KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Dovilė Rudokaitė-Margelevičienė

Dirichlet mišinių statistika paremto klasifikavimo metodo kūrimas ir tyrimas

Magistro darbas

Darbo vadovas
prof. H. Pranevičius

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Dovilė Rudokaitė-Margelevičienė

Dirichlet mišinių statistika paremto klasifikavimo metodo kūrimas ir tyrimas

Magistro darbas

Kalbos konsultantė
Lietuvių k. katedros
doc. J. Mikelionienė
2006-05

Vadovas
prof. H. Pranevičius
2006-05

Recenzentas
prof. J. Mockus
2006-05

Atliko
IFM-0/2 gr. stud.
Dovilė Rudokaitė-Margelevičienė
2006-05-15

Kaunas, 2006

SANTRAUKA

Sukurtų klasifikavimo metodų ir algoritmų yra daug, tačiau susidomėjimas klasifikavimo metodais nei kiek nemažėja. Didėjant duomenų ir informacijos kiekiams, jų įvairovei, iškyla reikmė duomenis patikimai apdoroti. Nuolat pasireiškia svarstymai, kuris metodas labiausiai tiktų duomenims klasifikuoti, - kuris jų tiksliausiai ir patikimai klasifikuotų duomenis. Šiame darbe pasiūlomas klasifikavimo metodas, paremtas Dirichlet mišinių statistika. Dirichlet mišiniai apjungia daugiau nei vieną skirstinį, kurie apibrėžiami ta pačia parametrų aibe, tačiau su skirtingomis jų reikšmėmis. Toks mišinys tampa jautrus atpažįstant įvairiai pasiskirsčiusius dydžius, todėl Dirichlet mišinio panaudojimas klasifikavime gali būti lanksti priemonė bet kokio tipo duomenims klasifikuoti.

Šiame darbe pasiūlomas metodas, kaip Dirichlet mišiniai gali būti panaudojami bet kokio tipo duomenims klasifikuoti. Atsižvelgiant į tai, projektuojamas Dirichlet mišinių klasifikatorius, kuris patikimai ir tiksliai klasifikuotų įvairaus tipo ir reikšmių pasiskirstymo duomenis. Suprojektuotas klasifikatorius klasifikuoja tiek skaitinės informacijos duomenis, tiek simbolinės. Dirichlet mišinių klasifikatorius realizuotas dvejopai: sukurtas galutinis programinis produktas vartotojui duomenims klasifikuoti ir įvairia forma analizuoti klasifikavimo rezultatus. Taip pat klasifikatoriaus realizacija apima programinės bibliotekos sukūrimą, kuri gali būti naudinga programų sistemų kūrėjams vystant didelės apimties dirbtinio intelekto aplinkas. Dirichlet mišinių parametrams optimizuoti įdiegti trys optimizavimo metodai: du išgaubto optimizavimo metodai ir genetinis algoritmas.

Sukurtas Dirichlet mišinių klasifikatorius testuojamas su trimis skirtingomis realių duomenų bazėmis: medicinine, biologine ir fizikine – ir lyginamas su kitais dviem klasifikavimo metodais. Visų klasifikatorių našumai vertinami keliais kokybiniais našumą apsprendžiančiais matais ir analizuojamas našumų skirtumų reikšmingumas. Visoms trims eksperimentuose naudotoms duomenų bazėms Dirichlet mišinių klasifikatorius pasirodė veikiantis našiausiai.

SUMMARY

Development and analysis of a Dirichlet mixture-based classifier

There exist many data classification methods and algorithms; however the importance of them has not diminished. The data and information quantities increase as well as the diversity of information, so the question is how to reliably process the data. Various considerations emerge concerning what method to choose or which of them fits for data best, i.e. which of them would classify data most accurately and reliably. This work presents a classification method based on the statistics of Dirichlet mixtures. Dirichlet mixture combines more than one Dirichlet densities that are described by the same set of parameters but with different values of them. Such a Dirichlet mixture becomes sensitive to recognize differently distributed variables (data) and hence, utilization of the Dirichlet mixtures in classification can provide a powerful tool for classification of data of any kind.

This thesis proposes a method describing how Dirichlet mixtures can be utilized for classification of data of any kind. With regard to this, a Dirichlet mixtures classifier is designed to classify data of any type and with any range of values. The designed classifier classifies numerical data as well as symbolic ones. The Dirichlet mixtures classifier is implemented in two ways: The first one concerns the classifier as the end-product for a user and the second one relates to a compiled library of classification routines. Using the Dirichlet mixtures classifier as the product, the user can classify data and to analyze classification results in the most appropriate form. A software system developer can use the compiled library of Dirichlet mixtures classification routines to develop large machine learning environments. To train Dirichlet mixtures classifier, three optimization methods have been installed in the computational kernel of the classifier: two convex optimization methods and genetic algorithm.

The developed Dirichlet mixtures classifier has been tested on three different real databases: medical, biological, and physical. The classifier is compared to other two classification methods. The performance of all the classification methods is assessed by several qualitative criteria and the significance of the performance differences is estimated. For all three databases used in the testing process, the Dirichlet mixtures classifier outperformed the other two classification methods.

TURINYS

PAVEIKSLŲ SĄRAŠAS

LENTELIŲ SĄRAŠAS

ALGORITMŲ SĄRAŠAS

1. ĮVADAS

1.1. Įvadas

„Klasifikavimas, įvairių stebėjimų išskyrimas ir įvardinimas, yra viena iš bendriausių kultūrinių žmogaus veiklų; jis sudaro pagrindą mūsų mokslui ir civilizacijai“[1] (Hampel, 2002: 5) [1]. Šiuos žodžius, turinčius taip pat filosofinę prasmę, parašė žymus statistikos profesorius Frankas Hampelis. Bendruoju atveju klasifikavimą galima apibrėžti kaip įvairių duomenų grupavimą pagal jų panašumą, remiantis tvarkinga susijusių jiems būdingų kategorijų aibe. Realiame pasaulyje duomenų įvairovė labai plati, jų kiekiai, kuriuos reikia apdoroti norint gauti reikšmingą rezultatą, skiriasi. Duomenų išskirtinumas ir kintamumas, aplinkybių ir įvairių veiksnių įvertinimo būtinumas priverčia sudaryti specialias ekspertines sistemas, o duomenis apibūdinančioms kategorijoms suteikti svorius. Visa tai apsunkina žmogaus pastangas tiesiogiai stebėti duomenų kaitą, juos interpretuoti, patvirtinti hipotezes, ar jas atmesti. Todėl klasifikavimo pagrindinis privalumas ir tikslas – supaprastinti realaus pasaulio vaizdavimą, išvesti taisykles, kuriomis vadovaujantis duomenys galėtų būti priskiriami kategorijoms. Kitaip sakant, duomenų klasifikavimas siūlo aiškesnį supratimą apie duomenis.

Kad stebėjimų duomenys būtų tinkami ir paruošti klasifikavimui, jie priskiriami reikšmingoms kategorijoms. Be tikslaus ir sisteminio duomenų kategorijų nustatymo pagal bendrąsias savybes, taikomi duomenims statistiniai modeliai negali užtikrinti tarpusavyje palyginamų ir patikimų rezultatų. Matematinių, statistinių metodų pagrindu sukurti modeliai formaliai aprašo duomenų pasiskirstymą pagal bendrąsias savybes, o tai reiškia, kad tokie modeliai gali būti taikomi klasifikuojant duomenis, kurių kategorija iš anksto nėra žinoma. Tačiau kad klasifikavimas būtų pavykęs, duomenys ir jų klasifikavimas turi atitikti keletą esminių reikalavimų, pavyzdžiui, stebėjimų duomenys turėtų būti priskiriami tik vienai kategorijai, kategorijų aibė turi būti išbaigta – duomenys turėtų būti priskiriami vienai iš kategorijų, klasifikavimas turi turėti konceptualų pagrindą ir loginę struktūrą.

Klasifikavimo tikslumas labai priklauso nuo sudaryto klasifikuojamiems duomenims matematinio modelio, arba kitaip – nuo to, kaip modeliu aprašomas duomenų pasiskirstymas atitinka realaus pasaulio stebimų duomenų pasiskirstymą. Metodų, kurių pagrindu kuriami modeliai yra daug, todėl erdvės kurti modeliais pagrįstas klasifikavimo architektūras, pritaikytas konkrečiai probleminei sričiai, plataus taikymo ar universalias, atsiranda taip pat daug.

[1] “Classification, the separation and naming of appearances, is one of the most basic cultural activities of humanity; it is a fundament for our science and civilization”

Vienas iš neseniai pasiūlytų klasifikavimo metodų [2], paremtas grafų teorija, duomenis priskiria kategorijoms pagal dinamiškai sudarytą grafą. Klasifikatorius pritaikytas vaizdų atpažinimo pagal skaitmeninius duomenis uždaviniuose ir lyginamas su atraminių vektorių metodo (AVM) klasifikatoriumi [3]. Kitame straipsnyje [4] pristatomas Gausso daugiadimensinių skirstinių kombinacija paremtas klasifikatorius, kurio parametrai optimizuojami matematinės vilties maksimizavimo (EM) algoritmu [5]. Gausso modeliu paremtas klasifikatorius taikytas magnetinio rezonanso vaizdavimo (MRV) srityje MRV duomenis klasifikuoti ir klasterizuoti (klasterinė analizė). AVM klasifikatoriai paskutiniu metu tapo ypač populiarūs bioinformatikos srityje. Įvairių architektūrų AVM klasifikatoriai [6-9] naudojami baltymų klasifikavimui į jų klases ir poklases (*folds*), be to, tam, kad klasifikavimas taptų našesnis, AVM konfigūruojami su neuroniniais tinklais [10], dinaminio programavimo algoritmais [11] ir kitais metodais. Išvardinti metodai atskleidžia tik mažą jų panaudojimo klasifikuojant realaus pasaulio duomenis galimybių dalį ir iliustruoja tik kelis praktikoje pritaikytų klasifikatorių pavyzdžius.

Šiame darbe kuriamas toks klasifikavimo metodas ir atliekami tyrimai su juo, kuris paremtas Dirichlet mišinių pasiskirstymo skaičiavimais. Matematinėje statistikoje Dirichlet mišiniai svarbūs, kur duomenų dažnuminėms charakteristikoms nusakyti naudojami multinominiai pasiskirstymai. Dirichlet mišiniai ir mišiniai, bendrai paėmus, išsiskiria tuo, jog gali būti pritaikyti klasifikavime duomenų, kurių atributai (savybės) gali būti logiškai grupuojamos sudarant sudėtingas duomenų struktūras, ar ypatingus derinius [12]. Pavyzdžiui, sudėtinio baltymų sekų išlygiavimo [13] amino rūgščių pasiskirstymas gali atskleisti įvairius biologiškai svarbius ir specifinius rajonus. Naudojant sudėtinių sekų analizės rezultate įvertintą *posteriorinio* pasiskirstymo dėsnį, galima modeliuoti biologiškai svarbius evoliucinius procesus [14, 15].

Šiame darbe pristatomas metodas gali būti naudojamas bet kokio tipo duomenims klasifikuoti. Nors Dirichlet mišinių metodas naudotas kai kuriems uždaviniams (pavyzdžiui, baltymų panašumų nustatymo) spręsti [16], išvesti bendrieji Bayeso mišinių modeliai [17, 18], Dirichlet mišinių statistiniais skaičiavimais paremtas metodas bet kokio tipo duomenų klasifikavimui, mūsų žiniomis, nebuvo sukurtas.

1.2. Darbo tikslai ir užduotys

Pagrindinis darbo tikslas – sukurti Dirichlet mišinių statistika paremtą klasifikavimo metodą ir atlikti šio metodo našumo tyrimą su pasirinktomis realiomis duomenų bazėmis. Siekiant šio tikslo, reikia atlikti tokias užduotis:

- Išanalizuoti pasirinktų, praktikoje patvirtintų klasifikavimo metodų teorinį pagrindimą ir jų ypatybes klasifikuoti duomenis.
- Išanalizuoti, kokie kokybiniai kriterijai naudojami klasifikavimo našumui vertinti. Apibrėžti aibę svarbiausių našumo vertinimo kriterijų, kurių atžvilgiu skirtingi klasifikavimo metodai bus tarpusavyje lyginami šiame darbe.
- Ištirti Dirichlet mišinio pasiskirstymo dėsnio ypatumus ir įvertinti jo panaudojimo galimybes duomenims klasifikuoti.
- Suprojektuoti universalų Dirichlet mišinių klasifikatorių, galintį interpretuoti bet kokio tipo duomenis ir juos klasifikuoti.
- Sukurti Dirichlet mišinių klasifikatoriaus programinį produktą: nepriklausomą, su grafine vartotojo aplinka klasifikavimo įrankį ir lanksčią (pernešamą Unix šeimos ir Windows operacijų sistemose) programinę biblioteką.
- Ištirti sukurto Dirichlet mišinių klasifikatoriaus našumą, testuojant jį su įvairaus tipo duomenų bazėmis. Palyginti Dirichlet mišinių klasifikatoriaus našumą su kitų klasifikavimo metodų našumais toms pačioms duomenų imtims. Įvertinti klasifikavimo metodų našumų skirtumų reikšmingumą.

1.3. Darbo struktūra

Šio darbo struktūra tokia: antrame skyriuje apžvelgiami du klasifikavimo metodai, su kuriais lyginamas sukurtas Dirichlet mišinių klasifikatorius. Trečiame skyriuje apibrėžiami kokybiniai klasifikavimo našumo matai, kuriais naudojantis vertinami sukurto ir lyginamų klasifikavimo metodų našumai. Ketvirtame skyriuje pristatoma Dirichlet mišinių statistika ir parodoma, kaip Dirichlet mišiniai gali būti naudojami duomenims klasifikuoti. Penktame skyriuje išdėstomi svarbiausi Dirichlet mišinių klasifikatoriaus projektavimo, algoritmizavimo ir realizavimo aspektai. Šeštame skyriuje atliekami tyrimai su trimis skirtingomis duomenų bazėmis: medicinine, biologine ir fizikine. Kiekvienu eksperimento atveju, Dirichlet mišinių klasifikatoriaus našumas lyginamas su kitų dviejų metodų našumu ir išskiriami našumų reikšmingi skirtumai. Septintame skyriuje pateikiamos bendrosios, šį darbą apibūdinančios išvados.

2. KLASIFIKAVIMO METODAI

Klasifikavimo metodų egzistuoja labai daug. Kas mėnesį, ar kelis, leidžiami žurnalai, skirti automatizuoto apmokymo algoritmams ir metodams, klasifikavimo metodams, žinių kaupimo metodologijoms ir metodams. Kas mėnesį pasirodo įvairioms tyrinėjimų sritims skirti žurnalai, kuriuose straipsniai daugiau ar mažiau nagrinėja įvairius klasifikavimo metodų ir jų tobulinimo aspektus. Todėl viename darbe apžvelgti visus egzistuojančius klasifikavimo metodus yra sunkiai įmanoma, o be to ir mažai prasminga.

Šiame darbe apžvelgsime du klasifikavimo metodus: griežtosiomis aibėmis paremtą klasifikavimą ir naivaus Bayeso metodą. Kodėl šie du metodai? Šiame darbe pristatomą Dirichlet mišinių statistika paremtą klasifikavimo metodą realizavome kaip nepriklausomą įrankį ir kaip programinę biblioteką. Programinę biblioteką realizavome taip, kad ją būtų galima nesunkiai pritaikyti gerai žinomame žinių kaupimo ir apdorojimo programų pakete ROSETTA [19]. Tai didelis duomenų apdorojimo paketas, kuris viename iš etapų naudoja griežtosiomis aibėmis paremtą klasifikavimą (šiek tiek plačiau apie paketą bus šnekama projektavimo dalyje). Klasifikavimas griežtosiomis aibėmis daugiausia pasiteisino medicinos ir biomedicinos srityse ir pradedama jį taikyti daugelyje klasifikavimo sričių. Dirichlet mišinių klasifikatoriaus palyginimas su klasifikavimu, paremtu griežtosiomis aibėmis, medicininiams ir kito pobūdžio duomenims atskleistų Dirichlet mišinių, kaip klasifikavimo metodo, galimybes varžytis su pažangiais klasifikavimo metodais. Šiame darbe mes lyginome Dirichlet mišinių klasifikatorių su klasifikavimu griežtųjų aibių teorijos kontekste ir su naivaus Bayeso metodu, kuris, nors ir nesudėtingas, kaip dažnai pažymima, pasižymi geru klasifikavimo našumu.

2.1. Griežtosios aibės

Griežtosios aibės skaičiavimais ir sąvokomis kartais gali priminti neapibrėžtąsias aibes (*fuzzy sets*) ir kitas aibių, ar jomis besiremiančias, teorijas, vis tiktai tai nepriklausomai 1982 m. išvesta ir pasiūlyta teorija, kurios pradininkas Zdzisław Pawlak. Griežtosios aibės – tai teorija, išvesta duomenims analizuoti, jiems aproksimuoti ir generuoti tikslias duomenis klasifikuojančias taisykles.

Nuo griežtųjų aibių sąvokos neatskiriama *informacinės sistemos* sąvoką. Informacinę sistemą sudaro netuščios ir baigtinės aibės: objektų aibė **U** ir atributų aibė **A**. Kiekvienas objektas $\mathbf{x} \in \mathbf{U}$ charakterizuojamas visų atributų $a \in \mathbf{A}$ reikšmėmis, todėl atributo reikšmę galima įsivaizduoti kaip funkciją, kuri duotam objektui grąžina reikšmę iš galimų to atributo reikšmių aibės $\mathbf{V}_a$ ($a : \mathbf{U} \to \mathbf{V}_a$). Informacinė sistema praplėsta specialiuoju atributu d vadinama *sprendimo sistema*: $\mathbf{S} = (\mathbf{U}, \mathbf{A} \cup \{d\})$. Specialusis atributas d vadinamas sprendimo

atributu, kuris nusako duotam objektui $\mathbf{x} \in U$ sprendimo kategoriją, kuriai objektas priklauso (1 pav.). Sprendimo sistemos naudojamos dvejopais tikslais: sugeneruoti griežtąsias aibes (tiksliau, klasifikuojančias taisykles) ir sudarytam griežtųjų aibių modeliui testuoti. Informacinė sistema netalpina informacijos apie objektų priklausomumą sprendimo kategorijoms, ir tokia sistema naudojama praktikoje po to, kai sudarytos griežtosios aibės ir klasifikuojančios taisyklės.

$\mathbf{x}_1$	$a_1(\mathbf{x}_1)$		$a_{\lvert A\rvert}(\mathbf{x}_1)$	$d(\mathbf{x}_1)$
$\mathbf{x}_2$	$a_1(\mathbf{x}_2)$	.	$a_{\lvert A\rvert}(\mathbf{x}_2)$	$d(\mathbf{x}_2)$
		. . .		
$\mathbf{x}_{\lvert U\rvert}$	$a_1(\mathbf{x}_{\lvert U\rvert})$	.	$a_{\lvert A\rvert}(\mathbf{x}_{\lvert U\rvert})$	$d(\mathbf{x}_{\lvert U\rvert})$

1 pav. Informacinės (sprendimo) sistemos pavyzdys. Kiekvienam objektui $x_i \in U$ (i = 1,...,|U|) žinomos atributų $a_j \in A$ (j = 1,...,|A|) reikšmės apsprendžia sprendimo kategoriją $d(x_i)$, kuriai objektas x_i priklauso. Žymėjimai |U| ir |A| žymi aibių dydžius.

Griežtosios aibės [20, 21] apibrėžia dviejų objektų neatskiriamumo (*indiscernibility*) sąryšį, kuris duotam atributų poaibiui $\mathbf{B} \subseteq \mathbf{A}$ grąžina aibes objektų iš **U**, kurie tarpusavyje negali būti atskiriami naudojantis atributais iš aibės **B**. Neatskiriamumo sąryšiu

$$IND_S(\mathbf{B}) = \left\{ \left\{ \mathbf{x} \in \mathbf{X} \middle| (\forall a \in \mathbf{B}) \wedge (\forall \mathbf{y} \in \mathbf{X}) : a(\mathbf{x}) = a(\mathbf{y}) \right\} \middle| \mathbf{X} \subseteq \mathbf{U} \right\} \subset \mathbb{P}\ \mathbf{U},$$

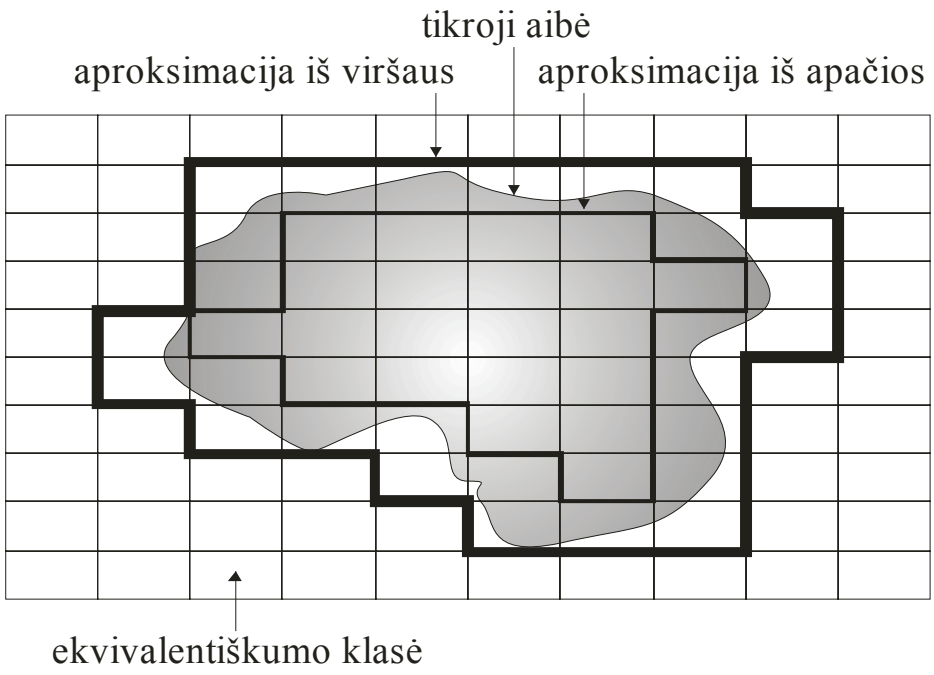
objektai sugrupuojami į taip vadinamas *ekvivalentiškumo klases*. Pavyzdžiui, 2 paveikslėlyje pavaizduotai informacinei sistemai neatskiriamumo sąryšis atributui a_1 lygus $IND_{S1}(\{a_1\}) = \{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_6\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5, \mathbf{x}_7\}\}$, o dviejų atributų aibei $\{a_1, a_2\}$ lygus:$IND_{S1}(\{a_1, a_2\}) = \{\{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5, \mathbf{x}_7\}, \{\mathbf{x}_6\}\}$.

	a_1	a_2	d
$\mathbf{x}_1$	1	3	2
$\mathbf{x}_2$	1	0	2
$\mathbf{x}_3$	3	1	1
$\mathbf{x}_4$	3	1	3
$\mathbf{x}_5$	4	2	1
$\mathbf{x}_6$	1	2	3
$\mathbf{x}_7$	4	2	3

2 pav. Išplėstinės informacinės sistemos S1 su konkrečiomis reikšmėmis pavyzdys.

Apdorojant duomenis, svarbiausia sudaryti objektų, priklausančių vienai sprendimo kategorijai, aibę, kadangi tik tokiu būdu galima pradėti klasifikatoriaus (klasifikavimo

metodo) apmokymą. Tačiau praktikoje gali pasitaikyti, kad objektai iš tos pačios ekvivalentiškumo klasės priklauso skirtingoms sprendimo kategorijoms. Tai reiškia, kad objektų atributų reikšmės vienodos, o sprendimo atributo reikšmė skiriasi, - tai sudaro dviprasmiškumus. Pavyzdžiui (2 pav.), objektų $\mathbf{x}_3$, $\boldsymbol{x}_4$ atributų a_1 ir a_2 reikšmės vienodos, tačiau skiriasi sprendimo kategorija d. Nors tokiais atvejais pagal sprendimo kategoriją tiksliai apibrėžti objektų aibės neįmanoma, įmanoma sudaryti aibę objektų iš pasirinktos sprendimo kategorijos, aibę objektų, kurie nepapuola į pasirinktą sprendimo kategoriją, ir aibę objektų, kurių sprendimo kategorija yra dviprasmiška. Tai yra aproksimuoti objektų aibę iš apačios, iš viršaus ir nustatyti ribojamą šių dviejų aproksimacijų ribą.



3 pav. Objektų aibės aproksimacijos grafinė iliustracija. Kiekvienas langelis žymi ekvivalentiškumo klasę, kuri talpina vieną, ar daugiau objektų. Fone pažymėta teritorija žymi tikrąją objektų aibę, kuriai atliekama aproksimacija. Aproksimacija iš apačios apims visus objektus, kurie sudaro ekvivalentiškumo klases ir ieškomų objektų poaibį. Aproksimacija iš viršaus apims taip pat ir tuos objektus, kurie pasiskirsto tarp kelių ekvivalentiškumo klasių, bet bent vienas iš tų klasių objektas yra iš ieškomų objektų aibės.

Formaliai aproksimacijos užrašomos taip. Jei ekvivalentiškumo klases $[\mathbf{x}]_{\mathrm{B}}$ apibrėšime objektų aibėmis, gautomis taikant neatskiriamumo sąryšį atributų poaibiui $\mathbf{B} \subseteq \mathbf{A}$, t.y. $[\mathbf{x}]_{\mathrm{B}} = \{\mathbf{y} \in \mathbf{X} | \mathbf{X} \subseteq \mathbf{U}, \mathbf{x} \in \mathbf{X}\} \in IND_{\mathrm{S}}(\mathbf{B})$, tuomet apatinės ir viršutinės aproksimacijos sąryšiai objektų poaibiui $\mathbf{X} \subseteq \mathbf{U}$ (kuris paprastai sudaromas pagal sprendimo kategoriją) atitinkamai apibrėžiami:

$$\underline{\mathrm{B}}\,\mathbf{X} = \{\mathbf{x} | [\mathbf{x}]_{\mathrm{B}} \subseteq \mathbf{X}\},$$

$$\overline{\mathrm{B}}\,\mathbf{X} = \left\{\mathbf{x} \middle| [\mathbf{x}]_{\mathrm{B}} \cap \mathbf{X} \neq \varnothing\right\}.$$

Šių dviejų aproksimacijų skirtumas vadinamas aproksimacijų riba: $BN_{\mathrm{B}}(\mathbf{X}) = \overline{\mathrm{B}}\,\mathbf{X} - \underline{\mathrm{B}}\,\mathbf{X}$. Jei aproksimacijų riba tuščia (dviprasmiškumų nėra), objektų aibė **X** vadinama tikslia. Priešingu atveju, objektų aibė **X** vadinama griežtąja. Iš čia kilęs griežtųjų aibių pavadinimas.

Aproksimacijų grafinė iliustracija pateikta 3 paveikslėlyje, o 2 paveikslėlio pavyzdžio objektų aibei $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ turėsime tokias aproksimacijas pasirinktam atributui a_1: $\underline{a_1}\,\mathbf{X} = \varnothing$, $\overline{a_1}\,\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_6\}$.

Iš kiekvienos ekvivalentiškumo klasės užtenka saugoti po vieną objektą, kadangi jų atributų reikšmės sutampa. Panašiai galima sumažinti atributų aibę: šalinti tuos atributus, kurių pašalinimas niekaip nepaveiks neatskiriamumo sąryšio ir objektų aibėms nustatytų aproksimacijų. Minimali sumažinta atributų aibė vadinama redukcijomis (*reducts*). Kitaip sakant, duotai informacinei sistemai **I** = (**U**, **A**) redukcija yra tokia minimali atributų aibė $\mathbf{B} \subseteq \mathbf{A}$, kad $IND_{\mathrm{I}}(\mathbf{B}) = IND_{\mathrm{I}}(\mathbf{A})$. Paprastai galima sudaryti kelias redukcijas, tačiau jų sudarymas yra NP-sudėtingumo uždavinys. Vis tik šiuo metu yra sukurta eilė euristinių algoritmų [19, 22-24], kurie gana tiksliai išsprendžia šį uždavinį per tenkinamą laiko tarpą.

Su redukcijomis neatsiejamai susijusios *atskiriamumo* (*discernibility*) matricų ir funkcijų sąvokos. Atskiriamumo matrica – tai kvadratinė $|\mathbf{U}|\times|\mathbf{U}|$ matrica, kurios elementas $\mathbf{m}_{\mathrm{xy}}$ atitinka atributų aibę, kuriais atskiriami du objektai **x** ir **y**:

$$\mathbf{m}_{\mathrm{xy}} = \left\{a \in \mathbf{A} \middle| a(\mathbf{x}) \neq a(\mathbf{y})\right\},\ i, j = 1,\ldots,|\mathbf{U}|.$$

Atskiriamumo funkcija $f_{\mathrm{I}}(\mathbf{x})$ objektui **x** išreiškiama atributų kombinacija, kuriomis objektas **x** gali būti atskiriamas (pagal atributus) nuo kitų objektų iš jų aibės **U**:

$$f_{\mathrm{I}}(\mathbf{x}) = \bigwedge_{\mathbf{y}\in\mathbf{U}} \left\{\bigvee a \middle| a \in \mathbf{m}_{\mathrm{xy}} \wedge \mathbf{m}_{\mathrm{xy}} \neq \varnothing\right\}.$$

Atskiriamumo funkcijos $f_{\mathrm{I}}(\mathbf{x})$ pirminė implikantė parodys minimalų atributų kiekį, kuriais objektas **x** gali būti atskiriamas nuo visų kitų objektų $\mathbf{y}\in\mathbf{U}$. Atskiriamumo funkcija 2 paveikslėlio pavyzdžio $\mathbf{x}_1$ objektui: $f_{\mathrm{I}}(\mathbf{x}_1) = (a_2)\wedge(a_1\vee a_2)\wedge(a_1\vee a_2)\wedge(a_1\vee a_2)\wedge(a_2)\wedge(a_1\vee a_2) \equiv (a_2)\wedge(a_1\vee a_2) \equiv a_2$. Taigi, šio pavyzdžio pirminė implikantė lygi a_2, o tai reiškia, kad objektas $\mathbf{x}_1$ nuo kitų objektų gali būti atskiriamas vieninteliu atributu a_2.

Visiems objektams $\mathbf{x}\in\mathbf{U}$ informacinėje sistemoje **I** atskiriamumo funkcija apibrėžiama tokia logine kombinacija:

$$g_{\mathrm{I}}(\mathbf{U}) = \bigwedge_{\mathbf{x}\in\mathbf{U}} f_{\mathrm{I}}(\mathbf{x}).$$

Funkcijos $g_{\mathrm{I}}(\mathbf{U})$ pirminė implikantė atskleis minimalų atributų kiekį, kuriais gali būti atskiriami visi objektai iš aibės **U**. Kitaip sakant, funkcijos $g_{\mathrm{I}}(\mathbf{U})$ pirminė implikantė atitiks informacinės sistemos **I** redukciją.

Griežtųjų aibių teorijos tikslas – sukurti tiksliai objektus klasifikuojančias taisykles. Klasifikavimo taisyklės generuojamos po to, kai nustatomos ekvivalentiškumo klasės ir sudaromos redukcijos. Taisyklės apima atributų ir jų reikšmių logine kombinacija, kurios implikuoja konkretų sprendimą, atitinkantį sprendimo kategoriją.

Jei duotai sprendimo sistemai $\mathbf{S} = (\mathbf{U}, \mathbf{A} \cup \{d\})$, kiekvienam objektui **x** gautas redukcijas žymėsime $\mathbf{B}(\mathbf{x}) \subseteq \mathbf{A}$, o atributų ir sprendimo kategorijos reikšmes objektams pažymėsime $a(\mathbf{x})$ ir $d(\mathbf{x})$ atitinkamai, tuomet taisyklių aibė objektui **x**:

$$R_{\mathrm{S}}(\mathbf{x}) = \left\{ \left(\bigwedge_{a\in\mathbf{B}(\mathbf{x})} (a = a(\mathbf{x})) \right) \Rightarrow (d = d(\mathbf{x})) \right\}.$$

Pavyzdžiui (2 pav.), objektui $\mathbf{x}_1$ redukcijų aibė $\mathbf{B}(\mathbf{x}) = \{a_2\}$, todėl taisyklių aibė $R_{\mathrm{S}}(\mathbf{x}_1) = \{(a_2{=}3) \Rightarrow (d{=}2)\}$, sakanti, kad, jei objekto atributo a_2 reikšmė lygi 3, sprendimo kategorija šiam objektui bus lygi 2. Taisyklių aibė visai sprendimo sistemai apims visiems objektams sudarytas taisykles:

$$\mathbf{R}_{\mathrm{S}} = \bigcup_{\mathbf{x}\in\mathbf{U}} R_{\mathrm{S}}(\mathbf{x}).$$

Dažnai realūs duomenys gaunami su triukšmais, ir dažno klasifikatoriaus uždavinys yra kaip įmanoma labiau atskirti ir nufiltruoti triukšmus. Griežtosios aibės šiam tikslui įveda *dinaminių redukcijų* sąvoką, kuri reiškia, jog redukcijos sudaromos dinamiškai ir apytikslės, t.y. „beveik“ tenkinančios neatskiriamumo sąryšį. Taip pat griežtosios aibės nagrinėja įvairius duomenų apdorojimo aspektus, tokius kaip atributų reikšmių diskretizavimas. Taikomi įvairūs algoritmai (Bulio algebros, entropijos, tolygiais skaičiavimais paremti algoritmai ir metodai [19]) atributų reikšmėms diskretizuoti, kad išvengti didelio reikšmių kintamumo ir sugeneruoti bendresnę, pasižyminčią gera diskriminuojama galia ir klasifikavimo našumu, taisyklių aibę. Griežtųjų aibių teorija apima eilę paruošiamųjų ir duomenis apdorojančių etapų, todėl ši teorija siejama ne tik su duomenų klasifikavimu, bet su žinių kaupimu ir atradimu (*knowledge discovery*) aplamai.

2.2. Naivaus Bayeso metodas

Tai paprastas klasifikavimo metodas, kuris, kaip dažnai pažymima, veikia gerai [25, 26]. Jei laikysimės 2.1 skyrelio žymėjimų ir tarsime, kad klasifikuojami objektai **x** su atributų reikšmėmis $a(\mathbf{x})$ patalpinti sprendimo sistemoje $\mathbf{S} = (\mathbf{U}, \mathbf{A} \cup \{d\})$, taikant šio tipo klasifikatorių, pagal sprendimo sistemoje esančią informaciją apskaičiuojamos sąlyginių tikimybių reikšmės: $P(a = a(\mathbf{x}) \mid d(\mathbf{x}) = k)$, čia a – atributo kintamasis, $d(\mathbf{x})$ – sprendimo kategorijos d reikšmė objektui **x**, k – dominanti sprendimo kategorijos reikšmė. *A posteriori* tikimybė kiekvienam naujam objektui **y** (kuris nebuvo naudotas sąlyginėms tikimybėms apskaičiuoti) priklausyti konkrečiai sprendimo kategorijai k, kai žinomos objekto atributų reikšmės, apskaičiuojama pagal Bayeso taisyklę:

$$P\left(d(\mathbf{y}) = k \middle| \{a = a(\mathbf{y})\}_{a \in \mathbf{A}}\right) = \frac{P(d(\mathbf{y}) = k) P\left(\{a = a(\mathbf{y})\}_{a \in \mathbf{A}} \middle| d(\mathbf{y}) = k\right)}{P\left(\{a = a(\mathbf{y})\}_{a \in \mathbf{A}}\right)}.$$

Pagal naivaus Bayeso metodą, atributų reikšmės yra nepriklausomi atsitiktiniai dydžiai, todėl skaitiklio sąlyginė tikimybė išskaidoma dauginamaisiais:

$$P\left(\{a = a(\mathbf{y})\}_{a \in \mathbf{A}} \middle| d(\mathbf{y}) = k\right) = \prod_{a \in \mathbf{A}} P\left(a = a(\mathbf{y}) \middle| d(\mathbf{y}) = k\right).$$

Tokia atributų nepriklausomumo prielaida smarkiai suprastina sąlyginių tikimybių įvertinimą duotai sprendimo sistemai **S**. Kadangi vardiklis nuo sprendimo kategorijos reikšmės nepriklauso, dažnai jis iš skaičiavimų pašalinamas, o skaitiklis normalizuojamas taip, kad jo suma taptų lygi 1.

Naivaus Bayeso klasifikatorius pasižymi dideliu spartumu, kadangi tokio klasifikatoriaus apmokymas yra trivialus – užtenka vieną kartą peržiūrėti sprendimo sistemą – metodo sudėtingumas $O(|\mathbf{U}|)$ eilės. Klasifikatorius taip pat yra paprastas, nes tikimybės $P(a = a(\mathbf{x}) \mid d(\mathbf{x}) = k)$ įvertinamos skaičiuojant atributų reikšmių pasikartojimo dažnius. Nepaisant to, metodas dažnai pasižymi geru klasifikavimo našumu [27].

2.3. Išvados

Šiame skyriuje apžvelgti du klasifikavimo metodai: griežtosios aibės ir naivaus Bayeso metodas. Pirmojo jų sąvoka yra platesnė ir apima eilę etapų, tarp kurių yra ir klasifikavimas. Griežtosios aibės neseniai sukurta teorija, kuri pradžioje buvo taikoma medicininiuose skaičiavimuose. Tačiau laikui bėgant, griežtosios aibės dėl gero klasifikavimo našumo buvo pradėtos naudoti daugelyje kitų klasifikavimo sričių, ir dabar ši teorija laikoma viena iš pažangių metodų duomenų klasifikavimo srityje.

Naivaus Bayeso metodas yra paprastas, tačiau dėl didelio spartumo apmokant klasifikatorių bei dėl nesudėtingų skaičiavimų šis metodas dažnai naudojamas įvairios paskirties duomenų klasifikavime. Taip pat dažnai pabrėžiamas šio metodo tikslumas.

Šiame darbe mes lyginsime pasiūlytą Dirichlet mišinių statistika paremto klasifikavimo metodą su griežtosiomis aibėmis ir su naivaus Bayeso metodu. Dviejų ir daugiau klasifikatorių palyginimas įmanomas tik griežtai apibrėžus našumo sąvokas ir kokybinius matus, pagal kuriuos palyginami klasifikatoriai. Sekantis skyrius apibrėžia klasifikavimo našumo sąvoką ir dydžius, kurie vienu ar kitu aspektu apibūdina klasifikatoriaus veikimą duotiems duomenims.

3. KLASIFIKAVIMO NAŠUMAS IR JO VERTINIMAS

Sukurti, suprojektuoti, įdiegti klasifikatorių yra vienas uždavinys, tačiau ne mažiau svarbu įvertinti klasifikatoriaus našumą. Naudojantis našumo įvertinimu, tiesiogiai įmanoma spręsti apie klasifikatoriaus efektyvumą klasifikuoti to tipo duomenis, kuriems jis buvo projektuojamas ir kuriamas. Be to, remiantis našumo įvertinimais, galima palyginti kelis klasifikatorius ir daryti išvadas apie klasifikatorių panaudojimo galimybes nagrinėjamuose uždaviniuose.

3.1. Duomenų aibės ir testavimo strategijos

Klasifikatorių galime įsivaizduoti kaip funkciją, vienareikšmiškai priskiriančią objektus sprendimo kategorijoms. Pažymėkime bet kokį klasifikatorių, realizuojantį tokią funkciją, raide ϖ. Kadangi klasifikatorius klasifikuoja objektus **x**, pažymėsime objektų aibę, kurią klasifikatorius ϖ turi apdoroti prieš atliekant sprendimą, raide **U**. Tikrąją sprendimo kategoriją, kuriai objektas **x** priklauso, žymėsime $d(\mathbf{x})$, o klasifikatoriaus ϖ sprendimą, atitinkantį spėjamąją kategoriją, žymėsime $\hat{d}_{\varpi}(\mathbf{x})$. Pažymėkime galimų sprendimo kategorijų aibę **D**, o šios aibės elementus: sprendimo kategorijas ne kaip funkcijas nuo objektų, bet kaip reikšmes – užrašysime $\{d_i\}_{i=1}^{|\mathbf{D}|}$, čia $|\mathbf{D}|$ reiškia aibės dydį.

Klasifikavimo našumui įvertinti naudojama duomenų aibė, kuriai taikomas klasifikatorius ϖ. Prižiūrimo klasifikatoriaus apmokymo proceso (*supervised learning*) eigoje naudojami „pažymėti“ objektai iš jų aibės **U**. „Pažymėjimas“ reiškia, kad kiekvienam objektui $\mathbf{x}\in\mathbf{U}$ yra žinoma to objekto tikroji sprendimo kategorija $d(\mathbf{x})$. Klasifikatorius ϖ ignoruoja pažymėjimą, kadangi jis dirba tik su faktiniais duomenimis. Tačiau klasifikavimo rezultate galima įvertinti kaip tiksliai klasifikatorius ϖ klasifikavo duomenis, kiekvienam objektui **x** palyginant žinomas reikšmes $d(\mathbf{x})$ ir sprendimo rezultatus $\hat{d}_{\varpi}(\mathbf{x})$. Kadangi visada stengiamasi gauti kaip galima tikslesnį klasifikatorių, šių dydžių skirtumų modulių suma turėtų būti minimizuojama. Tai yra, jei klasifikatoriaus modelis apibrėžiamas parametrų aibe Θ, „pažymėti“ duomenys taps naudingi optimizuojant klasifikatoriaus ϖ parametrus iš aibės Θ.

Klausimas kyla, kaip adekvačiai įvertinti klasifikatoriaus našumą. Jei klasifikatoriaus ϖ parametrai apmokomi naudojant pilną duomenų imtį, t.y. naudojami visi objektai $\mathbf{x}\in\mathbf{U}$, klasifikavimo tikslumo įvertinimas naudojant tos pačios duomenų imties objektus bus aiškiai nukrypęs (*biased*) ir neatspindės realaus klasifikatoriaus našumo. Vienas iš sprendimų išvengti neadekvataus našumo vertinimo – sudaryti dvi nepersidengiančias duomenų aibes: apmokymo ir testavimo. Apmokymo aibėje esantys objektai naudojami optimizuoti

klasifikatoriaus ϖ parametrus iš aibės Θ, ir klasifikatorius testuojamas naudojant testavimo aibės objektus. Jei šios dvi duomenų aibės nepriklausomos, toks testavimo metodas užtikrina nenukrypusį klasifikatoriaus ϖ našumo įvertinimą.

Kartais apmokymo duomenų aibės naudojimas klasifikatoriaus parametrams įvertinti gali sąlygoti permokymą (*overfit*), kuris reiškia, jog klasifikatorius puikiai geba atpažinti objektus iš apmokymo aibės, tačiau silpnai pasižymi bendrosiomis klasifikavimo savybėmis. Kitaip sakant, nenaudotus apmokymo aibėje objektus klasifikatorius sunkiai atpažįsta arba neatpažįsta. Sprendimai šito išvengti egzistuoja. Vienas iš jų iteratyvaus testavimo (*cross-validation*) metodo [28] naudojimas.

Iteratyvus testavimas naudojamas apmokymo etape ir taikomas apmokymo aibės duomenims arba pilnai duomenų aibei. Duomenų aibė dalinama į k lygių dalių. Klasifikatorius ϖ apmokomas naudojant $k-1$ dalį, ir einamasis našumas įvertinamas naudojant likusią vieną dalį. Kartojant procedūrą k kartų (iteracijų), kiekvienąkart parenkant skirtingas testavimo dalis, garantuojama, kad vidutinis našumo įvertinimas k skirtingoms testavimo duomenų dalims bus nenukrypęs ir adekvatus.

3.2. Nesutapimų matricos

Klasifikatoriaus ϖ našumas gali būti apibendrinamas taip vadinama nesutapimų matrica (*confusion matrix*). Nesutapimų matrica **C** – tai $|\mathbf{D}|\times|\mathbf{D}|$ dydžio matrica, nurodanti klasifikavimo rezultatą kiekvienai sprendimo kategorijai. Matricos elementas c_{ij} lygus objektų skaičiui, kurie priklauso sprendimo kategorijai d_i, bet klasifikatorius ϖ priskyrė kategorijai d_j. Formaliai:

$$c_{ij} = \left| \left\{ \mathbf{x} \in \mathbf{U} \middle| d(\mathbf{x}) = d_i \wedge \hat{d}_{\varpi}(\mathbf{x}) = d_j \right\} \right|. \tag{1}$$

Tikslaus klasifikatoriaus ϖ spėjimai $\hat{d}_{\varpi}(\mathbf{x})$ sutaps su tikromis sprendimų reikšmėmis $d(\mathbf{x})$, todėl tokio klasifikatoriaus nesutapimų matricos įstrižainės elementai bus didžiausi. Tikimybės: kad objektas, klasifikatoriaus ϖ priskirtas spėjimo kategorijai d_j, priklauso sprendimo kategorijai d_i - $P\left(d(\mathbf{x}) = d_i \middle| \hat{d}_{\varpi}(\mathbf{x}) = d_j\right)$; kad objektą **x**, priklausantį sprendimo kategorijai d_i, klasifikatorius priskirs kategorijai d_j - $P\left(\hat{d}_{\varpi}(\mathbf{x}) = d_j \middle| d(\mathbf{x}) = d_i\right)$; bei teisingo spėjimo tikimybė $P\left(d(\mathbf{x}) = \hat{d}_{\varpi}(\mathbf{x})\right)$ – gali būti nesunkiai įvertinamos naudojantis nesutapimų matricos reikšmėmis:

$$P\left(d(\mathbf{x}) = d_i \middle| \hat{d}_{\varpi}(\mathbf{x}) = d_j\right) = \frac{c_{ij}}{\sum_i c_{ij}}, \tag{2}$$

$$P\left(\hat{d}_{\varpi}(\mathbf{x}) = d_j \middle| d(\mathbf{x}) = d_i\right) = \frac{c_{ij}}{\sum_j c_{ij}}, \tag{3}$$

$$P\left(d(\mathbf{x}) = \hat{d}_{\varpi}(\mathbf{x})\right) = \frac{\sum_i c_{ii}}{\sum_{i,j} c_{ij}}. \tag{4}$$

Išraiška (4) išreiškia bendrą klasifikavimo tikslumą. Tikslumas priklauso nuo klasifikatoriaus ϖ spėjimo išrinkimo. Pagal Bayeso taisyklę, klasifikatoriaus spėjimas atitiks sprendimo kategoriją, kuriai spėjimas gautas su didžiausia tikimybe:

$$\hat{d}_{\varpi}(\mathbf{x}) = \arg\max_{d_i} P\left(d(\mathbf{x}) = d_i \middle| \mathbf{x}\right). \tag{5}$$

Tačiau toks spėjimo išrinkimas neįvertina klasifikavimo klaidų svorių, kurie gali kisti priklausomai nuo spėjimo, atitinkančio sprendimo kategoriją. Pavyzdžiui, jei objektas **x** priklauso sprendimo kategorijai $d(\mathbf{x}) \equiv d_i$, o klasifikatorius objektą **x** priskyrė kategorijai d_k, klaidos svoris gali būti priskiriamas kažkokiai reikšmei z_{ik}, kuri priklauso nuo tikrosios sprendimo kategorijos reikšmės ir nuo spėjamos kategorijos reikšmės. Sprendimas įvertinti klaidų svorius - panaudoti reikšmės z_{ik} ir išrinkti sprendimo kategoriją, kaip sumos $\sum_k z_{ik} P\left(d(\mathbf{x}) = d_i \middle| \mathbf{x}\right)$ minimizavimo rezultatą.

Šio darbo tyrimuose nenaudojame klaidų svorių ir visas klaidas baudžiame vienodai. Todėl spėjimas išrenkamas pagal didžiausią gautą tikimybę (5).

Nors tikslumas svarbi sąvoka klasifikavime, automatizuoto apsimokymo (*machine learning*) srityje dažnai sutinkamos tokios sąvokos, kaip teisingi pozityvai (*true positives*), neteisingi pozityvai (*false positives*), teisingi negatyvai (*true negatives*), neteisingi negatyvai (*false negatives*). Visos šios sąvokos apibrėžiamos atskirai sprendimo kategorijai d_i. Teisingi pozityvai (*TP*) reiškia objektų skaičiaus dalį, kurią klasifikatorius teisingai priskyrė sprendimo kategorijai d_i. Neteisingi pozityvai (*FP*) – tai dalis objektų priklausančių kategorijai d_i, kuriuos klasifikatorius klaidingai priskyrė sprendimo kategorijoms. Teisingi negatyvai (*TN*) – tai dalis objektų, kuriuos klasifikatorius teisingai atskyrė kaip nepriklausančius klasifikavimo kategorijai d_i. Neteisingi negatyvai (*FN*) išreiškia objektų dalį, kuriuos klasifikatorius neteisingai spėjo nepriklausančius kategorijai d_i. Šie dydžiai sprendimo kategorijai d_i nesunkiai išreiškiami nesutapimų matricos elementais:

$$TP = c_{ii}, \quad FP = \sum_{k:k\neq i} c_{ki}, \quad TN = \sum_{k:k\neq i} c_{kk}, \quad FN = \sum_{k:k\neq i} c_{ik}. \tag{6}$$

Naudojantis aukščiau apibrėžtomis sąvokomis, apibrėžiami neatsiejamai susiję su automatizuotu apmokymu dydžiai: jautrumas ir specifiškumas. Jautrumas γ išreiškia tikimybę, kad klasifikatorius ϖ objektą **x**, iš tiesų priklausantį kategorijai d_i, tai kategorijai ir priskirs: $P\left(\hat{d}_{\varpi}(\mathbf{x}) = d_i \middle| d(\mathbf{x}) = d_i\right)$. Specifiškumas η atitinka tikimybę, jog klasifikatorius ϖ objektą **x**, iš tiesų nepriklausantį kategorijai d_i, tai kategorijai nepriskirs: $P\left(\hat{d}_{\varpi}(\mathbf{x}) \neq d_i \middle| d(\mathbf{x}) \neq d_i\right)$. Šie dydžiai formaliai apibrėžiami:

$$\gamma = \frac{TP}{TP + FN}, \quad \eta = \frac{TN}{TN + FP}. \tag{7}$$

Jautrumas ir specifiškumas – tai vieni iš pagrindinių dydžių, kuriais remiantis nusakomas klasifikatoriaus našumas.

3.3. Diskriminuojanti galia ir kalibravimas

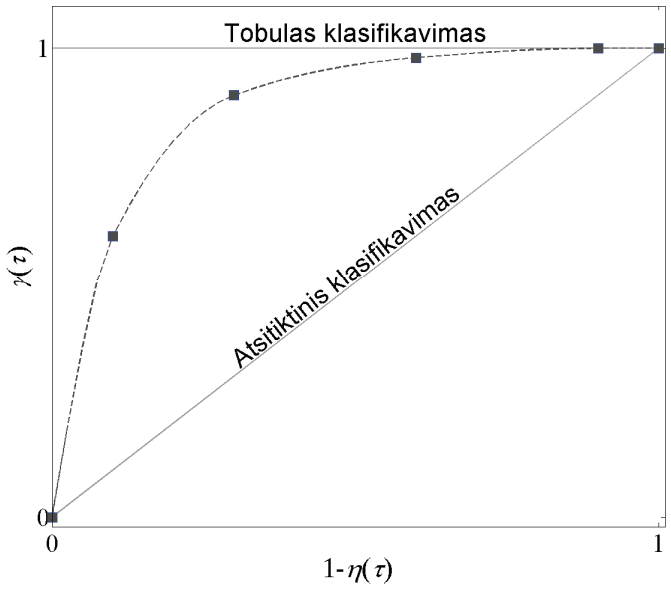
Klasifikatoriaus ϖ sprendimas objektą $\mathbf{x} \in \mathbf{U}$ priskirti sprendimo kategorijai d_i gali būti realizuojamas tikimybe, arba „įsitikinimo“ apie sprendimą reikšme iš intervalo $[0;1]$. Kuo ši reikšmė artimesnė vienetui, tuo klasifikatoriaus ϖ sprendimas patikimesnis. Kita vertus, jei reikšmė yra mažesnė už 0.5, klasifikatoriaus sprendimo laikyti tvirtu negalima, ir toks sprendimas paprastai yra atmetamas. Slenkstis, kurio neviršijus klasifikatoriaus sprendimas nebūtų laikomas patikimu, nebūtinai visada lygus 0.5 ir gali būti parenkamas kitai reikšmei. Dažnai tai priklauso nuo sprendžiamo uždavinio specifikos. Todėl klasifikatoriaus ϖ realizuojamą funkciją $\hat{d}_{\varpi}$ galima išskaidyti į dvi funkcijas, kurių pirmoji ϕ duotam objektui $\mathbf{x} \in \mathbf{U}$ grąžintų sprendimo reikšmę iš intervalo $[0;1]$, o antroji ψ, naudojantis slenkstiniu parametru τ ($0 \le \tau \le 1$), nuspręstų, ar sprendimas gali būti laikomas tvirtu, jog objektas priklauso sprendimo kategorijai d_i. Jei $\phi : \mathbf{U} \to [0;1]$ ir $\psi : [0;1] \to \mathbf{D}$, klasifikatoriui ϖ turėsime $\hat{d}_{\varpi}(\mathbf{x}) \equiv \psi(\phi(\mathbf{x}))$, čia $\psi(\phi(\mathbf{x})) = d_i$, jei $\phi(\mathbf{x}) \ge \tau$. Taigi, analizuojant sprendimo kategorijas d_i atskirai, galima nustatyti tokias τ reikšmes, kurias naudojant klasifikavimas d_i kategorijos atžvilgiu taptų tiksliausias.

Klasifikatoriaus diskriminuojanti galia yra vienas iš pagrindinių klasifikatoriaus našumą nusakančių matų ir nusako, kaip tiksliai klasifikatorius geba atskirti objektus, priklausančius kategorijai d_i, nuo kitų kategorijų objektų. Klasifikavimo progresą charakterizuojanti (*receiver operating characteristic*) kreivė [29], vadinama ROC kreive, - tai grafinis klasifikatoriaus diskriminuojančios galios vaizdavimas. Klasifikavimo teorijoje ROC kreivės plačiausiai naudojamos klasifikavimo našumui vaizduoti. ROC kreivės braižomos duotai

sprendimo kategorijai d_i atidedant ir sujungiant jautrumo-specifiškumo taškus, todėl tiesiogiai parodo santykį tarp šių dviejų dydžių. Kita vertus, jautrumas γ ir specifiškumas η priklausys nuo slenksčio τ, kadangi klasifikatoriaus sprendimas, kaip apibrėžta aukščiau, nuo jo priklauso. Todėl ROC kreivės taškų aibė **R** sudaroma keičiant slenkstinio parametro τ reikšmes:

$$\mathbf{R} = \bigcup_{\tau} \{(1-\eta(\tau), \gamma(\tau))\}. \tag{8}$$

Paprastai kad būtų gaunama vientisa ROC kreivė, prie aibės **R** dirbtinai pridedami du taškai: (0,0) ir (1,1), kurie reiškia, kad arba klasifikatorius išvis nepateikė jokio sprendimo, arba kad klasifikatorius visus objektus aklai priskyrė kategorijai d_i. ROC kreivės pavyzdys matomas 4 paveikslėlyje.



4 pav. ROC kreivės pavyzdys. Kiekvienas taškas gautas keičiant slenkstinio parametro τ reikšmę. Tobulas klasifikavimas pasižymi aukščiausiu jautrumu ir specifiškumu, t.y. nepriklausomai nuo τ reikšmės, klasifikatoriaus jautrumas ir specifiškumas išlieka lygūs 1. Atsitiktinis klasifikavimas pasižymi silpna diskriminuojančia galia, ir tokio klasifikavimo tikslumas, atpažįstant objektus, apylygis klaidų kiekiui.

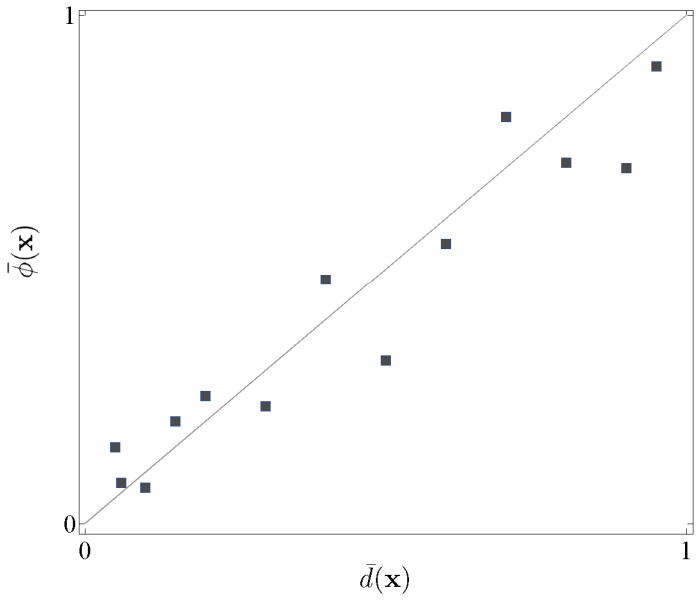
ROC kreivė nurodo klasifikatoriaus našumą keičiantis slenksčio τ reikšmėms, tačiau taikant klasifikatorių praktikoje, reikia įvertinti τ reikšmę, prie kurios klasifikatorius dirba tiksliausiai. Optimalus šios reikšmės parinkimas labai priklauso nuo uždavinio specifikos. Vienas iš būdų parinkti slenkstį τ, tai išrinkti artimiausią (0,1) taškui ROC kreivės tašką ir pagal jį nustatyti τ [22]. Tačiau toks metodas vienodai pasveria teisingų ir klaidingų spėjimų

skaičių, o tai praktikoje kartais gali būti nenaudinga. Kiti įvertinimo būdai įvertina teisingų ir klaidingų spėjimų svorius ir minimizuoja svorinę jautrumo ir specifiškumo sumą τ atžvilgiu.

Plotas po ROC kreive, žymimas AUC, taip pat turi išskirtine reikšmę: jis viena reikšme apibūdina klasifikatoriaus našumą duotai sprendimo kategorijai d_i, t.y. kaip tiksliai klasifikatorius ϖ sugeba atpažinti objektus, priklausančius kategorijai d_i:

$$\mathrm{AUC} = \int_0^1 \gamma(\tau) d\eta(\tau). \tag{9}$$

Atsitiktinio klasifikavimo, arba klasifikavimo, pasižyminčio silpna diskriminuojančia galia, AUC lygus 0.5, o tobulo klasifikavimo AUC = 1. Paprastai AUC plotas skaičiuojamas trapeciniu būdu, tačiau kiti metodai taikomi taip pat: naudojant bendruosius tiesinius modelius ir jų mišinius [30], naudojant empirinius tikimybinius santykius [31], naudojant glodinimo procedūras. Trapecinis būdas linkęs šiek tiek nepilnai įvertinti (*underestimate*) plotą ir susidaro nežymios paklaidos, tačiau skaičiavimo paklaidos įvertinimai leidžia įvertinti kokiu tikslumu yra gautas plotas, o metodo paprastumas padaro jo realizavimą nesunkiu.

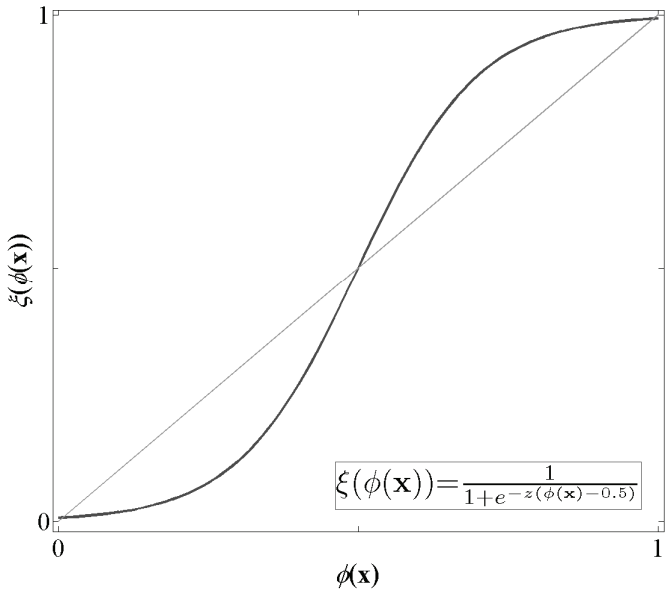


5 pav. Kalibravimo brėžinio pavyzdys. Idealiai sukalibruoto klasifikatoriaus vidutiniai *k* grupėse apskaičiuoti rezultatai (išėjimai) $\bar{\phi}(\mathbf{x})$ puikiai atitiks grupėse vidutines tikrų sprendimų reikšmes $\bar{d}(\mathbf{x})$, ir taškų išsidėstymas bus artimas tiesei ((0,0),(1,1)).

Diskriminuojanti galia nėra vienintelis klasifikatoriaus našumo matas - kalibravimas gali stipriai pagerinti klasifikatoriaus našumą. Kalibravimas, kaip ir ROC kreivių atveju, atliekamas atskirai sprendimo klasėms ir nusako klasifikatoriaus rezultato reikšmių $\phi(\mathbf{x})$ patikimumą, t.y. kokiu tikslumu $\phi(\mathbf{x})$ reikšmės atitinka objektų $\mathbf{x}$ priskyrimo dažnį sprendimo

klasei d_i. Tačiau svarbu pastebėti, kad nebūtinai gerai sukalibruotas klasifikatorius pasižymės stipria diskriminuojančia galia, arba atvirkščiai [22]. Šios sąvokos skirtingos, todėl tyrimuose turi būti stebima, kaip kalibravimas įtakoja diskriminuojančią galią.

Kad gauti kalibravimo brėžinį, sudaromas tikrų sprendimų ir klasifikatoriaus rezultato reikšmių sąrašas $\{(d(\mathbf{x}),\phi(\mathbf{x}))\}$, surūšiuotas pagal $\phi(\mathbf{x})$ (galima priimti, jog $d(\mathbf{x})$ įgyja dvi reikšmes: 1, jei objektas $\mathbf{x}$ priklauso kategorijai d_i, ir 0, jei objektas nepriklauso šiai kategorijai). Sąrašas padalinamas į k lygių dalių ir kiekvienai daliai apskaičiuojamos vidutinės vertės: $\left\{\left(\overline{d}_j(\mathbf{x}),\overline{\phi}_j(\mathbf{x})\right)\right\}_{j=1}^{k}$. Gautos vidutinės vertės atidedamos grafike (5 pav.). Idealiai sukalibruoto klasifikatoriaus rezultatų reikšmės $\phi(\mathbf{x})$ bus artimos tikriems sprendimams $d(\mathbf{x})$, ir tokiu būdu šių reikšmių poros išsidėsto 45^o tiesėje (5 pav.).



6 pav. Kalibravimo funkcijos pavyzdys. Pavyzdyje naudojama sigmoidinė kalibravimo funkcija $\xi(\phi(\mathbf{x}))$, kuri spėjimo reikšmes $\phi(\mathbf{x})$, mažesnes už 0.5, susilpnina, o didesnes už 0.5 – sustiprina; parametras z nusako reikšmių silpninimo ir stiprinimo lygį.

Kalibravimo funkcijų naudojimas yra pagrindinis būdas, kaip sukalibruoti klasifikatorių. Jei kalibravimo funkcija ξ yra griežtai monotoniškai didėjanti, klasifikatoriaus diskriminuojanti galia nesumažėja, ir ROC kreivės atspindi tą pačią diskriminuojančią galią. Todėl tokios kalibravimo funkcijos parinkimas užtikrina, kad ROC analizės rezultate gautas slenkstis τ nustatys didžiausio patikimumo ribą (neteisingų pozityvų skaičius bus santykinai minimalus, jei naudojamas slenkstis τ). Kalibravimo funkcijos $\xi(\phi(\mathbf{x}))$, kaip funkcijos nuo klasifikatoriaus rezultato $\phi(\mathbf{x})$, pavyzdys matomas 6 paveikslėlyje.

Dažnai tiksliai parinkti kalibravimo funkciją ξ sudėtinga, todėl įprasta taikyti tiesinės regresijos modeliavimą, kur gautas regresinis modelis geriausiai atspindėtų tikrų sprendimų ir klasifikatoriaus spėjimo taškų pasiskirstymą [22].

Panašiai kaip ROC kreivės gali būti įvertinamos vienu skaičiumi – AUC, – taip ir klasifikatoriaus kalibravimo kokybę įmanoma įvertinti vienu skaičiumi:

$$B = \frac{1}{|\mathbf{U}|}\sum_{\mathbf{x}\in\mathbf{U}}\left(\phi(\mathbf{x}) - d(\mathbf{x})\right)^2 . \qquad (10)$$

Tai Brierio įvertis [32, 33], kuris atspindi ne tik klasifikatoriaus kalibravimo kokybę, bet ir išreiškia santykį su diskriminavimo galia [33]. Kuo šis įvertis mažesnis, tuo klasifikatorius tikslesnis visų sprendimo kategorijų atžvilgiu.

3.4. Našumo vertinimai ir jų reikšmingumas

Klasifikavimo tikslumas (4), kaip matas vertinti klasifikavimo našumą, nėra pakankamas rodiklis klasifikatoriui vertinti, kadangi jis neatspindi klasifikatoriaus diskriminuojančios galios, klasifikavimo klaidų svorių, jei tokie naudojami, ir neparodo klasifikatoriaus gebėjimų atpažinti objektus iš visų sprendimo kategorijų. Klasifikatoriaus tikslumas gali būti aukštas, bet ROC analizė ir spėjimų pasiskirstymas gali parodyti, kad visi klasifikatoriaus spėjimai atitiko tik vieną sprendimo kategoriją, - suprantama, toks klasifikatorius praktikoje negali būti naudojamas.

Dėl šių priežasčių tikslumas negali būti vienintelis dydis, kuriuo lyginami dviejų klasifikatorių našumai. Plotas po ROC kreive – AUC – ne visada gali atskleisti skirtumus tarp lyginamų klasifikatorių: pavyzdžiui, skirtingos dviejų klasifikatorių ROC kreivės gali būti tokios, kad jų AUC būtų lygūs, nors aukšto specifiškumo lygiuose jų skirtumai gali būti labai reikšmingi. Tai yra svarbu bioinformatikos srityje, kur aukšto specifiškumo lygiai naudojami identifikuoti reikšmingus panašumus tarp biomolekulinių sekų [34]. Todėl kyla klausimas, kaip įvertinti dviejų lyginamų klasifikatorių našumo skirtumus, ir ar jie reikšmingi?

Kad įvertinti dviejų klasifikatorių ϖ_1 ir ϖ_2 klasifikavimo tikslumų skirtumų reikšmingumą, naudojamas taip vadinamas McNemaro[2] įvertinimas [26]. Jei pažymėsime n_{FF} abiejų klasifikatorių klaidingų spėjimų skaičiumi, n_{FT} - spėjimų skaičiumi, kai ϖ_1 pasiūlė teisingą spėjimą, o ϖ_2 – neteisingą, n_{TF} – spėjimų skaičiumi, kai ϖ_2 pasiūlė teisingą spėjimą, o ϖ_1 – neteisingą ir n_{TT} – spėjimų skaičiumi, kai ϖ_1 ir ϖ_2 buvo teisūs (1 lentelė), McNemaro statistika teigia, kad dydis:

[2] Ne parametrinis reikšmingumo įvertinimo metodas, pasiūlytas 1947 m. ir pavadintas pasiūliusio mokslininko Q. McNemar vardu

$$\chi^2 = \frac{\left(\left|n_{TF} - n_{FT}\right| - 1\right)^2}{n_{TF} + n_{FT}} \tag{11}$$

yra atsitiktinis dydis, pasiskirstęs pagal chi-kvadrato pasiskirstymo dėsnį su 1 laisvės laipsniu (stebėjimų ir parametrų skaičių skirtumas). Dydis χ^2 nesunkiai gali būti apskaičiuotas, o tikimybė atsitiktinai gauti tokios ar didesnės reikšmės dydį (pagal chi-kvadrato dėsnį) su reikšme mažesne už 0.05 reikš 95% reikšmingą skirtumą. Jei $n_{TF} + n_{FT} < 10$, dydis χ^2 nėra pasiskirstęs pagal chi-kvadrato dėsnį; tokiu atveju turėtų būti naudojami kiti įvertinimai (pavyzdžiui, Fisherio[3]).

1 lentelė. Klaidingų ir teisingų dviejų klasifikatorių spėjimų lentelės pavyzdys.

	ϖ_1 klaidos	ϖ_1 teisingi spėjimai
ϖ_2 klaidos	n_{FF}	n_{FT}
ϖ_2 teisingi spėjimai	n_{TF}	n_{TT}

McNemaro skaičiavimai naudojami tokiose programose, kaip MedCalc, SPSS ir kituose statistinio apdorojimo paketuose.

Dviejų klasifikatorių tikslumų skirtumų reikšmingumo įvertinimas nėra vienintelis būdas rezultato reikšmingumui nustatyti. Dažniau naudojamas klasifikatorių ϖ_1 ir ϖ_2 ploto po ROC kreive AUC_1 ir AUC_2 skirtumų reikšmingumo įvertis. Galima išskirti dvi AUC skirtumo reikšmingumo vertinimo kryptis: parametrinis vertinimas ir neparametrinis vertinimas. Vienas populiariausių ir dažniausiai cituojamų parametrinio vertinimo metodų – tai Hanley ir McNeilo [29, 35] metodas.

Dviejų klasifikatorių ROC kreivėms, gautoms naudojant tas pačias duomenų imtis (tik tokiu atveju klasifikatorių lyginimas yra prasmingas), Hanley ir McNeilas pasiūlė formulę skaičiuoti AUC kvadratinę paklaidą (pavyzdžiui, AUC skaičiavimas trapeciniu būdu visada įneš savas paklaidas, ir jos priklausys nuo taškų skaičiaus ROC kreivėje):

$$\sigma(\mathrm{AUC}) = \sqrt{\frac{\mathrm{AUC}(1-\mathrm{AUC}) + \left(\left|\mathbf{U}_P\right| - 1\right)\left(Q_1 - \mathrm{AUC}^2\right) + \left(\left|\mathbf{U}_N\right| - 1\right)\left(Q_2 - \mathrm{AUC}^2\right)}{\left|\mathbf{U}_P\right| \cdot \left|\mathbf{U}_N\right|}}. \tag{12}$$

Šioje formulėje naudojami tokie dydžiai: Q_1 ir Q_2 – tai atitinkamai tikimybė du neigiamus objektus įvertinti aukščiau nei atsitiktinai išrinktą teigiamą objektą ir tikimybė vieną neigiamą objektą įvertinti aukščiau nei atsitiktinai išrinktus du teigiamus objektus; šios tikimybės lygios $Q_1 = \mathrm{AUC}/(2 - \mathrm{AUC})$ ir $Q_2 = 2\,\mathrm{AUC}^2/(1 + \mathrm{AUC})$. Teigiami ir neigiami objektai atitinkamai

[3] 1922 m. R. A. Fisherio pasiūlytas statistinio reikšmingumo įvertinimo metodas mažoms duomenų apimtims

reiškia objektus iš sprendimo kategorijai priklausančios objektų aibės ($\mathbf{U}_P$) ir iš kategorijai nepriklausančios objektų aibės ($\mathbf{U}_N$). Remiantis įvestais pažymėjimais (6), galima užrašyti $\mathbf{U}_P \equiv TP + FN$ ir $\mathbf{U}_N \equiv TN + FP$. Kadangi du dviejų klasifikatorių ϖ_1 ir ϖ_2 plotai AUC_1 ir AUC_2 yra koreliuoti dydžiai ir kiekvienam ploto skaičiavimui gaunamos tam tikros kvadratinės paklaidos $\sigma(\mathrm{AUC}_1)$ ir $\sigma(\mathrm{AUC}_2)$, dviejų plotų skirtumo reikšmingumas priklausys nuo abiejų plotų reikšmių bei nuo plotų skirtumo kvadratinės paklaidos [35], kuri skaičiuojama

$$\sigma\left(\mathrm{AUC}_1 - \mathrm{AUC}_2\right) = \sqrt{\sigma^2\left(\mathrm{AUC}_1\right) + \sigma^2\left(\mathrm{AUC}_2\right) - 2r\sigma\left(\mathrm{AUC}_1\right)\sigma\left(\mathrm{AUC}_2\right)}. \qquad (13)$$

Šioje išraiškoje r išreiškia dviejų plotų AUC_1 ir AUC_2 koreliaciją. Koreliacijai r apskaičiuoti naudojama specialiai sudaryta lentelė [35], pagal kurią išrenkamos atitinkamos r reikšmės. Kad pasinaudoti lentele, turi būti žinomi dviejų plotų vidurkis bei koreliacijų koeficientų r_P ir r_N vidurkis, kurie reiškia: r_P – tai klasifikatorių ϖ_1 ir ϖ_2 spėjimo reikšmių $\phi(\mathbf{x})$ koreliacija objektams $\mathbf{x} \in \mathbf{U}_P$, r_N – klasifikatorių ϖ_1 ir ϖ_2 spėjimo reikšmių koreliacija objektams $\mathbf{x} \in \mathbf{U}_N$. Koreliacijos r_P ir r_N nesunkiai gali būti apskaičiuojamos naudojantis Pearsono koreliacijos formule[4].

Hanley ir McNeilas tame pačiame straipsnyje pažymi, kad dydis

$$z = \frac{\mathrm{AUC}_1 - \mathrm{AUC}_2}{\sigma\left(\mathrm{AUC}_1 - \mathrm{AUC}_2\right)} \qquad (14)$$

yra pasiskirstęs pagal normalųjį dėsnį su parametrais 0 ir 1, t.y. $z \sim N(0,1)$, ir gali būti naudojamas skirtumo statistiniam reikšmingumui įvertinti. Vadinasi, apskaičiuoto dydžio z tikimybės reikšmė (pagal normalųjį dėsnį) mažesnė už 0.05 reikš 95% reikšmingą skirtumą.

Neparametrinio klasifikatorių našumų skirtumo vertinimo metodo atveju, išvengiama parametrų skaičiavimų, tačiau naudojami intensyvesni skaičiavimai su teisingų pozityvų ir negatyvų reikšmėmis. Vienas populiariausių neparametrinio skirtumų vertinimo metodų yra DeLongo ir bendraautorių pasiūlytas metodas [36, 37]. Straipsnių autoriai pažymi, jog atsitiktinis dydis $z = \left(\hat{\theta}_1 - \hat{\theta}_2\right) / \sigma\left(\hat{\theta}_1 - \hat{\theta}_2\right)$, panašiai kaip ir parametrinio vertinimo metodo atveju, yra pasiskirstęs pagal standartinį normalųjį dėsnį, tačiau šiuo atveju dydžiai $\hat{\theta}_1$ ir $\hat{\theta}_2$ atitinka teisingų pozityvų ir negatyvų skaičiavimo santykius klasifikatoriams ϖ_1 ir ϖ_2, o standartinis nuokrypis σ skaičiuojamas nenaudojant koreliacijos reikšmių lentelių [36].

---

[4] Pearsono koreliacijos koeficientas $\rho_{x,y} = \dfrac{\overline{xy} - \overline{x}\,\overline{y}}{\sqrt{\overline{x^2} - (\overline{x})^2}\sqrt{\overline{y^2} - (\overline{y})^2}}$

Abi klasifikavimo našumų skirtumo vertinimo kryptys yra vienodai patikimos, paklaidos gautos dvejais metodais yra labai artimos [38], ir šios skirtingos kryptys naudojamos šiuolaikiniuose statistikos programų paketuose. Pavyzdžiui, MedCalc naudoja Hanley ir McNeilo metodą, SAS naudoja neparametrinį DeLongo metodą.

3.5. Išvados

Sukurto klasifikatoriaus klasifikavimo kokybei įvertinti atliekami našumo testai, kurie atskleidžia klasifikatoriaus galimybes klasifikuoti vienos ar kitos paskirties duomenis. Klasifikavimo našumas labai priklauso nuo pasirinktos testavimo strategijos. Duomenų išskaidymas į apmokymo ir testavimo imtis leidžia objektyviai įvertinti klasifikatoriaus našumą. Naudojantis iteratyvia testavimo strategija, išvengiama galimų testavimo aibės nuokrypių (kai duomenis sudaro daugiausia vienos sprendimo kategorijos objektai), ir klasifikatoriaus našumo įvertinimas tampa dar objektyvesnis. Našumą galima vertinti įvairiai: skaičiuojant nesutapimų matricas, įvertinant klasifikatoriaus diskriminuojančią galią. Didesnis našumo vertinimo kriterijų kiekis labiau atskleidžia klasifikatoriaus ypatybes. Diskriminuojanti galia yra viena iš pagrindinių našumą atspindinčių sąvokų, kuri atskleidžia klasifikatoriaus gebėjimą kiekvienai sprendimo kategorijai teisingai atpažinti kategorijai priklausančius ir nepriklausančius objektus. Klasifikatoriaus kalibravimas gali dar labiau padidinti jo diskriminuojančią galią. Palyginant du skirtingus klasifikatorius, svarbu nustatyti jų našumų skirtumo reikšmingumą. Našumo skirtumai gali būti vertinami bendro klasifikavimo tikslumo atžvilgiu ir diskriminuojančios galios atžvilgiu. Abiem atvejais, dviejų klasifikatorių našumo reikšmingas skirtumas rodo vieno klasifikatoriaus pranašumą kito atžvilgiu.

4. DIRICHLET MIŠINIŲ STATISTIKA KLASIFIKAVIME

Šiame skyriuje apibrėžiami Dirichlet skirstinys ir jų mišiniai. Trumpai pateikiama teorija, kuria remiantis sudaromas Dirichlet mišinių klasifikatorius; parodoma, kaip gali būti naudojamas Dirichlet mišinių pasiskirstymo dėsnis klasifikuojant realius duomenis.

4.1. Dirichlet skirstinys ir jų mišiniai

Dirichlet skirstinys g – tai tikimybių vektorių **p** pasiskirstymo funkcija [12, 16]. Įveskime kažkokią alfabeto aibę ir pažymėkime ją raide **A** su skirtingais |**A**| alfabeto simboliais. Kiekvienas įmanomas vektorius **p** atspindės *a priori* tikimybes alfabeto **A** simboliams pasiskirstyti. Dirichlet skirstinys apibrėžiamas parametrų vektoriumi $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{|\mathbf{A}|}$, kurio elementai $\alpha_i > 0$, ir išreiškiamas:

$$g(\mathbf{p} \mid \boldsymbol{\alpha}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{i=1}^{|\mathbf{A}|}\Gamma(\alpha_i)} \prod_{i=1}^{|\mathbf{A}|} p_i^{\alpha_i - 1} , \tag{15}$$

čia $\Gamma(\cdot)$ žymi *gamma* funkciją [39], $|\boldsymbol{\alpha}| = \sum_{i=1}^{|\mathbf{A}|} \alpha_i$, $p_i \geq 0$ ($i = 1,\ldots,|\mathbf{A}|$) ir $\sum_{i=1}^{|\mathbf{A}|} p_i = 1$.

Dirichlet skirstinių mišinys – tai individualių Dirichlet skirstinių matematinė kombinacija, kuri suformuoja naują tikimybių pasiskirstymą. Kiekvienam individualiam skirstiniui mišinyje suteikiami svoriai, kurie vadinami mišinio koeficientais. Kiekvienas individualus skirstinys mišinyje vadinamas mišinio komponentu. Dirichlet mišinio pasiskirstymas φ, sudarytas iš l komponentų, išreiškiamas:

$$\varphi = \sum_{j=1}^{l} q_j g_j , \tag{16}$$

čia g_j – Dirichlet skirstiniai, apibrėžiami parametrų aibėmis $\boldsymbol{\alpha}_j = \{\alpha_{ji}\}_{i=1}^{|\mathbf{A}|}$, q_j – mišinio koeficientai, tenkinantys sąlygą: $\sum_{j=1}^{l} q_j = 1$. Pilna mišinio parametrų aibė $\Theta = \left(\{\boldsymbol{\alpha}_j\}_{j=1}^{l}, \{q_j\}_{j=1}^{l}\right)$ vadinama Dirichlet mišinio modeliu. Kiekvienas komponentas mišinyje aprašo parametrais apibrėžtą pasiskirstymo dėsnį, todėl jų mišinys naudingas klasifikuojant duomenis, išsiskiriančius savybėmis, kurias gali atpažinti individualūs komponentai. Mišinys, sudarytas iš vieno komponento, susiprastina iki Dirichlet skirstinio. Komponentų skaičius mišinyje neribojamas, tačiau didelis jų skaičius padidina modelio parametrų skaičių ir apsunkina optimalių parametrų reikšmių radimą.

4.2. Klasifikavimas Dirichlet mišiniais

Priimkime, kad simboliai iš alfabeto **A** yra atsitiktiniai dydžiai, pasiskirstę pagal multinominį pasiskirstymo dėsnį, ir kiekvienam simboliui a_i žinomas jų pasikartojimo skaičius, kuris žymimas dažniu n_i. Tuomet pilnas dažnių vektorius $\mathbf{n} = \{n_i\}_{i=1}^{|\mathbf{A}|}$, ir dažnių vektoriaus tikėtinumas, kai žinomos simbolių pasikartojimų tikimybės, apibrėžiamas:

$$P(\mathbf{n} \mid \mathbf{p}) = \Gamma(|\mathbf{n}| + 1) \prod_{i=1}^{|\mathbf{A}|} \frac{p_i^{n_i}}{\Gamma(n_i + 1)}, \tag{17}$$

čia $|\mathbf{n}| = \sum_{i=1}^{|\mathbf{A}|} n_i$, p_i – simbolio a_i iš **A** pasirodymo tikimybė. Egzistuoja eilė metodų tikimybėms p_i įvertinti; vienas iš jų Dirichlet skirstinys ar jų mišiniai. Sakykime, kad atsitiktiniai dydžiai p_i pasiskirstę pagal Dirichlet dėsnį (15). Matyti, kad tikimybių vektorius **p** priklauso nuo parametrų vektoriaus $\boldsymbol{\alpha}_j$ ($\boldsymbol{\alpha} \equiv \boldsymbol{\alpha}_j$), ir tai reiškia, kad dažnių vektoriaus tikėtinumas priklauso nuo parametrų $\boldsymbol{\alpha}_j$ taip, kad

$$P(\mathbf{n} \mid \boldsymbol{\alpha}_j) = \int_{\mathbf{p} \in \mathcal{P}} P(\mathbf{n} \mid \mathbf{p}) g(\mathbf{p} \mid \boldsymbol{\alpha}_j) d\mathbf{p}, \tag{18}$$

čia integralas skaičiuojamas visoje galimų tikimybių vektorių **p** apibrėžimo srityje $\mathcal{P}$. Įstačius (15) ir (17) į (18), gauname [16]:

$$P(\mathbf{n} \mid \boldsymbol{\alpha}_j) = \frac{\Gamma(|\mathbf{n}| + 1)\Gamma(|\boldsymbol{\alpha}_j|)}{\Gamma(|\mathbf{n}| + |\boldsymbol{\alpha}_j|)} \prod_{i=1}^{|\mathbf{A}|} \frac{\Gamma(n_i + \alpha_{j,i})}{\Gamma(n_i + 1)\Gamma(\alpha_{j,i})}. \tag{19}$$

Jei priimsime, kad **p** pasiskirstęs pagal Dirichlet mišinių pasiskirstymo dėsnį, dažnių vektorius **n** priklausys nuo modelio Θ parametrų, ir vektoriaus **n** tikėtinumas tampa apibrėžiamas taip:

$$P(\mathbf{n} \mid \Theta) = \sum_{j=1}^{l} q_j P(\mathbf{n} \mid \boldsymbol{\alpha}_j). \tag{20}$$

Išbaigtą duomenų klasifikavimo procesą galima suskirstyti į dvi fazes: apmokymą ir klasifikavimą. Apmokymo fazėje klasifikatorius „apmokomas“ atpažinti būdingus klasėms duomenų objektus, todėl šiame etape tikimybės, kurios naudojamos dažnių (stebėjimų) pasiskirstymuose, turėtų būti įvertinamos kuo tiksliau. Klasifikavimo fazėje „apmokytas“ klasifikatorius klasifikuoja „apmokymuose“ nenaudotus duomenis ir dažnai šiuos duomenis sudaro naujai surinkti stebėjimai, kuriems sprendimas nėra žinomas. Todėl įvertintos

apmokymo etape tikimybės, manoma, geriausiai nusakys dažnių (stebėjimų) pasiskirstymus iš klasifikuojamų duomenų aibės. Tai reiškia, kad, jei apmokymo fazėje gautos optimalios tikimybės, labai tikėtina, kad klasifikavimo fazėje duomenys bus klasifikuojami tiksliai. Tačiau iškyla klausimas, kaip rasti optimalius tikimybių įvertinimus.

Posteriorinio vidurkio įvertinimas, maksimalaus tikėtinumo įvertinimas – tai keletas hipotezių testavimo metodų pavyzdžių. Žinoma [40, 41], kad *posteriorinio* vidurkio įvertis lygus

$$\hat{p}_i = \int\limits_{\mathbf{p}\in\mathcal{P}} p_i P(\mathbf{p} \mid \Theta, \mathbf{n}) d\mathbf{p}, \tag{21}$$

čia $\mathbf{p}$ yra tikimybių p_i vektorius, pasiskirstęs pagal Dirichlet mišinių pasiskirstymo dėsnį. Turėdami Dirichlet mišinių tankį, galime išreikšti

$$P(\mathbf{p} \mid \Theta, \mathbf{n}) = \sum_{j=1}^{l} P(\mathbf{p} \mid \boldsymbol{\alpha}_j, \mathbf{n}) P(\boldsymbol{\alpha}_j \mid \mathbf{n}, \Theta), \tag{22}$$

ir, įstačius pastarąją išraišką į (21) , gausime

$$\hat{p}_i = \sum_{j=1}^{l} P(\boldsymbol{\alpha}_j \mid \mathbf{n}, \Theta) \int\limits_{\mathbf{p}\in\mathcal{P}} p_i P(\mathbf{p} \mid \boldsymbol{\alpha}_j, \mathbf{n}) d\mathbf{p}. \tag{23}$$

Iš teorijos [34] žinoma, kad *posteriorinio* vidurkio įvertis, vieno Dirichlet skirstinio atveju, lygus

$$\hat{p}_i^{\mathrm{s}} = \int\limits_{\mathbf{p}\in\mathcal{P}} p_i P(\mathbf{p} \mid \boldsymbol{\alpha}_1, \mathbf{n}) d\mathbf{p} = \frac{n_i + \alpha_{1,i}}{|\mathbf{n}| + |\boldsymbol{\alpha}_1|}, \tag{24}$$

čia parametrai $\{\alpha_{1,i}\}$ sudaro vieną vektorių $\boldsymbol{\alpha}_1$ (mišinys susideda iš vieno Dirichlet skirstinio). Naudojantis Bayeso taisykle, galima išreikšti

$$P(\boldsymbol{\alpha}_j \mid \mathbf{n}, \Theta) = \frac{q_j P(\mathbf{n} \mid \boldsymbol{\alpha}_j)}{P(\mathbf{n} \mid \Theta)}. \tag{25}$$

Atlikus perstatymus – (24) ir (25) į (23), – gausime

$$\hat{p}_i = \frac{1}{P(\mathbf{n} \mid \Theta)} \sum_{j=1}^{l} q_j P(\mathbf{n} \mid \boldsymbol{\alpha}_j) \frac{n_i + \alpha_{j,i}}{|\mathbf{n}| + |\boldsymbol{\alpha}_j|}. \tag{26}$$

Pastaroji išraiška atitinka *posteriorinio* vidurkio įvertį tikimybėms p_i. Tačiau vis dar lieka nežinoma, kaip gauti optimalias modelio Θ parametrų reikšmes. Sugrįžkime trumpam prie tikėtinumo $P(\mathbf{n} \mid \Theta)$ išraiškos. Tikėtinumo reikšmė priklauso tiek nuo visų modelio parametrų, kurių reikšmes reikia rasti, tiek nuo dažnių (stebėjimų) vektoriaus **n**. Prieš priimant bet kokį sprendimą, žmonės, norėdami surinkti pakankamą eksperimentinių duomenų kiekį, paprastai atlieka daug matavimų. Panašiai yra su automatiniais klasifikavimo metodais: kuo daugiau skirtingų duomenų naudojama metodo apmokymui, tuo tikslesnius rezultatus galima tikėtis gauti taikant metodą duomenims. Pavyzdžiui, taisyklėmis paremtas klasifikatorius sugeneruos griežtesnę ir bendresnę neperteklinę taisyklių aibę pagal duomenis, surinktus iš daugelio pacientų, sakykime, su širdies ritmo sutrikimo negalavimais, nei tuo atveju, jei klasifikatorius būtų apmokomas pagal vieno paciento duomenis. Arba, baltymų klasifikavimas į jų šeimas negalėtų būti patikimas, jei klasifikatorius būtų apmokytas baltymų, paimtų po vieną iš šeimos, amino rūgščių pasiskirstymais. Todėl pakankamas duomenų kiekis ir tinkama jų kompozicija yra svarbūs kuriant tikslius klasifikatorius.

Lygtyse naudojamas vektorius **n** atitinka vieną stebėjimą (žinomi duomenys: n_i atitinka reikšmę vienam atributui). Tačiau stebėjimų paprastai atliekama daug, todėl kad rasti optimalias reikšmes ieškomiems parametrams, klasifikatorius dažnai apdoroja daug stebėjimų (dažnių) vektorių $\{\mathbf{n}_c\}_{c=1}^{N}$. Jei vektorius jų aibėje $\{\mathbf{n}_c\}$ laikysime nepriklausomais ir identiškai pasiskirsčiusiais atsitiktiniais dydžiais, tuomet, pagal maksimalaus tikėtinumo įvertinimą [41], modelio Θ parametrai gali būti optimizuojami maksimizuojant sandaugą $\Pi_c P(\mathbf{n}_c \mid \Theta)$. Kadangi logaritmo funkcija yra monotoniškai didėjanti funkcija, optimalius parametrus galima rasti minimizuojant logaritmų sumą vietoje to, kad maksimizuoti tiesioginę tikimybių sandaugą:

$$f(\Theta) = -\sum_{c=1}^{N} \log P(\mathbf{n}_c \mid \Theta). \qquad (27)$$

Paskutinė išraiška (27) atitinka tikslo funkciją, kuri turėtų būti minimizuojama, kad rasti optimalius klasifikatoriaus parametrus.

4.3. Išvados

Tiek Dirichlet skirstinys, tiek Dirichlet mišinio skirstinys gali būti naudojami duomenų klasifikavime. Abiem atvejais skiriasi optimizuojamų parametrų skaičius. Jei klasifikavime naudojamas Dirichlet mišinys, optimizuojamų parametrų skaičius išauga priklausomai nuo mišinio komponentų skaičiaus. Tačiau Dirichlet mišinio ypatybė vienu metu komponuoti keletą ar daug Dirichlet skirstinių leidžia atpažinti skirtingai pasiskirsčiusius dydžius

(Dirichlet skirstinių atsitiktinius dydžius), ir toks pasiskirstymo dėsnis gali būti efektyviai naudojamas duomenų klasifikavime. Kaip parinkti mišinio komponentų skaičių nėra griežtai nustatyta, ir efektyvus komponentų skaičius dažniausiai priklauso nuo sprendžiamo uždavinio specifikos. Dirichlet mišinio parametrai optimizuojami taikant maksimalaus tikėtinumo formuluotę ir priimant, kad stebėjimai tarpusavyje yra nepriklausomi dydžiai.

5. DIRICHLET MIŠINIŲ KLASIFIKATORIAUS PROJEKTAVIMAS IR REALIZACIJA

Dirichlet mišinių statistika paremtą klasifikatorių projektavome darbui su bet kokio tipo duomenimis ir realizavome kaip nepriklausomą programinį įrankį ir kaip klasifikavimo metodų programinę biblioteką. Šiame skyriuje apibrėžiami svarbiausi projektavimo ir realizacijos aspektai, kurie daro įtaką sukurtos sistemos našumui inžineriniu ir moksliniu atžvilgiais.

5.1. Klasifikatoriaus projektavimas

Dirichlet mišinius projektavome tikėdamiesi panaudoti integruotą žinių kaupimo sistemą Rosetta 1.0 [19]. Rosetta apima automatinio apsimokymo skaičiuojamąjį branduolį, paremtą griežtųjų aibių teorija [21, 23], taip pat Rosetta sistema apima kitus, paruošiamuosius žinių kaupimo etapus, kurių vykdymas palengvina apdoroti duomenis, sudaryti griežtesnes ir bendresnes taisyklių aibes. Tie kiti etapai – tai duomenų diskretizavimas, redukavimas, transformavimas (*scaling*), trūkstamų duomenų užpildymas ir kiti. Kiekviename etape yra apibrėžta aibė tam etapui būdingų metodų.

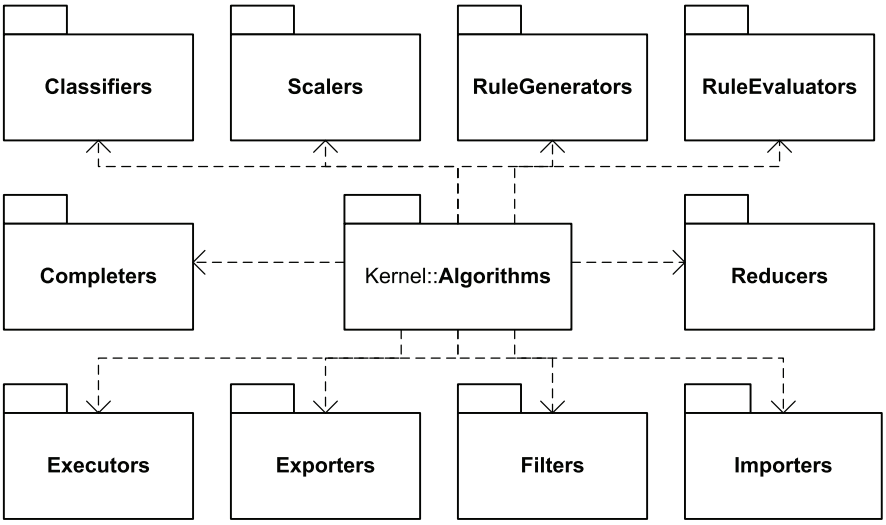
Rosetta sistemos programinis išeities tekstas ne komerciniams tikslams yra laisvai prieinamas [42], struktūriškas ir atitinka pakartotinio panaudojimo reikalavimus. Tačiau mūsų sprendimas naudotis šia sistema yra ne vien dėl jos viešumo, bet dėl jos ir joje realizuotų metodų pasiteisinimo moksliniuose tyrimuose. Skaičiavimų griežtosiomis aibėmis bendrai bei Rosetta sistemos taikymas įrodė esąs naudingas įvairiuose moksliniuose tyrimuose: nustatant ryšį tarp susirgimo vėžiu (skrandžio karcinomos) ir genų ekspresijos kitimo [43], anksti diagnozuojant vainikinių arterijų ligą [44], pagal genų ekspresijos duomenis nusakant baltymų funkcijas [45] bei identifikuojant baltymų jungimosi sritis [46] ir kituose tyrimuose. Šios sistemos panaudojimas Dirichlet mišiniais paremto klasifikavimo lyginimą su klasifikavimu, paremtu griežtosiomis aibėmis, padarytu savaime suprantamu. Kita vertus, dalis Rosetta programinio kodo[5] būtų panaudojama Dirichlet mišinių klasifikatoriaus kūrime, tokiu būdu kūrimo procesas taptų efektyvesnis, greitesnis ir atitiktų pakartotinio panaudojamumo kryptį, kuri tampa esminė kuriant šiuolaikines sudėtingas programines sistemas.

Dirichlet mišinių klasifikatorių projektavome išpildydami Rosetta sistemos struktūriškumą. Toks projektas užtikrino, kad sukurtas klasifikatorius bus taip pat struktūriškas ir tenkins pakartotino panaudojamumo reikalavimus. Dar daugiau, mes

[5] Rosetta paketą sudaro 722 bylos ir ~136 tūkst. eilučių

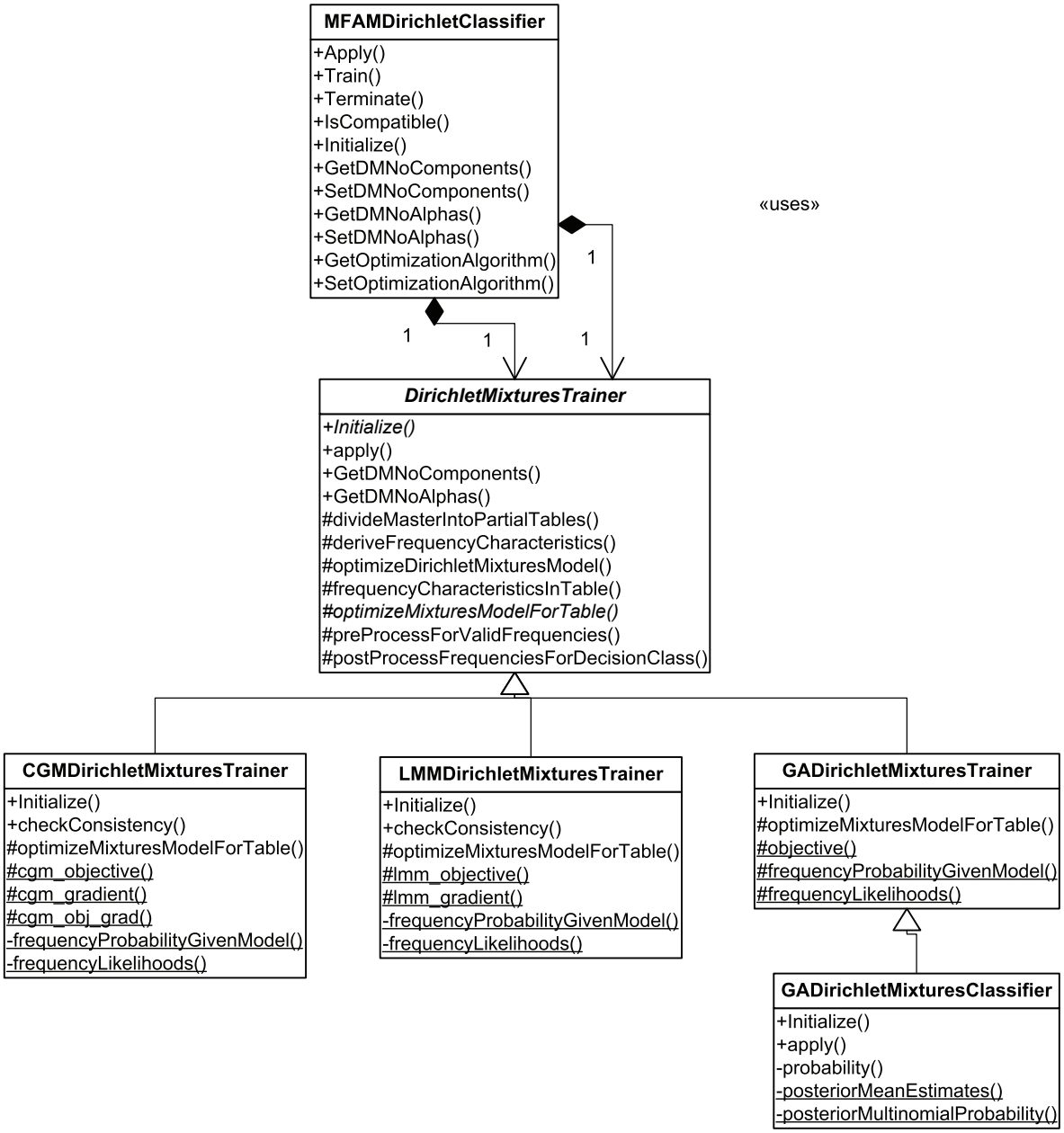
projektavome Dirichlet mišinių klasifikatorių taip, kad jis galėtų nesunkiai būti prijungiamas prie Rosetta paketo ir tokiu būdu praplėstų Rosetta žinių kaupimo aplinką nepriklausomu klasifikavimo metodu, arba galėtų būti naudojamas kaip nepriklausoma Dirichlet mišinių klasifikavimo programinė biblioteką. Nors Rosetta biblioteką Dirichlet mišinių kūrimo procese naudojome tik duomenims vaizduoti vidiniame Rosetta formate, atsižvelgiant į minėtus projektavimo aspektus, labai svarbu suprasti ir išlaikyti Rosetta sistemos struktūriškumo principus.

Rosetta sistema sudalinta į paketus, kurių svarbiausieji – tai *Structures* ir *Algorithms*. Paketas *Structures* – duomenų struktūroms skirtas paketas, *Algorithms* – įvairiems algoritmams ir metodams, apimant diskretizavimo, redukavimo, duomenų transformavimo, užpildymo ir kitus įvairius algoritmus, skirtas paketas (7 pav.).



7 pav. Sistemos Rosetta paketo *Algorithms* struktūra.

Trumpai galima taip apibrėžti paketo *Algorithms* dalinių paketų paskirtis ir turinį: *Importers* – sprendimo lentelės skaitymo į sistemą algoritmai (palaiko įvairius duomenų formatus), *Completers* – trūkstamų sprendimo lenetelės atributų užpildymo algoritmai, *Exporters* – sprendimo lentelės rašymo į išorinius duomenų formatus algoritmai, *Scalers* – sprendimo lentelės diskretizavimo algoritmai, *Reducers* – sprendimo lentelės redukavimo (atributų prastinimo) algoritmai, *Filters* – redukuotos sprendimo lentelės ir taisyklių filtravimo algoritmai, *RuleGenerators* – taisyklių generavimo pagal sudarytas redukcijas algoritmai, *RuleEvaluators* – klasifikavimo taisyklėmis diskriminuojančios galios įvertinimo algoritmai, *Executors* – nurodytų nuoseklių komandų vykdymo realizacija ir *Classifiers* – sprendimo lentelės objektų klasifikavimo algoritmai. Paketas *Classifiers* yra svarbiausias kuriant ir projektuojant Dirichlet mišinių klasifikatorių, kadangi Dirichlet mišinių klasifikatorius priskiriamas šiam paketui.



8 pav. Dirichlet mišinių klasifikatoriaus supaprastinta klasių struktūrinė diagrama.

Dirichlet mišinių klasifikatorius buvo projektuojamas prisilaikant Rosetta programinių standartų, kad sukurtas klasifikatorius galėtų būti naudojamas kaip programinė biblioteka, kaip nepriklausomas įrankis, ar integruotas tiesiogiai į Rosetta paketą. Supaprastinta Dirichlet mišinių klasifikatoriaus klasių struktūrinė diagrama pavaizduota 8 paveikslėlyje. Klasė *MFAMDirichletClassifier* – pagrindinis klasifikavimo Dirichlet mišiniais modulis. Šios klasės metodai naudojami apmokyti klasifikatorių bei, naudojantis apmokytu klasifikatoriumi, klasifikuoti duomenis. Dirichlet mišinio komponentų skaičius koreguojamas/grąžinamas metodais *SetDMNoComponents()*/*GetDMNoComponents()*; pseudo dažnių parametrų skaičius mišinio komponente atitinkamai koreguojamas/grąžinamas metodais *SetDMNoAlphas()* /*GetDMNoAlphas()*. Dirichlet mišinio modelio ir sprendimo lentelės tinkamumu galima

įsitinkinti naudojant metodą *IsCompatible()*. Duomenų klasifikavimas galimas tik, jei Dirichlet mišinių modelis buvo arba apmokytas, arba modelio parametrų reikšmės buvo nuskaitytos iš bylos. Klasifikatoriaus apmokymas pradedamas metodu *Train()*, o optimizavimo metodas nurodomas/grąžinamas metodais *SetOptimizationAlgorithm()*/ *GetOptimizationAlgorithm()*. Duomenų klasifikavimas pradedamas metodais *Initialize()* ir *Apply()*. Klasifikavimas gali būti nutraukiamas metodu *Terminate()*.

Abstrakti klasifikatoriaus apmokymo klasė *DirichletMixturesTrainer* įprasmina tris konkrečius optimizavimo metodus, kuriais paremtas klasifikatoriaus apmokymas: jungtinio gradiento metodą (*CGMDirichletMixturesTrainer*), kvazi Newtono Levenberg-Marquardt metodą (*LMMDirichletMixturesTrainer*), genetinį algoritmą (*GADirichletMixturesTrainer*, detaliau optimizavimo metodų pasirinkimo pagrindimas aprašytas žemiau). Klasifikatoriaus apmokymas pradedamas *initialize()* ir *apply()* metodais. Apmokymo procesas susideda iš trijų pagrindinių etapų:

- sprendimo lentelės išskaidymas į dalines pagal sprendimo kategorijos reikšmes (dalinės sprendimo lentelės su ta pačia sprendimo kategorijos reikšme, *divideMasterIntoPartialTables()*).
- Dažninių charakteristikų kiekvienam atributui skaičiavimas (*deriveFrequencyCharacteristics()*). Šis etapas apima dažninių charakteristikų skaičiavimą visoms dalinėms sprendimo lentelėms (*frequencyCharacteristicsInTable()*). Be to, kiekviena sprendimo lentelė apdorojama prieš skaičiuojant dažnines charakteristikas (*preProcessForValidFrequencies()*) ir po to (*postProcessFrequenciesForDecisionClass()*). Pastarieji skaičiavimai atliekami, kad įvertinti neigiamų atributų reikšmių pasiskirstymą ir atlikti reikšmių normalizavimą ir transformavimą, kuris būtinas norint išvengti didelių skirtingų atributų reikšmių skirtumų, galinčių neigiamai paveikti optimizavimo procesą.
- Dirichlet mišinio optimizavimas kiekvienai dalinei sprendimo lentelei (*optimizeDirichletMixturesModel()* ir *optimizeMixturesModelForTable()*). Optimizavimo metodas yra abstraktus, ir kiekvienas naudojamas optimizavimo metodas skirtingai realizuoja šį metodą.

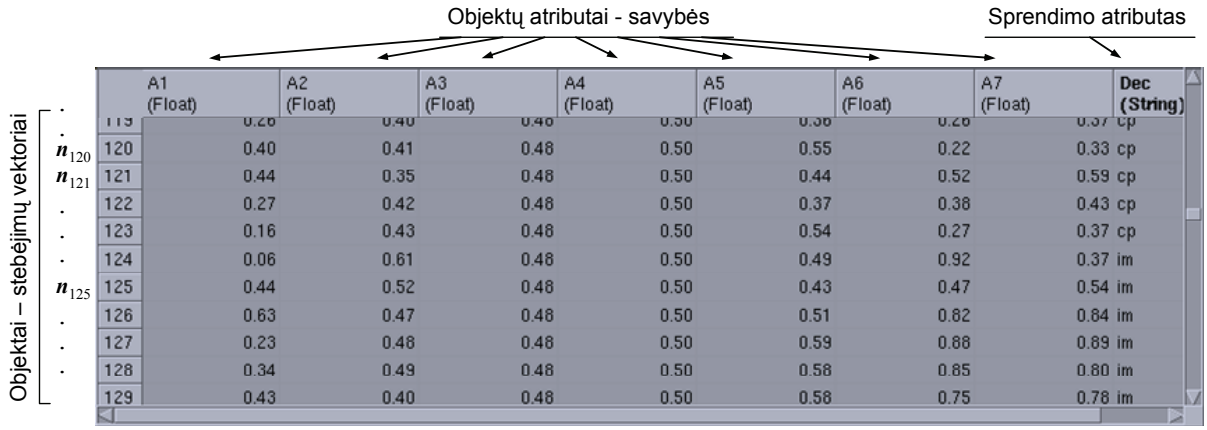
Jungtinio gradientų metodo, Levenberg-Marquardt metodo ir genetinio algoritmo nuosavi metodai *cgm_objective()*, *lmm_objective()*, *objective()* skirti skaičiuoti tikslo funkciją (27), metodai *cgm_gradient()*, *cgm_obj_grad()*, *lmm_gradient()* skirti skaičiuoti gradientus ir Jakobiano matricas išgaubto optimizavimo metodams.

Duomenų klasifikavimas, atlikus Dirichlet mišinių modelio optimizavimą, vykdomas naudojant klasės *GADirichletMixturesClassifier* metodus. Šios klasės paveldimumo ryšys su

genetinio algoritmo apmokymo klase *GADirichletMixturesTrainer* siejasi tik optimizuotų parametrų saugojimo vidiniu formatu. Duomenų klasifikavimas pradedamas *initialize()* ir *apply()* metodais. Maksimalaus tikėtinumo įvertis (27), *posteriorinio* vidurkio įvertis (26) ir multinominio pasiskirstymo dėsnio tikimybė (17) skaičiuojami metodais *probability()*, *posteriorMeanEstimates()*, *posteriorMultinomialProbability()* atitinkamai. Šie įverčiai gaunami kiekvienai sprendimo kategorijai, o didžiausios tikimybės nulemia galutinį klasifikatoriaus spėjimą.

5.2. Klasifikavimo Dirichlet mišiniais algoritmai

Kad galėtume aprašyti klasifikavimo Dirichlet mišiniais algoritmus, pradžioje apibrėšime informacinę sistemą, kurią naudoja projektuojama sistema. Duomenys vaizduojami lentele, kurioje kiekviena eilutė atitinka vieną stebėjimą, vadinamą objektu. Objektas charakterizuojamas stulpeliuose esančių savybių, vadinamų objekto atributais, reikšmėmis. Pažymėkime netuščią baigtinę objektų aibę **U**, o netuščią baigtinę atributų aibę pažymėkime **A**, tada informacinė sistema (lentelė) $\mathbf{I} = (\mathbf{U}, \mathbf{A})$. Kad atlikti duomenų klasifikavimą (klasifikatoriaus apmokymą), vienos informacinės sistemos nepakanka – kiekvienam objektui turi būti žinoma *a priori* informacija: objekto priklausomybės klasifikavimo kategorijoms, vadinamomis sprendimo klasėmis, požymiai. Vienas objektas gali priklausyti vienai ir tik vienai sprendimo klasei, ir tai reiškia, kad matavimai, surinkti objektui, atitinka žinomą klasifikavimo kategoriją.



	A1 (Float)	A2 (Float)	A3 (Float)	A4 (Float)	A5 (Float)	A6 (Float)	A7 (Float)	Dec (String)
120	0.40	0.41	0.48	0.50	0.55	0.22	0.33	cp
121	0.44	0.35	0.48	0.50	0.44	0.52	0.59	cp
122	0.27	0.42	0.48	0.50	0.37	0.38	0.43	cp
123	0.16	0.43	0.48	0.50	0.54	0.27	0.37	cp
124	0.06	0.61	0.48	0.50	0.49	0.92	0.37	im
125	0.44	0.52	0.48	0.50	0.43	0.47	0.54	im
126	0.63	0.47	0.48	0.50	0.51	0.82	0.84	im
127	0.23	0.48	0.48	0.50	0.59	0.88	0.89	im
128	0.34	0.49	0.48	0.50	0.58	0.85	0.80	im
129	0.43	0.40	0.48	0.50	0.58	0.75	0.78	im

9 pav. Sprendimo sistemos pavyzdys. Stebėjimų duomenys vienam objektui užrašomi vienoje eilutėje. Kiekvienas objektas apibrėžiamas globalių visiems objektams savybių aibe, vadinama atributais. Paskutinis atributas – tai sprendimo atributas, nurodantis sprendimo kategoriją, kuriai objektas priklauso. Atributai gali būti trijų tipų: sveikas skaičius (*Integer*), slankaus taško skaičius (*Float*), simbolinė eilutė (*String*).

Informacija *a priori* reikalinga apmokyti klasifikatorių, kuris vėliau sugebėtų atpažinti „nematytus“ objektus (kai *a priori* informacija nėra duota). Pažymėkime papildomą atributą

klasifikavimo kategorijoms $d \notin \mathbf{A}$, kuris vadinamas sprendimo atributu, tuomet išplėstinė informacinė sistema su sprendimo atributu vadinama sprendimo sistema ir žymima $\mathbf{S} = (\mathbf{U}, \mathbf{A} \cup \{d\})$ (žr. skyrių „Klasifikavimo metodai“). Sprendimo sistemos pavyzdys matomas 9 paveikslėlyje.

Sprendimo sistema naudojama klasifikatoriui apmokyti, tuo tarpu informacinė sistema be sprendimo atributo naudojama objektams informacinėje sistemoje klasifikuoti. Dirichlet mišinių klasifikatoriaus apmokymo algoritmas pateiktas žemiau (1 algoritmas).

1 algoritmas. Dirichlet mišinių klasifikatoriaus apmokymas.

Duota: Sprendimo sistema $\mathbf{S} = (\mathbf{U}, \mathbf{A} \cup \{d\})$, galimų sprendimo kategorijų aibė $\mathbf{D}$, Dirichlet mišinio komponentų skaičius C, komponentų pseudo dažnių skaičius F, normalizavimo konstanta *const*.

1 žingsnis: Sudaryti dalines sprendimo sistemas $\{\mathbf{S}_d\}_{d \in \mathbf{D}}$ taip, kad

$$\forall d \in \mathbf{D} : \mathbf{S}_d = \left(\mathbf{X}, \mathbf{A} \cup \{d\}\right) \middle| \mathbf{X} \subseteq \mathbf{U} \wedge \left(d(\mathbf{x}) = d \mid \mathbf{x} \in \mathbf{X}\right)$$

2 žingsnis:

for $\forall d \in \mathbf{D}$ **do**
begin
  for $\forall a \in \mathbf{A}$ **do**
  begin
    if $\min\limits_{\mathbf{x} \in \mathbf{X} | \mathbf{X} \in \mathbf{S}_d} a(\mathbf{x}) < 0$

      then $\forall \mathbf{x} \in \mathbf{X} \mid \mathbf{X} \in \mathbf{S}_d : m_a^d \leftarrow \left| \min\limits_{\mathbf{x}} a(\mathbf{x}) \right|$

      else $\forall \mathbf{x} \in \mathbf{X} \mid \mathbf{X} \in \mathbf{S}_d : m_a^d \leftarrow 0$

    $\forall \mathbf{x} \in \mathbf{X} \mid \mathbf{X} \in \mathbf{S}_d : n_a^d(\mathbf{x}) \leftarrow a(\mathbf{x}) + m_a^d$

    Normalizuoti vektorių $\mathbf{n}_a^d$: $\sum\limits_{\mathbf{x} \in \mathbf{X} | \mathbf{X} \in \mathbf{S}_d} n_a^d(\mathbf{x}) = const$

  end
end

3 žingsnis:

for $\forall d \in \mathbf{D}$ **do**
Begin
  Minimizuoti tikslo funkciją (27):

$$\hat{\Theta}^d \leftarrow \arg\min_{\Theta^d} \left\{ - \sum_{\mathbf{x} \in \mathbf{X} | \mathbf{X} \in \mathbf{S}_d} \log P\left(\mathbf{n}^d(\mathbf{x}) \mid \Theta^d\right) \right\} \Bigg|$$

$$\Theta^d = \left(\left\{ \boldsymbol{\alpha}_j^d \right\}_{j=1}^C, \left\{ q_j^d \right\}_{j=1}^C \right) \wedge \boldsymbol{\alpha}_j^d = \left\{ \alpha_{ji}^d \right\}_{i=1}^F$$

end

Rezultatas: $\left\{\hat{\Theta}^d\right\}_{d \in \mathbf{D}}$, $\left\{\left\{m_a^d\right\}_{a \in \mathbf{A}}\right\}_{d \in \mathbf{D}}$.

Klasifikatoriaus apmokymas susideda iš 3 pagrindinių žingsnių, kurie apibrėžti 5.1. skyriuje „Klasifikatoriaus projektavimas“. Pirmajame žingsnyje sprendimo sistema (lentelė) **S** pagal sprendimo atributo *d* reikšmes išskaidoma į dalines sistemas $\mathbf{S}_d$. Pagal kiekvieną dalinės

sprendimo sistemos $\mathbf{S}_d$ atributą $a \in \mathbf{A}$, kiekvienam objektui $\mathbf{x} \in \mathbf{X} \in \mathbf{S}_d$ sudaromi dažnių vektoriai $\left\{n_a^d(\mathbf{x})\right\}_{a\in\mathbf{A}}$. Dažnių vektorių pjūviai $\mathbf{n}_a^d = \left\{n_a^d(\mathbf{x})\right\}_{\mathbf{x}\in\mathbf{X}|\mathbf{X}\in\mathbf{S}_d}$ pagal sprendimo sistemos $\mathbf{S}_d$ atributus $a \in \mathbf{A}$ tikslinami ir normalizuojami (2 žingsnis). Reikšmių tikslinimas atliekamas, kad įvertinti galimas neigiamas dažnių reikšmes, o normalizavimas naudojamas, kad išvengti didelių reikšmių srities skirtumų dviems pasirinktiems sprendimo sistemos atributams. Šiame žingsnyje taip pat kiekvienam atributui a iš dalinės sprendimo sistemos išsaugojamos minimalios dažnių reikšmės m_a^d, kurios naudojamos duomenų klasifikavime (2 algoritmas). Paskutiniame – 3 žingsnyje – kiekvienai sprendimo lentelei $\mathbf{S}_d$ optimizuojamas Dirichlet mišinių modelis, ir rezultate gaunama optimizuotų mišinių modelių aibė $\left\{\hat{\Theta}^d\right\}_{d\in\mathbf{D}}$.

Optimizuoti modeliai naudojami apmokymuose nenaudotiems objektams klasifikuoti. Klasifikavimo Dirichlet mišiniais žingsniai atsispindi 2 algoritme.

2 algoritmas. Klasifikavimas Dirichlet mišiniais.

Duota: Informacinė sistema **I** = (**U**, **A**) ir objektas $\mathbf{x} \in \mathbf{U}$ iš informacinės sistemos **I**, galimų sprendimo kategorijų aibė **D**, Dirichlet mišinio komponentų skaičius C, komponentų pseudo dažnių skaičius F, normalizavimo konstanta *const*, optimizuotas Dirichlet mišinio modelis kiekvienai sprendimo kategorijai d: $\left\{\hat{\Theta}^d\right\}_{d\in\mathbf{D}}$, minimalių reikšmių atributų aibė $\left\{\left\{m_a^d\right\}_{a\in\mathbf{A}}\right\}_{d\in\mathbf{D}}$.

1 žingsnis: **for** $\forall d \in$ **D do**
begin

1.1 žingsnis: **for** $\forall a \in$ **A do**
begin

$n_a^d(\mathbf{x}) \leftarrow a(\mathbf{x}) + m_a^d$

Normalizuoti reikšmę $n_a^d(\mathbf{x})$: $0 < n_a^d(\mathbf{x}) \le const$

end

1.2 žingsnis: **if** PME **then** {*jei naudoti posteriorinį vidutinį įvertinimą*}

Skaičiuoti *posteriorinio* vidutinio įvertinimo tikimybių vektorių $\hat{\mathbf{p}}^d$ pagal (26):

$$\forall a \in \mathbf{A}: \hat{p}_a^d \leftarrow P_a\left(\mathbf{n}^d(\mathbf{x}) \middle| \hat{\Theta}^d\right) \hat{\Theta}^d = \left(\left\{\hat{\boldsymbol{\alpha}}_j^d\right\}_{j=1}^C, \left\{\hat{q}_j^d\right\}_{j=1}^C\right) \wedge \hat{\boldsymbol{\alpha}}_j^d = \left\{\hat{\alpha}_{ji}^d\right\}_{i=1}^F$$

Skaičiuoti multinominio pasiskirstymo tikimybę pagal (17):

$$\widetilde{P}^d(\mathbf{x}) \leftarrow P\left(\mathbf{n}^d(\mathbf{x}) \middle| \hat{\mathbf{p}}^d\right) \hat{\mathbf{p}}^d = \left\{\hat{p}_a^d\right\}_{a\in\mathbf{A}}$$

else

Skaičiuoti tikimybę pagal (20):

$$\widetilde{P}^d(\mathbf{x}) \leftarrow P\left(\mathbf{n}^d(\mathbf{x}) \middle| \hat{\Theta}^d\right) \hat{\Theta}^d = \left(\left\{\hat{\boldsymbol{\alpha}}_j^d\right\}_{j=1}^C, \left\{\hat{q}_j^d\right\}_{j=1}^C\right) \wedge \hat{\boldsymbol{\alpha}}_j^d = \left\{\hat{\alpha}_{ji}^d\right\}_{i=1}^F$$

end

2 žingsnis: $\hat{d}(\mathbf{x}) \leftarrow \arg\max_d \widetilde{P}^d(\mathbf{x})$

Rezultatas: $\hat{d}(\mathbf{x})$.

Klasifikavimo Dirichlet mišiniais algoritmas aprašo objekto $\mathbf{x} \in \mathbf{U}$ iš informacinės sistemos **I** klasifikavimo logiką. Šiame algoritme skiriami 2 pagrindiniai žingsniai. Pirmame žingsnyje atskirai kiekvienai sprendimo kategorijai $d \in \mathbf{D}$ ir kiekvienam atributui $a \in \mathbf{A}$ atliekamas dažnių $n_a^d(\mathbf{x})$ tikslinimas ir normalizavimas pagal apmokymo etape rastas minimalias dažnių reikšmes m_a^d (1.1 žingsnis) ir dažnių reikšmių srities ribas. Kiekvienai iš sprendimo kategorijų skaičiuojamos objekto priklausomybės toms kategorijoms tikimybės (1.2 žingsnis). Pagal pasirinkimą skaičiavimuose galima naudoti arba *posteriorinį* vidutinį įvertinimą (26), arba dažnių vektoriaus $\mathbf{n}^d(\mathbf{x}) = \{n_a^d(\mathbf{x})\}_{a \in \mathbf{A}}$ tikėtinumo išraišką (20). Galutinis klasifikatoriaus sprendimas atitiks maksimalią sprendimo kategorijos tikimybę (2 žingsnis), t.y. $\hat{d}(\mathbf{x})$ reikš sprendimo kategoriją, kuriai tikimybė, kad objektas **x** priklausys tai kategorijai, gauta didžiausia.

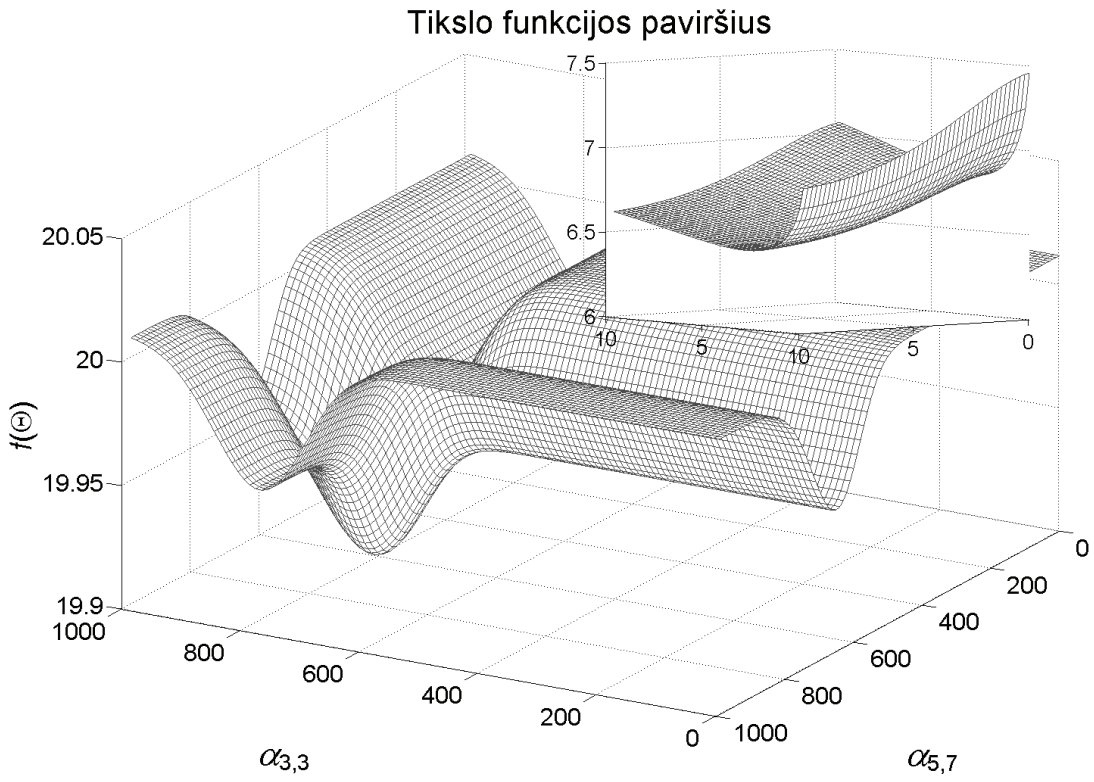
Paprastumo sumetimais, algoritmuose nepateikti pseudo dažnių skaičiaus F parinkimo galimybės ir skaičiavimai, priklausomi nuo F. Pagrindinis reikalavimas, kad pseudo dažnių skaičius F būtų atributų skaičiaus $|\mathbf{A}|$ informacinėje sistemoje **I** kartotinis. Algoritmai pateikti tuo atveju, kai $F = |\mathbf{A}|$. Tokia konfigūracija dažniausiai naudojama klasifikavime Dirichlet mišiniais. Priešingu atveju, jei $F \neq |\mathbf{A}|$ ir $|\mathbf{A}| \bmod F = 0$, atributai grupuojami į mažesnį jų skaičių (F) ir po to sudaromi dažnių vektoriai, kaip yra parodyta algoritmuose. Visuose mūsų atliktuose tyrimuose, naudojome Dirichlet mišinių konfigūraciją, pagal kurią $F = |\mathbf{A}|$.

5.3. Dirichlet mišinio parametrų optimizavimas

Dirichlet mišinių klasifikatoriaus apmokymo algoritme paskutiniame žingsnyje atliekamas Dirichlet mišinio parametrų optimizavimas. Tikslo funkcija (27), kurią reikia optimizuoti norint surasti optimalius Dirichlet mišinių parametrus, yra tolydi ir pakankamai sudėtinga:

a) stebėjimų skaičius, atitinkantis objektų skaičių lentelėje, neribojamas, ir dažnių vektorių gali būti daug;
b) tikslo funkcija skaičiuojama daugiamatėje erdvėje, kurios dydis (matavimų skaičius) priklauso nuo Dirichlet mišinių modelio parametrų skaičiaus.

Pavyzdžiui, jei Dirichlet mišinių modelį apibrėšime susidedantį iš 20 komponentų ir kiekvienas iš komponentų apibrėžiamas 20 pseudo dažnių parametrais $\{\alpha_{ji}\}_{i=1}^{20}$, tuomet iš viso reikės optimizuoti 420 parametrų. Tikslo funkcija turi daug lokalių minimumų (pavyzdys: 10 pav.), ir užduotis surasti globalaus minimumo tašką yra sudėtinga.



10 pav. Tikslo funkcijos (27) paviršius, gautas konkrečiam uždaviniui parinkus Dirichlet mišinių modelį, susidedantį iš 20 komponentų, kurių kiekvienas apibrėžiamas 7 pseudo dažnių parametrais. Paveikslėlyje matyti kaip kinta tikslo funkcija dviejų, atsitiktinai parinktų trečio ir penkto komponentų parametrų $\alpha_{3,3}$ ir $\alpha_{5,7}$ atžvilgiu. Likusieji 158 parametrai nekintami ir parinkti lokalaus minimumo taško kaimynystėje. Paveikslėlio fragmentas viršutiniame dešiniame kampe iliustruoja tikslo funkcijos paviršių dviejų, tų pačių parametrų $\alpha_{3,3}$ ir $\alpha_{5,7}$ atžvilgiu, jiems kintant kito, esančio arčiau koordinačių pradžios, lokalaus minimumo taško kaimynystėje.

Tikslo funkcijos optimizavimui naudosime tris optimizavimo metodus: genetinį algoritmą [47, 48], jungtinį gradientinį (*Conjugate Gradient*) metodą ir Levenberg-Marquardt metodą [49]. Genetinis algoritmas yra kombinatorinio optimizavimo metodas ir geriausiai tinka uždaviniams, kuriems tikslo funkcija yra diskreti arba turi daug lokalių minimumų. Tačiau kiti du metodai, kaip bus matyti iš atliktų tyrimų ir eksperimentų, naudojami ne be reikalo. Kai kuriems uždaviniams [50] genetinis algoritmas konverguoja gana lėtai ir gali prireikti atlikti didelį genetinių iteracijų kiekį. Išgaubto optimizavimo metodus (jungtinio gradiento ir Levenberg-Marquardt) įdiegėme [51], kad nustatytume, kuris optimizavimo metodas labiausiai tinka mūsų problematikai. Išgaubto optimizavimo metodų silpnoji vieta yra tai, jog dažnai globalaus minimumo radimo sėkmė labiausiai priklauso nuo pradinio taško pasirinkimo pradėti iteratyvią paiešką [49]. Kombinatorinio ir išgaubto optimizavimo metodų derinimas gali sąlygoti geresnius rezultatus.

5.4. Dirichlet mišinių klasifikatoriaus realizacija

Dirichlet mišinių klasifikavimo metodui testuoti parinkome kelias skirtingų mokslinių sričių duomenų bazes, kad palyginti ir įsitikinti metodo galimybėmis klasifikuoti bet kokio tipo duomenis. Tačiau klausimas kyla, kaip interpretuoti duomenis, kurių tipai (objektų atributų tipai) skirtingoms sprendimų lentelėms yra kintami dydžiai.

Kiekvienas sprendimo lentelės atributas yra apibrėžiamas konkrečiu duomenų tipu. Atributas gali būti trijų tipų: sveikas skaičius, slankaus taško skaičius, simbolinė eilutė (9 pav.). Slankaus taško skaičiai ir simbolinės eilutės vienareikšmiškai konvertuojami (atspindimi) į sveikus skaičius. Atlikus reikšmių konvertavimą, reikšmių pasiskirstymą sprendimų lentelėje galima laikyti dažnių pasiskirstymu, kuris kiekvienam sprendimų lentelės objektui sudaro dažnių vektorių $\mathbf{n}_c$. Neigiamoms atributų reikšmėms atliekamos tokios transformacijos: visos stulpelio, kuriame yra bent viena neigiama atributo reikšmė, reikšmės tiesiškai transformuojamos taip, kad tame stulpelyje reikšmės įgytų neneigiamas reikšmes. Atributų reikšmių sritis gali būti labai plati, todėl prieš tai, kai bus pradedamas Dirichlet mišinių parametrų optimizavimas, atliekamas atributų reikšmių srities siaurinimas išlaikant kiekvienos atributo reikšmės stulpelyje pradinį santykį su kitomis reikšmėmis (žr. 5.2. skyrių „Klasifikavimo Dirichlet mišiniais algoritmai“). Reikšmių srities siaurinimas atliekamas atskirai kiekvienam sprendimų lentelės stulpeliui. Toks reikšmių srities siaurinimas yra būtinas, kadangi kai kurios didelės atributų reikšmės gali neigiamai paveikti optimizavimo procesą, iškraipant optimizavimo erdvę, ir bendrą klasifikavimo rezultatą. Tai teigia teorija [49], tai patvirtino ir mūsų atlikti eksperimentai su Dirichlet mišinių klasifikatoriumi.

Tiek duomenų skaitymą/rašymą, tiek klasifikavimo Dirichlet mišiniais logiką, aprašytą 1 ir 2 algoritmais (žr. 5.2. skyrių „Klasifikavimo Dirichlet mišiniais algoritmai“), realizavome C++ programavimo kalba. Klasifikatoriaus realizaciją sudaro nepriklausomai veikiantis įrankis (programinė aplinka) ir programinė biblioteka, kuri gali būti naudojama išplėstinėms žinių kaupimo ir klasifikavimo aplinkoms kurti. Tokios bibliotekos ypatumas tame, kad vartotojas (programų sistemų kūrėjas) naudoja Dirichlet mišinių klasifikatoriaus C++ klases ir objektus ir taiko klasės metodus klasifikatorių apmokyti, o vėliau su juo klasifikuoti duomenis. Pavyzdys, kaip gali būti naudojama Dirichlet mišinių klasifikatoriaus C++ biblioteka programų sistemų kūrėjo reikmėms, parodytas 11 ir 12 paveikslėliuose.

Kad pradėti klasifikatoriaus apmokymą (11 pav.), naudojant sukurtą biblioteką, pirmiausia sukuriamas Dirichlet mišinių klasifikatoriaus objektas. Taip pat sukuriamas sprendimo sistemos objektas, kuris bus užpildytas duomenimis skaitant informaciją iš nurodytos bylos. Duomenų skaitymas atliekamas naudojantis tam skirto objekto `importer` metodais. Vėliau pasirenkama Dirichlet mišinių konfigūracija – mišinio komponentų skaičius ir pseudo dažnių

skaičiius komponentuose – ir parametrų optimizavimo metodas. Nustatomi pasirinkto optimizavimo metodo parametrai ir pradedamas Dirichlet mišinių klasifikatoriaus apmokymas.

```
//Sukuriamas Dirichlet mišinių klasifikatorius
Handle<MFAMDirichletClassifier>     classifier =
        dynamic_cast( MFAMDirichletClassifier*,
          ObjectManager::GetIdentifiedAlgorithm( MFAMDIRICHLETCLASSIFIER ));

//Sukuriamas projekto objektas
Handle<Structure> project = Creator::Create( PROJECT );

//Sprendimo sistemos importavimo algoritmas
MyDecisionTableImporter importer;

//Sukuriamas sprendimo sistemos objektas
Handle<KSDecisionTable> table =
        dynamic_cast( ::KSDecisionTable*,
          Creator::Create( KSDECISIONTABLE, project.GetPointer(), true ));

//Nustatomas bylos, kurioje saugoma sprendimo sistema, vardas
importer.SetFilename( String( "sprendimo_sistema.txt" ));

//Importuojama sprendimo sistema
if( importer.Apply( *table.GetPointer()) == NULL )
     //Sprendimo sistemos importuoti nepavyko
     return false;

//Nustatoma mišinio konfigūracija: komponentų ir parametrų skaičius
classifier->SetDMNoComponents( 9 );
classifier->SetDMNoAlphas(    20 );

//Pasirenkamas optimizavimo metodas: Levenberg-Marquardt
classifier->SetOptimizationAlgorithm( MFAMDirichletClassifier::OP_LMM );

//Nustatomi optimizavimo metodo parametrai
classifier->SetLMMInitialTau( 1e-3 );
classifier->SetLMMIterations( 1000 );
//Nustatomi kiti parametrai...

//Pradedamas Dirichlet mišinių klasifikatoriaus apmokymas
if( !classifier->Train( table, false ))
     //Klaida apmokant klasifikatorių
     return false;
```

11 pav. Dirichlet mišinių klasifikatoriaus apmokymo pavyzdys naudojant sukurtą klasifikavimo C++ programinę biblioteką.

Apmokyto Dirichlet mišinių klasifikatoriaus panaudojimas duomenims (sprendimo objektams) klasifikuoti, naudojant sukurtą biblioteką, yra toks pat nesudėtingas procesas (12 pav.). Jei sprendimo objektai, kurie turi būti klasifikuojami, skaitomi iš kitos sprendimo sistemos, nei iš tos, pagal kurią buvo apmokytas klasifikatorius, sprendimo sistema turi būti patikrinta, ar yra suderinama su klasifikatoriumi. Jei, pavyzdžiui, skirsis sprendimo sistemų atributų skaičius ar bent vienas jų tipas, klasifikatorius ir testuojama sprendimo sistema bus nesuderinami.

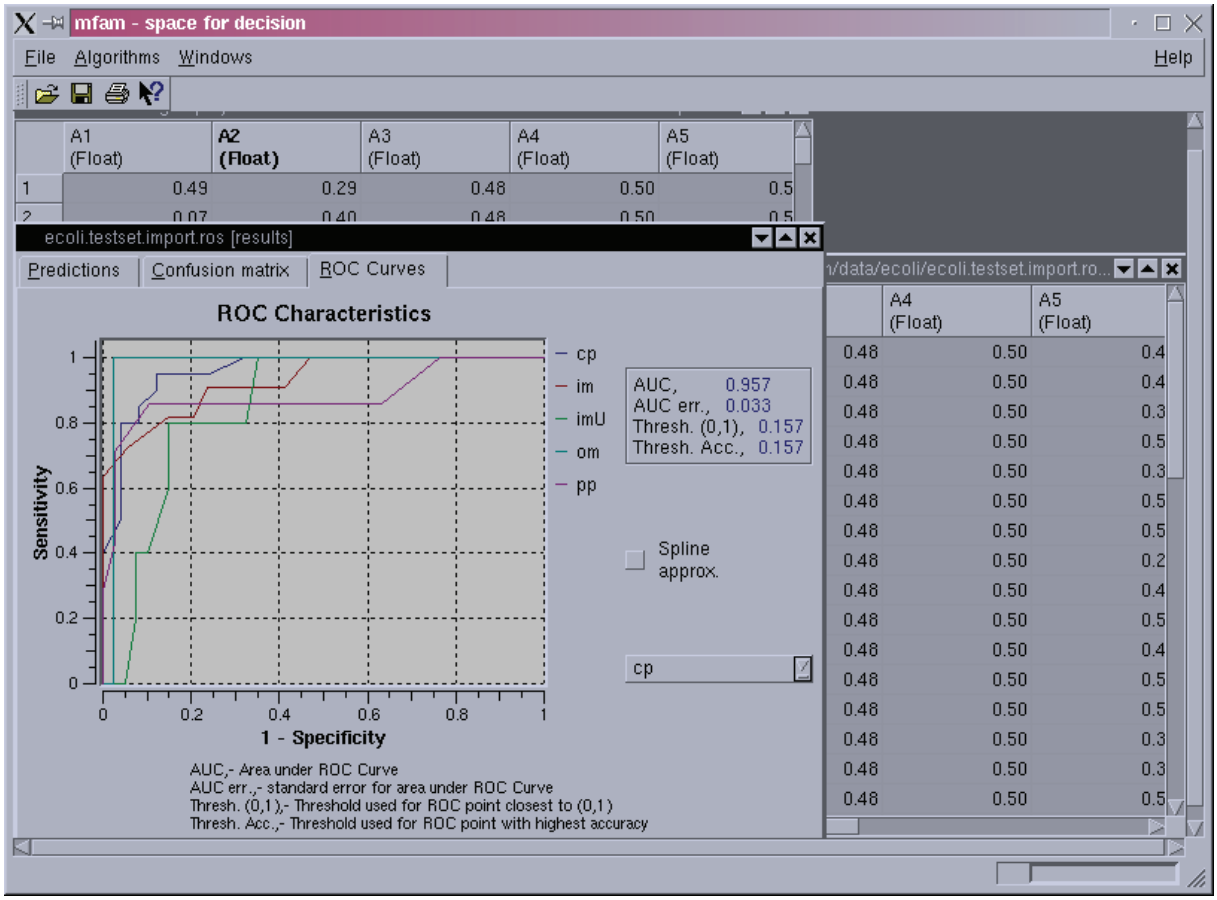
Sprendimo sistema skaitoma panašiu būdu, kaip klasifikatoriaus apmokymo atveju, ir dalis programinio teksto 12 paveikslėlyje, paprastumo ir aiškumo sumetimais, praleista. Sprendimo objektai iš sprendimo sistemos sudaromi po vieną ir kiekvienam iš jų pateikiamas klasifikatoriaus spėjimas. Kiekvienai iš sprendimo kategorijų pateikti klasifikatoriaus spėjimai gali būti toliau įvairiai apdorojami.

```
//Turime klasifikatoriaus objektą ‚classifier' (žr. aukščiau)
//Klasifikatoriaus ‚classifier' optimalūs parametrai rasti
//...

//Sprendimo sistemos objektas
Handle<KSDecisionTable>  test_table;

//Sukurti sprendimo sistemos objektą panašiai kaip klasifikatoriaus
//apmokymo atveju...

//Tikrinama, ar klasifikatorius suderinamas su importuota sprendimo sistema
if( !classifier->IsCompatible( *test_table, false ))
     //Klasifikatorius ir sprendimo sistema nesuderinami
     return false;

//Inicializuojamas klasifikatorius
if( !classifier->Initialize( *test_table, false ))
     //Klasifikatorius negali būti inicializuojamas
     return false;

//Sukuriamas vienas sprendimo objektas, kuriam bus gražinamas rezultatas
Handle<InformationVector>  inf = Creator::InformationVector();

//Objektų skaičius sprendimo sistemoje
int  no_objects = test_table->GetNoObjects( false );

//Klasifikuojami visi sprendimo objektai, esantys sprendimo sistemoje
for (i = 0; i < no_objects; i++)
{
     //Inicializuojamas sprendimo objektas
     if( !inf->Create( *test_table, i, false ))
        return false;

     //Atliekamas sprendimo objekto klasifikavimas
     Handle<Classification> result =
        dynamic_cast( Classification *, inf->Apply( *classifier ));

     //Apdorojamas rezultatas ‚result',- visi klasifikatoriaus spėjimai
     //(objekto priklausomybės kiekvienai sprendimo kategorijai tikimybės)
}
```

12 pav. Dirichlet mišinių klasifikatoriaus taikymo pavyzdys duomenims klasifikuoti, naudojant sukurtą klasifikavimo C++ programinę biblioteką.

Dirichlet mišinių klasifikatoriaus kaip įrankio realizacija – tai vartotojui skirtas galutinis programinis produktas. Sukurtoje klasifikavimo aplinkoje (13 pav.) vartotojas gali skaityti iš bylų, rašyti į bylas sprendimo sistemas, modifikuoti sprendimo sistemos reikšmes, konfigūruoti Dirichlet mišinių klasifikatorių, parinkti optimizavimo metodą klasifikatoriui apmokyti, nustatyti pasirinkto optimizavimo metodo parametrus, apmokyti klasifikatorių

naudojant pasirinktą optimizavimo metodą, testuoti klasifikatorių, t.y. klasifikuoti apmokymuose nenaudotus sprendimo (duomenų) objektus, tirti klasifikavimo rezultatus: analizuoti gautus spėjimų reikšmių pasiskirstymus, nesutapimų matricas, braižyti ROC kreives, išsaugoti klasifikavimo rezultatus bylose įvairia forma: galima išsaugoti ROC kreivių taškus, nesutapimų matricas ir klasifikatoriaus spėjimus.



13 pav. Sukurtos Dirichlet mišinių klasifikatoriaus programos veikimo pavyzdys. Vartotojas gali duomenis skaityti iš bylų ir rašyti į jas, apmokyti klasifikatorių, klasifikuoti duomenis ir analizuoti rezultatus jam palankia forma: paveikslėlyje parodyta ROC kreivių aibė vienam iš spręstų klasifikavimo uždavinių.

Dirichlet mišinių klasifikatoriaus grafinė aplinka sukurta kol kas Linux operacijų sistemai, tačiau klasifikatoriaus programinė biblioteka yra pernešama ir gali būti naudojamas tiek Unix šeimos operacijų sistemose, tiek Windows. Be to, grafinė klasifikatoriaus aplinka sukurta naudojantis pernešama C++ grafine biblioteka Qt, todėl Dirichlet mišinių klasifikatoriaus programinio įrankio pernešimas ir pritaikymas Windows operacijų sistemai yra nesudėtingas procesas.

5.5. Išvados

Dirichlet mišinių klasifikatorius suprojektuotas bet kokio tipo duomenims klasifikuoti. Duomenys kinta priklausomai nuo sprendžiamo uždavinio specifikos. Duomenų vaizdavimas informacinėmis ir sprendimo sistemomis aiškiai padalina informacinius vienetus į atskiras grupes. Sprendimo sistemos objektai charakterizuojami atributais, kurių tipas gali būti sveikas skaičius, slankaus taško skaičius arba simbolinė eilutė. Dirichlet mišinių klasifikatorius suprojektuotas taip, kad teisingai interpretuotų išvardintus duomenų tipus. Tam tikslui atliekamos atributų reikšmių transformavimo operacijos. Greta transformavimo operacijų, atributų reikšmių srities siaurinimas ir normalizavimas yra būtini skaičiavimai, kad užtikrinti optimizavimo metodų gerą veikimą. Dirichlet mišinių klasifikatoriaus parametrams optimizuoti naudojami trys optimizavimo metodai: jungtinio gradiento, Levenberg-Marquardt metodai ir genetinis algoritmas. Kiekvienas iš optimizavimo metodų turi savų privalumų ir trūkumų, todėl, priklausomai nuo nagrinėjamos uždavinio problematikos, pasirenkamas labiausiai tinkamas nagrinėjamam uždaviniui optimizavimo metodas. Dirichlet mišinių klasifikatorius sukurtas kaip nepriklausomai veikiantis programinis produktas ir kaip programinė biblioteka. Naudojant bibliotekos realizaciją, programų sistemų kūrėjas savo programiniame tekste (kode) nesunkiai gali panaudoti Dirichlet mišinių klasifikatorių duomenims klasifikuoti. Dirichlet mišinių klasifikatoriaus programinio produkto realizacija – tai galutinis su grafine aplinka produktas, skirtas vartotojui, iš kurio nereikalaujama nei programavimo, nei matematikos ir statistikos žinių.

6. EKSPERIMENTAI, REZULTATAI IR DISKUSIJOS

Klasifikavimo Dirichlet mišiniais našumo testavimui pasirinkome tris duomenų bazes iš Kalifornijos universiteto klasifikavimo duomenų archyvo [52]; šie duomenys: širdies aritmijos duomenų bazė, *E.Coli* baltymų klasifikavimo duomenų bazė, radaro signalų iš jonosferos duomenų bazė. Visais trim atvejais duomenys atsitiktinai buvo padalinami į nepersikertančias apmokymo ir testavimo duomenų aibes. Dirichlet mišinių klasifikatorius apmokomas pagal apmokymo duomenų aibę ir testuojamas duomenimis iš testavimo duomenų aibės. Apmokymo ir testavimo duomenų aibių santykiai aritmijos, *E.Coli* ir jonosferos duomenims sudarė 90:10, 86:14 ir 86:14 procentus atitinkamai. Testavimo duomenų aibės visais atvejais buvo surenkamos taip, kad kiekvienos klasifikavimo kategorijos duomenys testavimo aibėje sudarytų vienodą procentą nuo bendro tos klasifikavimo kategorijos duomenų kiekio duomenų bazėje. Vadinasi, jei jonosferos duomenų bazėje yra dvi klasifikavimo kategorijos: ‚g' ir ‚b',– testavimo duomenų aibėje duomenų iš šių klasifikavimo kategorijų bus po 14%.

Klasifikavimo Dirichlet mišiniais našumą vertinome ROC charakteringomis kreivėmis [29]. ROC kreivės apibūdina klasifikavimo tikslumą aibe taškų, atitinkančių skirtingus jautrumo ir specifiškumo lygius. Specifiškumas nurodo, kaip tiksliai (kokį procentą) klasifikatorius sugebėjo atskirti klasifikavimo kategorijai nepriklausančius objektus. Jautrumas nurodo, kaip tiksliai (kokį procentą) klasifikatorius sugebėjo atpažinti klasifikavimo kategorijai priklausančius objektus.

Kad įvertinti koks Dirichlet mišinių klasifikatoriaus našumas kitų metodų atžvilgiu, ROC kreives braižėme ir klasifikavimo metodams, su kuriais lyginome pasiūlytą klasifikatorių. Metodai, su kuriais lyginome Dirichlet mišinių klasifikatorių, - tai klasifikavimas taisyklėmis, sugeneruotomis griežtųjų aibių skaičiavimais, ir naivus Bayeso metodas. Visiems klasifikavimo metodams naudojome tas pačias duomenų imtis: tiek apmokymams, tiek testavimui. Prieš duomenis klasifikuojant taisyklėmis, jie diskretizuojami ir redukuojami griežtųjų aibių skaičiavimo kontekste, kad klasifikavimo taisyklės taptų kuo bendresnės ir atspindėtų realų objektų įvairumą kiekvienai klasifikavimo kategorijai. Šio tipo klasifikavimui ROC kreivės nubraižytos pagal geriausius gautus rezultatus, taikant įvairius diskretizavimo ir redukavimo algoritmus. Duomenų diskretizavimą atlikome taip pat ir naivaus Bayeso klasifikavimo metodo atveju: tai užtikrindavo geresnius rezultatus nei tuo atveju, jei nebūtų taikytas joks diskretizavimo algoritmas. ROC kreivės nubraižytos taip pat remiantis geriausiais klasifikavimo rezultatais. Klasifikavimo Dirichlet mišiniais atveju, nebuvo taikyti jokie diskretizavimo ir redukavimo algoritmai; tokiu būdu norėjome įsitikinti kaip

klasifikatorius sugeba atpažinti nematytus objektus, neatliekant jokių duomenų reikšmes iš esmės pakeičiančių transformacijų.

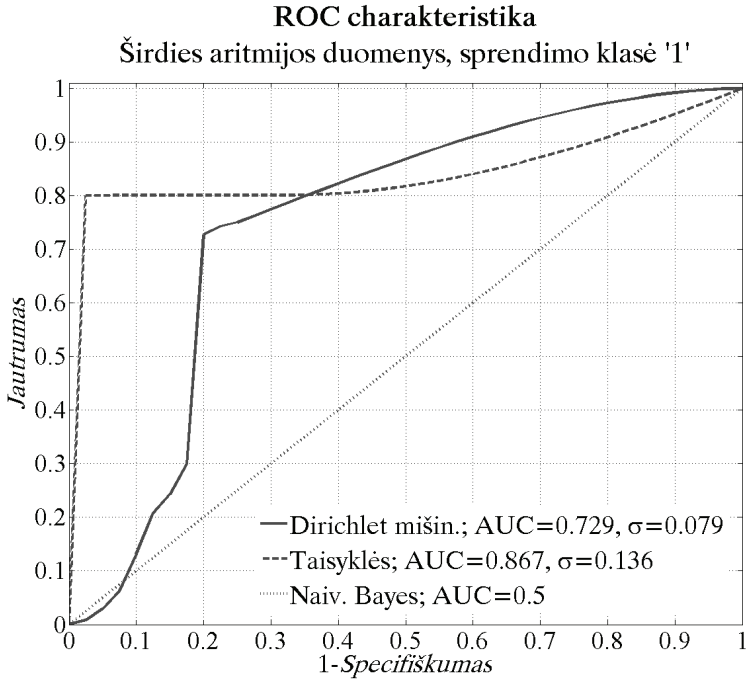
6.1. Širdies aritmijos duomenys

Šie duomenys surinkti [53] širdies aritmijos diagnozei atlikti. Šešiolika klasifikavimo kategorijų apibrėžia skirtingus širdies aritmijos lygius, pirma kategorija – „normali“. Duomenis sudaro 279 atributai: paciento amžius, lytis, aukštis, svoris, širdies dūžiai per minutę, kiti specifiniai rodikliai. Duomenų bazėje duomenys surinkti iš 452 pacientų (objektų skaičius). Kai kuriems objektams trūksta atributų reikšmių (0.33%).

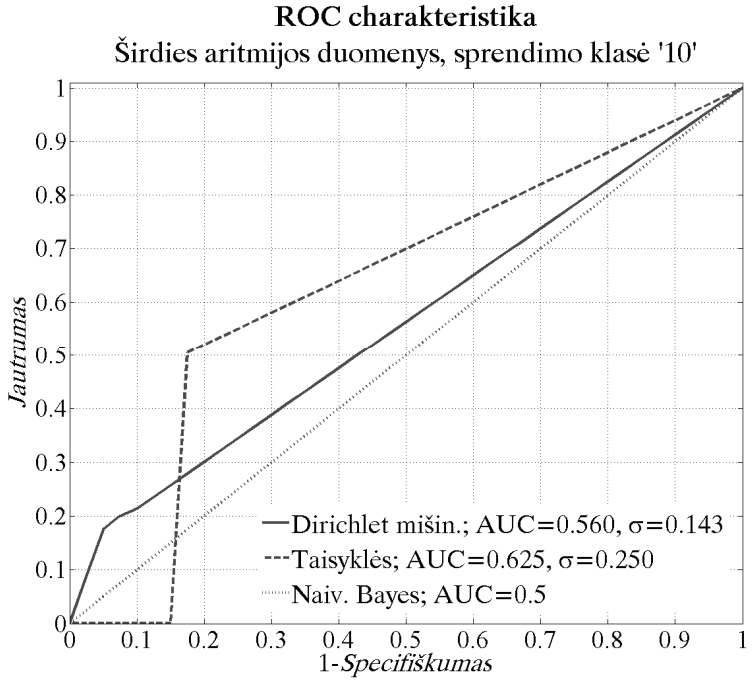
Dirichlet mišinių klasifikatorių apmokėme jungtinio gradiento metodu, Levenberg-Marquardt metodu ir genetiniu algoritmu. Tačiau tik genetiniu algoritmu gavome tenkinamus rezultatus, kiti du metodai nekonvergavo arba „įstrigdavo“ į netenkinamą lokalaus minimumo tašką. Analizuodami tikslo funkcijos paviršių, pastebėjome, jog paviršius yra gana plokščias, ir tai galėjo būti pagrindinė priežastis, kodėl išgaubto optimizavimo metodų taikymas būdavo nesėkmingas.

Klasifikuojant duomenis, išbandėme įvairius Dirichlet mišinių modelio derinius ir nustatėme, kad geriausiai šiems duomenims tinka paprastas (vieno komponento) Dirichlet klasifikatorius. Todėl Dirichlet mišinio modelis buvo sukomponuotas iš vieno komponento (Dirichlet skirstinys), priklausančio nuo 279 pseudo dažnių parametrų $\{\alpha_{1,i}\}_{i=1}^{279}$. Dirichlet modelis buvo optimizuojamas 200 genetinio algoritmo iteracijomis, taikant lyginio-nelyginio genų kryžminimo algoritmą ir genų pozicijų mutacijos algoritmą. Kiti genetinio algoritmo parametrai: kryžminimo tikimybė parinkta 0.9, mutacijos tikimybė lygi 0.1, populiacijos dydis – 60, vienoje kartoje (iteracijoje) individų pakeitimo skaičius – 9. Mutacijos tikimybė yra gana didelė, kadangi stengėmės simuliuoti greitąjį mutavimą ir sumažinti genetinių iteracijų skaičių. Didesnis iteracijų skaičius dažniausiai sąlygoja geresnius rezultatus, tačiau pabandėme optimizuoti Dirichlet modelį daugiau nei 2000 iteracijomis su mutavimo tikimybe 0.01 ir gavome panašius rezultatus. Todėl kai kuriais atvejais, nusprendėme naudoti „greitos mutacijos“ algoritmą, atliekant mažiau skaičiavimo iteracijų ir, atitinkamai, paspartinant skaičiavimus.

Žiūrint į klasifikavimo ROC kreives (14 pav.) matyti, kad klasifikavimai taisyklėmis ir Dirichlet mišiniais dviem klasifikavimo kategorijoms (‚1‘ ir ‚10‘) nedaug skiriasi. Naivaus Bayeso metodas šiems duomenims nepasiteisino: ploto po ROC kreive (AUC) reikšmė, lygi 0.5, tapati atsitiktinio klasifikavimo AUC reikšmei.

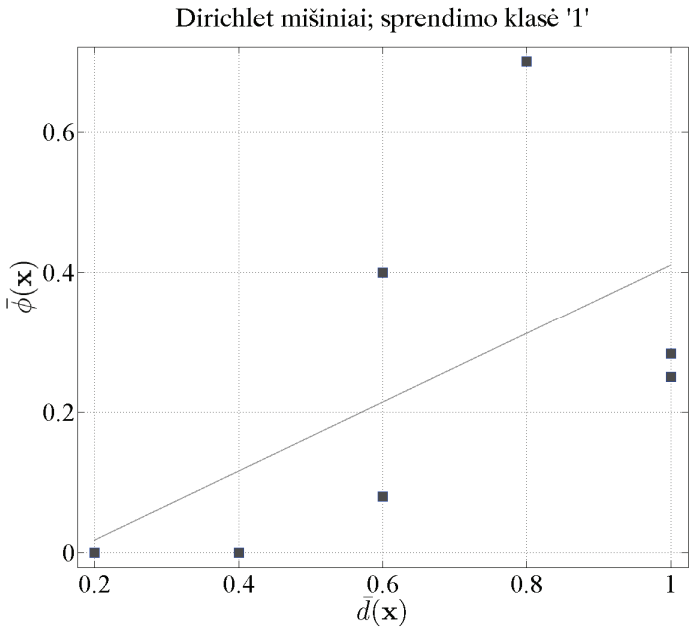


a)

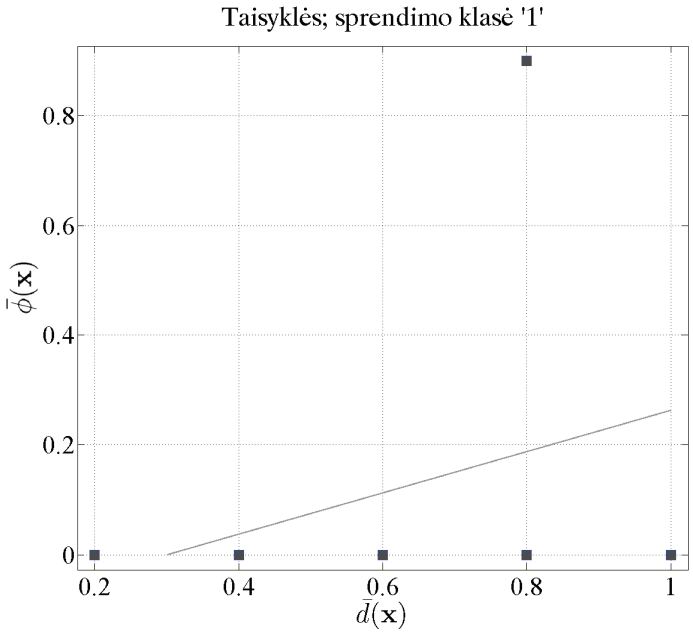


b)

14 pav. Trijų klasifikavimo metodų: klasifikavimo Dirichlet mišiniais (Dirichlet mišin.), klasifikavimo griežtųjų aibių skaičiavimais (Taisyklėmis) ir naivaus Bayeso (Naiv. Bayes) metodo – ROC kreivės. Charakteristika ROC kreivėmis atspindi širdies aritmijos duomenų klasifikavimo našumą klasifikavimo kategorijoms ‚1‘ (a) ir ‚10‘ (b). Plotas po ROC kreive (AUC) išreiškia klasifikavimo tikslumą viena reikšme; parametras σ žymi ploto po kreive skaičiavimo kvadratinę paklaidą.



a)



b)

15 pav. Kalibravimo brėžiniai, gauti duomenis klasifikuojant a) Dirichlet mišinių klasifikatoriumi ir b) taisyklėmis, sprendimo kategorijai ‚1'. Grupių, atspindinčių taškų kiekį brėžinyje, skaičius lygus 8. Abscisės ašyje atidėti vidutiniai tikrųjų sprendimo kategorijų reikšmių (grupėje) vidurkiai, ordinatėje – klasifikatoriaus spėjimų reikšmių vidurkiai. Tiesi linija atitinka tiesinės regresijos modelį duotiems taškams.

Pagal AUC reikšmes Dirichlet klasifikatorius šiek tiek nusileidžia klasifikavimui taisyklėmis, nors Dirichlet klasifikavimo tikslumas kategorijoms ‚1' ir ‚10' sudaro 75% ir 20% atitinkamai. Klasifikavimo taisyklėmis tikslumas šioms kategorijoms tesiekia 17% ir 0%. Bendras (visoms kategorijoms) klasifikavimo tikslumas Dirichlet mišiniu yra 47.5%, taisyklėmis – 10%. Klasifikavimo taisyklėmis atveju, ROC kreivės nubraižytos

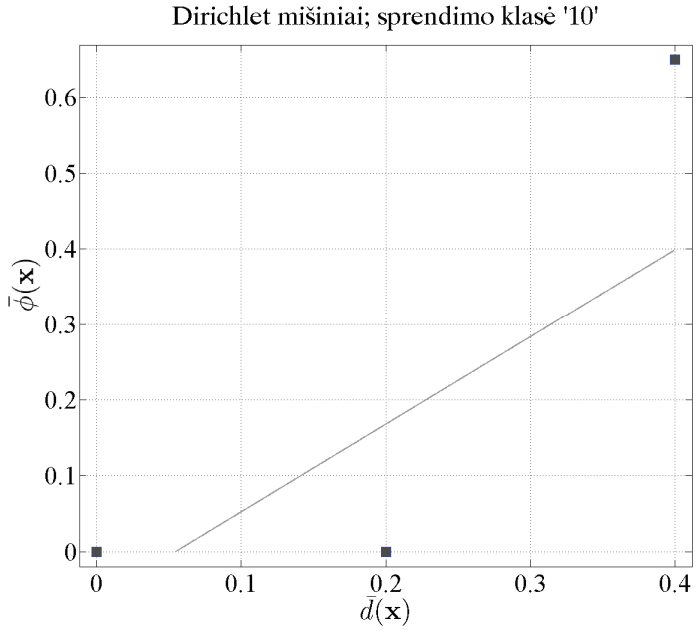
neįskaičiuojant tų atvejų, kai nei viena taisyklė nepateikdavo bet kokio sprendimo. Tuo tarpu Dirichlet mišinių klasifikatoriaus spėjimai įskaičiuojami visais atvejais, neatsižvelgiant į jų reikšmes. Toks rezultatų interpretavimas pakelia taisyklių klasifikavimo ROC našumą Dirichlet mišinių atžvilgiu, tačiau kalibravimo brėžiniai ir charakteristikos labiau atskleidžia Dirichlet mišinių klasifikatoriaus pranašumą.

2 lentelė. Dirichlet mišinių klasifikatoriaus ir klasifikavimo taisyklėmis spėjimų statistiniai parametrai sprendimo kategorijai ‚1‘.

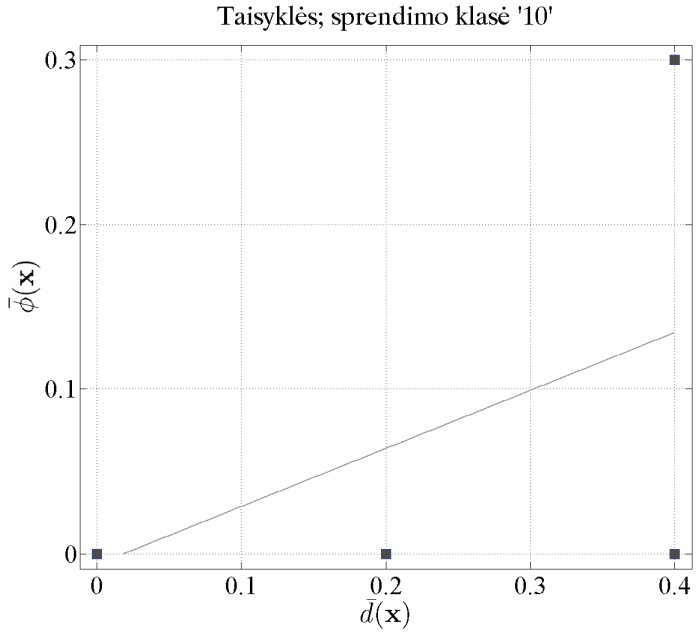
	Dirichlet mišiniai	**Taisyklės**
Brierio įvertis	0.3830	0.5063
Vidutinė sprendimų reikšmė	0.6	0.6
Vidutinė spėjimų reikšmė	0.2142	0.1125
Vidutinė spėjimų reikšmė neigiamiems objektams	0.1271	0.0313
Vidutinė spėjimų reikšmė teigiamiems objektams	0.2722	0.1667
Spėjimų reikšmių dispersija neigiamiems objektams	0.0762	0.0156
Spėjimų reikšmių dispersija teigiamiems objektams	0.0524	0.1449
Vidutinių spėjimų reikšmių riba	0.1451	0.1354
Teigimų ir neigiamų spėjimų reikšmių išsibarstymas	0.0619	0.0932
Klasifikatorių spėjimų koreliacija teigiamiems objektams	-0.0443	
Klasifikatorių spėjimų koreliacija neigiamiems objektams	-0.1228	

Dirichlet mišinių klasifikatoriaus Brierio įvertis (10) sprendimo kategorijai ‚1‘ lygus 0.38; Brierio įvertis klasifikavimo taisyklėmis atveju šiai kategorijai lygus 0.51. Tai rodo, kad Dirichlet mišinių spėjimai yra labiau suderinami su tikrųjų sprendimo kategorijų reikšmėmis, ir spėjimų reikšmės tendencingai didėja, didėjant sprendimo kategorijų reikšmėms. Tai iliustruoja 15 paveikslėlis. Kalibravimo brėžiniai sudaryti taip, kad sprendimo kategorijų ir klasifikatoriaus spėjimų porų skaičius grupėje būtų ne mažesnis nei 5. Tokiu būdu buvo gautos 8 grupės ir kiekvienai iš jų apskaičiuotos vidutinės sprendimo kategorijų reikšmės $\bar{d}(\mathbf{x})$ ir vidutinės spėjimų reikšmės $\bar{\phi}(\mathbf{x})$. Sprendimo kategorijų reikšmės grupėse gali įgauti reikšmes iš {1,0}, čia 1 reiškia kategoriją, kurios atžvilgiu atliekamas tyrimas (objektai, papuolantys į atitinkamą kategoriją, vadinami teigiamais), 0 – bet kokia kita kategorija (objektai iš bet kurios kitos kategorijos vadinami neigiamais). Kiti svarbesni klasifikatorių spėjimų parametrai pateikti 2 lentelėje.

Klasifikatorių spėjimų reikšmės 2 lentelėje parodo, jog Dirichlet mišinių spėjimų reikšmių dispersija yra mažesnė, skirtumas tarp dispersijų mažesnis, kuris reiškia didesnę reikšmių skiriamąją ribą atpažįstant sprendimo kategorijai priklausančius ir nepriklausančius objektus.



a)



b)

16 pav. Kalibravimo brėžiniai, gauti duomenims taikant a) Dirichlet mišinių klasifikatorių ir b) taisykles, sprendimo kategorijai ‚10'. Grupių, atspindinčių vieną tašką brėžinyje, dydis lygus 8. Abscisėje atidėti vidutiniai tikrųjų sprendimo kategorijų reikšmių vidurkiai, ordinatėje – klasifikatoriaus spėjimų reikšmių vidurkiai.

Tai patvirtina 15 paveikslėlis, iš dalies tai atspindi vidutinių spėjimų reikšmių riba, kuri skaičiuojama vidutinių spėjimų reikšmių teigiamiems ir neigiamiems objektams skirtumu. Teigiamų ir neigiamų spėjimų reikšmių išsibarstymas išreiškia spėjimų „triukšmą", kuris skaičiuojamas pagal $(\sigma_0^2 + \sigma_1^2) / (n_0 + n_1)$, čia σ_0^2 ir σ_1^2 – spėjimų dispersija neigiamiems ir teigiamiems objektams atitinkamai, o n_0 ir n_1 – atitinkamai neigiamų ir teigiamų objektų

kiekiai testavimo aibėje. Kuo išsibarstymas mažesnis tuo klasifikatoriaus spėjimai yra labiau kryptingi. Tiesa, kai kuriais atvejais, išsibarstymas gali neatspindėti klasifikatoriaus spėjimų kryptingumo, pavyzdžiui, jei visos spėjimų reikšmės bus vienodos.

Kad nustatyti Dirichlet mišinių ir klasifikavimo taisyklėmis našumo skirtumo reikšmingumą, apskaičiavome šiems klasifikatoriams Pearsono koreliacijos koeficientus neigiamai ir teigiamai objektų imtims. Tačiau matyti, jog abejais atvejais koreliacija yra neigiama, o tai reiškia, kad vieno klasifikatoriaus spėjimų santykinai didelės reikšmės koreliuoja su kito klasifikatoriaus mažomis reikšmėmis (matyti 15 pav.) ir klasifikatorių našumo reikšmingo skirtumo apskaičiuoti neįmanoma. Iš kitos pusės, ploto po ROC kreivėmis, gauto taikant taisykles, skaičiavimo kvadratinės paklaidos σ (12) yra didesnės nei gautos Dirichlet mišiniams (14 pav.) – klasifikavimo taisyklėmis našumo vertinimas vien tik pagal ROC kreives negali būti tikslus.

3 lentelė. Dviejų klasifikatorių klaidingų ir teisingų spėjimų lentelė; ϖ_1 - Dirichlet mišiniai, ϖ_2 – taisyklės.

	ϖ_1 *klaidos*	ϖ_1 *teisingi spėjimai*
ϖ_2 *klaidos*	20	16
ϖ_2 *teisingi spėjimai*	1	3

Panašūs rezultatai gauti kategorijai ‚10‘ (16 pav.), tik dėl mažos testavimo aibės apimties šiai kategorijai (5 objektai), parametrų skirtumai mažesni. Koreliacijos koeficientai dviems klasifikatoriams šiai kategorijai taip pat gauti neigiami, kurie patvirtina klasifikavimų santykio pastovumą šioms kategorijoms.

Dviejų klasifikatorių tikslumų skirtumų reikšmingumui įvertinti (3 lentelė) naudojome McNemaro statistiką (11), kuri parodė, jog McNemaro dydžo reikšmei χ^2=11.53 *P*-reikšmė, arba tikimybė, jog atsitiktinai bus gauta tokia arba didesnė reikšmė, lygi 6.8*e*-4. Vadinasi Dirichlet mišinių klasifikatoriaus tikslumo pranašumas taisyklių atžvilgiu yra *reikšmingas*.

Objektų iš klasifikavimo kategorijos ‚1‘ duomenų imtyje buvo 245, o objektų iš kategorijos ‚10‘ buvo 50. Testavimo imtys sudarytos tokiais santykiais: ‚1’ kategorijai priklausančių objektų skaičius lygus 24, ‚10‘ – jis lygus 5; kitose kategorijose objektų dar mažiau. Todėl reikšmingesni rezultatai gauti kategorijai ‚1‘: klasifikavimo tikslumo reikšmės Dirichlet mišiniams yra 75%, taisyklėms – 17%. Bendras klasifikavimo tikslumas nėra aukštas, bet galima daryti išvadą, jog šie duomenys yra komplikuoti automatiniams klasifikavimo metodams. Duomenų autoriai pažymi [53], kad jų naudotas metodas, paremtas intervalų diskretizavimo skaičiavimais, 10 iteracijų iteratyvaus validavimo (*cross-validation*)

procedūroje [28] pasiekia neaukštą 62% klasifikavimo tikslumą. Taip pat jie pažymi gana dažnus eksperto sprendimų ir klasifikatoriaus prognozių nesutapimus.

6.2. *E.coli* duomenys

Duomenų bazė skirta baltymų klasifikavimui į kategorijas, apibrėžiamas bakterijos *E.Coli* organelėmis, kuriose funkcionuoja baltymai [54]. Aštuonios klasifikavimo kategorijos nusako skirtingas ląstelės terpes, kuriose gali funkcionuoti baltymai: citoplazma (cp), vidinė membrana be jokių požymių apie amino rūgščių sekas (im), periplazma (pp), vidinė membrana, papildyta požymiais apie neperskeltas sekas (imU), ne lipoproteinų išorinė membrana (om), lipoproteinų išorinė membrana (omL), lipoproteinų vidinė membrana imL, vidinė membrana, papildyta požymiais apie perskeltas sekas (imS). Duomenis sudaro 7 klasifikavimui naudojami atributai: įvairi baltymų sekų informacija bei sekų analizės ir sekų atpažinimo metodų įverčiai. Objektai (baltymų informaciniai vektoriai) duomenų bazėje sudaro 336 įrašus. Objektams trūkstamų atributų reikšmių nėra.

Kaip ir širdies aritmijos atveju, išbandėme tris optimizavimo metodus: jungtinį gradiento, Levenberg-Marquardt metodus ir genetinį algoritmą. Priešingai nei širdies aritmijos duomenims, Levenberg-Marquardt metodas pasiteisino labiausiai, nors genetiniu algoritmu gauti artimi rezultatai. Naudojome ir bandėme įvairius Dirichlet mišinių modelius ir nustatėme, kad mišinys su 20 komponentų, kiekviename po 7 pseudo dažnių parametrus, buvo vienas iš tų, kurie geriausiai tiko šiai užduočiai. Gali būti, kad, jei būtume naudoję daugiau nei keletą tūkstančių genetinio algoritmo iteracijų, genetiniu algoritmu optimizuotas modelis galėjo būti tikslesnis ir sąlygoti tikslesnį klasifikavimą. Tačiau apsiribojome keletu tūkstančių genetiniu iteracijų, matydami, kad konvergavimas yra gana lėtas ir tęsėme tyrimus su Levenberg-Marquardt metodu, kuris šiems duomenims dirbo geriausiai.

Kadangi Levenberg-Marquardt metodas yra kvazi-Newtono metodas, kuris remiasi funkcijų išvestinių skaičiavimais, Jakobianas, tikslo funkcijos (27) dalinių išvestinių matrica, turi būti skaičiuojamas kiekvienoje metodo iteracijoje. Tikslo funkcijos (27) dalinės išvestinės, parametrų α_{ji} atžvilgiu, išreiškiamos (pilnas išvedimas gali būti randamas [16]):

$$\frac{\partial f(\Theta)}{\partial \alpha_{j,i}} = -\sum_{c=1}^{N} P(\boldsymbol{\alpha}_j \mid \mathbf{n}_c, \Theta)\left(\Psi(|\boldsymbol{\alpha}_j|) - \Psi(|\mathbf{n}_c| + |\boldsymbol{\alpha}_j|) + \Psi(n_{c,i} + \alpha_{j,i}) - \Psi(\alpha_{j,i})\right). \qquad (28)$$

Išraiškoje (28) naudojamas žymėjimas $\Psi(z) = \Gamma'(z) / \Gamma(z)$, kuris turi savo pavadinimą – tai *digamma* funkcija nuo argumento z. Kad apskaičiuoti dalinės išvestines, parametrų q_j atžvilgiu, įvedamas dydis Q_j ir naudojamas normalizavimas $q_j = Q_j / |Q|$, kad užtikrinti mišinio koeficientų (tikimybių) sumos lygybę 1; čia dydis $|Q| = \Sigma_j Q_j$. Kadangi dydžiai q_j ir Q_j tiesiškai

susiję, dalinių išvestinių skaičiavimas galimas Q_j atžvilgiu, tuo pačiu užtikrinant, kad reikiamos q_j reikšmės nesunkiai randamos žinant Q_j. Dalinės išvestinės Q_j atžvilgiu:

$$\frac{\partial f(\Theta)}{\partial Q_j} = \frac{N}{|Q|} - \frac{\sum_{c=1}^{N} P(\boldsymbol{\alpha}_j \mid \mathbf{n}_c, \Theta)}{Q_j}. \tag{29}$$

Galima nesunkiai įsitikinti, kad Jakobiano matricos skaičiavimas (28) ir (29) išraiškomis yra gana sudėtingas ir ilgai trunkantis procesas, jei įvertinsime tai, kad dažnių (stebėjimų) vektorių $\mathbf{n}_c$ skaičius neribojamas ir Dirichlet mišinių modelis gali susidėti iš daug komponentų, kurie taip pat gali priklausyti nuo daugelio parametrų. Tačiau žinoma, kad apytiksli Jakobiano matrica gali būti apskaičiuota baigtinių skirtumų metodu [55]. Levenberg-Marquardt optimizavimo eigoje naudojome tiek analitinį Jakobiano modelį, tiek apytikslio Jakobiano skaičiavimus. Įdomu tai, kad naudodami apytikslio Jakobiano skaičiavimus gavome optimizuotą Dirichlet mišinio modelį, kurio naudojimas duomenų klasifikavime lėmė geresnius rezultatus nei tuo atveju, kai buvo naudojamas analitinis Jakobiano modelis. Visuose atliktuose mūsų tyrimuose su Dirichlet mišiniais gaudavome, kad Jakobiano aproksimavimai būdavo efektyvūs tiek skaičiavimo trukmės, tiek tikslumo atžvilgiais. Tuo labai stebėtis nereiktų, kadangi dideliuose uždaviniuose Jakobiano aproksimavimas ir sėkmingas naudojimas vietoje analitinių Jakobiano modelių neprieštarauja teorijai [49].

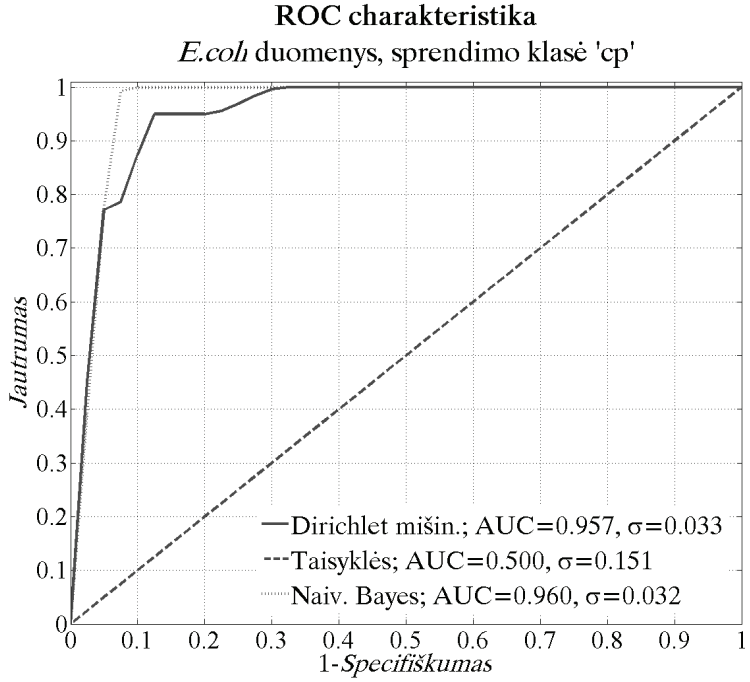
Dirichlet mišinių klasifikatoriaus rezultatai, pateikti 17, 18, 19 ir 21 paveikslėliuose, gauti Levenberg-Marquardt metodo apmokymais su apytiksliais Jakobiano matricos skaičiavimais. Dirichlet mišinį sudarė 5 komponentai su 7 pseudo dažnių parametrais. Apmokymo ir testavimo duomenų imtys pateiktos 4 lentelėje.

4 lentelė. Apmokymo ir testavimo duomenų imtys kiekvienai sprendimo kategorijai.

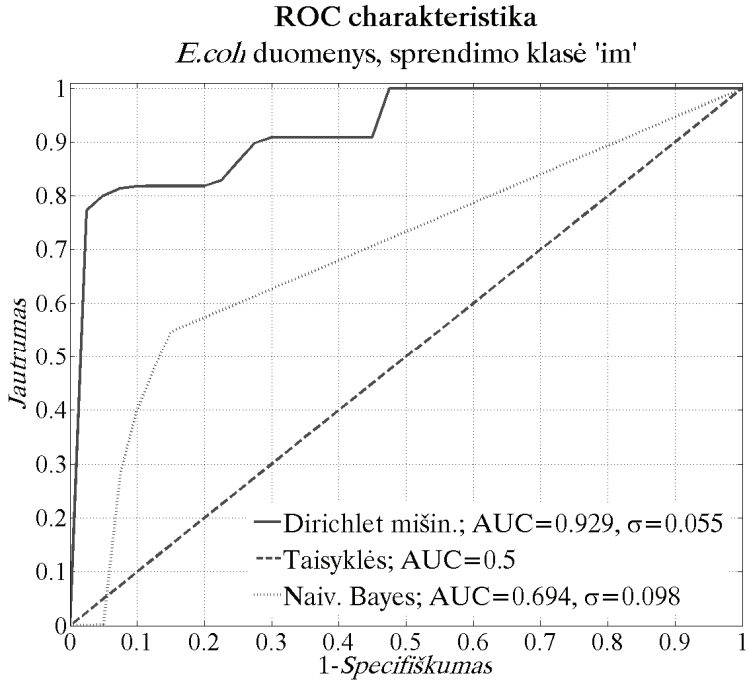
	Apmokymo duomenų imtis	*Testavimo duomenų imtis*
Kategorija ‚cp‘	143	20
Kategorija ‚im‘	77	11
Kategorija ‚imU‘	35	5
Kategorija ‚om‘	20	2
Kategorija ‚pp‘	52	7

Dirichlet mišinių klasifikatorius pranoko kitus du klasifikavimo metodus: naivaus Bayeso metodą ir klasifikavimą taisyklėmis. Nežiūrint į tai, kad naivaus Bayeso klasifikatorius panašiai tiksliai, kaip klasifikavimas Dirichlet mišiniais, klasifikavo duomenis klasifikavimo kategorijai ‚cp‘, kitoms kategorijoms jo klasifikavimo tikslumas yra visiškai prastas. Iš

tikrųjų, naivaus Bayeso metodas visus duomenų objektus priskyrė tai pačiai klasifikavimo kategorijai ‚cp‘, ir klasifikavimo tikslumas kitoms kategorijoms neviršija 0%.

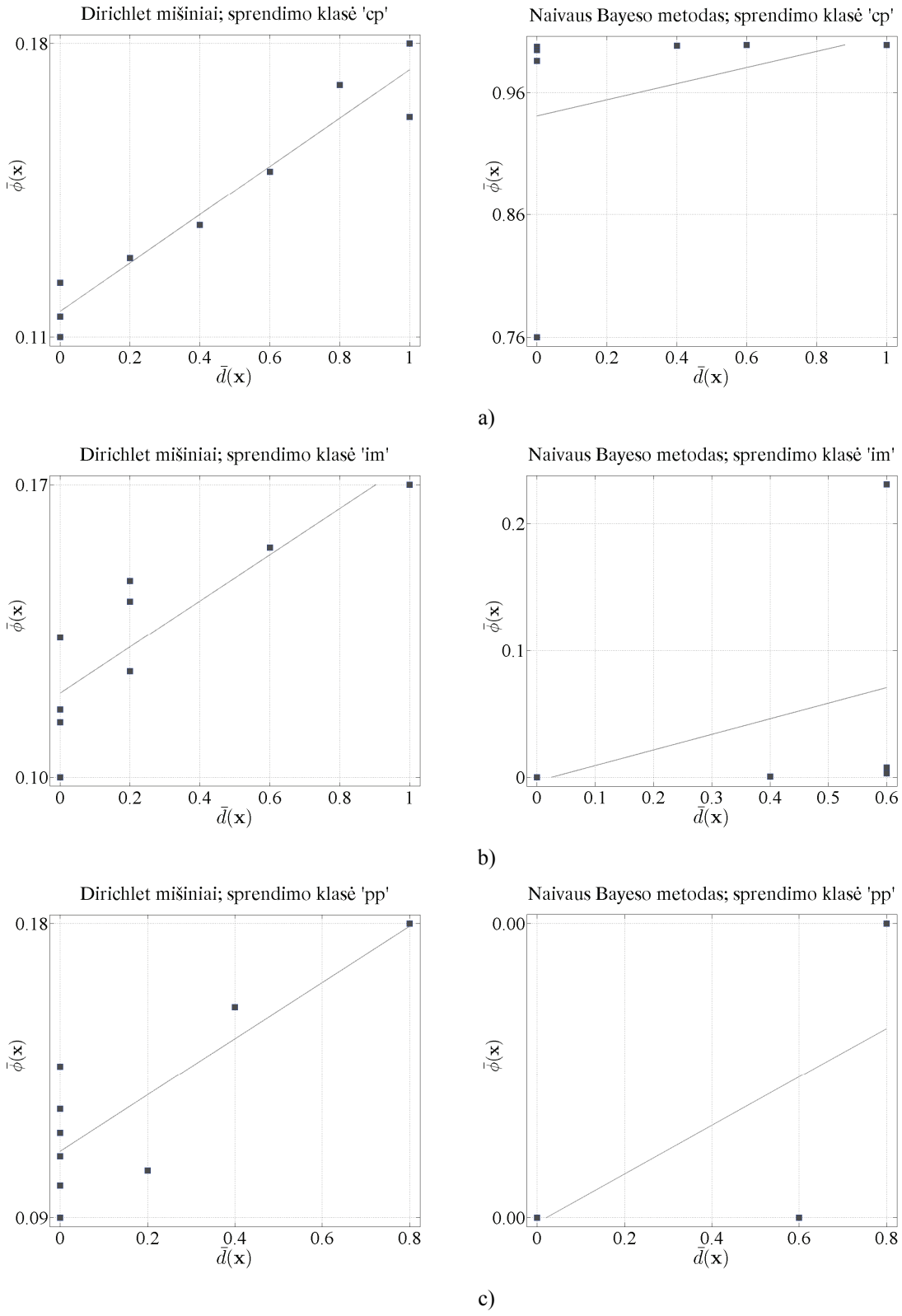


a)

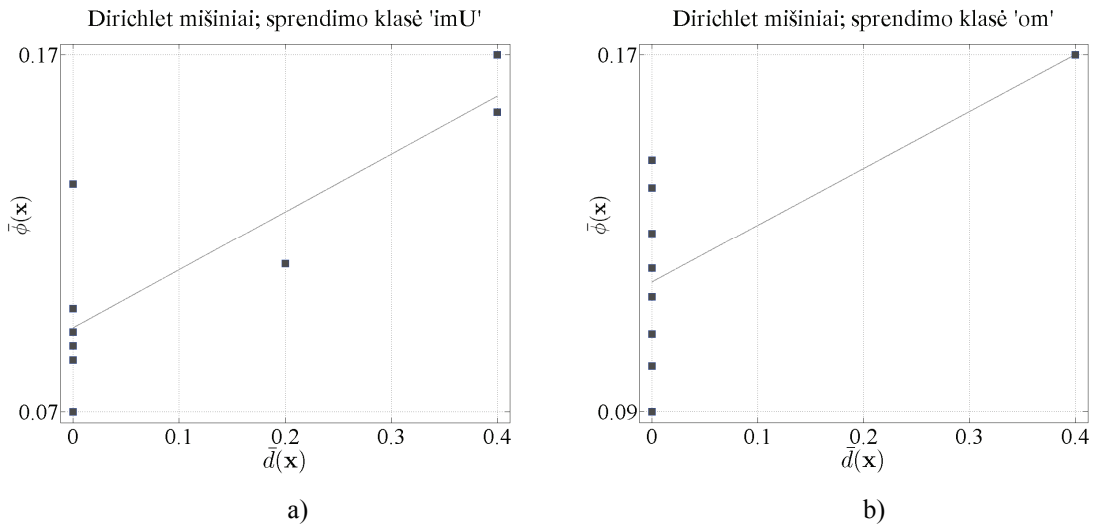


b)

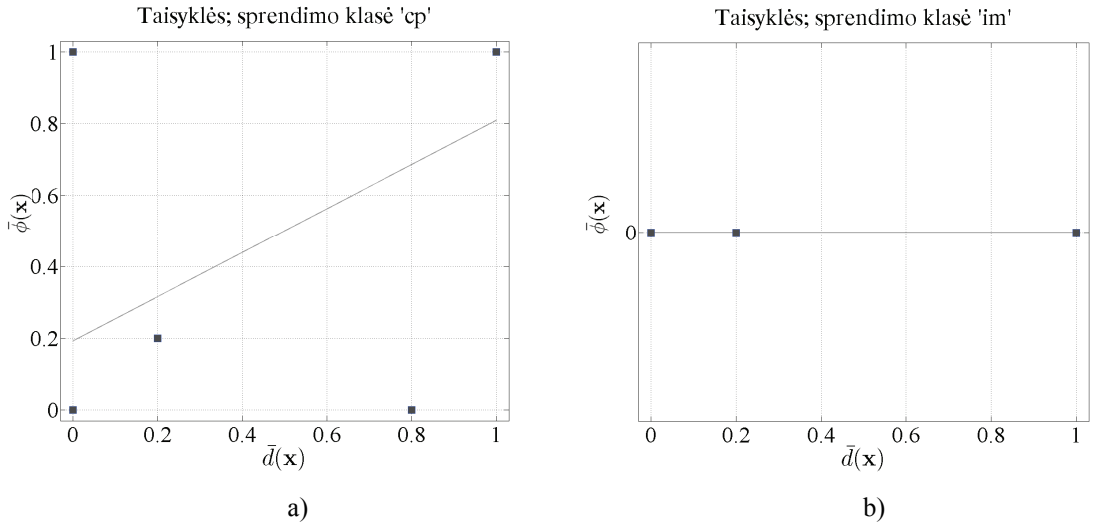
17 pav. Klasifikavimo našumas ROC kreivėmis dvejoms *E.coli* duomenų sprendimo klasėms (kategorijoms): ‚cp‘ (a) ir ‚im‘ (b). Grafikuose nubrėžtos kreivės šiems klasifikavimo metodams: Dirichlet mišinių metodui (Dirichlet mišin.), klasifikavimui taisyklėmis (Taisyklės) ir naivaus Bayeso metodui (Naiv. Bayes). AUC žymi plotą po ROC kreive, σ žymi ploto po kreive skaičiavimo kvadratinę paklaidą.



18 pav. Kalibravimo brėžiniai, gauti duomenis klasifikuojant Dirichlet mišinių klasifikatoriumi ir naivaus Bayeso metodu. Kiekvienoje paveikslėlio sekcijoje (a-c) pavaizduoti dviejų skirtingų klasifikatorių kalibravimo brėžiniai tai pačiai sprendimo kategorijai: a) kategorija ‚cp‘, b) kategorija ‚im‘, c) kategorija ‚pp‘. Grupių, atspindinčių taškų kiekį brėžinyje, skaičius lygus 9. Abscisės ašyje atidėti vidutiniai tikrųjų sprendimo kategorijų reikšmių vidurkiai, ordinatėje – klasifikatoriaus spėjimų reikšmių vidurkiai. Tiesi linija atitinka tiesinės regresijos modelį duotiems taškams.



19 pav. Dirichlet mišinių klasifikatoriaus kalibravimo brėžiniai: a) sprendimo kategorijai ‚imU‘, b) sprendimo kategorijai ‚om‘. Grupių skaičius lygus 9. Tiesi linija – tai tiesinės regresijos modelis duotiems taškams.



20 pav. Klasifikavimo taisyklėmis kalibravimo brėžiniai: a) sprendimo kategorijai ‚cp‘, b) sprendimo kategorijai ‚im‘. Grupių skaičius lygus 9.

Objektų, patenkančių į kategoriją ‚cp‘, yra gerokai daugiau nei kitose kategorijose, todėl tikėtina, kad naivaus Bayeso klasifikatorius „persimokino“ apmokymo etape ir nesugebėjo atpažinti objektų iš kitų klasifikavimo kategorijų. Bendras naivaus Bayeso klasifikatoriaus tikslumas nėra aukštas ir tesudaro 44.4%.

Nors Dirichlet mišinių klasifikatoriaus ir naivaus Bayeso metodo ROC kreivės sprendimo kategorijai ‚cp‘ artimos, kalibravimo brėžiniai geriau atspindi klasifikatorių spėjimų kokybę. Pavyzdžiui, sprendimo kategorijai ‚cp‘ naivaus Bayeso klasifikatorius spėjimai beveik visada vienodai aukšti, tai gerai iliustruoja, kodėl pastarasis klasifikatorius visus objektus priskyrė kategorijai ‚cp‘. Priešingai yra ‚im‘ sprendimo kategorijai: naivaus Bayeso klasifikatoriaus

spėjimai yra žemi ir išsibarstę. Kitoms kategorijoms pastarojo klasifikatoriaus spėjimai arba artimi nuliui, arba visi lygūs nuliui.

Dirichlet mišinių klasifikatoriaus spėjimų reikšmės nors ir nėra aukštos, bet tendencingai didėja, didėjant sprendimo kategorijų grupėse reikšmėms (18-19 pav.). Grupės buvo sudaromos tuo pačiu principu, kaip ir širdies aritmijos duomenų atveju: grupėje taškų skaičius ne mažesnis kaip 5, - todėl buvo sudarytos 9 grupės. Sprendimo kategorijų reikšmės grupėse gali įgauti reikšmes iš {1,0}; 1 reiškia kategoriją, kurios atžvilgiu atliekamas tyrimas (objektai toje kategorijoje vadinami teigiamais), 0 – bet kokia kita kategorija (objektai iš kitų kategorijų vadinami neigiamais).

Klasifikavimas taisyklėmis šiems duomenims nepasiteisino nepaisant to, kad buvo bandomi įvairūs duomenų diskretizavimo ir atributų aibės redukavimo algoritmai. ROC kreivės visoms kategorijoms primena atsitiktinį klasifikavimą. Bendras klasifikavimo taisyklėmis tikslumas nėra konkurencingas ir sudaro 35.6%. Kalibravimo brėžiniai (20 pav.) sprendimo kategorijoms ‚cp‘ ir ‚im‘ rodo prastą klasifikatoriaus spėjimų suderinamumą su sprendimo kategorijų reikšmėmis. Kitoms kategorijoms kalibravimo brėžiniai gauti analogiški kategorijai ‚im‘.

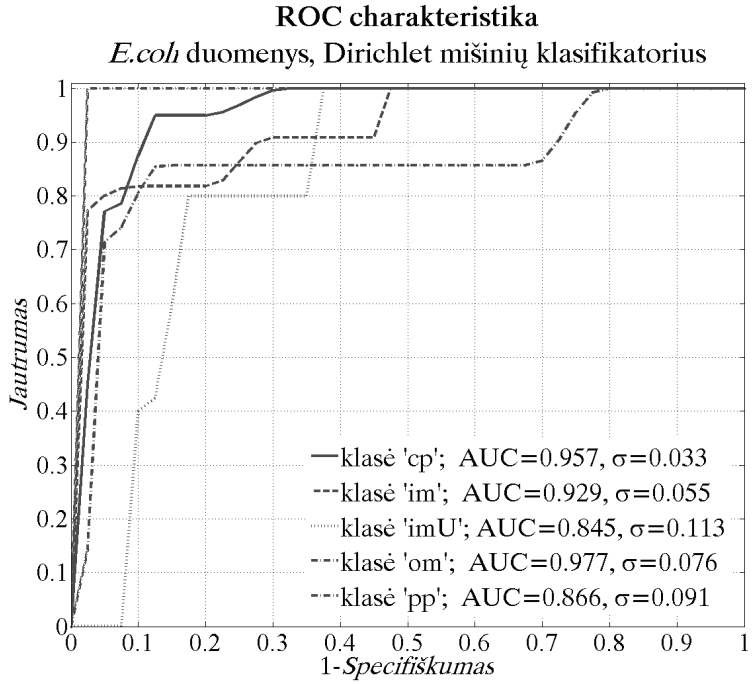
5 lentelė. Dirichlet mišinių klasifikatoriaus ir naivaus Bayeso metodo spėjimų statistiniai parametrai sprendimo kategorijai ‚cp‘.

	Dirichlet mišiniai	**Bayeso metodas**
Brierio įvertis	0.3229	0.5207
Vidutinė sprendimų reikšmė	0.4444	0.4444
Vidutinė spėjimų reikšmė	0.1381	0.9713
Vidutinė spėjimų reikšmė neigiamiems objektams	0.1215	0.9483
Vidutinė spėjimų reikšmė teigiamiems objektams	0.1588	1
Spėjimų reikšmių dispersija neigiamiems objektams	0.0002	0.0396
Spėjimų reikšmių dispersija teigiamiems objektams	0.0002	0
Vidutinių spėjimų reikšmių riba	0.0373	0.0517
Teigimų ir neigiamų spėjimų reikšmių išsibarstymas	0.0002	0.0220
Klasifikatorių spėjimų koreliacija teigiamiems objektams	-0.2179	
Klasifikatorių spėjimų koreliacija neigiamiems objektams	-0.2613	

Dirichlet mišinių klasifikatoriaus ir naivaus Bayeso metodo spėjimų statistiniai parametrai kategorijai ‚cp‘ pateikti 5 lentelėje. Iš lentelės matyti du priešingi spėjimų pasiskirstymai: jei Dirichlet mišinių spėjimai labiau pasiskirstę prie 0, tai naivaus Bayeso metodo spėjimai pasiskirstę prie 1. Nors kelių kokybinių parametrų (vidutinių spėjimų riba, išsibarstymas) skirtumai gauti nežymūs ir nereikšmingi, Brierio įvertis ir kalibravimo brėžinys (18a pav.)

labiau atskleidžia Dirichlet mišinių klasifikatoriaus spėjimų suderinamumą klasifikuojant šios kategorijos objektus. Klasifikatorių spėjimų koreliacija teigiamiems ir neigiamiems objektams yra neigiama dėl spėjimų reikšmių pasiskirstymo skirtinguose galimų reikšmių poliuose (0 ir 1). Todėl Dirichlet mišinių ir naivaus Bayeso klasifikatorių AUC reikšmingo skirtumo įvertinti pagal Hanley ir McNeilo statistiką (14) negalima.

Kad įvertinti Dirichlet mišinių ir naivaus Bayeso klasifikatorių tikslumų skirtumo reikšmingumą kategorijai ‚cp‘, McNemaro statistika negali būti taikoma, kadangi, kaip pažymėta antrame šio darbo skyriuje (žr. skyrių „Klasifikavimo našumas ir jo vertinimas“), $n_{FT} + n_{TF} = 4 < 10$ (dydžiai n_{FT} ir n_{TF} reiškia vieno klasifikatoriaus klaidingus spėjimus, kito teisingus, ir atvirkščiai), ir McNemaro statistinis dydis χ^2 nėra pasiskirstęs pagal chi-kvadrato dėsnį. Tačiau Fisherio statistika [56] parodė, kad dviejų klasifikatorių tikslumų skirtumas šiai kategorijai nėra reikšmingas (hipergeometrinio pasiskirstymo p reikšmė lygi 1). Vadinasi, Dirichlet mišinių ir naivaus Bayeso klasifikatorių tikslumų skirtumas yra atsitiktinis, ir išryškėjęs nedidelis naivaus Bayeso metodo pranašumas šiai kategorijai negali būti laikomas reikšmingu.



21 pav. Dirichlet mišinių klasifikavimo našumas ROC kreivėmis visoms *E.coli* duomenų sprendimo klasėms, kurių objektai buvo naudojami testavimo etape.

Kitoms sprendimo kategorijoms Dirichlet mišinių klasifikatorius tikslumu, našumu ir statistiniais parametrais gerokai pranoko kitus du klasifikavimo metodus. Tai matyti iš ROC

kreivių (17, 21 pav.) ir kalibravimo brėžinių (18-20 pav.). Nesutapimų matrica, atspindinti Dirichlet mišinių klasifikatoriaus spėjimus visoms kategorijoms pavaizduota 22 paveikslėlyje.

		Spėjimai						
		cp	**im**	**imL**	**imU**	**om**	**pp**	
Tikrosios reikšmės	**cp**	16	1	1	0	1	1	80.00%
	im	0	8	0	2	0	1	72.73%
	imL	0	0	0	0	0	0	--
	imU	0	0	1	4	0	0	80.00%
	om	0	0	0	0	2	0	100.00%
	pp	1	0	0	0	0	6	85.71%
		94.12%	88.89%	0%	66.67%	66.67%	75.00%	**80.00**%

22 pav. Dirichlet mišinių klasifikatoriaus nesutapimų matrica. Procentai dešiniame krašte atitinka objektų iš sprendimo kategorijų klasifikavimo tikslumą. Procentai apatinėje eilutėje reiškia klasifikatoriaus spėjimų tikslumą duotoms kategorijoms. Bendras klasifikavimo tikslumas yra matricos dešiniame apatiniame krašte.

Dirichlet mišinių klasifikatorius duomenis klasifikavo beveik tolygiai tiksliai, ir bendras tikslumas, pasiektas šiuo metodu, siekia 80%. Dirichlet mišinių klasifikavimo ROC kreivės visoms kategorijoms (21 pav.) rodo stiprią skiriamąją galią klasifikuojant *E.coli* baltymų duomenų objektus. Kad įvertinti Dirichlet mišinių tikslumo reikšmingumą, atlikome McNemaro statistikos skaičiavimus. Pagal 6 lentelėje pateiktus duomenis, apskaičiavome McNemaro statistinio dydžio reikšmę χ^2=9.38, o pagal chi-kvadrato pasiskirstymo dėsnį nustatėme, jog *P*-reikšmė, arba tikimybė, jog atsitiktinai bus gauta tokia arba didesnė reikšmė, lygi 0.0022. Vadinasi Dirichlet mišinių klasifikatoriaus bendro tikslumo pranašumas naivaus Bayeso metodo atžvilgiu yra *reikšmingas*.

6 lentelė. Dviejų klasifikatorių klaidingų ir teisingų spėjimų lentelė; ϖ_1 - Dirichlet mišiniai, ϖ_2 – naivaus Bayeso metodas.

	ϖ_1 *klaidos*	ϖ_1 *teisingi spėjimai*
ϖ_2 *klaidos*	5	20
ϖ_2 *teisingi spėjimai*	4	16

Šių duomenų (*E.Coli*) autoriai pažymi [54] pasiekę 81% tikslumą, naudojant specialiai šiems duomenims sukurtą tikimybinį modelį.

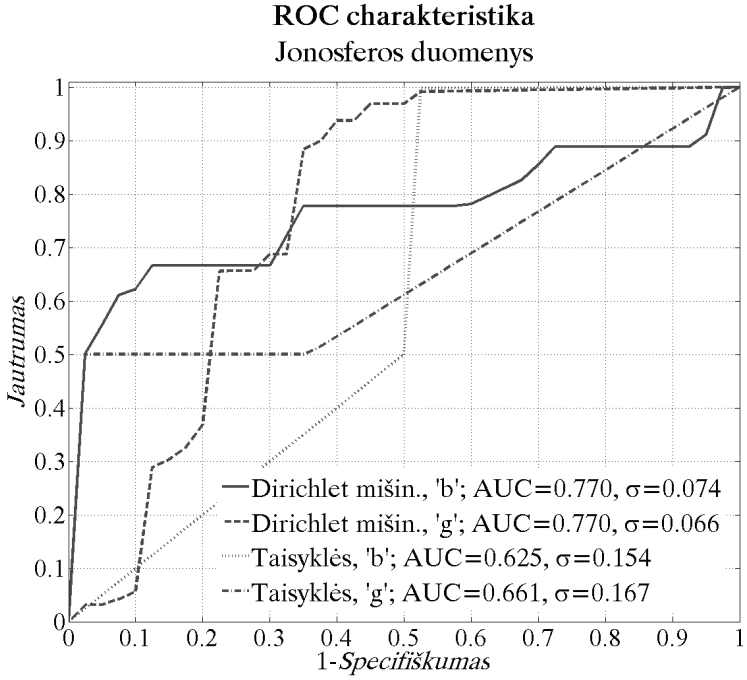
6.3. Jonosferos duomenys

Šioje duomenų bazėje sukauptos radaru užregistruotų laisvų elektronų jonosferoje signalų reikšmės naudojamos nustatyti, ar jonosferoje nėra pastebėta kokio nors tipo struktūra [57]. Tai dvejetainio klasifikavimo uždavinys, ir dvi klasifikavimo kategorijos: ‚g' ir ‚b' – reiškia „gerus" signalus (‚g'), rodančius kažkokio tipo struktūros aptikimą, ir „blogus" signalus (‚b'), reiškiančius, jog siųsti signalai jonosferoje nebuvo atspindėti. Kiekvienas gautas signalas buvo apdorotas auto-koreliacijos funkcija, kurios argumentais parinkti impulsų dažniai, o funkcijos reikšmės sudaro 34 skirtingus duomenų atributus. Šiuos duomenis perkėlėme į sprendimo lentelę su 34 atributų skaičiumi. Pilnus duomenis sudaro 351 skirtingas objektas (radaro signalo įrašai). Atributų trūkstamų reikšmių duomenyse nėra.

Šiems duomenims, taip pat kaip ir širdies aritmijos duomenims, klasifikavimas Dirichlet mišiniais buvo tiksliausias, kai naudojome genetinį algoritmą mišinio modeliui apmokyti ir modelio parametrams optimizuoti. Įdomu tai, kad panašūs rezultatai buvo gauti taikant maksimalaus tikėtinumo parametrų įvertinimo metodą (27) ir posteriorinio vidutinio įvertinimo metodą (26). Tai reikštų, kad abu metodai sėkmingai gali būti naudojami tikėtinumo išraiškų parametrams įvertinti.

Nustatėme, kad trisdešimt dviejų komponentų, kurių kiekvienas apibrėžiamas 34 pseudo dažnių parametrais $\{\alpha_{ji}\}_{i=1}^{34}$, Dirichlet mišinių modelis geriausiai tiko šiems duomenims klasifikuoti ir lėmė tiksliausius rezultatus. Kad apmokyti Dirichlet mišinio klasifikatorių atlikome 200 genetinio algoritmo iteracijų, taikant lyginio-nelyginio geno kryžminimo algoritmą ir geno pozicijų mutacijos algoritmą. Kryžminimo tikimybę parinkome 0.9, mutacijos tikimybę – lygią 0.1; populiacijos dydį nustatėme 60, vienoje kartoje individų pakeitimo skaičių – lygų 9. Taip pat bandėme įvairias genetines konfigūracijas: keisdami tinkamumo funkcijos (*fitness function*) skaičiavimo schemas (nenaudojama, tiesinė tikslo funkcijos kombinacija, laipsninė išraiškos schema, kt.), individų kartoje pakeitimo schemas (pakeičiamas tėvas, blogiausias individas, geriausias ar kiti individai), populiacijų saugojimo strategijas (kiek skirtingų populiacijų saugoti skaičiavimams: persidengiančias ar lygiagrečiai) [47]. Nustatėme, kad tinkamumo funkcijos skaičiavimas *sigma* nutraukimo algoritmu, blogiausių individų pakeitimas kartoje, persiklojančių populiacijų valdymas teigiamai įtakojo genetinio algoritmo eigą ir sąlygojo tiksliausią optimizuotą modelį, jei lyginsime su kitos konfigūracijos genetiniu algoritmu.

Klasifikavimas Dirichlet mišiniais, ROC kreivių atžvilgiu (23 pav.), pasirodė našesnis nei kiti du klasifikavimo metodai (naivaus Bayeso metodas šiems duomenims buvo netikslus – bendras tikslumas 2%, - ir tolimesnėje analizėje jo rezultatai praleidžiami).

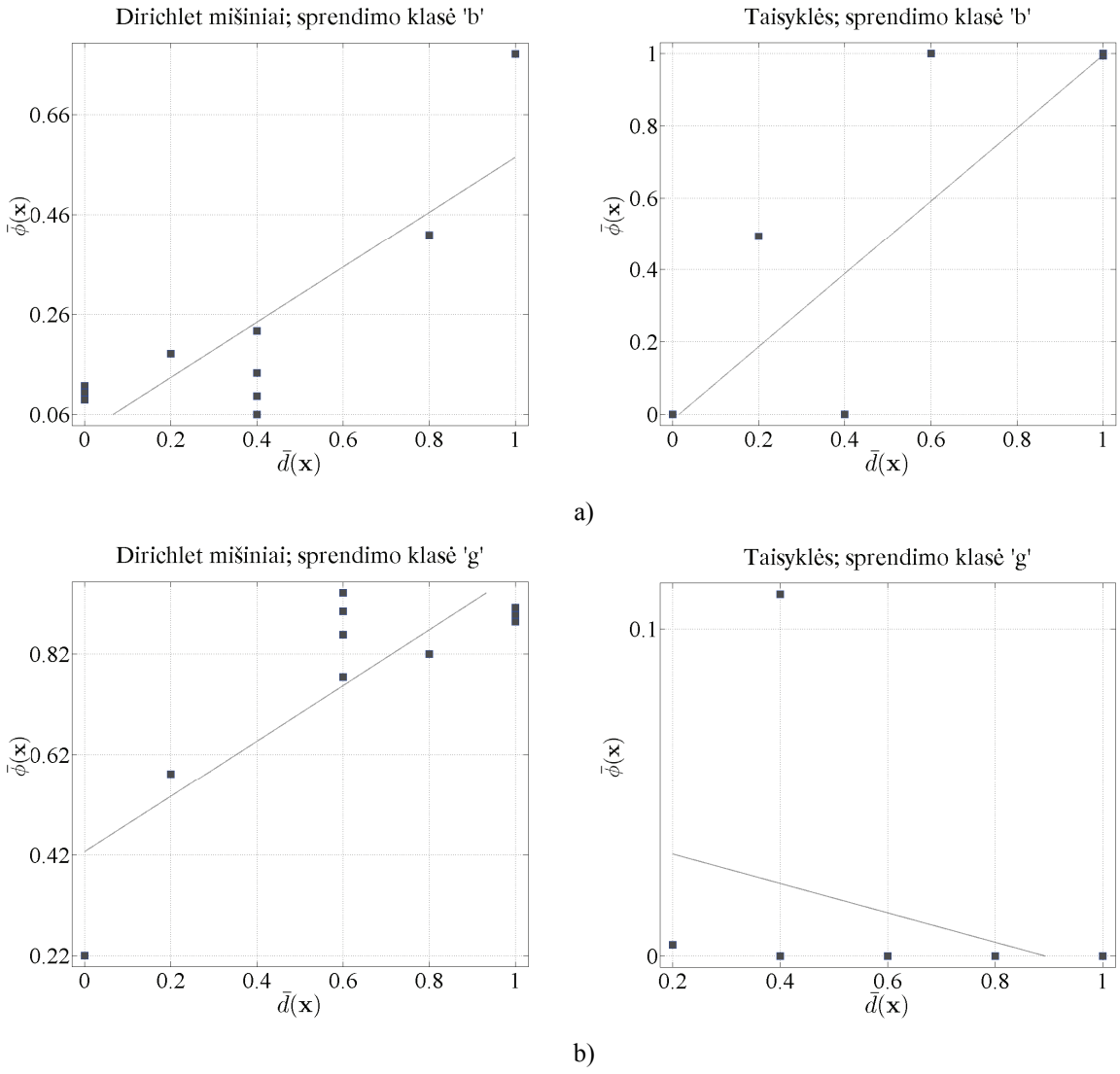


23 pav. Jonosferos duomenų klasifikavimo našumas ROC kreivėmis ‚g' ir ‚b' klasifikavimo kategorijoms. Paveikslėlyje pavaizduotos Dirichlet mišinių klasifikatoriaus (Dirichlet mišin.) ir klasifikavimo taisyklėmis (Taisyklės) našumo kreivės. Parametras σ žymi ploto po ROC kreive (AUC) skaičiavimo kvadratinę paklaidą.

ROC kreivės, gautos Dirichlet mišinių klasifikatoriui ir taisyklėms (23 pav.), skiriasi nedaug ir ruožais persidengia. Kad įvertinti dviejų klasifikatorių ploto po ROC kreive skirtumų reikšmingumą, nubraižėme kalibravimo brėžinius ir išvedėme klasifikatorių spėjimų statistinius parametrus. Kalibravimo grupių skaičius buvo parinktas taip, kad taškų kiekis grupėje būtų ne mažesnis nei 5. Sprendimo kategorijų reikšmės grupėse gali būti 1 arba 0; 1 reiškia kategoriją, kurios atžvilgiu atliekamas tyrimas (objektai toje kategorijoje vadinami teigiamais), 0 –kita kategorija (neigiami objektai).

Kalibravimo brėžiniai (24 pav.) rodo, jog kategorijai ‚b' abu klasifikatoriai spėdavo tendencingai, t.y. didėjant sprendimo kategorijų reikšmėms, klasifikatorių spėjimų reikšmės taip pat didėdavo. Tačiau kategorijai ‚g' brėžiniai skiriasi: jei Dirichlet mišinių klasifikatoriaus spėjimų reikšmės išlaiko tendencingą didėjimą, tai taisyklių spėjimų reikšmės yra mažos ir nesuderinamos su sprendimo kategorijų reikšmėmis. Iš tiesų, klasifikavimas taisyklėmis arba visus objektus priskyrė kategorijai ‚b', arba nepateikdavo visai jokio sprendimo (tikslumas kategorijai ‚g' – 0%). Dirichlet mišinių klasifikatoriaus Brierio įvertis kategorijai ‚g' lygus 0.1761, tuo tarpu taisyklėms jis lygus 0.6218. Dviejų klasifikatorių Pearsono koreliacijos koeficientai teigiamiems ir neigiamiems objektams atitinkamai lygūs -0.0647 ir -0.0671, rodantys, kad didesnių reikšmių Dirichlet mišinių klasifikatoriaus spėjimai koreliuoja su mažomis taisyklių spėjimų reikšmėmis. Dėl to kategorijai ‚g' dviejų

klasifikatorių našumų skirtumų reikšmingumo įvertinimas pagal Hanley ir McNeilo skaičiavimus neįmanomas. Šiai kategorijai taisyklių prastą klasifikavimą rodo ROC kreivės segmentas iki 0.5 specifiškumo lygio. Kita vertus, nežymų ROC kreivių plotų skirtumą galima paaiškinti dar tuo, jog, braižant ROC kreives taškai, kuomet taisyklės nepateikdavo visai jokio sprendimo, buvo ignoruojami. Taisyklėms tai pagerino ROC kreivių kokybę.



24 pav. Kalibravimo brėžiniai, gauti duomenis klasifikuojant Dirichlet mišinių klasifikatoriumi ir taisyklėmis. Paveikslėlyje pavaizduoti dviejų klasifikatorių kalibravimo brėžiniai tai pačiai sprendimo kategorijai: a) kategorijai ‚b‘, b) kategorijai ‚g‘. Grupių, atspindinčių taškų kiekį brėžinyje, skaičius lygus 10. Tiesi linija atitinka tiesinės regresijos modelį duotiems taškams.

Sprendimo kategorijai ‚b‘, kurios objektus abu klasifikatoriai klasifikavo panašiai tiksliai, klasifikatorių spėjimų statistiniai parametrai pateikti 7 lentelėje. Statistiniai parametrai gauti panašūs abiems klasifikatoriams, ir klasifikatorių spėjimų koreliacijos koeficientai teigiamiems ir neigiamiems objektams, nors ir nedideli, yra teigiami.

7 lentelė. Dirichlet mišinių klasifikatoriaus ir klasifikavimo taisyklėmis spėjimų statistiniai parametrai sprendimo kategorijai ‚b‘.

	Dirichlet mišiniai	**Bayeso metodas**
Brierio įvertis	0.1761	0.1418
Vidutinė sprendimų reikšmė	0.36	0.36
Vidutinė spėjimų reikšmė	0.2257	0.3486
Vidutinė spėjimų reikšmė neigiamiems objektams	0.1329	0.1087
Vidutinė spėjimų reikšmė teigiamiems objektams	0.3907	0.7752
Spėjimų reikšmių dispersija neigiamiems objektams	0.0030	0.0875
Spėjimų reikšmių dispersija teigiamiems objektams	0.0863	0.1818
Vidutinių spėjimų reikšmių riba	0.2578	0.6665
Teigimų ir neigiamų spėjimų reikšmių išsibarstymas	0.0330	0.1215
Klasifikatorių spėjimų koreliacija teigiamiems objektams	0.1038	
Klasifikatorių spėjimų koreliacija neigiamiems objektams	0.0363	

Nepaisant to, kad taisyklių kategorijos ‚b‘ objektų klasifikavimo specifiškumas yra prastas (pradinis ROC kreivės segmentas 23 pav.), plotas po ROC kreive nuo Dirichlet mišinių AUC reikšmės skiriasi nedaug. Skirtumo reikšmingumą nusprendėme įvertinti Hanley ir McNeilo statistika. Turėdami Pearsono koreliacijos koeficientus ir naudodami Hanley ir McNeilo lentelę [35], radome, kad $\sigma(\mathrm{AUC}_1 - \mathrm{AUC}_2) = 0.1668$ (13), o $z = 0.8693$ (14). Normalinio skirstinio z dydžio reikšmė lygi 0.2734, o P-reikšmę (P-reikšmė $= 1 - 1/2\left(1 + \mathrm{erf}\left(z/\sqrt{2}\right)\right)$) gavome lygią 0.1923. P-reikšmė išreiškia tikimybę, kad reikšmė $\geq z$ atsitiktinė. Vadinasi, Dirichlet mišinių klasifikatoriaus ir taisyklių AUC skirtumas pagal Hanley ir McNeilo skaičiavimus nėra reikšmingas.

Spėjimai

Tikros reikšmės		**g**	**b**	
	g	32	0	100.00%
	b	12	6	33.33%
		72.73%	100.00%	**76.00**%

a) Dirichlet mišiniai

Spėjimai

Tikros reikšmės		**g**	**b**	
	g	0	4	0.00%
	b	0	14	77.78%
		--	77.78%	**28.00**%

b) Taisyklės

25 pav. Dirichlet mišinių klasifikatoriaus (a) ir klasifikavimo taisyklėmis (b) nesutapimų matricos. Bendras klasifikavimo tikslumas yra matricų dešiniame apatiniame krašte. Tislumas kiekvienai kategorijai atskirai matomas dešiniuose matricų kraštuose. Klasifikavimas taisyklėmis nepateikė jokio spėjimo (matricoje neparodyta) 28 objektams iš kategorijos ‚g‘ ir 4 objektams iš kategorijos ‚b‘.

Tačiau žinant, jog klasifikavimas taisyklėmis didelei daliai objektų nepateikė išvis jokio spėjimo (25 pav.): 28 objektams iš kategorijos ‚g' ir 4 objektams iš kategorijos ‚b' - ir ROC kreivėse taškai, kuriems nebuvo gauta jokio spėjimo, neįtraukti, Hanley ir McNeilo statistika neturėtų būti vienintelis klasifikavimo našumų skirtumo reikšmingumo įvertinimo metodas. Skirtumo reikšmingumą galima vertinti tikslumų skirtumų pagrindu, pasitelkus McNemaro statistinius skaičiavimus (11). Dviejų klasifikavimo metodų nesutapimų matricos (25 pav.) atskleidžia klasifikavimo tikslumus kiekvienai kategorijai. Iš matricų matyti, jog skirtumai tarp Dirichlet mišinių klasifikatoriaus ir klasifikavimo taisyklėmis yra žymūs. Tą įrodo ir McNemaro statistika: McNemaro dydžio χ^2 reikšmė šiuo atveju lygi 12.60 (papildomi dydžiai reikalingi apskaičiuoti χ^2 pateikti 8 lentelėje), o atitinkama *P*-reikšmė, arba tikimybė atsitiktinai gauti tokį arba didesnį tikslumų skirtumą, lygi 3.9e-4. Tai reiškia, kad tikslumų skirtumas yra *reikšmingas*, ir Dirichlet klasifikatorius pagal McNemaro skaičiavimus šiai duomenų imčiai tikslumo atžvilgiu yra pranašesnis nei griežtosiomis aibėmis sugeneruotos taisyklės.

8 lentelė. Dviejų klasifikatorių klaidingų ir teisingų spėjimų lentelė; ϖ_1 - Dirichlet mišiniai, ϖ_2 – taisyklės.

	ϖ_1 *klaidos*	ϖ_1 *teisingi spėjimai*
ϖ_2 *klaidos*	3	33
ϖ_2 *teisingi spėjimai*	9	5

Bendras klasifikavimo tikslumas Dirichlet mišinių klasifikatoriui ir klasifikavimui taisyklėmis sudaro 76% ir 28% atitinkamai. Dirichlet mišinių klasifikatorius iš 32 testavimo aibėje buvusių ‚g' kategorijos duomenų visus teisingai priskyrė šiai kategorijai (100%) ir iš 18 testavimo aibėje buvusių ‚b' kategorijos duomenų teisingai priskyrė 6 (33.3%). Taisyklės nei vieno objekto nepriskyrė kategorijai ‚g' (0/32 – 0%) ir didesnę objektų dalį iš ‚b' kategorijos priskyrė teisingai (77.8%, 25 pav.), tačiau tai, kad taisyklės dažnai nepateikdavo jokio spėjimo, sumažina klasifikavimo patikimumą ir todėl taisyklių specifiškumas šiems duomenims yra žemas (23 pav.).

Atsiminus gautus rezultatus širdies aritmijos duomenims, *E.coli* duomenims, gauti rezultatai jonosferos duomenims rodo ir patvirtina, jog Dirichlet klasifikatorius jautrus apmokymo duomenų aibės apimčiai. Daugiausiai duomenų turinčios klasifikavimo kategorijos ‚g' objektus (225 objektai) klasifikatorius klasifikavo be klaidų, ir tikslumas sumažėjo ‚b' kategorijai, turinčiai viso 126 duomenų objektus. Iš kitos pusės, plotas po ROC kreive (23 pav.) šiai kategorijai gautas toks pat kaip kategorijai ‚g'. Tai sako, kad Dirichlet mišinių klasifikatoriaus prognozių kokybė kategorijai ‚b' gali būti padidinta, jei duomenų

klasifikavime butų naudojamos ROC analizės rezultate gautos specifiškumo-jautrumo slenkstinės reikšmės. Tokiu atveju, spėjimai, neviršijantys slenkstinės reikmšės, nebūtų laikomi patikimais.

Jonosferos duomenų autoriai pažymi [57] pasiekę bendrą 96% tikslumą, naudojant specialiai šiems duomenims sudarytą neuroninių tinklų architektūrą.

6.4. Išvados

Nors egzistuoja daugybė įvairių klasifikavimo metodų, kol kas nėra sukurto universalaus klasifikavimo modelio, kuris būtų tinkamas bet kokiai klasifikavimo užduočiai spręsti. Kad įrodyti šio metodo naudą klasifikuojant skirtingus savo prigimtimi duomenis iš įvairių tyrimų sričių, parinkome tris skirtingas duomenų bazes, kurioms taikėme klasifikavimo Dirichlet mišiniais metodą. Norėdami taip pat įvertinti, kaip Dirichlet mišinių metodas veikia kitų klasifikavimo metodų atžvilgiu, duomenis klasifikavome taip pat dviem papildomais klasifikavimo metodais. Atlikę ROC analizę, tvirtiname, kad medicininiams, biologiniams ir fizikiniams duomenims iš atitinkamų duomenų bazių Dirichlet mišinių klasifikatorius pranoko kitus du klasifikavimo metodus. Šiems duomenims nei klasifikavimas taisyklėmis, griežtųjų aibių teorijos kontekste, nei naivaus Bayeso metodas negalėjo tikslumu varžytis su klasifikavimu Dirichlet mišiniais. Dirichlet mišinių klasifikatoriaus palyginimui su originaliai sudarytomis minėtiems duomenims klasifikavimo architektūromis (duomenų autoriai publikavo rezultatus), sugretinome ir palyginome skirtingais metodais gautų rezultatų tikslumus. Paaiškėjo, kad Dirichlet mišinių klasifikatorius biologinius duomenis klasifikavo beveik taip pat tiksliai kaip specialiai šiems duomenims sukurta klasifikavimo sistema. Kitoms dvejoms duomenų bazėms mišinių klasifikatorius nepasiekė tokio tikslumo, kokį yra pasiekę duomenų autoriai (publikuota medžiaga). Tačiau originaliai sudarytoms klasifikavimo sistemoms mes negalėjome atlikti ROC analizės (neturime duomenų), kurios rezultate galėtume daugiau pasakyti apie šių metodų patikimumą ir našumą. Taip pat reikia pastebėti, kad apmokymo ir testavimo duomenų pasiskirstymai ir proporcijos, naudotos mūsų atliktuose testuose, neišvengiamai skiriasi nuo originaliuose darbuose parinktų duomenų aibių, taip pat kaip ir testavimo strategija. Visa tai sudaro bendro tikslumo kelių procentų paklaidą. Iš kitos pusės, mes neatlikome iteratyvios testavimo procedūros (kuria duomenų autoriai naudojosi), kurios rezultatu naudojantis galėtume tiksliau apibrėžti bendrą Dirichlet mišinių klasifikatoriaus tikslumą.

Taikydami Dirichlet mišinių klasifikatorių trims skirtingoms duomenų bazėms, norėjome išbandyti klasifikavimo metodo galimybes tiksliai dirbti su bet kokio tipo duomenimis. Dirichlet mišinių klasifikatoriaus lankstumas taip pat pasireiškia klasifikavimo modelio

konfigūravimo galimybe, kuria pasinaudojus modelis gali būti apibrėžiamas nurodytu komponentų skaičiumi ir kiekvieno komponento pseudo dažnių parametrų skaičiumi. Toks pasirinkimas įgalina Dirichlet mišinių modelį pritaikyti konkrečiai užduočiai, kaip buvo parodyta šiame skyriuje.

Klasifikavimas Dirichlet mišiniais gali būti naudingas daugelyje mokslo sričių, tarp jų ir bioinformatikoje. Nors Dirichlet mišiniai kaip priemonė modeliuoti biologinius evoliucinius procesus jau buvo naudojami [14], klasifikavimo kontekste jie dar nėra populiarūs, greičiausiai dėl mišinio parametrų optimizavimo ypatybių. Vis tiktai, ribotos konfigūracijos Dirichlet mišinių klasifikatorius gali būti efektyviai panaudojamas baltymams klasifikuoti. Neseniai pasirodė AVM pagrįsti metodai baltymams klasifikuoti [8, 58], kurie AVM branduoliui (*kernel*) sudaryti naudoja eilučių operacijas su baltymų sekomis ir gauna vienodo ilgio duomenų vektorius. Klasifikavimo kontekste šie vektoriai gali būti interpretuojami kaip objektai su nekintama atributų aibę. Sudaryti ir išsaugoti sprendimo sistemoje vektoriai nesunkiai gali būti klasifikuojami naudojant Dirichlet mišinių klasifikatorių. Tokie tyrimai kol kas nebuvo atliekami ir spręsti, ar Dirichlet mišinių klasifikatorius bus pranašesnis už AVM, galima tik atlikus su šiais duomenimis tyrimus, kurie galėtų pratęsti Dirichlet mišinių klasifikatoriaus taikymą ir tyrimus bioinformatikos srityje.

7. IŠVADOS

Klasifikavimas, kaip būdas formaliai aprašyti realaus pasaulio duomenis, išreikšti santykius tarp jų ir tuo pačiu kaip priemonė supaprastinti ir padaryti aiškesnį realaus pasaulio vaizdavimą, neišvengiamai tapo labai svarbus mokslinių tyrimų ir pramoninės veiklos sferose. Tačiau kol kas nėra sukurto universalaus klasifikavimo modelio, kuris būtų tinkamas bet kokiai klasifikavimo užduočiai spręsti.

Šiame darbe pristatytas Dirichlet mišinių statistika paremtas metodas, kuris tinkamas plačiam uždavinių spektrui. Taip pat šio darbo plėtotėje pasiekta:

- Išanalizuota griežtųjų aibių teorija ir naivaus Bayeso metodas, su kuriais lyginamas sukurtas Dirichlet mišinių klasifikatorius.
- Ištirti Dirichlet mišinio pasiskirstymo dėsnio ypatumai, jį pritaikant duomenims klasifikuoti.
- Sukurtas Dirichlet mišinių klasifikatorius bet kokio tipo duomenims klasifikuoti. Klasifikatorius gali būti naudojamas kaip programinis įrankis su grafine vartotojo aplinka ir kaip programinė biblioteka. Biblioteka gali tapti naudinga priemone kuriant automatizuoto apmokymo programines aplinkas. Tai padidina pasiūlyto metodo ir šio darbo praktinę vertę.
- Ištirtas Dirichlet mišinių klasifikatoriaus našumas naudojant tris realių duomenų bazes. Nustatyta, kad Dirichlet mišinių klasifikatorius pranoko kitus du klasifikavimo metodus, su kuriais jis buvo lyginamas. Visais atvejais gautas Dirichlet klasifikatoriaus ir kitų metodų našumų reikšmingas skirtumas.

PUBLIKACIJŲ SĄRAŠAS

1. D. Rudokaitė-Margelevičienė, H. Pranevičius, M. Margelevičius. Plataus taikymo autonominio apmokymo įrankis duomenų klasifikavimui. *Informacinės technologijos'2006*. Konferencijos pranešimų medžiaga. Kaunas: Technologija, 2006, p. 693-699.
2. D. Rudokaitė-Margelevičienė, H. Pranevičius, M. Margelevičius. Data classification using dirichlet mixtures. *Information Technology and Control*, 2006, 35(2), (priimta spaudai).

LITERATŪRA

[1] **Hampel, F.** Some thoughts about classification. In: Jajuga, K.; Sokolowski, A.; Bock, H.-H. (eds.). *Classification, Clustering, and Data Analysis. Recent advances and applications*. Springer, 2002, p. 5-26.

[2] **Eveland, C. K.; Socolinsky, D. A.; Priebe, C. E.; Marchette, D. J.** A hierarchical methodology for class detection problems with skewed priors. *Journal of Classification*, 2005, 22(1), p. 17-48.

[3] **Vapnik, V. N.** *The Nature of Statistical Learning Theory*. Springer, 2nd ed., 1999.

[4] **Wehrens, R.; Buydens, L. M. C.; Fraley, C.; Raftery, A. E.** Model-based clustering for image segmentation and large datasets via sampling. *Journal of Classification*, 2004, 21(2), p. 231-253.

[5] **Dempster, P.; Laird, N. M.; Rubin, D. B.** Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 1977, 39(1), p. 1-38.

[6] **Cheng, J.; Baldi, P.** A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 2006 Mar 17, p. to be published.

[7] **Han, S.; Lee, B.; Yu, S. T.; Jeong, C.; Lee, S.; Kim, D.** Fold recognition by combining profile-profile alignment and support vector machine. *Bioinformatics*, 2005, 21(11), p. 2667-2673.

[8] **Leslie, C.; Kuang, R.** Fast String Kernels using Inexact Matching for Protein Sequences. *Journal of Machine Learning Research*, 2004, 5, p. 1435-1455.

[9] **Rangwala, H.; Karypis, G.** Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 2005, 21(23), p. 4239-4247.

[10] **Ding, H.; Dubchak, I.** Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 2001, 17(4), p. 349-358.

[11] **Saigo, H.; Vert, J. P.; Ueda, N.; Akutsu, T.** Protein homology detection using string alignment kernels. *Bioinformatics*, 2004, 20(11), p. 1682-1689.

[12] **Santner, T. J.; Duffy, D. E.** *The statistical analysis of discrete data*. Springer-Verlag, 1989.

[13] **Lassmann, T.; Sonnhammer, E. L.** Kalign - an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, 2005, 6, p. 298.

[14] **Hughey, R.; Karplus, K.; Krogh, A.** SAM: Sequence Alignment and Modeling software system, version 3. *Technical Report UCSC-CRL-99-11*, University of California, 1999.

[15] **Karplus, K.; Hu, B.** Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics*, 2001, 17(8), p. 713-720.

[16] **Sjolander, K.; Karplus, K.; Brown, M.; Hughey, R.; Krogh, A.; Mian, I. S.; Haussler, D.** Dirichlet mixtures: a method for improving detection of weak but significant protein sequence homology. *CABIOS*, 1996, 12(4), p. 327-345.

[17] **Lartillot, N.; Philippe, H.** A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Mol. Biol. Evol.*, 2004, 21(6), p. 1095-1109.

[18] **Waterhouse, S. R.; MacKay, D.; Robinson, A. J.** Bayesian methods for mixtures of experts. *Advances in Neural Information Processing Systems*, 8, 1996, MIT Press, p. 351-357.

[19] **Ohrn, A.; Komorowski, J.; Skowron, A.; Synak, P.** The design and implementation of a knowledge discovery toolkit based on rough sets: The ROSETTA system. In: Polkowski, L.; Skowron, A. (eds.). *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*. Studies in Fuzziness and Soft Computing, 19, Physica-Verlag, 1998, p. 376-399.

[20] **Komorowski, J.; Pawlak, Z.; Polkowski, L.; Skowron, A.** Rough sets: A tutorial. In: Pal, S. K.; Skowron, A. (eds.). *Rough fuzzy hybridization - a new trend in decision-making*. Springer, 1999, p. 3-98.

[21] **Polkowski, L.** *Advances in soft computing: Rough sets*. Physica-Verlag, 2002.

[22] **Ohrn, A.** Discernibility and Rough Sets in Medicine: tools and applications. *Phd Thesis*. Trondheim, 1999.

[23] **Polkowski, L.; Skowron, A.** *Rough Sets in Knowledge Discovery 1: Methodology and Applications*. Studies in Fuzziness and Soft Computing, Springer-Verlag, 1998.

[24] **Espinoza, J. L.** Obtaining reducts with a genetic algorithm. In: Jajuga, K.; Sokolowski, A.; Bock, H.-H. (eds.). *Classification, Clustering, and Data Analysis. Recent advances and applications*. Springer, 2002, p. 219-225.

[25] **Baesens, B.; Verstraeten, G.; Van den Poel, D.; Egmont-Petersen, M.; Van Kenhove, P.; Vanthienen, J.** Bayesian network classifiers for identifying the slope of the customer - lifecycle of long-life customers. *Working Papers of Faculty of Economics and Business Administration*, 02/154, Ghent University, 2002.

[26] **Ripley, B. D.** *Pattern recognition and neural networks*. Cambridge University Press, 1996.

[27] **Rish, I.** An empirical study of the naive bayes classifier. *IJCAI-01 workshop on Empirical Methods in Artificial Intelligence*, 2001, p. 41--46.

[28] **Efron, B.; Tibshirani, R. J.** *An introduction to the bootstrap*. Chapman & Hall/CRC, 1994.

[29] **Hanley, J. A.; McNeil, B. J.** The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology*, 1982, 143(1), p. 29-36.

[30] **Liu, H.; Wu, T.** Estimating the area under a Receiver Operating Characteristic curve for repeated measures design. *Journal of Statistical Software*, 2003, 8(12), p.

[31] **Qin, G.; Zhou, X. H.** Empirical Likelihood Inference for the Area Under the ROC Curve. *UW Biostatistics Working Paper Series*, 268, Collection Of Biostatistics Research Archive, 2005.

[32] **Ikeda, M.; Ishigaki, T.; Yamauchi, K.** Relationship between Brier score and area under the binormal ROC curve. *Comput Methods Programs Biomed.*, 2002, 67(3), p. 187-194.

[33] **Redelmeier, D. A.; Bloch, D. A.; Hickam, D. H.** Assessing predictive accuracy: how to compare Brier scores. *J Clin Epidemiol.*, 1991, 44(11), p. 1141-1146.

[34] **Durbin, R.; Eddy, S. R.; Krogh, A.; Mitchison, G.** *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[35] **Hanley, J. A.; McNeil, B. J.** A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 1983, 148(3), p. 839-843.

[36] **DeLong, E. R.; DeLong, D. M.; Clarke-Pearson, D. L.** Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 1988, 44(3), p. 837-845.

[37] **Shaw, J. W.; Pharm, D.; Horrace, W. C.** Comparison of nonparametric Receiver Operating Characteristic analysis with a likelihood-ratio test for model selection. *Technical Report*, University of Arizona, 2001.

[38] **Hajian-Tilaki, K. O.; Hanley, J. A.; Joseph, L.; P., C. J.** A comparison of parametric and nonparametric approaches to ROC analysis of quantitative diagnostic tests. *Med Decis Making*, 1997, 17(1), p. 94-102.

[39] **Arfken, G.** *Mathematical methods for physicists*. Academic Press, 3rd ed., 1985.

[40] **Ewens, W. J.; Grant, G. R.** *Statistical methods in bioinformatics*. Springer, 2001.

[41] **Jeffreys, H.** *Theory of Probability*. Oxford University Press, 3rd ed., 1998.

[42] **Ohrn, A.** The ROSETTA C++ library. 2000. SourceForge web resource: http://rosetta.sourceforge.net.

[43] **Norsett, K. G.; Laegreid, A.; Midelfart, H.; Yadetie, F.; Falkmer, S.; Grønbech, J. E.; Waldum, H. L.; Komorowski, J.; Sandvik, A. K.** Gene expression based classification of gastric carcinoma. *Cancer Lett.*, 2004, 210(2), p. 227-237.

[44] **Dubey, A. K.** Using rough sets, neural networks, and logistic regression to predict compliance with cholesterol guidelines goals in patients with coronary artery disease. *AMIA Annu Symp Proc.*, 2003, p. 834.

[45] **Hvidsten, T. R.; Laegreid, A.; Komorowski, J.** Learning rule-based models of biological process from gene expression time profiles using gene ontology. *Bioinformatics*, 2003, 19(9), p. 1116-1123.

[46] **Hvidsten, T. R.; Wilczynski, B.; Kryshtafovych, A.; Tiuryn, J.; Komorowski, J.; Fidelis, K.** Discovering regulatory binding-site modules using rule-based learning. *Genome Res.*, 2005, 15(6), p. 856-866.

[47] **Goldberg, D. E.** *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[48] **Hromkovic, J.** *Algorithmics for Hard Problems*. Springer, 2nd ed., 2002.

[49] **Nocedal, J.; Wright, S. J.** *Numerical optimization*. Springer, 2000.

[50] **Bykov, Y.** Time-Predefined and Trajectory-Based Search: Single and Multiobjective Approaches to Exam Timetabling. *Phd Thesis*. Nottingham, 2003.

[51] **Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T.** *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[52] **Newman, J.; Hettich, S.; Blake, C. L.; Merz, C. J.** UCI Repository of machine learning databases. 1998. Web resource: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[53] **Güvenir, H. A.; Acar, B.; Demiröz, G.; Cekin, A.** A supervised machine learning algorithm for arrhythmia analysis. *Proceedings of the Computers in Cardiology Conference*, 24, 1997, p. 433-436.

[54] **Horton, P.; Nakai, K.** A probablistic classification system for predicting the cellular localization sites of proteins. *Intelligent Systems in Molecular Biology*, 1996, p. 109-115.

[55] **Levy, H.; Lessman, F.** *Finite Difference Equations*. Dover, 1992.

[56] **Fisher, R. A.** On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 1922, 85(1), p. 87-94.

[57] **Sigillito, V. G.; Wing, S. P.; Hutton, L. V.; Baker, K. B.** Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 1989, 10(3), p. 262-266.

[58] **Kuang, R.; Ie, E.; Wang, K.; Wang, K.; Siddiqi, M.; Freund, Y.; Leslie, C.** Profile-based string kernels for remote homology detection and motif extraction. *J Bioinform Comput Biol.*, 2005, 3(3), p. 527-550.

PRIEDAI

Simbolių lentelė

Simbolis	Apibrėžimas
$\mathbf{U}$	Objektų aibė
$\lvert\mathbf{U}\rvert$	Aibės **U** dydis, išreikštas elementų skaičiumi
$\mathbf{A}$	Atributų aibė
d	Sprendimo atributas
d_i	Sprendimo atributo reikšmė
$d(\mathbf{x})$	Sprendimo atributo reikšmė duotam objektui **x**
$\mathbf{I}$	Informacinė sistema
$\mathbf{S}$	Sprendimo sistema
$\mathbf{V}_a$	Galimų atributo *a* reikšmių aibė
$\mathbf{D}$	Galimų sprendimo atributo reikšmių aibė
$IND_{\mathrm{S}}(\mathbf{A})$	Neatskiriamumo sąryšis atributų aibei **A**
$[\mathbf{x}]_{\mathrm{B}}$	Ekvivalentiškumo klasė atributų aibei **B**
$\underline{\mathrm{B}}\,\mathbf{X}$	Apatinė aproksimacija objektų aibei **X**
$\overline{\mathrm{B}}\,\mathbf{X}$	Viršutinė aproksimacija objektų aibei **X**
$f_{\mathrm{I}}(\mathbf{x})$	Atskiriamumo funkcija objektui **x**
$g_{\mathrm{I}}(\mathbf{U})$	Atskiriamumo funkcija visiems objektams iš **U**
$R_{\mathrm{S}}(\mathbf{x})$	Taisyklių aibė objektui **x**
$\mathbf{R}_{\mathrm{S}}$	Taisyklių aibė sprendimo sistemai
$\mathbf{C}$	Nesutapimų matrica
ϖ	Klasifikatorius
$\hat{d}_{\varpi}(\mathbf{x})$	Klasifikatoriaus spėjimas objektui **x**
TP	Teisingi pozityvai
TN	Teisingi negatyvai
FP	Neteisingi pozityvai
FN	Neteisingi negatyvai
γ	Jautrumas
η	Specifiškumas
ϕ	Klasifikatoriaus spėjimo funkcija
ψ	Klasifikatoriaus sprendžiamoji funkcija
τ	Slenkstinis parametras
$\mathbf{R}$	ROC kreivės taškų aibė
AUC	Plotas po ROC kreive
$\overline{d}_j(\mathbf{x})$	Vidutinė sprendimo kategorijos grupėje reikšmė
$\overline{\phi}_j(\mathbf{x})$	Vidutinė klasifikatoriaus spėjimo grupėje reikšmė
ξ	Kalibravimo funkcija
B	Brierio įvertis
χ^2	Pagal chi-kvadrato dėsnį pasiskirstęs atsitiktinis dydis
$n_{FT}, n_{TF}, n_{FF}, n_{TT}$	Dviejų klasifikatorių spėjimų pasiskirstymai vienas kito atžvilgiu
$\sigma(\mathrm{AUC})$	AUC skaičiavimo kvadratinė paklaida
$\sigma(\mathrm{AUC}_1\text{-}\mathrm{AUC}_2)$	Dviejų klasifikatorių AUC skirtumo kvadratinė paklaida
r	Koreliacijos koeficientas

z	Hanley ir McNeil statistinis dydis
$\mathbf{p}$	Tikimybių vektorius
$g(\mathbf{p} \mid \boldsymbol{\alpha})$	Dirichlet skirstinys
Γ	*gamma* funkcija
Ψ	*digamma* funkcija
$\boldsymbol{\alpha}$	Dirichlet skirstinio parametrai: pseudo dažniai
φ	Dirichlet mišinio pasiskirstymo dėsnis
q_j	Dirichlet mišinio koeficientai
Θ	Dirichlet mišinio modelis
$\mathbf{n}$	Dažnių vektorius
$P(\mathbf{n} \mid \mathbf{p})$	Multinominio pasiskirstymo tikimybė
$\hat{p}_i$	Posteriorinio vidurkio įvertis
$\hat{p}_i^{\mathrm{s}}$	Posteriorinio vidurkio įvertis Dirichlet skirstinio atveju
$f(\Theta)$	Tikslo funkcija

Sutrumpinimų žodynas

Sutrumpinimas	Apibrėžimas
AVM	Atraminių vektorių metodas
EM	Matematinės vilties maksimizavimo algoritmas
MRV	Magnetinio rezonanso vaizdavimas
ROC	Klasifikavimo našumą apibūdinanti kreivė
AUC	Plotas po ROC kreive

Dirichlet mišinių klasifikatoriaus C++ biblioteka

Dirichlet mišinių klasifikatoriaus biblioteka, vadinama MFAM, sukurta C++ kalba, naudojant objektiškai orientuotą grafinę biblioteką Qt. Specialiosioms funkcijoms skaičiuoti naudojama C GSL (*GNU Scientific Library*) biblioteka, genetiniam algoritmui įdiegti naudojama GAlib C++ biblioteka.

MFAM C++ biblioteką sudaro tokia kataloginė struktūra:

algoritms. Dirichlet mišinio klasifikatoriaus sąsajai skirtas paketas.

basic. Paketas, talpinantis identifikatorių sąrašą bei aprašantis naujų algoritmų ir struktūrų kūrimą

common. Tai objektų valdymui ir instaliavimui skirtas paketas. Objektais šiuo atveju gali būti algoritmas ar duomenų struktūra.

library. Pagrindinius skaičiavimus, susijusius su Dirichlet mišinių klasifikatoriaus apmokymu ir testavimu, atliekantis paketas.

src. Vartotojo sąsają, programos aplinką, išlygiagretinimo bei papildomus skaičiavimus realizuojantis paketas.

structures. Paketas, aprašantis duomenų struktūras.

system. Su operacinės sistemos bibliotekomis susijęs paketas.

Šioje kataloginėje struktūroje išdėstytas programinis tektas yra originaliai sukurtas. Sukurtos Dirichlet mišinių klasifikatoriaus bibliotekos trumpa statistika pateikta žemiau lentelėje.

Katalogas	Bylų kiekis	Eilučių kiekis
algorithms	3	1578
basic	2	40
common	2	118
library	21	4755
src	61	10750
structures	2	819
system	2	4
	93	18064

Lentelėje pateikti skaičiai atspindi sukurtos sistemos apimtį bei neįtraukia ir neįvertina sistemoje išoriškai naudojamas bibliotekas.