

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

Verslo informatikos studijų programa
Kodas 62109P101

LINAS JUŠKEVIČIUS

MAGISTRO BAIGIAMASIS DARBAS

HIBRIDINIAI STATISTINIO ATPAŽINIMO METODAI

Kaunas 2010

**VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS**

INFORMATIKOS KATEDRA

LINAS JUŠKEVIČIUS

MAGISTRO BAIGIAMASIS DARBAS

HIBRIDINIAI STATISTINIO ATPAŽINIMO METODAI

Leidžiama gintis _____

Magistrantas _____
(parašas)

Darbo vadovas _____
(parašas)

(darbo vadovo mokslo laipsnis, mokslo
pedagoginis vardas, vardas ir pavardė)

Darbo įteikimo data _____

Registracijos Nr. _____

TURINYS

SANTRUMPŲ SĄRAŠAS	4
PAVEIKSLŲ SĄRAŠAS	5
LENTELIŲ SĄRAŠAS	6
SANTRAUKA	7
ĮVADAS	8
1. OBJEKTŲ ATPAŽINIMAS IR KLASIFIKAVIMAS	10
1.1. Objektų atpažinimo problemos formulavimas	10
1.2. Klasifikatoriaus kūrimo procesas	13
1.3. Duomenų klasifikavimas	15
1.3.1. Parametrų skaičiavimas	15
1.3.2. Neparamestrinis klasifikavimas ir klaidos skaičiavimas	17
1.3.2.1. Parzeno klasifikatorius	18
1.3.2.2. k-artimiausių kaimynų klasifikatorius	21
1.3.3. Parametrinis klasifikavimas	23
1.3.3.1. Bajeso tiesinis klasifikatorius	23
1.3.3.2. Tiesinio klasifikatoriaus kūrimas	24
1.3.3.3. Kvadratinio klasifikatoriaus kūrimas	25
1.4. Neuroniniai tinklai	26
1.4.1. Neuroninių tinklų savybės	27
1.4.2. Neuroninių tinklų struktūra	28
1.4.3. Neuroninių tinklų klasifikacija	29
2. HIBRIDINIAI METODAI	32
3. EKSPERIMENTINIS TYRIMAS	34
3.1. Tyrimo duomenų rinkimas ir jų segmentavimas	34
3.2. Garso ribų išskyrimas	42
3.3. Garso parametrų išskyrimas	47
3.4. Tyrimo rezultatai	49
IŠVADOS	53
LITERATŪROS SĄRAŠAS	54
PRIEDAI	56

SANTRUMPŲ SĄRAŠAS

IT – informacinės technologijos;

KNN (*angl. k-nearest neighbor*) – k-artimiausių kaimynų metodas;

LOG – logaritmas;

MAX – maksimumas;

MIN – minimumas;

MSE (*angl. minimum mean square error*) – minimali kvadratų vidurkio klaida;

SEG – segmentacija, segmentas;

PAVEIKSLŲ SĄRAŠAS

Paveikslas 1. Du objektų matavimai: a) signalas; b) simbolis.	11
Paveikslas 2. Etaloninių duomenų išskirstymas į tinkamus ir netinkamus.	12
Paveikslas 3. Klasifikatoriaus kūrimo proceso veiklos sekų diagrama.	14
Paveikslas 4. Parzeno klaida su skirtingomis r reikšmėmis: a) 1;-1; b) 1;-40; c) 1;-∞.	20
Paveikslas 5. Slenksčio reikšmė kNN klasifikavimui: a) 1;-1; b) 1;-40; c) 1;-∞.	22
Paveikslas 6. Adaptyvios sistemos kūrimas.	27
Paveikslas 7. Dirbtinio neurono sandara.	28
Paveikslas 8. Daugiasluoksnių perceptronų tipo neuroninis tinklas.	29
Paveikslas 9. Vienkrypčio ryšio neuroninis tinklas.	30
Paveikslas 10. Grįžtamojo ryšio neuroninis tinklas.	30
Paveikslas 11. Žodžių indeksavimas SEG WORD pagalba.	35
Paveikslas 12. Triukšmo išskyrimas žodžio fragmente.	36
Paveikslas 13. Žodis su tyliau išstartomis raidėmis.	37
Paveikslas 14. Diktoriaus tyliai tariamos frazės.	37
Paveikslas 15. Įkvėpimo problema diktuojant frazes.	38
Paveikslas 16. Netolygaus žodžių diktavimo pavyzdys.	38
Paveikslas 17. Diktoriaus vyro pirmųjų frazių dikcijos išskirtinumas.	39
Paveikslas 18. Diktorės moters pirmųjų frazių dikcijos išskirtinumas.	39
Paveikslas 19. Itin aiškus diktoriaus tarimas.	40
Paveikslas 20. Itin neaiškus diktoriaus tarimas.	40
Paveikslas 21. Žodžio „Šeši“ aiškus garso ribų išsidėstymas.	42
Paveikslas 22. Sunkiai suprantamos žodžio „Nulis“ garsų ribos.	43
Paveikslas 23. Sunkiai fonetiškai atpažįstamas žodelis „Taip“.	45
Paveikslas 24. Žodis „Aštuoni“ su „nukirpta“ paskutine raide.	45
Paveikslas 25. Dvibalsis „ie“ žodyje „Vienas“.	46

LENTELIŲ SĄRAŠAS

Lentelė 1. Diktoriaus vyro frazės „taip“ segmentai ir ilgiai	41
Lentelė 2. Apdorotų/nepavykusių frazių suvestinė	44
Lentelė 3. Žodžio „Nulis“ fonemų atpažinimo rezultatai	49
Lentelė 4. Žodžio „Du“ fonemų atpažinimo rezultatai.....	50
Lentelė 5. Žodžio „Trys“ fonemų atpažinimo rezultatai	50
Lentelė 6. Žodžio „Penki“ fonemų atpažinimo rezultatai	51
Lentelė 7. Žodžio „Šeši“ fonemų atpažinimo rezultatai.....	51

JUŠKEVIČIUS, Linas. (2010) Hybrid Statistical Methods for Identification. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 67 p.

SUMMARY

Data classification is a complex process in which the set of initial data is treated to create a system that allows the identification of automated objects and data flows. The paper presents and discusses the main methods of object identification, seeks to examine whether the modification of models increases the accuracy of object recognition.

The aim is to analyze the methods of statistical object detection, offer a new hybrid method and verify its reliability. In order to achieve this objective, analytical part deals with methods of object identification and analyzes the object recognition problems. Further, the description of the proposed hybrid method is suggested and its theoretical benefits are indicated. Experimental part describes the whole research process, data treatment processing and summarizes the results.

After evaluating the methods of object identification, hybrid method, that unifies linear discriminant and methods of neuron system, is proposed. Comparing the standard object recognition methods with the hybrid ones, it was demonstrated, that the hybrid methods allow for greater accuracy (approximately 10%) in the process of identification.

Paper consists of 51 sheer pages; inventory is enlivened by 25 pictures, 7 tables and 5 appendixes.

ĮVADAS

Šiuolaikinėje informacinėje visuomenėje vis didesnę reikšmę įgyja informacijos tikslumas ir konkretumas; žmonių ir mašinų gebėjimai atrinkti ir klasifikuoti informaciją. Kuriami nauji metodai ir modeliai, leidžiantys pagerinti reikiamų duomenų srautus.

Duomenų klasifikavimas – sudėtingas procesas, kurio metu pradinių duomenų aibės apdorojamos taip, kad būtų sukurta sistema, leidžianti automatizuoti objektų atpažinimą ir informacijos srautus. Darbe pristatomi ir aptariami pagrindiniai objektų atpažinimo metodai, siekiama iširti ar modelių modifikacijos padidina objektų atpažinimo tikslumą.

Statistinis objektų klasifikavimas – specifinė IT sritis, kurioje dar yra daug neatsakytų klausimų. Nors naujausiuose literatūros šaltiniuose statistinius objektų atpažinimo metodus yra aprašę daug garsių mokslininkų, tokių kaip K. Fukunaga, N. Friedman, R. Duda ar R. Duin, bet šių metodų modifikacijos ir hibridiniai metodai plačiau aptarti tik kelių autorių. Plačiausiai hibridinius statistinio atpažinimo metodus analizavo šveicaras H. Bunke ir amerikietis A. Kandel, kurie siūlė įvairius neuroninių tinklų ir kitų metodų rinkinius tikslesniam objektų atpažinimui.

Darbo objektas – hibridiniai objektų atpažinimo metodai.

Darbo tikslas – išanalizuoti statistinio objektų atpažinimo metodus, pasiūlyti naują hibridinį metodą ir patikrinti jo patikimumą.

Siekiant užsibrėžo tikslo, yra sprendžiami tokie **uždaviniai**:

- Išsiaiškinti pagrindines objektų atpažinimo procese kylančias problemas;
- Išnagrinėti statistinio objektų atpažinimo metodų ypatumus;
- Įvertinus objektų atpažinimo metodus, išanalizuoti naują hibridinį metodą;
- Iširti hibridinio metodo patikimumą fonetinio objektų atpažinimo procese.

Darbą struktūra: analitinėje dalyje aptariami objektų atpažinimo metodai, nagrinėjamos objektų atpažinimo problemos, toliau pateiktas siūlomo hibridinio metodo aprašymas ir jo teorinė nauda. Eksperimentinėje dalyje aprašyta visa tyrimo eiga, duomenų apdorojimo procesas, apibendrinami rezultatai.

Analitinėje/teorinėje dalyje **panaudota** 21 literatūros šaltinių, iš jų – 12 publikuotos knygos, 9 – internetiniai šaltiniai ir publikacijos. 8 šaltiniai publikuoti lietuvių kalba, visi kiti – anglų kalba. Analitinėje dalyje daugiausiai remtasi K. Fukunaga bei R. Duda ir bendra autorių knygų ištraukomis. Darbe remtasi VUKHF informatikos katedros magistro baigiamojo darbo metodiniais nurodymais¹.

¹ ŠARKIŪNAITĖ, Ingrida; KRIKŠČIŪNIENĖ, Dalia; SIMUTIS, Rimvydas. (2007) *Magistro baigiamojo darbo rengimo tvarka. Metodiniai nurodymai*.

Analitinėje/teorinėje dalyje panaudotas mokslinės **literatūros analizės ir apibendrinimo metodas**, hibridinio modelio pasiūlymo dalyje panaudotas **modelio kūrimo metodas**, eksperimentinėje dalyje panaudoti **statistinės analizės ir apibendrinimo metodai**.

Darbo eigoje pasiektas planuotas **rezultatas** – pasiūlytas metodas statistinės analizės būdu patikrintas ir įvertintas kaip patikimas, todėl jo panaudojimas objektų atpažinimo srityje yra rekomenduotinas.

Eksperimentinio tyrimo metu susidurta su **sunkumais** duomenų rinkimo ir apdorojimo metu. Renkant duomenis neišvengta problemų su diktorių balsinėmis savybėmis ir nekokybiškais garso įrašais dėl pašalinių triukšmų. Apdorojimo metu šios problemos neleido panaudoti visų tyrimo metu surinktų duomenų, trys per ilgos frazės buvo visiškai pašalintos, o visi nekokybiški įrašų fragmentai nebuvo įtraukti į tyrimo duomenų aibę.

Darbą sudaro 3 pagrindinės dalys: teorinė/analitinė dalis, metodo pasiūlymo dalis ir eksperimentinis tyrimas. Darbą sudaro 67 puslapiai, aprašą pagyvina ir jo dalis paaiškina 25 paveikslai, 7 lentelės ir 5 priedai

1. OBJEKTŲ ATPAŽINIMAS IR KLASIFIKAVIMAS

Objektų atpažinimas bendru atveju – tai procesas, kurio metu išskiriamas informacinis situacijos turinys ir įvyksta semantinė klasifikacija. Atpažinimo metu tam tikrais būdais (vaizdinė, garsinės medžiaga) gauta ir užfiksuota informacija apie objektą ar situaciją, palyginama su atmintyje saugomais arba duomenų bazėje jau esančiais ankstesniais tų objektų vaizdais bei aprašais ir jie priskiriami vienai ar kitai semantinei klasei (kategorijai).

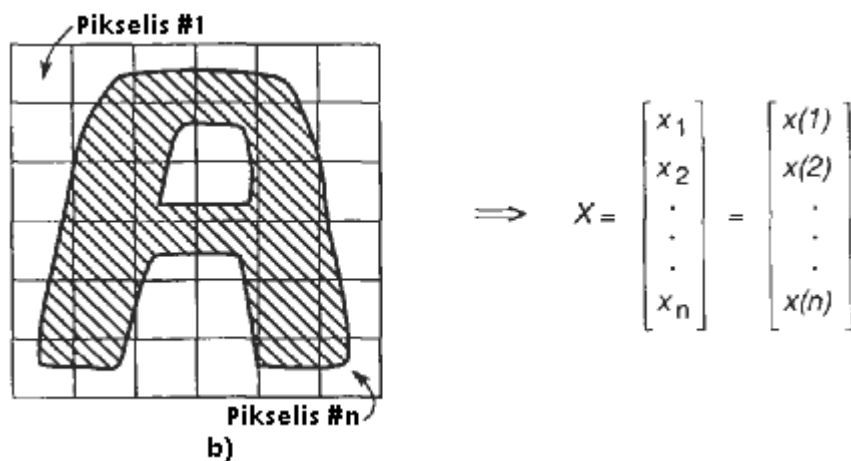
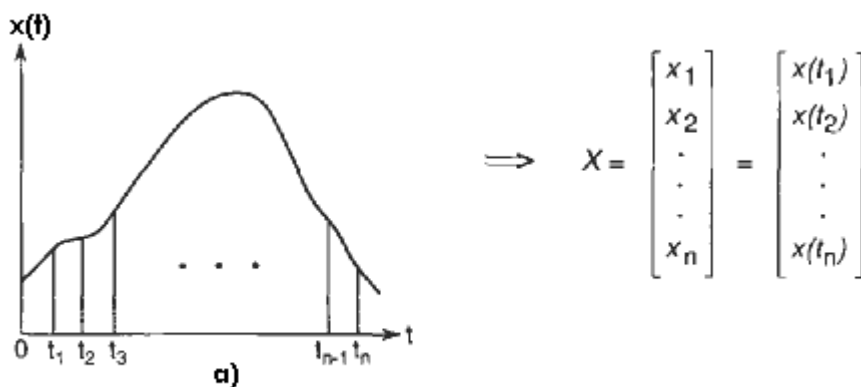
Toks objektų priskyrimas vienai ar kitai klasei atrodo nesudėtingas procesas, bet norint tai daryti automatizuotai ir kaip įmanoma tiksliau, susiduriama su daugybe klasifikavimo ir atpažinimo metu kylančių problemų. Tokių problemų kiekis priklauso ne tik nuo pradinių duomenų ir jų imties, bet ir nuo situacijos (uždavinio) sudėtingumo, klasifikavimui naudojamo tam tikro metodo bei laukiamų (tikėtinų) rezultatų. Šiame skyriuje aptariami objektų atpažinimo metodai, nagrinėjamos objektų atpažinimo metu kylančios problemos ir pateikiami pagrindiniai objektų klasifikacijos uždavinių sprendimai.

1.1. Objektų atpažinimo problemos formulavimas

Daugelis svarbiausių objektų atpažinimo uždavinių gali būti apibūdinami kaip signalų arba geometrinių figūrų klasifikavimas. Pavyzdžiui, problema, kurios metu yra testuojamas įrenginys, išvedantis tinkamą arba netinkamą informaciją yra stebimas skaičiuojant mikrofono išėjimo įtampos pakitimus per tam tikrą laiko tarpą. Šie skaičiavimai padeda supaprastinti paieškas tų įrenginių ir algoritmų, kurie daro didžiausią kiekį klaidų, ir jas pašalinti. Kita vertus, bet kokios kalbos spausdintų rašmenų atpažinimas remiasi geometrinių figūrų (raidžių) klasifikavimu (Jain, A. ir kiti, 2000, p. 1).

Norint kaip įmanoma teisingesnio klasifikavimo reikia surinkti ir turėti kuo daugiau stebimų pavyzdžių. Pats paprasčiausias, bet efektyviausias būdas surinkti šią informaciją yra išmatuoti visas tam tikru laiko momentu esančias signalo reikšmes $x(t_1), \dots, x(t_n)$ ir visus pilkos spalvos atspalvio pikselius tam tikrame objekte $x(1), \dots, x(n)$, kaip parodyta paveiksle 1 (Fukunaga, K. 1990, p. 2). Gauti matavimai n suformuoja duomenų masyvą X .

Tačiau suformuotame masyve galima išskirti reikšmes, kurios yra tikslios ir netikslios, tarp jų galima įterpti skyriklį vadinamą riba $g(x_1, x_2) = 0$. Žinodami šias dviejų reikšmių sritis ir ribą tarp jų, galime visas reikšmes išskirti į dvi kategorijas.



Šaltinis: FUKUNAGA, K. (1990) Introduction to Statistical Pattern Recognition, p. 3.

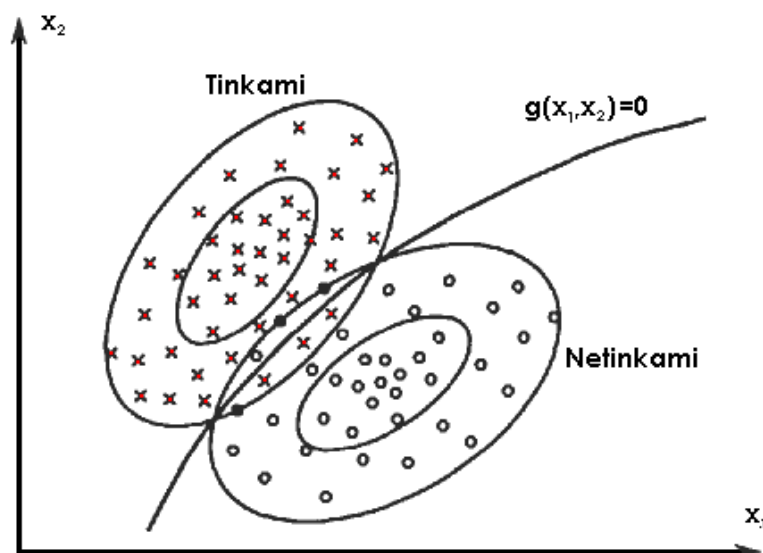
Paveikslas 1. Du objektų matavimai: a) signalas; b) simbolis.

Kai šios dvi kategorijos ir riba jau yra žinomi, tereikia pagal formules $g(x_1, x_2) < 0$ ir $g(x_1, x_2) > 0$ atskirti kurios iš reikšmių yra tinkamos modeliui, o kurios ne. Reiškiny $g(x_1, x_2)$ yra vadinamas diskriminantine funkcija (*angl. discriminant function*), o tinklą, kuris aptinka reiškinio $g(x_1, x_2)$ ženklus yra vadinamas objektų atpažinimo tinklu (*angl. pattern recognition network*) arba klasifikatoriumi (*angl. categorizer, classifier*). Norint sudaryti klasifikatorius, reikia išsiaiškinti duomenų masyvo X charakteristikas ir kiekvienam jų rasti tinkamą diskriminaninę funkciją. Šis procesą dar vadinamas apmokymu (*angl. learning, training*), o etaloniniai duomenys, kurių pagalba yra kuriami klasifikatoriai, vadinami apmokymo etaloniniais duomenimis (Webb, A. 2002; Fukunaga, K. 1990).

Pirmasis uždavinys kurį reikia spręsti – tai rasti teoriškai geriausią klasifikatorių, tariant kad skirtingų vektorių skirstiniai yra žinomi. Šis procesas vadinamas statistinės hipotezės tikrinimu, o kaip vienas geriausių klasifikatorių yra įvardijamas Bajeso klasifikatorius (*angl. Bayes classifier*), kuris minimizuoja klasifikavimo klaidų skaičių. Klaidos tikimybės išskyrimas yra vienas iš esminių parametrų objektų atpažinime. Bajeso klaida yra pati mažiausia klaida, kurią galima gauti atpažinimo procese (Fukunaga, K., 1990, p. 3).

Nors Bajeso klasifikatorius yra gana dažnas objektų atpažinime, bet jo panaudojimas yra gana sudėtingas dėl to, kad Bajeso tikimybes yra sunkus apskaičiuoti. Todėl dažnai yra naudojamas ir paprastesnis, parametrinis klasifikatorius (*angl. parametric classifier*). Parametriniai klasifikatoriai yra paremi pasirinktomis matematinėmis tankio arba diskriminantinės funkcijos formomis. Tokie klasifikatoriai kaip tiesiniai (*angl. linear*), kvadratiniai (*angl. quadratic*) arba gabalais tiesiniai (*angl. piecewise*) klasifikatoriai yra dažniausiai pasirenkami dėl savo paprastumo.

Tokiu atveju kai tankio funkcijoje neegzistuoja jokia parametrinė struktūra, funkcijai apskaičiuoti naudojami neparamestriniai metodai (*angl. nonparametric techniques*), tokie kaip Parzeno ir k-artimiausių kaimynų (*angl. k-nearest neighbor* – toliau bus naudojama termino santrumpa kNN). Bendrai paėmus, neparamestriniai metodai yra labai jautrūs parametrų skaičiui ir yra linkę pateikti klaidingus duomenis, nebent minėtų parametrų reikšmės yra kruopščiai parenkamos (Duda, R. ir kiti. 2000, p. 164).



Šaltinis: sukurta autoriaus pagal FUKUNAGA, K. (1990) Introduction to Statistical Pattern Recognition, p. 4.

Paveikslas 2. Etaloninių duomenų išskirstymas į tinkamus ir netinkamus.

Paveiksle 2 pavaizduota kaip atsitiktiniai sprendiniai yra suskirstytas į dvi dimensijas. Alternatyvus šiam metodui yra sprendimų priėmimas remiantis paieška sudarytame žodyne. Žodynai yra kuriami visus anksčiau gautus tyrimo rezultatus ir bandomuosius duomenis sudėjus į

vieną aibę (žodyną). Testavimo metu duotoji reikšmė yra priskiriama tai klasei, kuriai priklauso arčiausias etaloninis objektas. Šis procesas vadinamas artimiausių kaimynų klasifikavimo taisykle.

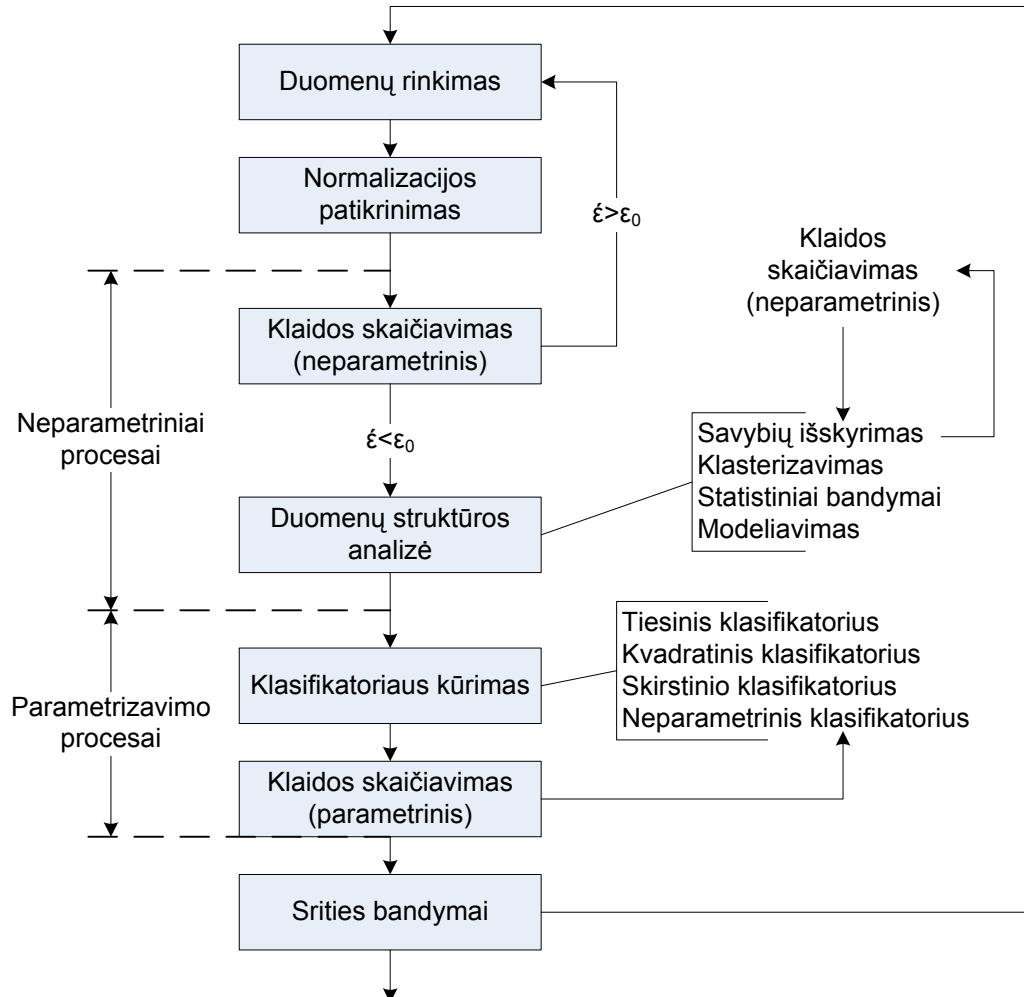
Žmogaus atliktas klasifikavimas dažniausiai remiasi mažu kiekiu savybių, tokių kaip viršūnės taškas, pagrindinis dažnis ir pan. Visi šie matavimai savyje turi tam tikrą informaciją apie jų klasifikavimą ir yra parenkami atsižvelgiant į nagrinėjamą problemos ar uždavinį. Savaimė suprantama, kad kuo mažesnės klasifikatoriui pateikiamų duomenų kiekis, tuo klasifikatoriaus kūrimas tampa lengvesnis. Siekiant pasinaudoti šiuo privalumu, reikia rasti toki būdą, kurio pagalba būtų galima iš pavyzdžių aibės išskirti svarbiausias savybes (Fukunaga, K., 1990, p. 4). Ši problema yra taip pat labai svarbi objektų atpažinimo procese ir yra vadinama savybių išskyrimu (*angl. feature selection, extraction*).

1.2. Klasifikatoriaus kūrimo procesas

Paveikslas 3 parodo veiksmų sekos modelį, pagal kurį yra kuriamas klasifikatorius. Po duomenų surinkimo proceso (*angl. data gathering*) etaloniniai objektai yra normalizuojami ir užregistruojami (*angl. normalization registration*). Sėkmingam klasifikatoriaus sukūrimui ypač didelę reikšmę turi normalizavimo ir registravimo procesai, dėl to, kad skirtingi duomenys reikalauja skirtingo tiek normalizavimo tiek registravimo.

Po normalizavimo ir registravimo procesų seka procesas, kurio metu yra pamatuojama kokiai klasei priklauso atitinkami duomenys. Šis procesas įgyvendinamas įvertinant Bajeso klaidos tikimybę matavimų plotmėje. Jei Bajeso gauta klaida yra didesnė negu galutinė klasifikatoriaus klaida, kurią norime pasiekti, reiškia kad duomenys nėra pilnai aprūpinti klasifikavimo informacija, kuri patenkintų reikalavimus. Parenkant savybes ir kuriant klasifikatorių tolesniuose etapuose, gali ženkliai padidėti klaidos tikimybė, todėl norint to išvengti reikia grįžti prie pirmojo etapo – duomenų rinkimo.

Kai jau Bajeso klaidos tikimybė yra mažesnė už leidžiamą (ϵ_0), galima pereiti prie kito projektavimo etapo, kuriame analizuojama duomenų struktūra (*angl. data structure analysis*). Šiame etape naudojami visi įmanomi duomenų analizės metodai, tokie kaip: savybių išskyrimas, klasterizavimas (*angl. clustering*), statistiniai testai, pavyzdinis modeliavimas ir pan.



Šaltinis: sukurta autoriaus pagal FUKUNAGA, K. (1990) Introduction to Statistical Pattern Recognition, p. 8.

Paveikslas 3. Klasifikatoriaus kūrimo proceso veiklos sekų diagrama.

Reikia atkreipti dėmesį į tai, kad kiekvieną kartą parenkant tam tikrą junginį savybių, yra tikrinama Bajeso klaidos tikimybė tame junginyje ir lyginama su galima klaida matavimų plotmėje. Skirtumas tarp jų parodo koks informacijos klasifikavimo kiekis yra prarandamas savybių išskyrimo procese (Friedman, J. ir kiti. 1997).

Kai jau duomenų struktūra yra daugmaž aiškiai apibrėžta, duomenys parodo kokį klasifikatorių kokiam atvejui reikia pritaikyti. Dažniausi pasirinkimai yra tiesinis, kvadratinis arba gabalais tiesinis klasifikatorius, retesniais atvejais pasirenkamas neparametrinis klasifikatorius. Neparametriniai metodai reikalingi autonominės (*angl. off-line*) analizės metu siekiant išsiaiškinti tokius itin svarbius duomenis kaip Bajeso klaidos tikimybė ir duomenų struktūra.

Po klasifikatoriaus sukūrimo, jis turi būti įvertintas atliekant tam tikras procedūras. Gautos klaidos yra palyginamos su Bajeso klaidomis savybių erdvėje. Skirtumas tarp šių dviejų klaidų parodo kaip padidėja klaidos tikimybė panaudojus atitinkamą klasifikatorių. Jei šis skirtumas yra per didelis ir nepriimtinas, reikia dar karta pergaltvoti klasifikatoriaus kūrimo procesą ar jį tiesiog perdaryti.

Galiausiai klasifikatorius yra ištestuojamas atitinkamame uždavinyje. Jei klasifikatorius neveikai taip kaip tikėtasi, reiškia, kad duomenys kurie buvo naudoti jo sudarymui skiriasi nuo tų duomenų, kurie yra naudojami kaip testiniai. Todėl siekiant maksimalaus tikslumo reikia išplėsti visą duomenų bazę arba pergaltvoti klasifikavimo procesą.

1.3. Duomenų klasifikavimas

Duomenų klasifikavimas visoje objektų atpažinimo srityje yra vienas iš svarbiausių aspektų, leidžiančių pagal klasifikavimo patikimumą įvertinti objektų atpažinimo tikslumą. Tam yra kuriami nauji metodai (taip pat tobulinami jau sukurti metodai), leidžiantys išskirti klasifikavimo metu kylančias problemas, dviprasmiškumus ir juos analizuojant, rasti tinkamiausius problemos sprendimo būdus. Skyriuje apžvelgiami pagrindiniai objektų klasifikavimo metodai, aprašomi veiksniai, darantys didžiausią įtaką objektų atpažinimo tikslumui. Visų pirma aptariamas pradinis duomenų, parametrų ir išraiškų formulavimas, vėliau aprašomi ir palyginami tarpusavyje parametrinio ir neparametrinio klasifikavimo metodai, įvertinamos juos veikiančios aplinkybės.

1.3.1. Parametrų skaičiavimas

Jei tankio funkcijas išreikštume tokiais parametrais kaip vidurkis, tikėtinai vektoriai ir kovariacinės matricos, galima būtų sukurti panašumo mato klasifikatorių, pagal kurį būtų galima atpažinti tam tikrus duomenis. Kitas būdas yra skaičiais ir parametrais išreikšti diskriminantines funkcijas, iš kurių būtų gaunamos tokios matematinės išraiškos kaip tiesinė arba kvadratinė funkcija. Bet koku atveju diskriminantinė funkcija yra tik tikėtinų vektorių ir kovariacinių matricų funkcija. Parametrinis atvejis yra vertinamas dėl jo nesudėtingumo lyginant su neparametriniu atveju, kuriame visi skaičiavimai yra paremti tankio ir diskriminantinėmis funkcijomis (McLachlan, G. 1992).

Daugelyje atpažinimo uždavinių yra daroma prielaida, kad parametrų reikšmės yra nekintančios. Deja, praktikoje parametrų tikrosios reikšmės nėra iš anksto žinomos ir turi būti gautos iš duoto skaičiaus etaloninių duomenų. Tai yra atliekama panaudojant etaloninių duomenų įvertinimo techniką nepriklausomai nuo to kad reikšmės yra atsitiktiniai dydžiai ir išsilaiko apytiksliai nenukrypdami nuo tikėtinų reikšmių

Kur kas svarbesnis matmuo objektų atpažinime yra klaidos tikimybė, kuri yra išreiškiama kaip sudėtinė dviejų parametrų funkcija: pirmasis yra parametrų rinkinys apibūdinantis klasifikatorių, antrasis – parametrų rinkinys apibūdinantis skirstinį, kuris bus naudojamas. Klaidos įvertinimas tampa labai sudėtingas ir sunkiai apskaičiuojamas dėl šių dviejų parametrų rinkinio. Toliau šiame skyriuje bus aptarta kokią įtaką tikėtinai klaidai gali turėti tikrųjų ir bandomųjų pavyzdžių kokybė.

Klasifikavimo klaidos skaičiavimas

Dar svarbesnis veiksnys kalbant apie objektų atpažinimą yra būsimo klasifikatoriaus veikimas. Diskriminantinė funkcija kai kuriems populiariems klasifikatoriams (įskaitant tiesinį ir kvadratinį klasifikatorius) yra M_1 , M_2 , Σ_1 , ir Σ_2 funkcijos, kitaip tariant, jie yra funkcinės grupės nariai. Laukiamo klasifikatoriaus veikimo suprastėjimas dėl baigtinio etaloninių duomenų kiekio, vyksta dėl dviejų priežasčių: riboto pavyzdžių rinkinio, naudojamo projektavimui ir riboto bandomųjų duomenų kiekio (Bishop, C. 2006).

Testavimo pavyzdžių poveikis. Klaidos skaičiavimo procedūra praktiškai yra vienintelis galimas variantas kai yra duotas tam tikras fiksuotas skaičius etaloninių duomenų klasifikatoriaus testavimui, t.y. pavyzdžiai yra ištestuojami klasifikatoriaus pagalba, o nesuklasifikuotų pavyzdžių skaičius yra sumuojamas. Kitas būdas yra apskaičiuoti testavimo pavyzdžių tankį ir panaudoti šiuos duomenis sudėtingesnėms situacijoms spręsti. Šis procesas yra gana painus ir sudėtingas net ir esant normaliajam skirstiniui su žinomu tikėtinų vektorių ir kovariacijos matricių skaičiumi.

Kūrimo pavyzdžių poveikis. Kur kas sudėtingesnis yra toks atvejis, kai yra duotas ribotas skaičius kūrimo pavyzdžių. Toliau norint išraišką išlaikyti kaip įmanoma bendresnę, bus nagrinėjama specifinė diskriminantinė funkcija.

Klaidos išraiška: tarkim, kad diskriminantinė funkcija yra dvi tikėtinų vektorių M_1 ir M_2 funkcijos, ir dvi kovariacinės matricos Σ_1 ir Σ_2 . Kaip tipinius klasifikavimo pavyzdžius galima įvardinti kvadratinį ir tiesinį klasifikatorius (Friedman, J. ir kiti. 1997). Jų bendroji klaidos išraiška atrodytų taip:

$$h(X) = \frac{1}{2} (M_2 - A_1^T \Sigma^{-1} X) + \frac{1}{2} (M_1^T \Sigma^{-1} M_1 - A_2^T \Sigma^{-1} M_2) \quad (1)$$

1.3.2. Nparametrinis klasifikavimas ir klaidos skaičiavimas

Nparametrinis klasifikatorius nepriklauso nuo jokių savybių, kurios būtų susijusios su tankio funkcijos struktūra. Todėl klasifikatorius tampa Bajeso klasifikatorium jeigu tankio koeficientai konverguoja į tikrąsias reikšmes. Bet taip įvyksta tik tokiu atveju, kai yra duotas baigtinis skaičius pavyzdžių.

Atsirandanti klaida yra Bajeso klaida – pasiekama mažiausia klaida, gauta iš duotojo skirstinio. Kaip jau buvo minėta, Bajeso klaida yra itin svarbus rodiklis objektų atpažinimo srityje, įgalinantis paprastesnį duomenų klasifikavimą ir matuojantis galimybę skirstyti objektus pagal atitinkamas savybes, dar prieš nustatant koks klasifikatorius bus naudojamas. Savybių (požymių) atrinkimas visuomet nulemia nedidelius klasifikuotumo praradimus. Šių praradimų kiekis ir pasekmės gali būti išmatuojami sulyginus Bajeso klaidą savybių erdvėje su Bajeso klaida realių duomenų erdvėje.

Visgi dažniausiai praktikoje nepasitaiko tokių atvejų kai pavyzdžių kiekis būtų begalinis, o dėl riboto pavyzdžių kiekio ir tankio skaičiavimų, vėliau Bajeso klaidos rezultatai turi didelius nuokrypius ir pakitimus, ypač didelių matavimų erdvėje (Fukunaga, K., 1990).

Panašios tendencijos yra ir parametrinio klasifikatoriaus atveju, tačiau nparametriniu atveju, ši tendencija tampa kiek griežtesnė.

Parzeno ir k-artimiausių kaimynų klasifikatoriai. Parzenas ir kNN siūlo panašius klasifikavimo ir klaidos apskaičiavimo algoritmus, be to pateikia ir panašius rezultatus. Visgi žvelgiant giliau, galima išvelgti gana didelius jų skirtumus ir įvertinti metodų gerąsias ir blogąsias savybes. Toliau skyriuje atskirai bus aptariami Parzeno ir kNN klasifikatoriai (požiūriai), apžvelgiant klasifikatorių kūrimą ir klaidos skaičiavimą, vėliau bus detalčiau aptarti kiekvieno šių metodų klaidos apskaičiavime kylantys sunkumai ir klaidos pakitimus lemiantys veiksniai.

1.3.2.1. Parzeno klasifikatorius

Panašumo mato klasifikatorius gali būti išreikiama išraiška $\ln p_1(X)/p_2(X) > t$, kur kintamojo t reikšmę įtakoja įvairūs reiškiniai, priklausomi nuo to koks klasifikatorius bus kuriamas. Šiame skyriuje aptariamas atvejis, kai tikrosios tankio funkcijos yra pakeičiamos į jų tikėtinas reikšmes. Kai Parzeno tankio reikšmė yra naudojama su branduolio funkcija $x_i(\cdot)$, tada panašumo mato klasifikatorius tampa toks (Fukunaga, K., 1990, p. 301):

$$- \ln \frac{\hat{P}_1(X)}{\hat{P}_2(X)} = - \ln \frac{\frac{1}{N_1} \sum_{j=1}^n \left(\left(\frac{1}{\omega_1} \right)^{\left(\frac{1}{\omega_1} \right)} \right)}{\frac{1}{N_2} \sum_{j=1}^n \left(\left(\frac{1}{\omega_2} \right)^{\left(\frac{1}{\omega_2} \right)} \right)} \quad (2)$$

čia $S = \{X_1^{(1)}, \dots, X_{N_2}^{(2)}\}$ yra duotųjų duomenų rinkinys. Išraiška (2) suklasifikuoja testavimo duomenis X į ω_1 arba ω_2 , priklausomai nuo to, ar kairioji lygties pusė yra didesnė ar mažesnė už išraišką t .

Klaidos apskaičiavimas: norint apskaičiuoti klasifikatoriaus klaidą iš duotų duomenų rinkinio S , galima naudoti pakartotinio sukeitimo (*angl. resubstitution*) ir slenkančio egzamino (*angl. leave-one-out*) metodus leidžiančius pasiekti viršutinę ir apatinę Bajeso klaidos ribas. Pakartotinio sukeitimo metodo atveju, visi galimi pavyzdžiai yra naudojami kuriant klasifikatorių ir tas pats rinkinys pavyzdžių yra testuojamas. Todėl, kai pavyzdys $X_k^{(1)}$ iš ω_1 yra ištestuojamas, yra naudojama formulė (3) (Fukunaga, K., 1990, p. 301):

$$- \ln \frac{\frac{1}{N_1} \sum_{j=1}^n \left(\left(\frac{1}{\omega_1} \right)^{\left(\frac{1}{\omega_1} \right)} \right)}{\frac{1}{N_2} \sum_{j=1}^n \left(\left(\frac{1}{\omega_2} \right)^{\left(\frac{1}{\omega_2} \right)} \right)} \quad (\text{Pakartotinio sukeitimo metodas}) \quad (3)$$

Jei aukščiau nurodyta išraiška yra mažesnė už t , tada $X_k^{(1)}$ yra teisingai suklasifikuotas, tuo tarpu jeigu išraiška yra didesnė už t , tuomet $X_k^{(1)}$ tampa neklasifikuojamu. Pakartotinio sukeitimo

atveju ω_1 -klaidos reikšmė yra gaunama ištestavus $X_1^{(1)}, \dots, X_{N_1}^{(1)}$, susumavus nesuklasifikuotų pavyzdžių kiekį, ir tą kiekį padalinus iš N_1 . Lygiai taip sekančios klaidos reikšmė yra apskaičiuojamas ištestavus rinkinį $X_1^{(2)}, \dots, X_{N_2}^{(2)}$.

Kita vertus, kai slenkančio egzamino metodas yra naudojamas norint ištestuoti $X_k^{(l)}$, jos reikšmės turi būti pašalintos iš kūrimo rinkinio. Todėl formulės (3) skaitiklis turi būti pakeistas į (Fukunaga, K., 1990, p. 302):

$$\hat{P}_{1L}(X_k^{(1)}) = \frac{1}{N_1 - 1} \left[\sum_{j=1}^{N_1} \kappa(X_k^{(1)} - X_j^{(1)}) - \kappa(X_k^{(1)} - X_j^{(1)}) \right] \quad (4)$$

Vėlgi, $X_k^{(l)}$ reikšmės yra testuojamos, o nesuklasifikuoti pavyzdžiai yra sumuojami. Paminėtina, kad atimties būdu gautas $\kappa_1(0)$ kiekis, nepriklauso nuo reikšmės k . Kai ω_2 -klaida yra ištestuojama, vardiklis formulėje (3) yra pakeičiamas tokiu pačiu būdu. Tipinės branduolio funkcijos dažniausiai patenkina nelygybę $\kappa_i(0) \geq \kappa_i(Y)$, o taip pat ir nelygybę $\kappa_i(0) \geq p_i(Y)$, tada gaunama tokia išraiška (Fukunaga, K., 1990, p. 301):

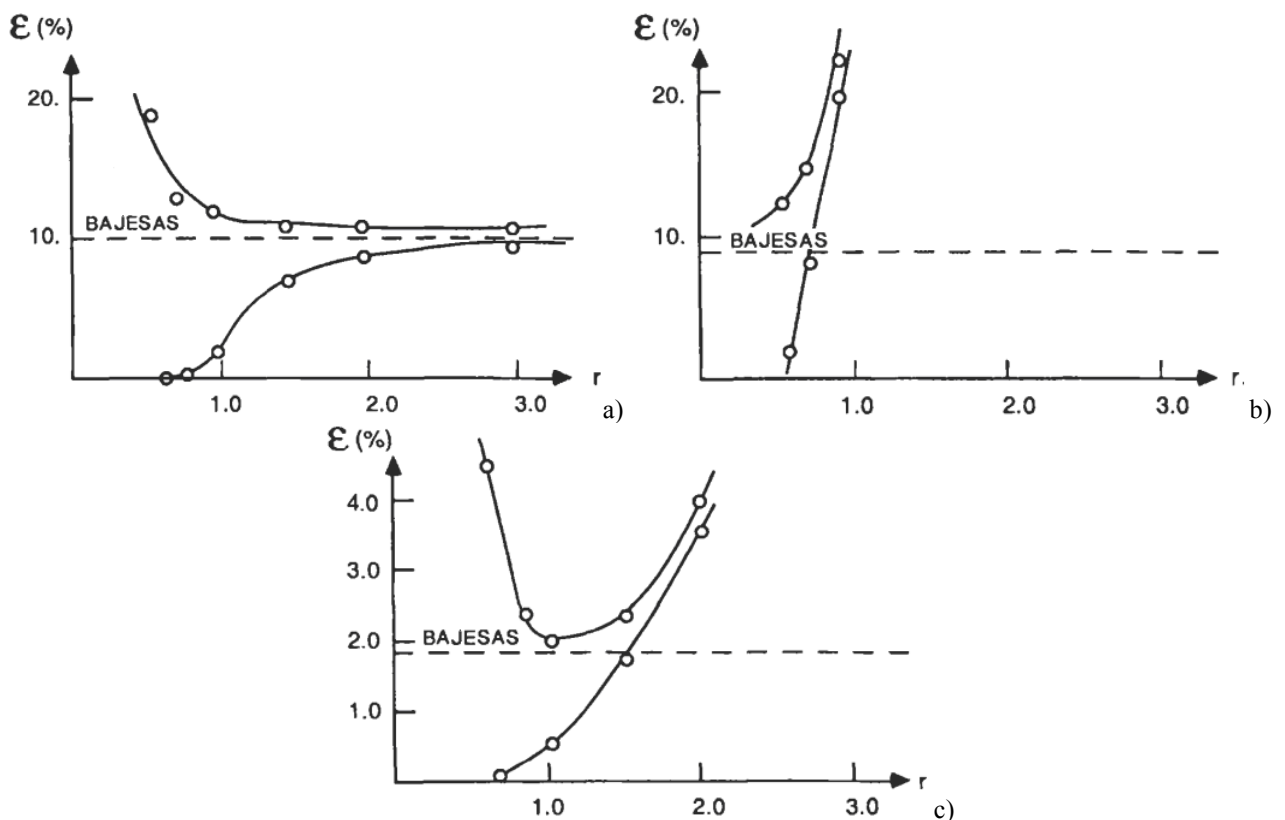
$$\hat{P}_{1L}(X_k^{(1)}) = \hat{p}_1(X_k^{(1)}) + \frac{1}{N_1 - 1} \left[\hat{p}_1(X_k^{(1)}) - \kappa_1(0) \right] \leq \hat{p}_1(X_k^{(1)}) \quad (5)$$

Tai reiškia, kad pakartotinio sukeitimo tankio reikšmė bus visada mažesnė negu slenkančio egzamino tankio reikšmė. Todėl kairioji lygties (3) pusė yra didesnė pakartotinio sukeitimo metodo atveju negu slenkančio egzamino metode, vadinasi reikšmė $X_k^{(l)}$ turi didesnius šansus būti nesuklasifikuota.

Taip pat verta paminėti, kad slenkančio egzamino tankio reikšmė gali būti gaunama iš pakartotinio sukeitimo tankio reikšmės panaudojant paprastą skaliarinę operaciją – atimant $\kappa_1(0)$ ir padalinant iš $(N_1 - 1)$. Todėl laikas, reikalingas apskaičiuoti tiek pakartotinio sukeitimo, tiek slenkančio egzamino tankių reikšmes yra beveik toks pat kaip vieno pakartotinio sukeitimo tankio funkcijos apskaičiavimo laikas.

Branduolio dydžio poveikis. Tankio funkcijos radimas ir klasifikavimas yra dvi skirtingai vykstantys procesai, todėl atrodytų tinkamiausias sprendimas vienam, gali būti visiškai netinkamas kitam. Pavyzdžiui, tankio skaičiavimuose kvadratų vidurkio klaidos reikšmė buvo naudojama siekiant surasti optimaliausią sprendimą.

Šis matavimas veda prie to, kad aukšto išsibarstymo sričių reikšmės yra svaresnės už žemo išsibarstymo sritis. Kita vertus, klasifikuojant, ryšys tarp dviejų tankio galų yra labai svarbus. Šiuo atveju kvadratų vidurkio klaida gali būti ne pats geriausias pasirinkimo variantas. Nepaisant šios srities mokslininkų pastangų, vis dar nėra tiksliai išsiaiškinta kaip optimizuotis branduolio funkcijos dydį klasifikavimo procesui. Vienas iš būdų įveikti šį sunkumą – nustatyti optimalų branduolio dydį eksperimentų pagalba (Webb, A., 2002). Kartojant klasifikavimo klaidos skaičiavimus tiek pakartotinio sukeitimo, tiek slenkančio egzamino metodais su skirtingomis r reikšmėmis, galima sudaryti lyginamąjį šių duomenų ir r reikšmių grafiką. Pagrindinis šio metodo trūkumas yra tai, kad skaičiavimo procedūra turi būti kartojama iki galo kiekvienai r reikšmei.



Šaltinis: FUKUNAGA, K. (1990) Introduction to Statistical Pattern Recognition, p. 324.

Paveikslas 4. Parzeno klaida su skirtingomis r reikšmėmis: a) 1;-1; b) 1;-40; c) 1; $-\infty$.

Paveiksle 4 pavaizduotos Bajeso klaidos viršutinė ir apatinė ribos, gautos panaudojant atitinkamai slenkančio egzamino ir pakartotinio sukeitimo metodus. Kaip matyti iš paveikslo, klaidos reikšmės yra labai panašios į r reikšmes, išskyrus (a) atvejį. Nepasirinkus tinkamo r , reikšmės tampa nesimetriškos ir net nebūtinai patenka į Bajeso klaidą. Norint išsiaiškinti kodėl klaidos reikšmės veikia taip kaip pavaizduota paveiksle 4 ir siekiant išsiaiškinti kaip parinkti reikiamus parametrus, reikėtų atlikti platesnę Parzeno klaidos skaičiavimo procedūros analizę, tačiau šiame darbe tai nebus aptarta dėl paties metodo sudėtingumo.

1.3.2.2. k-artimiausių kaimynų klasifikatorius

Naudojant kNN tankio reikšmes, anksčiau aptartas panašumo mato klasifikatorius tampa toks:

$$-1 \frac{\hat{P}_1(X)}{\hat{P}_2(X)} = -1 \frac{\epsilon_1 - \overline{N}_2 v_2(X)}{\epsilon_2 - \overline{N}_1 v_1(X)} = -1 \ln \frac{d_2 \left(\overline{N}_{k_2 NN}^{(2)}(X) \right)}{d_1 \left(\overline{N}_{k_1 NN}^{(1)}(X) \right)} = -1 \frac{\epsilon_1 - \overline{N}_2 |\Sigma_{\omega_1}|^{\frac{1}{2}}}{\epsilon_2 - \overline{N}_1 |\Sigma_{\omega_2}|^{\frac{1}{2}}} \leftrightarrow \quad (6)$$

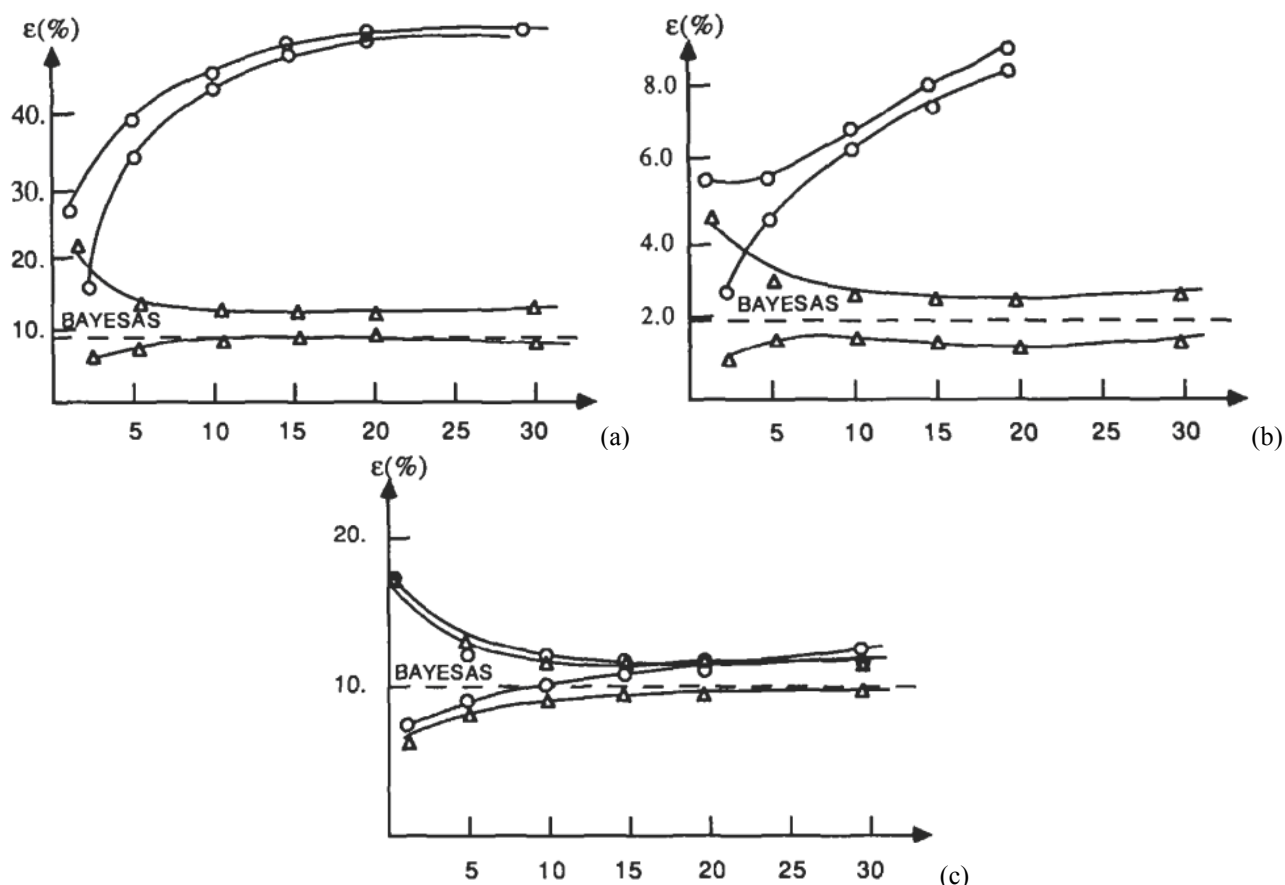
Norint suklasifikuoti testavimo duomenų pavyzdį X , visų pirma randami k_1 -asis narys iš reikšmės ω_1 ir k_2 -asis narys iš ω_2 , tuomet apskaičiuojamas atstumas tarp X ir minėtų reikšmių ir gautos reikšmės yra įdedamos į (6) kad būtų ištestuota ar kairė išraiškos pusė yra didesnė už t (Friedman, J. 2001, p. 94). Norint išvengti bereikalingos painiavos, šiuo atveju imama, kad k_1 reikšmė yra lygi k_2 .

Klaidos apskaičiavimas. Klasifikavimo klaida, paremta duotojo S rinkinio duomenų gali būti apskaičiuojama panaudojus pakartotinio sukeitimo ir slenkančio egzamino metodus. Kai $X_k^{(l)}$ iš išraiškos ω_1 yra testuojamas pakartotinio sukeitimo metodu, $X_k^{(l)}$ turi būti įtrauktas kaip etaloninių duomenų narys. Todėl, kai k-artimiausi kaimynai iš $X_k^{(l)}$ yra randamai iš ω_1 pavyzdžių rinkinio, $X_k^{(l)}$ pats yra įtraukiamas tarp kNN reikšmių. Kitame skyriuje bus plačiau aptarti sunkumai, su kuriais susiduriama klaidos apskaičiavimo procese tiek kNN, tiek ir Parzeno požiūriu (Hill, T; Lewicki, P. 2007).

Parametrų įtaka: Parzeno atveju, efektyviausias būdas sumažinti klaidos nuokrypį – nustatyti tinkamą slenkstį. Taip pat siekiant užtikrinti, kad slenkančio egzamino metodas parodys tiksliai Bajeso klaidos viršutinių ribas, būtina užtikrinti sąlygą, kad branduolio kovariacinė matrica būtų gaunama iš didelio kiekio nepriklausomų pavyzdžių arba naudojant slenkančio egzamino metodo skaičiavimų techniką.

Esant tinkamam slenksčio parinkimui, metodai, naudoti Parzeno požiūriu gali būti taip pat sėkmingai panaudoti ir kNN požiūriu. kNN tankio reikšmės yra tinkamai išsibarstę kai duomenų rinkinys yra tiksliai apibrėžtas, o slenkstis yra tinkamai pasirinktas. Tokiu atveju nuokrypio trukdžiai klasifikavimo proceso eigoje gali būti sumažinti arba visiškai panaikinti. Nėra nustatytos tinkamos reikšmės t netgi įprastiniams atvejams. Visgi kiekvienas iš retesnių metodų slenksčio pasirinkimui yra tiesiogiai pritaikomas kNN problemoms (Manning, C. ir kiti. 2008; Burges, C. 1998).

Paveiksle 5 pavaizduotas slenksčio pasirinkimo poveikis kNN klasifikavimo skaičiavimams. Palyginimui pateikti rezultatai pasiekti esant $t=0$. Kaip ir Parzeno atveju, slenkstis turi didžiulę reikšmę tais atvejais, kai duomenų kovariacijos yra skirtingos ir ypač tada kai skiriasi kovariacijų determinantai. Atveju (c) tankio reikšmių nuokrypis reikšmėms ω_1 ir ω_2 yra labai artimas Bajeso ribai ir todėl pasiekia gerus rezultatus net ir nenustatant atitinkamo slenksčio. Visgi (a) ir (b) atveju, kNN klaidos tampa stipriai išsibarstę ir visiškai nenaudingos tinkamai nepakoregavus slenksčio (Fukunaga, K., 1990, p. 349).



Šaltinis: FUKUNAGA, K. (1990) Introduction to Statistical Pattern Recognition, p. 350.

Paveikslas 5. Slenksčio reikšmė kNN klasifikavimui: a) 1;-1; b) 1;-40; c) 1;-∞.

Taigi, kalbant apie branduolio kovariacinės matricos skaičiavimus slenksčio egzamino metodu, galima teigti, kad tie patys veiksmai naudoti Parzeno požiūriu, gali būti pritaikyti ir kNN požiūriui.

1.3.3. Parametrinis klasifikavimas

Turint tankio funkcijas, ne visada įmanoma tiksliai apskaičiuoti tikimybių koeficientus. Laiko ir darbo reikalavimai tokiam klasifikavimui gali būti begaliniai. Todėl reikia nuolat plėtoti ir tobulinti paprastesnes objektų atpažinimo klasifikatorių kūrimo technikas. Paprastai galima tik nurodyti tam tikras paprastesnes klasifikatorių matematinės formas paliekant tam tikrą kiekį parametrų, kuriuos galima išgauti vėliau. Dažniausiai naudojami yra tiesiniai, kvadratiniai arba gabalais tiesiniai klasifikatoriai, kurie bus aptariami tolesniuose skyriuose.

Visų pirma reikia aptarti sąlygas, kuriomis Bajeso klasifikatorius tampa atitinkamai tiesiniu, kvadratinu arba gabalais tiesiniu. Bet visgi yra svarbu ir apžvelgti metodus, kurie yra būtini naudojant tada kai nei vieno iš minėtų klasifikatorių panaudoti nėra galimybės. Kaip bebūtų, Bajeso klasifikatorius yra pats geriausias visais atvejais ir jokie parametriniai klasifikatoriai negali palengvinti tikimybės ieškojimo proceso.

1.3.3.1. Bajeso tiesinis klasifikatorius

Dviems normaliesiems skirstiniams Bajeso taisyklė gali būti išreikšta kaip stebėjimų vektorius X kvadratinė funkcija:

$$\frac{1}{2}(X - \mu_1)^T \Sigma_1^{-1} (X - \mu_1) - \frac{1}{2}(X - \mu_2)^T \Sigma_2^{-1} (X - \mu_2) + \frac{1}{2} \ln \frac{|\Sigma_1|}{|\Sigma_2|} \stackrel{\omega}{\leftrightarrow} \frac{P_1}{P_2} \quad (7)$$

Yra daugybė klasifikatorių, kurie yra tobulinami signalų atpažinimo srityje, norint rasti naujus tinkamus būdus kaip klasifikuoti duomenis. Vienas tokių – koreliacijos klasifikatorius.

Koreliacinis klasifikatorius: Reiškiny $M_i^T X$ yra vadinamas koreliacija tarp M_i ir X . Kai X susideda iš bandomųjų reikšmių paimtų iš begalinio kintamo ciklo $x(t)$, koreliaciją galima užrašyti taip (Fukunaga, K., 1990, p. 125):

$$M_i^T X = \sum_{j=1}^n m_i(t_j) x(t_j) \quad (9)$$

O koreliacinio klasifikatoriaus struktūra gali būti aprašoma taip:

$$M_1^T X - \mu_2^T X \stackrel{\omega}{\leftrightarrow} \quad (10)$$

1.3.3.2. Tiesinio klasifikatoriaus kūrimas

Tiesiniai klasifikatoriai yra patys paprasčiausi ir lengviausiai įgyvendinami realiuose sprendimuose. Jie glaudžiai susiję su daug gerai žinomų reiškinių tokių kaip koreliacija ir Euklido atstumai. Visgi remiantis Bajesu, tiesiniai klasifikatoriai yra optimalūs tik normaliesiems skirstiniams su tolydžiosiomis kovariacinėmis matricomis. Tokiose srityse kaip signalų atpažinimas, tolydžiosios kovariacijos prielaida yra itin reikšminga, nes triukšmų savybės skirtinguose garsuose skiriasi labai nepastebimai (Burges, C. 1998, p. 12). Visgi daugelyje kitų objektų atpažinimo atvejų tolydžiosios kovariacijos prielaida nėra priimtina. Iš anksto nusistatę, kad bus naudojamas tiesinis klasifikatorius nepaisant duoto skirstinio, gauname tokį reiškinį:

$$h(X) = V^T X + v_0 \stackrel{m}{\longleftrightarrow} \stackrel{o}{\omega} \quad (11)$$

Čia $h(X)$ yra tiesinė X funkcija, kuri yra vadinama tiesine diskriminantine funkcija.

Minimali kvadratų vidurio klaida

Minimali kvadratų vidurkio klaida (minimum mean-square error MSE) yra dažnai naudojama siekiant atpažinimo problemų optimizavimo. Visgi minimali kvadratų vidurkio klaida gali būti integruojama į tiesinio klasifikatoriaus kūrimą.

Tarkim kad $\gamma(X)$ yra norima klasifikatoriaus išvestis, kuri norėtume kurti. Šiuo atveju kvadratų vidurkio klaida tarp pageidaujamo ir tikrųjų išvestinių duomenų bus išreiškiama taip (Vikipedija; Fukunaga, K. 1990 p. 324):

$$\hat{\varepsilon} = \mathbb{E}\{(h(X) - \gamma(X))^2\} = \mathbb{E}\{h^2(X)\} - E\{h(X)\gamma(X)\} + \mathbb{E}\{\gamma^2(X)\} \quad (12)$$

Pagal V ir v_0 minimizuojame gautą išraišką susumuodami tik pirmąją ir antrąją išraiškos dalis, taip gauname, kad:

$$E\{h(X)\gamma(X)\} = P_1 E\{h(X)(-)|\omega_-\} + P_2 E\{h(X)(+)|\omega_-\} \quad (13)$$

1.3.3.3. Kvadratinio klasifikatoriaus kūrimas

Kai X skirstinys yra normalusis tiek ω_1 tiek ir ω_2 atžvilgiu, Bajeso diskriminantinė funkcija tampa (7) kvadratine lygtimi. Net ir nenormaliesiems X , kvadratinis klasifikatorius yra vienas populiariausių, jis yra tinkamas daugeliui sričių. Kaip bebūtų, kol kas yra labai nedaug žinių apie tai kaip sukurti gerai veikiančią kvadratinį klasifikatorių, išskyrus tą atvejį kai išraiškos M_i ir \sum_i yra įkeliamos į išraišką (7). Taip pat kvadratiniai klasifikatoriai gali turėti ir keletą trūkumų, pagrinde susijusių su tuo, kad turi didesnius nuokrypius negu tiesiniai klasifikatoriai ir ypač tais atvejais kai yra mažas etaloninių duomenų kiekis.

Kvadratinio klasifikatoriaus kūrimo procesas. Paprastasis kvadratinis klasifikatorius gali būti išreiškiamas kaip:

$$h(X) = X^T Q X + V^T X + v_0 \leftrightarrow \omega \quad (14)$$

kur Q , V ir v_0 yra atitinkamai matrica, vektorius ir skaliaras. Todėl galima optimizuoti $\int (\eta_1, \eta_2, \sigma_1^2, \sigma_2^2) \eta = \int h(X) \omega$ ir $\sigma = \text{var}\{h(X) | \omega\}$ atsižvelgiant į Q , V ir v_0 , taip kai tai buvo padaryta tiesinio klasifikatoriaus atveju. Deja išraiškos $[n(n+1)/2] + n + 1$ parametrų skaičius yra per didelis, todėl optimizuoti $\int (\eta, \sigma, \sigma)$ būtų tiesiog nepraktiška (Fukunaga, K., 1990, p. 154).

Kvadratinis klasifikatorius Furjė atveju panaudojamas tariant, kad $F(k)$ yra normalusis skirstinys, o kvadratinis klasifikatorius Furjė atveju išreiškiamas taip (Fukunaga, K., 1990, p. 154):

$$h = \frac{1}{2} \sum_{j=0}^{n+1} v_{1j} |F(j) - E\{F(j) | \omega}|^2 - \frac{1}{2} \sum_{j=0}^{n-1} v_{2j} |F(j) - E\{F(j) | \omega}|^2 \quad (15)$$

Atkreiptinas dėmesys į tai, kad $F(j)$ kovariacinės matricos reiškiniams ω_1 ir ω_2 įstrižainės, o visos priešingos reikšmės tarp $F(j)$ ir $F(k)$ išnyksta.

Kovariacinės matricos aproksimacija: Daugiausiai sunkumų kuriant kvadratinį klasifikatorių kyla iš kovariacinių matricių. Jei priskirtume kokią nors struktūrą kovariacinei matriciai, būtų kur kas lengviau kurti kvadratinį klasifikatorių. Taip pat ir reikalingo kūrimo pavyzdžio dydis būtų sumažintas, o klasifikatorius taptų mažiau jautrus bandomojo skirstinio svyravimams. Visgi nekintamumo prielaida yra per nelyg suvaržanti visą atpažinimo procesą ir nėra gerai pritaikoma objektų atpažinimo srityje (Duda, R. ir kiti. 2000).

Kiti klasifikatoriai

Objektų atpažinimo srityje yra sukurta daug metodų, leidžiančių klasifikuoti duomenis tam tikrose probleminėse srityse, bet tokie klasifikatoriai nėra itin populiarūs nesudėtingų uždavinių sprendimuose.

Gabalais tiesiniai (piecewise) klasifikatoriai. Jei apribosime objektų atpažinimo metodus į dvi dalis – kvadratinis ir tiesinius klasifikatorius, pamatysime, kad jie užima didžiąją dalį viso atpažinimo proceso. Bet kai reikia apdoroti tris ar daugiau klasių, susiduriame su problema, kurios negali išspręsti nei vienas iš minėtų klasifikatorių. Net esant dviem klasėms, negalime pritaikyti vieno kurio nors jei klasės savyje turi keletą klasterių (Fukunaga, K., 1990, p. 169). Šiais atvejais klasifikatorių junginys, vadinamas gabalais tiesiniu klasifikatorium, leidžia palengvinti klasifikatorių panaudojimą.

Kvadratinis skirstinys kelių klasių atveju. Kelių klasių atveju keletą hipotezių patikrinimas Bajeso klaidos atžvilgiu sudaro geriausią klasifikatorių su polinkiu minimizuoti klaidas. Jei X ir L klasių skirstinys yra normalusis, tai:

$$\min \left[\frac{1}{2} (X - M_i)^T \sum_i^{-1} (X - M_i) + \frac{1}{2} \ln |\sum_i^{-1} P_i| \right] \quad (16)$$

kur max yra pakeičiamas į min dėl operacijos log minusinio ženklo. Atkreipiamas dėmesys į tai, kad normalizuotas atstumas tarp kiekvieno X ir kiekvienos klasės vidurkio M_i , turi būti sutvarkytas pagal dvi konstantų išraiškas: $(1/2) \ln |\sum|$ ir $\ln P_i$. Išraiška (16) suformuoja skirstinio kvadratinę ribą.

1.4. Neuroniniai tinklai

Dirbtiniai neuroniniai tinklai apima informacijos apdorojimo metodus, įkvėptus imituojant ir modeliuojant gyvūnų smegenyse vykstančius procesus. Dirbtinis neuroninis tinklas yra matematinių modelių rinkinys, kurio pagalba bandoma emuluoti įvairias biologinių sistemų savybes (Bishop, C. 2004, p. 2). Labiausiai viliojantis atrodo biologinių sistemų gebėjimas mokintis, prisitaikyti ir adaptuotis.

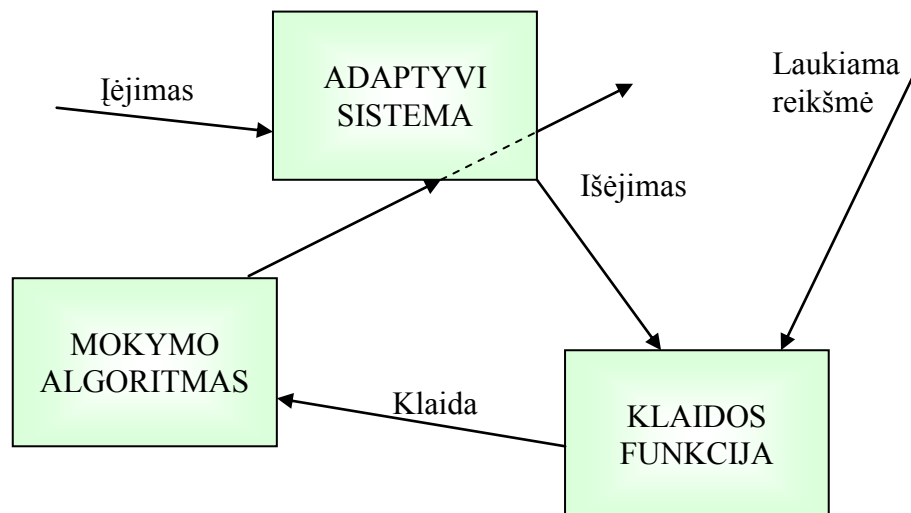
Dirbtiniai neuroniniai tinklai sudaryti iš daugelio skaičiuojančių elementų, kurie tarpusavyje glaudžiai susiję. Šie skaičiuojantys elementai yra panašūs į biologinius neuronus ir vieni su kitais sujungiami įvairaus stiprumo jungtimis, kurios yra analogiškos biologinių neuronų sinapsėms.

Mokymosi metu smegenyse keičiasi jungčių, siejančių neuronus, stiprumas. Tai galioja ir dirbtiniams neuroniniams tinklams. Neuroninių tinklų mokymui naudojami duomenų pavyzdžiai,

pagal kuriuos specialių algoritmų pagalba mokymo metu iteratyviai keičiami jungčių stiprumo koeficientai, vadinami svoriais. Informacija, reikalinga konkretaus uždavinio sprendimui, yra sukaupiama šių svorių reikšmėse.

1.4.1. Neuroninių tinklų savybės

Neuroninių tinklų adaptyvumas yra geriausia jų savybė. Neuroninės sistemos nėra sudaromos naudojant išankstines žinias pagal specifikaciją, formules ar aprašymą. Vietoje to, sistema naudoja išorinius duomenis savo parametrų nustatymui. Neuroniniai tinklai apmokomi žinant įėjimo ir atitinkamas išėjimo reikšmes (jos dar vadinamos užduoties reikšmėmis), gražinamas per ryšį, kuriame naudojama klaidos funkcija (Paveikslas 6).



Šaltinis: sudaryta autoriaus.

Paveikslas 6. Adaptyvios sistemos kūrimas.

Kaip matyti iš aukščiau pateikto paveikslo, sistemos daroma klaida naudojama sistemos parametrų keitimui. Suskaičiuota klaida gražinama sistemai, ir netiesiogiai vėl įtakoja išėjimą, keisdama parametrus. Pradžioje parametrai būna atsitiktiniai, ir sistema daro dideles klaidas. Mokymo metu parametrai derinami taip, kad klaida mažėtų. Sistema “susipažindama” su duomenimis suranda geriausią parametrų rinkinį. Adaptyvios sistemos pakankamai sudėtingos. Jos turi vykdyti reikalingas funkcijas, be to, jose papildomai naudojamos posistemės parametrų keitimui. Reikia pastebėti, jog pasikeitus duomenims ši metodika leidžia modifikuoti sistemos parametrus optimaliam uždavinio sprendimui. Ta pati sistema gali būti taikoma įvairių problemų sprendimui.

Naudojant neuroninius tinklus (kaip ir bet kurį kitą metodą), turi egzistuoti priklausomybė tarp žinomų įėjimo ir išėjimo reikšmių. Ši priklausomybė gali būti su tam tikru triukšmų lygiu, tačiau išėjimo reikšmės neturi būti sąlygojamos gryno atsitiktinumo arba faktorių, neatsispindinčių įėjimo duomenyse (Jain, A. 2000, p. 12).

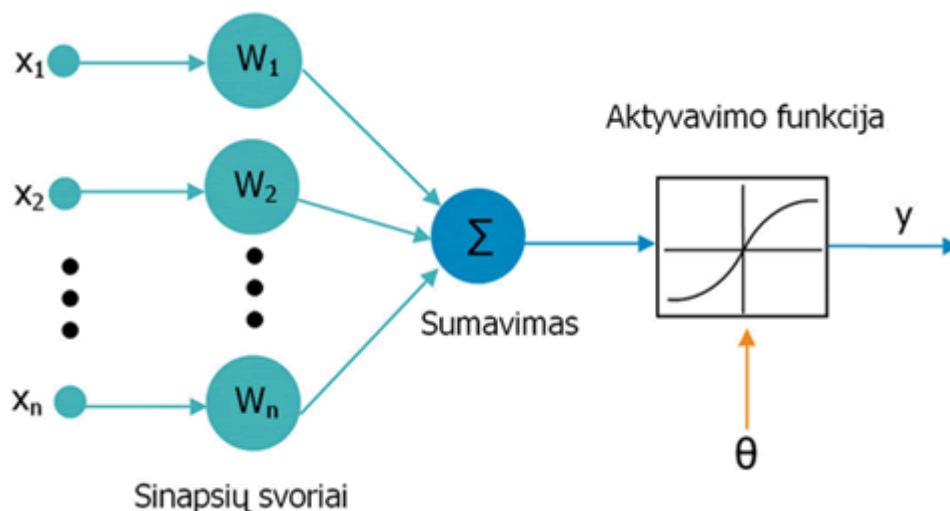
Bendru atveju, neuroniniai tinklai naudojami, kai nėra žinoma priklausomybė tarp įėjimo ir išėjimo duomenų. Jei ši priklausomybė žinoma, duomenys gali būti modeliuojami tiesiogiai. Neuroninis tinklas gali išmokyti įėjimo-išėjimo duomenų priklausomybę apmokymo metu ir tai yra viena iš pagrindinių neuroninių tinklų savybių.

1.4.2. Neuroninių tinklų struktūra

Dirbtinis neuronas (Paveikslas 7) yra svarbiausias neuroninio tinklo elementas. Jį sudaro trys pagrindiniai komponentai: svoriai, slenksčiai ir viena aktyvavimo funkcija.

$W = [W_1, W_2, \dots, W_n]$ yra svorio koeficientai, rodantys stiprumus atskirų įvesčių, aprašytų vektoriumi $X = [x_1, x_2, x_3, \dots, x_n]$. Kiekvienos įvesties signalas dauginamas iš svorio koeficiento. Tokiu būdu gaunama neuroninė jungtis XW (Makūnaitė, R. 2006).

Jei svorio koeficientas teigiamas, XW sužadina signalą išvestyje y , o jei neigiamas – XW slopina išvesties signalą. Vidinis neurono slenkstis veikia neurono y išvesties aktyvavimą štai tokiu būdu:



Šaltinis: MAKŪNAITĖ, R. (2006) Neuroniniai tinklai.

Paveikslas 7. Dirbtinio neurono sandara.

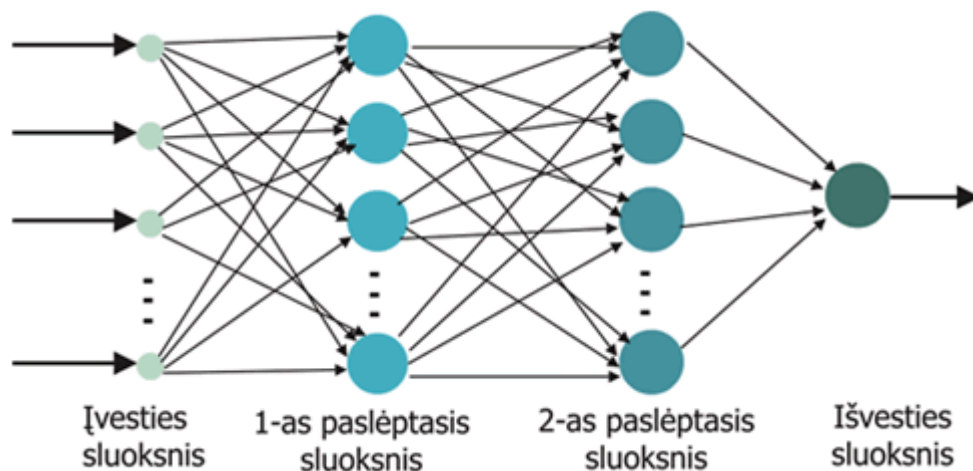
Aktyvavimo funkcija – tai matematinės operacijos su išvesties signalu. Kokio sudėtingumo aktyvavimo funkcija taikoma, priklauso nuo neuroninio tinklo sprendžiamo uždavinio.

Populiariausios – tiesinė, slenksčio, gabalais tiesinė (*angl. piecewise*), sigmoidinė ir hiperbolinė tangento aktyvavimo funkcijos.

1.4.3. Neuroninių tinklų klasifikacija

Skiriami vienasluoksnių ir daugiasluoksnių perceptronų neuroniniai tinklai. Dažniausiai naudojamą daugiasluoksnių perceptronų tipo neuroninį tinklą (Paveikslas 8) sudaro (Makūnaitė, R. 2006):

- Įvesties sluoksnis –neuronai, priimančys informaciją iš išorinių šaltinių ir siunčiantys ją apdoroti tinklui. Tai gali būti arba jutiklių įvestys, arba tinklo išorėje esančių sistemų siunčiami signalai.
- Paslėptasis sluoksnis –neuronai, priimančys informaciją iš įvesties sluoksnio ir apdorojantys ją tik jiems žinomu būdu. Šis sluoksnis tiesiogiai nesusijęs su išoriniu pasauliu, t.y. jungiasi tik su kitais neuroninio tinklo sluoksniais.
- Išvesties sluoksnis – neuronai, gaunantys apdorotą informaciją ir siunčiantys ją iš neuroninio tinklo.

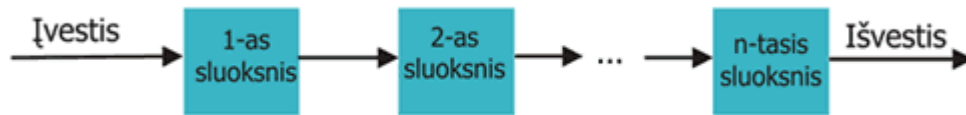


Šaltinis: MAKŪNAITĖ, R. (2006) Neuroniniai tinklai.

Paveikslas 8. Daugiasluoksnių perceptronų tipo neuroninis tinklas.

Atsižvelgiant į tai, kokia kryptimi siunčiami signalai, skiriami vienkrypčiai (asociatyvūs) ir grįžtamojo ryšio (autoasociatyvūs) tinklai.

Vienkrypčio ryšio neuroniniuose tinkluose vieno sluoksnio išvestys gali jungtis tik su kito sluoksnio įvestimis (Paveikslas 9). Neegzistuoja ryšiai tarp vieno sluoksnio išvesčių ir to paties ar prieš tai einančio sluoksnio įvesčių. Vieno sluoksnio išvestys susijusios su paskui einančio sluoksnio įvestimis (Makūnaitė, R. 2006).



Šaltinis: MAKŪNAITĖ, R. (2006) Neuroniniai tinklai.

Paveikslas 9. Vienkrypčio ryšio neuroninis tinklas.

Jeigu šakos svoris lygus nuliui, laikoma, kad tarp šakos jungiamųjų neuronų ryšio nėra. Paskutinio sluoksnio išvestys laikomos tinklo išvestimis.

Grįžtamojo ryšio neuroninio tinklo įvestis sudaro išorinės įvestys ir paties tinklo išvestis, kuriai būdingas tam tikras vėlinimas (Paveikslas 10). Puikūs grįžtamojo ryšio architektūros pavyzdžiai – Hopfieldo tinklas ir Boltzmano mašina (Makūnaitė, R. 2006).



Šaltinis: MAKŪNAITĖ, R. (2006) Neuroniniai tinklai.

Paveikslas 10. Grįžtamojo ryšio neuroninis tinklas.

Skiriami du neuroninių tinklų mokymo tipai – prižiūrimas ir neprižiūrimas.

Prižiūrimam mokymui reikalingas išorinis mokytojas (mokymas su mokytoju), valdantis mokymosi procesą ir teikiantis informaciją. Tai gali būti mokyti skirtų duomenų rinkinys arba stebėtojas, vertinantis neuroninio tinklo našumą. Prižiūrimo mokymo algoritmams skiriami mažiausio kvadratinio vidurkio, atgalinio skleidimo bei radialinės bazės funkcijos algoritmai. Prižiūrimo mokymo tikslas – priversti neuroninį tinklą pakeisti neuroninių jungčių svorius pagal etalonines įvestis ir išvestis. Mokymas baigiamas tinklui išmokus (galima minimali paklaida) sieti įvestis su išvestimis. Svarbus veiksnys – mokymo duomenų aibė, kuri turi būti suprantama ir privalo aprėpti visas praktines tinklo taikymo sritis. Taigi tinklas veiks gerai tik parinkus tinkamą mokymo aibę (Makūnaitė, R. 2006).

Neprižiūrimas mokymas neturi išorinio mokytojo (mokymas be mokytojo). Remdamasi vidiniais kriterijais ir tinklo informacija, sistema pati save turi suderinti. Tokiems neuroniniams tinklams pateikiami tik įvesčių pavyzdžiai, o sistema pati pagal požymius turi suklasifikuoti įvestis (Makūnaitė, R. 2006).

Kiekvienas iš šiame skyriuje išvardintų metodų turi savo privalumų ir trūkumų, todėl tiksliai nustatyti kurį metodą pasirinkti atpažinimo procesui, nėra lengva. Tam reikia gerai žinoti tiek sprendžiamą problemą, tiek gerai žinoti visų metodų savybes, taip pat svarbu žinoti kokie yra įėjimo duomenys ir laukiami rezultatai. Visgi sprendžiant sudėtingesnes atpažinimo problemas, vieno metodo panaudojimas gali neduoti laukiamo rezultato, o rezultatų tikslumas gali stipriai skirtis nuo lūkesčių. Todėl neretai siekiant didesnio klasifikavimo tikslumo, keli metodai yra jungiami į vieną ir tokiu būdu kuriami hibridiniai metodai, leidžiantys sumažinti objektų atpažinimo klaidos tikimybes. Tolesniuose skyriuose plačiau aptariamas tokių hibridinių metodų panaudojimas, įvardijami privalumai ir trūkumai.

2. HIBRIDINIAI METODAI

Hibridiniai metodai objektų atpažinime – tai dviejų ar daugiau klasifikavimo metodų rinkinys, kurio pagalba galima modeliuoti objektų atpažinimą tokiu pat arba dar didesniu tikslumu negu naudojant pavienius metodus. Naudojant hibridinius metodus objektų atpažinimo tikslumas priklauso nuo pasirinktų metodų derinio, nuo turimų ir laukiamų rezultatų kokybės ir kiekybės, pačios problemos esmės t.y. parenkant metodus, būtina atsižvelgti į jų tarpusavio suderinamumą, būtina atsižvelgti į turimų duomenų aibę ir žinoti kokie yra laukiami išvesties duomenys.

Hibridiniai metodai gaunami apjungiant tokius objektų atpažinimo metodus kaip tiesinis, kvadratinis klasifikatorius, neuroniniai tinklai bei visi parametriniai ir neparimetriniai klasifikatoriai. Plačiausiai hibridinius metodus yra išanalizavę mokslininkai Bunke, H. ir Kandel, A.² Jie siūlė įvairius neuroninių tinklų ir kitų metodų rinkinius tikslesniam objektų atpažinimui. Šiame darbe tyrimui pasirinktas tiesinės diskriminantės ir neuroninių tinklų derinys. Toliau bus aptartos priežastys kodėl pasirinktas būtent šis metodų rinkinys ir kodėl kuriamas būtent toks hibridinis metodas.

Neuroninis tinklas pasirinktas dėl paties metodo galimybės sukurti sudėtingą atpažinimo algoritmą ir modeliuoti sudėtingesnes atpažinimo kreives (funkcijas). Neuroniniuose tinkluose atpažinimo funkcijos gaunamos ne pagal išankstinius nusistatymus ar atsižvelgiant į pradinis duomenis, o pagal tai kaip tinklas sugeba prisitaikyti prie duomenų ir kaip pats tinklas išmoksta atpažinti tinkamus duomenis ir juos klasifikuoti. Neuroninis tinklas apmokomas naudojant mokymo imtį, kuri sudaroma iš įėjimo reikšmių kartu su atitinkamomis išėjimo reikšmėmis. Tokiu būdu, neuroninis tinklas mokosi nežinomos priklausomybės tarp jų. Jei tinklas tinkamai apmokytas, jis gali modeliuoti funkciją, siejančią įėjimo bei išėjimo kintamuosius. Taip neuroninis tinklas pagal įėjimo reikšmes gali išduoti nežinomas išėjimo reikšmes.

Mokymo proceso metu dažniausiai skaičiuojamas neuroninio tinklo darbą įvertinantis parametras. Tai gali būti neuroninio tinklo daroma klaida visiems duomenims, pavyzdžiui, skirtumas tarp tinklo išėjimo reikšmės ir norimos (užduoties) reikšmės arba kitoks neuroninį tinklą įvertinantis kriterijus.

Modeliuojant tradiciniais metodais (pavyzdžiui, naudojant tiesinį modeliavimą), įmanoma algoritmiškai nustatyti optimalius modelio parametrus, kurie absoliučiai minimizuoja klaidą. To dažniausiai negalima padaryti neuroniniam tinklui, kuris, bendru atveju, yra netiesinis modelis. Neuroninio tinklo svoriai nustatomi mokymo metu, iteraciniu (priartėjimo) keliu.

² BUNKE, Horst; KANDEL, Abraham. (2002) *Hybrid Methods in Pattern Recognition*.

Teisinis modeliavimas, kaip antroji hibridinio metodo komponentė pasirinkta dėl galimybės algoritmų pagalba nustatyti tinkamiausius modelio parametrus, kurie leidžia minimizuoti atpažinimo klaidą.

Tiesinės klasifikacijos uždavinys yra specifinis atpažinimo uždavinys, kurio esmė pagal pateiktus objekto (vaizdo, garso, proceso) skaitmeninius duomenis priskirti jį kokiai nors klasei. Klasifikacijos uždavinys skirtas sukurti duomenų žymėjimo algoritmą. Matematinė prasme klasifikatoriumi vadinamas bet koks vektorių atvaizdis į tam tikrą žymių (kategorijų) aibę. Kadangi praktiškai žymių skaičius nebūna labai didelis, todėl dažniausiai yra apskaičiuojamos funkcijų reikšmės, kurios yra interpretuojamos panašumo į atitinkamos kategorijos objektą ir išsirenkamas indeksas su didžiausia panašumo reikšme.

Klasifikavimo uždavinys yra skaidomas į dvi dalis. Pirmą yra surandamos objekto savybės (*angl. features*). Savybės yra apskaičiuojamos remiantis pradinio objektą aprašančiu vektoriumi. Parenkant objekto savybes yra ieškoma kompromiso tarp:

- Mažo požymių (savybių) skaičiaus;
- Kuo didesnio požymių informatyvumo.

Idealiu atveju iš savybių galima rekonstruoti pradinis objekto duomenis. Antrajame etape remiantis turimu požymių vektoriumi yra atliekama klasifikacija. Klasifikacija tiesinės diskriminantės metodu yra paremta aiškiu savybių išskyrimu, o pats metodas leidžia duomenų aibę išskirti panaudojant paprasčiausią tiesinę funkciją. Šio metodo paprastumas ir tuo pačiu gan aukštas atpažinimo tikslumas lėmė jo pasirinkimą šiam darbui.

Nors neuroninis tinklas ir tiesinė diskriminantė atrodo labai skirtingi ir nesuderinami metodai, bet jų gerosios savybės daro hibridinį metodą perspektyviu. Neuroninis tinklas sugeba modeliuoti sudėtingas klasifikavimo funkcijas, bet yra didesnė tikimybė atsirasti klaidai tinklui „persimokius“. Kad toks persimokymas neįvyktų, į atpažinimo procesą yra įtraukiamas tiesinės diskriminantės metodas, kuris įveda atpažinimo funkcijai tikslesnę kryptį ir tuo pačiu šiek tiek supaprastina atpažinimo procesą bei sumažina klaidos tikimybę. Šis dviejų atpažinimo metodų rinkinys tolesniuose skyriuose eksperimentinio tyrimo pagalba bus įvertintas ir palygintas su pavieniais metodais.

3. EKSPERIMENTINIS TYRIMAS

Eksperimentinio tyrimo pagalba siekiama patikrinti hibridinio metodo pranašumus prieš pavienius metodus, įvertinti objektų atpažinimo tikslumą ir iliustruoti analitinėje dalyje pateiktą medžiagą. Hibridinis metodas tyrimui gautas jungiant tiesinės diskriminantės ir neuroninių tinklų metodus ir trimis būdais įvertinant tyrimo eigoje sudarytus požymių vektorius. Visa eksperimentui reikalinga medžiaga surinkta tiesiogiai bendraujant su eksperimento dalyviais (eksperimento duomenų šaltiniais), o gauti duomenys apdoroti įvairių kompiuterinių programų pagalba, kurios (kaip ir visa tyrimo eiga) aprašomos tolimesniuose skyriuose.

3.1. Tyrimo duomenų rinkimas ir jų segmentavimas

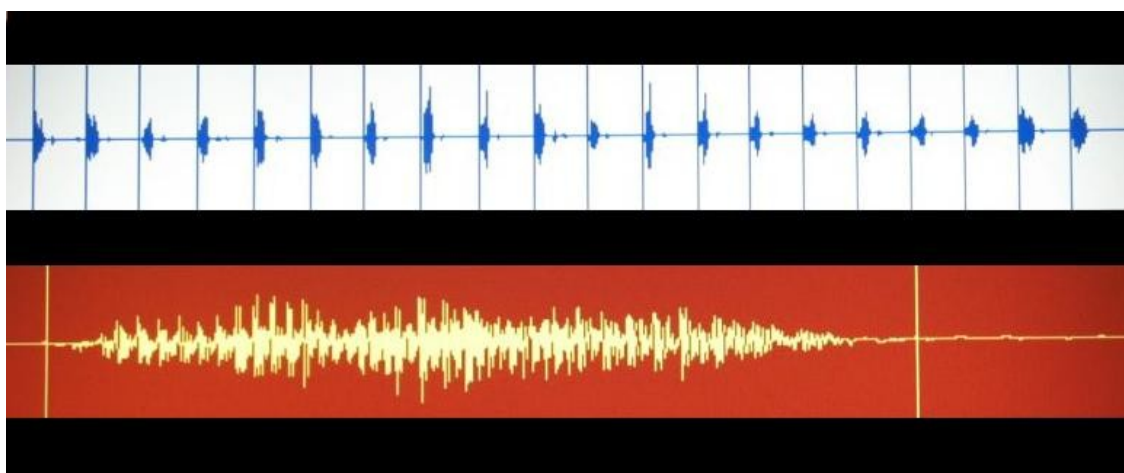
Medžiaga tiriamajai daliai rinkta su diktoriais bendraujant tiesiogiai (naudojama apklausos rūšis – interviu), balso įrašai rinkti mikrofono (ACME MK 200, palaikantis 20 – 16000 Hz dažnį, mikrofono jautrumas: $-58 \pm 2\text{dB}$) ir įrašų kūrimo programos Microsoft Sound Recorder 5.1 pagalba. Tyrimui parinktos gana paprastos frazės, bet pagal jų tarimo sudėtingumą ir įvairių balsių bei priebalsių maišymą, žodžiai pilnai atitinka tyrimo poreikius. Tyrimui surinktas 31 diktorius, jų uždavinys buvo sudiktuoti šias frazes:

- 0 (nulis);
- 1 (vienas);
- 2 (du);
- 3 (trys);
- 4 (keturi);
- 5 (penki);
- 6 (šeši);
- 7 (septyni);
- 8 (aštuoni);
- 9 (devyni);
- taip;
- ne;
- gal.

Kiekvienas diktorius aukščiau išvardintas frazes pakartoja po 20 kartų, taigi iš viso gaunama 620 vienos frazės (žodžio) variantų. Toks vienos frazės kiekis yra pakankamas atlikti pilnavertei žodžio atpažinimo analizei.

Tyrimo medžiagos rinkimo metu iš viso surinkta 8060 frazių su 31 diktoriaus pagalba. Visos 8060 frazių Java programavimo kalba parašyta programa SEG WORD (Paveikslas 11), segmentuotos, pažymint kiekvieno žodžio pradžią ir pabaigą, taigi duomenims apdoroti buvo padaryta 16120 žymų.

Žodžių segmentavimo pavyzdys pateiktas 11 paveiksle. Kaip matyti pateiktame piešinyje, programa suteikia galimybę išskirti atitinkamos atkarpos pradžią visoje laiko juostoje (baltame fone mėlyni svyravimai), o pažymėjus atitinkamą atkarpą, žemiau pateikiamas nuo pažymėtos srities sekundės trukmės fragmentas (raudoname fone geltoni svyravimai), kuriame ir nurodomos tikslios žodžio ar frazės ribos (pradžią ir pabaigą).

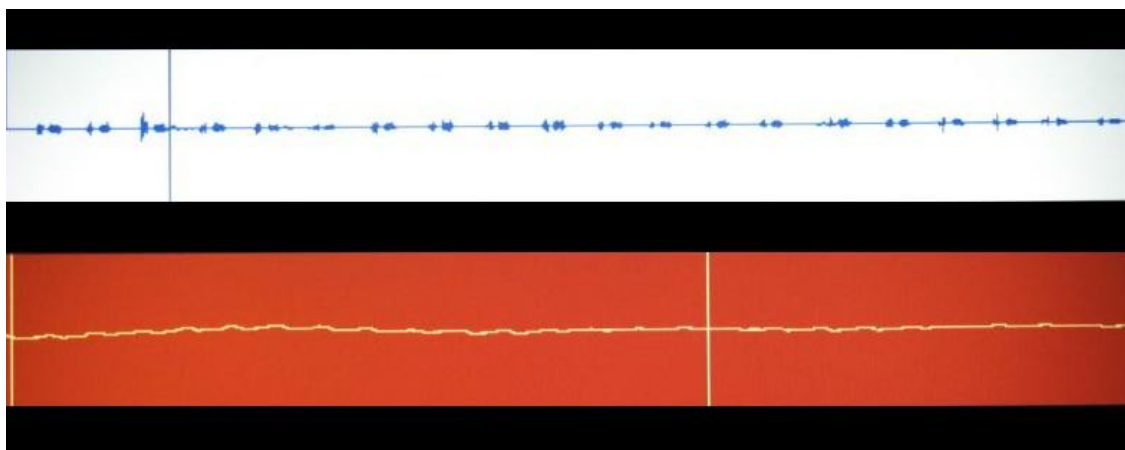


Šaltinis: sudaryta autoriaus.

Paveikslas 11. Žodžių indeksavimas SEG WORD pagalba.

Paveiksle matyti, kad laiko juostoje yra išskirta 20 ryškesnių svyravimų – tai ir yra vieno diktoriaus paeiliui 20 kartų tariama ta pati frazė. Šiuo atveju diktorius yra vyras, balso svyravimai yra ryškūs ir aiškiai išsiskiriantys iš tylos ir fono triukšmų, todėl išskirti žodžio pradžią ir pabaigą yra nesudėtinga.

Kur kas sudėtingiau žodžio išskyrimo procesas vyksta esant fono triukšmams ir diktoriaus įveliamoms klaidoms. Priklausomai nuo kiekvieno diktoriaus balsinių savybių taip pat kyla tokių problemų kaip įkvėpimas kalbėjimo metu, nereikalingo garso (triukšmo) išleidimas ar kalbėjimo manieros, tempas, tonas.



Šaltinis: sudaryta autoriaus.

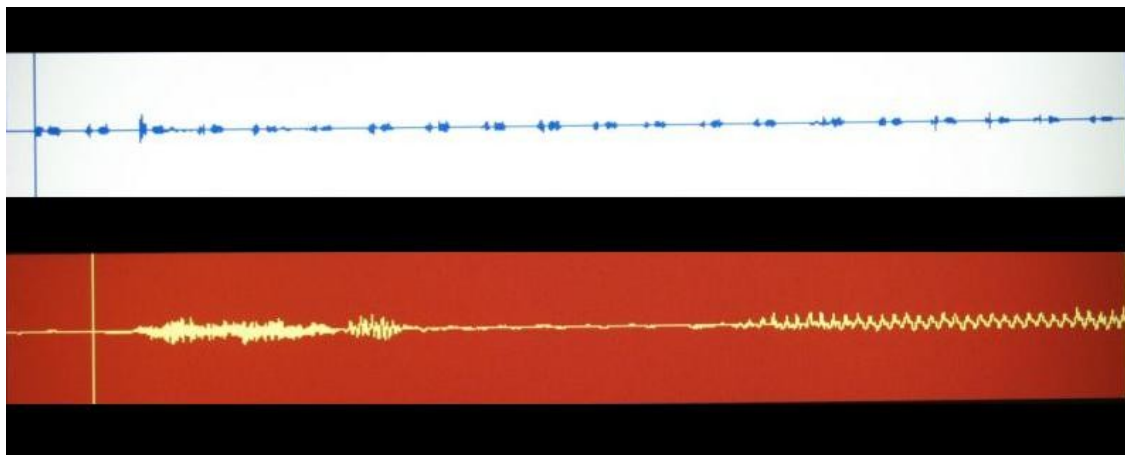
Paveikslas 12. Triukšmo išskyrimas žodžio fragmente.

Paveiksle 12 pateiktas diktorės moters netikėtas balso tono pakėlimas, kuris ryškiai išsiskiria tiek iš tariamos frazės srities, tiek iš tylėjimo atkarpos. Sekundės fragmente raudoname fone išskirti šio triukšmo svyravimai parodo tikslesnę šio triukšmo vietą.

Kaip matyti iš 11 ir 12 paveikslų, laiko juostoje tariamos frazės gana aiškiai atsispinti tolygiais bangų svyravimais ir yra išsidėstę vienodais intervalais vienas nuo kito, todėl garso atpažinimas, net ir esant triukšmams tampa dar lengvesnis. Kiekvienas garsas turi savo specifinį bangų išsidėstymą ir priklausomai nuo diktoriaus, gali nežymiai skirtis nuo kitų.

Paveiksle 11 pateiktas ryškus diktoriaus vyro tariamas fragmentas žodžio „nulis“, raudoname fone esančiuose svyravimuose matyti atskiri tariamos frazės parametrai ir tiksli žodžio atkarpa, paveiksle 12 šie svyravimai atrodo kiek kitaip, nes čia diktorė yra moteris, o įgarsinta frazė yra „septyni“.

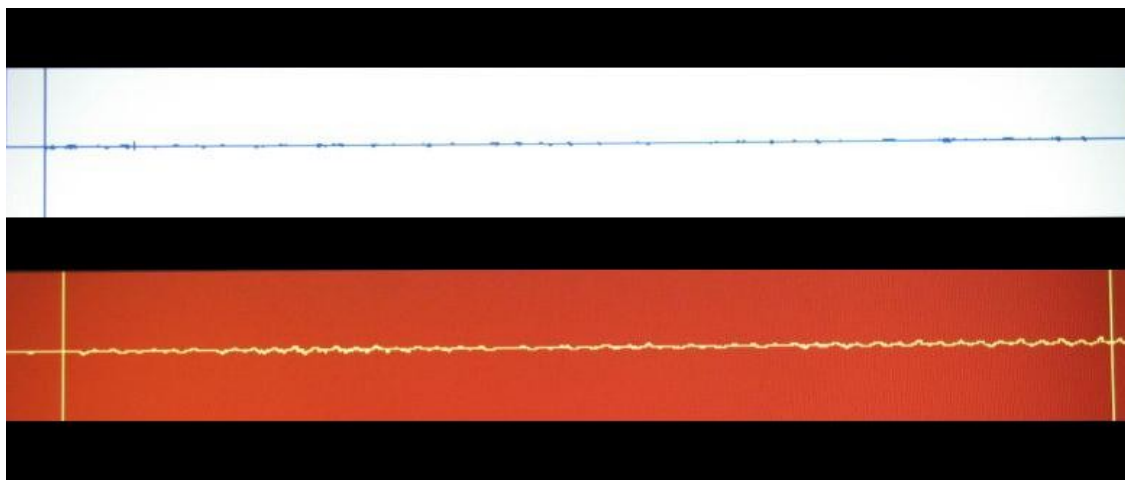
Kaip matome žodžio atkarpa susideda iš dviejų ryškių svyravimų su įterpta tyła (Paveikslas 13), o išdidinus svyravimų vaizdą, matosi kad kai kurie šio žodžio parametrai tariami tyliau. Toks kai kurių frazių išskirtinumas palengvina visą atpažinimo procesą



Šaltinis: sudaryta autoriaus.

Paveikslas 13. Žodis su tyliau ištartomis raidėmis.

Tyrimo medžiagos rinkimo metu susidurta ir su tokia problema, kaip per tylus diktorius balsas ir prasta įrašo kokybė, kuri puikiai atsispindi paveiksle 14. Šiuo atveju diktuosiu yra vyras ir tariama frazė – „devyni“.

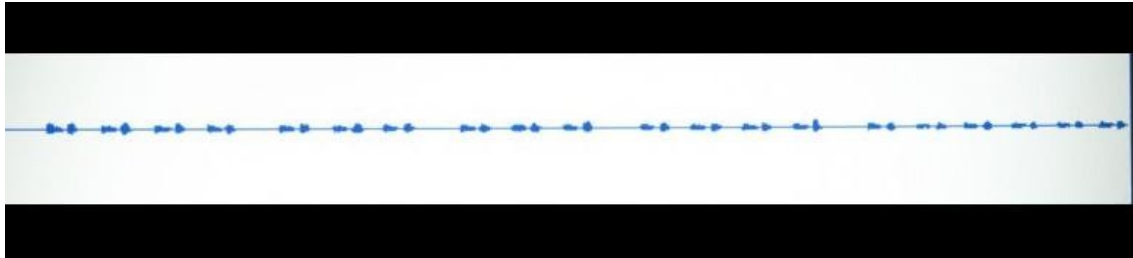


Šaltinis: sudaryta autoriaus.

Paveikslas 14. Diktorius tyliai tariamos frazės.

Kaip matyti paveiksle, žodis yra tariamas taip tyliai, kad laiko juostoje baltame fone neina išskirti tikslios reikiamo garso vietos, bangelių išsidėstymas yra chaotiškas, o balso atpažinimas tampa labai sudėtingas.

Šiuo atveju toks balso pavyzdys šiek tiek iškreipia visa frazės „devyni“ aibę, įterpdamas itin tylų šio garso pavyzdį, kuris eksperimente atpažinimo metu gali sukelti nepatogumų ir netikslumų dėl duomenų aibės išplėtimo. Kaip matyti iš aukščiau pateikto paveikslėlio, pirmieji žodžio „devyni“ bandymai dar yra matomi akimi laiko juostoje baltame fone, bet visi kiti išnyksta garsų visumoje ir atpažinimas tampa itin sudėtingas.

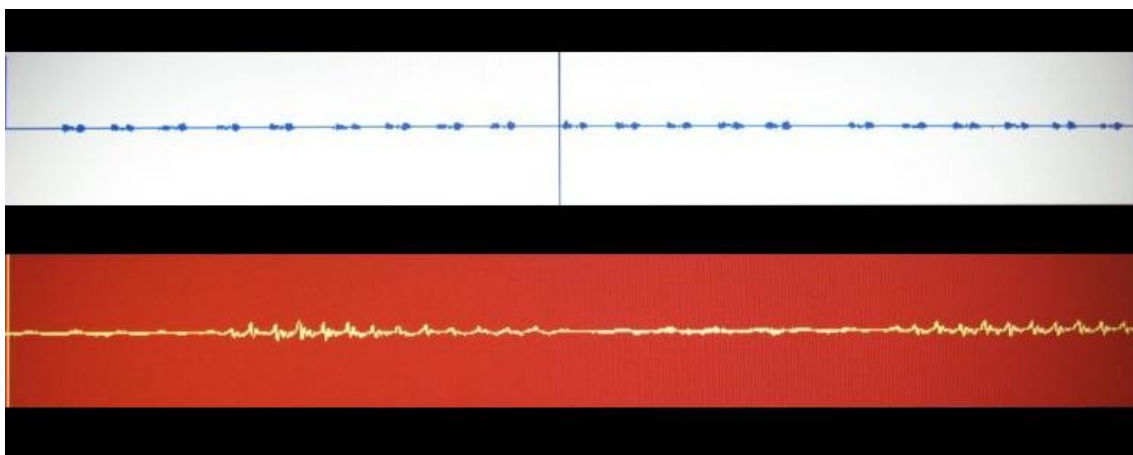


Šaltinis: sudaryta autoriaus.

Paveikslas 15. Įkvėpimo problema diktuojant frazes.

Dar viena problema, susijusi su medžiagos rinkimui tyrimui – tai netolygus diktorius kalbėjimas. Paveiksle 15 pateiktas pavyzdys diktorius, kuriam kilo problemų su tolygiu frazės pakartojimu 20 kartų ir su kvėpavimu.

Kaip matyti iš paveikslėlio, intervalai tarp žodžių yra nevienodi – tai įtakojo įkvėpimai pasakius 3-4 žodžiu. Šiuose tarpuose taip pat yra fiksuojamas triukšmas, kuris atsiranda dėl įkvėpimo metu sklaidžiamo garso ir tokiu būdu taip pat apsunkina visą atpažinimo procesą ir išplečia duomenų aibę.

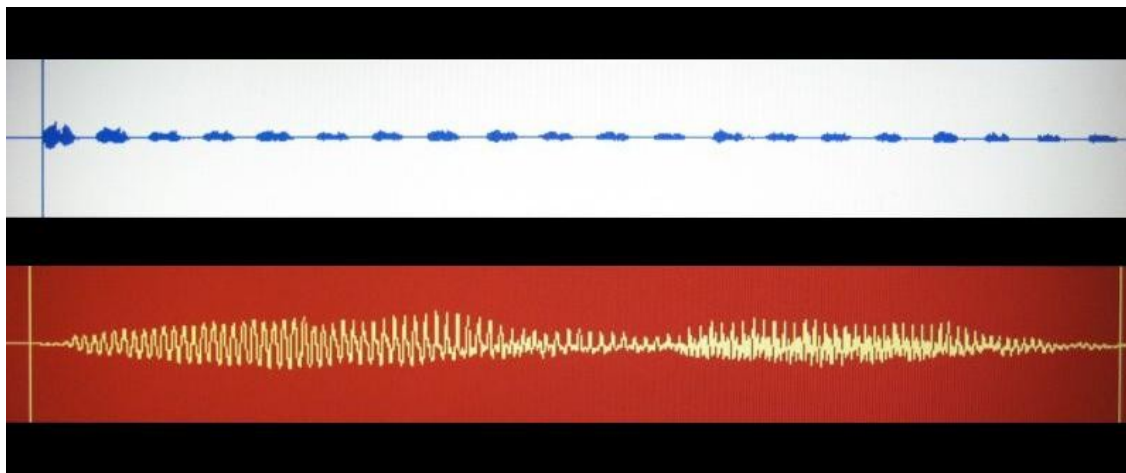


Šaltinis: sudaryta autoriaus.

Paveikslas 16. Netolygaus žodžių diktavimo pavyzdys.

Aukščiau esančiame paveiksle (Paveikslas 16) pateiktas dar vienas netolygaus žodžių diktavimo pavyzdys su triukšmų įsiterpimais tarp frazių, pažymėtais mėlynais taškais laiko juostoje.

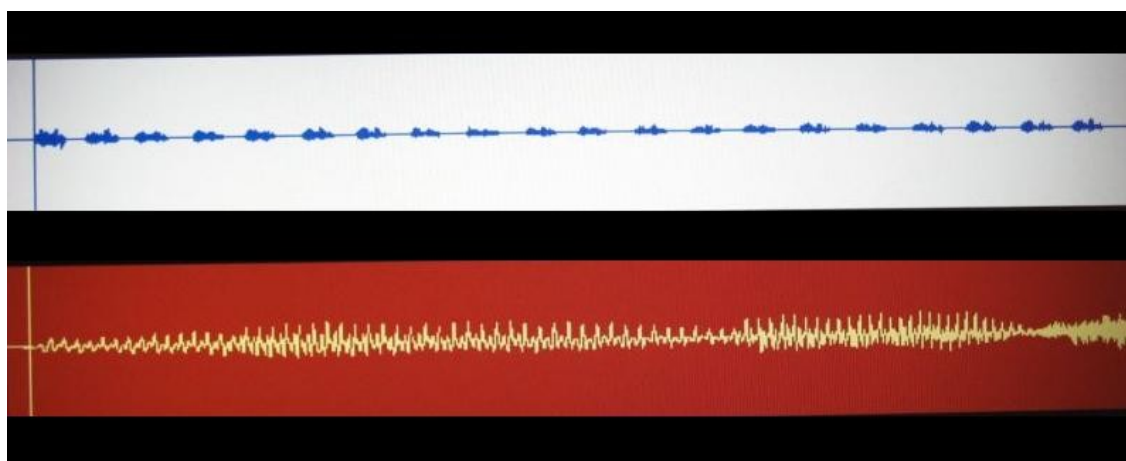
Kaip matyti iš raudoname fone esančių svyravimų, bangos skalė keičiasi nestipriai, todėl frazes išskirti iš tylos tampa dar sudėtingiau.



Šaltinis: sudaryta autoriaus.

Paveikslas 17. Diktoriaus vyro pirmųjų frazių dikcijos išskirtinumas.

Renkant medžiagą būsimam tyrimui pastebėta, kad beveik kiekvienas diktorius, pradeda diktuoti frazes kiek garsiau nei įprastai, todėl pirmieji 2-3 ištarti žodžiai išsiskiria iš kitų bangelių svyravimo atžvilgiu ir žodžio garsumu. Tai puikiai iliustruoja paveiksmai 17, 18 ir 19. Pirmosios frazės yra tariamos energingiau, o toliau, pagal normalią žmogaus dikciją diktavimo tempas šiek tiek sulėtėja ir nurimsta.



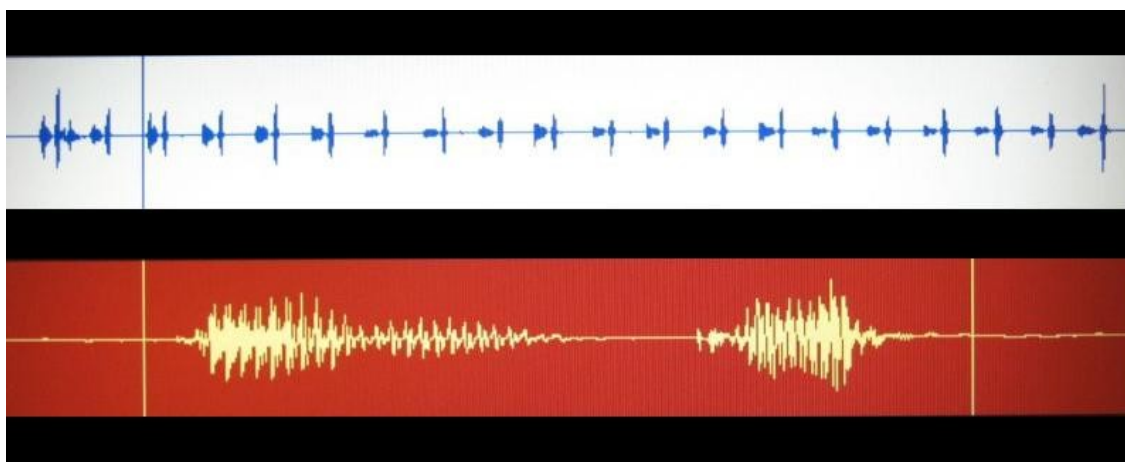
Šaltinis: sudaryta autoriaus.

Paveikslas 18. Diktorės moters pirmųjų frazių dikcijos išskirtinumas.

Pirmomis energingai ištartomis frazėmis renkant medžiagą pasižymėjo, ne tik vyrai diktoriai, bet ir moterys. Tiesa, moterų dikcija yra kiek tylesnė nei vyrų, todėl Paveiksle 18 pateiktas moters įrašytas balsas vaizdžiai išsiskiria pirmaisiais dviem ištartais žodžiais ir toliau tolygiai bals tembras krenta žemyn.

Paveiksluose 17 ir 18 pateikta ta pati diktuojama frazė „penki“ skiriasi gana žymiai, nes vyriškas balsas yra garsesnis ir išraiškingesnis. Kaip matyti abiejuose paveiksluose baltame fone esančių svyravimų forma yra panaši – žodžio viduryje pastebimas nežymus nutylėjimas, o tai šį žodį iš kitų daro išskirtinių.

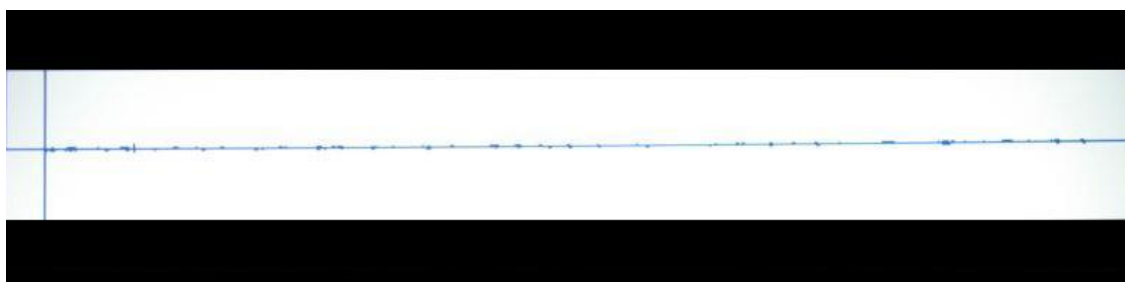
Kaip pastebima, toks požymis atsiranda visuose dviskiemeniuose skaitmenyse kaip „nulis“, „penki“ ir „šeši“. Šis požymis dar ryškiau išsiskiria paveiksle 19 pavaizduoto diktorius vyro tariamoje frazėje „penki“. Tiesa pirmame žodyje su garsiai išarta fraze atsiranda ir triukšmas, bet jis didelės reikšmės šiuo atveju nedaro, nes triukšmas atsiranda po frazės ir jį nesunkiai galima „nukirpti“.



Šaltinis: sudaryta autoriaus.

Paveikslas 19. Itin aiškus diktorius tarimas.

Iš visų surinktų diktorių frazių išskirtiniais galima pavadinti 5-6 diktorius, kurie daro lemiamą įtaką žodžių aibės plėtimui. Dviejų diktorių frazės buvo itin išraiškingos su intonacijomis ir balso variacijomis, todėl jų svyravimų skalė buvo plačiausia (Paveikslas 19).



Šaltinis: sudaryta autoriaus.

Paveikslas 20. Itin neaiškus diktorius tarimas.

Trijų diktorių tariamos frazės buvo itin tylios ir neaiškios (Paveikslas 20) – su jų išskyrimu kilo daugiausiai problemų. Taip pat vienas diktorius susidūrė su netolygiu diktavimu ir kvėpavimo problemomis 20 sekundžių kalbant be pertraukų.

Duomenų apdorojimo metu, vykdoma žodžių segmentacija, kuri pasitarnaus tolesnėje darbo eigoje. Segmentacijos laikų žymėjimai programos SEG WORD pagalba surašomi į tekstinį failą su *.rbb plėtiniu. Toliau šiuose failuose sukaupta informacija bus apdorojama atskirų frazių parametrų išskyrimui.

Žemiau pateiktoje lentelėje pavaizduota 20 kartų ištarto žodžio „taip“ segmentacija ir suskaičiuota kiekvieno žodžio tarimo trukmė sekundėmis. Šiuo atveju diktorius yra vyras su išraiškinga balso tonacija bei ilgu balso uždelstumu, todėl ir trumpo žodelio „taip“ tarimas tampa gana ilgas.

Lentelė 1

Diktoriaus vyro frazės „taip“ segmentai ir ilgiai

Eil. Nr.	Pradžia	Pabaiga	Ilgis (s)
1	37892	55440	0,48
2	102840	121508	0,52
3	169712	186512	0,47
4	229858	248312	0,51
5	295462	315622	0,56
6	356238	374264	0,50
7	424468	442816	0,51
8	494376	512404	0,50
9	564310	583990	0,55
10	629812	648478	0,52
11	700572	717424	0,47
12	776000	793332	0,48
13	841494	860908	0,54
14	910544	927664	0,48
15	977906	994812	0,47
16	1047702	1069036	0,59
17	1116520	1134494	0,50
18	1182642	1202374	0,55
19	1249828	1268334	0,51
20	1320920	1337132	0,45
Suma:			10,16

Šaltinis: sudaryta autoriaus.

Kaip matyti iš lentelėje 1 pateiktų duomenų, frazės tiesiogiai tariamos apie 10 sekundžių, o visas įrašas trunka apie 20 sekundžių, tai reiškia, kad kalbėjimo ir pauzės (tylos) viename įrašė yra po lygiai.

3.2. Garso ribų išskyrimas

Iš gautų duomenų toliau vykdomas frazių fonemų³ išskyrimas programos SEG PHON pagalba. Visas fonetinis apdorojimas vykdomas pasitelkiant šioje darbo dalyje surinktus ir apdorotus duomenis. Panaudojant gautą medžiagą atsiranda galimybė atlikti teorinėje dalyje nurodytų metodų palyginimą.

Diktorių kalbėjimo savybes pagal jų aiškumą ir suprantamumą būtų galima išskirti į tris grupes: puikiai suprantami, pakankamai aiškūs bei sunkiai arba visai nesuprantami. Iš 31 diktoriaus duomenų analizės metu pastebėta, kad tik 2-4 diktoriai pateko į puikiai suprantamų grupę, kurių balsinės savybės išsiskyrė iš kitų. Vienas iš tokių pavyzdžių pateiktas paveiksle žemiau:



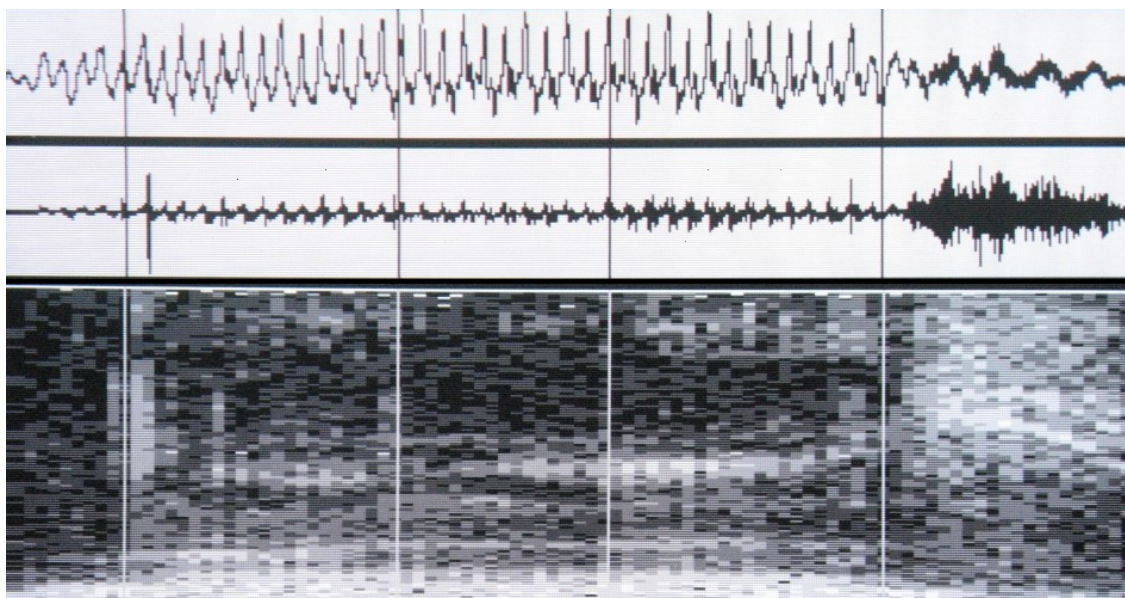
Šaltinis: sudaryta autoriaus.

Paveikslas 21. Žodžio „Šeši“ aiškų garso ribų išsidėstymas.

Paveiksle pavaizduotas vieno iš „geriausių“ diktorių tariama frazė „Šeši“. Jame atsispindi balsių (fonemų) „e“ ir „i“ ir duslumas (balta sritis antroje ir ketvirtoje zonoje apačioje) bei raidės „š“ šaižumas (Girdenis, A. 2003, p. 56).

³ Fonema – garsinės kalbos elementas, turintis skiriamąją (distinktyvinę) funkciją. Jeigu žodyje, pakeitus vieną kalbos garsą kitu, pasikeičia žodžio ar morfemos reikšmė, tai tokie garsai laikomi skirtingomis fonemomis (Vikižodynas).

Kaip paaiškėjo tyrimo metu, ne visa surinkta medžiaga buvo tinkama naudojimui. Ilgesni žodžiai, kaip „Septyni“, „Aštuoni“ ir „Devyni“ paprasčiausiai netilpo į SEG WORD skirtą sekundės intervalą ir darbo eigoje šias tris frazes teko atmesti dėl pernelyg mažo teisingai atpažintų garsų kiekio. Duomenų apdorojimo metu kilo ir daugiau problemų, jos bus aptartos šiame skyriuje.



Šaltinis: sudaryta autoriaus.

Paveikslas 22. Sunkiai suprantamos žodžio „Nulis“ garsų ribos.

Fonemos yra atskiriamos rankiniu būdu, todėl frazių aiškumas yra būtinas norint teisingai išskirti garsų pradžias ir pabaigas. Paveiksle 22 pavaizduotas sunkiai suprantamas žodis „Nulis“. Šioje vietoje ir išaiškėja kaip yra svarbu turėti kokybišką įrašą, nes visi triukšmai yra sužymėti apatinėje juostoje pilkais laukeliais.

Ne ką mažiau svarbios yra ir diktorius balsinės savybės. Prie sunkiai suprantamų diktorių galima priskirti 4-6, su kuriais turėta daugiausiai problemų fonemų atpažinimo metu. Lentelėje 2 pateikta visų pavykusių ir nepavykusių žodžių suvestinė, kuri atspindi ir išskiria problematiškiausius diktorius ir sunkiausiai apdorojamas frazes.

Kaip matyti žemiau esančioje lentelėje, X pažymėti laukeliai yra frazės, kurios nei vienas iš ištartų 20 žodžių buvo neatpažintas. Tai įtakojo tiek blogos balsinės diktorių savybės, tiek įrašų kokybė, tiek ir frazės sudėtingumas (žodžio „Taip“ pavyzdys).

Lentelė 2

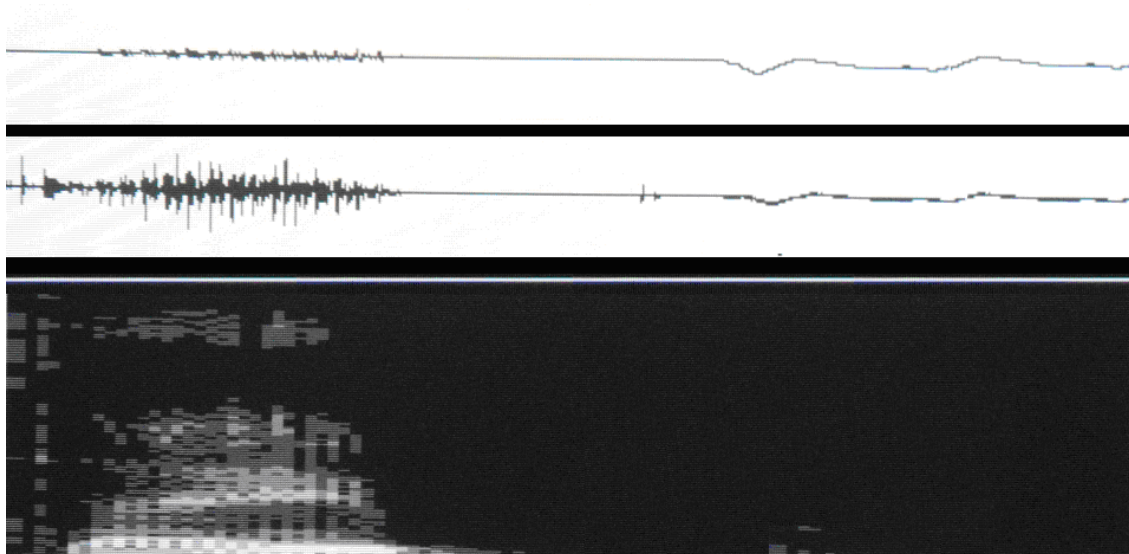
Apdorotų/nepavykusių frazių suvestinė

Frazė Diktorius	NULIS	VIENAS	DU	TRYS	KETURI	PENKI	ŠEŠI	TAIP	NE	GAL	Suma
# 1	+	+	+	+	+	+	+	+	+	+	10
# 2	+	+	+	+	+	+	+	+	+	+	10
# 3	+	+	+	+	+	+	+	X	+	+	9
# 4	+	+	+	+	+	+	+	+	+	+	10
# 5	+	+	+	X	+	+	+	X	+	X	7
# 6	+	+	+	X	+	+	+	+	+	+	9
# 7	+	+	+	+	+	+	+	+	+	+	10
# 8	+	+	+	+	+	+	+	X	+	+	9
# 9	+	+	+	+	+	+	+	X	+	+	9
# 10	+	+	+	+	+	+	+	+	+	+	10
# 11	+	+	+	+	+	+	+	X	+	+	9
# 12	+	+	+	+	+	+	+	+	+	X	9
# 13	+	+	+	+	+	+	+	X	+	+	9
# 14	+	+	+	+	+	+	+	X	+	X	8
# 15	+	X	+	+	+	+	+	X	+	+	8
# 16	+	X	+	+	+	+	+	X	+	+	8
# 17	+	+	+	+	+	+	+	X	+	+	9
# 18	+	+	+	+	+	+	+	X	+	+	9
# 19	+	+	+	+	+	+	+	+	+	+	10
# 20	+	+	+	+	+	+	+	X	+	+	9
# 21	+	+	+	+	+	+	+	X	+	+	9
# 22	+	+	+	+	+	+	+	X	+	+	9
# 23	+	+	+	+	+	+	+	X	+	X	8
# 24	+	+	+	+	+	+	+	X	+	X	8
# 25	+	+	+	+	+	+	+	X	+	+	9
# 26	+	+	+	+	+	+	+	X	+	+	9
# 27	+	+	+	+	+	+	X	X	+	+	8
# 28	+	+	+	+	+	+	+	X	+	+	9
# 29	+	+	+	+	+	+	+	X	+	+	9
# 30	+	+	+	+	+	+	+	X	+	+	9
# 31	+	+	+	+	+	+	+	X	+	+	9
Suma:	31	29	31	29	31	31	30	8	31	26	

Šaltinis: sudaryta autoriaus.

Lentelė puikiai parodo darbo metu pastebėtą ypatybę, jog tarp geriausių diktorių yra #4, #7, #10 ir #19, tarp blogiausių – #5, #14, #23 ir #24. Iš visų frazių išsiskiria žodeliai „Gal“ ir „Taip“. Tai įvyksta dėl priebalsių „l“ ir „p“ duslumo (Pakerys, A. 1995, p. 34), kuris neleidžia fonemos (garso) išskirti iš visumos.

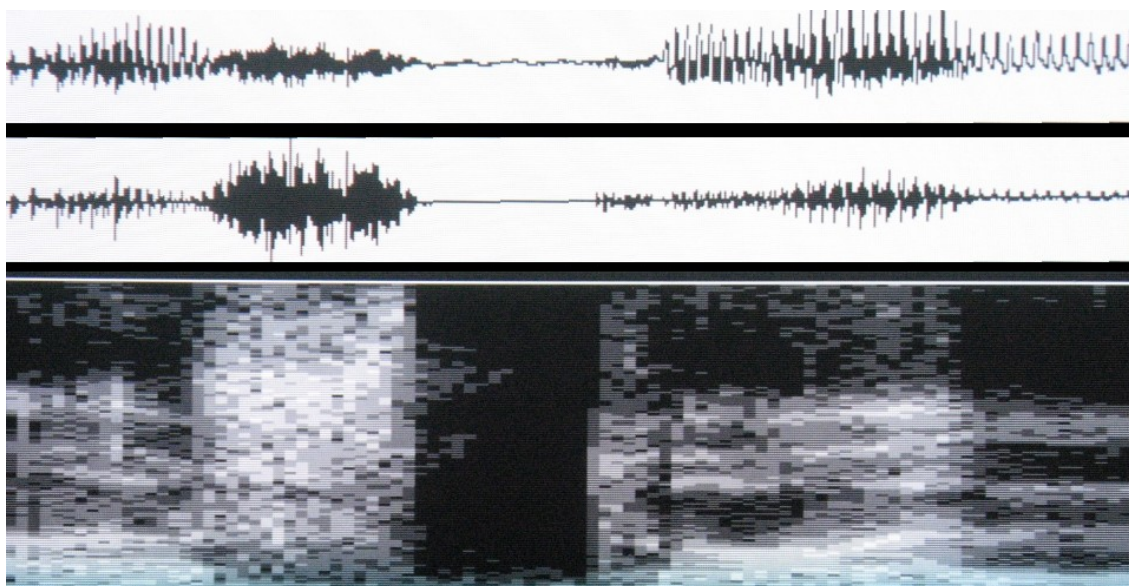
Žemiau pateiktame paveiksle pavaizduota žodžio „taip“ problema. Šiuo atveju išsiskiria tik dvibalsis „ai“, kuris yra matomas srities pradžioje, o raidės „t“ ir „p“ išnyksta tyloje arba triukšme.



Šaltinis: sudaryta autoriaus.

Paveikslas 23. Sunkiai fonetiškai atpažįstamas žodelis „Taip“.

Dar su viena problema susidurta pradėjus ilgesnių frazių apdorojimą, tokių kaip „Vienas“, „Keturi“, „Septyni“, „Aštuoni“, „Devyni“. Dauguma diktorių netilpo į SEG WORD programos leidžiamą sekundės intervalą vienam žodžiui, todėl didžioji daugumą ilgų žodžių buvo arba su „nukirpta“ pirmąją arba paskutiniąją fonema. Tokio žodžio pavyzdys pavaizduotas žemiau:

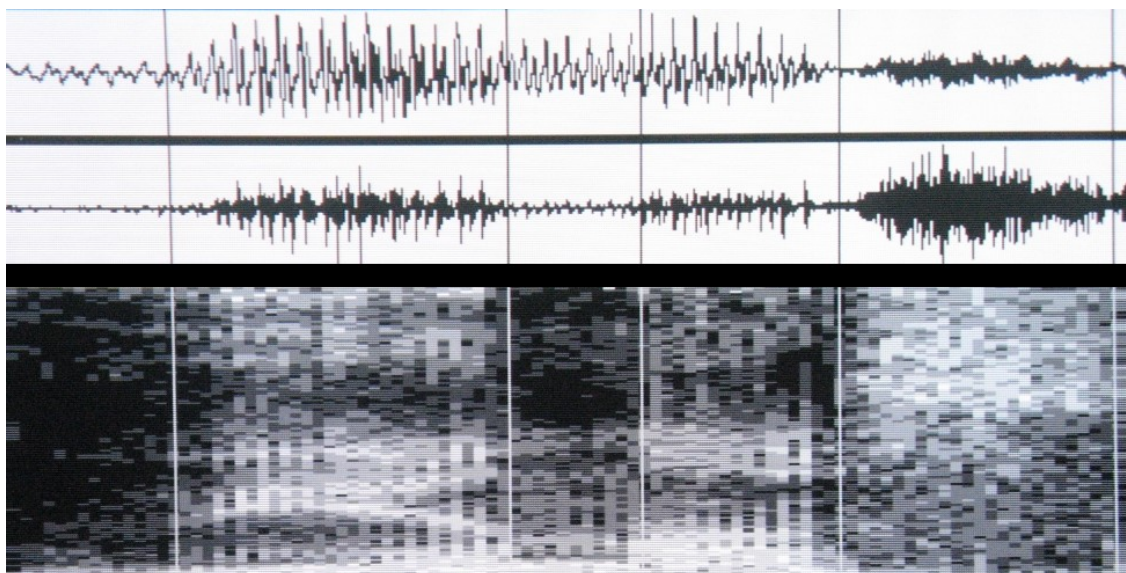


Šaltinis: sudaryta autoriaus.

Paveikslas 24. Žodis „Aštuoni“ su „nukirpta“ paskutine raide.

Kaip matyti iš paveikslo, aiškiai tariamos ir matomos yra raidės „a“, „š“, „t“, dvibalsis „uo“, o taip pat ir pusė garso „n“. Šiuo atveju fonema „i“ nebetelpa ir žodis tampa netinkamas analizei.

Per vidurį žodžio pastebimas keistas reiškinys – tyla, kuri pažymėta juoda spalva. Tylos tarpelis susidaro dėl to, kad po šaižiojo priebalsio „š“ yra tariamas kitas priebalsis „t“ ir tarp jų automatiškai atsiranda tylos intarpas. Taip pat galima pastebėti paveiksluose 21, 22, 24 ir 25 kaip iš visų garsų fonetiškai išsiskiria priebalsiai „š“ ir „s“. Jos viršutinėse garso bangos eilutėse pasižymi dažnesniu virpėjimu, o garsas tampa lengvai atpažįstamas.



Šaltinis: sudaryta autoriaus.

Paveikslas 25. Dvibalsis „ie“ žodyje „Vienas“.

Duomenų apdorojimo metu susidurta su dar vienu netikėtu reiškiniu – tai dvibalsių išskyrimas. Kadangi dvibalsiuose „ai“ ir „ie“ yra sudėtinga išskirti atskiras raides, todėl jie buvo traktuojami kaip vienas jungtinis garsas ir žymimi kaip viena tariama raidė. Paveiksle 25 šis reiškinys parodytas antroje fonemoje.

Iš viso fonetinio žymėjimo metu apdorota 10 frazių, iš jų gautą daugiau kaip 5000 žodžių ir padaryta daugiau kaip 22000 žymų ir išgauta 17 skirtingų fonemų iš kurių išskiriant garsines savybes – kovariacijas ir vidurkius – vykdomas klasifikatorių palyginimas.

3.3. Garso parametrų išskyrimas

Garso parametrų išskyrimo metu naudojama prieš tai esančiuose skyriuose aprašyta surinkta ir apdorota medžiaga. Norint apskaičiuoti kiekvienos fonemos savybes, reikia visas visų diktorių frazes sudėti į vieną eilę, prieš tai išskyrus kiekvieno garso pradžią ir pabaigą. Fonemų pradžios ir pabaigos tikslios vietos išskyrimas vykdomas programos MatLab pagalba. Programinis kodas, kurio pagalba yra atskiriama pirmoji žodžio raidė pateiktas žemiau:

```
k=1;
for j=1:count2/2
    zodis=rbb(j*2-1);
    for i=k:j
        raide=phn(i*interv-(interv-2));
        viskas1=y(zodis/2:raide/2);
        spc1=abs(fft(viskas1));
        fprintf(writeSPC1,'%f\n',spc1);
        k=i+1;
    end;
end;
fclose(writeSPC1);
```

Pirmoji raidė yra išskirtinė iš kitų tuo, kad SEG PHON programos pagalba yra žymimi garsų galai, o programos SEG WORD pagalba sužymima žodžio pradžia ir pabaiga. Automatiškai pirmoji raidė tampa garso intervalu tarp žodžio pradžios ir to garso pabaigos.

Likusios raidės yra lengviau atpažįstamos, nes kiekvieno garso pradžia yra prieš tai buvusio garso pabaiga. Žemiau pateikiamas programinis MatLab tekstas su likusių raidžių požymių išskyrimu

```
k=2;
for i=1:count1/interv
    viskas2=y(phn(k*3-4)/2:phn(k*3-1)/2);
    spc2=abs(fft(viskas2));
    fprintf(writeSPC2,'%f\n',spc2);
    k=k+kiekraidziu;
end;
fclose(writeSPC2);
```

Turint vieno žodžio visų fonemų pradžios ir pabaigos ribas, galima sukurti programinį kodą, kuris atskirtų visas likusių žodžių fonemas. Pilnas programinis fonemų požymių išskyrimo kodas pateiktas 1 Priede.

Išskyrus fonemų pradžias ir pabaigas belieka apskaičiuoti kiekvieno garso vidurkius ir kovariacijas. Jie randami programos MatLab pagalba, naudojant tokį sugeneruotą kodą:

```
fprintf(write, '%.f\n', spc01, spc02, spc03, spc04, spc05, spc06, spc07, spc08,
spc09, spc10, spc11, spc12, spc13, spc14, spc15, spc16, spc17, spc18, spc19,
spc20);
fclose(write);
read=fopen('F:\SEG_PHON\DATA\pasidedam.spc', 'r');
[spc, count]=fscanf(read, '%d');
fclose(read);
%<...>%
COVa=cov(spc)
MEANa=mean(spc)
fprintf(writeCOV, '%f\n', COVa);
fprintf(writeMEAN, '%f\n', MEANa);
fclose(writeCOV);
fclose(writeMEAN);
```

Pilnas kovariacijų ir vidurkių skaičiavimo kodas pateiktas 2 Priede. Tokiu būdu suskaičiuotos kovariacijų ir vidurkių reikšmės toliau naudojamos jau aprašytiems objektų atpažinimo ir klasifikacijos metodams, kurie panaudojami palyginti teorinėje dalyje aprašytus metodus.

Aukščiau esančiuose skyriuose aptartais būdais gauti požymių vektoriai vėlgi MatLab pagalba apdorojami ir taip gaunami galutiniai rezultatai. Galutinė tyrimo medžiaga gaunama tokiu MatLab programos kodu:

```
clear;
vector_size=12; %MFCC vektoriaus dydis 12
p=vector_size;
kart=100;
sweep=500;
r=p;
feat=vector_size;
l1l=[0:0.02:1];
LLL=1-l1l;
[s1, n1]=size(l1l);
%if 2*N-2<p r=2*N-2; end
    amvgetdata;
    for k=1:20 TEST(k, :)=A(k, :);
    end;
    for k=1:20 TEST(k+20, :)=B(k, :);
    end;
    for k=1:20 TEST(k+40, :)=C(k, :);
    end;
    for k=1:20 TEST(k+60, :)=D(k, :);
    end;
size(TEST)
ntest=size(TEST, 1);
tot_test=80;
amvregul;
amvpasuk;
```


Čia funkcija *amvregul* atlieka reguliarizuotą diskriminantinę analizę, o su *amvpaskuk* gaunamas rezultatas po neuroninio tinklo su pasukimu. Šių funkcijų pilni programiniai kodai pateikiami atitinkamai 3 ir 4 Prieduose.

3.4. Tyrimo rezultatai

Kiekviena tam tikro žodžio fonema tyrimo metu priskiriama jai būdingiems požymiams, o tiksliau jų vektoriams. Požymių vektoriai gaunami iš tam tikros fonemos centro parinkus 5 gretimus vektorius ir juos suvidurkinus (Driaunys, K. 2006). Toliau jie tiesinės diskriminantės ir neuroninio tinklo pagalba lyginami su atitinkamo metodo tos pašios srities vektoriais.

Hibridinio metodu atveju bendras panašumas imtas po 50% iš tiesinės diskriminantės ir neuroninio tinklo visų vektorių panašumų.

Ekspperimentui nebuvo panaudoti visi prieš tai surinkti ir apdoroti duomenys – iš visos aibės buvo atrinkta 19 geriausių diktorių, o tyrimas paremtas jų tariamomis frazėmis. Atpažinimo procese panaudota 9 diktorių tariamos frazės apmokymui, o 10 diktorių frazės – testavimui. Atpažinimo procesas pakartotas po 3 kartus su kiekvienu garsu ir iš visų vedamas bendras atpažinimo vidurkis, kuris atsispindi žemiau esančiose lentelėse.

Atpažinimo procese taip pat buvo panaudotos ne visos frazės – į tyrimą įtraukti tik skaičiai su nesikartojančiomis fonemomis, taigi viso tyrimo metu iš viso atpažinimo procese dalyvavo 13 lietuvių kalbos fonemų. Lentelėse 3-7 pateikti visi atpažinimo metu gauti rezultatai:

Lentelė 3

Žodžio „Nulis“ fonemų atpažinimo rezultatai

Metodas \ Fonema	N	U	L	I	S
Tiesinė diskriminantė	56%	61%	71%	48%	49%
Neuroninis tinklas	61%	72%	73%	55%	53%
Hibridinis metodas	63%	83%	77%	68%	52%

Šaltinis: sudaryta autoriaus.

Kaip matyti iš lentelėje 3 esančių atpažinimo rezultatų, fonemos „i“ ir „s“ buvo atpažintos su didžiausiomis klaidomis. Kadangi žodis „Nulis“ SEG WORD ir SEG PHON programomis apdorotas pats pirmas, todėl tokį palyginus nedidelį atpažinimo procentą įtakojo netikslus pradinių duomenų apdorojimas, kuris buvo įsisavintas tik tolesnėje darbo eigoje.

Kaip matyti jau su pirmuoju žodžiu matomas hibridinio metodo pranašumas tiek prieš tiesinę diskriminantę, tiek prieš neuroninį tinklą. Žodyje „Nulis“ visų 5 fonemų vidutinis atpažinimo tikslumas tiesinės diskriminantės metodu siekia 57%, neuroninio tinklo metodu – 63%, o hibridiniu metodu net 69%.

Lentelė 4

Žodžio „Du“ fonemų atpažinimo rezultatai

Metodas \ Fonema	D	U
Tiesinė diskriminantė	52%	63%
Neuroninis tinklas	63%	69%
Hibridinis metodas	68%	75%

Šaltinis: sudaryta autoriaus.

Lentelėje 4 pavaizduoti žodžio „Du“ atpažinimo rezultatai. Palyginus prieš tai gautus rezultatus su šiais, matyti, kad mažesnę kiekį garsų (fonemų) viename žodyje atskirti yra lengviau, nes trumpuose žodžiuose yra aiškiau apibrėžtos garsų (fonemų) ribos, o triukšmai šiuo atveju taip pat daro mažesnę įtaką garso atpažinimui.

Žodyje „Du“ abiejų fonemų vidutinis atpažinimo tikslumas tiesinės diskriminantės metodu siekia 58%, neuroninio tinklo metodu – 66%, o hibridiniu metodu net 72%. Pastebima, kad hibridinis metodas pateikia net 10% tikslesnes reikšmes negu tiesinė diskriminantė ir neuroninis tinklas kartu sudėjus.

Dar tikslesnis fonemų atpažinimas pastebimas žodyje „Trys“. Lentelė 5 jau leidžia daryti pradines atpažinimo išvadas, kad visos priebalsės tyrimo metu atpažįstamos su mažesnėmis paklaidomis nei balsės.

Lentelė 5

Žodžio „Trys“ fonemų atpažinimo rezultatai

Metodas \ Fonema	T	R	Y	S
Tiesinė diskriminantė	57%	74%	57%	53%
Neuroninis tinklas	67%	79%	69%	58%
Hibridinis metodas	64%	87%	73%	65%

Šaltinis: sudaryta autoriaus.

Kaip matyti iš aukščiau pateiktos lentelės, sunkiau atpažįstamos yra raidės „y“ ir „s“. Balsė „y“ tampa sunkiau atpažįstama dėl fonetinių jos tarimo ypatybių, o raidės „s“ mažesnis atpažinimo tikslumas gali būti įtakotas tuo, kad paskutinės žodžio raidės yra visada nutylinamos natūralioje kalboje, todėl ir atpažinimas šiuo atveju tampa sunkesnis.

Žodyje „Trys“ kaip ir žodyje „du“ visų fonemų vidutinis atpažinimo tikslumas yra gana panašus. Tiesinės diskriminantės metodu jis siekia 60%, neuroninio tinklo metodu – 68%, o hibridiniu metodu – 72%. Šiuo atveju hibridinis metodas neturėjo tokio ryškaus pranašumo prieš pirmuosius du metodus (o fonemos atpažinimas neuroniniu metodu gautas netgi truputį didesnis), lyginant su kitomis frazėmis, bet tai gali būti paaiškinama dideliu priebalsių kiekiu žodyje ir mažesne tikimybe atsirasti klaidai atpažinimo procese.

Lentelė 6

Žodžio „Penki“ fonemų atpažinimo rezultatai

Fonema \ Metodas	P	E	N	K	I
Tiesinė diskriminantė	61%	64%	59%	72%	54%
Neuroninis tinklas	70%	77%	63%	69%	60%
Hibridinis metodas	68%	84%	68%	76%	73%

Šaltinis: sudaryta autoriaus.

Žodyje „Penki“ visų 5 fonemų vidutinis atpažinimo tikslumas labai panašus į žodžio „Trys“ reikšmes: tiesinės diskriminantės metodu siekia 62%, neuroninio tinklo metodu – 68%, o hibridiniu metodu – 74%. Panašumas atsiranda vėlgi dėl didesnio kiekio priebalsių žodyje.

Kaip pavaizduota lentelėje 6, pirmoji raidė (žodžio „Trys“ atvejis) hibridinio metodo pagalba atpažįstama pora procentų prasčiau negu neuroninio tinklo metodu – tai galima paaiškinti netiksliu pradinių duomenų imtimi, nes priebalsį „p“ yra sunku išskirti iš tylos dėl jo duslumo savybės.

Didžiausias atpažinimo tikslumas pasiektas su žodžiu „Šeši“. Kaip matyti iš žemiau esančios lentelės duomenų (Lentelė 7), atpažinimo tikslumas viršija 80% ribą, o priebalsės „š“ atpažinimo atveju – net 90% ribą. Tai paaiškinama priebalsės šaižumo savybe, kuri aiškiai išskiria garsą tiek iš tylos, tiek iš kitų balsių ir priebalsių fono.

Lentelė 7

Žodžio „Šeši“ fonemų atpažinimo rezultatai

Fonema \ Metodas	Š	E	Š	I
Tiesinė diskriminantė	84%	72%	85%	58%
Neuroninis tinklas	92%	81%	91%	60%
Hibridinis metodas	95%	88%	97%	71%

Šaltinis: sudaryta autoriaus.

Lentelės duomenys parodo ženklų hibridinio metodo pranašumą prieš tiesinės diskriminantės metodą. Kaip ir prieš tai nagrinėtuose atvejuose, taip ir šiame išsiskiria paskutinės raidės nutylinimo savybė, kuri padidina atpažinimo klaidos tikimybę. Žodyje „Šeši“ visų 4 fonemų vidutinis atpažinimo tikslumas tiesinės diskriminantės metodu siekia 75%, neuroninio tinklo

metodu – 81%, o hibridiniu metodu net 88% - šie skaičiai dar kartą patvirtina hibridinio metodo pranašumą.

Tyrimo metu iš 45 lietuvių kalboje esančių priebalsinių fonemų (Kazlauskienė, A. 2008) panaudotos 9 („n“, „l“, „s“, „d“, „t“, „r“, „p“, „k“ ir „š“), o iš 12 balsinių fonemų (Kazlauskienė, A. 2008) panaudotos 4 („u“, „i“, „y“ ir „e“). Eksperimentinio tyrimo metu nustatyta, kad hibridinis metodas leidžia pasiekti vidutiniškai 6% didesnę atpažinimo tikslumą lyginant su neuroniniu tinklu ir net 12% didesnę tikslumą už tiesinės diskriminantės rezultatus. Tikslesni tyrimo skaičiavimai pateikti 5 priede. Šie skirtumai yra pakankamai reikšmingai dideli, nes bendras visų fonemų atpažinimo procentas siekė gana aukštą 69% rezultatą.

IŠVADOS

1. Darbe apibendrintos objektų atpažinimo procese kylančios problemos: pradinių duomenų tikslumas, daugiareikšmiškumas ir skirtingu klasių persidengimas;
2. Parodyta, kad skirtingų metodų panaudojimo efektyvumas priklauso nuo nagrinėjamos problemos sudėtingumo, skirtingiems uždaviniams labiau tinka skirtingi klasifikatoriai, didelę įtaką tikslumui turi apmokymo imtis ir duomenys;
3. Įvertinus objektų atpažinimo metodus, pasiūlytas hibridinis metodas, apjungiantis tiesinės diskriminantės ir neuroninio tinklo metodus. Toks hibridinis metodas, kai panaudojamos geriausios abiejų metodų savybės, potencialiai leistų sumažinti atpažinimo klaidą;
4. Lyginant standartinius objektų atpažinimo metodus su hibridiniais, parodyta, kad hibridiniai metodai leidžia pasiekti didesnę tikslumą atpažinimo procese – vidutinis teisingų atpažinimų skaičius su standartiniais metodais buvo 66%, o su hibridiniais – 75%.

LITERATŪROS SĄRAŠAS

1. BISHOP, Christopher. (2004) *Neural networks for pattern recognition*. Oxford university press Inc., New York. 1-73 p. ISBN 0-19-853864-2.
2. BISHOP, Christopher. (2006) *Pattern Recognition and Machine Learning*. Springer Science, Business Media, LLC, New York. 179 – 284 p. ISBN 0-38-731073-8.
3. BUNKE, Horst; KANDEL, Abraham. (2002) *Hybrid Methods in Pattern Recognition*. Series in Machine Perception and Artificial Intelligence – Vol. 47. World Scientific Publishing CO. Pte. Ltd., Singapore. 171-301 p. ISBN: 981-02-4832-6.
4. BURGESS, Christopher. (1998) *A Tutorial on Support Vector Machines for Pattern Recognition*. Kluwer Academic Publishers, Boston. 8-23 p.
5. DRIAUNYS, Kęstutis. (2006) *Lietuvių šnekamosios kalbos segmentavimo ir fonetinio atpažinimo tyrimas naudojant LTDIGITS garsyno įrašus*. Nepublikuota mokslų daktaro disertacija. Kaunas.
6. DUDA, Richard; HART, Peter; STORK, David. (2000) *Pattern Classification*. Wiley-Interscience, New York, 160-281 p. ISBN-10: 0471056693.
7. FRIEDMAN, Jerome; GEIGER, Dan; GOLDSZMIDT, Moises. (1997) *Bayesian Network Classifiers*. Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
8. FRIEDMAN, Jerome, HASTIE, Trevor, TIBSHIRANI, Robert. (2001) *The Elements of Statistical Learning*. Springer, New York, 79-114 p. ISBN-10: 0387952845.
9. FUKUNAGA, Keinosuke. (1990) *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 1-364 p. ISBN-10: 0122698517.
10. GIRDENIS, Aleksas. (2003) *Teoriniai lietuvių fonologijos pagrindai*. Mokslo ir enciklopedijų leidybos institutas, Vilnius. ISBN: 9785420015018.
11. HILL, Thomas; LEWICKI, Paul. (2007) *Statistics Methods and Applications*. [interaktyvus]. StatSoft, Tulsa, OK [žiūrėta 2009 m. lapkričio 23 d.]. Prieiga per Internetą: <<http://www.statsoft.com/textbook/k-nearest-neighbors/>>.

12. JAIN Anil; DUIN Robert; MAO Jianchang. (2000) *Statistical Pattern Recognition: a Review*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, 8-20 p.
13. KAZLAUSKIENĖ, Asta. (2008) *Dabartinės lietuvių kalbos fonetika*. Kauno technologijos universitetas. Kaunas: Technologija. ISBN 978-9955-25-572-7.
14. MAKŪNAITĖ, Rūta. (2006) *Neuroniniai tinklai* [interaktyvus]. Elektronika - elektronikos, informacinių ir ryšių technologijų portalas [žiūrėta 2010 m. gegužės 2 d.], , Prieiga per Internetą: <<http://www.elektronika.lt/theory/theme/160/4342/>>.
15. MANNING, Christopher; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. (2008) *Introduction to Information Retrieval* [interaktyvus]. Cambridge University Press, ISBN: 0521865719 [žiūrėta 2010 m. sausio 9 d.]. Ištrauka iš knygos Internetė: <<http://nlp.stanford.edu/IR-book/html/htmledition/k-nearest-neighbor-1.html>>.
16. MCLACHLAN, Geoffrey. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore, 27-96 p. ISBN 0-471-61531-5.
17. PAKERYS, Antanas. (1995) *Lietuvių bendrinės kalbos fonetika*. Vadovėlis respublikos aukštųjų mokyklų lietuvių kalbos ir literatūros specialybių studentams. Mokslas, Vilnius.
18. ŠARKIŪNAITĖ, Ingrida; KRIKŠČIŪNIENĖ, Dalia; SIMUTIS, Rimvydas. (2007) *Magistro baigiamojo darbo rengimo tvarka. Metodiniai nurodymai*. VU KHF Informatikos katedros Verslo informatikos (62109P101) ir Verslo informacijos sistemų (62103S138) studijų programų studentams, Vilniaus universiteto leidykla, Kaunas.
19. Vikipedija – laisvoji enciklopedija, straipsnis *Minimum mean square error* [interaktyvi duomenų bazė]. Atnaujinta 2010 m. gegužės 15 d. [žiūrėta 2010 m. gegužės 20 d.]. Prieiga per Internetą: <http://en.wikipedia.org/wiki/Minimum_mean_square_error>.
20. Vikižodynas – laisvasis žodynas, [interaktyvi duomenų bazė]. Puslapis *Fonema* [žiūrėta 2010 m. gegužės 20 d.]. Prieiga per Internetą: <<http://lt.wiktionary.org/wiki/fonema>>.
21. WEBB, Andrew. (2002) *Statistical Pattern Recognition*. Wiley-Interscience, Chichester West Sussex, , 1-223 p. ISBN-10: 0470845139.

PRIEDAI

1 PRIEDAS. Pilnas požymių išskyrimo kodas	57
2 PRIEDAS. Pilnas kovariacijų ir vidurkių skaičiavimo kodas	59
3 PRIEDAS. Pilnas funkcijos <i>amvregul</i> kodas	61
4 PRIEDAS. Pilnas funkcijos <i>amvpassuk</i> kodas	63
5 PRIEDAS. Galutiniai eksperimentinio tyrimo skaičiavimai.....	67

Pilnas požymių išskyrimo kodas

```

y=wavread('F:\SEG_PHON\DATA\KETURI\#1 (4).wav'); %keiciam
readPHN=fopen('F:\SEG_PHON\DATA\KETURI\#1 (4).phn','r'); %keiciam
readRBB=fopen('F:\SEG_PHON\DATA\KETURI\#1 (4).rbb','r'); %keiciam
writeZ=fopen('F:\SEG_PHON\DATA\KETURI\#1 Z.spc','w'); %keiciam
writeSPC1=fopen('F:\SEG_PHON\DATA\KETURI\#1 R1.spc','w'); %keiciam
writeSPC2=fopen('F:\SEG_PHON\DATA\KETURI\#1 R2.spc','w'); %keiciam
writeSPC3=fopen('F:\SEG_PHON\DATA\KETURI\#1 R3.spc','w'); %keiciam
writeSPC4=fopen('F:\SEG_PHON\DATA\KETURI\#1 R4.spc','w'); %keiciam
writeSPC5=fopen('F:\SEG_PHON\DATA\KETURI\#1 R5.spc','w'); %keiciam
writeSPC6=fopen('F:\SEG_PHON\DATA\KETURI\#1 R6.spc','w'); %keiciam
kiekraidziu=6; %keiciam
[phn,count1]=fscanf(readPHN,'%d');
fclose(readPHN);
[rbb,count2]=fscanf(readRBB,'%d');
fclose(readRBB);
interv=kiekraidziu*3;
%-----1 RAIDE-----%
k=1;
for j=1:count2/2
    zodis=rbb(j*2-1);
    for i=k:j
        raide=phn(i*interv-(interv-2));
        viskas1=y(zodis/2:raide/2);
        spc1=abs(fft(viskas1));
        fprintf(writeSPC1,'%f\n',spc1);
        k=i+1;
    end;
end;
fclose(writeSPC1);
%-----2 RAIDE-----%
k=2;
for i=1:count1/interv
    viskas2=y(phn(k*3-4)/2:phn(k*3-1)/2);
    spc2=abs(fft(viskas2));
    fprintf(writeSPC2,'%f\n',spc2);
    k=k+kiekraidziu;
end;
fclose(writeSPC2);
%-----3 RAIDE-----%
k=3;
for i=1:count1/interv
    viskas3=y(phn(k*3-4)/2:phn(k*3-1)/2);
    spc3=abs(fft(viskas3));
    fprintf(writeSPC3,'%f\n',spc3);
    k=k+kiekraidziu;
end;
fclose(writeSPC3);
%-----4 RAIDE-----%
k=4;
for i=1:count1/interv
    viskas4=y(phn(k*3-4)/2:phn(k*3-1)/2);
    spc4=abs(fft(viskas4));
    fprintf(writeSPC4,'%f\n',spc4);
    k=k+kiekraidziu;
end;
fclose(writeSPC4);
%-----5 RAIDE-----%

```

1 PRIEDAS (TĘSINYS)

```
k=5;
for i=1:count1/interv
    viskas5=y(phn(k*3-4)/2:phn(k*3-1)/2);
    spc5=abs(fft(viskas5));
    fprintf(writeSPC5,'%f\n',spc5);
    k=k+kiekraidziu;
end;
fclose(writeSPC5);
%-----6 RAIDE-----%
k=6;
for i=1:count1/interv
    viskas6=y(phn(k*3-4)/2:phn(k*3-1)/2);
    spc6=abs(fft(viskas6));
    fprintf(writeSPC6,'%f\n',spc6);
    k=k+kiekraidziu;
end;
fclose(writeSPC6);
%-----ZODIS-----%
for p=1:count2/2
    viskasZ=y(rbb(p*2-1)/2:rbb(p*2)/2);
    spcZ=abs(fft(viskasZ));
    fprintf(writeZ,'%f\n',spcZ);
end
fclose(writeZ);
```

Pilnas kovariacijų ir vidurkių skaičiavimo kodas

```

read01=fopen('F:\SEG_PHON\DATA\KETURI\#1 R1.spc','r'); %keiciam R
read02=fopen('F:\SEG_PHON\DATA\KETURI\#2 R1.spc','r'); %keiciam R
read03=fopen('F:\SEG_PHON\DATA\KETURI\#3 R1.spc','r'); %keiciam R
read04=fopen('F:\SEG_PHON\DATA\KETURI\#4 R1.spc','r'); %keiciam R
read05=fopen('F:\SEG_PHON\DATA\KETURI\#5 R1.spc','r'); %keiciam R
read06=fopen('F:\SEG_PHON\DATA\KETURI\#6 R1.spc','r'); %keiciam R
read07=fopen('F:\SEG_PHON\DATA\KETURI\#7 R1.spc','r'); %keiciam R
read08=fopen('F:\SEG_PHON\DATA\KETURI\#8 R1.spc','r'); %keiciam R
read09=fopen('F:\SEG_PHON\DATA\KETURI\#9 R1.spc','r'); %keiciam R
read10=fopen('F:\SEG_PHON\DATA\KETURI\#10 R1.spc','r'); %keiciam R
read11=fopen('F:\SEG_PHON\DATA\KETURI\#11 R1.spc','r'); %keiciam R
read12=fopen('F:\SEG_PHON\DATA\KETURI\#12 R1.spc','r'); %keiciam R
read13=fopen('F:\SEG_PHON\DATA\KETURI\#13 R1.spc','r'); %keiciam R
read14=fopen('F:\SEG_PHON\DATA\KETURI\#14 R1.spc','r'); %keiciam R
read15=fopen('F:\SEG_PHON\DATA\KETURI\#15 R1.spc','r'); %keiciam R
read16=fopen('F:\SEG_PHON\DATA\KETURI\#16 R1.spc','r'); %keiciam R
read17=fopen('F:\SEG_PHON\DATA\KETURI\#17 R1.spc','r'); %keiciam R
read18=fopen('F:\SEG_PHON\DATA\KETURI\#18 R1.spc','r'); %keiciam R
read19=fopen('F:\SEG_PHON\DATA\KETURI\#19 R1.spc','r'); %keiciam R
read20=fopen('F:\SEG_PHON\DATA\KETURI\#20 R1.spc','r'); %keiciam R
writeCOV=fopen('F:\SEG_PHON\DATA\Mean_Cov\TAIP\COV_20 R1.spc','w'); %keiciam R
writeMEAN=fopen('F:\SEG_PHON\DATA\Mean_Cov\TAIP\MEAN_20 R1.spc','w'); %keiciam R
write=fopen('F:\SEG_PHON\DATA\pasidedam.spc','w');
[spc01,count01]=fscanf(read01,'%d');
[spc02,count02]=fscanf(read02,'%d');
[spc03,count03]=fscanf(read03,'%d');
[spc04,count04]=fscanf(read04,'%d');
[spc05,count05]=fscanf(read05,'%d');
[spc06,count06]=fscanf(read06,'%d');
[spc07,count07]=fscanf(read07,'%d');
[spc08,count08]=fscanf(read08,'%d');
[spc09,count09]=fscanf(read09,'%d');
[spc10,count10]=fscanf(read10,'%d');
[spc11,count11]=fscanf(read11,'%d');
[spc12,count12]=fscanf(read12,'%d');
[spc13,count13]=fscanf(read13,'%d');
[spc14,count14]=fscanf(read14,'%d');
[spc15,count15]=fscanf(read15,'%d');
[spc16,count16]=fscanf(read16,'%d');
[spc17,count17]=fscanf(read17,'%d');
[spc18,count18]=fscanf(read18,'%d');
[spc19,count19]=fscanf(read19,'%d');
[spc20,count20]=fscanf(read20,'%d');
fprintf(write,'%f\n',spc01, spc02, spc03, spc04, spc05, spc06, spc07, spc08,
spc09, spc10, spc11, spc12, spc13, spc14, spc15, spc16, spc17, spc18, spc19,
spc20);
fclose(write);
read=fopen('F:\SEG_PHON\DATA\pasidedam.spc','r');
[spc,count]=fscanf(read,'%d');
fclose(read);
fclose(read01);
fclose(read02);
fclose(read03);
fclose(read04);
fclose(read05);
fclose(read06);
fclose(read07);

```

2 PRIEDAS (TĘSINYS)

```
fclose(read08);
fclose(read09);
fclose(read10);
fclose(read11);
fclose(read12);
fclose(read13);
fclose(read14);
fclose(read15);
fclose(read16);
fclose(read17);
fclose(read18);
fclose(read19);
fclose(read20);
COVa=cov(spc)
MEANa=mean(spc)
fprintf(writeCOV, '%f\n', COVa);
fprintf(writeMEAN, '%f\n', MEANa);
fclose(writeCOV);
fclose(writeMEAN);
```

Pilnas funkcijos *amvregul* kodas

```

%disp('Matricu ismatavimai');
%disp([size(A);size(B);size(C);size(D);size(TEST)]);

weab=ma-mb; weac=ma-mc; wead=ma-md; webc=mb-mc; webd=mb-md; wecd=mc-md;
v=0.25*(ma+mb+mc+md);

A=A-ones(nmt,1)*v; B=B-ones(nmt,1)*v;
C=C-ones(nmt,1)*v; D=D-ones(nmt,1)*v;

% testavimas
TEST=TEST-ones(ntest,1)*v;

disp('Pradedu pseudoinversija');

% diskriminantiniu funkciju sudarymas
S=0.5*(cov(A)+cov(B));
[UAB,DA,V]=svd(S); dsab=diag(DA);
PSinv=UAB(:,1:r)*inv(DA(1:r,1:r))*UAB(:,1:r)'; % Pseudoinversija
WPFAB=PSinv*weab'; WPFAB=[WPFAB',0];

S=0.5*(cov(A)+cov(C));
[UAC,DA,V]=svd(S); dsac=diag(DA);
PSinv=UAC(:,1:r)*inv(DA(1:r,1:r))*UAC(:,1:r)'; % Pseudoinversija
WPFAC=PSinv*weac'; WPFAC=[WPFAC',0];

S=0.5*(cov(A)+cov(D));
[UAD,DA,V]=svd(S); dsad=diag(DA);
PSinv=UAD(:,1:r)*inv(DA(1:r,1:r))*UAD(:,1:r)'; % Pseudoinversija
WPFAD=PSinv*wead'; WPFAD=[WPFAD',0];

S=0.5*(cov(B)+cov(C));
[UBC,DA,V]=svd(S); dsbc=diag(DA);
PSinv=UBC(:,1:r)*inv(DA(1:r,1:r))*UBC(:,1:r)'; % Pseudoinversija
WPFBC=PSinv*webc'; WPFBC=[WPFBC',0];

S=0.5*(cov(B)+cov(D));
[UBD,DA,V]=svd(S); dsbd=diag(DA);
PSinv=UBD(:,1:r)*inv(DA(1:r,1:r))*UBD(:,1:r)'; % Pseudoinversija
WPFBD=PSinv*webd'; WPFBD=[WPFBD',0];

S=0.5*(cov(C)+cov(D));
[UCD,DA,V]=svd(S); dscd=diag(DA);
PSinv=UCD(:,1:r)*inv(DA(1:r,1:r))*UCD(:,1:r)'; % Pseudoinversija
WPFCD=PSinv*wecd'; WPFCD=[WPFCD',0];

disp('Testuoju pseudoinversija');
PFi=testm(WPFAB,WPFAC,WPFAD,WPFBC,WPFBD,WPFCD,tot_test,feat,TEST);
disp('Pseudoinversijos rezultatai');
disp([PFi;(PFi/80.)]);
%pause;

disp('Pradedu reguliarizuota analize');
% reguliarizuota diskriminantine analize:
D2AB=1./(LLL'*dsab'+1111'*ones(1,p)); D2AC=1./(LLL'*dsac'+1111'*ones(1,p));
D2AD=1./(LLL'*dsad'+1111'*ones(1,p)); D2BC=1./(LLL'*dsbc'+1111'*ones(1,p));
D2BD=1./(LLL'*dsbd'+1111'*ones(1,p)); D2CD=1./(LLL'*dscd'+1111'*ones(1,p));

```

3 PRIEDAS (TĘSINYS)

```
for j=1:n1
    %disp(['speak;j']);

WAB=[weab*UAB*diag(D2AB(j,:))*UAB', 0]; WAC=[weac*UAC*diag(D2AC(j,:))*UAC', 0];
WAD=[wead*UAD*diag(D2AD(j,:))*UAD', 0]; WBC=[webc*UBC*diag(D2BC(j,:))*UBC', 0];
WBD=[webd*UBD*diag(D2BD(j,:))*UBD', 0]; WCD=[wecd*UCD*diag(D2CD(j,:))*UCD', 0];

% pratestuojam su regularizuota matrica
PFL1(j)=testm(WAB,WAC,WAD,WBC,WBD,WCD,tot_test,feat,TEST);
    %disp('Regularizuotos matricos rezultatai');
    %disp([PFL1(j); (PFL1(j)/40.)]);
end

[Preg1,nreg]=min(PFL1); % randu geriausia liambda reiksme
disp('Geriausi regularizuotos matricos rezultatai');
disp([Preg1; (Preg1/80.)]);
%pause;

disp('Pradedu neuronini tinkla');

[etest,swtest]=percvrea(A,B,C,D,sweep,0.2,0,zeros(1,p+1),TEST,1.03,tot_test,feat);
[PP,ppv]=min(etest);
disp('Geriausi neuroninio tinklo rezultatai');
disp([PP; (PP/80.)]);
%pause;
```

Pilnas funkcijos *amvpassuk* kodas

```

kart=25; k11=1; k12=25; sweepm=400;
U2AB=sqrtm(UAB); U4AB=sqrtm(U2AB); U8AB=sqrtm(U4AB); U16AB=sqrtm(U8AB);
ULAB=real(U16AB*U2AB*U4AB*U8AB*U16AB);
U2AC=sqrtm(UAC); U4AC=sqrtm(U2AC); U8AC=sqrtm(U4AC); U16AC=sqrtm(U8AC);
ULAC=real(U16AC*U2AC*U4AC*U8AC*U16AC);
U2AD=sqrtm(UAD); U4AD=sqrtm(U2AD); U8AD=sqrtm(U4AD); U16AD=sqrtm(U8AD);
ULAD=real(U16AD*U2AD*U4AD*U8AD*U16AD);
U2BC=sqrtm(UBC); U4BC=sqrtm(U2BC); U8BC=sqrtm(U4BC); U16BC=sqrtm(U8BC);
ULBC=real(U16BC*U2BC*U4BC*U8BC*U16BC);
U2BD=sqrtm(UBD); U4BD=sqrtm(U2BD); U8BD=sqrtm(U4BD); U16BD=sqrtm(U8BD);
ULBD=real(U16BD*U2BD*U4BD*U8BD*U16BD);
U2CD=sqrtm(UCD); U4CD=sqrtm(U2CD); U8CD=sqrtm(U4CD); U16CD=sqrtm(U8CD);
ULCD=real(U16CD*U2CD*U4CD*U8CD*U16CD);

disp('Pradedu pasukima');
for jk=k11:k12
    if jk==1 BTAB=eye(p); BTAC=eye(p); BTAD=eye(p);
        BTBC=eye(p); BTBD=eye(p); BTCD=eye(p);
        end
    if jk==2 BTAB=real(U16AB); BTAC=real(U16AC); BTAD=real(U16AD);
        BTBC=real(U16BC); BTBD=real(U16BD); BTCD=real(U16CD);
        end
    if jk==3 BTAB=real(U8AB); BTAC=real(U8AC); BTAD=real(U8AD);
        BTBC=real(U8BC); BTBD=real(U8BD); BTCD=real(U8CD);
        end
    if jk==4 BTAB=real(U16AB*U8AB); BTAC=real(U16AC*U8AC);
        BTAD=real(U16AD*U8AD); BTBC=real(U16BC*U8BC);
        BTBD=real(U16BD*U8BD); BTCD=real(U16CD*U8CD);
        end
    if jk==5 BTAB=real(U4AB); BTAC=real(U4AC); BTAD=real(U4AD);
        BTBC=real(U4BC); BTBD=real(U4BD); BTCD=real(U4CD);
        end
    if jk==6 BTAB=real(U4AB*U16AB); BTAC=real(U4AC*U16AC);
        BTAD=real(U4AD*U16AD); BTBC=real(U4BC*U16BC);
        BTBD=real(U4BD*U16BD); BTCD=real(U4CD*U16CD);
        end
    if jk==7 BTAB=real(U4AB*U8AB); BTAC=real(U4AC*U8AC);
        BTAD=real(U4AD*U8AD); BTBC=real(U4BC*U8BC);
        BTBD=real(U4BD*U8BD); BTCD=real(U4CD*U8CD);
        end
    if jk==8 BTAB=real(U4AB*U16AB*U8AB); BTAC=real(U4AC*U16AC*U8AC);
        BTAD=real(U4AD*U16AD*U8AD); BTBC=real(U4BC*U16BC*U8BC);
        BTBD=real(U4BD*U16BD*U8BD); BTCD=real(U4CD*U16CD*U8CD);
        end
    if jk==9 BTAB=real(U2AB); BTAC=real(U2AC); BTAD=real(U2AD);
        BTBC=real(U2BC); BTBD=real(U2BD); BTCD=real(U2CD);
        end
    if jk==10 BTAB=real(U2AB*U16AB); BTAC=real(U2AC*U16AC);
        BTAD=real(U2AD*U16AD); BTBC=real(U2BC*U16BC);
        BTBD=real(U2BD*U16BD); BTCD=real(U2CD*U16CD);
        end
    if jk==11 BTAB=real(U2AB*U8AB); BTAC=real(U2AC*U8AC);
        BTAD=real(U2AD*U8AD); BTBC=real(U2BC*U8BC);
        BTBD=real(U2BD*U8BD); BTCD=real(U2CD*U8CD);
        end
    if jk==12 BTAB=real(U2AB*U8AB*U16AB); BTAC=real(U2AC*U8AC*U16AC);
        BTAD=real(U2AD*U8AD*U16AD); BTBC=real(U2BC*U8BC*U16BC);
        BTBD=real(U2BD*U8BD*U16BD); BTCD=real(U2CD*U8CD*U16CD);

```

4 PRIEDAS (TĚSINYS 1)

```
        end
    if jk==13 BTAB=real(U2AB*U4AB); BTAC=real(U2AC*U4AC);
        BTAD=real(U2AD*U4AD); BTBC=real(U2BC*U4BC);
        BTBD=real(U2BD*U4BD); BTCD=real(U2CD*U4CD);
        end
    if jk==14 BTAB=real(U2AB*U4AB*U16AB); BTAC=real(U2AC*U4AC*U16AC);
        BTAD=real(U2AD*U4AD*U16AD); BTBC=real(U2BC*U4BC*U16BC);
        BTBD=real(U2BD*U4BD*U16BD); BTCD=real(U2CD*U4CD*U16CD);
        end
    if jk==15 BTAB=real(U2AB*U4AB*U8AB); BTAC=real(U2AC*U4AC*U8AC);
        BTAD=real(U2AD*U4AD*U8AD); BTBC=real(U2BC*U4BC*U8BC);
        BTBD=real(U2BD*U4BD*U8BD); BTCD=real(U2CD*U4CD*U8CD);
        end
    if jk==16 BTAB=real(U2AB*U4AB*U8AB*U16AB);
        BTAC=real(U2AC*U4AC*U8AC*U16AC);
        BTAD=real(U2AD*U4AD*U8AD*U16AD);
        BTBC=real(U2BC*U4BC*U8BC*U16BC);
        BTBD=real(U2BD*U4BD*U8BD*U16BD);
        BTCD=real(U2CD*U4CD*U8CD*U16CD);
        end
    if jk==17 BTAB=UAB; BTAC=UAC; BTAD=UAD; BTBC=UBC; BTBD=UBD; BTCD=UCD;
        end
    if jk==18 BTAB=real(UAB*U16AB); BTAC=real(UAC*U16AC);
        BTAD=real(UAD*U16AD); BTBC=real(UBC*U16BC);
        BTBD=real(UBD*U16BD); BTCD=real(UCD*U16CD);
        end
    if jk==19 BTAB=real(UAB*U8AB); BTAC=real(UAC*U8AC); BTAD=real(UAD*U8AD);
        BTBC=real(UBC*U8BC); BTBD=real(UBD*U8BD); BTCD=real(UCD*U8CD);
        end
    if jk==20 BTAB=real(UAB*U16AB*U8AB); BTAC=real(UAC*U16AC*U8AC);
        BTAD=real(UAD*U16AD*U8AD); BTBC=real(UBC*U16BC*U8BC);
        BTBD=real(UBD*U16BD*U8BD); BTCD=real(UCD*U16CD*U8CD);
        end
    if jk==21 BTAB=real(UAB*U4AB); BTAC=real(UAC*U4AC); BTAD=real(UAD*U4AD);
        BTBC=real(UBC*U4BC); BTBD=real(UBD*U4BD); BTCD=real(UCD*U4CD);
        end
    if jk==22 BTAB=real(UAB*U4AB*U16AB); BTAC=real(UAC*U4AC*U16AC);
        BTAD=real(UAD*U4AD*U16AD); BTBC=real(UBC*U4BC*U16BC);
        BTBD=real(UBD*U4BD*U16BD); BTCD=real(UCD*U4CD*U16CD);
        end
    if jk==23 BTAB=real(UAB*U4AB*U8AB); BTAC=real(UAC*U4AC*U8AC);
        BTAD=real(UAD*U4AD*U8AD); BTBC=real(UBC*U4BC*U8BC);
        BTBD=real(UBD*U4BD*U8BD); BTCD=real(UCD*U4CD*U8CD);
        end
    if jk==24 BTAB=real(UAB*U4AB*U16AB*U8AB); BTAC=real(UAC*U4AC*U16AC*U8AC);
        BTAD=real(UAD*U4AD*U16AD*U8AD); BTBC=real(UBC*U4BC*U16BC*U8BC);
        BTBD=real(UBD*U4BD*U16BD*U8BD); BTCD=real(UCD*U4CD*U16CD*U8CD);
        end
    if jk==25 BTAB=real(UAB*U2AB); BTAC=real(UAC*U2AC); BTAD=real(UAD*U2AD);
        BTBC=real(UBC*U2BC); BTBD=real(UBD*U2BD); BTCD=real(UCD*U2CD);
        end
end

for j=1:n1
    %disp([jk;j]);
    WAB=[weab*BTAB*diag(D2AB(j,:))*BTAB',0];
    WAC=[weac*BTAC*diag(D2AC(j,:))*BTAC',0];
    WAD=[wead*BTAD*diag(D2AD(j,:))*BTAD',0];
    WBC=[webc*BTBC*diag(D2BC(j,:))*BTBC',0];
    WBD=[webd*BTBD*diag(D2BD(j,:))*BTBD',0];
    WCD=[wecd*BTCD*diag(D2CD(j,:))*BTCD',0];
end
```


4 PRIEDAS (TĘSINYS 2)

```
% testavimas
PFLs(jk,j)=testm(WAB,WAC,WAD,WBC,WBD,WCD,tot_test,feat,TEST);
%disp([PFLs(jk,j);PFLs(jk,j)/40.]);

end
end

[Psuk,j]=min(min(PFLs)); [Psuka,jk]=min(PFLs(:,j)); PFLs(jk,j);
disp('Minimalus rezultatas su pasukimu');
disp([Psuk;Psuk/40.]);
%pause;
if jk==1 BTAB=eye(p); BTAC=eye(p); BTAD=eye(p);
    BTBC=eye(p); BTBD=eye(p); BTCD=eye(p);
    end
if jk==2 BTAB=real(U16AB); BTAC=real(U16AC); BTAD=real(U16AD);
    BTBC=real(U16BC); BTBD=real(U16BD); BTCD=real(U16CD);
    end
if jk==3 BTAB=real(U8AB); BTAC=real(U8AC); BTAD=real(U8AD);
    BTBC=real(U8BC); BTBD=real(U8BD); BTCD=real(U8CD);
    end
if jk==4 BTAB=real(U16AB*U8AB); BTAC=real(U16AC*U8AC);
    BTAD=real(U16AD*U8AD); BTBC=real(U16BC*U8BC);
    BTBD=real(U16BD*U8BD); BTCD=real(U16CD*U8CD);
    end
if jk==5 BTAB=real(U4AB); BTAC=real(U4AC); BTAD=real(U4AD);
    BTBC=real(U4BC); BTBD=real(U4BD); BTCD=real(U4CD);
    end
if jk==6 BTAB=real(U4AB*U16AB); BTAC=real(U4AC*U16AC);
    BTAD=real(U4AD*U16AD); BTBC=real(U4BC*U16BC);
    BTBD=real(U4BD*U16BD); BTCD=real(U4CD*U16CD);
    end
if jk==7 BTAB=real(U4AB*U8AB); BTAC=real(U4AC*U8AC);
    BTAD=real(U4AD*U8AD); BTBC=real(U4BC*U8BC);
    BTBD=real(U4BD*U8BD); BTCD=real(U4CD*U8CD);
    end
if jk==8 BTAB=real(U4AB*U16AB*U8AB); BTAC=real(U4AC*U16AC*U8AC);
    BTAD=real(U4AD*U16AD*U8AD); BTBC=real(U4BC*U16BC*U8BC);
    BTBD=real(U4BD*U16BD*U8BD); BTCD=real(U4CD*U16CD*U8CD);
    end
if jk==9 BTAB=real(U2AB); BTAC=real(U2AC); BTAD=real(U2AD);
    BTBC=real(U2BC); BTBD=real(U2BD); BTCD=real(U2CD);
    end
if jk==10 BTAB=real(U2AB*U16AB); BTAC=real(U2AC*U16AC);
    BTAD=real(U2AD*U16AD); BTBC=real(U2BC*U16BC);
    BTBD=real(U2BD*U16BD); BTCD=real(U2CD*U16CD);
    end
if jk==11 BTAB=real(U2AB*U8AB); BTAC=real(U2AC*U8AC);
    BTAD=real(U2AD*U8AD); BTBC=real(U2BC*U8BC);
    BTBD=real(U2BD*U8BD); BTCD=real(U2CD*U8CD);
    end
if jk==12 BTAB=real(U2AB*U8AB*U16AB); BTAC=real(U2AC*U8AC*U16AC);
    BTAD=real(U2AD*U8AD*U16AD); BTBC=real(U2BC*U8BC*U16BC);
    BTBD=real(U2BD*U8BD*U16BD); BTCD=real(U2CD*U8CD*U16CD);
    end
if jk==13 BTAB=real(U2AB*U4AB); BTAC=real(U2AC*U4AC);
    BTAD=real(U2AD*U4AD); BTBC=real(U2BC*U4BC);
    BTBD=real(U2BD*U4BD); BTCD=real(U2CD*U4CD);
    end
if jk==14 BTAB=real(U2AB*U4AB*U16AB); BTAC=real(U2AC*U4AC*U16AC);
    BTAD=real(U2AD*U4AD*U16AD); BTBC=real(U2BC*U4BC*U16BC);
    BTBD=real(U2BD*U4BD*U16BD); BTCD=real(U2CD*U4CD*U16CD);
    end
end
```

4 PRIEDAS (TĚSINYS 3)

```
if jk==15 BTAB=real(U2AB*U4AB*U8AB); BTAC=real(U2AC*U4AC*U8AC);
    BTAD=real(U2AD*U4AD*U8AD); BTBC=real(U2BC*U4BC*U8BC);
    BTBD=real(U2BD*U4BD*U8BD); BTCD=real(U2CD*U4CD*U8CD);
end
if jk==16 BTAB=real(U2AB*U4AB*U8AB*U16AB);
    BTAC=real(U2AC*U4AC*U8AC*U16AC);
    BTAD=real(U2AD*U4AD*U8AD*U16AD);
    BTBC=real(U2BC*U4BC*U8BC*U16BC);
    BTBD=real(U2BD*U4BD*U8BD*U16BD);
    BTCD=real(U2CD*U4CD*U8CD*U16CD);
end
if jk==17 BTAB=UAB; BTAC=UAC; BTAD=UAD;
    BTBC=UBC; BTBD=UBD; BTCD=UCD;
end
if jk==18 BTAB=real(UAB*U16AB); BTAC=real(UAC*U16AC);
    BTAD=real(UAD*U16AD); BTBC=real(UBC*U16BC);
    BTBD=real(UBD*U16BD); BTCD=real(UCD*U16CD);
end
if jk==19 BTAB=real(UAB*U8AB); BTAC=real(UAC*U8AC); BTAD=real(UAD*U8AD);
    BTBC=real(UBC*U8BC); BTBD=real(UBD*U8BD); BTCD=real(UCD*U8CD);
end
if jk==20 BTAB=real(UAB*U16AB*U8AB); BTAC=real(UAC*U16AC*U8AC);
    BTAD=real(UAD*U16AD*U8AD); BTBC=real(UBC*U16BC*U8BC);
    BTBD=real(UBD*U16BD*U8BD); BTCD=real(UCD*U16CD*U8CD);
end
if jk==21 BTAB=real(UAB*U4AB); BTAC=real(UAC*U4AC); BTAD=real(UAD*U4AD);
    BTBC=real(UBC*U4BC); BTBD=real(UBD*U4BD); BTCD=real(UCD*U4CD);
end
if jk==22 BTAB=real(UAB*U4AB*U16AB); BTAC=real(UAC*U4AC*U16AC);
    BTAD=real(UAD*U4AD*U16AD); BTBC=real(UBC*U4BC*U16BC);
    BTBD=real(UBD*U4BD*U16BD); BTCD=real(UCD*U4CD*U16CD);
end
if jk==23 BTAB=real(UAB*U4AB*U8AB); BTAC=real(UAC*U4AC*U8AC);
    BTAD=real(UAD*U4AD*U8AD); BTBC=real(UBC*U4BC*U8BC);
    BTBD=real(UBD*U4BD*U8BD); BTCD=real(UCD*U4CD*U8CD);
end
if jk==24 BTAB=real(UAB*U4AB*U16AB*U8AB); BTAC=real(UAC*U4AC*U16AC*U8AC);
    BTAD=real(UAD*U4AD*U16AD*U8AD); BTBC=real(UBC*U4BC*U16BC*U8BC);
    BTBD=real(UBD*U4BD*U16BD*U8BD); BTCD=real(UCD*U4CD*U16CD*U8CD);
end
if jk==25 BTAB=real(UAB*U2AB); BTAC=real(UAC*U2AC); BTAD=real(UAD*U2AD);
    BTBC=real(UBC*U2BC); BTBD=real(UBD*U2BD); BTCD=real(UCD*U2CD);
end

% gaunam pasukimo koeficientus
TAB=BTAB*sqrtm(diag(D2AB(j,:))); TAC=BTAC*sqrtm(diag(D2AC(j,:)));
TAD=BTAD*sqrtm(diag(D2AD(j,:))); TBC=BTBC*sqrtm(diag(D2BC(j,:)));
TBD=BTBD*sqrtm(diag(D2BD(j,:))); TCD=BTCD*sqrtm(diag(D2CD(j,:)));

disp('Pradedam neuronini tinkla su pasukimu');

[etest,swtest]=percpasu(A,B,C,D,TAB,TAC,TAD,TBC,TBD,TCD,sweepm,0.2,0,zeros(1,p+1),TEST,1.03,tot_test,feat);
[PPT,ppv]=min(etest);

disp('Minimalus rezultatas po neuroninio tinklo su pasukimu');
disp([PPT;PPT/40.]);
%pause;
```

Galutiniai eksperimentinio tyrimo skaičiavimai

Metodas Fonema	N	U	L	I	S	Vid.	D	U	Vid.	T	R	Y	S	Vid.	P	E	N	K	I	Vid.	Š	E	Š	I	Vid.	Vid.
Tiesinė diskriminantė	56	61	71	48	49	57	52	63	58	57	74	57	53	60	61	64	59	72	54	62	84	72	85	58	75	66
Neuroninis tinklas	61	72	73	55	53	63	63	69	66	67	79	69	58	68	70	77	63	69	60	68	92	81	91	60	81	
Hibridinis metodas	63	83	77	68	52	69	68	75	72	64	87	73	65	72	68	84	68	76	73	74	95	88	97	71	88	75
						63			65					67						68					81	
Tiesinė diskriminantė	7	22	6	20	3		16	12		7	13	16	12		7	20	9	4	19		11	16	12	13		12
Neuroninis tinklas	2	11	4	13	-1		5	6		-3	8	4	7		-2	7	5	7	13		3	7	6	11		6