VILNIAUS UNIVERSITETAS

Alma Molytė

VEKTORIŲ KVANTAVIMO METODŲ JUNGIMO SU DAUGIAMATĖMIS SKALĖMIS ANALIZĖ

Daktaro disertacija
Fiziniai mokslai (P 000), informatika (09 P),
informatika, sistemų teorija (P 175)

Vilnius, 2011

Disertacija rengta 2006–2011 metais Matematikos ir informatikos institute.

Moksliniai vadovai:

dr. Olga Kurasova (Vilniaus universiteto Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09 P) (2007–2011),
prof. habil. dr. Vydūnas Šaltenis (Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07 T) (2006–2007).

Padėka

Nuoširdžiai dėkoju mokslinio darbo vadovei dr. Olgai Kurasovai už nuoširdžias, atsakingas bei vertingas mokslines konsultacijas, nuoseklų vadovavimą, pagalbą ir kantrybę rengiant šią disertaciją. Labai jai dėkinga už visapusišką supratimą, optimizmo skiepijimą, padrąsinimą ir pasitikėjimą manimi.

Esu dėkinga disertacijos recenzentams prof. dr. Daliui Navakauskui ir dr. Viktorui Medvedevui atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų pastabų bei patarimų, padėjusių pagerinti šio darbo kokybę.

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.

Alma Molytė

Reziumė

Dažnai iškyla būtinybė nustatyti ir giliau pažinti daugiamačių duomenų struktūrą: susidariusius klasterius, itin išsiskiriančius objektus, objektų tarpusavio panašumą ir skirtingumą. Vienas iš sprendimų būdų – duomenų dimensijos mažinimas ir jų vizualizavimas. Kai analizuojamos didelės duomenų aibės, tikslinga prieš vizualizavimą sumažinti ne tik dimensiją, bet ir duomenų skaičių. Šio darbo tyrimų sritis yra daugiamačių duomenų skaičiaus mažinimas ir duomenų atvaizdavimas plokštumoje.

Disertacijoje nagrinėjami dirbtiniais neuroniniais tinklais grindžiami vektorių kvantavimo ir dimensijos mažinimu pagrįsti vizualizavimo metodai. Kaip alternatyva saviorganizuojančių neuroninių tinklų ir daugiamačių skalių junginiams, darbe pasiūlyti nuoseklus neuroninių dujų ir daugiamačių skalių junginys bei integruotas, atsižvelgiantis į neuroninių dujų metodo mokymosi eigą ir leidžiantis gauti tikslesnę daugiamačių vektorių projekciją plokštumoje. Junginiais gautų vaizdų kokybės vertinimui pasirinkti Konigo matas, Spirmano koeficientas bei MDS paklaida. Šie matai leidžia kiekybiškai įvertinti panašumų išlaikymą po daugiamačių duomenų transformavimo į mažesnės dimensijos erdvę. Taip pat pasiūlyti dvimačių vektorių pradinių koordinačių parinkimo būdai nuosekliame junginyje ir integruoto junginio pirmame mokymo bloke bei koordinačių reikšmių priskyrimo būdai integruoto junginio kituose mokymo blokuose. Eksperimentiškai nustatyta kvantavimo paklaidos priklausomybė nuo neuroninių dujų tinklo mokymo parametrų reikšmių, atliekamų mokymo epochų, neuronų ir neuronų nugalėtojų skaičiaus. Eksperimentiškai ištirta ir parodyta, kad daugiamačių duomenų vizualizavimui nuoseklus ir integruotas neuroninių dujų ir daugiamačių skalių junginiai yra tinkamesni negu saviorganizuojančio neuroninio tinklo ir daugiamačių skalių junginiai.

Abstract

Often there is a need to establish and understand the structure of multidimensional data: their clusters, outliers, similarity and dissimilarity. One of solution ways is a dimensionality reduction and visualization of the data. If a huge datasets is analyzed, it is purposeful to reduce the number of the data items before visualization. The area of research is reduction of the number of the data analyzed and mapping the data in a plane.

In the dissertation, vector quantization methods, based on artificial neural networks, and visualization methods, based on a dimensionality reduction, have been investigated. The consecutive and integrated combinations of neural gas and multidimensional scaling have been proposed here as an alternative to combinations of self-organizing maps and multidimensional scaling. The visualization quality is estimated by König's topology preservation measure, Spearman's rho and MDS error. The measures allow us to evaluate the similarity preservation quantitatively after a transformation of multidimensional data into a lower dimension space. The ways of selecting the initial values of two-dimensional vectors in the consecutive combination and the first training block of the integrated combination have been proposed and the ways of assigning the initial values of two-dimensional vectors in all the training blocks, except the first one, of the integrated combination have been developed. The dependence of the quantization error on the values of training parameters, the number of epochs, neurons and neuron-winners has been defined experimentally. The fact that the consecutive and integrated combinations of the neural gas and multidimensional scaling is more suitable than the combination of the self-organizing map and multidimensional scaling for visualization of the multidimensional data has been experimentally tested and proved.

Žymėjimai

Simboliai

d	projekcinės erdvės, į kurią transformuojamas *n*-matis vektorius, dimensija
$d(Y_i, Y_j)$	Euklido atstumas tarp dvimačių vektorių Y_i ir Y_j
$d(X_i, X_j)$	Euklido atstumas tarp *n*-mačių vektorių X_i ir Y_i
E_{g}, E_{f}	neuroninių dujų metodo mokymo taisyklės parametrai
$\hat{e}$	epochų skaičius
e	vykdomos epochos numeris
E_{KM}	Konigo matas
E_{MDS}	daugiamačių skalių (MDS) paklaidos funkcija
$\hat{E}_{\mathrm{MDS}}$	MDS paklaida vizualizavimo rezultatams vertinti
E_{QE}	kvantavimo paklaida
c_{kl}	parametrų x_k ir x_l kovariacijos koeficientas
c_{kk}	duomenų aibės parametro x_k dispersija, kai $k = 1$
c_k^i	klasterio centro C^i k-toji komponentė

$C = \{c_{kl}\}$	kovariacinė matrica
C^i	klasterio centras
γ	integruoto junginio mokymo proceso blokų skaičius
h_{ij}^c	SOM kaimynystės funkcija
κ	klasterių skaičius
k_x	stačiakampio SOM tinklo eilučių skaičius
k_y	stačiakampio SOM tinklo stulpelių skaičius
$K = \{K^1, K^2, \dots, K^\kappa\}$	klasterių aibė
K^i	*i*-tasis klasteris
$\lambda_{\mathrm{g}}, \lambda_{\mathrm{f}}$	neuroninių dujų metodo mokymo taisyklės parametrai
m	analizuojamų objektų (vektorių) skaičius
$\widehat{M}_c$	neuronas nugalėtojas
M_i	kvantuoti vektoriai
$M_{ij} = (m_n^{ij})$	kvantuotų vektorių (neuronų) masyvas
μ	*n*-matės erdvės R^n vektoriaus X_i artimiausių kaimynų skaičius
n	vektoriaus komponenčių skaičius
N	neuronų skaičius
q	integruoto junginio mokymo bloko numeris
ρ_{Sp}	Spirmano koeficientas
r	neuronų nugalėtojų skaičius
r_X'	*n*-mačių duomenų vektorių (taškų) visų porų atstumų eilės numeriai
r_Y'	*d*-mačių duomenų vektorių (taškų) visų porų atstumų eilės numeriai
r_{kl}	koreliacijos koeficientas tarp parametrų x_k ir x_l
$r_X(i, j)$	vektoriaus X_i *j*-tojo arčiausio kaimyno X_{ij} eilės numeris
$r_Y(i, j)$	vektoriaus Y_i, atitinkančio vektorių X_i, *j*-tojo arčiausio

	kaimyno Y_{ij} eilės numeris
$R = \{r_{kl}\}$	koreliacinė matrica
R^d	*d*-matė erdvė
R^n	*n*-matė erdvė
$\delta(X_i, X_j)$	*n*-mačių vektorių X_i ir X_j artumas (panašumas arba skirtingumas)
t	iteracijos numeris
$t_{\max}$	iteracijų skaičius
v'	integruoto junginio mokymo bloko epochų skaičius
v	*d*-matės erdvės R^d vektoriaus Y_i artimiausių kaimynų skaičius
w_{ij}	svorio koeficientas
x_{ij}	duomenų aibės *n*-mačio *i*-tojo vektoriaus *j*-oji komponentė
X	analizuojamų duomenų aibė
X_i	analizuojamos duomenų aibės X *i*-asis vektorius
y_{ij}	*d*-mačio *i*-tojo vektoriaus *j*-oji komponentė
Y_i	*d*-matis vektorius

Santrumpos

ND	neuroninės dujos (angl. *neural gas*)
MDS	daugiamačių skalių metodas (angl. *multidimensional scaling*)
PKA	pagrindinių komponenčių analizė (angl. *principal component analysis*)
SMACOF	daugiamačių skalių paklaidos minimizavimo algoritmas (angl. *scaling by majorizing a complicated function*)
SOM	saviorganizuojantis neuroninis tinklas (angl. *self-organizing maps*)
VKM	vektorių kvantavimo mokymas (angl. *learning vector quantization*)

Turinys

1

Įvadas

1.1. Tyrimų sritis ir problemos aktualumas

Dabartinėmis technologijomis galima gauti ir saugoti didelius duomenų kiekius, tačiau jų suvokimas gana sudėtingas uždavinys, ypač kai duomenys nurodo sudėtingą objektą ar reiškinį, kuris aprašytas daugeliu parametrų, ir kurie gali būti ne tik skaitiniai, bet ir loginiai bei tekstiniai. Tokie duomenys vadinami daugiamačiais duomenimis. Dažnai iškyla būtinybė nustatyti ir giliau pažinti šių daugiamačių duomenų struktūrą, t. y. susidariusius klasterius, itin išsiskiriančius objektus, objektų tarpusavio panašumą ir skirtingumą. Visų parametrų reikšmių junginys charakterizuoja vieną analizuojamos aibės konkretų objektą. Daugiamačiai duomenys gali būti analizuojami įvairiais statistikos metodais, tačiau kai duomenų kiekis yra didelis, dažnai jų nepakanka, todėl siekiant gauti daugiau žinių iš analizuojamų duomenų, yra naudojami įvairūs duomenų tyrybos (angl. *data mining*) metodai: klasifikavimo, klasterizavimo, vizualizavimo ir kt.

Šio darbo tyrimų sritis yra daugiamačių duomenų skaičiaus mažinimas ir duomenų atvaizdavimas plokštumoje (vizualizavimas).

Viena grupė metodų, įgalinančių atrasti naujas žinias analizuojamose duomenų aibėse, yra vektorių kvantavimo metodai. Vektorių kvantavimas (angl. *vector quantization*) – tai procesas, kurio metu n-mačiai įėjimo vektoriai yra kvantuojami į ribotą aibę n-mačių išėjimo vektorių, kurių skaičius yra mažesnis nei įėjimo vektorių. Dažniausiai šie metodai taikomi garsui ir vaizdui suspausti, tačiau jie tinka ir duomenims klasterizuoti bei klasifikuoti.

Daugiamačių duomenų vizualizavimo, dar kitaip vadinamo dimensijos mažinimo, metodais didelės dimensijos duomenys transformuojami į mažesnės dimensijos erdvę taip, kad išliktų esamos arba būtų atrastos „užslėptos“ analizuojamų duomenų savybės. Jais transformavus daugiamačius duomenis į trimatę erdvę ar plokštumą ir juos vizualizavus, daug paprasčiau suvokti duomenų struktūrą ir sąryšius tarp jų. Vienas iš dažnai taikomų metodų yra daugiamačių skalių (angl. *multidimensional scaling*). Daugiamačių skalių metodo vienos iteracijos skaičiavimų sudėtingumas yra $O(nm^2)$, čia m – objektų skaičius, n – dimensijų skaičius, todėl būtina ieškoti būdų skaičiavimams pagreitinti. Vienas iš būdų yra taikyti vektorių kvantavimo metodą ir juo sumažinti duomenų objektų skaičių prieš juos vizualizuojant. Duomenis sumažinti būtina taip, kad nauja duomenų aibė kaip galima labiau atspindėtų analizuojamos duomenų aibės savybes.

Daugiamačių skalių rezultatas labai priklauso nuo dvimačių vektorių pradinių koordinačių parinkimo būdo, kai daugiamačių skalių paklaida minimizuojama iteraciniu būdu. Siūlomi įvairūs pradinių koordinačių parinkimo būdai, tačiau šio uždavinio sprendimas išlieka aktuali problema.

Šioje disertacijoje sprendžiamos dvi pagrindinės problemos:

1. Duomenų aibės vektorių ir jų dimensijų skaičiaus sumažinimas vektorių kvantavimo ir daugiamačių skalių metodų junginiais, išlaikant duomenų struktūrą;

2. Gautų rezultatų priklausomybės nuo dvimačių vektorių, gautų daugiamačių skalių metodu, pradinių koordinačių parinkimas.

1.2. Darbo tikslas ir uždaviniai

Pagrindinis šio darbo tikslas – greitai ir tiksliai atvaizduoti didelės apimties duomenų aibes plokštumoje, tam sukuriant vektorių kvantavimo ir duomenų dimensijų mažinimo metodų junginį ir pasiūlant tinkamus dvimačių vektorių pradinių koordinačių parinkimo būdus.

Siekiant tikslo buvo sprendžiami šie uždaviniai:

- išnagrinėti vektorių kvantavimo strategijas duomenims klasterizuoti;
- ištirti vektorių kvantavimo metodų jungimo galimybes su vizualizavimo metodais, pagrįstais duomenų dimensijų skaičiaus mažinimu;
- ištirti dvimačių vektorių pradinių koordinačių reikšmių parinkimo nuosekliajame junginyje ir integruoto junginio pirmajame mokymo bloke būdus;
- ištirti dvimačių vektorių pradinių koordinačių priskyrimo integruoto junginio visuose mokymo blokuose, išskyrus pirmąjį, būdus;
- sukurti naujus nuoseklaus ir integruoto neuroninių dujų ir daugiamačių skalių metodų junginius, leidžiančius gauti tikslesnę daugiamačių vektorių projekciją plokštumoje ir atlikti išsamią jų lyginamąją analizę su saviorganizuojančio neuroninio tinklo ir daugiamačių skalių junginiais;
- atlikti gautų kvantavimo ir vizualizavimo rezultatų kokybės analizę.

1.3. Tyrimo objektas ir metodai

Analizuojant daugiamačius duomenis, norint geriau atskleisti jų struktūrą, vien tik klasikinių vizualizavimo metodų dažnai nepakanka. Disertacijos tyrimo objektai – dirbtiniais neuroniniais tinklais grindžiami vektorių kvantavimo metodai ir daugiamačių duomenų vizualizavimo metodai, pagrįsti dimensijų skaičiaus mažinimu. Su tyrimo objektu betarpiškai yra susiję šie dalykai: daugiamačių duomenų projekcijos į mažesnės dimensijos erdvę kokybės įvertinimo matai, dvimačių vektorių koordinačių parinkimo būdai ir jų atvaizdavimas plokštumoje.

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

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

1.4. Darbo mokslinis naujumas

1. Sukurtas nuoseklus neuroninių dujų ir daugiamačių skalių junginys ir integruotas, atsižvelgiantis į neuroninių dujų metodo mokymosi eigą ir leidžiantis gauti tikslesnę daugiamačių vektorių projekciją plokštumoje.
2. Pasiūlyti dvimačių vektorių pradinių koordinačių parinkimo būdai integruoto junginio pirmame mokymo bloke ir koordinačių reikšmių priskyrimo būdai integruoto junginio kituose mokymo blokuose.
3. Eksperimentiškai nustatyta kvantavimo paklaidos priklausomybė nuo neuroninių dujų tinklo mokymo parametrų reikšmių, atliekamų mokymo epochų, neuronų ir neuronų nugalėtojų skaičiaus.

4. Eksperimentiškai ištirta ir parodyta, kad daugiamačių duomenų vizualizavimui neuroninių dujų ir daugiamačių skalių junginys yra tinkamesnis negu saviorganizuojančio neuroninio tinklo ir daugiamačių skalių junginys.

1.5. Darbo rezultatų praktinė reikšmė

Tyrimų, atliktų naudojant įvairios prigimties realaus pobūdžio duomenis, rezultatai atskleidė, kad vektorių kvantavimo ir projekcijos mažinimo metodo junginiai gali būti plačiai taikomi daugiamačiams duomenims vizualizuoti. Analizuojant kitus realaus pobūdžio skaitinius duomenis, bus galima remtis išvadomis, gautomis šioje disertacijoje.

1.6. Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 8 moksliniuose leidiniuose: 5 periodiniuose recenzuojamuose mokslo žurnaluose, 3 straipsniai konferencijų pranešimų medžiagoje.

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

- 11-oji Lietuvos jaunųjų mokslininkų konferencija „Informatika“. 2008 m. balandžio 9 – 11d., Vilnius, Vilniaus Gedimino technikos universitetas, Lietuva.
- Lietuvos matematikų draugijos XLIX konferencija. 2008 m. birželio 25 – 26 d., Kaunas, Vytauto Didžiojo universitetas, Lietuva.
- XIII International Conference Applied Stochastic Models and Data Analysis (ASMDA–2009). June 30 – July 3, 2009, Vilnius, Lietuva.
- VI International Conference on Machine Learning and Data Mining

(MLDM 2009). July 23 – 25, 2009, Leipzig, Germany.

- Computer Graphics, Vision and Mathematics (GraVisMa 2009). September 2 – 4, 2009, Plzen, Czech Republic.
- XIV tarptautinė kompiuterininkų konferencija: Kompiuterininkų dienos – 2009. 2009 m. rugsėjo 25 – 26 d., Kaunas, Kauno technologijos universitetas, Lietuva.

1.7. Disertacijos struktūra

Disertaciją sudaro 5 skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Vektorių kvantavimo ir vizualizavimo metodai, Vektorių kvantavimo ir projekcijos metodų jungimas, Eksperimentiniai tyrimai, Bendrosios išvados. Papildomai disertacijoje pateikta: naudotų žymėjimų ir santrumpų sąrašai. Bendra disertacijos apimtis 135 puslapiai, kuriuose pateikta 50 paveikslų ir 16 lentelių. Disertacijoje remtasi 81 literatūros šaltiniu.

2

Vektorių kvantavimo ir vizualizavimo metodai

Dabartinės informacinės komunikacinės technologijos užtikrina didelių duomenų srautų generavimą ir jų saugojimą, tačiau tebelieka didelė spraga tarp duomenų surinkimo bei saugojimo ir jų suvokimo. Duomenų suvokimas yra gana sudėtingas uždavinys, ypač kai duomenys nurodo sudėtingą objektą, reiškinį, kuris aprašomas daugeliu parametrų (savybių). Sąvoka „objektas“ gali apimti įvairius dalykus: žmones, įrenginius, gamybos produktus ir kt. Visų parametrų reikšmių junginys charakterizuoja vieną konkretų analizuojamos aibės $X=\{X_1, X_2, ..., X_m\}$ objektą $X_i=(x_{i1}, x_{i2}, ..., x_{in})$, čia m yra analizuojamų objektų skaičius, n – parametrų skaičius ir i – objekto eilės numeris. Jei parametrų reikšmės yra skaitinės, tai $X_1, X_2, ..., X_m$ yra n-mačiai vektoriai. Dažnai jie interpretuojami kaip taškai n-matėje erdvėje R^n, čia n – erdvės dimensijos skaičius. Šie taškai atitinka vektorius, kurių pradžios koordinatės $(0,0,...,0)$. Savybių reikšmės $(x_{i1}, x_{i2}, ..., x_{in})$ yra vektoriaus X_i, $i=1,..,m$, komponentės (taškų koordinatės). Taigi, turime analizuojamų duomenų aibės

matricą $X=\{X_1, X_2, ..., X_m\}=\{x_{ij}, i=1,...,m,\ j=1,...,n\}$, jos eilutės yra vektoriai $X_i=(x_{i1}, x_{i2}, ..., x_{in})$, $i=1,..,m$, čia x_{ij} yra *i*-tojo vektoriaus *j*-oji komponentė. Ši duomenų matrica gali būti analizuojama įvairiais statistikos metodais, tačiau kai duomenų kiekis yra didelis, dažnai klasikinių statistikos metodų nepakanka. Siekiant gauti daugiau žinių iš analizuojamų duomenų, yra taikomi įvairūs duomenų tyrybos (angl. *data mining*) metodai: klasifikavimo, klasterizavimo, vizualizavimo ir kt. (Dunham 2003, Dzemyda *et al.* 2007, Han and Kamber 2006, Lorose 2004). Klasifikavimo uždavinių tikslas – turint aibę duomenų, kurių klasės įprastai yra žinomos, sukurti taisykles, pagal kurias duomenys, kurie nebuvo naudojami kuriant tas taisykles, automatiškai bus priskirti vienai ar kitai žinomai klasei. *Klasterizavimas* – tai analizuojamų objektų suskirstymas į skirtingas grupes (klasterius) taip, kad grupės viduje esantys objektai būtų panašūs tarpusavyje (pavyzdžiui artimi), o objektai iš skirtingų grupių būtų nepanašūs. Daugiamačių duomenų *vizualizavimo*, dar kitaip vadinamais dimensijos mažinimo, metodais didelės dimensijos duomenys transformuojami į mažesnės dimensijos erdvę taip, kad išliktų arba būtų atrastos „užslėptos“ analizuojamų duomenų savybės.

Vektorių *kvantavimo metodais* taip pat galima atrasti naujas žinias analizuojamose duomenų aibėse. *Saviorganizuojantys neuroniniai tinklai* (Kohonen 2001), *vektorių mokymo kvantavimas* (Kohonen 2001), *neuroninių dujų metodas* (Martinetz and Schulten 1991) yra metodai, grindžiami dirbtiniais neuroniniais tinklais. *Vektorių kvantavimas* – tai procesas, kurio metu *n*-mačiai įėjimo vektoriai yra kvantuojami į ribotą *n*-mačių išėjimo vektorių aibę, sudarytą iš mažesnio skaičiaus vektorių nei analizuojama duomenų aibė. Vektorių kvantavimo metodai yra taikomi duomenims klasterizuoti, suspausti. Jie gali būti taikomi duomenų su trūkstama reikšmių dalimi analizei ir pan. Vektorių kvantavimo ir vizualizavimo metodų jungimas leidžia greitai atvaizduoti dideles duomenų aibes plokštumoje neprarandant tikslumo.

2.1. Tyrimuose naudojami duomenys

Disertacijos eksperimentinėje dalyje yra analizuojamos įvairios duomenų aibės, turinčios tam tikrų specifinių savybių.

Duomenų aibės, paimtos iš duomenų bazės „UCI Repository of Machine Learning Databases“ (Asuncion and Newman 2007):

1. **Fišerio irisų duomenys** [150;4], kurie kartais vadinami tiesiog irisais arba irisų duomenimis, yra klasikiniai testiniai duomenys, naudojami daugiamačių duomenų analizei. Išmatuoti keturi 150-ies irisų (vilkdalgių) žiedų parametrai: vainiklapių pločiai (angl. *petal weight*), vainiklapių ilgiai (angl. *petal height*), taurėlapių pločiai (angl. *sepal weight*), taurėlapių ilgiai (angl. *sepal height*). Matuotos trijų rūšių gėlės: Iris Setosa (I klasė), Iris Versicolor (II klasė) ir Iris Virginica (III klasė). Sudaryti 4-mačiai vektoriai $X_1, X_2, ..., X_{150}$, čia ($X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$, $i = 1, 2, ..., 150$).
2. **Automobilių (autoMPG)** [392; 7], pagamintų 1970–1982 metais JAV, Japonijoje ir Europoje, duomenys (392 automobiliai). Trijų regionų automobilius charakterizuoja 9 parametrai: degalų sąnaudos (*angl. mpg*), cilindrų skaičius (angl. *cylinders*), variklio darbo tūris (angl. *displacement),* arklio jėgų kiekis (angl. *horsepower*), svoris (angl. *weight*), greitis (angl. *accelerator*), modelio pagaminimo metai (angl. *model years*), kilmės regionas (angl. *origin*), modelis (angl. *name*). Paskutiniai du parametrai nėra skaitiniai, todėl paprastai vizualizavimo procese tiesiogiai nėra naudojami – jie nusako objektų priklausomybę skirtingoms klasėms, pavyzdžiui, skirtingi kilmės regionai ar modeliai. Dažniausiai analizuojami 7-mačiai vektoriai $X_1, X_2, ..., X_{392}$, čia ($X_i = (x_{i1}, x_{i2}, ..., x_{i7})$, $i = 1, 2, ..., 392$).
3. **Krūties vėžio** duomenys [683; 9] surinkti Viskonsino universiteto (JAV) ligoninėje. Du naviko tipus – nepiktybinį (angl. *benign*) ir

piktybinį (angl. *malignant*) – apibūdina devyni skaitiniai parametrai: klampumas (angl. *clump thickness*), ląstelės dydžio vienodumas (angl. *uniformity of cell size*), ląstelės formos vienodumas (angl. *uniformity of cell shape*), kraštinis sulipimas (angl. *marginal adhesion*), vienasluoksnio epitelio ląstelės dydis (angl. *single epithelial cell size*), „nuogas“ branduolys (angl. *bare nuclei*), švelnus chromatinas (angl. *bland chromatin*), normalus (nepakitęs) branduolėlis (angl. *normal nucleoli*), mitozės (angl. *mitozes*). Sudaryti 9-mačiai vektoriai $X_1, X_2, ..., X_{683}$ ($X_i = (x_{i1}, x_{i2}, ..., x_{i9})$, $i = 1, 2, ..., 683$).

Duomenų aibės, paimtos iš interneto svetainės „Fundamental Clustering Problems Suite“ (FCPS 2011):

1. **Chainlink** [1000; 3], tai yra du sujungti žiedai. Tai 3-mačiai vektoriai $X_1, X_2, ..., X_{1000}$ ($X_i = (x_{i1}, x_{i2}, x_{i3})$, $i = 1, 2, ..., 1000$).
2. **Hepta** [212; 3], tai yra 7 atskiri klasteriai. Tai 3-mačiai vektoriai $X_1, X_2, ..., X_{212}$ ($X_i = (x_{i1}, x_{i2}, x_{i3})$, $i = 1, 2, ..., 212$).
3. **Target** [770; 2], tai yra taškai atsiskyrėliai. Tai 2-mačiai vektoriai $X_1, X_2, ..., X_{770}$ ($X_i = (x_{i1}, x_{i2})$, $i = 1, 2, ..., 770$).

Duomenų aibė **Elipsoidai** [1338; 100] paimta iš „Cluster generators: synthetic data for the evaluation of clustering algorithms“ duomenų bazės (Handl and Knowles 2010). Šios duomenų aibės vektoriai suformuoja 20 persidengiančių elipsoidinio tipo klasterių. Tai 100-mačiai vektoriai $X_1, X_2, ..., X_{1338}$ ($X_i = (x_{i1}, x_{i2}, ..., x_{100})$, $i = 1, 2, ..., 1338$).

Taip pat tyrime naudoti **kviečių duomenys** [400; 12] (Raudys *et al.* 2007). Analizuotos 5 kvietinių augalų rūšys: rugiai, avižos, miežiai, žieminiai ir vasariniai kviečiai. Buvo paimta kiekvieno grūdo skaitmeninė nuotrauka, išmatuota 12 parametrų: dydis (angl. *size*), forma (angl. *shape*), spalva (angl. *color*), plotas (angl. *area*), perimetras (angl. *perimeter*), ilgoji ašis

(angl. *major axis*), trumpoji ašis (angl. *minor axis*), skersmuo (angl. *equivalent diameter*), ekscentricitetas (angl. *eccentricity*), kontūro faktorius (angl. *shape factor*), apvalumas (angl. *roundness*), kompaktiškumas (angl. *compactness*). Sudaryti 12-mačiai vektoriai $X_1, X_2, ..., X_{400}$ ($X_i = (x_{i1}, x_{i2}, ..., x_{i12})$, $i = 1, 2, ..., 400$).

Tyrimuose naudoti ir dirbtinai sugeneruoti įvairių dydžių masyvai, sudaryti iš vektorių, kurių komponentės yra dydžiai, tolygiai pasiskirstę intervale (0, 1). Buvo sugeneruota 100 aibių, kurios sudarytos iš vektorių $X_1, X_2, ..., X_m$, čia ($X_i = (x_{i1}, x_{i2}, ..., x_{in}$, $i = 1, 2, ..., m$), $m = 20, 50, 100, 200, 1500$, o n = 5, 10, 15, 20, 50, 100, 150, 200 o taip pat duomenys, kurių charakteristikos pateiktos (2.1 lentelė).

2.1 lentelė. *Dirbtinai sugeneruotos duomenų aibės rand_clust* ir rand_data1500*

Nr.	Pavadinimas	m	n	Klasių skaičius	Apibūdinimas
1.	rand_clust5	100	5	5	
2.	rand_clust10	100	10	5	tolimesni klasteriai
3.	rand_clust57	100	10	5	tolimesni klasteriai
4.	rand_clust24	100	10	5	artimesni klasteriai
5.	rand_clust13	100	10	5	artimesni klasteriai
6.	rand_data1500	1500	5	1	

2.2. Vektorių kvantavimo metodai

Tarkime, kad turime analizuojamų duomenų aibę $X = \{X_1, X_2, ..., X_m\}$, sudarytą iš vektorių $X_i \in R^n$, $X_i = (x_{i1}, x_{i1}, ..., x_{in})$, $i = 1, ..., m$, čia x_{ij} yra i-tojo vektoriaus j-toji komponentė, n – komponenčių skaičius, m – analizuojamų vektorių skaičius.

Kvantavimo metodų tikslas – rasti vektorių $M_1, M_2, ..., M_N \in R^n$, $N < m$, rinkinį tokį, kad gauti kvantuoti vektoriai M_i, $i = 1, ..., N$, atspindėtų vektorių X_l, $l = 1, ..., m$, savybes.

Vektorių kvantavimo metodai yra klasifikuojami į:

1. Vektorių kvantavimo metodus be mokytojo (angl. *unsupervised*).
   1.1. Neuroninės dujos (ND) (angl. *neural gas*).
   1.2. Saviorganizuojantys neuroniniai tinklai (SOM) (angl. *self-organizing maps*).
2. Vektorių kvantavimo metodus su mokytoju (angl. *supervised*).
   2.1. Vektorių kvantavimo mokymo algoritmas VKM1 (angl. *learning vector quantization*).
   2.2. Vektorių kvantavimo mokymo algoritmas VKM2 (angl. *learning vector quantization*).

Neuroninių dujų metodas

Neuroninės dujos (angl. *neural gas*) – tai vienas iš kvantavimo metodų be mokytojo (angl. *supervised*), pagrįstas neuroniniu tinklu (Martinetz and Schulten 1991).

Tarkime, turime daugiamačius duomenis, kurie išreikšti *n*-matės erdvės vektoriais $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $i = 1,...,m$, čia x_{ij} yra duomenų *i*-tojo vektoriaus X_i *j*-toji komponentė, *n* – vektoriaus komponenčių skaičius, *m* – analizuojamų duomenų vektorių skaičius.

Neuroninių dujų metodu sukuriamas neuronų masyvas $M_1, M_2, ..., M_N \in R^n$, čia N – neuronų skaičius. Mokant neuroninį tinklą keičiamos neuronų komponentės. Mokymo pabaigoje neuronai tampa analizuojamos duomenų aibės vektorių $X_i \in R^n$, $i = 1,...,m$, kvantuotais vektoriais. Detaliau šis metodas aprašytas 3.1 skyrelyje.

Saviorganizuojantys neuroniniai tinklai

Saviorganizuojantis neuroninis tinklas (SOM) (angl. *self-organizing maps*) (Kohonen 2001) yra vektorių, išdėstytų dvimačio tinklelio (lentelės,

žemėlapio) mazguose, masyvas. Kiekvienas *n*-matis apmokymo aibės vektorius mokymo metu yra susiejamas su vienu tinklo neuronu, kuris taip pat yra *n*-matis vektorius. Mokymo pradžioje tinklo neuronų komponentės generuojamos, pavyzdžiui, atsitiktinai. Kiekviename mokymo žingsnyje vienas iš apmokymo aibės vektorių pateikiamas į tinklą. Tinklo neuronų reikšmės keičiamos pagal tam tikras taisykles. Apmokant neuroninį tinklą apskaičiuojami žemėlapio vektoriai ir tuos vektorius atitinkančių objektų numeriai, t. y. objektai pasiskirsto tarp žemėlapio elementų. Šis žemėlapis gali būti interpretuojamas kaip daugiamačių duomenų atvaizdavimas plokštumoje, nes įmanoma vizualiai stebėti objektų išsidėstymą. Išskirtinė tokio atvaizdavimo savybė – duomenų sugrupavimas (surūšiavimas, klasterizavimas) pagal jų panašumą (Yacoub *et al.* 2000), tačiau sunku pasakyti, kaip toli yra gretimuose lentelės langeliuose esantys vektoriai. Detaliau šis metodas aprašytas 3.2 skyrelyje.

Vektorių kvantavimo mokymas

Vektorių kvantavimas ir saviorganizuojantys neuroniniai tinklai yra glaudžiai susiję su vektorių kvantavimo mokymu, kuris apima visą klasę gimininingų algoritmų. Vektorių kvantavimo mokymo metodai apibrėžia mokymą su mokytoju algoritmus. Galime išskirti kelis pagrindinius vektorių kvantavimo mokymo algoritmus, kuriuos pažymėsime atitinkamai VKM1, VKM2.

VKM1 algoritmas

Turime duomenų aibės matricą $X=\{X_1, X_2, ..., X_m\}=\{x_{ij}, i=1,...,m, j=1,...,n\}$, jos eilutės yra vektoriai $X_i \in R^n$, t. y. $X_i=(x_{i1}, x_{i2}, ..., x_{in})$, $i=1,...,m$, čia x_{ij} yra *i*-tojo vektoriaus *j*-toji komponentė, $n-$ komponenčių skaičius, m – analizuojamų vektorių skaičius. Tarkime, kvantuojami vektoriai sudaro aibę $M=\{M_1, M_2, ..., M_N\}$, čia $M_k=\{m_{k1}, m_{k2}, ..., m_{kn}\}$, $k=1,...,N$, $N-$

kvantuojamų vektorių skaičius. Pradžioje pasirenkame kvantuotų vektorių skaičių ir atsitiktinai parenkame pradines kvantuotų vektorių koordinates. Tiek duomenų aibės vektoriams, tiek ir kvantuojamiems vektoriams yra nurodomi klasių numeriai. Į tinklą pateikus vektorių X_l, $l \in \{1,...,m\}$, suskaičiuojamas Euklido atstumas nuo jo iki visų kvantuojamų vektorių. Randame mažiausią atstumą. Tarkime tai vektorius M_c. Jeigu duomenų vektorius X_l, $l \in \{1,...,m\}$, ir kvantuojamas vektorius priklauso tai pačiai klasei, tai kvantuojamo vektoriaus reikšmės keičiamos pagal formulę:

$$M_c(t+1) = M_c(t) + \alpha \cdot (X_l(t) - M_c(t)).$$

Jeigu duomenų vektorius X_l, $l \in \{1,...,m\}$, ir kvantuojamas vektorius priklauso skirtingoms klasėms, tai kvantuojamo vektoriaus reikšmės keičiamos pagal formulę:

$$M_c(t+1) = M_c(t) - \alpha \cdot X_l(t) - M_c(t)).$$

Visų kitų kvantuojamų vektorių reikšmės nekeičiamos.

Čia α – mokymo parametras, o t – iteracijos numeris. Darbe T. Kohonen (2001) rekomenduoja, kad parametro α reikšmė būtų maža ir ne didesnė negu 0,1, o iteracijų skaičius 30 – 50 kartų didesnis negu yra kvantuojamų vektorių.

VKM2 algoritmas

Kaip ir VKM1 algoritme, taip ir šiame algoritme pradžioje pasirenkame kvantuotų vektorių skaičių ir atsitiktinai parenkame pradines kvantuotų vektorių koordinates. Tiek duomenų aibės vektoriams, tiek ir kvantuojamiems vektoriams yra nurodomi klasių numeriai. Kiekvienam į tinklą pateiktam vektoriui X_l, $l \in \{1,...,m\}$, surandami du artimiausi kvantuojami vektoriai M_c ir M_k, kurie yra gaunami paskaičiavus Euklido atstumus. Jeigu duomenų vektorius X_l, $l \in \{1,...,m\}$, priklauso bent vienai iš kvantuotų vektorių M_c arba M_k klasei (tarkime, X_l, $l \in \{1,...,m\}$, priklauso tai pačiai klasei, kaip ir

kvantuojamas vektorius M_c), tai kvantuojamų vektorių reikšmės keičiamos pagal formules:

$$M_c(t+1) = M_c(t) + \alpha \cdot (X_l(t) - M_c(t)),$$

$$M_k(t+1) = M_k(t) - \alpha \cdot (X_l(t) - M_k(t)).$$

Jeigu abu kvantuojami vektoriai M_c ir M_k priklauso tai pačiai klasei, kaip ir duomenų vektorius X_l, $l \in \{1,...,m\}$, tai kvantuojamų vektorių reikšmės keičiamos pagal atitinkamas formules:

$$M_c(t+1) = M_c(t) + \varepsilon \cdot \alpha \cdot (X_l(t) - M_c(t)),$$

$$M_k(t+1) = M_k(t) + \varepsilon \cdot \alpha \cdot (X_l(t) - M_k(t)).$$

Visų kitų kvantuojamų vektorių reikšmės nekeičiamos. Čia ε ir α – mokymo parametrai, o t – iteracijos numeris. ε reikšmė turi priklausyti intervalui nuo 0,1 iki 0,5. T. Kohonen (2001) rekomenduoja, kad parametro α reikšmė būtų maža ir ne didesnė negu 0,1.

K-vidurkių metodas

Vektoriams kvantuoti gali būti naudojamas *K-vidurkių* (angl. *K-means*) klasterizavimo metodas, kuris priskiriamas padalijimo klasterizavimo metodų grupei (MacQueen 1967, Vesento 2001). Padalijimo klasterizavimo metoduose stengiamasi duomenų aibę padalinti į nesusikertančius klasterius. Naudojant skirtingus algoritmus galima sukurti skirtingas klasterių aibes. Šio tipo klasterizavimo algoritmuose minimizuojama kriterijaus funkcija. Kriterijus sukuriamas toks, kad minimizuojant klasterių panašumą, klasterių skirtumas būtų maksimizuojamas. Tai gali būti vidutinis atstumas tarp klasterių arba kitas matas. Pradžioje inicijuojama klasterių aibė, tada iteraciniame procese objektai perkeliami iš vieno klasterio į kitą, siekiant gauti geriausią klasterizavimo kokybę pasirinkto mato prasme. Didžiausia tokios klasterinės analizės problema – nustatyti optimalų klasterių skaičių κ, kadangi parinkus kitą κ reikšmę, galima gauti kitus klasterius.

K-vidurkių klasterizavimo metodą galima laikyti ir kvadratinės paklaidos algoritmu (angl. *squared error clustering algorithm*) (Dunham 2003), nes jis minimizuoja kvadratinę paklaidą. Tegu klasteriui K^i priskirta objektų aibė $X=\{X_1^i, X_2^i, ..., X_{\mu_i}^i\}$, μ_i – objektų klasteryje K^i skaičius, $X_j^i=(x_{j1}^i, x_{j2}^i, ..., x_{jn}^i)$, $j=1,...,\mu_i$. Tada kvadratinė paklaida vienam klasteriui K^i yra Euklido atstumų tarp kiekvieno klasterio elemento ir klasterio centro C^i kvadratų suma $E_{K^i}=\sum_{j=1}^{\mu_i}\left\|X_j^i-C^i\right\|^2$, čia $C^i=(c_1^i, c_2^i, ..., c_n^i)$ klasterio centras, kurio komponentės randamos pagal formulę $c_k^i=\sum_{j=1}^{\mu_i}x_{jk}^i\Big/\mu_i$, $(k=1,...,n)$.

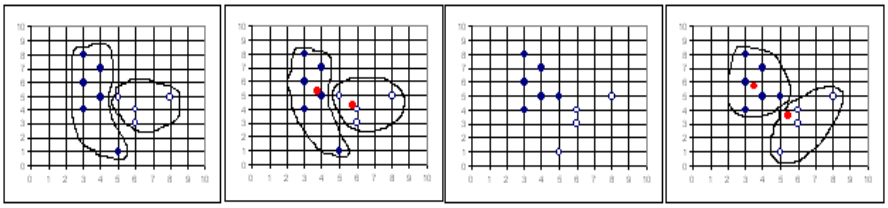
Kvadratinė paklaida klasterių aibei $K=\{K^1, K^2, ..., K^\kappa\}$ apskaičiuojama pagal formulę $E_{K\text{-}means}=\sum_{i=1}^{\kappa}E_{K^i}$.

Vieno iš galimų funkcijos minimizavimo algoritmų (K-vidurkių) struktūra yra tokia (2.1 paveikslas):

1. Inicijuojami κ klasterių centrai C^i, $(i=1,...,\kappa)$.
2. Kiekvienas analizuojamų duomenų aibės objektas priskiriamas tam klasteriui, iki kurio centro atstumas yra mažiausias.
3. Perskaičiuojami kiekvieno klasterio centrai.
4. Skaičiuojama kvadratinė paklaida pagal $E_{K\text{-}means}=\sum_{i=1}^{\kappa}E_{K^i}$ formulę.
5. 2–4 žingsniai kartojami, kol kvadratinės paklaidos reikšmė tampa mažesnė už pasirinktą slenkstinę reikšmę arba objektai nebepersiskirsto kitiems klasteriams.

K-vidurkių algoritmo trūkumas tas, kad randamas kvadratinės paklaidos lokalusis, o ne globalusis minimumas. Šis metodas pakankamai lėtai konverguoja; veikia tik su metriniais duomenimis. Rezultatui didelę įtaką daro

taškai-atsiskyrėliai (angl. *outliers*). Algoritmą būtina vykdyti kelis kartus pradedant su skirtingais klasterių centrais ir būtina žinoti klasterių skaičių, priešingu atveju, algoritmą reiktų vykdyti su skirtingomis κ reikšmėmis. Klasterių skaičiui įvertinti Milligan ir Cooper (1985) siūlo pasinaudoti Miligano algoritmu.



2.1 pav. *K-vidurkių metodo veikimo pavyzdys*

Yra sukurtų ir K-vidurkių metodo praplėtimų: darbe (Dempster *et al.* 1977) siūlo EM algoritmą (angl. *expectation maximization*), o darbe (Ball and Hall 1965) siūlomas ISODATA algoritmas.

K-vidurinių taškų klasterizavimo metodas

Klasterio centras C^i (vidurkis tarp klasterio elementų) nebūtinai yra klasterio elementas. Klasterinėje analizėje vartojama tokia sąvoka kaip vidurinis taškas (angl. *medoid*). Tai klasterio taškas, esantis arčiausiai klasterio centro. *K-vidurinių taškų metodo* idėja panaši į K-vidurkių metodą, tačiau čia vietoj klasterio centro naudojamas klasterio vidurio taškas (Kaufman and Rousseeuw 1987). Šis algoritmas literatūroje dar vadinamas PAM (angl. *partitioning around medoids*).

Algoritmo struktūra:

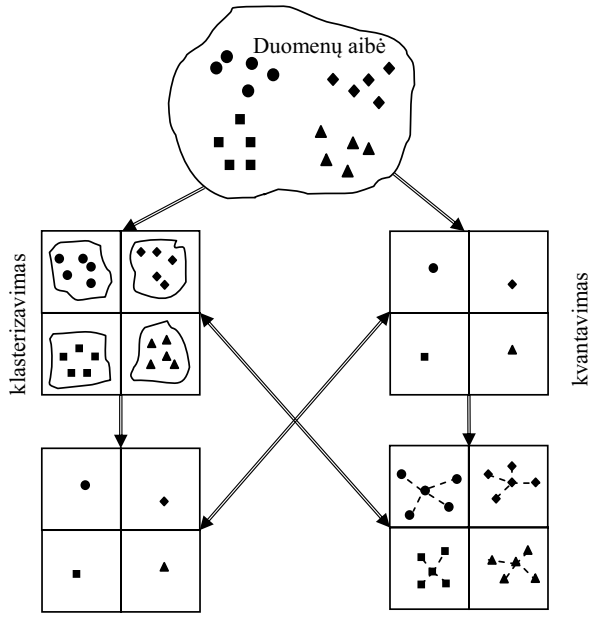
1. Atsitiktinai pasirenkame analizuojamos duomenų aibės κ elementų. Tarkime, kad jie bus viduriniais taškais.
2. Visus taškus (elementus) suskirstome į klasterius pagal artimiausius vidurinius taškus.

3. Skaičiuojame tikslo funkciją, kuri gali būti lygi atstumų nuo kiekvieno klasterio centro iki klasterio vidurinio taško sumai.
4. Pasirenkame bet kurį klasterio tašką, kuris nėra apibrėžtas kaip vidurinis taškas.
5. Pasirinktas taškas tampa viduriniu tašku, jei toks pakeitimas sumažintų tikslo funkcijos reikšmę.
6. 2–5 žingsniai kartojami, kol klasteriai nepersigrupuoja.

K-vidurinių taškų metodo privalumas lyginant su K-vidurkių metodu yra tas, kad jame labiau atsižvelgiama į taškus-atsiskyrėlius. Tačiau šis metodas tinka tik nedidelės duomenų aibės analizei ir yra lėtesnis už K-vidurkių metodą.

Klasterizavimo ir kvantavimo sąvokų paaiškinimas

Kaip matome *klasterizavimo* ir *kvantavimo* sąvokos yra panašios, bet tuo pačiu ir skirtingos. Kartais literatūroje jos laikomos net sinonimais.



2.2 pav. *Klasterizavimo ir kvantavimo palyginimo schema*

Skirtumą tarp šių sąvokų iliustruoja paprastas pavyzdys, pateiktas 2.2 paveiksle. Tarkime, kad turime duomenų aibę, kurią sudaro 20 dvimačių vektorių. Klasterizavimo metu visi duomenų aibės vektoriai suklasterizuojami į keturis klasterius. Vektorių kvantavimo metu sumažinamas analizuojamų vektorių skaičius (iki 4), t. y. randami keturi duomenų atstovai.

Turint klasterizuotus duomenis, galime apskaičiuoti klasterių atstovus. Tai būtų ekvivalentu kvantavimo metu gautam rezultatui. Kvantavimo metu rastam kiekvienam duomenų atstovui priskiriami artimiausi vektoriai. Gaunami klasteriai, ekvivalentūs klasterizavimo metu gautiems klasteriams.

2.3. Vizualizavimo (projekcijos) metodai

Daugiamačių duomenų *vizualizavimo metodai* padeda nustatyti ar įvertinti daugiamačių duomenų struktūrą: susidariusias grupes (klasterius), itin išsiskiriančius objektus (taškus atsiskyrėlius), panašumus tarp analizuojamų objektų ar jų grupių ir pan. Šie metodai intensyviai plėtojami siekiant didinti duomenų analizės efektyvumą, suprantamiau pateikti ir objektyviau įvertinti duomenų tyrybos rezultatus. Vizualizavimo metodai plėtojami dviemis pagrindinėmis kryptimis: tiesioginio vizualizavimo metodais, kai kiekvienas daugiamačio objekto parametras yra pateikiamas tam tikra vizualia forma ir projekcijos metodais, kai analizuojamą duomenų aibė transformuojama iš *n*-matės erdvės R^n į mažesnės dimensijos vaizdo erdvę R^d, kur duomenų aibės taškų išdėstymą galima stebėti vizualiai.

Daugiamačių duomenų tiesioginio vizualizavimo metoduose nėra apibrėžto formalaus matematinio kriterijaus. Visi daugiamačių duomenų parametrai pateikiami žmogui priimtina vizualia forma. Šie metodai yra klasifikuojami į *geometrinius* (taškiniai grafikai, linijiniai grafikai, lygiagrečios koordinatės, projekcijos paieška ir kt.) (Grinstein *et al.* 2001, Grinstein *et al.* 2002, Grinstein and Ward 2002, Hoffman and Grinstein 2002, Inselberg 1981,

Wegman and Luo 1996, Kruskal 1972), *simbolinius* (Černovo veidai, ženklų metodas, žvaigždžių metodas ir kt.) (Chernoff 1973, Kraus and Ertl 2001, Kaski 1997, Grinstein *et al.* 2002) ir *hierarchinio vizualizavimo metodus* (grotelių metodas, dimensijų įterpimas, fraktalai ir kt.) (Michalski 1978, Becker and Cleveland 1996, Rabenhorst 1994).

Daugiamačiams duomenims transformuoti į mažesnės dimensijos erdvę taikomi *projekcijos metodai.* Jie dar vadinami dimensijų skaičiaus mažinimo metodais (angl. *dimensionality reduction techniques*). Šie metodai gali būti naudojami ir daugiamačiams duomenims vizualizuoti, kai pasirinktas pakankamai mažas projekcinės erdvės R^d dimensijos skaičius (dažniausiai $d=2$). Analizuojant daugiamačius duomenis, kuriuos apibūdina *n* parametrų, norima vizualiai įvertinti jų išdėstymą *n*-matėje erdvėje. Kai $n>3$, tiesiogiai pamatyti šių taškų neįmanoma, tačiau yra galimybė rasti jų projekciją į dvimatę ar trimatę erdvę.

Vizualizuojant daugiamačius duomenis susiduriama su dviem dažnai vienas kitam prieštaraujančiais tikslais: viena vertus, norima supaprastinti uždavinį mažinant duomenų dimensiją, kita vertus, norima išlaikyti kiek galima daugiau originalios informacijos turinio, t. y. kuo mažiau iškraipyti analizuojamus duomenis. Projekcijos metoduose yra formalūs matematiniai projekcijos kokybės kriterijai. Jie optimizuojami siekiant gauti kiek galima tikslesnę daugiamačių duomenų projekciją mažesnio skaičiaus dimensijų erdvėje, t. y. kuo mažiau iškraipyti duomenis pasirinkto kriterijaus atžvilgiu, išsaugoti vizualizuojamų daugiamačių vektorių tarpusavio atstumų ar kitų artimumo įverčių proporcijas, taip pat išsaugoti ar net išryškinti kitas sprendimams priimti svarbias analizuojamos duomenų aibės charakteristikas, pavyzdžiui, klasterius.

Dažniausiai išskiriami tiesinės ir netiesinės projekcijos metodai.

1. *Tiesinės projekcijos metodai:*

1.1. Pagrindinių komponenčių analizė (angl. *principal component analysis*) (Opitz and Hilbert 2000, Taylor 2003, Yeung and Ruzzo 2001).

1.2. Tiesinė diskriminantinė analizė (angl. *linear discriminant analysis*) (Dunham 2003).

1.3. Projekcijos paieškos (angl. *projection pursuit*) (Kruskal 1972).

2. *Netiesinės projekcijos metodai:*

2.1. Daugiamačių skalių (angl. *multidimensional scaling*) (Borg and Groenen 2005).

2.2. Pagrindinės kreivės (angl. *principal curves*) (Delicado 1997).

2.3. Lokaliai tiesinio vaizdavimo (angl. *locally linear embedding*), (Roweis and Saul 2000).

2.4. ISOMAP (Tenenbaum *et al.* 2000).

Tiesinės projekcijos metodais ieškoma tiesinės analizuojamų duomenų transformacijos, o netiesinės projekcijos – netiesinės transformacijos. Bendru atveju yra įvairių tiesinės transformacijos galimybių: pasukimas, postūmis, atspindys, suspaudimas ir t. t. Pagrindinis skirtumas tarp jų yra tas, kad tiesiniai projekcijos metodai ieško tiesinio poerdvio, tokio kaip tiesė arba plokštuma, o netiesiniai metodai ieško netiesinio poerdvio.

Dabartinės technologijos leidžia kaupti didelius duomenų masyvus, milžiniški informacijos kiekiai saugomi genetinėje medžiagoje, tačiau dažnai įprasti duomenų analizės metodai ir įrankiai sunkiai susidoroja su tokiais didžiuliais duomenų kiekiais. Todėl būtini nauji metodai ir juos realizuojantys įrankiai, padėsiantys išgauti žinių iš tokių duomenų ir palengvinsiantys daryti tinkamus sprendimus.

Didelės apimties duomenų aibių vizualizavimas dažnai negalimas įprastais daugiamačių duomenų vizualizavimo metodais dėl kelių priežasčių:

- Techninės priežastys: skaičiavimuose reikalingos didelės duomenų matricos ir dažnai nepakanka operatyviosios kompiuterio atminties jas saugoti, o jas saugant diske, duomenų apdorojimo laikas labai sulėtėja; didelių duomenų masyvų apdorojimą gali riboti ir kompiuterio procesoriaus greitis.
- Informacijos suvokimo priežastys: jei labai daug duomenų bus pavaizduota duomenų vaizde, juos bus sunku suvokti, taškai, atitinkantys analizuojamus duomenis, persidengs.

Didelių apimčių duomenų aibių ($m \cdot n \approx 10^4$, kur m ir n – duomenų matricos eilučių ir stulpelių skaičiai) vizualizavimui taikomi įvairūs metodai. Vieni iš jų yra įvairios daugiamačių skalių modifikacijos, pavyzdžiui, diagonalinio mažoravimo algoritmas, santykinių daugiamačių skalių algoritmas (Williams and Munzner 2004, Naud 2004, de Silva and Tenenbaum 2003, Trosset and Groenen 2005, Agrafiotis *et al.* 2001, Chen *et al.* 2008, Hoffman and Grinstein 2002). A. Unwin (2006) monografijoje pateikiami įvairūs tiesioginio vizualizavimo metodai, pritaikyti didelėms aibėms vizualizuoti.

Daugiamačių duomenų vizualizavimo tematika pastaruoju metu yra plačiai analizuojama ir Lietuvoje. Tai nėra nauja mokslo sritis, joje dirba nemažai mokslininkų. Lietuvoje daugiamačių skalių metodą pirmasis pradėjo nagrinėti prof. habil. dr. V. Šaltenis (Šaltenis and Varnaitė 1975). Darbuose (Šaltenis and Aušraite 2002, Dzemyda *et al.* 2007, Karbauskaitė *et al.* 2007, Karbauskaitė and Dzemyda 2009) analizuojamos daugiamačių duomenų vizualizavimo idėjos, metodai bei problemos. Taip pat daugiamačių duomenų vizualizavimo algoritmus tyrinėja prof. G. Dzemyda, prof. A. Žilinskas, dr. J. Žilinskas. Detali vizualizavimo metodų apžvalga pateikta G. Dzemydos, O. Kurasovos ir J. Žilinsko (2008) vadovėlyje, skirtame informatikos krypties doktorantams ir magistrantams. A. Žilinskas ir J. Žilinskas (2009) tyrinėja daugiamates skales su miesto kvartalų (angl. *city-block*) metrika ir ieško

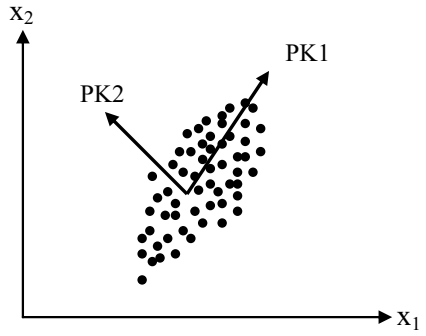
daugiamačių skalių įtempimo (angl. *stress*) funkcijos globalaus minimumo, tačiau didėjant taškų skaičiui ir projekcijos erdvės dimensijai *d*, uždavinio sudėtingumas auga eksponentiškai ir surasti globalų minimumą, kai $n \cdot d > 24$ ir $d \geq 2$, tampa labai sunku arba neįmanoma. Šiai problemai spręsti siūlomi šakų ir rėžių (Žilinskas and Žilinskas 2009) ir dviejų lygių hibridiniai algoritmai (Žilinskas and Žilinskas 2007, Žilinskas and Žilinskas 2008), kurie ypač efektyvūs dirbant su didelės apimties duomenimis. Pastaruoju metu apgintos kelios daktaro disertacijos, kuriose nagrinėjamos daugiamačių duomenų vizualizavimo problemos. O. Kurasovos (2005) disertacijoje nagrinėti saviorganizuojančių neuroninių tinklų ir daugiamačių skalių jungimo būdai. V. Medvedev (2007) ir S. Ivanikovo (2009) disertacijose detaliai išanalizuoti dirbtiniais neuroniniais tinklais grindžiamų daugiamačių duomenų vizualizavimo metodai, užtikrinantys efektyvų daugiamačių duomenų projekcijos paklaidos minimizavimą bei pagerinantys neuroninio tinklo mokymą. J. Bernatavičienės (2008) disertacijoje pateikta vizualios žinių gavybos metodologija, kuri leidžia padidinti duomenų analizės efektyvumą. V. Marcinkevičiaus (2010) disertacijoje detaliai išanalizuoti diagonalinio mažoravimo ir santykinių daugiamačių skalių algoritmai su pradinių vektorių inicijavimu pagal didžiausias dispersijas ir šie algoritmai palyginti su daugiamačių skalių SMACOF algoritmu. Daugiamačių duomenų vizualizavimo algoritmai ir metodai išlaikantys lokalią struktūrą, t. y. atstumų tarp artimiausių taškų santykių išlaikymas po analizuojamos daugiamačių duomenų aibės transformavimo iš didesnės dimensijos erdvės į mažesnės dimensijos erdvę detaliai išanalizuota R. Karbauskaitės (2010) disertacijoje.

2.3.1. Pagrindinių komponenčių analizė

Pagrindinių komponenčių analizė (PKA) (angl. *principal component analysis*) – tai klasikinis statistikinis tiesinės projekcijos metodas. Tai tiesinė duomenų transformacija, kuri plačiai naudojama duomenų analizėje kaip

daugiamačių duomenų dimensijos mažinimo metodas. Šiuo metodu ieškoma daugiamačių duomenų mažesnės dimensijos poerdvio, kuriame būtų išlaikyta daugiau originalios erdvės duomenų savybių bei informacijos.

Metodo tikslas – rasti kryptį, kuria dispersija yra didžiausia (Yeung and Ruzzo 2001, Opitz and Hilbert 2000, Taylor 2003). Didžiausią dispersiją turinti kryptis vadinama pirmąja pagrindine komponente (PK1). Ji eina per duomenų centrinį tašką. Taškų visumos vidutinis kvadratinis atstumas iki šios tiesės yra minimalus, t. y. ši tiesė yra kiek galima arčiau visų duomenų taškų (2.3 paveikslas). Antrosios pagrindinės komponentės (PK2) ašis taip pat turi eiti per duomenų centrinį tašką ir ji turi būti ortogonali pirmosios pagrindinės komponentės ašiai. Esminė PKA idėja yra sumažinti duomenų dimensijų skaičių atliekant tiesinę transformaciją ir atsisakant dalies po transformacijos gautų naujų komponenčių, kurių dispersijos yra mažiausios (Opitz and Hilbert 2000, Yeung and Ruzzo 2001, Taylor 2003). Tai padeda geriau vizualizuoti duomenis, o tuo pačiu ir palengvina duomenų suvokimą.



2.3 pav. *Pirmoji (PK1) ir antroji (PK2) pagrindinės komponentės*

Kaip pastebėjo C. Burtas (1949) šis metodas jau 1901 metais buvo analizuojamas K. Pirsono (1901), o 1933 metais jį išvystė H. Hotellingas (1933).

Tarkime, kad turime duomenų matricą, sudarytą iš vektorių $X=\{X_1, X_2,..., X_m\}=\{x_{ij},\ i=1,...,m,\ j=1,...,n\}$, kurios *i*-oji eilutė yra vektorius

$X_i = (x_{i1}, x_{i2}, ..., x_{in})$. Matricos X stulpeliai žymi duomenų parametrus (jų iš viso yra *n*), o eilutės – daugiamačius vektorius (jų iš viso yra *m*).

Tarp parametrų x_k ir x_l koreliacijos koeficientas r_{kl} skaičiuojamas pagal formulę:

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

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

Iš koreliacijos koeficientų, gautų iš (2.1) formulės, galima suformuoti *koreliacinę matricą* $R = \{r_{kl}, k, l = 1, ..., n\}$. Matricos R įstrižainės elementai lygūs vienetui.

Parametrų x_k ir x_l *kovariacijos koeficientas* c_{kl} yra skaičiuojamas pagal formulę:

$$c_{kl} = \frac{1}{m-1} \sum_{i=1}^{m} (x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l). \qquad (2.2)$$

Kai $k = l$, išraiška (2.2) yra dispersijos formulė, t. y. c_{kk} yra duomenų aibės parametro x_k dispersija.

Iš kovariacijos koeficientų, gautų remiantis (2.2) formule, galima suformuoti *kovariacinę matricą* $C = \{c_{kl}, k, l = 1, ..., n\}$. Ši matrica yra simetrinė.

Iš (2.1) ir (2.2) formulių gauname, kad koreliacijos koeficientas lygus $r_{kl} = c_{kl} / \sqrt{c_{kk} c_{ll}}$. Jei parametrai x_k ir x_l nekoreliuoti, tai jų kovariacijos koeficientas $c_{kl} = c_{lk} = 0$, $k \neq l$.

Apibrėžkime kovariacinės matricos tikrinius vektorius ir jų tikrines reikšmes. Matricos *tikrinis vektorius* (angl. *eigenvector*) E_k ir jį atitinkanti *tikrinė reikšmė* (angl. *eigenvalue*) λ_k yra lygties $CE_k = \lambda_k E_k$ sprendinys. Šioje lygtyje E_k yra vektorius-stulpelis; C – kovariacinė matrica; λ_k reikšmė

randama iš charakteringos lygties $|C-\lambda_k I|=0$. Čia I yra vienetinė matrica, kurios matmenys tokie pat kaip matricos C. Ženklu $|\cdot|$ apibrėžtas determinantas. Tikrinių vektorių skaičius yra lygūs kintamųjų skaičiui, t. y. n. Yra sukurta nemažai tikrinių reikšmių radimo uždavinio sprendimo metodų (Kvedaras and Sapagovas 1974).

Tikrinis vektorius, susijęs su didžiausia tikrine reikšme, turi tokią pat kryptį kaip pirmoji pagrindinė komponentė. Antroji pagrindinė komponentė atitinka antrąjį tikrinį vektorių ir t. t. Pagrindinių komponenčių skaičius parenkamas atsižvelgiant į projektinės erdvės dimensiją. Yra sukurti efektyvūs algoritmai pagrindinėms komponentėms rasti. Tam gali būti taikomi ir dirbtiniai neuroniniai tinklai.

Surūšiavus tikrinius vektorius E_k juos atitinkančių tikrinių reikšmių mažėjimo tvarka $(\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n)$, galima sukurti ortogonalią bazę, kurioje pirmas tikrinis vektorius yra nukreiptas didžiausios duomenų dispersijos kryptimi. Taigi, tikriniai vektoriai laikomi pagrindinėmis komponentėmis. Tikrinė reikšmė nurodo visos dispersijos dalį. Reikšmė λ_k – tai duomenų imties dispersija kryptimi E_k, o C_{kk} – duomenų imties dispersija k-tojo kintamojo x_k kryptimi.

Sudarykime pagrindinių komponenčių matricą $A=(E_1, E_2, ..., E_n)$. Jos stulpeliai yra tikriniai vektoriai E_k, $k=1,...,n$, atitinkantys tikrines reikšmes $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n$. Kiekvienas šios matricos A vektorius-stulpelis yra ortogonalus bet kuriam kitam. Transformuokime bet kurį duomenų vektorių X pagal formulę:

$$Y_i = (X_i - \overline{X})A, \tag{2.3}$$

čia $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $\overline{X} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_{in})$, $A = (E_1, E_2, ..., E_n)$.

Iš (2.3) formulės gauti $Y_i = (y_{i1}, y_{i2}, ..., y_{in})$ yra vektoriai (taškai) naujoje ortogonalioje koordinačių sistemoje $(y_1, y_2, ..., y_n)$, apibrėžtoje tikriniais vektoriais E_k, $k = 1, ..., n$. Vektoriaus Y_i, $i = 1, ..., m$, komponenčių $y_1, y_2, ..., y_n$ kovariacinė matrica lygi: $C_y = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$.

Dabar galima išreikšti originalų duomenų vektorių X_i, $i = 1, ..., m$, per vektorių Y_i pasinaudojus šia formule:

$$X_i = Y_I A^T + \overline{X}. \qquad (2.4)$$

Ji gauta iš (2.3) formulės pasinaudojus ortogonalių matricų savybe, kad $A^{-1} = A^T$, čia A^{-1} yra atvirkštinė, A^T – transponuota matrica.

Daugiamačių duomenų transformacijai galima naudoti ne visus tikrinius vektorius, o tik pirmuosius. Tegul matrica A_d sudaryta iš *d* pirmųjų tikrinių vektorių. Tada galima sukurti transformacijas analogiškas (2.3) išraiškai: $Y_i = (X_i - \overline{X})A_d$, $i = 1, ..., m$. Tokiu būdu randama duomenų vektoriaus projekcija į *d*-matę erdvę. Vizualizavimui pasirenkame $d = 2$ arba $d = 3$.

Jei duomenys sukoncentruoti tiesiniame poerdvyje, tai duomenų dimensijų skaičius sumažinamas neprarandant daug informacijos.

Jeigu PKA naudojama ne duomenų vizualizavimui, kyla klausimas, kiek pagrindinių komponenčių *d* parinkti. Dauguma skaičiaus *d* parinkimo taisyklės nėra griežtos (Yeung and Ruzzo 2001, Pan 2001). Galima parinkti tokį mažiausią sveiką skaičių *d*, kad procentinis (santykinis) dydis nuo visos dispersijos viršytų tam tikrą pasirinktą lygį, pavyzdžiui, 90 %. Taigi išdėsčius tikrinius vektorius jų tikrinių reikšmių mažėjimo tvarka ir imant *d* pirmųjų vektorių, galima apskaičiuoti, kokia paklaida bus daroma. Pagal leidžiamą paklaidą imama daugiau arba mažiau pagrindinių komponenčių.

Kai kurie autoriai pagrindinių komponenčių analizėje, užuot naudojęsi kovariacine matrica, renkasi koreliacinę matricą. Tai yra ekvivalentu tam tikram parametrų normavimui, o tik po to naujų parametrų (pagrindinių komponenčių) paieškai.

Jei egzistuoja tiesinės priklausomybės tarp parametrų, tai, taikant pagrindinių komponenčių metodą, duomenų dimensijų skaičius mažinamas su nedidelėmis paklaidomis. Tačiau gali egzistuoti netiesinės priklausomybės, kurių metodas negali įvertinti, ir tai yra šio metodo trūkumas. Taip pat standartiniai PKA metodai netinka duomenų, sudarytų iš daug parametrų, analizei, nes daugiamačių duomenų kovariacinė matrica yra labai didelė. Jos dydis yra $n \times n$. PKA metodas netinkamas vizualizuoti daugiamačius duomenis sudarytus iš kelių šimtų parametrų.

2.3.2. Daugiamatės skalės

Daugiamačių skalių (MDS) (angl. *multidimensional scaling*) metodas (Borg and Groenen 2005) – tai netiesinės projekcijos metodas, plačiai naudojamas daugiamačių duomenų analizei įvairiose šakose, ypač ekonomikoje, socialiniuose moksluose ir kt. Naudojant MDS metodą *n*-mačiai vektoriai projektuojami į mažesnę dimensijų skaičiaus erdvę (dažniausiai į R^2), siekiant išlaikyti analizuojamos aibės objektų artumus – panašumus arba skirtingumus (Kaski 1997, Jansson and Johansson 2003, Borg and Groenen 2005, Naud and Duch 2000). Gautuose vaizduose panašūs objektai išdėstomi arčiau vieni kitų, o skirtingi – toliau vieni nuo kitų.

Pradiniai daugiamačių skalių metodo duomenys yra kvadratinė simetrinė matrica, kurios elementai nusako artimumą tarp analizuojamų objektų. Tai gali būti arba panašumų arba skirtingumų matrica. Paprasčiausiu atveju tai yra Euklido atstumų tarp objektų matrica. Tačiau nebūtinai turi būti atstumai griežtai matematine prasme.

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

Vienas daugiamačių skalių metodo tikslų yra rasti optimalų daugiamačius objektus atitinkančių taškų (vektorių) vaizdą mažo skaičiaus dimensijų erdvėje. Yra daugybė skirtingų daugiamačių skalių variantų su skirtingomis paklaidų funkcijomis (angl. *stress*) ir jas optimizuojančiais algoritmais (Borg and Groenen 2005). Pagal analizuojamus duomenis daugiamačių skalių algoritmai gali būti skirstomi į *metrinius* ir *nemetrinius* (angl. *metric, non-metric*). Pirmasis daugiamačių skalių metodo algoritmas metriniams duomenims buvo pasiūlytas 1930 metais, vėliau MDS algoritmai buvo taikyti ir nemetriniams duomenims.

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

Tarkime kiekvieną *n*-matį vektorių $X_i \in R^n$, $i \in \{1,...,m\}$, atitinka mažesnio dimensijų skaičiaus vektorius $Y_i \in R^d$, $d < n$. Artumą (panašumą arba skirtingumą) tarp *n*-mačių vektorių X_i ir X_j pažymėkime $\delta(X_i, X_j)$, o atstumą tarp dvimačių vektorių Y_i ir Y_j – $d(Y_i, Y_j)$, $i, j = 1,...,m$. Jeigu artumas yra Euklido atstumas, tai $\delta(X_i, X_j) = d(X_i, X_j)$.

Naudojantis MDS algoritmu, bandoma atstumą $d(Y_i, Y_j)$ priartinti prie atstumo $d(X_i, X_j)$. Jei naudojama kvadratinė paklaidos funkcija, tai minimizuojama tikslo funkcija E_{MDS} gali būti užrašyta taip:

$$E_{\text{MDS}} = \sum_{i<j} w_{ij} (\delta(X_i, X_j) - d(Y_i, Y_j))^2. \tag{2.5}$$

Paklaidos funkcija E_{MDS} dar vadinama Stress funkcija. Dažniausiai naudojami tokie svoriai w_{ij}:

$$w_{ij} = \frac{1}{\sum_{i<j} (d(X_i, X_j))^2},$$

$$w_{ij} = \frac{1}{d(X_i, X_j) \sum_{k<l} d(X_k, X_l)},$$

$$w_{ij} = \frac{1}{m d(X_i, X_j)},$$

čia $d(X_i, X_j) \neq 0$, t. y. tarp vektorių $X_1, X_2, ..., X_m$ nėra sutampančių.

Funkcijų minimizavimui galima naudoti gradientinį nusileidimą. Pradėjus nuo atsitiktinės pradinių dvimačių taškų konfigūracijos, iteraciniame procese dvimačių vektorių $Y_i \in R^2$ koordinatės y_{ik}, $i = 1, ..., m$, $k = 1, 2$, keičiamos pagal formulę $y_{ik}(t+1) = y_{ik}(t) - \eta \partial E_{\text{MDS}}(t) / \partial_{ik}(t)$. Čia t yra iteracijos numeris, η – parametras, įtakojantis optimizavimo žingsnį.

Literatūroje minimi ir kiti paklaidos funkcijos optimizavimo būdai, tokie kaip *jungtinių gradientų metodas*, *kvazi-Niutono metodas*, *deterministinis atkaitinimo modeliavimo algoritmas* (angl. *simulated annealing*) (Klock and Buhmann 2000), *kombinatorinis MDS 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 2003), *SMACOF* (angl. *Scaling by MAjorizing*

a COmplicated Function) *algoritmas*, pagrįstas tikslo funkcijos mažoravimu (Borg and Groenen 2005).

Šioje disertacijoje naudojamas vienas populiariausių funkcijos E_{MDS} (2.5) minimizavimo iteracinis *SMACOF algoritmas*, kai svoriai $w_{ij}=1,\ \forall i,j$. Šis metodas užtikrina paklaidos funkcijos E_{MDS} monotonišką konvergavimą (Borg and Groenen 2005).

SMACOF algoritmo eiga:

1. Inicializuojami dvimačiai aibės $Y=\{Y_1,Y_2,...,Y_m\}$ vektoriai. Pradinė iteracija lygi $t=0$.
2. Apskaičiuojama projekcijos paklaida $E_{\text{MDS}}(Y(t))$ pagal (2.5) formulę.
3. Iteracijų numerį *t* padidiname vienetu.
4. Apskaičiuojame $Y(t)$ pagal formulę $Y(t+1)=m^{-1}B(Y(t))Y(t)$, čia matricos $B(Y(t))$ elementai apskaičiuojami pagal formules:

$$b_{ij}=\begin{cases}\dfrac{-d(X_i,X_j)}{d(Y_i,Y_j)}, & \text{kai } i\neq j \text{ ir } d(Y_i,Y_j)\neq 0,\\ 0, & \text{kai } i\neq j \text{ ir } d(Y_i,Y_j)=0;\end{cases}$$

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

5. Apskaičiuojame $E_{\text{MDS}}(Y(t))$ pagal (2.5). Jeigu $E_{\text{MDS}}(Y(t-1))-E_{\text{MDS}}(Y(t))<\varepsilon$ arba *t* yra lygus maksimaliam iteracijų skaičiui, tada iteracinis procesas stabdomas (ε yra maža teigiama konstanta). Priešingu atveju algoritmas kartojamas nuo 3 žingsnio.

MDS skaičiavimo vienos iteracijos skaičiavimų sudėtingumas SMACOF algoritme yra $O(nm^2)$, čia *n* – parametrų skaičius, *m* – objektų skaičius. Jeigu analizuojame didelės apimties duomenų aibes, tai svarbus MDS faktorius yra skaičiavimo laikas. Yra daug būdų MDS skaičiavimo laikui sumažinti. Vienas

iš jų, sumažinti analizuojamos duomenų aibės objektų skaičių *m ir* gautą mažesnę duomenų aibę analizuoti daugiamačių skalių metodu. Duomenų aibės objektų skaičių *m* galime sumažinti klasterizuojant arba pasinaudojant vektorių kvantavimo metodais.

2.3.3. Vektorių atvaizdavimo kokybės matai

Norint įvertinti vizualizavimo kokybę yra skaičiuojamos projekcijos paklaidos. Siekiant kiekybiškai įvertinti panašumų išlaikymą (pavyzdžiui, atstumus, kaimyniškumo ryšius ir kt.) po duomenų transformavimo į mažesnės dimensijos erdvę, reikia naudoti kiekybinius skaitinius matus. Šios problemos sprendimas plačiai nagrinėjamas moksliniuose darbuose (Goodhill and Sejnowski 1996, Estévez *et al.* 2005, Bernatavičienė *et al.* 2006, Karbauskaitė and Dzemyda 2009, Marcinkevičius 2010).

Disertacijoje nagrinėjami trys vizualizavimo kokybės įvertinimo matai. Vienas iš jų yra *Konigo matas* (Konig 2000), kuris yra taikomas darbuose (Estévez *et al.* 2005, Karbauskaitė and Dzemyda 2009). Šį matą vadinsime Konigo topologijos išlaikymo matu. Jis parodo, kaip išlaikomas taškų kaimyniškumas pereinant iš daugiamatės erdvės į dvimatę erdvę. Antras matas yra *Spirmano koeficientas* (angl. *Spearman's rho*). Šis matas naudojamas norint nustatyti, kaip gerai išlaikomas atstumų (tarp visų taškų *n*-matėje erdvėje) eiliškumas, pereinant į mažesnės dimensijos *d*-matę erdvę (Karbauskaitė *et al.* 2007, A3), t. y. kaip gerai atitinkamų mažesnės dimensijos erdvės taškų porų atstumų eilės numeriai atitinka didesnės dimensijos erdvės taškų porų atstumų eilės numerius didėjimo tvarka surūšiuotose atstumų sekose. Trečiasis vizualizavimo kokybės matas – *MDS paklaida*. Ja stengiamasi išlaikyti atstumus pereinant iš didesnės į mažesnės dimensijos erdvę.

Konigo topologijos išlaikymo matas turi du valdymo parametrus – artimiausių kaimynų skaičius: μ ir ν, $\mu < \nu$. Kaimynams įvertinti dažniausiai naudojami Euklido atstumai.

Tarkime, kad:

- X_{ij}, $j = 1,..,\mu$, yra *n*-matės erdvės vektoriaus X_i artimiausi kaimynai, čia atstumai tarp X_i ir jo kaimynų tenkina nelygybę $\left\|X_i - X_{ij_1}\right\| < \left\|X_i - X_{ij_2}\right\|$, kai $j_1 < j_2$. μ – kaimynų skaičius.
- Y_{ij}, $j = 1,..,\nu$, *d*-matės erdvės vektoriaus Y_i artimiausi kaimynai. ν – kaimynų skaičius.
- $r_X(i,j)$ – vektoriaus X_i *j*-tojo artimiausio kaimyno X_{ij} eilės numeris analizuojamoje duomenų aibėje.
- $r_Y(i,j)$ – vektoriaus Y_i, atitinkančio vektorių X_i, *j*-tojo artimiausio kaimyno Y_{ij} eilės numeris.

Konigo topologijos išlaikymo matas *i*-tajam taškui ir *j*-tajam kaimynui apskaičiuojamas pagal formulę:

$$E_{\text{KM}}^{ij} = \begin{cases} 3, \text{ kai } r_X(i,j) = r_Y(i,j), \\ 2, \text{ kai } r_X(i,j) = r_Y(i,l),\ l \in (1,...,\mu),\ i \neq l, \\ 1, \text{ kai } r_X(i,j) = r_Y(i,t),\ t \in (\mu+1,...,\nu),\ \mu < \nu, \\ 0, \text{ kitais atvejais.} \end{cases} \tag{2.6}$$

Bendrasis Konigo matas E_{KM} apskaičiuojamas pagal formulę:

$$E_{\text{KM}} = \frac{1}{3\mu m} \sum_{i=1}^{\mu} \sum_{j=1}^{m} E_{\text{KM}}^{ij}. \tag{2.7}$$

Darbe (Estévez *et al.* 2005) siūlomos parametrų reikšmės $\mu = 4$ ir $\nu = 10$. Konigo mato reikšmių kitimas yra nuo 0 iki 1. Jeigu $E_{\text{KM}} = 0$, tai kaimyniškumas nėra išlaikomas, o jeigu $E_{\text{KM}} = 1$, tai kaimyniškumas yra išlaikomas.

Spirmano koeficientas apskaičiuojamas pagal formulę:

$$\rho_{\mathrm{Sp}}(r'_X, r'_Y) = 1 - \frac{6}{(m')^3 - m'} \sum_{k=1}^{m'} (r'_X(k) - r'_Y(k))^2, \qquad (2.8)$$

čia r'_X ir r'_Y n-mačių ir d-mačių duomenų vektorių (taškų) visų porų atstumų eilės numeriai, $m' = m(m-1)/2$. Įprastai $-1 \le \rho_{\mathrm{Sp}} \le 1$. Jeigu Spirmano koeficiento reikšmė lygi 1, tai reiškia, kad kaimyniškumas išlaikomas 100 %.

Trečias tyrimuose naudojamas vizualizavimo matas, MDS paklaida, skaičiuojama pagal formulę:

$$\hat{E}_{\mathrm{MDS}} = \sqrt{\frac{\sum_{i<j} (d(X_i, X_j) - d(Y_i, Y_j))^2}{\sum_{i<j} (d(X_i, X_j))^2}}. \qquad (2.9)$$

Paklaida $\hat{E}_{\mathrm{MDS}}$ naudojama vietoj paklaidos E_{MDS} (2.5), kadangi jos normalizavimo parametras $\sum_{i<j} (d(X_i, X_j))^2$ leidžia gauti rezultatus, nepriklausomus nuo artimumų matavimo skalės. Pagrindinė priežastis, kodėl naudojama $\hat{E}_{\mathrm{MDS}}$ vietoj kvadratinės paklaidos $\hat{E}^2_{\mathrm{MDS}}$, yra tai, kad $\hat{E}^2_{\mathrm{MDS}}$ beveik visada labai maža (<1) (Borg and Groenen 2005), todėl $\hat{E}_{\mathrm{MDS}}$ reikšmės lengviau atskiriamos.

2.4. Antrojo skyriaus apibendrinimas ir išvados

Šiame skyriuje yra atlikta vektorių kvantavimo ir vizualizavimo metodų, naudojamų daugiamačių duomenų vizualizavimui, analitinė apžvalga. Susisteminti bei išnagrinėti vektorių kvantavimo metodai be mokytojo (neuroninių dujų, K-vidurkių, K-vidurinių taškų metodai bei saviorganizuojantys neuroniniai tinklai) ir su mokytoju (vektorių kvantavimo mokymo algoritmai VKM1 ir VKM2), kurie apima visą klasę gimininingų algoritmų ir gali būti jungiami su vizualizavimo metodais, dar kitaip vadinamais projekcijos ar dimensijų mažinimo metodais. Darbe siūloma

naudoti neuroninių dujų metodą ir saviorganizuojančius neuroninius tinklus, kadangi jie grindžiami dirbtiniais neuroniniais tinklais, todėl gali būti taikomi naujų duomenų, kurie nebuvo naudojami tinklui mokyti, analizei. Be to, šie tinklai mokomi be mokytojo būdu, todėl tinka duomenims, kurių klasės nėra žinomos.

Taip pat išanalizuoti pagrindiniai daugiamačių duomenų projekcijos metodai (daugiamatės skalės, pagrindinių komponenčių analizė), kai didelės dimensijos duomenys transformuojami į mažesnės dimensijos erdvę taip, kad išliktų arba būtų atrastos „užslėptos“ analizuojamų duomenų savybės. Daugiamačiams duomenims vizualizuoti darbe siūloma taikyti dažniausiai naudojamą netiesinės projekcijos metodą – daugiamačių skalių metodą.

Didelės apimties duomenų analizei siūloma naudoti vektorių kvantavimo ir vizualizavimo metodų junginius, kurie leidžia sumažinti analizuojamų duomenų skaičių ir juos atvaizduoti plokštumoje. Junginiais gautų vaizdų kokybės vertinimui pasirinkti šie skaitiniai matai: Konigo matas, Spirmano koeficientas bei MDS paklaida. Jie leidžia kiekybiškai įvertinti panašumų išlaikymą po daugiamačių duomenų transformavimo į mažesnės dimensijos erdvę.

3

Vektorių kvantavimo ir projekcijos metodų jungimas

Šiame skyriuje analizuojami vektorių kvantavimo metodų (neuroninių dujų ir saviorganizuojančio neuroninio tinklo) ir vieno iš projekcijos metodo – daugiamačių skalių (MDS) – junginiai. Pradžioje detaliai išanalizuoti neuroninių dujų (ND) metodo ir saviorganizuojančio neuroninio tinklo (SOM) algoritmai. Pateiktos tyrimuose naudojamos ND ir SOM tinklo mokymo taisyklės, tinklo mokymo ypatybės. Taip pat pateikti ir aprašyti saviorganizuojančio neuroninio tinklo ir daugiamačių skalių bei neuroninių dujų ir daugiamačių skalių metodo jungimo būdai. Pasiūlyti nuoseklus ir integruotas junginiai, kai daugiamačiai vektoriai analizuojami neuroninių dujų metodu arba vektorių analizei taikomi saviorganizuojantys neuroniniai tinklai, o gauti kvantuoti vektoriai vizualizuojami daugiamačių skalių metodu. Kaip bus parodyta vėliau, integruoti junginiai leidžia gauti tikslesnę daugiamačių taškų projekciją plokštumoje. Žinoma, kad net nedidelis paklaidos sumažinimas gali iš esmės pakeisti taškų išsidėstymą plokštumoje. Skyriuje

pateikti rezultatai publikuoti darbuose (A4, A5, B3 ir B2). Eksperimentinių tyrimų rezultatai bei junginių lyginamoji analizė pateikta 4 skyriuje.

3.1. Neuroninių dujų metodas

Neuroninės dujos (angl. *neural gas*) – tai vienas iš kvantavimo metodų, pagrįstas neuroniniu tinklu (Martinetz and Schulten 1991). Metodas pavadintas neuroninių dujų metodu dėl neuronų (vektorių) dinamikos, t. y. mokymo procese jie pasklinda erdvėje kaip dujos (dujų molekulės homogeninėje aplinkoje juda chaotiškai (netvarkingai), pasklinda po visą erdvę), nėra jokios specifinės topologijos.

Tarkime, turime duomenų aibės matricą $X = \{X_1, X_2, ..., X_m\} = \{x_{ij}, i = 1,..., m,\ j = 1,..., n\}$, jos eilutės yra vektoriai $X_i \in R^n$, t. y. $X_i = (x_{i1}, x_{i2}, ..., x_{in}), i = 1,..., m$, čia x_{ij} yra *i*-tojo vektoriaus *j*-toji komponentė, n – komponenčių skaičius, m – analizuojamų vektorių skaičius. Neuroninių dujų (ND) metodu sukuriamas neuronų masyvas *M*. Neuronai – tai vektoriai, kurių dimensijų skaičius lygus *n*. Neuroninių dujų metode neuronų tinklas yra vienmatis $M = \{M_1, M_2, ..., M_N\}$, čia $M_k = \{m_{k1}, m_{k2}, ..., m_{kn}\}$, $k = 1,..., N$, N – neuronų skaičius. ND, kaip vieno iš kvantavimo metodų, tikslas – pakeisti neuronų reikšmes taip, kad jie atspindėtų analizuojamos duomenų aibės vektorių X_l, $l = 1,..., m$, savybes, t. y. mokymo pabaigoje neuronai tampa vektorių X_l kvantuotais vektoriais.

Prieš tinklo mokymą generuojamos atsitiktinės pradinės neuronų komponenčių reikšmės. Mokymo metu vienas po kito mokymo aibės X vektoriai pateikiami į tinklą nustatytą kiekį kartų. Kiekvienas vektorius į tinklą pateikiamas $\hat{e}$ kartų. Kadangi analizuojamų vektorių skaičius yra lygus *m*, tai mokymo iteracijų skaičius $t_{\max} = \hat{e} \times m$. Į tinklą pateikus vektorių X_l, $l \in \{1,..., m\}$, suskaičiuojamas Euklido atstumas nuo jo iki visų tinklo neuronų.

Neuroninių dujų algoritmas:

1. **Pradinių reikšmių nustatymas.**

   Nustatome neuronų skaičiaus N reikšmę, pradines parametrų λ_g, λ_f, E_g, E_f, t_{max}, naudojamų mokymo taisyklėse, reikšmes ir epochų skaičių (epocha – tai mokymo proceso dalis, kai visus analizuojamus duomenis pateikiame į tinklą vieną kartą). Parametrų numatytosios reikšmės yra: $\lambda_g = N/2$, $\lambda_f = 0{,}01$, $E_g = 0{,}5$, $E_f = 0{,}01$, (Alhoniemi *et al.* 2000), $t_{max} = \hat{e} \times m$.

   Nustatome neuronų $M_1, M_2, ..., M_N$ komponenčių pradines reikšmes ($M_k = \{m_{k1}, m_{k2}, ..., m_{kn}\}, k = 1, ..., N$).

2. **Neuroninio tinklo mokymas.**

   Apskaičiuojame Euklido atstumus tarp visų neuronų ir duomenų vektoriaus X_l, $l \in \{1, ..., m\}$: $\|M_1 + X_l\|$, $\|M_2 + X_l\|$, ..., $\|M_N + X_l\|$.

   Gauti atstumai surūšiuojami didėjimo tvarka. Gaunama neuronų seka $W_1, W_2, ..., W_N$ tokia, kad $\|W_1 + X_l\| \le \|W_2 + X_l\| \le, ..., \le \|W_N + X_l\|$.

   Po surūšiavimo iki pirmo neurono W_1 atstumas yra mažiausias. Jis vadinamas neuronu nugalėtoju.

   Neuronų W_k, $k = 1, 2, ..., N$, komponenčių reikšmės yra keičiamos pagal taisyklę:

$$W_k(t+1) = W_k(t) + E(t) \cdot h_\lambda \cdot (X_l - W_k(t)), \qquad (3.1)$$

   čia: $E(t) = E_g (E_f / E_g)^{(t/t_{max})}$, $h_\lambda = e^{-(k-1)/\lambda(t)}$, $\lambda(t) = \lambda_g (\lambda_f / \lambda_g)^{(t/t_{max})}$.

3. 2 žingsnį kartojame visiems analizuojamiems vektoriams, nurodytą epochų skaičių.

Neuroninių dujų (ND) algoritmo pseudo kodas pateiktas 3.1 paveiksle.

function NG_training (*X*, *M*, $t_{\max}$, *N*, λ_{g} , λ_{f} , E_{g} , E_{f})

// įvestis: *X* – duomenų aibė, *M* – pradiniai neuronai, $t_{\max}$ – mokymo iteracijų skaičius,
// *N* – neuronų skaičius,
// λ_{g} , λ_{f} , E_{g} , E_{f} – parametrai (konstantos)
// išvestis: *W* – neuronai
BEGIN
FOR $t=0$ **TO** $t_{\max}$
  FOR $l=1$ **TO** *m* // *duomenų aibės vektorius* X_l *pateikiamas į neuroninį tinklą*
    FOR $i=1$ TO *N*
      $\|M_i - X_l\| := \sqrt{\sum_{p=1}^{n}(m_{ip} - x_{lp})^2}$ // *skaičiuojamas Euklido atstumas*
    END
    $\{W_1, W_2, ..., W_N\} := SORT\_ASCENDING(\|M_1 - X_l\|, ..., \|M_N - X_l\|)$
    // *čia* $W_k \in \{M_1, M_2, ..., M_N\}$, $k=1,...,N$, *ir* $\|W_1 - X_l\| \le, ..., \le \|W_N - X_l\|$
    $E(t) := E_{\mathrm{g}}(E_{\mathrm{f}} / E_{\mathrm{g}})^{(t/t_{\max})}$, $\lambda(t) := \lambda_{\mathrm{g}}(\lambda_{\mathrm{f}} / \lambda_{\mathrm{g}})^{(t/t_{\max})}$
    FOR $k=1$ TO *N*
      $h_\lambda := e^{-(k-1)/\lambda(t)}$,
      $W_k(t+1) := W_k(t) + E(t)\, h_\lambda\, (X_l - W_k(t))$ // *ND mokymo taisyklė*
    END
  END // *visų vektorių peržiūrėjimo pabaiga*
END // *mokymo pabaiga*
RETURN *W*
END

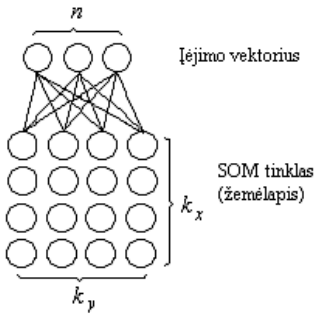
3.1 pav. *ND algoritmo pseudo kodas*

3.2. Saviorganizuojantys neuroniniai tinklai

Saviorganizuojančius neuroninius tinklus (žemėlapius) (SOM) (angl. *self-organizing maps*) T. Kohonen (2001) pradėjo tyrinėti apie 1982 metus. Pagal pradininko pavardę jie dar vadinami Kohoneno neuroniniais tinklais arba Kohoneno saviorganizuojančiais žemėlapiais. SOM tinklai yra nagrinėjami daugelio pasaulio mokslininkų bei plačiai taikomi įvairiose praktinėse srityse.

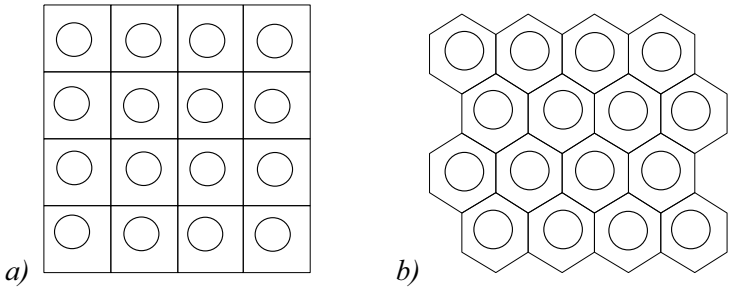
Šio tipo neuroninių tinklų pavadinimas kilo iš to, kad saviorganizuojantis žemėlapis, naudodamas mokymo aibę, pats save sukuria (save organizuoja). SOM tinklo tikslas – išlaikyti duomenų kaimyniškumus, t. y. taškai, esantys

arti įėjimo vektorių erdvėje, turi būti atvaizduojami arti vieni kitų ir SOM žemėlapyje. SOM žemėlapiai naudojami ir daugiamačiams duomenims klasterizuoti ir juos vizualizuoti, t. y. rasti projekcijas mažesnės dimensijos erdvėje, įprastai plokštumoje.



3.2 pav. *Dvimačio SOM tinklo schema*

Saviorganizuojantis neuroninis tinklas yra neuronų $M=\{M_1, M_2, ..., M_N\}$, čia $M_k=\{m_{k1}, m_{k2}, ..., m_{kn}\}$, $k=1,...,N$, $N-$ neuronų skaičius, paprastai išdėstytų dvimačio tinklelio (lentelės) mazguose, masyvas. Dvimačio neuroninio tinklo schema pateikta 3.2 paveiksle. Kiekvienas žemėlapio neuronas sujungtas su kiekvienu įėjimo vektoriumi (3.2 paveikslas, kad jo neperkrautume, pavaizduotos tik pirmos žemėlapio eilutės jungtys su įėjimo vektoriais). Galima stačiakampė (3.3 a paveikslas) arba šešiakampė (3.3 b paveikslas) tinklo struktūra.



3.3 pav. *SOM tinklo struktūra: a) stačiakampė (ortogonali); b) šešiakampė (heksagonali)*

Tyrimuose nagrinėsime stačiakampę SOM tinklo struktūrą, kuri yra dvimačių neuronų $M=\{M_{ij},\ i=1,...,k_x,\ j=1,...,k_y\}$ masyvas, čia $M_{ij}=(m_1^{ij},m_2^{ij},...,m_n^{ij})$, k_x yra lentelės eilučių skaičius, k_y – stulpelių skaičius. Bendras neuronų skaičius $N=k_x\times k_y$. Pagrindinis šio kvantavimo metodo tikslas yra pakeisti kvantuotų vektorių (neuronų) reikšmes taip, kad jie atspindėtų analizuojamų vektorių X_l, $l=1,...,m$, savybes. Mokymo pabaigoje neuronai tampa analizuojamos duomenų aibės vektorių X_l, kvantuotais vektoriais.

Saviorganizuojančio neuroninio tinklo algoritmas:

1. **Pradinių reikšmių nustatymas:**

   k_x yra lentelės eilučių skaičius, k_y – stulpelių skaičius. Bendras kvantuotų vektorių skaičius $N=k_x\times k_y$.

   $\hat{e}$ – tinklo mokymo epochų skaičius. Kiekvienas vektorius į tinklą pateikiamas $\hat{e}$ kartų, o mokymo iteracijų skaičius $t_{\max}=\hat{e}\times m$. Viena epocha – tai mokymo proceso dalis, kurios metu visi mokymo aibės vektoriai nuo X_1 iki X_m po vieną kartą pateikiami į tinklą nuosekliai arba atsitiktine tvarka.

   Įprastai pradinės kvantuotų vektorių $M_{ij}=(m_1^{ij},m_2^{ij},...,m_n^{ij})$, $\{i=1,...,k_x,\ j=1,...,k_y\}$ reikšmės parenkamos atsitiktinai ir $m_p^{ij}\in(0;1)$, $p=1,...,n$.

2. **Neuroninio tinklo mokymas:**

   Kiekviename mokymo žingsnyje vieną apmokymo aibės vektorių X_l, $l\in\{1,...,m\}$, pateikiame į tinklą.

Apskaičiuojame Euklido atstumus tarp duomenų vektoriaus X_l ir visų kvantuotų vektorių M_{ij}.

Neuronas $\hat{M}_c$, čia $c = \arg\min_{i,j}\{\|X_l - M_{ij}\|\}$, iki kurio Euklido atstumas nuo vektoriaus X_l yra mažiausias, pavadinamas neuronu (vektoriumi) nugalėtoju (angl. vector-winner).

Neurono (vektoriaus) M_{ij} komponentės keičiamos pagal taisyklę:

$$M_{ij}(t+1) = M_{ij}(t) + h_{ij}^{c}(t)(X_l - M_{ij}(t)), \tag{3.2}$$

čia h_{ij}^{c} – kaimynystės funkcija, $h_{ij}^{c}(t) \to 0$, kai $t \to \infty$.

3. 2 žingsnis kartojamas visiems analizuojamiems vektoriams, nurodytą epochų skaičių.

Įprastai kaimynystės funkcija

$$h_{ij}^{c}(t) = h_{ij}^{c}(\|r_c - r_{ij}\|, t), \tag{3.3}$$

čia $r_c \in R^2$ ir $r_{ij} \in R^2$ yra vektoriai, sudaryti iš neuronų $\hat{M}_c$ ir M_{ij} indeksų, t. y. jie nusako perskaičiuojamo neurono ir vektoriaus X_l neurono nugalėtojo vietą SOM tinkle. Didėjant atstumui $\|r_c - r_{ij}\|$, funkcijos $h_{ij}^{c}(t)$ reikšmė artėja prie nulio ($h_{ij}^{c}(t) \to 0$).

Saviorganizuojančio neuroninio tinklo (SOM) mokymo algoritmo pseudo kodas pateiktas 3.4 paveiksle.

SOM mokyme vienos iteracijos metu į tinklą pateikiamas vienas vektorius. Norint tinklą geriau išmokyti, tikslinga kiekvieną vektorių į tinklą pateikti kelis kartus. Galimi keli būdai:

- Įėjimo aibės vektoriai pateikiami iš eilės po vieną ciklišKai, t. y.,

pateikus visus vektorius, pirmasis vėl pateikiamas į tinklą ir t. t.

- Vektoriai pateikiami atsitiktine tvarka, t. y. vektoriai sumaišomi ir tada vienas po kito pateikiami į tinklą, kai visi jau pateikti, vėl sumaišomi ir vėl pateikiami į tinklą ir t. t.
- Į tinklą pateikiamas atsitiktinai parinktas įėjimo aibės vektorius, vėliau vėl atsitiktinai parenkamas kitas ir t. t.

Pirmais dviem atvejais visi vektoriai bus pateikti vienodą skaičių kartų, trečiu atveju nebūtinai. Antro būdo privalumas yra tai, kad išeliminuojama galimybė tinklui „prisiminti“ įėjimo vektorių pateikimo tvarką.

function SOM_training(*X*, *M*, $\hat{e}$, k_x, k_y**)**
// įvestis: *X* –duomenų aibė, *M* – pradiniai neuronai, $\hat{e}$ – tinklo mokymo epochų skaičius,
// k_x, k_y – eilučių ir stulpelių skaičius
// išvestis: M – neuronai
BEGIN
FOR $t=1$ **TO** $\hat{e}$
  FOR $l=1$ **TO** m // *duomenų aibės vektorius* X_l *pateikiamas į neuroninį tinklą*
    FOR $i=1$ TO k_x
      FOR $j=1$ TO k_y
        $\|M_{ij}-X_l\| := \sqrt{\sum_{p=1}^{n}(m_p^{ij}-x_{lp})^2}$ // *skaičiuojamas Euklido atstumas*
      END
    END
    $c := \arg\min_{i,j}\{\|X_l - M_{ij}\|\}$ // $\hat{M}_c$ – *vektoriaus* X_l *neuronas nugalėtojas*
    FOR $i=1$ TO k_x
      FOR $j=1$ TO k_y
        $M_{ij}(t+1) := M_{ij}(t) + h_{ij}^c(t)(X_l - M_{ij}(t))$ // *SOM mokymo taisyklė*
      END
    END
  END // *visų vektorių peržiūrėjimo pabaiga*
END // *mokymo pabaiga*
RETURN M
END

3.4 pav. *SOM algoritmo pseudo kodas*

Po SOM tinklo mokymo į tinklą pateikiami mokymo aibės arba nauji, dar tinklui „nematyti“, duomenų vektoriai. Randamas kiekvieno vektoriaus

neuronas nugalėtojas, t. y. vektoriai išsidėsto tarp žemėlapio (lentelės) elementų.

Bazinė SOM tinklo mokymo taisyklė išreiškiama (3.2) formule, tačiau galimos ir kitos naudojamų kaimynystės funkcijų h_{ij}^{c} (3.3) išraiškos. Tai yra euristinės funkcijos, todėl griežtų matematinių konvergavimo įrodymų nėra ir skirtingų mokymo taisyklių rezultatais gali būti šiek tiek kitokie žemėlapiai. Stabilios analizuojamų duomenų grupės įprastai išlieka visuose žemėlapiuose, tačiau gali būti duomenų, kurie priskiriami vis prie kitų grupių arba visai jų nesudaro.

Murtagh ir Hernindez-Pajares (1995) darbe pasiūlyta mokymo taisyklė, orientuota į mokymo proceso dalį – iteraciją. G. Dzemyda (2001) darbe dalinai modifikavo šią taisyklę, atsižvelgiant į mokymo proceso dalį – mokymo epochą. Ši taisyklė naudota ir darbuose (Dzemyda 2004, Dzemyda 2005). Viena iš galimų kaimynystės funkcijos h_{ij}^{c} išraiškų yra tokia (Dzemyda 2001):

$$h_{ij}^{c} = \frac{\alpha}{\alpha \eta_{ij}^{c} + 1}, \tag{3.4}$$

čia $\alpha = \max\left(\frac{\hat{e} + 1 - e}{\hat{e}}; 0{,}01 \right)$, $\hat{e}$ – prieš tinklo mokymą nustatytas viso mokymo epochų skaičius, e – vykdomos epochos numeris. Dydis η_{ij}^{c} vadinamas kaimynystės tarp neuronų $\hat{M}_c$ ir M_{ij} eile. Greta neurono nugalėtojo esantys neuronai vadinami pirmos eilės kaimynais, greta pirmos eilės kaimynų esantys neuronai, išskyrus jau paminėtus, – antros eilės kaimynais ir t. t.

Neurono nugalėtojo $\hat{M}_c$ kaimynystės funkcijos h_{ij}^{c} reikšmė yra maksimali. Ji mažėja didėjant epochų eilės numeriui e ir didėjant kaimynystės eilei η_{ij}^{c} neurono nugalėtojo atžvilgiu.

Kiekvienos mokymo epochos metu perskaičiuojami tie neuronai M_{ij}, kuriems galioja nelygybė:

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

Mokymo pradžioje perskaičiuojami tolimesni kaimynai, o vėliau tik artimesni.

3.3. ND ir SOM metodų kvantavimo paklaida

Baigus neuroninio tinklo mokymą SOM arba ND metodu, būtina nustatyti jo kokybę. Tam dažniausiai įvertinama *kvantavimo paklaida* (angl. *quantization error*). Kvantavimo paklaida parodo, kaip tiksliai tinklo neuronai prisiderina prie mokymo aibės vektorių. Jei vektoriaus X_l neuronas nugalėtojas būtų lygiai toks pat kaip pats vektorius X_l, tai paklaida būtų lygi 0. Dažniausiai duomenų vektorių skaičius viršija neuronų skaičių, todėl paklaida negali būti lygi 0. Kvantavimo paklaida (E_{QE}) – tai vidutinis atstumas tarp kiekvieno duomenų vektoriaus X_l ir jo vektoriaus nugalėtojo:

$$E_{\mathrm{QE}} = \frac{1}{m}\sum_{l=1}^{m}\left\|X_l - \hat{M}_c\right\|, \qquad (3.5)$$

čia $\hat{M}_c$ yra vektoriaus X_l neuronas nugalėtojas taikant SOM, ND metode $\hat{M}_c = W_1$.

3.4. Kvantavimo metodų ir daugiamačių skalių jungimo būdai

Neuroninės dujos ir saviorganizuojantys neuroniniai tinklai mokomi, jiems daug kartų pateikiant skirtingus objektus $X_1, X_2, ..., X_m$, nusakomais *n*-mačiais vektoriais. Kiekvienas įėjimo vektorius yra susijęs su artimiausiu neuronu. Dalis neuronų gali būti susiję su kai kuriais analizuojamais įėjimo vektoriais, o dalis ne. Neuronai, kurie yra susiję su analizuojamais įėjimo vektoriais, vadinami neuronais nugalėtojais. Dažniausiai neuronų nugalėtojų skaičius *r* yra

mažesnis nei neuronų skaičius N, $r \le N$. Neuronų skaičiaus parinkimo strategija pasiūlyta ir ištirta darbe (A1). SOM stačiakampės tinklo struktūros atveju galima nubraižyti lentelę, kurios langeliai atitinka neuronus, tačiau iš jos neaišku, kaip arti kaimyniniuose langeliuose esantys vektoriai yra *n*-matėje erdvėje. Kartais gautus rezultatus sudėtinga interpretuoti, todėl kyla idėja juos analizuoti vienu iš daugiamačių duomenų projekcijos metodu. Tuo tikslu gali būti naudojamas daugiamačių skalių metodas. Keletas SOM ir MDS junginių yra analizuoti darbe (Bernatavičienė *et al.* 2006). Neuronai nugalėtojai, kurie yra gaunami neuroninių dujų metodu, taip pat gali būti vizualizuojami daugiamačių skalių metodu (B3).

3.4.1. Nuoseklus ND, SOM ir MDS metodų junginys

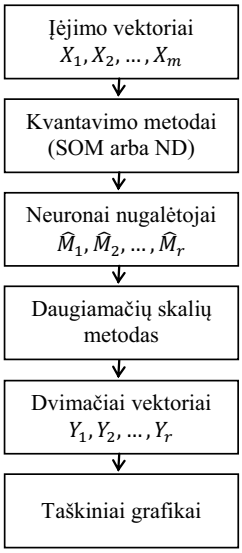
Nuosekliajame kvantavimo metodų ir daugiamačių skalių junginyje pradžioje vektoriai $X_1, X_2, ..., X_m$ yra kvantuojami SOM arba ND metodu, gauti vektoriai nugalėtojai analizuojami daugiamatėmis skalėmis. Gaunami dvimačiai vektoriai, kurie atvaizduojami Dekarto koordinačių sistemoje (3.5 paveikslas).

Nuoseklus SOM jungimas su daugiamatėmis skalėmis buvo tiriamas darbuose (Bernatavičienė *et al.* 2006, Dzemyda and Kurasova 2006). Darbuose (Estévez *et al.* 2005, A3, A4 ir B2) yra nagrinėjamas ne tik SOM, bet ir ND nuoseklusis junginys su MDS.

Vienas iš nuosekliojo junginio tikslų – pagerinti duomenų vizualizavimą, panaudojant saviorganizuojančius neuroninius tinklus. Kaip žinome, SOM gali pats vizualiai pateikti duomenis, pavyzdžiui, pagal unifikuotos atstumų matricos reikšmes (angl. *unified distance matrix*) (Kohonen 2001). Kaip bebūtų, iš SOM lentelės neaišku, kaip arti kaimyniniuose langeliuose esantys vektoriai yra *n*-matėje erdvėje. Tačiau pagrindinis nuosekliojo junginio tikslas – sumažinti skaičiavimo laiką, neprarandant vizualizavimo kokybės, atvaizduojant kvantuotus vektorius gautus taikant SOM arba ND metodą ir

juos vizualizuojant MDS metodu, lyginant su visos duomenų aibės vizualizavimo laiku, taikant tik MDS metodą.

Buvo analizuojamas neuroninių dujų ir saviorganizuojančio neuroninio tinklo jungimas su daugiamačių skalių metodu, kuriame projekcijos paklaida minimizuojama SMACOF algoritmu, pateiktu 2.3.2 skyrelyje.



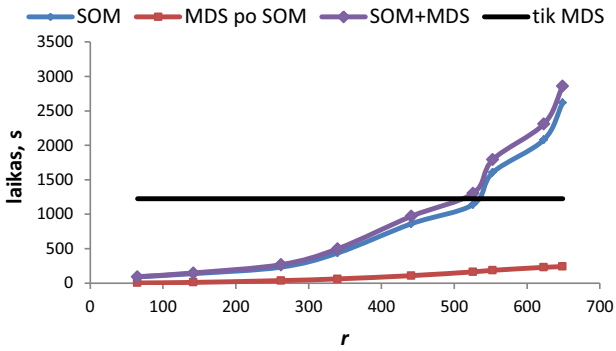
3.5 pav. *Neuronų nugalėtojų vizualizavimo daugiamačių skalių metodu schema (nuoseklus junginys)*

Kyla klausimas, kada tikslinga naudoti tik MDS, o kada jį jungti su vektorių kvantavimo metodais?

Pateikiamas pavyzdys. Tarkime, turime elipsoidų duomenų aibę [1338; 100] (Handl and Knowles 2010), kurią norime atvaizduoti plokštumoje. 3.6 paveiksle juoda linija pažymėtas skaičiavimo laikas, kai naudojant MDS metodą buvo vizualizuojami visi elipsoidų duomenų aibės vektoriai $(m=1338)$. SOM mokymas buvo kartojamas kelis kartus su įvairiomis neuronų N reikšmėmis. Buvo gautas įvairus neuronų nugalėtojų skaičius r. Mėlyna kreivė parodo SOM, o raudona tik MDS mokymo laiko priklausomybę nuo neuronų nugalėtojų skaičiaus r, kai ši duomenų aibė analizuojama SOM ir

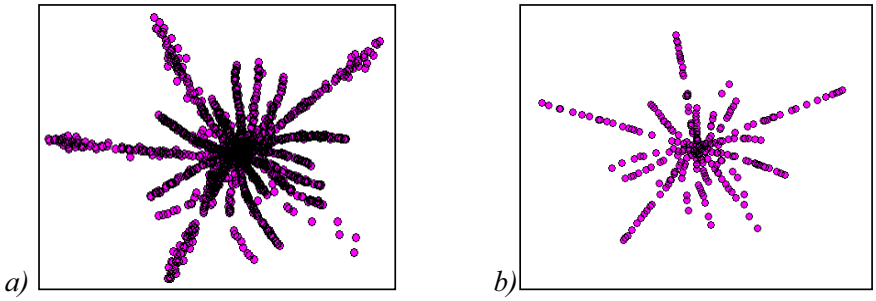
MDS junginiu. Violetinė kreivė parodo bendrą SOM ir MDS junginio laiką, kai vizualizuojami neuronai nugalėtojai, kurių skaičius lygus *r*.

Iš 3.6 paveikslo matome, kad tikslinga naudoti junginį, kai neuronų nugalėtojų skaičius *r* yra mažesnis negu 500, t. y. apytiksliai 37 % visų analizuojamų vektorių, kadangi junginio skaičiavimo laikas, lyginant su laiku, kai visa duomenų aibė vizualizuojama tik MDS, yra mažesnis. Kyla dar vienas klausimas, ar nenukenčia vizualizavimo kokybė?



3.6 pav. *Elipsoidų duomenų aibės vizualizavimo laikas naudojant tik MDS ir jo junginį su SOM*

3.7 a paveiksle pateikiamas visos elipsoidų duomenų aibės vaizdas, gautas daugiamačių skalių metodu, o 3.7 b paveiksle atvaizduoti tik 262 neuronai nugalėtojai, gauti naudojant SOM metodą.



3.7 pav. *Elipsoidų [1338; 100] duomenų aibės vizualizavimo rezultatai: a) atvaizduota visa duomenų aibė naudojant tik MDS; b) tik 262 neuronai nugalėtojai gauti SOM ir atvaizduoti naudojant MDS*

Matome, kad analizuojamos duomenų aibės elementų sumažinimas nepablogina vizualizavimo kokybės, yra išlaikoma duomenų struktūra, tačiau skaičiavimo laikas sumažėja iš esmės. Vizualizavus 262 neuronus nugalėtojus, taikant SOM ir MDS junginį, skaičiavimo laikas sumažėja 4,57 karto negu juos vizualizuojant tik daugiamačių skalių metodu.

Panašūs rezultatai yra gauti ir su kitomis analizuotomis duomenų aibėmis bei taikant ND ir MDS junginį. Taigi, kvantavimo metodų jungimas su MDS leidžia sumažinti skaičiavimo laiką neprarandant vizualizavimo kokybės, lyginant su rezultatais, gautais naudojant tik MDS metodą.

3.4.2. Integruotas ND, SOM ir MDS metodų junginys

Pastebėta, kad kai daugiamačių skalių MDS paklaida E_{MDS} minimizuojama iteraciniais metodais, labai svarbu tinkamai parinkti *d*-matės erdvės vektorių $Y_1, Y_2, ..., Y_m$ pradines reikšmes (mūsų atveju $d = 2$). MDS rezultatas priklauso nuo pradinių reikšmių parinkimo. Integruotas SOM ir MDS junginys, kaip naujas dvimačių vektorių inicializavimo būdas, pasiūlytas ir tyrinėtas darbuose (Kurasova 2005, Dzemyda and Kurasova 2006). Ten Samano algoritmas (Sammon 1969) yra naudojamas kaip vienas iš MDS grupės metodų. Disertacijoje yra siūloma tiek integruotame, tiek nuosekliajame junginyje vietoj SOM naudoti ND metodą, be to siūloma MDS paklaidą minimizuoti SMACOF algoritmu.

Integruoto junginio idėja yra ta, kad *n*-matės erdvės vektoriai $X_1, X_2, ..., X_m$ po SOM arba ND mokymo yra analizuojamas daugiamačių skalių metodu atsižvelgiant į SOM arba ND mokymosi eigą. Integruotas junginys susideda iš dviejų dalių, vykdomų pakaitomis:

1. SOM arba ND mokymas.
2. Neuronų nugalėtojų, gautų SOM arba ND metodu, dvimačių vektorių taškų skaičiavimas (paieška) daugiamačių skalių metodu.

Pagrindiniai žymėjimai ir apibrėžimai:

- Tarkime, kad mokymo aibę sudaro *n*-mačiai vektoriai $X_1, X_2, ..., X_m$ $(X_i = (x_{i1}, x_{i2}, ..., x_{in}),\ i = 1, ..., m)$. $Y_1, Y_2, ..., Y_m$ $(Y_i = (y_{i1}, y_{i2}),\quad i = 1, .., m)$ – dvimačių vektorių projekcijos plokštumoje.
- Neuroninis tinklas (SOM arba ND) apmokomas naudojant $\hat{e}$ mokymo epochų (*viena epocha* – tai SOM tinklo arba ND mokymo proceso dalis, kai visus vektorius tam tikra tvarka pateikiame tinklui po vieną kartą).
- Visas mokymo procesas, susidedantis iš $\hat{e}$ epochų, suskaidomas į lygius mokymo proceso blokus. Prieš neuroninio tinklo (SOM arba ND) apmokymą pasirenkame į kelis tokius blokus γ skaidysime visą mokymo procesą. Kiekvieną mokymo proceso bloką sudaro ν' mokymo epochų $(\hat{e} = \nu' \cdot \gamma)$. Raide q pažymėkime mokymo bloko, sudaryto iš ν' epochų, numerį $(q = 1, ..., \gamma)$.
- Pažymėkime neuronus nugalėtojus, gautus *q*-tajame mokymo bloke, $M_1^{(q)}, M_2^{(q)}, ..., M_r^{(q)}$, o jų dvimates projekcijas, apskaičiuotas daugiamačių skalių metodu – $Y_1^{(q)}, Y_2^{(q)}, ..., Y_r^{(q)}$ $(Y_i^{(q)} = (y_{i1}^{(q)}, y_{i2}^{(q)}),\quad i = 1, ..., r_q)$. Neuronų nugalėtojų skaičius r_q bus mažesnis arba lygus *m*.

Siūloma tokia integruoto junginio SOM arba ND ir MDS schema:

1 žingsnis: neuroninio tinklo mokymo pradžia ($q = 1$). Atlikus pirmąsias ν' mokymo epochas, mokymo procesas laikinai sustabdomas. Neuronai nugalėtojai $M_1^{(1)}, M_2^{(1)}, ..., M_{r_1}^{(1)}$, gauti po pirmojo mokymo proceso bloko $(q = 1)$, yra analizuojami daugiamačių skalių metodu. Pradinės dvimačių vektorių $Y_i^{(0)} = (y_{i1}^{(0)}, y_{i2}^{(0)}),\ i = 1, ..., r_1$, koordinatės gali būti parenkamos šiais būdais:

1. Atsitiktinai iš intervalo $(0;1)$.
2. Ant tiesės: $y_{i1}^{(0)} = i + 1/3$, $y_{i2}^{(0)} = i + 2/3$.
3. Priklausomai nuo dviejų didžiausių pagrindinių komponenčių (PK1 ir PK2).
4. Pagal vektorių komponentes, turinčias dvi didžiausias dispersijas.

Daugiamačių skalių metodu gaunamos neuronų nugalėtojų dvimačių vektorių projekcijos $Y_1^{(1)}, Y_2^{(1)}, ..., Y_r^{(1)}$.

Nuo 2 žingsnio iki γ: neuroninio tinklo mokymas tęsiamas toliau $(q = 2,..., \gamma)$. Neuronai nugalėtojai, gauti po q-tojo mokymo proceso bloko, yra analizuojami daugiamačių skalių metodu. Pradinės dvimačių vektorių $Y_1^{(q)}, Y_2^{(q)}, ..., Y_{r_q}^{(q)}$ koordinatės yra parenkamos atsižvelgiant į $(q-1)$-ajame bloke gautus rezultatus. Pažymėtina, kad bendru atveju $r_q \neq r_{q-1}$. Žemiau yra pateikiamas vienas iš galimų dvimačių vektorių pradinių koordinačių reikšmių priskyrimo būdų. Turime nustatyti kiekvieno dvimačio vektoriaus $Y_i^{(q)}$, atitinkančio neuroną nugalėtoją $M_i^{(q)}$, $i = 1,..., r_q$, pradines koordinates. Žingsnių seka yra tokia:

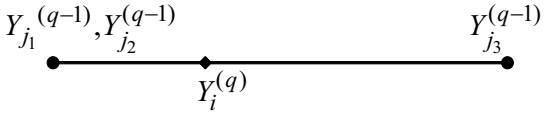
- Nustatykime, kurie vektoriai iš mokymo aibės $X_1, X_2, ..., X_m$ yra susiję su neuronu nugalėtoju $M_i^{(q)}$. Šiuos vektorius pažymėkime $X_{i_1}, X_{i2}, ...$ $(X_{i_1}, X_{i2}, ... \in \{X_1, X_2, ..., X_m\})$.
- Nustatykime, kurie $(q-1)$-ajame mokymo bloke gauti neuronai nugalėtojai yra susiję su $X_{i_1}, X_{i2}, ...$. Šiuos neuronus nugalėtojus pažymėkime $M_{j_1}^{(q-1)}$, $M_{j_2}^{(q-1)}, ...$ $(M_{j_1}^{(q-1)}, M_{j_2}^{(q-1)} ... \in \{M_1^{(q-1)}, M_2^{(q-1)}, ..., M_{r_{q-1}}^{(q-1)}\})$, ir jų dvimates projekcijas, gautas daugiamačių skalių metodu $Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ...$ $(Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ... \in \{Y_1^{(q-1)}, Y_2^{(q-1)}, ..., Y_{r_{q-1}}^{(q-1)}\})$.

Yra pasiūlyti du būdai dvimačių vektorių pradinių koordinačių priskyrimui:

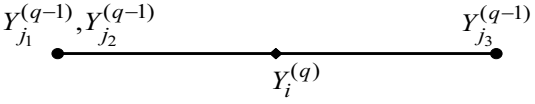
- **Pagal proporciją.** Pradinėms vektorių $Y_i^{(q)}$ koordinačių reikšmėms priskiriami vektorių $\{Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ...\}$ reikšmių vidurkiai. Iš 3.8 paveikslo matome, kad jeigu du taškai $Y_{j_1}^{(q-1)}$ ir $Y_{j_2}^{(q-1)}$ sutampa, tai taškas $Y_i^{(q)} = 1/3\left(Y_{j_1}^{(q-1)} + Y_{j_2}^{(q-1)} + Y_{j_3}^{(q-1)}\right)$ yra arčiau taško $Y_{j_1}^{(q-1)}$, negu $Y_{j_3}^{(q-1)}$.

- **Pagal vidurio tašką.** Tarp vektorių $\{Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ...\}$ gali būti sutampančių vektorių. Pradinėms vektorių $Y_i^{(q)}$ koordinačių reikšmėms priskiriami tik nesutampančių taškų reikšmių vidurkiai. Iš 3.9 paveikslo matome, kad $Y_i^{(q)} = 1/2\left(Y_{j_1}^{(q-1)} + Y_{j_3}^{(q-1)}\right)$.

Toliau daugiamačių skalių metodu skaičiuojamos neuronų nugalėtojų dvimatės projekcijos $Y_1^{(q)}, Y_2^{(q)}, ..., Y_{r_q}^{(q)}$ ($Y_i^{(q)} = (y_{i1}^{(q)}, y_{i2}^{(q)})$, $i = 1, ..., r_q$). Neuroninis tinklas mokomas, kol $q = \gamma$. Po γ-tojo bloko gaunamos *n*-mačių neuronų nugalėtojų $M_1^{(\gamma)}, M_2^{(\gamma)}, ..., M_{r_\gamma}^{(\gamma)}$ dvimatės projekcijos $Y_1^{(\gamma)}, Y_2^{(\gamma)}, ..., Y_{r_\gamma}^{(\gamma)}$, kurios atitinka *n*-mačių vektorių $X_1, X_2, ..., X_m$ dvimates projekcijas. Gauti dvimačiai vektoriai $Y_1^{(\gamma)}, Y_2^{(\gamma)}, ..., Y_{r_\gamma}^{(\gamma)}$ atvaizduojami plokštumoje. Integruoto junginio schema pateikta 3.10 paveiksle.

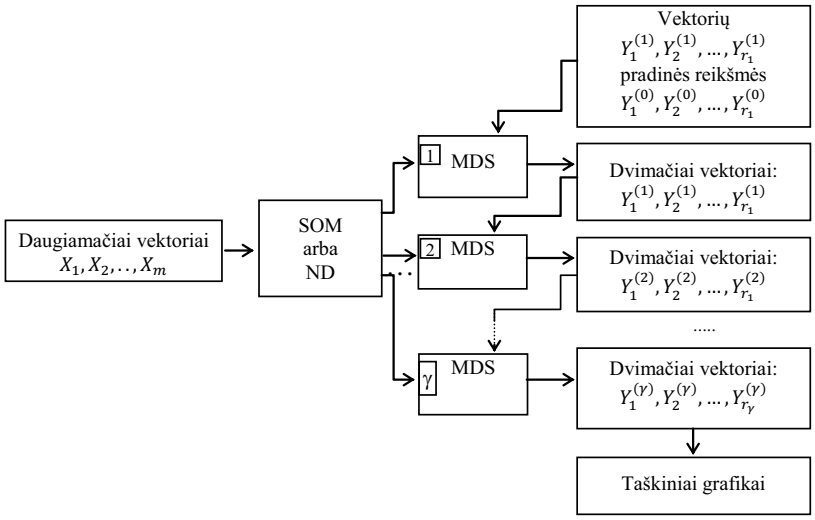


3.8 pav. *Pradinių dvimačių vektorių koordinačių priskyrimas pagal proporciją*



3.9 pav. *Pradinių dvimačių vektorių koordinačių priskyrimas pagal vidurio tašką*

Reikia pastebėti: jeigu analizuojamos duomenų aibės $X_1, X_2, ..., X_m$ po $(q-1)$-ojo mokymo bloko vektorių X_i ir X_j, $i, j \in \{1,...,m\}$, neuronas nugalėtojas yra tas pats, o po q-ojo mokymo bloko šių vektorių neuronai nugalėtojai yra skirtingi, tai pagal siūlomą integruoto junginio algoritmą, gautų taškų $Y_k = (y_{k1}, y_{k2})$, ir $Y_c = (y_{c1}, y_{c2})$, $k, c \in \{1,...,m\}$, koordinatės parenkamos tos pačios, t. y. šie taškai sutampa. Dėl šios priežasties prie sutampančių taškų koordinačių pridedame atsitiktinį skaičių ε, čia $\varepsilon < 0{,}01$.



3.10 pav. *SOM arba ND ir MDS integruoto junginio schema*

3.5. Trečiojo skyriaus apibendrinimas ir išvados

Šiame skyriuje detaliai išanalizuoti pagrindiniai disertacijoje naudojami vektorių kvantavimo metodai, grindžiami dirbtiniais neuroniniais tinklais – tai neuroninių dujų metodas bei saviorganizuojantys neuroniniai tinklai. Darbe pateiktas saviorganizuojančio neuroninio tinklo bei neuroninių dujų metodo ir daugiamačių skalių metodo jungimo teorinis pagrindas. Taip pat pateiktos ND metodo ir SOM tinklo mokymo taisyklės, naudojamos disertacijos tyrimuose. ND ir SOM tinklo mokymo kokybė bus vertinama pagal kvantavimo paklaidą.

Pasiūlyti nuoseklus ir integruotas ND ir daugiamačių skalių metodų junginiai, kaip alternatyva SOM ir MDS junginiams. Nuosekliajame junginyje po SOM ir ND gauti vektoriai nugalėtojai vizualizuojami daugiamačių skalių metodu. Integruotame junginyje po daugiamačių vektorių transformavimo į mažesnės dimensijos erdvę gauti dvimačiai vektoriai atvaizduojami plokštumoje daugiamačių skalių metodu, atsižvelgiant į SOM arba ND mokymosi eigą. Visas SOM ir ND mokymo procesas suskaidomas į pasirinktą blokų skaičių. Po kiekvieno SOM ir ND mokymo proceso bloko gauti vektoriai nugalėtojai analizuojami daugiamačių skalių metodu, parenkant dvimačių vektorių pradines koordinates atsižvelgiant į prieš tai buvusiame mokymo bloke gautas dvimates koordinates.

Šiame skyriuje pateikti nuosekliame bei integruoto junginio pirmajame mokymo bloke naudojami keturi dvimačių vektorių pradinių koordinačių parinkimo būdai: atsitiktinai iš intervalo (0; 1), ant tiesės, priklausomai nuo dviejų didžiausių pagrindinių komponenčių ir pagal didžiausias dispersijas.

Pasiūlytas dvimačių vektorių pradinių koordinačių reikšmių priskyrimas integruotame junginyje visuose, išskyrus pirmąjį, blokuose pagal vidurio tašką.

Kadangi teoriškai sudėtinga nustatyti, kuris iš junginių yra tinkamesnis duomenų analizei, todėl būtina atlikti eksperimentinius tyrimus, kurių metu tiriama kvantavimo paklaidos priklausomybė nuo neuronų, neuronų nugalėtojų bei mokymo epochų skaičiaus, vertinama duomenų atvaizdavimo, taikant pasiūlytus junginius, kokybė, bei atliekama jų lyginamoji analizė.

4

Eksperimentiniai tyrimai

Šiame skyriuje pateikiami vektorių kvantavimo metodų (neuroninių dujų ir saviorganizuojančių neuroninių tinklų) bei jų junginių su daugiamatėmis skalėmis eksperimentinių tyrimų rezultatai, publikuoti darbuose (A1–A5 ir B1–B3).

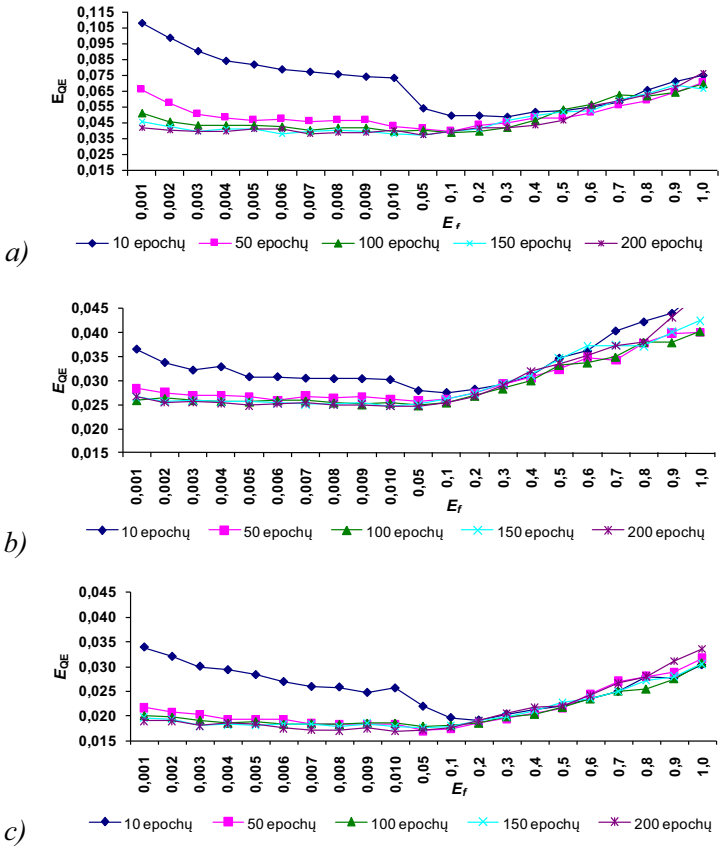
4.1. Mokymo taisyklės parametrų nustatymas neuroninių dujų metode

Šiame skyrelyje nagrinėjamas vienas iš vektorių kvantavimo metodų – *neuroninės dujos*, kuris pagrįstas neuroninio tinklo mokymu. Pateikti rezultatai publikuoti darbe (B1).

Neuroninių dujų metodu sukuriamas neuronų masyvas. Mokymo metu keičiamos neuronų komponentės, o mokymo pabaigoje neuronai tampa analizuojamos duomenų aibės vektorių kvantuotais vektoriais. Buvo tiriama kvantavimo paklaidos priklausomybė nuo kelių neuroninio tinklo mokymo parametrų ir atliekamų epochų skaičiaus parinkimo. Teorinių sprendinių

konvergavimo įrodymų nėra, todėl norint rasti tinkamiausią sprendinį, parametrai buvo parinkti empiriškai. Tyrimai buvo atlikti kelioms skirtingos prigimties duomenų aibėms: Fišerio irisams [149; 4], Automobilių [228; 7] ir kviečių duomenims [400; 12] (2.1 lentelė). Eksperimentiškai nustatytos tinklo mokymo parametrų reikšmės, kurioms esant gaunamos mažiausios kvantavimo paklaidos ir tinkamai parinktas atliekamų epochų skaičius. ND metode naudojama (3.1) mokymo taisyklė.

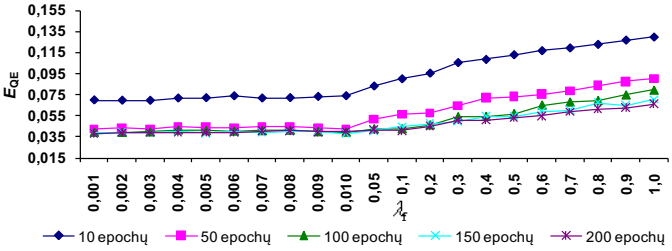
Kvantavimo paklaidos E_{QE} (3.5) priklausomybė nuo parametro E_{f}, kai neuronų skaičius $N=50$, o kitas parametras $\lambda_{\mathrm{f}} = 0{,}01$, pateikta 4.1 paveiksle.



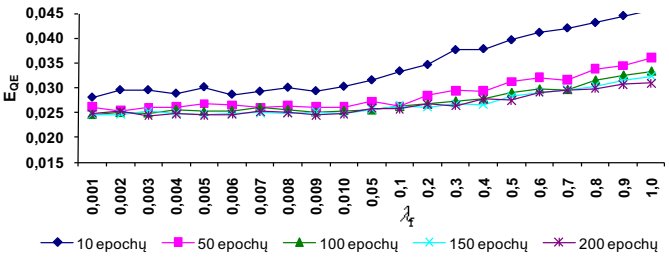
4.1 pav. *Kvantavimo paklaidos priklausomybė nuo parametro* E_{f}
a) irisų, b) kviečių, c) automobilių duomenims

Parametro E_f reikšmė buvo keičiama nuo 0,001 iki 1. Matome, kad visoms trims tirtoms duomenų aibėms E_{QE} mažėja, kai E_f reikšmė ne didesnė negu 0,1. Toliau didinat E_f reikšmę, kvantavimo paklaida didėja. Didžiausios paklaidos, lyginant įvairių epochų rezultatus, gaunamos, kai mokymo epochų skaičius lygus 10, kai $E_f < 0{,}1$. Didinant epochų skaičių, paklaida mažėja, tačiau epochų skaičių didinti daugiau nei 200 nėra prasmės, nes paklaidų skirtumas tampa nedidelis.

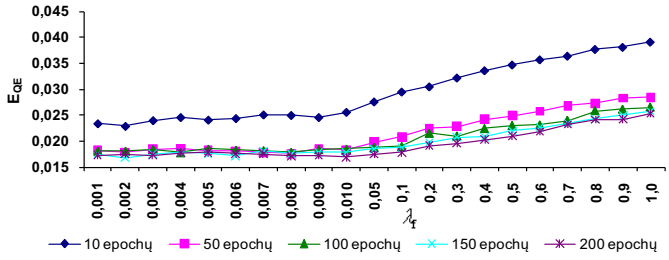
Kvantavimo paklaidos E_{QE} priklausomybė nuo parametro λ_f, kai neuronų skaičius $N = 50$, o kitas parametras $E_f = 0{,}01$, pateikta 4.2 paveiksle.



a)



b)



c)

4.2 pav. *Kvantavimo paklaidos priklausomybė nuo parametro* λ_f
a) irisų, b) kviečių, c) automobilių duomenims

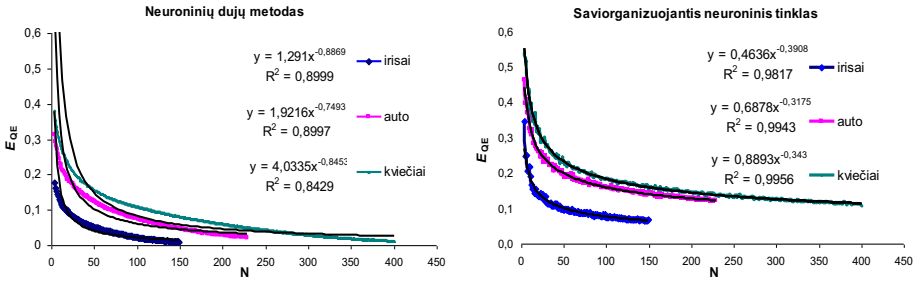
Parametro λ_f reikšmė buvo keičiama nuo 0,001 iki 1. Matome, kad visoms trims tirtoms duomenų aibėms kvantavimo paklaida taip pat mažėja, tačiau kai λ_f ne daugiau kaip 0,01. Toliau didinat λ_f reikšmę, kvantavimo paklaida pradeda didėti. Galima daryti išvadą, kad 200 mokymo epochų užtenka, norint gauti mažą kvantavimo paklaidos reikšmę.

Tyrimai parodė, kad kvantavimo paklaida priklauso nuo kelių tinklo mokymo parametrų. Atlikus eksperimentus nustatyta, kad parametro E_f reikšmė lygi 0,1, λ_f reikšmė lygi 0,01, o atliekamų epochų skaičius $\hat{e}=200$. Tolimesniuose tyrimuose parametrų E_g, λ_g reikšmės imtos, kaip nustatyta darbe (Alhoniemi *et al.* 2000), t. y. $E_g=0{,}5$, $\lambda_g=N/2$.

4.2. Kvantavimo paklaidos taikant SOM ir ND metodą

Šiame skyrelyje nagrinėjamos kvantavimo paklaidos E_{QE} (3.5) gaunamos naudojant saviorganizuojantį neuroninį tinklą ir neuroninių dujų metodą. Tyrimo tikslas – ištirti, kuriuo metodu gaunama mažesnė kvantavimo paklaida. Eksperimentai buvo atlikti naudojant tris realaus pobūdžio duomenų aibes: Fišerio irisus [149; 4], auto MPG [228; 7] ir kviečius [400; 12].

4.3 paveiksle pavaizduota kvantavimo paklaidų priklausomybė nuo neuronų skaičiaus N analizuojamoms skirtingų dimensijų duomenų aibėms. Mažiausia kvantavimo paklaida gaunama analizuojant irisų duomenis[149; 4], didesnė – automobilių [228; 7], didžiausia – kviečių [400; 12]. Be to pastebėta, kad paklaidos kitimo kreivės yra gana tiksliai aproksimuojamos laipsnine funkcija $y=ax^b$ su neigiamu laipsniu $b<0$, determinacijos koeficientas $R^2>0{,}98$ (SOM) arba $R^2>0{,}84$ (ND).



4.3 pav. *Kvantavimo paklaidos priklausomybė nuo neuronų skaičiaus N irisų, hepta ir kviečių duomenims*

4.3. Neuronų skaičiaus parinkimas

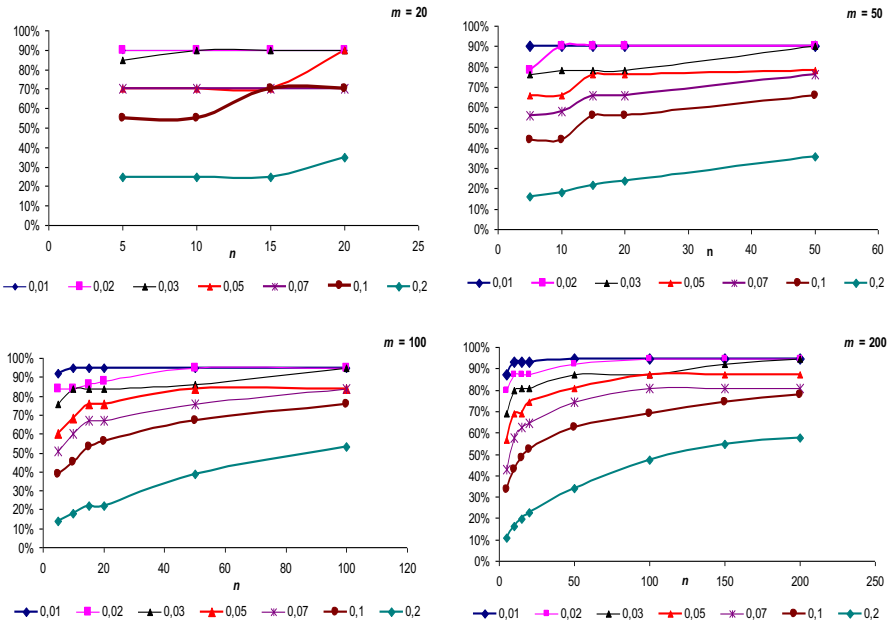
Šiame skyrelyje nagrinėjama neuronų skaičiaus parinkimo vektorių kvantavimo metoduose strategija. Analizuojami du neuroniniais tinklais grindžiami vektorių kvantavimo metodai: neuroninės dujos ir saviorganizuojantis neuroninis tinklas. Pateikti rezultatai publikuoti darbe (A1). Pasiūlytas būdas, pagal kurį parenkamas neuronų skaičius atsižvelgiant į analizuojamų duomenų specifiką.

Kvantavimo rezultatui įvertinti skaičiuojama kvantavimo paklaida E_{QE} (3.5). Tiek ND, tiek SOM metode ji yra mažiausia, kai $N=m$, t. y. neuronų skaičius N yra lygus analizuojamų vektorių skaičiui m. Tačiau nuo tam tikros $N=N'$ reikšmės paklaida skiriasi nežymiai palyginus su mažiausia. Tyrimuose nagrinėjamų vektorių kvantavimo metodų – saviorganizuojančių neuroninių tinklų ir neuroninių dujų metodo kvantuotų vektorių skaičius N vadinamas neuronų skaičiumi. Tyrimo tikslas – nustatyti neuronų skaičiaus N reikšmes atsižvelgiant į analizuojamų duomenų specifiką, kad kvantavimo paklaida būtų maža.

Eksperimentai buvo atlikti naudojant tris realaus pobūdžio duomenų aibes: Fišerio irisus [149; 4], hepta [212; 3], vėžio [683; 9], chainlink [1000; 3], elipsoidai [1338; 100]. Taip pat tyrime naudotos dirbtinai sugeneruotos

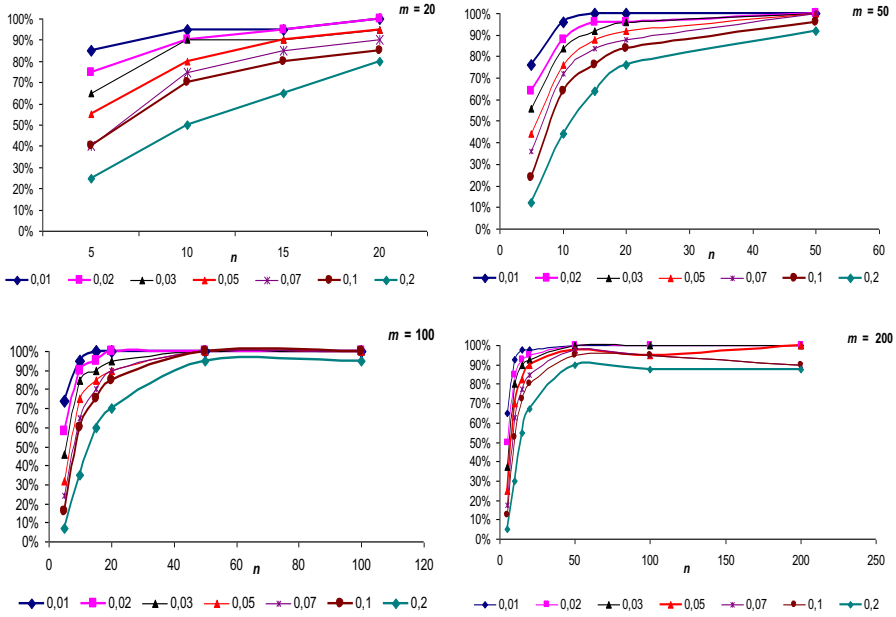
įvairaus didumo duomenų aibės, sudarytos iš vektorių, kurių komponentės yra dydžiai, tolygiai pasiskirstę intervale (0; 1).

Neuronų skaičiaus priklausomybės nuo duomenų vektorių dimensijų skaičiaus *n* tyrime naudoti dirbtinai sugeneruoti duomenys. Buvo sugeneruota 100 aibių, kurios sudarytos iš *m* vektorių, vektorių dimensija – *n*. Išmokomi SOM ir ND tinklai. Apskaičiuojamas kvantavimo paklaidų vidurkis. Eksperimentai atlikti, kai *m* yra lygu 20, 50, 100 ir 200, o *n* reikšmės yra nuo 5 iki atitinkamos *m* reikšmės. Neuronų skaičius $N = 4,...,m$. Fiksuotos neuronų skaičiaus $N' < N$ reikšmės. Esant šiam neuronų skaičiui, kvantavimo paklaidos reikšmė nuo mažiausios (kai $N = m$) skiriasi ne daugiau kaip $\varepsilon =$ 0,01, 0,02, 0,03, 0,05, 0,07, 0,1 ar 0,2. Apskaičiuota neuronų skaičiaus N' reikšmių procentinė dalis nuo visų neuronų skaičiaus $N = m$.



4.4 pav. *Neuronų skaičiaus procentinė dalis nuo analizuojamų vektorių skaičiaus SOM metode*

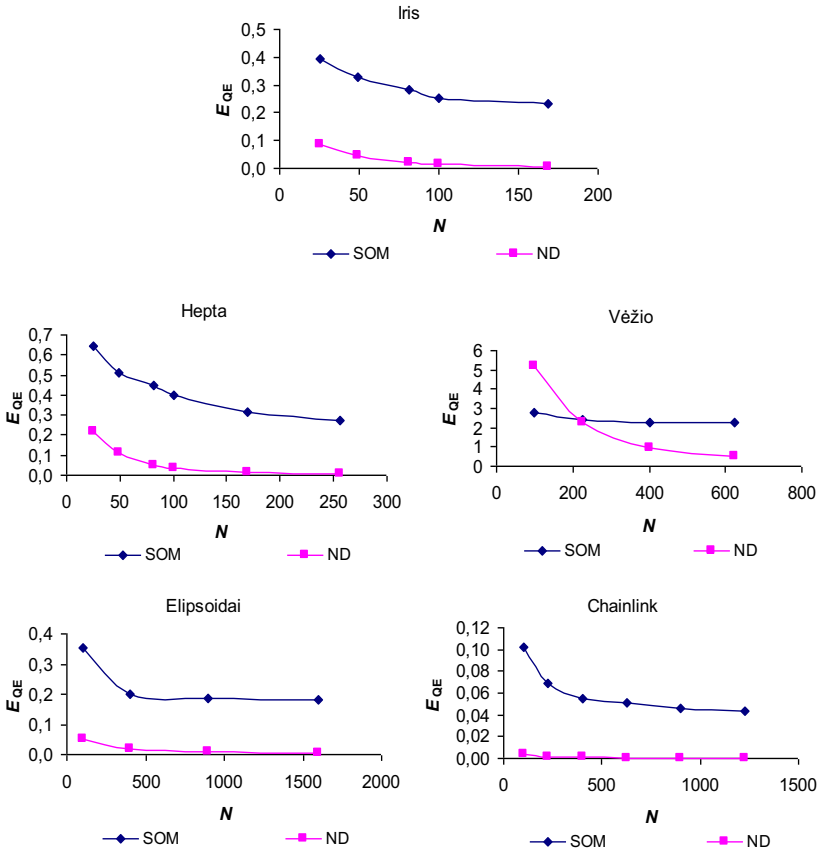
Neuronų skaičiaus N' priklausomybė nuo analizuojamų vektorių dimensijos n pavaizduota 4.4 ir 4.5 paveiksluose. Didesnė procentinė išraiška reiškia, kad gautume norimo tikslumo paklaidą, tinklą turime sudaryti iš didesnio neuronų skaičiaus.



4.5 pav. *Neuronų skaičiaus procentinė dalis nuo analizuojamų vektorių skaičiaus ND metode*

Esant fiksuotam paklaidų skirtumui ε iš 4.4 ir 4.5 paveikslų matome, kad didėjant n, procentinė neuronų dalis taip pat didėja. Tai reiškia, kad esant didesniam n, reikia imti daugiau neuronų, kad gautume gerus rezultatus kvantavimo paklaidos prasme. Palyginus SOM tinklo ir ND metodo rezultatus pastebėta, kad SOM metode galima atsisakyti žymiai didesnės dalies neuronų, lyginant su kvantavimo paklaida, kai $N=m$, neprarandant daug tikslumo. Pavyzdžiui, kai $m=50$, $n=20$, $\varepsilon=0{,}2$ SOM metode užtenka tik šiek tiek daugiau nei 20 % neuronų, o ND metode tam pačiam tikslumui pasiekti reikia beveik 80 % neuronų.

Toliau pateikti tyrimo rezultatai, kai vektoriai į tinklą pateikiami ne iš eilės, bet kiekvieno eksperimento metu ta pačia tvarka.



4.6 pav. *Kvantavimo paklaidos priklausomybė nuo neuronų skaičiaus N*

Šiame eksperimentiniame tyrime kiekvieno eksperimento metu pradinės neuronų reikšmės buvo parenkamos atsitiktinai, vektoriai į tinklą pateikiami ne iš eilės, o atsitiktine tvarka, bet visada ta pačia tvarka. Atlikta po 10 eksperimentų kiekvienam analizuojamam duomenų rinkiniui. Apskaičiuoti gautų paklaidų vidurkiai. Gauti rezultatai pateikiami 4.6 paveiksle.

Galima daryti išvadą, kad visoms analizuojamų duomenų aibėms, nepriklausomai nuo pradinių neuronų komponenčių parinkimo, kvantavimo

paklaida neuroninių dujų metodu yra mažesnė negu naudojant saviorganizuojantį neuroninį tinklą.

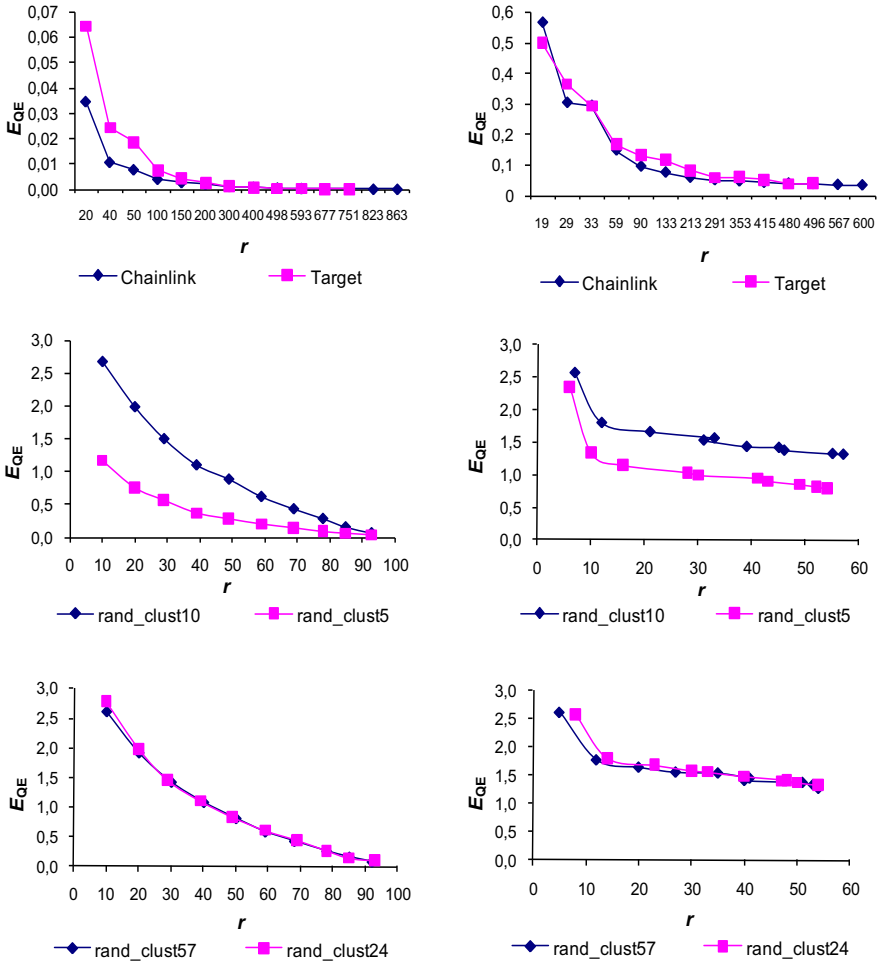
4.4. Kvantavimo paklaidos priklausomybė nuo neuronų nugalėtojų skaičiaus

Šiame skyrelyje pateikti eksperimentiniai rezultatai publikuoti darbuose (A2, A4 ir B2).

Tyrimo tikslas – ištirti, kokią įtaką daro neuronų nugalėtojų skaičius kvantavimo paklaidai. Neuronai nugalėtojai yra gaunami kvantuojant saviorganizuojančiu neuroniniu tinklu ir neuroninių dujų metodu. Nagrinėjama kvantavimo paklaidos priklausomybė nuo neuronų nugalėtojų skaičiaus. Norint įvertinti, kuris iš šių metodų yra tinkamesnis vektorių kvantavimui, buvo atliktas eksperimentinis tyrimas, kurio rezultatai pateikti 4.7 ir 4.8 paveiksluose.

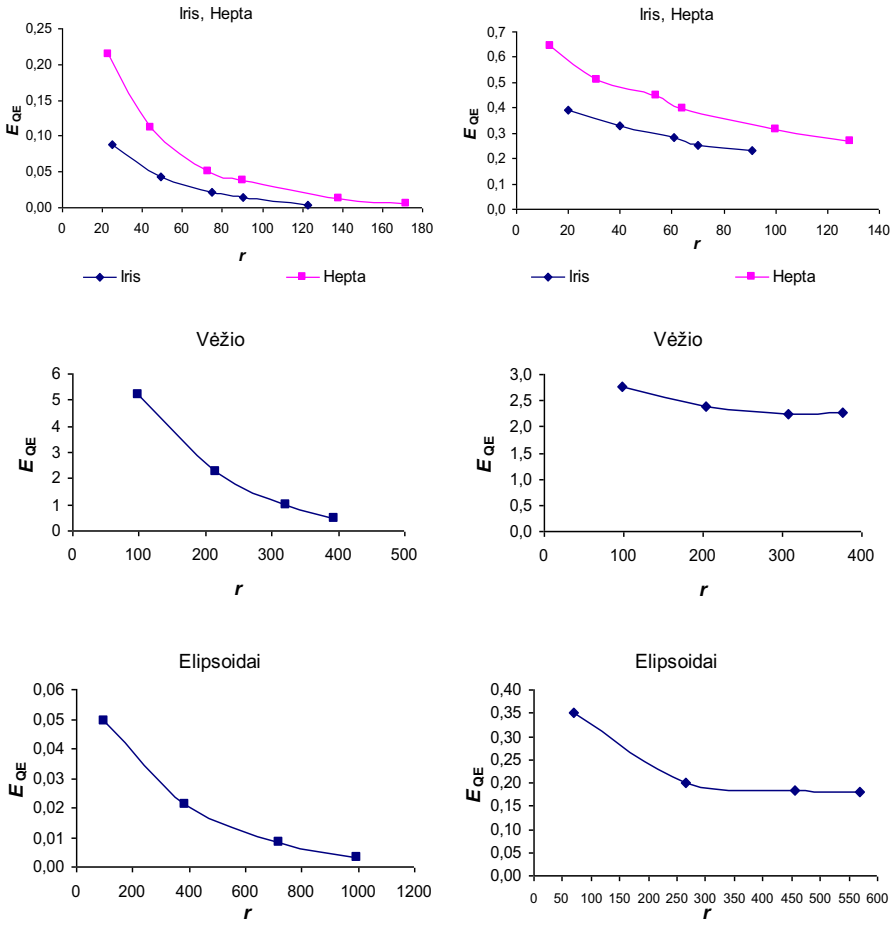
Tyrimuose naudotos šios duomenų aibės, turinčios specifinių savybių: Fišerio irisai [149; 4], hepta [212; 3], auto MPG [392; 7], target [770; 2], chainlink [1000; 3], ir rand_clust5 [100; 5], rand_clust10 [100; 10], rand_clust57 [100; 10] ir rand_clust24 [100; 10] (2.1 lentelė).

Eksperimentinio tyrimo metu reikėjo rasti tokį vektorių $M_1, M_2, ..., M_N$, $(N < m)$ rinkinį, kad SOM ir ND metodu gauti kvantuoti vektoriai M_i, $i = 1, ..., N$, atspindėtų vektorių X_l, $l = 1, ..., m$, savybes. Norint įvertinti kvantavimo kokybę buvo skaičiuojama kvantavimo paklaida E_{QE} (3.5). Ji parodo skirtumą tarp analizuojamų vektorių $X_1, X_2, ..., X_m$ ir kvantuotų vektorių (neuronų nugalėtojų) $\hat{M}_1, ..., \hat{M}_r$, čia r neuronų nugalėtojų skaičius. Kuo neuronų nugalėtojų yra daugiau, tuo kvantavimo paklaida mažesnė (4.7 ir 4.8 paveikslai).



4.7 pav. *Kvantavimo paklaidos priklausomybė nuo neuronų nugalėtojų skaičiaus r: ND (kairėje) ir SOM (dešinėje) (analizuotos chainlink, target, rand_clust10, rand_clust5, rand_clust57 ir rand_clust24 duomenų aibės)*

Iš 4.7 ir 4.8 paveikslų matome, kad neuroninių dujų metodu gauta kvantavimo paklaida yra žymiai mažesnė negu saviorganizuojančiu neuroniniu tinklu, kai neuronų nugalėtojų skaičius mažai skiriasi.

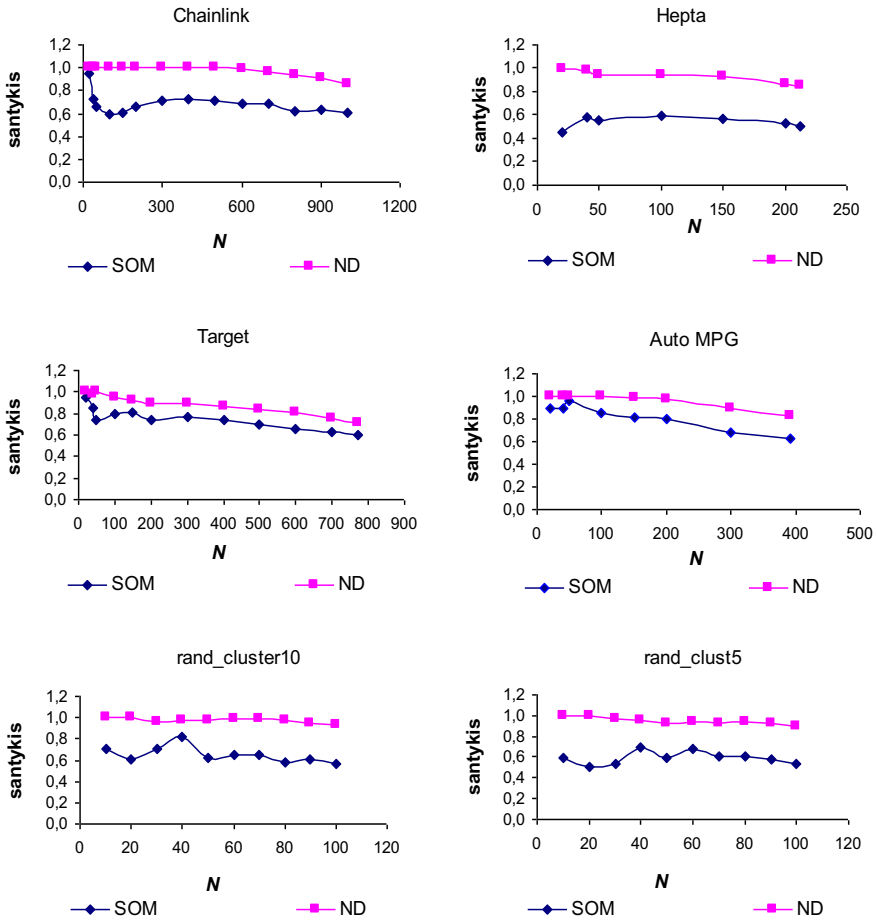


4.8 pav. *Kvantavimo paklaidos priklausomybė nuo neuronų nugalėtojų skaičiaus r: ND (kairėje) ir SOM (dešinėje) (analizuotos iris, hepta, elipsoidų ir vėžio duomenų aibės)*

4.9 paveiksle parodyta, kiek neuronų tampa neuronais nugalėtojais. Iš šio paveikslo matome, kad santykis tarp neuronų ir neuronų nugalėtojų gautų naudojant neuroninių dujų metodą yra didesnis negu naudojant saviorganizuojantį neuroninį tinklą. Neuroninių dujų metodu apie 80 % neuronų tampa neuronais nugalėtojais, o SOM tik apie 50 %.

Atlikto tyrimo rezultatai rodo, kad neuroninių dujų metodas yra tinkamesnis vektoriams kvantuoti, o saviorganizuojantys neuroniniai tinklai

yra labiau tinkami duomenims klasterizuoti. Tai patvirtina ir atlikta daugiamačių duomenų projekcijų plokštumoje kokybės analizė (4.8.2 skyrelis).



4.9 pav. *Neuronų ir neuronų nugalėtojų santykis ND ir SOM metodu*

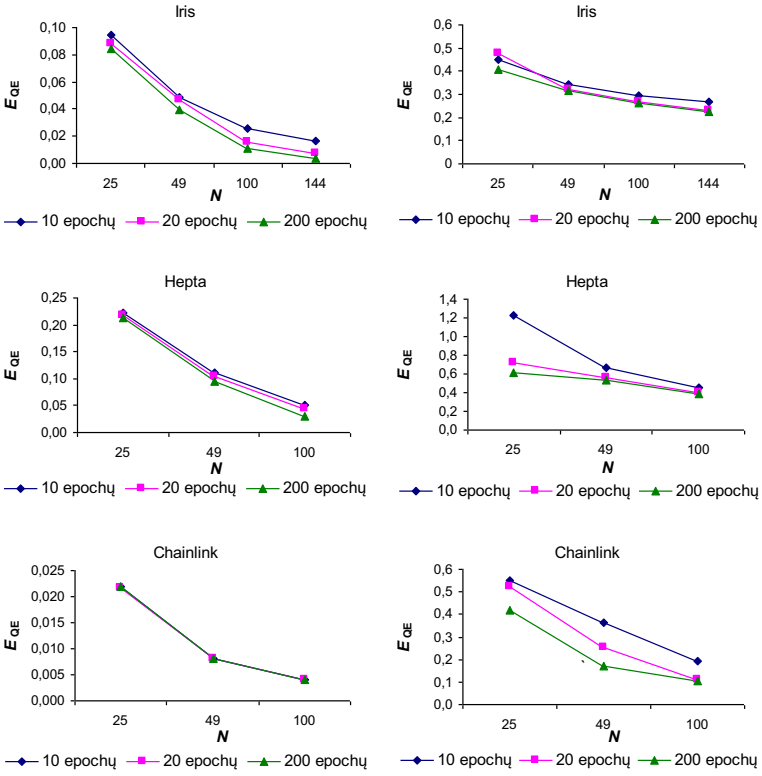
4.5. Kvantavimo paklaidos priklausomybė nuo mokymo epochų skaičiaus

Tyrimo tikslas – ištirti, kokią įtaką daro mokymo epochų skaičius kvantavimo paklaidai. Buvo atlikti tyrimai, kaip kvantavimo rezultatams daro

įtaką atliekamų epochų skaičius. Eksperimentinio tyrimo rezultatai pateikti 4.10 ir 4.11 paveiksluose.

Tyrimuose naudotos šios duomenų aibės, turinčios specifinių savybių: Fišerio irisai [149; 4], hepta [212; 3] auto MPG [392; 7], target [770; 2], chainlink [1000; 3], ir rand_clust5 [100; 5], rand_clust10 [100; 10], rand_clust57 [100; 10] ir rand_clust24 [100; 10] (2.1 lentelė).

Atlikti tyrimai parodė, kad didinant atliekamų epochų skaičių, kvantavimo paklaida mažėja tiek ND, tiek SOM metodu (4.10 paveikslas). Su visais analizuojamais duomenimis kvantavimo paklaida labiau skiriasi, kai yra atliekama 10 arba 200 epochų, tačiau atliekant 20 arba 200 epochų, paklaida beveik nesiskiria.

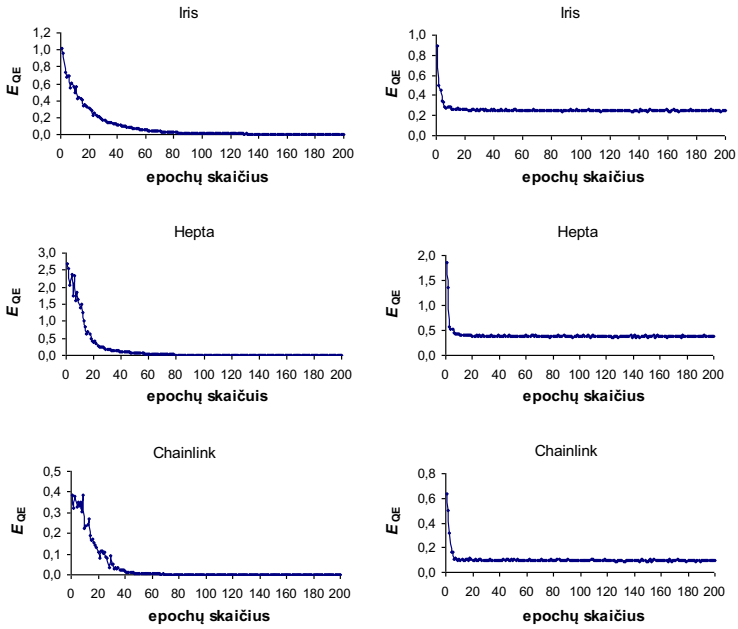


4.10 pav. *Kvantavimo paklaidos priklausomybė nuo neuronų skaičiaus: ND (kairėje) ir SOM (dešinėje)*

Atlikus tyrimus paaiškėjo, kad neuroninių dujų metodu gauta kvantavimo paklaida žymiai mažesnė negu gauta saviorganizuojančiu neuroniniu tinklu, kai atliekamų epochų skaičius yra 10, 20 ir 200. ND metodu pakanka atlikti apie 20 epochų.

Galime daryti išvadą, kad neuroninių dujų metodas kvantavimo paklaidos prasme yra stabilesnis negu saviorganizuojantis neuroninis tinklas, priklausomai nuo atliekamos epochos.

Toliau buvo tiriama, kaip keičiasi kvantavimo paklaida kiekvienos atliktos epochos metu, kai pradinės neuronų reikšmės yra atsitiktinės, o atliekamų epochų skaičius $\hat{e}=200$.



4.11 pav. *Kvantavimo paklaidos priklausomybė nuo epochų skaičiaus (pradinės neuronų komponentės atsitiktinės, $N=100$, $\hat{e}=200$):*
ND (kairėje) ir SOM (dešinėje)

Atlikta po 10 eksperimentų, kai pradinės neuronų komponentės parenkamos atsitiktinai, kiekvienam analizuojamam duomenų rinkiniui ir paskaičiuotas gautų paklaidų vidurkis. Iš 4.11 paveikslo matome, kad

kvantavimo paklaida per pirmąsias 5 epochas yra didesnė, o paskui stabilizuojasi ir toliau beveik nesikeičia iki 200 epochos SOM metodu. Taikant ND metodą kvantavimo paklaida nuo pirmos epochos yra didesnė negu SOM metodu ir stabilizuojasi tik apie 75 epochą, kuri ženkliai mažesnė negu SOM metodu. Iš 4.11 paveikslo matome, kad kvantavimo paklaida visiems analizuojamiems duomenims SOM metodu yra didesnė negu ND metodu.

4.6. Vektorių į tinklą pateikimo tvarkos tyrimas

Ištyrus neuronų skaičiaus parinkimo strategiją dviejuose vektorių kvantavimo metoduose – saviorganizuojančiame neuroniniame tinkle bei neuroninių dujų metode ir tinkamai parinkus atliekamų epochų skaičių, iškilo būtinybė išsiaiškinti, ar vektorių į tinklą pateikimo tvarka daro įtaką kvantavimo paklaidai. Eksperimentai buvo atlikti šioms duomenų aibėms: Fišerio irisai [149; 4], hepta [212; 3], vėžio [683; 9], chainlink [1000; 3] ir elipsoidai [1338; 100].

Šio tyrimo tikslas – nustatyti, kuris vektorių pateikimo į tinklą būdas yra tinkamiausias naudojant neuroninių dujų metodą ir saviorganizuojančius neuroninius tinklus.

Kvantavimo paklaidos skaičiuotos naudojant šiuos vektorių į tinklą pateikimo būdus:

1. **būdas** Kiekvieno eksperimento metu pradinės neuronų reikšmės parenkamos atsitiktinai. Vektoriai į tinklą pateikiami ne iš eilės, o atsitiktine, bet visada ta pačia tvarka.
2. **būdas** Eksperimento metu pradinės neuronų reikšmės parenkamos atsitiktinai, bet visada tos pačios. Vektoriai į tinklą pateikiami iš eilės.

3. būdas Kiekvieno eksperimento metu pradinės neuronų reikšmės parenkamos atsitiktinai, bet visada tos pačios. Vektoriai į tinklą pateikiami ne iš eilės, o atsitiktine tvarka.

Gautos kvantavimo paklaidos priklausomybės nuo pradinių neuronų reikšmių parinkimo, kai į tinklą vektoriai pateikiami ne iš eilės, bet visada tie patys (1 būdas), pateiktos 4.1–4.5 lentelėse. Atliekamų epochų skaičius $\hat{e}=25$.

4.1 lentelė. *Kvantavimo paklaida ir jos pasikliautinasis intervalas iris duomenų aibei, 1 būdas*

Metodas	N	E_{QE}	α	Pasikliautinasis intervalas	
				int-	int+
SOM	25	0,3937	0	0,3937	0,3937
	49	0,3295	0	0,3295	0,3295
	81	0,2815	0	0,2815	0,2815
	100	0,2541	0	0,2541	0,2541
ND					
	25	0,0879	0	0,0879	0,0879
	49	0,0440	0	0,0440	0,0440
	81	0,0217	0	0,0217	0,0217
	100	0,0141	0	0,0141	0,0141

4.2 lentelė. *Kvantavimo paklaida ir jos pasikliautinasis intervalas hepta duomenų aibei, 1 būdas*

Metodas	N	E_{QE}	α	Pasikliautinasis intervalas	
				int-	int+
SOM	25	0,6459	0	0,6459	0,6459
	49	0,5110	0	0,5110	0,5110
	81	0,4499	0,0019	0,4480	0,4518
	100	0,3976	0	0,3976	0,3976
	169	0,3147	0,0044	0,3103	
ND					
	25	0,2149	0	0,2149	0,2149
	49	0,1113	0	0,1113	0,1113
	81	0,0502	0	0,0502	0,0502
	100	0,0376	0	0,0376	0,0376
	169	0,0132	0,0008	0,0125	0,0140

Atlikta po 10 eksperimentų kiekvienai analizuojamų duomenų aibei. Apskaičiuoti gautų paklaidų vidurkiai ir jų pasikliautinieji intervalai „int-“ ir

„int+“, čia „int-“ = vidurkis – α , „int+“ = vidurkis + α . α – didžiausia reikšmė, kuria gali skirtis nustatytas vidurkis nuo realiai esančio, esant pasirinktam pasikliovimo lygmeniui (analizuojamu atveju 0,95).

Iš 4.1 ir 4.2 lentelių matome, kad naudojant SOM ir ND metodus α reikšmė beveik visada lygi 0, vadinasi 10 eksperimentų apskaičiuotų kvantavimų paklaidų vidurkis yra toks pat. Didinant neuronų skaičių α reikšmė ne visada lygi 0 (4.2–4.5 lentelėse).

4.3 lentelė. *Kvantavimo paklaida ir jos pasikliautinasis intervalas vėžio duomenų aibei, 1 būdas*

Metodas	N	E_{QE}	α	Pasikliautinasis intervalas	
				int-	int+
SOM	100	2,7660	0	2,7660	2,7660
	225	2,3883	0,0084	2,3799	2,3967
	400	2,2432	0,0098	2,2334	2,2530
	625	2,2674	0,0065	2,2609	2,2739
ND					
	100	5,2089	0	5,2089	5,2089
	225	2,2779	0	2,2779	2,2779
	400	0,9593	0	0,9593	0,9593
	625	0,4780	0,0074	0,4706	0,4853

4.4 lentelė. *Kvantavimo paklaida ir jos pasikliautinasis intervalas chainlink duomenų aibei, 1 būdas*

Metodas	N	E_{QE}	α	Pasikliautinasis intervalas	
				int-	int+
SOM	100	0,1017	0	0,1017	0,1017
	225	0,0688	0	0,0688	0,0688
	400	0,0544	0,0003	0,0541	0,0547
	625	0,0513	0,0005	0,0508	0,0518
	900	0,0453	0,0004	0,0449	0,0458
ND					
	100	0,0040	0	0,0040	0,0040
	225	0,0019	0	0,0018	0,0019
	400	0,0009	0	0,0009	0,0009
	625	0,0004	0	0,0004	0,0005
	900	0,0002	0	0,0002	0,0002

Kaip matome iš gautų rezultatų, didinant neuronų skaičių visoms analizuojamų duomenų aibėms, kvantavimo paklaida visada mažėja, o jos

pasikliautinojo intervalo galai sutampa su kvantavimo paklaida arba intervalas yra labai mažas. Taigi, galime daryti išvadą, jog mažoms duomenų aibėms, tokiomis kaip iris ar hepta kvantavimo paklaida nepriklauso nuo pradinių neuronų reikšmių parinkimo. Kai duomenų aibės yra didesnės (chainlink, elipsoidai) ir neuronų yra daugiau, SOM metodu kvantavimo paklaida labiau priklauso nuo pradinių neuronų reikšmių parinkimo, negu ND metodu.

4.5 lentelė. *Kvantavimo paklaida ir jos pasikliautinasis intervalas elipsoidų duomenų aibei, 1 būdas*

Metodas	*N*	E_{QE}	*α*	Pasikliautinasis intervalas	
				int-	int+
SOM	100	0,3521	0,0045	0,3476	0,3565
	400	0,1988	0,0008	0,1979	0,1996
	900	0,1844	0,0009	0,1835	0,1853
ND					
	100	0,0491	0,0001	0,0490	0,0492
	400	0,0209	0,0001	0,0207	0,0210
	900	0,0083	0,0001	0,0082	0,0084

4.6 lentelė. *Kvantavimo paklaidos priklausomybė nuo vektorių pateikimo į tinklą tvarkos, analizuojant iris duomenų aibę*

SOM	*N (r)*	Į tinklą vektorių pateikimo tvarka		*α*	Pasikliautinasis intervalas	
		Iš eilės (2 būdas)	Ne iš eilės (3 būdas)		int-	int+
E_{QE}	25 (22)	0,4488	**0,4190**	0,0180	0,4010	0,4370
	49(42)	**0,3224**	0,3275	0,0055	0,3220	0,3329
	81(62)	**0,2763**	0,2830	0,0028	0,2802	0,2858
	100 (73)	0,2679	**0,2665**	0,0060	0,2605	0,2725
	169 (98)	**0,2287**	0,2298	0,0021	0,2277	0,2318
ND						
E_{QE}	25 (25)	0,0954	**0,0886**	0,0013	0,0874	0,0899
	49(48)	0,0448	**0,0441**	0,0009	0,0433	0,0450
	81(75)	0,0252	**0,0214**	0,0009	0,0205	0,0223
	100 (90)	0,0163	**0,0140**	0,0008	0,0132	0,0148
	169 (128)	0,0056	**0,0044**	0,0005	0,0039	0,0050

Kvantavimo paklaidos gautos taikant 2 ir 3 būdą rezultatai, jos pasikliautinasis intervalas ir α reikšmės pateiktos 4.6–4.10 lentelėse. Mažiausia paklaida, lyginant 2 ir 3 būdus, yra paryškintu šriftu. Pasikliautinasis intervalas

yra labai mažas arba jo galai sutampa su gautomis paklaidomis, kai į tinklą vektoriai pateikiami ne iš eilės.

Atlikti eksperimentai parodė, kad visiems analizuojamiems duomenims, taikant 3 būdą SOM ir ND metodu kvantavimo paklaida priklauso nuo vektorių pateikimo į tinklą tvarkos. Mažoms duomenų aibėms, tokioms, kaip irisai, hepta ar vėžio, galime naudoti tiek 2, tiek 3 būdą, nes kvantavimo paklaidos mažai skiriasi. Didelėms duomenų aibėms (chainlink ir elipsoidai) 3 būdu gaunama mažesnė kvantavimo paklaida (paryškintu šriftu), todėl naudoti 2 vektorių pateikimo į tinklą būdą netikslinga, nes jis duoda prastesnį rezultatą, negu naudojant 3 būdą ND ir SOM metodu.

4.7 lentelė. *Kvantavimo paklaidos priklausomybė nuo vektorių pateikimo į tinklą tvarkos, analizuojant hepta duomenų aibę*

SOM	*N (r)*	Į tinklą vektorių pateikimo tvarka		*α*	Pasikliautinasis intervalas	
		Iš eilės (2 būdas)	Ne iš eilės (3 būdas)		int-	int+
E_{QE}	25 (18)	0,6771	**0,6278**	0,0155	0,6123	0,6433
	49(29)	**0,5090**	0,5307	0,0159	0,5148	0,5466
	81(49)	0,4469	**0,4461**	0,0076	0,4385	0,4537
	100 (67)	**0,3890**	0,3993	0,0076	0,3917	0,4069
	169 (95)	**0,3178**	0,3216	0,0045	0,3171	0,3261
	256 (128)	0,2780	**0,2750**	0,0048	0,2702	0,2798
ND						
E_{QE}	25 (24)	**0,2048**	0,2195	0,0036	0,2159	0,2231
	49(45)	**0,0993**	0,1033	0,0027	0,1005	0,1060
	81(72)	0,0559	**0,0520**	0,0013	0,0506	0,0533
	100 (91)	**0,0363**	0,0379	0,0011	0,0368	0,0390
	169 (136)	**0,0133**	0,0146	0,0012	0,0134	0,0158
	256 (168)	**0,0046**	0,0051	0,0007	0,0043	0,0058

Atlikus šių trijų neuronų pradinių reikšmių į tinklą pateikimo būdų analizę, paaiškėjo, kad iris ir hepta duomenims naudojant 1–3 būdą SOM metodu kvantavimo paklaidos mažai skiriasi, todėl galime naudoti bet kurį iš jų, o ND metodu irisų duomenų aibei geriau naudoti 1 arba 3 būdą, hepta – 1 arba 2 būdą. Vėžio duomenims mažiausios kvantavimo paklaidos gaunamos taikant 2 arba 3 būdą SOM metodu ir 1 arba 3 būdą ND metodu. Chainlink ir elipsoidų

aibėms geriausia naudoti 1 arba 3 būdą SOM metodu, o ND metodu chainlink duomenims nėra skirtumo, kurį iš trijų būdų naudosime, nes kvantavimo paklaidos labai mažai skiriasi.

4.8 lentelė. *Kvantavimo paklaidos priklausomybė nuo vektorių pateikimo į tinklą tvarkos, analizuojant vėžio duomenų aibę*

SOM	*N (r)*	Į tinklą vektorių pateikimo tvarka		*α*	Pasikliautinasis intervalas	
		Iš eilės (2 būdas)	Ne iš eilės (3 būdas)		int-	int+
E_{QE}	100 (98)	**2,7186**	2,7582	0,0170	2,7412	2,7751
	225 (192)	**2,3548**	2,3932	0,0082	2,3850	2,4013
	400 (289)	**2,2160**	2,2442	0,0078	2,2364	2,2520
	625 (356)	**2,2551**	2,2672	0,0073	2,2599	2,2745
ND						
E_{QE}	100 (100)	5,3567	**5,2119**	0,0467	5,1652	5,2586
	225 (212)	2,5458	**2,3009**	0,0635	2,2374	2,3645
	400 (315)	1,2209	**0,9689**	0,0523	0,9166	1,0212
	625 (408)	**0,4432**	0,4909	0,0280	0,4629	0,5189

4.9 lentelė. *Kvantavimo paklaidos priklausomybė nuo vektorių pateikimo į tinklą tvarkos, analizuojant chainlink duomenų aibę*

SOM	*N (r)*	Į tinklą vektorių pateikimo tvarka		*α*	Pasikliautinasis intervalas	
		Iš eilės (2 būdas)	Ne iš eilės (3 būdas)		int-	int+
E_{QE}	100 (74)	0,1041	**0,1029**	0,0019	0,1009	0,1048
	225 (175)	**0,0643**	0,0671	0,0010	0,0662	0,0681
	400 (294)	0,0547	**0,0545**	0,0011	0,0533	0,0556
	625 (381)	0,0535	**0,0516**	0,0007	0,0509	0,0522
	900 (488)	0,0461	**0,0460**	0,0007	0,0453	0,0466
	1225 (570)	0,0434	**0,0432**	0,0005	0,0427	0,0436
ND						
E_{QE}	100 (100)	**0,0039**	0,0041	0,0000	0,0041	0,0041
	225 (225)	**0,0018**	**0,0018**	0,0000	0,0018	0,0019
	400 (398)	**0,0009**	**0,0009**	0,0000	0,0009	0,0009
	625 (593)	0,0005	**0,0004**	0,0000	0,0004	0,0004
	900 (752)	**0,0002**	**0,0002**	0,0000	0,0002	0,0002
	1225 (859)	**0,0001**	**0,0001**	0,0000	0,0001	0,0001

4.10 lentelė. *Kvantavimo paklaidos priklausomybė nuo vektorių pateikimo į tinklą tvarkos, analizuojant elipsoidų duomenų aibę*

SOM	*N (r)*	Į tinklą vektorių pateikimo tvarka		α	Pasikliautinasis intervalas	
		Iš eilės (2 būdas)	Ne iš eilės (3 būdas)		int-	int+
E_{QE}	100 (71)	0,4216	**0,3558**	0,0066	0,3492	0,3624
	400 (238)	0,2127	**0,2007**	0,0014	0,1992	0,2021
	900 (439)	0,1878	**0,1856**	0,0009	0,1847	0,1865
	2500 (631)	0,1854	**0,1846**	0,0011	0,1835	0,1856
ND						
E_{QE}	100 (100)	**0,0491**	**0,0491**	0,0002	0,0489	0,0493
	400 (376)	0,0224	**0,0208**	0,0002	0,0206	0,0210
	900 (673)	0,0110	**0,0081**	0,0002	0,0079	0,0083
	2500 (1092)	0,0021	**0,0016**	0,0000	0,0015	0,0016

Šio eksperimentinio tyrimo rezultatai dar kartą patvirtina, kad ND metodu gauta kvantavimo paklaida yra mažesnė negu gauta SOM metodu. Be to, galima daryti išvadą, kad didesnėms duomenų aibėms (chainlink, elipsoidai) yra tikslinga keisti vektorių pateikimo į tinklą tvarką arba pradines neuronų reikšmes parinkti atsitiktinai. Mažoms duomenų aibėms nėra prasmės keisti vektorių pateikimo į tinklą tvarkos, nes kvantavimo paklaidos mažai skiriasi.

4.7. Tinklo mokymo skaičiavimo laiko tyrimas

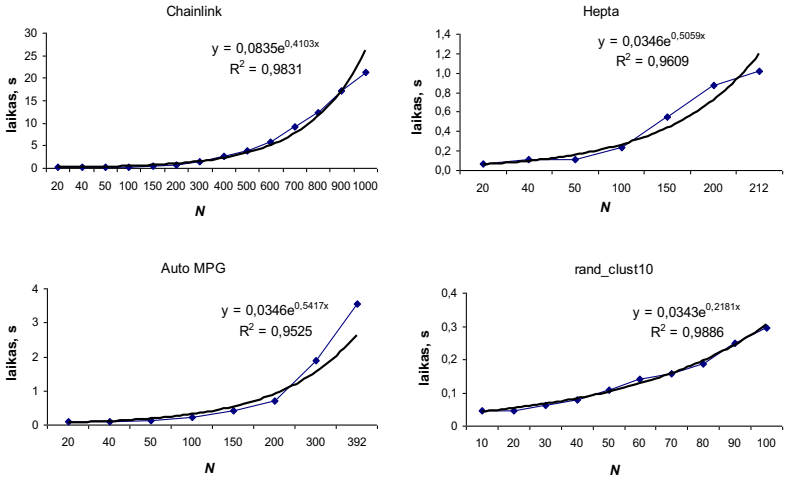
Šiame skyrelyje pateikta tinklo mokymo skaičiavimo laiko tyrimo, naudojant saviorganizuojantį neuroninį tinklą ir neuroninių dujų metodą, rezultatai. Buvo siekiama išsiaiškinti, kaip priklauso tinklo mokymo skaičiavimo laikas nuo neuronų ir epochų skaičiaus. Pateikti tyrimų rezultatai publikuoti darbe (B2).

Eksperimentai buvo atlikti su šiomis duomenų aibėmis: hepta [212; 3], auto MPG [392; 7], target [770; 2], chainlink [1000; 3], rand_clust10 [100; 10], rand_clust5 [100; 5], rand_clust57 [100; 10] ir rand_clust13 [100;10].

Eksperimentinio tyrimo metu buvo naudojama SOM (3.2) mokymo taisyklė, kai kaimynystės funkcija h_{ij}^{c} apskaičiuojama pagal (3.3) formulę.

Tinklo mokymo skaičiavimo laiko, kai pradinės neuronų komponentės yra atsitiktinės ir SOM mokymo taisyklės kaimynystės funkcija apskaičiuota pagal (3.3) formulę, tyrimo rezultatai pateikiami 4.12 paveiksle.

Iš 4.12 paveikslo matome, kaip tinklo mokymo skaičiavimo laikas priklauso nuo neuronų skaičiaus *N*. Atlikus eksperimentinę analizę nustatyta, kad gauti taškai pakankamai tiksliai aproksimuojami eksponentine funkcija $y = ae^{bx}$ su teigiamu laipsniu $b > 0$, daugeliu atvejų determinacijos koeficientas $R^2 > 0{,}9$. Tai reiškia, kad didinant neuronų skaičių, skaičiavimo laikas eksponentiškai didėja. Taip pat gauti taškai buvo aproksimuojami tiese, determinacijos koeficientai (R^2) pateikti 4.11 lentelėje. Matome, kad eksponentine funkcija aproksimuojama tiksliau, nes šios funkcijos determinacijos koeficientas yra didesnis negu tiesinės funkcijos.



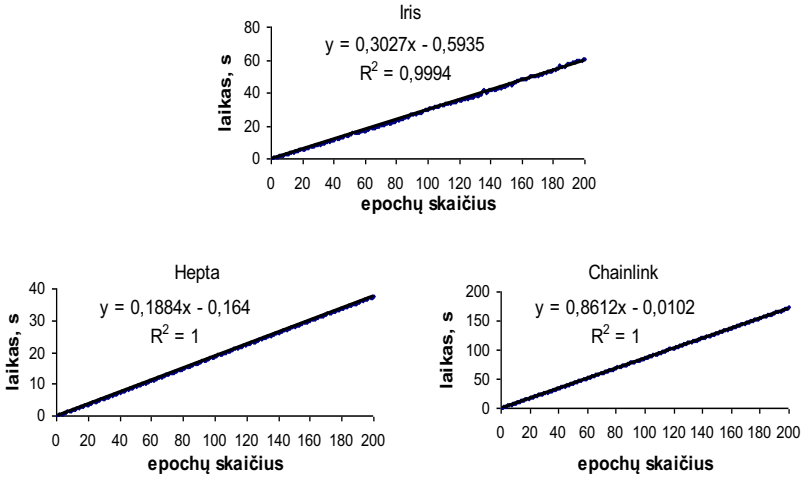
4.12 pav. *SOM skaičiavimo laiko priklausomybė nuo neuronų skaičiaus N*

Taikant SOM (3.2) mokymo taisyklę, kai kaimynystės funkcija h_{ij}^c apskaičiuojama pagal (3.4) formulę, tinklo mokymo skaičiavimo laikas pateiktas 4.13 paveiksle. Atlikus po 20 eksperimentų visoms analizuojamų duomenų aibėms, kai pradinės neuronų komponentės parenkamos atsitiktinai, neuronų skaičius $N = 100$, atliekamų epochų skaičius lygus 200, buvo

apskaičiuoti gautų kvantavimo paklaidų vidurkiai. Eksperimentai atlikti su kitomis *N* reikšmėmis, tačiau gauti rezultatai pagrindžia tą pačią tendenciją.

4.11 lentelė. *Eksponentinės ir tiesinės funkcijos determinacijos koeficientas* (R^2) *taikant SOM*

Duomenys	Eksponentinė funkcija		Tiesinė funkcija
	(R^2)	*b*	(R^2)
chainlink	**0,9831**	0,4103	0,7819
hepta	**0,9609**	0,5059	0,8923
auto MPG	**0,9525**	0,5417	0,6896
target	**0,9784**	0,4676	0,7670
rand_clust10	**0,9886**	0,2181	0,9375
rand_clust5	**0,9803**	0,2319	0,9498
rand_clust57	**0,9882**	0,2207	0,9486
rand_clust13	**0,9689**	0,2395	0,9176
rand_clust24	**0,9789**	0,2196	0,9329

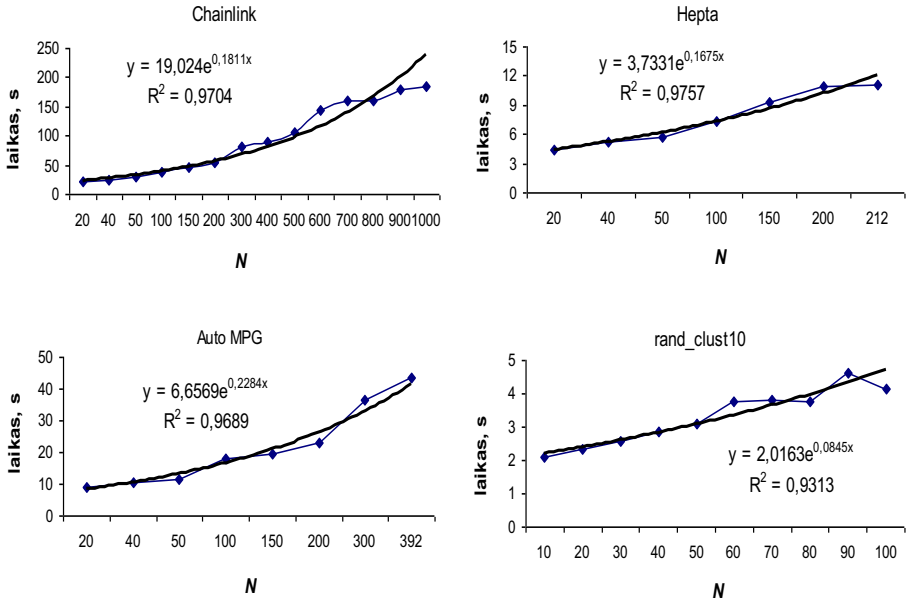


4.13 pav. *SOM tinklo mokymo skaičiavimo laiko priklausomybė nuo atliekamos epochos*

Iš 4.13 paveikslo matome, kad tinklo mokymo skaičiavimo laikas priklauso nuo atliekamos epochos numerio. Gauti rezultatai rodo, kad gauti taškai pakankamai tiksliai aproksimuojami tiesine funkcija $y = ax + b$, o determinacijos koeficientas (R^2) daugeliu atvejų artimas 1, t. y. daugiau už 0,9. Tai reiškia, kad didinant epochų skaičių, skaičiavimo laikas didėja tiesiškai.

Toliau eksperimentuose buvo naudotas ND metodas (3.1 skyrelis), kai pradinės neuronų komponentės yra atsitiktinės. Tyrimo rezultatai pateikiami 4.14 paveiksle.

Iš 4.14 paveikslo matome, kad ND metodu tinklo mokymo skaičiavimo laikas taip pat priklauso nuo neuronų skaičiaus *N*, kaip ir SOM metodu. Atlikus eksperimentinę analizę nustatyta, kad gauti taškai pakankamai tiksliai aproksimuojami eksponentine funkcija $y = ae^{bx}$ su teigiamu laipsniu $b > 0$, daugeliu atvejų determinacijos koeficientas $R^2 > 0{,}9$. Tai reiškia, kad didinant neuronų skaičių, skaičiavimo laikas eksponentiškai didėja. Taip pat gauti taškai buvo aproksimuojami ir tiese (4.12 lentelė). Tiesinės ir eksponentinės funkcijos determinacijos koeficientai skiriasi nedaug, be to eksponentinės funkcijos laipsnis *b* yra mažas, todėl taškai gerai aproksimuojami ir tiese.

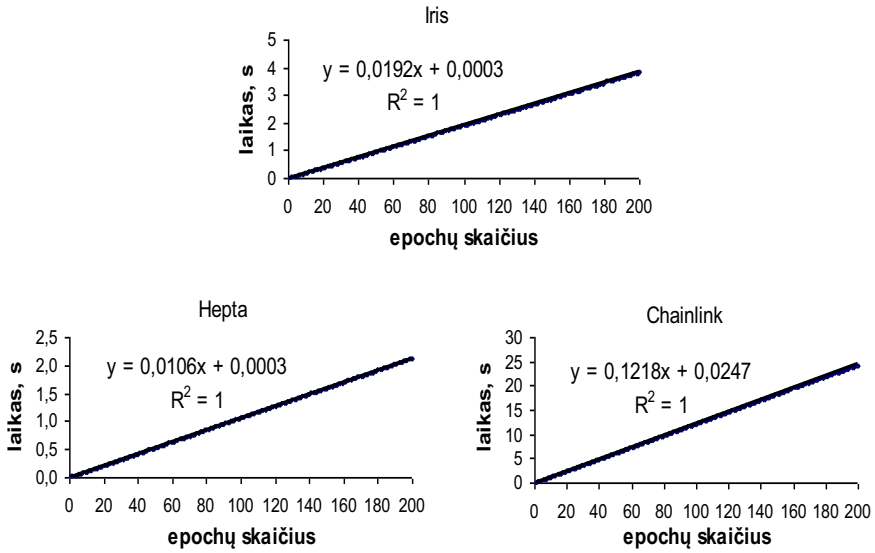


4.14 pav. *ND skaičiavimo laiko priklausomybė nuo neuronų skaičiaus N*

4.12 lentelė. *Eksponentinės ir tiesinės funkcijos determinacijos koeficientas* (R^2) *taikant ND*

Duomenys	Eksponentinė funkcija		Tiesinė funkcija
	(R^2)	b	(R^2)
chainlink	**0,9704**	0,1811	0,9642
hepta	**0,9757**	0,1675	0,9667
autoMPG	**0,9689**	0,2284	0,8893
target	**0,9123**	0,1894	0,8705
rand_clust10	**0,9313**	0,0845	0,9266
rand_clust5	0,9823	0,0723	**0,9875**
rand_clust57	0,9691	0,0790	**0,9821**
rand_clust13	**0,9609**	0,0866	0,9555
rand_clust24	0,9700	0,0752	**0,9943**

Taikant ND metodą (3.1 skyrelis) tinklo mokymo skaičiavimo laiko priklausomybė nuo epochų skaičiaus pateikta 4.15 paveiksle. Atlikus po 20 eksperimentų visoms analizuojamų duomenų aibėms, kai pradinės neuronų komponentės parenkamos atsitiktinai, neuronų skaičius $N=100$, atliekamų epochų skaičius lygus 200, buvo apskaičiuoti gautų kvantavimo paklaidų vidurkiai.



4.15 pav. *ND tinklo mokymo skaičiavimo laiko priklausomybė nuo atliekamos epochos*

Iš 4.15 paveikslo matome, kad tinklo mokymo skaičiavimo laikas priklauso nuo atliekamos epochos numerio. Gauti rezultatai rodo, kad gauti taškai pakankamai tiksliai aproksimuojami tiesine funkcija $y = ax + b$, o determinacijos koeficientas (R^2) lygus 1. Tai rodo, kad didinant epochų skaičių, skaičiavimo laikas didėja tiesiškai.

Naudojant saviorganizuojančius neuroninius tinklus visiems analizuojamiems duomenims tinklo mokymo skaičiavimo laikas, priklausantis nuo neuronų skaičiaus, didėja eksponentiškai, o naudojant ND metodą realaus pobūdžio duomenims tinklo mokymo skaičiavimo laikas, priklausantis nuo neuronų skaičiaus taip pat didėja eksponentiškai, tačiau dirbtinai sugeneruotoms duomenų aibėms rand_clust5, rand_clust57 ir rand_clust24 (2.1 lentelė) didėja tiesiškai. Tinklo mokymo laikas, priklausantis nuo atliekamos epochos SOM ir ND metodu, didėja tiesiškai.

4.8. Nuoseklaus SOM ir ND metodų ir MDS junginio tyrimas

Šiame skyrelyje yra pateikiama nuoseklaus saviorganizuojančio neuroninio tinklo ir daugiamačių skalių metodo bei neuroninių dujų ir daugiamačių skalių metodo junginių analizė (3.4.1 skyrelis). Daugiamačių skalių paklaida minimizuojama SMACOF algoritmu. Pateikti tyrimų rezultatai publikuoti darbuose (A2, A3, A4 ir B2).

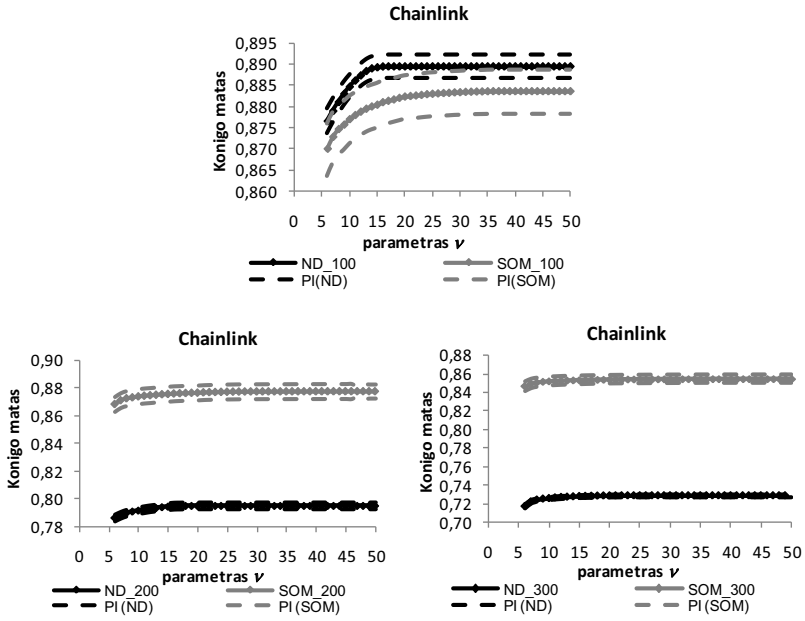
Tyrimuose naudotos duomenų aibės, turinčios specifinių savybių: hepta [212; 3], auto MPG [392; 7], chainlink [1000; 3], rand_clust10 [100; 10], rand_clust5 [100; 5] ir rand_clust57 [100; 10] (2.1 skyrelis).

4.8.1. Vektorių atvaizdavimo kokybės vertinimas

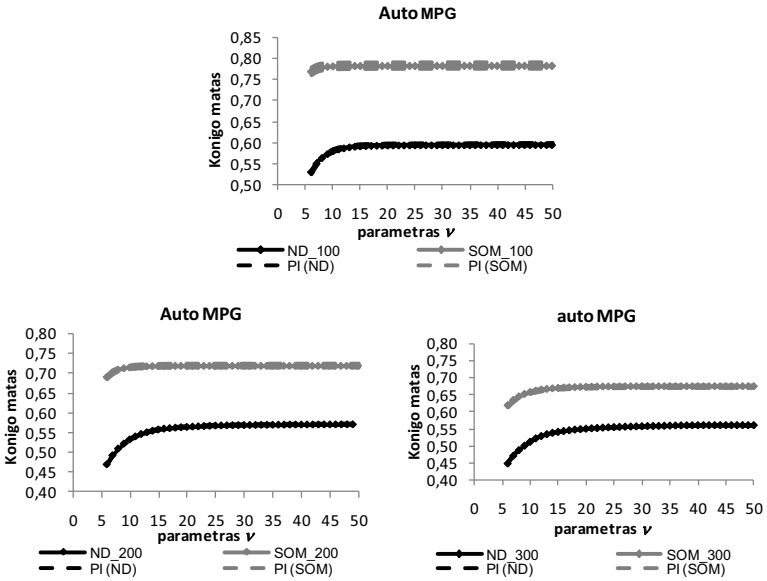
Tiriant nuoseklųjį neuroninių dujų ir saviorganizuojančių neuroninių tinklų jungimą su daugiamačių skalių metodu, nustatyta, kad neuronų nugalėtojų skaičius daro įtaką vizualizavimo rezultatų kokybei.

Šiame skyrelyje palyginami nuosekliojo junginio rezultatai pagal šiuos kokybės matus: Konigo matą E_{KM} (2.7), Spirmano koeficientą ρ_{Sp} (2.8) ir MDS paklaidą $\hat{E}_{\mathrm{MDS}}$ (2.9).

Įėjimo vektoriai $X_1, X_2, ..., X_m$ pateikiami į saviorganizuojantį neuroninį tinklą arba analizuojami neuroninių dujų metodu, po SOM arba ND mokymo gauti neuronai nugalėtojai $\hat{M}_1, \hat{M}_2, ..., \hat{M}_r$ daugiamačių skalių metodu atvaizduojami plokštumoje, gaunami dvimačiai vektoriai $Y_1, Y_2, ..., Y_r$. Neuronų skaičių N parenkame taip, kad neuronų nugalėtojų skaičius r būtų lygus 100, 200 ir 300 chainlink ir auto MPG duomenims, 50, 100 ir 150 iris, 50, 100 ir 200 hepta bei 50, 80 ir 100 rand_clust10 duomenims. Atlikta po 40 eksperimentų su kiekvienu įėjimo vektoriumi su skirtingomis pradinėmis neuronų reikšmėmis SOM ir ND metodu. Apskaičiuotas gautų reikšmių vidurkis ir jo pasikliautinasis intervalas (pasikliovimo lygmuo 0,95).



4.16 pav. *Konigo mato reikšmių priklausomybė nuo parametro* ν

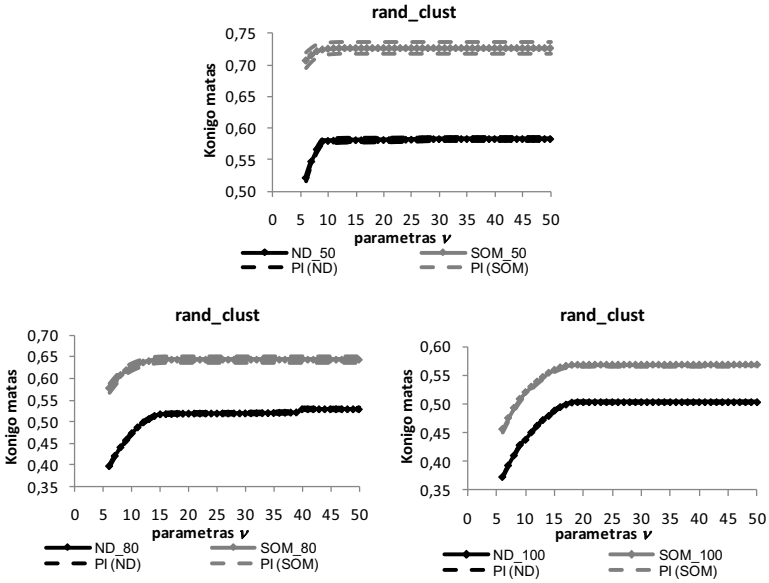


4.17 pav. *Konigo mato reikšmių priklausomybė nuo parametro* ν

Skaičiuojant Konigo kaimyniškumų išlaikymo matą E_{KM} yra labai svarbu parinkti dviejų parametrų μ ir ν reikšmes. Parametras μ parodo mažesnį artimiausių kaimynų ratą, o parametras ν – didesnį (platesnį) artimiausių kaimynų skaičių (ratą). Eksperimentai atlikti, kai $\mu = 4$, o parametras ν buvo keičiamas nuo 6 iki 50. Gauti E_{KM} reikšmių vidurkiai ir jų pasikliautinieji intervalai (PI) pateikti 4.16–4.18 paveiksluose.

Iš 4.16–4.18 paveikslų matome, kad E_{KM} reikšmė yra didesnė, kai neuronai nugalėtojai yra gaunami SOM metodu visiems duomenims, išskyrus chainlink duomenų aibę, kai neuronų nugalėtojų yra 100. Galime daryti išvadą, kad kaimyniškumas tiksliau išlaikomas, kai neuronai nugalėtojai yra gauti SOM metodu negu ND ir vizualizuojami daugiamačių skalių metodu. Chainlink duomenų aibei Konigo mato reikšmė gaunama didesnė naudojant ND metodą, tačiau šiuo atveju vidurkio pasikliautinasis intervalas platus ir persidengia, todėl gauti rezultatai nėra patikimi. Visoms analizuojamų duomenų aibėms, kai neuronų nugalėtojų yra daugiau, pasikliautinasis

intervalas yra siauresnis. Kai parametro v reikšmė yra maža, tai E_{KM} reikšmė mažesnė negu tada, kai v reikšmė didesnė. Nuo pradinės ($v=6$) iki tam tikros v reikšmės (pavyzdžiui, auto MPG duomenys, $6<v<10$ taikant SOM metodą ir $6<v<15$ taikant ND metodą, kai neuronų nugalėtojų skaičius 100) paklaida didėja, o toliau E_{KM} didėjimas nežymus.

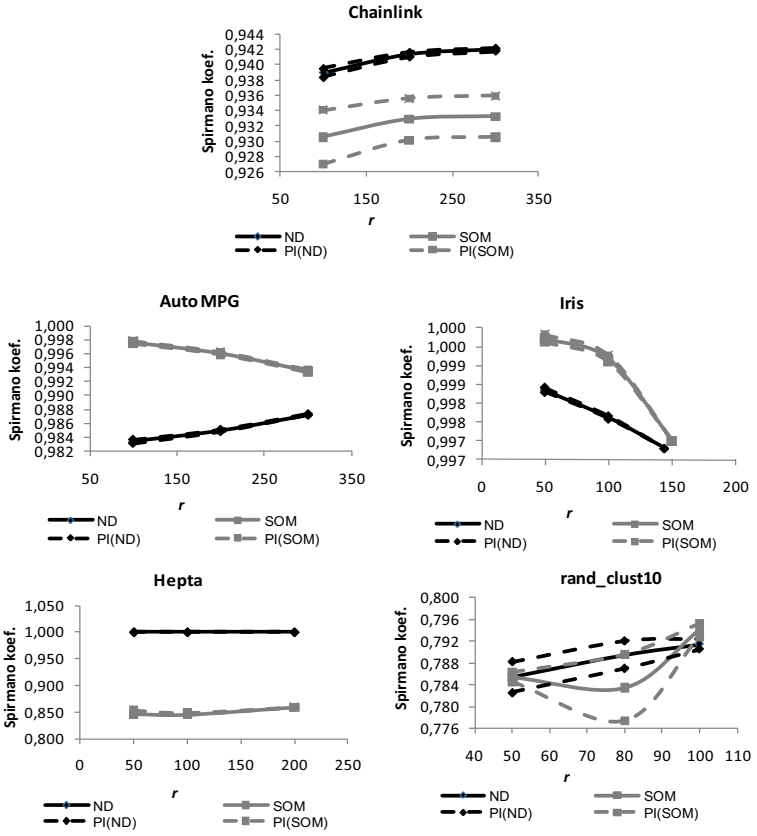


4.18 pav. *Konigo mato reikšmių priklausomybė nuo parametro* v

Eksperimentai parodė, kad naudojant Konigo matą kaimyniškumų išlaikymas yra daug tikslesnis, kai neuronai nugalėtojai gaunami SOM metodu ir vizualizuojami MDS. Išvada – SOM geriau išlaiko kaimyniškumus (artumus).

Spirmano koeficientas buvo naudojamas atstumų išlaikymui įvertinti. Jis parodo, ar visi atstumai santykinai vienodi daugiamatėje erdvėje ir dvimatėje erdvėje. Spirmano koeficiento ρ_{Sp} reikšmių vidurkis ir jo pasikliautinasis intervalas (PI) (pasikliovimo lygmuo 0,95) pateiktas 4.19 paveiksle. Spirmano koeficiento reikšmės yra didesnės, kai neuronai nugalėtojai yra gauti ND

metodu chainlink ir hepta duomenims, o SOM metodu auto MPG ir iris duomenų aibėms. Spirmano koeficiento reikšmės pakankamai didelės.



4.19 pav. *Spirmano koeficiento reikšmių priklausomybė nuo neuronų nugalėtojų skaičiaus r*

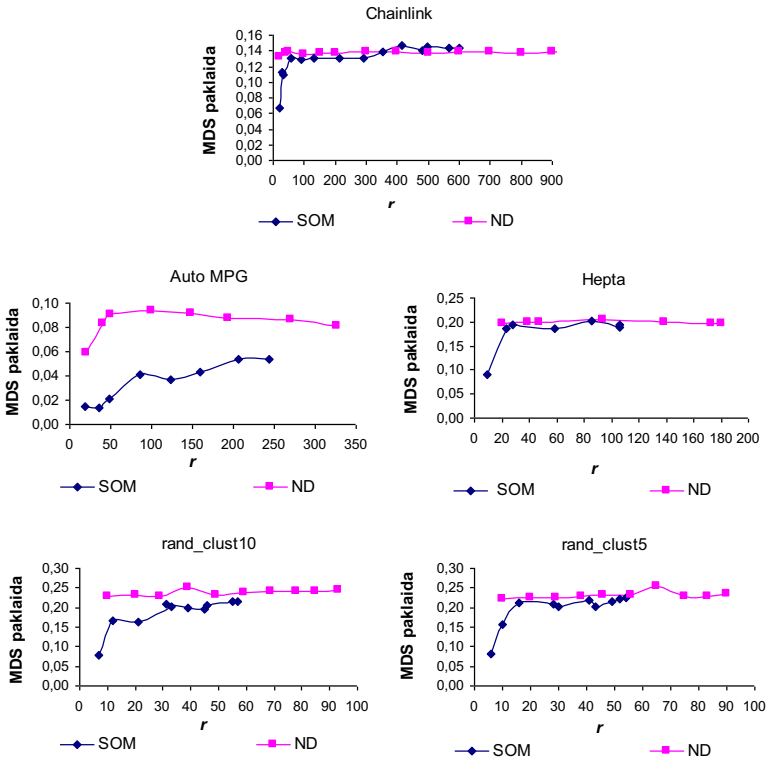
Daugeliu atvejų $\rho_{\mathrm{Sp}} > 0{,}9$, tai rodo, kad atvaizduojant duomenis iš *n*-matės erdvės į dvimatę santykinai išlaikomas duomenų kaimyniškumas. Įvertinti kaimyniškumo išlaikymo kokybę rand_clust10 duomenims yra sudėtinga, nes Spirmano koeficiento reikšmės yra įvairios, o reikšmių vidurkio pasikliautinasis intervalas yra platus ir persidengiantis.

Spirmano koeficiento reikšmės yra pakankamai didelės, todėl vizualizavimo kokybė yra gera atstumų išlaikymo prasme, tačiau pagal Spirmano koeficientą yra sunku pasakyti, kuris iš metodų (ND ar SOM) yra

geresnis. Atlikto tyrimo rezultatai parodė, kad abu kvantavimo metodai ND ir SOM yra tinkami jungimui su MDS.

Dar vienas matas, kuris buvo naudojamas vizualizavimo rezultatams įvertinti – tai MDS paklaida. Gauti rezultatai pateikti 4.20 paveiksle.

Kai vizualizuojami neuronai nugalėtojai, gauti SOM ir ND metodu, $\hat{E}_{\text{MDS}}$ paklaidos yra beveik lygios chainlink ir hepta duomenims. Šių duomenų dimensija n yra lygi 3. Duomenų aibėms (auto MPG, rand_clust10, rand_clust5), kai vizualizuojami neuronai nugalėtojai, gauti ND metodu, MDS paklaidos $\hat{E}_{\text{MDS}}$ yra mažesnės negu vizualizuojant neuronus nugalėtojus, gautus SOM metodu. Šių duomenų dimensijų skaičius n yra didesnis (n = 5, 7 arba 10).



4.20 pav. *MDS paklaidos priklausomybė nuo neuronų nugalėtojų skaičiaus ND ir SOM metodu*

Atliktų eksperimentinių tyrimų rezultatai leidžia daryti šias išvadas:

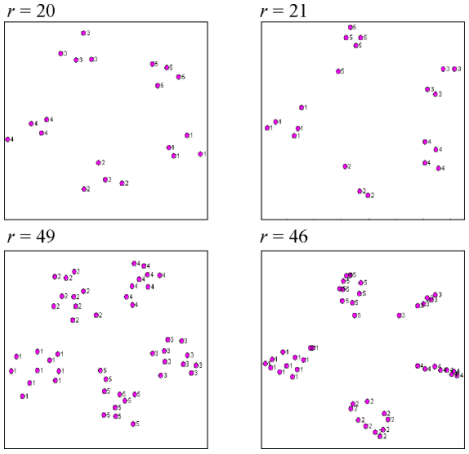
1. Kai neuronai nugalėtojai, gauti ND ir SOM metodu, yra vizualizuojami MDS metodu, MDS paklaidos $\hat{E}_{\text{MDS}}$ reikšmės yra beveik lygios duomenims, kurių dimensijų skaičius *n* yra mažas (n = 3). Kai vizualizuojami neuronai nugalėtojai gauti SOM metodu, $\hat{E}_{\text{MDS}}$ reikšmės yra mažos duomenims, kurių dimensijų skaičius *n* yra didelis (n = 5, n = 7 arba n = 10).
2. Konigo išlaikymo mato reikšmės yra mažesnės, kai neuronai nugalėtojai yra gauti ND metodu negu SOM metodu. SOM metodas labiau išlaiko kaimyniškumus.

4.8.2. Vaizdų, gautų nuosekliuoju junginiu, analizė

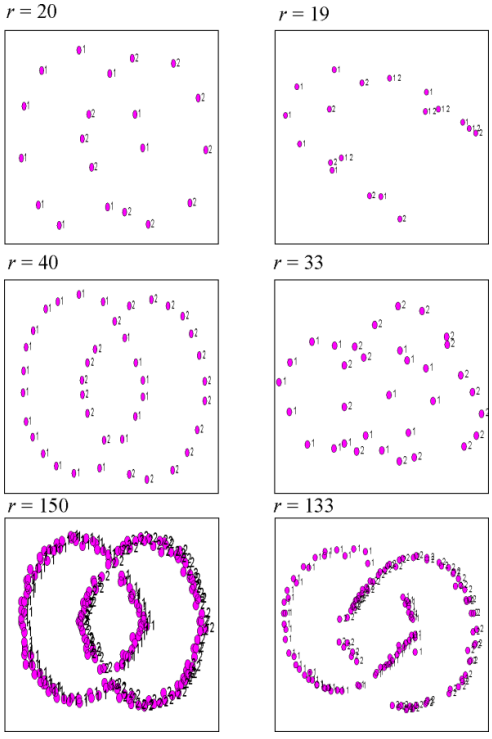
Šiame skyrelyje pateikiama daugiamačių duomenų projekcijų plokštumoje, gautų nuoseklaus junginio algoritmu, analizė (3.4.1 skyrelis). Tyrimo tikslas – ištirti plokštumoje gautų vaizdų kokybės priklausomybę nuo neuronų nugalėtojų ir atliekamų epochų skaičiaus. Vaizdų kokybė buvo vertinama subjektyviai, atsižvelgiant į duomenų žinomų savybių išlaikymą. Joks skaitinis kriterijus nebuvo naudotas. Eksperimentai buvo atlikti naudojant šias duomenų aibes: Fišerio irisus [149; 4], hepta [212; 3], target [770; 2], chainlink [1000; 3] ir rand_clust10 [100; 10].

Neuronai nugalėtojai, kurie yra daugiamačiai vektoriai $\hat{M}_1, \hat{M}_2, ..., \hat{M}_r$ atvaizduojami į dvimačius vektorius $Y_1, Y_2, ..., Y_r$ panaudojant daugiamačių skalių metodą. Neuronų nugalėtojų skaičius yra *r*. Gauti dvimačiai vektoriai pateikti 4.21–4.24 paveiksluose. Prie taškų esantys skaičiai žymi klasės numerį kuriai jie priklauso. Iš 4.21–4.24 paveikslų matome, kaip keičiasi vaizdai, didėjant neuronų nugalėtojų skaičiui. ND metodu duomenų struktūra jau „atsiskleidžia“, kai neuronų nugalėtojų skaičius *r* yra pakankamai mažas

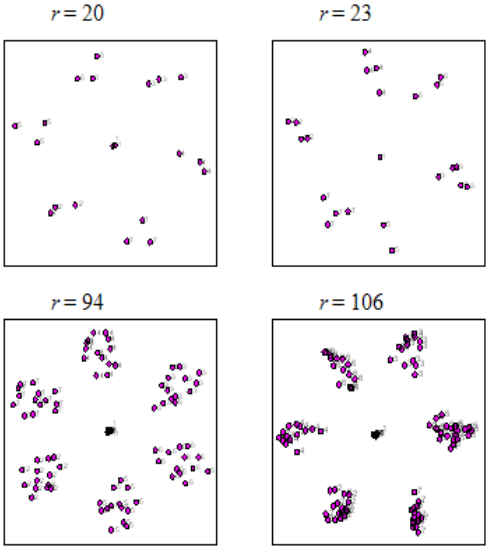
($r=20$). SOM metodu duomenų struktūra yra matoma tik, kai neuronų nugalėtojų skaičius yra didelis ($r=133;296;46$).



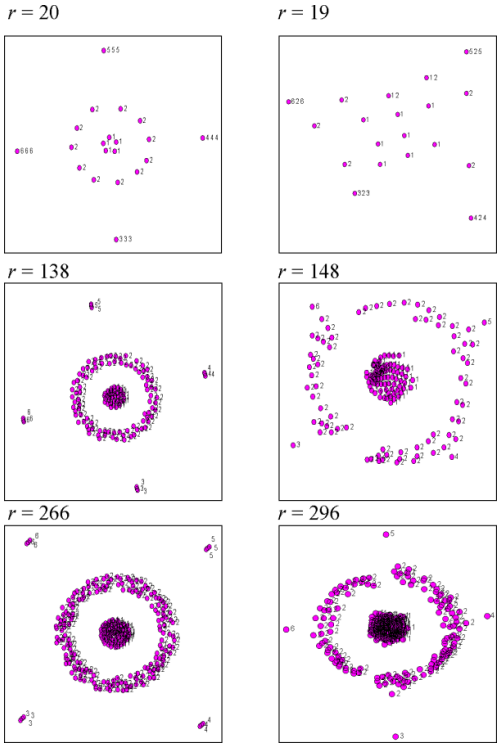
4.21 pav. *Duomenų rand_clust vaizdai esant skirtingam neuronų nugalėtojų skaičiui r: ND (kairėje) ir SOM (dešinėje)*



4.22 pav. *Duomenų chainlink vaizdai esant skirtingam neuronų nugalėtojų skaičiui r: ND (kairėje) ir SOM (dešinėje)*

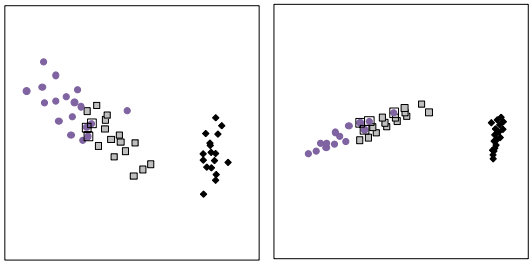


4.23 pav. *Duomenų hepta vaizdai esant skirtingam neuronų nugalėtojų skaičiui r: ND (kairėje) ir SOM (dešinėje)*



4.24 pav. *Duomenų target vaizdai esant skirtingam neuronų nugalėtojų skaičiui r: ND (kairėje) ir SOM (dešinėje)*

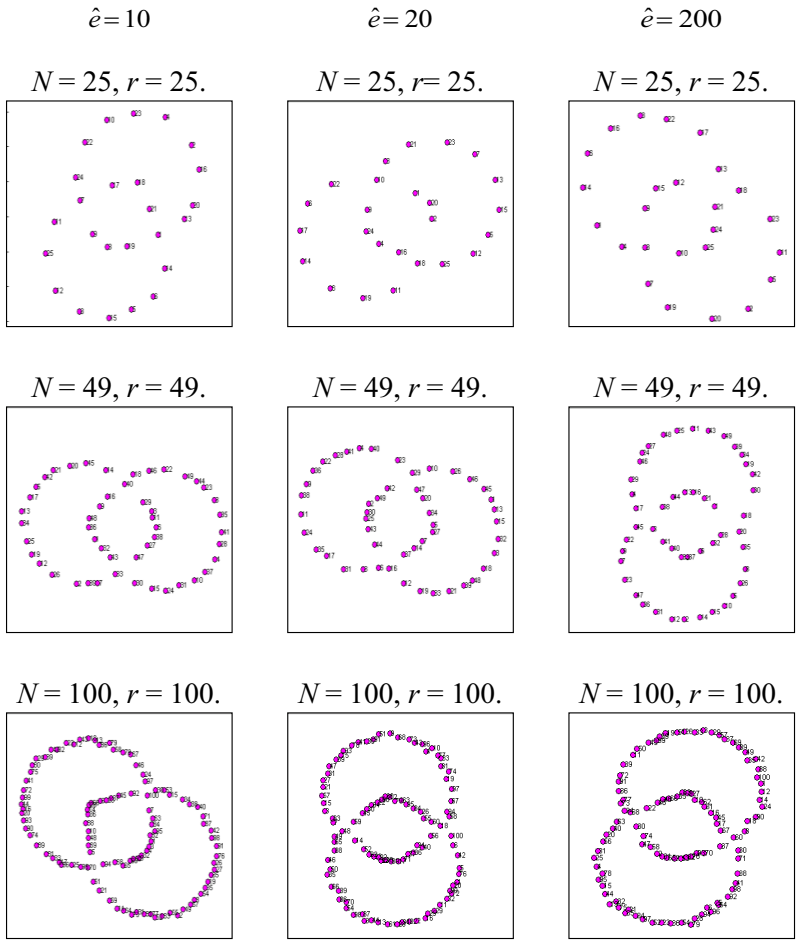
4.25 paveiksle pateikti iris duomenų aibės neuronai nugalėtojai gauti ND ir SOM metodu ir vizualizuoti daugiamačių skalių metodu. Taškai, atitinkantys pirmos veislės (Setosa) gėles, pažymėti mažais rombais; taškai, atitinkantys antros veislės (Versicolor) gėles, pažymėti kvadratėliais ir taškai, atitinkantys trečios veislės (Virginica), pažymėti skrituliukais. Taškai, atitinkantys antros ir trečios veislės gėles, yra pažymėti apibrėžtais skrituliukais. Kvantavimo paklaida gauta SOM metodu yra žymiai didesnė $(E_{\mathrm{QE}} = 0{,}3222)$ negu gauta ND metodu $(E_{\mathrm{QE}} = 0{,}0379)$. Matome, kad taškai gauti SOM metodu yra labiau susiklasterizavę, o taškai gauti ND metodu yra labiau išsibarstę, tačiau duomenų struktūra geriau „atsiskleidžia“ ND metodu.



4.25 pav. *Duomenų iris vaizdai: ND (kairėje) (*$E_{\mathrm{QE}} = 0{,}0379$*) ir SOM (dešinėje) (*$E_{\mathrm{QE}} = 0{,}3222$*)*

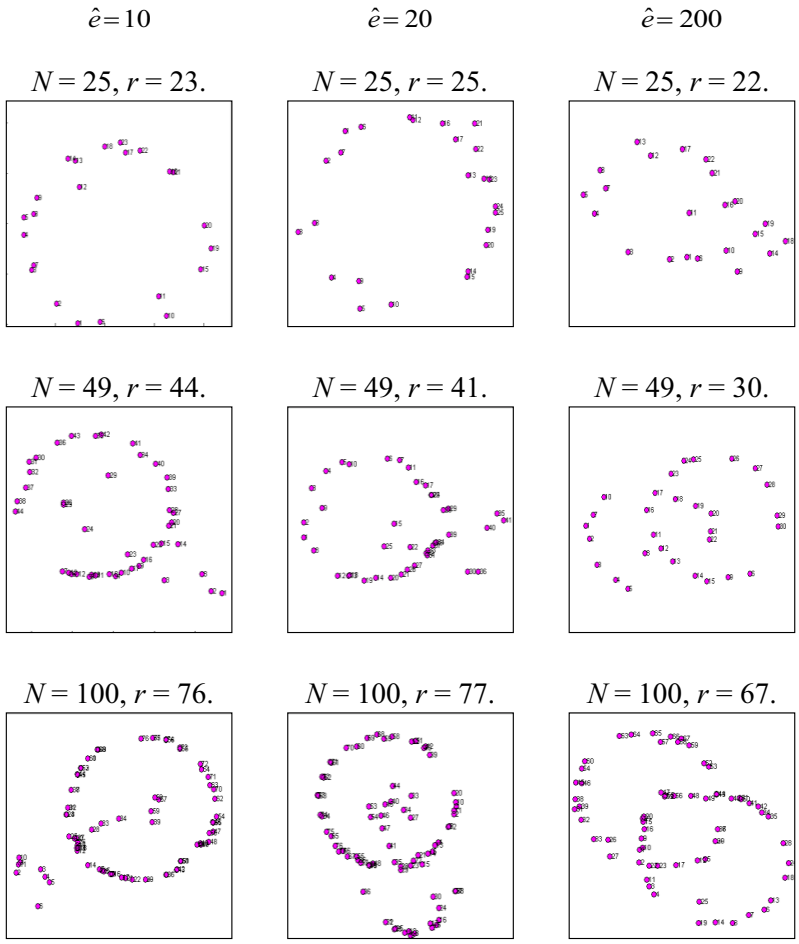
Taip pat atliktas tyrimas, kuriuo norima pamatyti, kaip atrodo vaizdai, gauti atlikus tam tikrą skaičių mokymo epochų.

4.26–4.29 paveiksluose pateiktas neuronų nugalėtojų, gautų, taikant saviorganizuojantį neuroninį tinklą, kai SOM mokymo taisyklėje naudojama kaimynystės funkcija h_{ij}^{c} apskaičiuota pagal (3.4) formulę, ir ND metodu, atvaizduotų plokštumoje daugiamačių skalių metodu, kai atliekama 10, 20 ir 200 epochų. Skaičius, esantis prie taško, žymi neurono nugalėtojo eilės numerį.



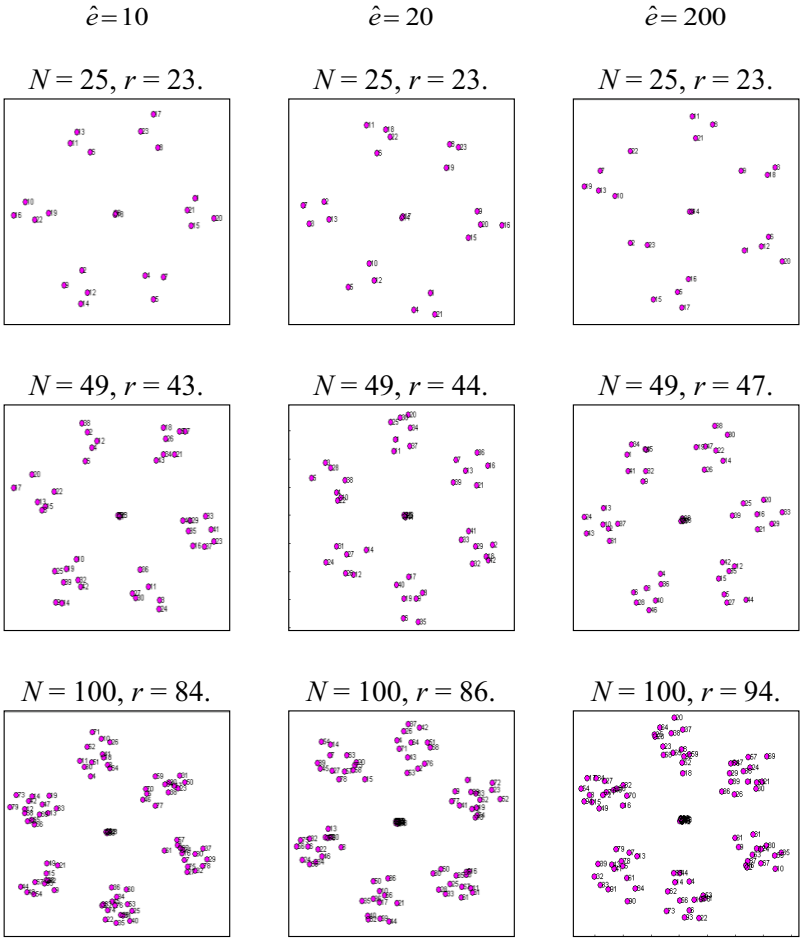
4.26 pav. *Duomenų chainlink vaizdai, kai atliekama 10, 20 ir 200 epochų (ND)*

Chainlink duomenų aibės struktūra jau gerai „atsiskleidžia“, kai neuronai nugalėtojai yra gauti ND metodu ir atliekama tik 10 epochų, o taikant SOM tinklą, reikia atlikti net 200 epochų, tačiau ir atliekant 200 epochų gautas vaizdas vis tiek yra prastesnis negu taikant ND metodą. Kuo neuronų skaičius yra didesnis, tuo chainlink duomenų aibės struktūra yra geriau matoma abiem metodais (4.26 ir 4.27 paveikslai).



4.27 pav. *Duomenų chainlink vaizdai, kai atliekama 10, 20 ir 200 epochų (SOM)*

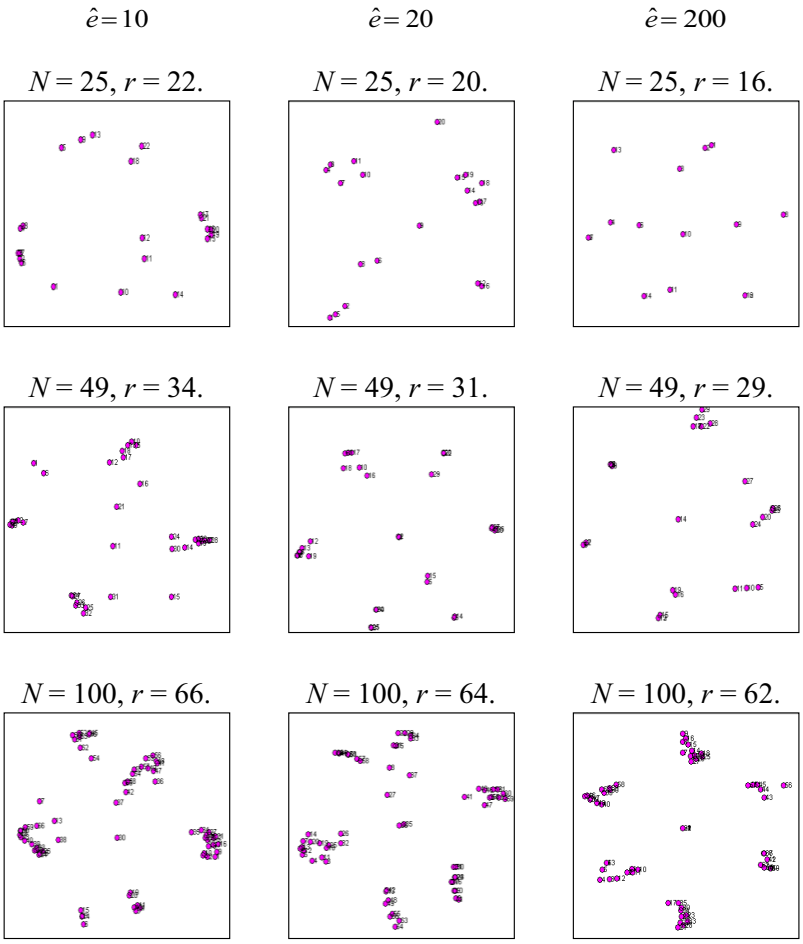
Iš 4.28 ir 4.29 paveikslų matome, kad hepta duomenų aibės struktūra taip pat jau gerai matoma, kai neuronai nugalėtojai yra gauti ND metodu ir atliekama tik 10 epochų, o taikant SOM tinklą, reikia atlikti 200 epochų. Taigi išlieka ta pati tendencija kaip ir chainlink duomenims, kad didinant neuronų skaičių hepta duomenų struktūra dar geriau „atsiskleidžia“ taikant SOM tinklą ir ND metodą.



4.28 pav. *Duomenų hepta vaizdai, kai atliekama 10, 20 ir 200 epochų (ND)*

Neuroninių dujų metodu duomenų struktūra gerai „atskleidžiama“ atlikus vos 10 epochų, o saviorganizuojančiu neuroniniu tinklu reikia atlikti 200 epochų. Tam, kad analizuojamų duomenų struktūra būtų gerai matoma taikant saviorganizuojančius neuroninius tinklus galimos dvi alternatyvos:

1. Didinti neuronų skaičių, kai atliekamų epochų skaičius yra mažas.
2. Didinti epochų skaičių, kai neuronų skaičius yra mažas.



4.29 pav. *Duomenų hepta vaizdai, kai atliekama 10, 20 ir 200 epochų (SOM)*

Atlikti eksperimentai dar kartą patvirtina išvadą, kad neuroninių dujų metodas yra tinkamesnis jungimui su daugiamatėmis skalėmis negu saviorganizuojantys neuroniniai tinklai. Priežastis, kodėl ND metodu gautus neuronus nugalėtojus atvaizdavus plokštumoje, analizuojama duomenų struktūra geriau „atskleidžiama“ lyginant su rezultatais, kai junginyje naudojamas SOM, yra suprantama: ND metodu gaunama mažesnė kvantavimo paklaida, t. y. vizualizuojami neuronai nugalėtojai, kurie labiau atitinka analizuojamą duomenų aibę.

4.9. Integruoto SOM ir ND metodų ir MDS junginio tyrimas

Šiame skyrelyje pateikiama integruoto saviorganizuojančio neuroninio tinklo ir daugiamačių skalių bei neuroninių dujų ir daugiamačių skalių metodo junginių tyrimo rezultatai ir jų lyginamoji analizė. Pateikti rezultatai publikuoti darbuose (A5 ir B3).

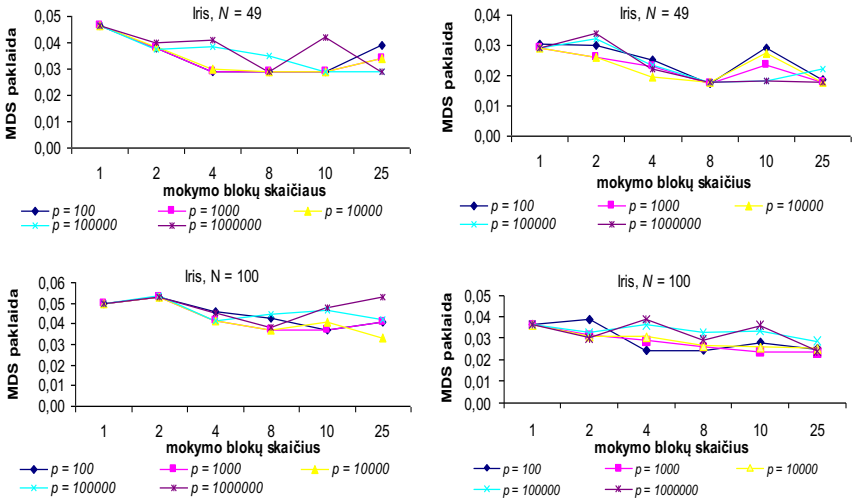
Eksperimentinio tyrimo metu buvo analizuojamas integruotas neuroninių dujų ir saviorganizuojančio neuroninio tinklo jungimas su daugiamačių skalių metodu, kuriame MDS paklaida minimizuojama SMACOF algoritmu (2.3.3 skyrelis). Dviejų vektorių kvantavimo metodų (SOM ir ND) su daugiamatėmis skalėmis integruoto junginio schema pateikta 3.4.2 skyrelio 3.10 paveiksle.

Tyrimuose naudotos duomenų aibės, turinčios tam tikrų specifinių savybių: Fišerio irisai [149; 4], hepta [212; 3], auto MPG [392; 7], vėžio [683; 9], chainlink [1000; 3], elipsoidai [1338; 100] ir rand_data1500 [1500; 5] (2.1 lentelė).

4.9.1. Nuoseklaus ir integruoto junginio palyginimas MDS paklaidos prasme

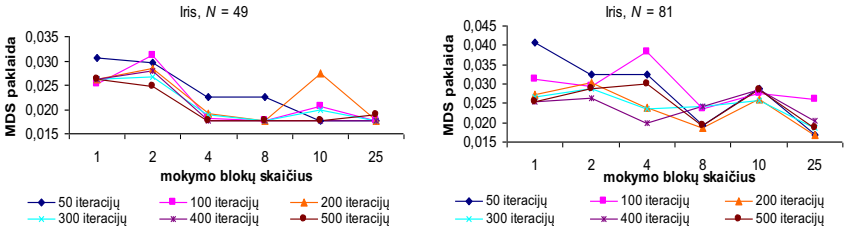
Kaip minėta 3.4.2 skyrelyje, integruotame junginyje gali atsitikti, kad priskiriant dvimačių vektorių pradines reikšmes, atsižvelgiant į prieš tai buvusį mokymo bloką, kelių vektorių visos koordinatės sutampa, todėl būtina ištirti, kaip mažo dydžio ε pridėjimas prie sutampančių koordinačių daro įtaką vizualizavimo rezultatams. Programiškai realizuota: $y(t,1)=y(t,1)+\varepsilon$, $y(t,2)=y(t,2)+\varepsilon$, čia $t-$ iteracijos numeris, $\varepsilon=\text{rand}()/p$, $\text{rand}()$ – atsitiktinai sugeneruotas skaičius, todėl kiekvieną kartą ε reikšmė gaunama vis kita. Tyrimo metu reikėjo nustatyti p reikšmę. Atlikti eksperimentai, kai $p=10^2,10^3,10^4,10^5,10^6$. 4.30 paveiksle pateikta MDS paklaidos priklausomybė nuo skirtingų mokymo blokų skaičiaus γ, esant įvairioms p reikšmėms. Jokių

esminių skirtumų tarp skirtingų p reikšmių panaudojimo nepastebėta (4.30 paveikslas). Tolimesni tyrimai atlikti, kai $p=10^3$.



4.30 pav. *p reikšmės įtaka MDS paklaidai: ND (kairėje) ir SOM (dešinėje)*

Kito eksperimento tikslas parodyti, kaip svarbu tinkamai parinkti MDS iteracijų skaičių priklausomai nuo mokymo blokų skaičiaus γ. Tyrimai atlikti, kai MDS iteracijų skaičius lygus 50, 100, 200, 300, 400 ir 500, o integruoto SOM ir MDS junginio blokų γ skaičius atitinkamai 1, 2, 4, 8, 10 ir 25, kai $\gamma=1$ – turime nuoseklų junginį. Vienos MDS iteracijos metu visų dvimačių vektorių koordinatės perskaičiuojamos vieną kartą.



4.31 pav. *MDS iteracijų priklausomybė nuo mokymo blokų skaičiaus*

Atlikus eksperimentus pastebėta tendencija, kad atlikus mažiau MDS iteracijų neuroninio tinklo mokymo procesą reikia skaidyti į daugiau mokymo

blokų, kad gautume tą pačią projekcijos paklaidą, ir priešingai, kuo daugiau atliekama MDS iteracijų, tuo reikia mažiau mokymo blokų (4.31 paveikslas).

Kiti eksperimentai buvo atlikti, kad išsiaiškintume, kuris iš dvimačių vektorių pradinių taškų parinkimo būdų nuosekliajame SOM ar ND ir MDS junginyje ar integruoto junginio pirmame bloke (kai taškai generuojami atsitiktinai iš intervalo (0;1), išdėliojami ant tiesės, panaudojant pagrindinių komponenčių ar didžiausių dispersijų metodą) yra geresnis, t. y. gaunama mažesnė projekcijos paklaida, bei kuris iš dvimačių vektorių koordinačių priskyrimas (pagal vidurio tašką ar pagal proporciją) integruotame junginyje, išskyrus pirmąjį mokymo bloką, yra tinkamesnis. Eksperimentai atlikti su įvairiu neuronų nugalėtojų skaičiumi.

Pateikiami eksperimentinių tyrimų rezultatai iris $[150;4]$, hepta $[212;3]$ ir rand_data1500 $[1500;5]$ duomenims. Neuroninis tinklas (SOM ir ND metodu) yra išmokytas atlikus 200 epochų ($\hat{e}=200$). Mokymo procesas integruotame junginyje dalinamas į γ = 2, 4, 8, 10, 25 blokus, o epochų skaičius ν' viename bloke atitinkamai lygus 100, 50, 25, 20 ir 8. Naudojant MDS metodą atliekama 100 iteracijų. Analizuojamų duomenų aibių MDS paklaidos $\hat{E}_{\text{MDS}}$ (2.9) rezultatų priklausomybė nuo pradinių dvimačių vektorių reikšmių parinkimo pateikta 4.13– 4.15 lentelėse. Naudojant atsitiktinį dvimačių vektorių taškų parinkimą buvo atlikta po 10 eksperimentų kiekvienai analizuojamų duomenų aibei ir paskaičiuotas gautų reikšmių vidurkis. Gauti rezultatai pateikti 4.13–4.15 lentelėse ir 4.32 paveiksle. Mažiausios paklaidos pažymėtos kursyvu, dažniausiai pasikartojančios paryškintu šriftu, o mažiausios pasikartojančios paklaidos pažymėtos paryškintu kursyvu. Neuronų skaičius N fiksuotas, o neuronų nugalėtojų skaičius r, gautas abiem vektorių kvantavimo metodais, yra arba tas pats, arba mažai skiriasi, todėl gautas MDS paklaidas $\hat{E}_{\text{MDS}}$ galime palyginti.

4.13 lentelė. *MDS paklaidos priklausomybė nuo pradinių reikšmių ir priskyrimo būdų iris duomenų aibei*

a) SOM ($E_{\mathrm{QE}} = 0{,}2225,\ r = 93$)

Nuoseklus			**Atsitiktinai**		**Ant tiesės**		**Pagal PK**		**Pagal dispersijas**	
			0,0363		0,0366		0,0276		***0,0265***	
Integruotas	γ	ν'	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***
	2	100	0,0385	0,0386	0,0484	0,0484	0,0395	0,0436	0,0438	0,0438
	4	50	0,0371	0,0373	***0,0265***	0,0271	0,0382	0,0269	0,0382	0,0382
	8	25	0,0335	0,0296	***0,0265***	***0,0265***	***0,0265***	***0,0265***	0,0347	***0,0265***
	10	20	0,0281	***0,0265***	0,0347	***0,0265***	***0,0265***	***0,0265***	***0,0265***	***0,0265***
	25	8	0,0298	0,0290	0,0347	***0,0265***	0,0347	***0,0265***	0,0347	***0,0265***

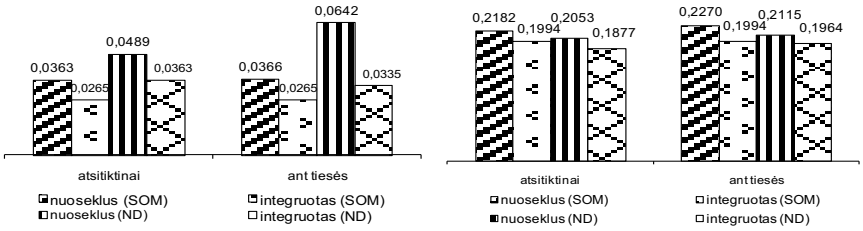
b) ND ($E_{\mathrm{QE}} = 0{,}0988,\ \ r = 94$)

Nuoseklus			**Atsitiktinai**		**Ant tiesės**		**Pagal PK**		**Pagal dispersijas**	
			0,0489		0,0642		***0,0335***		0,0358	
Integruotas	γ	ν'	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***
	2	100	0,0451	0,0452	0,0381	0,0561	***0,0335***	***0,0335***	***0,0335***	***0,0335***
	4	50	0,0399	0,0417	***0,0335***	***0,0335***	***0,0335***	***0,0335***	***0,0335***	***0,0335***
	8	25	0,0366	0,0363	***0,0335***	***0,0335***	***0,0335***	***0,0335***	***0,0335***	***0,0335***
	10	20	0,0392	0,0384	***0,0335***	***0,0335***	***0,0335***	0,0349	0,0349	0,0349
	25	8	0,0369	0,0388	0,0506	***0,0335***	***0,0335***	***0,0335***	***0,0335***	***0,0335***

Irisų duomenims MDS paklaida $\hat{E}_{\mathrm{MDS}}$ yra mažiausia, kai pradinės dvimačių vektorių taškų koordinatės yra gaunamos naudojant didžiausių dispersijų metodą, taikant SOM tinklą nuosekliajame junginyje ($\hat{E}_{\mathrm{MDS}} = 0{,}0265$) ir naudojant pagrindinių komponenčių metodą, ND ir MDS nuosekliajame junginyje ($\hat{E}_{\mathrm{MDS}} = 0{,}0335$), tačiau šios mažiausios MDS paklaidos integruotame junginyje gaunamos taikant kitus pradinių dvimačių vektorių reikšmių parinkimo būdus.

Iš 4.32 paveikslo matome, kad kai pradinių dvimačių vektorių reikšmių inicijavimas yra ant tiesės arba reikšmės parenkamos atsitiktinai, tai daugeliu

atvejų MDS paklaidos $\hat{E}_{\text{MDS}}$, gautos taikant integruotą junginį, yra mažesnės negu taikant nuoseklųjį junginį. Galime daryti išvadą, kad duomenų vizualizavimui integruotas junginys yra tinkamesnis negu nuoseklusis.



4.32 pav. *MDS paklaidos, gautos nuosekliuoju ir integruotu junginiu analizuojant iris duomenų aibę (kairėje) ir hepta duomenų aibę (dešinėje)*

$\hat{E}_{\text{MDS}}$ reikšmė yra šiek tiek didesnė junginiuose, kai vietoj SOM naudojamas neuroninių dujų metodas, tačiau kvantavimo paklaida E_{QE} (3.5) yra žymiai mažesnė, todėl neuroninių dujų metodas yra tinkamesnis naudoti junginiuose. Kai integruotame junginyje blokų skaičius γ didėja, tai gauta MDS paklaida $\hat{E}_{\text{MDS}}$ nežymiai svyruoja, tačiau ji nėra didesnė už reikšmę, gautą nuosekliuoju junginiu.

Hepta duomenims mažiausia MDS paklaida $\hat{E}_{\text{MDS}} = 0{,}1994$ gauta integruotu SOM tinklo ir MDS junginiu nepriklausomai nuo dvimačių vektorių pradinių reikšmių inicijavimo būdo. Naudojant ND metodą, dažnai pasikartojanti reikšmė $\hat{E}_{\text{MDS}} = 0{,}1964$ gauta nuosekliuoju junginiu, kai pradinių reikšmių parinkimui taikomas didžiausių dispersijų arba pagrindinių komponenčių metodas. Tokios pačios reikšmės gaunamos ir integruotu junginiu, kai pradinių reikšmių parinkimas yra ant tiesės. Jeigu naudojame atsitiktinį pradinių reikšmių parinkimą, mažiausia MDS paklaida $\hat{E}_{\text{MDS}} = 0{,}1877$ yra gaunama integruotu junginiu, kai $\gamma = 2$.

4.14 lentelė. *MDS paklaidos priklausomybė nuo pradinių reikšmių ir priskyrimo būdų hepta duomenų aibei*

a) SOM ($E_{\mathrm{QE}} = 0{,}3115,\ r = 86$)

Nuoseklus			**Atsitiktinai**		**Ant tiesės**		**Pagal PK**		**Pagal dispersijas**	
			0,2182		0,2270		0,2042		0,2042	
Integruotas	***γ***	***ν′***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***
	2	100	0,2004	0,2066	***0,1994***	***0,1994***	***0,1994***	***0,1994***	***0,1994***	***0,1994***
	4	50	0,2078	0,2345	***0,1994***	***0,1994***	***0,1994***	0,2042	0,2270	0,2487
	8	25	***0,1994***	0,2109	***0,1994***	0,2270	***0,1994***	***0,1994***	***0,1994***	0,2270
	10	20	***0,1994***	0,2051	***0,1994***	0,2042	***0,1994***	***0,1994***	***0,1994***	0,2042
	25	8	***0,1994***	0,2081	***0,1994***	***0,1994***	***0,1994***	***0,1994***	***0,1994***	***0,1994***

b) ND ($E_{\mathrm{QE}} = 0{,}1765,\ r = 94$)

Nuoseklus			**Atsitiktinai**		**Ant tiesės**		**Pagal PK**		**Pagal dispersijas**	
			0,2053		0,2115		**0,1964**		**0,1964**	
Integruotas	***γ***	***ν′***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***
	2	100	*0,1877*	*0,1877*	0,2043	0,2043	**0,1964**	**0,1964**	0,2043	0,2043
	4	50	0,2084	0,2084	0,2322	0,2322	0,2043	0,2043	0,2056	0,2056
	8	25	0,2194	0,2194	**0,1964**	**0,1964**	**0,1964**	**0,1964**	**0,1964**	**0,1964**
	10	20	0,2008	0,2052	**0,1964**	**0,1964**	**0,1964**	**0,1964**	**0,1964**	**0,1964**
	25	8	0,2115	0,2031	0,2115	**0,1964**	0,2115	**0,1964**	0,2115	**0,1964**

rand_data1500 duomenims negalime pastebėti iris ir hepta duomenų aibių MDS paklaidos $\hat{E}_{\mathrm{MDS}}$ mažiausių reikšmių gavimo tendencijos, tačiau šios paklaidos mažėja naudojant integruotus junginius, kai blokų skaičius γ didėja. Palyginus du priskyrimo būdus (pagal proporciją ir pagal vidurio tašką), galime daryti išvadą, kad nėra didelio skirtumo, kurį iš būdų naudoti integruotame junginyje.

Kvantavimo paklaidos gautos ND metodu yra žymiai mažesnės negu taikant SOM tinklą, todėl, kai neuronų nugalėtojų skaičius yra toks pats, tai

ND metodas tinkamesnis naudoti junginyje, nors MDS paklaida $\hat{E}_{\text{MDS}}$ yra šiek tiek didesnė.

4.15 lentelė. *MDS paklaidos priklausomybė nuo pradinių reikšmių ir priskyrimo būdų rand_data1500 duomenų aibei*

a) SOM ($E_{\text{QE}} = 0{,}21395,\ r = 394$)

Nuoseklus			**Atsitiktinai**		**Ant tiesės**		**Pagal PK**		**Pagal dispersijas**	
			0,3223		0,3189		0,3153		*0,3140*	
Integruotas	γ	ν'	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***
	2	100	0,3244	0,3247	0,3252	0,3237	0,3241	0,3239	0,3241	0,3216
	4	50	0,3217	0,3225	0,3217	0,3220	0,3217	0,3220	0,3218	0,3229
	8	25	0,3176	0,3200	0,3178	*0,*3148	0,3176	0,*3142*	0,3177	0,3206
	10	20	0,3157	0,3155	0,3164	0,3162	0,3164	0,3164	0,3164	0,3167
	25	8	0,3159	**0,3161**	0,3162	**0,3161**	0,3160	**0,3161**	0,3162	**0,3161**

b) ND ($E_{\text{QE}} = 0{,}1380,\ r = 400$)

Nuoseklus			**Atsitiktinai**		**Ant tiesės**		**Pagal PK**		**Pagal dispersijas**	
			0,3202		0,3223		0,3119		*0,3103*	
Integruotas	γ	ν'	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***	***pagal vidurio tašką***	***pagal proporciją***
	2	100	0,3192	0,3143	0,3179	0,3179	0,3125	0,3123	0,3140	0,3116
	4	50	0,3168	0,3159	0,3160	0,3160	0,3183	0,3187	**0,3115**	0,3140
	8	25	0,3129	0,3122	0,3132	0,3157	**0,3115**	**0,3115**	*0,3103*	**0,3115**
	10	20	0,3124	0,3131	0,3116	0,3223	0,3116	0,3119	**0,3115**	*0,3103*
	25	8	**0,3115**	**0,3115**	**0,3115**	0,3220	**0,3115**	**0,3115**	**0,3115**	**0,3115**

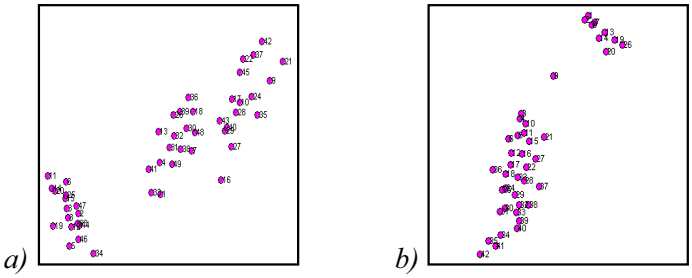
Jeigu $\hat{E}_{\text{MDS}}$ reikšmė, gauta nuosekliuoju junginiu, yra pakankamai maža, kai dvimačių vektorių pradinės reikšmės parenkamos naudojant pagrindinių komponenčių ar didžiausių dispersijų metodą, kartais vis tiek įmanoma ją dar labiau sumažinti naudojant integruotą junginį. Jeigu pradinės reikšmės generuojamos atsitiktinai arba parenkamos ant tiesės, tai geriau naudoti integruotą junginį negu nuoseklųjį.

4.9.2. Vaizdų kokybės priklausomybė nuo integruoto junginio mokymo blokų skaičiaus

Šiame skyrelyje pateikiama daugiamačių duomenų vaizdų projekcijų plokštumoje, gautų integruoto junginio algoritmu, analizė. Tyrimo tikslas – ištirti plokštumoje gautų vaizdų kokybės priklausomybę nuo neuronų nugalėtojų ir mokymo blokų skaičiaus. Eksperimentai buvo atlikti naudojant šias duomenų aibes: Fišerio irisus [149; 4], hepta [212; 3], chainlink [1000; 3] ir elipsoidus [1338; 100].

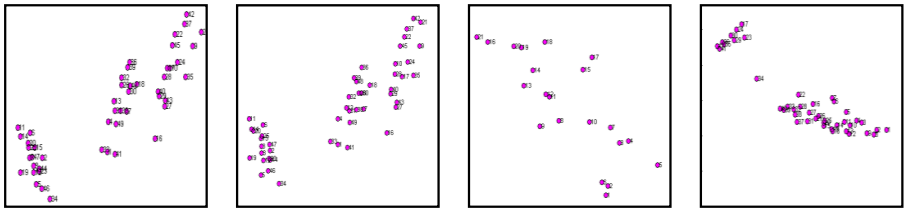
Eksperimentinis tyrimas atliktas, kai mokymo blokų skaičius $\gamma = 1$, 2, 4, 8, 10 ir 25, o $\hat{e} = 200$. Mokymo blokų ir atliekamų epochų skaičius nurodomas prieš mokymą. Daugiamačių skalių metodu buvo atliekama po 100 iteracijų. Dvimačių vektorių pradinės koordinatės parinktos atsitiktinai. Buvo vertinama, kaip keičiasi analizuojamų duomenų vaizdas didinant neuroninio tinklo mokymo blokų ir neuronų skaičių.

4.33 paveiksle pateiktas irisų duomenų aibės vizualizavimas MDS metodu, kai atliekamas pirmas mokymo blokas SOM ir ND metodu, t. y. nuoseklusis junginys. 4.34–4.37 paveiksluose pateikti vaizdai, gauti integruoto junginio įvairiuose blokuose.

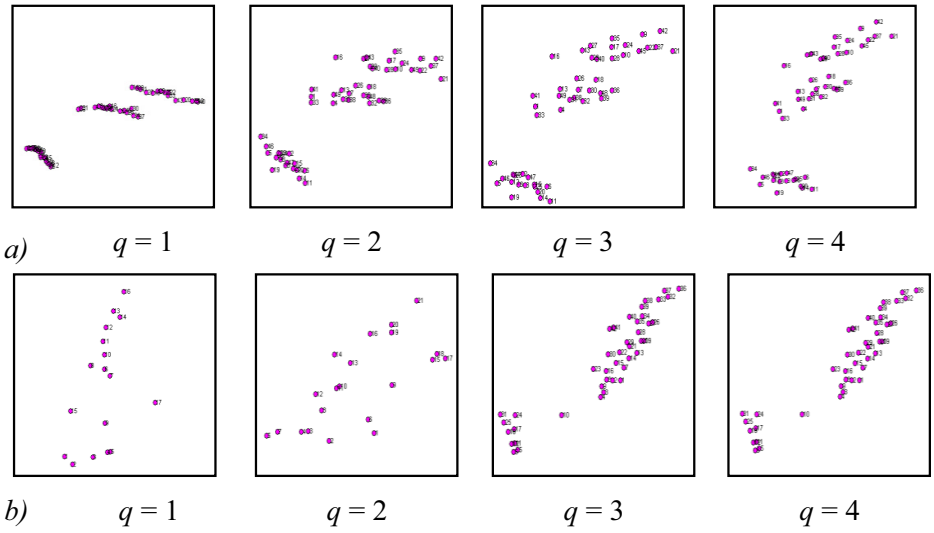


a) *b)*

4.33 pav. *Irisų duomenų aibės vizualizavimas, kai $\gamma = 1$. Neuroninio tinklo mokymas: a) ND metodu ($\hat{E}_{\mathrm{MDS}} = 0{,}0479$); b) SOM ($\hat{E}_{\mathrm{MDS}} = 0{,}0284$)*



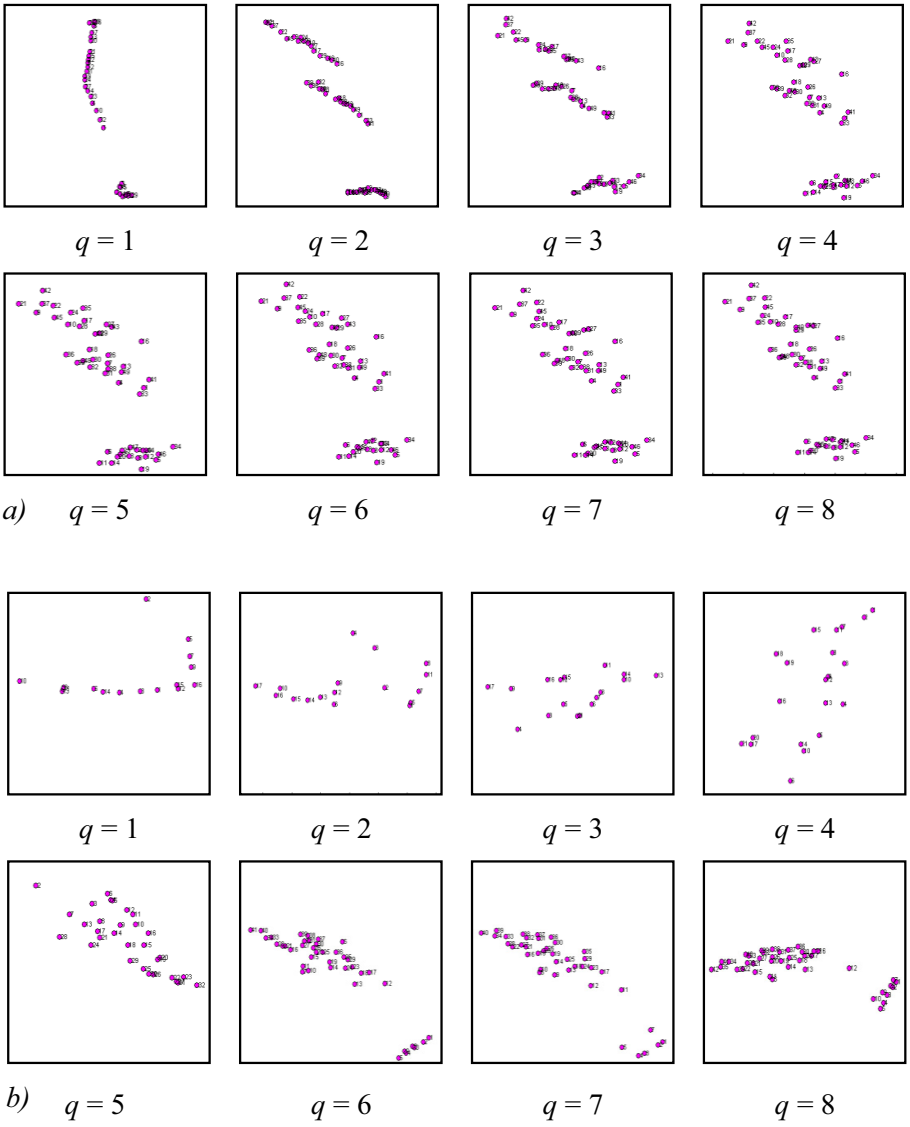
4.34 pav. *Irisų duomenų aibės vizualizavimas, kai $\gamma = 2$. Neuroninio tinklo mokymas: a) ND metodu; b) SOM*



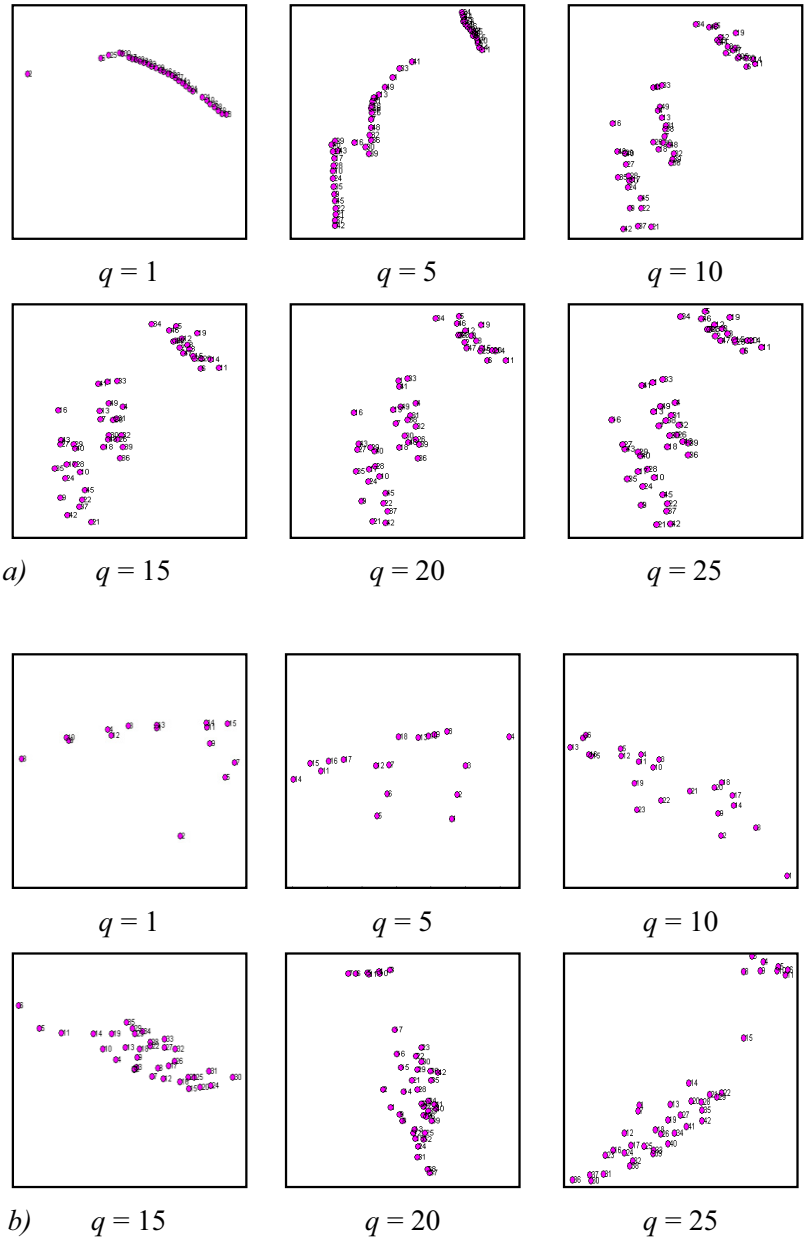
4.35 pav. *Irisų duomenų aibės vizualizavimas, kai $\gamma = 4$. Neuroninio tinklo mokymas: a) ND metodu; b) SOM*

Eksperimentų rezultatai parodė, kad analizuojamų duomenų struktūra, naudojant neuroninių dujų ir daugiamačių skalių metodų junginį, kai $\gamma = 2$ ir $\gamma = 4$, jau matoma vaizde, gautame po pirmo mokymo bloko, kai $\gamma = 8$ po antrojo mokymo bloko. Taikant saviorganizuojančio neuroninio tinklo ir daugiamačių skalių junginį, analizuojamų duomenų struktūra gerai matoma tik antrame mokymo bloke, kai $\gamma = 2$, trečiame mokymo bloke, kai $\gamma = 4$, ir šeštame, kai $\gamma = 8$. Neuroninio tinklo mokymą suskaidžius į 25 mokymo blokus, analizuojamų duomenų struktūra, naudojant ND ir MDS junginį, jau gerai matoma po 10 mokymo blokų, o naudojant SOM ir MDS junginį – po

15–20 mokymo blokų. Panašūs rezultatai gauti ir kitoms analizuojamų duomenų aibėms.



4.36 pav. *Irisų duomenų aibės vizualizavimas, kai γ = 8. Neuroninio tinklo mokymas: a) ND metodu; b) SOM*



4.37 pav. *Irisų duomenų aibės vizualizavimas, kai γ = 25. Neuroninio tinklo mokymas: a) ND metodu; b) SOM*

Atlikus gautų vaizdų lyginamąją analizę, galima daryti išvadą, kad taikant ND ir MDS junginį analizuojamų duomenų struktūra jau gerai matoma, kai mokymo blokų skaičius $q<1/3\gamma$, o taikant SOM ir MDS junginį, tik, kai mokymo blokų skaičius $q > 2/3\gamma$.

4.10. Ketvirtojo skyriaus rezultatai ir išvados

Eksperimentiškai ištirta kvantavimo paklaidos priklausomybė nuo kelių neuroninių dujų metodo mokymo taisyklės parametrų. Tyrimai parodė, kad mažiausia kvantavimo paklaida gaunama, kai parametras $E_{\mathrm{f}} = 0{,}1$, $\lambda_{\mathrm{f}} = 0{,}01$, o atliekamų mokymo epochų skaičius $\hat{e} = 200$.

Nagrinėta neuronų skaičiaus parinkimo strategija dviejuose kvantavimo metoduose – saviorganizuojančiame neuroniniame tinkle bei neuroninių dujų metode. Pasiūlytas būdas, pagal kurį parenkamas neuronų skaičius atsižvelgiant į analizuojamų duomenų specifiką. Ištyrus, iš kiek neuronų reikia sudaryti tinklą, kad analizuojama vektorių aibė būtų sumažinta taip, kad kvantavimo paklaida nuo mažiausios, kuri gaunama, kai neuronų skaičius sutampa su analizuojamų vektorių skaičiumi, skirtųsi mažu dydžiu ε. Nustatyta, kad esant didesniam vektorių dimensijų skaičiui, siekiant, kad kvantuoti vektoriai kuo tiksliau atspindėtų analizuojamų vektorių savybes, tinklą reikia sudaryti iš daugiau neuronų, nei kai vektorių skaičius yra mažesnis. Lyginant SOM ir ND rezultatus, SOM tinklą užtenka sudaryti iš žymiai mažesnio neuronų skaičiaus nei ND metode norint pasiekti tą patį tikslumą. Pavyzdžiui, kai $m = 50$, $n = 20$, $\varepsilon = 0{,}2$ SOM metode užtenka šiek tiek daugiau nei 20 % neuronų, o ND metode tam pačiam tikslumui pasiekti reikia beveik 80 % neuronų.

Ištyrus kvantavimo paklaidos priklausomybę nuo neuronų nugalėtojų ir mokymo epochų skaičiaus, gautos šios išvados:

- realaus pobūdžio duomenims neuroninių dujų metodu gauta kvantavimo paklaida yra ne mažiau nei 5 kartus mažesnė negu taikant saviorganizuojantį neuroninį tinklą, kai neuronų nugalėtojų skaičius mažai skiriasi; atsitiktinai generuotiems duomenims, kai neuronų nugalėtojų yra nedaug, SOM ir ND gautos kvantavimo paklaidos skiriasi nežymiai, tačiau esant didesniam neuronų nugalėtojų skaičiui, ND gauta kvantavimo paklaida mažesnė nei gauta SOM metodu; taigi neuroninių dujų metodas yra tinkamesnis vektoriams kvantuoti negu saviorganizuojantys neuroniniai tinklai, ir jis tinkamesnis naudoti junginyje, nors MDS paklaida $\hat{E}_{\mathrm{MDS}}$ yra šiek tiek didesnė;

- santykis tarp neuronų ir neuronų nugalėtojų neuroninių dujų metodu yra didesnis negu saviorganizuojančiu neuroniniu tinklu; neuroninių dujų metodu apie 80 % neuronų tampa neuronais nugalėtojais, o SOM tik apie 50 %;

- didinant neuronų ir atliekamų epochų skaičių, kvantavimo paklaida mažėja tiek ND, tiek SOM metodu; visiems analizuojamiems duomenims kvantavimo paklaida skiriasi, kai yra atliekama 10 epochų palyginus su paklaida, gauta atlikus 200 epochų, tačiau atliekant 20 epochų, paklaida beveik nesiskiria nuo gautos atlikus 200 epochų, todėl pakanka atlikti 20 epochų;

- saviorganizuojantys neuroniniai tinklai yra labiau tinkami duomenų klasterizavimui negu neuroninių dujų metodas.

Šiame skyriuje taip pat atlikta kvantavimo paklaidos priklausomybė nuo neuronų pradinių reikšmių į tinklą pateikimo tvarkos analizė. Nagrinėti trys vektorių pateikimo į tinklą būdai. Eksperimentiniai rezultatai parodė, kad:

- mažoms duomenų aibėms, tokioms kaip iris ar hepta, kvantavimo paklaida nepriklauso nuo pradinių neuronų reikšmių parinkimo; kai

duomenų aibės yra didesnės (chainlink, elipsoidai) ir neuronų yra daugiau, SOM metodu gauta kvantavimo paklaida labiau priklauso nuo pradinių neuronų reikšmių parinkimo negu gauta ND metodu;

- visiems analizuojamiems duomenims taikant trečią vektorių pateikimo į tinklą būdą SOM ir ND metodu, kvantavimo paklaida priklauso nuo vektorių pateikimo tvarkos; mažoms duomenų aibėms, tokioms kaip iris, hepta ar vėžio galime naudoti tiek antrą, tiek trečią būdą, nes kvantavimo paklaidos mažai skiriasi; didesnėms duomenų aibėms (chainlink ir elipsoidai) ND ir SOM metodu, naudoti antrą vektorių pateikimo į tinklą būdą netikslinga, nes jis duoda prastesnį rezultatą negu naudojant trečią vektorių pateikimo į tinklą būdą.

Buvo ištirta tinklo mokymo skaičiavimo laiko priklausomybė nuo neuronų ir atliekamų epochų skaičiaus. Galime daryti bendrą išvadą, kad tinklo mokymo skaičiavimo laikas priklauso nuo neuronų skaičiaus ir didėja pagal eksponentine funkciją, taikant SOM metodą, o pagal tiesinę funkciją, taikant ND, bei atliekamų epochų skaičiaus ir didėja pagal tiesinę funkciją, taikant ir SOM, ir ND metodus.

Atliktų nuoseklaus junginio tyrimų rezultatai parodė, kad visiems analizuotiems duomenims:

- pagal Konigo matą (E_{KM}) kaimyniškumo išlaikymas yra daug tikslesnis, kai neuronai nugalėtojai, gauti SOM metodu, yra vizualizuojami MDS metodu. SOM geriau išlaiko kaimyniškumus (artimumus) negu naudojant ND metodą;

- Spirmano koeficiento (ρ_{Sp}) reikšmės yra didesnės negu 0,85 (dirbtinai sugeneruotas duomenų aibėms didesnės negu 0,78), todėl vizualizavimo kokybė yra gera atstumų išlaikymo prasme, tačiau pagal Spirmano koeficientą yra sunku pasakyti, kuris iš metodų (ND ar SOM) yra geresnis;

- kai neuronai nugalėtojai, gauti ND ir SOM, yra atvaizduojami plokštumoje, MDS paklaidos $\hat{E}_{\text{MDS}}$ yra apytiksliai lygios duomenų aibėms, kurių dimensijų skaičius $n = 3$; kai vizualizuojami neuronai nugalėtojai gauti SOM metodu, $\hat{E}_{\text{MDS}}$ reikšmės yra mažos duomenims, kurių dimensijų skaičius $n = 3, 7, 10$;
- abu kvantavimo metodai ND ir SOM yra tinkami jungimui su MDS metodu;
- atvaizdavus duomenis plokštumoje, jų struktūra gerai matoma esant dar mažam neuronų nugalėtojų skaičiui gautu ND metodu, o SOM duomenų struktūra gerai matoma tik tada, kai neuronų nugalėtojų skaičius daug didesnis. Neuroninių dujų metodu duomenų struktūra gerai atskleidžiama atlikus vos 10 epochų, o saviorganizuojančiu neuroniniu tinklu reikia atlikti 200 epochų; kad analizuojamų duomenų struktūra būtų gerai matoma taikant saviorganizuojančius neuroninius tinklus galimos dvi alternatyvos: didinti neuronų skaičių, kai atliekamų epochų skaičius yra mažas arba didinti epochų skaičių, kai neuronų skaičius yra mažas.

Atlikus integruoto junginio tyrimus pastebėta tendencija, kad atlikus mažiau MDS iteracijų, neuroninio tinklo mokymo procesą reikia skaidyti į daugiau blokų, kad gautume tą pačią projekcijos paklaidą, ir priešingai, kuo daugiau atliekama MDS iteracijų, tuo mažiau reikia mokymo blokų.

Eksperimentiškai nustatyta, kad:

- dvimačių vektorių pradinių taškų koordinačių priskyrimas pagal proporciją ir pagal vidurio tašką gali būti naudojamas integruotame junginyje, nes gautuose rezultatuose nėra pastebėta esminių skirtumų;

- kai pradinės dvimačių vektorių reikšmės generuojamos atsitiktinai arba parenkamos ant tiesės, geriau naudoti integruotą junginį negu nuoseklųjį, nes MDS paklaida yra mažesnė;
- MDS paklaidos reikšmė, gauta nuosekliuoju junginiu, yra pakankamai maža, kai dvimačių vektorių pradinės reikšmės parenkamos naudojant pagrindinių komponenčių ar didžiausių dispersijų metodą, tačiau kartais vis tiek yra įmanoma ją dar sumažinti naudojant integruotą junginį;
- taikant ND ir MDS integruotą junginį, analizuojamų duomenų struktūra jau gerai matoma, kai mokymo blokų skaičius $q < 1/3\gamma$, o taikant SOM ir MDS integruotą junginį, mokymo blokų skaičius $q > 2/3\gamma$, čia γ – visų mokymo blokų skaičius.

5

Bendrosios išvados

Atlikti tyrimai atskleidė dviejų vektorių kvantavimo metodų – saviorganizuojančio neuroninio tinklo ir neuroninių dujų – jungimo su daugiamatėmis skalėmis naujas galimybes. Eksperimentinių tyrimų rezultatai leido daryti šias išvadas:

1. Neuroninių dujų metodu apie 80 % neuronų tampa nugalėtojais, o SOM tik apie 50 %, todėl SOM labiau tinkamas duomenims klasterizuoti. Tačiau kvantavimo paklaida esant tam pačiam neuronų nugalėtojų skaičiui neuroninių dujų metodu visiems analizuotiems duomenims visada mažesnė negu taikant SOM. Neuroninių dujų metodas tinkamesnis daugiamačiams duomenims kvantuoti, o tuo pačiu naudoti junginyje su daugiamačių skalių metodu.
2. Junginyje su daugiamačių skalių metodu naudojant SOM tinklą yra geriau išlaikomi kaimynystės ryšiai tarp taškų, pereinant iš daugiamatės erdvės į dvimatę erdvę Konigo mato prasme negu naudojant neuroninių dujų metodą. Spirmano koeficiento prasme šių

abiejų kvantavimo metodų naudojimas yra lygiavertis. MDS paklaida duomenims, kurių dimensijų skaičius $n = 5, 7, 10$, yra mažesnė, kai junginyje naudojamas SOM, nei ND metodas.

3. Taikant pasiūlytą integruotą neuroninių dujų ir daugiamačių skalių metodų junginį analizuojamų duomenų struktūra jau gerai matoma, kai mokymo blokų skaičius $q < 1/3\gamma$, o taikant SOM ir daugiamačių skalių metodo integruotą junginį, mokymo blokų skaičius turi būti $q < 2/3\gamma$, čia γ – visų mokymo blokų skaičius.

4. Pasiūlytas pradinių dvimačių vektorių koordinačių priskyrimas pagal vidurinį tašką integruotame junginyje, išskyrus pirmąjį mokymo bloką, gali būti naudojamas kaip alternatyva priskyrimui pagal proporciją, kadangi nepastebėta gautų rezultatų esminių skirtumų.

5. MDS paklaida, gauta nuosekliuoju junginiu, yra mažiausia, kai dvimačių vektorių pradinės reikšmės parenkamos pagal dvi pagrindines komponentes arba dvi didžiausias dispersijas turinčias komponentes, tačiau kartais įmanoma ją dar sumažinti naudojant integruotą junginį. Kai dvimačių vektorių pradinės reikšmės generuojamos atsitiktinai arba parenkamos ant tiesės, tai geriau naudoti integruotą junginį negu nuoseklųjį, nes MDS paklaida mažesnė.

Literatūra

Agrafiotis, D. K.; Rassokhin, D. N.; Lobanov, V. S. 2001. Multidimensional scaling and visualization of large molecular similarity tables. Journal of Computational Chemistry, 22, 488–500.

Alhoniemi, E.; Himberg, J.; Parhankangas, J.; Vesento, J. 2000. *SOM Toolbox for Matlab 5*. Paimta 2011 m. 03 06 d. iš http://www.cis.hut.fi/projects/ somtoolbox/

Asuncion, A.; Newman, D. J. 2007. *UCI Machine Learning Repository*. Paimta 2011 m. 03 23 d. iš http://www.ics.uci. edu/~mlearn/MLRepository.htm

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

Becker, R. A.; Cleveland, W. S. 1996. The design and control of Trellis display. Journal of Computational and Statistical Graphics, 5, 123–155.

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

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

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

Burt, C. 1949. Alternative methods of factor analysis and their relation to Pearson's method of principal axes. Brit. J. psychol., Statisti, 2, 98–121.

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

Chernoff, H. 1973. The use of faces to represent points in k-dimensional space graphically. Journal of the American Statistical Association, 68, 361–368.

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

de Silva, V.; Tenenbaum, J. B. 2003. Global versus local methods for nonlinear dimensionality reduction. In S. Becker, S. Thrun, K. Obermayer (Eds.). Advances in Neural Information Processing Systems, 15, 721–728.

Delicado, P. 1997. *Another Look at Principal Curves and Surfaces, Preprint, presented at the 1997 Joint Statistical Meeting*. Paimta 2011 m. 03 23 d. iš http://www-eio.upc.es/~delicado/ my-public-files/prcv_JMVA.pdf

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

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

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

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. 2004. Visualization of correlation-based environmental data. Environmetrics, 15, 827–836.

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.; Medvedev, V. 2007. Dimension Reduction and Data Visualization Using Neural Networks. *Emerging Artificial Intelligence Applications in Computer Engineering. Real Word AI Systems with Applications in eHealth. Frontiers in Artificial Intelligence and Applications*, T. 160, 25–49.

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

Estévez, P. A.; Figueroa, C. J.; Saito, K. 2005. Cross-Entropy Embedding of High-Dimensional Data Using the Neural Gas Model. Neural Networks, 18, 727–737.

FCPS. 2011. *Fundamental Clustering Problems*. Paimta 2011 m. 02 22 d. iš http://www.uni-marburg.de/fb12/datenbionik/data/

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

Grinstein, G. G.; Ward, M. O. 2002. Introduction to Data Visualization, Information Visualization in data Mining and Knowledge Discovery. (U. Fayyad; G. Grinstein; A. Wierse, Mont.) Information Visualization in Data Mining and Knowledge Discovery.

Grinstein, G. G.; Hoffman, P. E.; Picket, R. M. 2002. Benchmark Development for the Evaluation of Visualization for data Mining, Information Visualization in Data Mining and Knowledge Discovery. (U. Fayyad; G. Grinstein; A. Wierse; M. Kaufmann, Mont.)

Grinstein, G.; Trutschl, M.; Cvek, U. 2001. High-Dimensional Visualizations, In proceedings of Workshop on Visual data Mining. ACM Conference on Knowledge Discovery and data Mining, 1–14.

Han, J.; Kamber, M. 2006. *Data Mining, Concepts and Techniques.* London: Elsevier.

Handl, J.; Knowles, J. 2010. *Cluster generators: synthetic data for the evaluation of clustering algorithms*. Paimta 2011 m. 03 22 d. iš http://dbkgroup.org/handl/ generators/

Hastie, T. 1984. *Principal Curves and Surfaces, PhD Dissertation.* Stanford, California: Stanford Linear Accelerator Center, Stanford University.

Hoffman, P. E.; Grinstein, G. G. 2002. A Survey of Visualizations for High-Dimensional Data Mining. Information Visualization in Data Mining and Knowledge Discovery.

Hotelling, H. 1933. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417–441, 498–520.

Inselberg, A. 1981. *n-dimensional graphics, part I—lines and hyperplanes.* Los Angeles: Technical report G320-2711, IBM Los Angeles Scientific Center.

Ivanikovas, S. 2009. *Lygiagrečių skaičiavimų taikymo daugiamačiams duomenims vizualizuoti problemos. Daktaro disertacija.* Vilnius: Technologija.

Yacoub, M.; Frayssinet, D.; Badran, F.; Thiria, S. 2000. *Clustering and Classification Based on Expert Knowledge Propagation Using a probabilistic Self-Organizing Map: Application to Geophysics, Data analysis: scientific modelling and practical application.* Springer.

Yeung, K. Y.; Ruzzo, W. L. 2001. *An empirical study on Principal Component Analysis for clustering gene expression data.* Technical Report UW-CSE-01-04-02, University of Washington.

Jansson, M.; Johansson, J. 2003. *Interactive Visualization of Statistical Data using Multidimensional Scaling Techniques , Ph.D thesis*. Paimta 2011 m. 03 23 d. iš http://www.ep. liu.se /exjobb/ itn/2003/mt/008/

Karbauskaitė, R. 2010. *Daugiamačių duomenų vizualizavimo metodų, išlaikančių lokalią struktūrą, analizė. Daktaro disertacija.* Vilnius: Vytauto Didžiojo universitetas.

Karbauskaitė, R.; Dzemyda, G. 2009. Topology Preservation Measures in the Visualization of Manifold-Type Multidimensional Data. Informatica, 20(2), 235–253.

Karbauskaitė, R.; Kurasova, O.; Dzemyda, G. 2007. Selection of the Number of Neighbours of Each Data Point for the Locally Linear Embedding Algorithm, Information Technology and Control. 36(4), 359–364.

Kaski, S. 1997. *Data Exploration Using Self-Organizing Maps PhD thesis*. (Helsinki University of Technology, Department of Computer Science and Engineering) Paimta 2011 m. 03 23 d. iš http://www.cis.hut.fi/~sami/thesis/

Kaufman, L.; Rousseeuw, P. J. 1987. Clustering by means of medoids. Statistical Data Analysis based on the L1 Norm, 405–416.

Klock, H.; Buhmann, J. 2000. Data visualization by multidimensional scaling: A deterministic annealing approach. Pattern Recognition, 33, 651–669.

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

Konig, A. 2000. Interactive Visualization and Analysis of Hierarchical Neural Projections for Data Mining. IEEE Transactions on Neural Networks, 11(3), 615–624.

Kraus, M.; Ertl, T. 2001. Interactive Data Exploration with Customized Glyphs. In Proceedings of WSCG '01, 20–23.

Kruskal, J. B. 1972. Linear transformations of multivariate data to reveal clustering. In Multidimentional Scaling:Theory and Application in the Behavioural Sciences, I.

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

Kvedaras, B.; Sapagovas, M. 1974. *Skaičiavimo metodai.* Mintis.

Lorose, D. T. 2004. *Discovery Knowledge in Data: An Introduction to Data Mining.* Wiley-Intersience.

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

Marcinkevičius, V. 2010. *Netiesinės daugiamačių duomenų projekcijos metodų savybių tyrimas ir funkcionalumo gerinimas. Daktaro disertacija.* Vilnius: Matematikos ir Informatikos Institutas.

Martinetz, T. M.; Schulten, K. J. 1991. A neural-gas network learns topologies, in Artificial Neural Networks. (V. Kohonen; V. Mäkisara; O. Simula; J. Kangas, Mont.) 397–402.

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. Daktaro disertacija.* Vilnius: Technika.

Michalski, R. S. 1978. *A planar geometric model for representing multidimensional discrete spaces and multiple-valued logic functions.* Technical Report UIUCDCSR-78-897, University of Illinois at Urbaba-Champaign.

Milligan, G. W.; Cooper, M. C. 1985. An examination of procedures for determining the number of clusters in a data set. Psychometrika, 50, 159–179.

Murtagh, F.; Hernindez-Pajares, M. 1995. The Kohonen self-organizing map method: an assessment. Journal of Classification, 12, 165–190.

Naud, A. 2004. Visualization of high-dimensional data using an association of multidimensional scaling to clustering. *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.

Opitz, O.; Hilbert, A. 2000. *Visualization of Multivariate Data by Scaling and Property Fitting, Data analysis: scientific modeling and practical application.* (W. Gaul; O. Opitz; M. Schader, Mont.) Studies in Classification, Data Analysis and Knowledge Organization. Springer.

Pan, Z. 2001. *Principal Component Analysis Based Visualization and Human Melanoma Classification*. Paimta 2011 m. 03 23 d. iš http://www.cs.umd.edu/class/spring2001/ cmsc838b/Project/ Zhijian_Pan/pca.pdf

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, 2, 559–572.

Podlipskytė, A. 2003. *Daugiamačių duomenų vizualizacija ir jos taikymas biomedicininių duomenų analizei. Daktaro disertacija.* Kaunas: Vytauto Didžiojo universitetas.

Rabenhorst, D. A. 1994. Interactive exploration of multidimensional data. *In Proccedings of the In Proceedings of the SPIE Symposium on Electronic Imaging*, *2179*, 277–286.

Raudys, Š.; Kaan, B. Ö.; Babalik, A.; Denisov, V.; Bielskis, A. A. 2007. Classifiers Fusion in Recognition of Wheat Varieties. Lecture Notes in Computer Science, 4472, 62–71.

Roweis, S. T.; Saul, L. K. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, 290(5500), 2323–2326.

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

Šaltenis, V.; Aušraite, J. 2002. Data Visualization: ideas, methods, and problems. Informatics in education, 1, 129–148. ISSN 1648-5831.

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

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

Tenenbaum, J. B.; de Silva, V.; Langford, J. C. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 290(5500), 2319–2323.

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

Unwin, A.; Theus, M.; Hofmann, H. 2006. *Graphics of Large Datasets Visualizing a Million, Series: Statistics and Computing* (T. XIV).

Vesento, J. 2001. Importance of Individual Variables in the k-Means Algorithm. *In Proccedings of the PAKDD 2001,* Hong Kong, China. 513–518.

Wegman, E. J.; Luo, Q. 1996. *High Dimensional Clustering Using Parallel Coordinates and the Grand Tour.* Technical Report, 124, Center for Computational Statistics, George Mason University.

Williams, M.; Munzner, T. 2004. *Steerable, Progressive Multidimensional Scaling. Information Visualization, INFOVIS 2004.*

Žilinskas, A.; Žilinskas, J. 2008. A hybrid method for multidimensional scaling using city-block distances. Mathematical Methods of Operations Research, 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. Journal of Global Optimization, 38 (4), 581–596.

Autoriaus publikacijų sąrašas disertacijos tema

Straipsniai recenzuojamuose periodiniuose mokslo žurnaluose

A 1. Kurasova, O.; Molytė A., 2008. Neuronų skaičiaus parinkimas vektorių kvantavimo metoduose, *Lietuvos matematikos rinkinys,* T. 48/49, Spec. Nr., 354–359. ISSN 0132-2818.

A 2. Molytė, A.; Kurasova O., 2009. Vektorių kvantavimo metodų ir daugiamačių skalių junginys daugiamačiams duomenims vizualizuoti, *Informacijos mokslai*, Vilnius, Vilniaus universitetas, T. 50, 340–346. ISSN 1392-0561.

A 3. Kurasova, O.; Molytė, A., 2009. Combination of vector quantization and visualization. In: P. Perner (Ed.,) Machine Learning and Data Mining in Pattern Recognition - MLDM 2009, *Lecture Notes in Artificial Intelligence*, Springer Verlag, Heidelberg, Vol. 5632, 29–43. ISSN 0302-9743, ISBN 978-3-642-03069-7.

A 4. Kurasova, O.; Molytė, A., 2011. Quality of quantization and visualization of vectors obtained by neural gas and self-organizing Map, *Informatica*, Vilnius University, Vol. 22 (1), 115–134. ISSN 0868-4952. (ISI Web of Science, Impact Factor 2009: 1.040)

A 5. Kurasova, O.; Molytė, A., 2011. Integration of the self-organizing map and neural gas with multidimensional scaling, *Information Technology and Control*, Vol. 40 (1). ISSN 1392-124X. (ISI Web of Science, Impact Factor 2009: 0.495)

Straipsniai kituose mokslo leidiniuose

B 1. Molytė, A.; Kurasova, O., 2008. Vektorių kvantavimo metodo Neural-Gas tyrimas, 11-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ 2008 metų teminės konferencijos „Informatika“ straipsnių rinkinys, Vilnius, VGTU, 198–205. ISBN 978-9955-28-302-7.

B 2. Kurasova, O.; Molytė, A., 2009. Investigation of Quality of Mapping Vectors Obtained by Quantization Methods, Proceedings of the XIII International Conference on Applied Stochastic Models and Data Analysis (ASMDA-2009), Selected papers (L. Sakalauskas, C. Skiadas, E. K. Zavadskas (eds.)). Vilnius, Technika, 269–273. ISBN 978-9955-28-463-5.

B 3. Kurasova, O.; Molytė, A., 2009. Integrated Visualization of Vector Quantization by Multidimensional Scaling, Abstract Proceedings of Workshop on Computer Graphics, Vision and Mathematics (GraVisMa 2009) (D. Hildenbrand. V. Skala (eds.)), University of West Bohemia, 22. ISBN 978-80-86943-92-3.

Alma MOLYTĖ

VEKTORIŲ KVANTAVIMO METODŲ JUNGIMO
SU DAUGIAMATĖMIS SKALĖMIS ANALIZĖ

Daktaro disertacija

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

Alma MOLYTĖ

INVESTIGATION OF COMBINATIONS OF VECTOR QUANTIZATION METHODS WITH
MULTIDIMENSIONAL SCALING

Doctoral Dissertation

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