VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Jolita Bernatavičienė

VIZUALIOS ŽINIŲ GAVYBOS METODOLOGIJA IR JOS TYRIMAS

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07T)

Vilnius LEIDYKLA TECHNIKA 2008

Disertacija rengta 2004–2008 metais Matematikos ir informatikos institute.

Darbo mokslinis vadovas
prof. habil. dr. Gintautas Dzemyda (Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07T).
Konsultantas
prof. habil. dr. Vydūnas Šaltenis (Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07T).

http://leidykla.vgtu.lt
VGTU leidyklos TECHNIKA 1493-M mokslo literatūros knyga

ISBN 978-9955-28-278-5

Jolita BERNATAVIČIENĖ

VIZUALIOS ŽINIŲ GAVYBOS METODOLOGIJA
IR JOS TYRIMAS

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (07T)

2008 05 09. 7,75 sp. l. Tiražas 20 egz.
Vilniaus Gedimino technikos universiteto
leidykla „Technika“, Saulėtekio al. 11, LT-10223 Vilnius
http://leidykla.vgtu.lt
Spausdino UAB „Baltijos kopija“,
Kareivių g. 13B, 09109 Vilnius
www.kopija.lt

Reziumė

Disertacijos tyrimų sritis yra žinių gavybos iš daugiamačių duomenų procesas ir tiriamų duomenų suvokimo gerinimo būdai. Duomenų suvokimas yra sudėtingas uždavinys, ypač kai duomenys nurodo sudėtingą objektą, kuris aprašytas daugeliu parametrų. Norint gauti išsamią informaciją apie analizuojamus duomenis būtina kompleksinė jų analizė, kurios etapus apibrėžia žinių gavybos procesas. Disertacijos tyrimų objektas – vizualios žinių gavybos procesas. Su šiuo objektu betarpiškai susiję dalykai: daugiamačių duomenų pirminės aibės suformavimas; klasterizavimo, vizualizavimo ir klasifikavimo algoritmai; duomenų gavybos metodais gautų rezultatų įvertinimas; naujų daugiamačių duomenų atvaizdavimas; sprendimų priėmimas ir gautų žinių apibendrinimas, atsižvelgiant į analizės rezultatus. Pagrindinis disertacijos tikslas yra sukurti ir ištirti žinių gavybos vizualiais metodais metodologiją, kuri leistų padidinti duomenų analizės efektyvumą. Darbe atliktų tyrimų rezultatai atskleidė naujas medicininių (fiziologinių) duomenų analizės galimybes.

Disertaciją sudaro penki skyriai ir literatūros sąrašas. Bendra disertacijos apimtis 116 puslapių, 44 paveikslai ir 12 lentelių.

Tyrimų rezultatai publikuoti 9 moksliniuose leidiniuose: 1 straipsnis leidinyje, įtrauktame į Mokslinės informacijos instituto pagrindinį (Thomson ISI Web of Science) sąrašą; 2 straipsniai leidiniuose, įtrauktuose į Mokslinės informacijos instituto konferencijos darbų (Thomson ISI Proceedings) duomenų bazę; 2 straipsniai Lietuvos mokslo tarybos patvirtinto sąrašo tarptautinėse duomenų bazėse referuojamuose leidiniuose; 1 straipsnis recenzuojamoje konferencijų pranešimų medžiagoje ir 3 straipsniai kituose periodiniuose bei vienkartiniuose straipsnių rinkiniuose.

Tyrimų rezultatai buvo pristatyti ir aptarti 9 nacionalinėse ir tarptautinėse konferencijose Lietuvoje ir užsienyje.

Abstract

The research area of the thesis is the process of knowledge discovery from multidimensional data and the ways of improving the perception of the data investigated. Data perception is rather a complex problem, especially when the data refer to complicated object described by many parameters. In order to obtain exhaustive information on the analysed data, their all-round analysis is indispensable the stages of which are defined by the process of knowledge discovery. The object of dissertation research is the process of visual knowledge discovery. The following subjects are directly associated with this subject: formation of a primary set of multidimensional data; algorithms for clusterization, visualization, and classification; evaluation of the results obtained by data mining methods; mapping of a new multidimensional data; decision making and generalization of the knowledge obtained referring to the analysis results. The key target of the thesis is to develop and explore the methodology of knowledge discovery by visual methods that would allow us to increase the efficiency of data analysis. The research results of the work revealed new opportunities of medical (physiological) data analysis.

The dissertation is written in Lithuanian. It consists of 5 chapters, and the list of references. There are 116 pages of the text, 44 figures, 12 tables and 156 bibliographical sources.

The main results of this dissertation were published in 9 scientific papers: 1 article in a journal abstracted in Thomson ISI Web of Science database; 2 articles in scientific publications indexed in Thomson ISI Proceedings database; 3 articles in journals indexed in international databases approved by Science Council of Lithuania; 3 articles in the proceedings of scientific conferences. The main results of the work have been presented and discussed at 5 international and 4 national conferences.

Padėka

Nuoširdžiai dėkoju moksliniam vadovui prof. habil. dr. Gintautui Dzemydai už vertingas mokslines konsultacijas, nuoseklų vadovavimą, pagalbą ir kantrybę rengiant šią disertaciją.

Esu dėkinga prof. habil. dr. Vydūnui Šalteniui už vertingus patarimus, pastabas ir diskusijas.

Ačiū disertacijos recenzentams A. L. Lipeikai ir J. Žilinskui atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų patarimų bei kritinių pastabų.

Dėkoju Matematikos ir informatikos instituto Sistemų analizės skyriaus darbuotojams ir kolegoms už naudingus patarimus ir draugišką pagalbą.

Nuoširdžiai dėkoju savo artimiesiems ir draugams už jų paramą, moralinį palaikymą, kantrybę ir supratingumą.

Dėkoju Lietuvos valstybiniam mokslo ir studijų fondui už suteiktą finansinę paramą disertacijos rengimo metu.

Taip pat dėkoju visiems, kurie tiesiogiai ar netiesiogiai prisidėjo prie šio darbo.

Jolita Bernatavičienė

Turinys

1

Įvadas

1.1. Tyrimų sritis

Šiuolaikinės technologijos užtikrina didelių duomenų srautų gavimą ir jų saugojimą. Tačiau tebelieka didelė spraga tarp duomenų surinkimo bei saugojimo ir jų suvokimo. Dažna problema yra gauti svarbių žinių, suprasti duomenis, atskirti svarbią informaciją nuo menkavertės. Čia randa vietą žinių ir duomenų gavybos metodai (angl. *knowledge discovery, data mining*). Dažnai iškyla būtinybė nustatyti duomenų struktūrą: susidariusias grupes (klasterius), žymiai išsiskiriančius objektus (taškus-atsiskyrėlius) (angl. *outliers*), atstumus arba panašumus tarp objektų ir pan. Duomenų suvokimas yra sudėtingas uždavinys, ypač kai duomenys nurodo sudėtingą objektą, reiškinį, kuris aprašytas daugeliu parametrų, kurie gali būti skaitiniai, loginiai ir kt. Tokie duomenys vadinami daugiamačiais duomenimis.

Duomenų suvokimas yra ilgo ir sudėtingo žinių gavybos proceso rezultatas. Jis apima daug etapų: suformuluojami analizės tikslai ir uždaviniai; iškeliamos pirminės hipotezės apie duomenų struktūras; formuojama duomenų imtis tyrimams; pasirenkami, kuriami nauji duomenų gavybos ir analizės metodai; analizuojami duomenų gavybos metodais gauti rezultatai; gautos žinios apibendrinamos, paneigiamos arba priimamos iškeltos hipotezės.

Šio darbo tyrimų sritis yra žinių gavybos iš daugiamačių duomenų procesas ir tiriamų duomenų pažinimo ir suvokimo gerinimo būdai.

1.2. Darbo aktualumas

Technikoje, medicinoje, ekonomikoje, ekologijoje ir daugelyje kitų sričių nuolat susiduriama su daugiamačiais duomenimis. Vystantis technologijoms, tobulėjant kompiuteriams ir programinei įrangai, kaupiamų duomenų apimtys ypač sparčiai didėja. Tačiau tebelieka didelė spraga tarp duomenų surinkimo bei saugojimo, jų suvokimo bei gautų žinių pritaikymo sprendžiant praktinius uždavinius. Daugiamačių duomenų suvokimas yra ilgo ir sudėtingo žinių gavybos proceso rezultatas. Šis procesas – tai perėjimas nuo didelės analizuojamos duomenų aibės prie specifinių duomenų, iš kurių išskiriama informacija bei suformuojamos žinios apie tiriamų duomenų struktūrą, naujus sąryšius, duomenų grupes, kas turės įtaką tolimesnių sprendimų priėmimui.

Atskiri žinių gavybos proceso etapai yra detaliai išnagrinėti literatūroje, tačiau trūksta vientisos, visus žinių gavybos etapus apimančios, metodologijos. Ji įgalins tyrėją iš turimų duomenų išgauti maksimalų informacijos kiekį, apjungti šią informaciją su eksperto patirtimi ir suformuoti žinių banką, kuris padės išspręsti tyrime iškeltus uždavinius.

Sprendžiama **problema** – vizualaus žinių gavybos proceso vientisumo užtikrinimas.

1.3. Darbo tikslas ir uždaviniai

Pagrindinis disertacijos tikslas yra sukurti ir ištirti vizualios žinių gavybos metodologiją, kuri leistų padidinti duomenų analizės efektyvumą.

Norint pasiekti šį tikslą, reikėjo išspręsti tokius uždavinius:

1) analitiškai apžvelgti duomenų gavybos ir analizės metodus: klasifikavimo, klasterizavimo ir vizualizavimo;
2) išanalizuoti žinių gavybos procesą, apžvelgti ir palyginti esamus šio proceso modelius, ištirti vizualizavimo galimybių panaudojimą žinių gavybos procese; pasiūlyti ir ištirti daugiamačių duomenų vizualizavimo proceso modelį, kuriuo pagrindžiama kuriama metodologija;
3) ištirti pasirinktus algoritmus, naudojamus vizualios duomenų gavybos procese, ir sukurti efektyvesnes jų modifikacijas; ištirti naujų (papildomai gautų) daugiamačių duomenų vizualizavimo galimybes bei pagerinti tam naudojamų metodų efektyvumą;
4) pasiūlyti ir ištirti daugiamačių duomenų išdėstymo geometrijos keitimo būdus, siekiant tikslesnės analizuojamų duomenų projekcijos plokštumoje;

5) pritaikyti sukurtą metodologiją medicininių ir fiziologinių duomenų analizei.

Tyrimų metodikos pagrindą sudaro analitinė analizė, apibendrinimas ir eksperimentinis tyrimas.

1.4. Tyrimo objektas

Norint gauti išsamią informaciją apie analizuojamus duomenis būtina kompleksinė jų analizė, kurios etapus apibrėžia žinių gavybos procesas. Disertacijos tyrimų objektas – vizualios žinių gavybos iš daugiamačių duomenų procesas. Su šiuo objektu betarpiškai susiję dalykai:

1) daugiamačių duomenų pirminės aibės suformavimas;
2) klasterizavimo, vizualizavimo ir klasifikavimo algoritmai;
3) duomenų gavybos metodais gautų rezultatų įvertinimas;
4) naujų daugiamačių duomenų atvaizdavimas;
5) sprendimų priėmimas ir gautų žinių apibendrinimas, atsižvelgiant į analizės rezultatus.

1.5. Mokslinis naujumas

1. Sukurta žinių gavybos vizualiais metodais metodologija, kuri leidžia atlikti išsamią ir informatyvią tiriamų duomenų analizę.
2. Pasiūlyti santykinių daugiamačių skalių metodo efektyvumo gerinimo būdai:
   - sukurtos bazinių vektorių parinkimo strategijos;
   - ištirtos inicializavimo problemos santykinių daugiamačių skalių algoritme, nustatytas geriausias dvimačių vektorių inicializavimo būdas;
   - pasiūlytas optimalaus bazinių vektorių skaičiaus parinkimo būdas.
3. Sukurtas atstumų tarp daugiamačių duomenų koregavimo algoritmas, kuris pagerina vizualizavimo kokybę: geriau išryškina duomenų klasterius, mažiau iškraipo daugiamačių duomenų struktūras.
4. Pasiūlytas preliminaraus sveikatos būklės fiziologinių duomenų analizės pagrindu įvertinimo būdas, besiremiantis sukurta metodologija.

1.6. Ginamieji teiginiai

1. Vizualios žinių gavybos proceso susisteminimas leidžia visapusiškai įvertinti ir pritaikyti vizualizavimo metodų ir priemonių teikiamas galimybes duomenų analizės efektyvumui didinti.
2. Santykinių daugiamačių skalių efektyvumą galima pagerinti tinkamai parenkant bazinių vektorių skaičių, bazinių vektorių išrinkimo strategiją bei dvimačių vektorių inicializavimo būdą.
3. Daugiamačių vektorių vizualizavimo kokybę galima pagerinti taikant daugiamačių duomenų atstumų koregavimo transformaciją.
4. Sukurtą vizualios žinių gavybos metodologiją galima taikyti preliminariam sveikatos būklės vertinimui.

1.7. Praktinė vertė

Tyrimų rezultatai atskleidė naujas medicininių ir fiziologinių duomenų analizės galimybes. Tai leido sporto medicinos specialistams įvertinti nesportuojančiųjų sveikatos būklę ir jų galimybę sportuoti.

Tyrimai atlikti pagal:

- Lietuvos valstybinio mokslo ir studijų fondo prioritetinių Lietuvos mokslinių tyrimų ir eksperimentinės plėtros programą „Informacinės technologijos žmogaus sveikatai – klinikinių sprendimų palaikymas (e-sveikata), IT sveikata“; Registracijos Nr.: C-03013; Vykdymo laikas: 2003 m. 09 mėn. – 2006 m. 10 mėn.
- Lietuvos valstybinio mokslo ir studijų fondo aukštųjų technologijų plėtros programos projektą „Informacinės klinikinių sprendimų palaikymo ir gyventojų sveikatinimo priemonės e. Sveikatos sistemai (Info Sveikata)“; Registracijos Nr.: B-07019, Vykdymo laikas: nuo 2007 m. 09 mėn.

1.8. Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 9 moksliniuose leidiniuose: 1 straipsnis leidinyje, įtrauktame į Mokslinės informacijos instituto pagrindinį (Thomson ISI Web of Science) duomenų bazę; 2 straipsniai leidiniuose, įtrauktuose į Mokslinės informacijos instituto konferencijos darbų (Thomson ISI Proceedings) sąrašą; 2 straipsniai Lietuvos mokslo tarybos patvirtinto sąrašo tarptautinėse duomenų bazėse referuojamuose leidiniuose; 1 straipsnis recenzuojamoje konferencijų

pranešimų medžiagoje ir 3 straipsniai kituose periodiniuose bei vienkartiniuose straipsnių rinkiniuose.

Tyrimų rezultatai buvo pristatyti ir aptarti šiose nacionalinėse ir tarptautinėse konferencijose Lietuvoje ir užsienyje:

1. KTU mokslinė-teminė konferencija „Informacinės technologijos 2005“, Kaunas, Kauno tecnologijos universitetas, 2005 sausio 26–28.
2. Advanced Course on Knowledge Discovery (ACAI) complemented with 1st SEKT Summer School on Semantic – Web, Slovenia, Ljubljana, June 27 – July 5, 2005.
3. 9-oji tarptautinė konferencija „Biomedicininė inžinerija“, Kauno tecnologijos universitetas, 2005 spalio 27–28.
4. Optimal Process Design, International Networking for Young Scientists, Vilnius, Lithuania, 15–16 February 2006.
5. Lietuvos jaunųjų mokslininkų konferencija „Operacijų tyrimas ir taikymai“ (LOTD – 2006), Vilnius, 2006 gegužės 26.
6. The Eighth International Conference on Artificial Intelligence and Soft Computing (ICAISC 2006), Zakopane, Poland, 25–29 June 2006.
7. Optimization Challenges in Engineering: Methods, Software and Applications, EURO Summer Institute 2006, Lutherstadt Wittenberg, Germany, August 18 – September 2, 2006.
8. 11th Conference on Artificial Intelligence in Medicine (AIME 07), Doctoral Consortium, Amsterdam, Netherlands, 07–11 July 2007.
9. Informatikos doktorantų vasaros mokykla „Modernios duomenų gavybos ir analizės technologijos“, Druskininkai, 2007 rugsėjo 9–15.

1.9. Disertacijos struktūra

Disertaciją sudaro penki skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Vizualios analizės vieta duomenų gavyboje, Vizualios duomenų gavybos galimybių didinimas, Vizuali žinių gavyba analizuojant fiziologinius duomenis, Bendrosios išvados ir rekomendacijos. Disertacijos apimtis 116 puslapių, 44 paveikslai ir 12 lentelių.

2

Vizualios analizės vieta žinių gavyboje

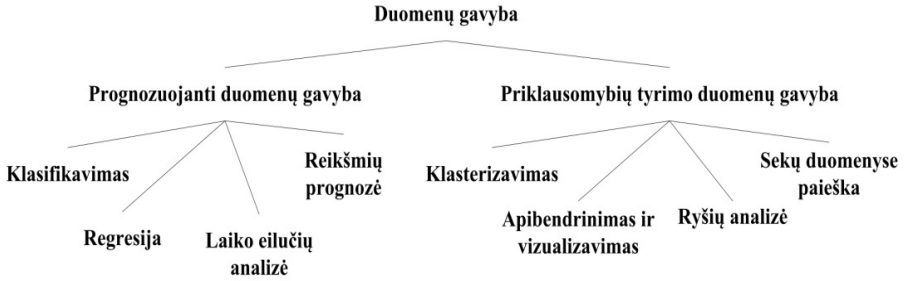
2.1. Duomenų gavybos sprendžiami uždaviniai ir jų sprendimui naudojami metodai

Duomenų gavyba ir analizė skirtingų sprendžiamų uždavinių vykdymui apima labai daug įvairių algoritmų. Visi algoritmai bando priderinti kuriamą modelį prie duomenų. Jie tiria duomenis ir apibrėžia tokį modelį, kurio charakteristikos yra labai artimos tiriamų duomenų charakteristikoms. 2.1 paveiksle pateikta duomenų gavybos sprendžiamų uždavinių schema.

Duomenų gavyba apima dvi plačias tyrimų sritis:

1. Priklausomybių (angl. *descriptive*) tyrimo duomenų gavyba. Jai priskiriama grupė uždavinių, kuriuose nustatomos struktūros (angl. *patterns*) duomenų imtyje neturint išankstinių žinių apie jau egzistuojančias tiriamų duomenų struktūras. Šiai uždavinių grupei yra priskiriamas klasterizavimas (grupavimas), ryšių analizė (angl. *association rules*), sekų duomenyse paieška (angl. *sequence discovery*), apibendrinimas (angl. *summarization*) ir vizualizavimas.

2. Prognozuojanti (angl. *predictive*) duomenų gavyba. Ji apima uždavinių grupę, kuriuose panaudojant visas žinias apie turimus duomenis atliekama prognozė naujiems duomenims. Pavyzdžiui, naujo tiriamojo priskyrimas tam tikrai susirgimų klasei, sekant akcijų kilimo ir kritimo tendencijas laike, daro prognozę apie akcijų kilimą ar kritimą artimiausiu laiku ir pan. Šiai uždavinių grupei yra priskiriamas klasifikavimas, regresija, laiko eilučių analizė, reikšmių prognozė.



2.1 pav. *Duomenų gavybos uždaviniai*

Klasterizavimas – sąvoka, naudojama tokiems metodams apibūdinti, kurie grupuoja panašius duomenų įrašus. Kiekvienas metodas turi savo panašumo matą. Duomenų įrašas gali apimti, pavyzdžiui, kiekvieno paciento sveikatos būklės aprašymą. Šiuo atveju grupavimo metodas grupuotų visus panašios sveikatos būklės pacientus, kartu tuo pačiu maksimizuodamas skirtumus tarp skirtingų pacientų grupių, sudarytų remiantis šiuo metodu. Paprastai grupavimas atliekamas remiantis klasterizavimo metodais. Duomenų grupės gali būti persidengiančios, hierarchinės ir nepersidengiančios (Fayyad *et al.* 1996a). Kiekvienos grupės narys panašus į savo grupės narius ir nepanašus į kitų grupių narius.

Išsiskiriančių duomenų (taškų atsiskyrėlių) analizė (angl. *outlier analysis*) yra tam tikra klasterizavimo forma, kuri sutelkia dėmesį į duomenis, nepriklausančius nė vienam iš aiškiai apibrėžtų klasterių (Han and Kamber 2006). Kartais tokie duomenys būna duomenų apdorojimo klaidos, o kitais atvejais išsiskiriantys duomenys suteikia naujos informacijos apie tiriamų duomenų ypatybes.

Klasterizavimo metodai gali būti suskirstyti į kelis pagrindinius tipus:

1. hierarchiniai metodai (angl. *hierarchical*) (Dash and Liu 2001), tai dendogramos, minimalaus jungimo medis (angl *minimal spanning tree*) ir kt. (Dunham 2003). Hierarchiniai metodai skaidomi į:

- sujungimo (angl. *agglomerative*) metodus; sujungimo metodų pradžioje visi objektai sudaro atskirus klasterius, vėliau objektai jungiami į grupes;
- išskaidymo (angl. *divisive*) (Jain and Dubes 1988) metodus; išskaidymo metodų pradžioje visi objektai sudaro vieną klasterį, iteraciniame procese jis skaidomas į mažesnius.

2. padalijimo metodai (angl. *partitioning*) stengiasi duomenų aibę padalinti į kelis nesusikertančius klasterius. Visų galimų klasterių perrinkimas reikalauja labai daug skaičiavimo sąnaudų, o kartais yra neįmanomas, todėl vietoj hierarchinių metodų naudojamos iteracinio optimizavimo euristikos. Paprastai šie euristiniai metodai skirtingais būdais iteratyviai perskisto taškus, paskirstant juos į *k* klasterių. Padalijimo metodai skirstomi į:
   - *k*-vidurkių klasterizavimo metodai ir jų modifikacijos: (Vesanto 2001), (Dunham 2003);
   - *k*-vidurinių taškų klasterizavimo metodai ir jų modifikacijos: PAM (angl. *partitioning around medoids*), CLARA (Kaufman and Rousseeuw 2005);
   - tikimybiniai klasterizavimo algoritmai: EM algoritmas (angl. *expectation maximization*) (Mitchell 1997).
3. duomenų tankiu pagrįsti klasterizavimo metodai – tai metodai, grupuojantys kaimyninius duomenų objektus į klasterius ne pagal atstumų matą, bet pagal objektų tankį. Tankis apibrėžiamas minimaliu taškų, tarp kurių yra tam tikras atstumas, skaičiumi. Čia įvedama taško kaimynystės sąvoka. Taškai, kurie yra vienas nuo kito nutolę tam tikru nustatytu atstumu, vadinami kaimynais. Klasteriai sudaromi arba pagal kaimynystės tankį, arba pagal tam tikrą tankio funkciją. Tankiu grindžiami klasterizavimo algoritmai sugeba atrasti sudėtingos struktūros klasterius, o taškai atsiskyrėliai nėra įtraukiami į klasterius: DBSCAN algoritmas (angl. *density based spatial clustering of applications with noise*) (Ester *et al.* 1996), OPTICS (angl. *ordering points to identify the clustering structure*) (Ankerst *et al.* 1999);
4. klasterizavimo metodai su ribojimais (Han and Kamber 2006). Klasterizavimo ribojimai - tai ribojimai atskiriems objektams (pvz. neseniai užsisakęs prekes užsakovas), ribojimai algoritmo parametrams (pvz. klasterių skaičius), ribojimai atskiram klasteriui ir pan.: COD algoritmas (angl. *clustering with obstructed distance*) (Tung *et al.* 2001), tikimybiniai klasterizavimo algoritmai su ribojimais aprašyti (Lange *et al.* 2005);
5. tinkleliu pagrįsti metodai (angl. *grid-based*) padalija erdvę į baigtinį skaičių langelių, t. y. suformuoja tinklelį; klasterizavimo operacijos atliekamos tinklo struktūroje: STING (angl. *statistical information*

grid-based method) metodas (Wang *et al.* 1997); WaveCluster (Sheikholeslami *et al.* 1998); CLIQUE (angl. *clustering in quest*) (Agrawal *et al.* 1998), FC algoritmas (angl. *fractal clustering*) (Barbara and Chen 2000);

6. neuroniniais tinklais grindžiami klasterizavimo algoritmai: SOM (Kohonen 2001), (Flexer 2001) ir įvairios modifikacijos.
7. genetiniais algoritmais grindžiami klasterizavimo algoritmai: GGA algoritmas (angl. *genetically guided algorithm*) (Hall *et al.* 1999), genetiniai algoritmai pritaikyti *k*-means tikslo funkcijai (Sarafis *et al.* 2002);
8. didelės dimensijos duomenų klasterizavimo algoritmai: CLARA metodas (angl. *clustering carge applications*) (Kaufman and Rousseeuw 1990); CLARANS metodas (angl. *clustering large applications based upon randomized search*) (Ng and Han 1994), (Ester *et al.* 1995); BIRCH (angl. *balanced iterative reducing and clustering using hierarchies*) (Zhang *et al.* 1996); DBCLASD (Xu *et al.* 1998) ir kt.

Ryšių (asociacijų) analizė, pristatyta 1993 metais straipsnyje (Agrawal *et al.* 1993), apima uždavinius, kuriuose reikia nustatyti ryšius, koreliacijas tarp duomenų, jų parametrų.

Gerai žinomas ryšių analizės pavyzdys yra pardavimo krepšelio analizė. Šiuo atveju, duomenų įrašai yra kliento pirktos prekės vienu metu. Pardavimo krepšelio analizė suranda prekių (kurias pirko skirtingi klientai) kombinacijas ir pagal asociacijas (arba ryšį) galima sudaryti vaizdą, kokie produktai perkami kartu (Dunham 2003).

Tačiau ryšių analizės vartotojas turi būti įspėtas, kad tokie ryšiai nėra funkcinės priklausomybės, gautos iš esamų duomenų.

Egzistuoja labai daug ryšių taisyklių sudarymo algoritmų. Juos galima būtų sugrupuoti į kelias grupes (Kotsiantis and Kanellopoulos 2006):

1. *nuoseklūs ryšių taisyklių sudarymo algoritmai*: AIS algoritmas (Agrawal *et al.* 1993) generuoja nuoseklias ryšių taisykles kiekvienam objektui; Apriori algoritmas (Agrawal and Srikant 1994) naudodamas apkarpymo (angl. *pruning*) techniką išrenka tik didžiausius kandidatų taisyklių poaibius ir kitos šio algoritmo modifikacijos Apriori-Gen, Aprior-TID, ryšių taisyklių sudarymo algoritmai naudojantys pavyzdžius (Toivonen 1996), (Parthasarathy 2002);
2. *lygiagretūs ryšių taisyklių sudarymo algoritmai*: FDM lygiagretus Apriori algoritmas dalina analizuojamą aibę ir taisykles kuria savo lokaliuose poaibiuose (Cheung *et al.* 1996a), (Parthasarathy *et al.* 2001) straipsnyje pateikta lygiagrečių algoritmų, naudojančių bendros atminties architektūrą apžvalga.

3. *ryšių taisyklių sudarymo algoritmai grįsti ribojimais:* RARM (angl. *rapid association rule mining*) (Das *et al.* 2001) algoritmas naudoja medžio struktūrą tiriamai duomenų bazei pateikti ir taip išvengia kandidatų generavimo proceso; Modifikuotas Apriori algoritmas su ribojimais aprašytas (Do *et al.* 2003).

Sekų duomenyse paieška apima duomenų gavybos metodikas, nustatančias tam tikrų įvykių dažnumus, kurie yra pritaikomi duomenų rinkinių analizei. Šiuo metu labai aktualu rasti tarp didelių duomenų masyvų mums svarbią informaciją, kurią būtų galima panaudoti ateityje. Vienas iš svarbiausių jos tikslų yra dažnų pasikartojamumų radimas. Dar visai neseniai bet kuriai informacijos apdorojimo sistemai pakakdavo spręsti įvairius paieškos (surasti, kur ir kiek kartų pasikartoja nurodytas įrašas) arba statistinius uždavinius: koks yra vidutinis avaringumas (gimstamumas, nusikalstamumas) respublikoje, duotame rajone, per kažkokį laikotarpį ir t.t. Sekų duomenyse paieška nagrinėja šiuos duomenis žymiai sudėtingiau ir pateikia gana detalius šios analizės rezultatus. Sekų duomenyse paieška leidžia atsakyti į klausimus, kokie žodžiai dažniausiai pasikartoja tekste, kokių kriterijų visuma turi įtaką avaringumui (gimstamumui, nusikalstamumui) respublikoje vertinant dažniausiai atsitinkančius įvykius, duotame rajone ar per kažkokį laiko tarpą. Ne ką mažesnę svarbą sekų duomenyse paieška turi medicinoje, nustatant žmogaus geno kodo pasikartojančias sekas, pagal kuriuos nustatoma, kad žmonės serga viena ar kita liga. Taip pat sekų duomenyse paieška yra populiari bankininkystėje, nustatant kokius kriterijus atitinka žmonės, kurie negali grąžinti paskolų. Prekybininkai irgi savo veikloje naudoja (arba gali naudoti) sekų duomenyse paieškos metodus, nustatant populiariausių prekių „krepšelius“, kurie buvo įsigyjami pirkėjų vieno apsipirkimo metu. Jeigu duomenys yra sudaryti iš tam tikrų aibės elementų, tai labai svarbu yra nustatyti dažnus tų elementų didžiausius poaibius duotoje duomenų bazėje.

Sekų duomenyse paieškos algoritmai pirmiausiai buvo nagrinėjami (Agrawal *et al.* 1993) ir (Agrawal and Srikant 1994). Šiuose darbuose buvo išnagrinėti pagrindiniai klasikinis Appriori ir jo patobulinta versija GSP algoritmai. Šių algoritmų pagrindinė idėja yra ta, kad dažnos sekos ieškomas eliminuojant nedažnus posekius iš galimos dažnos sekos. Vėliau buvo sukurtas žymiai efektyvesnis SPADE algoritmas (Zaki 2000), (Zaki 2001), paremtas lygiagrečiais skaičiavimais. Ieškant dažnų poaibių, visa duomenų bazė yra suskaidoma į tarpusavyje nesusietas ir nepriklausomas dalines gardeles. Šiose gardelėse lygiagrečiai galima vykdyti dažnų poaibių paiešką iš apačios į viršų, iš viršaus į apačią bei hibridiniu būdu. Populiariausi paieškos algoritmai:

1. *Eclat.* Algoritmas naudoja priesagų ekvivalentumo sąryšį ir taiko paieškos iš apačios į viršų metodą (priesaga – sekos dalis, esanti sekos arba posekio pradžioje, o ekvivalentumo sąryšis suskaido seką į

nesusikertančius posekius). Tokiu būdu surandami visi dažni rinkiniai (Zaki *et al.* 2005), (Chang *et al.* 2002), (Han *et al.* 2000).

2. *MaxEclat.* Algoritmas naudoja priesagų ekvivalentumo sąryšį ir taiko hibridinį paieškos metodą. Jis suranda ilgus maksimaliai dažnus rinkinius ir kai kuriuos nemaksimalius dažnus rinkinius (Lin and Kedem 2002), (Zaki *et al.* 1997), (Roddick and Spiliopoulou 2002).
3. *Clique.* Algoritmas naudoja maksimalų uždarų grupių pseudoekvivalentumo sąryšį (pseudoekvivalentumo sąryšis suskaido sekas į posekius, turinčius tą pačią priesagą) ir taiko paieškos iš apačios į viršų metodą. Randa visus dažnus rinkinius (Bayardo 1997), (Chattratichat *et al.* 1997), (Luo *et al.* 2006).
4. *MaxClique*. Šis algoritmas naudoja maksimalų uždarų grupių pseudoekvivalentumo sąryšį ir hibridinį paieškos metodą. Jis suranda ilgus maksimaliai dažnus rinkinius ir kai kuriuos nemaksimalius dažnus rinkinius (Mannila *et al.* 1997), (Silverstein *et al.* 2000), (Hipp *et al.* 2000).
5. *TopDown*. Naudoja maksimalų uždarų grupių pseudoekvivalentumo ryšį ir taiko paieškos iš viršaus į apačią metodą. Suranda tik maksimaliai dažnus rinkinius (Orlando *et al.* 2003), (Cheung *et al.* 1996b), (Savasere *et al.* 1995).
6. *AprClique*. Algoritmas naudoja maksimalų uždarų grupių pseudoekvivalentumo ryšį. Jį galima padalinti į du etapus:
   - Visi įmanomi maksimalaus elemento poaibiai kiekvienoje dalinėje gardelėje yra generuojami ir saugomi specialiuose maišos medžiuose (angl. *hash trees*), išvengiant dublikatų. Visiems *k*-rinkiniams yra skiriamas atskiras medis. Vidinis d gylio medžio mazgas turi lentelę, kurios elementai rodo į (d+1)-ąjį lygį. Visi rinkiniai yra saugomi medžio lapuose. Įterpimo į medį procedūra startuoja nuo medžio šaknies ir įterpia visus kandidatus į lapus.
   - Dažnumo skaičiavimo žingsnis yra panašus kaip Apriori algoritme. Kiekvienai transakcijai (transakcija – užbaigtas veiksmas su duomenų aibės įrašais, pvz. gautas vieno pirkėja pirkinių sąrašas) formuojami visi įmanomi poaibiai. Po to ieškomas tas poaibis medyje ir, radus, atnaujinamas skaitiklis (Fayyad *et al.* 1996b), (Klemettinen *et al.* 1994).

Apibendrinimas ir vizualizavimas. Prieš kuriant prognozės modelį, reikia suvokti tiriamus duomenis: įvertinti statistikas, tokias kaip vidurkis, dispersija, standartinis nuokrypis, histogramos ir pan., nustatyti duomenų pasiskirstymą,

iškelti hipotezes, kurias duomenų gavybos proceso metu teks patvirtinti ar atmesti. Apibendrinimo procesas duomenų gavyboje dar vadinamas *tiriamąja analize* (angl. *exploratory analysis*) (Larose 2004).

Dažnai daugiamačiams duomenis apibendrinti sudaromos priklausomybių lentelės. Apibendrinimas suskirsto tiriamus duomenis į poaibius, ir pateikia elementarų tų poaibių aprašymą.

Duomenų vizualizavimas padeda įžvelgti galimus ryšius tarp duomenų, suvokti duomenų struktūrą: naudojant histogramas įvertinti duomenų pasiskirstymą, naudojant taškinius grafikus galima įvertinti kintamųjų regresiją ir pan. Vizualizavimo metodai yra labai įvairūs (Hansen and Johnson 2004) ir suteikia daug pirminės informacijos apie turimus duomenis. Pavyzdžiui, naudojant daugiamačių skalių metodą (Borg and Groenen 1997) suprojektavus daugiamačius duomenis į plokštumą, kelti hipotezes apie tiriamų duomenų struktūrą, nustatyti taškus atsiskyrėlius ir pan. Detali vizualizavimo metodų apžvalga pateikta (Dzemyda *et al.* 2008), apie tai plačiau pateikiama ir 2.4. skyriuje.

Klasifikavimas. Klasifikavimo tikslas – identifikuoti parametrus, kurie nusakytų grupę (klasę), kuriai priklauso objektas. Ši sąvoka gali būti naudojama tiek esamų duomenų suvokimui, tiek naujų objektų charakteristikų prognozavimui. Klasifikavimo taikymo pavyzdys būtų banko klientų duomenų klasifikavimas, norint nuspręsti ar suteikti klientui paskolą ar ne, o jei nuspręsta suteikti – tai kokia bus šio kredito rizika. Klasifikavimo algoritmai reikalauja, kad klasės arba grupės būtų apibrėžiamos remiantis pasirinktų duomenų parametrų reikšmėmis. Dažniausiai klasifikavimo algoritmai apibrėžia klases orientuojantis į jau žinomų duomenų charakteristikas.

Duomenų struktūrų atpažinimas (angl. *pattern recognition*) yra tam tikra klasifikavimo rūšis, kur duomenys klasifikuojami į tam tikras klases remiantis panašumais į jau apibrėžtų klasių elementus.

Klasifikavimas dažnai atliekamas po grupavimo, kurio metu nustatomos klasės. Visus klasikavimo algoritmus galima suskirstyti į šias grupes:

1. Statistiniais metodais grindžiami klasifikavimo algoritmai. Šių klasifikatorių paskirtis yra nustatyti, kurios klasės tikėtinumas yra didžiausias tiriamiems duomenims pagal duotą turimą informaciją apie duomenis, prielaidas apie modelio struktūrą bei tikimybių pasiskirstymą. Galima paminėti tiesinius, netiesinius, logistinius regresinius klasifikavimo algoritmus (Dunham 2003), paprastasis Bayeso (angl. *Naïve Bayes*) klasifikatorius (Ramoni and Sebastiani 2003).
2. Atstumų skaičiavimais grindžiami klasifikavimo algoritmai kiekvieną objektą priskiria tai klasei, į kurios klasės objektus tiriamas objektas yra panašiausias. Dažniausiai tas panašumo mata yra atstumas, tačiau yra ir kitaip apibrėžiamų panašumo matų. Tokiems algoritmams yra priskiriami

artimiausių kaimynų klasifikatoriai (angl. *k nearest neighbors classifiers*) ir jų modifikacijos (Dunham 2003).

3. Sprendimų medžiais grindžiami klasifikavimo algoritmai; sukuriamas medis modeliuojantis klasifikavimo procesą, šie algoritmai sudalina tiriamą sritį į nedideles sritis, priklausančias atskiroms klasėms: ID3 algoritmas, C4.5 ir C5.0 algoritmai, CART algoritmas, SPRINT algoritmas ir pan. (Dunham 2003), (Fielding 2006).
4. Atraminių vektorių klasifikatoriai (angl. *support vector machines, SVM*). Pirmasis juos pristatė Vapnik, detali šių algoritmų apžvalga pateikta knygoje (Vapnik 1998).
5. Neuroniniais tinklais grindžiami klasifikavimo algoritmai: Bayeso neuroninių tinklų klasifikatorius (Heckerman 1996), klasifikatorius apmokomas pagal klaidos sklidimo atgal (angl. *backpropogation*) taisyklę (Hanson and Burr 1988), ir kt. Išsami neuroniniais tinklais grindžiamų klasifikavimo algoritmų taikymo apžvalga pramonėje, versle, moksle pateikta (Widrow *et al.* 1994), (Raudys 2001) ir kiti.
6. Genetiniais algoritmais grindžiami klasifikavimo algoritmai, tokio tipo algoritmai pateikti knygose (Michalewicz 1992), (Mitchell 1996).
7. Ryšių taisyklėmis grindžiami klasifikavimo algoritmai: CBA algoritmas (Liu *et al.* 1998), CMAR algoritmas (Li *et al.* 2001), CPAR algoritmas (Yin and Han 2003) ir kt.
8. Skirtingų klasifikavimo algoritmų junginiai: tokių algoritmų trumpa apžvalga pateikta (Han and Kamber 2006).

Regresija. Regresija prognozuojant naujų duomenų reikšmes naudojasi žinomais, jau turimais duomenimis. Ji naudoja standartinius statistinius metodus, pvz. tiesinę regresiją (Press *et al.* 1992). Deja, daugumos realių uždavinių duomenys nėra tiesiškai priklausomi nuo ankstesnių duomenų (pvz. rinkos kainos, pardavimų apimtys, perkamoji galia). Tai labai sunkiai prognozuojami dydžiai, nes priklauso nuo daugelio sudėtinių parametrų, todėl tokių duomenų prognozei naudojami daug sudėtingesni metodai: netiesinė regresija (Friedman 1991), logistinė regresija (Pearl 1988), regresijos medžiai (Breiman *et al.* 1984) ir pan.

Laiko eilučių analizė. Naudojant laiko eilučių analizės metodus yra tiriamas parametro reikšmių kitimas laike. Parametro reikšmės yra gaunamos fiksuotais laiko momentais (kas dieną, kas savaitę, kas valandą ir pan.). Prognozuojant laiko eilutės būsimas reikšmes remiamasi anksčiau gautais duomenimis, panašiai kaip ir regresijoje. Sukurtas laiko eilutės modelis atspindi laiko ypatumus,

sezoniškumą, tokius kalendoriaus ypatumus kaip atostogas ar švenčių dienas ir pan.

Laiko eilučių analizei naudojamos trys pagrindinės funkcijos: pirmoji funkcija, norint nustatyti tų eilučių panašumus ar skirtumus, matuoja atstumus tarp atitinkamų taškų skirtingose laiko eilutėse; antroji funkcija, norėdama apibrėžti eilutės elgesį, bando nustatyti kreivės struktūrą; trečioji – naudoja ankstesnį eilutės grafiką naujų reikšmių prognozei.

Statistiniai metodai skirti laiko eilučių analizei yra pateikti knygose (Chatfield 2003), (Shumway and Stoffer 2005), panašumų paieškos laiko eilutėse algoritmai pristatyti (Rafiei and Mendelzon 1997), (Shasha and Zhu 2004).

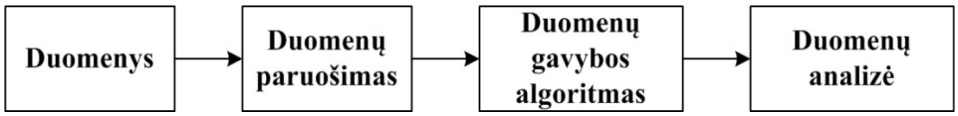
Reikšmių prognozavimas. Reikšmių prognozė gali būti apibrėžiama kaip tam tikra klasifikavimo atmaina. Skirtumas tas, kad prognozė nusako būsimą duomens reikšmę, o ne klasifikuoja esamą.

Dažnai duomenų gavybos uždaviniui išspręsti taikomi keli metodai iš eilės ar net sudėtingi jų deriniai (Han and Kamber 2006). Uždavinių bei metodų įvairovę papildo grupė duomenų gavybos algoritmų. Nė vienas jų nėra universalus ar nepriekaištingas. Parenkant algoritmus atsižvelgiama į jų operacinį ir loginį sudėtingumą), sugaištamą analizei kompiuterio laiką bei atmintį, analizės patikimumą.

2.2. Žinių gavybos procesas

Duomenų gavybos ir analizės procesą sudaro trys etapai:

1. *Duomenų paruošimas*: remiantis eksperto turimomis žiniomis ir nurodymais, duomenys surenkami, išvalomi, paruošiami priminei analizei.
2. *Duomenų gavybos algoritmas*: duomenų gavybos algoritmai naudojami svarbios informacijos duomenyse išskyrimui.
3. *Duomenų analizės fazė*: Gautų rezultatų analizė ir interpretavimas.



2.2 pav. *Duomenų gavybos ir analizės proceso schema*

Sėkmingas duomenų gavybos taikymas apima tokias turimų duomenų transformacijas, kurios duomenis padaro labiau kompaktiškus ir suprantamesnius, o ryšiai tarp duomenų yra aiškiai ir suprantamai apibrėžiami. Pilna duomenų gavybos schema pateikta 2.2 paveiksle.

2.1 lentelė. *Keturių žinių gavybos proceso modelių palyginimas*

4 žingsnių (Simoudis 1996) proceso modelis	**9 žingsnių (Fayyad *et al.* 1996a) proceso modelis**	**5 žingsnių (Cabena *et al.* 1998) proceso modelis**	**6 žingsnių (Cios *et al.* 2000) proceso modelis**
	1. Apibrėžiama taikymų sritis, suformuluojami tyrimo uždaviniai	1. Uždavinio tikslų suformulavimas	1. Analizuojamos problemos suformulavimas
1. Duomenų surinkimas	2. Sukuriama planuota duomenų aibė		2. Duomenų suvokimas
2. Duomenų transformacijos	3. Duomenų išgryninimas ir apdorojimas	2. Duomenų paruošimas ir apdorojimas	3. Duomenų paruošimas
	4. Duomenų dimensijos mažinimas ir jų projekcija		
	5. Suformuluojami reikalavimai duomenų gavybos metodui, naudojamam tyrime		
	6. Tiriamoji analizė, modelio parinkimas ir hipotezių iškėlimas		
3. Duomenų gavyba	7. Duomenų gavyba	3. Duomenų gavyba	4. Duomenų gavyba
4. Gautų rezultatų interpretavimas	8. Gautų rezultatų interpretavimas	4. Rezultatų analizė	5. Rezultatų įvertinimas ir validavimas
	9. Gautų žinių apibendrinimas	5. Turimų ir gautų žinių sulyginimas	6. Gautų žinių prijungimas ir panaudojimas

Žinių gavybos procesas naudojant duomenų gavybos ir analizės metodus yra iteracinis ir interaktyvus. Literatūroje yra siūlomi įvairūs žinių radimo proceso modeliai, kurių palyginimas yra pateiktas 2.1 lentelėje (Pal and Jain 2005).

Palyginus šiuos keturis modelius yra išskirti šeši pagrindiniai žinių radimo proceso žingsniai, su grįžtamaisiais ryšiais (2.3 paveikslas). Uždaviniai, sprendžiami kiekviename proceso žingsnyje yra aprašyti literatūroje (Larose 2004), (Sumathi and Sivanandam 2006):

1 žingsnis. Suformuluojami duomenų aibės analizės tikslai ir uždaviniai:

- ✓ kokias priklausomybes tikimasi duomenyse rasti, kokie dėsningumai domintų tyrėjus, kokias prognozes norima iš turimų duomenų daryti;
- ✓ Tikslai ir uždaviniai aprašomi duomenų gavybos uždavinių formalizavimo kalba;
- ✓ Pasirenkama strategija iškeltų uždavinių sprendimui.

2 žingsnis. Duomenų suvokimas:

- ✓ Surenkami duomenys analizei;
- ✓ Susipažinimui su surinktais duomenimis ir pirminių žinių juose nustatymui naudojama tiriamoji duomenų analizė;
- ✓ Įvertinama surinktų duomenų kokybė;
- ✓ Jei pastebimi dėsningumai tam tikruose duomenų poaibiuose, jie išskiriami atskirai duomenų analizei.

3 žingsnis. Duomenų paruošimas:

- ✓ Iš visos turimos duomenų aibės suformuojama galutinė duomenų imtis kuri bus naudojama tolimesnėse tyrimo stadijose;
- ✓ Pasirenkame tik tuos duomenų atvejus ir kintamuosius, kurie būdingi, svarbūs atliekamai duomenų analizei;
- ✓ Atliekame kintamųjų transformacijas, jei to reikia: pašalinamas iš duomenų triukšmas, mažinama duomenų dimensija, atliekamas duomenų normavimas, pašalinami taškai „atsiskyrėliai“, pasirenkama strategija, ką daryti su nepilnais duomenimis;
- ✓ Duomenys paruošiami pasirinktiems duomenų gavybos algoritmams.

4 žingsnis. Duomenų gavyba:

- ✓ Pasirenkami ir taikomi tinkamiausi duomenų gavybos metodai;
- ✓ Parenkami optimalūs metodų parametrai;
- ✓ Primename, kad tuo pačiu gali būti naudojami keli duomenų gavybos metodai;

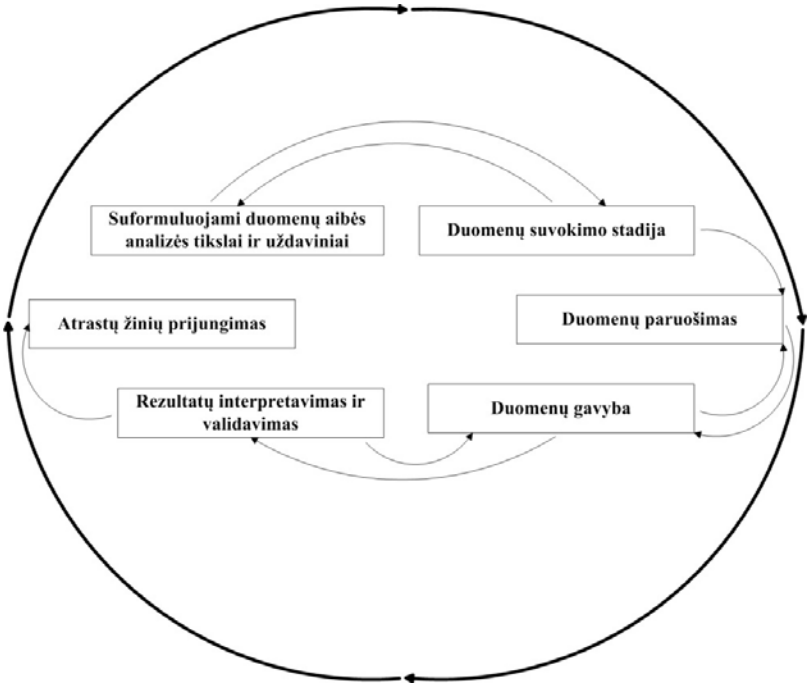
- ✓ Jei yra būtina, grįžtama į duomenų paruošimo stadiją, duomenų imtis koreguojama, papildant ją naujais kintamaisiais ir naujais įrašais.

5 žingsnis. Rezultatų įvertinimas ir validavimas:

- ✓ Įvertinama gautų rezultatų kokybė iškeltų uždavinių sprendimui;
- ✓ Įvertiname, ar gauti rezultatai tenkina iškeltus uždavinius pirmame žingsnyje;
- ✓ Nustatoma, ar visi tyrimo aspektai pakankamai įvertinti;
- ✓ Priimamas sprendimas, atsižvelgiant į naujas gautas žinias, besiremiančias duomenų analizės rezultatais.

6 žingsnis. Gautų žinių prijungimas ir panaudojimas:

- ✓ Gauti rezultatai pateikiami ekspertui, kuris gautas žinias lygina su anksčiau turėtomis jas interpretuoja, nusprendžia ar gautus rezultatus atmesti ar gautais rezultatais papildyti žinių banką;



2.3 pav. *Žinių gavybos proceso, naudojant duomenų gavybos ir analizės metodus, schema*

Visi žingsniai tarpusavyje yra susiję. Jie sudaro iteracinį ir prisitaikantį ciklą, sprendžiant vienus suformuluotus uždavinius kyla kiti, ir procesas tęsiamas, kol gaunamos tyrimus tenkinančios išvados. Be to kiekviename žingsnyje galima grįžti atgal ir pagerinti rezultatus, gautus ankstesniuose žingsniuose.

2.3. Vizualizavimas žinių gavybos procese

Visuose žinių gavybos, naudojant duomenų gavybos ir analizės metodus, proceso etapuose (žingsniuose) duomenų vizualizavimas yra labai svarbus. Tyrėjui dažnai yra sunku interpretuoti sprendimą, rezultatą, gautą automatizuotais duomenų gavybos metodais, o gautų rezultatų vizualus pateikimas (vaizdas, grafikas ar pan.) susistemina, apibendrina gautus rezultatus. Tuomet yra pasitikima žmogaus regos galimybėmis atrasti, pastebėti rezultatų dėsningumus ar ypatumus, gautus kiekviename žinių gavybos proceso žingsnyje: preliminarus duomenų vaizdas, srities specifinis vizualizavimas, gautų rezultatų pateikimas (Sumathi and Sivanandam 2006).

Pirmiausia apibrėžkime, kas yra informacijos vizualizavimas. (Card *et al.* 1999) įvardina *informacijos vizualizavimo sąvoką* kaip interaktyvų vaizdinį informacijos pateikimą, kuris padidina tiriamų duomenų pažinimo galimybes.

Negalima tapatinti informacijos vizualizavimo su moksliniu vizualizavimu. Moksliniame vizualizavime (Nielson *et al.* 1997) tiriami objektai turi fizinę prasmę, pavyzdžiui, vizualizuojama molekulės struktūra, medicininiai vaizdai, geodeziniai brėžiniai ir t.t.

Literatūroje yra nurodyta begalės duomenų grafikų, informacijos ir mokslinio vizualizavimo metodų. Tokių metodų apžvalgas galite rasti (Card *et al.* 1999), (Cleveland 1994), (Fayyad *et al.* 2002), (Nielson *et al.* 1997), (Tufte 1990), apie daugiamačių duomenų vizualizavimo metodus galima pasiskaityti detaliau (Dzemyda *et al.* 2008), (Grinstein and Ward 2002), (Hoffman and Grinstein 2002), (Keim 2002), (Wong and Bergeron 1997).

Šiame darbe mus labiausiai domina tiriamasis vizualizavimas, kurio tikslas naujų ryšių, struktūrų duomenyse radimas, hipotezių kėlimas iš gautų vizualizavimo rezultatų. Kitas tikslas yra vizualiai pateikti naują gautą informaciją ir parodyti jos vietą tarp esamos informacijos. Todėl ypatingai atkreipsime dėmesį į tuos vizualizavimo metodus, kurie:

- ✓ Skirti daugiamačių duomenų vizualizavimui;
- ✓ Jungiantys kelis duomenų gavybos ir vizualizavimo metodus.

Duomenų gavybos jungimas su vizualizavimu dar vadinamas *vizualia duomenų gavyba* (angl. *visual data mining*) (Keim 2002), (de Oliviera and Levkowitz 2003), (Wong 1999). Bendru atveju galima sakyti, kad vizuali duomenų gavyba yra sandūroje tarp duomenų gavybos ir vizualizavimo.

Vizualizavimas žinių gavyboje turi apimti visus šešis žinių gavybos proceso žingsnius (2.3 paveikslas). Tai vizualus pradinės duomenų aibės tyrinėjimas (1 – 3 žingsniai); rezultatų, gautų pasirinktais duomenų gavybos metodais, vizualizavimas ir interpretavimas; priimamas automatizuotas sprendimas ir vizualiai pateikiamas, parodant jo vietą tarp esamų sprendimų (4 – 5 žingsniai); vizualiai pateikiami gauti rezultatai ekspertui, kuris gautas žinias lygina su anksčiau turėtomis, jas interpretuoja, nusprendžia ar gautus rezultatus atmesti ar gautais rezultatais papildyti žinių banką (6 žingsnis).

Vizualizavimas tyrėjui daro žinių radimo procesą, naudojant duomenų gavybos metodus, daug aiškesniu ir suprantamesniu.

2.4. Vizualizavimo metodai

Daugiamačių duomenų vizualizavimas leidžia tyrinėtojui pačiam stebėti tų duomenų grupavimosi tendencijas, įvertinti atskirų daugiamačių taškų tarpusavio artumą, racionaliai priimti sprendimus. Kaip jau minėta anksteniasme skyriuje, yra daug apžvalgų susisteminančių vizualizavimo metodus: (Dzemyda *et al.* 2008), (Medvedev 2007), (Kurasova 2005), (Card *et al.* 1999), (Chambers *et al.* 1983), (Grinstein and Ward 2002), (Hoffman and Grinstein 2002), (Keim 2002), (Wong 1997). Vizualizavimo metodai grindžiami skirtingomis idėjomis, universalūs ir orientuoti specialioms duomenų struktūroms.

Galima išskirti dvi pagrindines vizualizavimo metodų grupes:

1. tiesioginio vizualizavimo metodai, kuomet kiekvienas daugiamačio taško elementas yra pateikiamas tam tikra vizualia forma;
2. projekcijos metodai, dar vadinami dimensijos mažinimo metodais (angl. *dimension reduction techniques*), kurie leidžia daugiamačius taškus pateikti mažesnės dimensijos erdvėje.

Pateikiant daugiamačių duomenų vizualizavimo, klasifikavimo, klasterizavimo pavyzdžių iliustracijas, bus naudojami **Fišerio irisų duomenys** (Fisher 1936), kurie dažnai vadinami tiesiog irisais arba irisų duomenimis. Tai klasikiniai testiniai duomenys, naudojami daugiamačių duomenų analizėje. Šią duomenų aibę galima rasti duomenų saugykloje „UCI Repository of Machine Learning Databases“ (Asuncion and Newman 2007)

Buvo matuota trijų veislių gėlių (Iris Setosa (I klasė), Iris Versicolor (II klasė) ir Iris Virginica (III klasė)) šie parametrai:

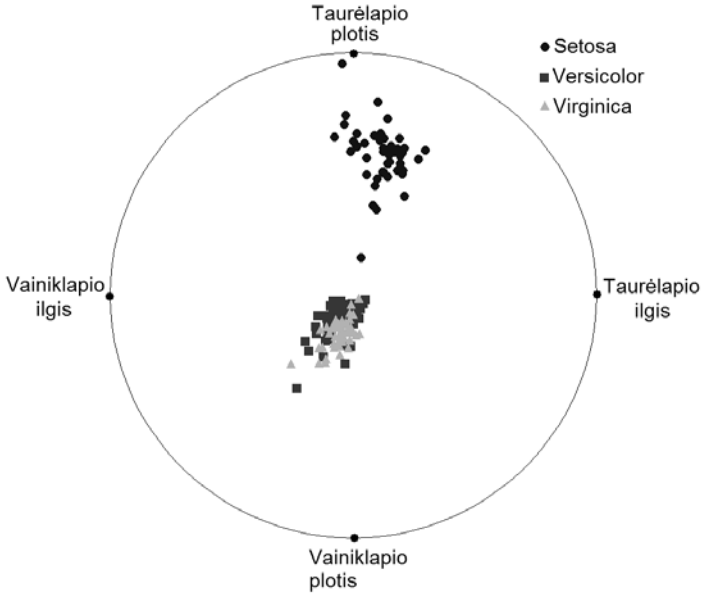
- vainiklapių pločiai (angl. *petal weight*),
- vainiklapių ilgiai (angl. *petal height*),
- taurėlapių pločiai (angl. *sepal weight*),
- taurėlapių ilgiai (angl. *sepal height*).

Iš viso matuota 150 gėlių. Sudaryti 4-mačiai ($n=4$) vektoriai $X_1, X_2, ..., X_{150}$ ($X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$, $i = 1,...,150$). Įvairiais metodais yra nustatyta, kad I klasės irisai „atsiskiria“ nuo kitų dviejų klasių (II ir III). II ir III klasės dalinai persidengia.

Toliau išvardijami vizualizavimo metodai, kurie yra priskiriami vienam iš metodų tipų. Taip pat pateikiami šių metodų taikymo pavyzdžiai.

Tiesioginio vizualizavimo metodai:

1) ***Geometriniai metodai***: taškiniai grafikai (angl. *scatter plots*), taškinių grafikų matricos (angl. *matrix of scatter plots*), linijiniai grafikai (angl. *line graphs, multiline graphs*), perstatymų matrica (angl. *permutation matrix*), apžiūros grafikai (angl. *survey plots*), Andrews kreivės, visų galimų projekcijų peržiūra (angl. *grand tours*), lygiagrečios koordinatės, spindulinis vizualizavimas (*RadViz*), jo modifikacijos (*GridViz*, *PolyViz*), projekcijos paieška (angl. *projection pursuit*).



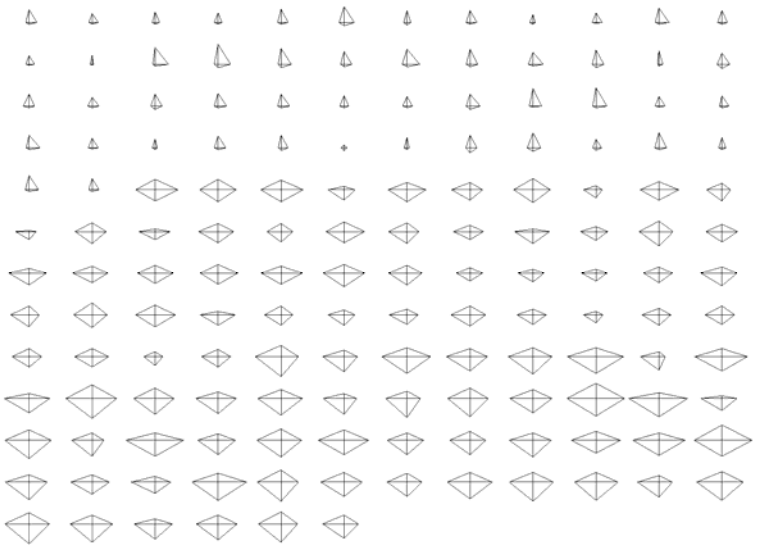
2.4 pav. *Spindulinio vizualizavimo metodu vizualizuoti irisų duomenys*

2.4 paveiksle pateikta vienos geometrinio metodo – spindulinio vizualizavimo (angl. *radial visualization*) (RadViz) – iliustracija. Čia vizualizuoti irisų duomenys. Spindulinio vizualizavimo metode duomenims vizualizuoti taikoma spyruoklės veikimo paradigma: iš skritulio centro nubrėžiama *n* spindulių (tiek, kiek yra analizuojamų duomenų komponenčių (parametrų)); taškai, kuriuose tie spinduliai susikerta su skritulio lanku, vadinami „inkarais“ (angl. *dimensional anchors*); vieni spyruoklių galai „pritvirtinami“ prie kiekvieno

„inkaro“, kiti galai – prie vizualizuojamo duomenų taško; spyruoklių konstantos imamos lygios vizualizuojamo taško komponenčių reikšmėms; duomenų taškas yra atidedamas tame skritulio taške, kuriame visų spyruoklių jėgų suma lygi nuliui. Detaliau apie šį metodą skaitykite (Dzemyda *et al.* 2008).

2) ***Simboliniai metodai*** (angl. *iconographic display*): Černovo veidai, žvaigždžių metodas (angl. *star glyphs*), brūkšnelinės figūros (angl. *stick figure*), ženklų metodas (angl. *glyphs*), nuspalvintos piktogramos (angl. *color icon*).

Vizualizavimo iliustracija, gauta naudojant simbolinį metodą, pateikiama 2.5 paveiksle. Čia irisų duomenys vizualizuoti naudojant žvaigždžių metodą (angl. *star glyphs*) (Kaski 1997) (naudota XmdvTool 7.0 sistema (http://davis.wpi.edu/~xmdv/)). Kiekvienas vizualizuojamas duomuo vaizduojamas stilizuota žvaigžde (2.5 paveikslas). Iš vieno taško nubraižoma tiek spindulių, kiek yra analizuojamų duomenų komponenčių (parametrų). Spindulio ilgis priklauso nuo jį atitinkančios komponentės (parametro) reikšmės. Išoriniai spindulių galai yra sujungiami linijomis. Iš paveikslo matome, kad I klasės (Iris Setosa) žvaigždės yra mažesnės už kitų dviejų klasių. Didžiausios žvaigždės atitinka III klasę (Iris Virginica).



2.5 pav. *Žvaigždžių metodu vizualizuoti irisų duomenys*

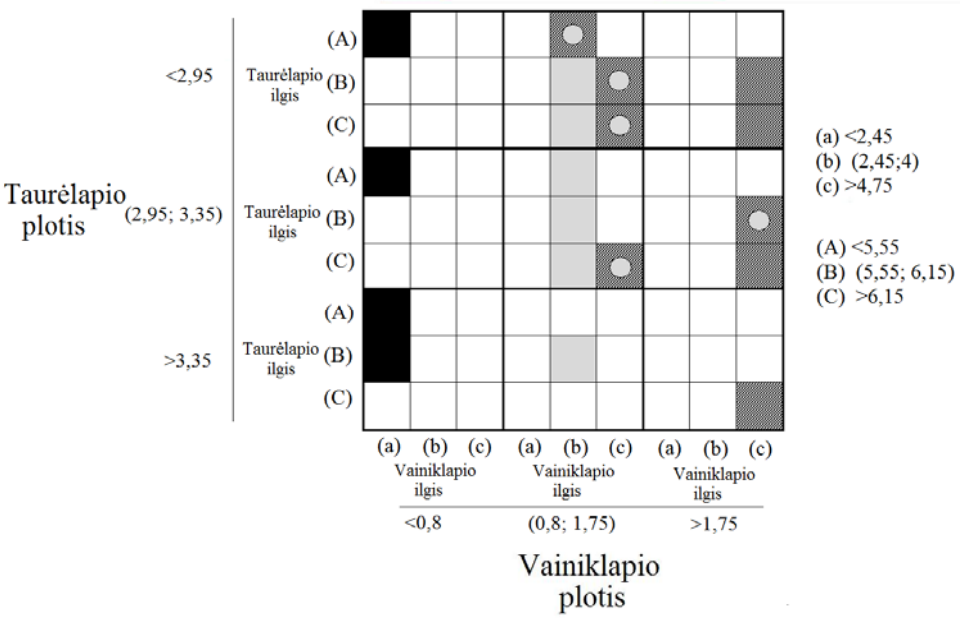
3) ***Hierarchinio vizualizavimo metodai*** (angl. *hierarchical display*): dimensijų įterpimo metodas (angl. *dimensional stacking*), grotelių metodas (angl. *trellis display*), „Pasaulis pasaulyje“ (angl. *world-within-world*), informacijos

kubas (angl. *info cube*), fraktalų metodas (angl. *fractal foam*), dendogramos, hierarchinės lygiagrečios koordinatės.

Vizualizavimo iliustracija, gauta naudojant vieną hierarchinio vizualizavimo metodą – dimensijos įterpimo metodą – pateikiama 2.6 paveiksle. Čia vizualizuoti irisų duomenys.

Dimensijos įterpimo metodas yra rekursinis atvaizdavimo metodas iš *n*-matės erdvės į dvimatę (LeBlanc *et al.* 1990). Metodo veikimo schema yra tokia: analizuojamų duomenų parametrų reikšmių kitimo intervalas suskaidomos į pointervalius; nubraižomas dvimatis tinklelis; jis padalyjamas į stačiakampius, kurių skaičius priklauso nuo pointervalių skaičiaus; du pasirinkti duomenų parametrai pavaizduojami *x* ir *y* ašyse, jie vadinami išoriniais parametrais; kiekviena kita parametrų pora yra „įspraudžiama“ į taip vadinamus vidinius stačiakampius; toks įterpimas tęsiamas tol, kol įterpiami visi parametrai; vidiniai stačiakampiai, kuriuose daugiau nėra vidinių stačiakampių, nuspalvinami, jeigu analizuojamoje duomenų aibėje yra vektorių, kurių parametrų reikšmės atitinka šiuos vidinius stačiakampius.

Analizuojant irisų duomenis, visų keturių irisus apibūdinančių parametrų reikšmės suskaidomos po tris pointervalius. Vainiklapio plotis ir taurėlapio plotis pasirenkami kaip išoriniai parametrai, o vainiklapio ilgis ir taurėlapio plotis – kaip vidiniai parametrai. Skirtingų veislių irisus atitinkantys langeliai nuspalvinami skirtingais atspalviais (2.6 paveikslas).

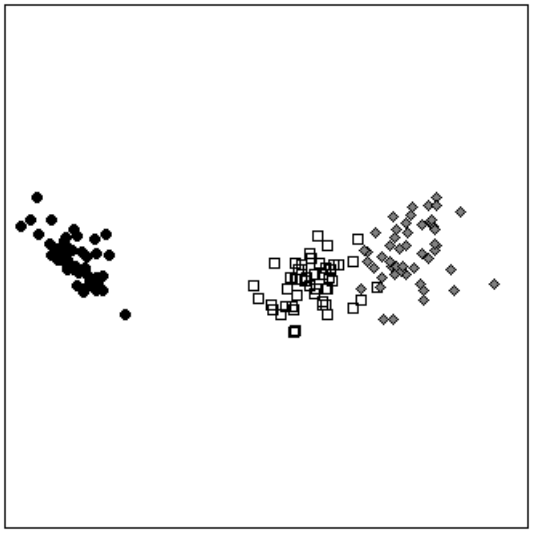


2.6 pav. *Dimensijų įterpimo metodo pavyzdys (irisų duomenų aibė)*

Projekcijos metodai:

Projekcijos metoduose yra formalus matematinis kriterijus, pagal kurį minimizuojamas projekcijos iškraipymas. Projekcijos metodai skirstomi į dvi grupes:

1) ***Tiesinės projekcijos metodai***: pagrindinių komponenčių analizė (angl. *principal component analysis*), tiesinės diskriminantinės analizės metodas (angl. *linear discriminant analysis*) (LDA), faktorinės analizės metodas; projekcijos paieška (angl. *projection pursuit*).



2.7 pav. *LDA metodu vizualizuoti irisų duomenys*

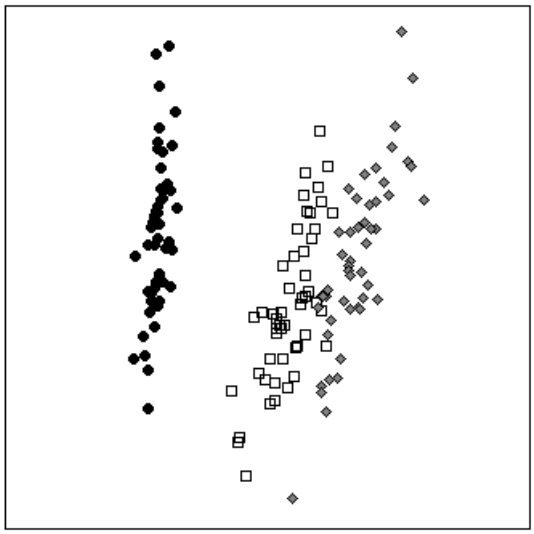
2.7 paveiksle pateiktas tiesinės diskriminantinės analizės metodu (angl. *linear discriminant analysis*) (LDA) gautas daugiamačių taškų išsidėstymas plokštumoje vizualizuojant irisų duomenis.

Šiuo metodu gali būti analizuojami duomenys, kurių klasės iš anksto žinomos. LDA metodas transformuoja daugiamatės erdvės duomenis į mažesnės dimensijos erdvę taip, kad klasių atskiriamumo kriterijaus reikšmė būtų optimali. Ieškoma tokių krypčių, kuriomis klasės yra geriausiai atskiriamos (Dzemyda *et al.* 2008), (Duda *et al.* 2000).

2) ***Netiesinės projekcijos metodai***: daugiamatės skalės (angl. *multidimensional scaling*) ir šio metodo atskiras atvejis – Sammono projekcija (angl. *Sammon mapping, projection*), pagrindinės kreivės (angl. *principal curves*), savireguliuojantys neuroniniai tinklai (SOM) (angl. *self organizing map*), trianguliacijos metodas, reliatyvios perspektyvos žemėlapis (angl. *relative*

perspective map), lokaliai tiesinis atvaizdavimas (angl. *locally linear embedding*) (LLE).

2.8 paveiksle pateiktas LLE metodu gautas daugiamačių taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant irisų duomenis. LLE metodas nereikalauja išlaikyti atstumų tarp labiausiai nutolusių duomenų taškų, o tik tarp artimiausių kaimynų. Detaliau apie šį metodą skaitykite (Medvedev 2007), (Roweis and Saul 2000).



2.8 pav. *LLE metodu vizualizuoti irisų duomenys*

Projekcijos metodai lyginant su tiesioginio vizualizavimo metodais yra populiaresni, nes jų rezultatai lengviau suvokiami ir interpretuojami.

Čia išvardinti toli gražu ne visi metodai. Yra bandymų apjungti kelis skirtingais principais grindžiamus vizualizavimo metodus. Dažnai klasikiniai vizualizavimo metodai jungiami su metodais, grindžiamais dirbtiniais neuroniniais tinklais, pavyzdžiui, SOM ir Sammon projekcijos junginiai (Kohonen 2002), (Kurasova 2005), SAMANN algoritmas (Mao and Jain 1995), (Medvedev 2007). Tokie skirtingų metodų junginiai leidžia atrasti daugiau naujų žinių apie analizuojamus duomenis, palyginus su atskirų metodų taikymu.

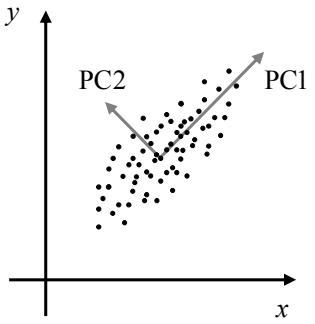
Toliau detalizuosime disertacijoje giliau tirtus ir taikytus metodus.

2.4.1. Pagrindinių komponenčių analizė

Pagrindinių komponenčių analizė (angl. *principal component analysis,* PCA) yra klasikinis statistikinis tiesinės projekcijos metodas, plačiai naudojamas

duomenų analizėje (pavyzdžiui, klasifikavimui arba atpažinimui). PCA metodas yra naudojamas daugiamačių duomenų dimensijai mažinti. Šiuo metodu ieškoma daugiamačių duomenų mažesnės dimensijos poerdvio, kuriame būtų išlaikyta kiek įmanoma daugiau originalios erdvės duomenų savybių bei informacijos. Esminė PCA idėja – sumažinti duomenų dimensiją kiek galima tiksliau išlaikant duomenų dispersijas. Tai padeda geriau vizualizuoti duomenis, o tuo pačiu ir palengvina duomenų suvokimą.

PCA metodo tikslas – rasti kryptį, kuria dispersija yra didžiausia (Jollife 1986), (Haykin 1999), (Taylor 2003). Didžiausią dispersiją turinti kryptis vadinama pirmąja pagrindine komponente (*PC1*). Ji eina per duomenų centrinį tašką. Taškų visumos kvadratinis vidutinis atstumas iki šios tiesės yra minimalus, t. y. ši tiesė yra kiek galima arčiau visų duomenų taškų (2.9 paveikslas). Antrosios pagrindinės komponentės (*PC2*) ašis taip pat eina per duomenų centrinį tašką ir ji ortogonali pirmosios pagrindinės komponentės ašiai.



2.9 pav. *Pirmoji (PC_1) ir antroji (PC_2) pagrindinės komponentės*

Tegu turime duomenų matricą $\mathbb{X}$, sudarytą iš vektorių $X_i = (x_{i1}, x_{i2}, ..., x_{in}) = \{x_{ij}\}$, $i = 1,...,m$, $j = 1,...,n$, čia m yra vektorių skaičius, n – parametrų skaičius.

Koreliacijos tarp *k*-tojo ir *l*-tojo vektorių-stulpelių $X_k = \{x_{ik}\}$ ir $X_l = \{x_{il}\}$, $i = 1,...,m$, koeficientas r_{kl} skaičiuojamas pagal (2.1) formulę. *Koreliacija* – tai dviejų kintamųjų tarpusavio sąryšis.

$$r_{kl} = \frac{\sum_{i=1}^{m} (x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l)}{\sqrt{\sum_{i=1}^{m} (x_{ik} - \bar{x}_k)^2 \cdot \sum_{i=1}^{m} (x_{il} - \bar{x}_l)^2}}, \qquad (2.1)$$

čia $\bar{x}_k = \frac{1}{m}\sum_{j=1}^{m} x_{ik}$, $\bar{x}_l = \frac{1}{m}\sum_{j=1}^{m} x_{il}$.

Koreliacinė matrica $R = \{r_{kl}\}$ yra suformuojama iš koreliacijos koeficientų, apskaičiuotų pagal (2.1) formulę. Matricos R įstražainės elementai lygūs vienetui.

Kovariacijos koeficientas C_{kl} yra skaičiuojamas pagal (2.2) formulę.

$$C_{kl} = \frac{1}{m-1}\sum_{i=1}^{m}(x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l), \text{ kai } k \neq l \ . \qquad (2.2)$$

Kai $k = l,$ (2.2) formulė yra dispersijos formulė.

Iš kovariacijos koeficientų, apskaičiuotų pagal (2.2) formulę, galima suformuoti *kovariacinę matricą* $C = \{C_{kl}\}$. Kovariacinė matrica yra simetrinė matrica. Jei du vektoriai nekoreliuoti, tai jų kovariacijos koeficientas $C_{kl} = C_{lk} = 0, k \neq l.$

Norint rasti duomenų pagrindines komponentes, reikia skaičiuoti duomenų kovariacinės matricos tikrines reikšmes λ_k (angl. *eigenvalue*) ir tikrinius vektorius e_k (angl. *eigenvector*). Tikrinių vektorių skaičius yra lygus kintamųjų skaičiui n. Tikrinis vektorius, susijęs su didžiausia tikrine reikšme, turi tokią pat kryptį kaip pirmoji pagrindinė komponentė. Antroji pagrindinė komponentė atitinka didžiausią iš likusių tikrinių vektorių ir t.t.

Pagrindinės komponentės yra lygties $Ce_k = \lambda_k e_k$ sprendinys e; čia e_k yra vektorius-stulpelis, C – duomenų kovariacinė matrica, λ_k – tikrinė reikšmė, randama iš charakteringos lygties $|C - \lambda_k I| = 0.$ Čia I yra vienetinė matrica, ženklu |.| apibrėžtas determinantas. Pagrindinių komponenčių skaičius parenkamas atsižvelgiant į projektinės erdvės dimensiją. Tikrinių vektorių ir reikšmių radimas nėra trivialus uždavinys, tačiau yra sukurta nemažai šio uždavinio sprendimo metodų.

Tegu matrica $A = \{e_j\}, \ j = 1,...,n$ yra sudaryta iš tikrinių vektorių e_j kaip vektorių-eilučių. Kiekvienas šios matricos vektorius-stulpelis yra ortogonalus bet kuriam kitam. Bet kurį duomenų vektorių X galime transformuoti pagal (2.3) formulę ir gauti tašką Y naujoje ortogonalioje koordinačių sistemoje, apibrėžtoje tikriniais vektoriais $e_j, \ j = 1,...,n.$

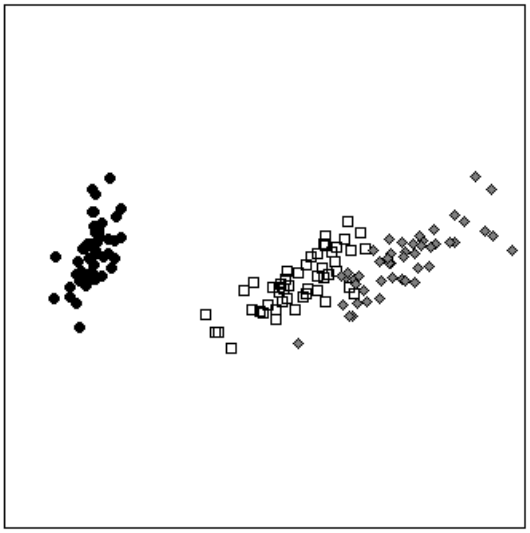
$$Y = (X - \bar{X})A, \qquad (2.3)$$

čia $X = (x_1, x_2, ..., x_n)$, $\bar{X} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$, $A = (e_1, e_2, ..., e_n)$.

Dabar galima išreikšti originalų duomenų vektorių X per vektorių Y pagal šią formulę: $X = YA^T + \bar{X}$. Ji gauta iš (2.3) formulės pasinaudojus ortogonalių matricų savybe, kad $A^{-1} = A^T$, čia A^{-1} yra atvirkštinė matrica, A^T – transponuota matrica.

Dažnai duomenys turi tokią savybę, kad tik dalis jų tikrinių reikšmių yra esminės, kitų reikšmės yra labai mažos, ir neverta skaičiuoti visų n pagrindinių komponenčių, o tik d $(d < n)$. Tuomet matrica A_d sudaroma tik iš d tikrinių vektorių turinčių didžiausias nuosavas reikšmes. Yra skurta įvairių metodikų, kaip iš anksto (ar algoritmo metu) nustatyti skaičiaus d reikšmę. Jei analizės tikslas – rasti pagrindines komponentes ir transformuotus duomenis pavaizduoti plokštumoje ar trimatėje erdvėje, t. y. juos vizualizuoti, tuomet $d = 2$ arba $d = 3$.

PCA metodu vizualizuoti irisų duomenys, kai duomenys atvaizduojami plokštumoje, pateikti 2.10 paveiksle.



2.10 pav. *PCA metodu vizualizuoti irisų duomenys*

Kaip minėta anksčiau, PCA metodas yra projekcijos metodas, kuriame nagrinėjamas tik tiesinės priklausomybės, todėl jis netinkamas daugiamačių duomenų dimensijai mažinti, kai tarp tų duomenų egzistuoja stiprūs netiesiniai sąryšiai. Taip pat standartiniai PCA metodai netinka duomenų, sudarytų iš daug parametrų, analizei. Tuomet daugiamačių duomenų kovariacinė matrica yra labai didelių apimčių. Jos dydis yra $n \times n$. PCA metodas nenaudotinas vizualizuoti daugiamačius duomenis, sudarytus iš kelių šimtų parametrų.

2.4.2. Daugiamačių skalių metodas

Daugiamačių skalių metodas (angl. *multidimensional scaling) (DS*) plačiai naudojamas daugiamačių duomenų analizėje įvairiose šakose: ekonomikoje, socialiniuose, medicinos moksluose ir kt. *n*-mačiai vektoriai projektuojami į mažesnės dimensijos erdvę (dažniausiai į dvimatę ar trimatę erdvę) siekiant kaip galima tiksliau išlaikyti atstumus (nepanašumus) tarp analizuojamos aibės objektų (Borg and Groenen 1997), (Cox, T. F. and Cox, M. A. A. 1994), (Kruskal and Wish 1984). Analizės rezultate gautuose dvimačiuose grafikuose tie objektai, kurie yra panašūs, yra pavaizduojami arčiau vieni kitų, o mažiau panašūs – toliau vieni nuo kitų. Minimizavimo problema yra sprendžiama taip, kad atstumai tarp taškų mažesnės dimensijos erdvėje atitiktų duotus nepanašumus kaip įmanoma geriau. Atvaizdavimas paprastai yra netiesinis, ir padeda atskleisti bendrą analizuojamų duomenų struktūrą.

Pradiniai duomenys, kurie analizuojami šiuo metodu, turi būti kvadratinė, simetrinė matrica, susidedanti iš ryšių tarp analizuojamų duomenų aibės elementų. Tai gali būti atstumų arba nepanašumų matrica. Ryšiais tarp aibės elementų gali būti ir Euklido atstumai. Tačiau, bendru atveju, tai nebūtinai turi būti atstumai griežtai matematine prasme.

Vienas DS pavyzdys galėtų būti toks: tarkime turime matricą, sudarytą iš atstumų tarp pagrindinių šalies miestų; DS analizės rezultate gautume miestų išdėstymą žemėlapyje, t. y. dvimatėje plokštumoje (Leeuw and Liere 2003). DS matricos pavyzdys yra koreliacijų tarp duomenų parametrų matrica. Jei tie duomenys traktuojami kaip panašumai, DS algoritmu stipriai koreliuoti parametrai atvaizduojami arti vieni kitų, silpnai koreliuoti – toliau vieni nuo kitų.

Vienas DS tikslų yra rasti optimalią daugiamačių duomenų konfigūraciją dvimatėje erdvėje. Yra daugybė skirtingų DS variantų su skirtingomis paklaidų funkcijomis (STRESS) ir jas optimizuojančiais algoritmais (Borg and Groenen 1997). Pagal analizuojamus duomenis DS algoritmai gali būti skirstomi į *metrinius* (angl. *metric*) ir *nemetrinius* (angl. *non-metric*). Pirmasis DS algoritmas metriniams duomenims buvo pasiūlytas W.S. Torgensono 1952 metais (Torgenson 1952), vėliau DS algoritmai buvo taikyti ir nemetriniams duomenis (Shepard 1962a), (Shepard 1962b).

Metriniai DS algoritmai (Taylor 2003) naudojami tada, kai įmanoma rasti Euklido atstumus tarp analizuojamų duomenų elementų, t. y. analizuojami metriniai duomenys. Pagrindinis šių algoritmų tikslas – pavaizduoti daugiamačius taškus dvimatėje erdvėje taip, kad atstumai tarp dvimačių taškų būtų kiek galima artimesni atstumams tarp atitinkamų daugiamačių taškų. Tam minimizuojama tam tikra paklaidos funkcija.

Tarkime kiekvieną *n*-matį vektorių $X_i \in R^n$ atitinka mažesnės dimensijos vektorius $Y_i \in R^d$, $(d < n)$. Pažymėkime atstumą tarp vektorių X_i ir X_j – d_{ij}^*,

o atstumą tarp vektorių Y_i ir Y_j – d_{ij}, $i, j = 1,...,m$. Metrinis DS algoritmas bando priartinti atstumus d_{ij} prie atstumų d_{ij}^*. Jei naudojama kvadratinė paklaidos funkcija, tai minimizuojama tikslo funkcija E_{DS} gali būti užrašyta taip:

$$E_{DS} = E_{DS}(Y) = \sum_{i<j}^{m} w_{ij}(d_{ij}^* - d_{ij})^2. \qquad (2.4)$$

Paklaidos funkcija E_{DS} dar vadinama STRESS funkcija. Faktiškai E_{DS} yra funkcija, kurios reikšmės priklauso nuo dvimačių vektorių $Y_i = (y_{i1}, y_{i2})$, čia $i = 1,...,m$, koordinačių reikšmių, t. y. E_{DS} priklauso nuo $2m$ kintamųjų. Todėl $E_{DS} = E_{DS}(Y)$, čia $Y = \{Y_1, Y_2, ..., Y_m\}$. Dažnai naudojami tokie svoriai:

$$w_{ij} = \frac{1}{\sum\limits_{\substack{k,l=1 \\ k<l}}^{m} (d_{kl}^*)^2}; \; w_{ij} = \frac{1}{d_{ij}^* \sum\limits_{\substack{k,l=1 \\ k<l}}^{m} d_{kl}^*}; \; w_{ij} = \frac{1}{m d_{ij}^*},$$

tačiau šioje disertacijoje naudojami svoriai $w_{ij} = 1$.

Vienas iš paprasčiausių šių funkcijų minimizavimo būdų – gradientinis nusileidimas. Pradėjus nuo atsitiktinės pradinių dvimačių taškų konfigūracijos, iteraciniame procese dvimačių vektorių $Y_i \in R^2$ koordinatės y_{ik}, $i = 1,...,m$, $k = 1, 2$, keičiamos pagal formulę $y_{ik}(m'+1) = y_{ik}(m') - \eta \frac{\partial E_{DS}(m')}{\partial y_{ik}(m')}$. Čia m' yra iteracijos numeris, o η – parametras, įtakojantis optimizavimo žingsnį.

Literatūroje minimi ir kiti paklaidos funkcijos optimizavimo būdai, tokie kaip jungtinių gradientų metodas, kvazi-Niutono metodas, deterministinis atkaitinimo modeliavimo algoritmas (angl. *simulated annealing*) (Klock and Buhmann 1999), evoliucinis algoritmas (Dzemyda *et al.* 2008), kombinatorinis DS algoritmas (Žilinskas, A. and Žilinskas, J. 2007), šakų ir rėžių algoritmas (Žilinskas, A. and Žilinskas, J. 2008), genetinio algoritmo ir lokalaus nusileidimo metodų kombinacijos (Mathar and Žilinskas 1993), (Podlipskytė 2003), SMACOF (angl. *scaling by majorization a complicated function*) algoritmas, pagrįstas tikslo funkcijos mažorizavimu (Borg and Groenen 1997).

Šioje disertacijoje tyrimams bus naudojamas iteracinis SMACOF algoritmas:

$$Y(m'+1) = V^+ B(Y(m'))Y(m'),$$

čia $B(Y(m'))$ matricos elementai apskaičiuojami pagal formules:

$$b_{ij} = \begin{cases} -\dfrac{w_{ij} d^*_{ij}}{d_{ij}}, & \text{kai } i \neq j \text{ ir } d_{ij} \neq 0 \\ 0, & \text{kai } i \neq j \text{ ir } d_{ij} = 0 \end{cases},$$

$$b_{ii} = -\sum_{j=1,\, j\neq i}^{s} b_{ij}.$$

V yra svorių matrica:

$$V = \begin{pmatrix} \sum w_{1s} & & & & \\ & \ddots & & -w_{ij} & \\ & & \ddots & & \\ & -w_{ij} & & \ddots & \\ & & & & \sum w_{ns} \end{pmatrix}.$$

V^+ yra matricos V Moore-Penrose pseudoinversinė matrica. Šiuo atveju visi svoriai w_{ij} =1, todėl $V^+ = \dfrac{1}{n} I.$

SMACOF algoritmo schema pateikiama 2.11 paveiksle. Šis algoritmas apibūdinamas keliais žingsniais:

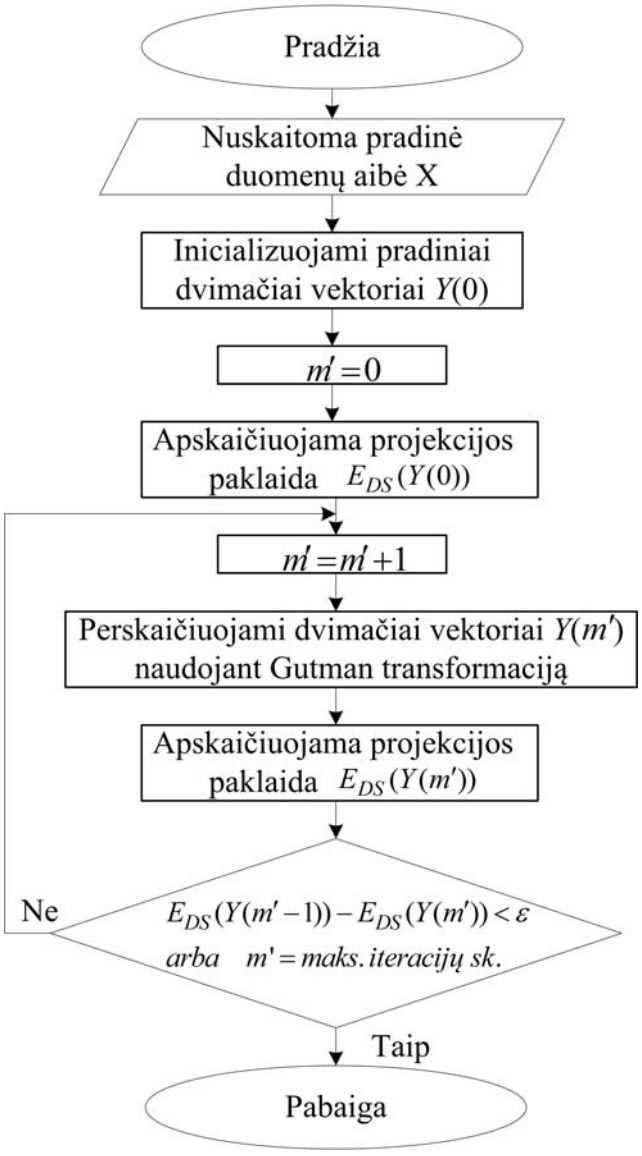
1. Inicializuojami dvimačiai vektoriai Y, pradinė iteracija lygi $m' = 0$.
2. Apskaičiuojama projekcijos paklaida $E_{DS}(Y(m'))$; čia Y dvimačiai vektoriai, apskaičiuoti pirmame žingsnyje $m' = 0.$
3. Iteracijų skaičių m' didiname vienetu.
4. Apskaičiuojame Guttman transformaciją

   $$Y(m') = V^+ B(Y(m'-1))Y(m'-1).$$

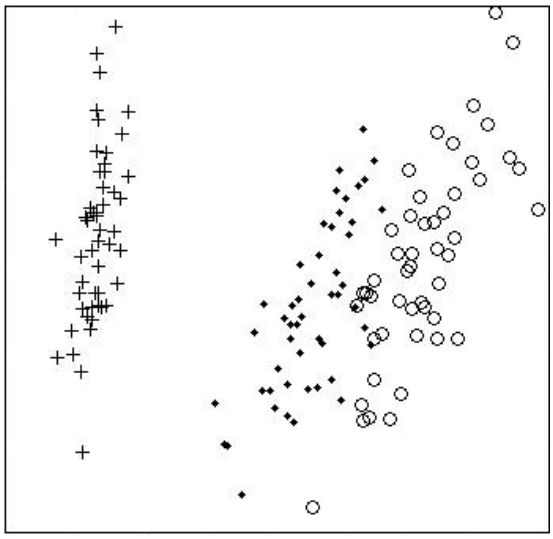
5. Skaičiuojame projekcijos paklaidą $E_M(Y(m'))$. Jei $E_{DS}(Y(m'-1)) - E_{DS}(Y(m')) < \varepsilon$ arba m' lygus maksimaliam nustatytam iteracijų skaičiui, tuomet stabdomas iteracinis procesas (ε yra nustatytas tikslumas), priešingu atveju einame į 3 punktą ir procesą kartojame.

Funkcijų STRESS ir SSTRESS minimizavimo uždaviniai yra sudėtingi dėl kriterijų daugiaekstremalumo (Žilinskas and Podlipskytė 2003). Darbe (Podlipskytė 2003) yra pateikta kriterijų STRESS ir SSTRESS galimų optimizavimo strategijų apžvalga bei jų tyrimai.

2.12 paveiksle pateiktas DS rezultate gautas daugiamačių taškų išsidėstymas dvimatėje plokštumoje vizualizuojant irisų duomenis.



2.11 pav. *Daugiamačių skalių SMACOF algoritmo schema*



2.12 pav. *DS metodu (SMACOF algoritmas) vizualizuoti irisų duomenys*

Kartais, analizuojant objektus, yra prasmingos ne atstumų skaitinės reikšmės, o atstumų tarp objektų eilės numeriai. Tada tikslinga naudoti *nemetrinius* DS algoritmus, kuriuose objektų nepanašumai nėra atstumai. Nepanašumą tarp *i*-tojo ir *j*-tojo objektų apibrėžkime realiu skaičiumi δ_{ij}. Kartais šis matas nėra tinkamas Euklido erdvei, todėl jis transformuojamas funkcija *f*: $f(\delta_{ij})$. Nemetriniuose DS algoritmuose dažniausiai minimizuojama (2.5) paklaidos (STRESS) funkcija:

$$\tilde{E}_{DS} = \sum_{i<j}^{m} w_{ik}(f(\delta_{kl}) - d_{kl})^2. \qquad (2.5)$$

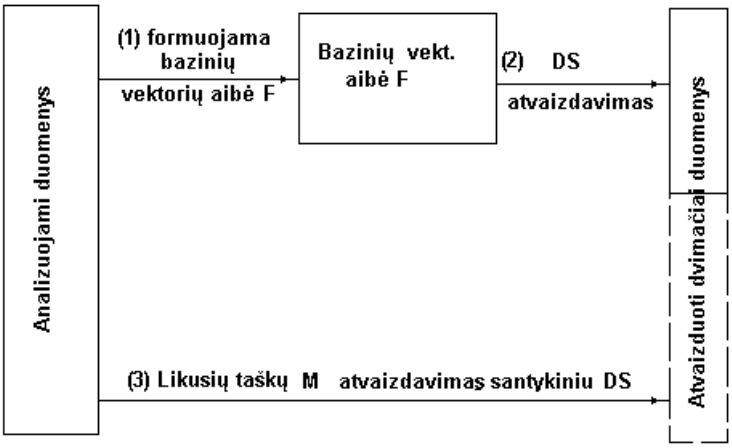
DS algoritmo trūkumai: DS algoritmo tikslo funkcijos optimizavimas reikalauja palyginimų tarp visų vektorių porų; DS algoritmai yra neefektyvus, dirbant su didelės apimties duomenų aibėmis; DS negali vizualizuoti naujus taškus tol, kol nebus perskaičiuoti visi analizuojami vektoriai. Todėl pastaruoju metu yra pasiūlyta daug DS algoritmo modifikacijų, kurios padeda šių problemų išvengti (Basalaj 1999), (Naud and Duch 2000), (Bernatavičienė *et al*. 2007a), (Bernatavičienė *et al* 2007c) ir kiti.

2.4.3. Santykinių daugiamačių skalių algoritmas

Santykinių daugiamačių skalių algoritmas (angl. *relative MDS*) yra detaliai aprašytas darbe (Naud and Duch 2000). Klasifikuojant ir atliekant vizualią duomenų analizę labai įdomu nustatyti naujo tiriamojo arba objekto duomenų

padėtį jau esamų duomenų atžvilgiu ir atsižvelgiant į gautus rezultatus priskirti jį vienai iš žinomų klasių. Naudojant klasikinį daugiamačių skalių metodą, mes negalime atidėti naujo taško tarp jau suprojektuotų taškų, reiktų konstruoti naują aibę ir ją visą projektuoti į plokštumą. Naujų taškų atvaizdavimui naudojamas santykinių daugiamačių skalių metodą (santykinių DS alg.). Tarkime, kad žinomų taškų skaičius yra N_{fiks}, naujų taškų skaičius N_{nauji}, o visų vizualizuojamų taškų skaičius yra N_{visi} ($N_{visi} = N_{fiks} + N_{nauji}$). Žinomų taškų aibė žymėsime *F*, o naujų taškų aibę – *M*. Algoritme išskiriami du etapai:

1. Naudojant DS metodą projektuojama bazinių vektorių aibė *F* (fiksuotų taškų skaičius lygus N_{fiks});
2. Nauji aibės *M* taškai yra projektuojami atsižvelgiant į bazinių taškų projekcijas, naudojant santykinių DS metodą (naujų taškų skaičius N_{nauji}).

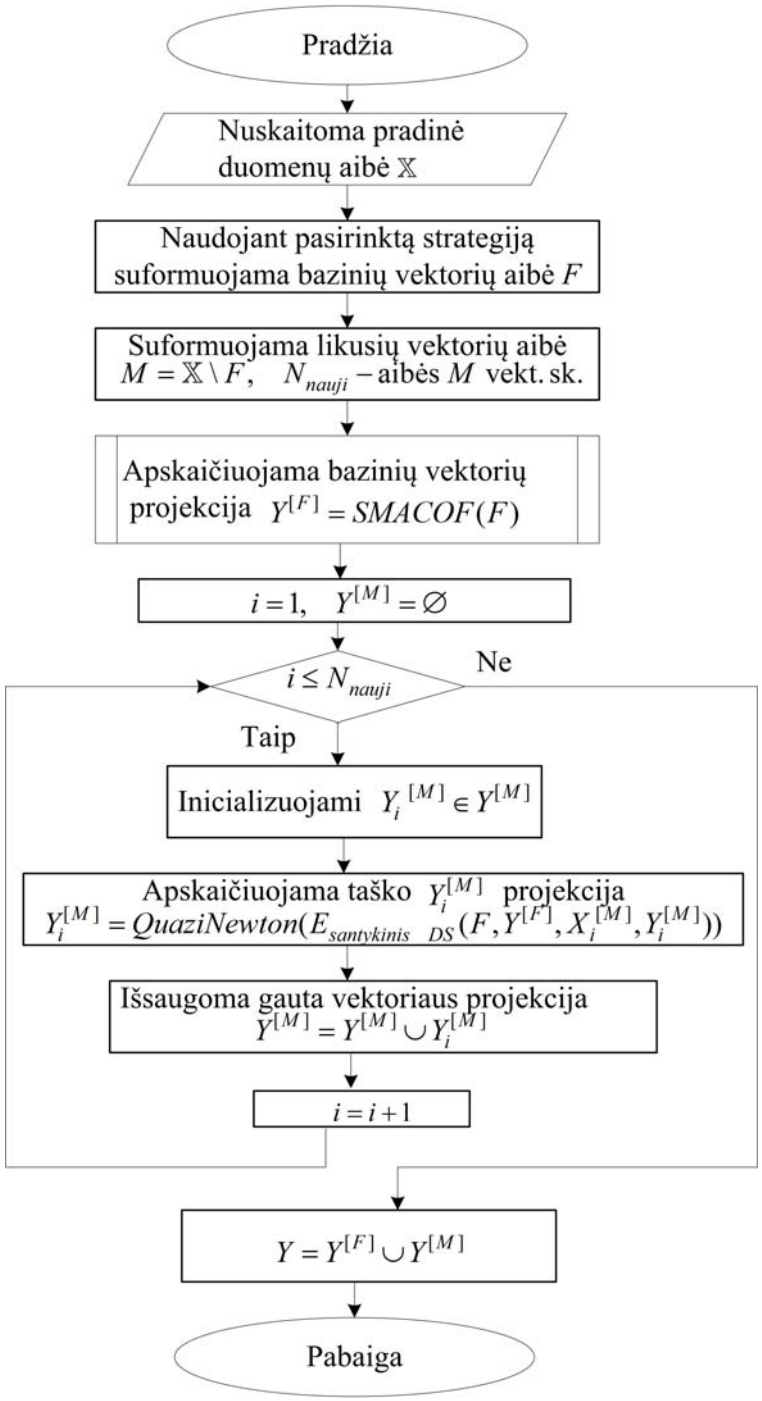


2.13 pav. *Santykinių DS algoritmas*

Santykinių DS algoritmas skiriasi nuo įprasto DS tuo, kad minimizuojant tikslo funkciją perskaičiuojamos tik naujų taškų projekcijos, o aibės *F* taškų projekcijos lieka fiksuotos. Santykinių DS algoritmo minimizuojama tikslo funkcija:

$$E_{santykinis\_DS} = \sum_{i<j}^{N_{nauji}} (d_{ij}^* - d_{ij})^2 + \sum_{i=1}^{N_{nauji}} \sum_{j=N_{nauji}+1}^{N_{visi}} (d_{ij}^* - d_{ij})^2 .$$

Eksperimentuose santykinių DS algoritmo tikslo funkcijos minimizavimui buvo naudotas kvazi-Niutono algoritmas. Detali šio algoritmo schema pateikta 2.14 paveiksle.



2.14 pav. *Santykinių DS algoritmo schema*

Schemoje naudojami žymėjimai:

$\mathbb{X}$ – tiriamų daugiamačių vektorių aibė;

$F = \{X_1^{[F]}, X_2^{[F]}, ..., X_{N_{fiks}}^{[F]}\}$ – bazinių vektorių aibė;

$M = \{X_1^{[M]}, X_2^{[M]}, ..., X_{N_{nauji}}^{[M]}\}$ – likusių vektorių aibė;

$Y^{[F]} = \{Y_1^{[F]}, Y_2^{[F]}, ..., Y_{N_{fiks}}^{[F]}\}$ – bazinių vektorių projekcija;

$Y^{[M]} = \{Y_1^{[M]}, Y_2^{[F]}, ..., Y_{N_{nauji}}^{[M]}\}$ – likusių taškų projekcija;

$Y^{[F]} = SMACOF(F)$ – funkcija, pagal SMACOF algoritmą apskaičiuoja bazinių vektorių projekciją;

$Y_i^{[M]} = QuaziNewton(E_{santykinis\_DS}(F, Y^{[F]}, X_i^{[M]}, Y_i^{[M]}))$ – funkcija, kuri kvazi-Niutono algoritmu minimizuoja $E_{santykinis\_DS}$ paklaidą ir grąžina taško $X_i^{[M]}$ projekciją;

Funkcija $E_{santykinis\_DS}(F, Y^{[F]}, X_i^{[M]}, Y_i^{[M]})$ suskaičiuoja santykinių DS paklaidą, šiai funkcijai pateikiami parametrai: $F, Y^{[F]}, X_i^{[M]}, Y_i^{[M]}$.

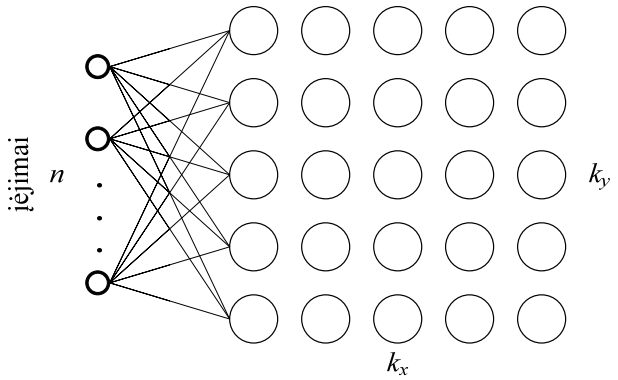
Nors šis algoritmas buvo pasiūlytas (Naud and Duch 2000), tačiau detalus šio algoritmo tyrimas, jo priklausomybė nuo bazinių vektorių parinkimo strategijos, bazinių vektorių skaičiaus nebuvo atliktas. 3 skyriuje bus pristatyta detali šio algoritmo analizė, pateikiamos išvados apie tinkamiausias šio algoritmo parametrų reikšmes.

2.4.4. SOM tinklo ir Sammono projekcijos integruotas junginys

Saviorganizuojantys neuroniniai tinklai (žemėlapiai) (angl. *self-organizing maps* (SOM)) (Kohonen 2001) dažnai yra vadinami Kohoneno neuroniniais tinklais arba Kohoneno saviorganizuojančiais žemėlapiais. Šio tipo neuroninių tinklų pavadinimas kylo iš jų savybės, kad saviorganizuojantis žemėlapis, naudodamas mokymo aibę, pats save sukuria (organizuoja). SOM tinklo tikslas yra išlaikyti duomenų topografiją: taškai, esantys arti vieni kitų įėjimo vektorių erdvėje, turi būti atvaizduojami arti vieni kitų ir SOM žemėlapyje. SOM žemėlapiai yra naudojami ir daugiamačiams duomenims klasterizuoti, ir juos vizualizuoti, t. y. rasti projekcijas mažesnės dimensijos erdvėje, įprastai plokštumoje.

Saviorganizuojantis neuroninis tinklas (SOM) yra neuronų $T = \{t_{ij}, i = 1, ..., k_x, j = 1, ..., k_y\}$, išdėstytų dvimačio tinklelio (lentelės) mazguose, masyvas. Dvimačio neuroninio tinklo schema pateikta 2.15 paveiksle. Kiekvienas žemėlapio neuronas sujungtas su kiekvienu įėjimo vektoriumi

(2.15 paveikslas, kad jo neperkrauti, pavaizduotos tik pirmos žemėlapio eilutės jungtys su įėjimo vektoriais). Čia k_x yra lentelės eilučių skaičius, k_y – stulpelių skaičius.



2.15 pav. *Dvimačio SOM tinklo schema*

SOM tinklo mokymo schema yra tokia: kiekvienas *n*-matis mokymo aibės vektorius $X \in \{X_1, X_2, ..., X_m\}$ mokymo metu yra susiejamas su vienu tinklo neuronu, kuris taip pat yra *n*-matis vektorius; mokymo pradžioje nustatomos vektorių-neuronų t_{ij}, $i = 1,...,k_x$, $j = 1,...,k_y$ komponenčių reikšmės, įprastai jos parenkamos atsitiktinai iš intervalo (0, 1); kiekviename mokymo žingsnyje vienas iš mokymo aibės vektorių *X* pateikiamas į tinklą; randama iki kurio neurono t_c nuo vektoriaus *X* Euklido atstumas yra mažiausias, vektorius t_c vadinamas neuronu nugalėtoju. Visų neuronų komponentės keičiamos pagal formulę:

$$t_{ij} \leftarrow t_{ij} + h_{ij}^c (X - t_{ij}),$$

čia, h_{ij}^c yra kaimynystės funkcija. Yra naudojamos įvairios šios funkcijos išraiškos. Viena iš jų, pateikta darbe (Dzemyda 2001): $h_{ij}^c = \dfrac{\alpha}{\alpha \eta_{ij}^c + 1}$, $\alpha = \max\left((e + 1 - \hat{e})/e; 0{,}01\right)$ (čia e yra mokymo epochų skaičius, $\hat{e}$ – vykdomos epochos numeris, viena mokymo epocha – tai mokymo proceso dalis, kai visus vektorius pateikiame tinklui po vieną kartą, ją sudaro *m* mokymo žingsnių). Dydis η_{ij}^c vadinamas kaimynystės tarp neuronų t_c ir t_{ij} eilė. Greta neurono-nugalėtojo esantys neuronai vadinami pirmos eilės kaimynais, greta pirmos eilės kaimynų esantys neuronai, išskyrus jau paminėtus, – antros eilės

kaimynais ir t.t. Kiekvienos epochos metu perskaičiuojami tie neuronai t_{ij}, kuriems galioja nelygybė:

$$\eta_{ij}^{c} \le \max[\alpha \max(k_x, k_y), 1].$$

Vienas iš tinklo apsimokymo kokybės įvertinimo kriterijų yra kvantavimo paklaida (angl. *quantization error*) $E_{SOM} = \frac{1}{m}\sum_{j=1}^{m} \| X_j - t_{c(j)} \|$. Tai vidutinis atstumas tarp kiekvieno *n*-mačio duomenų vektoriaus $X_j = (x_{j1}, x_{j2}, ..., x_{jn})$ ir jo vektoriaus-nugalėtojo $t_{c(j)}$, čia *m* – analizuojamų vektorių skaičius.

Sammono projekcija (Sammon 1969**)** yra vienas iš daugiamačių skalių grupės metodų. Tai netiesinis daugelio kintamųjų objektų atvaizdavimo mažesnių matmenų erdvėje metodas. Ieškoma daugiamačių vektorių projekcijos plokštumoje siekiant išlaikyti taškų tarpusavio atstumus. Sammono algoritmas minimizuoja projekcijos paklaidą

$$E_S = (\sum_{\substack{i,j=1 \\ i<j}}^{m} d_{ij}^{*})^{-1} \sum_{\substack{i,j=1 \\ i<j}}^{m} \frac{(d_{ij}^{*} - d_{ij})^2}{d_{ij}^{*}};\ E_S \text{ atitinka } E_{DS}, \text{ kai } w_{ij} = \frac{1}{d_{ij}^{*} \sum_{\substack{k,l=1 \\ k<l}}^{m} d_{kl}^{*}}.$$

Originaliame Sammono algoritme dvimačių vektorių koordinatės ieškomos pagal iteracinę formulę:

$$y_{ik}(m'+1) = y_{ik}(m') - \eta \frac{\frac{\partial E_S(m')}{\partial y_{ik}(m')}}{\left| \frac{\partial^2 E_S(m')}{\partial (y_{ik})^2 (m')} \right|},$$

čia m' yra iteracijos numeris, η – optimizavimo žingsnį įtakojantis parametras. Šiame darbe Sammono algoritmui taikyta Seidelio tipo pakoordinatinė paieška, analizuota darbe (Dzemyda *et al.* 2004), čia dvimačių vektorių koordinatės perskaičiuojamos atsižvelgiant ne tik į ankstesnėje iteracijoje gautas koordinates, bet ir skaičiuojamojoje iteracijoje jau gautas naujas koordinates.

SOM tinklo ir Sammono projekcijos integruotas junginys. Apmokant neuroninį tinklą apskaičiuojami SOM tinklo vektoriai (neuronai) ir nustatomi tuos vektorius atitinkančių analizuojamų objektų numeriai, t. y., objektai pasiskirsto tarp SOM tinklo elementų. Tai gali būti laikoma daugiamačių duomenų atvaizdavimu plokštumoje, nes įmanoma vizualiai stebėti objektų

tarpusavio išsidėstymą. Išskirtinė tokio atvaizdavimo savybė – duomenų sugrupavimas (surūšiavimas, klasterizavimas) pagal jų panašumą. Tačiau sunku pasakyti, kaip toli vienas nuo kito yra į gretimus tinklelio (lentelės) langelius pakliuvę objektai.

Kadangi SOM neuronai yra daugiamačiai vektoriai, todėl po tinklo mokymo gautus neuronus nugalėtojus galima analizuoti vienu iš DS tipo metodų, pavyzdžiui, Sammono algoritmu. Nuoseklus SOM tinklo ir Sammono algoritmo junginys tirtas darbuose (Dzemyda 2001), (Kaski 1997). Darbe (Dzemyda and Kurasova 2006) pateiktas ir detaliai išanalizuotas *integruotas junginys*, kai daugiamačiai duomenys analizuojami Sammono algoritmu atsižvelgiant į SOM tinklo mokymosi eigą: SOM tinklo mokymo procesas suskaidomas į pasirinktą skaičių blokų; po kiekvieno bloko gauti neuronai-nugalėtojai analizuojami Sammono algoritmu; jame pradinės dvimačių vektorių koordinatės imamos atsižvelgiant į prieš tai gautas dvimates koordinates.

Atlikta analizė darbe (Dzemyda and Kurasova 2006) parodė, kad integruotu junginiu gaunama tikslesnė daugiamačių duomenų projekcija plokštumoje Sammono paklaidos E_S prasme lyginant su nuosekliojo junginio rezultatais. Vietoj Sammono algoritmo sėkmingai gali būti naudojamas bet koks kitas daugiamačių skalių (DS) algoritmas.

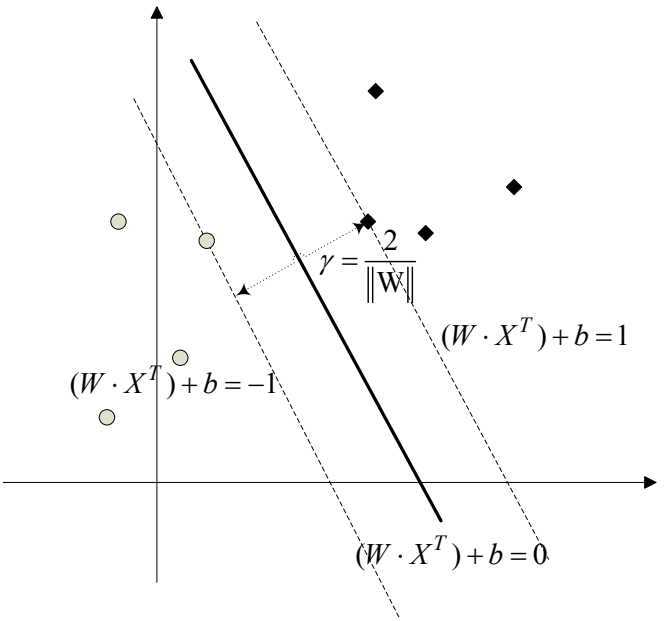
2.5. Duomenų gavybos metodų, naudojamų tyrimuose, apžvalga

2.5.1. Atraminių vektorių klasifikavimo algoritmas

Atraminių vektorių klasifikatorius (SVM) yra klasifikavimo su mokymu metodas, taikomas ir klasifikavime, ir regresinei analizei. Šio klasifikatoriaus iliustracija pateikta 2.16 paveiksle. Kai SVM naudojamas klasifikavimui, sukuriama hiperplokštuma, kuri duomenis atskiria į dvi klases (Cristianini and Shave-Taylor 2003). Tarkime turime mokymo duomenų aibę X_i, kur kiekvienas aibės objektas turi nustatytą klasę Z_i. Sudaromos tokios poros $(X_i, Z_i), i = 1,...,m,$ čia $X_i \in R^n$ ir $Z_i = \{1,-1\}$. Paprasčiausias atraminių vektorių klasifikatorius sukuria hiperplokštumą:

$$(W \cdot X^T) + b = 0,\ W \in R^n,\ b \in R,$$

atitinkančią tikslo funkciją $f(X) = \text{sgn}((W \cdot X^T) + b)$. Čia $W = (w_1, w_2, ..., w_n)$, $X = (x_1, x_2, ..., x_n)$ yra vektoriai eilutės, $W \cdot X^T = \sum_{i=1}^{n} w_i x_i$.



2.16 pav. *Atraminių vektorių klasifikatoriaus iliustracija*

Kuriant hiperplokštumą, mokymo aibės objektai suskirstomi į dvi dalis taip, kad atstumas tarp artimiausių elementų, priklausančių skirtingoms klasėms, iki tos hiperplokštumos būtų maksimalus.

Sukurta hiperplokštuma priklauso vien tik nuo mokymo aibės poaibio, sudaryto iš taip vadinamų atraminių vektorių. Konstruojant tinkamiausią hiperplokštumą, sprendžiamas toks optimizavimo su ribojimais uždavinys:

$$\min_{W,b} \frac{1}{2}\|W\|^2,$$

apribojimai $Z^i((W \cdot (X_i)^T) + b) \geq 1, \quad i = 1,...,m.$

Sudaroma Lagranžo forma:

$$L(W,b,\alpha) = \frac{1}{2}\| W \|^2 - \sum_{i=1}^{m} \alpha_i (Z_i((W \cdot (X_i)^T) + b) - 1).$$

Kaip parodyta (Chen, 2005), hiperplokštumos funkcija gali būti užrašyta taip: $f(X) = \text{sgn}(\sum_{i=1}^{m} Z_i \alpha_i (X \cdot (X_i)^T) + b)$. Vektoriai, kuriems $\alpha_i \neq 0$, yra vadinami atraminiais vektoriais. Maksimalus atstumas tarp vienos ir kitos klasių paviršių yra lygus $\gamma = \frac{2}{\|W\|}$.

Bendru atveju tikslo funkcija yra:

$$f(X) = \text{sgn}(\sum_{i=1}^{m} Z_i \alpha_i (\varphi(X) \cdot \varphi(X_i)^T + b)$$
$$= \text{sgn}(\sum_{i=1}^{m} Z_i \alpha_i k(X, X_i) + b).$$

čia $k(X, X_i)$ yra branduolio (angl. *kernel*) funkcija. Tai tokia funkcija, kai visiems $X_i, X_j \in \mathbf{X}$ $k(X_i, X_j) = (\varphi(X_i) \cdot \varphi(X_j)^T)$, čia φ yra aibės X atvaizdavimas į aibę *F*. Dažnai duomenys negali būti tiesiškai atskirti aibėje X, tačiau tai įmanoma aibėje *F*.

Branduolio funkcijos, naudojamos atraminių vektorių klasifikavimo modelyje:

tiesinė: $k(X_i, X_j) = (X_i \cdot (X_j)^T)$,

polinominė: $k(X_i, X_j) = \left(\gamma(X_i \cdot (X_j)^T) + r\right)^d$,

radialinių bazinių funkcijų (RBF): $k(X_i, X_j) = \exp\left(-\frac{\| X_i - X_j \|}{\gamma}\right)$,

sigmoidinė: $k(X_i, X_j) = \tanh\left(\gamma(X_i, (X_j)^T) + r\right)$.

Čia γ, r, d yra branduolio funkcijos parametrai. RBF yra dažniausiai naudojama branduolio funkcija atraminių vektorių klasifikatoriuje. Ji taip pat bus naudojama šio darbo eksperimentuose.

Optimizavimo uždavinys pertvarkomas į formą:

$$\max_{\alpha \in R^m} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j Z_i Z_j k(X_i, X_j),$$

apribojimai $\alpha_i \geq 0,\ i = 1,..,m$ ir $\sum_{i=1}^{m} \alpha_i Z_i = 0.$

Praktiškai, skiriamoji hiperplokštuma gali neegzistuoti, pavyzdžiui, kai klasės labai persidengia. Tada įvedamas parametras ir optimizavimo uždavinys tampa:

$$\min_{\xi, W, b} \left(W \cdot W^T + C \sum_{i=1}^{m} \xi_i \right),$$

apribojimai $Z_i (W^i \cdot (X_i)^T + b) \geq 1 - \xi_i,\ i = 1,...,m,\quad \xi_i \geq 0, i = 1,...,m.$

2.5.2. Paprastasis Bayeso klasifikatorius

Paprastasis Bayeso klasifikatorius yra pagrįstas Bayeso tikimybių taisykle. Tarkime, kad objektas X_i turi p nepriklausomų parametrų $\{x_1^i, x_2^i, ..., x_p^i\}$. Žinant tikimybes $P(x_k^i \mid C_j)$ kiekvienai klasei C_j ir parametrui x_k^i, mes galime įvertinti $P(X_i \mid C_j)$ tokiu būdu: $P(X^i \mid C_j) = \prod_{k=1}^{p} P(x_k^i \mid C_j)$. Mums reikia turėti apriorines tikimybes $P(C_j)$ kiekvienai klasei ir sąlyginę tikimybę $P(X_i \mid C_j)$. Kad suskaičiuotume $P(X_i)$, galime įvertinti su kokia tikimybe X_i priklauso kiekvienai klasei. Tikimybė, kad X_i priklauso klasei yra lygi sąlyginių kiekvieno parametro tikimybių sandaugai. Randama aposteriorinė tikimybė $P(C_j \mid X_i)$ kiekvienai klasei. Objektas priskiriamas klasei su didžiausia tikimybe.

Šis metodas neveikia turint tolygiai pasiskirsčiusius duomenis, tačiau problema sprendžiama juos išskaidžius į tam tikrus intervalus. Detalesnis šio algoritmo aprašymas pateikiamas (Ramoni and Sebastiani 2003), (Han and Kamber 2006).

2.5.3. k artimiausių kaimynų metodas (kNN)

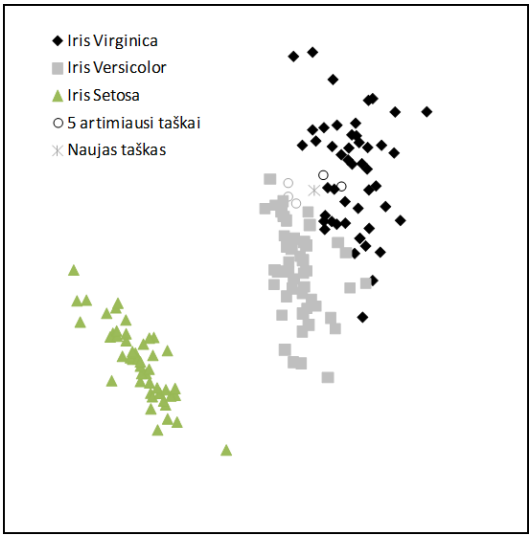
k artimiausių kaimynų metodas (kNN) yra klasifikavimo be mokymo metodas. Norint naują objektą priskirti kuriai nors klasei, yra skaičiuojami atstumai nuo to objekto iki visų mokymo aibės objektų.

Dažniausiai naudojamas Euklidinis atstumas, tačiau gali būti naudojamas ir miesto kvartalo (angl. *city-block, Manhattan*) atstumas, Čebyševo atstumas ar pan. (Dzemyda *et at.* 2008). Naujas objektas priskiriamas tai klasei, kuriai priklauso dauguma iš artimiausių *k* jo kaimynų.

Klasifikavimo tikslumas labai priklauso nuo kaimynų skaičiaus *k* parinkimo, kuris parenkamas eksperimentiškai. Didesnei mokymo aibei, didesnis turi būti pasirinktas kaimynų skaičius *k*. Daugiau informacijos apie šį metodą galima rasti (Han and Kamber 2006).

2.17 paveiksle pateikta *k* artimiausių kaimynų metodo iliustracija. Naujo taško klasės priskyrimas atliekamas atsižvelgiant į 5 artimiausius daugiamatėje erdvėje taškus. Paveiksle pateikta Iris duomenų projekcija, gauta daugiamačių skalių algoritmu, bei parodyti naujam taškui artimiausi kaimynai daugiamatėje erdvėje.

Projektuojant daugiamačius duomenis iškraipymai tarp duomenų neišvengiami. Todėl artimi taškai daugiamatėje erdvėje nebūtinai išlieka artimais ir projektuojamoje erdvėje. 2.17 paveikslas iliustruoja būtent šią situaciją.



2.17 pav. *Klasifikavimo rezultatai gauti k artimiausių kaimynų metodu: 5 kaimynai; iris duomenų aibė*

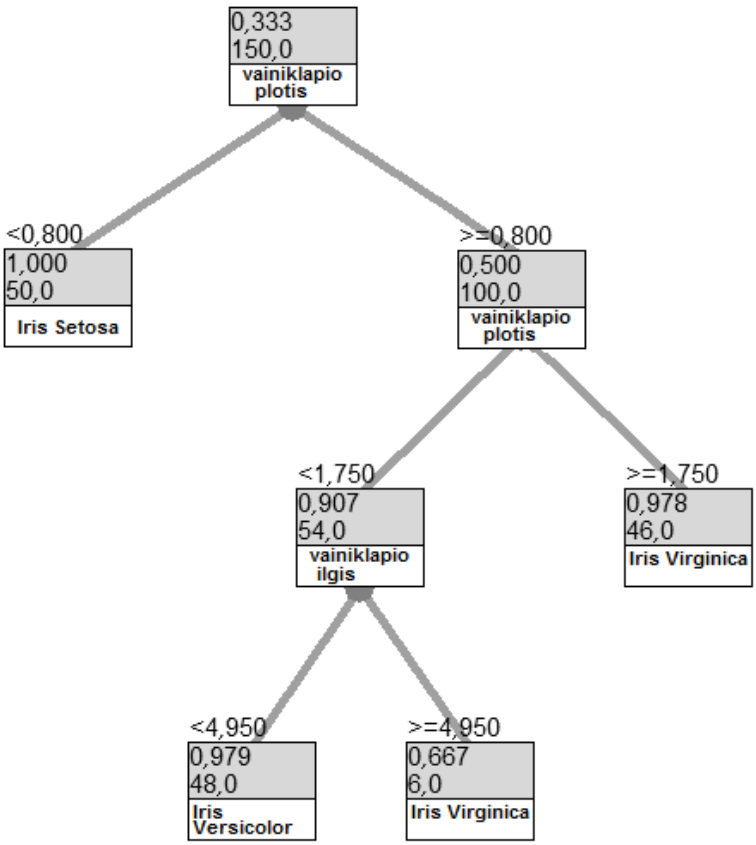
Daugiamatėje erdvėje 5 artimiausi naujam vektoriui kaimynai priklauso dviems skirtingoms klasėms: 2 kaimynai iš Iris Virginica klasės ir 3 kaimynai iš Iris Versicolor klasės. Naujas taškas priskirtas Iris Versicolor klasei, kadangi 3 artimiausi taškai iš 5 priklauso tai klasei.

2.5.4. Klasifikavimo medis

Klasifikavimo medžio metodu yra sukuriamas medis siekiant modeliuoti klasifikavimo procesą. Klasifikavimo medžių metodai yra tinkami tada, kai duomenų analizės (klasifikavimo arba prognozavimo) tikslas – sukurti taisykles, kurios gali būti lengvai suprantamos ir paaiškinamos.

Klasifikavimo medžio iliustracija pateikta 2.18 paveiksle. Šio metodo esmė padalinti tiriamą duomenų erdvę į hiperstačiakampius poerdvius, kurie yra suformuojami klasifikavimo medžio taisyklių. Virš blokų esančios nelygybės nurodo parametro, nurodyto bloko viduje, slenkstį, pagal kurį objektai priskiriami tam tikrai klasei. Slenkstis nustatomas pagal Relief kriterijų, detaliau apie jį (Kononenko 1994). Bloko viduje esantis pirmas skaičius nurodo tikimybę, su kuria objektai priklauso nurodytai daugumos objektų klasei. Antras skaičius nurodo daugumos objektų klasei priklausančių objektų skaičių.

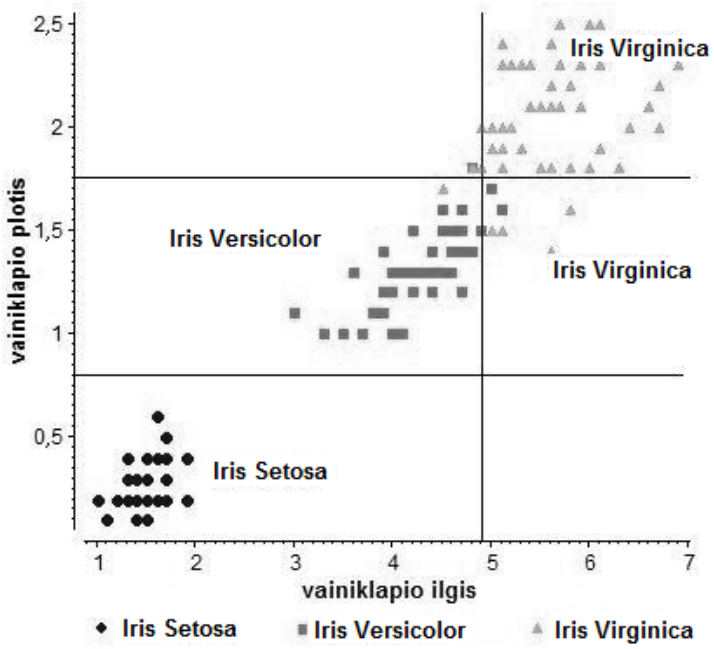
Kiekvienas klasifikavimo medžio lapas, kurį atitinka klasifikavimo medžio šakos paskutinis blokas, žymi klasę, kuriai priskiriami klasifikuojami duomenys.



2.18 pav. *Klasifikavimo medis; iris duomenų aibė*

Remiantis 2.18 paveiksle pateiktu klasifikavimo medžiu sukuriamos keturios klasifikavimo taisyklės, kurios suskirsto visus tiriamus duomenis į nepersidengiančias sritis:

1. Jei $vainiklapio\ plotis < 0,8$, tai $klasė = Iris\ Setosa$.
2. Jei $vainiklapio\ plotis \in [0,8;1,75)$ ir $vainiklapio\ ilgis \geq 4,95$, tai $klasė = Iris\ Virginica$.
3. Jei $vainiklapio\ ilgis < 4,95$ ir $vainiklapio\ plotis \in [0,8;1,75)$, tai $klasė = Iris\ Versicolor$.
4. Jei $vainiklapio\ plotis \geq 1,75$, tai $klasė = Iris\ Virginica$.



2.19 pav. *Iris duomenų taškinis grafikas su klasifikavimo medžio taisyklėmis apibrėžtomis sritimis*

Iris duomenų taškinis grafikas su klasifikavimo medžio taisyklėmis apibrėžtomis sritimis pateiktas 2.19 paveiksle. Kiekvienas naujas klasifikuojamas taškas priskiriamas tokiai klasei, priklausomai į kokią sritį tas taškas papuolė.

Detaliau šis metodas aprašytas knygose (Dunham 2003), (Han and Kamber 2006).

2.5.5. Klasifikavimo tikslumo vertinimo matai

Save mokančiose sistemose ir žinių gavyboje klasifikavimo kokybės įvertinimui naudojami įvairūs matai. Dažniausiai naudojamas matas yra bendras *klasifikavimo tikslumas* (angl. *clasification accuracy*), tačiau sprendžiant realius uždavinius dažnai jo nepakanka, kadangi reikia žinoti klasifikavimo tikslumą kiekvienai klasei atskirai. Medicininių duomenų analizėje dar naudojami vertinimo matai – *jautrumas* (angl. *sensitivity*) ir *specifiškumas* (angl. *specificity*).

Dviejų klasių atveju klasifikuojami objektai gali priklausyti klasei (tokius objektus vadinsime *teigiamais*) ir nepriklausyti klasei (neigiami objektai). Taip pat kiekvienas objektas klasifikatoriaus yra priskiriamas vienai iš klasių. Rezultatas taip pat gali būti arba teigiamas arba neigiamas (objektas

klasifikatoriaus priskirtas nurodytai klasei arba ne). Teigiamas objektas ir klasifikatoriaus priskirtas nurodytai klasei, jis vadinamas *tikrai teigiamu* (TT). Jeigu neigiamas objektas klasifikatoriaus nepriskirtas nurodytai klasei, jis vadinamas *tikrai neigiamu* (TN). Teigiamas objektas, klasifikatoriaus priskirtas klaidingai klasei, vadinamas *klaidingai neigiamu* (KN), o neigiamas objektas, klasifikatoriaus priskirtas tiriamai klasei, vadinamas *klaidingai teigiamu* (KT).

Tikrai teigiamas (TT): objektas X^i priskirtas klasei C_j ir iš tiesų jis jai priklauso;

Klaidingai teigiamas (KT): objektas X^i priskirtas klasei C_j, bet iš tiesų jis jai nepriklauso;

Tikrai neigiamas (TN): objektas X^i nepriskirtas klasei C_j ir iš tiesų jis jai nepriklauso;

Klaidingai neigiamas (KN): objektas X^i nepriskirtas klasei C_j, bet iš tiesų jis jai priklauso.

Tuomet klasifikavimo tikslumo įverčiai apibrėžiami pagal formules:

$$specifiškumas = \frac{\text{TN skaičius}}{\text{TN skaičius} + \text{KT skaičius}},$$

$$jautrumas = \frac{\text{TT skaičius}}{\text{TT skaičius} + \text{KN skaičius}},$$

$$bendras\ klasifikavimo\ tikslumas = \frac{\text{TT skaičius} + \text{TN skaičius}}{\text{visų objektų skaičius}}.$$

Pateiktų tikslumo matų prasmę iliustruosime medicininių duomenų klasifikavimo pavyzdžiu. Tarkime, kad grupei žmonių atliekamas tyrimas tam tikros ligos nustatymui. Keli žmonės iš šios grupės tikrai serga šia liga ir atliktas tyrimas rodo, kad šie žmonės serga. Tokius žmones vadinsime tikrai teigiamais (TT). Kita grupelė tiriamųjų serga šia liga, tačiau tyrimas rodo, kad jie neserga. Juos vadinsime klaidingai neigiamais (KN). Keli žmonės iš šios grupės neserga šia liga ir atliktas tyrimas rodo, kad šie žmonės neserga – juos vadinsime tikrai neigiamais (TN). Galiausiai, yra žmonių kurie neserga nurodyta liga, tačiau tyrimas parodė, kad jie serga. Tokius tiriamuosius vadinsime klaidingai teigiamais (KT). Visų tikrai teigiamų, klaidingai neigiamų, tikrai neigiamų ir klaidingai teigiamų tiriamųjų skaičius sudaro 100 % visų tiriamųjų aibės.

Jautrumo matas parodo santykį sergančiųjų, kurių liga patvirtinta tam tikru diagnostikos metodu arba tyrimu (tiktai teigiami tiriamieji), su visų sergančiųjų skaičiumi; šis matas rodo tikimybę, kad sergančio žmogaus tyrimo duomenys patvirtina ligą. Kuo didesnis jautrumo matas, tuo mažesnė tikimybė, kad tikrai sergančiam žmogui liga nebus diagnozuota. Specifiškumo matas parodo santykį

tarp tikrai neigiamų (neserga šia liga ir atliktas tyrimas rodo, kad neserga) ir visų nesergančių žmonių. Panašiai kaip ir jautrumo matas rodo tikimybę, kad sveiko žmogaus tyrimo duomenys patvirtina, jog jis yra sveikas. Kuo didesnė specifiškumo mato reikšmė, tuo mažiau sveikų žmonių yra priskirtų sergantiems. Detaliau apie šiuos klasifikavimo kokybės įvertinimo matus aprašyta (Han and Kamber 2006).

Šioje disertacijoje, klasifikavimo kokybės matai apskaičiuoti kryžminio tikrinimo strategiją, testavimo aibę sudaro 10 % mokymo aibės vektorių. Eksperimentuose naudota „Orange“ (Demsar *et al.* 2004) programinė įranga.

2.5.6. K-vidurkių klasterizavimo metodas

Iš klasterizavimo metodų grupės dažniausiai naudojamas *K*-vidurkių klasterizavimo metodas (angl. *k-means*) (MacQueen 1967), (Vesanto 2001). Šį klasterizavimo metodą galima laikyti ir kvadratinės paklaidos algoritmu (angl. *squared error clustering algorithm*) (Dunham 2003), nes jis minimizuoja kvadratinę paklaidą. Tegu klasteriui K_i priskirta objektų aibė $\{X_i^1, X_i^2, ..., X_i^{\mu_i}\}$, $i \in (1,...,\kappa)$, μ_i – objektų klasteryje K_i skaičius, $X_i^j = (x_{i1}^j, x_{i2}^j, ..., x_{in}^j)$, $j = (1,...,\mu_i)$. Tada kvadratinė paklaida vienam klasteriui K_i yra Euklido atstumų tarp kiekvieno klasterio elemento ir klasterio centro C_i kvadratų suma (2.6).

$$E_{K_i} = \sum_{j=1}^{\mu_i} \left\| X_i^j - C_i \right\|^2, \tag{2.6}$$

čia $C_i = (c_{i1}, c_{i2}, ..., c_{in})$ klasterio centras, kurio komponentės randamos pagal formulę $c_{ik} = \frac{\sum_{j=1}^{\mu_i} (x_{ik}^j)}{\mu_i}$, ($k = 1,...,n$).

Kvadratinė paklaida klasterių aibei $K = \{K_1, K_2, ..., K_\kappa\}$ apskaičiuojama pagal (2.7) formulę

$$E_K = \sum_{i=1}^{\kappa} E_{K_i}. \tag{2.7}$$

Vieno iš galimų funkcijos minimizavimo algoritmų (K-vidurkių) struktūra galėtų būti tokia:

1. Inicializuojami κ klasterių centrai C_i, $(i = 1,...,\kappa)$;

2. Kiekvienas analizuojamų duomenų aibės objektas priskiriamas tam klasteriui, iki kurio centro atstumas yra mažiausias;
3. Perskaičiuojami kiekvieno klasterio centrai;
4. Skaičiuojama kvadratinė paklaida pagal (2.7) formulę;
5. 2 – 4 punktai kartojami, kol kvadratinės paklaidos reikšmė tampa mažesnė už pasirinktą slenkstinę reikšmę arba objektai nebepersiskirsto kitiems klasteriams.

K-vidurkių algoritmo trūkumas tas, kad randamas kvadratinės paklaidos lokalus, o ne globalus minimumas; metodas pakankamai lėtai konverguoja; veikia tik su metriniais duomenimis; rezultatui didelę įtaką daro taškai-atsiskyrėliai; būtina algoritmą vykdyti kelis kartus pradedant su skirtingais klasterių centrais; būtina žinoti klasterių skaičių, priešingu atveju, algoritmą reiktų vykdyti su skirtingomis κ reikšmėmis.

Yra sukurtos *K*-vidurkių metodo modifikacijos: EM algoritmas (angl. *expectation maximization*), pasiūlytas darbe (Dempster *et at.* 1977), ISODATA algoritmas, pasiūlytas darbe (Ball and Hall 1965). Kai kurie autoriai šiuos praplėstus algoritmus priskiria prie modeliais pagrįstų klasterizavimo metodų.

2.6. Antrojo skyriaus apibendrinimas ir išvados

Šiame skyriuje yra atlikta duomenų gavybos sprendžiamų uždavinių ir metodų šiems uždaviniams spręsti analitinė apžvalga. Susisteminti bei išnagrinėti vizualizavimo metodai, kurie grindžiami skirtingomis idėjomis, yra universalūs ar orientuoti specifiniams duomenims. Taip pat apžvelgti duomenų gavybos metodai, kurie gali būti jungiami su vizualizavimo metodais daugiamačių duomenų analizei.

Išanalizuotas duomenų gavybos ir analizės procesas, apžvelgti ir palyginti keturi šio proceso modeliai. Parodyta, kad pagrindinis jų skirtumas yra detalizacijos lygmenyje. Apibendrinus gauname šešių žingsnių schemą, kuri apima visus žinių radimo etapus, įvardinama aiški vizualizavimo vieta kiekviename žinių radimo etape.

Analizė parodė, kad vizualizavimas užima svarbią vietą duomenų gavybos ir analizės procese, tačiau jo panaudojimas yra gana fragmentiškas. Čia yra reikalingas kompleksiškas požiūris apimantis visus analizės proceso etapus.

3

Vizualios žinių gavybos galimybių didinimas

Šiame skyriuje pateikiama vizualios žinių gavybos metodologija, santykinių daugiamačių skalių metodo efektyvumo gerinimo būdai bei daugiamačių duomenų tarpusavio atstumų koregavimo algoritmas, naudojamas vizualizuojant daugiamačius duomenis.

Pagrindiniai skyriaus rezultatai paskelbti šiuose straipsniuose: (Bernatavičienė *et at.* 2005b), (Bernatavičienė *et at.* 2006c), (Bernatavičienė *et at.* 2006d), (Bernatavičienė *et at.* 2007c).

3.1. Vizualios žinių gavybos iš daugiamačių duomenų metodologija

Daugiamačių duomenų suvokimas yra rezultatas ilgo ir sudėtingo žinių gavybos proceso, kuris apima daug etapų:

- suformuojami analizės tikslai ir uždaviniai, iškeliamos preliminarios hipotezės, kurias tyrimo rezultatai turėtų patvirtinti arba paneigti;
- formuojama duomenų imtis, pasirenkami analizei tinkami duomenų gavybos metodai;

- analizuojami tyrimo metu gauti rezultatai;
- gauta informacija apibendrinama, priimamos arba atmetamos tyrimo pradžioje iškeltos hipotezės.

Šis procesas – tai perėjimas nuo didelės analizuojamos duomenų aibės prie specifinių duomenų, iš kurių išskiriama informacija bei suformuojamos žinios apie tiriamų duomenų struktūrą, naujus sąryšius, duomenų grupes, kas turės įtaką tolimesnių sprendimų priėmimui.

Vizualizavimas yra pažinimo įrankis, kuriuo tyrėjas gali naudotis visame žinių gavimo procese. Naudojant duomenų gavybos metodus, vizualizavimas tyrėjui daro žinių radimo procesą daug aiškesniu ir suprantamesniu. Tam padeda šios galimybės:

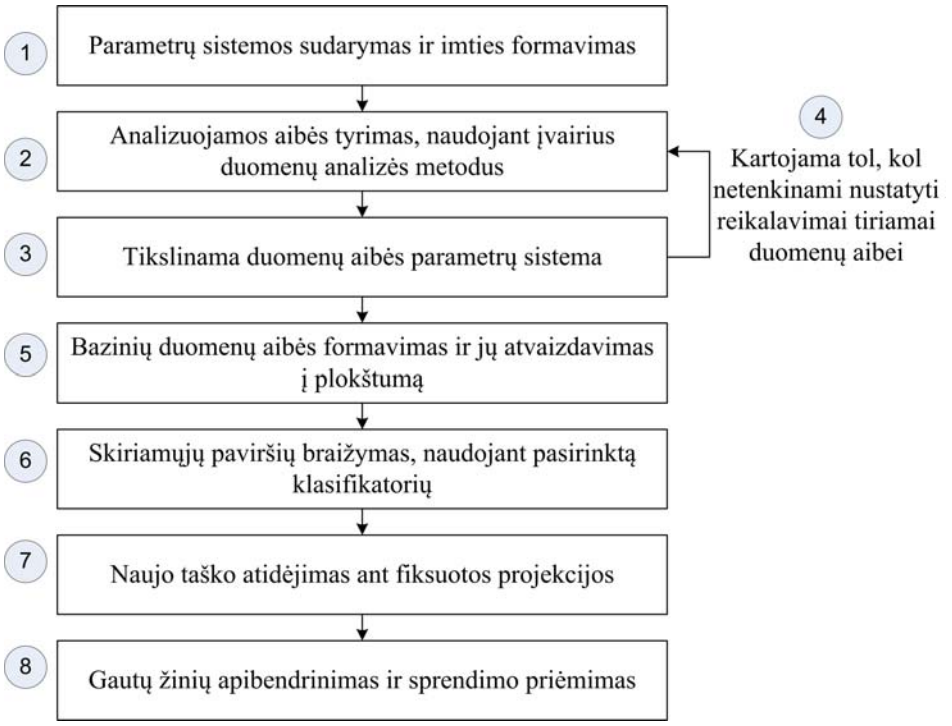
- vizuali pradinės duomenų aibės analizė;
- rezultatų, gautų pasirinktais duomenų gavybos metodais, vizualizavimas ir interpretavimas;
- gauto automatizuoto sprendimo vizualus pateikimas, parodant jo vietą tarp esamų sprendimų;
- vizualiai pateikiami gauti rezultatai ekspertui, kuris gautas žinias lygina su anksčiau turėtomis, jas interpretuoja, nusprendžia ar gautus rezultatus atmesti ar gautais rezultatais papildyti žinių banką.

Taigi, įvertinus visus duomenų gavybos ir analizės proceso etapus, nustatyta, kad fragmentiškas vizualizavimo panaudojimas neišnaudoja visų analizės galimybių, kurias gali suteikti vizualizavimas. Jį reikia integruoti į visus duomenų gavybos ir analizės proceso etapus.

Šiame darbe pasiūlyta žinių gavimo vizualiais metodais metodologija, kuria remiantis atliekama išsami vizuali duomenų analizė. 3.1 paveiksle pateikta siūloma žinių gavybos vizualiais metodais proceso schema:

1. Sudaroma analizuojamos aibės parametrų sistema, naudojant visus matuotus parametrus šiai duomenų imčiai.
2. Atliekama detali duomenų aibės analizė, taikant įvairius duomenų analizės metodus, duomenų transformacijas (šiuose tyrimuose naudojami klasifikavimo ir vizualizavimo metodai).
3. Tikslinama analizuojamos duomenų aibės parametrų sistema: atmetami neesminiai parametrai, pridedami informatyvesni parametrai.
4. Antras ir trečias žingsniai kartojami tol, kol analizuojamos duomenų aibės klasifikavimo ir vizualizavimo rezultatai netenkina nustatyto tikslumo.

5. Remiantis gautais rezultatais suformuojama bazinių duomenų aibė (bazinių duomenų klasės yra žinomos: klasifikatoriai juos priskyrė toms klasėms, o vizuali analizė tai patvirtino).
6. Baziniai duomenys (vektoriai) projektuojami į plokštumą ir braižomi klasių skiriamieji paviršiai naudojant vieną iš klasifikavimo metodų.
7. Nauji taškai, kurių klasė dar nenustatyta arba norima ją patikslinti, projektuojami į plokštumą atsižvelgiant į fiksuotą bazinių vektorių projekciją.
8. Priklausomai nuo taško, atitinkančio naujus duomenis, padėties tarp skiriamųjų paviršių daromas preliminarus sprendimas apie jo priklausomybę vienai ar kitai klasei.



3.1 pav. *Žinių gavybos vizualiais metodais proceso schema*

Žinių gavybos vizualiais metodais proceso susisteminimas leido visapusiškai įvertinti ir pritaikyti vizualizavimo metodų ir priemonių teikiamas galimybes duomenų analizės efektyvumui didinti.

Siūlomoje metodologijoje, kuri schematiškai pateikta 3.1 paveiksle, vizualizavimas yra integruotas į visus duomenų gavybos ir analizės proceso etapus.

Šios siūlomos schemos efektyvumas parodytas taikant ją preliminariai medicininei diagnozei. Tai detaliai pateikta 4 skyriuje, kur visi vizualizavimo proceso etapai betarpiškai taikomi atliekant fiziologinių duomenų analizę.

Toliau šiame skyriuje bus pateikti nauji moksliniai rezultatai, kurie buvo gauti disertaciniame darbe. Jie susiję su žinių gavybos vizualiais metodais proceso 2, 6, 7 punktais.

3.2. Santykinių daugiamačių skalių metodo efektyvumo gerinimas

Didelių aibių vizuali analizė yra iki šiol labai aktuali problema. Vizualizuojant dideles daugiamačių duomenų aibes standartiniu daugiamačių skalių algoritmu sugaištama daug skaičiavimo laiko, o kartais dėl operatyvinės atminties stokos vizualizuoti dideles duomenų aibes tampa neįmanoma. Santykinių daugiamačių skalių algoritmas (santykinių DS alg.) sėkmingai gali būti taikomas didelių aibių vizualizavimui išvengiant minėtų problemų. Šio algoritmo idėjas galima rasti straipsnyje (Žilinskas 1993), kuomet vizualizuojamas daugelio kintamųjų funkcijų optimizavimo procesas.

Santykinių DS algoritmas sudarytas iš dviejų etapų: suformuojama bazinių vektorių aibė, kuri standartiniu DS metodu suprojektuojama į plokštumą, likusieji aibės taškai po vieną projektuojami į plokštumą atsižvelgiant tik į fiksuotą bazinių vektorių projekciją.

Santykinių daugiamačių skalių algoritmas detaliai aprašytas 2.4.3. skyriuje. Nors šis algoritmas buvo pasiūlytas (Naud and Duch 2000), tačiau detalus šio algoritmo tyrimas nebuvo atliktas. Šiame skyriuje pateikiama detali šio algoritmo analizė, pateikiamos išvados apie tinkamiausias šio algoritmo parametrų reikšmes. Tyrimo metu nustatyta, kad vizualizavimo rezultatus, gautus santykinių daugiamačių skalių algoritmu įtakoja:

1. Bazinių vektorių parinkimo strategija.
2. Dvimačių vektorių inicializavimo būdas.
3. Bazinių vektorių skaičius.

3.2.1. Duomenys tyrimams

Tyrimams buvo naudotos penkios duomenų aibės, kurių dimensija kinta nuo n=3 iki n=50; skirtingas duomenų skaičius aibėse; yra žinoma struktūra ir kurios yra dažnai naudojamos duomenų analizės metodams tirti. Keturios duomenų aibės dirbtinai sugeneruotos ir viena reali:

1. *Ellipsoidal* [*m*; *n*] duomenų aibė, čia *m*=1115 yra vektorių skaičius, *n*=50 yra dimensija; duomenų aibės vektoriai suformuoja 20 persidengiančių elipsoidinio tipo klasterių.
2. *Ellipsoidal* [*m*; *n*] duomenų aibė, čia *m*=3140, *n*=50; duomenų aibės vektoriai suformuoja 10 nepersidengiančių elipsoidinio tipo klasterių.
3. *Gaussian* [*m*; *n*] duomenų aibė, čia *m*=2729, *n*=10; duomenų aibės vektoriai suformuoja 10 persidengiančių klasterių.
4. *Parraboloid* [*m*; *n*] duomenų aibė, čia *m*=2583, *n*=3; duomenų aibės vektoriai suformuoja du nepersidengiančius klasterius.
5. *Abalone* [*m*; *n*] duomenų aibė, čia *m*=4177, *n*=7, kurią sudaro 29 klasteriai.

1 ir 2 duomenų aibės sugeneruotos naudojant elipsoidinių klasterių generatorių (Handl and Knowles 2005). Šis generatorius sukuria elipsoidinius klasterius. Klasterių ribos apibrėžiamos keturiais parametrais:

- centras;
- tarpžidininis atstumas, kurio reikšmės tolygiai pasiskirstę intervale [1,0; 3,0];
- pagrindinės ašies kryptis tolygiai keičiama generuojant kiekvieną atskirą klasterį;
- maksimali atstumų nuo sugeneruoto taško iki dviejų židinių sumos reikšmė, priklausanti intervalui [1,05; 1,15].

Generuojami taškai kiekvienam klasteriui atskirai, tikrinama ar neperžengtos elipsoidui apibrėžtos ribos ir netinkami taškai atmetami. Naudojant tokį generatorių buvo sugeneruotos *Ellipsoidal* [1115; 50] ir *Ellipsoidal* [3140; 50] duomenų aibės. Jas galima rasti internete adresu http://dbkgroup.org/handl/generators/.

Gaussian [2729; 10] duomenų aibė yra generuota naudojant Gauso generatorių. Detalus šio generatoriaus aprašymas pateiktas (Handl and Knowles 2005).

Parraboloid [2583; 3] duomenų aibė sudaryta iš dviejų klasių taškų, kurie generuoti tokiu būdu: pirmos dvi koordinatės x_1, x_2 generuojamos atsitiktinai iš anksto apibrėžtoje srityje (pirmai klasei ši sritis yra skritulys, kurio spindulys yra lygus 0,4; antrai klasei ši sritis yra žiedas, kurio ribos apibrėžiamos dviem apskritimais, kurių spinduliai 0,7 ir 1,2). Trečia koordinatė pridedama naudojant taisyklę $x_3 = 1{,}8 \cdot \sqrt{x_1^2 + x_2^2}$. Sukurtas paraboloidas yra pasukamas.

Abalone [4177; 7] duomenų aibė paimta iš duomenų saugyklos „UCI repository“ (Asuncion and Newman 2007). Kiekvienas vektorius sudarytas iš 7 moliuskų parametrų:

- x_1 – ilgis (ilgiausia kiauto dalis);
- x_2 – skersmuo (statmenas ilgiui);
- x_3 – kiauto aukštis;
- x_4 – moliusko svoris kartu su kiautu;
- x_5 – moliusko svoris be kiauto;
- x_6 – vidaus organų svoris,
- x_7 – kiauto svoris be moliusko;

Moliusko žiedų skaičius nusako klasę. Duomenų aibės vektoriai tarpusavyje persidengia. Kadangi parametrų matavimų skalės skirtingos, duomenys buvo normuoti: suskaičiuoti parametro reikšmių vidurkis $\overline{x_j}$ ir dispersija σ_j^2; kiekvieno parametro reikšmė x_{ij} normuota naudojantis: $(x_{ij} - \overline{x_j})/\sigma_j$.

3.2.2. Bazinių vektorių parinkimo strategijos

Tiriant santykinių DS algoritmą, nustatyta, kad bazinių vektorių parinkimo strategija įtakoja vizualizavimo rezultatų kokybę. Esant mažoms duomenų aibėms baziniais vektoriais laikoma visa turima aibė, o nauji aibės taškai dedami ant gautos bazinių vektorių projekcijos.

Tačiau, kai analizuojamos aibės yra labai didelės, bazinių vektorių parinkimo strategija įtakoja vizualizavimo rezultatų kokybę, nes bazinių vektorių aibę sudaro tik dalis analizuojamos aibės taškų. Eksperimentuose buvo naudota keletas bazinių vektorių parinkimo strategijų:

I. Bazinių vektorių aibė F sudaryta iš klasterių centrų, gautų k-vidurkių klasterizavimo algoritmu suklasterizavus analizuojamus duomenis. Ši strategija buvo pasiūlyta (Naud 2004), (Naud 2006);

II. Suklasterizavus duomenų aibę k-vidurkių metodu, baziniais taškais pasirenkami klasterių centrams artimiausi duomenų aibės taškai ir nustatytas fiksuotas taškų skaičius iš kiekvieno klasterio (Bernatavičienė *et at.* 2006d);

III. Bazinių vektorių aibė F sudaryta iš taškų, atsitiktinai išrinktų iš analizuojamos aibės (X_i, $i = 1,...,m$) (Bernatavičienė *et at.* 2006d).

Naudojant I strategiją bazinių vektorių aibė suformuota tokiu būdu: analizuojamų duomenų aibė suklasterizuota į fiksuotą skaičių klasterių ir

suformuotų klasterių centrai sudaro bazinių vektorių aibę. Visa analizuojamų duomenų aibė vizualizuojama dviem etapais:

1. Bazinių vektorių aibė projektuojama į plokštumą naudojant standartinį DS algoritmą;
2. Likusieji vektoriai projektuojami į plokštumą naudojant santykinių DS algoritmą.

Kadangi klasterizavimo rezultatai labai priklauso, nuo pradinių klasterių centrų inicijavimo, tai eksperimentas buvo kartojamas 100 kartų su skirtingomis pradinėmis klasterių centrų reikšmėmis, gauti rezultatai vidurkinami. Tokiu būdu buvo suformuotos bazinių vektorių aibės, kai fiksuotas bazinių vektorių skaičius lygus $N_{fiks} = 100, 200, ..., 800$. Naudojant II strategiją, duomenys buvo klasterizuojami į 10,20,...,80 klasterių. Kiekvieno klasterio centras buvo pakeičiamas analizuojamos aibės artimiausiu vektoriumi. Bazinė vektorių aibė F buvo suformuota imant iš kiekvieno klasterio po tašką, artimiausią klasterio centrui, ir dar po 9 papildomus taškus iš kiekvieno klasterio. Jei klasterį sudaro mažiau nei 10 taškų, tai imami visi klasterio taškai. Tokiu būdu bazinių vektorių aibės skaičius N_{fiks} beveik visada lygus 100,200,..., 800. Klasterizavimas su skirtingu fiksuotu klasterių skaičiumi, buvo atliekamas 100 kartų. Kiekvieną kartą suformuojant vis kitą bazinių vektorių rinkinį. Duomenys vizualizuojami, skaičiuojami gautų rezultatų vidurkiai.

Naudojant III strategiją, bazinių vektorių aibė buvo formuojama atsitiktinai išrenkant N_{fiks} skaičių vektorių iš analizuojamos taškų aibės. Eksperimentas kartojamas 100 kartų, gauti rezultatai vidurkinami. Atlikti eksperimentai, kai bazinių vektorių skaičius lygus $N_{fiks} = 100, 200, ..., 800$. Gautų vizualizavimo rezultatų (kai gaunama visos aibės projekcija) palyginimui buvo skaičiuojama projekcijos paklaida:

$$E = \sqrt{\sum_{i<j}^{m} (d_{ij}^{*} - d_{ij})^2 \Big/ \sum_{i<j}^{m} (d_{ij}^{*})^2}. \qquad (3.1)$$

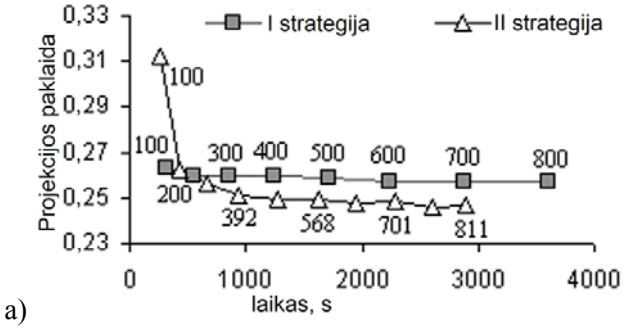
Standartinis DS algoritmas (SMACOF) minimizuoja paklaidą E_{DS}

$$E_{DS} = \sum_{i<j}^{m} (d_{ij}^{*} - d_{ij})^2, \qquad (3.2)$$

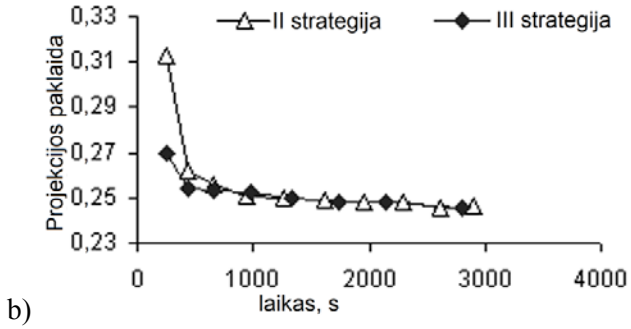
o santykinių DS algoritmas minimizuoją paklaidą $E_{santykinis\_DS}$:

$$E_{santykinis\_DS} = \sum_{i<j}^{N_{nauji}} (d_{ij}^{*} - d_{ij})^2 + \sum_{i=1}^{N_{nauji}} \sum_{j=N_{nauji}+1}^{N_{visi}} (d_{ij}^{*} - d_{ij})^2. \qquad (3.3)$$

Tačiau vaizdų projekcijų vertinimui, gautų skirtingais algoritmais buvo skaičiuojama paklaida E (3.1), nes normavimo parametro $\sum_{i<j}^{m}(d_{ij}^{*})^2$ įvedimas leidžia išvengti priklausomybės nuo mastelio ir atstumų skaičiaus *n*-matėje erdvėje. (Borg and Groenen 1997) knygoje yra įrodyta, kad tinkamai pakeitus mastelį, lokalius minimumus, gautus minimizuojant paklaidas E^2 ir E_{DS}, galima sutapatinti. Kadangi daugeliu atvejų praktikoje E^2 yra maža, tai yra patogiau naudoti paklaidą E, kuri duoda didesnę paklaidos reikšmę ir ją lengviau palyginti su kitomis paklaidos reikšmėmis. Šią paklaidą galima būtų naudoti ir standartiniame DS algoritme, tačiau jos negalima išskaidyti ir pritaikyti santykinių DS algoritme. Todėl E_{DS} (3.2) paklaida minimizuojama standartinio ir santykinių DS algoritmų vykdymo metu, o vizualizavimo kokybės įvertinimui buvo naudojama projekcijos paklaidos E (3.1) formulė.



a)



b)

3.2 pav. *Paklaidos priklausomybė nuo laiko*

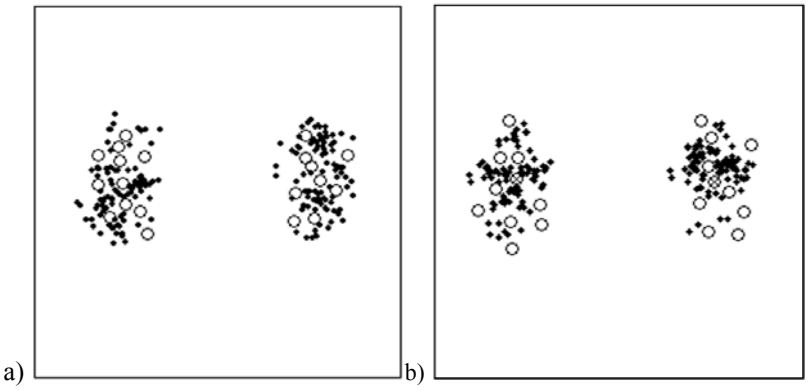
Taigi, naudojant kiekvieną bazinių vektorių parinkimo strategiją buvo atlikta po 100 eksperimentų, apskaičiuoti paklaidų ir skaičiavimo laiko vidurkiai. Gautos projekcijos paklaidos ir skaičiavimo laiko priklausomybės pateiktos 3.2 paveiksle. Skaičiai, esantys virš kreivių, žymi vidutinį bazinių vektorių skaičių

N_{fiks}, su kuriuo atlikti eksperimentai. Eksperimentai atlikti naudojant *Ellipsoidal* [1115; 50] duomenų aibę.

Kaip matome iš gautų rezultatų, naudoti I strategiją, pasiūlytą (Naud 2004), netikslinga, nes ji duoda prasčiausią rezultatą. Todėl tolimesniuose tyrimuose ši strategija nebus naudojama. Iš pateiktų rezultatų galima teigti, kad naudojant II ir III strategijas, vizualizavimo rezultatai gaunami gana panašūs, tačiau analizuojamos aibės klasterizavimas ir bazinių vektorių išrinkimas naudojant II strategiją užima daugiau laiko.

Didinant bazinių vektorių skaičių, naudojant visas minėtas strategijas, projekcijos paklaida mažėja. Straipsnyje (Naud 2006) buvo nustatyta, kad naudojant I strategiją, bazinių vektorių skaičių tikslinga didinti iki 500. Labiau didinant bazinių vektorių skaičių, taip pat auga ir klasterizavimo laikas, skaičiavimo laikas lėtėja, o paklaida kinta nežymiai (3.2 paveikslas).

Naudojant III strategiją (bazinius vektorius iš analizuojamos aibės parinkti atsitiktinai), bazinių vektorių skaičių galima didinti iki 1000 ir daugiau. Taip gauname tikslesnę projekciją, o taškų parinkimui laiko sugaištama nedaug, lyginant su kitomis strategijomis. Dėl šių priežasčių tolimesniuose tyrimuose bus naudojama III bazinių vektorių parinkimo strategija.



3.3 pav. *Dviejų nepersidengiančių 10-mačių sferų projekcijos: (a) naudojama III strategija,* $E = 0{,}11804$ *, (b) naudojama II strategija,* $E = 0{,}12265$

3.3 paveiksle pateikti dviejų 10-mačių sferų vizualizavimo rezultatai, naudojant II ir III bazinių vektorių parinkimo strategijas. Pirmiausia buvo suformuotos bazinių vektorių aibės *F*, ir naudojant standartinį DS algoritmą šios aibės vektoriai suprojektuoti į plokštumą.

Naudojant III strategiją, bazinių vektorių aibę sudaro 20 atsitiktinai pasirinktų analizuojamos aibės vektorių (3.3a paveiksle šie taškai pažymėti tuščiaviduriais apskritimais). Naudojant II strategiją, aibę *F* sudaro du poaibiai:

a) du aibės taškai, kurie yra artimiausi klasterių centrams, analizuojamą aibę suklasterizavus į du klasterius (3.3b paveiksle šie taškai žymimi perbrauktais apskritimais);

b) 9 taškai iš kiekvieno klasterio (3.3b paveiksle pažymėti tuščiaviduriais apskritimais).

Vizualizavimo rezultatai, gauti naudojant I strategiją yra analogiški pateiktiems 3.3b paveiksle naudojant II strategiją. Bazinių vektorių skaičius N_{fixed} abiem atvejais yra lygus 20 (II ir III strategijos). Likusieji aibės taškai (aibė *M*), paveiksle žymimi užpildytais apskritimais, į plokštumą suprojektuoti naudojant santykinių DS algoritmą.

Naudodami III strategiją gavome mažesnę projekcijos paklaidą (3.1) ($E = 0{,}11803$) nei naudodami II strategiją ($E = 0{,}12265$) (3.3 paveikslas).

Atliekant detalią santykinių DS algoritmo analizę, buvo nustatyta, kad vizualizavimo rezultatus įtakoja šie faktoriai:

I. Bazinių vektorių parinkimo strategija;

II. Pradinių dvimačių vektorių inicializavimo būdas;

III. Bazinių vektorių skaičius.

3.2.3. Inicializavimo problemos santykinių DS algoritme

Vizualizavimo rezultatai labai priklauso nuo dvimačių vektorių $Y_1, Y_2, ..., Y_m \in R^2$ pradinių reikšmių inicializavimo būdo. Tiriant standartinio DS algoritmo dvimačių vektorių inicializavimo būdus, buvo nustatyta (Bernatavičienė *et al.* 2006b), kad tiksliausia duomenų projekcija gaunama dvimačių vektorių inicializavimui naudojant pagrindinių komponenčių analizės (PCA) algoritmą. Todėl naudojant standartinio DS ir santykinių DS algoritmų junginį, dvimačiai vektoriai standartiniame DS bus inicializuojami remiantis PCA algoritmu.

Tačiau kokį inicializavimo būdą pasirinkti santykinių DS algoritme, dar nebuvo nustatyta. Taigi, šio tyrimo tikslas – nustatyti tinkamiausią dvimačių vektorių inicializavimo būdą, kuris bus toliau naudojamas santykinių DS algoritme. Buvo pasirinkti šeši skirtingi dvimačių vektorių inicializavimo būdai:

(a) Inicializavimo būdas, grįstas PCA algoritmu: daugiamatėje erdvėje bazinių vektorių koordinatėms yra suskaičiuojama vidurkių matrica $A[1 \times n]$ ir pasukimo matrica $T^*[n \times n]$. Iš matricos $T^*[n \times n]$ paimame dvi didžiausiais nuosavas reikšmes atitinkančius tikrinius vektorius, gaunama matrica

$T[2 \times n]$. Likusių vektorių dvimatės koordinatės yra inicializuojamos, naudojantis formule: $Y_i = (X_i - A)T'$, $i = 1,...,m$;

(b) pradinėms dvimačio vektoriaus koordinatėms iš likusių taškų aibės yra priskiriamos artimiausio bazinio vektoriaus dvimatės koordinatės;

(c) pradinėms dvimačio vektoriaus koordinatėms iš likusių taškų aibės yra priskiriamos atsitiktinės dvimačio vektoriaus koordinatės, sugeneruotos artimiausio bazinio vektoriaus aplinkoje, kurios spindulys yra lygus $r = 0,01$;

(d) ir (e) inicializavimo būdai yra analogiški (c) būdui, skiriasi tik aplinkos, kurioje generuojamas atsitiktinis dvimatis vektorius, spindulys: (d) $r = 0,1$ ir (e) $r = 1$;

(f) pradinėms dvimačio vektoriaus koordinatėms iš likusių taškų aibės yra priskiriamos atsitiktinės dvimačio vektoriaus koordinatės, sugeneruotos visoje bazinių vektorių projekcijų aplinkoje.

Eksperimentai buvo atlikti naudojant dvi duomenų aibes: *ellipsoidal* [1115; 50] ir *Gaussian* [2729; 10].

Naudojant (c), (d), (e), (f) inicializavimo būdus, buvo atliekama po 10 eksperimentų su kiekvienu skirtingu bazinių vektorių rinkiniu ir apskaičiuoti gautų paklaidų vidurkiai, nes inicijavimo rezultatą įtakojo tam tikras atsitiktinumo faktorius. Naudojant (a) ir (b) inicializavimo būdus atlikta po 1 eksperimentą su 100 skirtingų atsitiktinai atrinktų bazinių vektorių rinkinių.

Bazinių vektorių skaičius buvo keičiamas nuo 100 iki 1000 žingsniu 100. Buvo suformuota po 10 skirtingų fiksuoto kiekio bazinių vektorių rinkinių (naudota III bazinių vektorių parinkimo strategija). Naudojant visus šešis skirtingus dvimačių vektorių inicializavimo būdus, duomenys buvo projektuojami į plokštumą. Skaičiuotos paklaidos ir jų vidurkiai. Faktiškai, (c), (d), (e), ir (f) inicializavimo būdų atveju buvo vidurkinami paklaidų, gautų iš 10 eksperimentų kiekvienam bazinių vektorių rinkiniui, vidurkiai. Gauti rezultatai pateikiami 3.1 ir 3.2 lentelėse.

3.1 lentelė. *Eksperimento rezultatai ellipsoidal [1115; 50] duomenų aibei*

	100		300		500		700		900	
	vidurkis	dispersija	vidurkis	dispersija	vidurkis	dispersija	vidurkis	dispersija	vidurkis	dispersija
(a)	0,24609	0,00184	0,24261	0,00171	0,24103	0,00048	0,24061	0,00049	0,24023	0,00018
(b)	0,24620	0,00193	0,2426	0,00163	0,24103	0,00038	0,24059	0,00049	0,24023	0,00017
(c)	0,24617	0,00185	0,24261	0,00156	0,24103	0,00036	0,24059	0,00047	0,24023	0,00016
(d)	0,24616	0,00185	0,24261	0,00156	0,24103	0,00036	0,24059	0,00048	0,24023	0,00016
(e)	0,24636	0,00179	0,24266	0,00157	0,24105	0,00036	0,2406	0,00047	0,24023	0,00016
(f)	0,26150	0,00553	0,25252	0,00481	0,24683	0,00218	0,24384	0,00151	0,24231	0,00150

3.2 lentelė. *Eksperimento rezultatai Gausian [2729; 10] duomenų aibei*

	100		300		500		700		900	
	vidurkis	dispersija	vidurkis	dispersija	vidurkis	dispersija	vidurkis	dispersija	vidurkis	dispersija
(a)	0,28253	0,00640	0,27783	0,00493	0,27652	0,00511	0,27368	0,00061	0,27350	0,00052
(b)	0,28283	0,00685	0,27843	0,00507	0,27693	0,00516	0,27394	0,00065	0,27371	0,00041
(c)	0,28281	0,00647	0,27842	0,00482	0,27693	0,00492	0,27394	0,00062	0,27371	0,00039
(d)	0,28281	0,00647	0,27841	0,00483	0,27693	0,00492	0,27394	0,00063	0,27371	0,00039
(e)	0,28282	0,00647	0,27842	0,00483	0,27693	0,00492	0,27394	0,00063	0,27371	0,00039
(f)	0,28707	0,00733	0,28079	0,00516	0,27957	0,00505	0,27562	0,00090	0,27533	0,00103

Atlikti eksperimentai parodė, kad blogiausias inicializavimo būdas (f) yra atsitiktinis taškų parinkimas bazinių vektorių projekcijų srityje: gaunamas didžiausias projekcijos paklaidos vidurkis ir didžiausia paklaidos dispersija. Kitos strategijos duoda labai panašius rezultatus. Nors naudojant (a) inicializavimo būdą paremtą PCA algoritmu, paklaidos vidurkis šiek tiek mažesnis už paklaidų vidurkius gaunamus kitomis strategijomis (b, c, d, e), tačiau skirtumai tarp šių vidurkių statistiškai yra nereikšminiai.

Didinant bazinių vektorių skaičių paklaidos vidurkis ir dispersija mažėja, o paklaidos kitimas yra nereikšminis. Toks dėsningumas išlieka naudojant visus inicializavimo būdus.

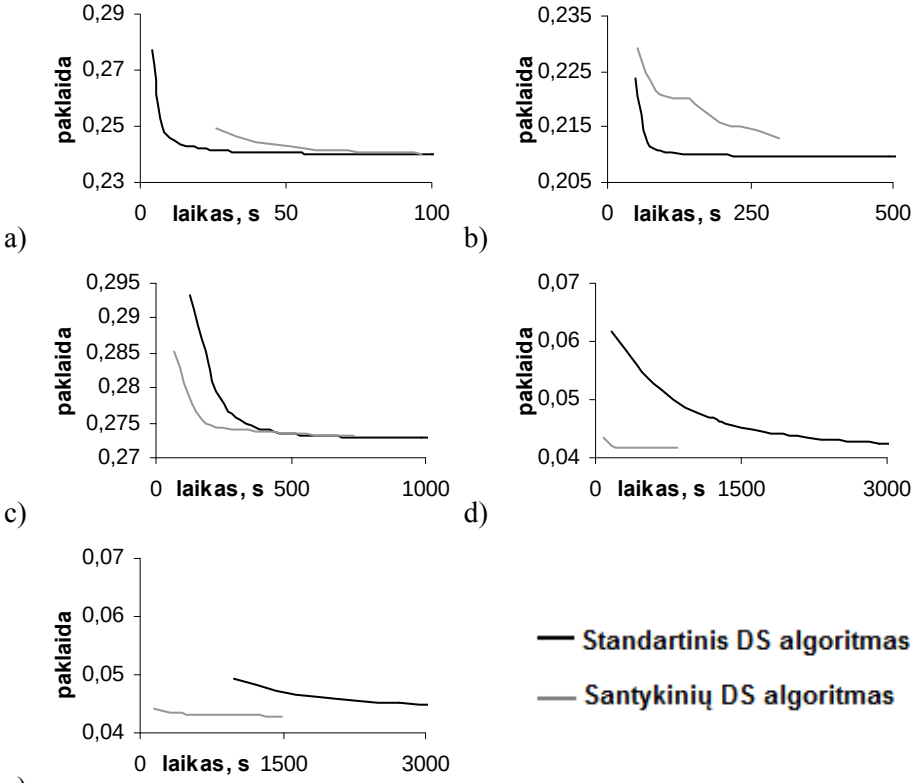
3.2.4. Santykinių DS algoritmo ir standartinio DS algoritmo lyginamoji analizė

Kito tyrimo tikslas buvo nustatyti, kada tikslinga naudoti santykinių DS, o kada standartinį DS algoritmą. Tam tikslui buvo vizualizuojamos penkios skirtingų dimensijų duomenų aibės su skirtingu vektorių skaičiumi.

Standartiniame DS algoritme buvo fiksuojamas skaičiavimo laikas ir paklaidos (3.1) dydis po kiekvienos iteracijos, o santykinių DS algoritme buvo fiksuojamas laikas ir paklaida (3.1) vizualizavus visą duomenų aibę, bazinių vektorių skaičiui kintant nuo 100 iki 1000 vizualizuojant mažas duomenų aibes ir nuo 100 iki 1500 vizualizuojant dideles duomenų aibes (3.4 paveikslas). Dar labiau didinant bazinių vektorių skaičių, labai didėja bazinių vektorių vizualizavimo laikas, o paklaida mažėja nežymiai.

Šiame skyriuje aprašomi eksperimentai buvo atlikti naudojant III bazinių vektorių parinkimo strategiją (baziniai vektoriai parenkami atsitiktinai iš visos analizuojamos duomenų aibės). Sudaryti bazinių vektorių rinkiniai, kai fiksuotas bazinių vektorių skaičius yra lygus $N_{fiks} = 100, 200, ..., 1500$. Buvo suformuota po 10 skirtingų bazinių vektorių rinkinių, naudojant vis kitą bazinių vektorių skaičių N_{fiks}. Duomenys vizualizuojami naudojant santykinių DS algoritmą, kiekvieną kartą apskaičiuojant projekcijos paklaidą ir skaičiavimo laiką. 3.4 paveiksle (projekcijos paklaidos priklausomybės nuo skaičiavimo laiko kreivė,

naudojant santykinių DS algoritmą, pažymėta pilka spalva) pateikti projekcijos paklaidų ir skaičiavimo laiko vidurkiai atlikus 10 eksperimentų. Skaičiavimo laiko ir projekcijos paklaidos priklausomybės kreivė (laikas, matuojamas sekundėmis, ir skaičiavimo paklaida fiksuoti po kiekvienos iteracijos), gauta naudojant standartinį DS algoritmą, yra pažymėta juoda spalva (3.4 paveikslas).

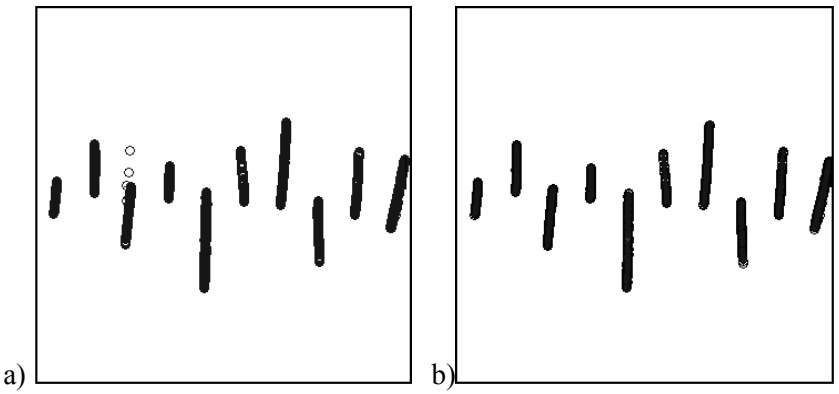


3.4 pav. *Projekcijos paklaidos priklausomybė nuo skaičiavimo laiko (sekundėmis): (a) ellipsoidal [1115; 50] duomenų aibė; (b) paraboloid [2583; 3] duomenų aibė; (c) Gaussian [2729; 10] duomenų aibė; (d) ellipsoidal [3140; 50] duomenų aibė; (e) abalone [4177; 7] duomenų aibė*

Gauti rezultatai rodo, kad vizualizuojant duomenų aibes turinčias daugiau nei 3000 vektorių ir kai vektorių dimensija didesnė nei 5, tikslinga naudoti santykinių DS algoritmą (3.4 paveikslas, (c), (d), (e)). Esant ribotam skaičiavimo laiko resursui santykinių DS algoritmas duoda tikslesnį vaizdą nei standartinis DS. Kai mažas vizualizuojamų taškų skaičius ir maža dimensija, santykinių DS

algoritmo naudoti netikslinga, nes jo gaunama paklaida yra didesnė lyginant su paklaida gaunama standartiniu DS algoritmu.

3.5 paveiksle pateikiamos *ellipsoidal* [3140; 50] duomenų aibės projekcijos, naudojant abu algoritmus. Iš pateiktų rezultatų matyti, kad vizualizuojant šią aibę DS algoritmu gaunama didesnė paklaida lyginant su paklaida gauta naudojant Santykinių DS algoritmą, ir sugaištama 9 kartus laiko ilgiau.



3.5 pav. *Ellipsoidal [3140; 50] duomenų aibės projekcija: (a) gauta naudojant standartinį DS, (b) gauta naudojant santykinių DS (bazinių vektorių skaičius lygus 1500)*

Atlikus 50 iteracijų standartiniu DS algoritmu, skaičiavimo laikas yra lygus 7530s, o projekcijos paklaida $E = 0,04174$. Naudojant santykinių DS algoritmą gaunama mažesnė projekcijos paklaida ir sutaupomas skaičiavimo laikas ($E = 0,04161$, 842 s). Kaip matome iš projekcijos, gautos standartiniu DS algoritmu (3.5a paveikslas), keli vieno klasterio taškai šiek tiek nutolę nuo kitų savo klasterio taškų.

3.3 lentelė. *Paklaida ir skaičiavimo laikas, gauti abiem algoritmais*

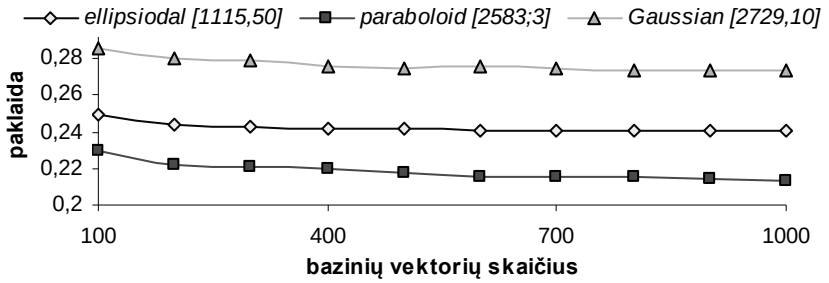
Duomenų aibė	DS po 50 iteracijų		Santykinių DS alg.	
	paklaida	laikas, s	paklaida	laikas, s
ellipsoidal [1115; 50]	0,240134	102	0,240232	96
ellipsoidal [3140; 50]	0,041745	7530	0,041615	842
Gaussian [2729; 10]	0,272748	1389	0,273075	732
paraboloid [2583; 3]	0,209169	1079	0,212924	300
abalone [4177; 7]	0,043681	24071	0,042907	1445

3.3 lentelėje pateikti skaičiavimo rezultatai, gauti vizualizavus 5 duomenų aibes abiem algoritmais: paklaida ir skaičiavimo laikas. Standartinis DS algoritmas buvo sustabdytas po 50 iteracijų. Tiek pat iteracijų buvo skirta

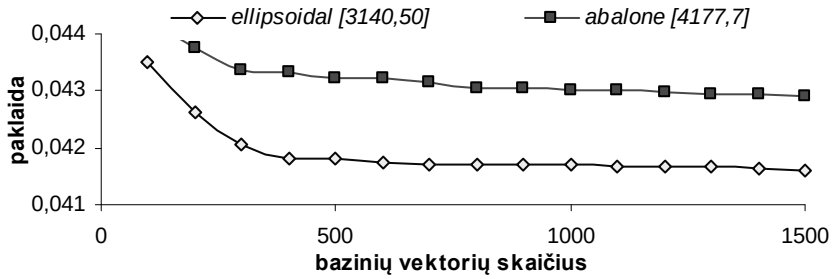
standartiniam DS algoritmui, projektuojant bazinius vektorius. Santykinių DS algoritme laikas buvo fiksuojamas, kai gaunamos visų taškų projekcijos. Bazinių taškų skaičius fiksuotas: aibėms *ellipsoidal* [1115; 50], *paraboloid* [2583; 3] lygus 1000, kitoms aibėms lygus 1500. Gauti rezultatai rodo, kad turint dideles aibes ir esant ribotiems skaičiavimo laiko resursams tikslesnę duomenų projekciją gauname naudojant santykinių DS algoritmą, ir tam sugaištame nedaug laiko.

3.2.5. Optimalaus bazinių vektorių skaičiaus parinkimas

Naudojant santykinių DS algoritmą didelių aibių vizualizavimui, labai svarbu apibrėžti optimalų bazinių vektorių skaičių. Bazinių vektorių skaičiaus didinimas lėtina skaičiavimus, tačiau gaunama mažesnė skaičiavimo paklaida ir tikslesnė projekcija. Taigi labai svarbu nustatyti optimalų bazinių vektorių skaičių, kad skaičiavimo kaštai nebūtų per daug dideli, o gauta projekcija būtų pakankamai informatyvi.



a)



b)

3.6 pav. *Projekcijos paklaidos priklausomybė nuo bazinių vektorių skaičiaus*

3.6 paveiksle pateikta projekcijos paklaidos priklausomybė nuo bazinių vektorių skaičiaus N_{fiks}. Eksperimentai buvo atlikti naudojant III bazinių vektorių parinkimo strategiją. Su kiekvienu fiksuotu bazinių vektorių skaičiumi

buvo atlikta po 100 eksperimentų, skaičiuotos projekcijų paklaidos ir gautas paklaidų vidurkis.

Kaip matyti iš 3.6 paveikslo, vidutinė projekcijos paklaida palaipsniui mažėja, didinant bazinių vektorių skaičių. Tačiau galima pastebėti, kad vizualizuojant mažesnes duomenų aibes (aibėse yra nuo 1000 iki 3000 vektorių) projekcijos paklaida stabilizuojasi, kai bazinių vektorių skaičius $N_{fiks.} \approx 700$ (3.6a paveikslas), o vizualizuojant didesnes duomenų aibes (aibėse yra daugiau nei 3000 vektorių), kai $N_{fiks} \approx 900$ (3.6b paveikslas). Labiau didinant bazinių vektorių skaičių projekcijos paklaidos (3.1) pokytis yra nereikšminis: paklaidos pokytis pastebimas tik 4 – 5 ženkle po kablelio.

Sekančio eksperimento tikslas yra parodyti, kaip svarbu tinkamai parinkti bazinius taškus. Baziniai vektoriai turėtų būti tolygiai pasiskirstę po visą duomenų aibės padengimo sritį.

Šiam eksperimentui naudosime II bazinių taškų parinkimo strategiją, kurioje klasterių centrai, gaunami *k*-vidurkių klasterizavimo algoritmu, gana tolygiai pasiskirsto po visą duomenų aibę, o imant papildomus vektorius iš kiekvieno klasterio, išlaikomas tolygumas didinant bazinių taškų skaičių.

3.4 lentelė. *Projekcijos paklaidos, gautos naudojant II bazinių vektorių parinkimo strategiją, ellipsoidal [1115; 50] duomenų aibei*

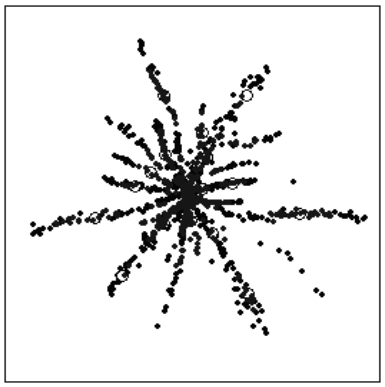
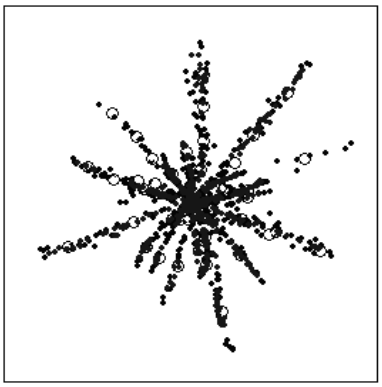
p \ *k*	**10**	**20**	**30**	**40**	**50**
5	0,3039	0,2640	0,2532	0,2499	0,2468
10	0,3045	0,2572	0,2469	0,2482	0,2448
15	0,2991	0,2576	0,2459	0,2448	0,2430
20	0,3038	0,2555	0,2446	0,2434	0,2413
25	0,2982	0,2503	0,2436	0,2427	0,2408

3.4 lentelėje yra pateiktos *ellipsoidal* [1115; 50] duomenų aibės projekcijų paklaidos, gautos su skirtingais fiksuoto bazinių vektorių skaičiaus rinkiniais. Bazinių vektorių rinkiniai buvo suformuoti tokiu būdu: duomenys suklasterizuoti į $k = 10, 20, ..., 50$ klasterių, iš kiekvieno klasterio paimta po $p = 5, 10, ..., 25$ vektorių.

Kaip rodo pateikti rezultatai, vizualizuojant duomenų aibės taškus, tikslingiau imti didesnį klasterių skaičių ir mažiau klasterio papildomų taškų, taip tolygiau padengiame vizualizuojamą duomenų aibę baziniais taškais. Pavyzdžiui, paėmus fiksuotą bazinių vektorių skaičių $N_{fiks.} = 300$ (3.4 lentelė), kai $k = 20$, o $p = 15$, paklaida lygi $E = 0,2576$, o kai $k = 30$, o $p = 10$, paklaida lygi $E = 0,2469$. Taigi šis eksperimentas parodo, kad kuo tolygiau padengiame

vizualizuojamą duomenų aibę baziniais taškais, tuo tikslesnę projekciją gauname, tai iliustruoja ir 3.7 paveikslas.

3.7 paveiksle pateikti *ellipsoidal* [1115; 50] duomenų aibės vizualizavimo rezultatai (bazinių vektorių skaičius lygus $N_{fiks} = 500$): (a) klasterių skaičius $k = 20$, o iš kiekvieno klasterio imama po $p = 25$ vektorių, (b) klasterių skaičius $k = 50$, o iš kiekvieno klasterio imama po $p = 10$ vektorių. (b) atveju gauta mažesnė projekcijos paklaida ir geresnė vizualizavimo kokybė.

a)  b) 

3.7 pav. *Ellipsoidal [1115, 50] duomenų aibės vizualizavimo rezultatai* ($N_{fiks.} = 500$)*: (a)* $k = 20$, $p = 25$, $E = 0,2503$, *(b)* $k = 50$, $p = 10$, $E = 0,2448$

Atlikti tyrimai leido padaryti šias išvadas:

- Vizualizavimo rezultatai labai priklauso nuo bazinių vektorių parinkimo strategijos.
- Buvo nustatyta, kad dvimačių vektorių inicializavimo būdas taip pat daro įtaką vizualizavimo rezultatams. Buvo pasiūlyti ir ištirti 6 skirtingi dvimačių vektorių inicializavimo būdai. Atlikti eksperimentai parodė, kad blogiausias inicializavimo būdas yra atsitiktinis taškų parinkimas bazinių vektorių projekcijų srityje: gaunamas didžiausias projekcijos paklaidos vidurkis ir didžiausia paklaidos dispersija. Kitos strategijos duoda labai panašius rezultatus. Nors naudojant inicializavimo būdą paremtą PCA algoritmu, paklaidos vidurkis mažesnis už paklaidų vidurkius gaunamus kitomis strategijomis, tačiau skirtumai tarp šių vidurkių statistiškai yra nereikšminiai.
- Vizualizuojant daugiamačius duomenis kai tiriamų duomenų dimensija yra didesnė už 5, o duomenų aibę sudaro daugiau nei 3000 vektorių, tikslingiau vietoj standartinio daugiamačių skalių algoritmo naudoti santykinių DS algoritmą. Tenkinant vizualizuojamai duomenų aibei

minėtus kriterijus, santykinių DS algoritmu gaunama pakankamai tiksli projekcija ir taupomas skaičiavimo laikas lyginat su standartiniu DS algoritmu. Turint ribotą skaičiavimo laiką, šiais atvejais santykinių DS algoritmu gauname tikslesnę projekciją nei standartiniu DS algoritmu.

- Kuo didesnė vizualizuojamų duomenų aibė, tuo didesnį bazinių vektorių skaičių reikia imti.
- Didinant bazinių vektorių skaičių gaunama tikslesnė projekcija. Tačiau per didelis bazinių vektorių skaičius lėtina skaičiavimus. Tyrimai parodė, kad mažesnėms duomenų aibėms (iki 3000 vektorių) tikslinga imti nuo 700 iki 1000 bazinių vektorių, o didelėms duomenų aibėms – nuo 900 iki 1500.
- Didinant bazinių vektorių skaičių, vidutinė projekcijos paklaida mažėja; paklaidos dispersijos mažėjimas nereikšminis.
- Kuo tolygiau baziniai vektoriai pasiskirstę po visą tiriamą aibę, tuo tikslesnė projekcija yra gaunama.

3.3. Atstumų koregavimas vizualizuojant daugiamačius duomenis

Dažnai duomenys būna daugiamačiai. Būtina ieškoti būdų juos pateikti žmogui suvokiama forma, pvz., vizualiai. Vizualizuoti tokius duomenis galima juos projektuojant į plokštumą. Šiam tikslui tinka daugiamačių duomenų projekcijos į mažesnės dimensijos erdvę metodai: pagrindinių komponenčių analizė (Taylor 2003), daugiamačių skalių (DS) (Torgerson 1952), Sammono projekcija (Sammon 1969) ir kt. Transformuojant daugiamačius duomenis į dvimatę plokštumą siekiama kuo mažiau iškraipyti daugiamačių duomenų tarpusavio atstumus. Kiekvienas projektavimo metodas naudoja savo iškraipymo kriterijų. Daugiamačių skalių metodo apibendrinta iškraipymo kriterijaus funkcija (paklaidos arba STRESS funkcija) (Groenen *et al.* 1996) yra

$$E_{DS} = \sum_{i=1}^{m} \sum_{j=i+1}^{m} w_{ij} (f((d_{ij}^{*})^2) - f((d_{ij})^2))^2,$$

čia d_{ij}^{*} – atstumas tarp daugiamačių vektorių $X_i, X_j \in R^n$, $i, j = 1,...,m,$ naudojamas vietoj nepanašumo mato δ_{ij}, naudojamo apibendrintoje daugiamačių skalių paklaidos formulėje pateiktoje (Groenen *et al.* 1996); d_{ij} yra atstumas tarp vektorius X_i, X_j atitinkančių dvimačių vektorių $Y_i, Y_j \in R^2$; w_{ij} yra fiksuotas

neneigiamas svoris. Funkcija *f(z)* transformuoja atstumus. (Groenen *et al.* 1996) straipsnyje pateikti trys funkcijos *f(z)* pavyzdžiai: Kruskalo raw STRESS funkcija $f(z) = z^{\frac{1}{2}}$ pateikta (Kruskal 1964), SSTRESS funkcija $f(z) = z$ pasiūlyta (Takane *et al.* 1977) ir Ramsay funkcija $f(z) = \log(z)$ pasiūlyta (Ramsay 1977).

Tačiau, net ir tiksliai optimizavus iškraipymų kriterijų arba kitaip vadinamą projekcijos paklaidą, tokios transformacijos sukelti tarpusavio atstumų iškraipymai dažnai apsunkina vizualią analizę. To priežastys – esmingai skirtingi atstumų tarp duomenų taškų pasiskirstymo dėsningumai didesnių matmenų srityse.

Vienetiniame kube atstumų tarp dviejų taškų pasiskirstymai buvo tirti darbuose (Sabirov 2000), (Žilinskas and Podlipskytė 2002). Atstumų pasiskirstymų vienmatėje, dvimatėje ir trimatėje erdvėse palyginimas pateiktas (Žilinskas and Podlipslytė 2002) straipsnyje, o teorinis pasiskirstymo apibrėžimas (Žilinskas 2003). Šiame darbe pasiūlyta trimačiame kube analitinė atstumų transformacija tokia, kuri atstumų pasiskirstymą trimačiame kube padaro tokiu pačiu kaip ir atstumų pasiskirstymas dvimačiame kube.

Šioje disertacijoje pateikiama kitokia atstumų transformacijos arba korekcijos funkcija pagrįsta statistiniu pasiskirstymų vertinimu. Ši transformacija gali būti taikoma ir didesnių dimensijų erdvėse. Ši atstumų korekcija (transformacija) pagerins projekcijos kokybę: paryškins duomenų klasterius, suprojektavus daugiamačius duomenis labiau išryškės duomenų struktūra, lengviau taps identifikuojami taškai atsiskyrėliai.

Šiame darbe bus naudojami du paprasčiausi daugiamačių skalių metodai. Vienas jų pasiūlytas (Torgerson 1952), kuriame optimizuojama tikslo funkcija yra atstumų nuokrypių kvadratų suma (2.4 formulė, kai $w_{ij} = 1$).

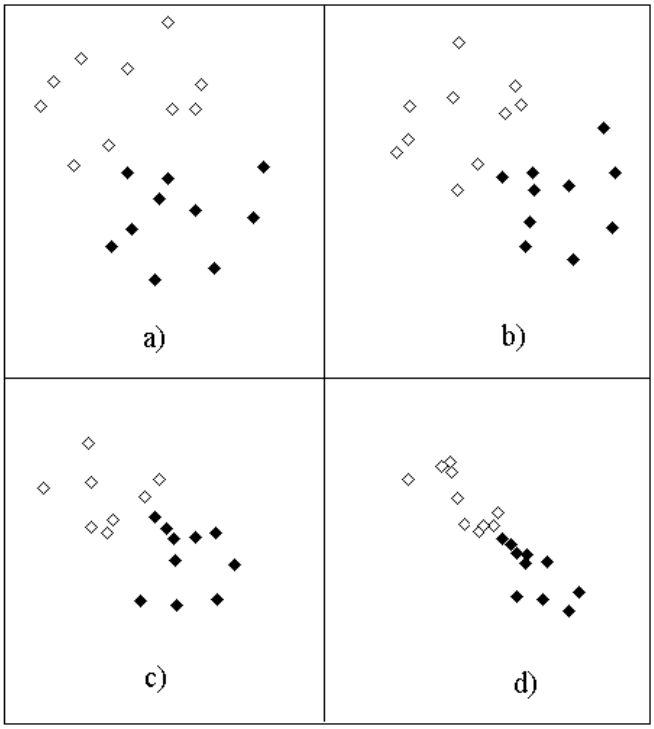
Kitas algoritmas, kuris bus naudojamas eksperimentuose yra Sammono algoritmas, jame naudojama kita tikslo funkcija (Sammon 1969):

$$E = \frac{1}{\sum\limits_{\substack{i,j=1 \\ i<j}}^{m} d_{ij}^{*}} \sum_{\substack{i,j=1 \\ i<j}}^{m} \frac{(d_{ij}^{*} - d_{ij})^2}{d_{ij}^{*}}.$$

3.3.1. Atstumų netiesinės korekcijos įtaka vizualizavimo rezultatams

Bet kuri netiesinė taškų tarpusavio atstumų korekcija turi įtakos transformacijos vaizdui. 3.8 paveiksle pateikti šešiamačių duomenų, tolygiai pasiskirsčiusių dviejose nežymiai (13 %) persidengiančiose sferose,

vizualizavimo rezultatai skirtingiem netiesinės korekcijos laipsniams, (naudotas DS kriterijus): a) nenaudojant korekcijos; b) naudojant korekciją daugiamačių taškų atstumams $(d_{ij}^*)^{1,2}$; c) naudojant korekciją $(d_{ij}^*)^{1,6}$; d) naudojant korekciją $(d_{ij}^*)^3$, čia d_{ij}^* – atstumai *n*-matėje erdvėje.



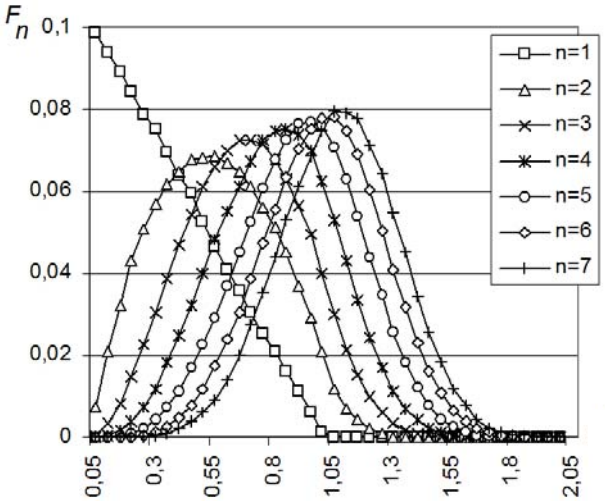
3.8 pav. *Šešiamačių duomenų, tolygiai pasiskirsčiusių dviejose nežymiai (13 %) persidengiančiose sferose, vizualizavimo rezultatai: a) nenaudojant korekcijos; b) korekcija* $(d_{ij}^*)^{1,2}$ *; c) korekcija* $(d_{ij}^*)^{1,6}$ *; d) korekcija* $(d_{ij}^*)^3$

Tokio pobūdžio korekcijos tarpusavyje glaudina artimesnius taškus, ką ir iliustruoja 3.8 paveikslas Geriausiu transformavimo rezultatu laikytina korekcijai $(d_{ij}^*)^{1,6}$ atitinkantis 3.8c paveikslas, tuo tarpu korekcija $(d_{ij}^*)^3$ (3.8d paveikslas) laikytina „per stipria“, nes tolimi taškai per daug nutolinami. Atstumai transformuojami tik daugiamatėje erdvėje.

Kaip matome, tiek atstumų korekcijos pobūdis, tiek jos laipsnis iš esmės keičia transformacijos kokybę, todėl tikslinga ieškoti tam tikra prasme optimalių korekcijų. Formuluojant reikalavimus korekcijai atsižvelgsime į atstumų tarp duomenų taškų pasiskirstymo dėsningumus didesnės dimensijos erdvėse.

3.3.2. Atstumų tarp tolygiai pasiskirsčiusių taškų daugiamačiame vienetiniame kube pasiskirstymai

Euklidinių atstumų tarp tolygiai pasiskirsčiusių taškų daugiamačiame vienetiniame kube pasiskirstymai tyrinėti darbe (Šaltenis 2004). Statistiškai modeliuojant gauti tarpusavio atstumų pasiskirstymų tankiai įvairiam matmenų skaičiui *n*. Pasiskirstymų tankių grafikai pateikti 3.9 paveiksle.



3.9 pav. *Tarpusavio atstumų daugiamačiame vienetiniame kube pasiskirstymų tankių grafikai įvairių matavimų erdvėse*

Toliau naudosimės ne šiais pasiskirstymų tankiais, o jiems atitinkančiomis pasiskirstymo funkcijomis $F_n(x)$. Tai tikimybės, kad tarpusavio atstumas ξ tarp atsitiktinių taškų bus mažesnis už duotą dydį x, priklausomybė nuo x:

$$F_n(x)=P(\xi<x).$$

3.3.3. Pagrindinė siūlomos korekcijos idėja

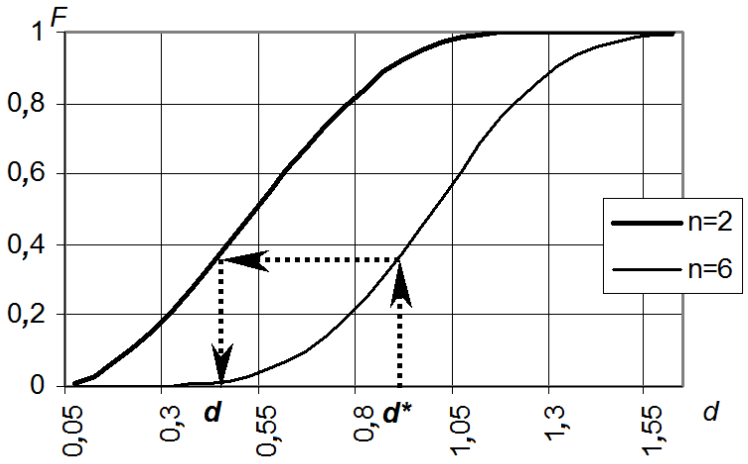
Akivaizdus tokiai atstumų korekcijai keliamas monotoniškumo reikalavimas. Labiau nutolusių duomenų pora ir po transformacijos turi likti labiau nutolusi.

Taip pat korekcija turi būti tokia, kad tolygiai pasiskirsčiusių taškų n-mačiame (n>2) vienetiniame kube tarpusavio atstumai $d^*[n]$, parametras n nusako erdvės matmenų skaičių, transformuoti į dvimatę erdvę, turėtų tarpusavio atstumų $d[n=2]$ pasiskirstymą, būdingą dvimatei erdvei (žiūr. 3.10 paveikslą).

Tam tarpusavio atstumai $d^*[n]$ turi būti trumpinami, dauginant juos iš tam tikro koregavimo koeficiento

$$k_n = \frac{d}{d^*},$$

čia $F_n(d^*) = F_2(d)$, $F_n(d^*)$, $F_2(d)$ – pasiskirstymo funkcijos n-matėje ir dvimatėje erdvėse.



3.10 pav. *Koregavimo koeficiento radimo idėjos iliustracija. Čia pateiktos tarpusavio atstumų pasiskirstymo funkcijos dvimačiam ir šešiamačiam atvejui;* $d^*[n=6]$ *– atstumo reikšmė šešiamačiu atveju,* $d[n=2]$ *– jam atitinkantis koreguotas atstumas dvimačiu atveju*

Tokiu būdu kiekvienam matmenų skaičiui n apskaičiuojamos monotoniškai augančios koregavimo koeficiento priklausomybės nuo atstumo daugiamatėje srityje $d^*[n]$ (atstumai yra normuoti intervale [0;1]).

Šias priklausomybes pakankamai gerai aproksimuoja išraiška:

$$k_n = 1 - \exp(-c_1(d^* - c_2)), \tag{3.4}$$

čia formulėje naudojamų koeficientų c_1 ir c_2 reikšmės priklauso nuo matmenų skaičiaus n ir yra pateiktos 3.5 lentelėje.

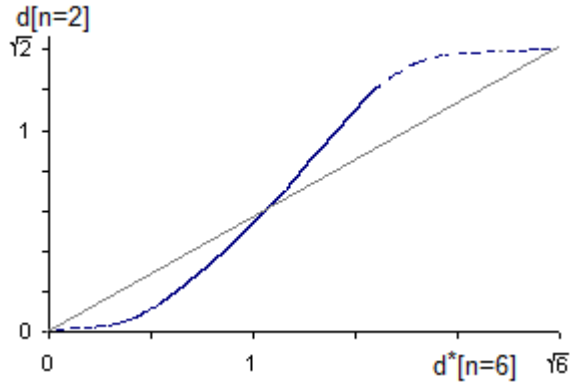
3.5 lentelė. *Koeficientų c_1 ir c_2 reikšmės*

Matmenų skaičius *n*	Koeficiento c_1 reikšmė	Koeficiento c_2 reikšmė
4	1,4	0,04
5	1,18	0,16
6	1,05	0,25

Tokios aproksimuojančios priklausomybės (3.4) skaičiuojant patogesnės už koregavimo koeficientų k_n lenteles, tačiau jos netinka mažoms (d_n<0,25) atstumų reikšmėms. Tokiu atveju tikslinga naudoti lentelines k_n reikšmes, kurios yra pateiktos 3.6 lentelėje.

3.6 lentelė. *Koeficientų k_n reikšmės skirtingoms dimensijoms n*

d_n	*n*			
	3	4	5	6
0	0	0	0	0
0,2	0,475	0,225	0,125	0,075
0,4	0,663	0,413	0,250	0,150
0,6	0,775	0,550	0,383	0,275
0,8	0,844	0,663	0,525	0,406
1	0,880	0,760	0,635	0,540
1,2	0,900	0,796	0,717	0,646



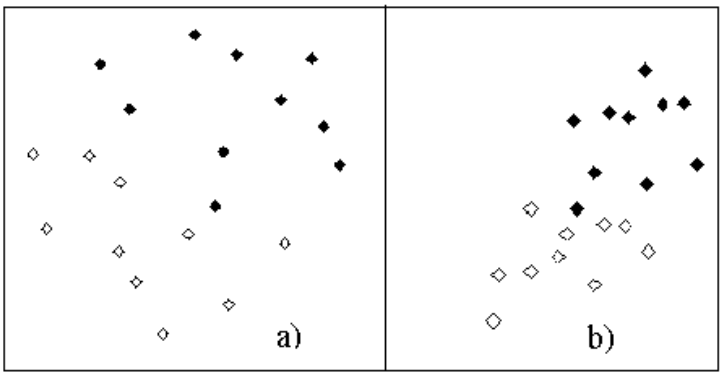
3.11 pav. *6-matės erdvės atstumų transformavimo funkcijos į dvimatės erdvės atstumus iliustracija*

6-matės erdvės atstumų transformavimo funkcijos į dvimatės erdvės atstumus iliustracija pateikta 3.11 paveiksle. Punktyrine linija pažymėta transformavimo funkcijos kreivės dalis gauta ekstrapoliuojant ir gautas reikšmes aproksimuojant.

3.3.4. Eksperimentinis korekcijų taikymo įvertinimas

Eksperimentai atlikti naudojant DS

Tiriant vizualizavimo kokybę iškraipymo kriterijaus funkcijos lokalaus minimumo radimui naudotasi populiariu kvazi Niutono metodu (Davidon 1959), (Fletcher and Powel 1963). Globaliam minimumui rasti lokalus optimizavimas kartotas 30 kartų.



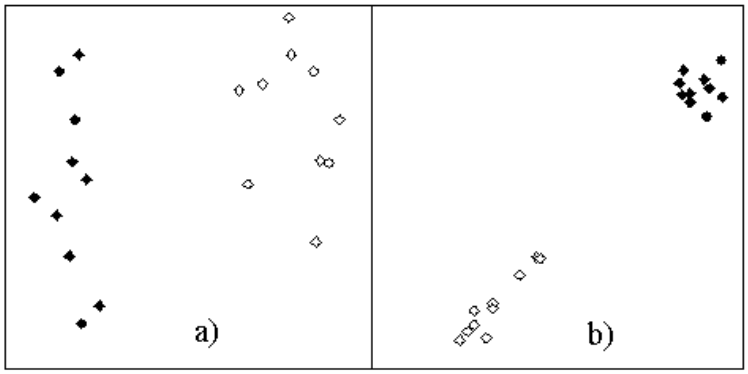
3.12 pav. *6-mačių duomenų, tolygiai pasiskirsčiusių dviejose žymiai (40 %) persidengiančiose sferose, vizualizavimo rezultatai: a) nenaudojant korekcijos; b) naudojant siūlomą korekcija*

Tyrimuose naudoti duomenys. Du vienodo spindulio tarpusavyje įvairiai nutolę arba dalinai susilieję sferiniai duomenų klasteriai. Kiekviename klasteryje – po 10 duomenų taškų.

3.12 paveiksle sferiniai duomenų klasteriai žymiai tarpusavyje persidengia (persidengimas sudaro 40 % atstumo tarp daugiamačių sferų centrų). Kaip matome, nenaudojant korekcijos (3.12a paveikslas) skirtingų klasterių duomenys praktiškai neatskiriami, jei duomenų taškai nebūtų skirtingai atžymėti. Tuo tarpu panaudojus korekciją (3.12b paveikslas) galima įžvelgti atsiskiriančias, neištįsusias taškų grupes.

3.13 paveiksle sferiniai duomenų klasteriai žymiai tarpusavyje nutolę (per 1,4 sferų spindulio tarp daugiamačių sferų centrų). Aišku, tiek nenaudojant korekcijos (3.13a paveikslas) tiek ją naudojant skirtingų klasterių duomenys gerai atsiskiria. Tačiau nekoreguojant atstumų klasterių vaizdai nekompaktiški.

Panaudojus korekciją galima matyti labiau primenančias daugiamačius klasterius taškų grupes.



3.13 pav. *6-mačių duomenų, tolygiai pasiskirsčiusių dviejose žymiai (per 1,4 sferos spindulio) viena nuo kitos nutolusiose sferose, vizualizavimo rezultatai:| a) nenaudojant korekcijos; b) naudojant siūlomą korekciją*

Eksperimentai atlikti naudojant Sammono algoritmą, kuriam taikytas koordinatinės paieškos principas.

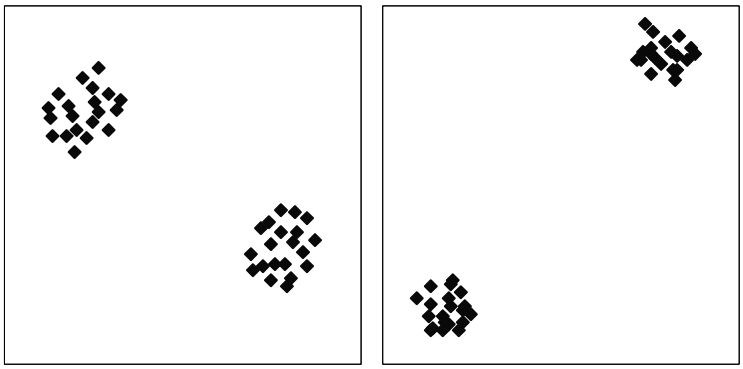
Tolesniuose eksperimentuose buvo naudotas Sammono algoritmas, kuriam taikytas koordinatinės paieškos principas (Dzemyda *et al.* 2004). Globaliam minimumui rasti lokalus optimizavimas kartotas 100 kartų.

Tyrimuose naudoti duomenys:

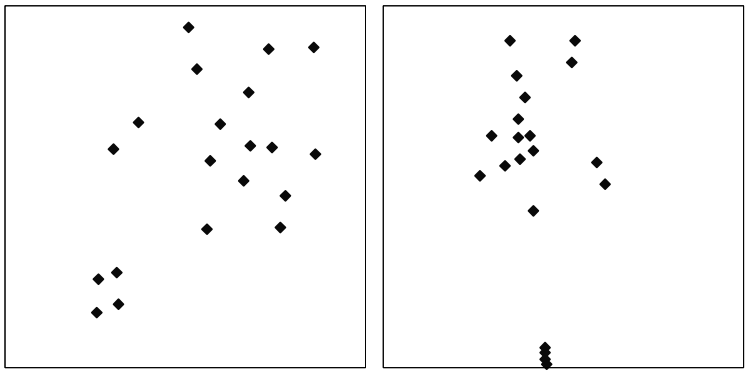
- Du vienodo spindulio tarpusavyje žymiai nutolę sferiniai duomenų klasteriai. Kiekviename klasteryje yra po 20 duomenų taškų.
- Wood duomenys (Draper and Smith 1966) (20 5-mačių vektorių, tarp kurių yra 4 taškai-atsiskyrėliai). Šie duomenys paprastai naudojami taškų–atsiskyrėlių išskyrimo algoritmų testavimui.

3.14 paveiksle pavaizduotos dviejų vienodo spindulio tarpusavyje žymiai nutolusių sferinių klasterių projekcijos be korekcijos ir su korekcija. Korekcija Sammono projekcijai neturi didelės įtakos, nes atvaizdavimo kokybė palyginus ir taip nebloga. Tačiau iš pateiktų vaizdų matyti, kad panaudojus korekciją artimi taškai dar labiau susispaudė, paryškindami klasterius.

3.15a paveiksle pateikta Wood duomenų projekcija netaikant korekcijos. Čia sunku atskirti taškus atsiskyrėlius. Tačiau panaudojus korekciją, artimi taškai susispaudė, todėl 3.15b paveiksle jau galime lengvai išskirti 4 susigrupavusius taškus atsiskyrėlius.



3.14 pav. *6-mačių duomenų, tolygiai pasiskirsčiusių dviejose žymiai viena nuo kitos nutolusiose sferose, vizualizavimo rezultatai naudojant Sammono algoritmą: a) be korekcijos; b) naudojant siūlomą korekciją*



3.15 pav. *Wood 5-mačių duomenų vizualizavimo rezultatai naudojant Sammono algoritmą: a) be korekcijos; b) naudojant siūlomą korekciją*

Disertacijoje pasiūlytas daugiamačių duomenų taškų tarpusavio atstumų koregavimas atliekant jų netiesinį projektavimą (DS arba Sammono algoritmais) į dvimatę plokštumą. Jis kartais pagerina vizualizavimo kokybę, koregavimas geriau išryškina duomenų klasterius, mažiau iškraipo daugiamačių duomenų struktūras.

Siūlomos korekcijos palyginus paprastos – pakanka tarpusavio atstumus daugiamatėje erdvėje dauginti iš atitinkamų koregavimo koeficientų, skirtingų įvairiems matmenų skaičiams. Pateikta koregavimo koeficientų reikšmių lentelė, o taip pat aproksimuojančios priklausomybės keliems daugiamačių duomenų atvejams.

3.4. Trečiojo skyriaus apibendrinimas ir išvados

1. Žinių gavybos vizualiais metodais proceso susisteminimas leido visapusiškai įvertinti ir pritaikyti vizualizavimo metodų ir priemonių teikiamas galimybes duomenų analizės efektyvumui didinti.
2. Detaliai ištyrus santykinių DS algoritmą, galime daryti šias išvadas:
   - Vizualizavimo rezultatai labai priklauso nuo bazinių vektorių parinkimo strategijos.
   - Buvo nustatyta, kad dvimačių vektorių inicializavimo būdas taip pat daro įtaką vizualizavimo rezultatams. Buvo pasiūlyti ir ištirti 6 skirtingi dvimačių vektorių inicializavimo būdai. Atlikti eksperimentai parodė, kad blogiausias inicializavimo būdas yra atsitiktinis taškų parinkimas bazinių vektorių projekcijų srityje: gaunamas didžiausias projekcijos paklaidos vidurkis ir didžiausia paklaidos dispersija. Kitos strategijos duoda labai panašius rezultatus. Nors naudojant inicializavimo būdą paremtą PCA algoritmu, paklaidos vidurkis šiek tiek mažesnis už paklaidų vidurkius gaunamus kitomis strategijomis, tačiau skirtumai tarp šių vidurkių statistiškai yra nereikšminiai.
   - Vizualizuojant daugiamačius duomenis kai tiriamų duomenų dimensija yra didesnė už 5, o duomenų aibę sudaro daugiau nei 3000 vektorių, tikslingiau vietoj standartinio daugiamačių skalių algoritmo naudoti santykinių DS algoritmą. Tenkinant vizualizuojamai duomenų aibei minėtus kriterijus, santykinių DS algoritmu gaunama pakankamai tiksli projekcija ir taupomas skaičiavimo laikas lyginat su standartiniu DS algoritmu. Turint ribotą skaičiavimo laiką, šiais atvejais Santykinių DS algoritmu gauname tikslesnę projekciją nei standartiniu DS algoritmu.
   - Kuo didesnė vizualizuojamų duomenų aibė, tuo didesnį bazinių vektorių skaičių reikia imti. Didinant bazinių vektorių skaičių gaunama tikslesnė projekcija. Tačiau per didelis bazinių vektorių skaičius lėtina skaičiavimus. Tyrimai parodė, kad mažesnėms duomenų aibėms (iki 3000 vektorių) tikslinga imti nuo 700 iki 1000 bazinių vektorių, o didelėms duomenų aibėms – nuo 900 iki 1500.
   - Didinant bazinių vektorių skaičių, vidutinė projekcijos paklaida mažėja; paklaidos dispersijos mažėjimas nereikšminis.
   - Kuo tolygiau baziniai vektoriai pasiskirstę po visą tiriamą aibę, tuo tikslesnė projekcija yra gaunama.
3. Pasiūlyta daugiamačių duomenų taškų tarpusavio atstumų koregavimo transformacija atliekant jų netiesinį projektavimą (DS arba Sammono

algoritmais) į dvimatę plokštumą. Atliktas atstumų koregavimo transformacijos tyrimas leido padaryti šias išvadas:

- Pasiūlyta transformacija pagerina vizualizavimo kokybę, koregavimas geriau išryškina duomenų klasterius, mažiau iškraipo daugiamačių duomenų struktūras;
- Siūlomos korekcijos palyginus paprastos – pakanka tarpusavio atstumus daugiamatėje erdvėje dauginti iš atitinkamų koregavimo koeficientų, skirtingų įvairiems matmenų skaičiams. Pateikta koregavimo koeficientų reikšmių lentelė, o taip pat aproksimuojančios priklausomybės keliems daugiamačių duomenų atvejams.

4

Vizuali žinių gavyba analizuojant fiziologinius duomenis

Šiame skyriuje pateikiamas eksperimentinis vizualios žinių gavybos metodologijos tyrimas, taikant metodologiją fiziologiniams duomenims.

Pagrindiniai skyriaus rezultatai paskelbti straipsniuose: (Bernatavičienė *et al.* 2005a), (Bernatavičienė *et al.* 2006a), (Bernatavičienė *et al.* 2006e), (Bernatavičienė *et al.* 2006f), (Bernatavičienė *et al.* 2007b).

4.1. Fiziologiniai duomenys

Žinių gavybos iš fiziologinių duomenų tikslas – įvertinti žmogaus sveikatos būklę ir jo galimybes sportuoti. Tai ypač aktualu kineziologams ir sporto medikams. Analizuojama aibė sudaryta iš trijų grupių vyrų duomenų:

- vyrai, sergantys išemine širdies liga (I grupė) (61 tiriamasis),
- ne sportininkai (II grupė) (110 tiriamųjų),
- profesionalūs sportininkai (vyrai) (III grupė) (161 tiriamasis).

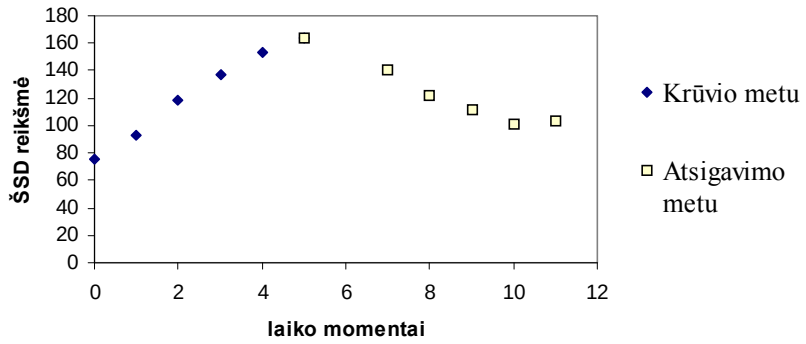
Šios duomenų aibės analizės tyrimai pateikti straipsniuose (Bernatavičienė *et al.* 2005a), (Bernatavičienė *et al.* 2006a), (Bernatavičienė *et al.* 2007b),

(Dzemyda *et al.* 2007). Šiam tyrimui buvo pasirinkti paprasčiausiai ir lengviausiai nustatomi elektrokardiogramos (EKG) bei arterinio kraujo spaudimo (AKS) dydžiai:

- širdies susitraukimų dažnis (ŠSD),
- sistolinis arterinio kraujo spaudimas (S),
- diastolinis arterinio kraujo spaudimas (D),
- intervalas elektrokardiogramoje nuo jungties taško J iki T bangos pabaigos (JT intervalas).

Šie keturi skaitiniai parametrai yra fiksuojami tam tikrais laiko momentais, atliekant ergometrinio dviračio tyrimą. Dar papildomai buvo apskaičiuoti du išvestiniai parametrai: (S-D)/S ir JT/RR (RR=60/ŠSD).

Prieš pradedant tyrimą, tiriamiesiems išmatuojami minėti 4 dydžiai: ŠSD, S, D, JT. Pradinis galingumas – 50 W. Tyrimo metu kas minutę galingumas didinamas po 50 W. Prieš kiekvieno galingumo padidinimą matuojami nurodyti keturi dydžiai. Tyrimas baigiamas, kai tiriamasis nepajėgia daugiau jo atlikti arba gydytojas pastebi žymius pakitimus širdies veikloje ir liepia baigti tyrimą. Po to seka organizmo atsigavimo periodas. Kas minutę vėl matuojami tie patys parametrai. Taigi, vieno tyrimo metu gaunamas kelių parametrų reikšmių rinkinys, be to, kiekvienam tiriamajam reikšmių skaičius skiriasi, kadangi jis priklauso nuo maksimalaus galingumo, kuriam esant tiriamasis pajėgė įveikti fizinį krūvį. 4.1 paveiksle pateikiamos vieno tiriamojo ŠSD parametro reikšmės matuotos tam tikrais laiko momentais didinant krūvį ir atsigavimo metu.



4.1 pav. *Širdies susitraukimų dažnio kitimas krūvio ir atsigavimo metu*

Toks reikšmių kitimas atspindi žmogaus širdies veiklą. Visos keturios fiziologinės charakteristikos (ŠSD, JT, S, D) yra svarbios, todėl būtina analizuoti

jų visumą. Tokia integrali analizė gydytojui gana sudėtingas uždavinys. Norint supaprastinti šį uždavinį, būtina rasti šios dinamikos integralius įverčius, kurie būtų lengviau suvokiami. Tam gali būti naudojamos įvairios strategijos, kuriomis remiantis kuriamos duomenų parametrų sistemos. Toliau disertacijoje pateikiami fraktalinės fiziologinių duomenų parametrų sistemos ir polinominės fiziologinių duomenų parametrų sistemos aprašymai.

4.2. Fiziologinių duomenų parametrų sistemos

Suformulavus tyrimui iškeltus uždavinius, apsibrėžus tikslus, kokias žinias norime gauti iš tiriamos aibės, turi būti suformuota parametrų sistema, bei sudaryta duomenų imtis tolimesniems tyrimams. Parametrų sistemos sudarymas ir imties formavimas apima **pirmą vizualios žinių gavybos metodologijos etapą**.

Buvo pasirinktos ir tirtos dvi parametrų sistemos: fraktalinė fiziologinių duomenų parametrų sistema ir polinominė parametrų sistema.

Fraktalinė fiziologinių duomenų parametrų sistema buvo pasiūlyta straipsnyje (Vainoras *et al.* 2005). Ji paremta fraktalinių dimensijų skaičiavimais.

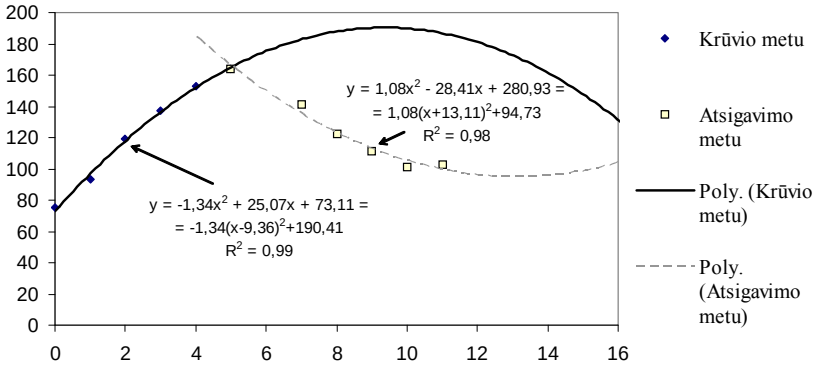
Fraktalinės dimensijos skaičiuojamos tokiu būdu:

1. Pradiniai duomenys – tai tyrimo metu gautos visų nagrinėjamų parametrų skaitinės reikšmės prie atitinkamų galingumų krūvio metu, o taip pat penkios reikšmės atsigavimo po krūvio metu. Gauti kiekvieno parametro taškiniai įverčiai yra interpoliuojami kubiniu splainu. Tokiu būdu gaunamas kiekvieno parametro funkcijos grafikas.
2. Fraktalinės dimensijos (užimtumo, informacinė, koreliacijos) skaičiuojama remiantis gautais grafikais pagal schemą:
   - nagrinėkime kvadratinį tinklelį (langelio dydis ε), uždėtą ant stebimo parametro funkcijos grafiko;
   - kiekvienoje tinklelio dalyje suskaičiuojamas į ją papuolusių taškų skaičius n_i. Jis dalinamas iš N bendro taškų skaičiaus $P_i(\varepsilon) = \frac{n_i}{N}$;
   - apibrėžkime informacinę funkciją $I \equiv -\sum_{i=1}^{N_\varepsilon} P_i(\varepsilon)\log[P_i(\varepsilon)]$, čia N_ε – užimtų langelių skaičius. Tuomet informacinė dimensija apibrėžiama taip $d_{\inf} \equiv -\lim_{\varepsilon\to 0}\frac{I}{\log(\varepsilon)} = \lim_{\varepsilon\to 0}\sum_{i=1}^{N_\varepsilon}\frac{P_i(\varepsilon)\log\left[P_i(\varepsilon)\right]}{\log(\varepsilon)}$. Įvedus pakeitimą

$I = \log N_\varepsilon$ gausime užimtumo dimensiją, o $I = \log \sum_i (P(\varepsilon))^2$ gausime koreliacijos dimensiją.

Apie fraktalinių dimensijų taikymą laiko eilutėms detaliau aprašyta (Higuchi 1988). Taigi, bendras parametrų skaičius yra 18: po 6 parametrus kiekvienai dimensijai. Yra nustatyta, kad informatyviausia yra informacinė dimensija, todėl daugiamačiai duomenys sudaryti būtent iš informacinės dimensijos 6 parametrų $(n = 6)$.

Polinominė fiziologinių duomenų parametrų sistema. Kitas naujas pasiūlytas fiziologinių duomenų parametrizavimo būdas – išskaičiuoti taškus aproksimuojančių kreivių parametrus (4.2 paveikslas). Pradžioje reikia parinkti aproksimavimo kreivės tipą. Šiuo atveju, eksponentė arba logaritminė kreivė netinka, kadangi aproksimuojamų taškų nėra daug ir nėra nusistovėjimo tendencijos. Paprasčiausias atvejis – aproksimavimas tiese, tačiau tai nėra tikslu. Antro laipsnio polinomo antra išvestinė parodo, kaip kreivė išlinkusi lyginant su tiese.



4.2 pav. *Širdies susitraukimų dažnio (ŠSD) kitimo krūvio ir atsigavimo metu aproksimavimas antro laipsnio polinomais*

Atlikus eksperimentinę analizę nustatyta, kad taškus aproksimuojant antro laipsnio polinomu $y = ax^2 + bx + c$, daugeliu atveju $R^2 > 0{,}9$ (R^2 yra aproksimavimo kokybės matas). Antrojo laipsnio polinomą galima suvesti į pilno kvadrato formą, t. y., $y = ax^2 + bx + c = a\left(x + \frac{b}{2a}\right)^2 + \left(c - \frac{b^2}{4a}\right)$, čia a –

parabolės išlinkio koeficientas (antra išvestinė), (*d*, *e*) – parabolės viršūnės koordinatės ($d = -\frac{b}{2a}$, $e = c - \frac{b^2}{4a}$).

Gana tiksliai matuotų fiziologinių charakteristikų reikšmių kitimą krūvio ir atsigavimo metu aprašo šie parametrai: fiziologinės charakteristikos reikšmė pradinėje būklėje (prieš tyrimą); charakteristikos reikšmė, esant maksimaliam krūviui; krūvio ir atsigavimo metu gautas reikšmes aproksimuojančių antro laipsnio polinomų koeficientai *a* (parabolių išlinkio koeficientai). Parabolių koeficientus *d* ir *e* tikslinga atmesti, kadangi jie nėra reikšmingi fiziologinių charakteristikų reikšmių kitimo aprašymui (nustatymui). Minėti parametrai nustatomi visoms keturioms fiziologinėms charakteristikoms (ŠSD, JT, S, D). Prie šių parametrų sistemos prijungiamas dar vienas parametras – maksimalus galingumas.

Sudaroma tokia parametrų sistema:

x_1 – maksimalus galingumas;

x_2 – krūvio metu išmatuotas ŠSD reikšmes aproksimuojančio antro laipsnio polinomo koeficientas *a*;

x_3 – parametro ŠSD reikšmė pradinėje būsenoje (prieš krūvį);

x_4 – parametro ŠSD reikšmė esant maksimaliam krūviui;

x_5 – atsigavimo metu išmatuotas ŠSD reikšmes aproksimuojančio antro laipsnio polinomo koeficientas *a*;

x_6, x_7, x_8, x_9 – atitinkami S parametrai;

x_{10}, x_{11}, x_{12}, x_{13} – atitinkami D parametrai;

x_{14}, x_{15}, x_{16}, x_{17} – atitinkami JT parametrai.

Šie parametrai apskaičiuojami visiems tiriamiesiems. Sudaryti daugiamačiai vektoriai $X_1, X_2, ..., X_m$, ($X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $i = 1, ..., m$), čia $m = 222$, $n = 17$. Kadangi parametrai yra iš skirtingų skalių būtina juos sunormuoti intervale [0; 1]. Turint daugiamačius vektorius juos galime analizuoti įprastais daugiamačių duomenų analizės metodais.

4.3. Fiziologinių duomenų parametrų sistemų lyginamoji analizė

Šiame skyriuje palyginami klasifikavimo rezultatai, naudojant dviejų skirtingų parametrų sistemų fiziologinius duomenis. Šis tyrimas apima **antrą vizualios žinių gavybos metodologijos etapą**.

4.1 lentelėje pateikiami klasifikavimo rezultatai, gauti klasifikuojant fraktalinės parametrų sistemos duomenis (kiekvienai fraktalinei dimensijai atskirai). Sukurtų klasifikatorių testavimui naudota kryžminio patikrinimo (angl. *cross-validation*) strategija. Buvo tiriamos dvi žmonių grupės: sportuojantys vyrai, ir vyrai, turintys išeminę širdies ligą.

4.1 lentelė. *Klasifikavimo rezultatai, gauti klasifikuojant fraktalinės parametrų sistemos duomenis (kiekvienai fraktalinei dimensijai atskirai)*

Klasifikatorius	Informacinė dim.			Koreliacinė dim.			Užimtumo dim.		
	Tiksl.	Spec.	Jautr.	Tiksl.	Spec.	Jautr.	Tiksl.	Spec.	Jautr.
NB	0,8215	0,8696	0,6984	0,8038	0,8491	0,6508	0,8265	0,8820	0,6825
kNN	0,8792	0,9441	0,7143	0,8034	0,8509	0,6825	0,8968	0,9317	**0,8095**
CT	0,8844	0,9255	0,7778	0,8536	0,9006	0,7302	0,8573	0,9068	0,7302
SVM	**0,9156**	**0,9689**	0,7778	0,8310	0,9255	0,5873	0,8970	0,9379	**0,7937**

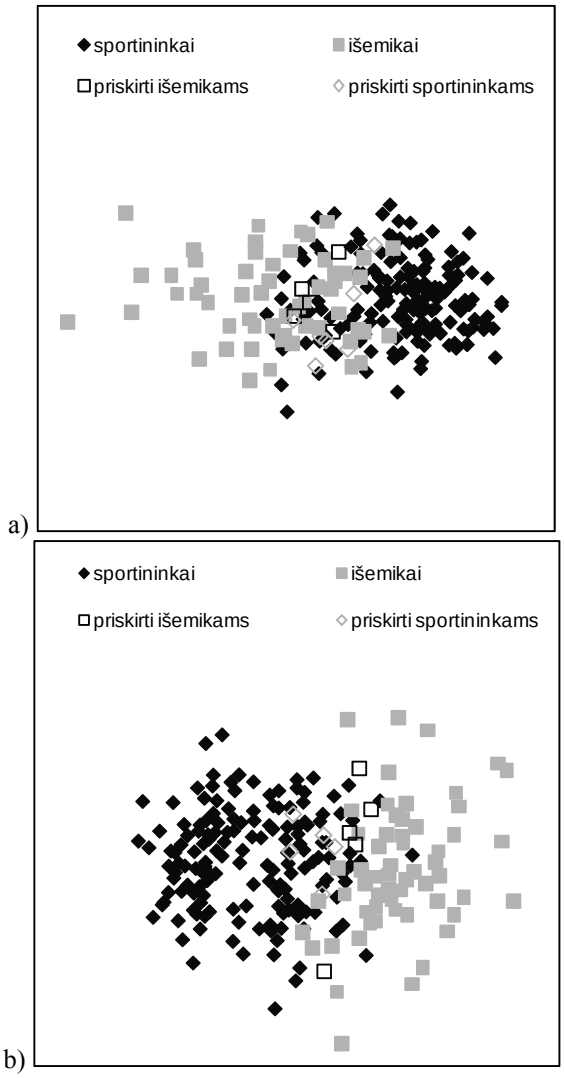
Vertintas bendras klasifikavimo tikslumas (angl. *classification accuracy*), jautrumas (angl. *sensitivity*) ir specifiškumas (angl. *specificity*). Šie vertinimo matai aprašyti 2.5.5. skyriuje. Analizuojant medicininius duomenis svarbiausias yra jautrumo matas, nes jis parodo kiek sergančių žmonių priskiriama sergantiems. Kuo didesnis jautrumo matas, tuo didesnė tikimybė identifikuoti sergantį žmogų. Geriausi gauti klasifikavimo rezultatai (4.1 lentelė) paryškinti juodai. Buvo naudoti artimiausių kaimynų (kNN), paprastasis Bayeso (NB), Klasifikavimo medžio (CT) ir atraminių vektorių klasifikatoriai (SVM). Plačiau apie šiuos klasifikatorius pateikta 2.5. skyriuje. Geriausias jautrumo matas gautas klasifikuojant užimtumo dimensijos daugiamačius duomenis artimiausių kaimynų (kNN) bei atraminių vektorių (SVM) klasifikatoriais. Kadangi kNN klasifikatoriumi specifiškumo matas bei klasifikavimo tikslumas yra šiek tiek prastesni, tai vizualizuodami duomenis naudosimės SVM klasifikatoriumi gautais klasifikavimo rezultatais. SVM klasifikatorius klaidingai priskyrė 5 sportininkus išemikams ir 7 išemikus sportininkams.

4.3a paveiksle pateikiama daugiamačių duomenų (užimtumo dimensijos parametrų sistema, $n = 4$) projekcija, gauta daugiamačių skalių SMACOF algoritmu. Šiame paveiksle įvedami tokie žymėjimai:

- juodi pilnaviduriai rombai žymi vektorius, atitinkančius sportininkus ir SVM klasifikatorius juos taip pat priskyrė sportininkų klasei;
- pilki pilnaviduriai kvadratai žymi vektorius, atitinkančius išemikus ir SVM klasifikatorius juos taip pat priskyrė išemikų klasei
- pilki tuščiaviduriai rombai žymi vektorius, atitinkančius išemikus tačiau SVM klasifikatorius juos priskyrė sportininkų klasei;

- juodi tuščiaviduriai kvadratai žymi vektorius, atitinkančius sportininkus tačiau SVM klasifikatorius juos priskyrė išemikų klasei.

Kaip matyti 4.3a paveiksle, išemikų ir sportininkų klasės gana stipriai persidengia.



4.3 pav. *Daugiamačių duomenų projekcijos, gautos DS algoritmu: (a) užimtumo dimensijos parametrų sistema ($n = 4$); (b) polinominė parametrų sistema ($n = 17$)*

4.3b paveiksle pateikiama daugiamačių duomenų (polinominė parametrų sistema, $n=17$) projekcija, gauta tuo pačiu daugiamačių skalių SMACOF algoritmu. Lyginant 4.3a ir 4.3b paveikslus galima teigti, kad 4.3b paveiksle abi duomenų klasės mažiau persidengę.

Blogai suklasifikuoti vektoriai, kurie atitinka sportininkus (4.3b paveikslas, tuščiaviduriai kvadratai), yra toliau nuo daugumos taškų, atitinkančių sportininkus. Jie labiau susimaišę su taškais, atitinkančiais išemikus. Tai gali būti perspėjimas, kad yra abejonių dėl sportininko sveikatos, ir jį reiktų detaliau tirti.

Duomenys, sudaryti iš polinominės parametrų sistemos, buvo suklasifikuoti naudojant tuos pačius klasifikatorius. Gauti klasifikavimo rezultatai pateikiami 4.2 lentelėje. Palyginus klasifikavimo tikslumą 4.1 ir 4.2 lentelėse matyti, kad, naudojant polinominę parametrų sistemą, klasifikavimo tikslumas taip pat gaunamas šiek tiek geresnis. Geriausias klasifikavimo jautrumo matas gaunamas paprastuoju Bayeso ir SVM klasifikatoriais, o bendras klasifikavimo tikslumas ir specifiškumo matas SVM klasifikatoriumi. Tolimesniuose tyrimuose bus naudojamas SVM klasifikatorius. Naudojant polinominę parametrų sistemą, SVM klasifikatorius klaidingai priskyrė 5 sportininkus išemikams ir 5 išemikus sportininkams. Reiktų atkreipti dėmesį, kad pašalinus tuos 10 blogai suklasifikuotų taškų, abi duomenų klasės persidengia visai nežymiai. Todėl reiktų patikrinti tų duomenų korektiškumą, bei atkreiti dėmesį į tuos tiriamuosius, kuriuos atitinka tie vektoriai, išsiaiškinti, kodėl jų duomenys tokie panašūs į priešingos klasės duomenis.

4.2 lentelė. *Klasifikavimo rezultatai, gauti klasifikuojant polinominės parametrų sistemos duomenis (n=17)*

Klasifikatorius	Tiksl.	Spec.	Jautr.
NB	0,9152	0,9441	**0,8413**
KNN	0,8840	0,9503	0,7143
CT	0,8887	0,9503	0,7302
SVM	0,9241	0,9627	**0,8254**

Taigi, palyginus rezultaus, gautus klasifikavimo ir vizualizavimo metodais, nustatyta, kad polinominė parametrų sistema tinkamesnė tolimesniems tyrimams. Todėl, toliau tiriami duomenys, sudaryti naudojant šią parametrų sistemą.

4.4. Parametrų įvertinimas polinominėje parametrų sistemoje

Trečias ir ketvirtas vizualios žinių gavybos metodologijos etapai skirti parametrų sistemos tikslinimui. Šiame skyriuje įvertinamas parametrų

reikšmingumas, bei atmetami nereikšminiai parametrai naudojant klasifikavimo ir vizualizavimo metodus.

Palyginus kelias pasiūlytas parametrų sistemas fiziologiniams duomenis, nustatyta, kad naudojant polinominę parametrų sistemą gaunami geresni klasifikavimo rezultatai nei naudojant fraktalinę parametrų sistemą. Todėl tolimesniuose tyrimuose naudosime tik polinominę parametrų sistemą.

Vizualizuojant 17-mačius duomenis, jie spaudžiami į dvimatę erdvę, tokiu atveju duomenų iškraipymai neišvengiami. Todėl tikslinga detaliau ištirti šią parametrų sistemą, įvertinti parametrų svarbą, gal būt, kai kurių atsisakyti.

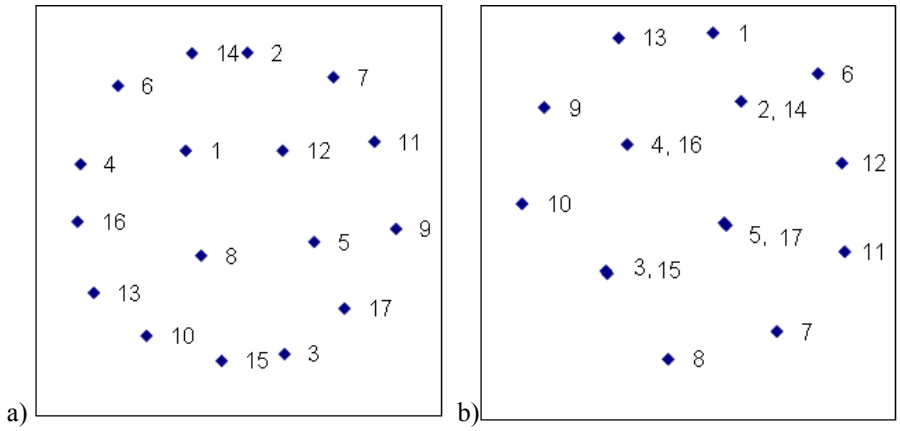
Mūsų tiriamos aibės vektoriai $X_1, X_2, ..., X_m$. Kiekvienas vektorius sudarytas iš parametrų $x_1, x_2, ..., x_n$ ($X_i = (x_{i1}, x_{i2}, ..., x_{in})$, čia n – parametrų (kintamųjų) skaičius, m – analizuojamų vektorių skaičius). Vieno parametro reikšmės gali priklausyti nuo kitų parametrų reikšmių, t. y., jie gali būti koreliuoti. Kyla problema, kaip nustatyti, kokie parametrai turi būti analizuojami kartu, o kuriuos verta atmesti, t. y., analizuojamus vektorius sudaryti tik iš dalies parametrų $x_1, x_2, ..., x_n$. Todėl tikslinga nagrinėti vektorius sudarančių parametrų sistemą.

Vienas iš parametrų sistemos analizės būdų – nagrinėti parametrų sistemą jų koreliacinės analizės pagrindu. Tikslas – stebėti parametrų išsidėstymą plokštumoje. Šis metodas vadinamas koreliacinių matricų vizualia analize, pasiūlytas (Dzemyda 2001) ir eksperimentiškai pagrįstas darbuose (Dzemyda 2005), (Dzemyda *et al.* 2007). Pirmiausia yra apskaičiuojama duomenų aibės $X_1, X_2, ..., X_m$ parametrų koreliacinė matrica $R = \{r_{x_i x_j}, i, j = 1, ..., n\}$. Čia $r_{x_i x_j}$ yra parametrų x_i ir x_j koreliacijos koeficientai. Daugiamačiai vektoriai $Y_1, Y_2, ..., Y_m \in S^n$, atitinkantys analizuojamų parametrų aibę $x_1, x_2, ..., x_n$ (šiuo atveju, $n = 17$), gauti apskaičiavus koreliacijos matricą. S^n yra n-matės Euklidinės erdvės poaibis R^n, sudarytas iš vienetinio ilgio n-mačių vektorių, t. y., S^n yra vienetinė hipersfera, jei $Y \in S^n$, tai $\| Y \| = 1$. $Y_1, Y_2, ..., Y_{17}$ vektorių aibė gali būti suprojektuota į plokštumą, tai leis mums vizualiai įvertinti parametrų $x_1, x_2, ..., x_{17}$ išsidėstymą plokštumoje. 4.4a paveiksle pateikta septyniolikos 17-mačių vektorių projekcija, gauta Sammono algoritmu. Skaičiai virš taškų paveiksle žymi parametrų $x_1, x_2, ..., x_{17}$ numerį. Iš 4.4a paveikslo matyti, kad taškai, atitinkantys $x_1, x_2, ..., x_{17}$ parametrus, plokštumoje pasiskirstę tolygiai ir sunku išskirti kokias nors parametrų grupes.

Toliau pateiksime, kaip daugiamačiai vektoriai $Y_1, Y_2, ..., Y_{17}$, atitinkantys parametrus $x_1, x_2, ..., x_{17}$, išsidėsto įvairaus dydžio SOM tinkle. Mažame SOM[4x4] tinkle vektoriai formuoja kelias grupes po 2 – 3 vektorius

(4.3a lentelė), didinant tinklo dydį kai kurios grupės persigrupuoja, kai kurios išlieka nepakitę. SOM[7x7] tinkle (4.3d lentelė) greta lieka tik dvi parametrų poros (2,14), (4,16).

Dabar vizualiai įvertinsime, kiek vektorių esančių kaimynėse celėse yra greta ir *n*-matėje erdvėje. Tam tikslui buvo panaudotas SOM tinklo ir Sammono projekcijos integruoto junginio algoritmas. 4.4b paveiksle pateikta SOM[7x7] tinklo vektorių nugalėtojų projekcija plokštumoje. Iš gauto vaizdo matyti kad greta yra šios parametrų poros (2, 14), (3, 15), (4, 16) ir (5, 17), nepaisant to, kad parametrai 3 ir 15, 5 ir 17 nepriklauso tai pačiai celei (4.3d lentelė), vektoriai nugalėtojai atitinkantys tuos vektorius yra visai šalia vienas kito (4.4b paveikslas).



4.4 pav. *Parametrų projekcija plokštumoje, gauta Sammono algoritmu (a), SOM tinklo ir Sammono projekcijos integruoto junginio algoritmu (b)*

Analizuojamos aibės apraše (4.1 skyriuje) pateikta, kad $x_2 - x_5$ yra širdies dažnio (ŠSD) parametrai, o $x_{14} - x_{17}$ yra atitinkami JT intervalo parametrai. Taigi, atlikus parametrų sistemos analizę galima teigti, kad ŠSD ir JT parametrai yra priklausomi, t. y. egzistuoja stipri koreliacija tarp atitinkamų parametrų. Iš to seka, kad vieną iš minėtų parametrų grupę galima eliminuoti. Sprendžiama problema supaprastėja, kadangi analizuojamų duomenų dimensija sumažėja nuo 17 iki 13. Taigi, dabar galima analizuoti vektorius sudarytus iš $x_1, x_2, ..., x_{13}$ parametrų, vienu atveju be JT parametrų, kitu vektorius $x_1, x_6, x_7 ..., x_{17}$ – be ŠSD parametrų. Medicinoje ryšys tarp ŠSD ir JT intervalo taip pat gerai žinomas (Bazett formulė), atliktas tyrimas patvirtina šį faktą. ŠSD geriau atspindi reguliacinės sistemos funkcijas, o JT intervalas labiau susijęs su miokardo metanolinėmis savybėmis. Žinant tai, tikslingiau atsisakyti ŠSD parametrų.

4.3 lentelė. *SOM tinklas (a) [4x4]; (b) [5x5]; (c) [6x6]; (d) [7x7]*

a)

2, 14	6, 10		8, 9
			4, 16
1, 13			
7, 11, 12		5, 17	3, 15

b)

2, 14		1, 10		3, 15
6, 12		7, 11		5, 17
4, 16		8, 9		13

c)

4, 16		8		15	3
		9			
1					17
7			6		5
11, 12		13	10		2, 14

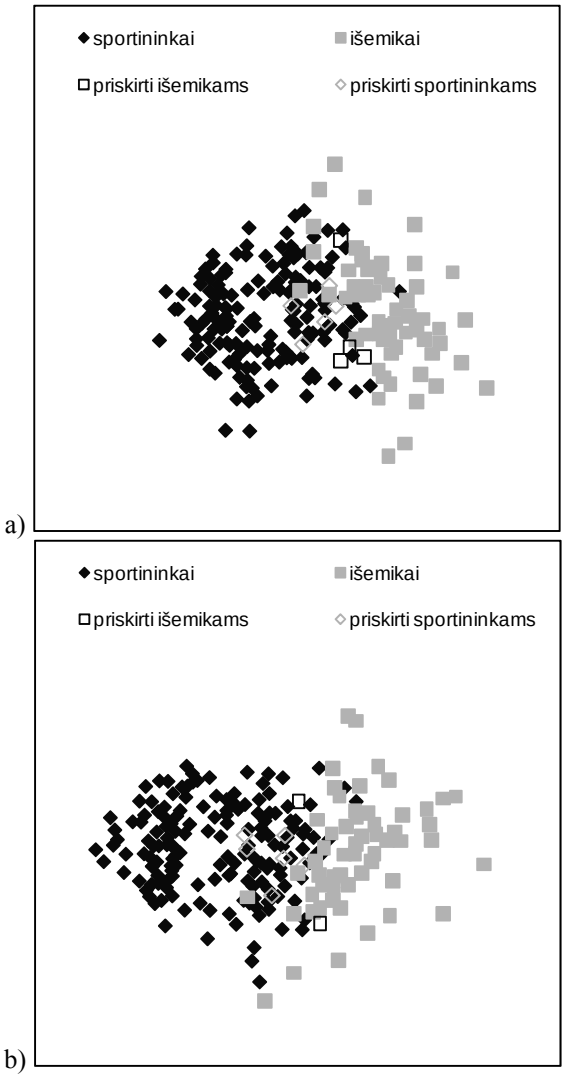
d)

4, 16		15	3		5	17
8						10
		9		13		
7						6
		12				
11				1		2, 14

4.4 lentelėje pateikti klasifikavimo rezultatai, gauti klasifikuojant duomenų aibę, kuri sudaryta iš parametrų, atmetus JT parametrus arba ŠSD parametrus. Geresnis klasifikavimo rezultatas gaunamas klasifikuojant duomenų aibę, kuri sudaryta iš parametrų, atmetus ŠSD parametrus: SVM klasifikatorius 2 sportininkus klaidingai priskyrė išemikams ir 6 išemikus priskyrė sportininkams.

4.4 lentelė. *Klasifikavimo rezultatai gauti klasifikuojant duomenų aibę, kuri sudaryta atmetus JT parametrus arba ŠSD parametrus*

klasifikatorius	**atmetus JT parametrus**			**atmetus ŠSD parametrus**		
	Tiksl.	**Spec.**	**Jautr.**	**Tiksl.**	**Spec.**	**Jautr.**
NB	0,9152	0,9503	0,8254	**0,9415**	0,9627	**0,8889**
kNN	0,8881	0,9379	0,7619	0,8666	0,9376	0,6825
CT	0,8978	0,9503	0,7619	0,8978	0,9503	0,7619
SVM	0,9113	0,9627	0,7778	0,9283	**0,9689**	0,8254



4.5 pav. *Duomenų aibių, kurios sudarytos iš parametrų (a) atmetus JT parametrus (b) atmetus ŠSD parametrus,projekcijos*

Klasifikuojant duomenų aibę, kuri sudaryta iš parametrų atmetus JT parametrus, gautas šiek tiek prastesnis rezultatas: SVM klasifikatorius klaidingai 4 sportininkus priskyrė išemikams ir 5 išemikus priskyrė sportininkams.

4.5 paveiksle pateiktos duomenų aibių, kurios sudarytos iš parametrų, atmetus JT parametrus arba ŠSD parametrus, projekcijos. Čia buvo naudotas DS SMACOF algoritmas. 4.5a ir 4.5b paveiksluose žymėjimai tokie patys, kaip ir 4.3 paveiksle. Lyginant 17-mačių ir 13-mačių vektorių projekcijas galima teigti, kad

gauti vaizdai iš esmės nesiskiria, nors 13-mačių vektorių (atmetus ŠSD parametrus) klasifikavimo kokybė šiek tiek pagerėjo.

4.5. Preliminarus sveikatos būklės įvertinimas naudojant pasiūlytą metodą

Šio skyriaus tyrimai apima **penktą – aštuntą vizualios žinių gavybos metodologijos etapus** (žr. 3.1.skyrių).

Norint nustatyti naujam pacientui preliminarią medicininę diagnozę, buvo pasiūlyta surasti taško, atitinkančio naują pacientą, padėtį tarp jau atvaizduotų taškų, atitinkančių jau ištirtus pacientus. Sprendimas apie sveikatos būklę yra priimamas nustačius taško, atitinkančio tiriamąjį pacientą, vietą skiriamojo paviršiaus atžvilgiu. Šiam tikslui yra naudojami įvairūs metodai (Basalaj 1999), (Kraaijvel and Mao 1995), (Naud and Duch 2000), (Pekalska *et al.* 1999), (Tiping 1996). Šiuose eksperimentuose mes naudosime santykinių DS metodą (Naud and Duch 2000). Priklausomai nuo taško, atitinkančio tiriamąjį pacientą vietos, naują pacientą galime priskirti vienai iš žinomų klasių arba naudoti papildomus metodus tiksliai naujai paciento klasei nustatyti.

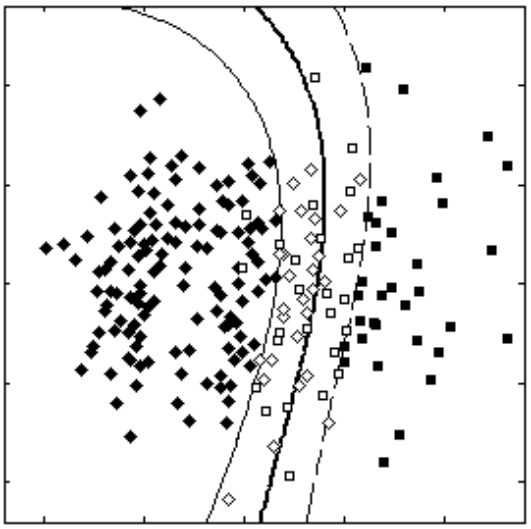
4.5 lentelė. *Klasifikavimo rezultatai*

erdvė	Tiksl.	Spec.	Jautr.
R^{17}	0,925	0,967	0,817
R^{2}	0,902	0,954	0,767

Skiriamąjį paviršių daugiamatėje erdvėje vizualiai įsivaizduoti neįmanoma, todėl skiriamieji paviršiai yra braižomi projekcijos duomenims. Kaip matyti iš 4.5 lentelėje pateiktų fiziologinių duomenų klasifikavimo rezultatų 17-matėje ir 2-matėje erdvėje (daugiamačių duomenų projekcijos) skirtumas yra nežymus. Dvimačiai vektoriai gauti naudojant daugiamačių skalių SMACOF algoritmą. 4.5 lentelėje pateikti klasifikavimo rezultatai gauti su nepilna duomenų aibe, keli vektoriai buvo eliminuoti iš mokymo duomenų aibės ir interpretuojami kaip nauji taškai. Klasifikavimo rezultatai ir skiriamieji paviršiai gauti naudojant atraminių vektorių klasifikatorių, naudojant "SVM Toolbox for Matlab" (Scwaighofer 2002). Čia Kernel funkcija k yra radialinė bazinė funkcija $k(X_i, X_j) = \exp(-\| X_i - X_j \|^2 / d\gamma)$, čia d analizuojamų duomenų dimensija (šiuo atveju $d = 2$), γ – parametras, parenkamas eksperimentiškai. Šiuo atveju $\gamma = 2$. Klasifikuojant dvimačius duomenis, SVM klasifikatorius klaidingai priskyrė 6 sportininkus išemikams ir 13 išemikų sportininkams.

4.6 paveiksle pateiktos duomenų projekcijos, klasių skiriamieji paviršiai ir atraminiai vektoriai:

- taškai, atitinkantys išemikus, pažymėti juodais kvadratais;
- taškai, atitinkantys sportininkus, pažymėti juodais rombais;
- atraminiai vektoriai pažymėti tuščiaviduriais rombais ir kvadratais (viso 55);
- paryškinta linija žymi skiriamąjį paviršių, plona linija žymi sportininkų skiriamąjį paviršių, punktyrinė linija žymi išemikų skiriamąjį paviršių.

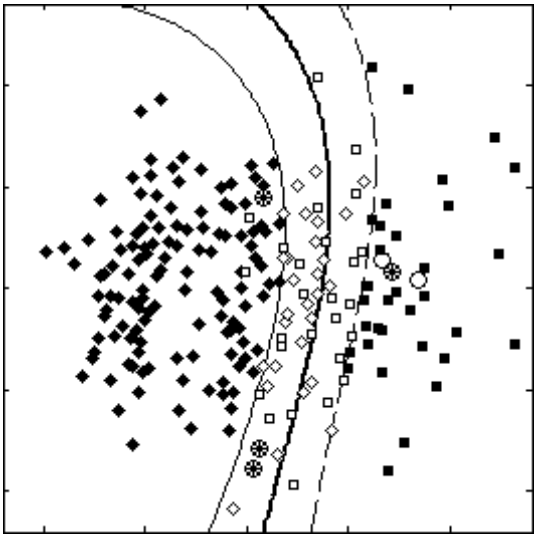


4.6 pav. *Fiziologinių duomenų projekcijos, klasių skiriamieji paviršiai ir atraminiai vektoriai*

Siekiant preliminariai nustatyti, ar naujas tiriamasis yra išemikas, reikia jį atitinkantį vektorių suprojektuoti į plokštumą, kur jau yra atidėti žinomų klasių atstovai. Jei taškas, atitinkantis tiriamąjį pacientą, patenka į „išemikų sritį“, galime įtarti, kad pacientas turi širdies veiklos sutrikimų, nes jo duomenys panašūs į išemine liga sergančių tiriamųjų. Jei taškas, atitinkantis tiriamąjį pacientą, patenka į „sportininkų sritį“, galime teigti, kad pacientas neturi širdies veiklos sutrikimų, nes jo duomenys panašūs į sportininkų. Jei taškas, atitinkantis tiriamąjį pacientą, patenka tarp plonos ir punktyrinės linijų, šį tiriamąjį reikia atidžiau ištirti.

Santykinių DS metodas gali būti naudojamas naujų taškų atidėjimui ant fiksuotos projekcijos. 4.7 paveiksle nauji taškai, atitinkantys išemikus, pažymėti

baltais skrituliukais, o nauji taškai, atitinkantys sportininkus, pažymėti užpildytais skrituliukais, yra atvaizduoti ant fiksuotos projekcijos. Galima teigti, kad vienas sportininkas neturi širdies veiklos sutrikimų, kadangi taškas, atitinkantis šį sportininką, patenka į „sportininkų sritį“. Du taškai, atitinkantys sportininkus, patenka į „sportininkų sritį“ tačiau jie yra tarp paryškintos linijos ir plonos linijos, taigi šie sportininkai turi būti atidžiau ištirti. Vienas taškas, atitinkantis sportininką, patenka į „išemikų sritį“. Gydytojai patvirtino, kad rimti sutrikimai yra pastebėti šio sportininko širdies veikloje ir jam neleidžiama toliau sportuoti.



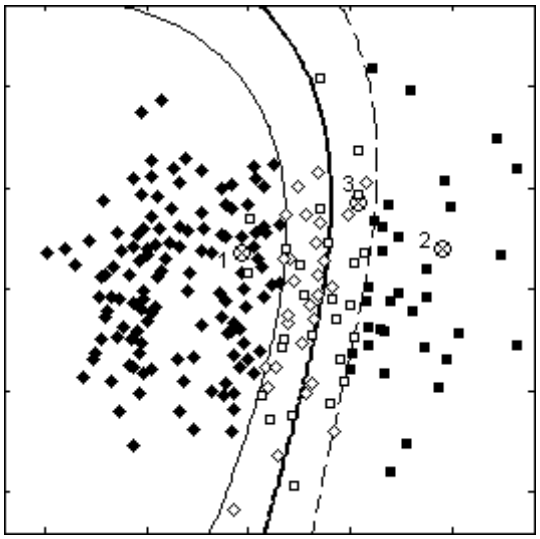
4.7 pav. *Naujų taškų, atitinkančių išemikus ir sportininkus projekcijos tarp fiksuotų taškų*

4.8 paveiksle taškai, pažymėti apibrėžtais kryžiukais, atitinka pacientus, kuriems diagnozė dar nenustatyta. Naudojant pasiūlytą metodą, galima daryti preliminarią diagnozę pacientams:

- pirmas pacientas (nr. 1) yra sveikas ir gali sportuoti, nes taškas, atitinkantis tiriamąjį pacientą, patenka į „sportininkų sritį“ (kairiau paryškintos ir plonos linijos);
- išemine širdies liga yra įtariama antram pacientui (nr. 2), nes taškas, atitinkantis šį pacientą, patenka į „išemikų sritį“ (dešiniau paryškintos ir punktyrinės linijos);

- trečiam pacientui (nr. 3) gali būti įtartas širdies veiklos sutrikimas, nes taškas, atitinkantis šį pacientą, yra tarp paryškintos linijos ir punktyrinės linijos; šis pacientas turėtų būti detaliau ištirtas.

Taigi, norint nustatyti preliminarią diagnozę naujam pacientui, yra siūloma paciento duomenis projektuoti į plokštumą, kur jau yra fiksuota etaloninė bazinių vektorių projekcija ir nustatyti jo padėtį tarp esamų taškų. Priklausomai nuo taško, atitinkančio tiriamąjį pacientą, padėties tarp skiriamųjų paviršių galime daryti preliminarų sprendimą apie jo sveikatos būklę.



4.8 pav. *Naujų taškų, atitinkančių naujus tiriamuosius, projekcijos tarp fiksuotų taškų*

Preliminarios diagnozės sprendimų priėmimo sistemos schema

Eksperimentų rezultatai parodė, kad preliminari diagnozė gali būti nustatoma naudojantis tokia schema:

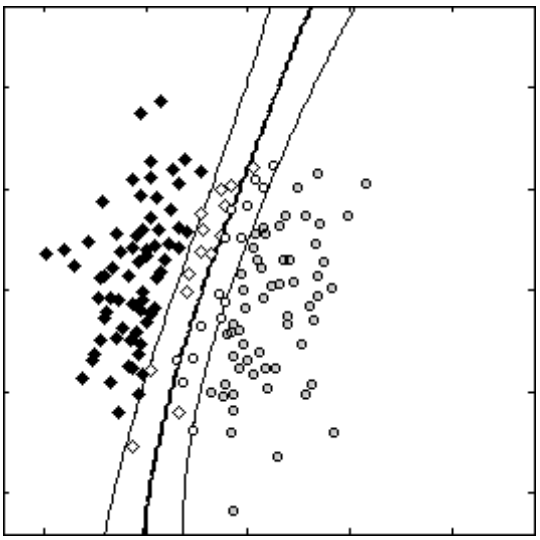
1. Sudaroma analizuojamos aibės parametrų sistema ir suformuojama turimų duomenų imtis.
2. Atliekama detali duomenų aibės analizė naudojant klasifikavimo ir vizualizavimo rezultatus.
3. Remiantis gautais rezultatais suformuojama bazinių vektorių aibė; bazinių vektorių klasės yra žinomos, klasifikatoriai juos priskyrė toms klasėms ir vizuali analizė tai patvirtino, ekspertas patvirtino tų duomenų korektiškumą.

4. Baziniai vektoriai projektuojami į plokštumą ir braižomi klasių skiriamieji paviršiai, naudojant vieną iš žinomų klasifikavimo metodų.
5. Nauji taškai, atitinkantys tiriamuosius, kurių diagnozė nenustatyta, projektuojami į plokštumą atsižvelgiant į fiksuotą bazinių vektorių projekciją.
6. Priklausomai nuo taško, atitinkančio tiriamąjį pacientą, padėties tarp skiriamųjų paviršių galime daryti preliminarų sprendimą apie jo sveikatos būklę.

Eksperimentai buvo atlikti naudojant fiziologinių duomenų aibę, tačiau pasiūlytą algoritmą galima taikyti bet kokių medicininių duomenų analizei, siekiant nustatyti preliminarią diagnozę.

Sportininkų aibės tyrimas

Atliekant fiziologinių duomenų analizę buvo pastebėta, kad sportininkų duomenys sudaro dvi atskiras grupes, kurios beveik nepersidengia.



4.9 pav. *Sportininkų duomenų projekcijos, klasių skiriamieji paviršiai ir atraminiai vektoriai*

Buvo atlikti šie eksperimentai:

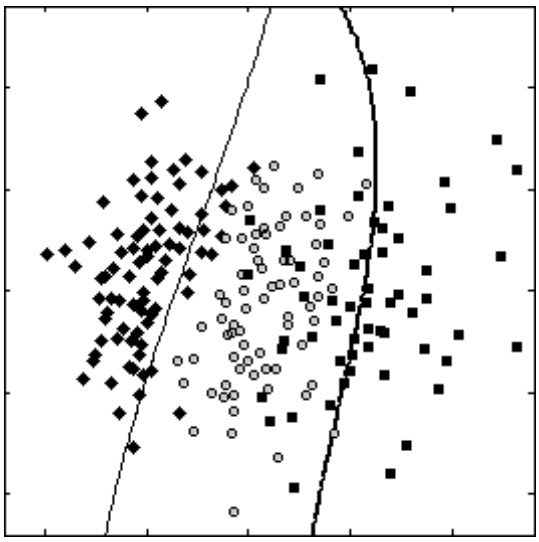
(1) Sportininkų duomenų aibė buvo suklasterizuota į dvi grupes naudojant standartinį *k*-vidurkių klasterizavimo algoritmą (Dunham 2003);

(2) Atsižvelgiant į klasterizavimo rezultatus, tie duomenys padalinti į dvi klases (sužymėti);

(3) Naudojant daugiamačių skalių SMACOF algoritmą, daugiamačiai duomenys suprojektuoti į plokštumą;

(4) Projekcijos duomenys suklasifikuojami naudojant SVM klasifikatorių ir braižomi skiriamieji paviršiai (4.9 paveikslas).

SVM klasifikatoriumi duomenys suklasifikuoti pakankamai tiksliai, tik trys vektoriai priskirti klaidingoms klasėms. Gauti rezultatai rodo, kad šių dviejų sportininkų grupių sveikatos būklė tikrai skiriasi.



4.10 pav. *Fiziologinių duomenų projekcija ir skiriamieji paviršiai (trys klasės: išemikų klasė ir dvi sportininkų klasės)*

4.10 paveiksle pateikiama visų fiziologinių duomenų projekcija (išskirtos abi sportininkų grupės) ir skiriamieji paviršiai:

- taškai, atitinkantys pirmos grupės sportininkus, pažymėti juodais rombais;
- taškai, atitinkantys antros grupės sportininkus, pažymėti pilkais apskritimais;
- taškai, atitinkantys išemikus, pažymėti juodais kvadratais;
- ryškia kreive žymimas išemikų skiriamasis paviršius, plona linija – abiejų sportininkų grupių skiriamasis paviršius.

Kaip matome 4.10 paveiksle, išemikų klasė persidengia tik su antrąja sportininkų klase. Gauti rezultatai taip pat patvirtina prielaidą, kad antros sportininkų grupės duomenys panašus į išemine širdies liga sergančiųjų

duomenis. Gal būt šie sportininkai per daug sportuoja ir viršija savo sveikatos ribas. Ši sportininkų grupė turėtų būti priskirta rizikos grupei, o šie sportininkai detaliau ištirti.

4.6. Ketvirtojo skyriaus rezultatai ir išvados

Disertacijoje buvo analizuojami fiziologiniai žmogaus sveikatos būklę nusakantys duomenys, remiantis sukurta vizualia žinių gavybos metodologija. Analizuoti profesionalūs sportininkai ir vyrai, sergantys išemine širdies liga. Ergometrinio dviračio testo metu matuojami 4 parametrai tam tikrais laiko momentais krūvio metu ir po krūvio sekančio atsigavimo metu. Daryti diagnostines išvadas iš tokių dinamiškai besikeičiančių duomenų yra gana sunku. Todėl tikslinga ieškoti tam tikro integralaus jų įverčio.

Buvo ištirtos dvi fiziologinių duomenų parametrizavimo sistemos: fraktalinių dimensijų parametrizavimo sistema ir polinominio aproksimavimo parametrų sistema. Tyrimą sudarė keturi etapai:

(1) atlikta dviejų pasiūlytų parametrų sistemų lyginamoji analizė naudojant kelis skirtingus duomenų gavybos metodus: klasterizavimo, klasifikavimo metodai, vizualizavimo metodai, bei jų junginiai;

(2) atlikta polinominės parametrų sistemos analizė paremta koreliacine bei vizualia analize;

(3) suklasifikuoti tiriami duomenys ir nubrėžti klasių skiriamieji paviršiai;

(4) nauji duomenys atidedami tarp skiriamųjų paviršių ir priklausomai nuo jų padėties tarp paviršių atliekama preliminari diagnozė.

Vizualizuojant fraktalinės dimensijos duomenis matomas gana ryškus grupių persidengimas. Geriausias klasifikavimo rezultatas gautas klasifikuojant užimtumo dimensijos duomenis. Atlikus šių duomenų klasifikavimą nustatyta, kad tiksliausiai klasifikuoja atraminių vektorių (SVM) klasifikatorius (bendras tikslumas $\approx 90\,\%$, jautrumas $\approx 80\,\%$). Neteisingai klasifikuojami 12 taškų iš 222.

Vizualizuojant polinominės parametrų sistemos duomenis, grupių persidengimas yra, bet ne toks žymus kaip vizualizuojant vektorius, sudarytus iš fraktalinės dimensijos parametrų. Atlikus šių duomenų klasifikavimą, tiksliausias yra taip pat atraminių vektorių klasifikatorius (bendras tikslumas $\approx 92\,\%$, jautrumas $\approx 83\,\%$). Neteisingai klasifikuoja 10 taškų iš 222. Lyginant rezultatus, kai klasifikuoti vektoriai, sudaryti iš fraktalinių dimensijų parametrų, klasifikavimo kokybė šiek tiek pagerėjo.

Šiame darbe taip pat atlikta polinominės parametrų sistemos analizė paremta koreliacine bei vizualia analize. Pradžioje apskaičiuojama parametrų koreliacinė matrica, iš jos atkuriama vektorių sistema. Stebimas gautų vektorių tarpusavio išsidėstymas plokštumoje. Naudojant SOM tinklo ir Sammono projekcijos junginį nustatyta, kad keli parametrai sudaro „tvirtas“ poras, t. y., jie yra arti vienas kito, todėl iš parametrų sistemos dalį parametrų galima atmesti:

a) atlikus analizę nustatyta, kad dydžio ŠSD parametrų grupė yra labai priklausoma nuo dydžio JT parametrų grupės, ir vieną parametrų grupę galima išmesti iš parametrų sistemos;

b) sumažinus parametrų skaičių nuo 17 iki 13, klasifikavimo kokybė beveik nepasikeičia. Geriausio klasifikatoriaus (SVM) tikslumas padidėja iki 93 %. Neteisingai klasifikuoja 9 taškus iš 222;

c) Vizualus 13-mačių duomenų projekcijos vaizdas nuo vektorių, sudarytų iš 17 parametrų, projekcijos vaizdo beveik nesiskiria, grupės vizualiai atrodo panašiai.

Kita sprendžiama problema: klasių skiriamojo paviršiaus nubraižymas. Jis vaizdžiai gali matytis tik dvimatėje erdvėje. Atlikus lyginamąją analizę klasifikuojant daugiamačius ir juos atitinkančius dvimačius duomenis, nustatyta, kad klasifikavimo kokybė pakinta nežymiai, todėl skiriamuosius paviršius galima braižyti dvimačiams vektoriams.

Norint nustatyti preliminarią diagnozę naujam pacientui yra siūloma paciento duomenis projektuoti į plokštumą, kur jau yra fiksuota etaloninė bazinių vektorių projekcija (suprojektuoti tiriamųjų duomenys su jau nustatytomis diagnozėmis, nubrėžti klasių skiriamieji paviršiai) ir nustatyti naujo taško padėtį tarp esamų taškų. Priklausomai nuo taško, atitinkančio tiriamąjį pacientą, padėties tarp skiriamųjų paviršių galime daryti preliminarų sprendimą apie šio tiriamojo sveikatos būklę. Pasiūlytas būdas pagelbėtų medikams vertinant pacientų sveikatos būklę, perspėtų apie besikeičiančią sportininko sveikatos būklę pradinėje stadijoje.

Tiriant polinominės parametrų sistemos duomenis nustatyta, kad sportininkų klasę sudaro dvi atskiros grupės. Galima daryti prielaidą, kad viena grupė yra tikrai sveikų sportininkų, o antros grupės sportininkams reiktų skirti didesnį dėmesį, nes šios grupės duomenys panašūs į išemine širdies liga sergančiųjų duomenis. Gal būt šie sportininkai per daug sportuoja ir viršija savo sveikatos galimybes. Ši sportininkų grupė turėtų būti priskirta rizikos grupei, o šie sportininkai detaliau ištirti.

Eksperimentų rezultatai parodė, kad preliminari diagnozė gali būti nustatoma naudojantis tokia schema:

1. Sukuriamas vizualus klasifikatorius:

(a) pasirenkama parametrų sistema, kuri remiasi žmogaus fiziologinėmis savybėmis ir apibūdina tiriamųjų sveikatos būklę;

(b) vertinant pasirinktus parametrus, suformuojama duomenų imtis;

(c) atliekama duomenų analizė (klasifikavimas, klasterizavimas, vizualizavimas), taip pat remiamasi jau žinoma informacija apie tiriamuosius, medikų nustatyta diagnoze.

2. Preliminari diagnozė naujam pacientui:

(a) nauji taškai, atitinkantys tiriamuosius, kurių diagnozė nenustatyta, projektuojami į plokštumą atsižvelgiant į fiksuotą bazinių vektorių projekciją;

(b) priklausomai nuo taško, atitinkančio tiriamąjį pacientą, padėties tarp skiriamųjų paviršių galime daryti preliminarų sprendimą apie jo sveikatos būklę.

Taip pat naudojant pasiūlytą schemą, galima stebėti ligos vystymosi dinamiką. Žinant tiriamojo duomenis, matuotus skirtingais gydymo momentais ir juos pavaizdavus šiuo būdu, vizualiai galima vertinti sveikatos pokytį.

Eksperimentai buvo atlikti naudojant fiziologinių duomenų aibę, tačiau pasiūlytą algoritmą galima taikyti bet kokių medicininių duomenų analizei siekiant nustatyti preliminarią diagnozę.

5

Bendrosios išvados ir rekomendacijos

1. Žinių gavybos vizualiais metodais proceso susisteminimas leidžia visapusiškai įvertinti ir pritaikyti vizualizavimo metodų ir priemonių teikiamas galimybes duomenų analizės efektyvumui didinti.

2. Detaliai ištyrus santykinių DS algoritmą, galime daryti šias išvadas:

   - Vizualizavimo rezultatai priklauso nuo bazinių vektorių parinkimo strategijos (kuo tolygiau baziniai vektoriai pasiskirstę po visą tiriamą aibę, tuo tikslesnė projekcija yra gaunama) ir bazinių vektorių skaičiaus, dvimačių vektorių inicializavimo būdo;
   - Naudojant inicializavimo būdą, paremtą PCA algoritmu, paklaidos vidurkis mažesnis už paklaidų vidurkius gaunamus kitomis strategijomis, tačiau skirtumai tarp šių vidurkių nereikšminiai. Blogiausias inicializavimo būdas yra atsitiktinis taškų parinkimas bazinių vektorių projekcijų srityje;
   - Vizualizuojant daugiamačius duomenis, kai tiriamų duomenų dimensija yra didesnė už 5, o duomenų aibę sudaro daugiau nei 3000 vektorių, tikslingiau vietoj standartinio daugiamačių skalių algoritmo naudoti santykinių DS algoritmą. Didinant bazinių vektorių skaičių gaunama tikslesnė projekcija. Tačiau per didelis bazinių vektorių skaičius lėtina

skaičiavimus. Tyrimai parodė, kad mažesnėms duomenų aibėms (iki 3000 vektorių) tikslinga imti nuo 700 iki 1000 bazinių vektorių, o didelėms duomenų aibėms – nuo 900 iki 1500.

3. Pasiūlyta daugiamačių duomenų taškų tarpusavio atstumų koregavimo transformacija, atliekant jų netiesinį projektavimą į dvimatę plokštumą, pagerina vizualizavimo kokybę, koregavimas geriau išryškina duomenų klasterius, mažiau iškraipo daugiamačių duomenų struktūras.

4. Buvo ištirtos ir palygintos dvi fiziologinių duomenų parametrizavimo sistemos: fraktalinių dimensijų parametrizavimo sistema ir polinominio aproksimavimo parametrų sistema. Vizualizuojant fraktalinės dimensijos duomenis matomas grupių persidengimas. Geriausias klasifikavimo rezultatas gautas klasifikuojant užimtumo dimensijos duomenis. Atlikus šių duomenų klasifikavimą nustatyta, kad tiksliausiai klasifikuoja atraminių vektorių (SVM) klasifikatorius (bendras tikslumas $\approx 90\,\%$, jautrumas $\approx 80\,\%$). Vizualizuojant polinominės parametrų sistemos duomenis, grupių persidengimas mažesnis. Atlikus šių duomenų klasifikavimą, tiksliausias yra taip pat atraminių vektorių klasifikatorius (bendras tikslumas $\approx 92\,\%$, jautrumas $\approx 83\,\%$).

5. Remiantis atlikta polinominės parametrų sistemos analize, kuri paremta koreliacine bei vizualia analize, nustatyta, kad dydžio ŠSD parametrų grupė yra labai priklausoma nuo dydžio JT parametrų grupės, ir ŠSD parametrų grupės galima atsisakyti.

6. Norint nustatyti preliminarią diagnozę naujam pacientui, yra siūloma rasti paciento duomenų projekciją plokštumoje, kur jau yra fiksuota etaloninė bazinių vektorių projekcija (suprojektuoti tiriamųjų duomenys su jau nustatytomis diagnozėmis, nubrėžti klasių skiriamieji paviršiai) ir nustatyti naujo taško padėtį tarp esamų taškų. Priklausomai nuo taško, atitinkančio tiriamąjį pacientą, padėties tarp skiriamųjų paviršių galime daryti preliminarų sprendimą apie šio tiriamojo sveikatos būklę. Pasiūlytas būdas yra naudingas medikams vertinant pacientų sveikatos būklę ir pastebint sportininko sveikatos pablogėjimą pradinėje to stadijoje.

7. Siūloma metodologija buvo taikyta fiziologinių duomenų analizei, tačiau ją galima taikyti bet kokių medicininių duomenų analizei siekiant nustatyti preliminarią diagnozę, o taip pat ir bendro pobūdžio daugiamačiams duomenims analizuoti. Tačiau pastaruoju atveju irgi būtina įsigilinti į tų duomenų kilmę ir specifiką.

Literatūros sąrašas

1. Agrawal, R.; Imielinski, T.; Swami, A. N. 1993. Mining association rules between sets of items in large databases, in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, P. Buneman and S. Jajodia, Eds. Washington, D.C., 207–216.
2. Agrawal, R.; Srikant, R. 1994. Fast algorithms for mining association rules in large databases, in *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, August 29–September 1, 1994, 487-499.
3. Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications, in *Proceedings of the ACM SIGMOD Conference*, Seattle, WA, 94–105.
4. Ankerst, M.; Breunig, M.; Kriegel, H. P.; Sander, J. 1999. OPTICS: Orderingpoints to identify clustering structure, in *Proceedings of the ACM SIGMOD Conference*, Philadelphia, PA, 49–60.
5. Asuncion, A.; Newman, D. J. 2007. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. [cited 9 April 2008]. Available from internet: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

6. Ball, G.; Hall, D. 1965. ISODATA, a novel method of data analysis and classification, *Technical Report AD-699616*, SRI, Stanford, CA.

7. Basalaj, W. 1999. Incremental multidimensional scaling method for database visualization, in *Proceedings of Visual Data Exploration and Analysis VI, SPIE*, 3647: 149–158.

8. Barbara, D.; Chen, P. 2000. Using the fractal dimension to cluster datasets, in *Proceedings of the 6th ACM SIGKDD*, Boston, MA, 260–264.

9. Bayardo, R. J. 1997. Brute-Force Mining of High-Confidence Classification Rules, in *Proc. of the Third Int ’1 Con. on Knowledge Discovery and Data Mining*, 123–126.

10. Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2006b. Optimal decisions in combining the SOM with nonlinear projection methods, *European Journal of Operational Research* 173(3): 729–745.

11. Bernatavičienė, J.; Dzemyda, G.; Marcinkevičius, V. 2007a. Diagonal majorization algorithm: properties and efficiency, *Information technology and control* 36(4): 353–358.

12. Borg, I.; Groenen, P. 1997. *Modern Multidimensional Scaling: Theory and Applications.* New York: Springer. 471 p.

13. Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. 1984. *Classification and Regression Trees.* Wadsworth International Group. 372 p.

14. Cabena, P.; Hadjinian, P.; Stadler, R.; Verhees, J.; Zanasi, A. 1998. *Discovering Data Mining: From Concepts to Implementation*. Prentice Hall. 195 p.

15. Card, S. K.; Mackinlay, J. D.; Shneiderman, B. eds. 1999. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann. 477 p.

16. Chambers, J. M.; Cleveland, W. S.; Kleiner, B.; Tukey, P. A. 1983. *Graphical Methods for Data Analysis*. Belmont, CA: Wadsworth. 395 p.

17. Chang, H. C.; Hsu, C. C.; Chen, E. 2002. *Mining Closed Frequent Itemsets for Incremental and Diminished Database with Lexicographic Tree Traversal*. Networks, Parallel and Distributed Processing, and Applications. 368 p.

18. Chatfield, C. 2003. *The Analysis of Time Series: An Introduction (6th ed.)*. Chapman and Hall. 334 p.

19. Chattratichat, J.; Darlington, J.; Ghanem, M. and et. al. 1997. Large Scale Data Mining: Challenges and Responses, in *Proceedings of the 3th*

International Conference on Knowledge Discovery and Data Mining, August, 143–146.

20. Cheung, D. W. L.; Ng, V. T.; Fu, A. W. C.; Fu, Y. 1996a. Efficient Mining of Association Rules in Distributed Databases. *IEEE Transactions on Knowledge and Data Engineering* 8(6): 911–922.

21. Cheung, D.; Han, J.; Ng, V.; Fu, A.; Fu, Y. 1996b. A fast distributed algorithm for mining association rules, in *Proc. of 1996 Int'l. Conf. on Parallel and Distributed Information Systems*, Miami Beach, Florida, 31–44.

22. Cios, K. J.; Teresinska, A.; Konieczna, S.; Potocka, J.; Sharma, S. 2000. Diagnosing myocardial perfusion from PECT bull's-eye maps – A knowledge discovery approach, *IEEE Engineering in Medicine and Biology Magazine, Special issue on Medical Data Mining and Knowledge Discovery* 19(4): 17–25.

23. Cleveland,W. S. 1994. *The Elements of Graphing Data. Hobart Press*, New Jersey, Summit. 297 p.

24. Cox, T. F.; Cox, M. A. A. 1994. *Multidimensional scaling.* London: Chapman & Hall. 213 p.

25. Cristianini, N.; Shawe-Taylor, J. 2003. Support Vector and Kernel Methods, In: *Berthold M, Hand DJ (eds.). Intelligent Data Analysis: An Introduction*, Springer-Verlag: 169–197.

26. Das, A.; Ng, W. K.; Woon, Y. K. 2001. Rapid association rule minint, in *Proceedings of the tenth international conference on Information and knowledge management*. ACM Press, 474–481.

27. Dash, M.; Liu, H. 2001. Efficient Hierarchical Clustering Algorithms Using Partially overlapping Partitions, *Advances in Knowledge Discovery and Data Mining, 5th Pacific-Asia Conference*, PAKDD 2001, Hong Kong, China, April 16–18, 2001: Proceedings (Lecture Notes in Artificial Intelligence), Springer. 595 p.

28. Davidon, W. C. 1959. Variable Metric Method for Minimization, *A.E.C. Research and Development Report*, ANL-5990.

29. Dempster, A. P.; Laird, N. M.; Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society Series B* 39(1): 1–38.

30. Demsar, J.; Zupan, B.; Leban, G. 2004. Orange: From Experimental Machine Learning to Interactive Data Mining, *White Paper, Faculty of*

Computer and Information Science, University of Ljubljana. [cited 9 April 2008]. Available from internet: < *http://www.ailab.si/orange* >.

31. Do, T. D.; Hui, S. C.; Fong, A. 2003. Mining Frequent Itemsets with Category-Based Constraints, *Lecture Notes in Computer Scien*ce 2843: 76–86.

32. Draper, N. R.; Smith, H. 1966. *Applied Regression Analysis*, John Wiley and Sons, New York. 407 p.

33. Duda, R. O.; Hart, P. E.; Stork, D. G. 2000. *Pattern Classification.* 2nd Edition. John-Wiley. 680 p.

34. Dunham, M. H. 2003. *Data Mining Introductory and Advanced Topics*. Pearson Education, Inc. Prentice Hall. 315 p.

35. Dzemyda, G. 2001. Visualization of a set of parameters characterized by their correlation matrix, *Computational Statistics and Data Analysis* 36(1): 15–30.

36. Dzemyda, G.; Bernatavičienė, J.; Kurasova, O.; Marcinkevičius, V. 2004. Sammono projekcijos paklaidos minimizavimo strategijos, *Lietuvos Matematikos Rinkinys* 44, Spec. nr., 628–633.

37. Dzemyda, G. 2005. Multidimensional data visualization in the statistical analysis of curricula, *Computational Statistics and Data Analysis* 49: 265–281.

38. Dzemyda, G.; Kurasova, O. 2006. Heuristic Approach for Minimizing the Projection Error in the Integrated Mapping, *European Journal of Operational Research* 171(3): 859–878.

39. Dzemyda, G.; Kurasova, O.; Vainoras, A. 2007. Parameter System for Human Physiological Data Representation and Analysis, *Pattern Recognition and Image Analysis – IbPRIA 2007, Lecture Notes in Computer Science, Springer* 4477: 209–216.

40. Dzemyda, G.; Kurasova, O.; Žilinskas, J. 2008. *Daugiamačių duomenų vizualizavimo metodai,* Mokslo Aidai. 167 p.

41. Ester, M.; Kriegel, H. P.; Sander, J.; Xu, X. 1995. A database interface for clustering in largespatial databases, in *Proceedings of the 1st ACM SIGKDD*, Montreal, Canada, 94–99.

42. Ester, M.; Kriegel, H. P.; Sander, J.; Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proceedings of the 2nd ACMSIGKDD*, Portland, Orego, 226–231.

43. Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. 1996a. *Knowledge discovery and data mining: Towards a unifying framework*, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD96), Portland, OR, AAAI Press, 82–88.

44. Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. 1996b. From Data Mining to knowledge Discovery: An Overview, *Advances in Knowledge Discovery and Data Mining,* Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padraic Smyth, and Ramasamy Uthurusamy, AAAI Press, 1–34.

45. Fayyad, U.; Grinstein, G. G.; Wierse, A. eds. 2002. *Information Visualization in Data Mining and Knowledge Discovery*. The Morgan Kaufman Series in Data Management Systems.Morgan Kaufman. 442 p.

46. Fielding, A. H. 2006. *Cluster and classification techniques in the BioSciences*. Cambridge University Press, Cambirdge. 220 p.

47. Fisher, R. A. 1936. The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics* 7: 179–188.

48. Fletcher, R.; Powell, M. J. D. 1963. A Rapidly Convergent Descent Method for Minimization, *Computer Journal* 6: 163–168.

49. Flexer, A. 2001. On the use of self-organizing maps for clustering and visualization, *Intelligent-Data-Analysis* 5(5): 373–384.

50. Friedman, J. H. 1991. Multivariate adaptive regression, *Annals of Statistics* 19: 1–141.

51. Grinstein, G. G.; Ward, M. O. 2002. *Introduction to Data Visualization. Information Visualization in data Mining and Knowledge Discovery*. Eds. U.Fayyad, G.G. Grinstein, A. Wierse. Morgan Kaufmann Publishers, 21–47.

52. Groenen, P. J. F.; Mathar, R.; De Leeuw, J. 1996. Least squares multidimensional scaling with transformed distances, in: W. Gaul & D. Pfeifer (Eds.), *Studies in classification, data analysis, and knowledge organization,* Berlin: Springer, 177–185.

53. Hall, L.O.; Ozyurt, B.; Bezdek, J. C. 1999. Clustering with a genetically optimized approach, *IEEE Trans. on Evolutionary Computation* 3(2): 103–112.

54. Han, J.; Pei, J.; Yin. Y. 2000. Mining frequent patterns without candidate generation, in *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data,* May, 1–12.

55. Han, J.; Kamber, M. 2006. *Data mining: concepts and techniques* (Morgan-Kaufman Series of Data Management Systems). Second edition, San Diego: Academic Press. 770 p.

56. Handl, J.; Knowles, J. 2005. Cluster generators for large high-dimensional data sets with large numbers of clusters. [cited 9 April 2008]. Available from internet: < http://dbkgroup.org/handl/generators/ >.

57. Hansen, C.; Johnson, C. 2004. *Visualization Handbook*, Elsevier Press. 984 p.

58. Hanson, S. J.; Burr, D. J. 1988. Minkowski back-propogation: Learning in connectionist models with non-euclidean error signals, in *Neural Information Processing Systems,* American Institute of Physics. 348 p.

59. Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation*. 2nd ed. Upper Saddle River, New Jersey: Prentice Hall. 842 p.

60. Heckerman, D. 1996. Bayesian networks for knowledge discovery, *Advances in Knowledge Discovery and Data Mining,* Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padraic Smyth, and Ramasamy Uthurusamy, AAAI Press, 273–305.

61. Higuchi, T. 1998. Approach to an Irregular Time Series on the Basis of Fractal Theory, *Physica D* 31: 277–283.

62. Hipp, J.; Güntzer, U.; Nakhaeizadeh, G. 2000. Algorithms for association rule mining – A general survey and comparison, *SIGKDD Explorations* 2(2): 1–58.

63. Hoffman, P. E.; Grinstein, G. G. 2002. A Survey of Visualizations for High-Dimensional Data Mining, *Information Visualization in Data Mining and Knowledge Discovery*, Ed. by U.Fayyad, G.G. Grinstein, A. Wierse. San Francisco: Morgan Kaufmann Publishers, 47–82.

64. Jain, A. K.; Dubes, R. C. 1988. *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall. 320 p.

65. Jollife, I. T. 1986. *Principal Component Analysis.* Springer-Verlag. 487 p.

66. Kaski, S. 1997. *Data Exploration Using Self-Organizing Maps PhD thesis.* Helsinki University of Technology, Department of Computer Science and Engineering, [cited 9 April 2008]. Available from internet: < http://www.cis.hut.fi/~sami/thesis/>.

67. Kaufman, L.; Rousseeuw, P. J. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY. 368 p.

68. Kaufman, L. Rousseeuw, P. J. 2005. *Finding groups in data: An introduction to cluster analysis*. John Wiley and Sons. 368 p.

69. Keim, D. A. 2002. Information Visualization and Visual Data Mining, *IEEE Transactions on Visualization and Computer Graphics* 8: 1–8.

70. Klemettinen, M.; Mannila, H.; Ronkainen, P.; Toivonen, H.; Verkamo, A. I. 1994. Finding Interesting Rules From Large Sets of Discovered Association Rules, in *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, 401–407.

71. Klock, H.; Buhmann, J. M. 1999. Data visualization by multidimensional scaling: A deterministic annealing approach, *Pattern Recognition* 33(4): 651–669.

72. Kohonen, T. 2001. *Self-Organizing Maps.* 3nd ed. Springer series in information sciences. Springer-Verlag 30. 501 p.

73. Kohonen, T. 2002. Self-Organizing Neural networks: Recent Advances and Applications, *Studies in Fuzziness and Soft Computing*, Heidelberg, New York: Physica-Verl. Ed U. Seiffert, L.C. Jain, 78: 1–11.

74. Kononenko, I. 1994. Estimating attributes: Analysis and extensions of Relief, in *L. De Raedt, & F. Bergadano (Eds.), Machine Learning: ECML-94,* Springer Verlag, 171–182.

75. Kotsiantis, S.; Kanellopoulos, D. 2006. Association Rules Mining: A Recent Overview, *GESTS International Transactions on Computer Science and Engineering* 32(1): 71–82.

76. Kraaijveld, M. A.; Mao, J.; Jain, A. K. 1995. A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps, *IEEE Transactions on Neural Networks* 6(3): 548–559.

77. Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, *Psychometrika* 29: 1–27.

78. Kruskal, J. B.; Wish, M. 1984. *Multidimensional Scaling*. Beverly Hills and London: Sage Publications. 93 p.

79. Kurasova, O. 2005. *Daugiamačių duomenų vizuali analizė taikant savireguliuojančius neuroninius tinklus.* Matematikos ir Informatikos Institutas (daktaro disertacija). 160 p.

80. Lange, T.; Law, M. H. C.; Jain, A. K.; Buhmann, J. M. 2005. *Learning with constrained and unlabelled data*. CVPR, 731 – 738.

81. Larose, D. T. 2004. *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley-Interscience. 222 p.

82. de Leeuw, W.; van Liere, R. 2003. Visualization of Multi Dimensional Data using Structure Preserving Projection Methods, *Data Visualization: The State of the Art*, 213–223.

83. LeBlanc, J.; Ward, M. O.; Wittels, N. 1990. Exploring N-Dimensional Databases*,* in *Visualization '*90, San Francisco, CA, 230–239.

84. Li, W.; Han, J.; Pei, J. 2001. CMAR: Accurate and efficient classification based on multiple class-assocoation rules, in *Proc. 2001 Int. Conf. Data Mining (ICDM'01),* San Jose, CA, 369–376.

85. Lin, D.; Kedem, Z. M. 2002. Pincer-search: An efficient algorithm for discovering the maximum frequent set, *IEEE Transactions on Knowledge and Data Engineering* 14(3): 553–566.

86. Liu, B.; Hsu, W.; Ma, Y. 1998. Integrating clssification and association rule mining, in *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98),* New York, NY, 80–86.

87. Luo, C.; Pereira, A. L.; Chung, S. M. 2006. Distributed Mining of Maximal Frequent Itemsets on a Data Grid System, *The Journal of Supercomputing, Springer Netherlands* 37(1): 71–90.

88. MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations, in Le Cam, L. M. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, I: Statistics, University of California Press, Berkeley and Los Angeles, CA, 281–297.

89. Mannila, H.; Toivonen, H.; Verkamo, A. I. 1997. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery* 1(3): 259–289.

90. Mao, J.; Jain, A. K. 1995. Artificial neural networks for feature extraction and multivariate data projection, *IEEE Transactions on Neural Networks* 6(2): 296–317.

91. Mathar, R.; Žilinskas, A. 1993. On Global Optimization in Two-Dimensional Scaling, *Acta Aplicandae Mathematicae* 33: 109–118.

92. Medvedev, V. 2007. *Tiesioginio sklidimo neuroninių tinklų taikymo daugiamačiams duomenims vizualizuoti tyrimai.* Matematikos ir Informatikos Institutas (daktaro disertacija). 144 p.

93. Michalewicz, Z. 1992. *Genetic Algorithms+Data Structures=Evolution Programs.* Springer–Verlag. 387 p.

94. Mitchell, M. 1996. *An Introduction to Genetic Algorithms.* MIT Press. 414 p.

95. Mitchell, T. 1997. *Machine Learning.* McGraw-Hill, New York, NY. 160 p.

96. Naud, A.; Duch, W. 2000. Interactive data exploration using MDS mapping, in *Proceedings of the Fifth Conference „Neural Networks and Soft Computing"*, 255–260.

97. Naud, A. 2004. Visualization of high-dimensional data using a association of multidimensional scaling to clustering, in *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems* 1: 252–255.

98. Naud, A. 2006. An Accurate MDS-Based Algorithm for the Visualization of Large Multidimensional Datasets, *Lecture Notes in Computer Science* 4029: 643–652.

99. Ng, R.; Han, J. 1994. Efficient and effective clustering methods for spatial data minint, in *Proceedings of the 20th Conference on VLDB*, Santiago, Chile, 144–155.

100. Nielson, G. M.; Hagen, H.; Muller, H. eds. 1997. *Scientific Visualization*. IEEE Computer Society, 577 p.

101. de Oliviera, M. C. F.; Levkowitz, H. 2003. FromVisual Data Exploration to Visual DataMining: A Survey*, IEEE Transactions on Visualization and Computer Graphics* 9(3): 378–394.

102. Orlando, S.; Lucchese, C.; Palmerini, P.; Perego, R.; Silvestri, F. 2003. Kdci: a multi-strategy algorithm for mining frequent sets, in Bart Goethals and Mohammed J. Zaki, editors, FIMI'03: *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, November, 2003, 1–10.

103. Pal, N.; Jain, L. C. 2005. *Advanced Techniques in Knowledge Discovery and Data Mining (Advanced Information and Knowledge Processing)*. Springer-Verlag New York. 272 p.

104. Parthasarathy, S., Zaki, M. J., Ogihara, M. 2001. Parallel data mining for association rules on shared-memory systems. *Knowledge and Information Systems: An International Journal* 3(1): 1–29.

105. Parthasarathy, S. 2002. Efficient Progressive Sampling for Association Rules, *ICMD 2002*, 354–361.

106. Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kauffman. 552 p.

107. Pekalska, E.; De Ridder, D.; Duin, R. P. W.; Kraaijveld, M. A. 1999. A new method of generalizing Sammon mapping with application to algorithm speed-up, in *Proc. of the 5th Annual Conference of the Advanced School for Computing and Imaging ASCI'99Boasson*, Ed. by J. A. Kaandorp, J. F. M. Tonino. Delft, 221–228.

108. Podlipskytė, A. 2003. *Daugiadimensinių duomenų vizualizacija ir jos taikymas biomediciniių duomenų analizei*. Daktaro disertacija. Kauno technologijos universitetas. 118 p.

109. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. 1992. *Numerical Recipes in C.* Second ed. Cambridge: Cambridge University Press. 1020 p.

110. Rafiei, D.; Mendelzon, A. 1997. Similarity-based queries for time series data, in *Proc. 1997 ACM-SIGMOID Int. Conf. Management of Data (SIGMOID'97),* Tocson, AZ, 13–25.

111. Ramoni, M.; Sebastiani, P. 2003. Bayesian Methods. *Intelligent Data Analysis: an Introduction.* Ed. by M. Berthold, D. J. Hand. Springer-Verlag, 131–168.

112. Ramsay, J. O. 1977. Maximum likelihood estimation in MDS. *Psychometrika* 42: 241–266.

113. Raudys, Š. 2001. *Statistical and neural Classifiers: an Integrated Approach to Design.* Advances in pattern recognition, Springer-Verlag. 326 p.

114. Roddick, J. F.; Spiliopoulou, M. 2002. A survey of temporal knowledge discovery paradigms and methods, *IEEE Transactions on Knowledge and Data Engineering* 14(4): 750–767.

115. Roweis, S. T.; Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding, *Science* 290(5500): 2323–2326.

116. Sabirov, Sh. 2000. Distribution function of the distance between two points inside a cube, *Random Oper. and Stoch*. Equ. 8: 339–342.

117. Sammon, J. W. 1969. A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers*, C-18(5): 401–409.

118. Sarafis, I.; Zalzala, A. M. S.; Trinder, P. W. 2002. A genetic rule-based data clustering tolkit, *Congress on Evolutionary Computation (CEC*), Honolulu, USA, 1238–1243.

119. Savasere, A.; Omiecinski, E.; Navathe, S. 1995. An efficient algorithm for mining association rules in large databases, in *Proceedings of the 21st VLDB Conference*, Zurich, Switzerland, 432–443.

120. Schwaighofer, A. 2002. SVM Toolbox for Matlab, [cited 20 April 2008]. Avialable from internet: <http://ida.first.fraunhofer.de/~anton/ /software.html>.

121. Shasha, D.; Zhu, Y. 2004. *High Performance Discovery in Time Series: Techniques and case Studies.* Springer. 190 p.

122. Sheikholeslami, G.; Hhatterjee, S.; Zhang, A. 1998. WaveCluster: A multiresolution clustering approach for very large spatial databases, in *Proceedings of the 24th on VLDB*, New York, NY, 428–439.

123. Shepard, R. N. 1962a. The analysis of proximities: Multidimensional scaling with an unknown distance function, *I. Psychometrika* 27(2): 125–140.

124. Shepard, R. N. 1962b. The analysis of proximities: Multidimensional scaling with an unknown distance function, *II. Psychometrika* 27(3): 219–246.

125. Shumway, R. H.; Stoffer, D. S. 2005. *Time Series Analysis and Its Applications.* Springer. 575 p.

126. Simoudis, E. 1996. Reality Check for Data mining, *IEEE Expert: Intelligent Systems and Their Applications* 11(5): 26–33.

127. Silverstein, C.; Brin, S.; Motwani, R.; Ullman J. D. 2000. Scalable techniques for mining causal structures, *Data Mining and Knowledge Discovery* 4(2/3): 163–192.

128. Sumathi, S.; Sivanandam. S. N. 2006. *Introduction to data mining and its applications*, Springer–Verlag, New York. 828 p.

129. Šaltenis, V. 2004. Outlier Detection Based on the Distribution of Distances between Data Points, *Informatica* 15(3): 399–410.

130. Takane, Y.; Young, F. W.; de Leeuw, J. 1977. Nonmetric individual differences in multidimensional Scaling: An alternating least squares method with optimal scaling features, *Psychometrika* 42: 7–67.

131. Taylor, P. 2003. *Statistical Methods. Intelligent Data Analysis: an Introduction.* Ed. by M. Berthold, D. J. Hand. Springer-Verlag, 69–129.

132. Tipping, M. E. 1996. *Topographic mappings and feed-forward neural networks*. Ph.D thesis, Aston University, Aston Street, Birmingham B4 7ET, UK. 157 p.

133. Toivonen, H. 1996. Sampling large databases for association rules, *The VLDB journal,* 134–145.

134. Torgerson, W. S. 1952. Multidimensional scaling: I. theory and method, *Psychometrika* 17: 401–419.

135. Tufte, E. R. 1990. *Envisioning Information*. Graphics Press. 126 p.

136. Tung, A. K. H.; Hou, J.; Han, J. 2001. Spatial clustering in the presence of obstacles, in *Proceedings of the 17th ICDE*, Heidelberg, Germany, 359–367.

137. Vainoras, A.; Ašeriškytė, D.; Poderys, J.; Navickas, Z. 2005. Fractal dimensions in evaluation of hear tfunction parameters during physical investigations, *Journal „Education. Physical Training. Sport"* 3(57): 61–66.

138. Vapnik, V. N. 1998. *Statistical Learning Theory.* John Wiley & Sons. 768 p.

139. Vesanto, J. 2001. Importance of Individual Variables in the k-Means Algorithm, in *Proceedings of PAKDD 2001, Hong Kong, China*, 513–518. [cited 20 March 2008] Avialable from internet < http://lib.hut.fi/ Diss/2002/isbn9512258978/article8.pdf>.

140. Wang, W.; Yang, J.; Muntz, R. R. 1999. STING+:An approach to actine spatialdata minint, in *proceedings 23th Conference on VLDB,* Athens, Greece, 186–195.

141. Widrow, B.; Rumelhart, D. E.; Lehr, M. A. 1994. Neural networks: Aplications in industry, business and science, *Comm. ACM* 37: 93–105.

142. Wong, P. C. 1999. Visual data minint, *IEEE Computer graficsand Applications* 19(5): 20–21.

143. Wong, P. C.; Bergeron, R. D. 1997. 30 Years of Multidimensional Multivariate Visualization, *Scientific Visualization*. IEEE Computer Society, 3–33.

144. Xu, X.; Ester, M.; Kriegel, H. P.; Sander, J. 1998. A distribution-based clustering algorithm for mining large spatial datasets, in *Proceedings of the 14th ICDE*, Orlando, FL, 324–331.

145. Yin, X.; Han, J. 2003. CPAR: Classification based on predictive association rules, in *Proc. 2003 SIAM Int. Conf. Data Mining (SDM' 03),* San Francisco, 331–335.

146. Zaki, M. J.; Parthasarathy, S.; Ogihara, M.; Li, W. 1997. New algorithms for fast discovery of association sules, in D. H. H. Mannila and D. Pregibon, editors, *Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD-97)*, Newport Beach, California, USA, August 14–17, AAAI Press, 283–286.

147. Zaki, M. J. 2000. Scalable Algorithms for Association Mining, *IEEE Transactions on Knowledge and Data Engineering*, May/June 12 (3): 372–390.

148. Zaki, M. J. 2001. SPADE: An Efficient Algorithm for Mining Frequent Sequences, *Machine Learning* 42(1-2): 31–60.

149. Zaki, M. J.; Parimi, N.; De N.; Gao, F.; Phoophakdee, B.; Urban, J.; Chaoji, V.; Hasan, M.; Salem, S. 2005. Towards Generic Pattern Mining, *Formal Concept Analysis*. Springer Berlin / Heidelberg 3403: 1–20.

150. Zhang, T.; Ramakrishnan, R.; Livny, M. 1996. BIRCH: an efficient data clustering method for very large databases, in *Proceedings of the ACM SIGMOD Conference*, Montreal, Canada, 103–114.

151. Žilinskas, A. 1993. On visualization of optimization process, in*: Parametric Optimization and Related Topics 3*, ed. by J. Guddat et al, Frankfurt am Main: Peter Lang Verlag, 549–556.

152. Žilinskas, A.; Podlipskyte, A. 2002. On symbolic computation in problem of geometric probabilities, *Information Technology and Control* 24: 49–54.

153. Žilinskas, A.; Podlipskytė, A. 2003. On multimodality of the SSTRESS criterion for metric multidimensional scaling, *Informatica* 14(1): 121–130.

154. Žilinskas, A. 2003. On the distribution of the distances between two points in a cube, *Random Operators and Stochastic Equations* 11: 21–24.

155. Žilinskas, A.; Žilinskas, J. 2007. Two level minimization in multidimensional scaling, *Journal of Global Optimization* 38(4): 581–596.

156. Žilinskas, A.; Žilinskas, J. 2008. Branch and bound algorithm for multidimensional scaling with city-block metric, *Journal of Global Optimization,* in press.

Autoriaus publikacijų sąrašas disertacijos tema

Straipsniai tarptautiniuose periodiniuose leidiniuose,
įtrauktuose į Mokslinės informacijos instituto pagrindinį sąrašą
(*Thomson ISI Web of Science*)

1A. Bernatavičienė, J.; Dzemyda, G.; Marcinkevičius, V. 2007c. Conditions for optimal efficiency of relative MDS, *Informatica* 18(2): 187–202. ISSN 0868-4952. [Current Abstracts; IAOR: International Abstracts In Operations Research; INSPEC; MatSciNet; Science Citation Index Expanded (Web of Science); Scopus; TOC Premier; VINITI; Zentralblatt MATH].

Straipsniai tarptautiniuose periodiniuose leidiniuose, įtrauktuose į
Mokslinės informacijos instituto konferencijos darbų sąrašą
(*Thomson ISI Proceedings*)

2A. Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2006a. Decision support for preliminary medical diagnosis integrating the data mining methods, *Simulation and optimisation in business and industry: International conference on operational research*: 17–20 May 2006, Kaunas: Technologija, 155–160. ISBN 9955-25-061-5.

3A. Bernatavičienė, J.; Šaltenis, V. 2006c. Correction of distances in the visualization of multidimensional data, *Series on computers and operations research* 7. Computer aided methods in optimal design and operations, New Jersey: World Scientific, 159–168. ISBN 981-256-909-X.

Straipsniai Lietuvos mokslo tarybos patvirtinto sąrašo tarptautinėse duomenų bazėse referuojamuose leidiniuose

4A. Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2006d. Strategies of selecting the basic vector set in the relative MDS, *Ūkio technologinis ir ekonominis vystymas [Technological and economic development of economy]* 12(4): 283–288. ISSN 1392-8619. [ASCE Civil Engineering Abstracts; Business Source Complete; Business Source Premier; Current Abstracts; ICONDA; SCOPUS; TOC Premier].

5A. Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V.; Medvedev, V. 2007b. The Problem of Visual Analysis of Multidimensional Medical Data, *Models and Algorithms for Global Optimization, Springer Optimization and Its Applications* 4: 277–298. ISSN 1931-6828. [SpringerLINK].

Straipsniai kituose recenzuojamuose mokslo leidiniuose, konferencijų pranešimų medžiagoje

6A. Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V.; Šaltenis, V.; Tiešis, V. 2006e. Visualization and analysis of the eye fundus parameters, in *Proceedings of the 6th Nordic conference on eHealth and telemedicine NCeHT2006*: 31 August – 1 September 2006, Finland, Helsinki, 267–268.

Straipsniai kituose periodiniuose leidiniuose, vienkartiniuose straipsnių rinkiniuose ir kt.

7A. Bernatavičienė, J.; Berškienė, K.; Ašeriškytė, D.; Dzemyda, G.; Vainoras, A.; Navickas, Z. 2005a. Fraktalinių dimensijų biomedicininio informatyvumo analizė, *Biomedicininė inžinerija: tarptautinės konferencijos pranešimų medžiaga*, Kaunas, 27–31. ISBN 9955-09-950-X.

8A. Bernatavičienė, J.; Šaltenis, V. 2005b. Atstumų koregavimas vizualizuojant daugiamačius duomenis, *Informacinės technologijos 2005: konf. pranešimų medžiaga,* Kaunas: Technologija, 102–107. ISBN 9955-09-788-4.

9A. Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Vainoras, A. 2006f. Integration of classification and visualization for diagnosis decisions, *International journal of information technology and intelligent computing* 1(1): 57–68. ISSN 1895-8648.