VYTAUTO DIDŽIOJO UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Virginijus MARCINKEVIČIUS

NETIESINĖS DAUGIAMAČIŲ DUOMENŲ PROJEKCIJOS METODŲ SAVYBIŲ TYRIMAS IR FUNKCIONALUMO GERINIMAS

DAKTARO DISERTACIJA

FIZINIAI MOKSLAI (P 000)
INFORMATIKA (09 P)
INFORMATIKA, SISTEMŲ TEORIJA (P 175)

VILNIUS, 2010

Disertacija rengta 2003–2010 metais Matematikos ir informatikos institute. Disertacija ginama eksternu.

Darbo mokslinis vadovas

Prof. habil. dr. Gintautas DZEMYDA (Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09 P) (2003–2010).

Padėka

Nuoširdžiai dėkoju darbo vadovui prof. habil. dr. G. Dzemydai už ypatingai atkaklų vadovavimą. Dėkoju savo kolegėms dr. J. Bernatavičienei, dr. O. Kurasovai ir R. Karbauskaitei už bendradarbiavimą, kuriant ir skelbiant šiame darbe pateiktus rezultatus.

Esu dėkingas disertacijos recenzentams prof. dr. (HP) A. L. Lipeikai ir prof. habil. dr. M. Sapagovui bei kolegai dr. P. Treigiui, atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų pastabų bei patarimų, padėjusių pagerinti šio darbo kokybę. Taip pat nuoširdžiai dėkoju I. Driukienei už pagalbą rengiant disertacijos santraukos tekstą.

Dėkoju Matematikos ir informatikos instituto sistemų analizės skyriaus kolegoms už kritiką ir draugišką pagalbą, rengiant disertaciją.

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.

Virginijus Marcinkevičius

Reziumė

Informacijos amžiuje susiduriame su nuolat didėjančiais duomenų kiekiais. Mūsų tikslas yra surasti esmines ar trokštamas žinias, slypinčias šiuose duomenyse, jas suvokti ir pritaikyti praktikoje. Šioje disertacijoje aprašomi tyrimai vykdomi duomenų vizualizavimo srityje, priskiriamoje aiškinamajai duomenų analizei, kurios tikslas pateikti grafinę informaciją, leidžiančią manipuliuoti duomenimis, naudojantis juose esančios informacijos vizualiu pateikimu.

Disertacijos objektas yra daugiamačiai duomenys ir daugiamatėmis skalėmis grindžiami dimensijos mažinimo arba projekcijos metodai. Analizuojant daugiamačius duomenis buvo taikomi įvairūs metodai: saviorganizuojantis neuroninis tinklas, SMACOF ir Sammono algoritmai, diagonalinis mažoravimo ir santykinių daugiamačių skalių algoritmai bei santykinės perspektyvos metodas. Šių netiesinės daugiamačių duomenų projekcijos metodų funkcionalumo gerinimas yra pagrindinis šio darbo tikslas.

Disertaciją sudaro įvadas, trys skyriai, išvados ir literatūros sąrašas. Pirmajame skyriuje apžvelgiami disertacijoje tirti daugiamačių skalių metodai ir pateikiami nauji teoriniai disertacijos autoriaus rezultatai. Antrajame skyriuje pristatomos metodų tyrimų ir pradinių taškų daugiamatėse skalėse parinkimo metodologijos. Trečiajame skyriuje eksperimentiškai pagrįsti atskirų disertacijoje nagrinėtų daugiamačių duomenų vizualizavimo metodų parametrų parinkimo būdai.

Bendra disertacijos apimtis - 105 puslapiai, 57 numeruotos formulės, 29 paveikslai ir 13 lentelių. Literatūros sąrašą sudaro 107 šaltiniai.

Tyrimų rezultatai publikuoti 8 recenzuojamuose periodiniuose mokslo žurnaluose ir 6 kituose mokslo leidiniuose. Rezultatai pristatyti ir aptarti 14 nacionalinėse ir tarptautinių konferencijų Lietuvoje ir užsienyje.

Abstract

In The Information Age people encounter with bigger and bigger amounts of data. We aim to find crucial or desirable knowledge, which lies in that data, to comprehend it and to apply it to practice. Researches, described in this dissertation, were done in the data visualization field, attributed to explanatory analysis of data, which aim to provide diagrammatic information, that allow manipulating the data, while using visual presentation of information of that data.

The object of the dissertation is multidimensional data and methods of dimensionality reduction or projection, that are validated by multidimensional scaling. To analyze multidimensional data there were various methods applied: self-organizing maps, SMACOF algorithm and Sammon's mapping algorithm, diagonal majorization algorithm and relative multidimensional scaling algorithm, and relative perspective method. The main objective of this work is to improve functionality of these nonlinear projection methods of multidimensional scaling.

The dissertation consists of 3 chapters, and the list of references. First chapter is dedicated to review multidimensional scaling methods, which were studied in this work, and to present new theoretical results of the author of the work. Second chapter introduce methodologies to study methods and to select initial points in multidimensional scaling. Third chapter gives substantiation by experiment of means to select parameters of particular multidimensional data visualization methods, analyzed in the dissertation.

The dissertation is written in Lithuanian. It consists of introduction, 3 chapters, conclusions, and the list of references. There are 105 pages of the text, 57 numbered formulas, 29 figures, 13 tables and 107 bibliographical sources.

The main results of this dissertation were published in 14 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; 5 articles in journals indexed in international database approved by Science Council of Lithuania; 6 publications are printed in the national and international conference proceedings. The main results of the work have been presented and discussed at 14 international and national conferences, workshops, seminars.

Žymėjimai

Simboliai

$\hat{\delta}_{ij}$	i - tojo ir j - tojo taškų (aibės X elementų) skirtumas, neatitikimas (*angl. disparity*); naudojamas nemetrinėse daugiamatėse skalėse.
$D^{(s)}$	visų kvadratinių nepanašumų matricų erdvė ($s \times s$).
E_{DS}	daugiamačių skalių (DS) paklaidos (tikslo, įtempimo) funkcija, paklaida ($E_{DS} = \sqrt{E_{norm}}$).
E_S	Sammono projekcijos paklaidos (tikslo, įtempimo) funkcija.
E_{SOM}	saviorganizuojančio neuroninio tinklo kvantavimo paklaida (arba $E_{SOM(kvant.)}$).
E_p	dalelių (disertacijoje – taškų) tarpusavio sąveikos potencinė energija. Standumo parametras $p\epsilon(-1, +\infty)$, suteikiantis galimybę kontroliuoti, kaip greitai stūmos jėgos mažės didėjant atstumui tarp dalelių.
R^n	n - matė Euklidinė erdvė, kurios elementai yra taškai arba vektoriai, kurių koordinatės yra realieji skaičiai.

$X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$	i - asis aibės X elementas. Taškas arba vietos vektorius, kurio koordinatės realieji skaičiai, t. y. $X_i \in X \subset R^n$.
$Y_i = \{y_{i1}, y_{i2}, \dots, y_{id}\}$	i - asis projekcijos taškas arba vietos vektorius $Y_i \in Y \subset R^d$, kurio koordinatės realieji skaičiai.
d_{ij}	atstumas tarp i - tojo ir j - tojo taškų. $d_{ij} \in D$.
$\hat{e}$	epochos numeris.
k_x	stačiakampio SOM tinklo stulpelių skaičius.
k_y	stačiakampio SOM tinklo eilučių skaičius.
$m_{ij} = \{m_{ij}^1, m_{ij}^2, \dots, m_{ij}^n\}$	neuroninio tinklo (SOM) i - tosios eilutės ir j - tojo stulpelio neuronas.
p_{ij}	i - tojo ir j - tojo taškų artimumas.
w_{ij}	svorio koeficientas.
δ_{ij}	i - tojo ir j - tojo taškų (aibės X elementų) skirtumingumas, nepanašumas ($\delta_{ij} \in \Delta$).
ρ_{Sp}	Spirmeno koeficientas.
$\sigma^2(X)$	dispersija.
Δ	nepanašumų (skirtingumų), tarp aibės X elementų, matrica žym. $\Delta \equiv (\delta_{ij})$.
D	nepanašumų matrica, kurią konstruojant remiamasi nepanašumų matrica Δ. Matrica $D \in D_D^{(s)}$ priklauso pasirinktai nepanašumų matricų klasei $D_D^{(s)}$.
$E(X)$	matricos X stulpelių vidurkių vektorius, vidurkis.
R	realiųjų skaičių laukas ($\mathbb{R}$ – realiųjų skaičių aibė).
S	bet kokios prigimties objektų ar elementų rinkinys (aibė).
V	svorių matrica $V \equiv (w_{ij})$.
$X = \{X_1, X_2, \dots, X_s\}$	duomenų aibė (matrica), kurios elementai (eilutės) yra taškai arba vietos vektoriai $X_i \in R^n$.
d	projekcijos taško koordinačių skaičius; projekcijos erdvės R^d $(d \leq n)$ matmenų skaičius (matavimų skaičius, dimensija).
e	mokymo epochų skaičius.
k	kaimyniškumo parametras, nusakantis į kiek greta (pagal tvarkos sąryšį) taško X_i esančių aibės X kaimynų reikia atsižvelgti apskaičiuojant jo DMA projekciją.
k'	stačiakampio SOM tinklo didžiausios kraštinės ilgis.
m'	iteracijos numeris.

n	taško koordinačių skaičius arba erdvės R^n matmenų skaičius (matavimų skaičius, dimensija).
s	visų imties taškų arba vektorių skaičius.
$\sigma(X)$	matricos X stulpelių vidutinis kvadratinis nuokrypis.

Santrumpos

DMA	diagonalinis mažoravimo algoritmas (*angl. Diagonal Majorization Algorithm*).
DS	metrinės daugiamatės skalės (*angl. multidimensional scaling*, MDS).
ISOMAP	izomerinis atvaizdavimas, praplečiantis metrines daugiamates skales, įvedant geometrinius atstumus, besiremiančius svertiniu grafu.
LLE	lokaliai tiesinis atvaizdavimas (*angl. Locally Linear Embeding*).
MJ	minimalaus jungimo kriterijus (*angl. minimal wiring*).
MTI	metrikos topologijos išsaugojimo kriterijus (*angl. metric topology preserving*).
PKA	pagrindinių komponenčių analizės metodas (*angl. principal component analysis*).
RPM	santykinės perspektyvos metodas (*angl. Relational Perspective Map*).
SDS	santykinių daugiamačių skalių algoritmas (*angl. relative MDS*).
SMACOF	daugiamačių skalių paklaidos minimizavimo algoritmas (*angl. Scaling by MAjorizing a COmplicated Function).*
SOM	saviorganizuojantis neuroninis tinklas (*angl. self-organizing map*).
SOM_Sammono	saviorganizuojančio neuroninio tinklo ir Sammono algoritmų junginys.
SOM_SMACOF	saviorganizuojančio neuroninio tinklo ir daugiamačių skalių algoritmų junginys.

Turinys

Įvadas

Tyrimų sritis

Informacijos amžiuje kasdien susiduriame su sparčiai didėjančiomis duomenų apimtimis, kur mūsų tikslas yra surasti esmines ar trokštamas žinias, slypinčias šiuose duomenyse, suprasti mus dominančią informaciją ir ją pritaikyti praktikoje. Šios informacijos paieška vykdoma tokiose svarbiose srityse, kaip statistika, duomenų vizualizavimas, duomenų bazės, duomenų gavyba (*angl. data mining*), šablonų atpažinimas arba atpažinimo teorija (*angl. pattern recognition*), sistemos mokymasis (*angl. machine learning*) ir dirbtinis intelektas. Kiekvienos jų tikslas – padidinti mūsų suvokimą apie turimus duomenis.

Istoriškai jau nuo seniai yra taikomi įvairūs statistiniai metodai, kurie bando perkelti turimus empirinius duomenis į matematinį modelį. Todėl dauguma statistikos darbų yra skirta iš anksto (*lot. a priori*) suformuluotų hipotezių apie duomenis tikrinimui. Darbe koncentruojamasi ties aiškinamąja duomenų analize arba duomenų vizualizavimu, kurio tikslas – pateikti grafinę informaciją, leidžiančią manipuliuoti duomenimis ir juos lengviau suvokti. Pagrindinis dėmesys disertacijoje kreipiamas į duomenų arba juose esančios informacijos vizualų pateikimą, naudojantis vizualizavimo, dimensijos mažinimo (*angl. dimensionality reduction*) arba projekcijos technikomis.

Darbo aktualumas

Duomenų suvokimas yra sudėtingas procesas, ypač kai duomenys nurodo sudėtingą objektą, reiškinį, kuris apibūdinamas daugeliu kiekybinių ir kokybinių parametrų ar savybių. Tokie duomenys vadinami daugiamačiais duomenimis ir gali būti interpretuojami kaip taškai arba vietos vektoriai daugiamatėje erdvėje. Analizuojant daugiamačius duomenis, dažnai į pagalbą pasitelkiame vieną svarbiausių duomenų analizės įrankių – duomenų vizualizavimą arba grafinį informacijos pateikimą. Pagrindinė vizualizavimo idėja – duomenis pateikti tokia forma, kuri leistų naudotojui lengviau suprasti duomenis, daryti išvadas ir tiesiogiai įtakoti tolesnį sprendimų priėmimo procesą. Vizualizavimas leidžia geriau suvokti sudėtingas duomenų aibes, gali padėti nustatyti tyrėją dominančius jų poaibius. Dimensijos mažinimo metodai leidžia atsisakyti tarpusavyje priklausomų duomenų komponenčių, o projekcijos metodais galima transformuoti daugiamačius duomenis į tiesę, plokštumą, trimatę erdvę ar į kitą žmogui vizualiai suvokiamą formą. Vizualią informaciją žmogus pajėgus suvokti daug greičiau ir paprasčiau negu skaitinę arba tekstinę. Iš kitos pusės toks suvokimas gali būti tik kaip dirva hipotezėms ir tolimesniems tyrimams, pagrįstiems griežtais matematiniais modeliais. Kokia informacija ir kaip ji turi būti vizualiai pateikiama, priklauso nuo naudotojo, dirbančio šioje srityje, todėl čia iškyla problemos, reikalaujančios atsakymų: kokius vizualizavimo metodus pasirinkti ir kaip optimaliai parinkti jų parametrus. Dėl nuolat didėjančių duomenų aibių, atsiranda vis nauji duomenų vizualizavimo metodai, tačiau išlieka aktuali problema – šių metodų aprobavimas ir taikymo pagrįstumo tyrimai.

Tyrimo objektas

Disertacijos tyrimo objektas yra daugiamačiai duomenys, jų atvaizdavimas netiesiniais daugiamačių skalių algoritmais ir saviorganizuojančiais neuroniniais tinklais, projekcijos kokybės vertinimas.

Darbo tikslas ir uždaviniai

Darbo tikslas – netiesinės daugiamačių duomenų projekcijos metodų funkcionalumo gerinimas, tiriant jų savybes.

Siekiant šio tikslo sprendžiami šie uždaviniai:

- Ištirti daugiamačių skalių algoritmų duomenų pradinio parinkimo būdus.
- Palyginti daugiamačių skalių SMACOF algoritmą, Sammono algoritmą ir santykinių daugiamačių skalių algoritmą topologijos išsaugojimą įvertinančiais kriterijais.
- Ištirti diagonalinio mažoravimo algoritmo efektyvumą, lyginant jį su daugiamačių skalių SMACOF ir santykinių daugiamačių skalių algoritmais.
- Teoriškai ištirti saviorganizuojančio neuroninio tinklo (SOM) neuronų nugalėtojų skaitinę priklausomybę nuo mokymo epochos.
- Ištirti naujas galimybes SOM tinklui vaizduoti.
- Modifikuoti santykinės perspektyvos metodo algoritmą, siekiant pagerinti jo konvergavimą.
- Ištirti santykinių daugiamačių skalių algoritmo parametrus, siekiant apskaičiuoti vienareikšmišką ir tikslią projekciją.

Tyrimų metodika

Analizuojant mokslinius ir eksperimentinius pasiekimus duomenų vizualizavimo srityje, naudoti informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai.

Kuriant programinę įrangą, naudotas programinio modeliavimo metodas. Teoriniai tyrimo metodai naudoti įrodant teoremas ir tiriant algoritmų konvergavimą. Taikytas matematinės indukcijos principas įrodant teiginius.

Remiantis eksperimentinio tyrimo metodu, atlikta statistinė duomenų ir tyrimų rezultatų analizė, kurios rezultatams įvertinti naudotas apibendrinimo metodas.

Darbo mokslinis naujumas ir jo reikšmė

Darbe atlikti tyrimai atskleidė naujas galimybes vystyti daugiamačių duomenų vizualizavimo metodus ir priemones.

Įrodyta, kad Sammono algoritme projekcijos duomenų pradinis parinkimas ant tiesės yra netinkamas. Remiantis tuo yra tikslinga naudotis principinių komponenčių analize ar didžiausių dispersijų metodu parenkant projekcijos pradinius taškus.

Parodyta, kad diagonalinio mažoravimo algoritmo efektyvumas nusileidžia daugiamačių skalių SMACOF realizacijai ir santykinėms daugiamatėms skalėms.

Teoriškai ištirta vienos epochos metu perskaičiuojamų stačiakampės formos SOM tinklo neuronų skaičiaus priklausomybė nuo mokymo epochos numerio.

Pasiūlytas naujas būdas neuroninio tinklo SOM vaizdavimui. Jame neuroninio tinklo lentelės ląstelių spalva parenkama kaip pilkos spalvos atspalvis, priklausantis nuo ląstelėse esantį neuroną atitinkančio vektoriaus ilgio.

Pasiūlytas naujas pradinių duomenų parinkimo būdas pagal didžiausias dispersijas, tinkamas visiems daugiamačių skalių klasės algoritmams.

Ištirtas santykinės perspektyvos metodo konvergavimas ir pasiūlyta naudoti dvi naujas atstumų funkcijas, taip užtikrinat RPM metodo konvergavimą.

Darbo rezultatų praktinė reikšmė

Tyrimų rezultatai taikyti tiriamuosiuose Lietuvos valstybinio mokslo ir studijų fondo ir Lietuvos mokslo tarybos projektuose:

- Prioritetinių Lietuvos mokslinių tyrimų ir eksperimentinės plėtros programoje „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.
- Aukštųjų technologijų plėtros programos projekte „Žmogaus genomo įvairovės ypatumų nulemti aterosklerozės patogenezės ypatumai (AHTHEROGEN)“; registracijos Nr.: U-04002; vykdymo laikas: 2004 m. 04 mėn. – 2006 m. 12 mėn.
- Aukštųjų technologijų plėtros programos projekte „Informacinės klinikinių sprendimų palaikymo ir gyventojų sveikatinimo priemonės e. Sveikatos sistemai (Info Sveikata)“; registracijos Nr.: B-07019; vykdymo laikas: 2007 m. 09 mėn. – 2009 m. 12 mėn.
- Prioritetinių Lietuvos mokslinių tyrimų ir eksperimentinės plėtros krypčių projekte „Genetinių ir genominių lūpos ir (arba) gomurio nesuaugimo pagrindų tyrimai (GENOLOG)“; registracijos Nr.: C-07022; vykdymo laikas: 2007 m. 04 mėn. – 2009 m. 12 mėn.
- Dvišalio bendradarbiavimo mokslo tyrimų ir eksperimentinės plėtros srityje Lietuvos – Prancūzijos integruotos veiklos programoje „Žiliberas“; registracijos Nr.: V-09059; vykdymo laikas: 2008 m. 04 mėn. – 2010 m. 12 mėn.

Ginamieji teiginiai

- Sammono algoritme projekcijos duomenų iniciacija ant tiesės yra netinkama, kadangi paklaidos konvergavimas iteracinio proceso pradžioje yra lėtas.
- Diagonalinis mažoravimo algoritmas paklaidos prasme nusileidžia daugiamačių skalių SMACOF algoritmui ir santykinėms daugiamatėms skalėms. DMA paklaida gaunama didesnė už SMACOF ir santykinių daugiamačių skalių algoritmo paklaidą, tačiau DMA yra greitesnis už SMACOF algoritmą.
- Stačiakampės formos SOM tinklo, kurio didesniąją briauną sudarančių neuronų yra k', permokomų neuronų skaičius laiptiškai mažėja didėjant mokymo epochos eilės numeriui ir sumažėja vienetu po $e' = \left[\frac{n'e}{k'}\right] - \left[\frac{(n'-1)e}{k'}\right]$ $(n' = 1, \dots, k' - 2)$ epochos.
- Galimos naujos atstumų funkcijos, kurios žymiai pagerina RPM algoritmo veikimą.
- Pradinių taškų parinkimo pagal didžiausias dispersijas būdas, daugiamačių skalių algoritmuose yra vienas tiksliausių ir efektyviausių.

Darbo rezultatų aprobavimas

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

1. Mathematical Modelling and Analysis 8-th International Conference. 2003 m. gegužės 28–31 d., Trakai, Lietuva.
2. Lietuvos matematikų draugijos XLIV konferencija. 2003 m. birželio 19–20 d., Vilnius, Vilniaus pedagoginis universitetas, Lietuva.
3. Kompiuterininkų dienos – 2003. 2003 m. rugpjūčio 28–30 d., Vilnius, Lietuva.
4. Informacinės technologijos 2004. 2004 m. sausio 28–29 d., Kaunas, Kauno technologijos universitetas, Lietuva.
5. EURO Summer Institute (ESI-XXII), 2004 m. liepos 9–25 d., Ankara, Middle East Technical University, Turkija.

6. Lietuvos matematikų draugijos XLV konferencija. 2004 m. birželio 17–18 d, Kaunas, Lietuvos žemės ūkio universitetas, Lietuva.
7. Lietuvos akių gydytojų suvažiavimas. 2005 m. lapkričio 4–5 d., Palanga, Lietuva.
8. Lietuvos jaunųjų mokslininkų konferencija, Operacijų tyrimas ir taikymai (LOTD – 2006), 2006 m. gegužės 26 d., Vilnius, Lietuva.
9. The 8th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2006. 2006 m. birželio 25–29 d., Zakopanė, Lenkija.
10. The 5th International Conference, Simulation and optimisation in business and industry, 2006 m. gegužės 17–20 d., Talinas, Estija.
11. Lietuvos jaunųjų mokslininkų konferencija, Operacijų tyrimas ir taikymai“ (LOTD – 2006), 2006 m. gegužės 26 d., Vilnius, Lietuva.
12. 11th Conference on Artificial Intelligence in Medicine (AIME 07), Doctoral Consortium, 2007 m. liepos 07–11 d., Amsterdamas, Olandija.
13. Informatikos doktorantų vasaros mokykla, Modernios duomenų gavybos ir analizės technologijos, 2007 m. rugsėjo 9–15 d., Druskininkai, Lietuva.
14. The 20th International Conference, EURO Mini Conference „Continuous Optimization and Knowledge-Based Technologies“ (EurOPT-2008), 2008 m. gegužės 20–23 d., Neringa, Lietuva.

Disertacijos struktūra

Disertaciją sudaro įvadas, trys skyriai ir išvados.

Darbo apimtis yra 105 puslapiai, neskaitant priedų, tekste panaudotos 57 numeruotos formulės, 29 paveikslai ir 12 lentelių. Rašant disertaciją buvo panaudotas 107 literatūros šaltinis.

1

Daugiamačių duomenų netiesinės projekcijos metodai

Šiame skyriuje ne tik apžvelgiami disertacijoje analizuoti netiesinės projekcijos metodai ir algoritmai daugiamačių duomenų vizualizavimui, bet ir išplėstos jų galimybės.

Su šiame skyriuje pateikta medžiaga susiję visi autoriaus publikuoti straipsniai (A1–B6).

Daugiamačių skalių atsiradimas siejamas su psichometrija, todėl pradžioje daugiamatės skalės buvo kuriamos ir tobulinamos psichometrų arba matematikų ir statistikų, dirbančių psichometrijos srityje. Bet greitai šis metodas buvo perimtas ir kitų sričių specialistų, tokių kaip geografų ar biržos prekiautojų, o dar vėliau - astronomų, genetikų, chemikų ir kt. (Torgenson 1952).

Bendrai daugiamatės skalės apima algoritmus, ieškančius paslėptų reguliarių struktūrų daugiamačiuose duomenyse. Duomenys - tai surikiuoti rinkiniai parametrų ir savybių, apibūdinančių kokį nors objektą ar reiškinį; jie gali būti interpretuojami kaip taškai arba vietos vektoriai daugiamatėje erdvėje. Šių algoritmų tikslas: objektų artimumo (*angl. proximity*) matricą, sukonstruotą pagal objektų parametrų dydžius, atvaizduoti į mažesnės dimensijos erdvės taškų atstumų matricą. Priklausomai nuo objektų kilmės, gali būti naudojamos tiek jų

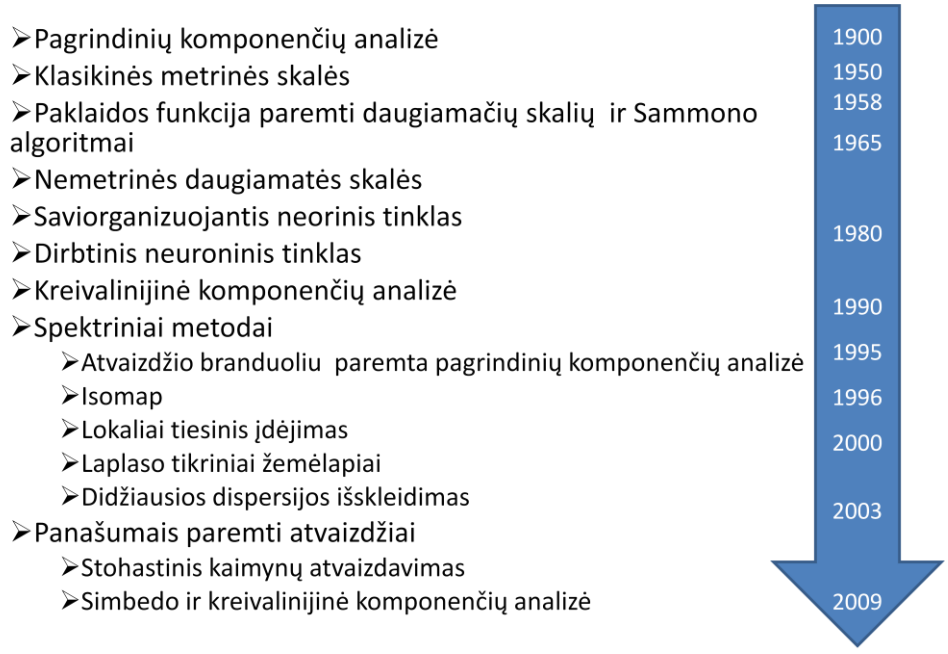
artimumo, tiek jų nepanašumo matricos (Cox and Cox 2001), nes, naudojant monotoniškai mažėjančią funkciją, galima nesunkiai pereiti nuo panašumų prie objektų tarpusavio nepanašumų.

Matematiniai pagrindai daugiamatėms skalėms buvo padėti 1958–1980 metais. Torgersonas (Torgerson 1958) sukūrė klasikinių daugiamačių skalių metodą (*angl. classical scaling*), kurio sprendinys sutampa su pagrindinių komponenčių analizės metodo sprendiniu. Klasikinių DS metodas remiasi tuo, kad objektų nepanašumai matuojami griežtai pagal Euklido metriką, nenaudojant jokių papildomų nepanašumų transformacijų (plačiau apie tai 1.3.1 skyriuje). 1977 de Leeuv (*orig. Jan de Leeuw*) apibendrino tuo metu plačiai tyrinėtų nemetrinių DS metodų principą ir pasiūlė naują metodą metrinėms daugiamatėms skalėms (de Leeuw 1977), pagrįstą sudėtingos funkcijos mažoravimo principu, vadinamu SMACOF (*angl. Scaling by MAjorizing a COmplicated Function*) arba metriniu SMACOF. Vėliau šis metodas patobulintas, pritaikant jį ne tik simetrinėms kvadratinėms nepanašumų matricoms, bet ir stačiakampėms nepanašumo matricoms. Taip pat atsirado nemetrinis SMACOF ir SMACOF su konfigūraciniais apribojimais (*angl. SMACOF with restrictions on the configurations*) bei individualių skirtumų SMACOF (*angl. SMACOF for individual differences*) (de Leeuw and Mair 2008). Prie daugiamačių skalių metodų matematinių pagrindų vystymo stipriai prisidėjo Gutmanas (*orig. Lois Guttman*) sukūręs Gutmano transformaciją (Guttman 1968) ir pasiūlęs, kaip ir Kruskalas (*orig. J. B. Kruskal*), algoritmą nemetrinių daugiamačių skalių problemos sprendimui (Guttman 1968; Kruskal 1964b).

Daugiamatės skalės priklauso dimensijos mažinimo (*angl. dimensionality reduction*) metodų klasei, kurios vystymąsi apžvelgia J. A. Lee savo pateiktyje (Lee and Verleysen 2010) ir kurios apibendrinimas pateiktas 1.1 paveiksle.

Daugiamačių skalių algoritmai ir jų taikymo sritys yra plačiai aprašyti (Borg and Groenen 2005), (Lee and Verleysen 2007) ir (Cox and Cox 2001) knygose. Šios knygos yra kertinės, norint suvokti daugiamačių skalių kilmę, vystymosi ištakas bei perspektyvas.

Lietuvoje daugiamačių duomenų vizualizavimas nėra nauja mokslo sritis ir joje dirba nemažai mokslininkų. Lietuvoje daugiamates skales pirmasis pradėjo nagrinėti prof. habil. dr. V. Šaltenis (Šaltenis and Varnaitė 1975). Jis kartu su savo kolege J. Valevičiene 1975 m. Valstybiniam algoritmų ir programų fondui pateikė Sammono projekcijos programinę realizaciją (Šaltenis 1975). Lygiagrečiai Sammono projekcijos tyrimus sėkmingai vykdė ir A. M. Montvilas. Taip pat prie daugiamačių algoritmų tyrinėjimo žymiai prisidėjo prof. A. Žilinskas, prof. G. Dzemyda, dr. (HP) J. Žilinskas bei jų auklėtiniai.



1.1 pav. Netiesinių dimensijos mažinimo metodų istorinė apžvalga (Lee and Verleysen 2010)

A. Žilinskas ir J. Žilinskas tyrinėja daugiamates skales su miesto kvartalų (*angl. city-block*) metrika ir ieško daugiamačių skalių paklaidos (įtempimo, tikslo) (*angl. Stress*) funkcijos globalaus minimumo. Kadangi paklaidos (1.2) funkcija su miesto kvartalų metrika tampa nediferencijuojama funkcija minimumo taškuose ($d \geq 2$) (Žilinskas and Žilinskas 2007), todėl klasikiniai Niutono–Rapsono (*angl. Newton-Rapson*) tipo metodai netinka. Siekiant surasti globalų minimumo tašką, naudojami kombinatoriniai, genetiniai, daugelio startų (*angl. multistart*) paieškos metodai. Tačiau augant taškų skaičiui ir projekcijos erdvės dimensijai d, uždavinio sudėtingumas auga eksponentiškai ir surasti globalų minimumą kai $n \cdot d > 24,\ d \geq 2$, tampa labai sunku arba neįmanoma (Žilinskas and Žilinskas 2009). Šiai problemai spręsti siūlomi šakų rėžių (Žilinskas and Žilinskas 2009) ir dviejų lygių hibridiniai (Žilinskas and Žilinskas 2007; Žilinskas and Žilinskas 2008) algoritmai, kurie ypač efektyvūs dirbant su didelės apimties duomenimis.

Pastaruoju metu sėkmingai apginta eilė disertacijų, skirtų daugiamačių duomenų vizualizavimo problemoms: A. Podlipskytė (Podlipskytė 2004), O. Kurasova (Kurasova 2005), V. Medvedev (Medvedev 2007),

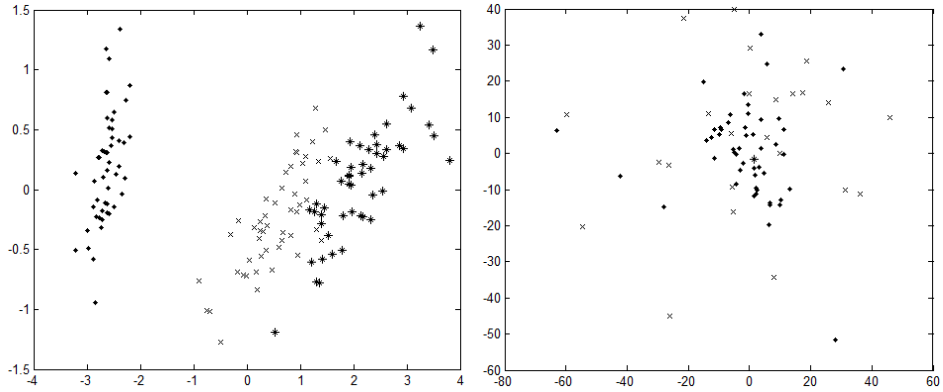
J. Bernatavičienė (Bernatavičienė 2008), S. Ivanikovas (Ivanikovas 2009). Pirmajai vadovavo prof. A. Žilinskas, likusioms prof. G. Dzemyda.

Išleistas vadovėlis informatikos krypties doktorantams ir magistrantams (Dzemyda *et al.* 2008).

1.1. Pagrindinės sąvokos

Šiame skyriuje nagrinėjami metodai, leidžiantys duomenų vektorius $X_i, X_j \in R^n$ atvaizduoti (suprojektuoti, kai $d < n$) į d matmenų erdvę R^d taip, kad šioje erdvėje būtų išlaikomi duomenų vektorių artimumai arba skirtingumai. Kitais žodžiais tariant, turime atvaizdavimą $\varphi: R^n \to R^d$.

Nepanašumams arba skirtumingumams (*angl. dissimilarity*) įvertinti erdvėse R^n ir R^d apibrėžiamos atitinkami nepanašumo matai δ_{ij} ir d_{ij}. Siekiant įvertinti bendrą duomenų aibės X ir jų vaizdų Y aibės panašumą naudojamos įvairios paklaidos funkcijos. Dažnai uždavinys yra dar sudėtingesnis, nes uždavinio pradžioje yra žinoma tik nepanašumų (ar panašumų) matrica $\Delta \equiv (\delta_{ij})$.



1.2 pav. „Iris“ duomenų aibė atvaizduota naudojantis klasikinėmis daugiamatėmis skalėmis. Kairėje matas yra Euklidinis atstumas, o dešinėje matas yra koreliacija (žr. 1.1 lentelę)

Varijuojant skirtingomis nepanašumo mato funkcijomis bei paklaidos funkcijomis gaunami vaizdai gali skirtis iš esmės (1.2 pav.), bet kartu tai leidžia pabrėžti vizualizuojamų duomenų skirtumingumus ar topologines savybes, nuspėti jų tikrąją (*angl. intrinsic*) dimensiją. Dažniausiai tyrinėjamos R^2 ir R^3 projekcijos erdvės ir naudojamas Euklidinis atstumas, kaip nepanašumo matas tarp elementų, nes taip analizuojat vektorių projekcijos vaizdus, žmogaus akis

gali geriausiai vizualiai įvertinti šių projekcijos taškų skirtumus arba panašumus (Morrison *et al.* 2003).

1.1. Apibrėžimas. Transformacija (atvaizdis) $\varphi: R^n \to R^d$ apibrėžta visoje R^n yra tiesinė jeigu galioje lygybės:

1. $\varphi(X_i + X_j) = \varphi(X_i) + \varphi(X_j)$, $\forall X_i, X_j \in R^n$ (adityvumas);
2. $\varphi(a \cdot X_i) = a \cdot \varphi(X_i)$, $\forall X_i \in R^n$ ir $a \in R$ (pirmo laipsnio homogeniškumas);

Tokia transformacija vadinama tiesinė arba tiesiniu atvaizdžiu (Trench 2010).

Tiesinės projekcijos metodo (tiesinio atvaizdžio) pavyzdys yra pagrindinių komponenčių analizė ir klasikinės metrinės daugiamatės skalės. Šie metodai yra ekvivalentūs, kuomet nepanašumai apskaičiuojami naudojant Euklido metrika (Lee and Verleysen 2007).

Daugiamačių duomenų netiesinės projekcijos metodai pasižymi tuo, kad transformacijai $\varphi: R^n \to R^d$ negalioja adityvumo savybė. Daugiamačių skalių atveju funkcija φ nėra apibrėžiama, tačiau minimizuojant paklaidos funkciją gaunamai projekcijai Y ir pradiniams duomenims X, nebegalioja tiesinio atvaizdžio (transformacijos) savybės.

Kaip minėta ankščiau, daugiamatės skalės skirstomos į metrines ir nemetrines priklausomai, ar tarp nagrinėjamos aibės X elementų yra įvesta metrika, ar ne. Dažniausiai nagrinėjami duomenys, priklausantys R^n edvei, kurioje yra įvesta metrika.

Metrinės daugiamatės skalės nuo nemetrinių skiriasi tuo, kad kai nagrinėjame nemetrines skales, turime situaciją, kai dalis skirtumų ar panašumų tarp objektų yra nežinomi, nes nėra žinomi visi objekto atributai arba žinomas tik jų tarpusavio išsidėstymas (eilė). Ankstyvieji daugiamačių skalių metodai buvo nemetrinių DS tipo, kadangi jos kilo būtent iš daugiamačių skalių taikymo psichologiniams testams įvertinti, kur duomenys dažnai yra kokybinio tipo ar nepilni (Kruskal 1964b; Guttman 1968).

Siekiant tiksliau apibrėžti skirtumus tarp metrinių ir nemetrinių daugiamačių skalių, reikia apibrėžti keletą sąvokų:

1.2. Apibrėžimas. Tegu turime aibę S, turinčią bent du elementus, ir nepanašumo (*angl. dissimilarity*) funkciją $\delta: S \times S \to R$, turinčią tokias savybes $\forall a, b \in S$:

1. $\delta(a, b) \geq 0$;
2. $\delta(a, a) = 0$;
3. $\delta(a, b) = \delta(b, a)$, komutatyvumas.

Tuomet funkcija δ vadinama nepanašumo funkcija, o jos reikšmė vadinama dviejų objektų nepanašumu arba skirtumingumu (žymima $\delta_{ij} \coloneqq \delta(a,b)$) (Critchley 2000).

Tam kad funkciją δ būtų galima vadinti metrika erdvėje S, reikia kad galiotų ir kitos metrikos savybės:

4. $\delta(a,b) \leq \delta(a,c) + \delta(c,b), \forall a,b,c \in S$, trikampio nelygybė;
5. $\delta(a,b) = 0$, tada ir tik tada, kai $a = b$.

Vadinasi, jeigu objektai priklauso metrinei erdvei, tuomet objektų nepanašumas gali būti prilyginami atstumams tarp šių taškų.

Porą (S,δ) vadinama nepanašumų erdve. Kadangi aibė S dažniausiai yra baigtinė arba yra didesnės aibės baigtinis poaibis, tai nepanašumus tarp visų tos aibės objektų galima sudėti į $[s \times s]$ simetrinę matricą $\Delta \equiv (\delta_{ij})$, kuri vadinama nepanašumų matrica.

Praktikoje dažnai objektų nepanašumai gaunami:

- tiesiogiai, pavyzdžiui atstumai tarp miestų ar skirtumai tarp reakcijos į dirgiklį psichometriniuose eksperimentuose ir kt.
- netiesiogiai, pavyzdžiui įvertis parametrų, atsižvelgiant į kuriuos du objektai skiriasi, ar kaip Mahalanobio atstumas tarp dviejų objektų.

Kartais neįmanoma išmatuoti nepanašumų tarp objektų arba jų tiesiog nežinome (pvz., nepanašumas tarp dviejų automobilių, darbuotojų ir t. t.), bet jeigu mokame nusakyti objektų artimumą p_{ij} (pvz., ekspertas surikiuoja skirtingų markių automobilius, priskirdamas jiems rikiavimo indeksą, tarkime automobilio patikimumo skalėje), tuomet galima naudojantis šiais artimumais taikyti nemetrinių daugiamačių skalių metodus ir atvaizduoti objektus.

Nemetrinių daugiamačių skalių metodo tikslas – vizualizuoti turimus objektų skirtumus (1.3 apibrėžimas).

1.3. Apibrėžimas. Dviejų objektų skirtumu arba neatitikimu (*angl. disparity*) vadinamas išvestinis dydis, monotoniškai priklausantis nuo dviejų objektų artimumo (*angl. proximity*).

Kuomet nepanašumų matricą galima apskaičiuoti arba jinai yra tiesiog duota, tuomet jos tyrimui galima taikyti nepanašumo analizės metodus. 1.3.3 skyriuje pateikiamas nemetrinių daugiamačių skalių metodo ir jame naudojamų monotoninių funkcijų išsamus aprašymas.

Nepanašumo analizės objektas yra matricos Δ reprezentacija kita matrica D, priklausančia pasirinktai nepanašumo matricų klasei $D \in D_D^{(s)}$. Dažnai $D_D^{(s)}$ priklauso visiems galimiems S atvaizdavimams į vizualizuojamą metrinę erdvę.

Jos šablonais gali būti:

1. R^d, su atstumų funkcija apibrėžiama Minkovskio p-norma ($1 \leq p \leq \infty$), arba bet kokia kita funkcija, tenkinanti skaliarinės sandaugos apibrėžimą.
2. Nekryptingai sujungto grafo teigiamų viršūnių aibė, kur d_{ij} lygus trumpiausiam keliui tarp viršūnių i ir j.

Apibendrinant, kiekvienai reprezentacinei klasei $D_D^{(n)}$ galima išskirti tris aspektus (Critchley 2000):

1. Vizualizavimas: kaip atrodo objektai, kurių komponentes mes žinome? Tai gali būti ir vektoriai Euklidinėje erdvėje, ir dendrogramos klasteriai.
2. Skaičiavimas: kaip apskaičiuojami d_{ij} tarp objektų projekcijos erdvėje?
3. Charakterizavimas: kada nepanašumų matrica Δ priklauso $D_D^{(s)}$? T. y. kuomet galima surasti vienareikšmišką atvaizdavimą Δ į D?

Siekiant surasti kaip įmanoma geresnį Δ atvaizdą D, reikalinga atstumo funkcija $\phi(\cdot,\cdot)$, kurią minimizuojant atžvilgiu Δ ($\phi(\Delta, D)$), būtų galima sakyti, kad surandamas geriausias atvaizdas D. Funkcija $\phi(\cdot,\cdot)$ vadinama paklaidos arba tikslo funkcija ir disertacijoje pateiktuose tyrimuose žymima sutrumpintai raide E. E papildomai žymima su indeksu, leidžiančiu tiksliau įvardinti funkciją (1.1) – (1.5).

Tegu $v(S)$ yra vektorius, kurio komponenčių skaičius $m = s(s-1)/2$ ir koordinatės yra simetrinės $(s \times s)$ matricos S viršutinės trikampės matricos elementai. Tuomet funkcija ϕ gali būti tokia: $\phi(\Delta, D) = \|v(\Delta - D)\|_p$ ($1 \leq p \leq \infty$), arba $\phi(\Delta, D) = v(\Delta - D)^T \cdot V \cdot v(\Delta - D)$, kur V vadinama svorių matrica $(m \times m)$. Kai svorių matrica V ekvivalenti vienetų matricai, turime primityvią (*angl. raw stress*) tikslo funkciją:

$$E_p(Y) = \sum_{\substack{i,j=1 \\ i<j}}^{s} \left(\delta_{ij} - d_{ij}(Y)\right)^2. \qquad (1.1)$$

$E_p(\mathrm{Y})$ minimizuoti galima naudojant paprastą mažiausių kvadratų, jungtinių gradientų ar kitus metodus.

Kai svorių matrica V yra diagonalinė, bei jos elementai yra bet kokie teigiami realūs skaičiai, tuomet turime bendresnį funkcijos $E_p(\mathrm{Y})$ atvejį, vadinamą tikslo funkcija su svoriais (*angl. weighted stress*) (1.2).

$$E_w(Y) = \sum_{\substack{i,j=1 \\ i<j}}^{s} w_{ij}\left(\delta_{ij} - d_{ij}(Y)\right)^2. \qquad (1.2)$$

Disertacijoje nagrinėjamas būtent šis paklaidos atvejis, kada yra ieškomas vektorių atvaizdavimas į Euklidinę vektorinę erdvę R^d.

Pats bendriausias atvejis, kai V matricos visi elementai yra teigiami realūs skaičiai. Tuomet ši tikslo funkcija vadinama apibendrinta (*angl. generalized*):

$$E_{ap}(Y) = \sum_{i<j}\left(\delta_{ij} - d_{ij}(Y)\right)\sum_{i<k\le j} w_{ik}\left(\delta_{ik} - d_{ik}(Y)\right). \qquad (1.3)$$

Kaip ir E_p(Y) atveju, funkcijas E_w(Y) ir E_{ap}(Y) galima minimizuoti atitinkamai naudojant mažiausių kvadratų metodą su svoriais ar apibendrintą mažiausių kvadratų metodą.

Kadangi funkcijos (1.1), (1.2) ir (1.3) priklauso nuo mastelio, t. y. pervedus duomenis iš vieno mastelio į kitą, paklaida proporcingai gali sumažėti arba padidėti, todėl įvedamas paklaidos normalizavimas arba santykinės paklaidos. Galimi du normalizavimo atvejai:

- Pirmuoju atveju, kai dalinama iš konstantos, lygios nepanašumų matricos Δ elementų sumai, gaunama normalizuota paklaida (*angl. normalized stress*):

$$E_{norm}(Y) = \frac{\sum_{\substack{i,j=1 \\ i<j}}^{s} w_{ij}\left(\delta_{ij} - d_{ij}(Y)\right)^2}{\sum_{\substack{i,j=1 \\ i<j}}^{s} w_{ij}\delta_{ij}^2}. \qquad (1.4)$$

- Antruoju atveju, kai dalinama iš gautos projekcijos objektų nepanašumo matricos D elementų sumos ir ištraukiama šaknis, gaunama literatūroje naudojama Kruskal paklaida (*angl. Kruskal stress, stress-1*) (Kruskal 1964b):

$$E_{Kruskal}(Y) = E_{stress-1}(Y) = \sqrt{\frac{\sum_{\substack{i,j=1 \\ i<j}}^{s} w_{ij}\left(\delta_{ij} - d_{ij}(Y)\right)^2}{\sum_{\substack{i,j=1 \\ i<j}}^{s} w_{ij}d_{ij}^2\,(Y)}}. \qquad (1.5)$$

Darbe (Borg and Groenen 2005) įrodyta, kad kai Y^* yra minimumo taškas ir tinkamai parinktas mastelis, tuomet $E_{norm}(Y^*) = E^2_{stress-1}(Y^*)$, todėl bet kuri iš paklaidos funkcijų (1.4) ir (1.5) gali būti naudojama ir viena keičiama kita.

Minimizuojant paklaidos funkcijas ieškoma, formos leidžiančios pamatyti tai, kas yra abstraktu ar nematoma, t. y. vizualizuoti. Norint vizualizuoti pasirinktus duomenis reikia atlikti visą eilę veiksmų: pradinių duomenų atrinkimas ir transformavimas, projekcijos algoritmo vykdymas, gautos projekcijos apdorojimas ir išvedimas į kompiuterio ekraną. Toks projekcijos apskaičiavimo procesas vadinamas vizualizavimu (Robertson and De Ferrari 1994). Disertacijoje nagrinėjamos projekcijos į vieno, dviejų ir trijų matmenų vektorines erdves, kurias nesunku atvaizduoti žmogui lengviau suvokiama forma.

Tolimesniuose skyriuose nagrinėsime įvairius galimus nepanašumo matricos Δ atvaizdavimo į matricą D metodus, jų privalumus ir trūkumus.

1.2. Pagrindinių komponenčių analizė

Pagrindinių komponenčių analizė (PKA, *angl. principal component analysis*) plačiai naudojama dimensijos mažinimui, duomenų suspaudimui atsisakant nereikšmingų parametrų, esminių savybių suradimui ir duomenų vizualizavimui (Jolliffe 2002). Disertacijoje pagrindinių komponenčių analize yra paremta viena iš pradinių duomenų iniciacijos strategijų, naudojama netiesiniuose daugiamačių duomenų projekcijos metoduose. Tai yra netiesioginis PKA taikymas duomenų dimensijos mažinimui ir jų vizualizavimui. PKA yra geriausias iš tiesinės projekcijos metodų, todėl rekomenduojama prieš pradedant taikyti sudėtingesnius projekcijos metodus pritaikyti būtent šį metodą, ir tik tuomet, jeigu gautas rezultatas netenkina, taikyti kitus metodus.

Pagrindinių komponenčių analizės algoritmas gaunamas remiantis ekvivalenčiais 1.4 ir 1.5 apibrėžimais.

1.4. Apibrėžimas. Pagrindinių komponenčių analizė – ortogonali duomenų projekcija į mažesnės dimensijos (matmenų skaičiaus) erdvę, kuri maksimizuoja suprojektuotų duomenų dispersiją (Hotelling 1933). Mažesnės dimensijos erdvė yra vadinama pagrindiniu poerdviu (*angl. principal subspace*).

1.5. Apibrėžimas. Pagrindinių komponenčių analizė – tiesinė projekcija, kuri minimizuoja vidutinę projekcijos paklaidą, apibrėžiamą kaip Euklidinių atstumų tarp duomenų ir jų projekcijų kvadratų vidurkis, apskaičiuojamas pagal (1.6) formulę (Pearson 1901).

$$E = \frac{1}{s}\sum_{i=1}^{s}\|X_i - Y_i\|^2, \tag{1.6}$$

čia $Y_i = \sum_{k=1}^{d} a_{ik}U_k + \sum_{k=d+1}^{n} b_i U_k$, kur $\{U_k\}$ ortogonalūs baziniai vektoriai, $\|\cdot\|$ – Euklidinis atstumas.

Pagal antrąjį apibrėžimą turime, kad PKA metodas yra tiesinė projekcija. Nors abu apibrėžimai ekvivalentūs, tačiau PKA dažniausiai siejamas su pirmuoju apibrėžimu, nes remiantis pirmuoju apibrėžimu buvo kuriami pagrindinių komponenčių analizės algoritmai, kurie susideda iš duomenų suvidurkinimo, jų kovariacinės matricos apskaičiavimo ir vėliau šios kovariacinės matricos (1.7) d tikrinių vektorių, atitinkančių d didžiausias tikrinės reikšmes, suradimo.

$$\Sigma = E\big(X - E(X)\big)\big(X - E(X)\big)^T. \tag{1.7}$$

Žinant tikrinius vektorius galima apskaičiuoti PKA projekciją. Pagrindinis uždavinys surandant PKA projekciją yra tikrinių vektorių apskaičiavimas, kuris gali būti vykdomas remiantis keletu strategijų:

- kovariacinės matricos dekompozicija (Golub and Van Loan 1996), kurios sudėtingumas $O(n^3)$. Efektyvesnis yra laipsninis metodas (*angl. power method*) (Golub and Van Loan 1996), kurio sudėtingumas $O(dn^2)$;
- EM (*angl. expectation-minimization*) algoritmas yra naudojamas, kai duomenų komponenčių skaičius n yra labai didelis, tuomet vien kovariacinės matricos apskaičiavimas reikalauja $O(sn^2)$ operacijų. Šis algoritmas laimi būtent todėl, kad jame neskaičiuojama kovariacinė matrica ir jo sudėtingumas tampa lygus $O(dsn)$ (Roveis 1998);

PKA siekia surasti „teisingą“ projekcijos kryptį, kuria atvaizduoti duomenys atskleidžia savo struktūrą. Jeigu duomenų dimensija yra didelė arba jie priklauso netiesinei daugdarai, tuomet tokios krypties dažnai surasti neįmanoma (Friedman and J.W 1974).

Patys metodai detaliai aprašyti (Dzemyda *et al.* 2008; Jolliffe 2002; Bishop 2006) knygose.

1.3. Daugiamatės skalės

Daugiamatės skalės (DS) – tai metodas, kuris atvaizduoja nepanašumus (arba panašumus) tarp dviejų objektų kaip atstumą tarp šių taškų žemesnės dimensijos

erdvėje (projekcijos erdvėje). Tokios žemesnės dimensijos erdvės pavyzdžiu yra trimatė erdvė, dvimatė Euklidinė plokštuma ar tiesė. Stebint gautą projekciją Euklidinėje plokštumoje, galima spręsti apie objektų tarpusavio santykį bei jų formuojamas struktūras, kadangi laikoma, kad mažai nutolę vienas nuo kito taškai projekcijos erdvėje atspindi panašius ir artimus pagal savo požymius objektus. Daugiamatės skalės naudojamos tirti kaip objektų požymiai, išskiriantys juos iš kitų objektų, įtakoja jų tarpusavio padėtį projekcijos erdvėje, o taip pat norint surasti daugiamatės erdvės dimensiją, kuri yra pakankama norint įvertinti objektų panašumą ar nepanašumą (Borg and Groenen 2005). Simetrija tiesės atžvilgiu ir pasukimas yra invariantinės transformacijos DS algoritme.

Daugiamačių skalių teorijos pradininkai buvo matematikai psichologai, kurių pagrindiniai rezultatai paskelbti žurnale Psihometrika (*angl. Psychometrika*) (Chen *et al.* 2008).

Kaip jau paminėta šio skyriaus pradžioje, daugiamatės skalės buvo, tyrinėjamos matematikų, dirbusių psichometrijos srityje. Knygoje (Young and Hamer 1987) galima surasti pagrindinius žingsnius ankstyvojoje daugiamačių skalių vystymosi istorijoje. Keletas faktų susijusių su disertacijoje tyrinėjamų metodų vystymosi:

1. Pirmas žingsnis – metrinės daugiamatės skalės:
   - (Torgerson 1958) pirmas pateikė taip vadinamą daugiamačių skalių siūlymą, tenkinantį Euklidinės edvės modelį.
   - (Attneave 1950) straipsnyje pasiūlė ne-Euklidinės erdvės modelį tenkinančias daugiamates skales.
2. Antras žingsnis – nemetrinės daugiamatės skalės:
   - (Kruskal 1964b) pateikė matematiškai ir statistiškai racionalų siūlymą nemetrinėms daugiamatėms skalėms.
   - (Guttman 1968) pasiūlė visiškai naują metodą nemetrinėms daugiamatėms skalėms, pagrįstą Gutmano transformacija.
3. Trečias žingsnis – individualių skirtumų daugiamatės skalės:
   - (Bloxom 1968) pirmasis aprašė daugiamačių skalių modelį su svoriais.
4. Ketvirtas žingsnis – vystymosi konsolidavimas:
   - (de Leeuw 1977) ir (de Leeuw and Heiser 1977) straipsniuose pateikė SMACOF algoritmą, tinkantį tiek metrinėms, tiek ir

nemetrinėms daugiamatėms skalėms su svoriais ir be jų. Nepanašumų matas jame yra atstumai.

Šiais laikais daugiamačių skalių algoritmai taikomi naudojant nesimetrines nepanašumų matricas: lyginantys atskirų individų trokštamą objektų pasirinkimą (*angl. unfolding*) ar nepanašumų matricų aibes, apibūdinančias tą patį objektą skirtingose situacijose.

Daugiamačių skalių metodai yra efektyviai skaičiuojami ir sugebantys atskleisti daugiamačių duomenų struktūrą, kai duomenys yra išsidėstę tiesiniame ar beveik tiesiniame daugiamatės erdvės poerdvyje (Tenenbaum *et al.* 2000). Kuomet duomenys formuoja iš esmės netiesinę struktūrą, tuomet šios struktūros DS metodais atskleisti nepavyksta. Tais atvejais naudojami, taip vadinami, netiesiniai dimensijos mažinimo metodai (*angl. nonlinear dimensionality reduction*): kaip ISOMAP (Tenenbaum *et al.* 2000) ar lokaliai tiesinis atvaizdavimas (LLE) (Roweis and Saul 2000).

Kadangi ne visuomet galima surasti atvaizdį į mažesnės dimensijos erdvę, išlaikantį visus atstumus, todėl įvedama paklaidos funkcija, kurią minimizuojant atstumai atvaizduojami į kaip įmanoma artimesnius atstumams įvesties erdvėje. Taškų projekciją galima gauti tiek metrinių, tiek ir nemetrinių skalių metodais.

1.3.1. Klasikinės daugiamatės skalės

Klasikinės daugiamatės skalės dar kitaip vadinamos Torgersono ar Torgersono - Goverio (*angl. Torgerson-Gower scaling*) skalėmis. Torgersonas ir Goveris yra pirmieji autoriai, praktiškai pritaikę (Young and Householder 1938) teoremas susijusias su matricos dekompozicijos egzistavimu.

Pagrindinė klasikinių daugiamačių skalių idėja – nepanašumai tarp objektų yra lygūs atstumams tarp šių objektų ,ir tikslas yra surasti šių objektų koordinates. Šis uždavinys susiveda į Gramo (*angl. Gramian*) matricos (1.8) tikrinių reikšmių dekompoziciją (*angl. eigendecomposition*). Pagrindinių komponenčių analizės metodas (aptartas 1.2 skyriuje) atlieka vektorių X_i kovariacinės matricos tikrinių reikšmių dekompoziciją. Nors kovariacinė matrica $(n \times n)$ ir Gramo matrica $(s \times s)$ yra skirtingų dydžių, bet esant centruotai X, nesunku, remiantis įvesties matricos X dekompozicija, įrodyti, kad klasikinių skalių sprendinys sutampa su pagrindinių komponenčių sprendiniu (Lee and Verleysen 2007):

$$G = XX^T. \tag{1.8}$$

Uždavinio suvedimas į Gramo matricą pateiktas (Borg and Groenen 2005). Trumpai tariant, atstumų kvadratų matrica $D^{[2]}(X)$ pagal prielaidą yra lygi turimai nepanašumų matricai Δ, kurios kiekvienas elementas pakeltas kvadratu

$\Delta^{[2]}$. Be to, galioja (1.9) lygybės. Pirmoji lygybė vadinama dvigubo matricos centravimo (*angl. double centering*) procedūra, antroji atitinka matricos G_Δ tikrinių reikšmių dekompoziciją.

$$G_\Delta = -\frac{1}{2} J \Delta^{[2]} J^T = Q \Lambda Q^T. \quad (1.9)$$

Žinant Q ir Λ, bei pasirinkus norimą didžiausių teigiamų tikrinių reikšmių skaičių d, galima vienareikšmiškai suskaičiuoti matricą Y, t. y. surasti projekciją į R^d (1.10).

$$Y = Q_+ \Lambda_+^{1/2}, \quad (1.10)$$

čia $\Lambda_+ - d$ didžiausių teigiamų tikrinių reikšmių matrica, Q_+ – šias tikrines reikšmes atitinkančių tikrinių vektorių matrica.

Kadangi imamos tik teigiamos tikrinės reikšmės ir $d < n$, todėl taikant klasikines daugiamates skales neišvengiamai gaunama paklaida, kurią galima įvertinti pagal (1.11) formulę. Ši paklaida vadinama ištempimo (*angl. strain*) paklaida.

$$E_{ištempimo} = \|YY^T - G_\Delta\|^2. \quad (1.11)$$

Pagrindinis klasikinių daugiamačių skalių metodo privalumas yra tas, kad jis neturi parametrų, nuo kurių priklausytų, ir todėl gaunama vienareikšmė taškų projekcija. Vienas iš pagrindinių trūkumų, kaip ir PKA metode, yra tai, kad klasikinių daugiamačių skalių metodas yra tiesinis atvaizdis. Vadinasi, jeigu projektuojama daugdara yra netiesinė, iš gautos projekcijos negalima atskleisti duomenų topologijos.

1.3.2. Metrinės daugiamatės skalės

Metrinės daugiamatės skalės tam tikra prasme yra klasikinių daugiamačių skalių praplėtimas.

Daugiamačių skalių metodu galima atvaizduoti s vektorių $X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ iš n - matės erdvės R^n į mažesnio matmenų skaičiaus erdvę R^d taip, kad kuo geriau būtų išsaugoti projekcijos vektorių $Y_i = \{y_{i1}, y_{i2}, \dots, y_{in}\}$ nepanašumai arba skirtumingumai, apskaičiuoti lyginant atitinkamus projektuojamus vektorius X_i. Dažniausiai naudojama projekcijos (vaizdo) erdvė yra R^2. Tik metrinėse daugiamatėse skalėse nepanašumų arba skirtumingumų matu yra imamas atstumas tarp atitinkamus vektorius atitinkančių taškų, bet nepanašumu gali būti bet kuri funkcija tenkinanti 1.2. apibrėžimą. Lentelėje 1.1 pateiktas sąrašas dažniausiai naudojamų objektų nepanašumo funkcijų. Pažymėkime δ_{ij} skirtumingumą tarp i-tojo ir j-tojo

vektorių R^n erdvėje ir atitinkamai d_{ij} atstumą erdvėje R^d. Tuomet daugiamačių skalių metodo tikslas – surasti tokias taškų X_i projekcijas Y_i, kad atstumai tarp taškų erdvėje R^n kaip įmanoma tiksliau atitiktų atstumus tarp tų taškų projekcijų erdvėje R^d. Sąvoką „kaip įmanoma tiksliau“, matematiškai galima apibrėžti kaip paklaidą, apskaičiuojamą pagal formulę (1.1).

1.1. lentelė. Populiariausi kiekybinių duomenų artimumo matai

Mato pavadinimas	**Matematinė išraiška**
Euklidinis atstumas	$\delta_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2}$
Euklidinis atstumas su svoriais	$\delta_{ij} = \sqrt{\sum_{k=1}^{n} w_k \left(x_{ik} - x_{jk}\right)^2}$
Mahalanobio (*angl. Mahalanobi*) atstumas	$\delta_{ij} = \sqrt{\left(x_{ik} - x_{jk}\right)^T \Sigma^{-1} \left(x_{ik} - x_{jk}\right)}$
Miesto kvartalų atstumas	$\delta_{ij} = \sum_{k=1}^{n} \left\lvert x_{ik} - x_{jk} \right\rvert$
Minkovskio (*angl. Minkowski*) atstumas	$\delta_{ij} = \left(\sum_{k=1}^{n} \left\lvert x_{ik} - x_{jk} \right\rvert^{\lambda} \right)^{\frac{1}{\lambda}}, \lambda \geq 1$
Kanbera (*angl. Canberra*) atstumas	$\delta_{ij} = \sum_{k=1}^{n} \frac{\left\lvert x_{ik} - x_{jk} \right\rvert}{\left\lvert x_{ik} + x_{jk} \right\rvert}$
Bray-Curtis atstumas	$\delta_{ij} = \sum_{k=1}^{n} \frac{\left\lvert x_{ik} - x_{jk} \right\rvert}{\left(x_{ik} + x_{jk}\right)}$
Nukrypimas (*angl. divergence*)	$\delta_{ij} = \frac{1}{n} \sum_{k=1}^{n} \frac{\left(x_{ik} - x_{jk}\right)^2}{\left(x_{ik} + x_{jk}\right)^2}$
Kampinis atskyrimas (*angl. angular separation*)	$\delta_{ij} = 1 - \frac{\sum_{k=1}^{n} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{n} x_{ik}^2 \sum_{k=1}^{n} x_{jk}^2}}$
Koreliacija	$\delta_{ij} = 1 - \frac{\sum_{k=1}^{n} (x_{ik} - \bar{x}_i)\left(x_{jk} - \bar{x}_i\right)}{\sqrt{\sum_{k=1}^{n} (x_{ik} - \bar{x}_i)^2 \sum_{k=1}^{n} \left(x_{jk} - \bar{x}_i\right)^2}}$

Kiekvienas iš pateiktų artimumo matų turi ir savo taikymo sritį, pavyzdžiui: Bray-Curtis atstumas dažnai naudojamas ekologijoje; koreliacija ir kampinio atskyrimo matai naudojami, kai duomenų komponenčių skaičius yra nedidelis; miesto kvartalų metrika tinka, kai norima palyginti objektą aprašančius atskirus požymius; Euklido metrika plačiausiai naudojama, nes ji atspindi dviejų objektų tarpusavio padėtį plokštumoje. Šios metrikos trūkumas, kad jos rezultatui didesnę įtaką turi didžiausias absoliutines reikšmes turintys vektoriaus elementai. Atstumo tarp taškų absoliutus dydis turi didelę įtaką paklaidos E_{DS} (1.1) reikšmei. Šią įtaką galima sumažinti naudojantis (1.2) formule.

Literatūroje minimi įvairūs paklaidos funkcijos E_{DS} (1.1) optimizavimo būdai, tokie kaip jungtinių gradientų metodas, kvazi-Niutono (*angl. Quasi-Newton*) metodas, deterministinis atkaitinimo modeliavimo algoritmas (*angl. simulated annealing*) (Klock and Buhmann 1999), evoliucinis algoritmas (Dzemyda *et al.* 2008), kombinatorinis daugiamačių skalių algoritmas (Žilinskas and Žilinskas 2007), šakų ir rėžių algoritmas (Žilinskas and Žilinskas 2009), genetinio algoritmo ir lokalaus nusileidimo metodų kombinacijos (Mathar and Žilinskas 1993; Podlipskytė 2004), tuneliavimo (*angl. tunneling*) globalaus minimizavimo metodas (Groenen and Heiser 1996), atstumų glaistymo (*angl. distance smoothing*) metodas (Groenen *et al.* 1998), SMACOF algoritmas, dar vadinamas Gutmano (*angl. Guttman*) mažorizavimo algoritmu, kuris pagrįstas tikslo funkcijos mažorizavimo principu (Borg and Groenen 2005).

Šioje disertacijoje tyrimams naudojamas iteracinis SMACOF algoritmas ir šio algoritmo modifikacija, vadinama diagonaliniu mažoravimo algoritmu (DMA) (Trosset and Groenen 2005), kuri skirta atvaizduoti didesnėm duomenų aibėms. SMACOF algoritmas yra vienas iš geriausių optimizavimo algoritmų, taikomas E_{DS} paklaidos minimizavimui (Borg and Groenen 1997; Groenen and van de Vaelden 2004). SMACOF algoritmas yra paprastas, bet efektyvus, kadangi garantuoja paklaidos funkcijos konvergavimą į lokalų minimumą su tiesiniu konvergavimo greičiu (de Leeuw 1988).

SMACOF algoritmas išsamiai aprašytas (Borg and Groenen 2005). Kadangi tai vienas iš pagrindinių disertacijoje naudotų duomenų vizualizavimo algoritmų, trumpai bus aprašyti jo principai.

SMACOF algoritmas, tai iteracinis algoritmas, pirmą kartą pasiūlytas (Guttman 1968) straipsnyje ir matematiškai užrašomas yra

$$Y(m'+1) = V^{+}B\big(Y(m')\big)Y(m'), \qquad (1.12)$$

kur matricą $B\big(Y(m')\big)$ sudaro elementai b_{ij} apskaičiuojami pagal (1.13) formulę.

$$b_{ij} = \begin{cases} -\frac{w_{ij}\delta_{ij}}{d_{ij}}, & kai\ i \neq j\ ir\ d_{ij} \neq 0; \\ 0, & kai\ i \neq j\ ir\ d_{ij} = 0; \\ -\sum_{j=1, j\neq i}^{s} b_{ij}, & kai\ i = j. \end{cases} \quad (1.13)$$

Svorių matrica V yra

$$V = \begin{pmatrix} \sum w_{1j} & & & & \\ & \ddots & & -w_{ij} & \\ & & \ddots & & \\ & -w_{ij} & & \ddots & \\ & & & & \sum w_{sj} \end{pmatrix}. \quad (1.14)$$

V^+ yra matricos V Moore-Penrose pseudoinversinė matrica (Gower and Groenen 1991). Tuo atveju, kai visi svoriai $w_{ij} = 1$, tuomet $V^+ = \frac{1}{n}(I - \frac{1}{n}1 \cdot 1^T)$. Čia I yra vienetinė matrica. 1 yra matrica, kurios visi elementai lygūs vienam. Tuomet iteracinė formulė (1.12) tampa lygi:

$$Y(m'+1) = \frac{1}{n}B\big(Y(m')\big)Y(m') \quad (1.15)$$

SMACOF algoritmo schema pagal (Borg and Groenen 2005):

1. Pasirinktu metodu parenkami pradiniai vektoriai $Y_i \epsilon R^d$ ir priskiriama $m' = 0$.
2. Apskaičiuojama paklaida $E_{DS}\big(Y(m')\big)$ pagal (1.1) formulę.
3. Iteracijų skaičius m' padidinamas vienetu.
4. Apskaičiuojama Gutmano transformacija pagal (1.12) formulę ir gaunama nauja taškų projekcija $Y(m')$.
5. Apskaičiuojama daugiamačių skalių paklaida $E_{DS}\big(Y(m')\big)$.
6. Jeigu $E_{DS}\big(Y(m'-1)\big) - E_{DS}\big(Y(m')\big) < \varepsilon$ arba iteracijų skaičius $m' = m'_{max}$, tai algoritmas stabdomas, kitu atveju kartojamas algoritmas nuo 3 žingsnio.

Šis metodas buvo taikomas, kaip pagrindinis daugiamačių skalių metodas, (A1, A3, A4 ir A8) straipsniuose.

1.3.3. Nemetrinės daugiamatės skalės

Nemetrinės daugiamatės skalės buvo plėtojamos Šepardo (Shepard 1962a), vėliau jas vystė Kruskalas (Kruskal 1964a; Kruskal 1964b). Dažnai negalima turimų objektų artimumų pavadinti atstumais arba tiesiog nežinoma nepanašumų tarp objektų dydžių. Šiai problemai spręsti ir buvo pasiūlytos nemetrinės daugiamatės skalės. Čia remiamasi tuo, kad nors nepanašumų reikšmių nežinome, tačiau turime objektų rikiavimą, sudarytą pagal šių objektų nepanašumus. Pavyzdžiui, automobilių ekspertas gali surikiuoti turimą aibę automobilių pagal tam tikrą požymį (surinkimo kokybė, liekamoji vertė, ir t. t.), tačiau kokia to požymio reikšmė, išlieka neaišku.

Kai objektų artimumas p_{ij} (*angl. proximity*) apibrėžiamas eilės numeriu (rangu), tuomet ieškant atvaizdžio siekiama, kad jis tenkintų rikiavimo modelį (1.16).

Rikiavimo modelis (*angl. ordinal model*) reikalauja, kad:

$$jei\ p_{ij} < p_{kl}, tai\ d_{ij} \leq d_{kl}, \tag{1.16}$$

čia p_{ij}, p_{kl} – objektų X_i, X_j ir atitinkamai X_k, X_l artimumai (rangai), d_{ij}, d_{kl} – atstumai tarp šių objektų projekcijos taškų. Jei $p_{ij} = p_{kl}$, tai d_{ij} ir d_{kl} gali būti bet kokie.

Taigi nemetrinės DS skalės stengiasi išlaikyti rikiavimą projekcijoje kaip įmanoma „artimą“ (izometrinį) įvesties duomenų rikiavimui. Kadangi artimumai p_{ij} yra ne visada patogūs naudoti tiesiogiai, todėl jie transformuojami į skirtumus arba neatitikimus $\hat{\delta}_{ij} = f(p_{ij})$ (*angl. disparity*) (1.3 apibrėžimas).

Skirtumai $\hat{\delta}_{ij}$ yra apskaičiuojami iš artimumų p_{ij}, naudojant įvairias monotonines funkcijas f, pateiktas (1.17) – (1.21) išraiškomis.

Pirmo laipsnio polinomas:

$$\hat{\delta}_{ij} = a + b \cdot p_{ij}. \tag{1.17}$$

Antro laipsnio polinomas:

$$\hat{\delta}_{ij} = a + b \cdot p_{ij} + c \cdot p_{ij}^2. \tag{1.18}$$

Eksponentinė funkcija:

$$\hat{\delta}_{ij} = a + b \cdot exp(p_{ij}). \tag{1.19}$$

Logaritminė funkcija:

$$\hat{\delta}_{ij} = a + b \cdot log(p_{ij}). \tag{1.20}$$

Antro laipsnio polinomas:

$$\hat{\delta}_{ij} = a + b \cdot p_{ij} + c \cdot p_{ij}^2, \quad (1.21)$$

čia $a, b, c \in R$.

Padarius prielaidą, kad apskaičiuoti skirtumai $\hat{\delta}_{ij}$ apytiksliai lygūs atstumams (nepanašumams) tarp taškų daugiamatėje erdvėje δ_{ij}, gauname metrinėmis daugiamatėmis skalėmis sprendžiamą problemą. Tada taškų X projekciją galima apskaičiuoti standartiniais daugiamačių skalių algoritmais (SMACOF, pseudo-Niutono), vietoje nepanašumų naudojat skirtumingumus. Pačios projekcijos tikslumas vertinamas Kruskalo paklaida (1.5).

1.4. Sammono algoritmas

Lietuvoje Sammono algoritmas plačiai tyrinėtas O. Kurasovos (Dzemyda and Kurasova 2006), G. Dzemydos (Dzemyda *et al.* 2008) ir A. M. Montvilo (Montvilas 2003) darbuose.

Užsienyje Sammono algoritmas nagrinėjamas nuo pat 1969 metų. Šis algoritmas skirtas išsaugoti tiek mažus, tiek didelius atstumus tarp taškų, esančių daugiamatėje erdvėje taip, kad jie liktų kaip įmanoma panašesni į atstumus tarp taškų projekcijos plokštumoje (Sammon 1969). Dažnai teigiama, kad Sammono algoritmu gauta projekcija yra reikšmingai geresnė topologijos išsaugojimo prasme už SOM (Bezdek and Pal 1995). Pats Sammono algoritmas yra vienas iš greičiausiai veikiančių dimensijos mažinimo algoritmų (Biswas *et al.* 1981). Metodo kilmė, kaip ir daugelio tuo metu atsiradusių metrinių ir nemetrinių metodų, yra problemos, kylančios iš tuo metu egzistavusių labai populiarių, klasterizavimo algoritmų. Klasterizavimo algoritmai priklauso nuo daugelio parametrų, tokių kaip: klasterių panašumo mato, klasterio panašumo ribų, iteracijų skaičiaus ar ribos, nuo kurios priklauso, kiek bus klasterių surasta. J. W. Sammonas pasiūlė savo algoritmą, leidžiantį, autoriaus nuomone, išvengti visų šitų problemų arba, teisingiau, perkelti šias problemas ant žmonių, interpretuojančių gautas Sammono algoritmu projekcijas, pečių.

Tai yra vienas iš populiariausių vizualizavimo algoritmų, kuris iš tiesų yra daugiamačių skalių atskiras atvejis. Sammono algoritmo tikslo funkcija (paklaida) (*angl. Sammon stress*) (1.23), kurioje maži atstumai turi didesnį svorį negu dideli atstumai, gaunama iš (1.4) formulės, parenkant svorius pagal formulę:

$$w_{ij} = \frac{1}{\delta_{ij} \sum_{\substack{i=1 \\ i<j}}^{s} \delta_{\mathrm{ij}}}. \quad (1.22)$$

Tada galima užrašyti:

$$E_S = \frac{1}{\sum_{\substack{i=1 \\ i<j}}^{s} \delta_{ij}} \sum_{\substack{i=1 \\ i<j}}^{s} \frac{\left(\delta_{ij} - d_{ij}\right)^2}{\delta_{ij}}, \tag{1.23}$$

čia $\sum_{\substack{i=1 \\ i<j}}^{s} \delta_{ij}$ naudojama kaip normalizavimo konstanta, pašalinanti paklaidos priklausomybę nuo duomenų mastelio. Nepanašumai d_{ij} ir δ_{ij} yra lygūs atstumams tarp taškų tiek projekcijos, tiek pradinėje erdvėse. Atstumai gali būti apskaičiuojami pagal bet kokią metriką, bet autorius siūlo naudoti tiktai Euklidinę metriką (Sammon 1969).

Sammono algoritmo paklaidai minimizuoti naudojamas iteracinis gradientinis pseudo–Niutono (*orig. pseudo–Newton*) metodas, pagrįstas Hesiano (*angl. Hessian*) matricos diagonale aproksimacija. Projekcijos taškų $Y_i \in R^2$ koordinatės $y_{ik},\ i = \overline{1,s},\ k = 1{,}2$ apskaičiuojamos pagal iteracinę formulę:

$$y_{ik}(m'+1) = y_{ik}(m') - \alpha \frac{\dfrac{\partial E_S(m')}{\partial y_{ik}(m')}}{\left|\dfrac{\partial^2 E_S(m')}{\partial y_{ik}^2(m')}\right|}, \tag{1.24}$$

čia m' - iteracijos numeris, o koeficientas α vadinamas žingsnio ilgiu arba „magišku faktoriumi“ (Kohonen T. 2001; Sammon 1969).

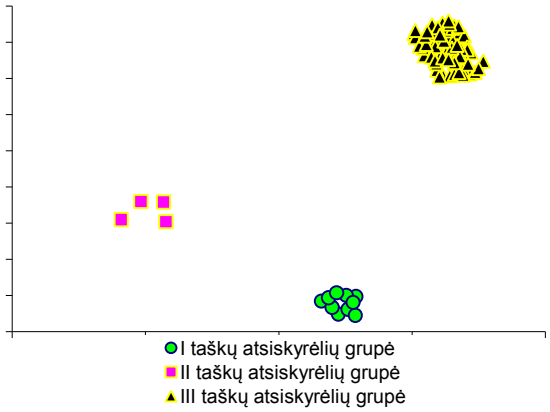
Pirmos ir antros eilės dalinės išvestinės apskaičiuojamos pagal formules:

$$\frac{\partial E_S}{\partial y_{ik}} = -\frac{2}{\sum_{\substack{i=1 \\ i<j}}^{s} \delta_{ij}} \sum_{j=1, j\neq i} \left(\frac{\delta_{ij} - d_{ij}}{\delta_{ij} \cdot d_{ij}}\right)\left(y_{ik} - y_{jk}\right), \tag{1.25}$$

$$\frac{\partial^2 E_S}{\partial y_{ik}^2} = -\frac{2}{\sum_{\substack{i=1 \\ i<j}}^{s} \delta_{ij}} \sum_{j=1, j\neq i} \frac{1}{\delta_{ij} \cdot d_{ij}} \left(\left(\delta_{ij} - d_{ij}\right) - \frac{(y_{ik} - y_{jk})^2}{d_{ij}} \left(1 + \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right) \right). \tag{1.26}$$

Rekomenduojama žingsnio reikšmė, užtikrinanti pakankamai gerą Samono paklaidos konvergavimą į minimumą yra intervale $\alpha \in [0{,}3;\ 0{,}4]$ (Kohonen T. 2001; Sammon 1969), naudojant klasikinį Sammono algoritmą. Priklausomybė nuo žingsnio α sumažėja, kai naudojamas modifikuotas Sammono algoritmas, kurio metu naujos taškų koordinatės $y_{ik}(m'+1)$ apskaičiuojamos ne tik atsižvelgiant į ankstesnės iteracijos taškų koordinates $y_{ik}(m')$ bet ir į vykdomos

iteracijos metu gautų taškų koordinates. Zeidelio (*Philipp Ludwig von Seidel*) arba Gauso-Zeidelio (*Gauss-Seidel*) metodas yra naudojamas tiesinių lygčių sistemomis spręsti, bet originali idėja, kurią galima pritaikyti ir DS algoritme, yra ta, kad nauji taškai iteracinio proceso eigoje apskaičiuojami remiantis jau prieš tai, toje pačioje iteracijoje, apskaičiuotais taškais. Tačiau daugiamačių skalių atveju tai reikštų, kad reikia perskaičiuoti ir atstumų matricą po kiekvieno pakeisto taško. Nors toks metodo konvergavimo greitis išauga, tačiau išauga ir laiko sąnaudos. Todėl yra pasiūlyta modifikacija, kuri skiriasi nuo Gauso-Zeidelio tipo modifikacijos tuo, kad pirmos ir antros eilės išvestinės perskaičiuojamos tik kiekvienos iteracijos pabaigoje (Dzemyda and Kurasova 2006). Tai nereikalauja papildomų skaičiavimų pirmos ir antros eilės išvestinėms perskaičiuoti ir tuo pačiu padidina projekcijos paklaidos konvergavimo greitį. Sammono projekcijos pavyzdys pateiktas 1.3 paveiksle. Jame nėra vaizduojamos taškų koordinatės koordinačių ašyse, kadangi Sammono algoritmas nepriklauso nuo pasukimo, perkėlimo ir kitų tiesinių transformacijų, o tai lemia ekvivalenčias projekcijas įvairiose plokštumos vietose.



1.3 pav. Modifikuoto Sammono algoritmo projekcijos pavyzdys, atvaizduojant HBK duomenis (žr. 2.1 skyriuje)

Labai svarbi problema, kuri įtakoja tiek Sammono algoritmo konvergavimo greitį, tiek gautos paklaidos dydį ir tikslumą, tai pradinių vektorių parinkimas $y_{ik}(m'=0),\ i=\overline{1,s},\ k=1,2$. Sammonas rekomenduoja parinkti pradines vektorių koordinates atsitiktinai (Sammon 1969), bet taip pat mini, kad praktikoje geriausia parinkti jas lygias didžiausias dispersijas turinčioms X_i koordinatėms, t. y. $y_{i1}(m'=0)=x_{ik_1}, y_{i2}(m'=0)=x_{ik_2}, \dots, y_{id}(m'=0)=x_{ik_d}$, kur dispersijos atitinkamai tenkina nelygybes $\sigma^2\left(x_{ik_1}\right)\geq\sigma^2\left(x_{ik_2}\right)\geq\cdots\geq$

$\sigma^2(x_{ik_n})$. Taip pat dažnai pasitaikantis pradinių vektorių iniciacijos būdų yra suprojektuoti daugiamačius duomenis naudojant klasikinį pagrindinių komponenčių analizės metodą (detalus aprašymas 1.2 skyriuje) į d dimensiją ir šios projekcijos taškų koordinates naudoti, kaip pradinius vektorius Sammono algoritmui.

Pradinių koordinačių parinkimo pagal didžiausias dispersijas pagrindinis privalumas tas, kad jo sudėtingumas $O(sn)$. Be to gaunama projekcija yra artima pagrindinių komponenčių analizės metodu gaunamai projekcijai, kadangi $\sum_{k=1}^{d}\sigma^2(x_{ij_k}) \leq \sum_{k=1}^{d}\lambda_k$, kur λ_k yra matricos Δ tikrinės reikšmės, surūšiuotos mažėjimo tvarka. Šie metodai tarpusavyje lyginami šios disertacijos autoriaus darbuose (A6, A7 ir A8).

1.5. Diagonalinis mažoravimo algoritmas

Diagonalinis mažoravimo algoritmas (DMA) detaliai aprašytas (Trosset and Groenen 2005). Tai atskiras SMACOF algoritmo atvejis, nes DMA algoritme naudojama paprastesnė mažoravimo funkcija, kadangi dauguma skaičiavimuose naudojamų svorių lygūs 0, ko pasekoje (1.12) formulė tampa lygi:

$$Y(m'+1) = Y(m') + \frac{1}{2}diag(V)^{-1}[B(Y(m') - V)]Y(m'). \quad (1.27)$$

DMA algoritmu gaunama šiek tiek didesnė projekcijos paklaida, lyginant su SMACOF (A2), nors (1.27) išreikštinės formulės skaičiavimas vyksta daug greičiau, taip pat nereikia skaičiuoti pseudo-inversinės V^+ matricos (1.28).

$$V^+ = (V + 11^T)^{-1} - n^{-2}11^T. \quad (1.28)$$

Dauguma matricos V (1.14) elementų w_{ij} yra lygūs nuliui, todėl galima neskaičiuoti atitinkamų matricos B elementų b_{ij}. Iteraciniu būdu perskaičiuojant dvimatei plokštumai priklausančių projekcijos vektorių $Y_i = \{y_{i1}, y_{i2}\}$ koordinates, atsižvelgiama ne į visus atstumus d_{ij} tarp vektorių Y_i ir Y_j, taip ženkliai pagreitina projekcijos suradimo procesą bei sutaupo kompiuterio atmintį.

Šio algoritmo sudėtingumas, naudojant standartinę V matricą, yra $O(n^2)$. Atsižvelgdami į tai autoriai (Trosset and Groenen 2005) rekomenduoja naudoti tik dalį matricos V svorių ir siūlo du matricos V formavimo būdus:

- Pirmas būdas. Svoriai apibrėžiami remiantis formule $w_{ij} = 0{,}4/(0{,}4 + d_{ij})$, kai $d_{ij} < 0{,}6$, priešingu atveju $w_{ij} = 0$. Šis būdas disertacijos tyrimuose nebuvo naudojamas, nes sunku įvertinti

atstumų d_{ij} skaičių, į kuriuos bus atsižvelgiama skaičiuojant projekcijos vektorių priklausančių dvimatei plokštumai koordinatės. Taip pat nėra apibrėžta kaip parinkti slenkstį atstumams konkrečiai analizuojamai duomenų aibei. Todėl disertacijos eksperimentinėje dalyje buvo naudojamas antras svorių matricos konstravimo būdas.

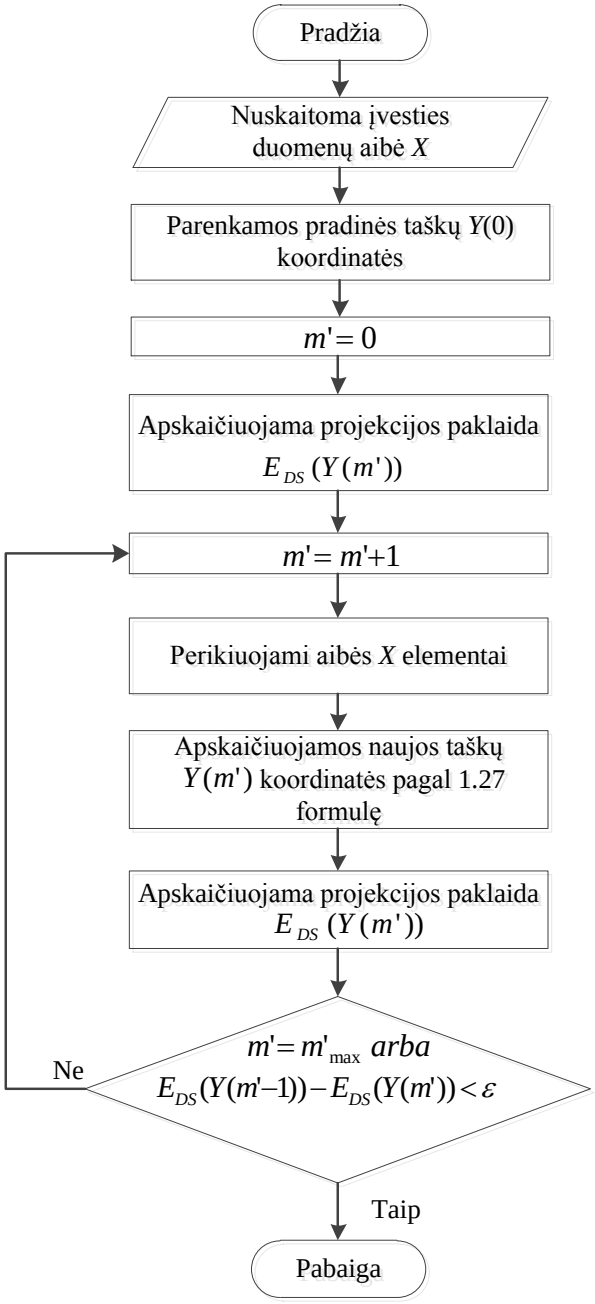
- Antras būdas. Matricos V svoriai imami $w_{ij} = 1$, kai $0 \le i - k < j < i + k \le m$, ir $w_{ij} = 0$ visais kitais atvejais. Čia k – daugiamačių vektorių kaimyniškumo parametras, nusakantis, į kiek kaimyninių vektorių analizuojamoje duomenų aibėje bus atsižvelgiama apskaičiuojant X_i. Naudojant šį būdą gaunama svorių matrica V yra pavidalo (1.29).

$$V = \begin{pmatrix} k & -1 & \overbrace{\dots}^{k-2} & -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & k+1 & -1 & \dots & -1 & 0 & 0 & \dots & 0 \\ -1 & -1 & k+2 & -1 & \dots & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -1 & \dots & -1 & 2k & -1 & \dots & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & -1 & \dots & -1 & k+1 & -1 \\ 0 & \dots & 0 & 0 & 0 & -1 & \dots & -1 & k \end{pmatrix}. \quad (1.29)$$

Matricos X eilutes galima keisti vietomis pagal pasirinktą eilučių išmaišymo būdą tiek algoritmo pradžioje, tiek po kiekvienos iteracijos. Tuo tarpu matrica V išlieka visuomet tokia pati. Tokiu būdu perskaičiuojant naujas projekcijos taškų koordinates atsižvelgiama į vis kitus įvesties taškus. Tai leidžia algoritmui konverguoti į minimumo tašką, turintį mažesnę paklaidos reikšmę. Galima atvirkštinė strategija, kurios metu matrica X nekeičiama, bet keičiama matricos V elementų padėtis. Ši strategija disertacijoje nebuvo taikoma, kadangi galima pasinaudoti matricoje X slypinčia papildoma informacija (klasteriai, taškai atsiskyrėliai ir t. t.).

DMA algoritmo sudėtingumas tiesiogiai priklauso nuo matricos V nelygių nuliui elementų skaičiaus. Žinodami kvadratinės svorių matricos V dimensiją s ir parametrą k, gauname, kad DMA algoritmo sudėtingumas lygus $O(s^2 - (s - 2k - 1)^2)$, kai $s \ge 2k + 1$. Kai naudojami visi svoriai, turime DMA algoritmo sudėtingumą lygų $O(s^2)$. Taigi DMA skaičiavimo laikas, priklausomai nuo pasirinkto k, sutrumpėja proporcingai santykiui $\frac{s^2}{s^2-(s-2k-1)^2}$, čia $s \ge 2k + 1$. Esant pakankamai dideliems s ir turint $k = s/10$ skaičiavimo laikas sumažėja iki 2,77 karto. Kuomet $k = s/100$, tuomet skaičiavimo laikas sumažėja iki 25,25 karto (Bernatavičienė *et al.* 2007).

DMA algoritmo schema pateikta 1.4 paveiksle.



1.4. pav. Diagonalinio mažoravimo algoritmo blokinė schema

Pagal (Trosset and Groenen 2005) algoritmą sudaro šie žingsniai:

1. Pasirinktu metodu parenkami pradiniai vektoriai $Y_i \epsilon R^d$ ir priskiriamas $m' = 0$.
2. Pagal (1.1) formulę apskaičiuojama paklaida $E_{DS}\big(Y(m')\big)$.
3. Iteracijų skaičius m' padidinamas vienetu.
4. Perrikiuojami aibės X elementai.
5. Apskaičiuojama nauja taškų projekcija $Y(m')$ pagal (1.27) formulę.
6. Apskaičiuojama daugiamačių skalių paklaida $E_{DS}\big(Y(m')\big)$.
7. Jeigu $E_{DS}\big(Y(m'-1)\big) - E_{DS}\big(Y(m')\big) < \varepsilon$ arba iteracijų skaičius $m' = m'_{max}$, tai algoritmas stabdomas, kitu atveju algoritmas vykdomas nuo 3 žingsnio.

1.6. Santykinių daugiamačių skalių algoritmas

Santykinių daugiamačių skalių algoritmas (*angl. relative MDS*) pasiūlytas ir aprašytas darbe (Naud and Duch 2000). Šis algoritmas skirtas didelių aibių bei naujų taškų priklausančių daugiamatei erdvei vizualizavimui, naudojant prieš tai apskaičiuotą bazinių taškų projekciją.

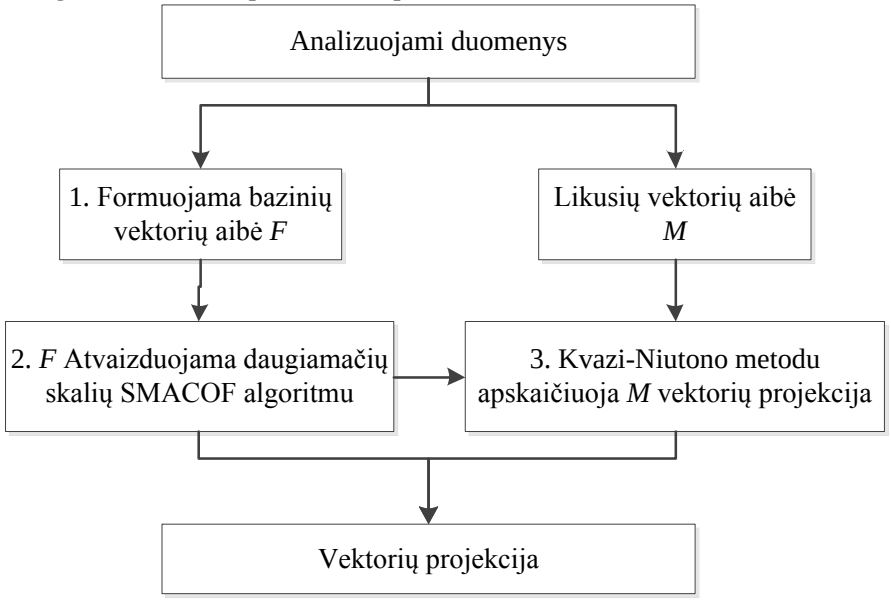
Naudojant klasikinį daugiamačių skalių metodą, negalima atidėti naujo taško neperskaičiuojant visos turimos duomenų aibės projekcijos. Todėl naujų taškų atvaizdavimui gali būti naudojamas santykinių daugiamačių skalių algoritmas (SDS). Nors šis metodas nėra toks tikslus kaip SMACOF ar Sammono metodai, tačiau jis gali atvaizduoti dideles aibes, tam pareikalaudamas mažai kompiuterio skaičiavimo resursų. Kadangi SDS algoritme nereikia saugoti ir perskaičiuoti kiekvienos iteracijos metu didelių atstumų matricų D.

Santykinių daugiamačių skalių metode turimą duomenų aibę X padaliname į du netuščius poaibius F ir M, taip kad $X = F \cup M$ ir $F \cap M = \emptyset$. Aibę (aibės X poaibį) F vadinsime bazinių taškų aibe, o aibę M – naujų taškų aibe. Nors aibė M gali būti nuolat papildoma naujais elementais, tačiau kiekvienu fiksuotu laiko momentu laikysime, kad turime baigtinę aibę, kurios tūrį (elementų skaičių) žymėsime s_M. Atitinkamai bazinių taškų aibės elementų skaičius žymimas s_F, be to galioja lygybė $s = s_F + s_M$.

Tuomet ieškodami aibės X projekcijos santykinių daugiamačių skalių metodu atliekame šiuos žingsnius:

1. Pagal pasirinktą metodiką reikia išrinkti bazinius vektorius ir sukonstruoti bazinių vektorių aibę; pvz., naudojantis *k*-vidurkių (*angl. k-means*) algoritmu (Naud and Duch 2000; Vesento 2001). Daug įvairių metodikų pasiūlyta disertacijoje (Bernatavičienė 2008).
2. Naudojant DS metodą projektuojama bazinių vektorių aibė *F*;
3. Naujų aibės *M* taškų projekcijos apskaičiuojamos atsižvelgiant į bazinių taškų projekcijas ir į naujų taškų tarpusavio atstumus. Kvazi-Niutono (*angl. quasi-Newton*) metodas naudojamas minimizuojama santykinių daugiamačių skalių tikslo funkcija (1.30).

Algoritmo schema pateikta 1.5 paveiksle.



1.5 pav. Santykinių daugiamačių skalių algoritmo veikimo principas

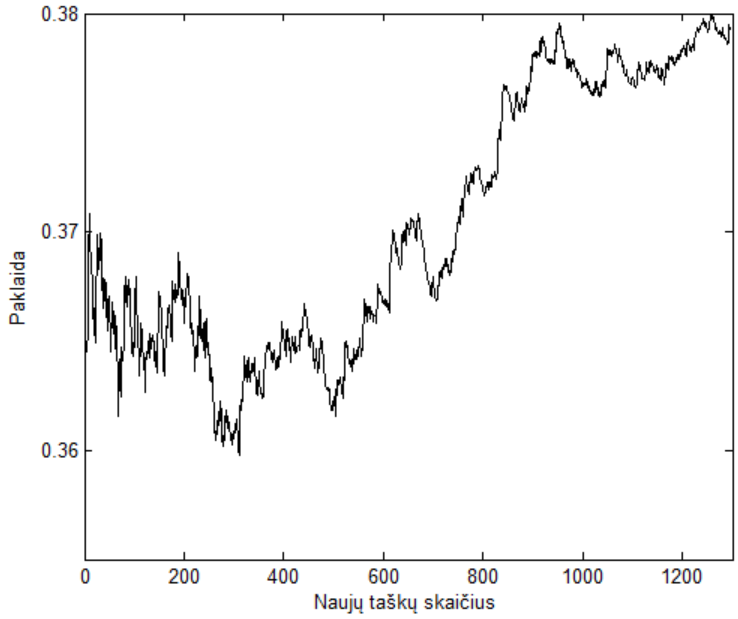
Svarbus santykinių daugiamačių skalių bruožas – aibę *M* taip pat galima dalinti į smulkesnius ar didesnius poaibius ir jų elementus (taškus) atidėti minimizuojant tą pačią tikslo funkciją (1.30).

$$E_{SDS} = \sum_{i<j}^{s_M} \left(\delta_{ij} - d_{ij}\right)^2 + \sum_{i=1}^{s_F} \sum_{j=s_F+1}^{s_M+s_F} \left(\delta_{ij} - d_{ij}\right)^2. \qquad (1.30)$$

J. Bernatavičienės disertacijoje (Bernatavičienė 2008) pateikta detali algoritmo schema, kurioje siūloma naujus taškus atidėti po vieną. Taip atidedant

taškus, trečiajame algoritmo žingsnyje naudojamas kvazi-Niutono metodas veikia efektyviausiai. Atidedant naujus taškus didesnėmis grupėmis tikėtina geresnė projekcija, nes yra įvertinamas didesnis kiekis taškų tarpusavio atstumų, tačiau dėl ilgiau skaičiuojamos naujų taškų projekcijos, prarandamas vienas pagrindinių SDS algoritmų privalumų – skaičiavimo greitis. Todėl, kuo didesnėmis grupėmis atidedami nauji taškai, tuo paklaidos ir laiko santykis yra artimesnis gaunamam SMACOF algoritmu ar tiesiog pagrindinių komponenčių analizės metodu gaunamam paklaidos ir laiko santykiui (A1).

Disertacijoje siūloma ne tik įvairios M keitimo strategijos, bet ir aibės F formavimo strategijos. Pavyzdžiui, jei aibė M nuolat papildoma naujais elementais, tuomet SDS algoritmas po tam tikro atidėtų taškų skaičiaus pradeda grąžinti netikslią projekciją normalizuotos paklaidos (1.4) prasme (1.6 pav.). Taigi reikia performuoti aibę F, pridedant daugiau taškų ir atlikti visus skaičiavimus iš naujo, arba galima po keleto žingsnių jau atvaizduotus naujus taškus priskirti baziniams (pridėti prie aibės F) taškams ir toliau tęsti naujų taškų atidėjimą.



1.6. pav. Normalizuotos paklaidos priklausomybė nuo naujų taškų skaičiaus, kai santykinių daugiamačių skalių algoritmu atvaizduojami 1446 „Sferos“ taškai iš kurių 150 yra baziniai

Dvi santykinių skalių projekcijos tarpusavyje nebuvo lyginamos remiantis (1.30) formule tiek dėl jos priklausomybės nuo mastelio, tiek ir dėl galimo skirtingo bazinių vektorių skaičiaus. Ji keičiama (1.4) arba (1.5) paklaidos formulėmis, pagal kurias galima palyginti gautas projekcijas ir su kitomis DS gaunamomis projekcijomis.

1.7. Santykinės perspektyvos metodas

Santykinės perspektyvos metodas (*angl. relational perspective map*), yra nagrinėjamas kaip vienas iš originalių daugiamačių skalių atmainų, kuris buvo pasiūlytas J. X. Li (Li 2004). Metodo tikslas, kaip ir įprastų daugiamačių skalių, yra artimumus išsaugantis atvaizdavimas į R^2 erdvę. Santykinės perspektyvos metodas (RPM) algoritmo pagrindinė idėja yra imituoti dalelių sistemą ant uždaro paviršiaus, kai tarp dalelių veikia tarpusavio stūmimo ir traukos jėgos. Šios jėgos, veikdamos ant uždaro paviršiaus, turi tenkinti pusiausvyros sąlygą. Autorius J. X. Li teigia, kad šio metodo viena iš gerų savybių yra gebėjimas sudėtingos struktūros duomenų aibę suskaidyti į mažesnes ant uždaro paviršiaus tarpusavyje nepersidengiančias aibes. Taip pat kaip daugiamačių skalių Sammono algoritmas, RPM gerai išsaugo artimų kaimynų tarpusavio nepanašumus ant projekcijos paviršiaus. Naujausiame J. X. Li sukurtos programinės įrangos pakete „Visumap“ (Li 2010), pateikiama RPM modifikacija, leidžianti taškus atvaizduoti ne tik ant toro, bet ir ant sferos, ištiesintos sferos (*angl. flat sphere)*, ištiesinto Kleino-butelio (*angl. flat klein-bottle*), ar tiesiog realių skaičių plokštumoje.

Vienas iš RPM algoritmų trūkumų – jis nekonverguoja į minimumą. Konvergavimui pasiekti naudojamas euristinis žingsnio mažinimas (Li 2004). Disertacijoje buvo suformuoti uždaviniai ištirti ir pagerinti RPM algoritmo konvergavimą (A5).

RPM algoritmas siekia atvaizduoti aibę X ant toro (žiedo formos toro) paviršiaus, taip kad atstumai δ_{ij} tarp taškų $X_i, X_j \in X$, būtų lygūs atstumams d_{ij} tarp taškų $Y_i, Y_j \in T$ ant toro paviršiaus. Šis metodas skiriasi nuo standartinių DS metodų tuo, kad minimizuoja įtempimų (potencinės energijos) funkciją (1.31) ir naudoja ypatingą atstumų metriką (1.32) panašią į Minkovskio.

Toras geometrijoje yra sukinio paviršius, kurį apibrėžia apskritimas, besisukantis apie ašį, lygiagrečią jo plokštumai bei jo neliečiančią (Weisstein 2010). Paprastai toru laikomas žiedo formos toras, nors egzistuoja daug kitų formų torų (elipsinis, plokščias ir kt.) (Gray *et al.* 2006).

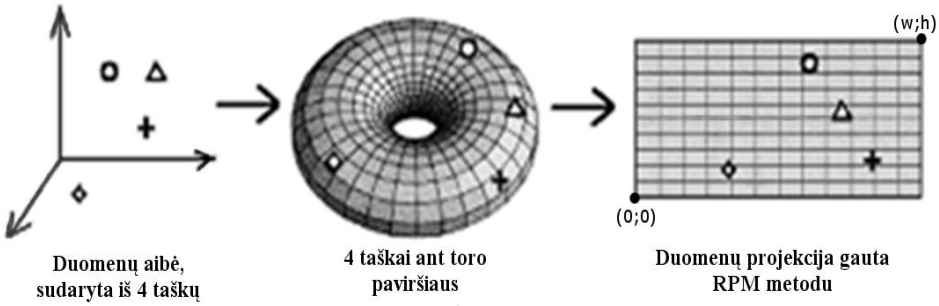
Potencinė energija apskaičiuojama pagal formulę:

$$E_p = \sum_{i<j} \frac{\delta_{ij}}{p d_{ij}^p}, kai\ p = -1\ arba\ p > 0,$$
$$E_0 = -\sum_{i<j} \delta_{ij}\, ln\big(d_{ij}\big), kai\ p = 0. \tag{1.31}$$

Atstumai ant toro $T = [0, w] \times [0, h] \subset R^2$ gali būti apskaičiuojami pagal formulę:

$$d_r\big(Y_i, Y_j\big) = \Big(min\big\{\big|y_{i1} - y_{j1}\big|, w - \big|y_{i1} - y_{j1}\big|\big\}^r + min\big\{\big|y_{i2} - y_{j2}\big|, h - \big|y_{i2} - y_{j2}\big|\big\}^r\Big)^{\frac{1}{r}}, \tag{1.32}$$

čia w ir h yra toro išklotinės (1.7 pav.) plotis ir aukštis. $r > 0$.



1.7 pav. Santykinės perspektyvos metodo modelis

Eksperimentiškai nustatyta, kad geriausia projekcija ant toro gaunama kai $p = r = 0$ (Li 2004). Todėl tyrimuose minimizuojama E_0 su $d_{ij} = d_0\big(Y_i, Y_j\big)$.

E_0 minimizuojama taikant iteracinį Niutono-Rapsono metodą (Press *et al.* 2002; Žilinskas 2000), apskaičiuojant projekcijos taško $Y_i(y_{i1}, y_{i2})$, kiekvieną koordinatę atskirai. Šiuo atveju paklaidos funkcija E_0, nagrinėjama kaip vieno kintamojo funkcija $f(y_{ik})$, atžvilgiu y_{i1} arba y_{i2}. Nauja kintamojo reikšmė apskaičiuojama pagal (1.33) formulę.

$$y_{ik}(m' + 1) = y_{ik}(m') - \frac{\partial E_p / \partial y_{ik}}{\partial^2 E_p / \partial y_{ik}^2}, k = 1,2. \tag{1.33}$$

Įstatę į (1.33) E_p pirmąją ir antrąją dalines išvestines gauname:

$$y_{ik}(m'+1) = y_{ik}(m') + \frac{1}{p+1}\frac{\sum_{i\neq j} h_{ijk} f_{ij}}{\sum_{i\neq j} f_{ij}/d_{ij}}, k = 1,2, \qquad (1.34)$$

kur f_{ij} apskaičiuojama pagal (1.35) formulę, o h_{ijk} pagal (1.36) formulę.

$$f_{ij} = \frac{\partial E_p}{\partial d_{ij}} = -\frac{\delta_{ij}}{d_{ij}}, i < j. \qquad (1.35)$$

Atstumo $d_{ij} = d_0(Y_i, Y_j)$ dalinė išvestinė pagal y_{ik}, kai $k = 1$ lygi:

$$h_{ij1} = \frac{\partial d_{ij}}{\partial y_{i1}} = \begin{cases} 1\,, kai\ |y_{i1} - y_{j1}| < \frac{w}{2}\ ir\ y_{i1} > y_{j1}; \\ -1, kai\ |y_{i1} - y_{j1}| < \frac{w}{2}\ ir\ y_{i1} < y_{j1}; \\ -1, kai\ |y_{i1} - y_{j1}| > \frac{w}{2}\ ir\ y_{i1} > y_{j1}; \\ 1\,, kai\ |y_{i1} - y_{j1}| > \frac{w}{2}\ ir\ y_{i1} < y_{j1}. \end{cases} \qquad (1.36)$$

Analogiškai galima apskaičiuoti h_{ij2}, formulėje (1.36) w pakeitus h, o y_{i1} atitinkamai y_{i2}.

Nesunku pastebėti, kad h_{ij1} neturi dalinės išvestinės, kai $|y_{i1} - y_{j1}| = \frac{w}{2}$ arba $y_{i1} = y_{j1}$. Funkcija E_p yra nediferencijuojama šiuose taškuose. Vadinasi Niutono-Rapsono metodas negarantuoja funkcijos minimumo suradimo. Maža to, taškų $|y_{i1} - y_{j1}| = \frac{w}{2}$ ir $y_{i1} = y_{j1}$ aplinkoje E_p įgyja skirtingo ženklo dalines išvestines (h_{ij1}) iš kairės ir iš dešinės. Toks staigus išvestinės ženklo pasikeitimas sąlygoja, kad iteracinio proceso metu taškai, esantys arti toro išklotinės kraštinių, ima virpėti ir neleidžia pasiekti funkcijos E_p minimumo (Karbauskaitė *et al.* 2006).

Kita problema, kad parinkus labai besiskiriančius savo dydžiu w ir h, RPM algoritmas apskirtai nelinkęs konverguoti. Tai galima paaiškinti tuo, kad nors taško Y_i koordinatės y_{i1} ir y_{i2} apskaičiuojamos individualiai, tačiau jos įtakoja viena kitos reikšmes. Apskaičiuojant atstumą d_{ij} įvertinamos abi koordinatės. Jeigu vienos iš jų įtaka žymiai didesnė (tai atsitinka, kai $w \gg h$ arba $w \ll h$), tuomet neišvengiamai atsiranda neproporcinga taško koordinačių y_{i1} ir y_{i2} įtaka viena kitos reikšmei. Šią problemą dalinai pavyko išspręsti darbe (A5), kuriame pasiūlyta naudoti atstumą d_n, kuris gaunamas normuojant atstumą d_0 ir jį apskaičiuojant pagal (1.37) formulę.

$$d_n(Y_i, Y_j) = min\left\{\frac{|y_{i1} - y_{j1}|}{w}, 1 - \frac{|y_{i1} - y_{j1}|}{w}\right\} + min\left\{\frac{|y_{i2} - y_{j2}|}{h}, 1 - \frac{|y_{i2} - y_{j2}|}{h}\right\}. \quad (1.37)$$

Problema išspręsta tik dalinai, nes, taikant normuotą atstumą d_n, RPM algoritmas nebepriklauso nuo toro parametrų w ir h, ir be papildomų stabdymo parametrų gaunama projekcija, tačiau vis tik ši projekcija nėra stabili. Taškai esantys šalia toro kraštų šokinėja iš vienos toro pusės į kitą, taip išjudindami visus likusius projekcijos taškus. Naudojant tolydžią atstumų funkciją:

$$d_t(Y_i, Y_j) = \frac{w}{4}\left(1 - cos\left(\frac{2\pi}{w}(y_{i1} - y_{j1})\right)\right) + \frac{h}{4}\left(1 - cos\left(\frac{2\pi}{h}(y_{i2} - y_{j2})\right)\right). \quad (1.38)$$

Tuomet, E_p, tampa diferencijuojama visuose taškuose ant toro paviršiaus. Tačiau E_p funkcijos konvergavimo tai nepakeičia. Galima priežastis yra ta, kad $d_t(Y_i, Y_j)$, netenkina trikampio nelygybės savybės.

Atstumų funkcija d_t, buvo išvesta remiantis atstumo d_0 dalinėmis išvestinėmis h_{ijk}. Tarkime $z_{i1} = |y_{i1} - y_{i2}|$, tuomet funkciją $h_{ij1}(z)$ galima pakeisti sinusoide, bei remiantis funkcijos pilnu diferencialu galima išvesti naują atstumų funkciją d_t. Svarbu pastebėti, kad h_{ij1} taškuose $|y_{i1} - y_{j1}| = \mathrm{w}/2$ ir $y_{i1} = y_{j1}$ yra lygi nuliui, o šioje aplinkoje yra arti nulio. Taigi taškų, esančių arti toro išklotinės briaunų, padėtis kinta nežymiai ir tolygiai.

Norint stabilaus E_p konvergavimo į minimumo tašką, reikia laikyti, kad E_p yra ne vieno kintamojo, o dviejų kintamųjų funkcija $f(y_{i1}, y_{i2})$.

1.8. Savioganizuojantis neuroninis tinklas

Saviorganizuojantis neuroninis tinklas (*SOM, angl. self-organizing map*) arba Kohoneno saviorganizuojantis savybių žemėlapis taip vadinamas jį 1982 m. sukūrusio autoriaus vardu (Kohonen T. 2001; Kaski *et al.* 1998), skirtas išsaugoti dažnus topologinius ryšius tarp duomenų taškų. SOM tinklas sėkmingai naudojamas duomenų klasterizavime ir jų vizualizavime (Flexer 2001).

1.6. Apibrėžimas. Saviorganizuojantis neuroninis tinklas – daugiamačių daugdarų netiesinis, sutvarkytas, tolygus atvaizdavimas į reguliarios struktūros ir mažos dimensijos elementų masyvą (Kohonen T. 2001).

Intuityviai daugdara suvokiama kaip glodžioji aibė su lokaliai joje apibrėžta Euklidinė koordinačių sistema (Adler and Taylor 2007). Taip apibūdinama daugdara dar vadinama topologine daugdara arba topologine erdve. Euklidinė koordinačių sistema apibrėžia Euklidinę erdvę R^n, kurios dimensija laikoma ir daugdaros dimensija. Topologinę erdvę galima įsivaizduoti kaip aibę X su joje apibrėžta topologija $\mathcal{U}$ (aibės sudalinimu į atvirus poaibius) (Munkres 1975; Livré 2004).

Pagrindinė SOM tinklo paskirtis – įvesties duomenų aibės suspaudimas išlaikant svarbiausias topologines, metrines savybes (Kohonen T. 2001). Taškai, artimi įėjimo (*angl. input*) vektorių erdvėje, yra atvaizduojami arti vieni kitų ir SOM tinkle. SOM tinklai gali būti naudojami siekiant vizualiai pateikti duomenų klasterius bei ieškant daugiamačių duomenų projekcijų į mažesnės dimensijos erdvę, įprastai į plokštumą.

Šis atvaizdavimas realizuotas panašiai kaip klasikinis vektorių kvantavimo metodas, tik jis turi griežtai apibrėžtą struktūrą bei skirtingą paklaidos minimizavimo algoritmą (Kohonen T. 2001; Kohonen T. 2002b). Viena iš modernių krypčių yra neuroninių tinklų ir klasikinių daugiamačių skalių algoritmų jungimas. Pavyzdžiui, jungiant dirbtinį neuroninį tinklą su Sammono algoritmų gaunamas SAMANN algoritmas (Mao and K. 1995), kuris efektyviai taikomas didelių duomenų aibių atvaizdavimui. Dirbtiniai neuroniniai tinklai praktikoje taikomi ne tik duomenų vizualizavimui (Bernatavičienė *et al.* 2006), bet ir tiesiogiai klasifikavimui (Treigys *et al.* 2008).

1.7. Apibrėžimas. Kvantavimas yra tolydaus dydžio interpretavimas, naudojantis diskrečių reikšmių baigtine aibe (Evans 2010).

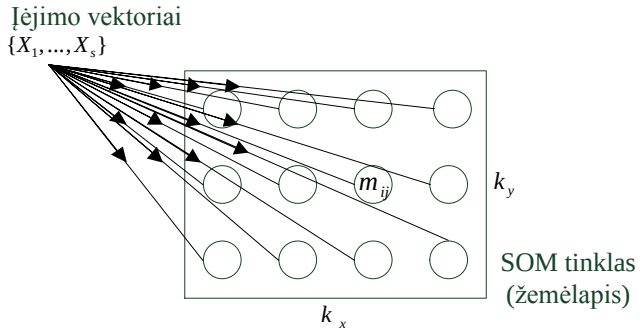
1.8. Apibrėžimas. Vektorių kvantavimu (*angl. vector quantization*) vadiname įėjimo (*angl. input*) vektorių $X_i \in R^n$ aproksimavimą baigtiniu skaičiumi vektorių taip vadinamais kodų-knygos (neuronais) vektoriais (*angl. codebook vector*) (Kohonen T. 2001).

Disertacijoje kodų-knygos vektoriai vadinami neuronais arba kvantavimo vektoriais (neuronais, nes SOM žemėlapis yra neuroninis tinklas; kvantavimo vektoriais, nes SOM gali būti tapatinamas su vektorių kvantavimu).

Saviorganizuojantis neuroninis tinklas (SOM) yra neuronų $m_{ij} = \{m_{ij}^1, m_{ij}^2, \dots, m_{ij}^n\}$ $i = 1, \dots, k_x,\ j = 1, \dots, k_y$, išdėstytų dvimačio tinklelio (lentelės) mazguose, masyvas. Galima stačiakampė (*angl. rectangular*) arba šešiakampė (*angl. hexagonal*) tinklo struktūra. Keturkampės tinklo struktūros atveju, k_x yra lentelės eilučių skaičius, k_y – stulpelių skaičius. Kiekvienas įėjimo aibės vektorius $X_i \in (X_1, X_2, \dots, X_s) \subset R^n$ mokymo metu yra susiejamas su vienu tinklo neuronu m_{ij}, kuris taip pat yra vektorius,

priklausantis R^n. Kadangi SOM tinklas apmokomas vektoriais $X_1, X_2, \ldots, X_s$, todėl jie vadinami dar ir apmokymo vektoriais. O neuronas m_{ij} vadinamas neuronu-nugalėtoju, jei jis po mokymo susietas bent su vienu įėjimo vektoriumi ir žymimas m_{ij}^c.

Dvimačio stačiakampio neuroninio tinklo schema pateikta 1.8 paveiksle.



1.8 pav. Stačiakampio saviorganizuojančio neuroninio tinklo schema

Tegu turime įėjimo vektorių aibę iš R^n, kuri bus naudojama SOM tinklo apmokymui. Kiekvienas tinklo neuronas sujungtas su kiekvienu įėjimo vektoriumi (1.8 pav.). Šioje disertacijoje naudojama tik stačiakampė tinklo struktūra.

SOM tinklas apmokomas mokymo be mokytojo būdu. Apie šį mokymo procesą plačiai aprašyta straipsnyje (Dzemyda *et al.* 2008). Apmokius SOM tinklą, būtina nustatyti jo apsimokymo kokybę. Dažniausiai vertinamos dvi paklaidos: kvantavimo (*angl. quantization error*) (1.39) ir topografinė (*angl. topographic error*) (1.40).

Kvantavimo paklaida parodo, kaip tiksliai tinklo neuronai m_{ij} prisiderina prie mokymo aibės vektorių:

$$E_{SOM(kvant)} = \frac{1}{s}\sum_{i=1}^{s} \|X_i - m_i^c\|, \qquad (1.39)$$

čia m_i^c yra vektoriaus X_i neuronas-nugalėtojas.

Topografinė paklaida parodo, kaip gerai SOM tinklas išlaiko analizuojamų duomenų topografiją, t. y. tarpusavio išsidėstymą. Topografinė paklaida $E_{SOM(topo)}$ skaičiuojama pagal tokią formulę:

$$E_{SOM(topo)} = \frac{1}{s}\sum_{i=1}^{s} u(X_i). \quad (1.40)$$

Jeigu SOM žemėlapyje vektoriaus X_i neuronas-nugalėtojas yra šalia neurono, iki kurio atstumas nuo X_i yra mažiausias (neskaičiuojant iki neurono-nugalėtojo), tai $u(X_i) = 0$, priešingu atveju $u(X_i) = 1$.

Stačiakampės struktūros tinklo atveju, k_x yra lentelės eilučių skaičius, k_y – stulpelių skaičius. Kiekvienas apmokymo aibės vektorius $X_i \in (X_1, X_2, \ldots, X_s)$ mokymo metu yra susiejamas su vienu tinklo neuronu, kuris taip pat yra vektorius priklausantis R^n. Mokymo pradžioje vektorių m_{ij} komponentės generuojamos atsitiktinai. Kiekviename mokymo žingsnyje vienas iš apmokymo aibės vektorių $X_i \in (X_1, X_2, \ldots, X_s)$ pateikiamas į tinklą. Randama, iki kurio neurono m_i^c vektoriaus X_i Euklidinis atstumas yra mažiausias. Neuronų komponentės keičiamos pagal (1.41) bendrą iteracinę formulę:

$$m_{ij} \leftarrow m_{ij} + h_{ij}^c\left(X_i - m_{ij}\right), \quad (1.41)$$

čia disertacijoje naudojamas atskiras h_{ij}^c atvejis: $h_{ij}^c = \frac{\alpha}{\alpha\eta_{ij}^c+1}$, $\alpha = \max\left(\frac{e+1-\hat{e}}{e}, 0{,}01\right)$, e – mokymo epochų skaičius, $\hat{e}$ – vykdomos epochos numeris (Dzemyda 2001). Viena mokymo epocha – tai mokymo proceso dalis, kai visus vektorius pateikiame tinklui po vieną kartą. Ją sudaro s mokymo žingsnių. Dydis η_{ij}^c yra neurono m_{ij}^c (indeksas c yra neurono m_{ij} unikalus numeris) kaimynystės tarp neuronų m_{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. (Agrawal *et al.* 1998). Kiekvienos epochos metu perskaičiuojami tie neuronai m_{ij}, kuriems galioja:

$$\eta_{ij}^c \le max\left[\alpha\, max\left(k_x, k_y\right), 1\right]. \quad (1.42)$$

Pažymėkime funkciją $\eta(\hat{e}) = \max\left[\alpha \max\left(k_x, k_y\right), 1\right] = \max\left[\alpha k', 1\right]$, kur $k' = \max\left(k_x, k_y\right)$. Sveikas skaičius n' rodo, kiek sumažėjo kaimynystės eilė lyginant su didžiausia (pagal (1.42) formulę didžiausia kaimynystė eilė $\eta_{ij}^c = k'$, kai $\hat{e} = 1$). Paprastai k_x ir k_y, o tuo pačiu ir k', neviršija kelių dešimčių.

Tyrinėjant mokymo procesą, ypač siekiant sukurti SOM lygiagretųjį algoritmą, svarbu žinoti, kaip keičiasi tinklo mokymo operacijų skaičius. Todėl buvo suformuluota ir įrodyta tokia teorema (A7):

1.1 Teorema. Jei turime stačiakampį SOM tinklą, kurio kraštinė $k' = \max(k_x, k_y) \le 100$, ir mokymo epocha tenkina nelygybę $1 \le \hat{e} \le e + 1 - e/k'$, tai epochose $\hat{e} = \left[\frac{(n'-1)e}{k'}\right] + 2$, $(n' = 1, \dots, k' - 1)$ bet kurio neurono m_{ij}^c maksimali kaimynystės eilė η_{ij}^c (1.42) yra mažesnė vienetu, lyginant su $(\hat{e} - 1)$ epocha, jei $1 \le \hat{e} \le e + 1 - e/k'$. Maksimali kaimynystės eilė nebemažėja ir lieka lygi vienam ($\eta_{ij}^c = 1$), kai $e + 1 - e/k' \le \hat{e} \le e$.

Įrodymas. Nesunku įsitikinti, kad augant vykdomos epochos numeriui $\hat{e}$, dydis $\alpha = \max\left(\frac{e+1-\hat{e}}{e}, 0{,}01\right)$ mažėja. Jis pasiekia ribinį tašką, kai $\frac{e+1-\hat{e}}{e} = 0{,}01 \Longrightarrow \hat{e} = 0{,}99e + 1$.

Tada $\alpha = \max\left(\frac{e+1-\hat{e}}{e}, 0{,}01\right) = \begin{cases} \frac{e+1-\hat{e}}{e}, & \text{kai } 1 \le \hat{e} \le 0{,}99e + 1, \\ 0{,}01, & \text{kai } 0{,}99e + 1 < \hat{e} \le e. \end{cases}$

Augant vykdomos epochos numeriui $\hat{e}$, funkcija $\eta(\hat{e}) = \max[\alpha k', 1]$ mažėja.

Ji pasiekia minimumą, kai $\alpha k' = 1 \Longrightarrow \frac{e+1-\hat{e}}{e} k' = 1 \Longrightarrow \hat{e} = e + 1 - \frac{e}{k'}$.

Tada, kai tenkinama nelygybė $1 \le \hat{e} \le 0{,}99e + 1$, funkcija $\eta(\hat{e})$ įgyja pavidalą:

$$\eta(\hat{e}) = \begin{cases} \dfrac{e+1-\hat{e}}{e} k', & \text{kai } 1 \le \hat{e} \le e + 1 - \dfrac{e}{k'}, \\ 1, & \text{kai } e + 1 - \dfrac{e}{k'} < \hat{e} \le e. \end{cases}$$

Kai $0{,}99e + 1 < \hat{e} \le e$, funkcija $\eta(\hat{e})$ ekvivalenti šiai:

$$\eta(\hat{e}) = \begin{cases} 0{,}01k, & \text{kai } k' \ge 100, \\ 1, & \text{kai } 0 < k' < 100. \end{cases}$$

Tarp sąlygų $1 \le \hat{e} \le 0{,}99e + 1$ ir $1 \le \hat{e} \le e + 1 - e/k'$, kai $k \le 100$, griežtesnė yra antroji, tada:

$$\eta(\hat{e}) = \begin{cases} \dfrac{e+1-\hat{e}}{e} k', & \text{kai } 1 \le \hat{e} \le e + 1 - \dfrac{e}{k'}, \\ 1, & \text{kai } e + 1 - \dfrac{e}{k'} < \hat{e} \le e. \end{cases}$$

Reikia rasti slenksčio tašką $\bar{e} \in R$, kuriame $\eta(\bar{e}) = k' - (n' - 1)$. Tai bus slenksčio taškas, iki kurio perskaičiuojamų neuronų kaimynystės eilė $\eta_{ij}^c = k' - (n' - 1)$. Pirmoje epochoje $\hat{e}$ didesniu numeriu už $\bar{e}$ $(\hat{e} = \bar{e} + 1)$ kaimynystės eilė sumažės vienetu, lyginant su $(\hat{e} - 1)$ iteracija. Iš čia $\frac{e+1-\bar{e}}{e} k' = k' - (n' - 1) \Longrightarrow \bar{e} = \frac{k' + (n'-1)e}{k'} = \frac{(n'-1)e}{k'} + 1$.□

Iš įrodytos teoremos seka, kad perskaičiuojamų neuronų skaičiaus priklausomybė nuo epochos numerio turi laiptinę formą. Po kiekvieno $e' = \left\lceil \frac{n'e}{k'} \right\rceil - \left\lceil \frac{(n'-1)e}{k'} \right\rceil$ $(n' = 1, \ldots, k' - 2)$ skaičiaus epochų perskaičiuojamų neuronų skaičius mažėja.

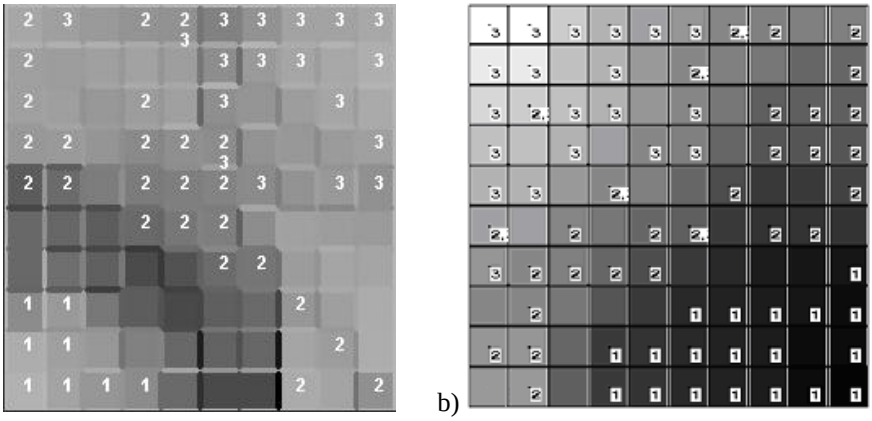
Teoremos rezultatai naudingi tuomet, kai kuriamas lygiagretusis integruotas SOM ir Sammono algoritmų junginys, nes leidžia tikslai apskaičiuoti, kada apsikeitimas duomenimis tarp procesorių yra pats efektyviausias (Dzemyda *et al.* 2003; Dzemyda and Kurasova 2006).

Pavaizduoti SOM tinklą galima kaip unifikuotą atstumų matricą arba U-matricą (*angl. U-matrix, unified distance matrix*). Tai yra vienas iš populiaresnių SOM tinklo vizualizavimo būdų, pristatytas, darbuose (Kraaijveld 1995), (Ultsch 1990). Pagal šį būdą vidutiniai atstumai tarp kaimyninių neuronų yra pateikiami pilkos spalvos atspalviais (vėliau imta naudoti ir kitų spalvų skales) (Kohonen T. 2001; Dzemyda and Kurasova 2002). Jei vidutiniai atstumai tarp kaimyninių neuronų yra maži, tuos neuronus atitinkantys tinklo langeliai spalvinami šviesia spalva (tamsi spalva reiškia didelius atstumus). Naudojant Nenet sistemą (Hassinen *et al.* 1999). 1.9 a paveiksle pateikta U-matricos iliustracija, kurioje matyti, kaip pirmos klasės irisai atsiskiria nuo kitų dviejų, griežtos ribos tarp antros ir trečios klasių nėra. Klasteriai yra nustatomi pagal šviesius atspalvius, o jų ribos – pagal tamsesnius (Kohonen T. 2001; Kohonen T. 2002a). Algoritmas, kaip apskaičiuoti U-matricą, pateikta (Dzemyda *et al.* 2008).

SOM tinklo vizualizavimui Kleiveg (Kleiweg 1996) pasiūlė pilkos spalvos atspalviais nuspalvinti tik briaunas tarp kaimynų, remiantis atstumais tarp jų. Papildomai gautą paveikslą papildant linijomis sujungiančiomis artimiausius kaimynus, pagal minimalaus jungimo medžio (*angl. minimal spanning tree*) algoritmą (Kleiweg 1996). Tai leidžia detaliau atskleisti neuronų tarpusavio ryšius.

Saviorganizuojančio tinklo neuronai yra vektoriai, kurių atskiros koordinatės (komponentės) gali turėti savyje informaciją apie atskirų komponenčių koreliacijas ar tarpusavio ryšius (Seiffert and Jain 2002). Todėl SOM tinklas dažnai atvaizduojamas atskiromis komponenčių plokštumomis (*angl. component planes*). Tinklo ląstelės pateikiamos pilkos spalvos atspalviais priklausomai nuo komponentės reikšmės.

a) b)



1.9 pav. Saviorganizuojančio neuroninio tinklo vizualizavimas:
a) U-matrica; b) pagrįstas neurono vektoriaus ilgiu

Naudojantis šiomis idėjomis, disertacijoje siūlomas SOM tinklo atvaizdavimo būdas, kuriame SOM tinklo lentelės ląstelės yra nuspalvinamos pilkos spalvos atspalviais priklausomai nuo to, koks yra ląstelėje esančio neurono ilgis (1.9 b pav.). Šiame vaizdavime neurono padėtis SOM tinkle nurodo jo kryptį, o spalva panašumą į kitus tinklo neuronus. Turima informacija leidžia identifikuoti klasterius, tačiau kartu neuroninio tinklo vaizdą padaro labiau suprantamą lyginant jį su U-matrica.

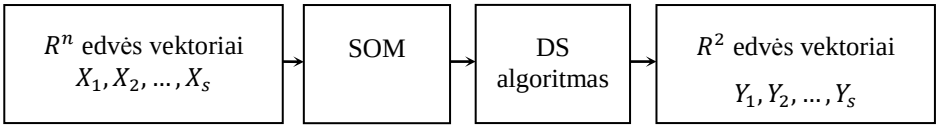
Toks atvaizdavimo būdas netinka tuomet, kai tinklas apmokomas vienodo ilgio vektoriais.

1.8.1. SOM ir daugiamačių skalių algoritmų junginiai

Vizualizuojant didelių apimčių duomenų aibes tikslinga naudoti saviorganizuojančio neuroninio tinklo ir daugiamačių skalių (klasikinės daugiamatės skalės, Sammono algoritmas, SMACOF algoritmas, santykinės daugiamatės skalės ir pan.) algoritmų junginį (SOM-DS junginys). SOM algoritmu duomenys klasterizuojami, o vektorių-nugalėtojų DS algoritmu projektuojami į plokštumą. Toks jungtinis algoritmas pasiūlytas (Bernatavičienė *et al.* 2005) darbe. Tam turėjo įtaką darbas (Dzemyda and Kurasova 2002).

SOM tinklo apmokymo pabaigoje kiekvienam iš įėjimo vektorių X_i yra priskiriamas vienintelis neuroninio tinklo vektorius m_i^c, kuris vadinamas neuronu-nugalėtoju. Neuronų-nugalėtojų skaičius paprastai būna mažesnis nei s ir priklauso nuo pasirinkto SOM tinklo dydžio, nes kelis aibės X vektorius paprastai atitinka vienas SOM tinklo neuronas. Tai gali žymiai sumažinti įvesties

vektorių skaičių DS algoritmui, nuo kurio tiesiogiai priklauso operacijų skaičius, reikalingas projekcijai apskaičiuoti.



1.10 pav. Nuoseklaus saviorganizuojančio neuroninio tinklo ir daugiamačių skalių algoritmo junginio schema

Taigi, algoritmų junginio pagrindinė idėja (1.10 pav.) yra atvaizduoti šiuos neuronus-nugalėtojus į plokštumą, naudojant kuriuos nors daugiamačių skalių algoritmus, nes SOM tinklas atlikdamas vektorių klasterizavimą, sumažina įvesties vektorių aibę iki vektorių–nugalėtojų aibės.

1.9. Skyriaus išvados

Šiame skyriuje apžvelgti pagrindiniai disertacijoje naudojami daugiamačių skalių metodai ir jų algoritmai, bei jų kaita. Atskleistos jų savybės ir išplėstos galimybės. Disertacijoje nagrinėtų metodų teorijai ir realizacijoms yra naudingi šie pirmame skyriuje pateikti disertacinio darbo rezultatai:

1. Teoriškai išanalizuoti daugiamačių skalių klasės algoritmai, nurodyti jų privalumai ir trūkumai.
2. Įrodyta, kad SOM tinklo permokomų neuronų skaičiaus laiptiškai mažėja, didėjant mokymo epochos eilės numeriui ir sumažėja vienetu po $e' = \left[\frac{n'e}{k'}\right] - \left[\frac{(n'-1)e}{k'}\right]$ $(n' = 1, \dots, k' - 2)$ epochos. Čia $k' \leq 100$ stačiakampės formos neuroninio tinklo didesniąją briauną sudarančių neuronų skaičius.
3. Nesant įėjimo vektorių normavimo pagal vektoriaus ilgį, galima naudoti SOM tinklo ląstelių spalvinimą pilkos spalvos atspalviais, priklausantį nuo ląstelės neurono ilgio.
4. RPM algoritme galima naudoti atstumų funkciją, užtikrinančią paklaidos minimizavimo algoritmo konvergavimą, atsisakant konvergavimą skatinančių papildomų parametrų.

2

Tyrimų metodologija

Šiame skyriuje nagrinėjamos tyrimuose naudojamos duomenų parengimo, prieš pateikiant juos vizualizavimo algoritmams, problemos. Taip pat pateikti teoriniai rezultatai, sprendžiantys Sammono algoritmo pradinių vektorių iniciacijos problemą. Nemažą skyriaus dalį užima projekcijos topologijos išsaugojimo kriterijų analitinė apžvalga ir naudojimo pagrindimas. Taip pat nagrinėjamos ir vizualizavimo algoritmų kūrimo problemos.

2.1. Tyrimuose naudojami duomenys

Prieš gerą dešimtmetį vizualios analizės eksperimentai buvo atliekami su nedidelės apimties duomenų aibėmis, nes turėtų kompiuterių techninės galimybės ribodavo domenų apimtis. Tyrimuose tyrinėjami duomenys kurių eilė yra apie $s \cdot n \approx 10^4$, kur s ir n – duomenų matricos eilučių ir stulpelių skaičiai. Sparčiai tobulėjant kompiuterinei įrangai (didėjant branduolių skaičiui procesoriuose, operatyvios atminties kiekiui, kompiuterio pagrindinės magistralės pralaidumui, bei kieto disko darbo efektyvumui) labai sutrumpėja algoritmų skaičiavimo laikas, bei kinta pačių algoritmų schemos. Taigi algoritmų tyrimui galima naudoti vis didesnės apimties duomenų aibes ir

algoritmus reikalaujančius daugiau operacijų atliekamų kompiuterio procesoriuose, kas sąlygojo skirtingų apimčių duomenų aibių pasirinkimą.

Disertacijos eksperimentinėje dalyje buvo analizuojamos šios duomenų aibės:

1. Kasikinė Fišerio irisų („Irisų“) duomenų aibė (Fisher 1936). Ją sudaro 150 irisų vainiklapio pločio ir aukščio, bei taurėlapio pločio ir aukščio matavimų rinkinys. Rinkinyje yra po 50 egzempliorių iš trijų skirtingų irisų gėlės rūšių: „Iris Setosa“, „Iris Versicolor“ ir „Iris Virginica“.
2. „Wood“ duomenų aibę (Draper and Smith 1966) sudaro 20 vektorių priklausančių penkiamatei edvei iš kurių keturi (jų numeriai 4, 6, 8, 19) yra žymiai nutolę nuo bendros grupės (angl. outliers) ir suformuoja atskirą klasterį.
3. „HBK“ duomenų aibę (Hawkins *et al.* 1984) sudaro 75 vektoriai priklausantys keturmatei eidvei, kuriuos galima suskirstyti į tris grupes: 1–10 vektoriai suformuoja pirmąją grupę, 11–14 antrąją ir 15–75 trečiąją grupę.
4. „Vyno“ (angl. wine) duomenų aibę galima parsisiųsti iš testinių duomenų aibių saugyklos „UC Irvine machine learning repository“ (Frank and Asuncion 2010). Ši duomenų aibė suformuota pagal cheminių tyrimų rezultatus, tiriant vyno brendimą. Ištirtos trijų to paties regiono vynuogių kultūros ir gauta 178 vektoriai priklausantys trylikamatei erdvei.
5. „Krūties vėžio“ arba „Vėžio“ duomenų aibė (Frank and Asuncion 2010) sudaryta iš 699 vektorių priklausančių devynmatei erdvei. Šią duomenų aibę sukūrė Viskonsino (angl. Wisconsin) universitetinė ligoninė.
6. „Klasteriai“ yra 100 vektorių duomenų aibė sugeneruojama imant 10 vektorių priklausančių R^{10} erdvei ir apie kiekvieną iš jų pagal normalųjį skirstinį parenkant papildomai 9 vektorius.
7. „Ellipsoidal“ $[s; n]$ duomenų aibė, čia $s = 3140$ ir $n = 50$; duomenų aibės vektoriai suformuoja 10 persidengiančių elipsoidinio tipo klasterių.

7 duomenų aibė sugeneruota naudojant elipsoidinių klasterių generatorių (Handl and Knowles 2010). Šis generatorius sukuria elipsoidinius klasterius. Klasterių ribos apibrėžiamos keturiais parametrais:

- centras;

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

Taškai generuojami kiekvienam klasteriui atskirai. Tikrinama, ar neperžengtos elipsoidui apibrėžtos ribos, jei taip – netinkami vektoriai atmetami. Naudojant tokį generatorių buvo sugeneruota „Ellipsoidal“ $[3140; 50]$ duomenų aibė.

8. „Gaussian“ $[s; n]$ duomenų aibė, čia $s = 2729$ ir $n = 10$; duomenų aibės vektoriai suformuoja 10 persidengiančių klasterių (2.1 pav. kairėje). „Gaussian“ $[2729; 10]$ duomenų aibė yra generuota naudojant Gauso generatorių. Detalus šio generatoriaus aprašymas pateiktas (Handl and Knowles 2010).

2.1 pav. SMACOF algoritmu naudojant pagrindinių komponenčių metodą vektoriams parinkimui ir atlikus 1000 iteracijų gautos projekcijos: kairėje „Gaussian“ ($E_{DS} = 0{,}272756$) ir dešinėje „Paraboloid“ duomenų aibės ($E_{DS} = 0{,}208293$)

9. „Paraboloid“ [s; n] duomenų aibė, čia $\mathrm{s} = 2583$ ir $\mathrm{n} = 3$. Duomenų aibės vektoriai suformuoja du nepersidengiančius klasterius (2.1 pav. dešinėje). „Paraboloid“ $[2583; 3]$ duomenų aibė sudaryta iš dviejų klasių vektorių, kurie generuoti tokiu būdu: pirmos dvi koordinatės x_{i1}

ir x_{i2} 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 $x_{i3} = 1{,}8 \cdot \sqrt{x_{i1}^2 + x_{i2}^2}$. Sukurtas Paraboloidas yra pasukamas apie koordinačių pradžios vektorių.

10. „Sferos“ $[1446; 3]$, tai tolydžiai vienas nuo kito ant vienetinės sferos išsidėsčiusių vektorių aibė (2.2 pav. kairėje).
11. „Abalone“ $[s;\ n]$ duomenų aibė, čia $s = 4177$ ir $n = 7$, kurią sudaro 29 klasteriai (2.2 pav. dešinėje).

„Abalone“ $[4177; 7]$ duomenų aibė paimta iš duomenų saugyklos „UCI machine learning repository“ (Frank and Asuncion 2010). Kiekvienas vektorius sudarytas iš 7 moliuskų parametrų:

- x_{i1} – ilgis (ilgiausia kiauto dalis);
- x_{i2} – skersmuo (statmenas ilgiui);
- x_{i3} – kiauto aukštis;
- x_{i4} – moliusko svoris kartu su kiautu;
- x_{i5} – moliusko svoris be kiauto;
- x_{i6} – vidaus organų svoris,
- x_{i7} – 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 $\bar{x}_j$, $j = \overline{1,n}$ ir dispersija σ_j^2, $j = \overline{1,n}$ pagal s turimų reikšmių, kiekvieno parametro reikšmė x_{ij} normuota naudojantis formule: $\tilde{x}_{ij} = (x_{ij} - \bar{x}_j)/\sigma_j$.

12. „Satimage“ $[6435; 36]$, tai duomenų aibė nusakanti septynias klases, kurios vektorių atributai yra skaičiai nuo 0 iki 255 (Handl and Knowles 2010). 2.2 pav. apačioje pateikta šios aibės projekcija SMACOF algoritmu.

2.2 pav. SMACOF algoritmu naudojant pagrindinių komponenčių metodą vektoriams parinkimui ir atlikus 1000 iteracijų gautos projekcijos: kairėje „Sferos“ ($E_{DS} = 0{,}217195$), dešinėje Abolone“ ($E_{DS} = 0{,}012506$), apačioje „Satimage“ ($E_{DS} = 0{,}094820$) duomenų aibės

Naudojami duomenys eksperimentuose buvo centruojami atimant vidurkius ir normuojami pagal dispersiją, kadangi disertacijoje naudojamų DS algoritmų rezultatui duomenų tiesinės transformacijos įtakos neturi (Gower 1966; Bronshtein *et al.* 2004).

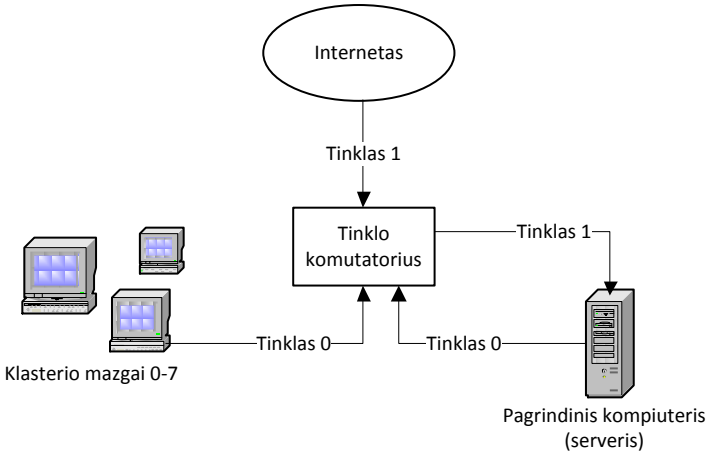
2.2. Tyrimuose naudota kompiuterinė įranga

Rengiant disertacijos darbą buvo naudojami dviejų tipų kompiuteriai:

1. Kompiuteriai su 1,8 Ghz „Pentium IV“ procesoriumi ir 1024 MB atminties. Šie kompiuteriai buvo sujungti į vietinį tinklą ir turėjo įdiegtą

MPI (1.2.0 versija) servisą. Tai juose įgalino atlikti tiek nuoseklius, tiek ir lygiagrečius skaičiavimus.

2. Devyni kompiuteriai su dviejų branduolių „AMD Athlon“ procesoriais veikiančiai 2,4 Ghz dažniu ir turintys 2048 MB operatyvios atminties. Visi kompiuteriai veikė, kaip paskirstytos atminties klasteris (2.3 pav.).



2.3 pav. Naudoto klasterio struktūros schema

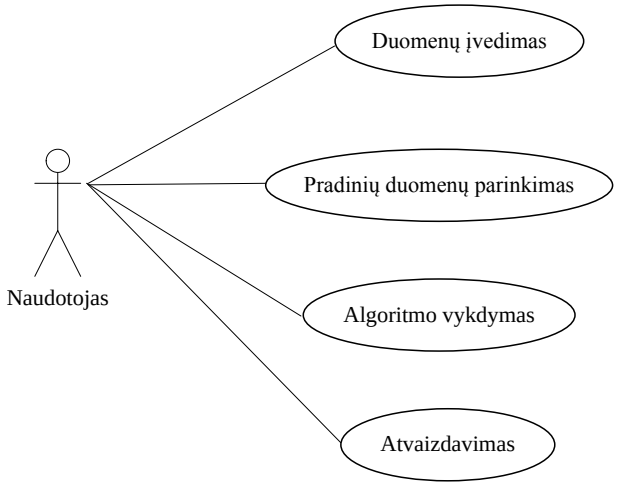
Klasteryje yra įdiegta 32 bit „Rocks cluster“ (www.rocksclusters.org) 5.1 versija, turinti OpenMPI (1.2.7 versija), Qsub servisus, todėl galima klasteryje kompiliuoti ir vykdyti norimus lygiagrečius skaičiavimus. Klasterį, kurio schema pateikta 2.3 paveiksle, sudaro vienas pagrindinis (*angl. front*) kompiuteris ir aštuoni pagalbiniai (*angl slave*) kompiuteriai arba dar kitaip vadinami klasterio mazgais (*angl. node*). Klasterio internetinės prieigos adresas yra „cluster.mii.lt“.

2.3. Tyrimuose naudota programinė įranga

Šiame skyriuje aptariamos daugiamačių duomenų vizualizavimui naudotos programinės įrangos kūrimo problemos. Programinės įrangos tikslas – apjungti vizualizavimo priemones į visumą, kurią galima būtų efektyviai panaudoti tiek užduoties formulavimui, tiek ir jos sprendimo efektyviam suradimui.

Buvo siekiama specifikuoti programinę įrangą, leidžiančią naudotojui atlikti pagrindines funkcijas tokias kaip: duomenų įvedimas, pradinių duomenų iniciacija, daugiamačių skalių tipo algoritmų vykdymas ir gautos projekcijos

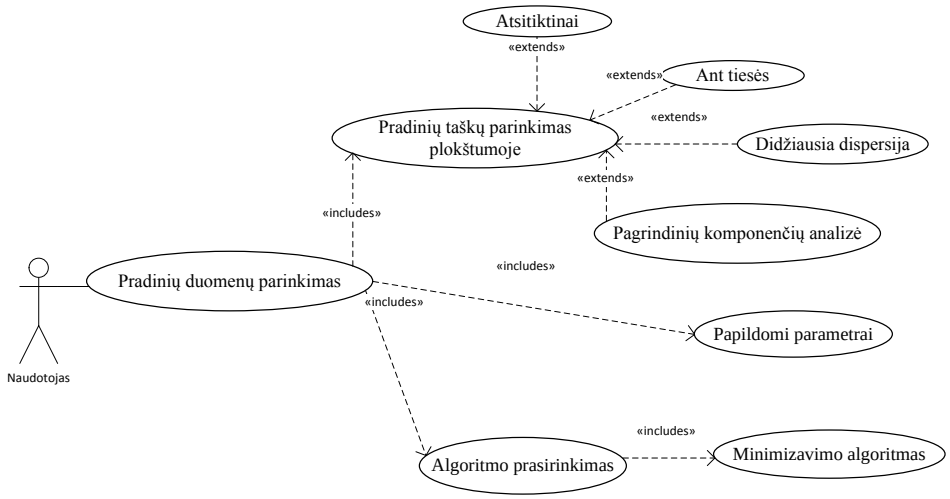
pateikimas arba vizualizavimas. Apibendrinti funkciniai reikalavimai pateikti 2.4 paveiksle.



2.4 pav. Daugiamačių duomenų vizualizavimo algoritmus apjungiančios sistemos naudojimo būdo diagrama

Naudotojas turi turėti galimybes pateiktas 2.5 paveiksle:

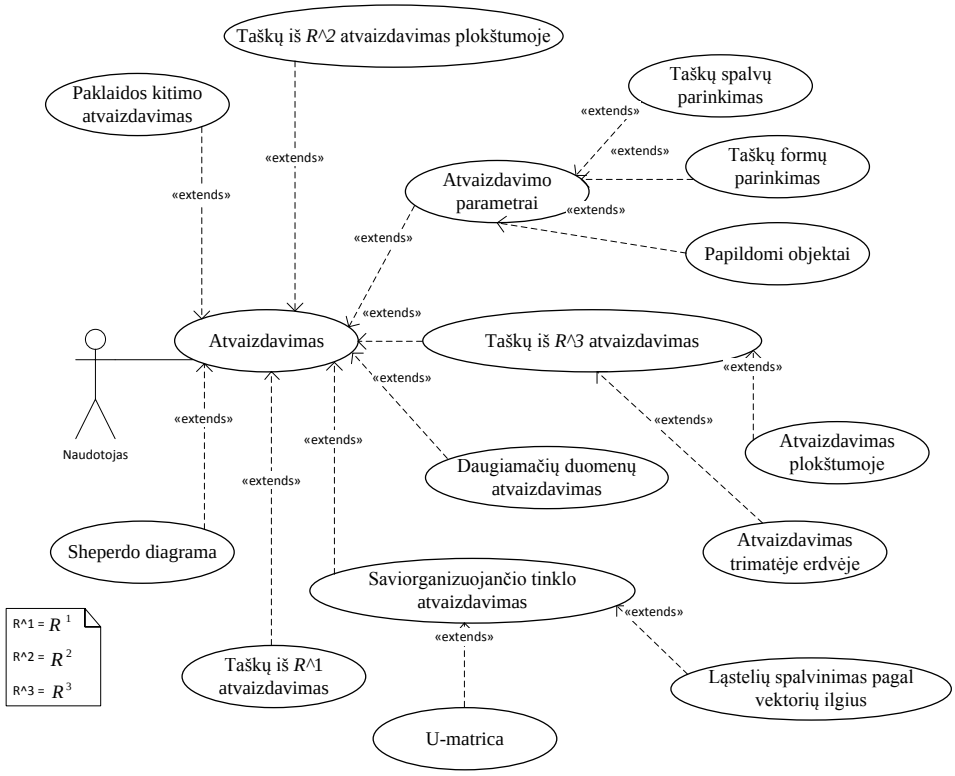
1. Nurodyti programai duomenis, su kuriais bus atliekami skaičiavimai. Šiuos duomenis esant reikalui galima sumaišyti, išrinkti, šalinti eilutes ir stulpelius, nurodyti klases ir juos normalizuoti.
2. Parinkti pradinius vektorius, pagal norimą pradinių vektorių parinkimo strategiją ar būdą (PKA, atsitiktinai tam tikroje srityje, didžiausių dispersijų, ant tiesės). Šis naudojimo būdas (*angl. use case*) apima ir projekcijos algoritmo parametrų, tokių kaip projekcijos paklaidos minimizavimo metodas, parinkimą. Kiekvieną iš paminėtų programos aspektų galima įsivaizduoti kaip atskirą (smuklesnį) naudojimo būdą, o jų tarpusavio ryšiai detaliai pateikti 2.5 paveiksle.
3. Įvykdyti pasirinktam algoritmui reikalingus skaičiavimus, nepriklausomai nuo operacinės sistemos, kurioje šie skaičiavimai atliekami.
4. Atvaizduoti apskaičiuotas įvesties duomenų transformacijas kompiuterio ekrane (2.6 pav.).



2.5 pav. Pradinių duomenų parinkimo naudojimo būdo diagrama

Šiuo atveju programa apima tik disertacijoje taikytus iniciacijos metodus: atsitiktinė iniciacija, pagal pagrindinių komponenčių analizę, atsitiktinai ant tiesės ir pagal didžiausias duomenų aibės vektorių komponenčių dispersijas. Taip pat naudotojas gali pasirinkti vizualizavimo algoritmus, pagrįstus daugiamatėmis skalėmis: SMACOF, Sammon, SOM, RPM ir t. t. Tarp kitų naudotojui leistinų pasirinkti parametrų gali būti nepanašumo funkcijos ar algoritmo sustojimo sąlygos pasirinkimas.

Duomenų vizualizavimo naudojimo būdas (2.6 pav.) apima galimus scenarijus, skirtus tiek išryškinti tam tikras duomenų savybes, tiek keisti pačios projekcijos savybes. Suprojektuotos programos galimybės apima tik bazinius duomenų pateikties kompiuterio ekrane būdus: Šepardo (*angl Shepard*) diagrama (de Leeuw 2005) ir jos įvairios modifikacijos (Fayyad *et al.* 2002), SOM tinklo atvaizdavimas U-matrica ir remiantis vektorių ilgiu, atvaizdavimas tiesėje, plokštumoje ir erdvėje. Papildomai sistema gali atvaizduoti projekcijos paklaidos kitimą algoritmo vykdymo metu, bei turi galimybę keisti projekcijos išvaizdos savybes (taškų forma ir spalva, jungimo medis, ir t. t.).



2.6 pav. Atvaizdavimo naudojimo būdų diagrama

Tokios programinės įrangos sukūrimo poreikis iškilo tuomet, kai prireikia apjungti turimas įvairių DS algoritmų realizacijas į vieną sistemą. Tai savo ruožtu leistų pasinaudoti objektinio programavimo galimybes ir išvengti programinio kodo dubliavimo. Taip pat programinė įranga turi turėti: apibrėžtą struktūrą, klasių ir duomenų tipų hierarchijas tam, kad būtų galimą lengvai papildyti turimą programą naujomis funkcijomis ir vizualizavimo algoritmais. Išsamus programos projektas buvo parengtas kartu su Vilniaus kolegijos praktikantu E. Abariumi ir yra aprašytas jo baigiamajame darbe (Abarius 2009).

Dažnai iškylanti problema yra programos veikimo užtikrinimas skirtingose operacinėse sistemose, tokiose kaip: *Unix, Linux ar Windows*. Taigi programai yra keliami reikalavimai programos kodui, kuris turi būti kaip įmanoma universalesnis jo kompiliavimo skirtingais kompiliatoriais prasme. Taip pat sukompiliuota programa turi veikti efektyviai ir leisti atlikti sudėtingus eksperimentus. Šiems reikalavimams įgyvendinti yra tinkama C programavimo kalba, o grafinį programos interfeisą tikslinga perkelti į interneto serverį,

kuriamą naudojantis HTML, JavaScript, PHP technologijomis. Sukurtas veikiantis tokios programos pavyzdys, skirtas vykdyti eksperimentus kompiuterių klasteryje, prieinamas adresu *http://cluster.mii.lt/visualization*.

2.4. Pradinių vektorių reikšmių parinkimas

Pradinių projekcijos vektorių Y_i parinkimas turi įtaką galutiniam netiesinės projekcijos metodo rezultatui. Vizualizavimo algoritmuose naudojami optimizavimo metodai dažnai suranda lokalų, o ne globalų projekcijos kokybę charakterizuojančios funkcijos minimumą. Todėl pradinių Y_i vektorių parinkimas yra labai svarbus, nes parinkus skirtingas Y_i vektorių aibes, gaunami skirtingi lokalūs minimumo taškai.

Pradinių vektorių $Y_i \in R^d$, $d = 1,2,3$ parinkimas gali būti atliktas įvairiais būdais. Vienas iš paprasčiausių būdų yra atsitiktinis šių vektorių koordinačių parinkimas tam tikroje srityje. Srities forma dažniausiai yra kvadratas ar kubas, nors gali būti ir kitokios formos, kaip tiesė, plokštuma, sfera ir t. t. Naudojant atsitiktinį vektorių koordinačių parinkimą, projekcijos algoritmas kartojamas keletą kartų su skirtingais pradiniais Y_i vektorių rinkiniais. Projekcija, turinti mažiausią minimizuojamos funkcijos reikšmę, yra pasirenkama kaip šio algoritmo projekcijos rezultatas. Šis metodas naudojamas SOM_PAK programinėje įrangoje (Kohonen *et al.* 1996), tačiau su maža modifikacija: pirmoji pradinio vektoriaus koordinatė parenkama atsitiktinai, o antroji taip, kad gautas vektorius Y_i būtų lygiagretus pasirinkai tiesei. Ši modifikacija dažnai taikoma nors neturi teoriškai pagrįsto paaiškinimo, kodėl toks pradinių vektorių parinkimo būdas yra tinkamas.

Teoriškai panagrinėsime, kaip atveju $d = 2$ keičiasi taškų projekcijos jei pradiniai $Y_i = \{y_{i1}, y_{i2}\}$ vektoriai parenkami ant tiesės (Murtagh 2004):

$$y_{i1} = a \cdot y_{i2} + b, \tag{2.1}$$

čia a ir b konstantos.

Sammono projekcijos metodas minimizuoja paklaidą E_S (1.23) naudojantis pseudo-Niutono (*angl. pseudo-Newton)* algoritmu (1.24).

Pirmoji paklaidos E_S funkcijos išvestinė kintamojo y_{ik} atžvilgiu skaičiuojama pagal formulę (2.2).

$$\frac{\partial E_S}{\partial y_{ik}} = -\frac{2}{c}\sum_{\substack{j=1,\\ j\neq i}}^{s}\left(\frac{\delta_{ij} - d_{ij}}{\delta_{ij}d_{ij}}\right)\left(y_{ik} - y_{jk}\right), \qquad k = 1,2, \tag{2.2}$$

kur $c = \sum_{\substack{i=1 \\ j<i}}^{n} \delta_{ij}$.

Įterpus (2.1) į (2.2), kai $k = 1$ gauname:

$$\frac{\partial E_S}{\partial y_{i1}} = -\frac{2}{c}\sum_{\substack{j=1, \\ j\neq i}}^{s}\left(\frac{\delta_{ij} - d_{ij}}{\delta_{ij} d_{ij}}\right)\left(a \cdot y_{i2} + b - \left(a \cdot y_{j2} + b\right)\right) = \\ = -\frac{2a}{c}\sum_{\substack{j=1, \\ j\neq i}}^{s}\left(\frac{\delta_{ij} - d_{ij}}{\delta_{ij} d_{ij}}\right)\left(y_{i2} - y_{j2}\right) = a\frac{\partial E_S}{\partial y_{i2}}. \tag{2.3}$$

Antroji paklaidos E_S funkcijos išvestinė kintamojo y_{ik} atžvilgiu skaičiuojama pagal formulę (2.4).

$$\frac{\partial^2 E_S}{\partial y_{ik}^2} = -\frac{2}{c}\sum_{\substack{j=1, \\ j\neq i}}^{s}\frac{1}{\delta_{ij} d_{ij}}\left[\left(\delta_{ij} - d_{ij}\right)\right. \\ \left. - \frac{\left(y_{ik} - y_{jk}\right)^2}{d_{ij}}\left(1 + \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right)\right], k = 1{,}2\,. \tag{2.4}$$

Įstačius (2.1) į (2.4), kai $k = 1$ gauname:

$$\frac{\partial^2 E_S}{\partial y_{ik}^2} = -\frac{2}{c}\sum_{\substack{j=1, \\ j\neq i}}^{s}\frac{1}{\delta_{ij} d_{ij}}\left[\left(\delta_{ij} - d_{ij}\right)\right. \\ \left. - \frac{a^2\left(y_{i2} - y_{j2}\right)^2}{d_{ij}}\left(1 + \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right)\right]. \tag{2.5}$$

Apskaičiuokime skirtumą:

$$\frac{\partial^2 E_s}{\partial y_{i1}^2} - \frac{\partial^2 E_s}{\partial y_{i2}^2}.$$

Antrųjų dalinių išvestinių skirtumas yra lygus:

$$\frac{\partial^2 E_S}{\partial y_{i1}^2} - \frac{\partial^2 E_S}{\partial y_{i2}^2} = -\frac{2}{c}\sum_{\substack{j=1,\\ j\neq i}}^{s} \frac{1}{\delta_{ij} d_{ij}} \cdot$$

$$\cdot \left[\left(\delta_{ij} - d_{ij}\right) - \frac{a^2 \left(y_{i2} - y_{j2}\right)^2}{d_{ij}} \left(1 + \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right) \right] +$$

$$+ \frac{2}{c}\sum_{\substack{j=1,\\ j\neq i}}^{s} \frac{1}{\delta_{ij} d_{ij}} \cdot \left[\left(\delta_{ij} - d_{ij}\right) - \frac{\left(y_{i2} - y_{j2}\right)^2}{d_{ij}} \left(1 + \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right) \right] = \tag{2.6}$$

$$= \frac{2(a^2 - 1)}{c} \sum_{\substack{j=1,\\ j\neq i}}^{s} \frac{1}{\delta_{ij} d_{ij}} \left[\frac{\left(y_{i2} - y_{j2}\right)^2}{d_{ij}} \left(1 + \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right) \right].$$

Kadangi atstumai erdvėse R^2 ir R^n yra teigiami ($d_{ij} \geq 0, \delta_{ij} \geq 0$) ir bendru atveju visi kartu nelygūs nuliui, kai $y_{i2} \neq y_{j2}$, todėl teisinga 2.1 išvada.

2.1 Išvada. $\frac{\partial^2 E_S}{\partial y_{i1}^2} - \frac{\partial^2 E_S}{\partial y_{i2}^2} = 0$ tada ir tik tada, kai taškai yra išsidėstę ant tiesės, kurios krypties koeficientas a tenkina lygybę $a^2 - 1 = 0$, t. y. $a = \pm 1$.

2.1 Teorema. Jei pradiniai Sammono projekcijos taškai $Y_i = \{y_{i1}, y_{i2}\}$ yra išsidėstę ant tiesės $y_{i1} = a \cdot y_{i2} + b$, kur $a = \pm 1, b \in R$, tai projekcijos taškai, gauti minimizuojant Sammon paklaidą pagal iteracinę formulę (1.24), taip pat bus išsidėstę ant tiesės $y_{i1} = a \cdot y_{i2} + b$.

Įrodymas.

Teorema įrodoma remiantis matematinės indukcijos metodu.

Jeigu taškas $Y_i = \{y_{i1}, y_{i2}\}$ priklauso tiesei, tai jisai tenkina lygybę:

$$y_{i1}(m') = a \cdot y_{i2}(m') + b. \tag{2.7}$$

Teiginys, kad projekcijos taškai yra ant tiesės $y_{i1}(m') = a \cdot y_{i2}(m') + b$ yra teisingas, kai $m' = 0$, nes pradiniai taškai yra parenkami ant šios tiesės.

Tarkime, kad teiginys (2.7) teisingas, kai iteracijos numeris lygus m'.

Įrodykime, kad teiginys (2.7) yra teisingas, kai iteracijos numeris lygus $m' + 1$. Taškų $Y_i \in R^2$ koordinatės y_{ik}, $i = \overline{1,n}$, $k = 1,2$ $(m' + 1)$ - osios iteracijos metu persiskaičiuojamos pagal (1.24) formulę:

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

Įstačius (2.1) į (1.24), kai $k = 1$ ir pritaikius (2.7) prielaidą, gauname:

$$\begin{aligned} y_{i1}(m'+1) &= a \cdot y_{i2}(m') + b - \alpha \frac{a\frac{\partial E_S(m')}{\partial y_{i2}(m')}}{\left|\frac{\partial^2 E_S(m')}{\partial y_{i1}^2(m')}\right|} = \\ &= a \cdot \left(y_{i2}(m') - \alpha \frac{\frac{\partial E_S(m')}{\partial y_{i2}(m')}}{\left|\frac{\partial^2 E_S(m')}{\partial y_{i1}^2(m')}\right|} \right) + b. \end{aligned} \tag{2.8}$$

Atsižvelgiant į 2.1 išvadą, gauname

$$\begin{aligned} y_{i1}(m'+1) &= a \cdot \left(y_{i2}(m') - \alpha \frac{\frac{\partial E_S(m')}{\partial y_{i2}(m')}}{\left|\frac{\partial^2 E_S(m')}{\partial y_{i2}^2(m')}\right|} \right) + b = \\ &= a \cdot y_{i2}(m'+1) + b. \end{aligned} \tag{2.9}$$

Taigi teiginys, kad $y_{i1}(m') = a \cdot y_{i2}(m') + b$ (visi projekcijos taškai yra ant tiesės) yra teisingas, tada ir tik tada kai $a = \pm 1, b \in R$. □

Tokia pati teorema gali būti suformuluota ir įrodyta daugiamatėms skalėms, kurios naudoja pseudo-Niutono metodą paklaidai minimizuoti. Šiuo atveju a ir b yra bet kokie skaičiai.

Išvada 2.2. Iš teoremos 2.1 seka, kad projekcijos taškai visada bus ant tos pačios tiesės. Tačiau, eksperimentiniai tyrimai parodė, kad jau po pirmos iteracijos taškai nežymiai nutolsta nuo šios tiesės. Atlikus dar keletą iteracijų, projekcijos taškai išsibarsto po visą plokštumą. Tai įvyksta todėl, kad atliekant skaičiavimus kompiuteriu, gautus rezultatus įtakoja kompiuterio skaičiavimo ir skaičių apvalinimo paklaidos. Taigi pradinių taškų iniciacija ant tiesės yra galima, tačiau šiuo atveju paklaidos konvergavimas iteracinio proceso pradžioje yra lėtas.

Sudėtingesnis pradinių taškų iniciacijos būdas yra pagrindinių komponenčių analizės metodo taikymas. Tokiu būdu pirmiausiai daugiamačiai duomenys yra suprojektuojami į plokštumą naudojantis PKA metodu ir gauti taškai yra laikomi

atitinkamų įvesties taškų projekcijos (pradiniais) taškais. Deja, pagrindinių komponenčių paieška reikalauja papildomų laiko ir kompiuterio resursų.

Disertacijoje pasiūlytas paprastesnis būdas: apskaičiuojame kiekvienos duomenų aibės X vektorių komponentės dispersiją. Gautos dispersijos palyginamos tarpusavyje ir surandamos dvi didžiausias dispersijas turinčios vektorių komponentės, kurių numeriai yra k_1, k_2. Šias komponentes atitinkančios vektoriaus X_i koordinatės yra laikomos jo projekcijos vektoriaus Y_i pradinėmis koordinatėmis $y_{i1} = x_{ik_1}, y_{i2} = x_{ik_2}$, o tokį pradinių koordinačių iniciacijos metodą vadinsime didžiausių dispersijų metodu. Eksperimentai parodė (Bernatavičienė *et al.* 2005), kad šis metodas yra vienas iš efektyviausių pradinių taškų iniciacijos metodų. Plačiau apie eksperimentus žr. 3 skyriuje.

2.5. Kiekybiniai atvaizdavimo įvertinimo kriterijai

Daugiamačius duomenis galima atvaizduoti įvairiais metodais, šie metodai dažnai minimizuoja skirtingas paklaidos funkcijas, todėl iškyla objektyvaus atvaizdavimų tarpusavio palyginimo problema. Norint ją išspręsti reikia parinkti kriterijų, kuris apibūdintų projekcijos kokybę ir tiktų skirtingiems metodams. Pateiksime keletą tokių projekcijų panašumo kriterijų.

Metrinės topologijos išsaugojimas (MTI).

2.1. **Apibrėžimas.** Tegu duotos erdvės R^n $(X_i \in R^n)$ ir R^d $(Y_i \in R^d)$, $1 \le d < n$, tuomet projekciją (transformaciją) $\Phi: X \to T$ vadinsime išsaugančią metrinę topologiją, jeigu

$$\delta_{ij} \le \delta_{kl} \Longrightarrow d_{ij} \le d_{kl}, \forall i, j, k, l, \qquad (2.10)$$

čia δ_{ij} ir d_{ij} yra atstumai (skirtumingumai arba panašumai) tarp taškų atitinkamai erdvėse R^n ir R^d (Bezdek and Pal 1995).

Metrinės topologijos transformacija užima tarpinę vietą tarp transformacijos, kuri nekeičiančia kaimynų, bet neišsaugo atstumų eilės, ir izometrinio atvaizdžio, kuris nekeičia ne tik atstumų eilės, bet ir pačių atstumų. Galima sakyti, kad MTI transformacija išsaugo kaimynus tarp atitinkamų taškų R^n ir R^d erdvėse bei reliatyvius atstumų sąryšius.

Norint įvertinti turimos transformacijos Φ metrinės topologijos išsaugojimą, reikia naudoti Spirmeno koeficientą.

Spirmeno koeficientas (*angl. Spearmen coefficient (rho), Kendall tau*). Atstumų eilės ir dydžių išlaikymas yra svarbi MTI transformacijos charakteristika. Siekiant ją įvertinti ji pervadinama $\Delta \equiv (\delta_{ij})$ į (δ_k), kur $\delta_k = \delta_{ij}$ ir $k = (i-1)\left(\frac{2s-i}{2}\right) + (j-i)$. Analogiškai galima pernumeruoti matricą (d_{ij}). Taip sunumeruotus atstumus surūšiuojame didėjimo tvarka, tuomet k - ojo matricų (δ_k) ir (d_k) elemento padėties indeksas yra vadinamas

jo rangu ir žymimas atitinkamai $r_X(k)$ ir $r_Y(k)$. Vektoriai $r_X = \{r_X(k) \mid k = \overline{1, n^2}\}$ ir $r_Y = \{r_Y(k) \mid k = \overline{1, n^2}\}$ yra vadinami rangų vektoriais (Bezdek and Pal 1995).

Nukrypimas nuo MTI gali būti apskaičiuotas rangų koreliacijos koeficientu (2.11), kitaip Spirmeno koeficientu:

$$\rho_{Sp}(r_Y, r_X) = 1 - \frac{6\sum_{k=1}^{T}\left(r_Y(k) - r_X(k)\right)^2}{T^3 - T}, \tag{2.11}$$

čia $T = \frac{\mathrm{s(s-1)}}{2}$.

Spirmeno koeficiento reikšmė yra intervale $-1 \leq \rho_{Sp} \leq 1$. Metrinė topologija pilnai išlaikoma, kai ši reikšmė yra lygi 1. Pagal (Bezdek and Pal 1995), transformacija yra MTI tada ir tik data, kai $\rho_{Sp} = 1$. Transformacija vadinama anti-MTI, jei $\rho_{Sp} = -1$, tai yra, visi atstumai tarp projekcijos taškų yra išdėstyti atvirkštine tvarka negu atstumai duomenų erdvėje R^n.

Spirmeno koeficientas taip pat tinka įvertinti projekcijos topologijos išsaugojimą. Jis gali būti naudojamas įvertinti ir lokalius atstumus išsaugančių projekcijos algoritmų projekcijoms (Karbauskaitė and Dzemyda 2009).

Kai rangai gali būti ir vienodi, tuomet vietoje Spirmeno koeficiento naudojama Pirsono (*angl. Pearson*) koreliacija arba tiesiog standartinė koreliacija, apskaičiuojama pagal (2.12) formulę:

$$corr(r_X, r_Y) = \frac{E\left(r_X - E(r_X)\right)E\left(r_Y - E(r_Y)\right)}{\sigma(r_X)\sigma(r_Y)}. \tag{2.12}$$

Minimalaus jungimo (MJ) kriterijus. Tarkime erdvėse R^n ir R^d apibrėžta simetriška panašumo funkcija, kuri kiekvienai erdvės taškų porai priskiria teigiamą skaičių, atspindintį šių taškų panašumą (arba skirtingumą). Pavadinkime šias funkcijas F ir G atitinkamai erdvėms R^n ir R^d, bei apibrėžkime paklaidos (arba baudos) funkciją:

$$C = \sum_{i=1}^{s}\sum_{j<i} F(i,j)G\left(M(i), M(j)\right), \tag{2.13}$$

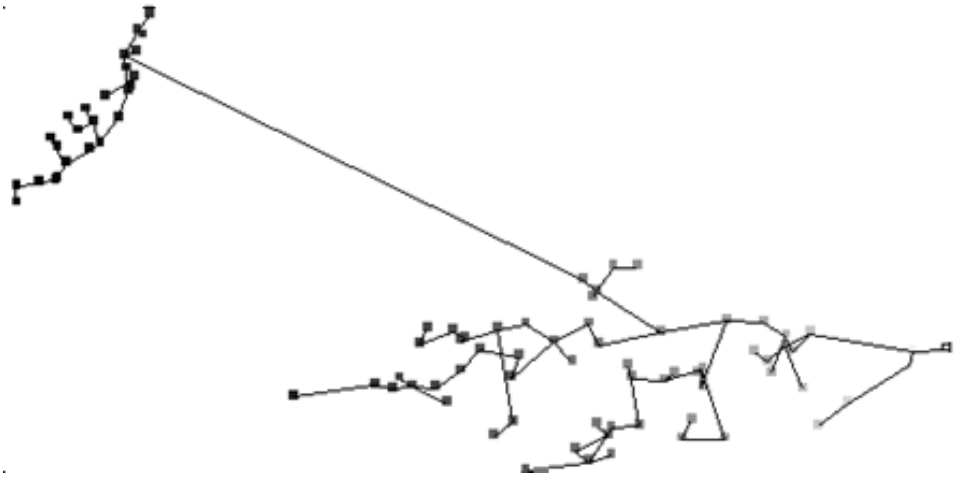
čia i ir j yra R^n taškų numeriai, o $M(i)$ ir $M(j)$ šių taškų projekcijos (vaizdai) erdvėje R^d.

Paklaidos funkciją (2.13) laikysime MJ kriterijumi, kai

$$F(i,j) = \begin{cases} 1, & i, j\ kaimynai; \\ 0, & ne\ kaimynai. \end{cases} \tag{2.14}$$

$$G\big(M(i), M(j)\big) = \|M(i) - M(j)\|^p. \qquad (2.15)$$

Taigi, geru atvaizdavimu pagal MJ kriterijų laikomas vaizdas, kuriame artimi R^n erdvės taškai yra arti vienas kito ir erdvėje R^d. Taškų artumas (2.15) formulėje gali būti matuojamas tiek Euklido atstumu ($p = 2$), tiek ir bet kokiu kitu Minkovskio atstumu (Goodhill and Sejnowski 1996). Disertacijoje kriterijaus reikšmei apskaičiuoti naudojamas Euklidinis atstumas.



2.7 pav. Minimalaus jungimas projektuojant „Irisų“ duomenų aibę ($MJ = 20{,}74308$)

Mažesnės MJ kriterijaus reikšmės atitinka tikslesnes taškų projekcijas. MJ kriterijaus taikymo pavyzdys pateiktas 2.7 paveiksle. Pats jungimo medis, pagal kurį skaičiuojamas kriterijus, efektyviausiai apskaičiuojamas Prim (*angl. Prim*) algoritmu (Prim 1957).

2.6. Skyriaus išvados

Skyriuje apžvelgtos tyrimuose naudojamos duomenų parengimo, prieš pateikiant šiuos duomenis vizualizavimo algoritmams, problemos. Taip pat pateikti teoriniai rezultatai, sprendžiantys Sammono algoritmo pradinių vektorių iniciacijos problemą. Nemažą skyriaus dalį užima projekcijos topologijos išsaugojimo kriterijų, kurie buvo naudojami eksperimentinėje disertacijos dalyje, analitinė apžvalga ir naudojimo pagrindimas. Pateikta vizualizavimui skirtos sistemos vizija, kuria buvo naudojamasi kuriant programinę įrangą, reikalingą eksperimentiniams tyrimams.

Apibendrinat skyrių galima daryti šias išvadas:

1. Sammono projekcijos algoritme pradinių taškų parinkimas ant tiesės, kurios krypties koeficientas lygus $a = \pm 1$, yra netaikytinas, kadangi tai mažina algoritmo konvergavimo greitį.
2. Pradinių duomenų parinkimas ant tiesės daugiamačių skalių algoritmuose, kai paklaida minimizuojama pseudo-Niutono algoritmu, yra netikslingas, dėl lėtesnio konvergavimo.
3. Kadangi taškai ant tiesės, kurios krypties koeficientas lygus $a = \pm 1$, teoriškai turėtų išlikti ant tos pačios tiesės, taikant daugiamačių skalių algoritmus, bet tik dėl kompiuterio skaičiavimo ir skaičių apvalinimo paklaidų taškai palieka tiesę ir po keleto iteracijų išsibarsto po visą projekcijos plokštumą. Todėl yra tikslinga naudoti kitus iniciacijos būdus, kaip pavyzdžiui, pagrindinių komponenčių analizė ar didžiausių dispersijų metodas.
4. Siekiant palengvinti daugiamačių skalių metodų pritaikymą skirtingoms operacinėms sistemoms ir lygiagretiems skaičiavimams, būtinas vientisos sistemos projektas (šiame skyriuje jis yra pateiktas).
5. Vizualizuojant daugiamačius duomenis skirtingais daugiamačių skalių metodais, iškyla atvaizdavimų tarpusavio palyginimo problema. Daugiamačiai duomenys dažnai priklauso arba yra šalia daugdaros, todėl siūloma daugiamačių skalių metodus palyginti taikant daugdaros topologijos išsaugojimą įvertinančius kriterijus.

3

Eksperimentiniai tyrimai

Šiame skyriuje pateikiami gauti eksperimentinių tyrimų rezultatai, publikuoti autoriaus darbuose (A1, A2, A3 ir A4). Tyrimai buvo papildyti naujais eksperimentų su didesnėmis aibėmis rezultatais.

3.1. Sammono ir SMACOF algoritmų tyrimas

Čia pateikiami autoriaus rezultatai publikuoti (A1) straipsnyje nagrinėjant Sammono ir SMACOF algoritmų junginius su SOM algoritmu.

Šiame skyriuje nagrinėjamas daugiamačių duomenų atvaizdavimo optimizavimas. Tiriama Sammono projekcija, daugiamačių skalių SMACOF realizacija ir jų nuoseklūs junginiai su savireguliuojančiu neuroniniu tinklu, naudojantys atstumus, apskaičiuojamus pagal Euklido metriką. Tyrime analizuojamos algoritmų ir jų junginių savybės. Tiriami pradinių vektorių parinkimo metodai, bei atliekama jų lyginamoji algoritmų analizė, vertinant juos įvairiais kiekybiniais kriterijais. Kiekybinių kriterijų panaudojimas leidžia įvertinti projekcijos rezultatus ir pasirinkti geriausią. Tyrimai buvo atlikti naudojant šešias skirtingos kilmės duomenų aibes.

3.1.1. Sammono ir SMACOF skaičiavimo laikas

Norint įvertinti, kuris iš algoritmų veikia greičiau arba reikalauja mažiau operacijų, buvo atliktas eksperimentinis tyrimas, kurio rezultatai pateikti 3.1 lentelėje.

Iš pateiktų rezultatų galime daryti išvadą, kad DS SMACOF algoritmo skaičiavimo laikas yra apie 2,3 karto mažesnis nei Sammono projekcijos algoritmo, kai iteracijų skaičius yra pakankamai didelis ir vienodas abiem algoritmams. Čia viena iteracija apima visus skaičiavimus, reikalingus perskaičiuoti visų duomenų aibės taškų naujas projekcijas plokštumoje vieną kartą.

3.1 lentelė. Sammono projekcijos ir SMACOF skaičiavimo laiko vidutinė priklausomybė nuo iteracijos numerio

Iteracija	„Irisų“ duomenų aibė			„Klasteriai“		
	Sammono projekcija, *s*	SMACOF, *s*	Santykis	Sammono projekcija, *s*	SMACOF, *s*	Santykis
5	0,031000	0,015999	1,937534	0,078999	0,046000	1,717391
55	0,250000	0,109000	2,293574	0,625000	0,296999	2,104378
105	0,484999	0,217999	2,224771	1,187999	0,531000	2,237287
155	0,703000	0,328999	2,136778	1,733999	0,766000	2,263707
205	0,937000	0,405999	2,307882	2,266000	1,014999	2,232513
255	1,172000	0,515999	2,271319	2,828000	1,250000	2,262400
305	1,406000	0,593999	2,367004	3,375000	1,500000	2,250000
355	1,609000	0,718999	2,237831	3,921999	1,735000	2,260519
405	1,844000	0,796999	2,313677	4,655999	1,969000	2,364652
455	2,078000	0,905999	2,293599	5,030999	2,219000	2,267237
505	2,296999	1,000000	2,297000	5,717999	2,484999	2,301006

Skaičiavimo laiko santykis artėja prie 2,3 todėl, kad esant nedideliam iteracijų skaičiui, santykiui didelę įtaką turi laikas, sugaištas pradinių vektorių iniciacijai, nepanašumų matricos apskaičiavimui ir t. t.

3.1.2. SOM junginio su DS metodais kokybės lyginamoji analizė

Šiame poskyriuje buvo tiriami SOM_Sammono ir SOM_SMACOF algoritmų junginiai naudojant įvairias vizualizuojamų duomenų aibes. Kadangi neuroninio

tinklo SOM apmokymo rezultatai priklauso nuo jo neuronų-vektorių $m_{ij} = \{m_{ij}^1, m_{ij}^2, \dots, m_{ij}^n\}$ pradinių reikšmių, todėl tikslinga pasirinkti mažiausią kvantavimo paklaidą ($E_{SOM(kvant)}$) (1.39**)** turintį neuroninį SOM tinklą iš keleto su skirtingomis pradinėmis neuronų reikšmėmis apmokytų neuroninių tinklų. Atliekant eksperimentus SOM tinklo apmokymas buvo kartojamas 100 kartų, parenkant skirtingus atsitiktinai sugeneruotus pradinius SOM tinklo neuronus bei skirtingą įėjimo (mokymo) duomenų aibės X eilę. Tikslas – surasti neuronus-vektorius, su kuriais neuroninio tinklo kvantavimo paklaida $E_{SOM(kvant)}$ yra mažiausia. Sammono projekcijos ir DS algoritmų iteracijų skaičius eksperimentuose buvo parenkamas taip, kad abiejų algoritmų skaičiavimo laikas būtų apytiksliai vienodas. Algoritmų vykdymo laiką siekiama suvienodinti tam, kad būtų galima lyginti vien tik projekcijos paklaidas.

Algoritmų junginiais SOM_Sammono ir SOM_SMACOF gautos duomenų projekcijos buvo lyginamos remiantis 2.5 skyriuje pateiktais kriterijais.

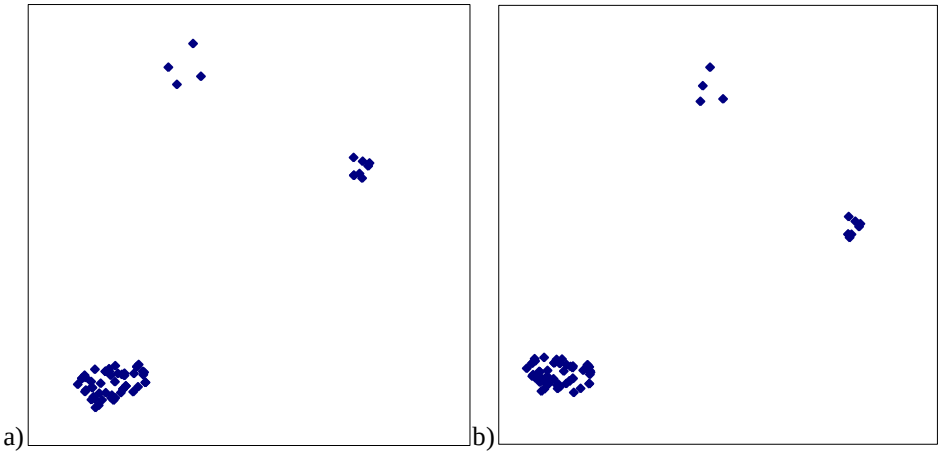
SOM tinklo mokymo rezultatas priklauso nuo to, kokios parenkamos pradinės neuronų koordinatės, būtent todėl eksperimentai buvo kartojami po 100 kartų, siekiant išrinkti vektorius-nugalėtojus su kuriais SOM tinklo kvantavimo paklaida yra mažiausia. Gauti vektoriai-nugalėtojai, buvo atvaizduojami į dvimatę plokštumą Sammono ir SMACOF algoritmais. Rezultatai pateikti 3.2 lentelėje.

3.2 lentelė. SOM_Sammono ir SOM_SMACOF junginiais gautų projekcijų kokybės kriterijų reikšmės

Kriterijus	Tipas	SOM_Sammono					
		„Irisų“	„HBK“	„Wood“	„Vyno“	„Vėžio“	„Klasteriai“
Minimalaus jungimo	mažėja	**21,94382**	**4,23492**	**1,35353**	**88,19783**	164,0650	**44,15650**
Spirmeno koeficientas	didėja	**0,99664**	**0,98705**	**0,95675**	**0,98805**	**0,98310**	0,83153
Kriterijus		SOM_SMACOF					
Minimalaus jungimo	mažėja	20,92690	3,95750	1,27246	86,29917	**181,15250**	35,78920
Spirmeno koeficientas	didėja	0,99864	0,99026	0,96069	0,98919	0,98318	**0,81109**

3.2 lentelėje stulpelis „Tipas“ nurodo, kuria, didėjimo ar mažėjimo, kryptimi kinta kriterijaus reikšmė, kai gerėja atvaizdavimo kokybė (paklaida). Taigi, jeigu kriterijaus tipas yra „mažėja“, vadinasi mažesnė jo reikšmė atitinka geresnę duomenų projekciją.

Daugumoje atvejų, pateiktų 3.2 lentelėje, SOM_SMACOF algoritmų junginiu gautos projekcijos kokybė yra geresnė už gautą SOM_Sammono algoritmų junginiu. Kokybės kriterijų reikšmės skiriasi statistiškai nereikšmingai, todėl abiejų algoritmų duomenų projekcijos yra panašios (3.1 pav.), t. y. naudojantis Kolmogorovo–Smirnovo testu (Messey 1951) buvo patikrinta hipotezė, kad dvi imtys turi tą pačią pasiskirstymo funkciją. Taigi abu algoritmų junginiai gali būti naudojami daugiamačių duomenų vizualizavimui, nes užtikrina vienodą projekcijos kokybę.



3.1 pav. „HBK“ duomenų projekcija apskaičiuota: a) SOM_Sammono junginiu; b) SOM_SMACOF junginiu

Didinat savirorganizuojančio neuroninio tinklo dydį, didėja ir neuronų-nugalėtojų skaičius, kuriuos reikia atvaizduoti į plokštumą aprašytais algoritmų junginiais. Todėl atliekant eksperimentus buvo atsižvelgiama į šią SOM tinklo savybę, parenkant skirtingus SOM tinklo dydžius ir eksperimentams taikant tik tuos tinklus, kurie labiausiai atskleidžia tyrinėjamų duomenų topologijos savybes, tokias kaip taškai atsiskyrėliai, klasteriai, kaimyniškumas ir t. t.

Pagrindinis atlikto tyrimo rezultatas yra tai, kad algoritmų junginiai (SOM_Sammono ir SOM_SMACOF) yra panašūs, nes jie abu užtikrina identišką daugiamačių duomenų projekcijos kokybę. Tai leidžia efektyviai taikyti ne tik daugumos tyrinėtojų naudojamą SOM ir Sammono junginį, bet ir jam panašų pagal paklaidą, bet greitesnį pagal skaičiavimo laiką, SOM ir SMACOF algoritmų junginį.

Eksperimentiškai parodyta, kad tirtos DS realizacijos skaičiavimo laikas yra apie 2,3 karto trumpesnis už Sammono projekcijos realizacijos skaičiavimo laiką, esant pakankamai dideliems abiejų algoritmų iteracijų skaičiams. Tai yra

susiję su tuo faktu, jog Sammono projekcijos algoritmas naudoja daugiau procesoriaus operacijų reikalaujančių matematinių operatorių (pvz. dalyba).

3.2. Diagonalinio mažoravimo algoritmo tyrimas

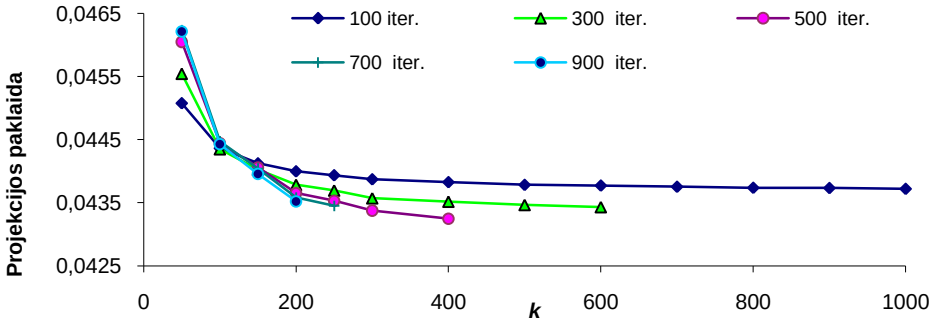
Šio skyriaus poskyriuose tyrinėjamas diagonalinio mažoravimo algoritmo efektyvumas. Taip pat tiriama, kaip priklauso projekcijos paklaida nuo iteracijų skaičiaus bei kaimyniškumo parametro k. Pateikiamų rezultatai buvo publikuoti autoriaus darbe (A2).

3.2.1. Kaimyniškumo parametro *k* parinkimas

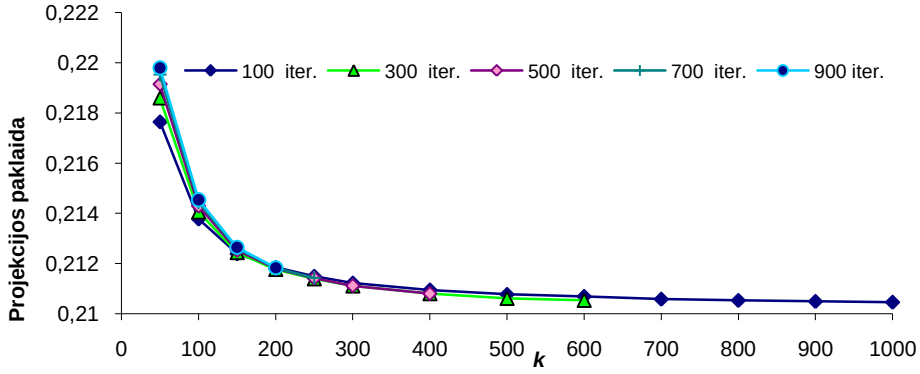
Daugiamačių duomenų aibės X vektorių kaimyniškumo parametras k yra aprašytas 1.5 skyriuje pateiktame diagonalinio mažoravimo algoritme. Nuo parametro k priklauso svorių matricos V (1.29) nenulinių elementų skaičius. Šis parametras parodo, į kiek maksimaliai i-tojo vektoriaus kaimynų pagal vietą nepanašumų matricoje reikia atsižvelgti apskaičiuojant vektoriaus X_i projekciją DMA algoritmu vienos iteracijos metu.

Atliekant preliminarius skaičiavimus su DMA algoritmu buvo pastebėta, kad nuo parametro k parinkimo labai priklauso projekcijos paklaida $E_{DS} = \sqrt{E_{norm}}$ ir gaunami vizualizavimo rezultatai. Todėl buvo nuspręsta atlikti tyrimą, kurio tikslas nustatyti, kaip kinta projekcijos paklaida, keičiant parametro k reikšmę. Čia skaičiavimo laikas ir iteracijų skaičius buvo fiksuotas. Eksperimento pradžioje kiekvienos analizuojamos aibės elementai parenkami atsitiktinai, taip siekiama, kad šalia būtų kuo mažiau panašių taškų. Su kiekviena skirtinga parametro k reikšme atliekama po 50 eksperimentų. Parametro k reikšmė kinta nuo 100 iki 1000, k didinamas kas 100. Vėliau apskaičiuojami gautų projekcijos paklaidų vidurkiai.

SMACOF ir DMA algoritmuose pradiniai vektoriai dvimatėje plokštumoje parenkami naudojant pagrindinių komponenčių analizės metodą.



3.2 pav. Projekcijos paklaidos priklausomybė nuo kaimyniškumo parametro k („Abalone“ duomenų aibė)



3.3 pav. Projekcijos paklaidos priklausomybė nuo kaimyniškumo parametro k („Ellipsoidal“ duomenų aibė)

Kaip rodo atlikto eksperimento rezultatai (3.2 pav. ir 3.3 pav.), kai parametro k reikšmė didesnė už 400 ir fiksuotas skaičiavimo laikas, jau po 300 iteracijų gaunami pakankamai tikslūs rezultatai. Projekcijos paklaida, gauta DMA algoritmu, ir paklaida, gauta SMACOF algoritmu, skiriasi mažiau nei 1 %. Didinant iteracijų skaičių projekcijos paklaida kinta nežymiai. Toliau didinant parametro k reikšmę, ilgėja skaičiavimo laikas, o vizualizavimo rezultatai artimi rezultatams, gautiems SMACOF algoritmu. SMACOF algoritmu gaunamos paklaidos lygios: $E_{DS} = 0{,}043970$ „Abalone“ duomenų aibei ir $E_{DS} = 0{,}210109$ „Ellipsoidal“ duomenų aibei. Taip pat iš šio eksperimento matyti, kad kai k parametras labai mažas ($k \approx 50$), didinant iteracijų skaičių projekcijos paklaida ne mažėja, o atvirkščiai – didėja.

3.2.2. Vektorių pradinis surikiavimas

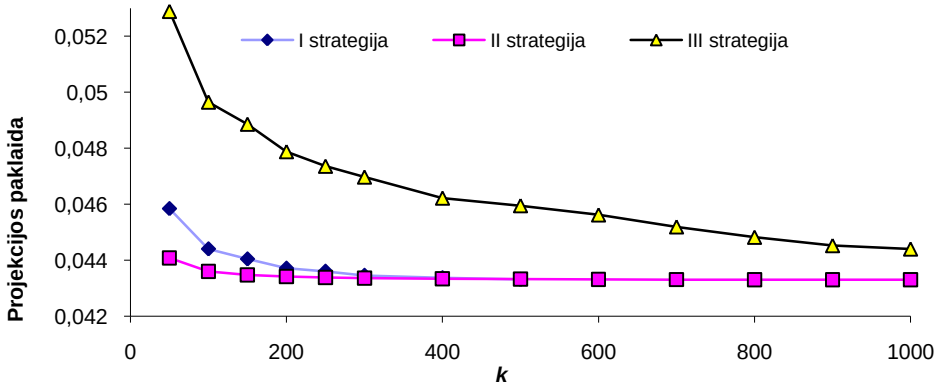
Atliekant eksperimentus su skirtingomis duomenų aibėmis nustatyta, kad projekcijos paklaidai didelę įtaką daro pradinės daugiamačių duomenų aibės suformavimas, t. y. analizuojamų vektorių pradinis surikiavimas. Atsižvelgiant į šį faktą, buvo atliekami tyrimai, kai pradinė duomenų aibė buvo formuojama naudojant tris skirtingas daugiamačių vektorių rikiavimo strategijas:

I. Analizuojamos aibės daugiamačiai vektoriai vieną kartą atsitiktinai išmaišomi algoritmo vykdymo pradžioje.

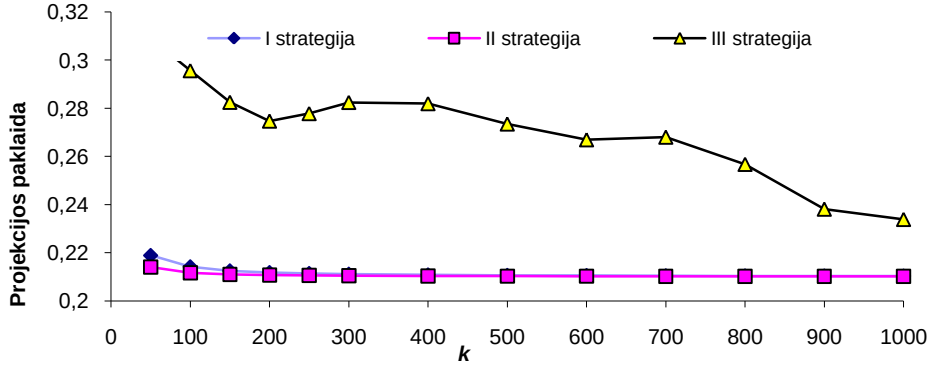
II. Analizuojamos aibės daugiamačiai vektoriai ir juos atitinkantys dvimatės plokštumos vektoriai, kurių koordinatės buvo apskaičiuotos ankstesnėse iteracijose, atsitiktinai perrikiuojami prieš kiekvieną atliekamą iteraciją.

III. Naudojant pagrindinių komponenčių analizės metodą daugiamačiai duomenys pradžioje projektuojami į tiesę. Tokiu būdu yra įvertinamas vektorių panašumas, ir daugiamačiai vektoriai surikiuojami atsižvelgiant į jų išdėstymą tiesėje (artimesni taškai turi gretimus eilės numerius).

Naudojant I ir II daugiamačių vektorių rikiavimo strategijas, buvo atlikta po 50 eksperimentų su skirtingomis parametro k reikšmėmis. Čia parametro k reikšmės keičiamos nuo 50 iki 1000, vizualizuojami daugiamačiai duomenys, apskaičiuojamas projekcijos paklaidos vidurkis, standartinis nuokrypis bei skaičiavimo laikas. 3.2.1 skyriuje eksperimentai parodė, kad projekcijos paklaida po 300 iteracijų nusistovi ir kinta labai nežymiai, todėl atliekant šį tyrimą buvo atliekama tik po 300 iteracijų. Pradinės vektorių koordinatės buvo parenkamos pagal pagrindinių komponenčių analizės metodą. Tiriant parametro k įtaką buvo naudojama I daugiamačių vektorių rikiavimo strategija (3.2 pav. ir 3.3 pav.). Nustatyta, kad parametrą k keičiant nuo 300 iki 1000, projekcijos paklaidos kitimą galima apibrėžti formule $E_{DS} = -0{,}0002\ln(k) + C$, čia C – konstanta. Šiuo atveju tai reiškia, kad taip keičiant k projekcijos paklaida ($E_{DS} = \sqrt{E_{norm}}$) sumažės apytiksliai tik iki 0,08 %.



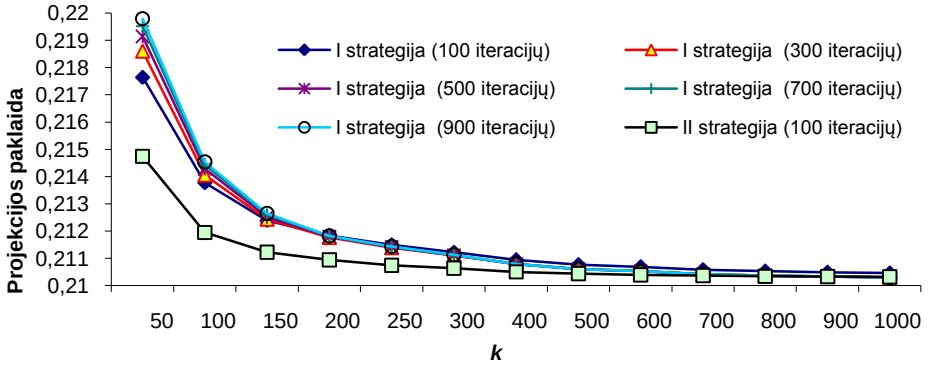
3.4 pav. Projekcijos paklaidos priklausomybė nuo kaimyniškumo parametro k („Abalone“ duomenų aibė), naudojant skirtingas daugiamačių vektorių rikiavimo strategijas



3.5 pav. Projekcijos paklaidos priklausomybė nuo kaimyniškumo parametro k („Ellipsoidal“ duomenų aibė), naudojant skirtingas daugiamačių vektorių rikiavimo strategijas

Atlikti eksperimentai rodo, kad prasčiausi vizualizavimo su DMA rezultatai gaunami naudojant III duomenų rikiavimo strategiją (3.4 ir 3.5 pav.). Tai reiškia, kad kaimynų parinkimas pagal jų padėtį ant tiesės, kurios krypties vektorius sutampa su tikrinių vektoriumi atitinkančiu didžiausią PKA tikrinę reikšmę, yra netinkamas. Kai atsižvelgiama tik į artimiausius kaimynus, projekcijos paklaida konverguoja lėtai ir pasiekimas lokalus minimumas yra daugiau negu 5 % didesnis už gaunamą naudojant I ir II strategijas. Tad, daugiamačių taškų vizualizavimui naudojant DMA algoritmą, reikia, kad analizuojamoje aibėje

šalia būtų ir artimų, ir nutolusių taškų, kadangi į juos atsižvelgiama skaičiuojant vektorių, priklausančių dvimatei plokštumai, koordinates. Atsitiktinis vektorių perrikiavimas prieš kiekvieną iteraciją reiškia, kad skaičiuojant dvimatės plokštumos vektorių koordinates atsižvelgiama į daugiau ir įvairesnių analizuojamos aibės taškų. Dėl šios priežasties, naudojant II duomenų rikiavimo strategiją, gaunama tikslesnė projekcijos paklaida ir su mažiausiu iteracijų kiekiu (pakanka 100). Rezultatai pateikti 3.6 paveiksle.



3.6 pav. Projekcijos paklaidos priklausomybė nuo kaimyniškumo parametro k („Ellipsoidal“ duomenų aibė). Naudojamos I ir II daugiamačių vektorių rikiavimo strategijos bei skirtingas iteracijų skaičius

Lyginant visas tris rikiavimo strategijas, geriausi rezultatai gaunami naudojant II strategiją. Naudojant II-ąją daugiamačių duomenų rikiavimo strategiją, jau po 100 iteracijų gaunamas pakankamai tikslus rezultatas ir, didinant parametrą k, projekcijos paklaidos kitimas tampa labai nežymus (3.4, 3.5 ir 3.6 pav.).

Tuo pačiu eksperimentiškai buvo palygintos SMACOF ir DMA algoritmų projekcijos paklaidos ir skaičiavimo laikai. Lygimui naudotos dvi geriausios k parametro parinkimo strategijos. Gauti rezultatai pateikti 3.3 lentelėje.

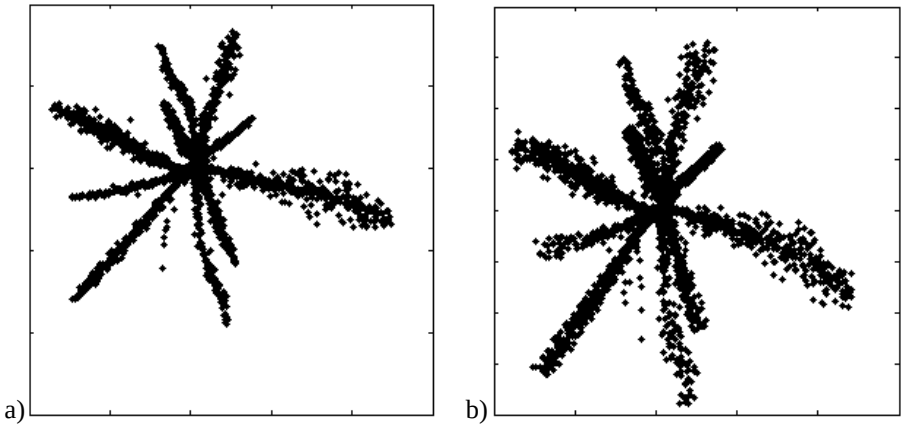
Atlikus eksperimentus su keturiomis skirtingomis duomenų aibėmis nustatyta, kad projekcijos paklaida, naudojant SMACOF algoritmą, yra nežymiai mažesnė nei naudojant DMA algoritmą (abiejuose algoritmuose pradinės projekcijos vektorių koordinatės parenkamos PKA metodu). Skirtumas tarp projekcijos paklaidų yra nedidelis („Abalone“, „Gaussian“ ir „Ellipsoidal“ duomenų aibėms $\leq 1\,\%$, o „Paraboloid“ duomenų aibei $\leq 4\,\%$), tačiau skaičiavimo laikas žymiai ilgesnis, t. y. kuo didesnė duomenų aibė, tuo ryškesnis skirtumas.

3.3 lentelė. Projekcijos paklaida, projekcijos paklaidos standartinis nuokrypis ir skaičiavimo laikas, gautas naudojant SMACOF ir DMA algoritmus

Aibė	SMACOF (100 iter.)		DMA (I strategija, 300 iter., k=400)			DMA (II strategija, 100 iter., k=400)		
	E_{DS}	t, s	$E(E_{DS})$	$\sigma^2(E_{DS})$	$E(t)$, s	$E(E_{DS})$	$\sigma^2(E_{DS})$	$E(t)$, s
„Abalone“	0,043497	151,82	0,043493	0,000065	29,10	0,043726	0,000015	21,96
„Paraboloid“	0,208653	22,40	0,217755	0,004121	16,98	0,214324	0,002012	9,74
„Ellipsoidal“	0,210109	105,27	0,210794	0,000145	23,44	0,210501	0,000059	15,81
„Gaussian“	0,283866	109,28	0,285405	0,000290	28,60	0,284212	0,000039	19,10

3.7 paveiksle pateiktos „Ellipsoidal“ duomenų projekcijos, gautos SMACOF ir DMA algoritmais. Tiriant šiuos algoritmus buvo atlikta po 100 iteracijų. DMA algoritme naudota II daugiamačių vektorių rikiavimo strategija. Abiem atvejais (3.7 a ir 3.7 b pav.) gauta duomenų struktūra gana aiški. Nors projekcijų paklaidos skiriasi vos 0,35 %, tačiau skirtumas tarp skaičiavimo laikų gana ryškus, projekcija DMA algoritmu gaunama 7 kartus greičiau.

Skirtumas tarp skaičiavimo laikų mažėja, kai yra didinamas analizuojamos aibės vektorių skaičius ir mažinamas parametras k. To priežastis – duomenų paruošimas prieš iteracinį procesą, t. y. duomenų rikiavimas naudojant II strategiją reikalauja atlikti daugiau skaičiavimų.



3.7 pav. „Ellipsoidal“ [3140; 50] duomenų aibės projekcija:
(a) gauta naudojant SMACOF algoritmą: $t = 94{,}22s;\ E_{DS} = 0{,}210109$
(b) gauta naudojant DMA algoritmą: $k = 400; t = 13{,}84s; E_{DS} = 0{,}21085$

Tyrimai parodė, kad vizualizuojant dideles duomenų aibes ir taupant skaičiavimo laiką efektyvu naudoti diagonalinį mažoravimo algoritmą. Tačiau reikia atkreipti dėmesį į kelis faktorius, kurie įtakoja šiuo algoritmu gaunamos projekcijos kokybę: analizuojamos aibės vektorių rikiavimo strategiją ir kaimyniškumo eilės parametro *k* parinkimą.

Atlikti tyrimai leido padaryti šias išvadas:

- Projekcijos paklaida, gauta DMA algoritmu, šiek tiek blogesnė nei gauta SMACOF algoritmu, tačiau parenkant kaimyniškumo eilės parametrą $k \geq 400$ arba $k \approx \frac{s}{10}$ (tirtoms aibėms) ir naudojant I ar II kaimynų parinkimo strategijas, ši paklaida gaunama pakankamai artima SMACOF algoritmo paklaidai, nes skirtumas neviršija 5 %.
- Pasiūlytos kelios vektorių rikiavimo strategijos, kurias naudojant gaunamos mažesnės projekcijos paklaidos. Naudojant šias strategijas reikia atsižvelgti į mažesnį kaimynų skaičių (naudoti mažesnį kaimyniškumo parametrą k). Esant $k \approx \frac{s}{10}$, galima iki trijų kartų sutaupyti skaičiavimo laiko.
- Skaičiuojant vektoriaus projekciją DMA algoritmu, kiekvienos iteracijos metu atsižvelgiama tik į k jau suprojektuotų vektorių. Tikslinga analizuojamos aibės daugiamačius vektorius perrikiuoti prieš kiekvieną iteraciją, tokiu atveju skaičiuojant duomenų projekciją atsižvelgiama į beveik visus daugiamačius aibės taškus.
- Eksperimentiškai nustatyta, kad kai k parametras labai mažas, didinant iteracijų skaičių projekcijos paklaida ne mažėja, o atvirkščiai didėja.

3.3. Santykinių daugiamačių skalių algoritmo tyrimas

Santykinis daugiamačių skalių algoritmas priklauso nuo daugelio veiksnių, tokių kaip bazinių vektorių parinkimo strategijos, vektorių iniciacijos dvimatėje plokštumoje būdo, bazinių vektorių skaičiaus. Šio algoritmo pradinis tyrimas buvo atliktas ir aprašytas kartu su bendraautoriais straipsnyje (Bernatavičienė *et al.* 2007), bei disertacijoje (Bernatavičienė 2008).

Atlikti tyrimai leido padaryti šias išvadas:

1. Vizualizavimo rezultatai labai priklauso nuo bazinių vektorių parinkimo strategijos. Nustatyta, kad kuo tolygiau baziniai vektoriai pasiskirstę po visą tiriamą aibę, tuo tikslesnė projekcija yra gaunama.

2. Naujų vektorių pradinių koordinačių parinkimo būdas taip pat daro įtaką vizualizavimo rezultatams. Buvo pasiūlyti ir ištirti 6 skirtingi vektorių parinkimo dvimatėje plokštumoje būdai (Bernatavičienė *et al.* 2007) Atlikti eksperimentai parodė, kad blogiausias bazinių vektorių parinkimo būdas yra atsitiktinis taškų parinkimas bazinių vektorių projekcijų srityje. Šiuo atveju gaunamas didžiausias projekcijos paklaidos vidurkis ir didžiausia paklaidos dispersija. Naudojant pradinių vektorių parinkimo būdą, paremtą PKA algoritmu, paklaidos vidurkis mažesnis už paklaidų vidurkius, gaunamus kitomis strategijomis, tačiau skirtumai tarp šių vidurkių yra nereikšminiai.

Šioje disertacijoje pristatyti ir ištirti du nauji vektorių koordinačių dvimatėje projekcijos plokštumoje parinkimo būdai: pasirenkamos artimiausios bazinio vektoriaus koordinatės arba didžiausią dispersiją turinčios dvi įvesties vektoriaus koordinatės. Šis tyrimas pateiktas 3.3.1 poskyryje.

Disertacijoje (Bernatavičienė 2008) teigiama, kad kuo didesnė vizualizuojamų duomenų aibė, tuo didesnį bazinių vektorių skaičių reikia imti. Didinant bazinių vektorių skaičių gaunama tikslesnė projekcija. Tyrimai parodė, kad mažesnėms duomenų aibėms (iki 3000 vektorių) tikslinga naudoti nuo 700 iki 1000 bazinių vektorių, o didelėms duomenų aibėms – nuo 900 iki 1500.

Daugeliu atvejų šis teiginys nėra visiškai teisingas, nes ne visuomet didinant bazinių vektorių skaičių paklaida mažėja. Galimi ir tokie atvejai, kuomet paklaida didinant bazinių vektorių skaičių didėja, o tinkamai parinktas bazinių vektorių skaičius garantuoja mažesnę projekcijos SDS algoritmu paklaidą už geriausią paklaidą gautą SMACOF algoritmu. Buvo atliktas tyrimas siekiant nustatyti tikslesnį bazinių vektorių skaičiaus parinkimo būdą. Tai plačiau aprašyta 3.3.2 poskyryje.

3.3.1. Bazinių vektorių išrinkimas

Kaip minėta anksčiau, vizualizavimo rezultatai, gauti daugiamačius duomenis vizualizuojant SDS algoritmu, labai priklauso nuo bazinių vektorių parinkimo strategijos. Baziniai vektoriai buvo parenkami naudojant dvi strategijas:

I. Bazinių vektorių aibė F sudaryta iš vektorių, atsitiktinai išrinktų iš analizuojamos aibės $X_i, i = \overline{1,m}$ (Bernatavičienė *et al.* 2007).

II. Pradiniai duomenys klasterizuojami k-vidurkių metodu. Baziniais vektoriais pasirenkami klasterių centrams artimiausi duomenų aibės taškai ir nustatomas fiksuotas taškų skaičius iš kiekvieno klasterio (Naud A. 2004; Naud A. 2006).

Atlikti tyrimai parodė, kad kuo tolygiau baziniai vektoriai pasiskirstę po visą tiriamą aibę, tuo tikslesnė projekcija yra gaunama. Tačiau naudojant

(Bernatavičienė 2008) disertacijoje pasiūlytas strategijas, vizualizavimo rezultatus įtakoja atsitiktinumo faktorius (atsitiktinai parenkami baziniai vektoriai, pradiniai klasterių centrai taip pat parenkami atsitiktinai, naudojant *k*-vidurkių metodą), todėl kiekvieną kartą gaunama vis kita projekcija. Siekiant išvengti atsitiktinumo ir gauti vienareikšmę projekciją, siūlomos dvi naujos bazinių vektorių parinkimo strategijos:

III. Naudojant PKA metodą daugiamačiai duomenys projektuojami į tiesę, ir žingsniu $[s/s_F]$ išrenkamas nustatytas bazinių vektorių skaičius.

IV. Apskaičiuojamos duomenų aibės X vektorių X_i visų n komponentčių dispersijos. Gautos dispersijos palyginamos tarpusavyje ir išrenkama didžiausią dispersiją turinti vektorių komponentė. Vektoriai surikiuojami šios komponentės didėjimo tvarka, ir žingsniu $[s/s_F]$ išrenkamas nustatytas bazinių vektorių skaičius.

III ir IV strategijos atitinka taškų statmeną projekciją į tiesę, kurios kryptį nusako tiesės krypties vektorius. Nors krypties vektoriai skiriasi, tačiau jeigu turima taškų aibė nėra ant sferos ar sukinio, tuomet taškai, suprojektuoti ant tiesės, sudaro identiškai sutvarkytą aibę (atstumai taip taškų ant tiesės gali skirtis). Naudojant šias strategijas buvo atlikti tyrimai su penkiomis skirtingomis testinėmis duomenų aibėmis, gauti rezultatai palyginti (3.4 lentelė).

Atlikus pradinius tyrimus nustatyta, kad blogiausia SDS projekcijos paklaida gaunama, kai bazinių vektorių koordinatės ir patys vektoriai parenkami atsitiktinai. Dėl šios priežasties dvimatės plokštumos vektorių koordinatės, parenkant bazinius vektorius pagal I ir IV strategijas, buvo parenkami naudojant didžiausių dispersijų metodą. Naudojant III bazinių vektorių parinkimo strategiją, baziniai vektorių koordinatės parinktos remiantis PKA algoritmu (plačiau vektorių priklausančių R^2 pradinių koordinačių parinkimo būdai aprašyti 3.4 skyriuje). Gauti rezultatai rodo, kad didelėms duomenų aibėms tinkamiausias bazinių vektorių parinkimo būdas yra naudoti III strategiją, paremtą PKA algoritmu. Tai iliustruoja ir rezultatai, pateikti 3.4 lentelėje. Šiuo atveju gaunama vienareikšmė paklaida, t.y. gaunama nepriklausoma nuo atsitiktinių faktorių (tokių kaip bazinių vektorių koordinačių parinkimas) projekcija, kurios paklaida visuomet įgyja tą pačią reikšmę.

Bazinius vektorius parenkant pagal I strategiją, o pradines vektorių koordinates dvimatėje plokštumoje parenkat remiantis didžiausių dispersijų metodu, galima gauti mažesnę paklaidą. Bet taip pat maža paklaida gaunama kai vektorių koordinatės parenkamos pagal PKA, o patys baziniai vektoriai pagal III strategiją. Antruoju atveju paklaida gaunama vienareikšmiškai. Norint nustatyti priežastis, nuo kurių priklauso vienoms domenų aibėms gaunama paklaida.

3.4 lentelė. Projekcijos paklaidos, gautos naudojant santykinių daugiamačių skalių algoritmą ir keičiant bazinių vektorių parinkimo strategijas

Aibė	Did. dispersijų + I strategija	PKA + III strategija	Did. dispersijų + IV strategija
	Paklaida po 1000 iteracijų, 800 baz. vektorių		
„Paraboloid“ [2573x3]	**0,213618**	0,226991	0,227451
„Gaussian“ [2729x10]	0,285396	**0,280194**	0,289072
	Atlikta 1000 iteracijų, 1500 baz. vektorių		
„Ellipsoidal“ [3140x50]	0,207102	**0,206759**	0,208443
„Abalone“ [4177x7]	**0.012776**	0,013226	0,013479
„Satimage“ [6435x36]	0,109498	**0,095214**	0,119422

Tyrimai parodė, jei aibė naujų taškų aibė nuolat papildoma naujais elementais, tuomet SDS algoritmo paklaida po tam tikro atidėtų taškų skaičiaus pradeda didėti. Todėl reikia performuoti aibę bazinių taškų aibę, pridedant į ją daugiau taškų, ir atlikti visus skaičiavimus iš naujo, arba galima po keleto žingsnių jau atvaizduotus naujus taškus priskirti baziniams taškams ir toliau tęsti naujų taškų atidėjimą.

3.3.2. Bazinių vektorių skaičius

Tiriant santykinių daugiamačių skalių algoritmo vizualizavimo rezultatų priklausomybę nuo daugiamačių vektorių iniciacijos strategijos, buvo pastebėta, kad didinant bazinių vektorių skaičių projekcijos paklaida taip pat ima didėti (3.5 lentelė).

Siekiant surasti tokį bazinių vektorių parinkimą, su kuriuo paklaida yra mažiausia arba kaip įmanoma artima mažiausiai, reikia atlikti daug eksperimentų su skirtingais bazinių vektorių rinkiniais. 3.5 lentelėje pateiktos atsitiktiniu bazinių vektorių parinkimu gautos vidutinės projekcijos paklaidos, atlikus po 128 eksperimentus su kiekviena duomenų aibe. Skaičiavimai buvo atliekami kompiuterių klasteryje (aprašyta 2.2 skyriaus 2 punkte), naudojant 16 procesorių ir skaičiavimus kartojant 8 kartus. Kiekviename klasterio procesoriuje buvo vykdomas tas pats algoritmas su skirtingais baziniais vektoriais.

Šio eksperimento tikslas – ištirti, kaip kinta projekcijos paklaida didinant bazinių vektorių skaičiui. Su 3.5 lentelėje nurodytu bazinių vektorių skaičiumi ir pasirinktu bazinių vektorių iniciacijos metodu SDS algoritmas atlikdavo po 600 iteracijų. Nebazinių vektorių pradinės koordinates parenkamos imant jas lygias artimiausio bazinio vektoriaus koordinatėms dvimatėje plokštumoje.

Kaip rodo gauti rezultatai, pateikti 3.5 ir 3.6 lentelėse, kai bazinių vektorių skaičius didesnis už 800, projekcijos paklaida tampa nestabili, o daugeliu atvejų ima didėti. Šis faktas būdingas naudojant skirtingus bazinių vektorių iniciacijos būdus, o tai reiškia, kad pradinių vektorių koordinačių parinkimo metodas projekcijos paklaidos nestabilumui įtakos neturi. Taigi, parenkant bazinių vektorių skaičių egzistuoja riba, nuo kurios toliau didinant bazinių vektorių skaičių paklaida daugeliu atvejų pradeda didėti.

3.5 lentelė. „Abalone“ projekcijos paklaidos, gautos naudojant santykinių daugiamačių skalių algoritmą, keičiant bazinių vektorių skaičių ir iniciacijos metodą

Bazinių vektorių skaičius	Paklaida, kai vektorių pradinių koordinačių parinkimo metodas yra			
	Atsitiktinis	Ant tiesės	Didžiausia dispersiją	PKA
100	0,017593	0,017339	0,013502	0,013610
200	**0,017105**	0,018041	0,013050	0,013320
400	0,017152	0,018341	0,012894	0,012802
800	0,017229	**0,017010**	0,012800	0,012661
1600	0,01800	0,019108	0,012808	0,012653
2000	0,018194	0,017575	**0,012768**	0,012644
2500	0,018448	0,019545	0,012842	0,012595
3200	0,0184875	0,018444	0,012801	0,012576
4000	0,0189052	0,019721	0,012820	**0,012555**

3.6 lentelė. „Gaussian“ projekcijos paklaidos, gautos naudojant santykinių daugiamačių skalių algoritmą, keičiant bazinių vektorių skaičių ir iniciacijos metodą

Bazinių vektorių skaičius	Paklaida, kai vektorių pradinių koordinačių parinkimo metodas yra			
	Atsitiktinis	Ant tiesės	Didžiausia dispersiją	PKA
100	0,292790	0,293559	0,287760	0,287443
200	0,284600	0,284006	0,281275	0,280405
400	0,281672	0,279928	0,280249	0,276137
800	**0,278576**	**0,276970**	0,275353	0,274023
1600	0,279848	0,279064	0,274999	0,273112
2000	0,278780	0,278403	0,273906	0,272931
2500	0,279734	0,279672	**0,273807**	**0,272757**

3.4. Pradinių vektorių koordinačių parinkimo problema

Ištyrus disertacijoje pateiktus algoritmus nustatyta, kad pradinių vektorių koordinačių dvimatėje plokštumoje parinkimas labai įtakoja vizualizavimo rezultatus. Todėl šio eksperimento tikslas – nustatyti, kuris koordinačių parinkimo būdas yra tinkamiausias, naudojant daugiamačių skalių algoritmus. Pirmiausia eksperimentai buvo atlikti naudojant mažas duomenų aibes. Eksperimentuose naudoti SMACOF, SDS ir Sammono algoritmai.

Projekcijos paklaidos skaičiuotos naudojant šiuos pradinių duomenų komponenčių parinkimo būdus:

1. Dvimatėje plokštumoje vektoriai parenkami atsitiktinai. Eksperimentai atliekami 10 kartų, apskaičiuojamas gautų projekcijos paklaidų vidurkis.
2. Pirmoji pradinio vektoriaus dvimatėje plokštumoje koordinatė parenkama atsitiktinai, o antroji taip, kad gautas vektorius Y_i būtų lygiagretus pasirinkai tiesei.
3. Dvimatėje plokštumoje vektoriai apskaičiuojami naudojant pagrindinių komponenčių analizės algoritmą.
4. Apskaičiuojamos duomenų aibės X vektorių visų komponenčių dispersijos. Gautos dispersijos palyginamos tarpusavyje ir išrenkamos dvi didžiausias dispersijas turinčios vektorių komponentės. Šias komponentes atitinkančios vektorių koordinatės laikomos vektoriaus, esančio dvimatėje plokštumoje, pradinėmis koordinatėmis. Šį pradinių koordinačių parinkimo metodą vadinsime didžiausių dispersijų metodu.

Eksperimentų rezultatai, pateikti 3.7 ir 3.8 lentelėse parodė, kad mažiausia projekcijos paklaida gaunama projekcijos vektorių koordinates parenkant naudojantis PKA arba didžiausių dispersijų metodais, tačiau PKA metodas reikalauja didesnių skaičiavimo resursų. PKA algoritmo sudėtingumas yra $O(s^3)$, bet jeigu skaičiuojame tik d tikrinių reikšmių tuomet efektyviausio PKA algoritmo (pvz. EM algoritmas, *angl. expectation-maximization algorithm*) sudėtingumas lygus $O(dsn)$ (Roveis 1998). Tuo tarpu koordinates parenkant pagal didžiausias dispersijas sudėtingumas lygus $O(sn)$. Todėl, kaip pakankamai gera alternatyva PKA, kitiems tyrimams pradinių vektorių koordinačių parinkimui naudojamas didžiausių dispersijų metodas.

3.7 lentelė. Sammono algoritmo projekcijos paklaidos

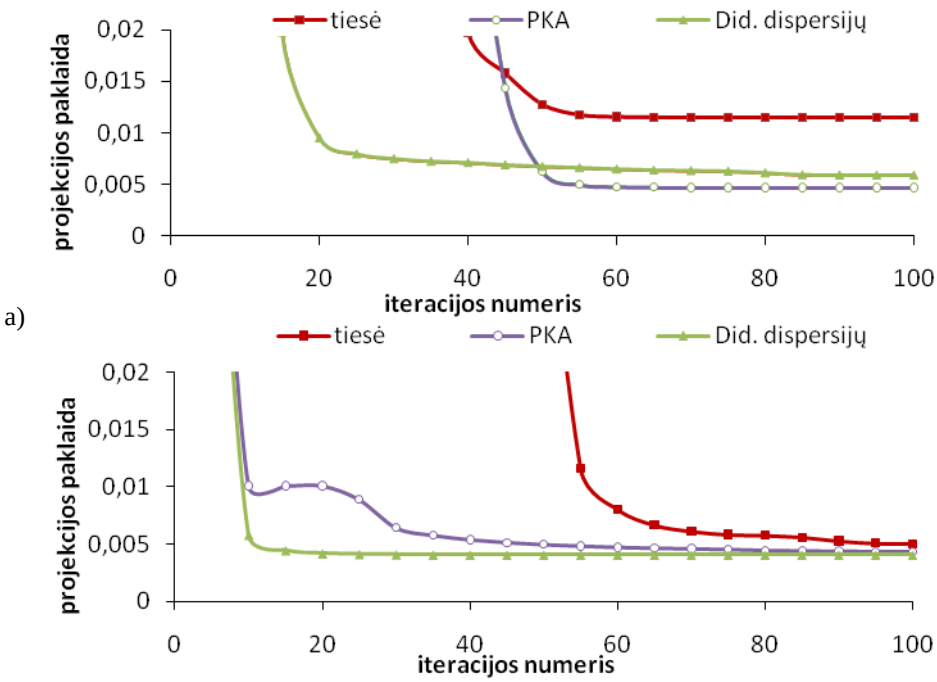
Aibė	Paklaida, kai vektorių pradinių koordinačių parinkimo metodas			
	Atsitiktinis	Tiesėje	PKA	Did. dispersija
„HBK“	0,006483	0,01140	**0,00464**	0,00555
„Wood“	0,025269	0,02536	**0,02432**	0,02537
„Irisų“	0,004997	0,00491	**0,00397**	0,00406
„Vyno“	0,000140	0,00012	**0,00003**	**0,00003**
„Klasteriai“	0,071625	0,07103	0,07115	**0,06667**

3.8 lentelė. SMACOF algoritmo projekcijos paklaidos

Aibė	Paklaida, kai vektorių pradinių koordinačių parinkimo metodas			
	Atsitiktinis	Tiesėje	PKA	Did. dispersija
„HBK“	**0,003497**	0,00431	0,0044563	0,0044563
„Wood“	0,004883	**0,004281**	0,0044773	0,0044773
„Irisų“	0,006914	0,004082	**0,0038487**	0,0040419
„Vyno“	0,009474	0,003755	0,0037548	**0,0024549**
„Klasteriai“	0,334236	0,37769	**0,3164181**	**0,3164181**

Paklaidos priklausomybė nuo pradinių taškų koordinačių parinkimo metodo ir iteracijų skaičiaus yra pateikta 3.8 paveiksle. Rezultatai rodo, kad PKA ir didžiausių dispersijų iniciacijos metodai yra geresni paklaidos prasme už pradinių taškų iniciaciją ant tiesės ir atsitiktinai.

Gautų rezultatų patikrinimui buvo atlikti analogiški eksperimentai naudojant didesnes duomenų aibes ir kitus daugiamačių skalių metodus (SDS ir DMA).



3.8 pav. Projekcijos paklaidos priklausomybė nuo iniciacijos metodo: a) „HBK“ duomenų aibė; b) Irisų duomenų aibė

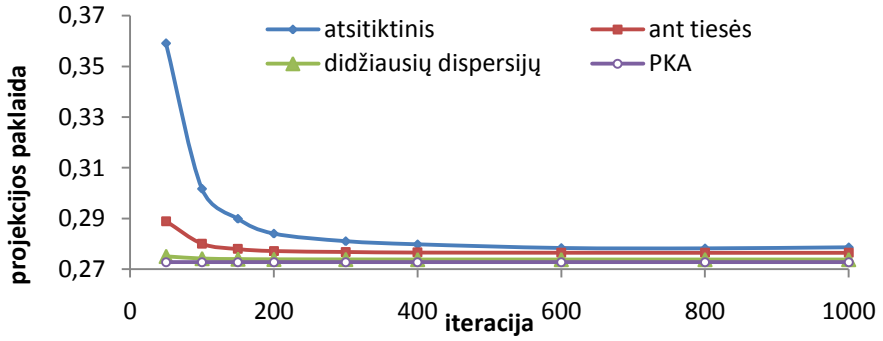
3.9 Lentelėje pateikti rezultatai gauti DS SMACOF algoritmu. Eksperimentai buvo atliekami esant tokioms sąlygoms:

- Naudojant atsitiktinį taškų pradinių koordinačių parinkimo būdą su kiekviena duomenų aibe buvo atlikta po 256 eksperimentus ir apskaičiuotas gautų paklaidų vidurkis.
- Pradines taškų koordinatės parenkant ant tiesės didžiausių dispersijų metodu ir pagal pagrindinių komponenčių analizę, eksperimentai buvo kartojami 32 kartus, kadangi visais atvejais gaunama beveik vienareikšmė paklaida. Paklaida gaunama nevienoda vien tik dėl skaičių apvalinimo paklaidų.
- Kiekvienas algoritmas atliko po 200 iteracijų.

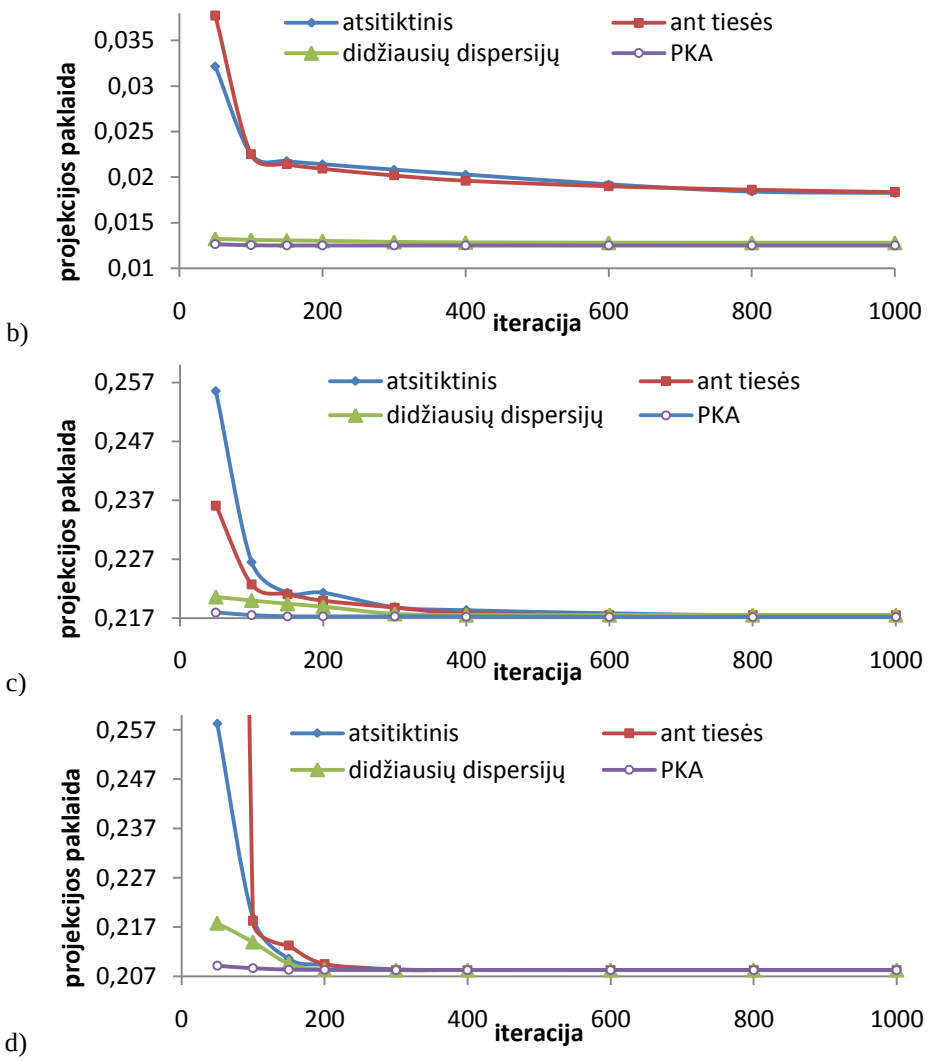
3.9 lentelė. Projekcijos paklaidos ir skaičiavimo laikai, gauti daugiamačių skalių SMACOF algoritmu

Koordinačių parinkimo būdas	Paklaida, laikas	„Abalone“	„Paraboloid“	„Gaussian“	„Sferos“
Atsitiktinis	$E\left(\sqrt{E_{norm}}\right)$	0,013019	0,209435	0,284020	0,219793
	$E(t)$, s	233,81	**78,46**	90,80	25,20
Ant tiesės	$\sqrt{E_{norm}}$	0,020931	0,209510	0,277183	0,219941
	$E(t)$, s	**233,53**	78,50	**90,64**	25,20
Didžiausių dispersijų	$\sqrt{E_{norm}}$	0,013019	0,208405	0,273857	0,218949
	$E(t)$, s	233,81	78,52	90,87	**25,12**
PKA	$\sqrt{E_{norm}}$	**0,012513**	**0,208306**	**0,272727**	**0,217274**
	$E(t)$, s	234,48	79,17	91,80	25,50375

Geriausias rezultatas gaunamas naudojant pagrindinių komponenčių analizės iniciacijos būdą, o sugaištamas laikas beveik nepriklauso nuo pradinių taškų koordinačių parinkimo būdo.



a)



3.9 pav. Projekcijos paklaidos priklausomybė nuo vektorių pradinių koordinačių parinkimo metodo, naudojant SMACOF algoritmą: a) „Gaussian“; b) „Abalone“; c) „Sferos“; d) „Paraboloid“ duomenų aibėms.

3.9 paveiksle pateikta projekcijos paklaidos priklausomybė nuo pasirinkto pradinių vektorių koordinačių parinkimo būdo. Pateikti rezultatai rodo, kad tiksliausia projekcijos paklaida gaunama naudojant didžiausių dispersijų iniciacijos ir PKA būdus. Jau atlikus 50 iteracijų gaunama pakankamai tiksli

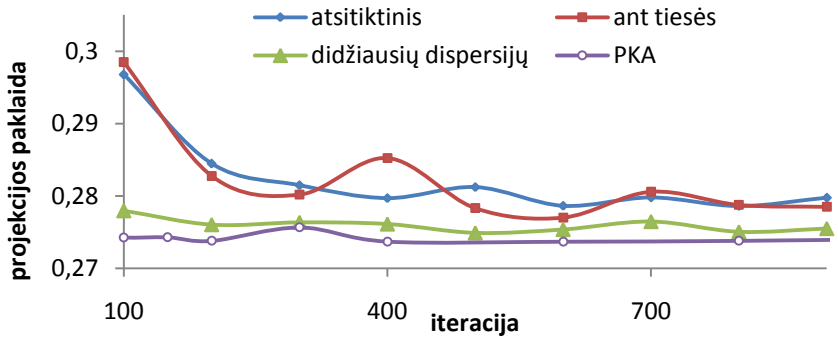
projekcijos paklaida, o didinant iteracijų skaičių, projekcijos paklaida kinta nežymiai (1-3 %).

Atliekant eksperimentus su santykiniu daugiamačių skalių algoritmu buvo naudojami šie parametrai: 800 bazinių vektorių, 600 iteracijų jų projekcijai apskaičiuoti SMACOF algoritmu. Atlikta po 128 eksperimentus naudojant atsitiktinį bazinių pradinių vektorių iniciacijos būdą bei po 16 eksperimentų naudojant pradinių vektorių parinkimą ant tiesės pagal didžiausias dispersijas ir pagrindinių komponenčių analizę. Likusių aibės taškų iniciacijai imamos artimiausio bazinio taško koordinatės dvimatėje plokštumoje. Apskaičiuoti gautų paklaidų ir skaičiavimo laiko vidurkiai.

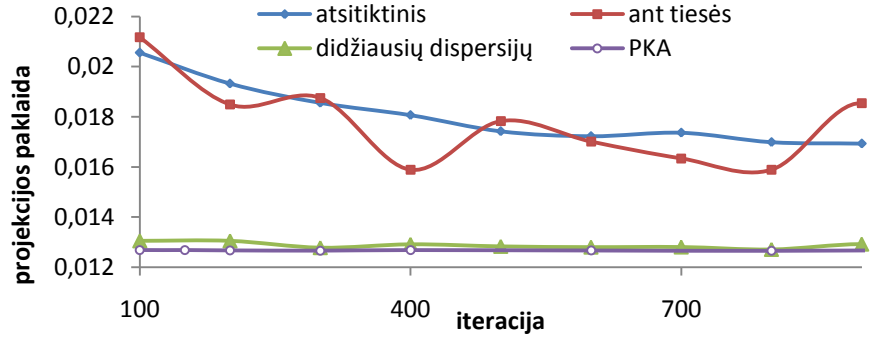
3.10 lentelė. Projekcijos paklaidos ir skaičiavimo laikai gauti santykinių daugiamačių skalių algoritmu

Koordinačių parinkimo būdas	Paklaida, laikas	„Abalone“	„Paraboloid“	„Gaussian“	„Sferos“
Atsitiktinis	$E\left(\sqrt{E_{norm}}\right)$	0,017230	0,211763	0,278649	0,219625
	$E(t)$, s	29,71	24,96	25,98	23,62
Ant tiesės	$\sqrt{E_{norm}}$	0,017010	0,210841	0,274914	0,219839
	$E(t)$, s	29,83	**24,88**	25,95	**23,53**
Didžiausių dispersijų	$\sqrt{E_{norm}}$	0,012800	0,217671	0,274007	0,220005
	$E(t)$, s	29,44	25,02	25,98	23,61
PKA	$\sqrt{E_{norm}}$	**0,012666**	**0,212438**	**0,273695**	**0,218846**
	$E(t)$, s	**29,10**	24,99	**25,79**	23,54

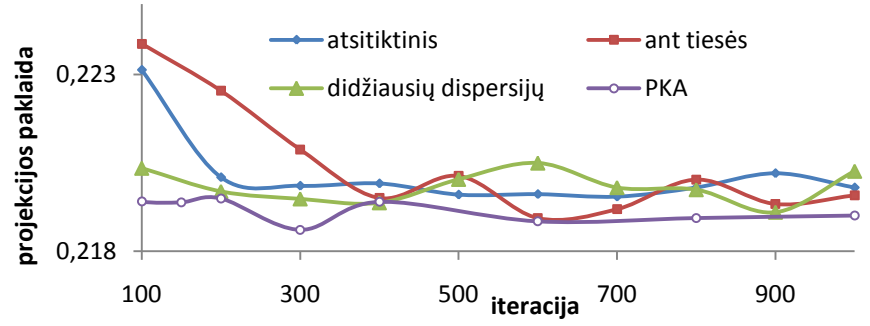
Gauti rezultatai rodo, kad SDS algoritmo atveju geriausi pradinių vektorių iniciacijos būdai – didžiausių dispersijų ir PKA. Naudojant šiuos iniciacijos būdus, jau pirmosiose iteracijose gaunama pakankamai tiksli paklaida (3.10 pav.). Didinant iteracijų skaičių projekcijos paklaidos kitimas nėra toks stabilus, kaip SMACOF algoritmo atveju, kadangi SDS algoritmo rezultatas priklauso nuo bazinių vektorių kiekio ir jų parinkimo būdo.


atsitiktinis
ant tiesės
didžiausių dispersijų
PKA
projekcijos paklaida
0,3
0,29
0,28
0,27
100
400
700
iteracija

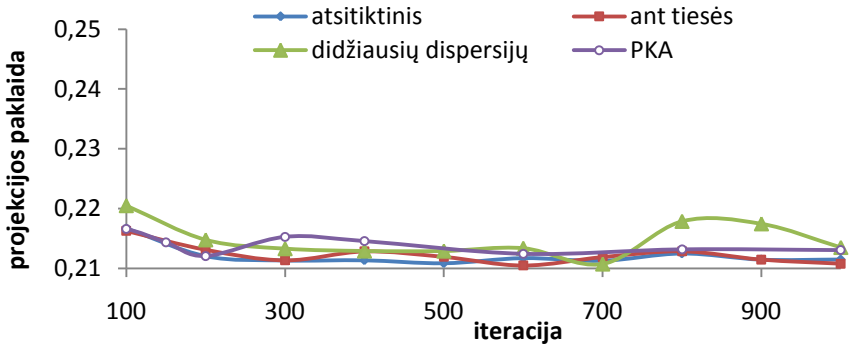
a)


atsitiktinis
ant tiesės
didžiausių dispersijų
PKA
projekcijos paklaida
0,022
0,02
0,018
0,016
0,014
0,012
100
400
700
iteracija

b)


atsitiktinis
ant tiesės
didžiausių dispersijų
PKA
projekcijos paklaida
0,223
0,218
100
300
500
700
900
iteracija

c)



d)

3.10 pav. Projekcijos paklaidos priklausomybė nuo iniciacijos metodo, naudojant santykinių daugiamačių skalių algoritmą:
a) „Gaussian“; b) „Abalone“; c) „Sferos“; d) „Paraboloid“ duomenų aibėms

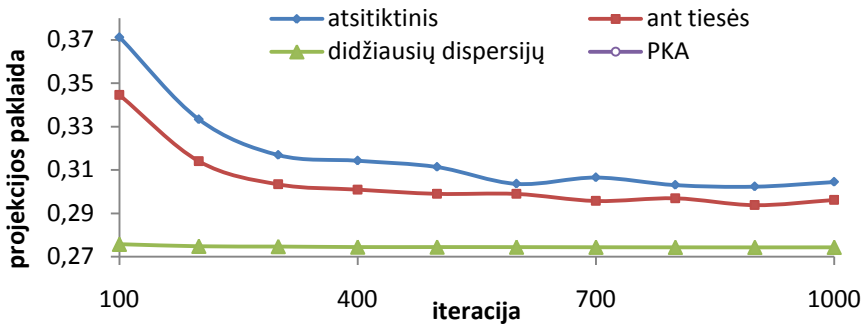
DMA algoritmo tyrimo rezultatai pateikti 3.11 lentelėje. DMA algoritmui buvo parinkta $k = 400$ kaimynų, bei vektorių aibė perrikiuojama prieš kiekvieną iteraciją (3.2.1 skyriuje gauti DMA algoritmo optimalūs parametrai). Su kiekviena duomenų aibe atlikta po 300 iteracijų.

3.11 lentelė. Projekcijos paklaidos ir skaičiavimo laikai gauti diagonaliniu mažoravimo algoritmu

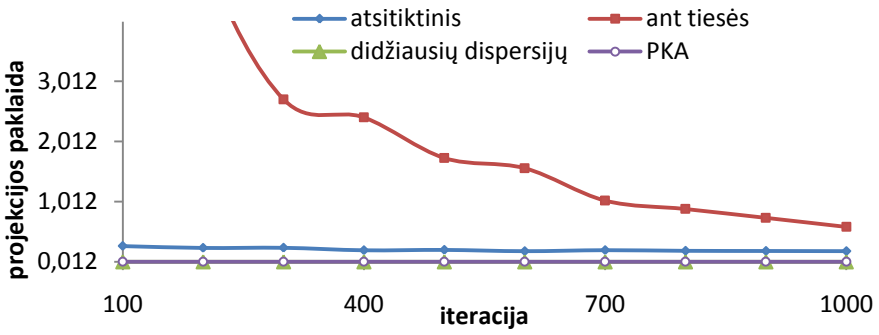
Koordinačių parinkimo būdas	Paklaida, laikas	„Abalone“	„Paraboloid“	„Gaussian“	„Sferos“
Atsitiktinis	$E\left(\sqrt{E_{norm}}\right)$	0,245783	0,267107	0,316895	0,235206
	$E(t)$, s	96,01	54,04	58,79	26,95
Ant tiesės	$\sqrt{E_{norm}}$	2,711916	0,738870	0,303371	0,239015
	$E(t)$, s	95,97	54,02	58,77	26,91
Didžiausių dispersijų	$\sqrt{E_{norm}}$	0,0131146	**0,2134101**	0,2746866	0,220871
	$E(t)$, s	96,04	54,03	58,73	27,01
PKA	$\sqrt{E_{norm}}$	**0,012612**	0,217371	**0,273613**	**0,219985**
	$E(t)$, s	**92,46**	**52,02**	**56,3**	**26,00**

Atliekant eksperimentus su diagonaliniu mažoravimo algoritmu nustatyta, kad šis algoritmas labiausiai priklauso nuo vektorių iniciacijos būdo ir blogai parinkus pradinę vektorių projekciją dvimatėje plokštumoje labai sulėtėja paklaidos konvergavimo greitis. Iš 3.11 lentelės matyti, kad dvimatėje plokštumoje pradinių vektorių parinkimas ant tiesės DMA algoritmui yra visiškai netinkamas. Tai ypač pastebima naudojant „Abalone“ duomenų aibę. Tinkamiausias pasirodė pradinių vektorių parinkimo pagal PKA iniciacijos būdas. Pradinių vektorių didžiausių dispersijų iniciacijos būdas duoda stabilią, ir artima PKA iniciacijos būdui, projekcijos paklaidą bei didėjant iteracijų skaičiui jos pokytis yra nereikšminis.

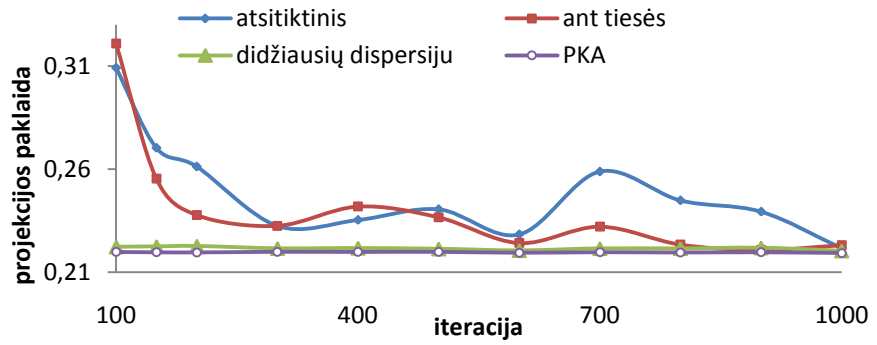
a)



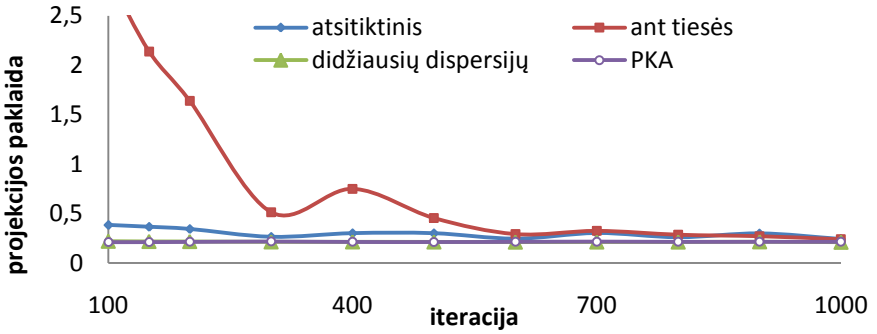
b)



c)



d)



3.11 pav. Projekcijos paklaidos priklausomybė nuo iniciacijos metodo, naudojant diagonalinio mažoravimo (II vektorių rikiavimo strategija) algoritmą: a) „Gaussian“; b) „Abalone“; c) „Sferos“; d) „Paraboloid“ duomenų aibėms

3.5. DS klasės algoritmų lyginamoji analizė

Ištyrus tris daugiamačių skalių klasės algoritmus: SMACOF, diagonalinį mažoravimo, bei santykinių daugiamačių skalių algoritmus, iškilo būtinybė juos palyginti tarpusavyje, bei nustatyti, kuris iš jų tinkamiausias didelių aibių vizualizavimui. Algoritmai buvo vertinami remiantis kiekybiniais atvaizdavimo kriterijais.

3.12 lentelėje pateikti duomenys gauti, esant tokioms algoritmų pradinėms sąlygoms:

1. SMACOF algoritmas atliko 600 iteracijų, plokštumos vektorių pradinis koordinačių parinkimas atliktas taikant didžiausių dispersijų metodą.
2. SDS algoritme baziniai vektoriai projektuojami daugiamačių skalių SMACOF algoritmu atliekant 600 iteracijų, vektorių pradinis

parinkimas atliktas taikant didžiausių dispersijų metodą ir baziniai vektoriai išrenkami pagal didžiausių dispersijų strategiją. Bazinių vektorių skaičius aibėms iki 4000 vektorių lygus 1000, o didesnėms nei 4000 lygus 1500 bazinių vektorių.

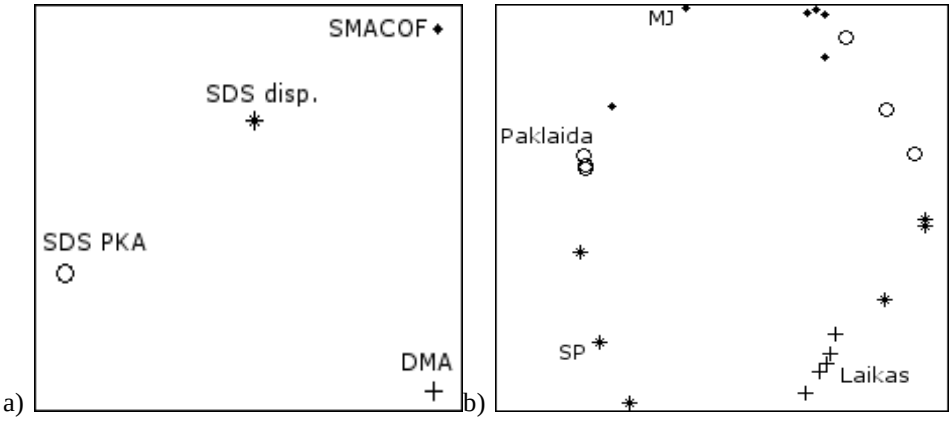
3. SDS algoritme baziniai vektoriai projektuojami DS algoritmu atliekant 600 iteracijų. Vektorių pradinis parinkimas atliktas taikant PKA metodą ir baziniai vektoriai išrenkami pagal PKA strategija. Bazinių vektorių skaičius aibėms iki 4000 vektorių lygus 1000, o didesnėms nei 4000 lygus 1500.
4. DMA algoritmas atlieka 600 iteracijų, prieš kiekvieną iteraciją vektoriai perrikiuojami ir skaičiuojant naudojama $k = 400$ kaimynų. Vektorių koordinatės parenkamos naudojantis didžiausių dispersijų metodu.

3.12 lentelė. DS klasės algoritmais gautų projekcijų kokybės kriterijų vidutinės reikšmės

Kriterijus	SMACOF					
	„Sferos“	„Gaussian“	„Paraboloid“	„Elipsoidal“	„Abalone“	„Satimage“
MJ	**4229,69**	**14844,34**	**115,6487**	184,6303	109,3453	67255,92
SP	**0,861522**	**0,812781**	**0,893843**	0,928474	0,999592	0,980790
Norm. paklaida	**0,217515**	**0,273772**	**0,208293**	**0,207143**	0,012816	0,1165890
Laikas	73,46	268,87	232,7953	360,58	693,79	2717,75
Kriterijus	SDS su iniciacija pagal dispersijas					
MJ	4237,45	14554,89	116,91	**179,42**	108,01	68713,81
SP	0,859215	0,810587	0,885582	**0,928548**	0,999592	0,981856
Norm. paklaida	0,219058	0,274714	0,213209	0,207212	0,012779	0,109482
Laikas	**35,78**	**38,29**	**37,33**	**39,85**	**42,43**	120,77
Kriterijus	SDS su iniciacija pagal PKA					
MJ	4698,30	16594,15	121,04	188,57	108,53	**63134,47**
SP	0,811134	0,753909	0,878545	0,930524	0,999597	**0,985516**
Norm. paklaida	0,272489	0,301998	0,251414	0,205049	0,012656	**0,0952139**
Laikas	36,14	39,04	37,76	40,22	42,74	**95,59**
Kriterijus	DMA					
MJ	4728,84	15310,89	229,78	248,8662	112,1474	266496,6285
SP	0,858383	0,811120	0,888252	0,927513	0,999583	0,915439
Norm. paklaida	0,219763	0,274438	0,212582	0,208014	0,012949	0,204888
Laikas	52,75	116,16	104,76	131,98	189,76	302,61

Tyrimo rezultatai rodo, kad nėra vienareikšmiškai geriausio algoritmo, tinkančio didelių aibių vizualizavimui. Laiko atžvilgiu pats efektyviausias yra santykinių daugiamačių skalių algoritmas, o kitų kriterijų atžvilgiu – SMACOF algoritmas. Neišsiskiriantis nei pagal vieną kriterijų iš kitų algoritmų yra diagonalinio mažoravimo algoritmas.

Siekiant apibendrinti ir palyginti algoritmų rezultatus, pagal 3.12 lentelės rezultatus sukonstruoti keturi vektoriai, atitinkantys skirtingus algoritmus. Šie vektoriai turi 24 koordinates, atitinkančias kriterijų reikšmes skirtingoms duomenų aibėms. Gautieji vektoriai suprojektuoti į dvimatę plokštumą SMACOF algoritmu, siekiant vizualiai įvertinti jų skirtumus (3.12 pav.).



3.12 pav. SMACOF projekcija: a) kriterijų vektorių; b) kriterijų vektorių komponenčių

Projekcijos rezultatai patvirtina (3.12 a pav.), kad tirti algoritmai skiriasi ir skirtumas tarp bet kurių dviejų algoritmų yra beveik vienodas, kadangi beveik vienodi tarpusavio atstumai tarp projekcijos taškų.

Kai normuoti vektoriai yra nepriklausomi (erdvės bazė, Pareto aibė, vienetinės matricos elementų aibė), tuomet projekcijos plokštumoje jie suformuoja taisyklingą iškiląjį daugiakampį (arba kai galima per juos nubrėžti apskritimą). Šiuo atveju visi jie išsidėstę elipsės pavidalo kreivėje ir sugrupuoti grupėmis pagal kriterijų, t. y. kriterijai pagal kuriuos vertinami algoritmai, yra tarpusavyje mažai koreliuoti (3.12 b pav.). Vadinasi, jeigu algoritmas optimizuojamas vienam kriterijui, jis dažnai labai nusileidžia pagal kitus. Ne visi kriterijai vienodai svarbūs, sprendžiant konkrečią problemą, todėl įvedami skirtingi kriterijų svoriai ir taip randamas geriausias sprendinys. Vizualizuojant didelės apimties aibes svarbiausias kriterijus yra skaičiavimo laikas, antroje vietoje projekcijos paklaida. Pagal šiuos kriterijus keturiais iš penkių tirtų atvejų,

geriau pasirodė santykinių daugiamačių skalių algoritmas su pradinių vektorių koordinačių parinkimu pagal didžiausias dispersijas.

3.6. Skyriaus išvados

Apibendrinant gautus rezultatus galima padaryti tokias išvadas:

1. Palyginus rezultatus, gautus naudojant skirtingus DS tipo algoritmus, nustatyta, kad optimalu vektorius dvimatėje plokštumoje parinkti naudojant didžiausių dispersijų metodą. Šis metodas suteikia geras pradines sąlygas paklaidos konvergavimui į lokalų minimumą ir jau po pirmųjų 100 iteracijų gaunama pakankamai tiksli projekcijos paklaida.
2. Paklaidos priklausomybė nuo pradinių taškų iniciacijos metodo ir iteracijų skaičiaus rodo, kad PKA ir didžiausių dispersijų iniciacijos metodai yra žymiai geresni paklaidos prasme už iniciaciją ant tiesės.
3. Algoritmų junginiai (SOM_Sammono ir SOM_SMACOF) yra panašūs, nes abu užtikrina identišką daugiamačių duomenų projekcijos kokybę. Tai leidžia taikyti ne tik daugumos tyrinėtojų naudojamą SOM ir Sammono junginį, bet ir jam panašų SOM ir SMACOF algoritmų junginį, taip sutaupant skaičiavimas reikalingo laiko.
4. Eksperimentiškai parodyta, kad daugiamačių skalių SMACOF realizacijos skaičiavimo laikas yra apie 2 kartus mažesnis už Sammono projekcijos realizacijos skaičiavimo laiką, esant pakankamai dideliems abiejų algoritmų iteracijų skaičiams.
5. Tyrimai parodė, kad vizualizuojant dideles duomenų aibes ir taupant skaičiavimo laiką, efektyvu naudoti diagonalinį mažoravimo algoritmą. Tačiau reikia atkreipti dėmesį į analizuojamos aibės daugiamačių vektorių rikiavimo strategijos ir kaimyniškumo parametro *k* parinkimą. Ištyrus DMA algoritmo rezultatų priklausomybę nuo daugiamačių vektorių rikiavimo strategijos, gautos mažesnės projekcijos paklaidos atsižvelgiant į mažesnį kaimynų skaičių k. Visa tai leidžia iki trijų kartų sutaupyti skaičiavimo laiko, kai $k \approx \frac{s}{10}$.
6. DMA algoritmo projekcijos paklaida, yra didesnė už SMACOF algoritmo projekcijos paklaidą, tačiau parenkant kaimyniškumo eilės parametrą $k \geq 400$ arba $k \approx \frac{s}{10}$ (tirtoms aibėms) ir naudojant kaimynų parinkimo I ar II strategijas, šių paklaidų skirtumas yra mažesnis už 5 %.

7. Skaičiuojant vektoriaus projekciją DMA algoritmu, kiekvienos iteracijos metu atsižvelgiama tik į k jau suprojektuotų vektorių. Tikslinga analizuojamos aibės daugiamačius vektorius perrikiuoti prieš kiekvieną iteraciją, tokiu atveju skaičiuojant duomenų projekciją atsižvelgiama į beveik visus daugiamačius aibės taškus.
8. Egzistuoja bazinių vektorių riba nuo kurios projekcijos paklaida, gauta santykinių daugiamačių skalių algoritmu, pradeda didėti.
9. Bazinių vektorių parinkimui naudojant PKA ir didžiausių dispersijų metodus, gaunama vienareikšmiška projekcija.
10. Santykinių daugiamačių skalių algoritmas su pradinių vektorių iniciacija pagal didžiausias dispersijas leidžia efektyviai ir tiksliai apskaičiuoti didelės apimties duomenų aibių projekcijas.

Išvados

Darbe atlikti tyrimai atskleidė vizualizavimo metodų, grindžiamų daugiamatėmis skalėmis, naujas galimybes.

Teoriniai ir eksperimentiniai tyrimai leido padaryti šias išvadas:

1. Teoriškai įrodyta, kad Sammono projekcijos algoritme pradinių taškų parinkimas ant tiesės, kai jos krypties koeficientas lygus $a = \pm 1$, yra netaikytinas. Teoriškai, naudojant tokią taškų iniciaciją, šie taškai turėtų išlikti ant tos pačios tiesės. Dėl skaičiavimo ir skaičių apvalinimo paklaidų taškai palieka tiesę ir po keleto iteracijų išsibarsto po visą dvimatę projekcijos plokštumą. Tikslinga naudoti tokius iniciacijos būdus, kaip pagrindinių komponenčių analizė ar didžiausių dispersijų metodas. Pagrindinių komponenčių analizės ir didžiausių dispersijų iniciacijos metodai yra žymiai geresni paklaidos prasme už iniciaciją ant tiesės.
2. Palyginus rezultatus, gautus naudojant skirtingus daugiamačių skalių tipo algoritmus, nustatyta, kad optimalu pradinius vektorius dvimatėje plokštumoje parinkti naudojant didžiausių dispersijų metodą. Šis iniciacijos metodas pagreitina paklaidos konvergavimą ir jau po pirmųjų iteracijų gaunama pakankamai artima minimaliai projekcijos paklaida.

3. Tyrimai parodė, kad vizualizuojant dideles duomenų aibes ir taupant skaičiavimo laiką, efektyvu naudoti diagonalinį mažoravimo algoritmą. Tačiau reikia atkreipti dėmesį į analizuojamos aibės daugiamačių vektorių rikiavimo strategijos ir kaimyniškumo eilės parametro k parinkimą. Ištyrus DMA algoritmo rezultatų priklausomybę nuo daugiamačių vektorių rikiavimo strategijos, gautos mažesnės projekcijos paklaidos atsižvelgiant į mažesnį kaimynų skaičių k. Visa tai leidžia iki trijų kartų sutaupyti skaičiavimo laiką, kai $k \approx \frac{s}{10}$.
4. Diagonalinio mažoravimo algoritmo projekcijos paklaida yra didesnė už SMACOF algoritmo projekcijos paklaidą, tačiau, parenkant kaimyniškumo eilės parametrą $k \geq 400$ arba $k \approx \frac{s}{10}$ (tirtoms aibėms) ir naudojant kaimynų perrikiavimo strategijas, kai kaimynai perrikiuojami algoritmo pradžioje arba po kiekvienos iteracijos; šių paklaidų skirtumas yra mažesnis už 5.
5. SOM tinklo permokomų neuronų skaičius laiptiškai mažėja didėjant mokymo epochos eilės numeriui ir sumažėja vienetu po $e' = \left[\frac{n'e}{k'}\right] - \left[\frac{(n'-1)e}{k'}\right]$ $(n' = 1, \dots, k' - 2)$ epochos.
6. Jeigu analizuojamos duomenų aibės vektoriai nėra sunormuoti pagal vektoriaus ilgį, tuomet galima naudoti SOM tinklo ląstelių spalvinimą pilkos spalvos atspalviais, priklausančiais nuo ląstelės neurono ilgio. Šiame vaizdavime neurono padėtis SOM tinkle nurodo jo panašumą į kitus tinklo neuronus pagal kryptį, o spalva – pagal neurono ilgį.
7. SOM_Sammono ir SOM_SMACOF junginiai yra užtikrinantys panašią daugiamačių duomenų projekcijos kokybę. Tai leidžia taikyti ne tik dažai naudojamą SOM ir Sammono junginį, bet ir jam panašų SOM ir SMACOF algoritmų junginį, taip sutaupant skaičiavimas reikalingo laiko.
8. RPM algoritme galima naudoti atstumų funkciją, leidžiančią paklaidos minimizavimo algoritmui konverguoti nenaudojant papildomų konvergavimą skatinančių parametrų.

Literatūra

Abarius, E. 2009. *Daugiamačių duomenų vizualizavimo algoritmus apjungiančios sistemos projektas.* Baigiamasis darbas, Vilniaus kolegija, Elektronikos ir informatikos faluktetas, Vilnius.

Adler, R. J.; Taylor, J. E. 2007. *Random Fields and Geometry.* New York: Springer Science + Business Media LLC. ISBN-13:978-0-387-48112-8

Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. *In Proccedings of the ACM SIGMOD Internat. Confer. on Management of Data.*

Attneave, F. 1950. Dimensions of similarity. American Journal of Psychology, 3, 567–587.

Bernatavičienė, J. 2008. *Vizualios žinių gavybos metodologija ir jos tyrimas.* Vilnius: Technika. ISBN 978-9955-28-278-5

Bernatavičienė, J.; Dzemyda, G.; Marcinkevičius, V. 2007. Conditions for optimal efficiency of relative MDS. Informatica, 18(2), 187–202. ISSN 0868-4952

Bernatavičienė, J.; Dzemyda, G.; Marcinkevičius, V. 2007. Diagonal majorization algorithm: properties and efficiency. Information technology and control, 36(4), 353–358. ISSN 0392-124X

Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2006. Decision Support for Preliminary Medical Diagnosis Integrating the Data Mining Methods.

In Proccedings of the Simulation and Optimisation in Business and Industry: 5th International Conference on operational research, Kaunas: Technologija. 155-160.

Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2005. Optimal decisions in combining the SOM with nonlinear projection methods. European Journal of Operational Research, 173(3), 729–745. ISSN 0377-2217

Bezdek, C. J.; Pal, R. N. 1995. Index of topological preservation for feature extraction. Pattern Recognition, 28(3), 381–391.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning.* Singapore: Springer Science+Buisness. ISBN-10:0-387-31073-8

Biswas, G.; Jain, A.; Dubes, R. 1981. An Evaluation of Projection Algorithms. *In Proccedings of the IEEE Trans. on Pattern Analysis and Machine Intelligence, , PAMI-3*, 702–708.

Bloxom, J. 1968. *Individual differences in multidimensional scaling (Tech. Rep. No. ETS RM 68-45).* Princeton, NJ: Educational Testing Service.

Borg, I.; Groenen, J. P. 1997. *Modern Multidimensional Scaling: Theory and Applications.* New York: Springer-Verlag. ISBN 0-387-94845-7

Borg, I.; Groenen, J. P. 2005. *Modern Multidimensional Scaling: Theory and Applications* (Second Edition). New York: Springer.

Bronshtein, I. N.; Semendyayev, K. A.; Musion, G.; Muehling. 2004. *Handbook of Mathematics.* Berlin: Springer-Verlag. ISBN 3-540-43491-7

Chen, C.; Hardle, W.; Unwin, A. (Mont.). 2008. *Handbook of Data Visualization.* Berlin: Spinger. ISBN 978-3-540-33036-3

Cox, F. T.; Cox, A. M. 2001. *Multidimensional Scaling* (Second Edition). New York, USA: Chapman & Hall/CRC. ISBN 1-58488-094-5

Critchley, F. 2000. On a Framework for Dissimilarity Analysis. Esantis W. Gaul; O. Opitz; M. Schader (Mont.), *Data Analysis: Scientific Modeling and Practical Application* (121–143). Berlin: Springer-Verlag.

de Leeuw, J. 1977. Applications of convex analysis to multidimensional scaling. Esantis J. Barra; F. Brodeau; G. Romier; B. van Cutsem, *Recent developments in statistics* (133–145). Amsterdam: The Netherlands: North-Holland.

de Leeuw, J. 1988. Convergence of the majorization method for multidimensional scaling. Jornal of Classification, 5, 163–180.

de Leeuw, J. 2005. Shepard diagram. Esantis *The Encyclopedia of Statistics in Behavioral.* Wiley.

de Leeuw, J.; Heiser, W. 1977. Convergence of correction-matrix algorithms for multidimensional scaling. *In Proccedings of the Geometric representations of relational data,* Ann Arbor, MI: Mathesis. 735–752.

de Leeuw, J.; Mair, P. 2008. Multidimensional Scaling Using Majorization: SMACOF in R. Journal of Statistical Software, 31(3).

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

Dzemyda, G. 2001. Visualization of a set of parameters characterized by their correlation matrix. Computational Statistics and Data Analysis, 36(10), 15–30.

Dzemyda, G.; Kurasova, O. 2002. Comparative analysis of the graphical result presentation in the SOM software. Informatica, 13(3), 275–286.

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.

Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2003. Lygiagretūs skaičiavimai savireguliuojančio neuroninio tinklo junginyje su Sammono algoritmu. Lietuvos matematikos rinkinys, T. 43, Spec. nr., 218–222. ISSN 0132-2818.

Dzemyda, G.; Kurasova, O.; Žilinskas, J. 2008. *Daugiamačių duomenų vizualizavimo metodai* . Vilnius: Mokslo Aidai.

Evans, B. 2010. *Lecture on Quantization.* Paimta 2010 m. 06 16 d. iš http://users.ece.utexas.edu/~bevans/courses/realtime/lectures/08_Quantization/lecture8.ppt

Fayyad, U.; Grinstein, G. G.; Wierse, A. (Mont.). 2002. *Information Visualization in Data Mining and Knowledge Discovery.* Sant Diego: Academic Press. ISBN 1-55860-689-0

Fisher, A. F. 1936. The Use of Multiple Measurements in Axonomic Problems. Annals of Eugenics 7, 179–188.

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

Frank, A.; Asuncion, A. 2010. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Friedman, J.; J.W, T. 1974. A projection pursuit algorithm for exploratory data analysis. IEEEE transactions on Computers, 23(9), 881–890.

Golub, G.; Van Loan, C. K. 1996. *Matrix Computations* (Third ed.). Jon Hopkins University Press.

Goodhill, G. J.; Sejnowski, J. T. 1996. Quantifying neighbourhood preservation in topographic mappings. *In Proccedings of the 3rd Joint Symposium on Neutral Computation*, California: University of California. 61–82.

Gower, J. 1966. Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika, 53, 325–338.

Gower, J.; Groenen, P. 1991. Applications of the modified Leverrier-Faddeev algorithm for the construction of explicit matrix spectral decompositions and inverses. Utilitas Mathematica, 51–64.

Gray, A.; Abbena, E.; Salamon, S. 2006. *Modern Differential Geometry of Curves and Surfaces with Mathematica, Third Edition (Studies in Advanced Mathematics).* Boca Raton: Chapman and Hall/CRC. ISBN-10: 1584884487

Groenen, P. J.; Heiser, W. 1996. The tunneling method for global optimization in mutidimensional scaling. Psychometrika, 61, 529–550.

Groenen, P. J.; Heiser, W.; Meulman, J. 1998. City-block scaling: smoothing strategies for avoiding local minima. Esantis I. Balderjahn; R. Mathar; M. Schader (Mont.), *Classification, Data analysis, and Data Highways* Springer. 46–53.

Groenen, P.; van de Vaelden, M. 2004. *Multidimensional scaling.* Econometric Institute Report EI2004-15.

Guttman, L. 1968. A general nonmetric technique for finding the smallest coordinate space for a configuration of points. Psychometrika, 33, 469–506.

Handl, J.; Knowles, J. 2010. *Cluster generators: synthetic data for the evaluation of clustering algorithms*. Paimta 2010 iš http://dbkgroup.org/handl/generators/

Hassinen, P.; Elomaa, J.; Rönkkö; Halme, J. 1999. *Screen shots taen from program called Nenet V1.1a*. Paimta 2006 m. iš Neural Networks Information Homepage: http://koti.mbnet.fi/phodju/nenet/Nenet/General.html

Hawkins, D.; Bradu, D.; Kass, G. 1984. Location of several outliers in multiple regression data using elemental sets. Technometrics, 26, 197–208.

Hotelling, H. 1933. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417–441.

Ivanikovas, S. 2009. *Lygiagrečių skaičiavimų taikymo daugiamačiams duomenims vizualizuoti problemos.* Vilnius: Vytauto didžiojo universitetas.

Young, F.; Hamer, R. 1987. *Multidimensional Scaling: History, Theory and Applications.* New York: Erlbaum.

Young, G.; Householder, A. S. 1938. Discussion of a set of points in terms of their mutual distances. Psychometrika, 19–22.

Jolliffe, I. 2002. *Principal Componenet Analysis* (Second ed.). Springer.

Karbauskaitė, R.; Dzemyda, G. 2009. Topology Preservation Measures in the Visualization of Manifold-Type Multidimensional Data. 20(2).

Karbauskaitė, R.; Marcinkevičius, V.; Dzemyda, G. 2006. Testing the relational perspective map for visualization of multidimensional data. Technological and Economic Development of Economy, 12(4), 289–294. ISSN 1392–8619

Kaski, S.; Kangas, J.; T., K. 1998. Bibliography of self-organizing map (SOM) papers: 1981-1997. Neuroal Computing Surveys, 1(3&4), 1–176.

Kleiweg, P. 1996. *Neurale netwerken: Een inleidende cursus met practica voor de studie Alfa-Informatica.* Master's thesis, Rijksuniversiteit Groningen.

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

Kohonen, T. 2002b. Overture. Esantis U. Seifert; L. Jain (Mont.), *Self-Organizing Neural Networks* (1–10). New York: Springer-Verlag Company.

Kohonen, T. 2001. *Self-Organizing Maps* (third ed., Vol. 30). Springer-Verlag.

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

Kohonen, T.; Hynninen, J.; Kangas, J.; Laaksonen, J. 1996. *„SOM_PAK: The Self-Organizing Map Program Package".* Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Neural Networks Research Centre, Espoo, Finland.

Kraaijveld, M. A. 1995. A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps. IEEE Transactions on Neural Networks, 6(3), 548–559.

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

Kruskal, J. B. 1964b. Nonmetric multidimensional scaling: A numerical method. Psychometrika, 115–129.

Kurasova, O. 2005. *Daugiamačių duomenų vizuali analizė taikant savireguliuojančius neuroninius tinklus (SOM).* Vilnius: Technika.

Lee, A. J.; Verleysen, M. 2010. *Data visualization using nonlinear dimensionality reduction techniques: method review and quality assessment.* Paimta 2010 iš Université catholique de Louvain: http://www.inma.ucl.ac.be/news/slides/lee09.pdf

Lee, A. J.; Verleysen, M. 2007. *Nonlinear Dimensionality Reduction.* New York: Springer Science+Business Media. ISBN-13: 978-0-387-39350-6

Li, J. X. 2010. Paimta 2010 iš VisuMap Technologies: Visualizing high dimensional complex data: http://www.visumap.net/

Li, J. X. 2004. Visualization of high dimensional data with relational perspective map. Information Visualization, 3(1), 49–59. ISSN:1473-8716

Livré, R. 2004. *Topology.* Paimta 2010 iš http://www.math.huji.ac.il/~erezla/TOP/top.pdf

Mao, J.; K., J. A. 1995. Artificial neural networks for feature extraction and multivariate data projection. IEEE Transactions on Neural Networks, 6(2), 487–500.

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

Medvedev, V. 2007. *Tiesioginio sklidimo neuroninių tinklų taikymo daugiamačiams duomenims vizualizuoti tyrimas.* Vilnius: Technika.

Messey, F. J. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. Journal of the American Statistical Association, 46(253), 68–78.

Montvilas, A. M. 2003. Features of Sequential Nonlinear Mapping. Informatica, 14(3), 337–348.

Morrison, A.; Ross, G.; Chalmers, M. 2003. Fast multidimensional scaling through sampling, springs and interpolation. Information Visualization, 2(1), 67–77.

Munkres, J. R. 1975. *Topology*. Englewood Cliffs, New Jersey: Prentice-Hall.

Murtagh, F. 2004. *Multivariate Data Analysis Software and Resources*. Paimta 2005 iš http://astro.u-strasbg.fr/~fmurtagh/mda-sw/

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

Naud, A. 2004. Visualization of high-dimensional data using association. *In Proccedings of the Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, , 1*, 252–255.

Naud, A.; Duch, W. 2000. Interactive data exploration using MDS mapping. *In Proccedings of the Fifth Conference „Neural Networks and Soft Computing“, ,* Zakopane, Poland. 255–260.

Pearson, K. 1901. On lines and planes of closest fit to systems of points in space. The London, Edinburgh and Dublin Philosophical Magazine and Journal of Cience, Sixth Series, 559–572.

Podlipskytė, A. 2004. *Daugiamačių duomenų vizualizacija ir jos taikymas biomediciniių duomenų analizei* . Kaunas: Vytauto Didžiojo universitetas.

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. 2002. *Numerical Recipes in C++* (2nd). Cambridge: Cambridge University Press.

Prim, C. R. 1957. Shortest connection networks and some generalizations. 36.

Robertson, P.; De Ferrari, L. 1994. *Systematic Approaches to Visualization: Is a Reference Model Needed? in Scientific Visualization, Advances and Challenges.* (L. Rosenblum; R. Earnshaw; J. Encarnacao; H. Hagen; A. Kaufman; S. Klimenko; et al., Mont.) Academic Press.

Roveis, S. 1998. EM Algorithms for PCA and SPCA. Neural Information Processing Systems 10 (NIPS'97), 626–632.

Roweis, S. T.; Saul, L. K. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, 2323–2326.

Sammon, W. J. 1969. A nonlinear mapping for data structure analysis. 18, 401–409.

Seiffert, U.; Jain, L. C. 2002. *Self-Organizing Neuroal Networks. Recent Advances and Applications.* New York: Physica-Verlag Heidelberg. ISSN 1434-9922

Shepard, R. N. 1962a. The analysis of proximities: multidimensional scaling with an unknown distance function. I. Psychometrika, 27, 125–140.

Šaltenis, V. 1975. *Mapping of multidimensional points to lower dimensionality. (in Russian).* Inst. Math. Cybern. Vilnius: The State Fund of Algorithms and Programs. Reg. No. P001298.

Šaltenis, V.; Varnaitė, A. 1975. On the method of dimensionality reducing in multiextremal problems. Esantis A. Žilinskas (Mont.), *Teorija Optimaljnych Reshenij* (T. 1, 23–42). Vilnius: Inst. Math. Cybern.

Tenenbaum, J. B.; deSilva, V.; C., L. J. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 290, 2319–2323.

Torgenson, W. 1952. Multidimensional scaling: I. Theory and method. Psychometrika, 17, 401–419.

Torgerson, W. 1958. *Theory and Methods of Scaling.* New York: Wiley.

Treigys, P.; Šaltenis, V.; Dzemyda, G.; Barzdžiukas, V.; Paunksnis, A. 2008. Automated optic nerve disc parametrization. Informatica, 19(3), 403–420. ISSN 0868-4952

Trench, F. W. 2010. *Introduction to real analysis* (Free edition availible on http://ramanujan.math.trinity.edu/wtrench/texts/TRENCH_REAL_ANALYSIS.PDF). Library of Congress Cataloging-in-PublicationData. ISBN 0-13-045786-8

Trosset, W. M.; Groenen, P. J. 2005. Multidimensional Scaling Algorithms for Large Data Sets. *In Proccedings of the Computing Science and Statistics.*

Ultsch, A. S. 1990. Kohonen's Self-Organizing Feature Maps for Exploratory Data Analysis. *In Proccedings of the INNC'90, International Neural Network Conference,* , Dordrecht. 305-308.

Vesento, J. 2001. Importance of Individual Variables in the k-Means Algorithm. *In Proccedings of the PAKDD 2001,* Hong Kong: Availible from internet <http://lib.hut.fi/Diss/2002/isbn9512258978/article8.pdf>. 513–518.

Weisstein, E. W. 2010. *Torus.* Paimta 2010 iš From MathWorld--A Wolfram Web Resource: http://mathworld.wolfram.com/Torus.html

Žilinskas, A. 2000. *Matematinis programavimas.* Kaunas: Vytauto Didžiojo universiteto leidykla.

Žilinskas, A.; Žilinskas, J. 2008. A hybrid method for multidimensional scaling using city-block distances. Math. Meth. Oper. Res., 429–443.

Žilinskas, A.; Žilinskas, J. 2009. Branch and bound algorithm for multidimensional scaling with city-block metric. Journal of global optimization, 357–372.

Žilinskas, A.; Žilinskas, J. 2007. Two level minimization in multidimensional scaling. Jornal Global Optim., 581–596.

Autoriaus publikacijų disertacijos tema sąrašas

Straipsniai recenzuojamuose mokslo žurnaluose

A1. Bernatavičienė J., Dzemyda G., Marcinkevičius V., 2007. Conditions for Optimal Efficiency of Relative MDS, *Informatica*, Vol. 18(2), 187–202. ISSN 0868-4952. (*Current Abstracts. IAOR: International Abstracts In Operations Research. INSPEC. MatSciNet. ISI Web of Science. Scopus. TOC Premier. VINITI. Zentralblatt MATH*)

A2. Bernatavičienė J., Dzemyda G., Marcinkevičius V., 2007. Diagonal Majorization Algorithm: Properties and Efficiency, *Information Technology and Control*, Vol. 36(4), 353–358. ISSN 1392-124X. (*ISI Web of Science. VINITI. INSPEC*)

A3. Bernatavičienė J., Dzemyda G., Kurasova O., Marcinkevičius V., 2006. Optimal Decisions in Combining the SOM with Nonlinear Projection Methods, *European Journal of Operational Research*, Elsevier, Vol. 173(3), 729–745. ISSN 0377-2217. (*ISI Web of Science. Science Direct. INSPEC. Business Source Complete. GeoRef. Computer Abstracts International Database. Compendex*)

A4. Bernatavičienė J., Dzemyda G., Kurasova O., Marcinkevičius V., 2006. Strategies of Selecting the Basic Vector Set in the Relative MDS, *Technological and Economic Development of Economy*, Vol. 12(4), 283–288. ISSN 1392-8619. (*ASCE*

Civil Engineering Abstracts. Business Source Complete. Business Source Premier. Current Abstracts. ICONDA. SCOPUS. TOC Premier)

A5. Karbauskaitė R., Marcinkevičius V., Dzemyda G., 2006. Testing the Relational Perspective Map for Visualization of Multidimensional Data, *Technological and Economic Development of Economy*, Vol. 12(4), 289–294. ISSN 1392-8619. (*ASCE Civil Engineering Abstracts. Business Source Complete. Business Source Premier. Current Abstracts. ICONDA. SCOPUS. TOC Premier)*

A6. Dzemyda G., Bernatavičienė J., Kurasova O., Marcinkevičius V., 2004. Sammono projekcijos paklaidos minimizavimo strategijos, *Lietuvos matematikos rinkinys*, T. 44, Spec. nr., 1–6. ISSN 0132-2818. (*MatSciNet. CIS: current index to statistics. VINITI. Zentralblatt MATH*)

A7. Dzemyda G., Kurasova O., Marcinkevičius V., 2003. Lygiagretūs skaičiavimai savireguliuojančio neuroninio tinklo junginyje su Sammono algoritmu, *Lietuvos matematikos rinkinys*, T. 43, Spec. nr., 218–222. ISSN 0132-2818. *(MatSciNet. CIS: current index to statistics. VINITI. Zentralblatt MATH*)

A8. Dzemyda G., Kurasova O., Marcinkevičius V., 2003. MPI programų paketo taikymas lygiagrečiam vizualizavimui, *Informacijos mokslai*, Vilnius, Vilniaus universitetas, T. 26, 230–235. ISSN 1392-0561.

Straipsniai kituose leidiniuose

B1. Karbauskaitė R, Dzemyda G., Marcinkevičius V., 2008, Selecting a Regularisation Parameter in the Locally Linear Embedding Algorithm, *The 20th International Conference EURO Mini Conference Continuous Optimization and Knowledge-Based Technologies" EurOPT'2008: May 20-23, 2008*, Neringa, Lithuania: selected papers. Vilnius : Technica 59-64. *(Conference Proceedings Citation Index*)

B2. Marcinkevičius V., 2008. Statistical Estimation of the Multidimensional Data Visualization Algorithms, *Science and Supercomputing in Europe Report 2007*, Bologna: CINECA *Consorzio Interuniversitario*, 382–384. ISBN 978-88-86037-21-1.

B3. Bernatavičienė J., Dzemyda G., Kurasova O., Marcinkevičius V., Medvedev V., 2007. The Problem of Visual Analysis of Multidimensional Medical Data. Models and Algorithms for Global Optimization, *Springer Optimization and Its Applications*, New York, Springer, Vol. 4, 277–298. ISBN 978-0-387-36720-9. (*SpringerLINK*)

B4. Bernataviciene, J., Dzemyda, G., Kurasova, O., Marcinkevičius, V. 2006. Decision Support for Preliminary Medical Diagnosis Integrating the Data Mining Methods, *Simulation and Optimisation in Business and Industry: 5th International Conference on operational research*: May 17–20, 2006, Kaunas, Technologija, 155–160. ISBN 9955-25-061-5. (*ISI Proceedings*)

B5. Dzemyda G., Bernatavičienė J., Kurasova O., Marcinkevičius V. 2005. Minimization of the Mapping Error Using Coordinate Descent, *The 13-th*

International Conference in Central Europe on Computer Graphics, Visualization and Computer vision 2005 in Co-operation with Eurographics, Plzen, University of West Bohemia, 169–172. ISBN 80-903100-9-5.

B6. Marcinkevičius V., Dzemyda G., 2004. Daugiamačių duomenų vizualizavimas apmokytu SOM ir Sammono algoritmų junginiu, *Informacinės technologijos 2004*, *konferencijos pranešimų medžiaga*, Kaunas, Technologija, 350–355. ISBN 9955-09-588-1.

Virginijus MARCINKEVIČIUS

NETIESINĖS DAUGIAMAČIŲ DUOMENŲ
PROJEKCIJOS METODŲ SAVYBIŲ
TYRIMAS IR FUNKCIONALUMO GERINIMAS

Daktaro disertacija

Fiziniai mokslai (P 000),
Informatika (09 P)
Informatika, sistemų teorija (P 175)

Virginijus MARCINKEVIČIUS

INVESTIGATION AND FUNCTIONALITY IMPROVEMENT OF NONLINEAR
MULTIDIMENSIONAL DATA PROJECTION METHODS

Doctoral Dissertation

Physical sciences (P 000),
Informatics (09 P)
Informatics, systems theory (P 175)

2010 08 20 . 7,5 sp. l. Tiražas 20 egz.
Išleido Matematikos ir informatikos institutas
Akademijos g. 4, LT-08663 Vilnius.
Interneto svetainė: *http://www.mii.lt.*
Spausdino „Kauno technologijos universiteto spaustuvė“,
Studentų g.54, LT-51424 Kaunas