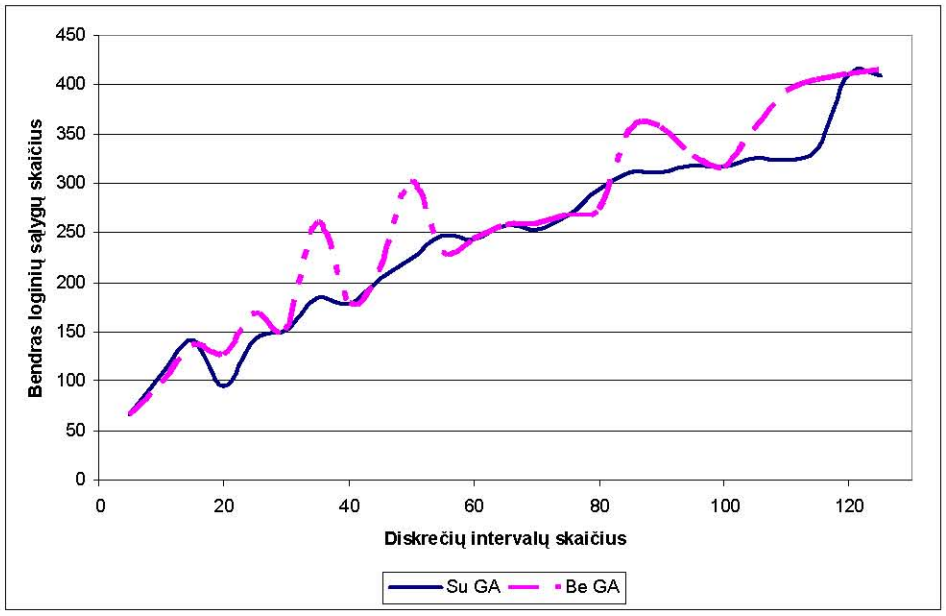
3.2.4. Genetinio algoritmo įtaka klasterizacijos taisyklėms

Klasterizacijos taisyklių sudarymui gali būti taikomi genetinio algoritmo (*GA*) modulio perdengimo ir mutacijos operatoriai. Jų dėka yra formuojami mazgai, kurie duomenų rinkiniui suteikia papildomos informacijos [2]. *GA* modulio taikymas leidžia skirti mažiau kompiuterio atminties ir procesoriaus resursų klasterizacijos algoritmo vykdymui. *GA* naudingas, kai turimas mažas egzempliorių skaičius [11]. Šiame etape tiriama genetinio algoritmo modulio įtaka *WINE* duomenų rinkinio klasterizacijos taisyklėms, kai yra pateiktas skirtingas diskrečių intervalų skaičius. Gautiems rezultatams taikomos prieš tai aprašytos vykdymo ribos. Kiekvienas eksperimentas skirtingam diskretizacijos intervalui atliekamas 10 kartų.



35 pav. ***Taisyklių generavimo WINE duomenims laikas esant skirtingam diskrečių intervalų skaičiui***

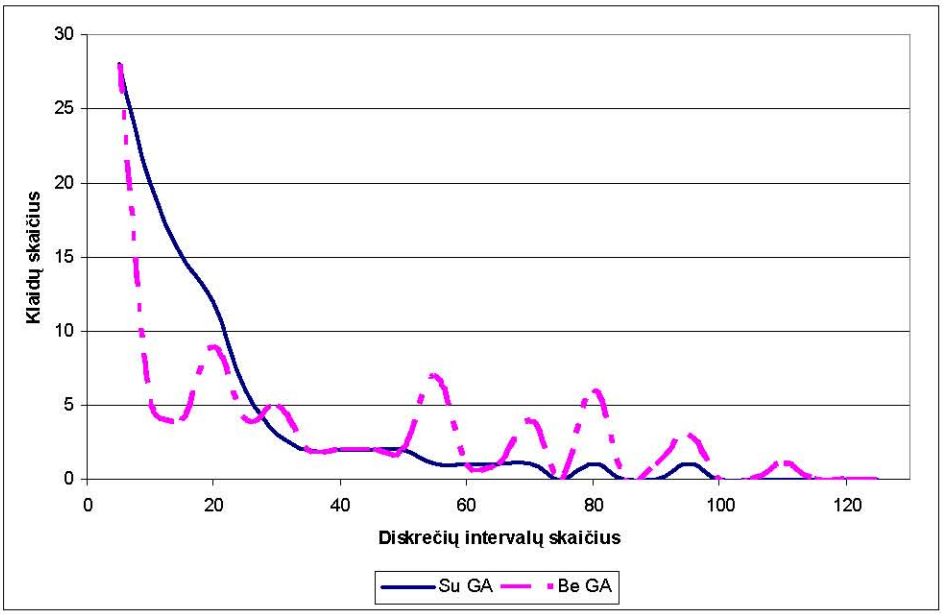
Iš 35 pav. matoma, kad taikant genetinio algoritmo modulį objektų priskyrimo klasteriams taisyklės sudaromos greičiau. Didinant diskretizacijos intervalų skaičių, algoritmo vykdymo laikas tampa pastoviu. Tuo pasižymi klasterizacijos algoritmas, kai yra naudojamas *GA* modulis.



36 pav. ***WINE duomenims sudarytų taisyklių sudėtingumas esant skirtingam diskrečių intervalų skaičiui***

WINE duomenims sudarytų klasterizacijos taisyklių sudėtingumas yra mažesnis, kai taikomas genetinio algoritmo modulis (36 pav.). Čia sudėtingumas suprantamas, kaip visų

taisyklių loginių sąlygų skaičius. Kuo objektų priskyrimo klasteriams taisyklių sudėtingumas mažesnis, tuo jos lengviau taikomos.



37 pav. ***WINE duomenims sudarytų taisyklių klaidų skaičius esant skirtingam diskrečių intervalų skaičiui***

WINE duomenims sugeneruotų klasterizacijos taisyklių padaromas klaidų skaičius yra mažesnis, kai taikomas genetinio algoritmo modulis (37 pav.). Čia klaidomis yra laikomas skaičius atvejų, kai sudarytos taisyklės nenusako tiriamo rinkinio objektų arba nusako *NEG* rinkinio egzempliorius. Jei nėra naudojamas *GA* modulis, tai klaidų pasirodymas yra labiau kintantis. Didinant diskrečių intervalų skaičių, šis pasirodymas tampa lygus 0.

Konceptualios klasterizacijos algoritmo *GA* modulis padeda sudaryti paprastesnes ir geresnes taisyklės. Esant ribotiems skaičiavimų resursams, genetinio algoritmo modulis naudingas, nes jo dėka sugeneruojamos klasterizacijos taisyklės per trumpesnį laiko tarpą [11].

3.2.5. Neigiamo duomenų rinkinio įtaka klasterizacijos taisyklėms

CLIP3 ir CLIP4 konceptualios klasterizacijos algoritmai generuoja taisykles pagal pateiktus *POS* ir *NEG* duomenų rinkinius. Nuo neigiamų egzempliorių išdėstymo tvarkos priklauso sudaromi mazgai bei galutinis rezultatas. Šiame etape tiriama *NEG* duomenų rinkinio egzempliorių užrašymo tvarkos įtaka klasterizacijos taisyklėms.

Klasterizacijos algoritmas kito lygmens mazgus generuoja pagal esamą lygmenį ir nagrinėjamą *NEG* objektą. Šie objektai iš *NEG* rinkinio imami tokia tvarka, kuria jie yra užrašyti. Galima tirti kelis jų atrankos būdus:

- eilės tvarka, kuria buvo pateikti,

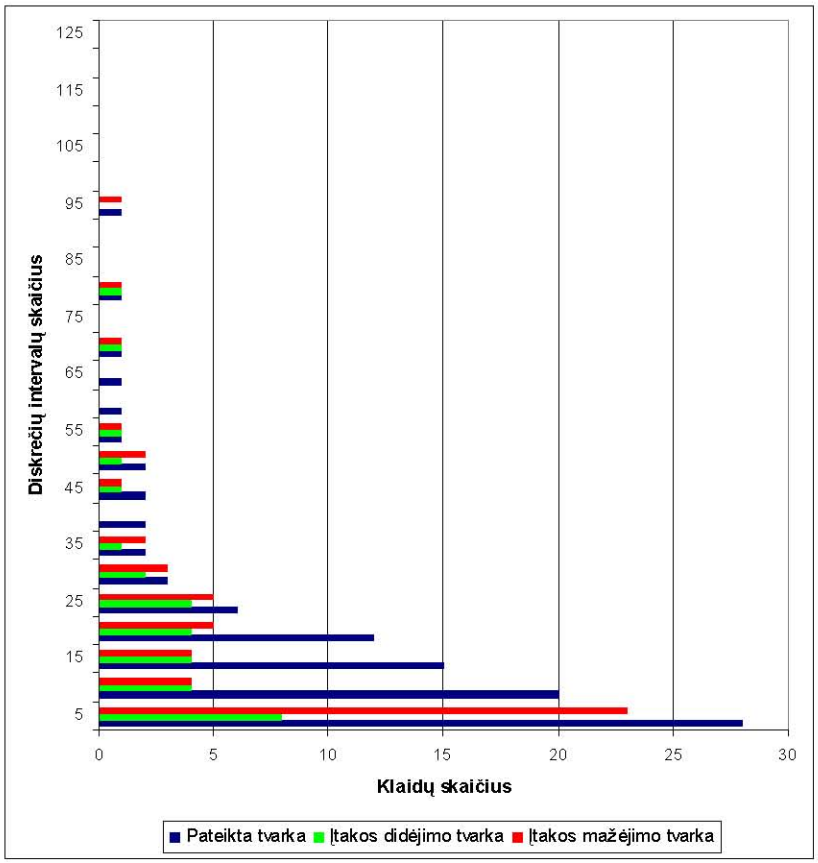
- atsitiktiniu būdu.

Neigiamo objekto išrinkimo eilės tvarka būdas nėra patrauklus, nes tuomet *NEG* duomenų rinkinio formavimo metu yra nulemiami klasterizacijos taisyklių generavimo rezultatai. Esant tam tikrai objektų užrašymo tvarkai gali būti gauti blogi arba geri rezultatai.

Neigiamo objekto išrinkimas atsitiktiniu būdu gali būti gero arba blogo rezultato priežastis. Galima peržiūrėti rezultatus, kurie gauti atsitiktiniu būdu parenkant dar nenagrinėtus neigiamus objektus, ir juos palyginti įvestų kriterijų atžvilgiu. Taip atrenkami geriausi rezultatai. Šis būdas reikalauja daug laiko ir skaičiavimo resursų.

NEG objektų užrašymas sistemingu būdu turėtų leisti išdėstyti taip neigiamus egzempliorius, kad gaunami rezultatai būtų pakankamai geri. Tačiau tam nėra pateiktas sprendimo būdas. Galima nagrinėti šį variantą – išdėstyti *NEG* egzempliorius pagal jų įtaką *POS* duomenų rinkiniui.

CLIP3 ir CLIP4 konceptualios klasterizacijos algoritmai pagrįsti minimizavimo uždavinio sprendimu. Jų metu yra sudaromos dvejetainės matricos. Jeigu tam tikro teigiamo objekto požymio reikšmė skiriasi nuo pasirinkto neigiamo objekto atitinkamos požymio reikšmės, tai į dvejetainę matricą įrašomas 1. Priešingu atveju, jeigu tam tikro teigiamo objekto požymio reikšmė sutampa su neigiamo objekto atitinkamo požymio reikšme, tai į dvejetainę matricą įrašomas 0. Tokiu būdu galima sudaryti dvejetaines matricas kiekvienam neigiamam stebėjimui iš *NEG* duomenų rinkinio. Susumavus šių binarinių matricų vienetų skaičių, gaunamas dydis, kuris nusako neigiamo objekto įtaką tiriamam duomenų rinkiniui. Kuo didesnis pastarasis įvertinimas, tuo mažiau atitinkamų *POS* rinkinio reikšmių nusako neigiamas stebėjimas – jo įtaka yra maža. Pagal šiuos įvertinimus galima palyginti visus *NEG* rinkinio objektus.



38 pav. ***NEG rinkinio užrašymo įtaka esant skirtingam WINE duomenų diskrečių intervalų skaičiui***

38 pav. pateikta *NEG* rinkinio objektų užrašymo tvarkos įtaka klasterizacijos rezultatams, kai yra keičiamas *WINE* duomenų rinkinio diskrečių intervalų skaičius. Nagrinėjami šie atvejai – kai neigiamas egzempliorius parenkamas pateikta eilės tvarka, ir kai neigiami egzemplioriai surūšiuoti mažėjimo bei didėjimo įtakos *POS* rinkinio egzemplioriams tvarka. Mažas klaidų skaičius yra tuomet, kai neigiami egzemplioriai pateikti įtakos didėjimo tvarka. Tuomet pirmiausiai yra sudaromi mazgai pagal tuos *NEG* rinkinio objektus, kurie aprašo mažiausiai teigiamų egzempliorių atributų reikšmių.

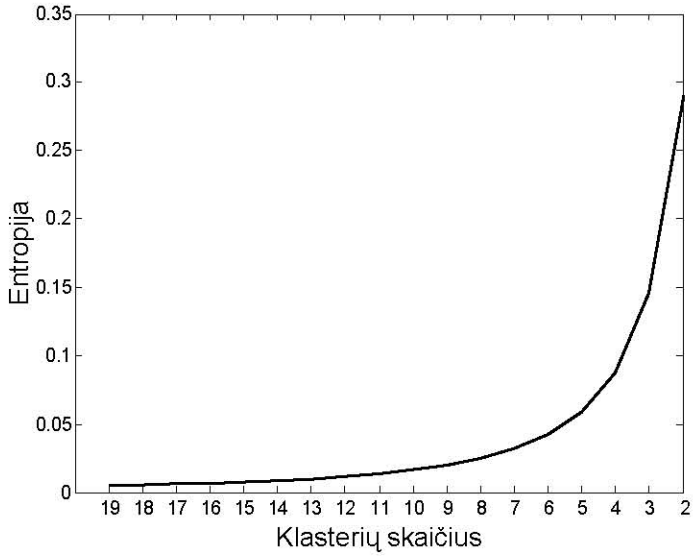
Taigi *NEG* rinkinio egzempliorių užrašymo tvarka daro skirtingą įtaką klasterizacijos taisyklių generavimui. Nustatyta, kad *WINE* duomenims mažiausias klaidų skaičius yra tuomet, kai *NEG* rinkinio objektai yra surūšiuoti įtakos *POS* egzemplioriams didėjimo tvarka. Geri rezultatai gali būti ir tada, kai neigiami egzemplioriai atrenkami atsitiktiniu būdu.

3.2.6. Virtualios mokymo(si) aplinkos duomenų klasterizavimas ir taisyklių generavimas

Pastaruoju metu naudojamos įvairios nuotolinių studijų galimybės. Viena iš jų – virtuali mokymo(si) aplinka. Šioje aplinkoje gali būti kaupiama informacija apie besimokančiųjų pasiekimus. Dažniausiai pastarieji pasiekimai siejami su žinių patikrinimo rezultatais. Pagal

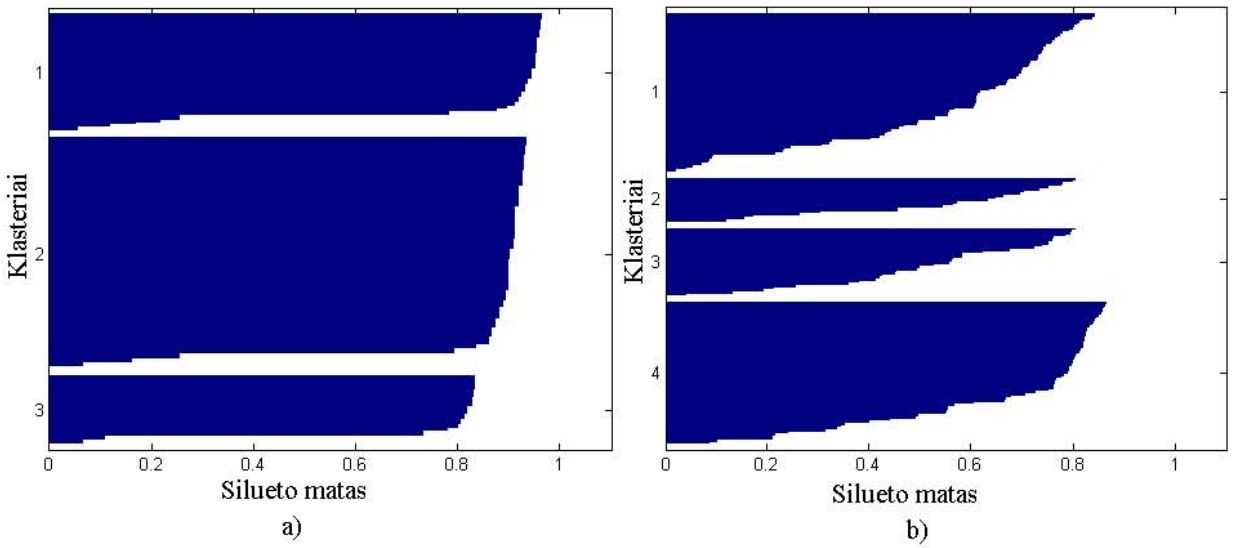
šiuos rezultatus galima sudaryti klasterius ir jiems generuoti taisykles. Tokiu būdu besimokantieji suskirstomi į natūraliai suformuotas grupes, kur kiekvienai grupei taikomos tam tikros taisyklės. Pavyzdžiui, besimokančiųjų klasteriui galima taikyti specializuotą kurso pateikimo būdą.

Šiame etape naudojami *AP* ir *ML* duomenų rinkiniai, kurie sukaupti KU virtualioje mokymo(si) aplinkoje. Jiems sudarome klasterius ir klasterizacijos taisykles modifikuotu Renyi-CLIPx konceptualios klasterizacijos algoritmu.

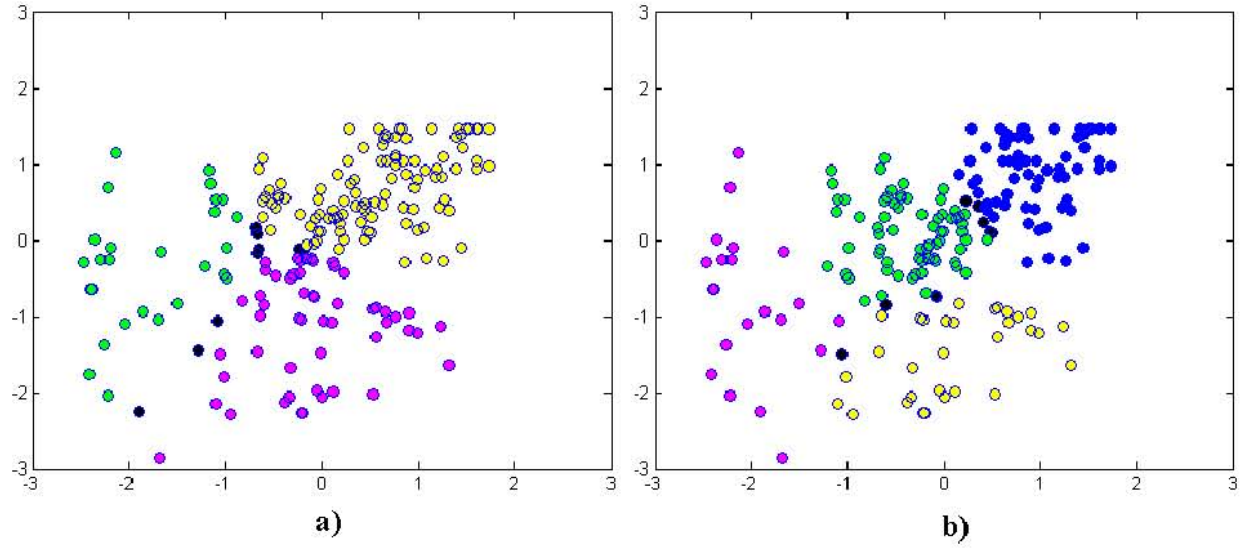


39 pav. ***AP duomenų rinkinio Renyi entropijos pokytis***

39 pav. pateiktas *AP* duomenų rinkinio Renyi entropijos pokytis, kuris gaunamas mažinant klasterių skaičių. Didelis pokytis užfiksuotas, kai klasterių skaičius pakeičiamas iš 3 iki 2. Akivaizdus entropijos pokytis atsiranda, kai sudaromi ir 3 klasteriai. Taigi yra 3 arba 4 natūraliai atskirti klasteriai. Atliekant pateikto rinkinio analizę yra nustatytos ir pašalintos 6 išskirtys ir 2 pasikartojantys objektai.



40 pav. ***AP duomenų rinkinio perskirstytų klasterių gerumas, kai yra suformuoti 3 (a) ir 4 (b) klasteriai***



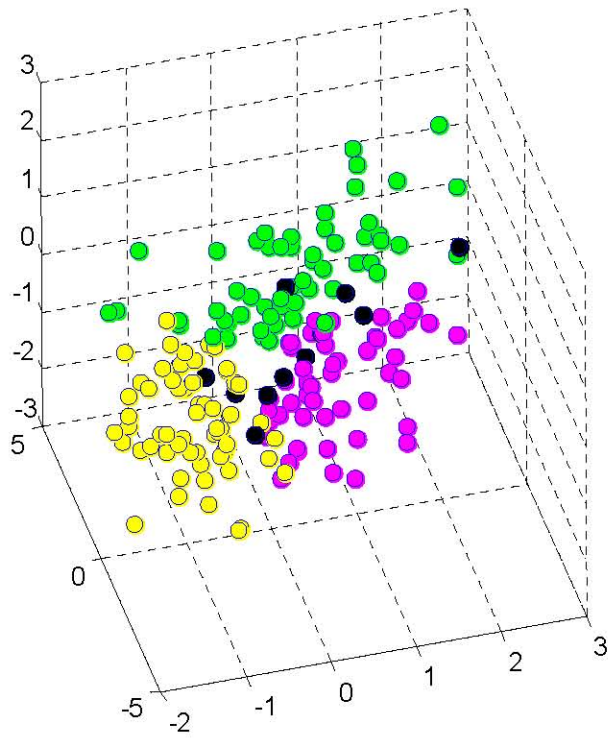
41 pav. ***AP duomenų rinkinio sudarytų klasterių objektų grafinis pavaizdavimas, kai yra suformuoti 3 (a) ir 4 (b) klasteriai***

40 pav. pateiktos 3-jų ir 4-ių sudarytų klasterių siluetų mato diagramos. 3-jų klasterių bendras vertinimas yra aukštesnis nei 4-ių klasterių – 0,77 ir 0,59 atitinkamai. Todėl daroma prielaida, kad tikrasis klasterių skaičius yra 3. Objektų priskyrimas klasteriams vaizdžiai pateiktas 41 pav. Čia matavimų skalės susietos su normalizuotais studentų įvertinimo balais, o skirtingomis spalvomis yra pažymėti objektai, priklausantys skirtingiems klasteriams. Juoda spalva žymimi tie objektai, kurių priskyrimas klasteriams yra abejotinas – jų priskyrimo gerumo vertinimas mažesnis nei 0,1. Pastarieji objektai patenka į ribą, kuri skiria vieną klasterį nuo kito.

Atliekant klasterizacijos taisyklių generavimą, nagrinėjama situacija, kai yra sudaryti 3 klasteriai. Šiuo atveju pirmam klasteriui priskirta 58, antram – 106, o trečiam 30 objektų. Gauti šie rezultatai:

- Pirmajam klasteriui sudarytos 2 taisyklės, kurių kiekviena aprašo 91,10% ir 49.69%, o jos kartu dengia 100% tiriamo klasterio objektų.
- Antrajam klasteriui sudarytos 3 taisyklės, kurios atitinkamai dengia 96,23%, 50,00%, 72,64% egzempliorių. Šios taisyklės kartu aprašo visus tiriamo klasterio objektus.
- Trečiajam klasteriui sudarytos 2 taisyklės, kurios aprašo 96,67% ir 76.67%, o jos kartu tenkina 100% tiriamo klasterio objektų.

Nei viena sudaryta taisyklė neaprašo *NEG* rinkinio objektų taip, kad jie būtų priskirti į tiriamą *POS* rinkinį.



42 pav. ***ML duomenų rinkinio objektų priskyrimas klasteriams***

Duomenų rinkinio *ML* objektus nusako 3 atributų reikšmės. Šio rinkinio objektų priskyrimą klasteriams galima pavaizduoti trimatėje erdvėje. 42 pav. pateiktas tiriamo rinkinio objektų priskyrimas klasteriams. Čia matavimų skalės susietos su normalizuotais studentų įvertinimo balais. Nagrinėjamu atveju yra suformuoti 3 klasteriai, kurių bendras gerumo vertinimas, kai duomenys yra normalizuoti, – 0,79. Akivaizdu, kad objektai gerai priskirti šiems klasteriams. Sudarytos klasterizacijos taisyklės aprašo visus tiriamus objektus.

Taigi modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas sudaro gerai atskirtus klasterius iš virtualios mokymo(si) aplinkos duomenų apie besimokančiųjų pasiekimus ir pateikia taisykles. Remiantis šiomis taisyklėmis naujas objektas priskiriamas į atitinkamą klasterį. Pagal šiuos rezultatus kiekvienai sudarytai besimokančiųjų grupei galima taikyti

specializuotą kurso pateikimo būdą. Tokiu būdu siekiama praplėsti KU virtualios mokymo(si) aplinkos galimybes.

3.2.7. Modifikuoto Renyi-CLIPx klasterizacijos algoritmo programinė realizacija

Renyi-CLIPx algoritmo modulių, Renyi entropijos, CLIP3 ir CLIP4 klasterizacijos algoritmų praktinė realizacija atlikta taikant MATLAB 7.1 (R14 SP3) matematinį paketą. MATLAB (iš žodžių MATrix LABoratory) yra daugiaplatformė MathWorks programinė įranga, skirta įvairių mokslo šakų problemoms spręsti, ypač matematinėms. Ji turi puikias galimybes manipuliacijoms su matricomis. Šis paketas turi lengvai perprantamą programavimo kalbą [20]. MATLAB 7 programinės įrangos licencijų yra įsigijusi Klaipėdos universiteto Informatikos katedra.

Klasterių sudarymas ir objektų priskyrimo klasteriams taisyklių generavimas modifikuotu Renyi-CLIPx klasterizacijos algoritmu atliekamas žemiau pateiktais etapais.

1. Duomenų rinkinio paruošimas klasterių ir taisyklių sudarymui.

   Tiriamas duomenis rinkinys išsaugomas tekstiniame dokumente (praplėtimas **.txt*). Vykdymo byloje *renyi_data.m* nurodome nagrinėjamo duomenų rinkinio dokumento pavadinimą.

   renyi_data byloje gali būti parenkami šie veiksmai:

   - pasikartojančių eilučių pašalinimas,
   - sąlyginių išskirčių ir išskirčių pašalinimas,
   - duomenų rinkinio normalizavimas.

2. Klasterių sudarymas.

   Pateikto duomenų rinkinio klasteriai sudaromi komandų eilutėje įvykdžius komandą *renyi*.

   Vykdymo byloje *renyi.m* galima keisti šiuos nustatymus:

   - inicijuojamų klasterių skaičių K_{init},
   - inicijuojamų klasterių elementų skaičių N_{init}.

3. Klasterių objektų perskirstymas.

   Klasterių objektų perskirstymas atliekamas komandų eilutėje įvedus komandą *renyi_info(X)*. Čia X lygus skaičiui klasterių, kurių objektus perskirstome.

4. Taisyklių generavimas.

   Taisyklių generavimas atliekamas komandų eilutėje įvedus komandą *clip3_main* arba *clip4_main*.

Vykdymo byloje *clip4_main.m* (*clip3_main.m*) galima keisti diskretizacijos intervalų skaičių.

Vykdymo byloje *clip4.m* (*clip3.m*) galima keisti šiuos nustatymus:

- triukšmo ribą,
- geriausios taisyklės ribą,
- sustojimo ribą,
- mažiausią loginių sąlygų skaičių,
- mažiausias taisyklės stebimą skirtingų požymių skaičių,
- genetinio algoritmo vykdymo aktyvavimą,
- neigiamų stebėjimų užrašymo tvarką.

Įvykdžius šias komandas, modifikuotas Renyi-CLIPx klasterizacijos algoritmas sudaro perskirstus klasterius ir suformuoja klasterizacijos taisykles. Taip pat pateikiami grafikai, rodantys tarpklasterinę entropiją, sudarytų klasterių gerumo vertinimą pagal siluetų matą ir objektų priskyrimą klasteriams.

4. IŠVADOS

1. Atlikus Renyi entropijos, ITERATE, CLIP3 ir CLIP4 klasterizacijos algoritmų analizę, pasiūlytas modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas. Jis pateikia vieningą kompleksą žingsnių, kuris be papildomų priemonių įvedimo sudaro klasterius ir objektų priskyrimo klasteriams taisykles.
2. Siekiant užtikrinti aukštą gaunamų klasterių atitikimą ekspertų nuomonei, rekomenduojama atlikti šiuos veiksmus:
    - normalizuoti duomenų rinkinį,
    - standartinį nuokrypį apskaičiuoti iš nagrinėjamų objektų,
    - inicijuoti didelį pradinių klasterių skaičių,
    - perskirstyti klasterizacijos rezultatus pagal siluetų matų vertinimą.

    Įvykdžius šiuos veiksmus, modifikuoto Renyi-CLIPx klasterizacijos algoritmo sudaromų klasterių vidutinis atitikimas ekspertų nuomonei yra 93,82%, kai tiriamas tarptautinėje praktikoje naudojamas *WINE* duomenų rinkinys.
3. Modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas pagerina generuojamas taisykles ir loginių sąlygų skaičių tirtame *WINE* duomenų rinkinyje leidžia sumažinti vidutiniškai nuo 500 iki 300, t.y. daugiau kaip 40%, jeigu:
    - diskretizuojamas duomenų rinkinys,
    - taikomi apribojimai,
    - naudojamas genetinio algoritmo modulis,
    - neigiamų stebėjimų duomenų rinkinio objektai yra surūšiuoti įtakos teigiamiems stebėjimams didėjimo tvarka.
4. Modifikuotu Renyi-CLIPx konceptualios klasterizacijos algoritmu galima sudaryti gerai atskirstus klasterius ir objektų priskyrimo klasteriams taisykles virtualios mokymo(si) ir kitokios prigimties duomenų rinkiniams. Normalizuotų *AP* ir *ML* duomenų vertinimas siluetų matų požiūriu yra 0,77 ir 0,79 atitinkamai, o sudarytos taisyklės aprašo visus objektus.

5. SANTRUMPŲ ŽODYNĖLIS

1. ADO – didžiausių skirtumų rūšiavimo algoritmas.
2. AI VMA – Adaptyvi Intelektuali Virtuali Mokymo(si) Aplinka.
3. AP – Asemblerinio programavimo ir valdymo kurso duomenų rinkinys.
4. BIN – binarinė matrica.
5. CLIP – sutr. iš angl. k. termino: *Cover Learning using Integer Linear Programming*.
6. CM – patrikrintoji matrica (angl. k. *Checked Matrix*).
7. CU – klasės kategorijos naudingumas.
8. DX – atsitiktinio dydžio X dispersija.
9. EX – atsitiktinio dydžio X vidurkis.
10. GA – genetinis algoritmas.
11. IP – sutr. iš angl. k. termino: *Integer Liner Programming*.
12. IQR – kvartilių skirtumas.
13. ML – Matematinės logikos kurso duomenų rinkinys.
14. NEG – neigiamų stebėjimų duomenų rinkinys.
15. POS – teigiamų stebėjimų duomenų rinkinys.
16. PS – skirsnio rezultatas (angl. k. *Partition Score*).
17. TM – šabloninė matrica (angl. k. *Template Matrix*).
18. VMA – Virtuali Mokymo(si) Aplinka.

6. LITERATŪRA

1. Abidi S., Ong J. A Data Minning Strategy for Inductive Data Clustering. *IEEE International Symposium on Information Theory*. 2000, volume 2, p. 568-573.
2. Aghagolzadeh M., Zadeh H., Araabi B., Aghagolzadeh A. A hierarchical clustering based on mutual information maximization. *Image Processing, 2007. ICIP 2007. IEEE International Conference*. 2007, p. 277-283.
3. Antos A., Kontoyiannis I. Estimating the entropy of discrete distributions. *IEEE International Symposium on Information Theory*. 2001, p. 34-38.
4. Baziukaitė D. (2007) *Besimokančiojo atžvilgiu virtualios mokymo(si) aplinkos gebėjimus gerinantys metodai*, Daktaro disertacija, Vytauto Didžiojo Universitetas, Matematikos ir informatikos institutas (Disertacija rengta 2002 m. - 2007 m. Klaipėdos universitete), Kaunas, p. 112. Fiziniai mokslai, informatika (09P).
5. Baziukaitė D. Approach to an Adaptive and Intelligent Learning Environment. *Advances in Computer, Information, and Systems Sciences, and Engineering, Proceedings of IETA 2005*. TeNe 2005 and EIAE 2005, Elleithy, K.; Sobh, T.; Mahmood, A.; Iskander, M.; Karim, M. (Eds.) 2006, XV, ISBN 1-1020-5260-X, Springer, p. 399-406 [Springer].
6. Baziukaitė D. Making Virtual Learning Environment more intelligent: the problem of software agent's mental state recognition. *Liet. matem. rink.* 2005, T. 45, p. 459-464.
7. Baziukaitė D., Narščius A., Rimkus E. Vienodo pločio, dažnio ir K-vidurkių diskretizacijos algoritmų lyginamasis tyrimas. *Informacinės technologijos 2007*. 2007, Kaunas: Technologija; p. 443-446, ISSN 1822-6337.
8. Bercher J., Vignat C. A Renyi entropy convolution inequality with application. *IEEE International Symposium on Information Theory*. 2002, p. 263 -268.
9. Biswas G., Member S., Fisher J. ITERATE: A Conceptual Clustering Algorithm for Data Mining, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*. 1998, volume 28, p. 219-230.
10. Biswas G., Weinberg J., Li C. ITERATE: A Conceptual Clustering Method for Knowledge Discovery in Databases. *Innovative Applications of Artificial Intelligence in the Oil and Gas Industry*. 1995, 32-51.
11. Cios K., Kurgan A. CLIP4: Hybrid inductive machine learning algorithm that generates inequality rules. *Information Sciences*. 2004, volume 163, issues 1-3, p. 37-83.
12. Cios K., Pedrycz W., Swiniarski R. Data Mining Methods for Knowledge Discovery. *Neural Networks, IEEE Transactions*. 1998, volume 9, issue 6, p. 1533-1539.
13. Cios K., Wedding D., Liu N. CLIP3: cover learning using integer programming. *Kybernetes: The International Journal of Systems & Cybernetics*. 1997, volume 26, number 5, p. 513-536(24).
14. Čekanavičius V., Murauskas G. *Statistika ir jos taikymai I*, Vilnius: TEV, 2003. ISBN 9986-546-93-1.
15. Gokcay E., Principe J. A new clustering evaluation function using Renyi's information potential. *Acoustics, Speech, and Signal Processing, 2000. 2000 IEEE International Conference*. 2000, Volume 6, p. 3490-3497.
16. Gokcay E., Principle J. Information theoretic clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions*. 2002, Volume 24, Issue 2, p. 158-171.
17. Jenssen R., Erdogmus D., Principe J. Clustering using Renyi's Entropy. *Neural Networks, 2003. Proceedings of the International Joint Conference*. 2003, volume 1, p. 523-528.
18. Kurgan A., Cios K. Ensemble of Classifiers to Improve Accuracy of the CLIP4 Machine Learning Algorithm. *Proc. of the SPIE International Conference on Sensor Fusion: Architectures, Algorithms, and Applications VI*. 2002, p. 32-42.

19. Mirkin B. *Clustering for Data Mining: A data recovery approach*. London: Chapman & Hall/CRC, 2005 m. ISBN 1-58488-534-3.
20. Moler C. The Origins of MATLAB [interaktyvus]. 2007. http://www.mathworks.com/company/newsletters/news_notes/clevescorner/dec04.html [žiūrėta 2008-05-14].
21. Narščius A. Modifikuotas Renyi-CLIPx konceptualios klasterizacijos algoritmas. *Fundamentiniai tyrimai ir inovacijos mokslų sandūroje: Respublikinės jaunųjų mokslininkų konferencijos darbai*. 2008, Klaipėda: KU leidykla; p. 4-11, ISBN 978-9955-18-329-7.
22. Narščius A., Golouchova J. Renyi, Iterate ir CLIP3 klasterizacijos algoritmų lyginamoji apžvalga. *Lietuvos pajūrio aplinkos tyrimai, planavimas ir tvarkymas*. 2007, Klaipėda: KU leidykla; p. 19-25, ISBN 978-9955-18-221-4.
23. Narščius A., Golouchova J., Baziukaitė D., Ramašauskas O. Konceptualios klasterizacijos ir PEST analizės taikymas tyrimui apie Lietuvos gyventojų naudojimąsi viešojo administravimo institucijų teikiamomis elektroninėmis paslaugomis. *Technologijos mokslo darbai vakarų Lietuvoje XI*. 2008, Klaipėda: KU leidykla; p. 49-54, ISSN 1822-4652.