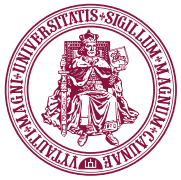


VYTAUTO DIDŽIOJO UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Rasa KARBAUSKAITĖ

# DAUGIAMAČIŲ DUOMENŲ VIZUALIZAVIMO METODŲ, IŠLAIKANČIŲ LOKALIĄ STRUKTŪRĄ, ANALIZĖ

**DAKTARO DISERTACIJA**

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

VILNIUS

VYTAUTO DIDŽIOJO UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

**Rasa KARBAUSKAITĖ**

# DAUGIAMAČIŲ DUOMENŲ VIZUALIZAVIMO METODŲ, IŠLAIKANČIŲ LOKALIĄ STRUKTŪRĄ, ANALIZĖ

Daktaro disertacija

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

Vilnius, 2010

Disertacija rengta 2006-2010 metais Matematikos ir informatikos institute.

**Darbo mokslinis vadovas**

prof. habil. dr. Gintautas Dzemyda (Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09 P).

# Padėka

*Labiausiai dėkoju savo moksliniam vadovui prof. habil. dr. Gintautui Dzemydai už vertingas mokslines konsultacijas, nuoseklų vadovavimą, visokeriopą pagalbą ir kantrybę rengiant šią disertaciją. Labai jam dėkinga už visapusišką supratimą, optimizmo skiepijimą, padrąsinimą ir pasitikėjimą manimi.*

*Be galo dėkinga draugei dr. Olgai Kurasovai už vertingus patarimus, pastabas ir diskusijas rengiant disertaciją, o labiausiai dėkoju jai už draugišką pagalbą, supratimą ir moralinį palaikymą.*

*Ačiū disertacijos recenzentams prof. habil. dr. Mifodijui Sapagovui ir dr. Viktorui Medvedevui, atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų patarimų bei kritinių pastabų.*

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

*Dėkoju Vilniaus pedagoginio universiteto informatikos ir matematinės analizės ir geometrijos katedrų darbuotojams už moralinį palaikymą. Ypač dėkinga doc. dr. Edmundui Mazėčiui už vertingus mokslinius patarimus ir draugišką pagalbą.*

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

*Nuoširdžiai dėkoju savo tėvams už jų moralinį palaikymą ir supratimą.*

*Taip pat dėkoju visiems kitiems, kurie prisidėjo prie šio darbo vertingomis pastabomis ir pasiūlymais.*

*Rasa Karbauskaitė*

# Reziumė

Disertacijos tyrimų sritis susijusi su duomenų analizėje bei vizualizavime sprendžiamu uždaviniu – atvaizduoti duomenis iš labai didelės dimensijos erdvės į mažesnės dimensijos projekcinę erdvę taip, kad kiek galima labiau būtų išlaikyta duomenų struktūra, bei leisti vizualiai pažvelgti į sudėtingas daugiamačių duomenų aibes. Disertacijoje koncentruojamasi į tuos duomenų dimensijos mažinimo metodus, kurie išlaiko lokalią struktūrą bei nagrinėja specifinius duomenis – daugdaros tipo daugiamačius duomenis. Taigi disertacijos tyrimų objektas – daugiamačių duomenų vizualizavimo algoritmai ir metodai, išlaikantys lokalią struktūrą, bei daugiamačių duomenų projekcijų mažesnės dimensijos erdvėje vertinimo kriterijai. Pagrindinis disertacijos tikslas yra išanalizuoti daugiamačių duomenų vizualizavimo algoritmus, išlaikančius lokalią struktūrą, juos modifikuoti bei ištirti nagrinėjamų algoritmų valdymo parametrų svarbą ir pasiūlyti būdus šiems parametrams parinkti, siekiant gauti tikslesnę duomenų projekciją.

Pirmame skyriuje atskleidžiama nagrinėjamos problematikos svarba, įvardinamas tyrimų objektas, aprašomi keliami tikslai bei uždaviniai, mokslo naujumas ir kt. Antras skyrius skirtas daugiamačių duomenų projekcijos metodų analitinei apžvalgai. Ji leido išgryninti kelias aktualias tyrimų kryptis, kurias vienija lokalios struktūros išlaikymo būtinumas. Trečiame skyriuje lokalios struktūros išlaikymo idėjos pritaikomos duomenų aibių papildymui naujais duomenimis. Ketvirtame skyriuje nagrinėjamas metodas, leidžiantis daugiamačius duomenis atvaizduoti mažesnės dimensijos erdvėje taip, kad jų projekcijos nepersidengtų. Čia irgi akivaizdi lokalios struktūros išlaikymo panaudojimo idėja. Penktas ir šeštas skyriai skirti nagrinėti dviem netiesinės daugdaros atpažinimo metodams, kurie daugiamačius duomenis transformuoja į mažesnės dimensijos erdvę, išlaikant kaimyniškumą tarp artimiausių taškų ir atskleidžiant daugiamačių duomenų netiesinę struktūrą. Septintas skyrius skirtas daugdaros topologijos išlaikymo matams tyrinėti. Aštuntame skyriuje pateiktos bendrosios išvados.

Disertaciją sudaro aštuoni skyriai ir literatūros sąrašas. Bendra disertacijos apimtis: 168 puslapiai, 103 paveikslai ir 6 lentelės.

Tyrimų rezultatai publikuoti 6 moksliniuose leidiniuose (keturi iš jų žurnaluose): 2 straipsniai tarptautiniuose periodiniuose leidiniuose, įtrauktuose į Mokslinės informacijos instituto pagrindinį sąrašą (*ISI Web of Science*), 2 straipsniai kituose recenzuojamuose mokslo žurnaluose, 2 straipsniai konferencijų pranešimų rinkiniuose. Tyrimų rezultatai buvo pristatyti ir aptarti 5-iose respublikinėse ir tarptautinėse konferencijose.

# Žymėjimai

## Simboliai

| | |
|---|---|
| $C^i = \{c^i_{jl},\ j,l = \overline{1,k}\}$ | duomenų taško $X_i$ (lokalioji) Gramo matrica |
| $\lvert C^i \rvert$ | Gramo matricos $C^i$ determinantas |
| $d$ | vaizdo erdvės (projekcinės erdvės, į kurią atvaizduojamas $n$-matės erdvės taškas) dimensija (matmenų, matavimų skaičius), $d < n$; $n$-matės erdvės taškų vidinė dimensija; daugdaros dimensija |
| $d(X_i, X_j)$ | atstumas tarp $n$-matės erdvės taškų $X_i$ ir $X_j$ |
| $d(Y_i, Y_j)$ | atstumas tarp $d$-matės erdvės taškų $Y_i$ ir $Y_j$, $d < n$ |
| $D$ | lokaliosios Gramo matricos reguliarizacijos algoritmo R2 valdymo parametras – reguliarizuotos Gramo matricos $C^i$ determinantas |

| | |
|---|---|
| $E_S$ | Sammono projekcijos paklaida |
| $E_p$ | dalelių (taškų) tarpusavio sąveikos potencinė energija. Standumo (*rigidity*) parametras $p \in (-1;+\infty)$ suteikia galimybę kontroliuoti, kaip greitai stūmos jėgos mažės didėjant atstumams tarp dalelių (taškų) |
| $\varepsilon$ | lokaliosios Gramo matricos reguliarizacijos parametras LLE algoritme |
| $f_l$ | matricos *l*-asis tikrinis vektorius |
| $G = (V, E)$ | kaimynystės grafas, $V$ – viršūnių aibė, $E$ – briaunų aibė |
| $G(X_i, X_j)$ | geodezinis atstumas tarp daugdaros taškų $X_i$ ir $X_j$ |
| $k$ | netiesinės daugdaros atpažinimo metodų (LLE, LE, HLLE, LTSA, ISOMAP) valdymo parametras – taško artimiausių kaimynų skaičius |
| $k_{geod}$ | geodezinių atstumų skaičiavimo algoritmo parametras – artimiausių kaimynų skaičius, reikalingas kaimynystės grafui sudaryti |
| $k^*$ | LE algoritmo modifikacijos valdymo parametras – maksimalus artimiausių kaimynų skaičius |
| $k_1$ | KM mato valdymo parametras – mažesnysis artimiausių kaimynų skaičius |
| $k_2$ | KM mato valdymo parametras – didesnysis artimiausių kaimynų skaičius ( $k_1 < k_2$ ) |
| $K$ | MRRE mato valdymo parametras – artimiausių kaimynų skaičius kiekvienam duomenų taškui |
| $L = \{l_{ij}, i, j = \overline{1,m}\}$ | grafo $G$ Laplaso matrica |
| $\lambda_l$ | matricos *l*-ąjį tikrinį vektorių atitinkanti tikrinė reikšmė |
| $m$ | analizuojamų objektų (taškų) skaičius |

| | |
|---|---|
| $m'$ | iteracijos numeris |
| $M$ | $d$-matė daugdara, įdėta erdvėje $R^n$ |
| $\widetilde{M}$ | išretinta (*sparse*) matrica, kurios $d$ tikriniai vektoriai suformuoja taškų $Y_i$ koordinates LLE algoritme |
| $n$ | taško $X_i$ koordinačių skaičius; erdvės, kuriai priklauso taškas $X_i$, dimensija; objektą apibūdinančių parametrų skaičius |
| $r$ | lokaliosios Gramo matricos $C^i$ rangas, t. y. matricos $C^i$ tikrinių reikšmių nelygių nuliui skaičius |
| $\widetilde{r}$ | RPM algoritmo pradinis mokymo greitis |
| $R^n$ | $n$-matė erdvė |
| $\rho_{Sp}$ | Spirmano koeficientas |
| $t$ | lokaliosios Gramo matricos reguliarizacijos algoritmo R1 valdymo parametras |
| $T$ | LE algoritmo valdymo parametras – šiluminio branduolio parametras, naudojamas Gauso branduolio funkcijoje svoriams $w_{ij}$ apskaičiuoti |
| $V_i$ | $i$-oji grafo $G$ viršūnė, atitinkanti duomenų tašką $X_i$ |
| $w^*$ | LE algoritmo modifikacijos valdymo parametras – svorinis slenkstis, virš kurio svoriai laikomi tinkamais, t. y. tuos svorius turintys taškai turi įtakos nagrinėjamam taškui ir yra laikomi artimiausiais kaimynais |
| $w_{ij}$ | kaimynystės grafo $G$ briaunos, jungiančios viršūnes $V_i$ ir $V_j$, svoris LE algoritme; svoris tarp $i$-ojo taško ir jo $j$-ojo kaimyno LLE algoritme; svoris daugiamatėse skalėse |

| | |
|---|---|
| $W$ | kaimynystės grafo $G$ briaunų svorių matrica LE algoritme; svorių matrica LLE algoritme |
| $x_{ij}$ | duomenų taško $X_i$ $j$-oji koordinatė; objektą $X_i$ apibūdinantis $j$-asis parametras |
| $x_j$ | objektą apibūdinantis $j$-asis parametras |
| $X$ | analizuojamų duomenų matrica (aibė) |
| $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ | $i$-asis duomenų taškas (vektorius), $X_i \in R^n$ |
| $X_i^T$ | transponuotasis vektorius $X_i$ |
| $X_{ij} = (x_{i1}^j, x_{i2}^j, ..., x_{in}^j)$ | duomenų taško $X_i$ $j$-asis kaimynas |
| $y_j$ | vaizdo taško $j$-oji koordinatė |
| $Y$ | vaizdo taškų matrica (aibė) |
| $Y_i = (y_{i1}, y_{i2}, ..., y_{id})$ | vaizdo taško koordinatės, taško $X_i$ transformacija mažesnės dimensijos (vaizdo) erdvėje $R^d$, $d < n$ |
| $Y_{ij} = (y_{i1}^j, y_{i2}^j, ..., y_{id}^j)$ | vaizdo taško $Y_i$ $j$-asis kaimynas; taško $X_{ij}$ transformacija mažesnės dimensijos (vaizdo) erdvėje $R^d$, $d < n$ |
| $\overline{Y} = [0, w] \times [0, h] \subset R^2$ | stačiakampis, gautas išardžius torą, kuriame RPM metodu pavaizduotos $n$-matės erdvės taškų $X_i$ transformacijos $Y_i$; $w$ – stačiakampio plotis, $h$ – stačiakampio aukštis |
| $\|X_i - X_j\|$ | Euklido atstumas tarp $n$-matės erdvės taškų $X_i$ ir $X_j$ |

## Santrumpos

| | |
|---|---|
| HLLE | Hesės matricos tikriniai žemėlapiai (*Hessian eigenmaps, hessian-based locally linear embedding, Hessian LLE*) |
| ISOMAP | izometrinis požymių vaizdavimas (*isometric feature mapping*) |
| KM | Konigo matas – topologijos išlaikymo matas |
| LE | Laplaso matricos tikriniai žemėlapiai (*Laplacian eigenmaps*) |
| LLE | lokaliai tiesinis vaizdavimas (*locally linear embedding*) |
| LTSA | lokaliųjų liečiamųjų erdvių rikiavimas (*local tangent space alignment*) |
| MDS | daugiamatės skalės (*multidimensional scaling*) |
| MRRE | kaimynystės klaidos – topologijos išlaikymo matas |
| MST | minimalaus jungimo medis (*minimal spaning tree*) |
| PCA | pagrindinių komponenčių analizė (*principal component analysis*) |
| R1 | lokaliosios Gramo matricos reguliarizacijos algoritmas, pasiūlytas Roweis ir Saul (2000) |
| R2 | lokaliosios Gramo matricos reguliarizacijos algoritmas, pasiūlytas šiame darbe |
| RPM | santykinės perspektyvos metodas (*relational perspective map*) |

# Turinys

# 1

# Įvadas

## 1.1. Tyrimų sritis

Sparčiai vystantis technologijoms, tobulėjant kompiuteriams ir programinei įrangai, informacija gaunama labai greitai, o kaupiamų duomenų apimtys ypač sparčiai didėja. Todėl susiduriame su būtinybe analizuoti bei vertinti didžiulius duomenų kiekius. Lieka esminė problema – kaip tuos duomenis suvokti ir interpretuoti, kaip iš turimų duomenų gauti reikiamą informaciją, atskiriant menkaverčius faktus. Paprastai atsiranda būtinybė nustatyti ir giliau pažinti tokių duomenų struktūrą: susidariusias grupes (klasterius), itin išsiskiriančius objektus (taškus atsiskyrėlius), objektų tarpusavio panašumą ar skirtingumą. Suvokti duomenis nelengva, ypač kai jie yra *daugiamačiai*, t. y. kai nurodo sudėtingą reiškinį ar objektą, apibūdinamą daugeliu parametrų (požymių, savybių), kurie gali būti ne tik skaitiniai, bet ir loginiai, tekstiniai ir kt. Šio darbo tyrimų sritis ir yra daugiamačių duomenų analizė bei tų duomenų suvokimo gerinimo būdai.

## 1.2. Darbo aktualumas

Realiame gyvenime sprendžiami uždaviniai dažnai susiję su daugiamačiais duomenimis. Nėra tokios žmonių veiklos srities, kur nebūtų kaupiami ir analizuojami tokie duomenys. Su jais susiduriame medicinoje, ekonomikoje, ekologijoje, sociologijoje, technikoje ir daugelyje kitų sričių.

Realus daugiamačių duomenų pavyzdys iš medicinos srities – krūties vėžio duomenys. Buvo matuoti devyni fiziologiniai parametrai 683 moterims, kurioms diagnozuotas piktybinis ar nepiktybinis krūties navikas. Gauti 683 taškai, išsibarstę devynmatėje erdvėje. Kitas tipiškas daugiamačių duomenų pavyzdys susijęs su vaizdų apdorojimu (*image processing*). Dažnai duomenis sudaro to paties objekto paveikslėliai, gauti palaipsniui pasukant objektą tam tikru kampu arba nufotografuojant jį skirtingais momentais. Kiekvienas paveikslėlis yra skaitmenizuojamas, t. y. duomenų taško koordinatės yra sudarytos iš paveikslėlio taškų spalvinių savybių, ir todėl šio taško koordinačių skaičius yra labai didelis.

Kuo didesnės apimties duomenys (didesnis objektų skaičius arba didesnis objektą charakterizuojamų parametrų skaičius), tuo sunkiau iš duomenų lentelėje pateiktų daugybės skaičių suvokti objektų visumos ypatybes. Daugiamačiai duomenys gali būti analizuojami įvairiais duomenų analizės metodais: klasifikavimo, klasterizavimo, statistinės analizės, vizualizavimo ir kt. Svarbu duomenis pateikti tokia forma, kad tyrėjui būtų lengviau juos suprasti, t. y. būtų galima nustatyti duomenų struktūrą, tarpusavio ryšius, susidariusias grupes ir pan. Vienas iš galimų būdų yra vizualizavimas. *Vizualizavimas* – tai grafinis informacijos pateikimas. Vizualią informaciją žmogus pajėgus suvokti daug greičiau negu tekstinę.

Siekiant palengvinti duomenų interpretavimą, efektyviai pateikti ir įvertinti duomenų gavybos rezultatus, daugiamačių duomenų vizualizavimo metodai vystomi gana intensyviai. Plačiai naudojama vizualizavimo metodų grupė – projekcijos metodai, dar vadinami dimensijos mažinimo metodais. Jais transformavus daugiamačius duomenis į dvimatę ar trimatę vaizdo erdvę ir juos vizualizavus, daug paprasčiau suvokti duomenų struktūrą ir sąryšius tarp jų.

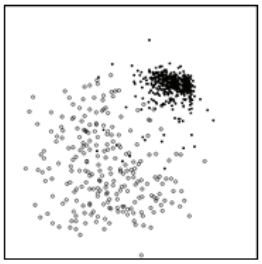
Pavyzdžiui, žvelgiant į 1.1 ir 1.2 paveiksluose pateiktas duomenų lenteles, nieko negalima pasakyti apie analizuojamus duomenis, nes taškų skaičius (1.1 pav.) arba taškų koordinačių skaičius (1.2 pav.) labai didelis. Tačiau vizualizavus daugiamačius duomenis plokštumoje, jau galime daryti tam tikras išvadas apie analizuojamų duomenų aibių struktūrą. Didžioji dalis taškų 1.1 paveiksle, atitinkančių nepiktybinio naviko duomenis (juodi taškai), susikoncentruoja vienoje srityje, o kiti taškai, atitinkantys piktybinio naviko duomenis, pasiskirsto plačiai. Matome aiškiai išskiriančias taškų sankaupos sritis. 1.2 paveiksle pavaizduota, kaip plokštumoje išsidėstę daugiamatės erdvės

taškai, atitinkantys pasuktos taupyklės paveikslėlius. Didesni skrituliukai atitinka šalia jų pateiktus paveikslėlius. Taupyklė buvo laipsniškai sukama aplink 360° kampu, todėl plokštumoje taškai išsidėstė ratu. Turėdami naują paveikslėlį, radę jo vietą plokštumoje tarp jau esančių paveikslėlių su žinomais figūros posūkio kampais, galime įvertinti ir naujame paveiksle esančios figūros posūkio kampą.

**Analizuojami duomenys (683 pacientės)**

| | 1 | 2 | 3 | … | 8 | 9 |
|---|---|---|---|---|---|---|
| **1** | 4,05 | 0,05 | 0,48 | … | 0,83 | 0,93 |
| **2** | 4,86 | 3,90 | 3,14 | … | 1,34 | 0,81 |
| **3** | 2,48 | 0,09 | 0,11 | … | 0,87 | 0,73 |
| **4** | 5,33 | 7,14 | 7,96 | … | 6,68 | 0,79 |
| **5** | 3,13 | 0,09 | 0,74 | … | 0,80 | 0,36 |
| **…** | … | … | … | … | … | … |
| **680** | 4,40 | 9,52 | 9,20 | … | 3,89 | 0,48 |
| **681** | 4,36 | 9,41 | 9,54 | … | 9,51 | 1,80 |
| **682** | 3,66 | 7,68 | 5,79 | … | 5,93 | 0,17 |
| **683** | 3,91 | 7,88 | 8,00 | … | 3,32 | 1,00 |

**Duomenų projekcija**

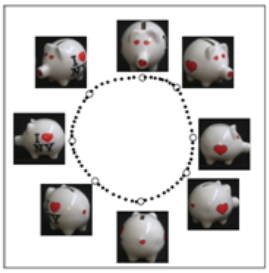


**1.1 pav.** Krūties vėžio duomenų vizualizavimas

**Analizuojami duomenys (72 paveikslėliai)**

| | 1 | 2 | … | 25000 | … | 35000 | … | 49152 |
|---|---|---|---|---|---|---|---|---|
| **1** | 53 | 26 | … | 195 | … | 81 | … | 26 |
| **2** | 53 | 25 | … | 191 | … | 132 | … | 25 |
| **3** | 53 | 25 | … | 188 | … | 114 | … | 25 |
| **4** | 53 | 25 | … | 188 | … | 106 | … | 25 |
| **5** | 53 | 25 | … | 187 | … | 86 | … | 25 |
| **…** | … | … | … | … | … | … | … | … |
| **69** | 53 | 25 | … | 187 | … | 22 | … | 25 |
| **70** | 53 | 25 | … | 188 | … | 26 | … | 25 |
| **71** | 53 | 25 | … | 194 | … | 27 | … | 25 |
| **72** | 53 | 25 | … | 193 | … | 31 | … | 25 |

**Duomenų projekcija**



**1.2 pav.** Taupyklės paveikslėlių duomenų vizualizavimas

Duomenis transformuojant į mažesnės dimensijos erdvę, neišvengiami duomenų projekcijų iškraipymai. Todėl gautų projekcijų kokybės įvertinimas išlieka aktualia problema.

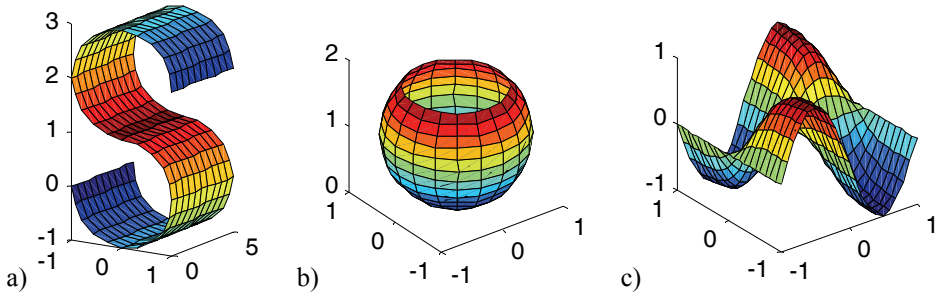
Dažnai tenka dirbti su duomenų aibėmis, kurios pastoviai papildomos naujais duomenimis. Labai svarbu greitai atvaizduoti naujus duomenų taškus, neprarandant didelio tikslumo. Todėl naujų taškų atvaizdavimas, jų įterpimas tarp anksčiau atvaizduotų taškų – viena iš disertacijoje nagrinėjamų problemų.

Daugiamačių duomenų projekcijos metodai susiduria su dviem pagrindinėmis problemomis. Pirma, reikia rasti daugiamačių duomenų projekcijas mažesnės dimensijos erdvėje (dvimatėje ar trimatėje), siekiant kuo tiksliau išlaikyti analizuojamos aibės objektų artimumus – panašumus ar skirtingumus. Antra, daugiamačius duomenis atvaizduoti mažesnės dimensijos

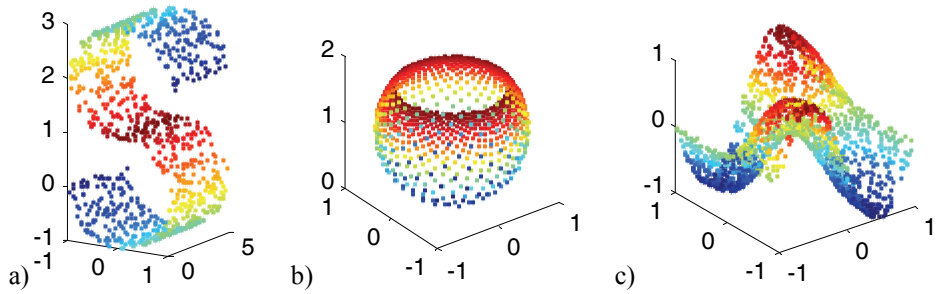
erdvėje taip, kad jų projekcijos nepersidengtų. Ši problema taip pat yra viena iš disertacijoje sprendžiamų problemų.

Dažnai praktiniuose uždaviniuose yra sukaupiami daugiamačiai duomenys, kuriuos atitinkantys taškai nagrinėjami didelės dimensijos erdvėje, o iš tikrųjų jie yra kokios nors mažesnės dimensijos daugdaros arba tai daugdarai artimi taškai. *Daugdara* – tai aibė, kurios kiekvieno taško aplinkoje įvesta koordinačių sistema. Nors daugdara apibendrina bet kokios dimensijos erdvėje kreivės ir paviršiaus sąvokas, bet lokaliai daugdara nesiskiria nuo Euklido erdvės: ji sudaryta iš suklijuotų Euklido erdvės gabalų (Adler and Taylor 2007).

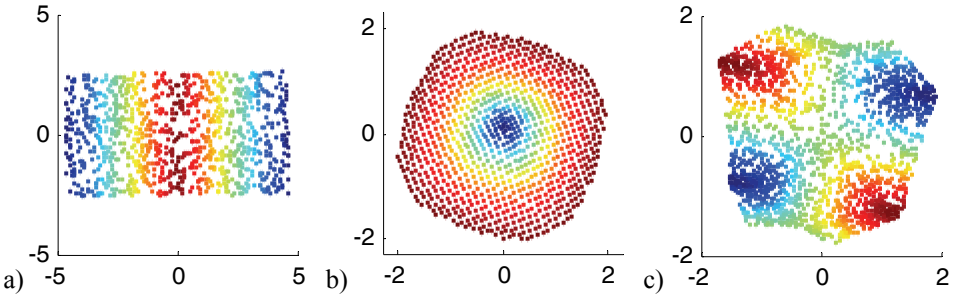
Netiesinių dvimačių daugdarų pavyzdžiai pateikti 1.3 paveiksle. 1.4 paveiksle pavaizduoti trimatės erdvės taškai, išsidėstę ant šių daugdarų. Kadangi disertacijoje tiriami netiesinės daugdaros atpažinimo metodai yra taikomi tik daugiamatės erdvės taškams, priklausantiems mažesnės dimensijos daugdarai, tai šiuo atveju analizei būtų pateikiami duomenų rinkiniai pavaizduoti 1.4 paveiksle, o tikslas – transformuoti šiuos duomenis į *dvimatę* erdvę, t. y. į plokštumą (nes taškai priklauso *dvimatėms* daugdaroms) (1.5 pav.).




**1.3 pav.** Netiesinės dvimatės daugdaros: a) S-formos daugdara, b) sferos nuopjova, c) „Dvynės viršukalnės“




**1.4 pav.** Trimatės erdvės taškai, išsidėstę ant netiesinių dvimačių daugdarų: a) S-formos daugdaros, b) sferos nuopjovos, c) daugdaros „Dvynės viršukalnės“




**1.5 pav.** Trimatės erdvės taškų, išsidėsčiusių ant netiesinių dvimačių daugdarų: a) S-formos daugdaros, b) sferos nuopjovos, c) daugdaros „Dvynės viršukalnės“ projekcijos plokštumoje

Taigi viena iš pagrindinių disertacijos problemų – atrasti mažesnio matavimo netiesinę daugdarą didelio matavimo erdvėje ir tada transformuoti duomenų taškus, esančius ant arba arti tos daugdaros, į mažesnio matavimo erdvę.

Svarbus su daugdara susijęs dalykas yra jos *topologija*, t. y. daugdaros visų atvirųjų poaibių rinkinys. Topologijos išlaikymui įvertinti sukurta daugybė įvairių matų: (Siegel and Castellan 1988; Goodhill and Sejnowski 1996; Konig 2000; Tenenbaum *et al*. 2000; Venna and Kaski 2001; Lee and Verleysen 2007). Skirtingiems uždaviniams turi būti parenkami skirtingi topologijos išlaikymo matai (Goodhill and Sejnowski 1996). Svarbi disertacijoje sprendžiama problema – rasti ir ištirti tuos matus, kurie būtų tinkamiausi analizuoti daugdaros topologijos išlaikymą po jos transformavimo į mažesnės dimensijos erdvę.

## 1.3. Tyrimo objektas

Disertacijos tyrimų objektas – daugiamačių duomenų vizualizavimo algoritmai ir metodai, išlaikantys lokalią struktūrą, bei daugiamačių duomenų projekcijų mažesnės dimensijos erdvėje vertinimo kriterijai. Čia lokalios struktūros išlaikymu vadiname atstumų tarp artimiausių taškų santykių išlaikymą po analizuojamos daugiamačių duomenų aibės transformavimo iš didesnio matavimo erdvės į mažesnio matavimo erdvę. Su šiuo objektu betarpiškai susiję dalykai: dimensijos mažinimo (projekcijos) algoritmai, netiesinės daugdaros atpažinimo metodai, daugiamačių duomenų projekcijos į mažesnio matavimo erdvę paklaidos, naujų daugiamačių taškų atvaizdavimas.

## 1.4. Darbo tikslas ir uždaviniai

Pagrindinis disertacijos tikslas yra išanalizuoti daugiamačių duomenų vizualizavimo algoritmus, išlaikančius lokalią struktūrą, juos modifikuoti bei ištirti nagrinėjamų algoritmų valdymo parametrų svarbą ir pasiūlyti būdus šiems parametrams parinkti, siekiant gauti tikslesnę duomenų projekciją.

Norint pasiekti šį tikslą, reikėjo išspręsti tokius uždavinius:

1. išanalizuoti esamus daugiamačių duomenų vizualizavimo metodus ir apsibrėžti tiriamų vizualizavimo metodų, išlaikančių lokalią struktūrą, grupę;
2. išanalizuoti pasirinktus metodus ir jais vizualizuoti daugiamačių duomenų aibes;
3. tiriamais metodais gautus vizualizavimo rezultatus palyginti su rezultatais, kurie gauti metodais, išlaikančiais ne tik lokalią struktūrą;
4. įvertinti įvairiais vizualizavimo metodais gautų duomenų projekcijų kokybę (naudojant ekspertinį ir kiekybinius skaitinius matus);
5. sukurti daugiamačių duomenų vizualizavimo algoritmų, išlaikančių lokalią struktūrą, modifikacijas, siekiant gauti tikslesnę analizuojamų duomenų projekciją;
6. įvertinti modifikuotų algoritmų rezultatus lyginant su rezultatais, gautais originaliais algoritmais;
7. ištirti nagrinėjamų algoritmų valdymo parametrų svarbą ir pasiūlyti būdus šiems parametrams parinkti, siekiant gauti tikslesnę analizuojamų duomenų projekciją.

## 1.5. Mokslinis naujumas

Eksperimentiškai ištyrus trianguliacijos metodo realizacijas, naudojančias antrojo arčiausiojo kaimyno ir atramos taško metodus atraminiams taškams parinkti, nustatyta, jog abiem atvejais projekcijos paklaida labai priklauso nuo taškų atvaizdavimo sekos, o tai įrodo, kad naudoti vien tik trianguliacijos metodą duomenims vizualizuoti nėra pakankama. Pasiūlyta Sammono ir trianguliacijos metodų junginio nauja realizacija naujiems taškams atvaizduoti.

Eksperimentiškai ištirta santykinės perspektyvos metodo (RPM) priklausomybė nuo parametrų – stačiakampio, kuriame vizualizuojami duomenys, pločio ir aukščio, bei pradinio mokymo greičio. Pasiūlyta nauja RPM metodo realizacija, leidžianti šios priklausomybės beveik išvengti.

Eksperimentiškai ištirta lokaliai tiesinio vaizdavimo (LLE) metodo priklausomybė nuo parametrų – artimiausių kaimynų skaičiaus kiekvienam duomenų taškui ir lokaliosios Gramo matricos reguliarizacijos parametro. Siekiant gauti kuo tikslesnes duomenų projekcijas, pasiūlytas naujas būdas artimiausių kaimynų skaičiui parinkti LLE algoritme. Taip pat sukurtas naujas algoritmas lokaliajai Gramo matricai reguliarizuoti.

Eksperimentiškai ištirta parametrų (artimiausių kaimynų skaičiaus kiekvienam duomenų taškui ir šiluminio branduolio parametro, naudojamo Gauso branduolio funkcijoje svoriams apskaičiuoti) svarba Laplaso matricos tikrinių žemėlapių (LE) algoritme. Pasiūlyta LE algoritmo modifikacija, kurioje yra tik vienas svarbus valdymo parametras – maksimalus artimiausių kaimynų skaičius.

Panaudojant du kriterijus – topologijos išlaikymo kokybę ir skaičiavimo sąnaudas – ištirti ir palyginti trys topologijos išlaikymo matai (Spirmano koeficientas, Konigo matas (KM) ir kaimynystės klaidos (MRRE)), kurie tinkami analizuoti daugdaros topologijos išlaikymą po jos transformavimo į mažesnio matavimo erdvę.

## 1.6. Ginamieji teiginiai

1. Taupant skaičiavimų laiką ir mažai teprarandant tikslumą, naujiems daugiamatės erdvės taškams atvaizduoti, pradinius taškus vizualizavus MDS tipo metodu, gali būti naudojamas trianguliacijos metodas.
2. Santykinės perspektyvos, lokaliai tiesinio vaizdavimo ir Laplaso matricos tikrinių žemėlapių metodų valdymo parametrai labai įtakoja gautų projekcijų kokybę, tačiau egzistuoja strategijos, leidžiančios mažinti parametrų skaičių ar reglamentuoti jų reikšmių parinkimą.
3. Konigo matas (KM) ir kaimynystės klaidos (MRRE) visada gerai nusako daugdaros topologijos išlaikymą po jos transformacijos į mažesnio matavimo erdvę, o Spirmano koeficientas sėkmingai gali būti taikomas tik paprastesnės struktūros daugdarų topologijos išlaikymui įvertinti.

## 1.7. Praktinė vertė

Tyrimų rezultatai atskleidė daugdaros tipo daugiamačių duomenų analizės galimybes. Parodyta, jog netiesinės daugdaros atpažinimo metodai gali būti plačiai naudojami įvairiose srityse, tarp jų ir medicinoje.

Tyrimai atlikti pagal Lietuvos valstybinio mokslo ir studijų fondo aukštųjų technologijų plėtros programos projektą „Informacinės klinikinių sprendimų palaikymo ir gyventojų sveikatinimo priemonės e. Sveikatos sistemai (Info Sveikata)“; Registracijos Nr.: B-07019; Vykdymo laikas: 2007 m. 09 mėn. – 2009 m. 12 mėn.

## 1.8. Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 6 moksliniuose leidiniuose (keturi iš jų žurnaluose): 2 straipsniai tarptautiniuose periodiniuose leidiniuose, įtrauktuose į Mokslinės informacijos instituto pagrindinį sąrašą (*ISI Web of Science*), 2 straipsniai kituose recenzuojamuose mokslo žurnaluose, 2 straipsniai konferencijų pranešimų rinkiniuose.

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

1. Lietuvos jaunųjų mokslininkų konferencija „Operacijų tyrimas ir taikymai“ (LOTD – 2006), Vilnius, 2006 m. gegužės 26 d.
2. Lietuvos jaunųjų mokslininkų konferencija „Operacijų tyrimas ir taikymai“ (LOTD – 2007), Vilnius, 2007 m. gegužės 18 d.
3. Informatics Summer School „Modern Data Mining Technologies“, Druskininkai, 2007 m. rugsėjo 9–15 d.
4. The 20th International Conference, EURO Mini Conference „Continuous Optimization and Knowledge-Based Technologies“ (EurOPT-2008), Neringa, 2008 m. gegužės 20–23 d.
5. The 13th International Conference „Applied Stochastic Models and Data Analysis“ (ASMDA-2009), Vilnius, 2009 m. birželio 30 – liepos 3 d.

## 1.9. Disertacijos struktūra

Disertaciją sudaro aštuoni skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Daugiamačių duomenų projekcijos metodų analizė, Trianguliacijos ir Sammono metodų bei jų jungimo tyrimas, Santykinės perspektyvos metodo realizacijų tyrimas, Lokaliai tiesinio vaizdavimo metodo parametrų parinkimo strategijos, Laplaso matricos tikrinių žemėlapių metodo realizacijų tyrimas, Topologijos išlaikymo matai daugdaros tipo daugiamačių duomenų vizualizavime, Bendrosios išvados. Disertacijos apimtis 168 puslapiai, 103 paveikslai ir 6 lentelės.

# 2

# Daugiamačių duomenų projekcijos metodų analizė

Šiame skyriuje yra pateikta ne tik išsami duomenų dimensijos mažinimo (projekcijos) metodų apžvalga, bet ir atliktų tyrimų, analizuojant tuos metodus, rezultatai.

Daugiamačių duomenų vizualizavimo metodai padeda analizuoti ir atvaizduoti žmogui suprantamesne forma sudėtingais ir dažnai nežinomais tarpusavio ryšiais susietus daugiamačius duomenis. Pagrindinis daugiamačių duomenų vizualizavimo tikslas – tai duomenų pavaizdavimas mažesnės dimensijos erdvėje, išsaugant jų tarpusavio panašumo struktūras. Daugiamačių duomenų vizualizavimas suteikia galimybę tyrinėtojui pačiam stebėti tų duomenų grupavimosi tendencijas, įvertinti atskirų daugiamatės erdvės taškų tarpusavio artimumą, racionaliai priimti sprendimus.

Kalbėdami apie daugiamačius duomenis, susiduriame su šiomis sąvokomis: *objektas* ir *parametras*. Sąvoka objektas apima įvairius dalykus: žmones, įrenginius, augalus, gyvūnus, gamtos reiškinius ir kt. Objektai, sudarantys konkrečią analizuojamų objektų aibę, yra apibūdinami bendrais parametrais (požymiais, savybėmis). Objektų tokioje aibėje skaičius $m$ yra baigtinis, tačiau bendru atveju gali būti ir labai didelis. Tam tikras visų parametrų reikšmių rinkinys nusako vieną konkretų analizuojamos aibės objektą

$X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $i = \overline{1,m}$, čia $n$ yra parametrų skaičius, dar vadinamas duomenų dimensija, $i$ yra objekto eilės numeris. Kai objektą $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ apibūdina daugiau nei vienas parametras, duomenys (objektai) yra *daugiamačiai*. Objektai $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ dar vadinami vektoriais ar taškais, o parametrai $x_1, x_2, ..., x_n$ – atitinkamai komponentėmis ar koordinatėmis. Galima suformuoti analizuojamų duomenų aibę $X$, sudarytą iš taškų (vektorių) $X_i \in R^n$, ($X_i = (x_{i1}, x_{i2}, ..., x_{in}) = \{x_{ij}\}$, $i = \overline{1,m}$, $j = \overline{1,n}$). Kiekvienas parametras turi tam tikras skaitines reikšmes, iš kurių įmanoma sudaryti skaičių lentelę. Kuo duomenų dimensija didesnė, tuo sunkiau iš lentelės išgauti informacijos apie santykius tarp atskirų objektų. Tokioje situacijoje gelbsti vizualus duomenų pateikimas, kuris leidžia tyrinėtojui „pajusti“ daugiamačius objektus atitinkančių taškų tarpusavio atstumus, o tai palengvina duomenų visumos pažinimą. Kai dimensija daugiau nei trys, daugiamačių duomenų tiesioginis vaizdavimas yra sudėtingas, todėl tokių duomenų pateikimui naudojami įvairūs vizualizavimo metodai. Tačiau, vizualizuojant duomenis (mažinant duomenų dimensiją), dažnai neįmanoma pateikti visų duomenų parametrų neprarandant jokios informacijos, todėl būtina ieškoti būdų, leidžiančių tuos praradimus minimizuoti.

Dėl šios priežasties sukurta ir toliau sėkmingai vystoma daug skirtingo pobūdžio daugiamačių duomenų vizualizavimo metodų. Siekiant kuo mažiau prarasti informacijos vizualizavimo metu, konkrečiam uždaviniui (konkrečiai duomenų aibei vizualizuoti) reikia parinkti tinkamiausią metodą. Vieningos vizualizavimo metodų klasifikacijos nėra, kadangi metodai gali būti klasifikuojami pagal įvairius kriterijus. Įvairios vizualizavimo metodų klasifikacijos ir jų apžvalgos pateiktos šiuose darbuose: (Dzemyda *et al*. 2008; Medvedev 2007; Lee and Verleysen 2007; Kurasova 2005; Keim and Ward 2003; Hoffman and Grinstein 2002; Šaltenis and Aušraite 2002; Grinstein and Ward 2002; Grinstein *et al.* 2001; Sachinopoulou 2001; Wegenkittl *et al.* 1997; Wong and Bergeron 1997; Keim and Kriegel 1996) ir kt.

Galima išskirti dvi pagrindines vizualizavimo metodų grupes:

1. *tiesioginio vizualizavimo metodai*, kuriais kiekvienas daugiamačio objekto parametras yra pateikiamas tam tikra vizualia forma;
2. *projekcijos*, dar vadinamieji *dimensijos mažinimo* (*dimensional reduction techniques*) metodai, leidžiantys daugiamačius duomenų objektus atitinkančius taškus pateikti mažesnės dimensijos erdvėje.

Projekcijos metodai lyginant su tiesioginio vizualizavimo metodais yra populiaresni, nes jų rezultatai lengviau suvokiami ir interpretuojami. Todėl šioje disertacijoje bus nagrinėjami tik projekcijos metodai.

Lietuvoje daugiamačių duomenų vizualizavimu domimasi jau seniai. Šioje srityje dirba nemažai mokslininkų. Pirmasis pradėjo nagrinėti daugiamates skales prof. V. Šaltenis (1975 m.). Tuo pat metu Sammono projekcijos tyrimus sėkmingai vykdė ir A. M. Montvilas. Daugiamačių duomenų vizualizavimo metodus daug metų jau tiria ir tobulina prof. A. Žilinskas, prof. G. Dzemyda bei jų auklėtiniai.

Prof. A. Žilinskas ir J. Žilinskas pastaruoju metu tyrinėja daugiamates skales su miesto kvartalų metrika (*city–block*) ir ieško daugiamačių skalių paklaidos (įtempimo, tikslo) funkcijos globalaus minimumo (Žilinskas and Žilinskas 2007; Žilinskas and Žilinskas 2009).

Nuo 2004 metų sėkmingai apginta eilė disertacijų, susijusių su daugiamačių duomenų vizualizavimo problemomis: (Podlipskytė 2004; Kurasova 2005; Medvedev 2007; Bernatavičienė 2008; Ivanikovas 2009). Pirmajai vadovavo prof. A. Žilinkas, likusioms keturioms – prof. G. Dzemyda.

Disertacijoje (Podlipskytė 2004) eksperimentiškai palyginti į populiarius programinius paketus įtraukti daugiamačių skalių algoritmai, parodyti šių algoritmų trūkumai didelių daugiamačių duomenų masyvų (pvz. biomedicininių) vizualizacijos požiūriu. Sukurtas efektyvus algoritmas, pagrįstas daugiamačių skalių metodologija ir orientuotas į biomedicininių duomenų vizualizacijos uždavinius. Sukurtasis algoritmas naudotas duomenų, surinktų apie pacientų miego sutrikimus, gyvenimo kokybę, vizualizacijai.

Disertacijoje (Kurasova 2005) tirta daugiamačių duomenų vizuali analizė taikant saviorganizuojančius neuroninius tinklus (SOM). Pagrindinis disertacijos tikslas buvo sukurti algoritmus, kuriuose saviorganizuojantis neuroninis tinklas taip jungiamas su daugiamatėmis skalėmis, kad būtų kiek galima tiksliau atskleista duomenų, sudarančių tam tikras grupes, struktūra. Sukurtas ir ištirtas naujas integruotas SOM tinklo ir Sammono algoritmo junginys, atsižvelgiantis į SOM tinklo mokymosi eigą ir leidžiantis gauti tikslesnę daugiamatės erdvės taškų projekciją plokštumoje. Pasiūlytas ir realizuotas lygiagretusis integruoto junginio algoritmas.

Disertacijoje (Bernatavičienė 2008) tyrimų sritis yra žinių gavybos iš daugiamačių duomenų procesas ir tiriamų duomenų suvokimo gerinimo būdai. Pagrindinis disertacijos tikslas buvo sukurti ir ištirti žinių gavybos vizualiais metodais metodologiją, kuri leistų padidinti duomenų analizės efektyvumą. Detaliai ištirtas santykinių daugiamačių skalių metodas.

Disertacijose (Medvedev 2007; Ivanikovas 2009) nagrinėjami dirbtiniais neuroniniais tinklais grindžiami algoritmai daugiamačiams duomenims vizualizuoti. Pagrindinis šių disertacijų tikslas buvo sukurti ir tobulinti metodus, kuriuos taikant būtų efektyviai minimizuojamos daugiamačių duomenų projekcijos paklaidos naudojantis dirbtiniais neuroniniais tinklais bei projekcijos algoritmais, o taip pat pagreitinamas dirbtinio neuroninio tinklo mokymas.

Disertacijose detaliai ištirtas SAMANN algoritmas bei pasiūlytos jo modifikacijos.

Šioje disertacijoje nagrinėjami tokie daugiamačių duomenų vizualizavimo metodai, kurie nebuvo analizuoti aukščiau minėtose disertacijose, t. y. atlikta daugiamačių duomenų vizualizavimo metodų, išlaikančių lokalią struktūrą, analizė. Detaliai ištirtas trianguliacijos metodas, santykinės perspektyvos metodas bei pastaruoju metu labai populiarūs netiesinės daugdaros atpažinimo metodai, tokie kaip lokaliai tiesinis vaizdavimas ir Laplaso matricos tikrinių žemėlapių metodas.

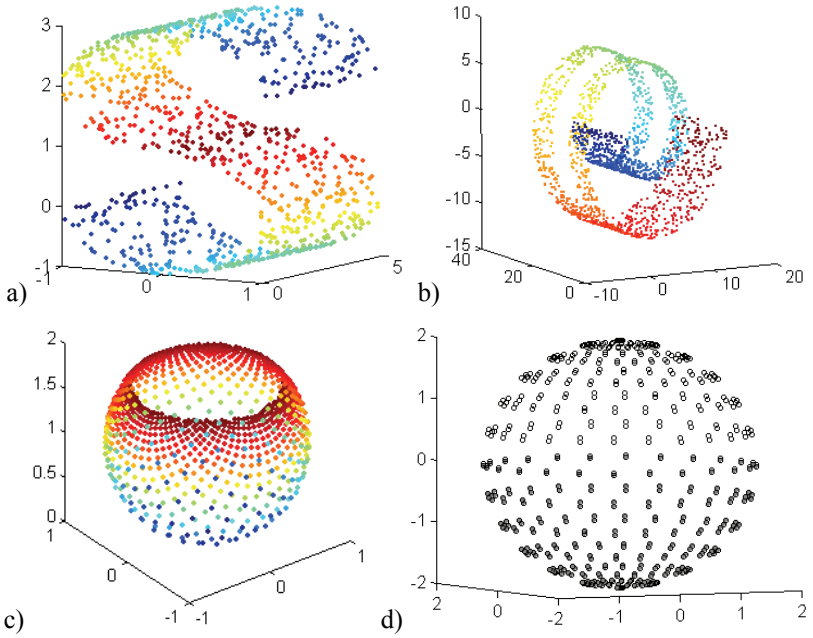
## 2.1. Analizuojami duomenys

Skyriuje daugiamačių duomenų vizualizavimui naudojamos šios duomenų aibės:

- 1000 trimatės erdvės taškų, išsidėsčiusių ant netiesinės dvimatės *S-formos daugdaros* (2.1a pav.).
- 2000 trimatės erdvės taškų, esančių ant netiesinės dvimatės *spiralinio cilindro, turinčio centre stačiakampę išpjovą, daugdaros*. Spiralinis cilindras (*„swiss roll“*) – tai cilindrinis paviršius, kurio vedamoji yra spiralė (2.1b pav.).
- 1000 trimatės erdvės taškų, išsidėsčiusių ant netiesinės dvimatės daugdaros, vadinamos *sferos nuopjova* (*„punctured sphere“*) (2.1c pav.).
- *Sferos* taškų aibė (2.1d pav.) – tai aibė 576 taškų, kurių koordinatės $(x_1, x_2, x_3)$ yra apskaičiuojamos pagal šias parametrines lygtis:

$$\begin{cases} x_1 = 2\cos\alpha\cos\beta, \\ x_2 = 2\sin\alpha\cos\beta, \\ x_3 = 2\sin\beta \end{cases}$$

vienodais intervalais keičiant parametrų $\alpha$ ir $\beta$ reikšmes, kai $\alpha \in [0;360^\circ]$, $\beta \in [0;360^\circ]$.




**2.1 pav.** Trimatės erdvės taškai, išsidėstę ant netiesinių dvimačių daugdarų

- *Nespalvotų ančiuko paveikslėlių duomenų aibė* (Nene *et al*. 1996). Duomenis sudaro to paties objekto (ančiuko) nespalvoti paveikslėliai, gauti palaipsniui apsukant ančiuką 360° kampu. Kiekvienas paveikslėlis yra skaitmenizuojamas, t. y. duomenų taško koordinatės yra sudarytos iš paveikslėlio taškų spalvinių savybių, ir todėl šio taško koordinačių skaičius yra labai didelis. Analizuojamų paveikslėlių, tuo pačiu daugiamatės erdvės taškų skaičius $m = 72$. Kiekvienas paveikslėlis sudarytas iš $128 \times 128$ nespalvotų taškelių, taigi vizualizuojamų taškų koordinačių skaičius $n = 16384$.

- *Fišerio irisų duomenys* (Fisher 1936), kurie kartais vadinami tiesiog irisais arba irisų duomenimis. Tai yra klasikiniai testiniai duomenys, naudojami daugiamačių duomenų analizėje. Išmatuoti keturi 150-ies irisų (vilkdalgių) žiedų parametrai: taurėlapių ilgiai (*sepal length*), taurėlapių pločiai (*sepal width*), vainiklapių ilgiai (*petal length*), vainiklapių pločiai (*petal width*). Matuotos trijų veislių gėlės: *Iris Setosa* (I klasė), *Iris Versicolor* (II klasė) ir *Iris Virginica* (III klasė). Sudaryti 150 4-matės erdvės taškai, kiekvienoje klasėje yra po 50 taškų. Įvairiais metodais nustatyta, kad I klasės irisai skiriasi nuo kitų dviejų (II ir III) klasių, o pastarųjų – labiau giminingi.

- *Panevėžio miesto ir rajono tiriamų mokyklų duomenys*. Švietimo informacinių technologijų centre (ITC) yra sukurta pedagogų duomenų bazių sistema. Iš šių pedagogų duomenų bazių analizei pasirinktos 19 Panevėžio miesto (tyrime jos pažymėtos numeriais nuo 1 iki 19) ir 9 Panevėžio rajono (pažymėtos nuo 20 iki 28) mokyklos: dvi gimnazijos (jų numeriai 1 ir 2), likusios – vidurinės mokyklos. Analizuojami mokytojai, dirbantys 5–12 klasėse, taip pat gimnazinėse klasėse. Norėta išnagrinėti, kokią įtaką mokyklai daro mokytojų kvalifikacija, amžius, mokytojų skaičiaus dinamika, kaip mokyklos grupuojasi pagal pasirinktus parametrus. Remtasi 1999/2000 mokslo metų duomenimis. Mokykloms charakterizuoti yra pasirinkti šie rodikliai:
    - $x_1$ – mokytojų-ekspertų ir mokytojų-metodininkų procentas;
    - $x_2$ – mokytojų, neturinčių tinkamos kvalifikacijos (dirbančių ne pagal savo turimą specialybę), procentas;
    - $x_3$ – mokytojų, kurių amžius viršija 55 metus, procentas;
    - $x_4$ – mokytojų, kurių amžius iki 35 metų, procentas;
    - $x_5$ – mokytojų skaičiaus augimo (mažėjimo) procentas.

## 2.2. Projekcijos metodai

Gamtos ir socialinių mokslų pateikti realūs duomenys dažnai yra charakterizuojami labai dideliu parametrų skaičiumi, t. y. duomenys yra labai didelės dimensijos. Todėl labai sunku šiuos duomenis suprasti. Tačiau daugeliu atveju neaiški, reikalaujanti gilios analizės duomenų struktūra gali būti aprašyta mažu požymių skaičiumi. Dimensijai mažinti sukurta daugybė metodų.

*Projekcijos metodai* (dimensijos mažinimo metodai) taikomi, kai norima daugiamačius duomenis transformuoti į mažesnės dimensijos erdvę. Jų tikslas – pateikti daugiamačius duomenis mažesnės dimensijos erdvėje taip, kad būtų kiek galima tiksliau išlaikyta duomenų struktūra (išsaugotos objektų tarpusavio išsidėstymo charakteristikos – atstumai ar pan.). Projekcijos metodai gali būti naudojami ir daugiamačiams duomenims vizualizuoti, kai pasirinkta pakankamai maža projekcinės erdvės $R^d$ dimensija ($d = 2$ arba $d = 3$). Erdvė $R^d$ dar vadinama vaizdo erdve, nes dažnai jos elementus galima stebėti vizualiai.

Tarkime, kad turime daugiamačius duomenis, kurie išreikšti *n*-matės erdvės taškais $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ ($X_i \in R^n$), $i = \overline{1, m}$; čia $x_{ij}$ yra duomenų *i*-ojo taško

$X_i$ *j*-oji koordinatė, atitinkanti *j*-ąjį parametrą, $n$ – taško $X_i$ koordinačių skaičius, t. y. erdvės, kuriai priklauso taškas $X_i$, dimensija, $m$ – analizuojamų objektų (taškų, vektorių) skaičius. Tikslas – rasti taško $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ transformaciją $Y_i = (y_{i1}, y_{i2}, ..., y_{id})$ mažesnės dimensijos (projekcinėje arba vaizdo) erdvėje $R^d$, $d < n$. Jeigu projekcijos metodo tikslas – vizualizuoti duomenis, tai $d = 2$ arba $d = 3$. Galima nagrinėti ir atvejį, kai $d = 1$, tačiau tada vizualizuojant prarandama daugiau informacijos ir vaizdumo.

Projekcijos metoduose yra naudojamas formalus matematinis kriterijus, pagal kurį minimizuojamas projekcijos iškraipymas

Projekcijos metodai yra skirstomi į dvi dideles grupes:

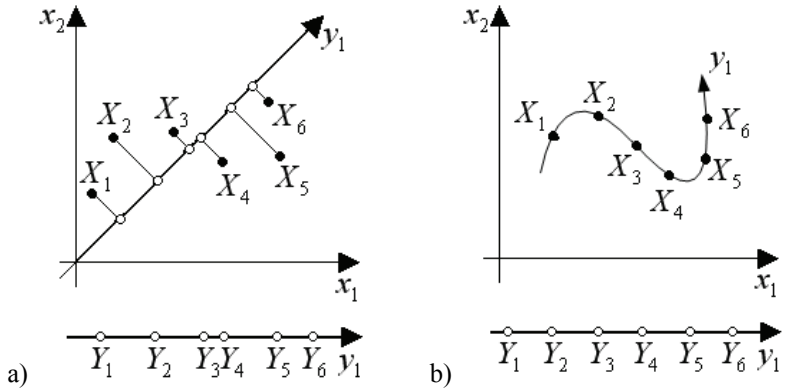
1. *tiesinės projekcijos metodai*: pagrindinių komponenčių analizė (*principal component analysis*), tiesinė diskriminantinė analizė (*linear discriminant analysis*), faktorinė analizė (*factor analysis*), nepriklausomų komponenčių analizė (*independent component analysis*), projekcijos paieška (*projection pursuit*) ir kiti;
2. *netiesinės projekcijos metodai*: daugiamatės skalės (*multidimensional scaling*) ir šio metodo atskiras atvejis – Sammono projekcija (*Sammon mapping, projection*), pagrindinės kreivės (*principal curves*), saviorganizuojantys neuroniniai tinklai (SOM) (*self organizing map*), trianguliacijos metodas (*triangulation method*), lokaliai tiesinis vaizdavimas (*locally linear embedding*) ir kiti.

Tiesinės projekcijos metodais ieškoma tiesinės analizuojamų duomenų transformacijos, o netiesinės projekcijos – netiesinės transformacijos. Tiesinės projekcijos metodai labai efektyvūs, kai duomenys pasiskirsto po tam tikrą poerdvį. Be to, jie reikalauja mažiau skaičiavimų negu netiesinės projekcijos metodai. Tačiau tikslesnė duomenų struktūra išlaikoma naudojant netiesinės projekcijos metodus. Deja ir šiuo atveju duomenų vizualizavimo iškraipymai yra neišvengiami.

Skirtumas tarp tiesinės ir netiesinės projekcijos pavaizduotas 2.2 paveiksle. Plokštumos taškai $X_1, X_2, ..., X_6$ išdėstyti taip, kad tarp artimiausių taškų būtų vienodi atstumai, t. y. $d(X_i, X_{i+1}) = d(X_{i+1}, X_{i+2})$, $i = \overline{1,4}$. Jei šiuos taškus atvaizduosime į vienmatę erdvę (į tiesę $y_1$) naudodami tiesinę projekciją, tai atstumai tarp taškų nebus išlaikyti (2.2a pav.). Tačiau netiesinės projekcijos atveju, radus tinkamą transformaciją, atstumai tarp artimiausių taškų išliks vienodi (2.2b pav.).

Aukščiau minėti projekcijos metodai aprašyti darbuose (Dzemyda *et al*. 2008; Medvedev 2007). Toliau detalizuosime tik tuos projekcijos metodus, kurie išsaugo *lokalią struktūrą*. Čia lokalios struktūros išlaikymu vadiname

atstumų tarp artimiausių taškų santykių išlaikymą po analizuojamos daugiamačių duomenų aibės transformavimo iš didesnio matavimo erdvės į mažesnio matavimo erdvę. Taip pat trumpai bus apžvelgiami dažnai naudojami projekcijos metodai, išsaugantys ne tik lokalią, bet ir *globalią struktūrą*, t. y. atstumų santykius tarp visų duomenų taškų, nes šių metodų trūkumai, atsiskleidę nagrinėjant tam tikras duomenų aibes (daugdaros tipo duomenis), išryškina nagrinėjamų metodų privalumus.



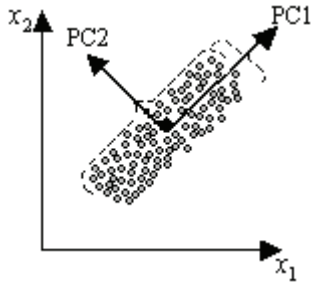

**2.2 pav.** Projekcijos: a) tiesinė ir b) netiesinė

## 2.2.1. Pagrindinių komponenčių analizė

Pagrindinių komponenčių analizė (*principal component analysis*, PCA) yra klasikinis statistikos metodas, sukurtas 1901 m. Karlo Pirsono (*Karl Pearson*) ir nuo to laiko plačiai naudojamas duomenų analizei (pvz., klasifikavimui, atpažinimui). Faktiškai, tai yra tiesinės projekcijos metodas.

PCA metodo tikslas – sumažinti duomenų dimensiją atliekant tiesinę transformaciją ir atsisakant dalies po transformacijos gautų naujų komponenčių, kurių dispersijos yra mažiausios (Jolliffe 1989; Opitz and Hilbert 2000; Yeung and Ruzzo 2001; Taylor 2003). Taigi PCA metodu siekiama maksimaliai išlaikyti dispersiją duomenyse.

Iš pradžių ieškoma krypties, kuria dispersija yra didžiausia. Didžiausią dispersiją turinti kryptis vadinama pirmąja pagrindine komponente (PC1). Ji eina per duomenų centrinį tašką. Tai taškas, kurio koordinatės yra analizuojamą duomenų aibę sudarančių taškų atskirų koordinačių vidurkiai. Visų taškų vidutinis atstumas iki šios tiesės yra minimalus, t. y. ši tiesė yra kiek galima arčiau visų duomenų taškų. Antrosios pagrindinės komponentės (PC2) ašis taip pat turi eiti per duomenų centrinį tašką ir ji turi būti statmena pirmosios pagrindinės komponentės ašiai (2.3 pav.).




**2.3 pav.** Pirmoji (PC1) ir antroji (PC2) pagrindinės komponentės

Tegu turime duomenų matricą $X = \{X_1, X_2, ..., X_m\} = \{x_{ij}, i = \overline{1,m},\ j = \overline{1,n}\}$, kurioje eilutės žymi vektorius $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ (jų yra $m$), o stulpeliai – duomenų parametrus $x_1, x_2, ..., x_n$ (jų yra $n$).

Norint rasti pagrindines komponentes, reikia sudaryti duomenų kovariacinę matricą ir apskaičiuoti jos tikrines reikšmes (*eigenvalues*) bei tikrinius vektorius (*eigenvectors*). Matricos $C$ *tikrinis vektorius* (*eigenvector*) $f_l$ ir jį atitinkanti *tikrinė reikšmė* (*eigenvalue*) $\lambda_l$ yra lygties $Cf_l = \lambda_l f_l$ sprendinys. Šioje lygtyje $f_l$ yra vektorius stulpelis, $C$ – kvadratinė $n \times n$ kovariacinė matrica, $\lambda_l$ reikšmė randama iš charakteristinės lygties $|C - \lambda_l I| = 0$. Čia $I$ yra vienetinė matrica, kurios matmenys tokie pat kaip matricos $C$; ženklu |.| apibrėžtas determinantas. Tikrinių vektorių ir tikrinių reikšmių radimas nėra trivialus uždavinys, tačiau yra sukurta nemažai šio uždavinio sprendimo metodų (Kvedaras and Sapagovas 1974).

Surūšiavus tikrinius vektorius $f_l$ juos atitinkančių tikrinių reikšmių mažėjimo tvarka ($\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$), sudaroma matrica $A = (f_1, f_2, ..., f_n)$, kurios stulpeliai yra surūšiuoti tikriniai vektoriai $f_l$, $l = \overline{1,n}$. Ši matrica vadinama pagrindinių komponenčių matrica, kurios kiekvienas vektorius stulpelis yra ortogonalus bet kuriam kitam. Bet kurį duomenų vektorių $X_i$, $i = \overline{1,m}$ galima transformuoti pagal formulę $Y_i = (X_i - \overline{X})A$, čia $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $\overline{X} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$, $\overline{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$. Gauti $Y_i = (y_{i1}, y_{i2}, ..., y_{in})$ yra taškai naujoje ortogonalioje koordinačių sistemoje $(y_1, y_2, ..., y_n)$, apibrėžtoje tikriniais vektoriais $f_l$, $l = \overline{1,n}$.

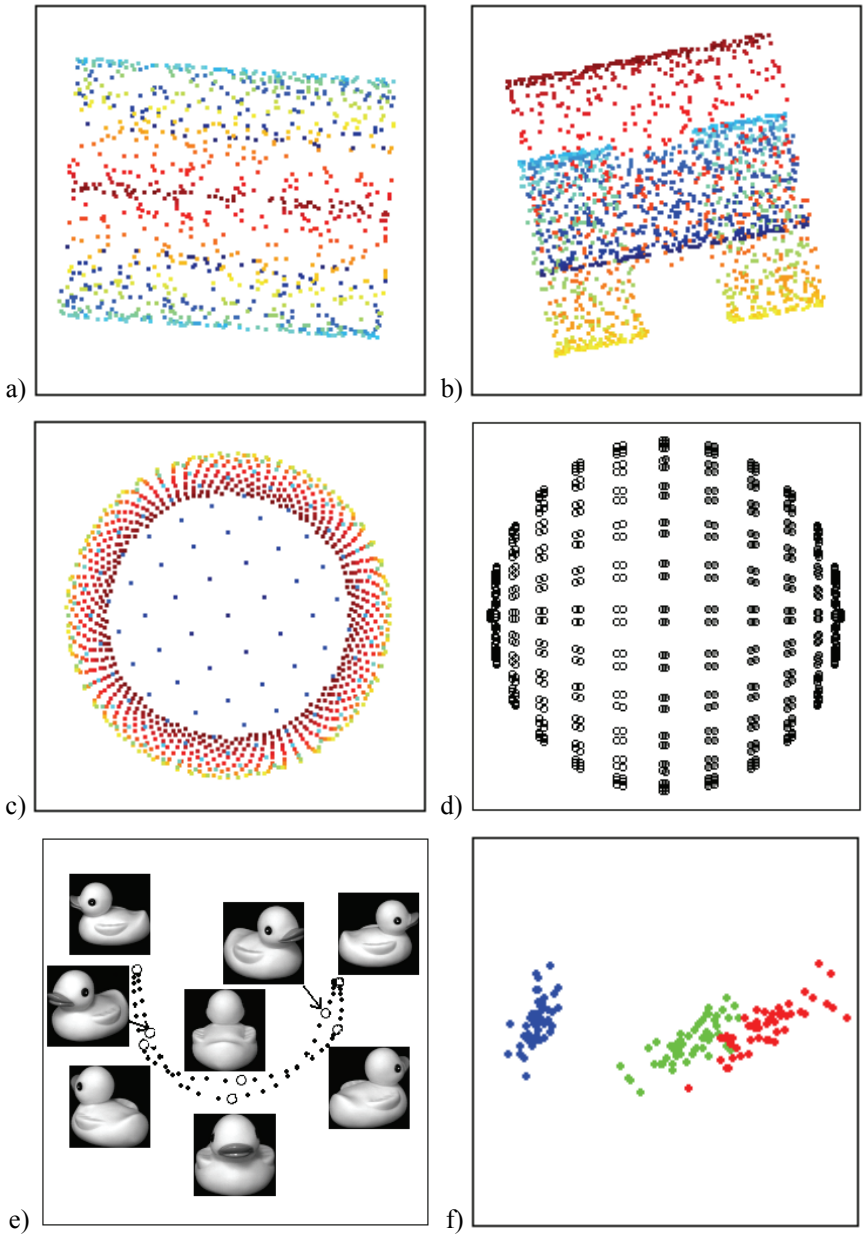
Pirmasis tikrinis vektorius $f_1$ aprašo naują parametrą $y_1$, kurio dispersija $\lambda_1$ yra pati didžiausia tarp $\lambda_1, \lambda_2, ..., \lambda_n$, t. y. šis parametras ir yra pirmoji pagrindinė komponentė. Antrasis tikrinis vektorius $f_2$ aprašo parametrą $y_2$, kurio dispersija yra antra pagal dydį ir kuris yra antroji pagrindinė komponentė, ir t. t.

Dažnai tik dalis tikrinių reikšmių yra esminės, nes kitų reikšmės yra labai mažos. Todėl daugiamačių duomenų transformacijai galima naudoti ne visus *n* tikrinius vektorius, o tik pirmuosius *d* ( $d < n$ ). Tuomet matrica $A$ sudaroma tik iš *d* tikrinių vektorių, turinčių didžiausias tikrines reikšmes. Yra skurta įvairių metodikų, kaip iš anksto (ar algoritmo metu) nustatyti skaičiaus *d* reikšmę. Jei analizės tikslas – rasti pagrindines komponentes ir transformuotus duomenis pavaizduoti plokštumoje ar trimatėje erdvėje, t. y. juos vizualizuoti, tuomet $d = 2$ arba $d = 3$.

PCA metodu vizualizuoti duomenys, daugiamatės erdvės taškai priklausantys įvairioms netiesinėms dvimatėms daugdaroms (a-e atvejai) ir irisų duomenys (f atvejis), kai duomenys atvaizduojami plokštumoje, pateikti 2.4 paveiksle.

Pagrindinių komponenčių metodo privalumas yra jo idėjos paprastumas. Gal dėl to šis metodas populiarus jau daugiau nei 100 metų ir iki šiol plačiai taikomas. Tačiau PCA metodas turi ir trūkumų:

- jei egzistuoja tiesinės priklausomybės tarp parametrų $x_1, x_2, ..., x_n$, tai duomenų dimensija mažinama su nedidelėmis paklaidomis; tačiau dažniausiai tarp realių duomenų egzistuoja stiprūs netiesiniai sąryšiai, kurių šis metodas negali įvertinti;
- vizualizavimo rezultatai labai priklauso nuo taškų atsiskyrėlių (*outliers*), nes šie taškai *n*-matėje erdvėje yra ryškiai nutolę nuo kitų ir dirbtinai padidina dispersijos reikšmę, o tai daro didelę įtaką nustatytoms pagrindinėms komponentėms bei duomenų projekcijos kokybei;
- netinka vizualizuoti duomenis, išsidėsčiusius ant netiesinės daugdaros, nes tolimi taškai ant daugdaros paviršiaus gautoje projekcijoje tampa artimi (2.4a pav., 2.4b pav.).




**2.4 pav.** PCA metodu vizualizuoti duomenys: a) S-formos daugdaros taškai, b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai, c) sferos nuopjovos taškai, d) sferos taškai, e) ančiuko paveikslėlių duomenys, f) irisų duomenys

## 2.2.2. Daugiamatės skalės

Daugiamačių skalių (*multidimensional scaling*, MDS) metodas (Borg and Groenen 2005) naudojamas daugiamačių duomenų analizei įvairiose mokslo srityse, pvz., socialiniuose moksluose, medicinoje ir t. t. Sukurta daug šio metodo realizacijų, kurios skiriasi naudojamais vizualizavimo kokybės kriterijais, optimizavimo algoritmais ar prielaidomis apie duomenis.

Naudojantis MDS, ieškoma daugiamačių duomenų projekcijų mažesnės dimensijos erdvėje (dažniausiai $R^2$ arba $R^3$), siekiant išlaikyti analizuojamos aibės objektų artimumus – panašumus arba skirtingumus. 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.

Vienas daugiamačių skalių metodų tikslų yra rasti optimalų daugiamačius objektus atitinkančių taškų vaizdą mažos dimensijos erdvėje.

Tegu kiekvieną *n*-matės erdvės tašką $X_i \in R^n$, $i = \overline{1,m}$, atitinka mažesnės dimensijos erdvės taškas $Y_i \in R^d$, $d < n$. Atstumą tarp taškų $X_i$ ir $X_j$ pažymėkime $d(X_i, X_j)$, o atstumą tarp taškų $Y_i$ ir $Y_j$ – $d(Y_i, Y_j)$, $i, j = \overline{1,m}$.

Naudojantis MDS algoritmu, bandoma atstumus $d(Y_i, Y_j)$ priartinti prie atstumų $d(X_i, X_j)$. Jei naudojama kvadratinė paklaidos funkcija, tai minimizuojama tikslo funkcija $E_{MDS}$ gali būti užrašyta taip: $E_{MDS} = \sum_{i<j} w_{ij} \left( d(X_i, X_j) - d(Y_i, Y_j) \right)^2$. Paklaidos funkcija $E_{MDS}$ dar vadinama *Stress* funkcija. Svoriai $w_{ij}$ apskaičiuojami pagal formules: $w_{ij} = \frac{1}{\sum_{i<j} \left( d(X_i, X_j) \right)^2}$, $w_{ij} = \frac{1}{d(X_i, X_j) \sum_{k<l} d(X_k, X_l)}$, $w_{ij} = \frac{1}{m d(X_i, X_j)}$, čia $d(X_i, X_j) \neq 0$.

Literatūroje minimi įvairūs paklaidos funkcijos optimizavimo būdai: gradientinis nusileidimas, jungtinių gradientų metodas, kvazi-Niutono metodas, deterministinis atkaitinimo modeliavimo algoritmas (*simulated annealing*) (Klock and Buhmann 2000), evoliucinis algoritmas (Michalewicz 1996), 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;

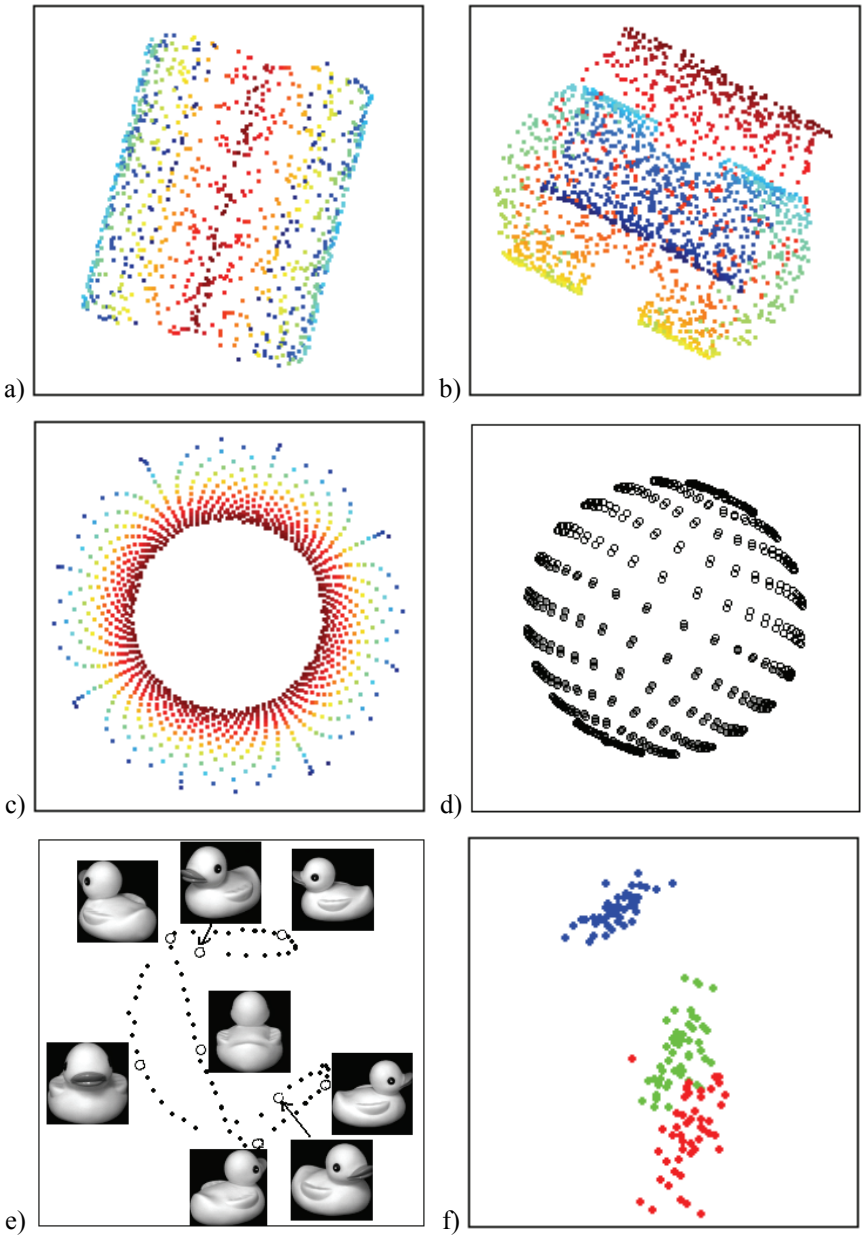
Podlipskytė 2004), SMACOF (*scaling by majorization a complicated function*) algoritmas, pagrįstas tikslo funkcijos mažorizavimu (Borg and Groenen 2005). Šioje disertacijoje tyrimams bus naudojamas SMACOF algoritmas.

2.5 paveiksle parodytas MDS metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant įvairių netiesinių dvimačių daugdarų taškus (a-f) ir irisų (g) duomenis.

Įprastinio MDS algoritmo trūkumai:

- algoritmo tikslo funkcijos optimizavimas reikalauja palyginimų tarp visų taškų porų;
- algoritmai yra neefektyvūs, dirbant su didelės apimties duomenų aibėmis;
- negalima vizualizuoti naujų taškų tol, kol nebus perskaičiuoti visi analizuojami taškai;
- įprastinis MDS metodas netinka daugdaros tipo duomenims vizualizuoti, nes tolimi taškai ant daugdaros paviršiaus gautoje projekcijoje tampa artimi (2.5 pav.).

Siekiant išvengti minėtų problemų, pasiūlyta įvairios MDS algoritmo modifikacijos (Basalaj 1999; Naud and Duch 2000; Bernatavičienė *et al*. 2007; Tenenbaum *et al*. 2000) ir kiti.




**2.5 pav.** MDS metodu vizualizuoti duomenys: a) S-formos daugdaros taškai, b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai, c) sferos nuopjovos taškai, d) sferos taškai, e) ančiuko paveikslėlių duomenys, f) irisų duomenys

### 2.2.3. Sammono algoritmas

Sammono projekcija, dažnai vadinama Sammono metodu ar algoritmu, (Sammon 1969) yra netiesinis objektų, apibūdinamų daugeliu parametrų, atvaizdavimas mažesnės dimensijos erdvėje $R^d$, dažniausiai $d = 2$. Tai vienas iš daugiamačių skalių (MDS) grupės metodų (Bezdek and Pal 1995). Metodo idėja – atvaizduoti daugiamačius objektus atitinkančius taškus mažesnės dimensijos erdvėje išlaikant atstumų tarp atitinkamų daugiamatės erdvės taškų ir jų projekcijų santykius. Sammono projekcija minimizuoja projekcijos iškraipymą (paklaidą) $E_S$, kuri darbe bus vadinama Sammono paklaida:

$$E_S = \frac{1}{\sum_{i<j} d(X_i, X_j)} \sum_{i<j} \frac{\left(d(X_i, X_j) - d(Y_i, Y_j)\right)^2}{d(X_i, X_j)}.$$

Funkcija $E_S$ sutampa su $E_{MDS}$ funkcija, kai $w_{ij} = \frac{1}{d(X_i, X_j)\sum_{k<l} d(X_k, X_l)}$. Detaliau apie Sammono metodą žr. 3.2 skyrelyje.


**2.6 pav.** Sammono metodu vizualizuoti Panevėžio miesto ir rajono tiriamų mokyklų duomenys

### 2.2.4. Trianguliacija

Darbe (Lee *et al*. 1977) pateiktas nuoseklaus daugiamatės erdvės taškų atvaizdavimo plokštumoje būdas taikant trianguliacijos metodą. Jo esmė: duomenų taškai atvaizduojami ne iš karto visi, bet vienas paskui kitą; taškas dedamas atsižvelgiant į anksčiau atvaizduotų taškų išsidėstymą. Šiuo metodu, skirtingai nei daugiamatėmis skalėmis, nesistengiama išlaikyti atstumų tarp visų daugiamatės erdvės taškų ir jų projekcijų plokštumoje santykius. Atvaizduojant daugiamačius duomenis plokštumoje, tiksliai išlaikomi atstumai nuo kiekvieno taško iki kažkurių dviejų anksčiau atvaizduotų taškų; atstumai tarp tų dviejų taškų taip pat yra tiksliai išlaikyti. Tuo būdu, atvaizduojant duomenis iš daugiamatės erdvės į dvimatę, tiksliai išlaikomi $(2m-3)$ atstumai iš galimų $m(m-1)/2$ atstumų (čia $m$ – analizuojamų taškų skaičius).

Galimi du metodai, pagal kuriuos nustatoma, iki kurių dviejų jau atvaizduotų duomenų aibės taškų atstumų didumas turi būti tiksliai išlaikytas, atvaizduojant naują tašką iš *n*-matės į dvimatę erdvę. Tai antrojo arčiausiojo kaimyno (*second nearest neighbour*) ir atramos taško (*reference point*) metodai. Taikant antrojo arčiausiojo kaimyno metodą, tiksliai išlaikomi atstumai nuo kiekvieno vizualizuojamo taško iki dviejų artimiausių jo kaimynų. Naudojant atramos taško metodą, iš duomenų aibės taškų vienas taškas pasirenkamas atramos tašku, iki kurio atstumai nuo visų kitų taškų bus visada išlaikomi. Taigi šiuo atveju išlaikomi kiekvieno taško atstumai iki pirmojo arčiausiojo kaimyno bei atramos taško (detaliau žr. 3.1 skyrelyje).

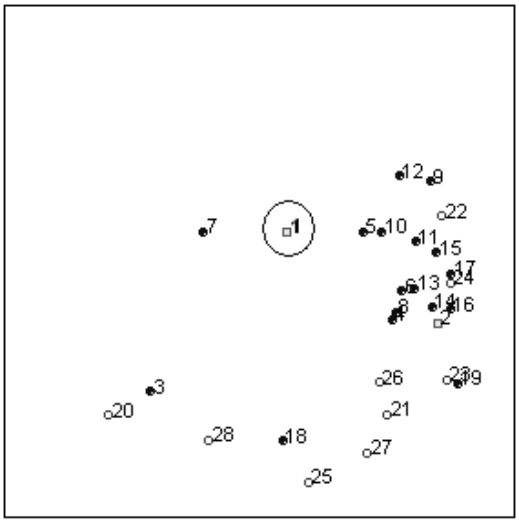
Atramos taško metodas naudojamas, norint parodyti visų analizuojamų duomenų padėtį vieno pasirinkto taško (atramos taško) atžvilgiu. Siekiant kuo geriau ištirti duomenis, gali būti naudojami įvairūs atramos taškai. Tokiu būdu galima gauti daugybę skirtingų atvaizdavimų iš tų pačių duomenų. Tai tas pats, kas stebėti objektą erdvėje iš skirtingų pozicijų (Lee *et al*. 1977).

Šioje disertacijoje skirtumas tarp antrojo arčiausiojo kaimyno ir atramos taško metodų išryškintas, analizuojant Panevėžio miesto ir rajono tiriamų mokyklų duomenis. 2.7 paveiksle parodytas antrojo arčiausiojo kaimyno metodo rezultatas, kur pradiniu tašku vizualizavimo metu buvo pasirinkta Panevėžio „Saulėtekio“ vidurinė mokykla, pažymėta numeriu 10. Kiekviena mokykla atvaizduota dviejų artimiausių pagal nagrinėjamus požymius mokyklų atžvilgiu.

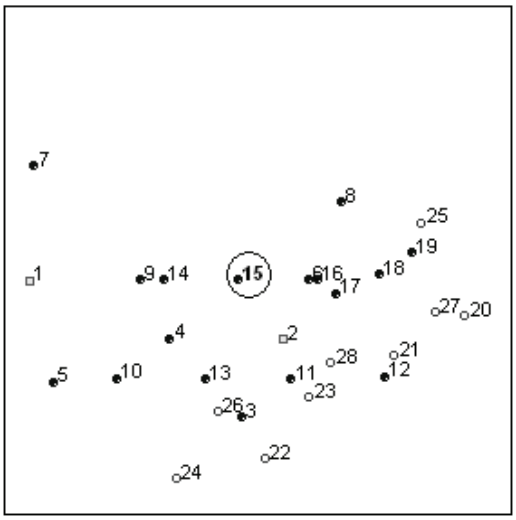
2.8 paveikslas pateikia mokyklų išsidėstymą, gautą panaudojus atramos taško metodą, kai atramos tašku pasirinkta Juozo Balčikonio gimnazija (nr. 1). Galima vizualiai vertinti mokyklų panašumą. Matome, kad dauguma Panevėžio miesto mokyklų (nr. 3–19, užtušuoti skrituliukai) yra arčiau Juozo Balčikonio gimnazijos nei rajono mokyklos (nr. 20–28, tušti skrituliukai). Todėl galima daryti išvadą, kad miesto mokyklos panašesnės pedagoginio personalo kvalifikacijos, amžiaus ir kaitos požiūriu į Juozo Balčikonio gimnaziją nei

rajono mokyklos. 2.9 paveiksle atramos tašku pasirinkta miesto mokykla – Alfonso Lipniūno vidurinė mokykla (numeris 15), o 2.10 paveiksle pasirinkta rajono mokykla – Krekenavos Mykolo Antanaičio vidurinė mokykla (numeris 21).


**2.7 pav.** Panevėžio miesto ir rajono tiriamų mokyklų duomenų vizualizavimas antrojo arčiausiojo kaimyno metodu




**2.8 pav.** Panevėžio miesto ir rajono tiriamų mokyklų duomenų vizualizavimas atramos taško metodu, kai atramos tašku pasirinkta Juozo Balčikonio gimnazija (numeris 1)




**2.9 pav.** Panevėžio miesto ir rajono tiriamų mokyklų duomenų vizualizavimas atramos taško metodu, kai atramos tašku pasirinkta Panevėžio miesto Alfonso Lipniūno mokykla (numeris 15)


**2.10 pav.** Panevėžio miesto ir rajono tiriamų mokyklų vizualizavimas atramos taško metodu, kai atramos tašku pasirinkta Panevėžio rajono Krekenavos Mykolo Antanaičio mokykla (numeris 21)

Taigi trianguliacijos metodo privalumas – naudojant tą pačią duomenų aibę, tačiau pasirenkant skirtingus, tam tikra prasme tyrėjui įdomius atramos taškus, galima gauti daugybę skirtingų atvaizdavimų, kurie visumoje padeda geriau suvokti analizuojamus duomenis.

Lyginant su pagrindinių komponenčių analizės metodu, trianguliacijos metodo privalumas yra tai, kad šiuo metodu galima analizuoti ir ne skaitmeninius duomenis. Darbe (Lee *et al*. 1977), taikant trianguliacijos metodą, analizuojami baltymo duomenys, susiję su amino rūgštimis baltymo molekulėje, citochromu *c*, rasto gyvūnų ir aukštesnių augalų mitochondrijoje; užrašytos tiktai tos pozicijos, kurios skiriasi. Kadangi šie duomenys nėra skaitmeniniai (duomenų taško kiekvienos koordinatės reikšmė yra didžioji raidė), negalima naudoti Euklido atstumų. Todėl naudojami Hamingo atstumai (*Hamming distance*). Hamingo atstumas tarp dviejų taškų $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ ir $X_j = (x_{j1}, x_{j2}, ..., x_{jn})$ apibrėžiamas taip:

$$d(X_i, X_j) = \sum_{l=1}^{n} \delta(x_{il}, x_{jl}),$$

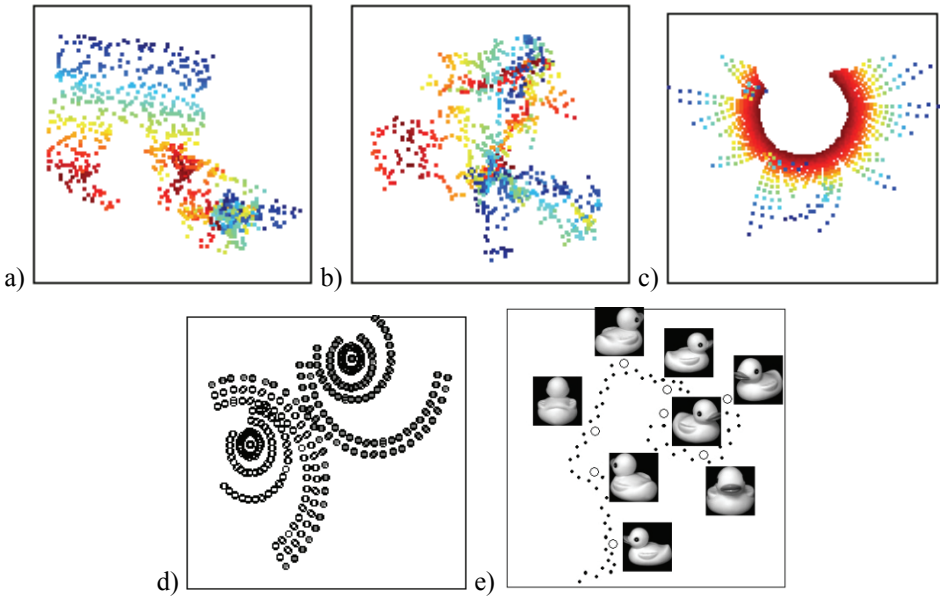
kur $\delta(x_{il}, x_{jl}) = 1$, jei $x_{il} \neq x_{jl}$, ir $\delta(x_{il}, x_{jl}) = 0$, jei $x_{il} = x_{jl}$.

Disertacijoje parodyta, kad trianguliacijos metodu galima greitai ir su nedidele paklaida atvaizduoti naujus duomenų taškus, kai pradiniai duomenų aibės taškai vizualizuoti vienu iš MDS grupės metodų (detaliau žr. 3.3 skyrelyje). Tai dar vienas trianguliacijos metodo privalumas.

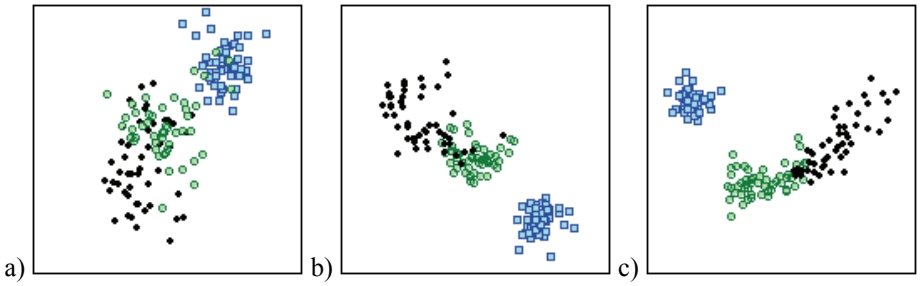
Trianguliacijos metodu vizualizavus plokštumoje daugiamatės erdvės taškus, išsidėsčiusius ant arba arti netiesinės daugdaros, nustatyta, kad šis metodas nėra tinkamas netiesinės daugdaros tipo duomenims vizualizuoti (2.11 pav.).

Tyrimas su irisų duomenimis atskleidė dar vieną trianguliacijos metodo trūkumą – gauta projekcijos paklaida labai priklauso nuo taškų atvaizdavimo sekos (pirma taškai surūšiuojami tam tikra tvarka, o po to nuosekliai atvaizduojami plokštumoje). Trianguliacijos metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant irisų duomenis, parodytas 2.12 paveiksle. Taškai, atitinkantys pirmos klasės irisus, pažymėti mėlynais kvadratėliais, antros klasės irisus – žaliais skrituliukais, trečios klasės – juodais skrituliukais. 2.12a paveiksle taškai atvaizduoti plokštumoje taikant antrojo arčiausiojo kaimyno metodą, kai pradiniu tašku vizualizuojant duomenis pasirinktas pirmas duomenų aibės taškas. Matome, kad antros klasės taškai įsimaišo tarp pirmos ir trečios klasės taškų, t. y. duomenys tarsi sudaro vieną klasterį. 2.12b paveiksle taškai atvaizduoti plokštumoje taikant antrojo arčiausiojo kaimyno metodą, kai pradiniu tašku pasirinktas duomenų aibės

taškas, kurio numeris 102. Šiuo atveju taškai, atitinkantys pirmos klasės irisus, sudaro atskirą grupę, tačiau antros ir trečios klasės taškai gerokai persidengia. 2.12c paveiksle panaudotas atramos taško metodas, kai atramos tašku pasirinktas pirmas duomenų aibės taškas. Gautoje projekcijoje antros ir trečios klasių sankirta mažesnė nei 2.12b paveiksle.




**2.11 pav.** Trianguliacijos metodu vizualizuoti daugiamatės erdvės taškai, esantys ant arba arti netiesinių dvimačių daugdarų: a) S-formos daugdaros (antrojo arčiausiojo kaimyno metodu), b) S-formos daugdaros (atramos taško metodu), c) sferos nuopjovos (antrojo arčiausiojo kaimyno metodu), d) sferos (antrojo arčiausiojo kaimyno metodu), e) ančiuko paveikslėlių duomenys (antrojo arčiausiojo kaimyno metodu)




**2.12 pav.** Trianguliacijos metodu vizualizuoti irisų duomenys

## 2.2.5. Santykinės perspektyvos metodas

Santykinės perspektyvos metodas (*relational perspective map,* RPM) (Li 2004) – tai netiesinis duomenų dimensijos mažinimo metodas, atvaizduojantis duomenis ant toro paviršiaus (stačiakampyje). *Torą* apibrėžia spindulio *R* apskritimas, besisukantis apie jo plokštumoje esančią tiesę, nutolusią nuo apskritimo centro atstumu $a > R$. RPM, kaip ir daugiamačių skalių (*multidimensional scaling*) metodai, stengiasi išlaikyti atstumų tarp atitinkamų daugiamatės erdvės taškų ir jų projekcijų ant toro santykius. Tačiau RPM gali išsaugoti daugiau sudėtingos duomenų aibės lokalių savybių nei kiti daugiamačių skalių metodai. Pagrindinė RPM metodo savybė – galėjimas padalyti sudėtingą duomenų aibę į dalis ir vizualizuoti duomenis plokštumoje taip, kad jų projekcijos nepersidengtų. Todėl RPM metodas išlaiko atstumus nedidelėje srityje (*short-range distances*) tiksliau negu kiti projekcijos metodai.

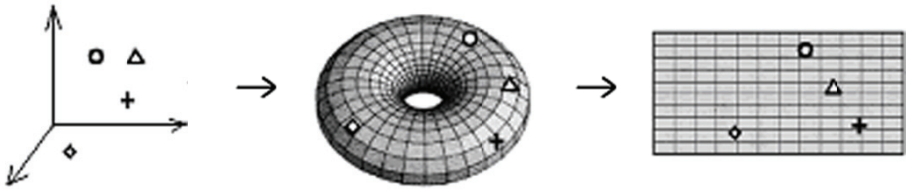
Tam, kad vizualizuojamų taškų projekcijos nepersidengtų ir atstumai būtų kuo tiksliau išlaikyti, taškų atvaizdavimui pasirinktas toro paviršius. Vizualizuojant taškus ant toro, gaunami dvimačiai taškai, kurie suprantami kaip dalelės, galinčios laisvai judėti ant toro paviršiaus, bet negalinčios nuo jo atitrūkti. Veikiamos stūmos jėgų, dalelės juda ant toro paviršiaus, kol jų išsidėstymas atitinka atstumus tarp daugiamatės erdvės taškų.

Kaip pavaizduota 2.13 paveiksle, RPM metodas pirmiausia atvaizduoja daugiamatės erdvės taškus ant toro paviršiaus, o tada, torą išardžius (torą perpjauname pasirinktoje vietoje ir gauname cilindrą, kurį vėl pjauname per vieną jo sudaromąją), gaunamas stačiakampis, kuriame ir matome daugiamatės erdvės taškų projekcijas.

Santykinės perspektyvos metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant irisų duomenis, parodytas 2.14 paveiksle.

Kaip geriau suprasti daugiamačių duomenų atvaizdavimą ant toro paviršiaus? Vizualizuokime sferos taškus (2.1d pav.) RPM metodu. Taškų atvaizdavimas ant toro pateiktas 2.15 paveiksle. Norint įsivaizduoti, kaip taškai išsidėsto ant toro, reikėtų sulipdyti priešingas stačiakampio kraštines. Tačiau to padaryti mes negalime, todėl sudedame iš eilės keletą atvaizdavimų (2.15 pav.). 2.16 paveiksle vaizdžiai matome viršutinę (šviesūs taškai) ir apatinę (tamsūs taškai) pussferes.

Šiame skyrelyje RPM metodu vizualizuoti ir trimatės erdvės taškai, išsidėstę ant netiesinės dvimatės S-formos daugdaros (2.1a pav.). Tyrimas parodė (2.17 pav.), jog RPM metodas nėra tinkamas vizualizuoti šiuos taškus.

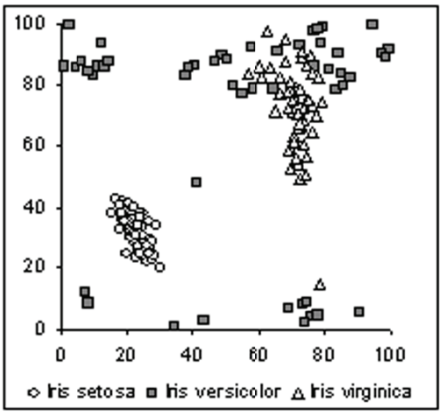


Duomenų aibė, sudaryta iš 4 trimatės erdvės taškų

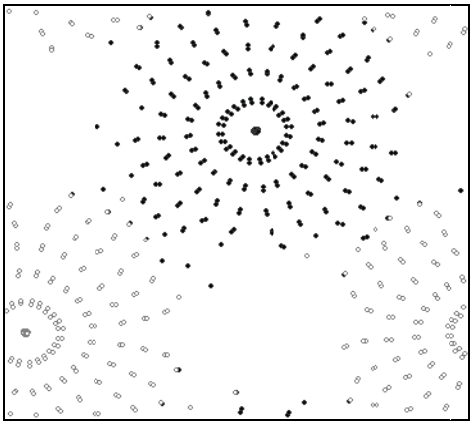
4 taškai ant toro paviršiaus

Duomenų vizualizavimas RPM metodu

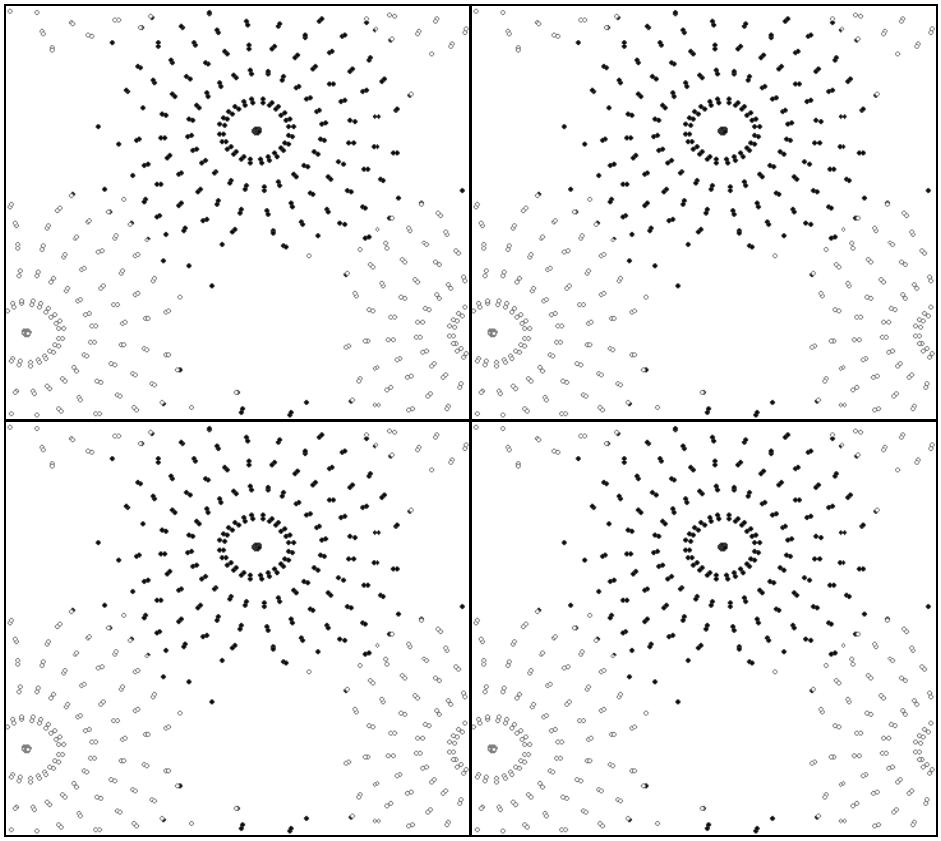
**2.13 pav.** RPM metodo modelis



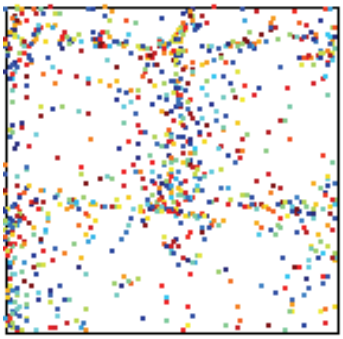

**2.14 pav.** RPM metodu vizualizuoti irisų duomenys



**2.15 pav.** Sferos taškų vizualizavimas RPM metodu



**2.16 pav.** Sferos taškų atvaizdavimo ant toro paviršiaus paaiškinimas



**2.17 pav.** RPM metodu vizualizuoti netiesinės dvimatės S-formos daugdaros taškai

## 2.3. Netiesinės daugdaros atpažinimo metodai

Skyrelio pradžioje pateiksime keletą svarbių sąvokų paaiškinimų (Adler and Taylor 2007; Lee and Verleysen 2007).

**Apibrėžimas**. *Topologinė erdvė* yra aibė $X$, kurioje nurodyta jos *topologija* $\widetilde{T}$, t. y. aibės $X$ poaibių šeima (rinkinys), pasižyminti šiomis savybėmis: dviejų šeimos $\widetilde{T}$ elementų sankirta yra tos šeimos elementas, bet kuri šeimos $\widetilde{T}$ elementų sąjunga yra tos šeimos elementas, tuščioji aibė ir visa aibė $X$ yra šeimos $\widetilde{T}$ elementai. Topologijos $\widetilde{T}$ elementai vadinami atvirosiomis aibėmis (Lee and Verleysen 2007).

Disertacijoje nagrinėjame tik Euklido erdvę. Kadangi ši erdvė yra metrinė (apibrėžtas atstumas), tai joje įvedama metrinė topologija. Šiuo atveju tarp atvirųjų aibių taškų galima skaičiuoti atstumus ir rasti taškų kaimynus.

**Apibrėžimas**. *Atvirąja aibe* vadinama aibė, kurioje apie bet kurį jos tašką galima apibrėžti atvirąjį rutulį, kuris priklausytų tai aibei.

**Apibrėžimas**. Tarkime, kad turime taško atvirąją aplinką (atvirąją aibę), kurioje yra tas taškas. Tuomet visus toje aplinkoje esančius taškus vadinsime to taško *kaimynais*.

*Daugdara* yra topologinė erdvė, kurioje kiekvieno taško aplinkoje įvesta koordinačių sistema. Nors daugdara apibendrina bet kokio matavimo erdvėje kreivės ir paviršiaus sąvokas, bet lokaliai daugdara nesiskiria nuo Euklido erdvės: ji sudaryta iš suklijuotų Euklido erdvės gabalų (Adler and Taylor 2007).

Kalbant apie daugdarą, svarbi yra jos dimensija. Pavyzdžiui, linija yra vienmatė, plokštuma – dvimatė ir pan. Vienmatėje daugdaroje kiekvieno taško aplinka yra panaši į tiesės segmentą. Linija, apskritimas yra vienmatės daugdaros, t. y. daugdarų dimensija lygi vienam. Dvimatėje daugdaroje kiekvieno taško aplinka panaši į skritulį. Plokštuma, sferos paviršius, toro paviršius yra dvimatės daugdaros, t. y. šių daugdarų dimensija lygi dviem.

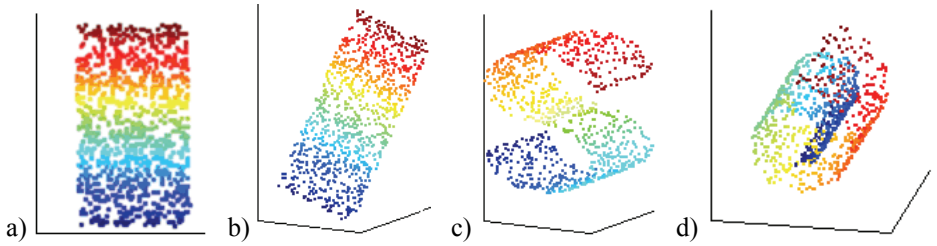
**Apibrėžimas**. Tegu $M$ yra $d$-matė daugdara, t. y. kiekvienas jos taškas $X_i$ turi aplinką $U_{X_i}$, kuri bijektyviai atvaizduojama į aibės $R^d$ atvirąjį poaibį, t. y. bet kuriam aplinkos $U_{X_i}$ taškui $X_k$ galima rasti $d$ realiųjų skaičių – jo koordinačių $(x_{k1}, x_{k2}, ..., x_{kd})$. Jei $V_{X_i}$ – kita taško $X_i$ aplinka, $(\widetilde{x}_{k1}, \widetilde{x}_{k2}, ..., \widetilde{x}_{kd})$ – kitos taško $X_k$ koordinatės, tai ryšys tarp koordinačių $(x_{k1}, x_{k2}, ..., x_{kd})$ ir $(\widetilde{x}_{k1}, \widetilde{x}_{k2}, ..., \widetilde{x}_{kd})$ užrašomas funkcijomis $\widetilde{x}_{ki} = f_i(x_{k1}, x_{k2}, ..., x_{kd})$, $i = \overline{1,d}$. Jei funkcijos $f_i$ tolydžios, tai *daugdara topologinė*, o jei funkcijos $f_i$ diferencijuojamos, tai *daugdara* yra *glodi*.

**Whitney teorema (1936).** Bet kurią glodžiąją *d*-matę daugdarą galima įdėti į pakankamai didelės dimensijos *n* Euklido erdvę $R^n$.

Šioje disertacijoje yra nagrinėjami daugiamatės erdvės taškai, esantys ant arba arti netiesinių glodžiųjų daugdarų.

Įvairiose mokslo srityse kaupiami ir analizuojami daugiamačiai duomenys, kuriuos atitinkantys taškai nagrinėjami labai didelės dimensijos erdvėje, o iš tikrųjų jie yra kokios nors mažesnės dimensijos daugdaros taškai arba tai daugdarai artimi taškai. Netiesinės daugdaros atpažinimo metodų tikslas – atrasti mažesnio matavimo netiesinę daugdarą didelio matavimo erdvėje ir tada transformuoti duomenų taškus, išsidėsčiusius ant arba arti tos daugdaros, į mažesnio matavimo erdvę, t. y. atskleisti (*unfold*) daugdarą.

2.18 paveiksle pateiktas paprastas pavyzdys. Nagrinėjami trimatės erdvės taškai, priklausantys dvimatei daugdarai (2.18a pav.), kuri yra įdėta į trimatę erdvę trimis skirtingais būdais: tiesinis įdėjimas (plokštuma) – 2.18b pav., S-forma –2.18c pav., spiralinis cilindras (*„swiss roll“*) – 2.18d pav. S-formos daugdara ir spiralinis cilindras yra tarsi susuktas stačiakampis popieriaus lapas (2.18a pav.) Analizei būtų pateikiamas duomenų rinkinys, pavaizduotas 2.18b, 2.18c ir 2.18d paveiksluose, o tikslas – transformuoti šiuos duomenis į mažesnio matavimo erdvę (2.18a pav.).




**2.18 pav.** Taškai, priklausantys dvimatei daugdarai (a), įdėtai į trimatę erdvę trimis skirtingais būdais: (b) tiesinis įdėjimas (plokštuma), (c) S-forma, (d) spiralinis cilindras („*swiss roll*“)

*Duomenų vidinė dimensija* (*intrinsic dimensionality*) paprastai apibrėžiama kaip minimalus parametrų ar paslėptų kintamųjų (*latent variables*) skaičius, reikalingas duomenims aprašyti (Lee and Verleysen 2007). Dažnai paslėpti kintamieji dar vadinami duomenų *laisvės laipsniais* (*degrees of freedom*) (Tenenbaum *et al*. 2000; Lee and Verleysen 2007). Tegu analizuojamų duomenų dimensija yra *n*. Labai didelės dimensijos duomenys gali turėti reikšmingas mažesnės dimensijos struktūras, paslėptas analizuojamoje daugiamatėje erdvėje, t. y. duomenų vidinė dimensija *d* yra daug mažesnė ($d \ll n$). Tuomet laikoma, kad šie duomenys yra ant arba arti glodžiosios mažesnės dimensijos, t. y. *d*-matės daugdaros. Sprendžiant daugdaros atpažinimo uždavinį, šie duomenų

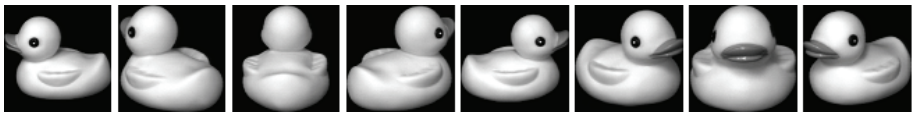
taškai turi būti transformuojami į tokios dimensijos erdvę, kokia yra daugdaros dimensija *d*. Tačiau vizualizavimo atveju, projekcinės erdvės dimensija turi būti pasirinkta 2 arba 3 nepriklausomai nuo daugdaros dimensijos. Tai gali sukelti sunkumų, taikant daugdaros atpažinimo metodus, nes visų metodų vienas parametras ir yra daugdaros dimensija *d*. Netinkamos šio parametro reikšmės labai įtakoja rezultatus. Iš vienos pusės, per didelė *d* reikšmė sustiprina triukšmo efektus, t. y. projekcijos tampa triukšmingos ir, kai kuriais atvejais, nestabilios, tuo tarpu, per maža šio parametro reikšmė nulemia projekcijų persidengimą (Levina and Bickel 2005; Yin *et al*. 2007). Šioje disertacijoje sprendžiamas tik vizualizavimo uždavinys, o taip pat analizuojamos tik dvimatės daugdaros (parametro reikšmė $d = 2$), todėl nenagrinėjami daugdaros dimensiją nustatantys metodai, kurie aprašyti darbuose (Camastra and Vinciarelli 2002; Brand 2003; Costa and Hero 2004; Kegl 2005; Levina and Bickel 2005; Weinberger and Saul 2006) ir kt.

Svarbus su daugdara susijęs dalykas yra jos *topologija*, t. y. daugdaros visų atvirųjų poaibių rinkinys. Topologijos išlaikymas reiškia kaimynystės išlaikymą, t. y. transformuojant daugdarą į mažesnio matavimo erdvę, kiekvieno taško aplinka su joje esančiais kaimynais transformuojama į kitą aplinką taip, kad joje taško kaimynai liktų tie patys. Netiesinės daugdaros atpažinimo metodai yra topologiją išlaikantys metodai. Šių metodų tikslas – išlaikyti atstumus, atvaizduojant daugiamačius duomenis į mažesnės dimensijos erdvę, taip kad duomenų taškai, esantys arti pradinėje duomenų (didelės dimensijos) erdvėje (*input space*), būtų taip pat artimi ir projekcinėje (mažesnės dimensijos) erdvėje (*output space*), t. y. tokiu būdu siekiama išsaugoti kiekvieno taško kaimynus.

Per paskutinius dešimt metų sukurta daug netiesinės daugdaros atpažinimo metodų: lokaliai tiesinis vaizdavimas (*locally linear embedding*, LLE) (Roweis and Saul 2000; Saul and Roweis 2003), Laplaso matricos tikriniai žemėlapiai (*Laplacian eigenmaps*, LE) (Belkin and Niyogi 2002; Belkin and Niyogi 2003), Hesės matricos tikriniai žemėlapiai (*Hessian LLE*, HLLE) (Donoho and Grimes 2003), lokaliųjų liečiamųjų erdvių rikiavimas (*local tangent space alignment*, LTSA) (Zhang and Zha 2004), izometrinis požymių vaizdavimas (*Isometric feature mapping*, ISOMAP) (Tenenbaum *et al*. 2000) ir kiti (Roweis *et al*. 2002; Brand 2003; Teh and Roweis 2003; Verbeek *et al*. 2004; Weinberger *et al*. 2005; Sha and Saul 2005; Weinberger and Saul 2006; Dollar *et al*. 2007; Weinberger *et al*. 2007; Lee and Verleysen 2007). Tuo tarpu, kai LLE, LE, HLLE, LTSA stengiasi išlaikyti daugdaros lokalią struktūrą, ISOMAP metodo tikslas išsaugoti tiek lokalią, tiek globalią daugdaros struktūrą.

Praktikoje netiesinės daugdaros atpažinimo metodai ypač taikomi vaizdų apdorojime (*image processing*). Paveikslas (nuotrauka) yra skaitmenizuojamas, t. y. duomenų taško koordinatės sudarytos iš paveikslėlio ar nuotraukos taškų spalvinių savybių, ir todėl šio taško koordinačių skaičius yra labai didelis (Law

and Jain 2006). Dažnai duomenis sudaro to paties objekto paveikslėliai, gauti palaipsniui pasukant objektą tam tikru kampu (2.19 pav.) arba nufotografuojant objektą skirtingais momentais (2.20 pav.) ir t. t. Tokiu būdu, gretimi taškai nedaug skiriasi vienas nuo kito, o visas taškų rinkinys aproksimuoja tam tikrą daugdarą. Detalūs pavyzdžiai pateikti straipsniuose (Tenenbaum *et al*. 2000; Kouropteva *et al*. 2002; Saul and Roweis 2003) (veidų analizė) ir straipsnyje (Karbauskaitė *et al*. 2007) (objekto, pasukto skirtingais kampais, paveikslėlių palyginimas). Netiesinės daugdaros atpažinimo metodų tikslas – suprasti ir išanalizuoti šiuos paveikslėlius jų kintamumo požiūriu, pavyzdžiui, kaip keičiasi žmogaus pozicija, veido išraiška ar to paties objekto pasukimas (Weinberger and Saul 2006). Tai naudinga identifikuojant objekto nežinomą poziciją, turint to objekto nuotraukų įvairiose pozicijose rinkinius.



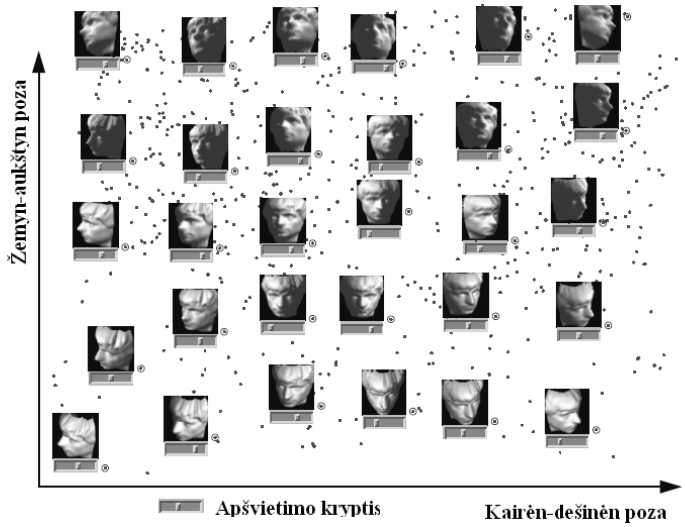
**2.19 pav.** Ančiuko, pasukto skirtingais kampais, paveikslėliai (http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php)

**2.20 pav.** Golfą žaidžiančio žmogaus nuotraukos (http://www.golfswingphotos.com)

Paprastai paveikslėliai ar nuotraukos yra charakterizuojami daug mažesniu laisvės laipsnių skaičiumi (*degrees of freedom*) nei tikruoju taško, nusakančio paveikslėlį ar nuotrauką, koordinačių skaičiumi *n*. Jei duomenų aibės paveikslėliai yra efektyviai charakterizuojami mažu skaičiumi kintamųjų, tuomet šiuos paveikslėlius atitinkantys taškai yra išsidėstę ant arba arti mažesnės

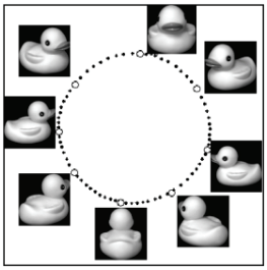
dimensijos daugdaros, įdėtos didelės dimensijos erdvėje (Weinberger and Saul 2006). Besikeičiantis apšvietimas, veido išraiška, orientacija ir kt. gali būti laikomi vidiniais kintamaisiais, kuriais nusakomi taškai aproksimuoja netiesinę veido daugdarą pradinių duomenų erdvėje (Zhang *et al*. 2004).

Panagrinėkime pavyzdį, pateiktą darbe (Tenenbaum *et al*. 2000). Tegu pradinius duomenis sudaro 698 nesurūšiuoti žmogaus veido paveikslėliai, besiskiriantys veido poza ir apšvietimu. Nors pradinių duomenų dimensija gana didelė ( $n = 4096$ , nes kiekvienas paveikslėlis sudarytas iš $64 \times 64$ nespalvotų taškų), tačiau reikšminga šių duomenų struktūra turi daug mažiau nepriklausomų laisvės laipsnių (šiuo atveju 3). Šios 4096-matės erdvės viduje visi paveikslėlius atitinkantys taškai yra išsidėstę ant trimatės daugdaros, kitaip tariant duomenų taškai sudaro paviršių, kuris charakterizuojamas trimis kintamaisiais: dviejų pozų ir apšvietimo kampu. Tikslas – turint nesurūšiuotus pradinius daugiamatės erdvės taškus, rasti jų projekcijas mažesnės dimensijos erdvėje, tokioje kad taškų projekcijų koordinačių skaičius būtų lygus duomenų aibės vidiniam laisvės laipsnių skaičiui, t. y. šiuo atveju randamos projekcijos, turinčios tris koordinates. 2.21 paveiksle parodytos taškų projekcijos plokštumoje, kur didesni skrituliukai nurodo atitinkamus paveikslėlius, o po paveikslėlių esančios horizontalios slinkties juostos žymi trečią koordinatę. Kiekviena koordinačių ašis (kairėn-dešinėn poza, žemyn-aukštyn poza, apšvietimo kryptis) susijusi su vienu laisvės laipsniu.




**2.21 pav.** 698 žmogaus veido paveikslėlius atitinkančių 4096-matės erdvės taškų, išsidėsčiusių ant trimatės daugdaros, dvimatės projekcijos, gautos ISOMAP metodu, kai algoritmo parametras $k = 6$ (Tenenbaum *et al*. 2000)

Kai duomenų taškai atitinka paveikslėlius, gautus nufotografavus vaizdą ar objektą iš skirtingų kampų (arba skirtingoje pozoje) ir esant tam tikram apšvietimui, t. y. keičiasi žiūrėjimo kampas ir apšvietimo kryptis, tai tuomet šie taškai yra išsidėstę ant daugdaros, kurios dimensija yra 3 (2.21 pav.). Jei apšvietimas objekto neįtakoja, tai tuomet taškai priklausys dvimatei daugdarai. Tokia situacija gaunama analizuojant besisukančio ančiuko paveikslėlius (2.22 pav.).



**2.22 pav.** 72 besisukančio ančiuko paveikslėlius atitinkančių 16384-matės erdvės taškų, išsidėsčiusių ant dvimatės daugdaros, dvimatės projekcijos, gautos ISOMAP metodu, kai algoritmo valdymo parametras $k = 2$

Netiesinės daugdaros atpažinimo metodų praktinė vertė pademonstruota skirtinguose pritaikymuose, tokiuose kaip veido pozos atskleidimas (*face pose detection*) (Hadid *et al*. 2002; Li *et al*. 2001), veido atpažinimas (*face recognition*) (Yang 2002; Zhang *et al*. 2004), veido išraiškos analizė (*analysis of facial expressions*) (Elgammal and Lee 2004a; Chang *et al*. 2004), žmogaus judesių interpretavimas (*human motion data interpretation*) (Jenkins and Mataric 2004), eisenos analizė (*gait analysis*) (Elgammal and Lee 2004a; Elgammal and Lee 2004b), medienos sandaros analizė (*wood texture analysis*) (Niskanen and Silven 2003) bei medicininių duomenų analizė – magnetinio rezonanso būdu gautų vaizdų tyrimas (Varini *et al*. 2004).

Radioterapijoje trimatis taikinys (navikas) yra vaizduojamas tomografo vaizde suskaidant jį į vokselius (erdvinius pikselius, pvz. 80×80×80). Šis navikas gali būti švitinamas radio spinduliais iš įvarių švitinimo prietaisų, siekiant sukoncentruoti švitinimą į taikinį ir kuo mažiau paliesti aplinkinius audinius. Kiekvienas švitinimo įtaisas kompiuteryje pavaizduojamas plokščiu langu suskaidytu į spindulius (pvz., 80×80). Yra tiriamas kiekvieno spindulio poveikis kiekvienam vokseliui, todėl kintamųjų gaunasi labai daug (Pena *et al*. 2009; Shepard *et al*. 1999). Čia galimi ir netiesinės daugdaros atpažinimo uždaviniai.

### 2.3.1. Lokaliai tiesinis vaizdavimas

2000 metais Lawrence K. Saul (USA) ir Sam T. Roweis (UK) pasiūlė vieną iš pirmųjų netiesinės daugdaros atpažinimo metodų – lokaliai tiesinį vaizdavimą (*locally linear embedding*, LLE) (Roweis and Saul 2000; Saul and Roweis 2003). Nors pavadinime egzistuoja žodis „tiesinis“, tačiau šis metodas priskiriamas netiesinių duomenų dimensijos mažinimo metodų grupei. LLE metodo savybės: atvaizduojant daugiamačius duomenis į mažesnės dimensijos erdvę, išlaikomi kaimynystės ryšiai tik tarp artimiausių taškų; atskleidžiama netiesinės daugdaros globali struktūra – bendras taškų aibės išsidėstymas, įvertinant mažesnės dimensijos daugdaros egzistavimą; duomenys vienareikšmiškai atvaizduojami mažesnės dimensijos erdvėje.

LLE algoritmo schema pateikta 2.23 paveiksle. Algoritmą sudaro trys etapai:

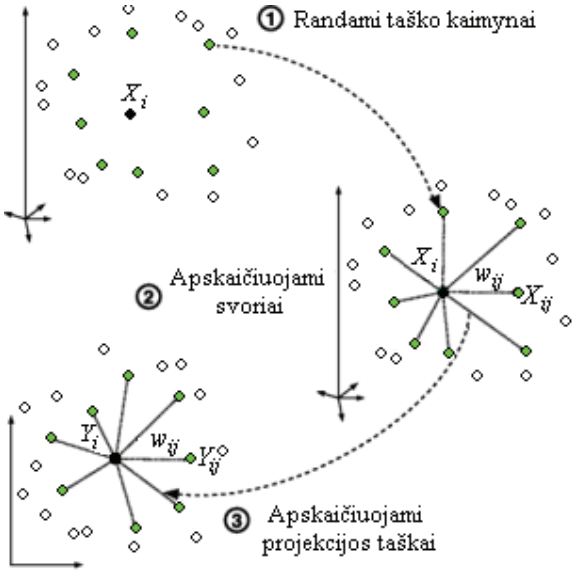
1. Randami kiekvieno vizualizuojamo duomenų taško $X_i$ kaimynai. Čia artimumo matas yra Euklido atstumas. Kaimyniniai taškai gali būti randami dvejopai: arba ieškoma nustatyto skaičiaus $k$ artimiausių kaimynų, arba ieškoma kaimynų iš tam tikro fiksuoto dydžio spindulio $\delta$ atvirojo rutulio, kurio centras yra taškas $X_i$. Darbe naudojamoje LLE algoritmo realizacijoje ieškoma $k$ artimiausių taškų $X_{ij}$, $j = \overline{1,k}$, kuriuos vadinsime taško $X_i$ $k$ artimiausiais kaimynais.
2. Kiekvienas daugiamatės erdvės taškas $X_i$ išreiškiamas jo artimiausių kaimynų tiesine kombinacija $\sum_{j=1}^{k} w_{ij} X_{ij}$. Čia neišvengiama paklaida (gaunami nauji, taškams $X_i$ gana artimi taškai $\overline{X}_i = \sum_{j=1}^{k} w_{ij} X_{ij}$), kurią būtina minimizuoti, randant optimalias svorių $w_{ij}$ reikšmes.
3. Fiksavus svorius $w_{ij}$, apskaičiuojami projekcijos taškai $Y_i$.

LLE metodo privalumai lyginant su MDS ir PCA metodais (Roweis and Saul 2000):

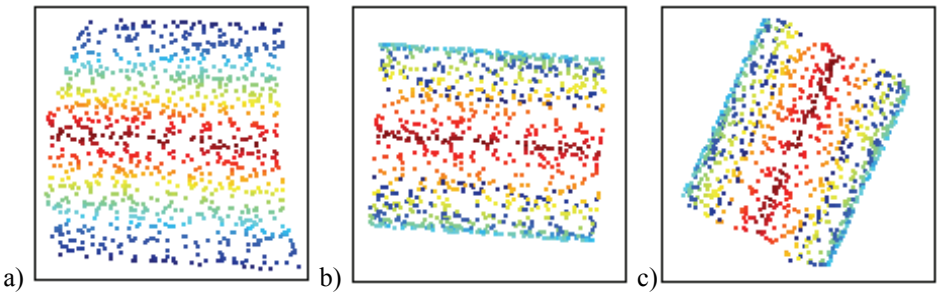
- MDS grupės metodais stengiamasi išlaikyti atstumus tarp visų duomenų aibės taškų. LLE nereikalauja išlaikyti atstumų tarp labiausiai nutolusių duomenų taškų, o tik tarp artimiausių kaimynų.
- Mažinant duomenų dimensiją LLE metodu, pasiseka identifikuoti daugdaros nežinomą struktūrą. Tuo tarpu įprastiniu MDS ir PCA

metodais tolimi daugiamatės erdvės taškai ant daugdaros atvaizduojami į artimus taškus plokštumoje, tokiu būdu suardoma daugdaros struktūra.

Tuo nesunku įsitikinti vizualizavus 1000 3-matės erdvės taškų, priklausančių netiesinei dvimatei S-formos daugdarai (2.1a pav.), LLE (2.24a pav.), PCA (2.24b pav.) ir MDS (2.24c pav.) metodais. Matome, kad LLE metodu daugdaros struktūra yra išlaikoma, S-formos daugdara tiesiog ištiesinama: tolimiausi taškai ant daugdaros (tamsiai mėlyna spalva) išlieka tolimiausi ir projekcijoje. Tačiau įprastiniu MDS ir PCA metodais gautose projekcijose tolimiausi taškai yra šviesiai mėlynos spalvos, o šie taškai nėra tolimiausi taškai ant daugdaros.




**2.23 pav.** Lokaliai tiesinio vaizdavimo algoritmo schema



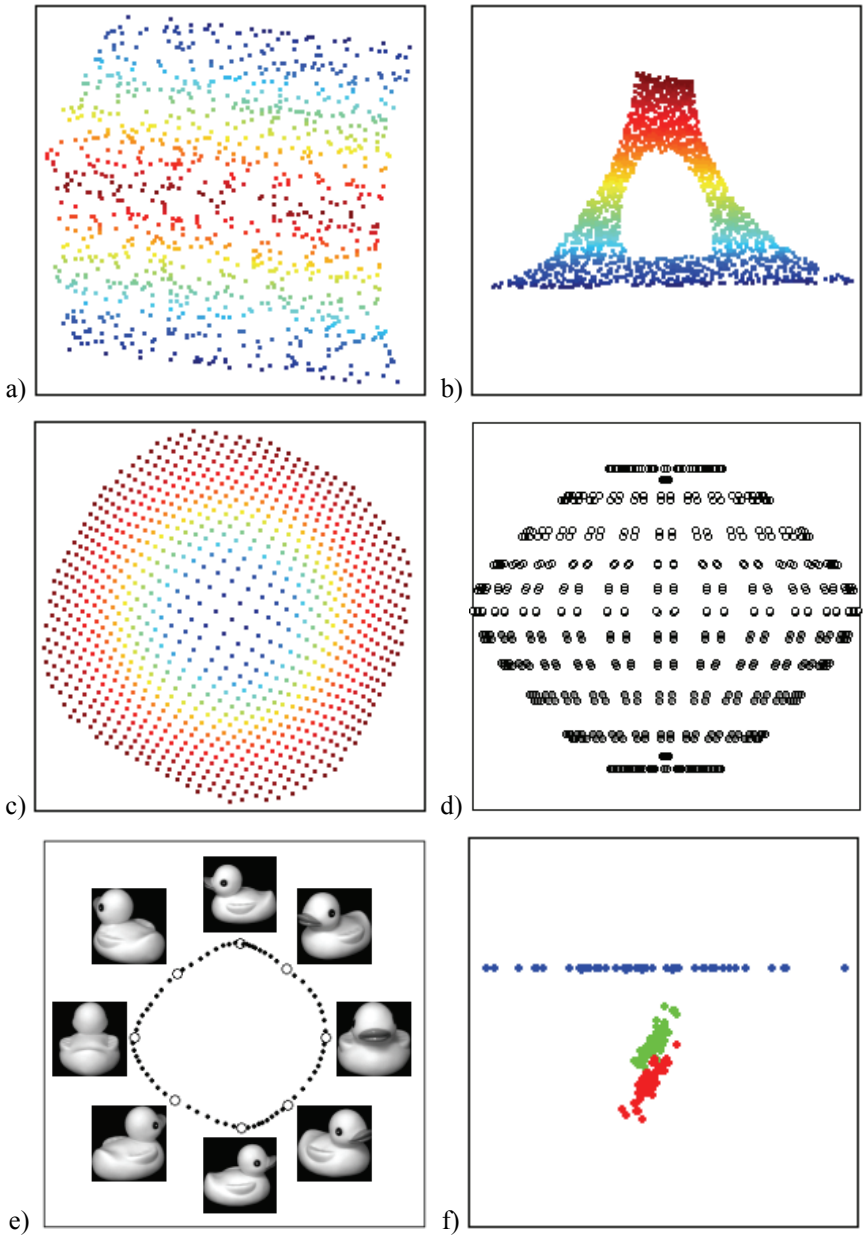

**2.24 pav.** a) projekcija gauta LLE, b) projekcija gauta PCA, c) projekcija gauta MDS

Lokaliai tiesinio vaizdavimo metodas yra labiau tinkamas vizualizuoti duomenis, kurie išsidėstę ant vientisos daugdaros (sudaro vientisą klasterį). Kai duomenų aibė sudaryta iš kelių visiškai atskirų duomenų klasterių, nėra gaunami geri vizualizavimo rezultatai (2.25f pav.). To priežastis yra tai, kad analizuojami duomenys nėra išsidėstę ant vientisos daugdaros.

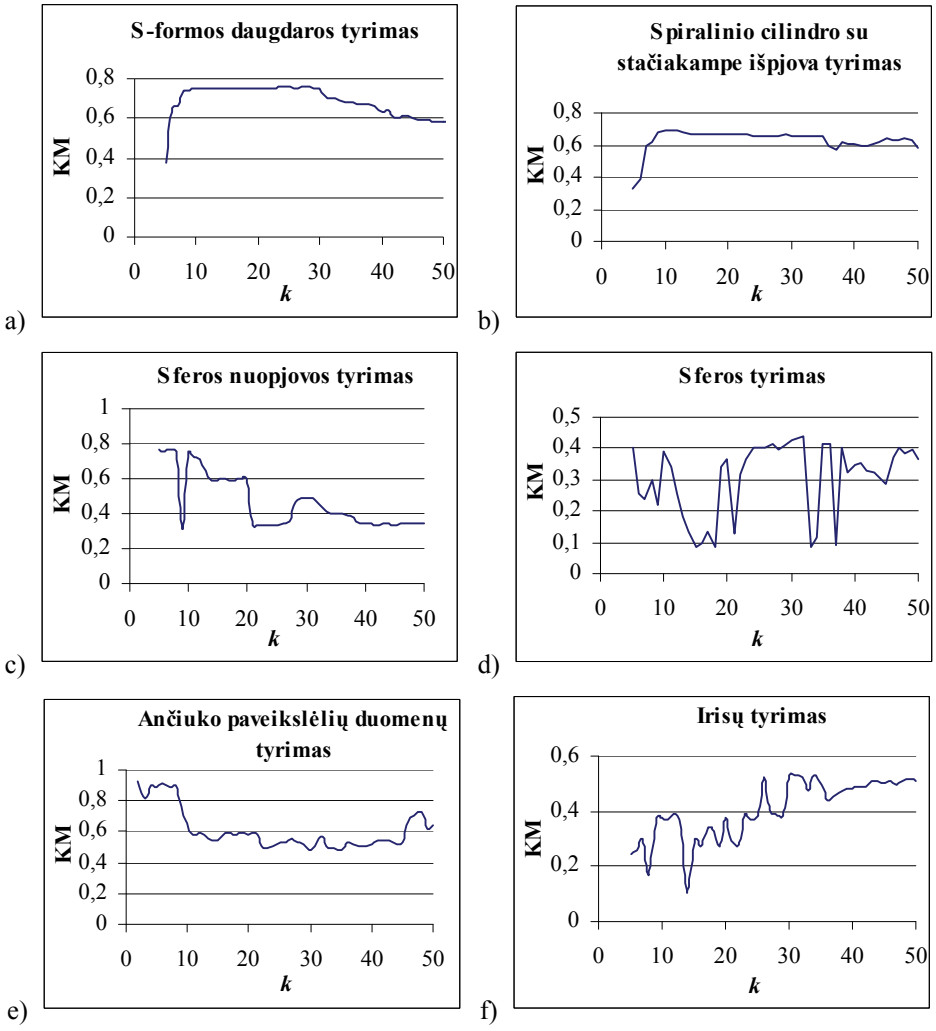
Sukurta daug LLE metodo modifikacijų, pavyzdžiui, toliau disertacijoje nagrinėjami Laplaso matricos tikrinių žemėlapių metodas (*Laplacian eigenmaps*, LE) (Belkin and Niyogi 2002; Belkin and Niyogi 2003), Hesės matricos tikrinių žemėlapių metodas (*Hessian LLE*, HLLE) (Donoho and Grimes 2003). Darbe (DeCoste 2001) ištirtas atstumų, pagrįstų Mercer branduolių funkcijomis (*Mercer kernels*), panaudojimas LLE algoritme ir pasiūlyta nauja branduolinė LLE forma, pavadinta KLLE (*kernelized LLE*). Taip pat sukurtos LLE algoritmo modifikacijos, paremtos mokymo su mokytoju principu (Ridder *et al*. 2003; Ridder *et al*. 2004). Darbuose (Bengio *et al*. 2004; Kouropteva *et al*. 2005) LLE algoritmas pritaikytas naujiems taškams atvaizduoti. Sukurti LLE ir kitų metodų, pavyzdžiui, PCA (Abusham *et al*. 2005), SOM (Xiao *et al*. 2005) junginiai.

Originalus LLE algoritmas ir jo modifikacijos plačiai taikomos veido atpažinimo srityje (Mekuz *et al*. 2005; Zhu and Zhu 2006; Zhao *et al*. 2005; Hadid *et al*. 2002). Lokaliai tiesinis vaizdavimas sėkmingai panaudotas vaizdais paremtoje veido animacijos sistemoje, aprašant burnos vaizdus (Liu *et al*. 2006). Eisenos atpažinimas – tai vienas iš pagrindinių būdų asmens autentiškumui patvirtinti iš labai didelio atstumo. Straipsniuose (Li and Li 2004; Li *et al*. 2005) parodyta, kad LLE metodas tinkamas analizuoti eiseną ilgą laiko tarpą ir dinaminius požymius (savybes) šiame eisenos laikotarpyje. LLE metodas taip pat panaudotas rankos gesto atpažinimui ir rankos judėjimo sekimui (Ge *et al*. 2008). Be to, LLE naudojamas ekonominiams duomenims vizualizuoti (Liou and Kuo 2002) bei ranka rašytiems skaitmenims atpažinti (Ridder *et al*. 2003). Straipsnyje (Jain and Saul 2004), LLE pritaikytas tiriant ir vizualizuojant kalbos ir garsų apdorojime gautas duomenų aibes. LLE metodas pritaikytas net medicininiams duomenims analizuoti – magnetinio rezonanso būdu gautiems vaizdams tirti. Rezultatai parodė, kad LLE metodas gali atskleisti piktybinių auglių įvairiarūšiškumą (Varini *et al*. 2004).

LLE metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant trimatės erdvės taškus, priklausančius netiesinėms dvimatėms daugdaroms, bei irisų duomenis, parodytas 2.25 paveiksle. Nuo LLE algoritmo parametro $k$ reikšmės labai priklauso duomenų vizualizavimo kokybė (žr. 5 skyrių). Todėl, prieš ieškant daugiamačių duomenų projekcijų (2.25 pav.), skaičiuotos daugdaros topologijos išlaikymo mato KM (žr. 7 skyrių) priklausomybės nuo LLE parametro $k$ ($k \in [5;50]$, tik e) atveju $k \in [2;50]$) (2.26 pav.). Jomis remiantis pasirinkta ir LLE algoritme panaudota tokia $k$ reikšmė, su kuria KM reikšmė didžiausia (geriausia).




**2.25 pav.** LLE metodu vizualizuoti duomenys: a) S-formos daugdaros taškai ($k = 10$), b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai ($k = 11$), c) sferos nuopjovos taškai ($k = 7$), d) sferos taškai ($k = 32$), e) ančiuko paveikslėlių duomenys ($k = 4$), f) irisų duomenys ($k = 34$)




**2.26 pav.** Daugdaros topologijos išlaikymo mato KM priklausomybės nuo LLE parametro *k*

### 2.3.2. Laplaso matricos tikrinių žemėlapių metodas

Laplaso matricos tikrinių žemėlapių metodas (*Laplacian Eigenmaps*, LE) (Belkin and Niyogi 2002; Belkin and Niyogi 2003) skirtas netiesiniams duomenų dimensijos mažinimo uždaviniams spręsti. Be to, šis metodas priklauso pastaruoju metu labai išvystytų netiesinės daugdaros atpažinimo metodų grupei. Kaip ir LLE, LE yra lokalus metodas (*local approach*), t. y. išlaiko daugdaros lokalią struktūrą. LE metodas remiasi grafų teorijos idėjomis. LE kuria duomenų aibės kaimynystės grafą, kuriuo atvaizduojama daugdaros lokali struktūra. LE yra pagrįstas atstumų tarp artimiausių taškų mažinimu. Be to, šis mažinimas yra su ribojimais, kad būtų išvengtas trivialus sprendinys, kai visi taškai yra atvaizduojami į vieną tašką (visi atstumai yra lygūs nuliui) (Lee and Verleysen 2007).

Kaip parodyta darbuose (Belkin and Niyogi 2003; Nadler *et al*. 2006), šis metodas yra glaudžiai susijęs su spektriniu klasterizavimu (Ng *et al*. 2002). Ryšys tarp LE ir spektrinio klasterizavimo parodo, kad LE metodas labai tinkamas duomenų klasterizavimui (Lee and Verleysen 2007). Vizualizavimo prasme tai privalumas, nes siekiant išsaugoti projekcijose lokalią duomenų struktūrą, algoritmas netiesiogiai pabrėžia ir natūralius klasterius duomenyse. Šis algoritmas palyginti nejautrus taškams atsiskyrėliams ir triukšmui (Belkin and Niyogi 2003).

Tarkime, kad vizualizuojamų duomenų aibę $X$ sudaro *m* *n*-matės erdvės taškų $X_i = (x_{i1},...,x_{in}), i = \overline{1,m}$ ($X_i \in R^n$), priklausančių glodžiajai netiesinei *d*-matei daugdarai, kuri įdėta erdvėje $R^n$.

*LE algoritmo schema:*

1. Iš visų aibės $X$ taškų $X_i, i = \overline{1,m}$ sukonstruojamas kaimynystės grafas (*Adjacency Graph*), turintis *m* viršūnių (kiekviena viršūnė atitinka vieną tašką) ir aibę briaunų, jungiančių kaimyninius taškus.
2. Parenkami grafo briaunų svoriai.
3. Randamos daugiamatės erdvės taškų $X_i, i = \overline{1,m}$ projekcijos $Y_i = (y_{i1},...,y_{id}), \ i = \overline{1,m}$ *d*-matėje erdvėje iš šio grafo Laplaso matricos tikrinių vektorių.

**Apibrėžimas.** *Laplaso matrica* (kartais dar vadinama įėjimo (*admittance*) arba *Kirchhoff* matrica) – tai grafo pavaizdavimui skirta matrica.

Tarkime, kad grafas $G = (V, E)$ ($V = \{V_1, V_2,..., V_m\}$ – viršūnių aibė, $E$ – briaunų aibė, $m$ – viršūnių skaičius) yra neorientuotas, nesvorinis grafas, neturintis kilpų ar kelių briaunų iš vienos viršūnės į kitą (nėra multigrafas).

Tuomet šio grafo Laplaso matrica yra $m \times m$ simetrinė matrica, kurioje kiekvienai viršūnei skiriama viena eilutė ir vienas stulpelis. Laplaso matrica $L = \left\{ l_{ij}, i, j = \overline{1,m} \right\}$ apibrėžiama tokiu būdu:

$$L = \widetilde{D} - A,$$

kur $\widetilde{D} = diag\left(d_1,...,d_m\right)$ yra diagonalioji matrica, suformuota iš grafo viršūnių laipsnių (*degree matrix*), o $A$ yra gretimumo matrica (*adjacency matrix*). Viršūnės $V_i$ laipsnis – tai skaičius viršūnių, gretimų viršūnei $V_i$; viršūnės yra gretimos, jei jos sujungtos briauna. Grafo $G = \left(V, E\right)$ gretimumo matrica – tai kvadratinė *m*-tosios eilės matrica $A = \left\{ a_{ij}, i, j = \overline{1,m} \right\}$, kurios elementas $a_{ij}$ apibrėžiamas taip:

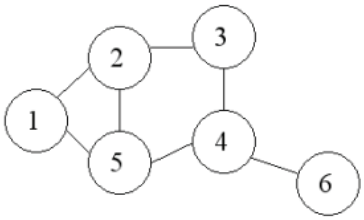
$$a_{ij} = \begin{cases} 1, \text{ jei } V_i \text{ ir } V_j \text{ yra gretimos,} \\ 0, \text{ priešingu atveju.} \end{cases}$$

Laplaso matricos elementai gali būti apskaičiuojami tiesiog pagal formulę:

$$l_{ij} = \begin{cases} d_i, \text{ jei } i = j, \\ -1, \text{ jei } i \neq j \text{ ir } V_i \text{ ir } V_j \text{ yra gretimos,} \\ 0, \text{ priešingu atveju.} \end{cases}$$

Pavyzdys:

Žymėtasis grafas




Laplaso matrica

$$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

LE algoritme Laplaso matrica *L* apskaičiuojama šiek tiek kitokiu būdu, nes iš visų aibės $X$ taškų sukonstruojamas svorinis kaimynystės grafas, todėl formulėje $L = \widetilde{D} - A$ gretimumo matrica $A$ pakeičiama grafo briaunų svorių matrica *W*, o $\widetilde{D}$ yra diagonalioji svorių matrica, kurios elementai yra matricos *W* stulpelių (arba eilučių, nes matrica *W* simetrinė) sumos.

Yra du būdai briaunų svoriams parinkti (Belkin and Niyogi 2003):

1. naudojant šiluminio branduolio (*Heat kernel*) parametrą $T$ ($T \in R$). Jei grafo viršūnės $V_i$ ir $V_j$ yra sujungtos, tai briaunos svoris apskaičiuojamas pagal formulę $w_{ij} = e^{-\|X_i - X_j\|^2 / T}$, priešingu atveju, $w_{ij} = 0$;
2. supaprastintas būdas, neturintis parametrų ($T = \infty$). Jei grafo viršūnės $V_i$ ir $V_j$ yra sujungtos briauna, tai $w_{ij} = 1$, priešingu atveju, $w_{ij} = 0$.

Naudojant antrąjį būdą, gaunamas atvejis, kai grafas yra nesvorinis ir jo Laplaso matrica apskaičiuojama pagal formulę $L = \widetilde{D} - A$.

LE metodas yra LLE metodo modifikacija. LE metodas turi nedaug valdymo parametrų: jei šiluminio branduolio (*Heat kernel*) parametro *T* reikšmę pasirinksime $\infty$, tai algoritme liks tik vienas parametras *k*, ($k \in N$) (artimiausių kaimynų skaičius) arba $\delta$ ($\delta \in R$) (spindulys atvirojo rutulio, į kurį patekę taškai bus laikomi kaimynais). Taigi šis algoritmas turi tik vieną parametrą. Šiuo atžvilgiu, LE metodas pranašesnis už LLE metodą, kuris turi du parametrus: artimiausių kaimynų skaičių $k$ ir lokalios Gramo matricos reguliarizacijos parametrą $\varepsilon$.

Skaičiuojamuoju požiūriu, LE metodas labai panašus į LLE metodą: skaičiuojama grafo Laplaso matrica *L* (LLE metode buvo skaičiuojama išretinta matrica $\widetilde{M}$), po to sprendžiamas tikrinių vektorių, atitinkančių mažiausias tikrines reikšmes, radimo uždavinys. Esminis skirtumas tarp šių metodų:

- LLE metode, radus taškų kaimynus, kiekvienam taškui skaičiuojama Gramo matrica, kuri naudojama **svoriams apskaičiuoti**. Iš gautos svorių matricos *W* sudaroma simetrinė, teigiamai pusiau apibrėžta, išretinta (*sparse*) matrica: $\widetilde{M} = (I - W)^T (I - W)$ ($I$ – vienetinė matrica) ir sprendžiamas tikrinių vektorių radimo uždavinys.
- LE metode sudaromas kaimynystės grafas (*Adjacency Graph*) ir jo briaunoms **svoriai yra parenkami**. Naudojant šiuos svorius, sudaroma simetrinė, teigiamai pusiau apibrėžta Laplaso matrica *L* ir sprendžiamas tikrinių vektorių radimo uždavinys.

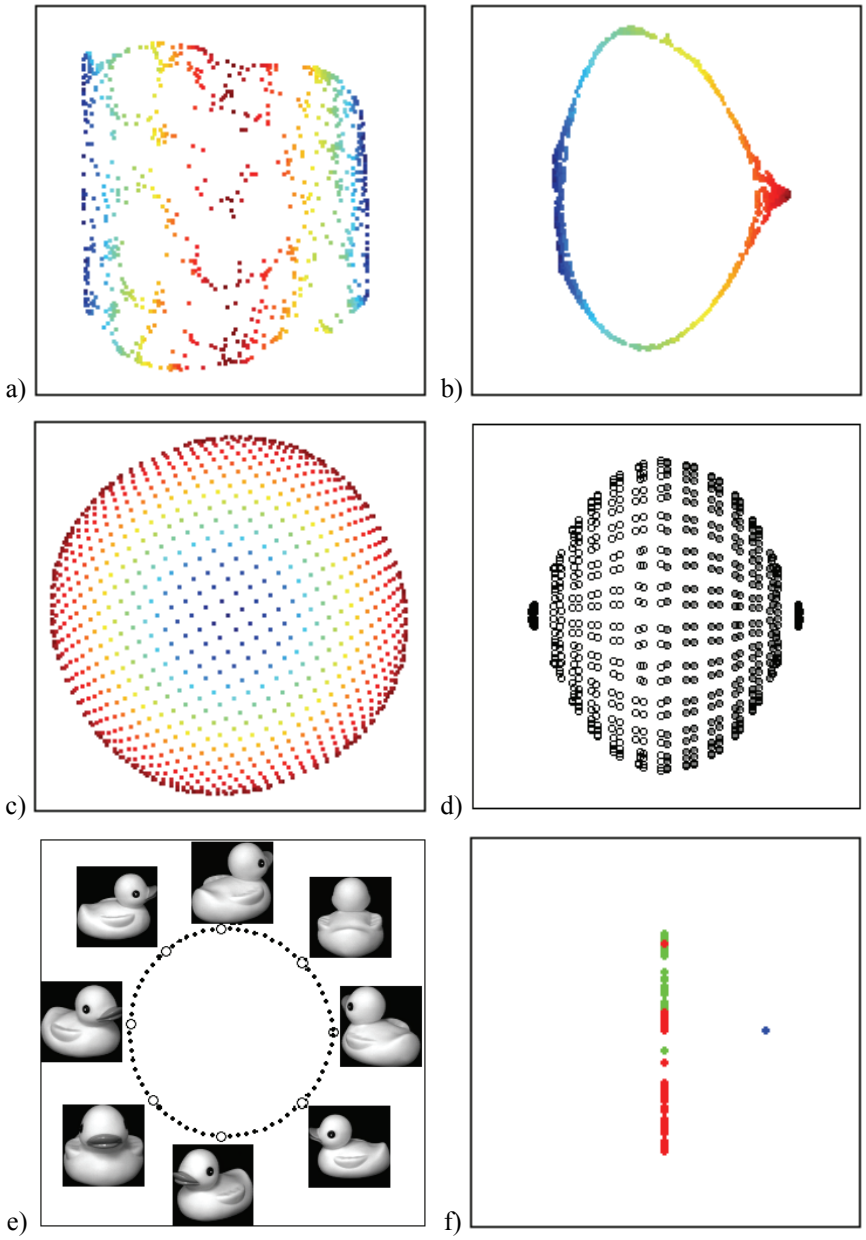
Yra nemažai LE metodo variantų. Išsklaidymo atvaizdavimai (*Diffusion maps,* DM) (Lafon and Lee 2006; Nadler *et al*. 2006), kaip ir LE metodas, naudoja šiluminio branduolio (*heat kernel*) funkciją ir turi analogišką algoritmą (sprendžiamas tikrinių vektorių ir tikrinių reikšmių radimo uždavinys). Verta paminėti, kad „vietą saugančios projekcijos“ metodas (*locality preserving projections*, LPP) (He and Niyogi 2004), dar žinomas kaip Laplaso veidai

(*Laplacianfaces*) (He *et al*. 2005), yra LE metodo tiesinis variantas. LPP veikia tuo pačiu principu kaip ir PCA, sudarant $d \times n$ transformacijų matricą, kurią galima panaudoti bet kuriam vektoriui iš aibės $R^n$. Šis metodas išlaiko daugelį PCA metodo privalumų, nors tikslo funkcija yra kitokia: tuo tarpu, kai PCA stengiasi išsaugoti duomenų aibės globalią struktūrą, LPP stengiasi išlaikyti lokalią struktūrą. Kaip ir LE, LPP metode panaudotas $k$ artimiausių kaimynų principas. Straipsnyje (Bengio *et al*. 2004) trumpai pateiktas LE metodo išplėtimas naujiems taškams atvaizduoti. LE metodas ir jo modifikacijos sėkmingai buvo pritaikytos klasterizavimo ir veido atpažinimo uždaviniams spręsti (He *et al*. 2005).

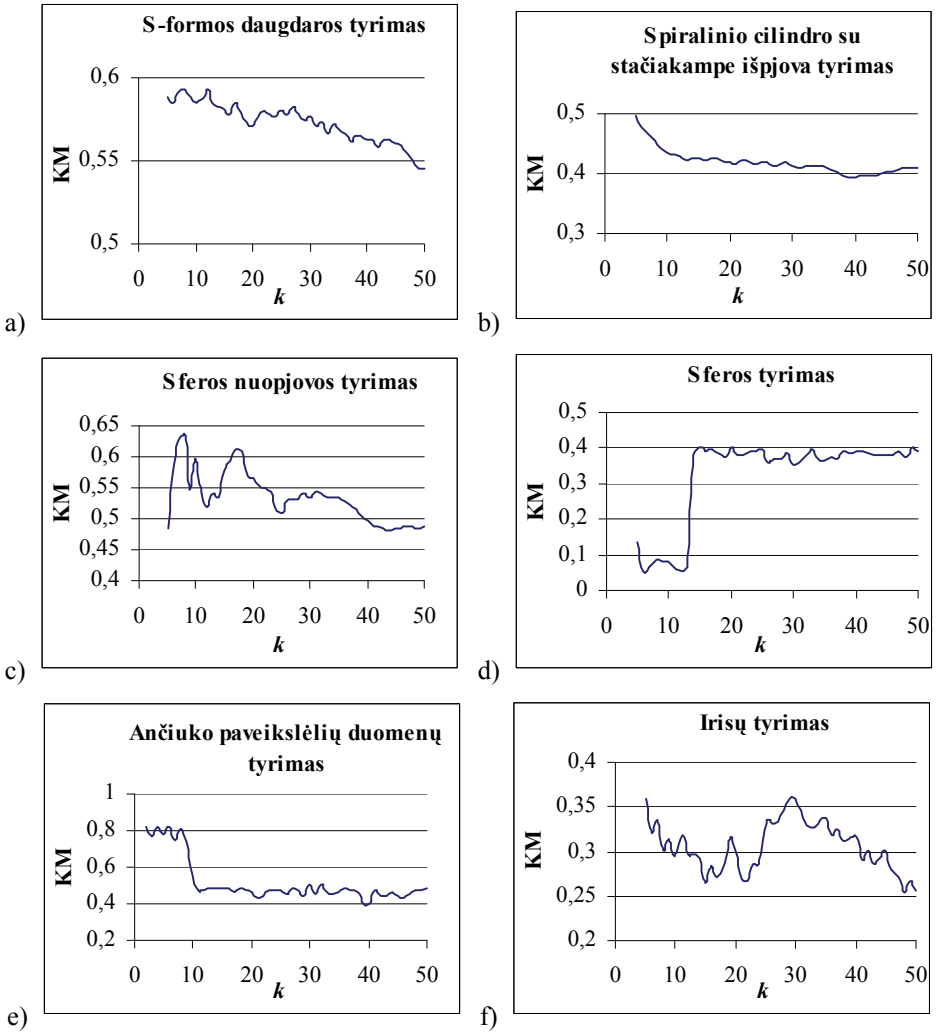
LE metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant netiesinių daugdarų taškus bei irisų duomenis, parodytas 2.27 paveiksle.

Nuo LE algoritmo parametro $k$ ($k$ – artimiausių kaimynų skaičius) reikšmės labai priklauso duomenų vizualizavimo kokybė. Todėl, prieš ieškant daugiamačių duomenų projekcijų (2.27 pav.), skaičiuotos daugdaros topologijos išlaikymo mato KM (žr. 7 skyrių) priklausomybės nuo LE parametro $k$ ($k \in [5; 50]$, tik e) atveju $k \in [2; 50]$) (2.28 pav.). Jomis remiantis pasirinkta ir LE algoritme panaudota tokia $k$ reikšmė, su kuria KM reikšmė didžiausia (geriausia).

Atlikti tyrimai parodė, kad LE metodas pateikia blogesnes daugiamatės erdvės taškų projekcijas plokštumoje nei LLE metodas. LE algoritmas turi tendenciją sumažinti atstumus tarp artimiausių taškų, tuo padidindamas atstumus tarp tolimesnių taškų. Dėl šios priežasties projekcijose dažnai matomos „skylės“ ir negalima matyti taškų išsidėstymo klasterių viduje (2.27a,b pav.). 2.27f paveiksle matome, jog pirmos klasės irisų duomenys atvaizduojami net į vieną tašką, kitų dviejų klasių duomenis atitinkantys taškai taip pat labai suspaudžiami.




**2.27 pav.** LE metodu vizualizuoti duomenys: a) S-formos daugdaros taškai ($k = 8$), b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai ($k = 5$), c) sferos nuopjovos taškai ($k = 8$), d) sferos taškai ($k = 15$), e) ančiuko paveikslėlių duomenys ($k = 2$), f) irisų duomenys ($k = 5$)




**2.28 pav.** Daugdaros topologijos išlaikymo mato KM priklausomybės nuo LE parametro *k*

### 2.3.3. Hesės matricos tikrinių žemėlapių metodas

Tam, kad lengviau būtų suprasti šį metodą, pateiksime keletą svarbių sąvokų paaiškinimų.

**Apibrėžimas.** Tarkime, kad turime funkciją $f: R^n \to R$, $f(x_1, x_2, ..., x_n)$. Jei egzistuoja šios funkcijos visos antros eilės dalinės išvestinės, tai funkcijos $f$ *Hesės matrica* yra tokia kvadratinė matrica:

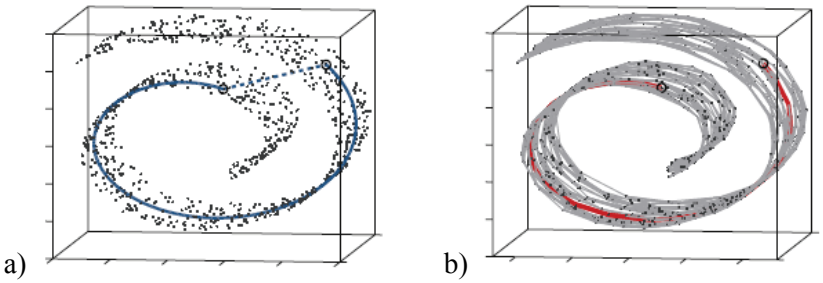
$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

**Apibrėžimas.** Erdvės $R^d$ *poaibis* $Y$ vadinamas *iškiliuoju* (*convex*), jeigu bet kuriems dviems jo taškams $Y_i$ ir $Y_j$ juos jungianti atkarpa irgi priklauso aibei $Y$.

**Apibrėžimas.** *Geodezinis*, arba *kreivinis*, *atstumas* tarp dviejų daugdaros taškų – tai trumpiausias lankas iš visų daugdaros kreivių lankų, jungiančių tuos taškus.

Kadangi nagrinėjame tik daugdaros taškų aibę, o ne pačią daugdarą, tai geodezinis atstumas tarp dviejų daugdaros taškų yra apytiksliai lygus trumpiausio kelio tarp tų taškų ilgiui einant daugdaros kreivu paviršiumi. Tuo tikslu iš visų daugdaros taškų sudaromas svorinis kaimynystės grafas ir trumpiausias kelias grafe randamas naudojant Floido (Floyd 1962) arba Dijkstros (Dijkstra 1959) algoritmus.

2.29a paveiksle pavaizduoti atstumai tarp dviejų spiralinio cilindro taškų: Euklido atstumas pažymėtas punktyrine, o geodezinis atstumas – ištisine linija. 2.29b paveiksle iš daugdaros taškų sukonstruotas kaimynystės grafas. Raudona kreivė žymi trumpiausio kelio grafe ilgį tarp dviejų pažymėtų taškų. Tai yra tikrojo geodezinio atstumo tarp nurodytų daugdaros taškų aproksimacija.




**2.29 pav.** a) Euklido (punktyrinė linija) ir geodezinis (ištisinė linija) atstumai tarp dviejų spiralinio cilindro taškų, b) trumpiausias kelias grafe tarp dviejų pažymėtų daugdaros taškų (raudona linija) (Tenenbaum *et al*. 2000)

**Apibrėžimas.** (Donoho and Grimes 2003) *Izometrija* – tai visų daugiamatės erdvės taškų, priklausančių daugdarai $M$, įdėtai Euklido erdvėje $R^n$, transformavimas į mažesnės dimensijos erdvę $R^d$ taip, kad geodeziniai atstumai tarp daugdaros taškų būtų lygūs Euklido atstumams tarp tų taškų projekcijų, t. y. $G(X_i, X_j) = \|Y_i - Y_j\|$, $\forall X_i \leftrightarrow Y_i, X_j \leftrightarrow Y_j$, kur $G(\cdot)$ apibrėžia geodezinį atstumą tarp daugdaros taškų $X_i \in R^n$, o $\|\cdot\|$ – Euklido atstumą tarp projekcijų $Y_i \in R^d$.

**Apibrėžimas.** (Donoho and Grimes 2003) *Lokalioji izometrija* (*local isometry*) – tai toks daugiamatės erdvės taškų, priklausančių daugdarai $M$, įdėtai Euklido erdvėje $R^n$, transformavimas į mažesnės dimensijos erdvę $R^d$, kai kiekvieno daugdarai priklausančio taško $X_i \in R^n$ pakankamai mažoje aplinkoje geodeziniai atstumai iki kaimyninių taškų $X_{ij} \in R^n$, $i = \overline{1,m}, j = \overline{1,k}$ yra lygūs Euklido atstumams tarp jų projekcijų $Y_i \in R^d$ ir $Y_{ij} \in R^d$, t. y. $G(X_i, X_{ij}) = \|Y_i - Y_{ij}\|$, $\forall X_i \in R^n \leftrightarrow Y_i \in R^d$, $\forall X_{ij} \in R^n \leftrightarrow Y_{ij} \in R^d$, $i = \overline{1,m}, j = \overline{1,k}$, kur $m$ – taškų skaičius, $k$ – artimiausių kaimynų skaičius.

Hesės matricos tikrinių žemėlapių metodas (*hessian-based locally linear embedding*, *Hessian eigenmaps, Hessian LLE,* HLLE) (Donoho and Grimes 2003) yra lokaliai tiesinio vaizdavimo (LLE) modifikacija, o teoriškai šis metodas pagrįstas Laplaso matricos tikrinių žemėlapių metodu (LE), tik kvadratinė Laplaso matrica pakeista kvadratine Hesės matrica.

HLLE metodu transformuojant (atvaizduojant) daugdarą, įdėtą labai didelio matavimo erdvėje, į mažesnės dimensijos erdvę, minimizuojamas daugdaros kreivumas taip, kad duomenų atvaizdavimas mažesnės dimensijos erdvėje būtų lokaliai izometrinis. Tai pasiekiama, atlikus Hesės matricos *H* tikrinių vektorių analizę. Hesės matrica aprašo daugdaros kreivumą apie duomenų taškus.

Tegu turime analizuojamų duomenų matricą

$$X = \{X_1, X_2, ..., X_m\} = \{x_{ij}, i = \overline{1,m}, j = \overline{1,n}\},$$

kurios *i*-oje eilutėje yra vektorius $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $X_i \in R^n$. Tegu šie duomenų taškai yra išsibarstę ant netiesinės *d*-matės daugdaros. HLLE metodu ieškosime šių duomenų transformacijų $Y_i = (y_{i1}, y_{i2}, ..., y_{id})$, $Y_i \in R^d$ mažesnės dimensijos erdvėje ($d < n$).

*HLLE algoritmą sudaro šie etapai (Donoho and Grimes 2003):*

1. Naudojant Euklido atstumą, randami kiekvieno duomenų taško $X_i \in R^n$ $i = \overline{1,m}$ $k$ artimiausi kaimynai.
2. Apibrėžiama lokaliosios liečiamosios erdvės kiekviename taške $X_i$ bazė ir randamos taško $X_i$ artimiausių kaimyninių taškų projekcijų toje lokalioje liečiamojoje erdvėje koordinatės.
3. Panaudojus gautas kaimyninių taškų koordinates, kiekviename daugdaros taške $X_i$ apskaičiuojama matrica $H_i$, kuri aproksimuoja Hesės matricą tame taške.
4. Iš lokalių matricų $H_i$ sudaroma matrica *H*, aproksimuojanti Hesės matricą visai daugdarai.
5. Randamos daugiamatės erdvės taškų $X_i \in R^n$ projekcijos $Y_i \in R^d$ mažesnės dimensijos erdvėje, atlikus matricos *H* tikrinių vektorių ir juos atitinkančių tikrinių reikšmių analizę. Ieškomi matricos *H* tikriniai vektoriai, atitinkantys *d* mažiausias, nelygias nuliui tikrines reikšmes. Šie vektoriai ir suformuoja daugiamatės erdvės taškų projekcijų mažesnės dimensijos erdvėje koordinates $Y_i \in R^d$.

HLLE metodo idėja pagrįsta lokaliąja izometrija: daugdara *M*, įdėta Euklido erdvėje $R^n$, yra lokaliai izometrinė atviram ir jungiam Euklido erdvės $R^d$ poaibiui *Y*. Kadangi ši projekcija neturi būti iškila, tai galima nagrinėti įvairesnių situacijų (pavyzdžiui, daugdaroje yra kiaurymių) negu originaliu ISOMAP

algoritmu (Tenenbaum *et al*. 2000). Originalus ISOMAP metodas nagrinėja tik vieną atvejį, kai daugdara *M* yra geodeziškai iškila, todėl ir projekcija yra iškila. Tačiau paprastai yra tyrinėjami realūs daugiamačiai duomenys, natūraliai sudarantys daugdarą, kuri dažnai nėra geodeziškai iškila. Todėl lokali izometrija HLLE algoritme yra tinkamesnė, tyrinėjant realius duomenis, nei globali izometrija ISOMAP algoritme (Donoho and Grimes 2003). Pagrindinis HLLE privalumas – šis metodas gali būti taikomas neiškiloms duomenų aibėms.

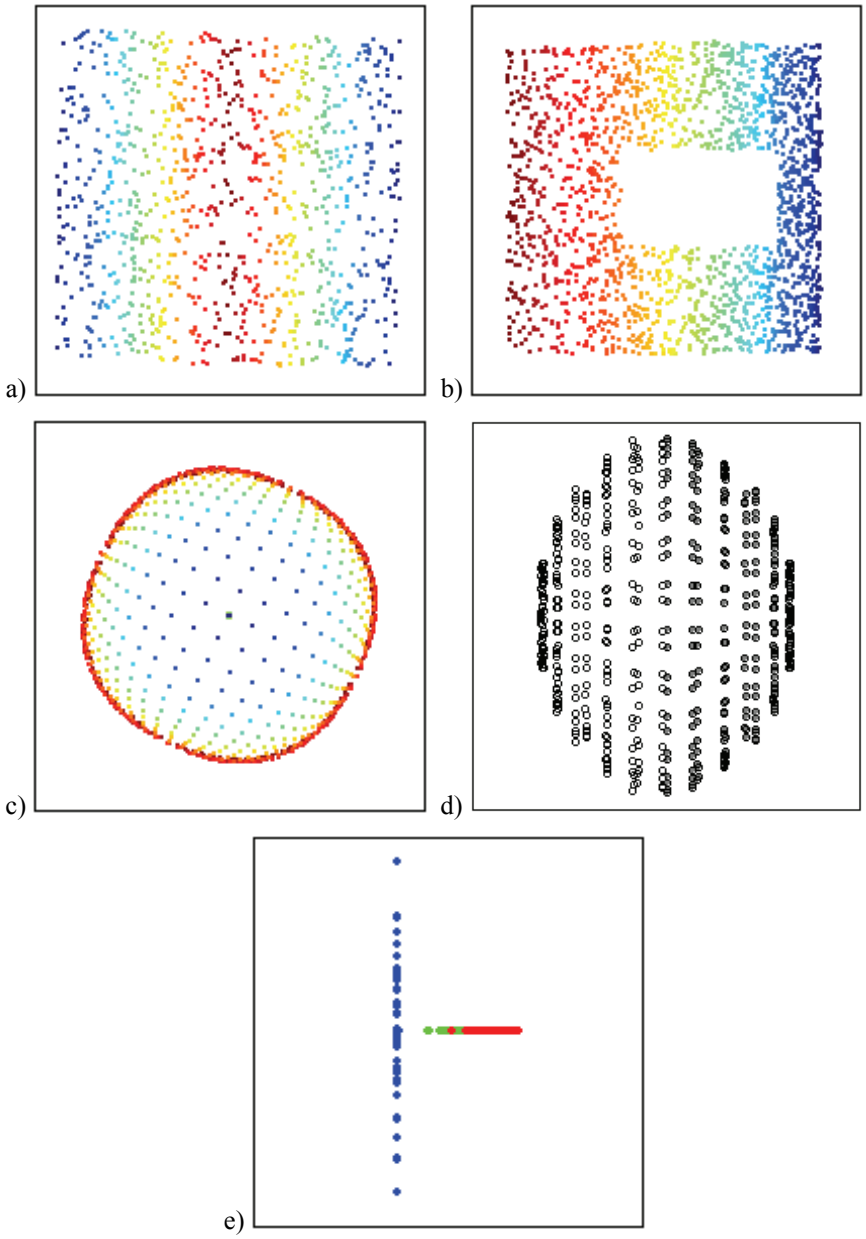
Naudojant HLLE metodą, reikia išspręsti *m* atskirų $k \times k$ tikrinių vektorių radimo uždavinių (*eigenproblem*) ir, panašiai kaip originaliame LLE algoritme, vieną $m \times m$ išretintą tikrinių vektorių radimo uždavinį. Kadangi pradiniai skaičiavimai atliekami tik mažose kaimynystėse, tai HLLE metodas išsprendžia didesnio duomenų skaičiaus *m* (net keliasdešimt tūkstančių) uždavinius nei LLE ar ISOMAP metodai.

HLLE metodo trūkumas lyginant su originaliu LLE, LE, LTSA ar ISOMAP metodais yra tai, kad HLLE metode reikia apskaičiuoti antros eilės išvestines, o tai padaryti sunku, kai pradinių duomenų dimensija labai didelė (Donoho and Grimes 2003). Todėl neįmanoma vizualizuoti daugiamatės erdvės taškus, pavyzdžiui, atitinkančius besisukančio ančiuko paveikslėlius (taškų koordinačių skaičius $n = 16384$).

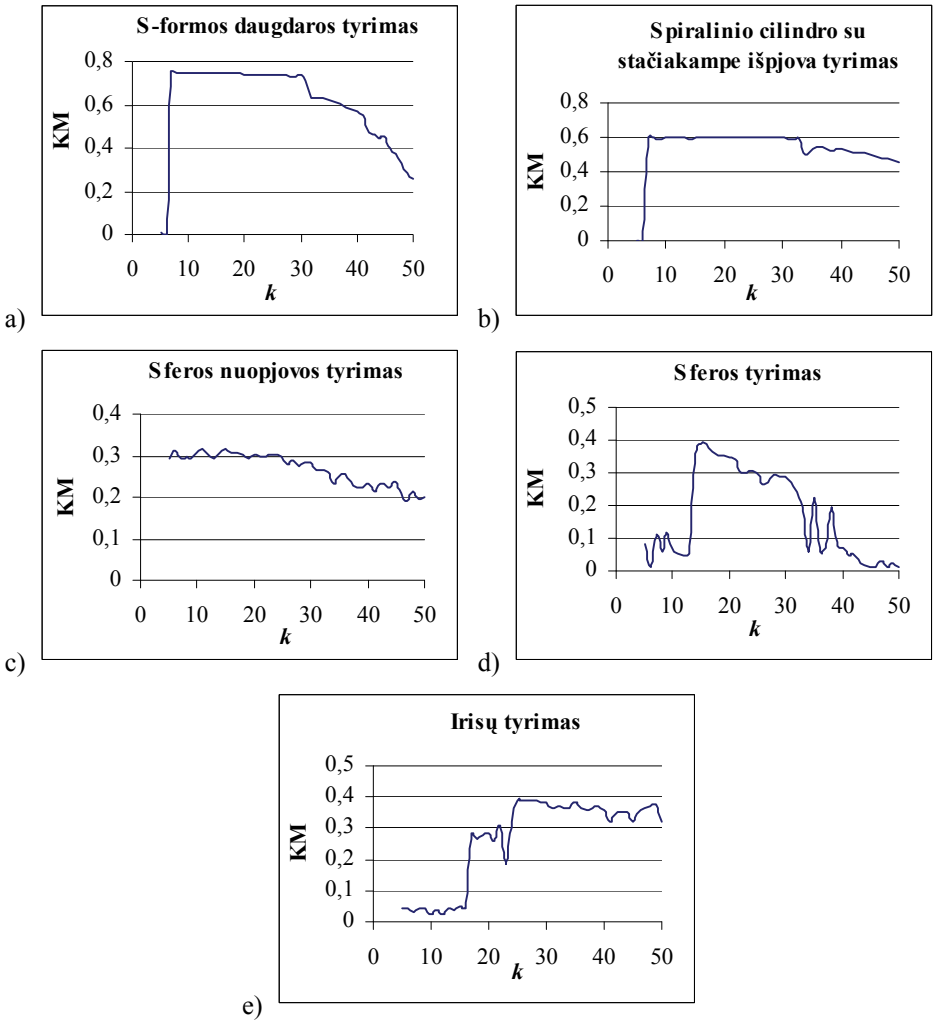
HLLE metodas sėkmingai pritaikytas jutiklio lokalizavimui (*sensor localization*) (Patwari and Hero 2004).

HLLE metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant netiesinių daugdarų taškus bei irisų duomenis, parodytas 2.30 paveiksle. HLLE algoritmo parametro *k* (*k* – artimiausių kaimynų skaičius) reikšmė labai įtakoja duomenų vizualizavimo kokybę. Todėl, prieš ieškant daugiamačių duomenų projekcijų (2.30 pav.), skaičiuotos daugdaros topologijos išlaikymo mato KM (žr. 7 skyrių) priklausomybės nuo HLLE parametro *k*, $k \in [5; 50]$ (2.31 pav.). Jomis remiantis pasirinkta ir HLLE algoritme panaudota tokia *k* reikšmė, su kuria KM reikšmė didžiausia (geriausia).

LLE, LE, HLLE ir ISOMAP metodų skirtumas akivaizdžiai matomas, vizualizavus spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškus (2.1b pav.). Kadangi ši daugdara geodeziškai neiškila ir yra lokaliai izometrinė į Euklido erdvę, tai LLE, LE ir ISOMAP metodais gaunamos deformuotos projekcijos: LLE metodu kai kurių simetriškų daugiamatės erdvės taškų projekcijos gaunamos nesimetrinės (stačiakampė išpjova deformuota) (2.25b pav.), o LE (2.27b pav.) ir ISOMAP (2.34b pav.) metodo atveju trūkstama sritis labai išplečiama, deformuojant aplink esančias projekcijas. HLLE metodu gautoje projekcijoje trūkstama sritis yra stačiakampė kaip ir pradiniuose duomenyse (2.30b pav.).




**2.30 pav.** HLLE metodu vizualizuoti duomenys: a) S-formos daugdaros taškai ($k = 10$), b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai ($k = 15$), c) sferos nuopjovos taškai ($k = 11$), d) sferos taškai ($k = 15$), e) irisų duomenys ($k = 25$)




**2.31 pav.** Daugdaros topologijos išlaikymo mato KM priklausomybės nuo HLLE parametro *k*

### 2.3.4. Lokaliųjų liečiamųjų erdvių rikiavimas

Lokaliųjų liečiamųjų erdvių rikiavimas (*local tangent space alignment*, LTSA) (Zhang and Zha 2004) – tai dar vienas netiesinės daugdaros atpažinimo metodas. Panašiai kaip ir HLLE metodas, šis metodas aprašo daugiamačių duomenų lokaliąsias savybes, naudojant kiekvieno duomenų taško lokaliąją liečiamąją erdvę. LTSA paremtas tuo, kad, jei tariama, jog daugdara yra lokaliai tiesinė, tai analizuojamas daugiamatės erdvės taškas tiesiškai atvaizduojamas į jo lokaliąją liečiamąją erdvę, o taip pat ir atitinkamas ieškomas mažesnės dimensijos erdvės taškas tiesiškai atvaizduojamas į tą pačią lokaliąją liečiamąją erdvę. LTSA metodas tuo pat metu ieško mažesnės dimensijos erdvės taškų koordinačių (projekcijų) ir mažesnės dimensijos erdvės taškų tiesinių atvaizdavimų į pradinių daugiamatės erdvės taškų lokaliąją liečiamąją erdvę.

Kaip ir LLE, LE, HLLE ir ISOMAP algoritmuose, LTSA algoritme yra valdymo parametras $k$ – artimiausių kaimynų skaičius. Taigi LTSA metodas, kaip ir prieš tai nagrinėti netiesinės daugdaros atpažinimo metodai, pagrįstas kaimyniškumo išsaugojimo principu.

Tegu turime analizuojamų duomenų matricą

$$X = \{X_1, X_2, ..., X_m\} = \{x_{ij}, i = \overline{1,m}, j = \overline{1,n}\},$$

kurios $i$-oje eilutėje yra vektorius $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $X_i \in R^n$. Tegu šie duomenų taškai išsidėstę ant arba arti netiesinės $d$-matės daugdaros. LTSA metodu ieškosime šių duomenų transformacijų $Y_i = (y_{i1}, y_{i2}, ..., y_{id})$, $Y_i \in R^d$ mažesnės dimensijos erdvėje ($d < n$).

*LTSA algoritmą sudaro šie etapai (Zhang and Zha 2004)*:

1. Kiekvienam duomenų taškui $X_i$, $i = \overline{1,m}$ a) randami $k$ artimiausi kaimynai; b) šiuos kaimyninius taškus panaudojus koreliacinei matricai gauti, surandami tos matricos $d$ tikriniai vektoriai $f_1, f_2, ..., f_d$, atitinkantys $d$ didžiausias tikrines reikšmes, ir sudaroma matrica $F_i = \left(1_k / \sqrt{k}, f_1, ..., f_d\right)$, čia $1_k$ – $k$-matis vektorius-stulpelis, kurio komponentės lygios 1.

2. Sudaroma rikiavimo matrica $B$. Matricos $B = \{b_{ij}, i, j = \overline{1,m}\}$ elementai gaunami, atliekant iteracinį sumavimą (visoms matricoms $F_i = \left(1_k / \sqrt{k}, f_1, ..., f_d\right)$ (matricoje $F_i$ stulpelis žymi vektorių) ir pradedant nuo $B = 0$): $B(K_i, K_i) \leftarrow B(K_i, K_i) + I_k - F_i F_i^T$, $i = \overline{1,m}$, kur

kaimynystės aibė, t. y. vektorius $K_i$ atvaizduoja duomenų taško $X_i$ $k$ artimiausių kaimynų indeksus, $I_k$ – vienetinė $k$ dydžio matrica.

3. Randami projekcijos taškai $Y_i \in R^d$. Ieškomi matricos $B$ tikriniai vektoriai, atitinkantys $d$ mažiausias, nelygias nuliui tikrines reikšmes. Šie vektoriai ir suformuoja daugiamatės erdvės taškų projekcijų mažesnės dimensijos erdvėje koordinates $Y_i \in R^d$.

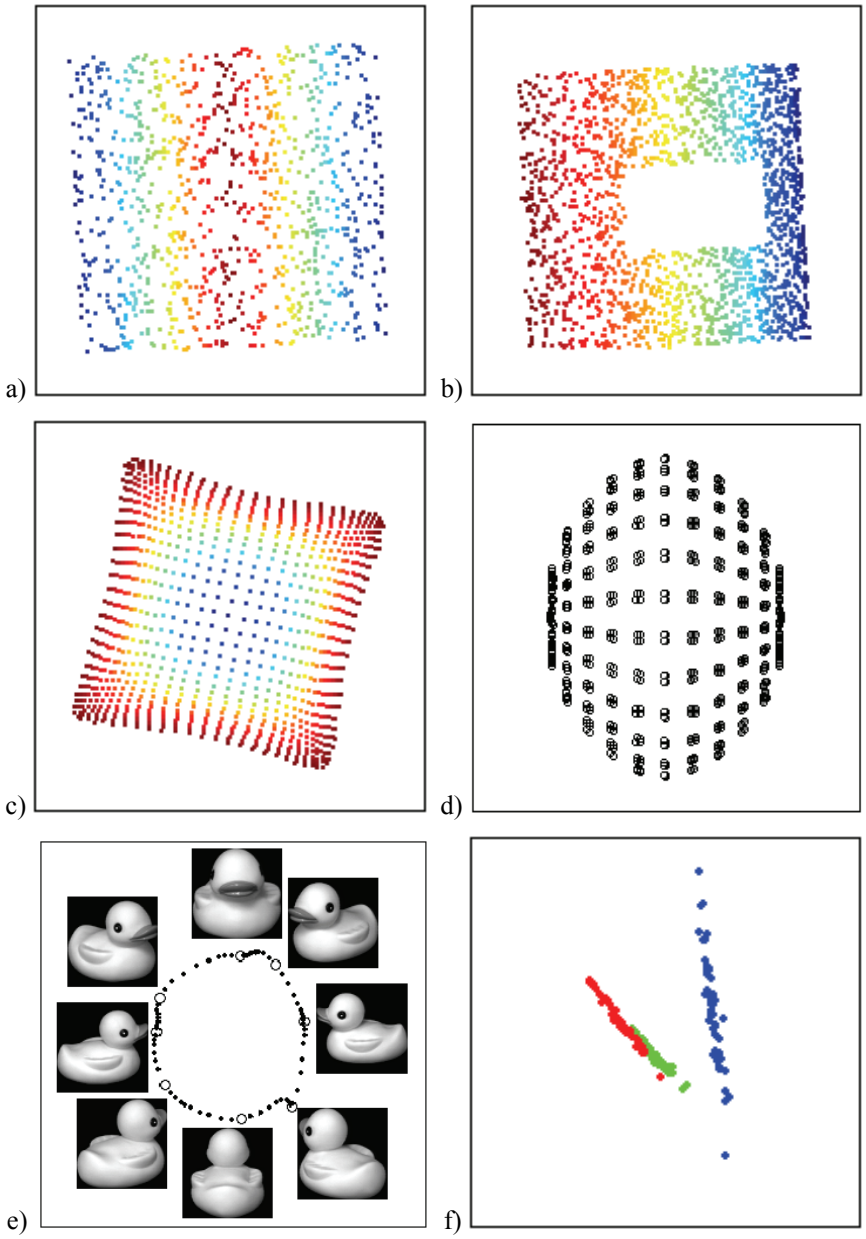
Darbe (Teng *et al*. 2005) LTSA metodas sėkmingai pritaikytas genetiniams duomenims analizuoti.

LTSA metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant taškus, priklausančius netiesinėms dvimatėms daugdaroms, bei irisų duomenis, parodytas 2.32 paveiksle.

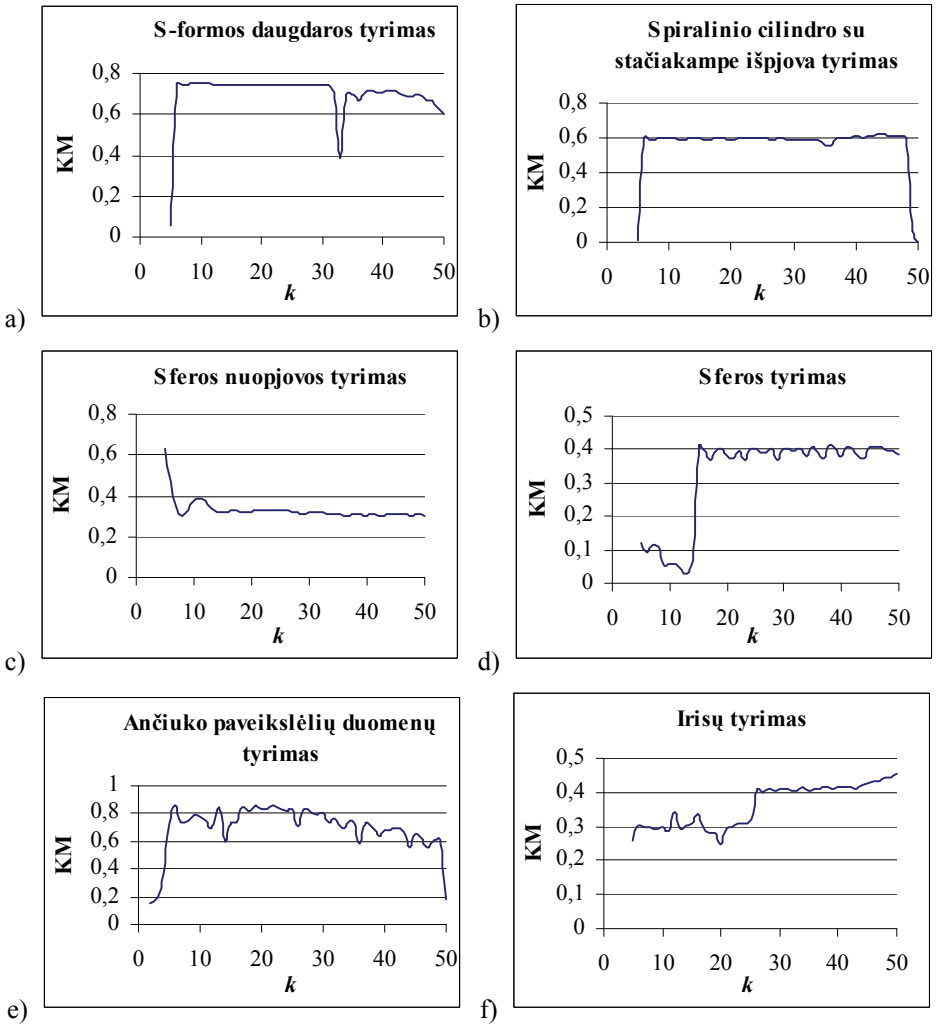
LTSA algoritmo parametro $k$ ($k$ – artimiausių kaimynų skaičius) reikšmės parinkimas lemia duomenų vizualizavimo kokybę. Todėl, prieš ieškant daugiamačių duomenų projekcijų (2.32 pav.), skaičiuotos daugdaros topologijos išlaikymo mato KM (žr. 7 skyrių) priklausomybės nuo LTSA parametro $k$ ($k \in [5; 50]$, tik e) atveju $k \in [2; 50]$) (2.33 pav.). Jomis remiantis pasirinkta ir LTSA algoritme panaudota tokia $k$ reikšmė, su kuria KM reikšmė didžiausia (geriausia).

## 2.3.5. ISOMAP metodas

Izometrinis požymių vaizdavimas (*Isometric feature mapping*, ISOMAP) (Tenenbaum *et al*. 2000) paprastai priskiriamas prie netiesinės daugdaros atpažinimo metodų, išlaikančių ne tik lokalią, bet ir globalią daugdaros struktūrą. Tačiau ISOMAP metodą galima laikyti ir daugiamačių skalių (MDS) grupės metodu, kuris skirtas duomenų dimensijai mažinti, o taip pat ir daugiamačiams duomenims vizualizuoti. Šis metodas skiriasi nuo įprastinio MDS metodo tuo, jog kitaip yra apibrėžiamas atstumų tarp analizuojamų duomenų matas. Įprastiniuose MDS metoduose paprastai tarp daugiamatės erdvės taškų yra skaičiuojami Euklido atstumai. Tokiu būdu neatsižvelgiama į daugdaros formą, todėl susiduriama su sunkumais atvaizduojant netiesines duomenų struktūras, tokias kaip spiralinį cilindrą ar pan. Taikant ISOMAP metodą, daroma prielaida, kad pradinėje erdvėje analizuojamus duomenis atitinkantys taškai yra išsidėstę ant mažesnės dimensijos netiesinės daugdaros, todėl skaičiavimuose naudojami geodeziniai atstumai.




**2.32 pav.** LTSA metodu vizualizuoti duomenys: a) S-formos daugdaros taškai ($k = 10$), b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai ($k = 12$), c) sferos nuopjovos taškai ($k = 5$), d) sferos taškai ($k = 15$), e) ančiuko paveikslėlių duomenys ($k = 6$), f) irisų duomenys ($k = 50$)




**2.33 pav.** Daugdaros topologijos išlaikymo mato KM priklausomybės nuo LTSA parametro *k*

Siekiant apskaičiuoti geodezinius atstumus tarp duomenų taškų $X_1, X_2, ..., X_m$, pirmiausia reikia nustatyti kiekvieno taško $X_i$, $i = \overline{1,m}$ kaimynus. Čia artimumo matas yra Euklido atstumas. Kaimyniniai taškai gali būti randami dvejopai: arba ieškoma nustatyto skaičiaus *k* artimiausių kaimynų, arba ieškoma kaimynų iš tam tikro fiksuoto dydžio spindulio $\delta$ atvirojo rutulio, kurio centras yra taškas $X_i$. Tuomet sudaromas svorinis kaimynystės grafas, kuris jungia kiekvieną tašką $X_i$, $i = \overline{1,m}$ su visais jo kaimynais. Grafo briaunų svoriai yra Euklido atstumai tarp taško $X_i$ ir jo kaimynų. Trumpiausio kelio grafe radimui naudojami Floido (Floyd 1962) arba Dijkstros (Dijkstra 1959) algoritmai. Floido algoritmas labiau tinkamas, kai analizuojamos duomenų aibės yra nedidelės (taškų skaičius nedidesnis nei 1000). Analizuojant didžiules duomenų aibes ir siekiant sutrumpinti skaičiavimo laiką, rekomenduojama naudoti Dijkstros algoritmą, kuris naudoja kaimynystės grafo išretintą struktūrą (Euklido atstumų matricoje saugomi atstumai tik tarp kaimyninių taškų) skaičiuojant trumpiausio kelio atstumus. Dijkstros algoritmas veikia ypač greitai net ir su mažomis duomenų aibėmis, jei jame naudojamos Fibonači medžio duomenų struktūros (*Fibonacci heap data structures*) (Tenenbaum *et al*. 2000). Naudojantis vienu iš šių algoritmų, randami trumpiausių kelių ilgiai tarp visų taškų porų. Šie ilgiai ir yra geodezinių atstumų tarp taškų porų įverčiai. Tokiu būdu suformuojama $m \times m$ geodezinių atstumų matrica, kurią galima analizuoti bet kuriuo daugiamačių skalių metodu.

Taikant ISOMAP metodą, ieškoma tokios transformacijos į vaizdo erdvę, kur būtų geriausiai išlaikomi geodeziniai atstumai tarp vizualizuojamų objektų (taškų).

*ISOMAP algoritmą sudaro trys etapai:*

1. Daugiamatėje erdvėje randami kiekvieno taško kaimynai.
2. Skaičiuojami geodeziniai atstumai tarp visų taškų porų.
3. Daugiamačių skalių metodu randamos daugiamatės erdvės taškų projekcijos mažesnės dimensijos – vaizdo erdvėje.

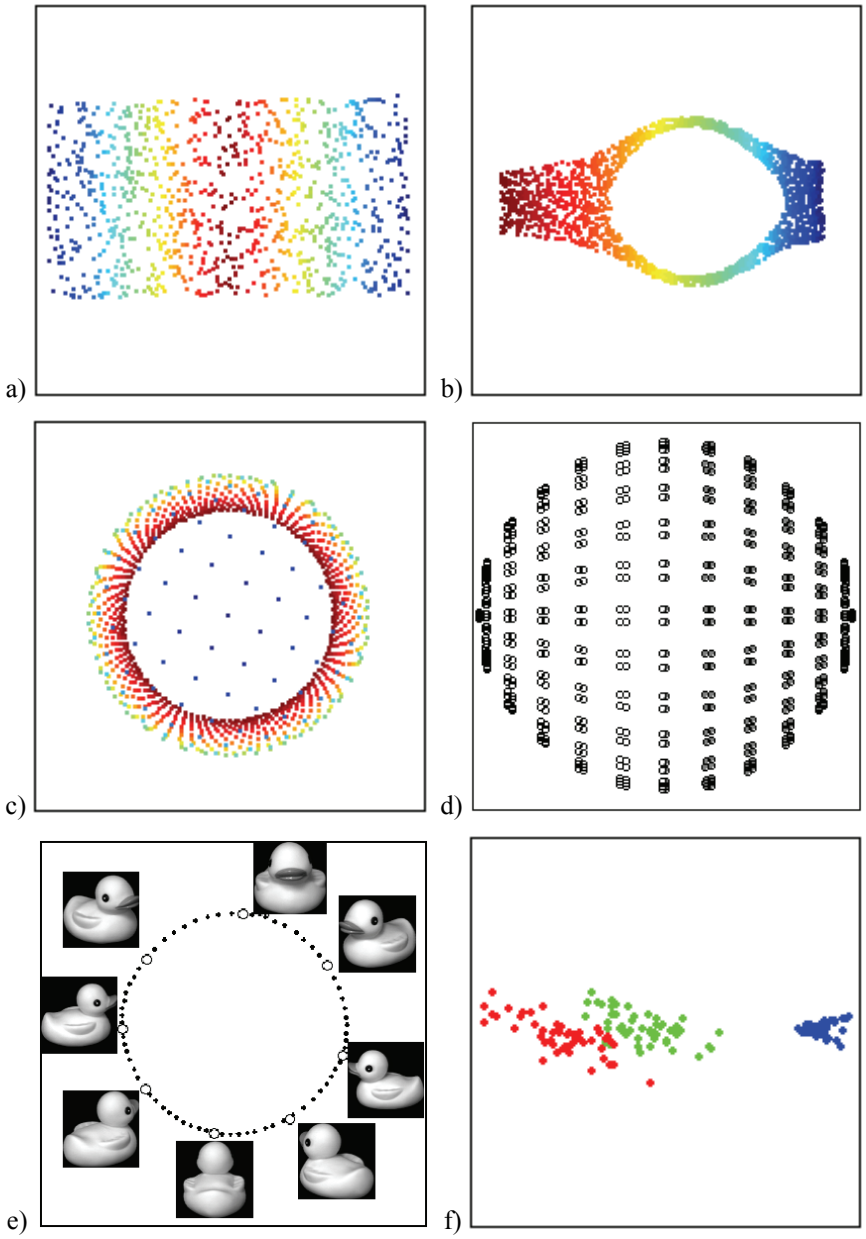
Pasak V.De Silva ir J.B.Tenenbaum (2003), turint labai dideles duomenų aibes, nepraktiška saugoti kompiuterio atmintyje pilną $m \times m$ ($m$ – taškų skaičius) atstumų matricą. Daugeliu atveju, kai duomenys yra išsidėstę ant mažesnės dimensijos daugdaros, antrame algoritmo žingsnyje apskaičiuoti atstumai yra „smarkiai pertekliški“ (*heavily redundant*) ir dauguma jų gali būti ignoruojami, nepadarius esminės įtakos gautoms projekcijoms. Taigi autoriai siūlo algoritmo modifikaciją (Landmark ISOMAP), kurioje, užuot išlaikius geodezinius atstumus tarp visų duomenų taškų, pasirenkamas duomenų taškų

poaibis (*landmark points*) ir išlaikomi atstumai nuo šių taškų iki visų kitų taškų. Kadangi gaunama stačiakampė geodezinių atstumų matrica, tai taikoma modifikuota MDS realizacija daugiamatės erdvės taškų projekcijoms rasti (Silva and Tenenbaum 2003a). Darbe (Silva and Tenenbaum 2003b) pasiūlytas metodas, kuriame koreguojami grafo atstumai, atsižvelgiant į duomenų struktūrą ir tankį, t. y. tankios duomenų sritys praretinamos, o retos labiau suspaudžiamos. Darbe (Yang 2004) pasiūlyta, realizuojant ISOMAP metodą, vietoj įprastinio MDS metodo naudoti Sammono algoritmą. Taip pat sukurta ISOMAP metodo modifikacija naujiems duomenų taškams atvaizduoti (Bengio *et al*. 2004; Law and Jain 2006).

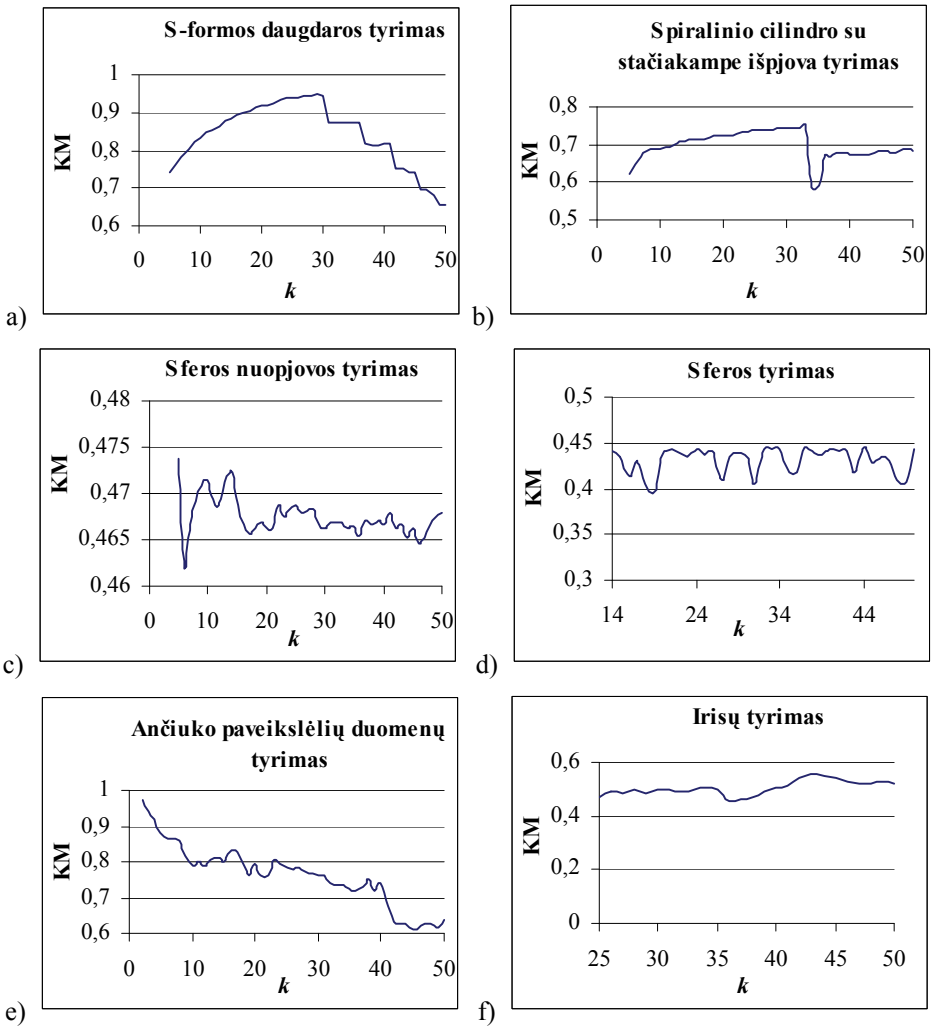
ISOMAP metodo trūkumai:

- trumpiausio kelio grafe radimas, o tuo pačiu ir geodezinių atstumų tarp taškų porų įverčiai, labai priklauso nuo analizuojamų duomenų (taškų skaičiaus, triukšmo) bei nustatytų artimiausių kaimynų (parametro $k$ ar atvirojo rutulio spindulio $\delta$ parinkimo). Blogai parinkus parametrų $k$ ar $\delta$ reikšmes, kaimynystės grafe sudaromos klaidingos jungtys, o tai nulemia projekcijų kokybę (Lee and Verleysen 2007).
- ISOMAP sėkmingai gali būti naudojamas tik, kai taškai išsidėstę ant daugdaros, kuri yra geodeziškai iškila (joje nėra kiaurymių) ir jos projekcija yra iškila (Donoho and Grimes 2003). Tačiau, jei daugdara yra geodeziškai neiškila (pavyzdžiui, spiralinis cilindras, turintis centre stačiakampę išpjovą) ir yra lokaliai izometrinė į projekcinę erdvę, tuomet ISOMAP metodu gaunamos deformuotos projekcijos: trūkstama sritis labai išplečiama, deformuojant aplink esančias projekcijas (2.34b pav.).

ISOMAP metodu gautas daugiamatės erdvės taškų išsidėstymas dvimatėje plokštumoje, vizualizuojant netiesinių daugdarų taškus bei irisų duomenis, parodytas 2.34 paveiksle. Kadangi, ISOMAP algoritmo parametras $k$ labai įtakoja duomenų vizualizavimo kokybę, tai, prieš ieškant daugiamačių duomenų projekcijų (2.34 pav.), skaičiuotos daugdaros topologijos išlaikymo mato KM (žr. 7 skyrių) priklausomybės nuo ISOMAP parametro $k$ (a), b), c) atvejais $k \in [5; 50]$, d) atveju $k \in [14; 50]$, e) atveju $k \in [2; 50]$, f) atveju $k \in [25; 50]$) (2.35 pav.). Jomis remiantis pasirinkta ir ISOMAP algoritme panaudota tokia $k$ reikšmė, su kuria KM reikšmė didžiausia (geriausia).




**2.34 pav.** ISOMAP metodu vizualizuoti duomenys: a) S-formos daugdaros taškai ($k = 30$), b) spiralinio cilindro, turinčio centre stačiakampę išpjovą, taškai ($k = 33$), c) sferos nuopjovos taškai ($k = 5$), d) sferos taškai ($k = 14$), e) ančiuko paveikslėlių duomenys ($k = 2$), f) irisų duomenys ($k = 43$)




**2.35 pav.** Daugdaros topologijos išlaikymo mato KM priklausomybės nuo ISOMAP parametro *k*

## 2.4. Antrojo skyriaus apibendrinimas ir išvados

Duomenų analizėje bei vizualizavime įprastas uždavinys – atvaizduoti duomenis iš labai didelės dimensijos erdvės į mažesnės dimensijos projekcinę erdvę taip, kad kiek galima labiau būtų išlaikyta duomenų struktūra bei sudaryti galimybę vizualiai pažvelgti į sudėtingas daugiamačių duomenų aibes. Daugiamačių duomenų vizualizavimas suteikia galimybę tyrinėtojui pačiam stebėti tų duomenų grupavimosi tendencijas, įvertinti atskirų daugiamatės erdvės taškų tarpusavio artumą, racionaliai priimti sprendimus.

Šiame skyriuje yra atlikta duomenų dimensijos mažinimo (projekcijos) metodų analitinė apžvalga. Susisteminti ir išnagrinėti tie projekcijos metodai, kuriais daugiamačius duomenis transformuojant į mažesnės dimensijos erdvę siekiama išlaikyti tik lokalią struktūrą, t. y. atstumus tarp artimiausių taškų: trianguliacija, santykinės perspektyvos metodas, lokaliai tiesinis vaizdavimas, Laplaso matricos tikrinių žemėlapių metodas, Hesės matricos tikrinių žemėlapių metodas, lokaliosios liečiamosios erdvės rikiavimas. Taip pat trumpai apžvelgti dažnai naudojami projekcijos metodai (pagrindinių komponenčių analizė, daugiamatės skalės), kuriais stengiamasi išsaugoti ne tik lokalią, bet ir globalią struktūrą, t. y. atstumus tarp visų duomenų taškų, nes šių metodų trūkumai, atsiskleidę nagrinėjant tam tikras duomenų aibes (daugdaros tipo duomenis), išryškina nagrinėjamų metodų privalumus.

Analitinė apžvalga leido išgryninti kelias aktualias tyrimų kryptis, kurias vienija lokalios struktūros išlaikymo būtinumas. Toliau jas detalizuosime.

Netiesinės daugdaros atpažinimo metodai (*nonlinear manifold learning methods*) remiasi prielaida, kad duomenys yra išsidėstę ant mažesnės dimensijos netiesinės daugdaros, įdėtos į labai didelės dimensijos erdvę. Šių metodų tikslas – atrasti šią netiesinę daugdarą ir ją perkelti į mažesnio matavimo erdvę. Atlikus išsamią daugdaros atpažinimo metodų analizę, o ypač apžvelgus jų taikymo sritis, galima padaryti išvadą, jog šie metodai šiandien labai populiarūs ir plačiai taikomi, ypač vaizdų apdorojime. Todėl yra tikslinga tolesnius tyrimus koncentruoti į netiesinės daugdaros atpažinimo metodus, t. y. tuos dimensijos mažinimo metodus, kurie nagrinėja specifinius duomenis – daugdaros tipo daugiamačius duomenis, siekiant dar labiau didinti metodų efektyvumą.

Svarbus su daugdara susijęs dalykas yra jos topologija. Topologijos išlaikymui įvertinti sukurta daugybė įvairių matų, tačiau skirtingiems uždaviniams turi būti parenkami skirtingi topologijos išlaikymo matai. Todėl kita nemažiau svarbi spręstina problema yra rasti ir ištirti tuos matus, kurie būtų tinkami įvertinti daugdaros topologijos išlaikymą po jos transformavimo į mažesnės dimensijos erdvę.

Dažnai tenka dirbti su duomenų aibėmis, kurios pastoviai papildomos naujais duomenimis. Labai svarbu greitai atvaizduoti naujus duomenų taškus,

daug neprarandant tikslumo. Čia gali būti naudingos lokalios struktūros išlaikymo idėjos, jų kombinacijos su kitais netiesinės projekcijos metodais. Tam reikalingi tyrimai.

Daugiamačių duomenų projekcijos metodai susiduria su dviem pagrindinėmis problemomis. Reikia ne tik atvaizduoti daugiamačius objektus atitinkančius taškus mažesnės dimensijos erdvėje (dvimatėje ar trimatėje), kuo tiksliau išlaikant atstumų tarp atitinkamų daugiamatės erdvės taškų ir jų projekcijų santykius, bet ir daugiamačius duomenis atvaizduoti mažesnės dimensijos erdvėje taip, kad jų projekcijos nepersidengtų. Čia irgi akivaizdi lokalios struktūros išlaikymo panaudojimo idėja, kurią reikėtų vystyti.

# 3

# Trianguliacijos ir Sammono metodų bei jų jungimo tyrimas

Šiame skyriuje tiriami du netiesiniai duomenų dimensijos mažinimo metodai – Sammono projekcija (Sammon 1969) ir trianguliacijos metodas (Lee *et al*. 1977) bei jų junginys. Sammono projekcija priklauso daugiamačių skalių (MDS) metodų klasei. Sammono metodas yra vienalaikio (*simultaneous*) atvaizdavimo metodas, t. y. duomenų taškai atvaizduojami visi iš karto. Juo bandoma išlaikyti atstumus tarp visų duomenų taškų. Trianguliacijos metodas yra nuoseklaus (*sequential*) atvaizdavimo metodas, t. y. duomenų taškai atvaizduojami ne iš karto visi, bet vienas paskui kitą; taškas dedamas atsižvelgiant į anksčiau atvaizduotų taškų išsidėstymą. Trianguliacijos metodu atvaizdavus naują tašką, tiksliai išlaikomi jo atstumai tik iki dviejų anksčiau atvaizduotų taškų; atstumai tarp tų dviejų taškų taip pat yra tiksliai išlaikyti. Darbe pasiūlyta Sammono ir trianguliacijos metodų junginio nauja realizacija, leidžianti atvaizduoti naujus daugiamatės erdvės taškus pakankamai tiksliai ir greitai, neatliekant Sammono projekcijos visai analizuojamų taškų aibei. Atlikta algoritmų lyginamoji analizė pagal šiuos aspektus: vizualus daugiamatės erdvės taškų projekcijų įvertinimas; laiko, per kurį atvaizduojami taškai, įvertinimas; projekcijos paklaidos įvertinimas (Sammono paklaidos prasme).

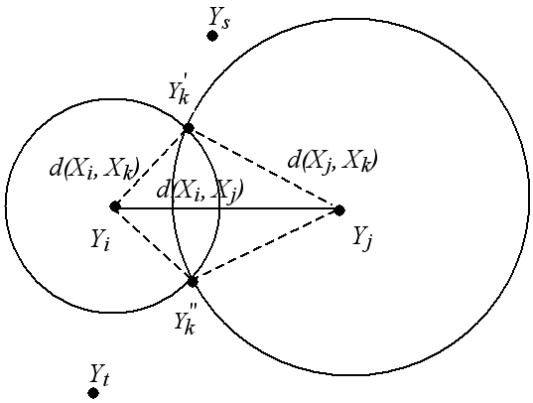
Pagrindiniai skyriaus rezultatai paskelbti straipsnyje (Karbauskaitė and Dzemyda 2006).

## 3.1. Trianguliacijos metodas

Tarkime, kad turime tris daugiamatės erdvės taškus $X_i$, $X_j$ ir $X_k$. Pirmiausia, plokštumoje yra atidedamos taškų $X_i$ ir $X_j$ projekcijos $Y_i$ ir $Y_j$ taip, kad tiksliai išlaikomas atstumas tarp taškų $X_i$ ir $X_j$, tai yra $d(X_i, X_j) = d(Y_i, Y_j)$. Tada ieškoma trečiojo taško $X_k$ projekcijos $Y_k$ plokštumoje taip, kad visi atstumai tarp daugiamatės erdvės taškų $X_i$, $X_j$ ir $X_k$ būtų tiksliai išlaikyti. Tai pasiekiama nubrėžiant du apskritimus, kurių centrai yra $Y_i$ ir $Y_j$, o spinduliai lygūs $d(X_i, X_k)$ ir $d(X_j, X_k)$ (3.1 pav.).

Dėl trikampio nelygybės apskritimai arba susikerta dviejuose taškuose, arba liečiasi. Jei apskritimai tik liečiasi, tai bus vienintelis taškas $Y_k$. Jei apskritimai kertasi, tuomet galimos dvi taško $Y_k$ padėtys plokštumoje, t. y. $Y_k'$ ir $Y_k''$. Tada reikia nuspręsti, kurioje vietoje atvaizduoti tašką $Y_k$. Kiekvienai galimai taško $Y_k$ padėčiai $Y_k'$ ir $Y_k''$ tarp jau atvaizduotų taškų galime rasti arčiausią kaimyną (išskyrus $Y_i$ ir $Y_j$). Pažymėkime plokštumos taškų $Y_k'$ ir $Y_k''$ kitus artimiausius kaimynus atitinkamai $Y_s$ ir $Y_t$. Negalime tiksliai išlaikyti pradinių atstumų tarp $Y_k$ ir $Y_s$ ir tarp $Y_k$ ir $Y_t$. Tačiau suskaičiuosime klaidą, padarytą pažymint $Y_k$ $Y_k'$ vietoje, ir klaidą, kuri padaryta, padedant $Y_k$ į $Y_k''$ padėtį. Tada tašką $Y_k$ padėsime į tą padėtį, kur bus padaryta mažesnė klaida. Tegu $a = \left| d(X_k, X_s) - d(Y_k', Y_s) \right|$, $b = \left| d(X_k, X_t) - d(Y_k'', Y_t) \right|$. Jei $a < b$, tai $Y_k = Y_k'$, priešingu atveju, $Y_k = Y_k''$ (Karbauskaitė and Dzemyda 2006). Galima ir kitaip (Dzemyda *et al*. 2008): jei daugiamatis taškas $X_k$ yra arčiau $X_s$, tai $Y_k = Y_k'$, o jei $X_k$ yra arčiau $X_t$, tai $Y_k = Y_k''$.

Trianguliacijos metodo dėka, galima nuosekliai daugiamatės erdvės taškų aibę atvaizduoti į plokštumą. Atvaizdavus daugiamatės erdvės taškus, kiekvienam plokštumos taškui $Y_k$ visada egzistuoja du taškai $Y_i$ ir $Y_j$, tokie, kad visi atstumai tarp taškų $Y_i$, $Y_j$ ir $Y_k$ yra tiksliai išlaikyti, t. y. $d(X_i, X_j) = d(Y_i, Y_j)$, $d(X_i, X_k) = d(Y_i, Y_k)$ ir $d(X_j, X_k) = d(Y_j, Y_k)$.




**3.1 pav.** Daugiamatės erdvės taškų atvaizdavimas plokštumoje trianguliacijos metodu

Tarkime, kad turime $m$ daugiamatės erdvės taškų aibę. Atvaizdavus pirmuosius tris taškus į plokštumą, tiksliai išlaikomi trys atstumai. Atvaizduojant likusius $(m-3)$ taškus, kiekvienas taškas išlaiko dar du atstumus. Todėl, vizualizuojant plokštumoje $m$ daugiamatės erdvės taškų trianguliacijos metodu, tiksliai išlaikytų atstumų skaičius yra $3+2(m-3)=2m-3$. Kadangi iš viso atstumų yra $m(m-1)/2$, tai bus išlaikyti tik informatyviausi.

Siekiant atvaizduoti duomenų taškų aibę, pirmiausia, iš analizuojamų $n$-matės erdvės taškų turi būti sudarytas minimalaus jungimo medis (*minimal spaning tree*, MST) (Graham and Hell 1985). Minimalaus jungimo medžio idėja paremta grafų teorija. Grafo jungimo medis (*spanning tree*) yra grafas, kuriame visos viršūnės sujungtos į medį. Grafo šakos turi svorius arba ilgius. Medžio svoris – visų šakų svorių suma. Medis, kurio svoris yra mažiausias, vadinamas minimalaus jungimo medžiu. Yra daug skirtingų algoritmų minimalaus jungimo medžiui rasti (Kruskal 1956; Prim 1957). Visi atstumai minimalaus jungimo medyje yra tiksliai išlaikyti. Jei turime $m$ taškų, tai bus išlaikyta tik $(m-1)$ atstumų. Tačiau trianguliacijos metodas gali išlaikyti $(2m-3)$ atstumus, todėl privalome suteikti papildomos informacijos. Galimi du metodai, pagal kuriuos nustatoma, iki kurių dviejų duomenų aibės taškų atstumų didumas turi būti tiksliai išlaikytas, atvaizduojant naują tašką iš $n$-matės į dvimatę erdvę. Tai antrojo arčiausiojo kaimyno (*second nearest neighbour*) ir atramos taško (*reference point*) metodai.

*Antrojo arčiausiojo kaimyno metodas.* Tarkime, kad taškas $X_j$ jau atvaizduotas plokštumoje ir gautas dvimatės jo projekcijos taškas $Y_j$. Taškas $X_k$

yra tiesiogiai sujungtas su $X_j$ minimalaus jungimo medyje, be to, iš visų jau atvaizduotų taškų taškas $X_j$ yra arčiausias taškui $X_k$. Tegu tarp visų atvaizduotų taškų, kurių atstumai iki $X_j$ yra tiksliai išlaikyti pereinant iš *n*-matės erdvės į dvimatę, taškas $X_i$ yra arčiausias iki taško $X_k$ (taško $X_i$ projekcija yra taškas $Y_i$). Įprastai $X_i$ turi tiesioginę jungtį su $X_j$ minimalaus jungimo medyje. Norime tašką $X_k$ atvaizduoti plokštumoje, t. y. rasti dvimatį tašką $Y_k$. Taikant trianguliacijos metodą, taškas $Y_k$ plokštumoje atidedamas taip, kad atstumai nuo jo iki dviejų taškų $Y_i$ ir $Y_j$ būtų lygūs atstumams nuo taško $X_k$ iki dviejų taškų $X_i$ ir $X_j$, kurie ir yra du artimiausi taško $X_k$ kaimynai.

*Atramos taško metodas.* Jį taikant parenkamas vienas atramos taškas $X_i$, iki kurio atstumai nuo visų kitų taškų būtų visada išlaikomi pereinant iš *n*-matės į dvimatę erdvę. Taigi išlaikomi kiekvieno taško $X_k$ atstumai iki dviejų taškų: iki to, kuris tiesiogiai sujungtas su tašku $X_k$ minimalaus jungimo medyje, t. y. pirmojo arčiausiojo kaimyno, ir iki atramos taško $X_i$.

Šiame skyriuje daugiamatės erdvės taškų aibei atvaizduoti į plokštumą trianguliacijos metodu buvo naudotas toks algoritmas (Lee *et al*. 1977):

- Iš duotos taškų aibės, remiantis Primo algoritmu (Prim 1957), sudaromas minimalaus jungimo medis (MST).
- MST šaknimi (pradiniu tašku) pasirenkamas bet kuris taškas, t. y. sudaromas vienkryptis medis.
- Nustatoma taškų atvaizdavimo tvarka. Realizacijoje panaudotas pirma į plotį medžio ieškojimo metodas (Lee *et al*. 1977).
- Analizuojami taškai nustatyta tvarka iš eilės atvaizduojami plokštumoje.

## 3.2. Sammono projekcija

Sammono projekcija, dažnai dar vadinama Sammono metodu ar algoritmu, yra netiesinis objektų, apibūdinamų daugeliu parametrų, atvaizdavimas mažesnės dimensijos erdvėje (Sammon 1969). Tai vienas iš daugiamačių skalių (MDS) grupės metodų (Bezdek and Pal 1995). Metodo idėja – atvaizduoti daugiamačius objektus atitinkančius taškus mažesnės dimensijos erdvėje išlaikant atstumų tarp

atitinkamų daugiamatės erdvės taškų ir jų projekcijų santykius. Nagrinėsime atvejį, kai projekcinės erdvės, į kurią atvaizduojame *n*-matės erdvės taškus, dimensija yra 2, t. y. atvaizduojame į plokštumą. Sammono projekcija negali greitai apdoroti naujų taškų be papildomų skaičiavimų. Visi atstumai tarp taškų turi būti skaičiuojami iš naujo, naudojant visą turimą duomenų aibę. Todėl tai sudaro sunkumų, kai duomenys pastoviai atnaujinami arba papildomi naujais taškais, ir kai duomenų aibės yra labai didelės.

Tarkime, kad turime daugiamatės erdvės taškus $X_i = (x_{i1}, x_{i2}..., x_{in})$, $i = \overline{1,m}$ priklausančius erdvei $R^n$. Sprendžiamas uždavinys – šiuos *n*-matės erdvės taškus $X_1, X_2, ..., X_m \in R^n$ atvaizduoti (gauti projekciją) plokštumoje $R^2$. Juos atitiks dvimatės erdvės taškai $Y_1, Y_2, ..., Y_m \in R^2$. Čia $Y_i = (y_{i1}, y_{i2})$, $i = \overline{1,m}$. Atstumą tarp daugiamatės erdvės taškų $X_i$ ir $X_j$ pažymėkime $d(X_i, X_j)$, o atstumą tarp juos atitinkančių plokštumos taškų $Y_i$ ir $Y_j$ – $d(Y_i, Y_j)$, $i, j = \overline{1,m}$. Sammono projekcija minimizuoja projekcijos iškraipymą (paklaidą) $E_S$:

$$E_S = \frac{1}{\sum\limits_{\substack{i,j=1 \\ i<j}}^{m} d(X_i, X_j)} \sum_{\substack{i,j=1 \\ i<j}}^{m} \frac{\left(d(X_i, X_j) - d(Y_i, Y_j)\right)^2}{d(X_i, X_j)}. \tag{3.1}$$

Sammono paklaida (*Sammon's stress (error)*) $E_S$ – tai matas, kuris parodo, kaip tiksliai išlaikomi atstumai tarp taškų pereinant iš didesnės dimensijos erdvės į mažesnės dimensijos erdvę. Pagrindinis uždavinys – minimizuoti šią paklaidos funkciją $E_S$. Tam gali būti naudojami įvairūs optimizavimo metodai. J. W. Sammon darbe (Sammon 1969) pasiūlė vieną funkcijos $E_S$ minimizavimo strategiją, pagal kurią dvimačių taškų $Y_i \in R^2$ komponentės $y_{il}$, $i = \overline{1,m}$, $l = 1, 2,$ randamos pagal iteracinę formulę:

$$y_{il}(m'+1) = y_{il}(m') - \eta \frac{\dfrac{\partial E_S(m')}{\partial y_{il}(m')}}{\left|\dfrac{\partial^2 E_S(m')}{\partial y_{il}^2(m')}\right|}, \tag{3.2}$$

čia $m'$ yra iteracijos numeris, o $\eta$ – optimizavimo žingsnio ilgį reguliuojantis parametras. Viena iteracija perskaičiuojamos visų $m$ taškų $Y_i \in R^2$, $i = \overline{1,m}$ koordinatės.

Dalinėms išvestinėms rasti naudojamos šios formulės:

$$\frac{\partial E_S}{\partial y_{il}} = -\frac{2}{c}\sum_{\substack{j=1\\ i\neq j}}^{m}\left(\frac{d(X_i,X_j)-d(Y_i,Y_j)}{d(X_i,X_j)d(Y_i,Y_j)}\right)(y_{il}-y_{jl}), \tag{3.3}$$

$$\frac{\partial^2 E_S}{\partial y_{il}^2} = -\frac{2}{c}\sum_{\substack{j=1\\ i\neq j}}^{m}\frac{1}{d(X_i,X_j)d(Y_i,Y_j)}\left[\left(d(X_i,X_j)-d(Y_i,Y_j)\right)-\right.$$
$$\left.-\frac{(y_{il}-y_{jl})^2 d(X_i,X_j)}{d^2(Y_i,Y_j)}\right], \tag{3.4}$$

$$c = \sum_{\substack{i,j=1\\ i<j}}^{m} d(X_i,X_j). \tag{3.5}$$

Gauta projekcijos paklaida $E_S$ priklauso ir nuo parametro $\eta$ ir nuo taškų $Y_i$ $i=\overline{1,m}$, koordinačių $y_{il}$, $l=1,2$ pradinių reikšmių parinkimo. Eksperimentiškai nustatyta, kad mažiausia paklaida gaunama, kai $\eta \in [0{,}3;\ 0{,}4]$ (Sammon 1969; Kohonen 2001), tačiau šio parametro reikšmė gali būti parinkta ir didesnė (Dzemyda and Kurasova 2006).

Nors šiuo metu yra ir kitų paklaidos $E_S$ optimizavimo metodų, J. W. Sammono pasiūlytas variantas sėkmingai naudojamas daugelyje darbų (Dzemyda 2004; Dzemyda 2005; Kohonen 2001; Kaski 1997; Konig 2000).

*Apibendrintoji Sammono algoritmo schema (Dzemyda et al. 2008)*:

1. Skaičiuojami atstumai tarp analizuojamų taškų pradinėje erdvėje.
2. Atsitiktinai parenkamos vaizdo erdvės taškų koordinačių reikšmės.
3. Skaičiuojama projekcijos paklaida $E_S$ (3.1).
4. Atnaujinamos vaizdo erdvės taškų koordinačių reikšmės pagal (3.2) formulę.
5. Jeigu projekcijos paklaidos reikšmė mažesnė už pasirinktą slenkstį arba iteracijų skaičius viršija nustatytąjį, tuomet algoritmas sustabdomas, priešingu atveju grįžtama ir kartojama nuo 3 žingsnio.

## 3.3. Trianguliacijos metodo jungimas su Sammono projekcija

Darbe (Biswas *et al*. 1981) pasiūlyta trianguliacijos metodą jungti su Sammono algoritmu. Trianguliacijos metodas yra pakankamai greitas, tačiau pereinant iš *n*-matės į dvimatę erdvę juo tiksliai išlaikoma tik $(2m-3)$ atstumų tarp analizuojamų taškų. Sammono algoritmu bandoma išlaikyti visus $m(m-1)/2$ atstumus tarp taškų, tačiau jis yra gana lėtas. Norint pagreitinti skaičiavimus, nors ir atsisakant šiek tiek tikslumo, verta naudoti šių dviejų metodų junginį.

Iš pradžių parenkama $\tilde{m}$ *n*-matės erdvės taškų $X_1, X_2, ..., X_{\tilde{m}}$ $(\tilde{m} < m)$ ir pagal Sammono algoritmą atvaizduojama plokštumoje. Jie vadinami baziniais taškais. Likusieji $(m-\tilde{m})$ taškai nuosekliai vienas po kito atvaizduojami plokštumoje taikant trianguliacijos metodą taip, kad būtų tiksliai išlaikyti atstumai nuo kiekvieno taško, nepriklausančio baziniams taškams, iki artimiausių dviejų bazinių taškų. Dėl Sammono metodo, kuriuo atvaizduojami baziniai taškai, gali būti neišlaikoma trikampio nelygybė. Tai sukelia sunkumų, nes trianguliacijos metodas remiasi šia nelygybe. Biswas, Jain ir Dubes (1981) išvengė problemos, tikrindami trikampio nelygybę kiekvienam atvaizduojamam taškui ir, kai ji nebūdavo patenkinta, jie vieną po kito pasirinkdavo iš bazinių taškų trečią arčiausią kaimyną, ketvirtą ir taip toliau, kol nelygybė būdavo patenkinta.

Kyla klausimas, koks turi būti skaičius $\tilde{m}$? Kai žinomos taškų klasės, baziniais taškais turi būti keli atstovai iš kiekvienos klasės. Kitu atveju prieš analizę duomenys klasterizuojami ir jų centrai imami baziniais taškais. Darbe (Biswas *et al*. 1981) tokio junginio rezultatai palyginti su kitais projekcijos metodais, parodyta, kad toks metodas veikia gana greitai, o prarandama gana nedidelis tikslumas. Taip pat šis metodas gali būti taikomas, kai reikia greitai atvaizduoti naujus analizuojamos aibės taškus.

### 3.3.1. Sammono ir trianguliacijos metodų junginio realizavimas

Šiame skyrelyje pasiūlyta Sammono ir trianguliacijos metodų junginio nauja realizacija. Skirtingai nuo (Biswas *et al*. 1981), mūsų pasiūlytoje Sammono – trianguliacijos metodo realizacijoje nereikalaujama, kad būtų patenkinta trikampio nelygybė, kai atliekamas nuoseklus daugiamatės erdvės taškų atvaizdavimas. Analizuojamas taškas tiesiog atvaizduojamas jo dviejų arčiausių kaimynų artimoje aplinkoje. Šiuo atveju gauname trianguliacijos idėjos praplėtimą, kuris, galimas atvejis, turėtų padidinti susidomėjimą šiuo metodu.

Toliau pateikiame Sammono ir trianguliacijos algoritmų sujungimo ir nuoseklaus atvaizdavimo detales.

Sammono algoritmu bandoma išlaikyti visus atstumus tarp taškų, tačiau vis tiek gaunama tam tikra paklaida. Todėl, norint atvaizduoti naują tašką, taikant trianguliacijos metodą, gali būti gaunami trys atvejai: apskritimai liečiasi, apskritimai kertasi ir apskritimai nei liečiasi, nei kertasi. Tikėtiniausias trečiasis atvejis, kuris labiausiai komplikuotas atvaizduojant naujus taškus. Žemiau pasiūlytas naujas sprendimo būdas.

Tarkime, kad ieškoma daugiamatės erdvės taško $X_k$ projekcijos $Y_k = (y_{k1}, y_{k2})$ plokštumoje. Erdvėje $R^n$ randami taško $X_k$ du arčiausi kaimynai, t. y. $X_{k1} = \left(x_{k1}^1, x_{k2}^1, ..., x_{kn}^1\right)$ ir $X_{k2} = \left(x_{k1}^2, x_{k2}^2, ..., x_{kn}^2\right)$. Jų projekcijos plokštumoje bus taškai $Y_{k1} = \left(y_{k1}^1, y_{k2}^1\right)$ ir $Y_{k2} = \left(y_{k1}^2, y_{k2}^2\right)$. Pažymėkime: $d(Y_{k1}, Y_{k2}) = d^*$, $d\left(X_k, X_{k1}\right) = r_1$, $d\left(X_k, X_{k2}\right) = r_2$.

Brėžkime du apskritimus, kurių centrai yra $Y_{k1} = \left(y_{k1}^1, y_{k2}^1\right)$ ir $Y_{k2} = \left(y_{k1}^2, y_{k2}^2\right)$, o spinduliai lygūs $r_1$ ir $r_2$. Apskritimai nei liečiasi, nei kertasi, kai: a) $r_1 > d^* + r_2$; b) $r_2 > d^* + r_1$; c) $d^* > r_1 + r_2$. Analizuokime a) atvejį. Tegu $r_1 > d^* + r_2$. Situacijos gali būti tokios kaip parodyta 3.2 paveiksle. Išveskime formules plokštumos taško $Y_k$ koordinatėms $(y_{k1}, y_{k2})$ rasti. Pažymėkime: $\hat{d} = (r_1 - d^* - r_2)/2$.

Tiesės, einančios per du taškus $Y_{k1} = \left(y_{k1}^1, y_{k2}^1\right)$ ir $Y_{k2} = \left(y_{k1}^2, y_{k2}^2\right)$, lygtis (3.2a pav.):

$$\frac{y_1 - y_{k1}^1}{y_{k1}^2 - y_{k1}^1} = \frac{y_2 - y_{k2}^1}{y_{k2}^2 - y_{k2}^1}. \tag{3.6}$$

Pertvarkę šią lygtį, gauname:

$$y_2 = y_{k2}^1 + \frac{\left(y_1 - y_{k1}^1\right)\left(y_{k2}^2 - y_{k2}^1\right)}{y_{k1}^2 - y_{k1}^1}, \quad y_{k1}^2 \neq y_{k1}^1. \tag{3.7}$$

Trikampiai $Y_{k1}Y_aY_{k2}$ ir $Y_{k1}Y_bY_k$ yra panašūs, todėl jų atitinkamos kraštinės proporcingos:

$$\frac{y_{k1}^2 - y_{k1}^1}{d^*} = \frac{y_{k1} - y_{k1}^1}{r_1 - \hat{d}}. \tag{3.8}$$

Pertvarkę (3.8) formulę, gauname:

$$y_{k1} = y_{k1}^1 + \frac{\left(y_{k1}^2 - y_{k1}^1\right)\left(r_1 - \hat{d}\right)}{d^*}. \tag{3.9}$$

Įstatę (3.9) į (3.7) formulę, gauname plokštumos tašką $Y_k = (y_{k1}, y_{k2})$.

Tarkime, kad yra tokia situacija kaip parodyta 3.2b, 3.2c paveiksluose. Tuomet taško $Y_k = (y_{k1}, y_{k2})$ koordinatės randamos iš formulių:

$$y_{k1} = y_{k1}^1, \;\; y_{k2} = y_{k2}^2 - r_2 - \hat{d} \text{ (3.2b pav.)}, \tag{3.10}$$

$$y_{k1} = y_{k1}^1, \;\; y_{k2} = y_{k2}^2 + r_2 + \hat{d} \text{ (3.2c pav.)}. \tag{3.11}$$

Kartais galime gauti du koncentrinius apskritimus (3.2d pav.). Tuomet *n*-matėje erdvėje randame taško $X_k$ trečią artimiausią kaimyną $X_{k3} = \left(x_{k1}^3, x_{k2}^3, ..., x_{kn}^3\right)$ ($X_{k1}$ ir $X_{k2}$ – atitinkamai pirmas ir antras artimiausi kaimynai). Daugiamačio taško $X_{k3}$ projekcija plokštumoje bus taškas $Y_{k3} = \left(y_{k1}^3, y_{k2}^3\right)$. Brėžiame tiesę $Y_{k1}Y_{k3}$. Jos lygtis yra:

$$\frac{y_1 - y_{k1}^1}{y_{k1}^3 - y_{k1}^1} = \frac{y_2 - y_{k2}^1}{y_{k2}^3 - y_{k2}^1}. \tag{3.12}$$

Pertvarkę šią lygtį, gauname:

$$y_1 = y_{k1}^1 + \frac{\left(y_{k1}^3 - y_{k1}^1\right)\left(y_2 - y_{k2}^1\right)}{y_{k2}^3 - y_{k2}^1}, \; y_{k2}^3 \neq y_{k2}^1. \tag{3.13}$$

Po to, brėžiame apskritimą, kurio centras yra taškas $Y_{k1} = \left(y_{k1}^1, y_{k2}^1\right)$, o spindulys lygus $r_3 = r_2 + \frac{r_1 - r_2}{2}$. Apskritimo lygtis yra:

$$\left(y_1 - y_{k1}^1\right)^2 + \left(y_2 - y_{k2}^1\right)^2 = r_3^2. \tag{3.14}$$

Tiesės ir apskritimo susikirtimo taškas ir yra ieškomas taškas $Y_k = (y_{k1}, y_{k2})$. Šio taško ordinatė randama iš formulės:

$$y_2 = y_{k2}^1 \pm \sqrt{\frac{r_3^2\left(y_{k2}^3 - y_{k2}^1\right)^2}{\left(y_{k2}^3 - y_{k2}^1\right)^2 + \left(y_{k1}^3 - y_{k1}^1\right)^2}}. \tag{3.15}$$

Įstatę (3.15) į (3.13) formulę, gauname dvi galimas taško $Y_k = (y_{k1}, y_{k2})$ padėtis plokštumoje $Y_k^{'} = (y_{k1}^{'}, y_{k2}^{'})$ ir $Y_k^{''} = (y_{k1}^{''}, y_{k2}^{''})$. Nustatome, kuris iš dviejų gautų taškų tinka: skaičiuojame atstumus nuo šių taškų iki trečio artimiausio kaimyno projekcijos, t. y. $d\left(Y_k^{'}, Y_{k3}\right)$ ir $d\left(Y_k^{''}, Y_{k3}\right)$ ir imame tą tašką, nuo kurio atstumas iki trečio artimiausio kaimyno projekcijos yra mažesnis.

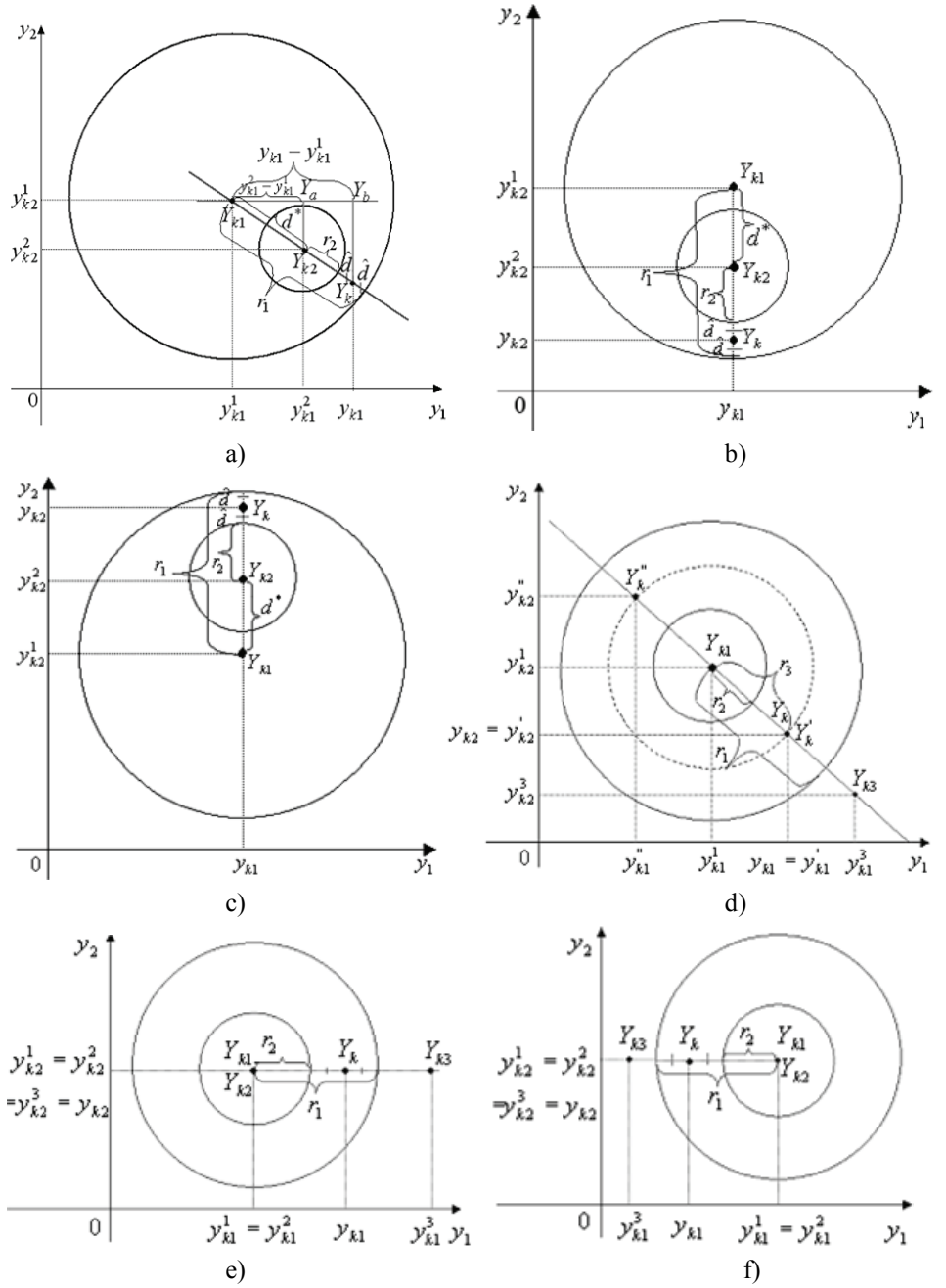
3.2e, 3.2f paveiksluose pateiktas specifinis prieš tai nagrinėtos situacijos atvejis, kai visų trijų artimiausių kaimynų projekcijų ordinatės yra lygios, t. y. $y_{k2}^1 = y_{k2}^2 = y_{k2}^3$. Tuomet taško $Y_k = (y_{k1}, y_{k2})$ koordinatės randamos iš formulių:

$$y_{k2} = y_{k2}^1,$$

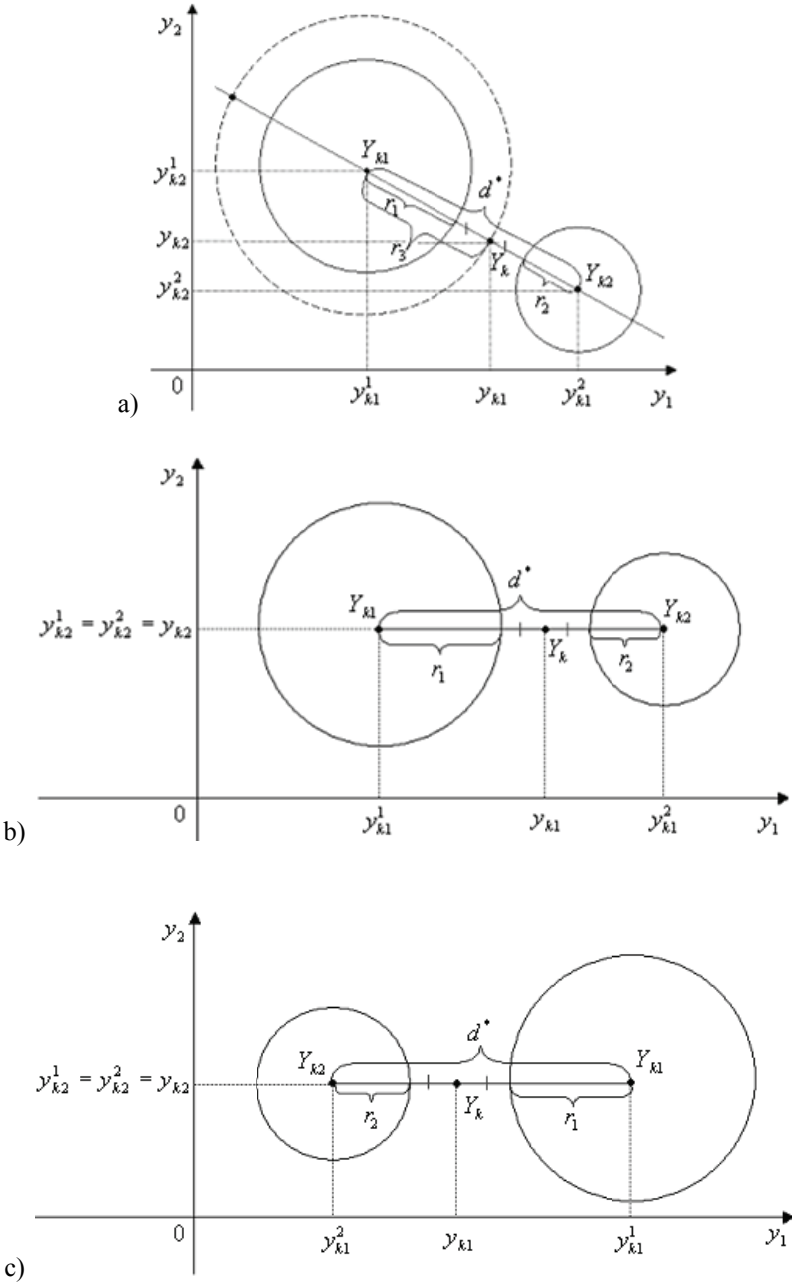
$$y_{k1} = y_{k1}^2 + r_2 + \frac{r_1 - r_2}{2}, \text{ jei } y_{k1}^3 \geq y_{k1}^1 \text{ (3.2e pav.)},$$

$$y_{k1} = y_{k1}^2 - r_2 - \frac{r_1 - r_2}{2}, \text{ jei } y_{k1}^3 < y_{k1}^1 \text{ (3.2f pav.)}.$$

Kai $r_2 > d^* + r_1$, gauname analogišką nagrinėtam atvejui algoritmą. Skirtumas tik tas, kad apskritimas, kurio centras yra $Y_{k1} = \left(y_{k1}^1, y_{k2}^1\right)$, o spindulys lygus $r_1$, bus apskritimo su centru $Y_{k2} = \left(y_{k1}^2, y_{k2}^2\right)$ ir spinduliu, lygiu $r_2$, viduje. Panašiu principu remiamasi ieškant naujo taško koordinačių, kai $d^* > r_1 + r_2$. Grafiškai vaizdas parodytas 3.3 paveiksle.




**3.2 pav.** Taško atvaizdavimas trianguliacijos metodu, kai apskritimai yra vienas kito viduje




**3.3 pav.** Taško atvaizdavimas trianguliacijos metodu, kai apskritimai yra vienas kito išorėje

## 3.4. Eksperimentinio trianguliacijos ir Sammono metodų bei jų junginio tyrimo rezultatai

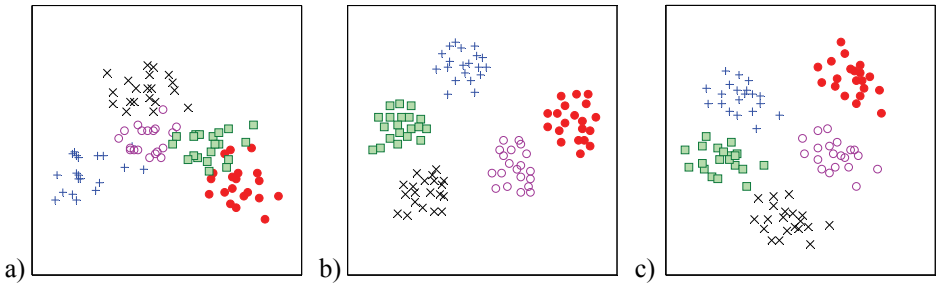
Trianguliacijos ir Sammono metodų bei jų junginio tyrimui naudojami testiniai duomenys – 100 10-matės erdvės taškų, kurie sudaro 5 klases. Generuojamas penkių 10-matės erdvės taškų, pasiskirsčiusių pagal normalųjį dėsnį, masyvas. Kiekvieno šio taško aplinkoje generuojama 20 atsitiktinių 10-matės erdvės taškų. Gaunamas 100 taškų masyvas. Iš 100 turimų taškų pasirenkami 75 taškai po 15 iš kiekvienos klasės. Jie laikomi pradiniais, o likusieji 25 sudarys naujų taškų aibę.

Atvaizduojant taškus trianguliacijos metodu (naudotas antrojo arčiausiojo kaimyno metodas), iš pradinių taškų sudaromas minimalaus jungimo medis (MST) ir jie vizualizuojami plokštumoje. Likusieji 25 taškai nuosekliai atvaizduojami plokštumoje, tiksliai išlaikant atstumus iki anksčiau atvaizduotų dviejų artimiausių taškų (3.4a pav.). Taip pat šie 100 taškų (75 pradiniai ir 25 nauji taškai) atvaizduojami tik Sammono metodu (3.4b pav.). Naudojant Sammono ir trianguliacijos metodų junginį, pradiniai 75 taškai vizualizuojami plokštumoje Sammono metodu, o nauji 25 taškai nuosekliai atvaizduojami plokštumoje trianguliacijos metodu (3.4c pav.). Projekcijos paklaida (Sammono paklaida) skaičiuojama atvaizdavus visus duomenis (100 taškų) (3.1 lentelė).

Iš gautų vaizdų (3.4 pav.) bei laiko, reikalingo taškams atvaizduoti, ir projekcijos paklaidos įverčių (3.1 lentelė), pastebime, kad

- Trianguliacijos metodas yra pakankamai greitas, bet projekcijos paklaida didelė. Sammono algoritmu gauta projekcijos paklaida nedidelė, tačiau jis yra gana lėtas. Sammono algoritmu duomenys (100 taškų) vizualizuojami maždaug 7,5 karto ilgiau nei trianguliacijos metodu, tačiau Sammono metodu gauta projekcijos paklaida yra 4 kartus mažesnė.
- Sammono ir trianguliacijos metodų junginys veikia gana greitai, o prarandamas tikslumas gana nedidelis. Sammono-trianguliacijos junginiu visi duomenys (pradiniai ir nauji) vizualizuojami 1,7 kartus greičiau, o paklaida yra didesnė tik maždaug 3% lyginant su rezultatais, gautais vizualizuojant duomenis vien tik Sammono algoritmu. Be to, naudojant Sammono-trianguliacijos junginį nauji taškai atvaizduojami net 83 kartus greičiau nei Sammono metodu.

Šis tyrimas parodė, jog naudoti vien tik trianguliacijos metodą duomenims vizualizuoti nėra pakankama, nes gauta duomenų projekcijos paklaida labai didelė. Sammono ir trianguliacijos metodų junginį verta naudoti, kai reikia greitai atvaizduoti naujus analizuojamos aibės taškus neprarandant didelio tikslumo.




**3.4 pav.** Testinių duomenų vizualizavimas: a) 100 taškų atvaizduota trianguliacijos metodu, b) 100 taškų atvaizduota Sammono metodu, c) 75 pradiniai taškai atvaizduoti Sammono ir 25 nauji taškai atvaizduoti trianguliacijos metodu

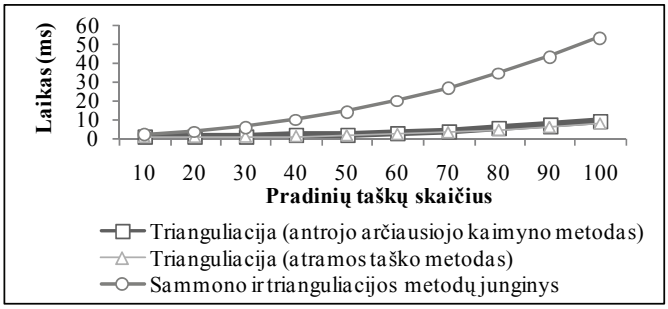
**3.1 lentelė.** Laiko ir projekcijos paklaidos įverčiai, gauti vizualizuojant testinius duomenis trianguliacijos, Sammono metodais bei jų junginiu

| Metodas | Laikas (ms) | | | Projekcijos paklaida (Sammono paklaida) |
|---|---|---|---|---|
| | 75 pradiniai taškai | 25 nauji taškai | Visi 100 taškų | |
| Trianguliacija | 605 | 78 | 683 | 0,1821811 |
| Sammono projekcija | 2890 | 5125 | 5125 | 0,0448481 |
| Sammono ir trianguliacijos metodų junginys | 2890 | 62 | 2952 | 0,0462552 |

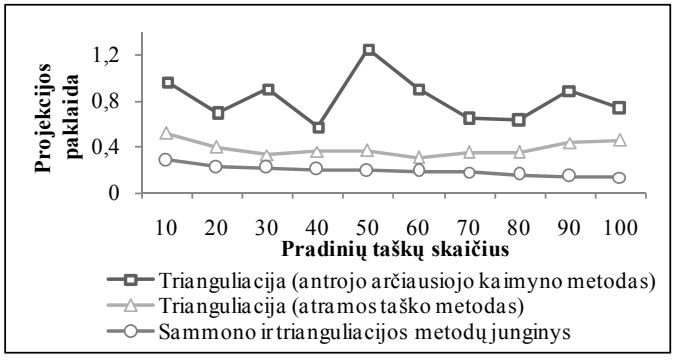
Šiame skyrelyje taip pat analizuojama, kaip laikas ir projekcijos paklaida priklauso nuo to, kiek atvaizduota pradinių taškų ir kiek naujų. Tyrimas atliktas su 100 15-mačiais atsitiktiniais duomenimis. Atsitiktiniai skaičiai sugeneruoti intervale (0;1). Pradinių taškų pasirenkama 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Iš viso atvaizduojama 100 taškų. Taigi atliekant tyrimą, pradinių taškų skaičius po truputį didinamas, o naujų taškų skaičius mažinamas. Projekcijos paklaida skaičiuojama atvaizdavus visus jau turimus duomenis. Trianguliacijos metodas realizuotas naudojant du metodus: antrojo arčiausiojo kaimyno metodą ir atramos taško metodą.

Atliekant šį tyrimą pastebėta, kad, atvaizduojant taškus tiek antrojo arčiausiojo kaimyno metodu, tiek atramos taško metodu, tiek Sammono bei trianguliacijos metodų junginiu, gaunama didėjanti laiko funkcija (3.5 pav.). Kuo didesnis pradinių taškų skaičius, tuo daugiau laiko užtrunkama atvaizduojant taškus. 3.6 paveiksle matome, kad projekcijos paklaida, gauta Sammono ir trianguliacijos metodų junginiu, didinant pradinių taškų skaičių

palaipsniui mažėja. Tačiau projekcijos paklaida, gauta vizualizuojant duomenis antrojo arčiausiojo kaimyno metodu arba atramos taško metodu, yra nepastovi, negalima nustatyti jokio dėsningumo. Neaišku, kada ji didesnė: ar padidinus pradinių taškų skaičių, ar jį sumažinus (3.6 pav.). Tada atliktas tyrimas su kopų duomenimis. Tai ekologiniai duomenys, nusakantys Suomijos pajūrio kopas ir jų vegetaciją (Hellemaa 1998). Tai šešiolika 16-matės erdvės taškų, gautų analizuojant kopas charakterizuojančių parametrų koreliacinę matricą (Dzemyda 2004). Atliekant tyrimą su kopų duomenimis, MST šaknimi (atramos tašku) pasirinkti pirmas ir antras duotos aibės taškai. Iš 3.7 paveikslo matyti, kad gautos skirtingos projekcijų paklaidos net esant tam pačiam pradinių taškų skaičiui. Vadinasi, projekcijos paklaida priklauso nuo taškų atvaizdavimo tvarkos.

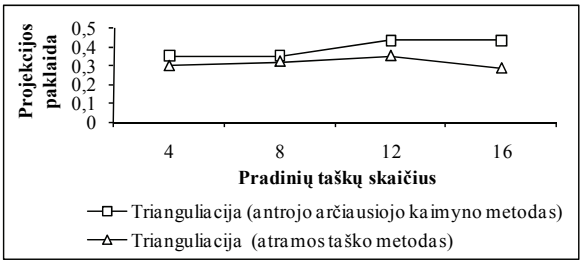



**3.5 pav.** Laiko, per kurį vizualizuojami 100 15-matės erdvės atsitiktinių taškų, priklausomybės nuo pradinių taškų skaičiaus

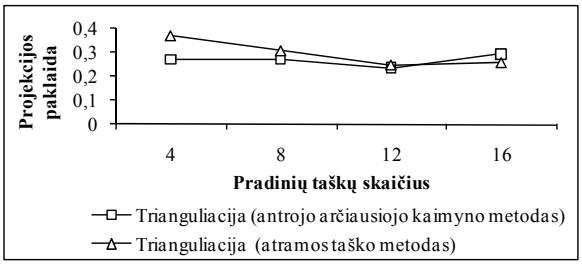



**3.6 pav.** Projekcijos paklaidos $E_s$ (3.1), gautos transformuojant į plokštumą 100 15-matės erdvės atsitiktinių taškų, priklausomybės nuo pradinių taškų skaičiaus

a)




b)




**3.7 pav.** Projekcijos paklaidos priklausomybės, gautos tiriant kopų duomenis antrojo arčiausiojo kaimyno ir atramos taško metodais, kai MST šaknis: a) I-as taškas, b) II-as taškas

## 3.5. Trečiojo skyriaus apibendrinimas ir išvados

Šiame skyriuje išnagrinėtas trianguliacijos metodas, Sammono algoritmas bei jų abiejų junginys. Eksperimentiškai ištirtos trianguliacijos metodo realizacijos, naudojančios antrojo arčiausiojo kaimyno ir atramos taško metodus atraminiams taškams parinkti. Nustatyta, jog abiem atvejais projekcijos paklaida labai priklauso nuo taškų atvaizdavimo sekos, be to, ši paklaida gana didelė. Vadinasi, naudoti vien tik trianguliacijos metodą duomenims vizualizuoti nėra pakankama.

Trianguliacijos metodas yra pakankamai greitas, tačiau juo išlaikomi tik $(2m-3)$ atstumai tarp analizuojamų taškų. Sammono algoritmas bando išlaikyti visus $m(m-1)/2$ atstumus tarp analizuojamų taškų, tačiau jis yra gana lėtas: kai turi būti atvaizduotas naujas taškas, visa atvaizdavimo procedūra turi būti pakartota iš naujo. Ši problema išsprendžiama trianguliacijos metodą jungiant su Sammono algoritmu. Toks metodas naudotinas, kai reikia greitai atvaizduoti naujus analizuojamos aibės taškus. Jis veikia gana greitai, o prarandamas gana nedidelis tikslumas.

Apskritai, bet kuris kitas daugiamačių skalių (MDS)-tipo metodas gali būti naudojamas junginyje vietoj Sammono metodo.

# 4

# Santykinės perspektyvos metodo realizacijų tyrimas

Lyginant su kitais žinomais dimensijos mažinimo metodais, santykinės perspektyvos metodas (RPM) yra visiškai kitoks metodas, sprendžiantis kompromisus, kurie iškyla, kai reikia išlaikyti tiek lokalią, tiek ir globalią duomenų struktūrą. Taigi RPM metodas siūlo naują būdą daugiamačiams duomenims tirti. Šiame skyriuje detaliai išnagrinėtas RPM metodas. Ištyrus jį eksperimentiškai, nustatyti šio metodo trūkumai, todėl pasiūlyta modifikacija, leidžianti jų išvengti. Pagrindiniai skyriaus rezultatai paskelbti straipsnyje (Karbauskaitė *et al*. 2006).

## 4.1. RPM metodo ypatumai

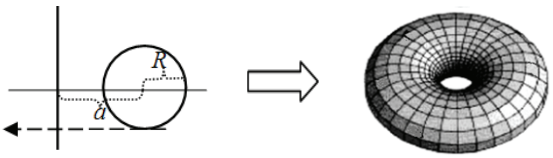
Santykinės perspektyvos metodas (*relational perspective map*, RPM) (Li 2004) skirtas daugiamačiams duomenims vizualizuoti ant toro paviršiaus. RPM, kaip ir daugiamačių skalių (*multidimensional scaling*) metodai, stengiasi išlaikyti atstumų tarp atitinkamų daugiamatės erdvės taškų ir jų projekcijų ant toro santykius. Tačiau pagrindinė RPM metodo savybė – galėjimas padalyti sudėtingą duomenų aibę į dalis ir vizualizuoti duomenis taip, kad jų projekcijos

nepersidengtų. Todėl RPM metodas išlaiko atstumus nedidelėje srityje (*short-range distances*) tiksliau negu kiti projekcijos metodai.

Tarkime, kad analizuojamus duomenis sudaro $m$ $n$-matės erdvės taškų $X_i = (x_{i1},...,x_{in}), i = \overline{1,m}$ ($X_i \in R^n$), sudarančių $m \times n$ dydžio matricą $X$. Atstumą tarp taškų $X_i$ ir $X_j$ pažymėkime $d(X_i, X_j)$, $i, j = \overline{1,m}$. RPM algoritmas šiuos daugiamatės erdvės taškus atvaizduoja į dvimatės erdvės taškus $Y_i = (y_{i1}, y_{i2}), i = \overline{1,m}$ ($Y_i \in R^2$), tarp kurių atstumai yra $d(Y_i, Y_j)$, $i, j = \overline{1,m}$, taip, kad atstumai tarp vizualizuojamų taškų ir jų projekcijų būtų kiek galima tiksliau išlaikyti.

Tam, kad vizualizuojamų taškų projekcijos nepersidengtų ir atstumai būtų kuo tiksliau išlaikyti, taškų atvaizdavimui pasirinktas toro paviršius.

**Apibrėžimas.** *Torą* apibrėžia spindulio $R$ apskritimas, besisukantis apie jo plokštumoje esančią tiesę, nutolusią nuo apskritimo centro atstumu $a > R$.




**4.1 pav.** Toro gavimas

**Fizikinė prasmė.** Vizualizuojant daugiamatės erdvės taškus ant toro, gaunami dvimatės erdvės taškai, kurie suprantami kaip dalelės, galinčios laisvai judėti ant toro paviršiaus, bet negalinčios nuo jo atitrūkti. Veikiamos stūmos jėgų, dalelės juda ant toro paviršiaus, kol išsidėsto taip, jog atstumų tarp jų ir atitinkamų daugiamatės erdvės taškų santykiai lygūs.

RPM algoritmas, atvaizduodamas taškus ant toro, minimizuoja potencinę energiją, kuri ir charakterizuoja taškų atvaizdavimą:

$$E_p = \sum_{i<j} \frac{d(X_i, X_j)}{p d^p(Y_i, Y_j)}, \text{ kur } E_0 = -\sum_{i<j} d(X_i, X_j) \ln\left(d(Y_i, Y_j)\right). \tag{4.1}$$

(4.1) formulėje naudojamas parametras $p$ vadinamas standumu, tvirtumu (*rigidity*). Jo reikšmė yra realus skaičius iš intervalo $(-1; +\infty)$. Daleles veikiančios jėgos aprašomos formule:

$$f_{ij} = \frac{\partial E_p}{\partial d(Y_i, Y_j)} = -\frac{d(X_i, X_j)}{d^{p+1}(Y_i, Y_j)}, \text{ kur } i < j. \tag{4.2}$$

Fizikiniu požiūriu, dalelės juda ant toro, veikdamos viena kitą stūmos jėga, ir todėl minimizuoja potencinę energiją. Iš (4.2) formulės matome, kad stūmos jėga $f_{ij}$, veikianti du toro paviršiaus taškus $Y_i$ ir $Y_j$, yra tiesiog proporcinga atstumui erdvėje tarp šių taškų – $d(X_i, X_j)$, todėl daugiamatės erdvės taškai, tarp kurių atstumas erdvėje yra didesnis, bus atvaizduoti toliau vienas nuo kito ant toro paviršiaus. Tarus, kad $p = 0$, remiantis (4.2) formule, galime teigti, kad stūmos jėga $f_{ij}$ tarp dviejų toro paviršiaus taškų $Y_i$ ir $Y_j$ yra atvirkščiai proporcinga atstumui tarp jų ant toro – $d(Y_i, Y_j)$. Tai reiškia, kad arčiau esantys toro paviršiaus taškai labiau įtakoja potencinę energiją $E_p$ ir todėl pateikia daugiau informacijos apie originalią duomenų aibę. Apskritai, parametras *p* suteikia galimybę kontroliuoti, kaip greitai stūmos jėgos mažės didėjant atstumams tarp toro paviršiaus taškų.

Kaip pavaizduota 2.13 paveiksle, RPM metodas pirmiausia atvaizduoja daugiamatės erdvės taškus ant toro paviršiaus, o tada, torą išardžius (torą perpjauname pasirinktoje vietoje ir gauname cilindrą, kurį vėl pjauname per vieną jo sudaromąją), gaunamas stačiakampis, kuriame ir matome analizuojamų taškų projekcijas.

## 4.2. RPM algoritmas

Prieš pateikiant algoritmą, pirma vertėtų išsiaiškinti, kas matematiškai yra toras ir daugiamatės erdvės taškų atvaizdavimas ant jo. Ar toro paviršius yra metrinė erdvė, kurioje toks atvaizdavimas būtų įmanomas?

Tarkime, kad $\overline{Y} = [0, w] \times [0, h] \subset R^2$ dvimatėje Dekarto koordinačių sistemoje apibrėžia stačiakampį, kurio plotis *w* ir aukštis *h*. $X = \{X_i = (x_{i1}, ..., x_{in}), i = \overline{1, m}\}$, $X_i \in R^n$ yra nagrinėjama daugiamatės erdvės taškų aibė, $Y = \{Y_i = (y_{i1}, y_{i2}), i = \overline{1, m}\}$, $Y_i \in \overline{Y}$ – gautų dvimatės erdvės taškų aibė. Tuomet taškų atvaizdavimas ant toro apibrėžiamas taip:

$$\varphi : X \to Y,\ X_i \to Y_i. \tag{4.3}$$

Siekiant pavaizduoti toro topologiją, apibrėžkime atstumo funkciją srityje $\overline{Y}$. Tegu $Y_i = (y_{i1}, y_{i2})$ ir $Y_j = (y_{j1}, y_{j2})$ yra du taškai iš aibės $Y$, tuomet atstumas tarp jų apibrėžiamas taip:

$$d(Y_i, Y_j) = \min\left\{|y_{i1} - y_{j1}|, w - |y_{i1} - y_{j1}|\right\} + \min\left\{|y_{i2} - y_{j2}|, h - |y_{i2} - y_{j2}|\right\}. \quad (4.4)$$

Dėl šios atstumų funkcijos priešingos stačiakampio $\overline{Y}$ kraštinės sulipdomos ir gauname paviršių, topologiškai ekvivalentų torui, kaip pavaizduota 2.13 paveiksle. $d(Y_i, Y_j)$ yra miesto-kvartalo atstumo metrika (*city-block distance metric*) ant toro paviršiaus; ji apibrėžia atstumą tarp dviejų taškų kaip trumpiausią atstumą tarp jų einant tik horizontaliai ir vertikaliai.

Kaip teigiama straipsnyje (Li 2004), tokia atstumo metrika pasirinkta todėl, kad 1) supaprastėja skaičiavimai, 2) eksperimentiškai įrodyta, kad gaunami tikslesni atvaizdavimai nei naudojant Euklido atstumus, nors teorinio paaiškinimo ir nėra.

Apibrėžę $\overline{Y}$ ir $d(Y_i, Y_j)$, galime teigti, kad toro paviršius, naudojamas RPM algoritme, iš tikrųjų yra metrinė erdvė $\left(\overline{Y}, d(Y_i, Y_j)\right)$. RPM algoritmo tikslas – rasti atvaizdavimą $\varphi$ (4.3), kuris minimizuotų potencinę energiją (4.1).

Energijos funkcijai (4.1) minimizuoti RPM algoritme pritaikomas NR (*Newton-Raphson*) metodas (Press *et al*. 2002). Jei tarsime, kad $f(z)$ yra vieno kintamojo funkcija, tai jos minimumo taškas randamas iš formulės:

$$z^{(m'+1)} = z^{(m')} - \frac{f'\left(z^{(m')}\right)}{f''\left(z^{(m')}\right)}. \quad (4.5)$$

Norint pritaikyti NR metodą nagrinėjamai optimizavimo problemai, reikia apskaičiuoti funkcijos $E_p$ pirmos ir antros eilės dalines išvestines pagal visus kintamuosius $y_{i1}$ ir $y_{i2}$. Pateiksime dalinių išvestinių formules tik pagal kintamuosius $y_{i1}$. Dalinių išvestinių skaičiavimas pagal kintamuosius $y_{i2}$ yra analogiškas.

Funkcijos $E_p$ pirmos eilės dalinės išvestinės pagal $y_{i1}, i = \overline{1,m}$ apskaičiuojamos pagal formulę:

$$\frac{\partial E_p}{\partial y_{i1}} = \sum_{i<j} h_{ij} f_{ij}, i, j = \overline{1,m}. \quad (4.6)$$

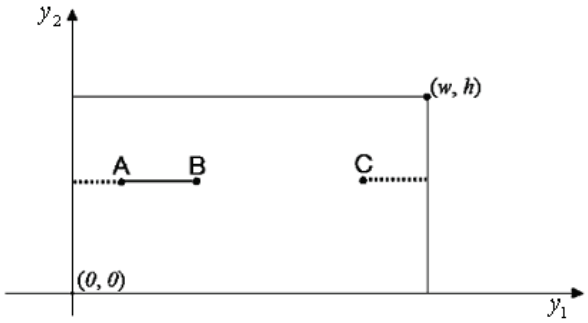
Čia $f_{ij}$ žymi stūmos jėgą ant toro tarp taškų $Y_i$ ir $Y_j$ ((4.2) formulė), $h_{ij}$ - operatorius, jungiantis toro topologiją su Dekarto koordinačių sistema. Operatorius $h_{ij}$ randamas iš formulių:

$$h_{ij}=\frac{\partial d(Y_i,Y_j)}{\partial y_{i1}}=\begin{cases}\dfrac{\partial\left(\left|y_{i1}-y_{j1}\right|\right)}{\partial y_{i1}},\text{jei}\left|y_{i1}-y_{j1}\right|<\dfrac{w}{2},\\ \dfrac{\partial\left(w-\left|y_{i1}-y_{j1}\right|\right)}{\partial y_{i1}},\text{jei}\left|y_{i1}-y_{j1}\right|>\dfrac{w}{2}\end{cases}=$$

$$=\begin{cases}+1,\text{jei}\left|y_{i1}-y_{j1}\right|<\dfrac{w}{2},y_{i1}>y_{j1},\\ -1,\text{jei}\left|y_{i1}-y_{j1}\right|<\dfrac{w}{2},y_{i1}<y_{j1},\\ -1,\text{jei}\left|y_{i1}-y_{j1}\right|>\dfrac{w}{2},y_{i1}>y_{j1},\\ +1,\text{jei}\left|y_{i1}-y_{j1}\right|>\dfrac{w}{2},y_{i1}<y_{j1}.\end{cases} \tag{4.7}$$

Jei $\left|y_{i1}-y_{j1}\right|=\dfrac{w}{2}$ arba $\left|y_{i1}-y_{j1}\right|=0$, tai gali būti naudojamos abi reikšmės: $h_{ij}=1$ arba $h_{ij}=-1$. Mes RPM algoritmo realizacijoje pasirinkome $h_{ij}=1$.

Kaip veikia operatorius $h_{ij}$, aiškiai iliustruoja 4.2 paveikslas. Paveiksle matome tris toro paviršiaus taškus A, B, ir C, pavaizduotus Dekarto koordinačių sistemoje. Trumpiausias kelias nuo A iki B yra tiesi linija tarp jų. Trumpiausias kelias nuo A iki C yra taškinė linija, kuri apsisuka apie torą. Taigi jei taškas A šiek tiek pasislenka į dešinę, atstumas tarp A ir B sumažėja, bet atstumas tarp A ir C padidėja. Šis veiksmas kaip tik ir charakterizuojamas šiuo operatoriumi. Šiuo atveju $h_{AB}=1$, $h_{AC}=-1$.




**4.2 pav.** Trumpiausias atstumas tarp taškų dėl toro topologijos

Funkcijos $E_p$ antros eilės dalinės išvestinės pagal $y_{i1}, i=\overline{1,m}$ apskaičiuojamos pagal formulę:

$$\frac{\partial^2 E_p}{\partial y_{i1}^2} = -(p+1)\sum_{i<j}\frac{f_{ij}}{d(Y_i,Y_j)}. \tag{4.8}$$

Įstatę (4.6) ir (4.8) į (4.5) formulę, gauname iteracinę formulę daugiamatės erdvės taškų projekcijų koordinatėms rasti:

$$y_{i1}^{(m'+1)} = y_{i1}^{(m')} + \frac{1}{p+1}\frac{\sum_{i<j} h_{ij} f_{ij}}{\sum_{i<j}\frac{f_{ij}}{d(Y_i,Y_j)}}. \tag{4.9}$$

Straipsnyje (Li 2004) pasiūlyta modifikuota formulė:

$$y_{i1}^{(m'+1)} = y_{i1}^{(m')} + c^{(m')}\frac{\sum_{i<j} h_{ij} f_{ij}}{\sum_{i<j}\frac{f_{ij}}{d(Y_i,Y_j)}}. \tag{4.10}$$

(4.10) formulėje konstanta $\frac{1}{p+1}$ pakeista parametru $c^{(m')}$, kuris vadinamas mokymo greičiu $m'$-oje iteracijoje. Didinant iteracijų skaičių $m'$, $c^{(m')}$ turėtų artėti į nulį. Parametras $c^{(m')}$ apskaičiuojamas pagal formulę:

$$c^{(m')} = \tilde{r}a^{m'}, \tag{4.11}$$

kur $\tilde{r}$ – pradinis mokymo greitis, $a \in (0;1)$. Skaičiai $\tilde{r}$ ir $a$ nustatomi empiriškai. Kuo labiau $a$ artėja prie 1, tuo optimizavimo procesas ilgesnis ir tuo tikslesnis taškų atvaizdavimas.

*Apibendrinta RPM algoritmo schema:*

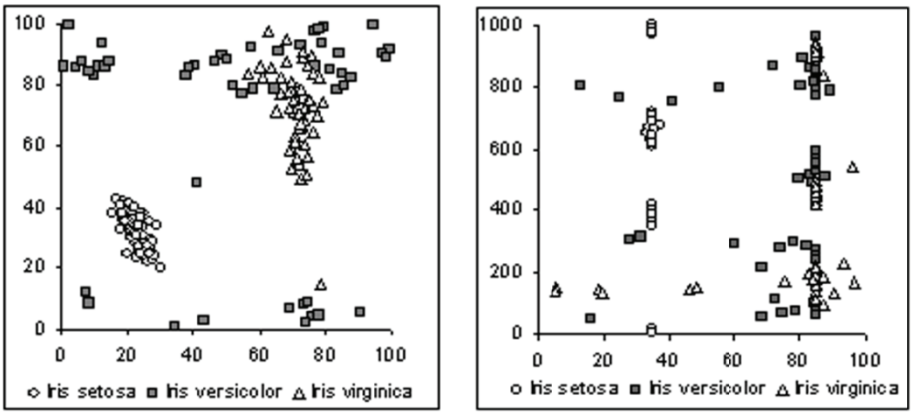
- daugiamatės erdvės taškai atsitiktinai išdėstomi ant toro;
- po to, jų projekcijos keičiamos pagal iteracinę (4.10) formulę ( $y_{i2}^{(m'+1)}$ apskaičiuojamos analogiškai) tol, kol taškų koordinačių pokyčių suma $\sum_{i=1}^{m}\left|y_{i1}^{(m'+1)} - y_{i1}^{(m')}\right| + \left|y_{i2}^{(m'+1)} - y_{i2}^{(m')}\right|$ tampa mažesnė už 0,0001.

## 4.3. RPM metodo eksperimentinio tyrimo rezultatai

Straipsnyje (Li 2004) tiksliai nenurodoma, kokios turėtų būti parametrų $w$ (stačiakampio plotis), $h$ (stačiakampio aukštis) ir $\tilde{r}$ (4.11) reikšmės, norint minimizuoti potencinę energiją ir gauti kuo tikslesnį daugiamačių duomenų atvaizdavimą plokštumoje (stačiakampyje).

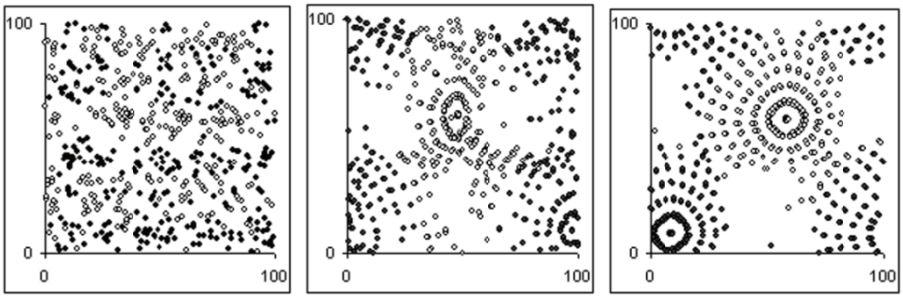
Tyrimus atlikome su dviem duomenų aibėmis: sferos taškais ir irisų duomenimis (žr. 2.1 skyrelį), norėdami išsiaiškinti, kaip nuo parametrų $\tilde{r}$, $w$, $h$ reikšmių priklauso gauti vaizdai (4.3, 4.4 pav.). Straipsnyje (Li 2004) teigiama, kad parametro $a$ reikšmė turėtų būti kuo artimesnė 1, todėl, atlikdami eksperimentus, pasirinkome $a = 0{,}975$. Taip pat, kaip rekomenduota (Li 2004), pasirinkome $p = 0$. Daugiamatės erdvės taškų koordinačių pradinės reikšmės plokštumoje (stačiakampyje) sugeneruotos atsitiktinai.

Atvaizduojant irisų duomenis, buvo pasirinktos skirtingos parametrų $w$ ir $h$ reikšmės ir tirtas potencinės energijos $E_p$ kitimas. Parametro $\tilde{r}$ reikšmė fiksuota: $\tilde{r} = 4$. Su kiekvienu parametrų $w$ ir $h$ rinkiniu (pvz., $w = 100$, $h = 100$) atlikta po 100 eksperimentų ir apskaičiuotos gautų potencinės energijos reikšmių vidutinės reikšmės. Pastebėta, kuo šių parametrų santykis didesnis, tuo potencinė energija pasiekia mažesnę reikšmę (4.5 pav.), tačiau taškų vizualizavimo plokštumoje kokybė blogesnė (4.3 pav.): 4.3b paveiksle klasės visai nesusidaro. Taigi turėdami skirtingo pločio ir ilgio stačiakampius, negalime lyginti potencinių energijų minimumų ir teigti, kad kuo mažesnė potencinė energija, tuo tikslesnis taškų atvaizdavimas.



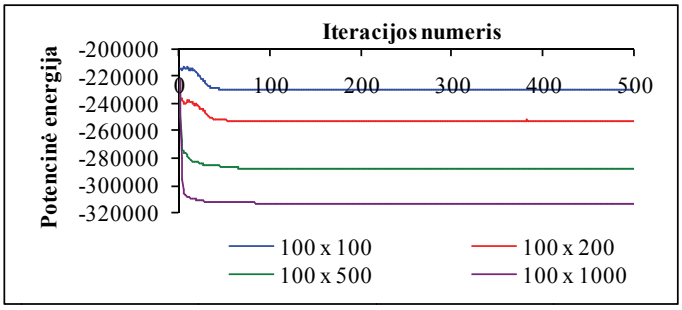

a) $w = 100$, $h = 100$ $(100 \times 100)$ b) $w = 100$, $h = 1000$ $(100 \times 1000)$

**4.3 pav.** Irisų duomenų vizualizavimas plokštumoje (stačiakampyje) RPM metodu, esant skirtingoms parametrų $w$ ir $h$ reikšmėms ($\tilde{r} = 4$)



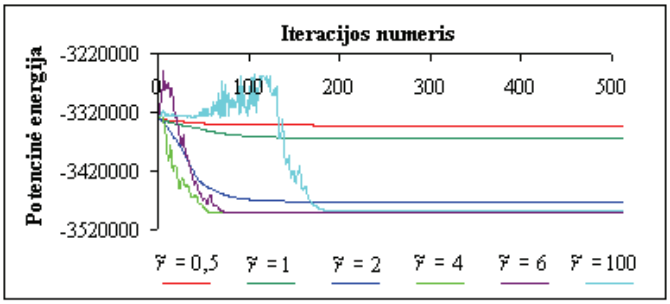

$\tilde{r} = 0{,}5,\ E_p = -3342223$ $\tilde{r} = 2,\ E_p = -3472738$ $\tilde{r} = 4,\ E_p = -3491417$

**4.4 pav.** Sferos taškų vizualizavimas plokštumoje (stačiakampyje) RPM metodu, esant skirtingoms parametro $\tilde{r}$ reikšmėms ( $w = h = 100$ )




**4.5 pav.** Potencinės energijos priklausomybės nuo iteracijos numerio, esant skirtingoms parametrų *w* ir *h* reikšmėms ( $w \times h$ )
(RPM metodu analizuoti irisų duomenys)

Atliekant tyrimus su sferos taškų aibe, buvo stebima potencinės energijos $E_p$ priklausomybė nuo parametro $\tilde{r}$ . Parametrų *w* ir *h* reikšmės buvo parinktos tokios: $w = h = 100$ (kadangi irisų duomenys vizualizuoti pakankamai gerai su šiomis *w* ir *h* reikšmėmis). 4.6 paveiksle pavaizduota, kaip, vizualizuojant sferos taškus, kinta potencinė energija, esant skirtingoms parametro $\tilde{r}$ reikšmėms. Nors energijos santykinė reikšmė labai žymiai nesikeičia (daugiausia (4%) ji pasikeičia tarp $\tilde{r} = 0{,}5$ ir $\tilde{r} = 4$ ), kintant $\tilde{r}$, tačiau ir nežymus energijos sumažėjimas iš esmės pakeičia taškų projekcijas: kai $\tilde{r} = 0{,}5$ , sferos viršutinė ir apatinė dalys visai neišryškėja, o kai $\tilde{r} = 4$ , gaunama pakankamai tiksli projekcija, t. y. matosi viršutinė ir apatinė pussferės (4.4 pav.).




**4.6 pav.** Potencinės energijos priklausomybės nuo iteracijos numerio, esant skirtingoms parametro $\tilde{r}$ reikšmėms
(RPM metodu analizuota sferos taškų aibė)

## 4.4. Nauja RPM metodo realizacija

Šiame skyrelyje pasiūlyta nauja RPM algoritmo modifikacija. Pagrindinės šios modifikacijos idėjos:

- kitu būdu apibrėžta atstumo funkcija srityje $\overline{Y}$;
- kiekvienoje iteracijoje perskaičiuojame ne visų taškų koordinates iš karto, bet imame po vieną tašką ir perskaičiuojame jo koordinates, atsižvelgdami į taškus, kurių koordinatės jau perskaičiuotos ir į taškus, kurių koordinatės dar nepakeistos.

Atstumo funkcija srityje $\overline{Y}$ apibrėžiama taip:

$$d\left(Y_i,Y_j\right)=\min\left\{\frac{\left|y_{i1}-y_{j1}\right|}{w},1-\frac{\left|y_{i1}-y_{j1}\right|}{w}\right\}+\min\left\{\frac{\left|y_{i2}-y_{j2}\right|}{h},1-\frac{\left|y_{i2}-y_{j2}\right|}{h}\right\}. \tag{4.12}$$

Įvedę tokią atstumo funkciją, gauname, kad $d\left(Y_i,Y_j\right)\in\left[0;1\right]$. Dėl šios atstumo funkcijos turime perskaičiuoti (4.7), (4.8) ir (4.9) formules:

$$h_{ij} = \frac{\partial d(Y_i, Y_j)}{\partial y_{i1}} = \begin{cases} +\frac{1}{w}, \text{ jei} \left| y_{i1} - y_{j1} \right| < \frac{w}{2}, y_{i1} > y_{j1}, \\ -\frac{1}{w}, \text{ jei} \left| y_{i1} - y_{j1} \right| < \frac{w}{2}, y_{i1} < y_{j1}, \\ -\frac{1}{w}, \text{ jei} \left| y_{i1} - y_{j1} \right| > \frac{w}{2}, y_{i1} > y_{j1}, \\ +\frac{1}{w}, \text{ jei} \left| y_{i1} - y_{j1} \right| > \frac{w}{2}, y_{i1} < y_{j1}. \end{cases} \tag{4.13}$$

Jeigu $\left| y_{i1} - y_{j1} \right| = \frac{w}{2}$ arba $\left| y_{i1} - y_{j1} \right| = 0$, tai tada gali būti naudojama $h_{ij} = \frac{1}{w}$ arba $h_{ij} = -\frac{1}{w}$. Mes pasirinkome $h_{ij} = \frac{1}{w}$.

Funkcijos $E_p$ antros eilės dalinės išvestinės pagal $y_{i1}, i = \overline{1,m}$ apskaičiuojamos pagal formulę:

$$\frac{\partial^2 E_p}{\partial y_{i1}^2} = -\frac{p+1}{w^2} \sum_{i<j} \frac{f_{ij}}{d(Y_i, Y_j)}. \tag{4.14}$$

Iteracinė formulė daugiamatės erdvės taškų projekcijų koordinatėms rasti bus tokia:

$$y_{i1}^{(m'+1)} = y_{i1}^{(m')} + \frac{w^2}{p+1} \frac{\sum_{i<j} h_{ij} f_{ij}}{\sum_{i<j} \frac{f_{ij}}{d(Y_i, Y_j)}} + wb. \tag{4.15}$$

(4.15) formulėje daugiklis *wb* pridėtas tam, kad būtų patenkinta sąlyga $0 \le y_{i1}^{(m'+1)} < w$, kur *b* yra sveikas skaičius.

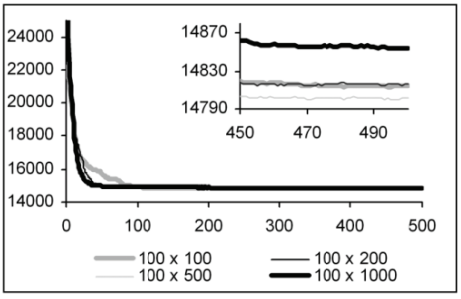
Skaičiavimai $y_{i2}$ atžvilgiu yra visiškai analogiški.

## 4.5. RPM algoritmo modifikacijos eksperimentinio tyrimo rezultatai

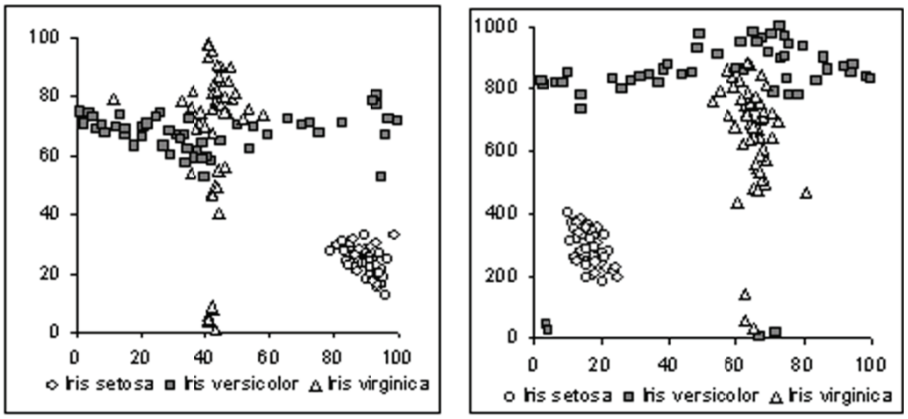
Vizualizuojant irisų duomenis, buvo pasirinktos skirtingos parametrų *w* ir *h* reikšmės ir tirtas potencinės energijos $E_p$ kitimas. Kiekvienai parametrų *w* ir *h*

porai (pavyzdžiui, $w = 100, h = 100$) atlikta po 100 eksperimentų ir apskaičiuotos gautų potencinės energijos reikšmių vidutinės reikšmės. Pastebėta, kad, naudojant RPM algoritmo modifikaciją, išvengiama energijos didelės priklausomybės nuo stačiakampio ilgio ir pločio (4.7 pav.), o gauti duomenų atvaizdavimai plokštumoje panašūs (klasių atskyrimo atžvilgiu) (4.8 pav.). Naudojant modifikuotą algoritmą, energijos santykinė reikšmė daugiausia pakinta 0,4% (4.7 pav.), tuo tarpu, naudojant RPM pradinį algoritmą (Li 2004), ji gali kisti net iki 27% (4.5 pav.). Pastebėkime: potencinė energija nekonverguoja į minimumą (4.7 pav.).

4.9 paveikslas iliustruoja sferos taškų vizualizavimą plokštumoje (stačiakampyje), kai $w = h = 100$.

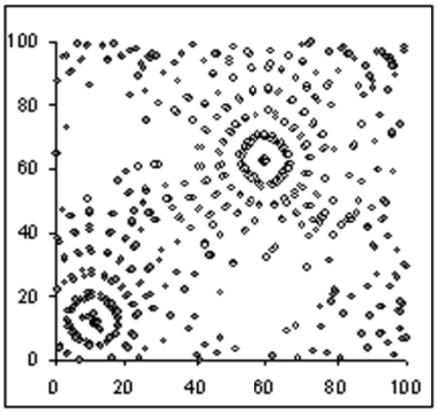



**4.7 pav.** Potencinės energijos priklausomybės nuo iteracijos numerio, esant skirtingoms parametrų $w$ ir $h$ reikšmėms ( $w \times h$ )
(RPM algoritmo modifikacija analizuoti irisų duomenys)




a) $w = 100$, $h = 100$ b) $w = 100$, $h = 1000$

**4.8 pav.** Irisų duomenų vizualizavimas plokštumoje (stačiakampyje) RPM algoritmo modifikacija, esant skirtingoms parametrų $w$ ir $h$ reikšmėms




**4.9 pav.** Sferos taškų vizualizavimas plokštumoje (stačiakampyje) naudojant RPM algoritmo modifikaciją ($w$ = 100, $h$ = 100)

## 4.6. Ketvirtojo skyriaus apibendrinimas ir išvados

Eksperimentai su įvairiais duomenimis parodė, kad RPM tipo algoritmai gali būti sėkmingai taikomi daugiamačių duomenų vizualiai analizei. Šio metodo privalumas – ne tik stengiamasi išlaikyti atstumų tarp atitinkamų daugiamatės erdvės taškų ir jų projekcijų ant toro santykius, bet duomenys vizualizuojami taip, kad jų projekcijos nepersidengtų.

RPM algoritmo (Li 2004) rezultatai labai priklauso nuo parametrų $w$, $h$ ir $\tilde{r}$. Deja, nėra suformuluotų aiškių taisyklių, kaip šiuos parametrus parinkti.

Mūsų pasiūlytame algoritme nėra didelės priklausomybės nuo parametrų $w$ ir $h$. Tačiau, atsisakius parametro $\tilde{r}$, potencinė energija nekonverguoja į minimumą. Tyrimų metu pastebėta, kad, atlikus apie 100 iteracijų, procesas stabilizuojasi. Norint tiksliai apibrėžti sustojimo sąlygą, reikalingi papildomi detalesni tyrimai.

RPM metodo trūkumas yra tai, kad funkcija $E_p$ yra nediferencijuojama kai kuriuose taškuose: jei yra tokios poros $y_{i1}$ ir $y_{j1}$, kad $y_{i1} = y_{j1}$ arba $|y_{i1} - y_{j1}| = w/2$, tada taško $y_{i1}$ aplinkoje funkcijos kairiosios ir dešiniosios išvestinių reikšmės nesutampa. Taikydami NR (*Newton-Raphson*) metodą, pasirinkome funkcijos kairiosios išvestinės reikšmę. Siekiant išvengti šio RPM metodo trūkumo, reikėtų arba naudoti funkcijos minimizavimo metodus, kurie nereikalauja funkcijos diferencijuojamumo, arba atstumo funkciją srityje $\overline{\overline{Y}}$ apibrėžti kaip funkciją, diferencijuojamą visuose toro paviršiaus taškuose.

# 5

# Lokaliai tiesinio vaizdavimo metodo parametrų parinkimo strategijos

Šiame skyriuje detaliai išnagrinėtas vienas iš netiesinės daugdaros atpažinimo metodų – lokaliai tiesinis vaizdavimas (LLE). Pasiūlyti nauji būdai LLE algoritmo valdymo parametrams parinkti. Skyriaus rezultatai paskelbti straipsniuose (Karbauskaitė *et al*. 2007; Karbauskaitė *et al*. 2008).

## 5.1. LLE algoritmas

Lokaliai tiesinis vaizdavimas (*locally linear embedding,* LLE) (Roweis and Saul 2000; Saul and Roweis 2003) yra netiesinis metodas, skirtas duomenų dimensijai mažinti ir daugdarai atpažinti. Naudojant LLE metodą, taškai, esantys ant daugdaros didelės dimensijos erdvėje, transformuojami į mažesnės dimensijos erdvę taip, kad daugdara yra „išvyniojama“.

Nagrinėkime glodžiąją (*smooth*) netiesinę daugdarą, kurią lokaliai galima aproksimuoti tiesiškai. Tarkime, kad ant šios daugdaros pasirinkta labai daug taškų ir kiekvienas taškas ir jo kaimynai yra ant arba arti daugdarą sudarančių tiesinių hiperplokštumų. Todėl kiekvienas duomenų taškas gali būti

aproksimuojamas jo kaimynų svorine tiesine kombinacija. Pagrindinė LLE metodo idėja yra ta, kad tokia tiesinė kombinacija lieka nepakitusi atlikus tiesines transformacijas (posūkį (*rotation*), mastelio keitimą (*rescaling*), perkėlimą (*translation*)) ir todėl nepasikeičia, kai daugdara yra „išvyniojama“ mažesnės dimensijos erdvėje. Duomenų projekcija mažesnės dimensijos erdvėje gaunama išsprendus du mažiausių kvadratų optimizavimo uždavinius su ribojimais.

Tarkime, kad analizuojamus duomenis sudaro $m$ $n$-matės erdvės taškų (vektorių) $X_i = (x_{i1},...,x_{in}), i = \overline{1,m}$ ($X_i \in R^n$), sudarančių $n \times m$ dydžio matricą $X$, o ieškomus (projekcijos) duomenis – $m$ $d$-matės erdvės taškų (vektorių) $Y_i = (y_{i1},...,y_{id}), i = \overline{1,m}$ ($Y_i \in R^d$), sudarančių $d \times m$ dydžio matricą $Y$.

LLE algoritmas turi tris žingsnius. Pirmame žingsnyje randami kiekvieno vizualizuojamo duomenų taško $X_i$ kaimynai. Čia artimumo matas yra Euklido atstumas. Kaimyniniai taškai gali būti randami dvejopai: arba ieškoma nustatyto skaičiaus $k$ artimiausių kaimynų, arba ieškoma kaimynų iš tam tikro fiksuoto dydžio spindulio $\delta$ atvirojo rutulio, kurio centras yra taškas $X_i$. Darbe naudojamoje LLE algoritmo realizacijoje ieškoma $k$ artimiausių taškų $X_{ij}$, $j = \overline{1,k}$, kuriuos vadinsime taško $X_i$ $k$ artimiausiais kaimynais.

Antrame žingsnyje randami tokie koeficientai (svoriai) $w_{ij}$, kad kiekvienas daugiamatės erdvės taškas $X_i, i = \overline{1,m}$ būtų išreiškiamas jo artimiausių kaimynų tiesine kombinacija $\sum_{j=1}^{k} w_{ij} X_{ij}$. Deja, čia neišvengiama paklaida, nes gaunami nauji, taškams $X_i$ gana artimi taškai $\overline{X}_i = \sum_{j=1}^{k} w_{ij} X_{ij}$. Paklaida įvertinama tokia funkcija:

$$E(W) = \sum_{i=1}^{m} \left\| X_i - \sum_{j=1}^{k} w_{ij} X_{ij} \right\|^2. \qquad (5.1)$$

Svoris $w_{ij}$ nusako, kokią įtaką daro taškas $X_{ij}$ taško $X_i$ rekonstrukcijai. Čia $X_{ij} = (x_{i1}^j, ..., x_{in}^j), i = \overline{1,m}, j = \overline{1,k}$ ir $\|\cdot\|$ yra Euklido atstumas. Norint apskaičiuoti svorių matricą $W$, minimizuojama (5.1) funkcija, esant ribojimui: taško $X_i$ visų svorių suma turi būti lygi 1, $\sum_{j=1}^{k} w_{ij} = 1$. Tai yra tipinis mažiausių

kvadratų optimizavimo uždavinys su ribojimu. Jo sprendinys lengvai randamas, išsprendus tiesinių lygčių sistemą.

Imkime konkretų duomenų tašką $X_i$, jo artimiausius $k$ kaimynus $X_{ij}$ bei svorius $w_{ij}$, $j=\overline{1,k}$, tenkinančius sąlygą $\sum_{j=1}^{k} w_{ij}=1$. Tuomet transformacijos paklaidą galima užrašyti tokiu būdu:

$$E^{(i)}(W)=\left\| X_i-\sum_{j=1}^{k} w_{ij}X_{ij}\right\|^2=\left\|\sum_{j=1}^{k} w_{ij}\left(X_i-X_{ij}\right)\right\|^2 = \sum_{j,l=1}^{k} w_{ij}w_{il}c^i_{jl}=\sum_{j=1}^{k} w_{ij}\sum_{l=1}^{k} c^i_{jl}w_{il}. \tag{5.2}$$

Čia $C^i=\{c^i_{jl},\ j,l=\overline{1,k}\}$, yra $k\times k$ lokalioji Gramo matrica, kurios elementai apskaičiuojami pagal šią formulę:

$$c^i_{jl}=\left(X_i-X_{ij}\right)\cdot\left(X_i-X_{il}\right), \tag{5.3}$$

kur $X_{ij}$ ir $X_{il}$ yra taško $X_i$ kaimynai.

LLE algoritmas gali būti apibendrinamas, vietoj Euklido atstumų naudojant kitus metrinius atstumus. Pavyzdžiui, užuot ieškojus kaimynų pradinių duomenų erdvėje (kaip daroma originaliame LLE algoritme), gali būti naudojamas branduolio atstumas (*kernel distance*) kaimynams rasti branduolio požymių erdvėje (*kernel feature space*). Branduoliu pagrįsti mokymo metodai (*kernel-based learning methods*) (atraminių vektorių metodai (*support vector machines*), branduolio PCA metodas (*kernel PCA*) ir kiti (Cristianini and Taylor 2000)) dažnai naudojami duomenų analizėje. Straipsnyje (DeCoste 2001) ištirtas atstumų, pagrįstų Mercer branduolių funkcijomis (*Mercer kernels*), panaudojimas. Rezultate pasiūlyta nauja branduolinė LLE forma, pavadinta KLLE.

Toliau pateikiame šiek tiek žinių apie Mercer branduolių funkcijas.

Primename, kad analizuojamus duomenis sudaro $m$ $n$-matės erdvės vektorių $X_i=\left(x_{i1},...,x_{in}\right), i=\overline{1,m}$ ($X_i\in R^n$). Tegu $\phi(\cdot)$ yra atvaizdavimo funkcija iš $n$-matės erdvės į kitą didelės dimensijos, galbūt net begalinės dimensijos požymių (*feature*) erdvę. Tegu $X_i$ ir $X_j$ yra du vektoriai iš erdvės $R^n$, tuomet branduolio funkcija apskaičiuojama tokiu būdu:

$$\kappa(X_i,X_j)=\phi(X_i)\cdot\phi(X_j), \tag{5.4}$$

t. y. $\kappa(X_i, X_j)$ yra dviejų vektorių $\phi(X_i)$ ir $\phi(X_j)$, priklausančių branduolio požymių erdvei, skaliarinė sandauga, gaunama neapskaičiavus tikslių šių vektorių koordinačių. Tokiu būdu, branduolių funkcijos įgalina ištirti dideles netiesines požymių erdves, išvengiant didžiulės dimensijos. Tačiau kai kuriais atvejais vektorių komponenčių branduolio požymių erdvėje tikslus apskaičiavimas gali būti naudingas (Dzemyda 2001).

Pažymėkime branduolio funkcijas $\kappa(X_i, X_j) = \kappa_{ij}$, $i, j = \overline{1, m}$ bei sudarykime branduolio funkcijų matricą iš $m$ eilučių ir $m$ stulpelių $K = \left\{\kappa_{ij}, i, j = \overline{1, m}\right\}$. Tegu turime $m$-matės erdvės vektorių $C = (c_1, c_2, ..., c_m)$ (komponentės paprastai yra realūs skaičiai). Tuomet $CKC^T = \sum_{i=1}^{m}\sum_{j=1}^{m} \kappa_{ij} c_i c_j$.

**Mercerio teorema.** Tegu turime du vektorius $X_a, X_b \in R^n$. Jei $\kappa(X_a, X_b)$ yra teigiamai pusiau apibrėžta branduolio funkcija, t. y. $CKC^T \geq 0$, tai $\kappa(X_a, X_b)$ galima išreikšti taip: $\kappa(X_a, X_b) = \phi(X_a) \cdot \phi(X_b)$.

Straipsnyje (DeCoste 2001) parodyta, jog LLE algoritme galima vartoti įvairias Mercer branduolių funkcijas: polinominę (*polynomial kernel*), radialinių bazinių funkcijų (*radial basis function kernel*), tiesinę (*linear kernel*). Čia branduolio Gramo matricos (*kernel Gram matrix*) elementai apibrėžiami tokia lygtimi:

$$\begin{aligned} c^i_{jl} &= \left(\phi(X_i) - \phi(X_{ij})\right) \cdot \left(\phi(X_i) - \phi(X_{il})\right) = \\ &= \kappa(X_i, X_i) - \kappa(X_i, X_{ij}) - \kappa(X_i, X_{il}) + \kappa(X_{ij}, X_{il}), \end{aligned} \tag{5.5}$$

kur $X_{ij}$ ir $X_{il}$ yra $X_i$ kaimynai.

Šiame skyriuje, kaip ir LLE algoritmo autoriai, naudojame tiesinę branduolio funkciją: $\phi(X_a) = X_a$ ir $\kappa(X_a, X_b) = X_a \cdot X_b$. Netiesinių branduolio funkcijų pritaikymas yra ateities tyrimų objektas. Žemiau parodyta, jog panaudojus tik tiesinę branduolio funkciją, galime gauti gerus vizualizavimo rezultatus.

(5.2) paklaida gali būti minimizuojama naudojant Lagranžo daugiklį, esant apribojimui $\sum_{j=1}^{k} w_{ij} = 1$. Panaudojant atvirkštinę Gramo matricą, optimalūs svoriai randami pagal formulę:

$$w_{ij} = \frac{\sum_{l=1}^{k} \left(c_{jl}^{i}\right)^{-1}}{\sum_{p=1}^{k}\sum_{q=1}^{k} \left(c_{pq}^{i}\right)^{-1}}, \tag{5.6}$$

kur $\left(c_{jl}^{i}\right)^{-1}$, $j,l = \overline{1,k}$ yra atvirkštinės Gramo matricos $\left(C^{i}\right)^{-1}$ elementai. Tokiu būdu ieškant svorių, sugaištama labai daug laiko. Paprastesnis būdas ieškoti optimalių svorių – spręsti tiesinių lygčių sistemą:

$$\sum_{l=1}^{k} c_{jl}^{i} w_{il} = 1, \tag{5.7}$$

ir gautus svorius perskaičiuoti taip, kad jų suma būtų lygi 1:

$$w_{ij} \leftarrow w_{ij} / \sum_{l=1}^{k} w_{il}. \tag{5.8}$$

Kai artimiausių kaimynų skaičius yra didesnis už pradinių duomenų dimensiją ($k > n$), tai lokalioji Gramo matrica yra singuliari (išsigimusi) arba beveik singuliari. Tokiu atveju, prieš sprendžiant sistemą (5.7), reikia atlikti lokaliosios Gramo matricos reguliarizaciją, t. y. prie matricos pagrindinės įstrižainės elementų pridėti labai mažą teigiamą konstantą $\varepsilon$:

$$c_{jl}^{i} \leftarrow c_{jl}^{i} + \delta_{jl}\varepsilon, \tag{5.9}$$

kur $\delta_{jl} = 1$, jei $j = l$, ir $\delta_{jl} = 0$, priešingu atveju. Konstanta $\varepsilon$ toliau bus vadinama reguliarizacijos parametru ir apskaičiuojama pagal formulę:

$$\varepsilon = \mathrm{tr}\left(C^{i}\right)t, \tag{5.10}$$

kur $\mathrm{tr}\left(C^{i}\right)$ – matricos $C^{i}$ pėdsakas, $t$ – valdymo parametras, parenkamas vartotojo ($t > 0$, $t \ll 1$).

Trečiame žingsnyje randami daugiamačių duomenų $X_i$ projekcijos vektoriai $Y_i = \left(y_{i1},\ldots,y_{id}\right), i = \overline{1,m}$, ($Y_i \in R^d$). LLE algoritmo tikslas – kiek galima tiksliau išlaikyti daugiamatės erdvės lokalią tiesinę struktūrą mažesnės dimensijos erdvėje. Todėl svoriai $w_{ij}$, kurie transformuoja kiekvieną duomenų tašką $X_i$ jo artimiausių kaimynų atžvilgiu *n*-matėje erdvėje, turi transformuoti suprojektuotus daugdaros taškus *d*-matėje erdvėje. Taigi rasti svoriai fiksuojami ir projekcijos vektoriai $Y_i$ apskaičiuojami minimizuojant tokią funkciją:

$$\Phi(Y)=\sum_{i=1}^{m}\left\|Y_i-\sum_{j=1}^{k}w_{ij}Y_{ij}\right\|^2, \tag{5.11}$$

atsižvelgiant į du apribojimus, kurie užtikrina vienintelį sprendinį:

- $\frac{1}{m}\sum_{i=1}^{m}Y_iY_i^T=I$, kur $I$ – vienetinė matrica, sudaryta iš $d$ eilučių ir $d$ stulpelių. Projekcijų koordinatės turi turėti vienetinę kovariaciją, kad pašalintų posūkio (*rotation*) laisvės laipsnį ir fiksuotų skalę.
- $\sum_{i=1}^{m}Y_i=0$, 0 yra $d$-matis nulinis vektorius. Pašalina perkėlimo (*translation*) laisvės laipsnį, reikalaujant, kad gauti vektoriai būtų sucentruoti koordinačių sistemos pradžioje.

Tegu turime matricą $W=\left\{\left(\overline{w}_{1j},\overline{w}_{2j},...,\overline{w}_{mj}\right),\ j=\overline{1,m}\right\}$, kurios elementai $\overline{w}_{ij}$, $i,j=\overline{1,m}$ yra svoriai tarp taškų $X_i$ ir $X_j$. Be to, $\overline{w}_{ij}=0$, jei $X_i$ ir $X_j$ nėra artimiausi kaimynai, priešingu atveju, $\overline{w}_{ij}$ reikšmės gaunamos minimizuojant funkciją (5.1). $I$ – vienetinė matrica, sudaryta iš $m$ eilučių ir $m$ stulpelių.

Norint paprasčiausiu būdu apskaičiuoti vektorių $Y_i$, $i=\overline{1,m}$ $d$-mates komponentes, pakanka rasti simetrinės išretintos matricos $\widetilde{M}=(I-W)^T(I-W)$ apatinius $d+1$ tikrinius vektorius. Šie tikriniai vektoriai yra susiję su matricos $\widetilde{M}$ $d+1$ mažiausiomis tikrinėmis reikšmėmis. Iš šių vektorių reikia atmesti apatinį tikrinį vektorių, kurio tikrinė reikšmė lygi 0. Jis yra vienetinio ilgio su vienodomis komponentėmis. Likusieji $d$ tikriniai vektoriai suformuoja projekcijų $Y_i$ koordinates. Tašką $Y_i$ sudaro gautų tikrinių vektorių $i$-ųjų komponenčių rinkinys.

*LLE algoritmą apibendrinanti schema:*

1. Randami kiekvieno duomenų taško $X_i$ kaimynai.
2. Minimizuojant funkciją (5.1), apskaičiuojami svoriai $w_{ij}$, kuriuos naudojant kiekvienas daugiamatės erdvės taškas $X_i$ išreiškiamas jo kaimynų tiesine kombinacija.
3. Fiksavus svorius $w_{ij}$, apskaičiuojami projekcijos taškai $Y_i$, minimizuojant funkciją (5.11).

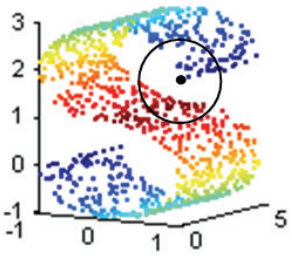
## 5.2. Parametrų parinkimo LLE algoritme tyrimas

LLE algoritmas turi tris valdymo parametrus: artimiausių kaimynų skaičių kiekvienam duomenų taškui, lokaliosios Gramo matricos reguliarizacijos parametrą bei duomenų vidinę dimensiją (daugdaros dimensiją) *d*. Nuo šių parametrų labai priklauso duomenų vizualizavimo kokybė. Šiame skyrelyje pasiūlyti nauji būdai artimiausių kaimynų skaičiui bei reguliarizacijos parametro reikšmei parinkti. Metodai eksperimentiškai ištirti su keliomis duomenų aibėmis: dirbtiniais ir realiais duomenimis, gautais skaitmenizavus paveikslėlius. Sprendžiant daugdaros atpažinimo uždavinį, ant daugdaros esantys duomenų taškai turi būti transformuojami į tokios dimensijos erdvę, kokia yra daugdaros dimensija *d.* Tačiau šioje disertacijoje nagrinėjami daugiamatės erdvės taškai, išsidėstę tik ant dvimačių daugdarų, todėl pasirinkta $d = 2$. Taigi LLE algoritme lieka tik du valdymo parametrai.

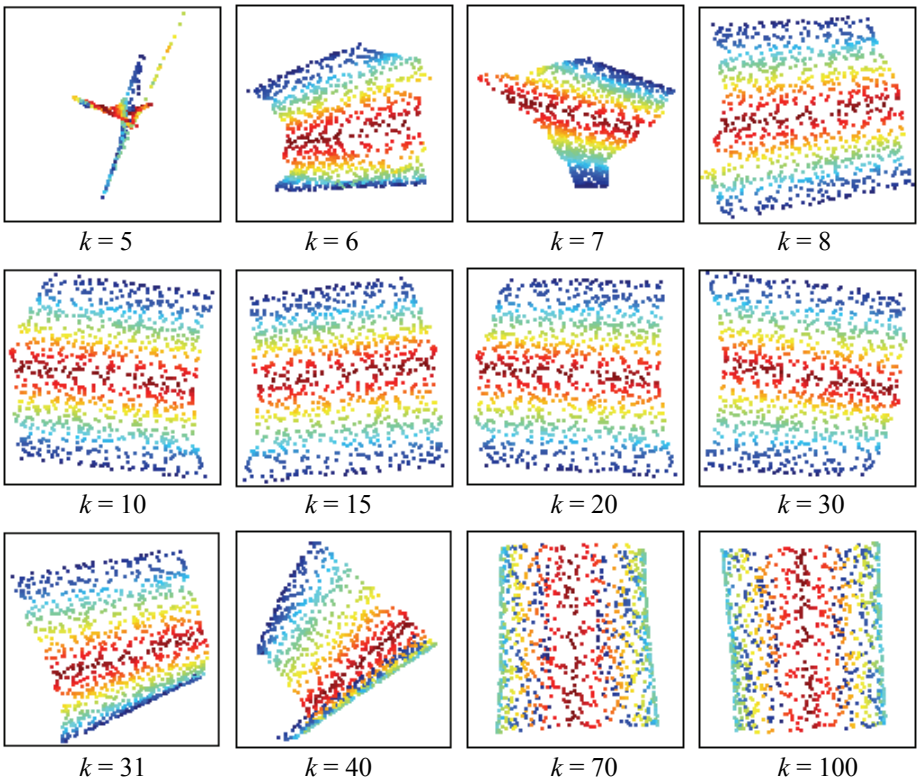
### 5.2.1. Artimiausių kaimynų skaičiaus parinkimas LLE algoritme

Šiame skyrelyje tyrimai atliekami su 1000 3-matės erdvės taškų, priklausančių netiesinei dvimatei S-formos daugdarai (5.1 pav.).

Svarbiausias LLE algoritmo žingsnis, nulemiantis rezultatų tikslumą, yra pirmasis, t. y. nustatyti artimiausių kaimynų skaičių *k* kiekvienam duomenų taškui. Nuo šio parametro labai priklauso duomenų vizualizavimo kokybė. Jei *k* reikšmė per maža, vientisa daugdara suardoma ir atvaizdavimas neatspindi duomenų globalių savybių (5.2 pav., pvz., $k = 5$). Jei *k* reikšmė per didelė, atvaizdavimas praranda savo netiesiškumo savybę ir LLE algoritmas elgiasi kaip tiesiniai dimensijos mažinimo metodai, pavyzdžiui, tradicinis PCA (2.24b pav.), kai visa duomenų aibė yra suvokiama kaip viena kaimynystė (5.2 pav., pvz., $k = 100$).



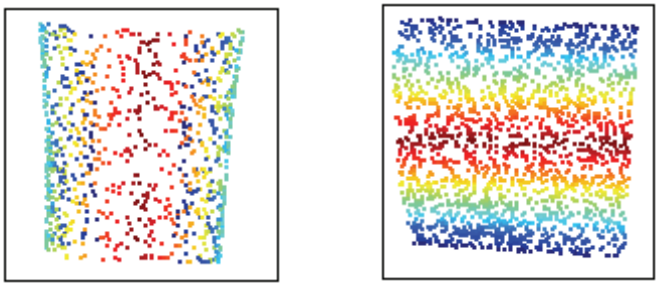

**5.1 pav.** 1000 3-matės erdvės taškų, išsidėsčiusių ant netiesinės dvimatės S-formos daugdaros




**5.2 pav.** S-formos daugdaros taškų ($m = 1000$) projekcijos plokštumoje, gautos LLE metodu, esant įvairiam artimiausių kaimynų skaičiui $k$

LLE algoritmu gauti vizualizavimo rezultatai paprastai yra stabilūs, kai artimiausių kaimynų skaičius $k$ parenkamas iš tam tikro dydžio intervalo (Saul and Roweis 2003). 5.2 paveiksle pavaizduota daugybė projekcijų, gautų LLE algoritmu vizualizavus tą pačią duomenų aibę, tik parinkus skirtingą artimiausių kaimynų skaičių $k$. Tikėtinos projekcijos (daugdara su analizuojamais taškais „išvyniojama“) yra gaunamos gana dideliame artimiausių kaimynų skaičiaus intervale, t. y. $k \in [8; 30]$. Šios projekcijos bus laikomos tinkamomis (teisingomis). Tačiau, kaip minima (Saul and Roweis 2003), to intervalo dydis labai priklauso nuo įvairių duomenų savybių, tokių kaip taškų tankumas, daugdaros geometrija. LLE rezultatų priklausomybė nuo taškų tankumo parodyta 5.3 paveiksle. Buvo tiriama dvimatė S-formos daugdara. Vieną kartą joje parinkta 1000 taškų, o kitą kartą – 2000 taškų. Abiem atvejais gautos projekcijos plokštumoje, kai artimiausių kaimynų skaičius $k = 50$. Iš 5.3b paveikslo matyti, kad LLE metodu sėkmingai atskleista S-formos

daugdaros struktūra, kai $m = 2000$. Tuo tarpu, kai taškų skaičius $m = 1000$, S-formos daugdaros „išvynioti" nepavyko (5.3a pav.).



a) $m = 1000$ , $k = 50$ b) $m = 2000$ , $k = 50$

**5.3 pav.** S-formos daugdaros taškų projekcijos plokštumoje, gautos LLE metodu

Jei daugdaros struktūra iš anksto yra žinoma, tai galime pasinaudoti ekspertiniu įvertinimu ir pasakyti, su kokiomis *k* reikšmėmis gaunamos teisingos (tinkamos), t. y. atskleidžiančios daugdaros globalią struktūrą, jos taškų projekcijos. Bet ką galime pasakyti apie projekcijos, apskaičiuotos naudojant tam tikrą *k* reikšmę, patikimumą, kai daugdaros struktūra nėra iš anksto aiški? Taigi, norint įvertinti gautas projekcijas, būtina naudoti kiekybinius skaitinius kriterijus (matus). Automatinis artimiausių kaimynų skaičiaus parinkimas pasiūlytas straipsnyje (Kouropteva *et al*. 2002).

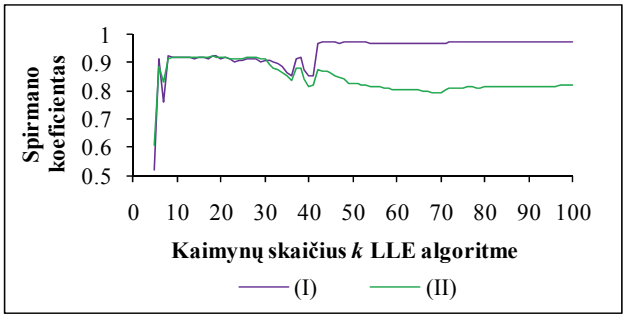
**Naujas būdas tinkamam artimiausių kaimynų skaičiaus intervalui parinkti**

Iš 5.2 paveikslo matome, kad nebūtina ieškoti optimalaus artimiausių kaimynų skaičiaus, bet pakanka rasti tinkamo artimiausių kaimynų skaičiaus intervalą. Šiame skyrelyje pasiūlytas naujas būdas šiai problemai išspręsti. Norint kiekybiškai įvertinti topologijos išlaikymą po transformacijos į mažesnio matavimo erdvę, dažnai yra skaičiuojamas Spirmano koeficientas $\rho_{Sp}$ (*Spearman's rho*) (Siegel and Castellan 1988). Detaliau šis matas aprašytas 7.1.1 skyrelyje. Geriausia Spirmano koeficiento reikšmė lygi 1.

Spirmano koeficiento skaičiavime naudojami atstumai ir plokštumoje, ir daugiamatėje erdvėje. Kyla klausimas: kokius atstumus tikslinga vertinti skaičiuojant Spirmano koeficientą: Euklido ar geodezinius? Euklido atstumai paprastai vartojami plokštumoje. Daugiamatėje erdvėje gali būti naudojami tiek Euklido, tiek ir geodeziniai atstumai. Straipsnyje (Aggarwal *et al*. 2001) teigiama, kad Euklido atstumas netinka, norint rasti trumpiausią atstumą tarp

taškų, neišeinant už daugdaros ribų. Straipsnyje (Tenenbaum *et al*. 2000) teigiama, jog norint išlaikyti daugdaros globalią struktūrą, būtina naudoti geodezinius atstumus. Kadangi nagrinėsime tik dvimačių daugdarų taškus, tai tolesniuose eksperimentuose plokštumoje vertinsime tik Euklido atstumus.

Tyrimas atliktas su S-formos daugdaros taškais $(m=1000, n=3)$. LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [5;100]$. Kiekvieną kartą buvo apskaičiuojamas Spirmano koeficientas (5.4 pav.). Gautos dvi Spirmano koeficiento priklausomybės nuo $k$: (I) erdvėje vertinami Euklido atstumai, (II) erdvėje vertinami geodeziniai atstumai. Nagrinėkime atvejį, kai artimiausių kaimynų skaičius $k=100$. Matome, kad, vertinant erdvėje Euklido atstumus, Spirmano koeficiento reikšmė yra arti 1 $(\approx 0{,}97)$, o vertinant erdvėje geodezinius atstumus Spirmano koeficiento reikšmė mažesnė $(\approx 0{,}82)$. Kai $k=100$, labai gerai išlaikyti Euklido atstumai, tačiau suardoma daugdaros struktūra (5.2 pav., $k=100$), o mes siekiame ją išsaugoti. Šis eksperimentas patvirtina faktą, kad erdvėje būtina vertinti geodezinius atstumus, todėl tolesniuose eksperimentuose erdvėje bus vertinami tik geodeziniai atstumai.

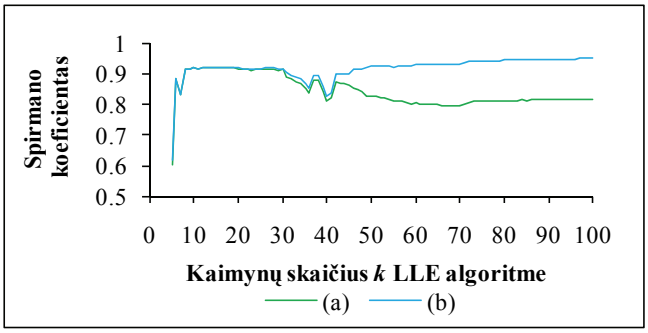



**5.4 pav.** Spirmano koeficiento priklausomybės nuo *k*, gautos transformavus į plokštumą S-formos daugdaros taškus LLE metodu
(I) trimatėje erdvėje vertinami Euklido atstumai,
(II) trimatėje erdvėje vertinami geodeziniai atstumai

Geodezinių atstumų skaičiavimo algoritme yra vienas parenkamas parametras – artimiausių kaimynų skaičius, reikalingas kaimynystės grafui sudaryti. Jį žymėsime $k_{geod}$. Toks pat parametras – artimiausių kaimynų skaičius $k$ yra ir LLE algoritme. Kokia turi būti $k_{geod}$ reikšmė, skaičiuojant geodezinius atstumus? Ar $k_{geod}$ turi sutapti su LLE algoritme pasirinktu artimiausių kaimynų skaičiumi $k$?

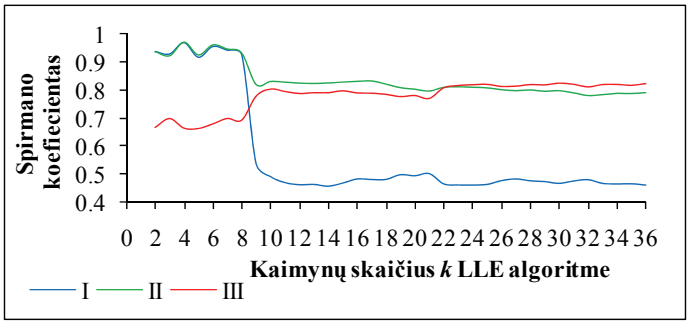
5.5 paveiksle gautos dvi Spirmano koeficiento priklausomybės nuo $k$: (a) skaičiuojant erdvėje geodezinius atstumus, fiksuojamas labai mažas

artimiausių kaimynų skaičius, pvz., $k_{geod}=10$, (b) skaičiuojant geodezinius atstumus, artimiausių kaimynų skaičius kinta kaip ir LLE algoritme, t. y. $k_{geod}=k$. Kai $k=100$, Spirmano koeficiento reikšmė, pagal (a) priklausomybę, yra pakankamai maža ($\approx 0{,}82$), nusileidusi priklausomybė pakyla nežymiai. Iš to seka, kad atstumai blogai išlaikomi, atvaizdavimas neatspindi globalios struktūros. Tačiau iš (b) priklausomybės matome, kad Spirmano koeficiento reikšmė arčiau 1 ($\approx 0{,}95$). Iš to sektų, kad LLE rezultatas yra gana geras. Tačiau akivaizdu, kad vizualizavus šiuos duomenis LLE algoritmu, kai $k=100$, gautas atvaizdavimas neatskleidžia daugdaros globalios struktūros (5.2 pav., $k=100$), nors Spirmano koeficiento reikšmė ir yra arti 1. Priežastis tokia: kai, skaičiuojant geodezinius atstumus erdvėje, artimiausių kaimynų $k_{geod}$ parenkama labai daug, tai suardoma netiesinės daugdaros struktūra, t. y. taško kaimynais erdvėje (grafe) gali tapti taškai, kurie sutinkami pereinant skersai daugdaros (randant kaimynus, skaičiuojami Euklido atstumai). 5.1 paveiksle taško, pažymėto juodu skrituliu, kaimynai yra apibrėžto rutulio viduje. Šiuo atveju ir LLE algoritme artimiausių kaimynų yra tiek pat, kiek ir geodezinių atstumų skaičiavimo algoritme: $k=k_{geod}$ (LLE algoritme kaimynai randami skaičiuojant Euklido atstumus). Dėl to tolimi taškai ant daugdaros traktuojami kaip artimi tiek erdvėje, tiek plokštumoje. Dėl šios priežasties, Spirmano koeficiento reikšmė didėja didinant artimiausių kaimynų skaičių. Tinkamos projekcijos 5.2 paveiksle yra gaunamos, kai (a) priklausomybė 5.5 paveiksle įgyja didžiausias (geriausias) reikšmes. Todėl Spirmano koeficientas su fiksuota gana maža $k_{geod}$ reikšme gali būti naudojamas kaip kriterijus daugdaros taškų vizualizavimo kokybei įvertinti.




**5.5 pav.** Spirmano koeficiento priklausomybės nuo *k*, gautos transformavus S-formos daugdaros taškus į plokštumą LLE metodu. Trimatėje erdvėje buvo vertinami geodeziniai atstumai, kai (a) $k_{geod}=10$, (b) $k_{geod}=k$

Antras tyrimas atliktas su nespalvotų ančiuko paveikslėlių duomenimis ($m = 72, n = 16384$) (žr. 2.1 skyrelį). LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [2; 36]$. Kiekvieną kartą buvo apskaičiuojamas Spirmano koeficientas. 5.6 paveiksle parodytos trys Spirmano koeficiento priklausomybės nuo $k$: (I), skaičiuojant erdvėje geodezinius atstumus, fiksuojamas labai mažas artimiausių kaimynų skaičius, pvz., $k_{geod} = 2$, (II), skaičiuojant geodezinius atstumus, artimiausių kaimynų skaičius kinta kaip ir LLE algoritme, t. y. $k_{geod} = k$, (III) – erdvėje vertinami Euklido atstumai. Iš 5.6 paveikslo, matome, kad didžiausias Spirmano koeficiento reikšmes, t. y. $0{,}91 \le \rho_{Sp} \le 0{,}97$, kai $k \in [2; 8]$, įgyja (I) ir (II) priklausomybės, o (III) priklausomybės $\rho_{Sp}$ reikšmės, kai $k \in [2; 8]$, daug mažesnės: $0{,}66 \le \rho_{Sp} \le 0{,}7$. Kai $k \ge 9$, (I) priklausomybės Spirmano koeficiento reikšmės smarkiai sumažėja ($\rho_{Sp} \approx 0{,}54$, kai $k = 9$), (II) priklausomybės – truputį mažiau sumažėja ($\rho_{Sp} \approx 0{,}82$, kai $k = 9$), o (III) priklausomybės reikšmės, priešingai, padidėja. Projekcijos, gautos vizualizavus šiuos duomenis LLE metodu, pavaizduotos 5.7 paveiksle. Kadangi objektas palaipsniui buvo apsuktas 360° kampu, tai tikėtina, kad teisingas atvaizdavimas yra gautas a) atveju, kai $k \in [2; 8]$ (plokštumoje taškai išsidėstę ratu). Iš to seka, kad teisingą rezultatą duoda (I) ir (II) priklausomybės, ypač ryškų skirtumą tarp sprendinių atspindi (I) priklausomybė.

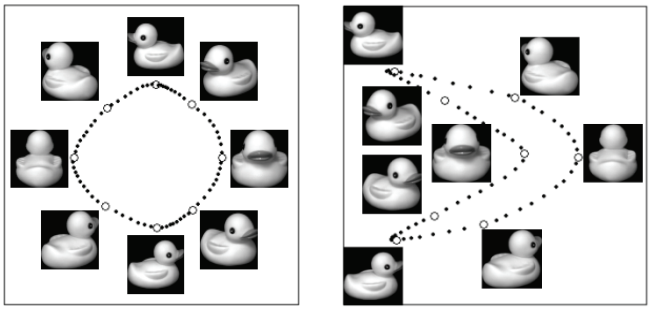



**5.6 pav.** Spirmano koeficiento priklausomybės nuo $k$, gautos transformavus į plokštumą besisukančio ančiuko paveikslėlius atitinkančius 16384-matės erdvės taškus LLE metodu:

(I) daugiamatėje erdvėje skaičiuojami geodeziniai atstumai, $k_{geod} = 2$;

(II) daugiamatėje erdvėje skaičiuojami geodeziniai atstumai, $k_{geod} = k$;

(III) daugiamatėje erdvėje skaičiuojami Euklido atstumai



a) $k = 4$ b) $k = 9$

**5.7 pav.** LLE metodu, naudojant $k$ artimiausių kaimynų, vizualizuoti besisukančio ančiuko paveikslėlių duomenys. Didesni apkritimai žymi greta esančius paveikslėlius

## 5.2.2. Reguliarizacijos parametro parinkimas LLE algoritme

Šiame skyrelyje atskleista LLE algoritmo priklausomybė nuo reguliarizacijos parametro bei pateiktas naujas algoritmas lokaliajai Gramo matricai reguliarizuoti. Pasiūlytas algoritmas detaliai išnagrinėtas teoriškai ir eksperimentiškai realizuotas su trimis dirbtinėmis duomenų aibėmis. Šio algoritmo rezultatai palyginti su rezultatais, gautais taikant reguliarizacijos algoritmą, pateiktą straipsnyje (Roweis and Saul 2000).

### 5.2.2.1. Naujas reguliarizacijos algoritmas LLE metodui

Šiame skyrelyje pasiūlytas naujas lokaliosios Gramo matricos $C^i$ reguliarizacijos algoritmas. Prieš aprašant algoritmą, pirmiausia pateiksime būtinų teorinių žinių iš matricų teorijos. Tegu turime $m$ vektorių $X_1, X_2, ..., X_m \in R^n$, $X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $i = \overline{1,m}$ ir sudarykime matricą $X$, kurios stulpeliai yra šie vektoriai. Matrica $C = X^T X$, kurios elementai yra vektorių $X_i$ ir $X_j$, $i, j = \overline{1,m}$, skaliarinės sandaugos, vadinama Gramo matrica, o jos determinantas – Gramo determinantu.

**Teorema**. Jei prie kvadratinės matricos pagrindinės įstrižainės elementų pridėsime konstantą $c$, tai ir šios matricos tikrinės reikšmės padidės $c$, ir atvirkščiai, jei visas kvadratinės matricos tikrines reikšmes padidinsime konstanta $c$, tai tos matricos pagrindinės įstrižainės elementai padidės $c$.

Pažymėkime:

- $A$ yra kvadratinė matrica, sudaryta iš $k$ eilučių ir $k$ stulpelių,

- $\lambda_i, i = \overline{1,n}$ yra matricos $A$ tikrinės reikšmės,
- $B$ – diagonalioji matrica, sudaryta iš $k$ eilučių ir $k$ stulpelių, kurios elementai $b_{ij} = \begin{cases} c, \text{ jei } i = j, \\ 0, \text{ jei } i \neq j. \end{cases}$

Tuomet $\forall \lambda_i, i = \overline{1,n}$ teisingos lygybės:

$$\lambda_i = \max_{x \in R^n, |\vec{x}|=1} \left(xAx^T\right),$$

$$\hat{\lambda}_i = \max_{x \in R^n, |\vec{x}|=1} \left(x(A+B)x^T\right) = \max_{x \in R^n, |\vec{x}|=1} \left(xAx^T + xBx^T\right).$$

Kadangi bet kokiam $x \in R^n$, tenkinančiam sąlygą $|\vec{x}| = 1$, galioja $xBx^T = c$, tuomet $\hat{\lambda}_i = \max_{x \in R^n, |\vec{x}|=1} \left(xAx^T + c\right) = \max_{x \in R^n, |\vec{x}|=1} \left(xAx^T\right) + c = \lambda_i + c.$

LLE algoritme $C^i$ yra lokalioji Gramo matrica, sudaryta iš $k$ eilučių ir $k$ stulpelių, kurios elementai apskaičiuojami pagal (5.3) formulę. Kai $k > n$, Gramo determinantas $|C^i| = 0$, t. y. matrica $C^i$ yra singuliari. Tai seka iš tiesinės algebros elementarių faktų (Gantmacher 1988): bet kurie $n+1$ vektoriai $n$-matėje erdvėje tiesiškai priklausomi; Gramo determinantas lygus nuliui tada ir tik tada, kai vektoriai $X_1, X_2, ..., X_m \in R^n$ yra tiesiškai priklausomi.

Tačiau dėl skaičiavimo paklaidų $|C^i|$ gali būti ne lygus, bet labai arti nulio, t. y. Gramo matrica yra beveik singuliari. Taigi negalima išspręsti (5.7) tiesinių lygčių sistemos arba svoriai $w_{ij}$ nėra vienareikšmiškai nustatomi. Taigi susiduriame su „blogai-sąlygotu“ uždaviniu (*ill-posed problem*). Norint šią sistemą išspręsti, reikia pakeisti matricą $C^i$ taip, kad pakeistos matricos determinantas būtų pakankamai didelis. Toks matricos $C^i$ pokytis yra tirtas eksperimentinėje dalyje 5.2.2.3 skyrelyje.

**Apibrėžimas.** „Gerai-sąlygotas“ uždavinys (*well-posed problem*) – tai uždavinys, pasižymintis šiomis savybėmis: 1) sprendinys egzistuoja; 2) jis yra vienintelis; 3) sprendinys tolygiai priklauso nuo pradinių duomenų, t. y. sprendinys nesikeičia šuoliu nuo pradinių duomenų. Uždavinys, kuris nėra „gerai-sąlygotas“ (*well-posed problem*), vadinamas „blogai-sąlygotu“ uždaviniu (*ill-posed problem*).

Matricos determinantas yra lygus šios matricos tikrinių reikšmių sandaugai: $|C^i| = \prod_{j=1}^{k} \lambda_j$. Atliksime matricos $C^i$ reguliarizaciją, pridėdami prie jos pagrindinės įstrižainės elementų mažą teigiamą konstantą $\varepsilon$. Remiantis teorema, tikrinės reikšmės taip pat padidės $\varepsilon$. Tuomet reguliarizuotos matricos determinantas bus $D = \prod_{j=1}^{k} (\lambda_j + \varepsilon)$.

Tegu *D* reikšmė iš anksto fiksuota ($D > |C^i|$), o matricos $C^i$ rangas yra *r*, t. y. matrica $C^i$ turi *r* tikrinių reikšmių, nelygių nuliui. Tada galime apytiksliai apskaičiuoti $\varepsilon$, kuriam esant gausime tą determinanto reikšmę:

$$\varepsilon \approx \sqrt[k-r]{\frac{D}{\prod_{j=1}^{r} \lambda_j}}. \tag{5.12}$$

*Apibendrinta siūlomo reguliarizacijos algoritmo schema tokia:*

Algoritmo pradžioje nustatoma norima determinanto *D* reikšmė. Po to, kiekvienam duomenų aibės taškui vykdomi tokie žingsniai:

1. Randami matricos $C^i$, sudarytos iš *k* eilučių ir *k* stulpelių, tikriniai vektoriai $f_j = (f_{j1},...,f_{jk})$ ir tikrinės reikšmės $\lambda_j$, $j = \overline{1,k}$.
2. Apskaičiuojamas matricos $C^i$ rangas *r*.
3. Pagal apytikslę (5.12) formulę randamas $\varepsilon$.
4. Visos tikrinės reikšmės padidinamos $\varepsilon$: $\lambda^*_j = \lambda_j + \varepsilon$, $j = \overline{1,k}$.
5. Atstatoma Gramo matrica (Dzemyda 2001):

   a) $q_{ij} = \sqrt{\lambda^*_j} f_{ji}$, $i, j = \overline{1,k}$;

   b) $C^i = Q^T Q$, kur $Q = (Q_1, Q_2, ..., Q_k) = \begin{pmatrix} q_{11}\ q_{21} \cdots q_{k1} \\ q_{12}\ q_{22} \cdots q_{k2} \\ \cdots\ \cdots\cdots\ \cdots \\ q_{1k}\ q_{2k} \cdots q_{kk} \end{pmatrix}$.

### 5.2.2.2. Reguliarizacijos algoritmų savybės

Pažymėkime reguliarizacijos algoritmą, pasiūlytą (Roweis and Saul 2000), R1, o naująjį (pasiūlytą 5.2.2.1 skyrelyje) – R2. Šiame skyrelyje pateikiama lyginamoji šių reguliarizacijos algoritmų analizė. Tyrimui pasirinkta 1000 3-matės erdvės taškų, priklausančių netiesinei dvimatei S-formos daugdarai (5.1 pav.).

Lokalioji Gramo matrica $C^i$ yra sudaroma kiekvieno taško aplinkoje, t. y. kiekvienam duomenų taškui ir jo artimiausiems kaimynams. Jei duomenų aibę sudaro 1000 taškų, tai reikės sudaryti ir reguliarizuoti 1000 Gramo matricų. Žemiau pateikti ir palyginti reguliarizacijos algoritmų R1 ir R2 rezultatai, gauti ištyrus tik vieną Gramo matricą, kai $k=10,\ t=0{,}01$ (R1 algoritme), $D=10^{-20}$ (R2 algoritme).

Gramo matrica $C^i$ iki reguliarizacijos yra tokia:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0,008** | 0,006 | 0,022 | 0,024 | 0,013 | 0,002 | -0,020 | 0,039 | -0,028 | 0,019 |
| 0,006 | **0,008** | 0,023 | 0,023 | 0,026 | 0,020 | 0,002 | 0,036 | -0,005 | 0,038 |
| 0,022 | 0,023 | **0,073** | 0,074 | 0,067 | 0,042 | -0,020 | 0,120 | -0,042 | 0,100 |
| 0,024 | 0,023 | 0,074 | **0,076** | 0,057 | 0,027 | -0,041 | 0,126 | -0,065 | 0,085 |
| 0,013 | 0,026 | 0,067 | 0,057 | **0,109** | 0,109 | 0,072 | 0,086 | 0,060 | 0,161 |
| 0,002 | 0,020 | 0,042 | 0,027 | 0,109 | **0,130** | 0,123 | 0,034 | 0,123 | 0,162 |
| -0,020 | 0,002 | -0,020 | -0,041 | 0,072 | 0,123 | **0,181** | -0,077 | 0,204 | 0,107 |
| 0,039 | 0,036 | 0,120 | 0,126 | 0,086 | 0,034 | -0,077 | **0,211** | -0,115 | 0,129 |
| -0,028 | -0,005 | -0,042 | -0,065 | 0,060 | 0,123 | 0,204 | -0,115 | **0,236** | 0,090 |
| 0,019 | 0,038 | 0,100 | 0,085 | 0,161 | 0,162 | 0,107 | 0,129 | 0,090 | **0,239** |

Gramo matricos savybės, dar neatlikus reguliarizacijos, yra tokios:

a) rangas $r=3$,

b) tikriniai vektoriai:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0,021** | **0,113** | **0,076** | 0,202 | -0,077 | -0,250 | -0,083 | -0,133 | 0,919 | 0,048 |
| **0,078** | **0,080** | **0,113** | 0,202 | -0,070 | -0,088 | -0,321 | -0,059 | -0,086 | -0,899 |
| **0,184** | **0,296** | **0,006** | 0,198 | 0,518 | 0,165 | 0,041 | 0,726 | 0,114 | -0,042 |
| **0,137** | **0,334** | **-0,053** | 0,280 | 0,449 | -0,190 | -0,458 | -0,443 | -0,237 | 0,297 |
| **0,383** | **0,108** | **0,262** | 0,219 | 0,130 | 0,403 | 0,590 | -0,431 | 0,025 | -0,110 |
| **0,426** | **-0,072** | **0,349** | 0,280 | -0,303 | -0,596 | 0,164 | 0,232 | -0,240 | 0,168 |
| **0,363** | **-0,396** | **-0,158** | 0,383 | -0,291 | 0,494 | -0,413 | 0,088 | 0,046 | 0,174 |
| **0,197** | **0,564** | **-0,662** | 0,035 | -0,414 | -0,024 | 0,165 | 0,014 | -0,049 | -0,029 |
| **0,348** | **-0,514** | **-0,530** | -0,104 | 0,391 | -0,314 | 0,172 | -0,086 | 0,081 | -0,169 |
| **0,568** | **0,161** | **0,210** | -0,718 | -0,034 | 0,074 | -0,275 | -0,008 | 0,098 | 0,013 |

c) tikrinės reikšmės:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0,695 | 0,569 | 0,006 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Gramo matrica $C^i$, atlikus reguliarizaciją R1:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0,021** | 0,006 | 0,022 | 0,024 | 0,013 | 0,002 | -0,020 | 0,039 | -0,028 | 0,019 |
| 0,006 | **0,021** | 0,023 | 0,023 | 0,026 | 0,020 | 0,002 | 0,036 | -0,005 | 0,038 |
| 0,022 | 0,023 | **0,086** | 0,074 | 0,067 | 0,042 | -0,020 | 0,120 | -0,042 | 0,100 |
| 0,024 | 0,023 | 0,074 | **0,089** | 0,057 | 0,027 | -0,041 | 0,126 | -0,065 | 0,085 |
| 0,013 | 0,026 | 0,067 | 0,057 | **0,122** | 0,109 | 0,072 | 0,086 | 0,060 | 0,161 |
| 0,002 | 0,020 | 0,042 | 0,027 | 0,109 | **0,143** | 0,123 | 0,034 | 0,123 | 0,162 |
| -0,020 | 0,002 | -0,020 | -0,041 | 0,072 | 0,123 | **0,194** | -0,077 | 0,204 | 0,107 |
| 0,039 | 0,036 | 0,120 | 0,126 | 0,086 | 0,034 | -0,077 | **0,224** | -0,115 | 0,129 |
| -0,028 | -0,005 | -0,042 | -0,065 | 0,060 | 0,123 | 0,204 | -0,115 | **0,249** | 0,090 |
| 0,019 | 0,038 | 0,100 | 0,085 | 0,161 | 0,162 | 0,107 | 0,129 | 0,090 | **0,252** |

Gramo matricos savybės, atlikus reguliarizaciją R1, yra tokios:

a) rangas $r = 10$,

b) tikriniai vektoriai:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0,021** | **0,113** | **0,076** | -0,209 | 0,281 | -0,658 | 0,531 | 0,303 | -0,177 | -0,145 |
| **0,078** | **0,080** | **0,113** | -0,224 | 0,098 | 0,619 | 0,696 | -0,075 | 0,205 | -0,021 |
| **0,184** | **0,296** | **0,006** | 0,075 | 0,256 | -0,234 | -0,009 | -0,810 | 0,189 | -0,248 |
| **0,137** | **0,334** | **-0,053** | 0,036 | 0,271 | 0,331 | -0,239 | 0,249 | -0,404 | -0,633 |
| **0,383** | **0,108** | **0,262** | 0,516 | 0,170 | -0,045 | -0,045 | 0,379 | 0,575 | -0,005 |
| **0,426** | **-0,072** | **0,349** | 0,412 | -0,235 | 0,023 | 0,213 | -0,170 | -0,596 | 0,192 |
| **0,363** | **-0,396** | **-0,158** | -0,136 | -0,542 | -0,112 | 0,093 | -0,001 | 0,181 | -0,566 |
| **0,197** | **0,564** | **-0,662** | 0,124 | -0,323 | -0,033 | 0,153 | 0,093 | -0,001 | 0,227 |
| **0,348** | **-0,514** | **-0,530** | 0,072 | 0,539 | 0,066 | 0,024 | -0,025 | -0,093 | 0,158 |
| **0,568** | **0,161** | **0,210** | -0,652 | 0,018 | -0,008 | -0,312 | 0,056 | 0,016 | 0,287 |

c) tikrinės reikšmės:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0,708 | 0,582 | 0,019 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 | 0,013 |

Taigi atlikus reguliarizaciją R1,

a) visos tikrinės reikšmės padidėjo $\varepsilon = 0{,}013$;

b) reguliarizuotos matricos pagrindinės įstrižainės elementai padidėjo $\varepsilon = 0{,}013$;

c) tikriniai vektoriai, atitinkantys nelygias nuliui pradines tikrines reikšmes, liko nepakitę.

Iš (5.12) formulės seka, kad rekomenduojama $\varepsilon$ reikšmė yra 0,003.

Gramo matrica $C^i$, atlikus reguliarizaciją R2:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0,011** | 0,006 | 0,022 | 0,024 | 0,013 | 0,002 | -0,020 | 0,039 | -0,028 | 0,019 |
| 0,006 | **0,011** | 0,023 | 0,023 | 0,026 | 0,020 | 0,002 | 0,036 | -0,005 | 0,038 |
| 0,022 | 0,023 | **0,076** | 0,074 | 0,067 | 0,042 | -0,020 | 0,120 | -0,042 | 0,100 |
| 0,024 | 0,023 | 0,074 | **0,080** | 0,057 | 0,027 | -0,041 | 0,126 | -0,065 | 0,085 |
| 0,013 | 0,026 | 0,067 | 0,057 | **0,112** | 0,109 | 0,072 | 0,086 | 0,060 | 0,161 |
| 0,002 | 0,020 | 0,042 | 0,027 | 0,109 | **0,133** | 0,123 | 0,034 | 0,123 | 0,162 |
| -0,020 | 0,002 | -0,020 | -0,041 | 0,072 | 0,123 | **0,184** | -0,077 | 0,204 | 0,107 |
| 0,039 | 0,036 | 0,120 | 0,126 | 0,086 | 0,034 | -0,077 | **0,214** | -0,115 | 0,129 |
| -0,028 | -0,005 | -0,042 | -0,065 | 0,060 | 0,123 | 0,204 | -0,115 | **0,239** | 0,090 |
| 0,019 | 0,038 | 0,100 | 0,085 | 0,161 | 0,162 | 0,107 | 0,129 | 0,090 | **0,242** |

Gramo matricos savybės, atlikus reguliarizaciją R2, yra tokios:

a) rangas $r = 10$,

b) tikriniai vektoriai:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0,021** | **0,113** | **0,076** | -0,007 | -0,026 | 0,028 | 0,815 | -0,500 | -0,091 | -0,240 |
| **0,078** | **0,080** | **0,113** | 0,046 | -0,039 | -0,325 | 0,383 | 0,796 | -0,076 | -0,282 |
| **0,184** | **0,296** | **0,006** | -0,261 | 0,031 | 0,469 | -0,157 | 0,087 | -0,711 | -0,228 |
| **0,137** | **0,334** | **-0,053** | -0,323 | 0,768 | -0,305 | -0,100 | -0,099 | 0,186 | -0,162 |
| **0,383** | **0,108** | **0,262** | -0,208 | -0,273 | 0,370 | -0,097 | 0,033 | 0,602 | -0,382 |
| **0,426** | **-0,072** | **0,349** | 0,063 | -0,272 | -0,597 | -0,277 | -0,292 | -0,270 | -0,155 |
| **0,363** | **-0,396** | **-0,158** | 0,626 | 0,385 | 0,217 | -0,028 | -0,003 | -0,036 | -0,313 |
| **0,197** | **0,564** | **-0,662** | 0,292 | -0,296 | -0,134 | -0,051 | -0,045 | 0,087 | -0,029 |
| **0,348** | **-0,514** | **-0,530** | -0,543 | -0,100 | -0,087 | 0,135 | 0,031 | -0,031 | 0,043 |
| **0,568** | **0,161** | **0,210** | 0,080 | 0,117 | 0,124 | 0,217 | 0,101 | 0,008 | 0,718 |

c) tikrinės reikšmės:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0,698 | 0,572 | 0,009 | 0,003 | 0,003 | 0,003 | 0,003 | 0,003 | 0,003 | 0,003 |

Taigi atlikus reguliarizaciją R2,

a) visos tikrinės reikšmės padidėjo $\varepsilon = 0{,}003$;

b) reguliarizuotos matricos pagrindinės įstrižainės elementai padidėjo $\varepsilon = 0{,}003$;

c) tikriniai vektoriai, atitinkantys nelygias nuliui pradines tikrines reikšmes, liko nepakitę.

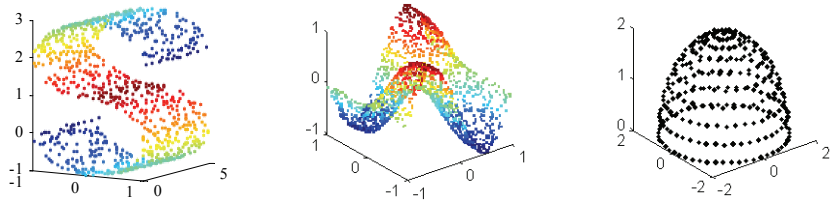
Atlikus šią reguliarizacijos algoritmų R1 ir R2 analizę, galime teigti, kad abu algoritmai pateikia panašią galimybę analizei, kiekvienas iš jų turi po vieną valdymo parametrą: *t* R1 atveju ir *D* R2 atveju. Tačiau parametro *D* privalumas lyginant su *t* yra tai, kad parametras *D* turi realią prasmę (jis yra reguliarizuotos Gramo matricos determinantas), o *t* yra tiktai tam tikras daugiklis.

Kitame skyrelyje bus pateikti eksperimentinių tyrimų su keliomis dirbtinėmis duomenų aibėmis rezultatai. Įvertinus gautas duomenų projekcijas,

naudojant Spirmano koeficientą, bus bandoma nustatyti šių reguliarizacijos algoritmų priklausomybę nuo valdymo parametrų. Taip pat nustatysime, kokias reikėtų parinkti valdymo parametrų $D$ ir $t$ reikšmes, kad rezultatas būtų geriausias, t. y. Spirmano koeficiento reikšmė būtų didžiausia (geriausia).

### 5.2.2.3. Reguliarizacijos algoritmų eksperimentinis tyrimas

Tyrimams atlikti buvo naudojamos šios duomenų aibės: 1000 3-matės erdvės taškų, išsidėsčiusių ant S-formos daugdaros (5.8a pav.), 2000 3-matės erdvės taškų, esančių ant daugdaros „Dvynės viršukalnės“ (*„Twin peaks“*) (5.8b pav.) ir 294 3-matės erdvės taškai, priklausantys pussferei (5.8c pav.).



a) S-formos daugdaros taškai

b) daugdaros „Dvynės viršukalnės“ taškai

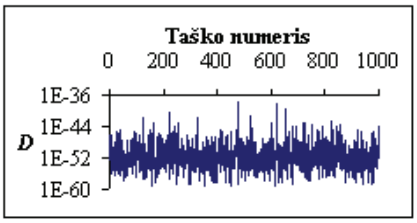
c) pussferės taškai

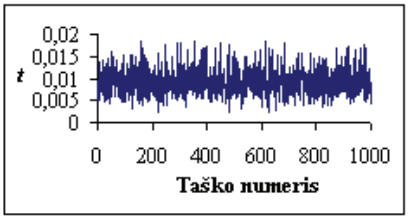
**5.8 pav.** Netiesinių dvimačių daugdarų taškai

Naujasis reguliarizacijos algoritmas R2 skiriasi nuo pasiūlyto straipsnyje (Roweis and Saul 2000) algoritmo R1 valdymo parametrais: naujajame algoritme valdymo parametras yra reguliarizuoto Gramo determinanto $D$ reikšmė, o algoritme R1 – parametro $t$ reikšmė. Abiejų reguliarizacijos metodų R1 ir R2 esmė yra tai, kad randama reguliarizacijos parametro $\varepsilon$ reikšmė, kuri pridedama prie matricos $C^i$ pagrindinės įstrižainės elementų, o tuo pačiu ir prie jos tikrinių reikšmių. R1 algoritme fiksuojama parametro $t$ reikšmė ir randamas $\varepsilon$ (parametro $D$ reikšmė gali būti apskaičiuota) (5.9a pav.). R2 algoritme, priešingai, fiksuojama parametro $D$ reikšmė ir randamas $\varepsilon$ (parametro $t$ reikšmė gali būti apskaičiuota) (5.9b pav.). Iš (5.10) formulės seka, kad $t = \varepsilon / \mathrm{tr}(C^i)$. Dėl labai mažų skaičių 5.9a paveiksle $y$ ašyje naudojama logaritminė skalė. Matome, kad R1 atveju determinanto $D$ reikšmė kinta labai dideliuose rėžiuose ir tai gali sukelti nestabilumų sprendžiant (5.7) tiesinių lygčių sistemą.

5.2.1 skyrelyje pasiūlėme būdą artimiausių kaimynų skaičiui nustatyti. Jame parodėme, jog kiekybinis matas – Spirmano koeficientas (Siegel and Castellan 1988) – yra tinkamas įvertinti daugiamačių duomenų topologijos išlaikymą, vizualizavus duomenis LLE metodu. Šiame skyrelyje taip pat naudojamas Spirmano koeficientas, siekiant įvertinti projekcijas, gautas

vizualizavus daugdarų taškus LLE metodu ir panaudojus abu reguliarizacijos algoritmus.

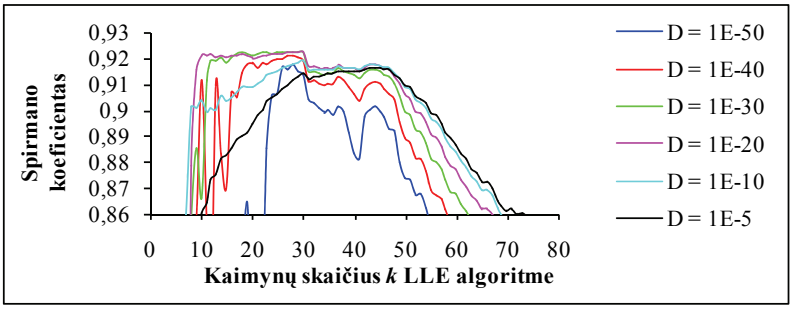
a) 


b) 


**5.9 pav.** Parametrų $D$ ir $t$ kitimas, esant a) $t = 10^{-3}$, $k = 26$ (algoritmas R1), b) $D = 10^{-30}$, $k = 26$ (algoritmas R2) (analizuota S-formos daugdara)

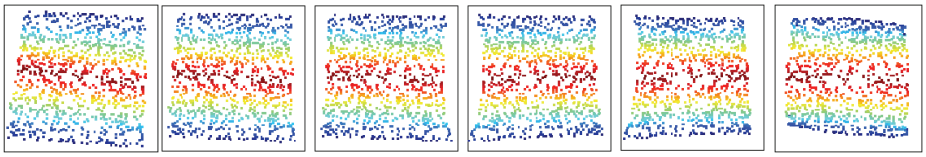
**Reguliarizacijos algoritmo R2 tyrimas**

Pirmas tyrimas atliktas su netiesinės dvimatės S-formos daugdaros taškais ($m = 1000, n = 3$) (5.10 pav.). LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [6; 100]$, kiekvieną kartą apskaičiuojant Spirmano koeficientą $\rho_{Sp}$. Gautos šešios Spirmano koeficiento priklausomybės nuo $k$, kai valdymo parametras – fiksuotas determinantas – įgyja tokias reikšmes: $D = \{10^{-50}, 10^{-40}, 10^{-30}, 10^{-20}, 10^{-10}, 10^{-5}\}$.




**5.10 pav.** Spirmano koeficiento priklausomybės nuo $k$, gautos transformavus S-formos daugdaros taškus į dvimatę erdvę LLE metodu, esant skirtingoms $D$ reikšmėms

Iš 5.10 paveikslo matome, kad didžiausia (geriausia) Spirmano koeficiento reikšmė yra $\approx 0{,}92$. Empiriškai nustatėme: esant $\rho_{Sp} \geq 0{,}915$, galima teigti, kad artimiausių kaimynų skaičius yra tinkamas, norint gauti teisingas (atskleidžiančias daugdaros globalią struktūrą) projekcijas (5.11 pav.).

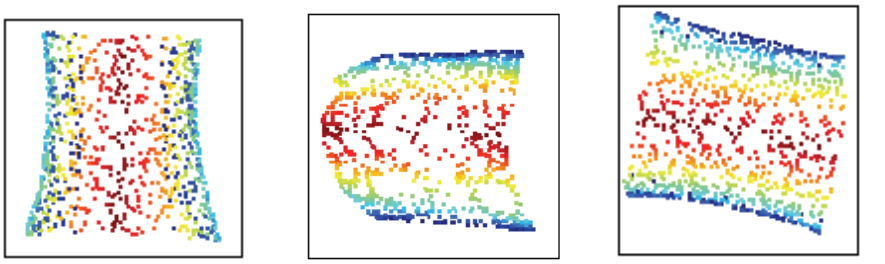


$D = 10^{-50}$ $D = 10^{-40}$ $D = 10^{-30}$ $D = 10^{-20}$ $D = 10^{-10}$ $D = 10^{-5}$

$k = 28$ $k = 28$ $k = 28$ $k = 30$ $k = 30$ $k = 43$

**5.11 pav.** Projekcijos plokštumoje, gautos vizualizavus S-formos daugdaros taškus LLE metodu, kai Spirmano koeficiento reikšmė yra geriausia ($\approx 0{,}92$)

Esant mažesnėms Spirmano koeficiento reikšmėms, gaunamos neteisingos projekcijos (5.12 pav.).



$D = 10^{-10}, k = 89, \rho_{Sp} \approx 0{,}89$ $D = 10^{-10}, k = 8, \rho_{Sp} \approx 0{,}90$ $D = 10^{-20}, k = 50, \rho_{Sp} \approx 0{,}91$

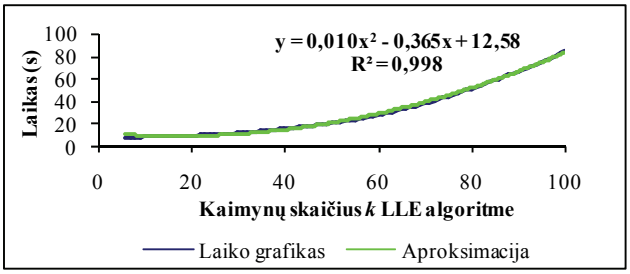
**5.12 pav.** Projekcijos plokštumoje, gautos vizualizavus S-formos daugdaros taškus LLE metodu, kai Spirmano koeficiento reikšmė nėra geriausia

Didžiausios Spirmano koeficiento reikšmės ($\rho_{Sp} > 0{,}92$), esant nedideliam artimiausių kaimynų skaičiui, gaunamos fiksavus determinantus, $D = 10^{-30}$ ir $D = 10^{-20}$. Be to, tinkama parametro $k$ reikšmė gali būti parenkama iš gana plataus intervalo. Didinant determinantą ($D = 10^{-10}$, $D = 10^{-5}$) arba mažinant jį ($D = 10^{-40}$, $D = 10^{-50}$), su mažesne parametro $k$ reikšme įgyjamos vis mažesnės Spirmano koeficiento reikšmės. Be to, tinkamo artimiausių kaimynų skaičiaus intervalas yra vis siauresnis ir pirma tinkama parametro $k$ reikšmė vis didesnė (5.1 lentelė).

Labai svarbu, kad didžiausia Spirmano koeficiento reikšmė būtų gaunama su mažesne $k$ reikšme, nes kuo didesnis artimiausių kaimynų skaičius, tuo didesnės LLE algoritmo vykdymo laiko sąnaudos. LLE algoritmo vykdymo laikas didėja polinomiškai didinant $k$ (5.13 pav.).

**5.1 lentelė.** Parametrų $k$ ir $D$ reikšmės, su kuriomis Spirmano koeficiento reikšmės gaunamos geriausios ( $\rho_{Sp} \approx 0{,}92$ ), transformavus S-formos daugdaros taškus į plokštumą LLE metodu

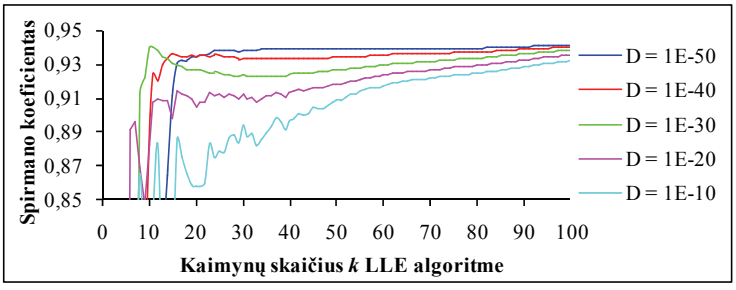
| $D$ | $10^{-50}$ | $10^{-40}$ | $10^{-30}$ | $10^{-20}$ | $10^{-10}$ | $10^{-5}$ |
|---|---|---|---|---|---|---|
| $\rho_{Sp} \approx 0{,}92$, kai $k$ priklauso intervalui | [26; 28] | [19; 30] | [12; 30] | [9; 47] | [25; 47] | [39; 47] |




**5.13 pav.** Laiko priklausomybė nuo $k$, gauta transformavus S-formos daugdaros taškus į plokštumą LLE metodu

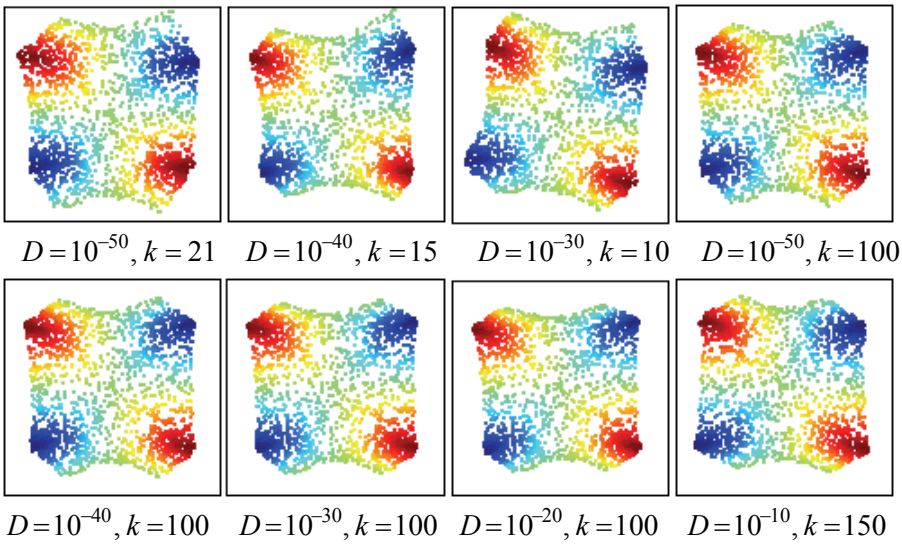
Taigi, vizualizuojant S-formos daugdaros taškus LLE metodu, mažiausiai sugaištama laiko ir gaunamas tinkamos projekcijos, kai $D = \{10^{-30}, 10^{-20}\}$.

Taip pat atliktas tyrimas su netiesinės dvimatės daugdaros „Dvynės viršukalnės“ taškais ( $m = 2000, n = 3$ ) (5.14 pav.). LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [5;100]$. Kiekvieną kartą buvo apskaičiuojamas Spirmano koeficientas $\rho_{Sp}$. Gautos penkios Spirmano koeficiento priklausomybės nuo $k$, kai valdymo parametras – fiksuotas determinantas – įgyja tokias reikšmes: $D = \{10^{-50}, 10^{-40}, 10^{-30}, 10^{-20}, 10^{-10}\}$.



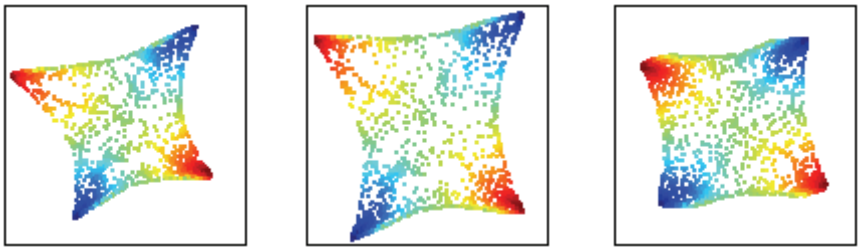

**5.14 pav.** Spirmano koeficiento priklausomybės nuo $k$, gautos transformavus daugdaros „Dvynės viršukalnės“ taškus į dvimatę erdvę LLE metodu, esant skirtingoms valdymo parametro $D$ reikšmėms

Iš 5.14 paveikslo matome, kad didžiausia (geriausia) Spirmano koeficiento reikšmė $\approx 0{,}94$. Empiriškai nustatėme: esant $\rho_{Sp} \geq 0{,}935$, galima teigti, kad artimiausių kaimynų skaičius yra tinkamas, norint gauti teisingas projekcijas (5.15 pav.).




**5.15 pav.** Projekcijos plokštumoje, gautos vizualizavus daugdaros „Dvynės viršukalnės“ taškus LLE metodu, kai su atitinkamomis parametrų $D$ ir $k$ reikšmėmis gauta Spirmano koeficiento reikšmė yra geriausia ($\approx 0{,}94$)

Esant mažesnėms Spirmano koeficiento reikšmėms, gaunamos neteisingos projekcijos (5.16 pav.).

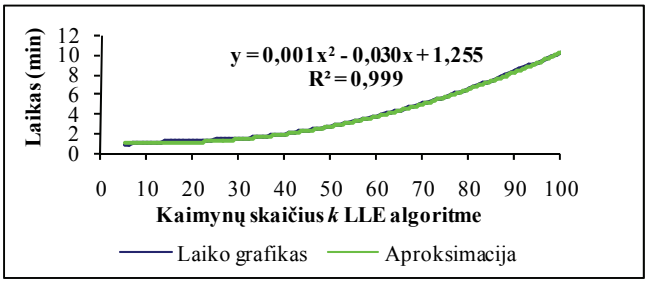



**5.16 pav.** Projekcijos plokštumoje, gautos vizualizavus daugdaros „Dvynės viršukalnės“ taškus LLE, kai Spirmano koeficiento reikšmė nėra geriausia

**5.2 lentelė.** Parametrų $k$ ir $D$ reikšmės, su kuriomis Spirmano koeficiento reikšmės gaunamos geriausios ( $\rho_{Sp} \approx 0{,}94$ ), transformavus daugdaros „Dvynės viršukalnės“ taškus į plokštumą LLE metodu

| $D$ | $10^{-50}$ | $10^{-40}$ | $10^{-30}$ | $10^{-20}$ | $10^{-10}$ |
|---|---|---|---|---|---|
| $\rho_{Sp} \approx 0{,}94$, jei $k$ priklauso intervalui | $[21; 100]$ | $\{15, 16, 21, 24, 25\} \cup [58; 100]$ | $[10; 12] \cup [86; 100]$ | $[99; 100]$ | $> 100$ |

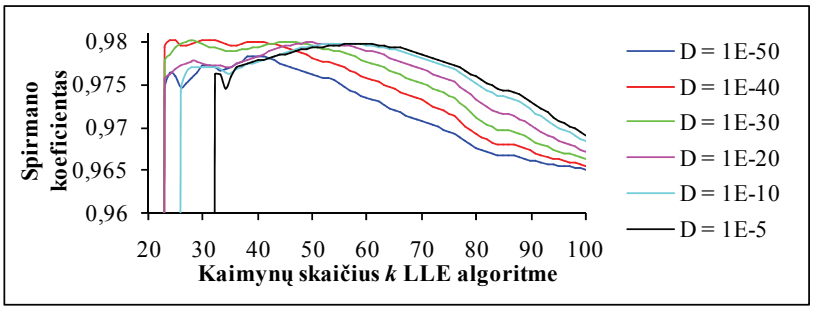
5.2 lentelėje pateikti tinkamo artimiausių kaimynų skaičiaus $k$ intervalai (juose $\rho_{Sp} \geq 0{,}935$) su įvairiomis $D$ reikšmėms. Iš šios lentelės ir 5.14 paveikslo pastebime, jog, didinant nustatytą determinanto $D$ reikšmę, Spirmano koeficiento reikšmės vis mažėja ( $k > 20$ ) ir tinkamo artimiausių kaimynų skaičiaus $k$ intervalai vis siaurėja. Pavyzdžiui, kai $D = 10^{-20}$, didžiausia Spirmano koeficiento reikšmė ( $\rho_{Sp} \approx 0{,}94$ ) pasiekiama, kai $k = 99$, o kai $D = 10^{-10}$ – tik kai $k > 100$. Labai svarbu, kad didžiausia Spirmano koeficiento reikšmė būtų gaunama su mažesne $k$ reikšme, nes kuo didesnis artimiausių kaimynų skaičius, tuo didesnės LLE algoritmo vykdymo laiko sąnaudos (5.17 pav.). Pavyzdžiui, kai $D = 10^{-30}$, teisingos daugdaros taškų projekcijos gali būti gaunamos ir kai $k = 10$, ir kai $k = 100$ (5.15 pav.), tačiau LLE algoritmo vykdymo laikas skiriasi 10 kartų.




**5.17 pav.** Laiko priklausomybė nuo $k$, gauta transformavus daugdaros „Dvynės viršukalnės“ taškus į plokštumą LLE metodu, kai $D = 10^{-30}$

Taigi vizualizuojant daugdaros „Dvynės viršukalnės“ taškus LLE metodu mažiausiai sugaištama laiko ir gaunamos teisingos projekcijos, kai $D = 10^{-50}, k = 21$, $D = 10^{-40}, k = 15$, $D = 10^{-30}, k = 10$.

Trečias tyrimas atliktas su pussferės taškais ( $m = 294, n = 3$ ) (5.18 pav.). LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [6;100]$, kiekvieną kartą apskaičiuojant Spirmano koeficientą. Gautos šešios Spirmano koeficiento priklausomybės nuo $k$, kai valdymo parametras – pasirinktas determinantas – įgyja reikšmes: $D = \{10^{-50}, 10^{-40}, 10^{-30}, 10^{-20}, 10^{-10}, 10^{-5}\}$.



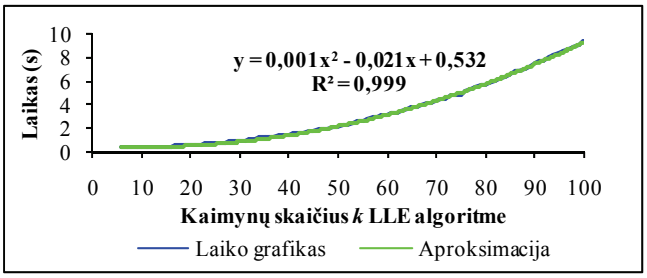

**5.18 pav.** Spirmano koeficiento priklausomybės nuo $k$, gautos transformrmavus pussferės taškus LLE metodu į dvimatę erdvę, esant skirtingoms parametro $D$ reikšmėms

Iš 5.18 paveikslo ir 5.3 lentelės matome, kad didžiausios Spirmano koeficiento reikšmės ( $\approx 0{,}98$ ) gali būti gaunamos fiksavus bet kurį iš išvardintų determinantų, tik būtinas mažiausias artimiausių kaimynų skaičius yra kitoks.

**5.3 lentelė.** Parametrų $k$ ir $D$ reikšmės, su kuriomis Spirmano koeficiento reikšmės gaunamos geriausios ( $\rho_{Sp} \approx 0{,}98$ ),transformrmavus pussferės taškus LLE metodu į plokštumą

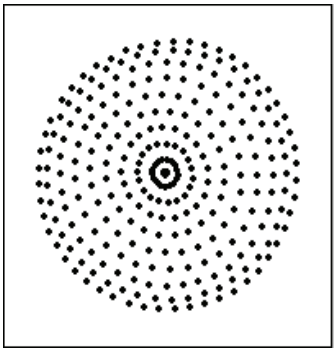
| $D$ | $10^{-50}$ | $10^{-40}$ | $10^{-30}$ | $10^{-20}$ | $10^{-10}$ | $10^{-5}$ |
|---|---|---|---|---|---|---|
| $\rho_{Sp} \approx 0{,}98$, jei $k$ priklauso intervalui | $[24; 55]$ | $[23; 62]$ | $[23; 70]$ | $[23; 76]$ | $[27; 80]$ | $[32; 82]$ |

Kaip ir ankstesniais atvejais, LLE algoritmo vykdymo laikas didėja polinomiškai didinant $k$ (5.19 pav.), todėl vertėtų pasirinkti mažiausias parametro $k$ reikšmes, su kuriomis $\rho_{Sp} \approx 0{,}98$. Šiuo atveju mažiausiai sugaištama laiko ir gaunamos teisingos projekcijos, kai $k = 23$, $D = \{10^{-40}, 10^{-30}, 10^{-20}\}$; $k = 24$, $D = 10^{-50}$.




**5.19 pav.** Laiko priklausomybė nuo *k*, gauta transformavus pussferės taškus į plokštumą LLE metodu

5.20 paveiksle pavaizduota pussferės taškų projekcija plokštumoje, gauta LLE algoritme parinkus tokias parametrų *D* ir *k* reikšmes, kad $\rho_{Sp} \approx 0{,}98$.



**5.20 pav.** Projekcija plokštumoje, gauta vizualizavus pussferės taškus LLE metodu, esant tokioms parametrų *D* ir *k* reikšmėms, kad $\rho_{Sp} \approx 0{,}98$

Ištyrus tris dirbtines duomenų aibes, nustatyta, kad:

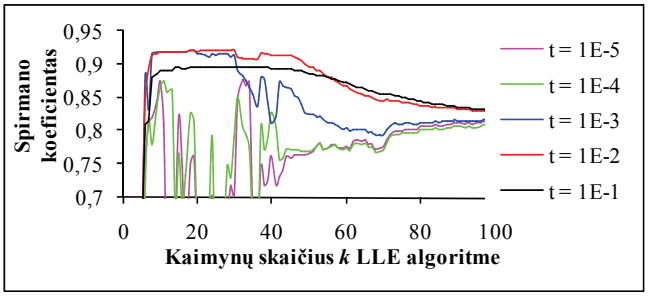
- Galima pasirinkti bet kokią determinanto reikšmę $D \in [10^{-50}; 10^{-10}]$ ir, apskaičiavus Spirmano koeficiento reikšmes, braižyti Spirmano koeficiento grafiką, imant parametro *k* reikšmes iš gana plataus intervalo. Po to rasti didžiausias (geriausias) Spirmano koeficiento reikšmes atitinkančius *k*.
- LLE algoritmo vykdymo laikas didėja polinomiškai, didinant *k*. Siekiant sugaišti mažiau laiko vizualizuojant duomenis, rekomenduojama braižyti Spirmano koeficiento grafikus su keliomis *D* reikšmėmis, imant mažesnį parametro *k* kitimo intervalą, ir rasti didžiausias (geriausias) Spirmano

koeficiento reikšmes atitinkančius $k$ ir $D$, atsižvelgiant į tai, kad artimiausių kaimynų skaičius turi būti kuo mažesnis.

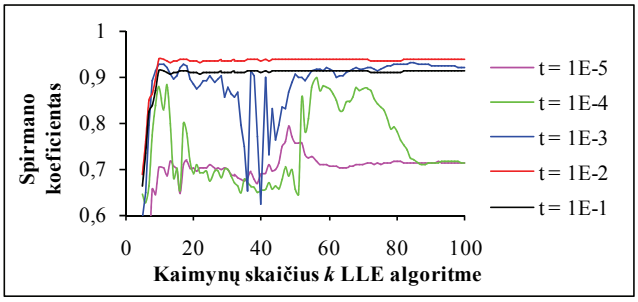
- Visais trimis išnagrinėtais atvejais gaunamos teisingos projekcijos esant nedideliam artimiausių kaimynų skaičiui, kai reguliarizacijos algoritme R2 valdymo parametras – fiksuotas determinantas – įgyja reikšmę $10^{-30}$.

**Reguliarizacijos algoritmo R1 tyrimas**

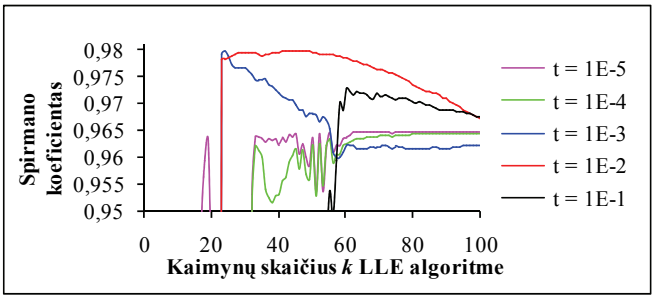
Analogiški tyrimai su šiomis duomenų aibėmis atlikti ir naudojant reguliarizaciją R1. Kiekvienu atveju LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis $k \in [6; 100]$, kiekvieną kartą apskaičiuojant Spirmano koeficientą. Gautos penkios Spirmano koeficiento priklausomybės nuo $k$, kai valdymo parametras įgyja tokias reikšmes: $t = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ (5.21, 5.22, 5.23 pav.). Visais trimis išnagrinėtais atvejais geriausios Spirmano koeficiento reikšmės (nustatytos tiriant reguliarizaciją R2) įgyjamos, kai $t = \{10^{-3}, 10^{-2}\}$.




**5.21 pav.** Spirmano koeficiento priklausomybės nuo $k$, gautos transformavus S-formos daugdaros taškus LLE metodu į dvimatę erdvę, esant skirtingoms $t$ reikšmėms




**5.22 pav.** Spirmano koeficiento priklausomybės nuo $k$, gautos transformavus daugdaros „Dvynės viršukalnės“ taškus LLE metodu į dvimatę erdvę, esant skirtingoms $t$ reikšmėms

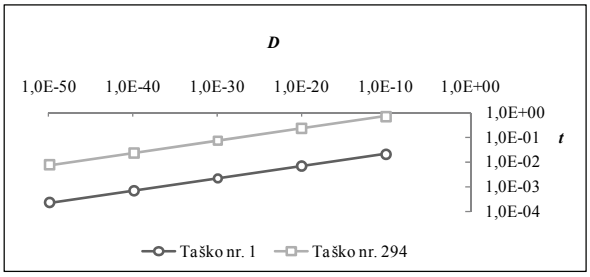



**5.23 pav.** Spirmano koeficiento priklausomybės nuo *k*, gautos transformavus pussferės taškus LLE metodu į dvimatę erdvę, esant skirtingoms *t* reikšmėms

### Ryšio tarp valdymo parametrų *D* ir *t* nustatymas

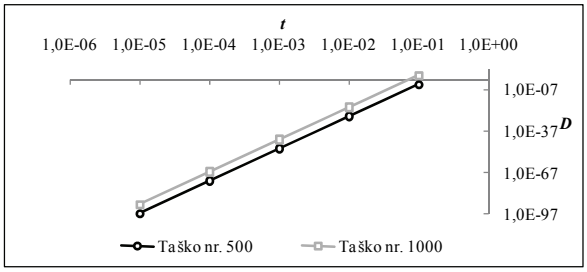
Siekiant nustatyti ryšį tarp valdymo parametrų *D* ir *t*, su S-formos daugdaros ir pussferės taškais atliktas toks tyrimas:

- Tirta, kaip kinta parametro *t* reikšmė atskiram duomenų taškui, didinant determinanto reikšmę. Kadangi tyrimo duomenys (parametro *D* reikšmės) ir rezultatai (gautos *t* reikšmės) yra labai maži skaičiai, jie pavaizduoti logaritminėje skalėje. Iš 5.24 paveikslo matome, kad didinant determinanto *D* reikšmę, didėja ir parametro *t* reikšmė.
- Tirta, kaip kinta parametro *D* reikšmė atskiram duomenų taškui, didinant parametro *t* reikšmę. Kadangi tyrimo duomenys (parametro *t* reikšmės) ir rezultatai (gautos parametro *D* reikšmės) yra labai maži skaičiai, jie pavaizduoti logaritminėje skalėje. Iš 5.25 paveikslo matome, kad didėjant *t* reikšmei, didėja ir determinanto reikšmė.



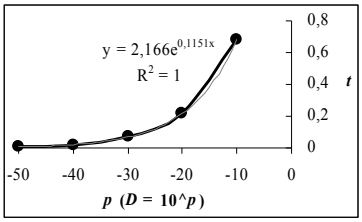

**5.24 pav.** Parametro *t* reikšmės kitimas atskiram duomenų taškui didinant determinantą (analizuoti pussferės taškai)

5.25 paveiksle matyti, kad, parinkus per daug mažą parametro *t* reikšmę, galime gauti per daug mažą determinanto reikšmę, o tai gali nulemti kokius nors skaičiuojamuosius nestabilumus.

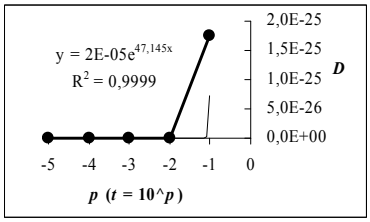



**5.25 pav.** Determinanto reikšmės kitimas atskiram duomenų taškui didinant parametro *t* reikšmę (analizuoti S-formos daugdaros taškai)

Iš šio tyrimo galima daryti išvadą, kad valdymo parametrai *D* ir *t* yra glaudžiai tarpusavyje susiję: didinant vieną parametrą, didėja ir kitas. Be to, didinant vieno parametro reikšmes, kito parametro reikšmės didėja eksponentiškai (5.26 pav.).




a) parametro *t* priklausomybė nuo *D*

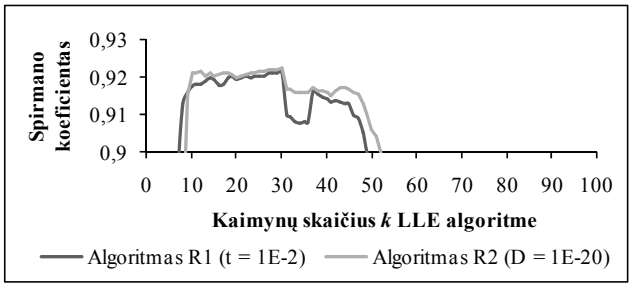



b) parametro *D* priklausomybė nuo *t*

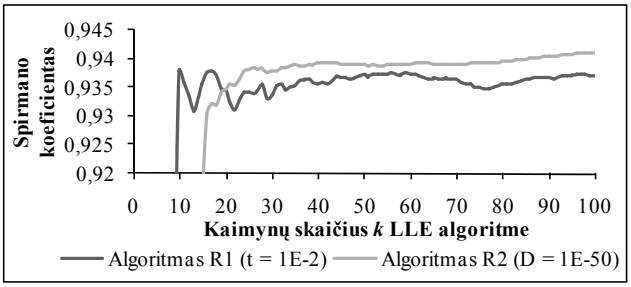
**5.26 pav.** Parametrų *D* ir *t* eksponentinė priklausomybė (analizuotas 294-as pussferės taškas)

**Geriausių rezultatų, gautų taikant reguliarizacijos algoritmus R1 ir R2 LLE algoritme, palyginimas**

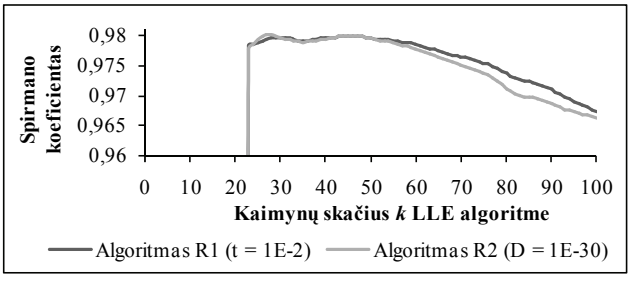
5.2.2.2 skyrelyje, išanalizavę reguliarizacijos algoritmus, įsitikinome, kad abu algoritmai pateikia panašią galimybę analizei, kiekvienas iš jų turi po vieną valdymo parametrą: *t* R1 atveju ir *D* R2 atveju. Ankstesniame skyrelyje atskleistas glaudus šių parametrų tarpusavio ryšys. Žemiau palyginti geriausi rezultatai, gauti taikant reguliarizacijos algoritmus R1 ir R2 LLE algoritme. Iš 5.27 paveikslo matome, kad naudojant šiuos reguliarizacijos algoritmus gaunami panašūs rezultatai Spirmano koeficiento prasme.




a) analizuoti netiesinės dvimatės S-formos daugdaros taškai




b) analizuoti netiesinės dvimatės daugdaros „Dvynės viršukalnės" taškai




c) analizuoti pussferės taškai

**5.27 pav.** Spirmano koeficiento priklausomybės nuo *k*, gautos taikant reguliarizavimo algoritmus R1 ir R2 LLE metode

## 5.3. Penktojo skyriaus apibendrinimas ir išvados

5 skyriuje nagrinėjamas netiesinis dimensijos mažinimo metodas – lokaliai tiesinis vaizdavimas, kuris daugiamačius duomenis transformuoja į mažesnės dimensijos erdvę, išlaikant kaimyniškumą tarp artimiausių taškų ir atskleidžiant daugiamačių duomenų netiesinę struktūrą. Nuo algoritmo valdymo parametrų – artimiausių kaimynų skaičių kiekvienam duomenų taškui ir lokaliosios Gramo matricos reguliarizacijos parametro – reikšmių labai priklauso duomenų vizualizavimo kokybė.

5.2.1 skyrelyje siūlome naują būdą artimiausių kaimynų skaičiui parinkti. Metodas eksperimentiškai ištirtas su dviem duomenų aibėmis: dirbtiniais ir realiais duomenimis, gautais skaitmenizavus paveikslėlius. Kad kiekybiškai įvertintume daugdaros topologijos išlaikymą, skaičiuojame Spirmano koeficientą. Eksperimentai parodė, kad kiekybinis matas – Spirmano koeficientas – yra tinkamas duomenų topologijos išlaikymui įvertinti, duomenis transformavus LLE algoritmu į mažesnės dimensijos erdvę. Tam, kad Spirmano koeficientas tinkamai atspindėtų gautas projekcijas, skaičiuojant jo reikšmę reikia *n*-matėje erdvėje vertinti geodezinius, o ne Euklido atstumus, ir geodezinių atstumų skaičiavimo algoritme fiksuoti gana mažą artimiausių kaimynų skaičių.

5.2.2 skyrelyje pasiūlytas naujas algoritmas lokaliajai Gramo matricai reguliarizuoti. Pasiūlytas reguliarizacijos algoritmas ištirtas detaliai teoriškai ir eksperimentiškai su trimis dirbtinėmis duomenų aibėmis. Palyginus su algoritmu, pasiūlytu straipsnyje (Roweis and Saul 2000), gauti panašūs rezultatai. Taigi abu algoritmai gali būti alternatyviai naudojami LLE metode reguliarizuojant Gramo matricą. Abu algoritmai pateikia panašią galimybę analizei, kiekvienas iš jų turi po vieną valdymo parametrą: $t$ R1 atveju ir $D$ R2 atveju. Tačiau parametro $D$ privalumas lyginant su $t$ yra tai, kad parametras $D$ turi realią prasmę (jis yra reguliarizuotos Gramo matricos determinantas), o $t$ yra tiktai tam tikras daugiklis.

Literatūroje yra ir daugiau reguliarizacijos algoritmų. Tichonovo (*Tikhonov*) reguliarizacija (Tikhonov and Arsenin 1977) yra vienas iš dažniausiai naudojamų reguliarizacijos metodų, skirtų „blogai-sąlygotiems“ uždaviniams (*ill-posed problems*). Taigi LLE algoritme galima taikyti ir Tichonovo reguliarizaciją. Tačiau, norint įvertinti Tichonovo reguliarizacijos efektyvumą, būtini papildomi detalūs tyrimai. Kito tyrimo objektas gali būti modernesnių (sudėtingesnių) (*more sophisticated*) branduolio funkcijų pritaikymas Gramo matricai gauti – ne tik tiesinių, bet taip pat polinominių, radialinių bazinių, Ermito (*Hermitian*) ir kitų branduolio funkcijų.

# 6

# Laplaso matricos tikrinių žemėlapių metodo realizacijų tyrimas

Šiame skyriuje detaliai ištirtas ir modifikuotas dar vienas netiesinis duomenų dimensijos mažinimo bei daugdaros atpažinimo metodas – Laplaso matricos tikrinių žemėlapių metodas (*Laplacian Eigenmaps*, LE) (Belkin and Niyogi 2003). Pasiūlyta nauja metodo realizacija, ištirtas jos efektyvumas lyginant su ankstesne. LE algoritmo modifikacija skiriasi nuo pradinio LE algoritmo parametrais, įtakojančiais vizualizavimo kokybę. Ištirta abiejų LE algoritmų priklausomybė nuo parametrų. Skyriaus rezultatai paskelbti straipsnyje (Karbauskaitė and Dzemyda 2009b).

## 6.1. Laplaso matricos tikrinių žemėlapių metodas

Tarkime, kad vizualizuojamų duomenų aibę $X$ sudaro $m$ $n$-matės erdvės taškų (vektorių) $X_i = (x_{i1},...,x_{in}), i = \overline{1,m}$ ($X_i \in R^n$), priklausančių glodžiajai netiesinei $d$-matei daugdarai, kuri įdėta erdvėje $R^n$. Šiame algoritme analizuojamų duomenų matrica $X = \{X_1, X_2,..., X_m\} = \{x_{ij}, i = \overline{1,m}, j = \overline{1,n}\}$, kurios eilutės yra $n$-matės erdvės vektoriai, sudaryta iš $m$ eilučių ir $n$ stulpelių.

Šie daugiamatės erdvės taškai gali būti pavaizduoti grafu $G=(V,E)$. Kiekviena grafo viršūnė $V_i$, $i=\overline{1,m}$ susieta su duomenų aibės tašku $X_i$, $i=\overline{1,m}$, o briauna jungia viršūnes $V_i$ ir $V_j$, jei atitinkami duomenų aibės taškai $X_i$ ir $X_j$ yra kaimynai. Yra du būdai kaimynams parinkti:

- parenkamas parametras $\delta$, $\delta \in R$ – spindulys atvirojo rutulio, į kurį patekę taškai bus laikomi kaimynais. $X_j$ bus laikomas taško $X_i$ kaimynu, jei tenkinama nelygybė $\|X_i - X_j\|^2 < \delta$, čia $\|\cdot\|$ – Euklido atstumas. Privalumai: būdas yra geometriškai motyvuotas, ryšys tarp abiejų kaimyninių taškų yra simetriškas. Trūkumai: dažnai gaunamas nejungusis grafas, sudarytas iš kelių dalių, sunku parinkti $\delta$.
- parenkamas artimiausių kaimynų skaičius $k$ ($k \in N$). Privalumai: parametrą $k$ lengviau parinkti, grafas visada bus jungusis. Trūkumai: būdas yra geometriškai mažiau intuityvus.

Kaimynystės ryšiai gali būti užšifruoti specialioje duomenų struktūroje arba, paprasčiau, gretimumo matricoje $A$. Dvejetainės (binarinės) reikšmės $a_{ij} \in \{0,1\}$ nurodo, ar daugiamatės erdvės taškai $X_i$ ir $X_j$ yra kaimynai (1) ar ne (0). Gretimumo matrica $A$ privalo būti simetrinė, turint omeny, kad grafas *G* privalo būti neorientuotas.

LE metodo tikslas – atvaizduoti daugiamačių duomenų aibę *X* į mažesnės dimensijos duomenų aibę *Y*, kurioje išlaikomi tie patys kaimynystės ryšiai. Duomenų, gautų po projekcijos, matrica $Y=\{Y_1,Y_2,...,Y_m\}=\{y_{ij}, i=\overline{1,m}, j=\overline{1,d}\}$ ($d \ll n$) yra sudaryta iš *m* eilučių ir *d* stulpelių.

Šiam tikslui pasiekti reikia minimizuoti tokią tikslo funkciją:

$$E_{LE}=\frac{1}{2}\sum_{i,j=1}^{m}\|Y_i-Y_j\|^2 w_{ij}, \tag{6.1}$$

kur simetrinės matricos $W$ elementai $w_{ij}$ yra susiję su gretimumo matricos $A$ elementais $a_{ij}$ tokiu būdu: $w_{ij}=0$, jei $a_{ij}=0$, priešingu atveju, $w_{ij}\geq 0$. Yra du būdai nenulinėms $w_{ij}$ reikšmėms parinkti:

- Naudojant šiluminio branduolio (*Heat kernel*) parametrą $T$ ($T \in R$) Gauso branduolio funkcijoje:

$$w_{ij} = e^{-\frac{\|X_i - X_j\|^2}{T}}; \tag{6.2}$$

- Supaprastintas būdas, neturintis parametrų: $w_{ij} = 1$, kai $a_{ij} = 1$. Šis variantas gaunamas, kai Gauso branduolio funkcijoje pasirenkame $T = \infty$. Šio būdo privalumas – atsisakoma parametro *T*.

Atsižvelgus į tai, kad $W$ yra simetrinė matrica, kriterijus $E_{LE}$ gali būti užrašomas tokia matricine forma:

$$E_{LE} = \mathrm{tr}\left(Y^T L Y\right). \tag{6.3}$$

Šioje lygtyje *L* yra grafo *G* Laplaso matrica. Laplaso matrica yra simetrinė, teigiamai pusiau apibrėžta ir apskaičiuojama pagal formulę:

$$L = \widetilde{D} - W, \tag{6.4}$$

čia $\widetilde{D}$ yra diagonalioji svorių matrica, kurios elementai yra matricos $W$ stulpelių (arba eilučių, nes matrica $W$ simetrinė) sumos, $\widetilde{d}_{ii} = \sum_{j=1}^{m} w_{ji}$, $i = \overline{1,m}$.

Norint įrodyti (6.3), pakanka pastebėti, kad *d*-matėms projekcijoms teisinga:

$$\begin{aligned}
E_{LE} &= \frac{1}{2}\sum_{i,j=1}^{m}\left\|Y_i - Y_j\right\|^2 w_{ij} = \frac{1}{2}\sum_{p=1}^{d}\sum_{i,j=1}^{m}\left(y_{ip} - y_{jp}\right)^2 w_{ij} \\
&= \frac{1}{2}\sum_{p=1}^{d}\sum_{i,j=1}^{m}\left(y_{ip}^2 + y_{jp}^2 - 2y_{ip}y_{jp}\right) w_{ij} \\
&= \frac{1}{2}\sum_{p=1}^{d}\left(\sum_{i=1}^{m} y_{ip}^2 \widetilde{d}_{ii} + \sum_{j=1}^{m} y_{jp}^2 \widetilde{d}_{jj} - 2\sum_{i,j=1}^{m} y_{ip}y_{jp}w_{ij}\right) \\
&= \frac{1}{2}\sum_{p=1}^{d} 2f_p^T \widetilde{D} f_p - 2f_p^T W f_p = \frac{1}{2}\sum_{p=1}^{d} 2f_p^T L f_p = \mathrm{tr}\left(Y^T L Y\right),
\end{aligned}$$

kur $f_p$ yra *m*-matės erdvės vektorius, duodantis kiekvienam projekcijos taškui *p*-tąją koordinatę, t. y. $f_p = \left(y_{1p}, y_{2p}, ..., y_{mp}\right)^T$ (vektorius-sulpelis), o $f_p^T$ yra matricos *Y* *p*-tojo stulpelio transpozicija.

Tikslo funkcijos $E_{LE}$ reikšmės ypač išauga, jei kaimyniniai taškai $X_i$ ir $X_j$ atvaizduojami į tolimus taškus $Y_i$ ir $Y_j$. Todėl ją minimizuojant ir stengiamasi užtikrinti, kad jei $X_i$ ir $X_j$ yra artimi, tai ir $Y_i$ ir $Y_j$ būtų taip pat artimi.

Siekiant rasti aibę *Y*, funkcijos $E_{LE}$ minimizavimas, esant ribojimui $Y^T \widetilde{D} Y = I_{d\times d}$, suvedamas į apibendrintą tikrinių vektorių ir tikrinių reikšmių radimo uždavinį:

$$Lf = \lambda \widetilde{D} f, \tag{6.5}$$

kai ieškoma matricos *L* *d* tikrinių vektorių, susijusių su mažiausiomis, nelygiomis nuliui tikrinėmis reikšmėmis.

Kai projekcinės erdvės dimensija $d = 1$, šis ribojimas apsaugo nuo situacijos, kai visi taškai yra atvaizduojami į vieną tašką, kai $d = 2$ – į vieną tiesę, ir t. t. Ieškant *d*-mačių projekcijų, šis ribojimas apsaugo, kad taškai neatsivaizduotų į poerdvį, kurio dimensija mažesnė negu *d*.

Tegu $f_0, ..., f_{m-1}$ yra lygties (6.5) sprendiniai (tikriniai vektoriai), sutvarkyti pagal jų tikrinių reikšmių dydį:

$$Lf_0 = \lambda_0 \widetilde{D} f_0,$$

$$Lf_1 = \lambda_1 \widetilde{D} f_1,$$

...

$$Lf_{m-1} = \lambda_{m-1} \widetilde{D} f_{m-1},$$

$$0 = \lambda_0 \le \lambda_1 \le ... \le \lambda_{m-1}.$$

Kadangi *L* yra simetrinė ir teigiamai pusiau apibrėžta, visos tikrinės reikšmės yra realios ir nemažesnės už nulį.

Ieškant *n*-matės erdvės taškų projekcijos į *d*-matę erdvę, tikrinis vektorius $f_0$, atitinkantis tikrinę reikšmę $\lambda_0 = 0$, neįtraukiamas (kad būtų tenkinamas ribojimas $Y^T \widetilde{D} Y = I_{d\times d}$), o naudojami sekantys *d* tikriniai vektoriai $f_1, f_2, ..., f_d$ ($f_j = (y_{1j}, y_{2j}, ..., y_{mj}), j = \overline{1,d}$):

$$X \to Y = (f_1, f_2, ..., f_d) = \begin{pmatrix} y_{11}\ y_{12} \cdots y_{1d} \\ y_{21}\ y_{22} \cdots y_{2d} \\ \cdots\ \cdots\cdots\ \cdots \\ y_{m1}\ y_{m2} \cdots y_{md} \end{pmatrix}, \ Y_i = (y_{i1}\ y_{i2} \ldots y_{id}), \ i = \overline{1,m}.$$

Straipsnyje (Belkin and Niyogi 2003) teigiama, jog šis sprendinys yra optimalus.

*LE algoritmo schema (Belkin and Niyogi 2003)*:

1. Sukonstruojamas kaimynystės grafas (*Adjacency Graph*).
2. Parenkami grafo briaunų svoriai.
3. Randamos daugiamatės erdvės taškų projekcijos mažesnio matavimo erdvėje iš šio grafo Laplaso matricos tikrinių vektorių.

## 6.2. LE algoritmo modifikacija

Šiame skyrelyje pasiūlyta LE algoritmo modifikacija skiriasi nuo pradinio LE algoritmo (Belkin and Niyogi 2003) parametrais, kurie įtakoja vizualizavimo kokybę. Pradiniame LE algoritme galimi du parametrai, kurių reikšmės parenkamos: artimiausių kaimynų skaičius $k$ kiekvienam duomenų taškui ir šiluminio branduolio parametras $T$, naudojamas Gauso branduolio funkcijoje. Straipsnyje (Belkin and Niyogi 2003) teigiama, jog neaišku, kaip parinkti parametrą $T$. Ypač sunku teisingai parinkti šį parametrą, esant labai dideliam artimiausių kaimynų skaičiui $k$. Taip pat parodyta, jog nuo pasirinktų parametrų $k$ ir $T$ reikšmių labai priklauso gautų duomenų projekcijų kokybė.

Mūsų modifikuotas LE algoritmas turi šiuos parametrus:

- Parametrą $w^*$, $w^* \in R, 0 < w^* < 1$. Tai svorinis slenkstis, virš kurio svoriai laikomi tinkamais, t. y. tuos svorius turintys taškai turi įtakos nagrinėjamam taškui ir yra laikomi artimiausiais kaimynais. Šis parametras yra konstanta, nes jo reikšmė vykdant algoritmą nekinta.
- Šiluminio branduolio parametrą $T$ ($T \in R$). Jo reikšmės parinkti nereikia – ji nustatoma automatiškai.
- Maksimalų artimiausių kaimynų skaičių $k^*$. Artimiausių kaimynų skaičius kiekvienam duomenų taškui yra kintamas. Tačiau numatyta

galimybė riboti kaimynų skaičių, imant loginę sąlygą, kad kiekvienas duomenų taškas gali turėti ne daugiau kaip $k^*$ artimiausių kaimynų.

Pagrindinė LE modifikacijos idėja tokia. Pirmiausia, apskaičiuojami svoriai tarp visų duomenų taškų. Tada, remiantis gauta svorių matrica ir pasirinktu svoriniu slenksčiu, randami kiekvieno duomenų taško artimiausi kaimynai ir sudaromas svorinis grafas.

*Apibendrinta modifikuoto LE algoritmo schema:*

1. Fiksuojamos algoritmo parametrų ir tarpinių kintamųjų pradinės reikšmės

Fiksuojamos parametrų $w^*$ ($w^* \in R, 0 < w^* < 1$), $k^*$ ($k^* \in N$) reikšmės. Parenkama pradinė parametro $T$ ($T \in R$) reikšmė. Taip pat parenkamos kintamojo *a* tokios reikšmės, kad $a \in R$, $a > 1$.

2. Apskaičiuojami svoriai tarp visų duomenų taškų

a) Mažinama parametro *T* reikšmė: $T = T / a$;

b) Apskaičiuojami svoriai tarp visų duomenų taškų, naudojant Gauso branduolio funkciją:

$$w_{ij} = \begin{cases} e^{-\frac{\|X_i - X_j\|^2}{T}}, \text{ jei } i \neq j,\ i, j = \overline{1, m}; \\ 0, \text{ jei } i = j, \end{cases}$$

c) Randame, kiek kiekvienam duomenų taškui $X_i$ egzistuoja taškų $X_j$, $j = \overline{1, m}$, kurių svoriai $w_{ij}$ tenkina sąlygą $w_{ij} \geq w^*$, ir iš gautų rezultatų randamas didžiausias skaičius $k_{\max}$.

d) Lyginama gauta $k_{\max}$ reikšmė su pasirinktu maksimaliu artimiausių kaimynų skaičiumi $k^*$:

Jei $k_{\max} < k^*$, tai atliekami tokie žingsniai:

- atstatoma ankstesnė parametro *T* reikšmė: $T = T \cdot a$;
- atstatoma ankstesnė kintamojo $k_{\max}$ reikšmė, tokia, kad $k_{\max} > k^*$;
- nežymiai sumažinama kintamojo *a* reikšmė: $a = a \cdot 0.999$;
- grįžtama į antrojo žingsnio pradžią.

Jei $k_{\max} > k^*$, tai grįžtama į antrojo žingsnio pradžią.

Pasibaigus antrajam žingsniui, $k_{\max} = k^*$ ir yra gaunama svorių matrica $W$.

3. Sukonstruojamas svorinis kaimynystės grafas

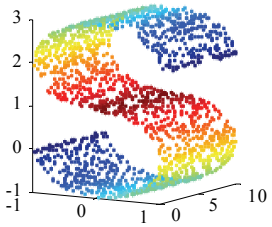
Jei daugiamatės erdvės taškai $X_i$ ir $X_j$ yra kaimynai, tai grafo viršūnės $V_i$ ir $V_j$ sujungiamos briauna, kurios svoris $w_{ij} \neq 0$, priešingu atveju, briaunos svoris $w_{ij} = 0$. Taškas $X_j$ laikomas taško $X_i$ kaimynu, jei $w_{ij} \geq w^*$. Jei kuris nors duomenų taškas $X_i$ neturi nė vieno kaimyno, tai jo kaimynu laikomas tas taškas $X_j$, kurio svoris $w_{ij}$ ($w_{ij} < w^*$) yra didžiausias.

4. Randamos daugiamatės erdvės taškų projekcijos mažesnio matavimo erdvėje iš šio grafo Laplaso matricos tikrinių vektorių

Visi šio žingsnio veiksmai yra tokie pat kaip ir Belkin ir Niyogi pasiūlyto LE algoritmo 3 žingsnyje (Belkin and Niyogi 2003).

## 6.3. Parametrų *k* ir *T* svarba LE algoritme

LE algoritmas (Belkin and Niyogi 2003) turi du valdymo parametrus: artimiausių kaimynų skaičių $k$ kiekvienam duomenų taškui ir šiluminio branduolio parametrą $T$, naudojamą Gauso branduolio funkcijoje svoriams apskaičiuoti. Straipsnyje (Belkin and Niyogi 2003), atliekant eksperimentus su keliais realiais duomenų rinkiniais, parodoma, jog galima naudoti ir paprastesnę algoritmo versiją, pasirenkant $w_{ij} \in \{0,1\}$ arba $T = \infty$. Šiuo atveju algoritmas turi tik vieną parametrą $k$ ir, pasak autorių, veikia gerai. Bet ar šis teiginys teisingas bendru atveju?




**6.1 pav.** 2000 3-matės erdvės taškų, priklausančių netiesinei dvimatei S-formos daugdarai

Nagrinėkime S-formos daugdaros taškus ($m = 2000, n = 3$) (6.1 pav.). Tarkime, kad $T = \infty$. Iš 6.2 paveikslo matome, kad, esant šiai parametro $T$ reikšmei, S-formos daugdara „išvyniojama“, tik kai parametras $k \leq 50$. Kai

$k \geq 100$, daugdaros „išvynioti“ nepavyksta. Vadinasi, ne visada galima imti parametro $T$ reikšmę $\infty$. Būtina dar atsižvelgti ir į artimiausių kaimynų skaičių $k$. Iš atliktų eksperimentų su S-formos daugdaros taškais matyti, kad nuo pasirinktų parametrų $k$ ir $T$ reikšmių labai priklauso gautų projekcijų kokybė. Projekcijų kokybei įvertinti naudojame Spirmano koeficientą. Nesunku pastebėti, jog siekiant kuo geriau „išvynioti“ daugdarą didinant artimiausių kaimynų skaičių $k$, parametro $T$ reikšmę reikia mažinti. 6.1 lentelėje pateiktos daugdaros „išvyniojimui“ būtinos šių parametrų reikšmės. Taigi akivaizdu, kad, esant labai dideliam artimiausių kaimynų skaičiui $k$, sunku teisingai parinkti parametrą $T$. Todėl buvo siekiama modifikuoti LE algoritmą taip, kad parametro $T$ reikšmės parinkti nereikėtų, o ji būtų apskaičiuojama, vykdant algoritmą.

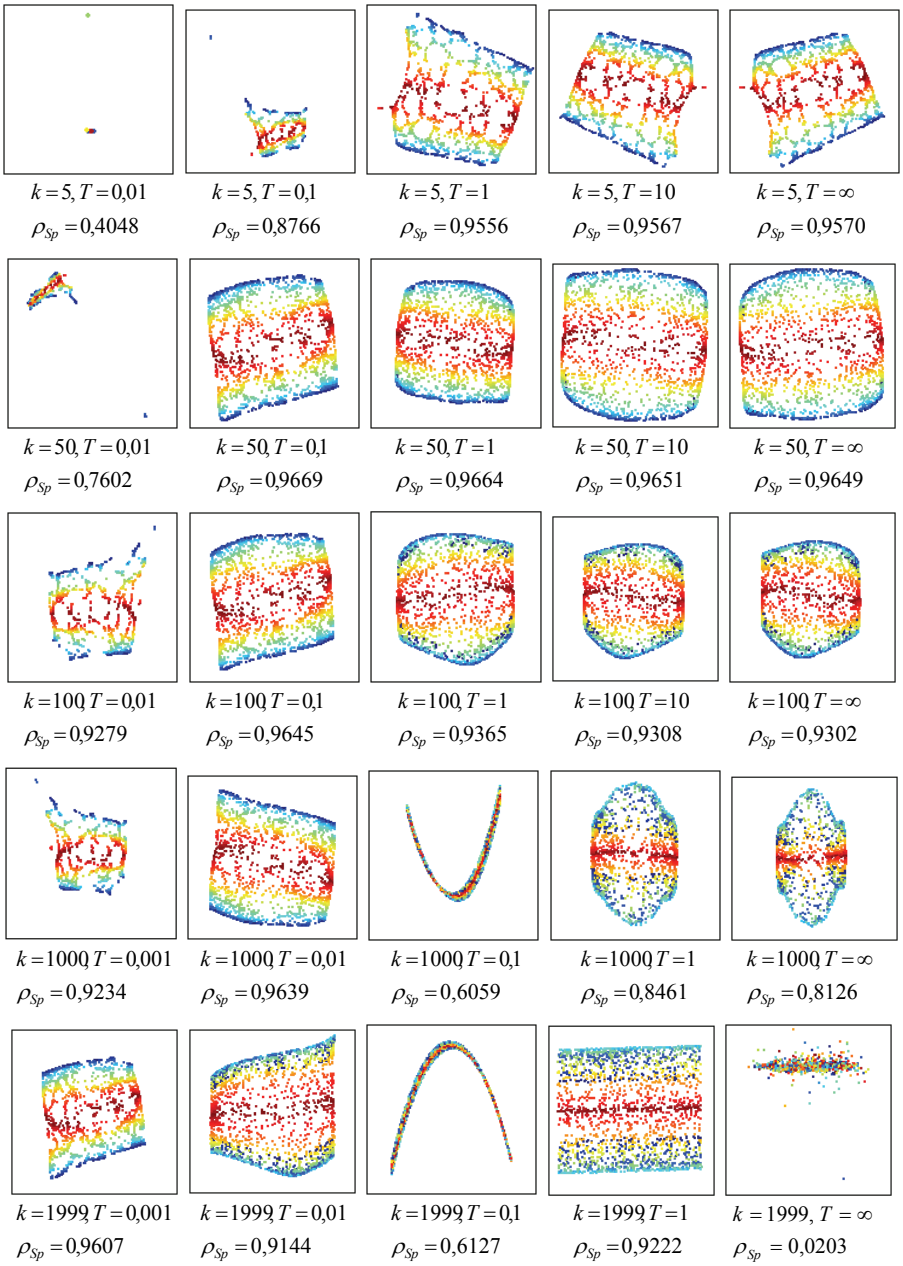
**6.1 lentelė.** Parametrų $k$ ir $T$ reikšmės, reikalingos S-formos daugdarai „išvynioti“

| $k$ | 5 | 50 | 100 | 1000 | 1999 |
|---|---|---|---|---|---|
| $T$ | (1; $\infty$) | (0,1; $\infty$) | 0,1 | 0,01 | 0,001 |
| $\rho_{Sp}$ | 0,96 | 0,97 | 0,96 | 0,96 | 0,96 |

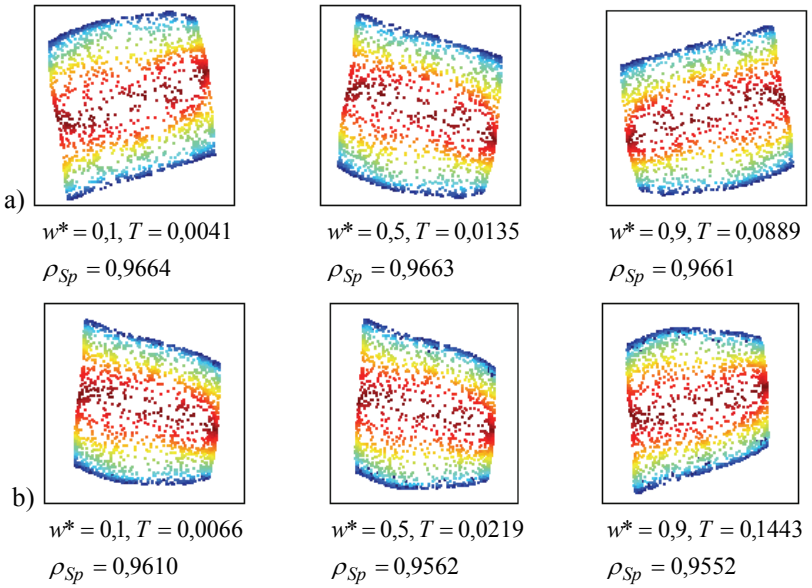
## 6.4. Parametrų svarba LE algoritmo modifikacijoje

LE algoritmo modifikacijoje svorinio slenkščio $w^*$ ($w^* \in R, 0 < w^* < 1$) ir maksimalaus artimiausių kaimynų skaičiaus $k^*$ reikšmės parenkamos ir algoritmo vykdymo metu jos nekinta. Šiluminio branduolio parametro $T$ ($T \in R$) pradine reikšme gali būti bet koks realus skaičius, nes, vykdant algoritmą, ji vis tiek kinta. Taigi tikslios parametro $T$ reikšmės parinkinėti nereikia – ji nustatoma automatiškai, tokia, kad bet kurio taško $X_i$ kaimynų skaičius $k_i$ neviršytų $k^*$.

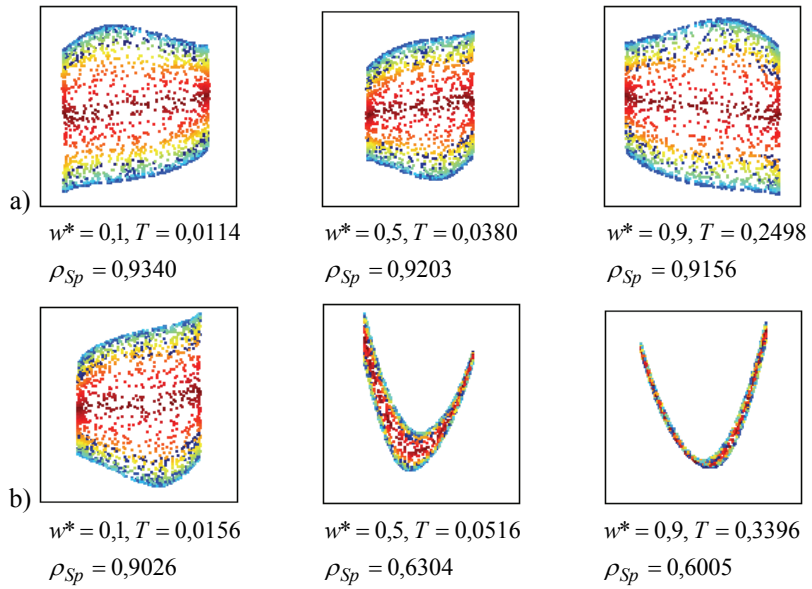
Dabar išsiaiškinkime, kaip nuo parametro $w^*$ skirtingų reikšmių priklauso gautų projekcijų kokybė. Nagrinėkime S-formos daugdaros taškus (6.1 pav.), kai pradinė $T = 10$. Kai maksimalus artimiausių kaimynų skaičius $k^* \in \{100, 150\}$, Spirmano koeficiento reikšmės gaunamos pakankamai didelės: $\rho_{Sp} \approx 0{,}97$ (6.3a pav.) ir $\rho_{Sp} \approx 0{,}96$ (6.3b pav.), t. y. daugdara „išvyniojama“ gana gerai su įvairiomis parametro $w^*$ reikšmėmis ($w^* \in \{0{,}1; 0{,}5; 0{,}9\}$). Galima pastebėti, kad, didinant parametro $w^*$ reikšmę, Spirmano koeficiento reikšmė mažėja, bet labai nežymiai, o parametro $T$ reikšmės didėja (6.3 pav.).




**6.2 pav.** S-formos daugdaros taškų projekcijos plokštumoje, gautos pradiniu LE algoritmu, esant skirtingoms parametrų $k$ ir $T$ reikšmėms




**6.3 pav.** S-formos daugdaros taškų projekcijos plokštumoje, gautos modifikuotu LE algoritmu, kai $k^* = 100$ (a) ir $k^* = 150$ (b)




**6.4 pav.** S-formos daugdaros taškų projekcijos plokštumoje, gautos modifikuotu LE algoritmu, kai $k^* = 300$ (a) ir $k^* = 500$ (b)

Kai $k^* = 300$ (6.4a pav.), $k^* = 500$ (6.4b pav.), S-formos daugdaros „išvynioti“ nepavyksta su jokia $w^*$ reikšme. Vadinasi, kai pasirinktas tinkamas artimiausių kaimynų skaičius $k^*$, parametro $w^*$ reikšme gali būti bet koks realus skaičius iš intervalo $(0;1)$. Tačiau, norint gauti kuo didesnę Spirmano koeficiento reikšmę (pasiekti kuo geresnę vizualizavimo kokybę), patartina pasirinkti parametro $w^*$ reikšmę kuo mažesnę, pavyzdžiui, $w^* = 0{,}1$. Atlikus šį tyrimą, galima teigti, kad mūsų pasiūlytoje LE algoritmo modifikacijoje yra tik vienas reikšmingas valdymo parametras – maksimalus artimiausių kaimynų skaičius $k^*$, nuo kurio labai priklauso gautų projekcijų kokybė.

## 6.5. Šeštojo skyriaus apibendrinimas ir išvados

Šiame skyriuje ištirtas LE algoritmas, kuris skirtas netiesiniam dimensijos mažinimui: netiesinės daugdaros atpažinimui ir jos vizualizavimui. Tirtos dvi LE algoritmo realizacijos – pasiūlyta metodo autorių ir siūloma šiame darbe. Abi LE algoritmo realizacijos eksperimentiškai ištirtos su keliomis dirbtinėmis duomenų aibėmis. Su visomis aibėmis gauti panašūs testavimo rezultatai. Todėl šiame skyriuje pateikiame tiktai rezultatus, gautus ištyrus S-formos daugdaros taškus, nes ši daugdara yra gerai žinoma ir dažnai naudojama analizuojant įvairius daugdaros atpažinimo metodus.

LE algoritmas turi du valdymo parametrus *k* ir *T*. Kaip žinoma, sunku teisingai parinkti *T*, jei parametro *k* reikšmės didelės. Šiame skyriuje pasiūlyta LE algoritmo modifikacija, turinti tris valdymo parametrus ($T$, $k^*$, $w^*$). Šios modifikacijos privalumas yra tai, kad šiluminio branduolio parametro *T* reikšmės parinkinėti nereikia, nes šiame skyriuje pasiūlyta galimybė ją įvertinti automatiškai. Kai pasirinktas tinkamas maksimalus artimiausių kaimynų skaičius $k^*$, parametro $w^*$ reikšme gali būti bet koks realus skaičius iš intervalo $(0;1)$. Tačiau, norint pasiekti kuo geresnę vizualizavimo kokybę, patartina pasirinkti parametro $w^*$ reikšmę kuo mažesnę. Taigi mūsų LE modifikacijoje lieka tik vienas svarbus valdymo parametras $k^*$ – maksimalus artimiausių kaimynų skaičius. Kintamas (iš anksto nefiksuotas) artimiausių kaimynų skaičius kiekvienam taškui yra šios modifikacijos unikalumas.

# 7

# Topologijos išlaikymo matai daugdaros tipo daugiamačių duomenų vizualizavime

Topologijos išlaikymui įvertinti sukurta daugybė įvairių matų: (Siegel and Castellan 1988; Goodhill and Sejnowski 1996; Konig 2000; Tenenbaum *et al*. 2000; Venna and Kaski 2001; Lee and Verleysen 2007) ir kt. Skirtingiems uždaviniams turi būti parenkami skirtingi topologijos išlaikymo matai (Goodhill and Sejnowski 1996). Mūsų tikslas – rasti ir ištirti tuos matus, kurie būtų tinkami analizuoti daugdaros topologijos išlaikymą po jos transformavimo į mažesnio matavimo erdvę. Šiame skyriuje nagrinėjami trys kiekybiniai matai: Spirmano koeficientas (*Spearman's rho*) (Siegel and Castellan 1988), Konigo matas (Konig 2000) ir kaimynystės klaidos (*mean relative rank errors*, MRRE) (Lee and Verleysen 2007). Spirmano koeficientas dažnai naudojamas topologijos išlaikymui tirti siekiant sumažinti dimensiją (Bezdek and Pal 1995; Goodhill and Sejnowski 1996; Kouropteva *et al*. 2005; Bernatavičienė *et al*. 2006). 5.2.1 skyrelyje parodyta, kad Spirmano koeficientas taip pat tinka topologijos išlaikymui įvertinti, vizualizavus daugdaros tipo duomenis LLE algoritmu. Konigo matas buvo panaudotas daugiamačių duomenų atvaizdavimų, gautų taikant saviorganizuojančius neuroninius tinklus, topologijos išlaikymui įvertinti (Konig 2000; Estevez *et al*. 2005). Šiame skyriuje parodyta, jog šis matas taip

pat gali būti sėkmingai naudojamas siekiant įvertinti topologijos išlaikymą, vizualizavus daugiamačius duomenis LLE metodu. Kaimynystės klaidos naudotos, vertinant topologijos išlaikymą vaizdų, gautų vizualizuojant dirbtinius ir tikrus veidus įvairiais netiesiniais dimensijos mažinimo metodais (Lee and Verleysen 2007). Šiame skyriuje palyginti šie topologijos išlaikymo matai ir išryškinti Konigo mato ir kaimynystės klaidų privalumai lyginant su Spirmano koeficientu. Šio skyriaus rezultatai paskelbti straipsnyje (Karbauskaitė and Dzemyda 2009a).

# 7.1. Trys topologijos išlaikymo matai

## 7.1.1. Spirmano koeficientas

Norint kiekybiškai įvertinti topologijos išlaikymą, dažnai yra skaičiuojamas Spirmano koeficientas (*Spearman's rho*) (Siegel and Castellan 1988). Šis kiekybinis skaitinis matas įvertina atitikimą (sąryšį) eiliškumo surūšiuotuose duomenyse, t. y. kaip gerai atitinkamų mažesnės dimensijos erdvės taškų (projekcijų) porų atstumų eilės numeriai atitinka didesnės dimensijos erdvės taškų porų atstumų eilės numerius surūšiuotose didėjančiai atstumų sekose. Spirmano koeficientas apskaičiuojamas pagal formulę:

$$\rho_{Sp} = 1 - \frac{6\sum_{i=1}^{S}\left(\hat{r}_X(i) - \hat{r}_Y(i)\right)^2}{S^3 - S}, \tag{7.1}$$

kur $S = m(m-1)/2$, $S$ – lyginamų atstumų skaičius, $m$ – taškų skaičius, $\hat{r}_X(i)$, $i = \overline{1,S}$ – analizuojamų ($n$-mačių) duomenų taškų visų porų atstumų eilės numeriai surūšiuotoje didėjimo tvarka atstumų sekoje, $\hat{r}_Y(i)$, $i = \overline{1,S}$ – suprojektuotų ($d$-mačių) taškų visų porų atstumų eilės numeriai surūšiuotoje didėjimo tvarka atstumų sekoje.

Be to, $-1 \le \rho_{Sp} \le 1$. Geriausia Spirmano koeficiento reikšmė lygi 1.

5.2.1 skyrelyje parodyta, kad Spirmano koeficientas yra tinkamas topologijos išlaikymui įvertinti transformavus dvimačių daugdarų taškus LLE algoritmu į plokštumą. Skaičiuojant atstumus tarp originalių duomenų taškų (porų) (dėl $\hat{r}_X(i)$), būtina naudoti geodezinius atstumus, kuriuos skaičiuojant turi būti pasirenkamas gana mažas artimiausių kaimynų skaičius ($\le 10$). Projekcijos atveju, kai projekcinės erdvės dimensija sutampa su duomenų vidine dimensija,

tiek Euklido, tiek geodeziniai atstumai gali būti naudojami skaičiuojant atstumus tarp suprojektuotų duomenų taškų (porų) (dėl $\hat{r}_Y(i)$).

### 7.1.2. Konigo matas

Topologijos išlaikymo matas, naudojamas straipsnyje (Konig 2000), paremtas kaimyninių taškų eilės numerių pradinėje ir projekcinėje erdvėse tvarkos įvertinimu. Pažymėkime šį matą KM. Jis turi du valdymo parametrus – artimiausių kaimynų skaičius: $k_1$ ir $k_2$ ($k_1 < k_2$). Kaimynams rasti skaičiuojami Euklido atstumai, t. y. Euklido atstumas yra artimumo matas. Pažymėkime:

1. $X_{ij}$, $j = \overline{1,k_1}$ $n$-matės erdvės taško $X_i$ $k_1$ artimiausius kaimynus, kur atstumai tarp $X_i$ ir jo kaimynų tenkina šią nelygybę $\left\|X_i - X_{ij_1}\right\| < \left\|X_i - X_{ij_2}\right\|$, kai $j_1 < j_2$;
2. $Y_{ij}$, $j = \overline{1,k_2}$ $d$-matės erdvės taško $Y_i$ $k_2$ artimiausius kaimynus;
3. $\bar{r}_X(i,j)$ taško $X_i$ $j$-tojo arčiausiojo kaimyno $X_{ij}$ eilės numerį analizuojamoje duomenų aibėje $X = \{X_1,...,X_m\}$;
4. $\bar{r}_Y(i,j)$ taško $Y_i$, atitinkančio tašką $X_i$, $j$-tojo arčiausiojo kaimyno $Y_{ij}$ eilės numerį aibėje $Y = \{Y_1,...,Y_m\}$.

Topologijos išlaikymo matas $i$-tajam taškui ir $j$-tajam kaimynui apskaičiuojamas tokiu būdu:

$$\mathrm{KM}_{ij} = \begin{cases} 3, \text{ jei } \bar{r}_X(i,j) = \bar{r}_Y(i,j), \\ 2, \text{ jei } \bar{r}_X(i,j) = \bar{r}_Y(i,l),\ l = \overline{1,k_1},\ j \neq l, \\ 1, \text{ jei } \bar{r}_X(i,j) = \bar{r}_Y(i,t),\ t = \overline{k_1+1,k_2},\ k_1 < k_2, \\ 0, \text{ priešingu atveju.} \end{cases}$$

Bendrasis matas KM apskaičiuojamas pagal formulę:

$$\mathrm{KM} = \frac{1}{3k_1 m} \sum_{i=1}^{m} \sum_{j=1}^{k_1} \mathrm{KM}_{ij}.$$

KM reikšmių kitimo sritis yra tarp 0 ir 1, kur 0 reiškia blogą kaimynystės išsaugojimą, o 1 – gerą.

**Topologijos išlaikymo mato KM parametrų analizė**

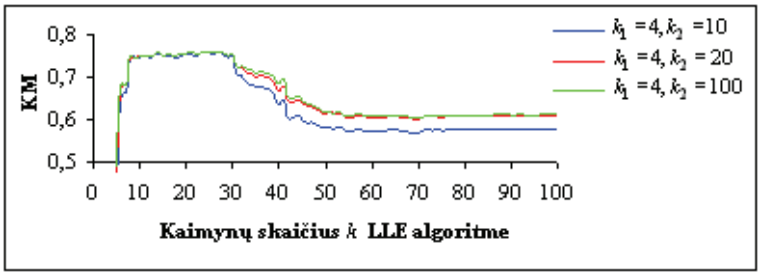
Topologijos išlaikymo matas KM turi du valdymo parametrus: mažesnįjį artimiausių kaimynų skaičių $k_1$ ir didesnįjį artimiausių kaimynų skaičių $k_2$ $(k_1 < k_2)$ kiekvienam taškui. Siekiant išanalizuoti šių parametrų įtaką gautai KM reikšmei, atlikta keletą tyrimų.

Pirmas tyrimas (7.1 pav.) atliktas su netiesinės dvimatės S-formos daugdaros taškais $(m = 1000, n = 3)$ (7.16a pav.). LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [5; 100]$. Kiekvienai pasirinktai $k$ reikšmei buvo apskaičiuojamas topologijos išlaikymo matas KM. Iš pradžių, skaičiuojant KM reikšmes, buvo fiksuotas mažesnysis artimiausių kaimynų skaičius $k_1$ $(k_1 = 4)$, o didesnysis artimiausių kaimynų skaičius $k_2$ buvo po truputį didinamas ($k_2 = \{10, 20, 100\}$). Gautos trys KM priklausomybės nuo LLE algoritmo parametro $k$, esant skirtingoms $k_1$ ir $k_2$ kombinacijoms: {4, 10}, {4, 20} ir {4, 100}. Iš 7.1a paveikslo matyti, jog, kai KM reikšmė yra geriausia, t. y. $\mathrm{KM} \approx 0{,}75$, visų priklausomybių reikšmės apytiksliai lygios. Kai $k > 30$, KM reikšmės ima mažėti. Be to, kai parametras $k_2 \geq 20$, priklausomybės apytiksliai sutampa. Kai $k_2 = 10$, gauta KM priklausomybė įgyja mažesnes reikšmes nei kitos ($k_2 = 20$ ar $k_2 = 100$). Tačiau vidutinis skirtumas tarp $\mathrm{KM}(k_2 = 10)$ ir $\mathrm{KM}(k_2 = 100)$ yra tik $\approx 6\%$. Taigi, skaičiuojant KM, parametras $k_2$ neturi didelės įtakos.

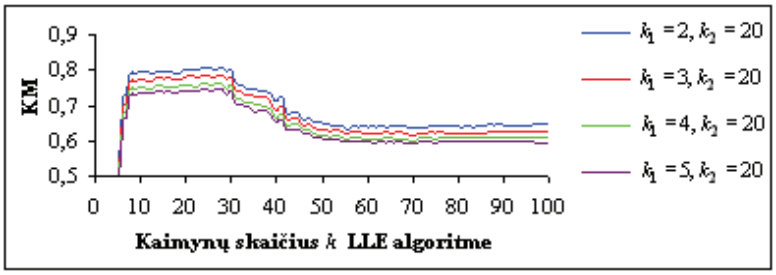
Po to, skaičiuojant KM reikšmes, buvo fiksuotas didesnysis artimiausių kaimynų skaičius $k_2$ ($k_2 = 20$), o mažesnysis artimiausių kaimynų skaičius $k_1$ buvo po truputį didinamas ($k_1 = \{2, 3, 4, 5\}$). Gautos keturios KM priklausomybės nuo LLE algoritmo parametro $k$, esant skirtingoms $k_1$ ir $k_2$ kombinacijoms: {2, 20}, {3, 20}, {4, 20} ir {5, 20}. Pastebėta, kad didinant parametro $k_1$ reikšmę, KM reikšmės mažėja (7.1b pav.). Tačiau nuo parametro $k_1$ reikšmės priklauso tik topologijos išlaikymo mato KM reikšmės dydis, bet KM priklausomybės nuo $k$ forma lieka panaši. Todėl galima naudoti bet kurią iš šių priklausomybių, ieškant tokio artimiausių kaimynų skaičiaus $k$ (arba jo intervalo) LLE algoritme, kad daugdara, ant kurios išsidėstę analizuojami taškai, būtų sėkmingai „išvyniojama“ mažesnės dimensijos erdvėje.

Analogiški tyrimai buvo atlikti ir su spiralinio cilindro taškais $(m = 1000, n = 3)$ (7.16b pav.). Gauti panašūs rezultatai kaip ir S-formos daugdaros atveju (7.2 pav.).

a)

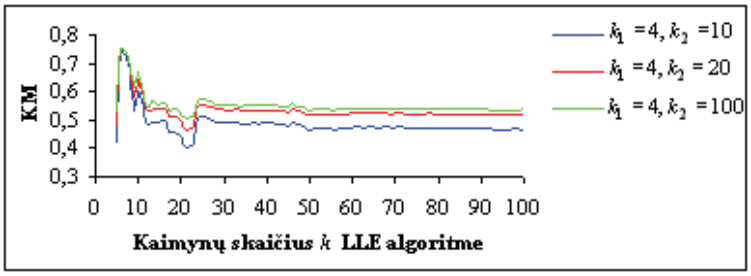



b)

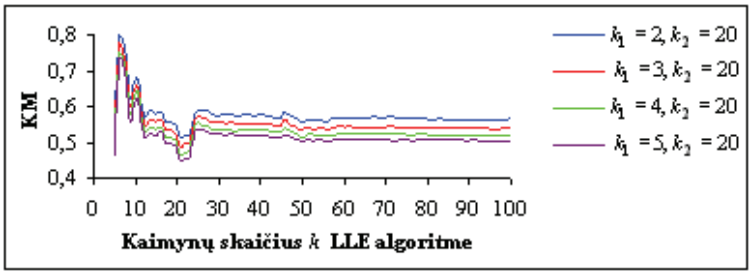



**7.1 pav.** KM priklausomybės nuo LLE parametro $k$, esant skirtingoms $k_1$ ir $k_2$ kombinacijoms, gautos transformavus S-formos daugdaros taškus LLE metodu į dvimatę erdvę

a)




b)




**7.2 pav.** KM priklausomybės nuo LLE parametro $k$, esant skirtingoms $k_1$ ir $k_2$ kombinacijoms, gautos transformavus spiralinio cilindro taškus LLE metodu į dvimatę erdvę

### 7.1.3. Kaimynystės klaidos

Straipsnyje (Lee and Verleysen 2007) pasiūlytas topologijos išlaikymo matas, paremtas vienoje erdvėje esančių kaimyninių taškų eilės numerių artumo išlaikymu ir kitoje erdvėje. Eilės numeris $\overline{\overline{r}}_X(i, j)$ apskaičiuojamas taip:

- Naudojant analizuojamą duomenų aibę $X$, apskaičiuojami Euklido atstumai nuo *i*-tojo taško $X_i$ iki visų likusių taškų, t. y. $\|X_i - X_s\|$, kai $s = \overline{1, m}$, $s \neq i$.
- Gauti atstumai surūšiuojami didėjimo tvarka. Tuomet ieškomas dydis $\overline{\overline{r}}_X(i, j)$ bus taško $X_j$ eilės numeris pagal surūšiuotus atstumus. Pastebėkime, kad, jei $j = \arg\min_{1 \le s \le m,\, s \neq i} \|X_i - X_s\|$, tada $\overline{\overline{r}}_X(i, j) = 1$.

Kaimynystės klaidos (*mean relative rank errors*) skaičiuojamos taip:

$$\text{a) MRRE}(X \to Y) = \frac{1}{c} \sum_{i=1}^{m} \sum_{j \in \mathrm{N}_K(X_i)} \frac{\left| \overline{\overline{r}}_X(i, j) - \overline{\overline{r}}_Y(i, j) \right|}{\overline{\overline{r}}_X(i, j)},$$

$$\text{b) MRRE}(Y \to X) = \frac{1}{c} \sum_{i=1}^{m} \sum_{j \in \mathrm{N}_K(Y_i)} \frac{\left| \overline{\overline{r}}_X(i, j) - \overline{\overline{r}}_Y(i, j) \right|}{\overline{\overline{r}}_Y(i, j)},$$

kur $\mathrm{N}_K(X_i)$ apibrėžia taško $X_i$ $K$ artimiausių kaimyninių taškų eilės numerių aibę. Normavimo faktorius toks:

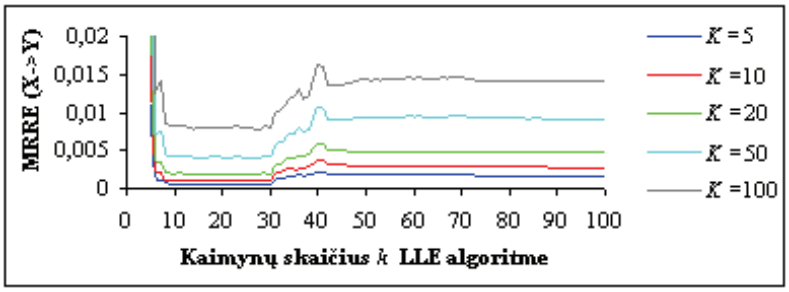
$$c = m \sum_{l=1}^{K} \frac{|2l - m - 1|}{l}.$$

Jis suveda klaidą į intervalą $[0; 1]$. Abu matai ($\text{MRRE}(X \to Y)$ ir $\text{MRRE}(Y \to X)$) artėja prie nulio, jei kiekvieno taško artimiausi $K$ kaimynai abiejose erdvėse randasi toje pačioje eilėje (vietoje). Taigi MRRE geriausia reikšmė lygi nuliui.

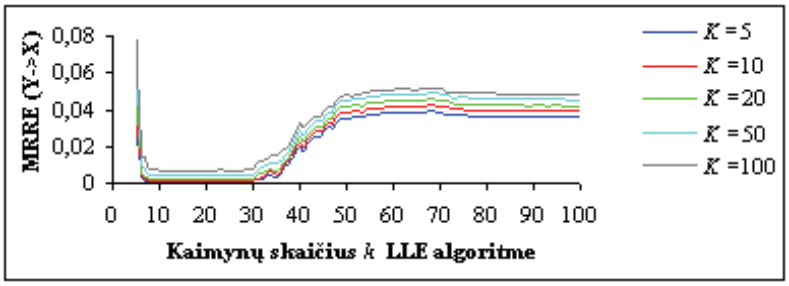
**Topologijos išlaikymo mato MRRE parametro *K* analizė**

MRRE formulėse yra tik vienas valdymo parametras – artimiausių kaimynų skaičius $K$ kiekvienam duomenų taškui. Siekiant išsiaiškinti, kokią įtaką šis parametras turi minėtoms klaidoms ($\text{MRRE}(X \to Y)$ ir $\text{MRRE}(Y \to X)$),

atliktas tyrimas su netiesinės dvimatės S-formos daugdaros taškais $(m = 1000, n = 3)$ (7.16a pav.). LLE algoritmas vykdytas su skirtingomis parametro $k$ reikšmėmis, $k \in [5;100]$, apskaičiuojant MRRE reikšmes kiekvienai pasirinktai $k$ reikšmei. Iš 7.3 ir 7.4 paveikslų matyti, jog, didinant artimiausių kaimynų skaičių $K$ kiekvienam duomenų taškui $n$-matėje (7.3 pav.) ir $d$-matėje (7.4 pav.) erdvėse, MRRE reikšmės didėja, tačiau visos MRRE priklausomybės nuo $k$ išlaiko panašią formą. Todėl galima naudoti bet kurią iš šių priklausomybių, ieškant tokio artimiausių kaimynų skaičiaus $k$ (arba jo intervalo) LLE algoritme, kad daugdara būtų sėkmingai „išvyniojama“ mažesnės dimensijos erdvėje.




**7.3 pav.** MRRE($X \to Y$) priklausomybės nuo LLE parametro $k$, esant skirtingoms $K$ reikšmėms MRRE algoritme, gautos transformavus S-formos daugdaros taškus LLE metodu į dvimatę erdvę




**7.4 pav.** MRRE($Y \to X$) priklausomybės nuo LLE parametro $k$, esant skirtingoms $K$ reikšmėms MRRE algoritme, gautos transformavus S-formos daugdaros taškus LLE metodu į dvimatę erdvę

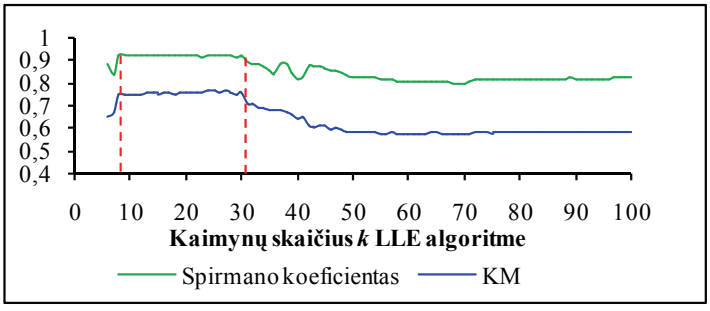
## 7.2. Topologijos išlaikymo matų palyginimas

Šiame skyrelyje palyginsime tris topologijos išlaikymo matus: Spirmano koeficientą, Konigo matą (KM) ir kaimynystės klaidas (MRRE) bei nustatysime, kurie iš jų geriau nusako daugdaros topologijos išlaikymą, transformavus ją LLE algoritmu į mažesnės dimensijos *d*-matę erdvę.

Tuo tikslu analizuojami įvairios struktūros netiesinių dvimačių daugdarų taškai. LLE algoritmu gavus jų projekcijas plokštumoje, apskaičiuojamos šių topologijos išlaikymo matų reikšmės su įvairiomis LLE parametro *k* reikšmėmis. Tokiu būdu gaunamos Spirmano koeficiento, KM ir MRRE priklausomybės nuo *k*.

Teigsime, kad pasirinkta LLE parametro *k* reikšmė yra tinkama, jei LLE metodu pavyks atrasti nežinomą daugdarą, t. y. LLE gerai išlaikys daugdaros topologiją, transformavus ją į mažesnio matavimo erdvę.

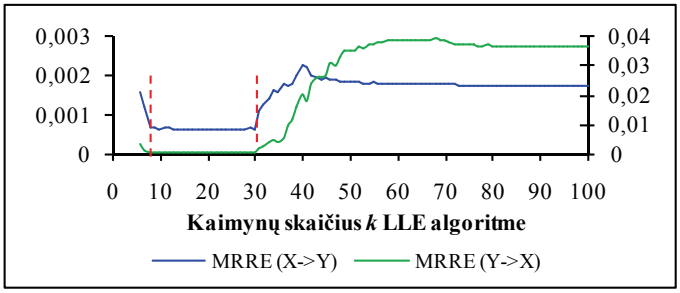
Pirmas tyrimas atliktas su netiesinės dvimatės S-formos daugdaros taškais ($m = 1000, n = 3$) (7.16a pav.). 5.2.1 skyrelyje buvo gauta Spirmano koeficiento priklausomybė nuo *k* ir parodyta, jog daugdara sėkmingai „išvyniojama“, kai $k \in [8; 30]$. Apskaičiavus KM reikšmes ($k_1 = 4, k_2 = 10$), kai $k \in [6; 100]$, gaunamas tas pats tinkamas artimiausių kaimynų skaičiaus intervalas, t. y. $k \in [8; 30]$ (7.5 pav.). Norime atkreipti dėmesį į tai, jog, nors šios priklausomybės ir pavaizduotos viename paveiksle, tačiau jų reikšmių tarpusavyje lyginti negalima, nes Spirmano koeficiento reikšmės visada gaunamos didesnės nei KM. Svarbu tik, kad sutaptų geriausias abiejų matų reikšmes atitinkantys artimiausių kaimynų skaičiaus intervalai.




**7.5 pav.** Spirmano koeficiento ir KM priklausomybės nuo LLE parametro *k*, gautos transformavus S-formos daugdaros taškus į plokštumą LLE metodu

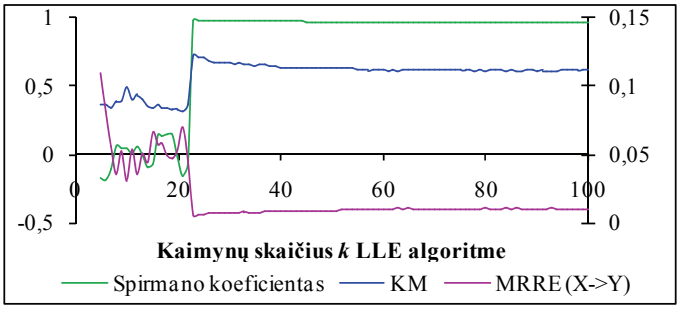
Apskaičiavus MRRE ($K = 5$) reikšmes, kai $k \in [6; 100]$, irgi gaunama, kad S-formos daugdara geriausiai „išvyniojama“, kai $k \in [8; 30]$ (7.6 pav.).

MRRE($X \to Y$) reikšmės randamos kairėje, o MRRE($Y \to X$) – dešinėje skalėje.

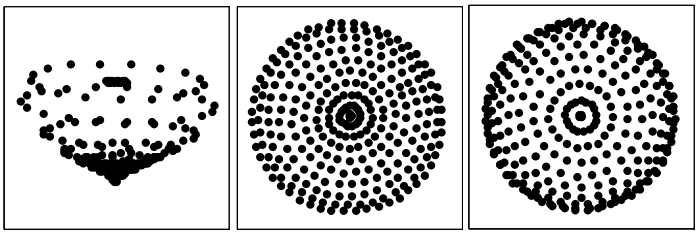



**7.6 pav.** MRRE priklausomybės nuo LLE parametro *k*, gautos transformavus S-formos daugdaros taškus į plokštumą LLE metodu

Antras tyrimas atliktas su pussferės taškais ($m = 294, n = 3$) (7.16e pav.). Iš 7.7 paveikslo matome, kad visų trijų topologijos išlaikymo matų (Spirmano koeficiento, KM, kai $k_1 = 4, k_2 = 10$, ir MRRE($X \to Y$), kai $K = 5$) reikšmės yra geriausios, kai $k \geq 23$. Spirmano koeficiento ir KM reikšmės randamos kairėje, o MRRE – dešinėje skalėje (taip bus visuose tolesniuose paveiksluose). Vadinasi, kai $k \geq 23$, pussferės lokali struktūra atskleidžiama geriausiai. 7.8 paveikslas iliustruoja pussferės taškų vizualizavimą plokštumoje, esant skirtingoms parametro *k* reikšmėms.



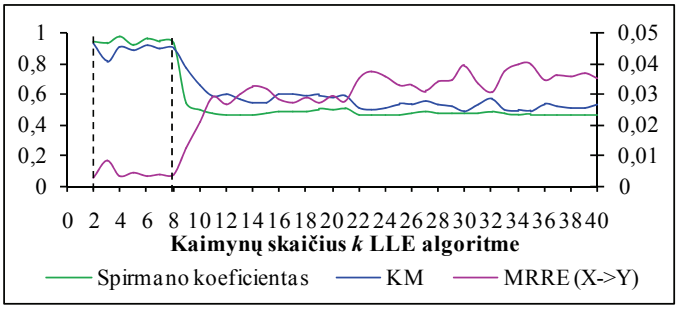

**7.7 pav.** Spirmano koeficiento, KM ir MRRE priklausomybės nuo LLE parametro *k*, gautos transformavus pussferės taškus į plokštumą LLE metodu



$k = 22$ $k = 23$ $k = 100$

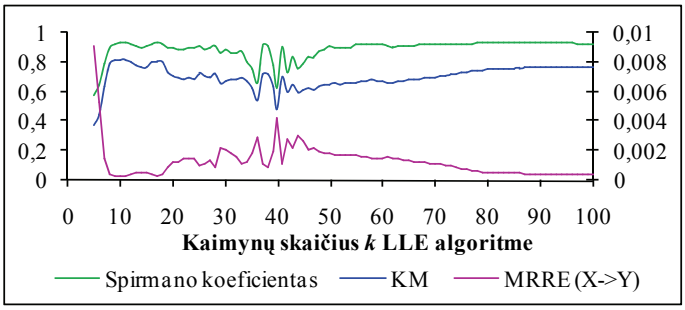
**7.8 pav.** Pussferės taškų projekcijos plokštumoje, gautos LLE metodu

Trečias tyrimas atliktas su sukamo ančiuko paveikslėlių duomenimis $(m = 72, n = 16384)$ (žr. 2.1 skyrelį). 5.2.1 skyrelyje buvo gauta Spirmano koeficiento priklausomybė nuo $k$ ir parodyta, jog teisingos duomenų projekcijos gaunamos, kai $k \in [2; 8]$. Apskaičiavus KM ($k_1 = 4, k_2 = 10$) ir MRRE($X \to Y$) ($K = 5$) reikšmes, kai $k \in [2; 40]$, gaunamas tas pats tinkamas artimiausių kaimynų skaičiaus intervalas kaip ir Spirmano koeficiento atveju (7.9 pav.).




**7.9 pav.** Spirmano koeficiento, KM ir MRRE priklausomybės nuo LLE parametro *k*, gautos transformavus sukamo ančiuko paveikslėlių duomenis atitinkančius daugiamačius taškus į plokštumą LLE metodu

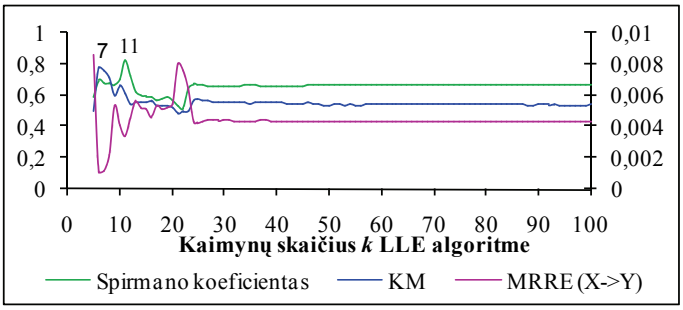
Spirmano koeficientas sėkmingai buvo pritaikytas duomenų topologijos išlaikymui įvertinti ir tiriant daugdaros „Dvynės viršukalnės“ taškus $(m = 2000, n = 3)$ (7.16c pav.) 5.2.2.3 skyrelyje. Iš 7.10 paveikslo matyti, jog ir šiuo atveju visi trys matai: Spirmano koeficientas, KM ($k_1 = 4, k_2 = 20$) ir MRRE($X \to Y$) ($K = 5$) taip pat įgyja savo geriausias reikšmes apytiksliai tuose pačiuose artimiausių kaimynų skaičiaus *k* intervaluose.



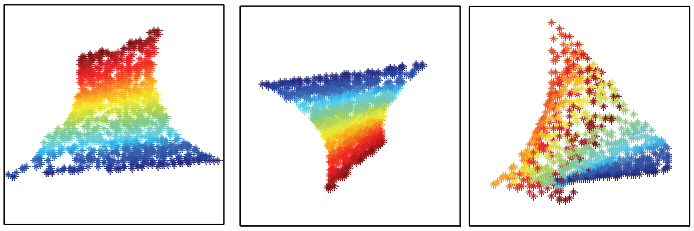

**7.10 pav.** Spirmano koeficiento, KM ir MRRE priklausomybės nuo LLE parametro *k*, gautos transformavus daugdaros „Dvynės viršukalnės“ taškus į plokštumą LLE metodu

Ištyrus šias daugdaras (S-formos daugdaros taškus, pussferės taškus, besisukančio ančiuko paveikslėlių duomenis, daugdaros „Dvynes viršukalnės“ taškus), galima teigti, kad visi trys topologijos išlaikymo matai – Spirmano koeficientas, KM ir MRRE – sėkmingai gali būti taikomi daugdarų topologijos išlaikymui įvertinti, vizualizavus jas plokštumoje LLE algoritmu. Tačiau atlikime tyrimus su sudėtingesnės struktūros daugdarų (spiralinio cilindro, sferos nuopjovos) taškais, ir patikrinkime nagrinėjamų topologijos išlaikymo matų tinkamumą jų atveju.

Spiralinis cilindras (*„swiss roll“*) ($m=1000, n=3$) (7.16b pav.) labai susuktas į spiralę. Taigi „išvynioti“ jį gana sunku. 7.11 paveiksle matyti, jog matai KM ($k_1=3, k_2=20$) ir MRRE($X \to Y$) ($K=5$) įgyja geriausias reikšmes, kai $k=6$, $k=7$. Tuo tarpu Spirmano koeficiento reikšmės nėra geriausios, esant šioms parametro *k* reikšmėms. Tačiau Spirmano koeficientas įgyja geriausią reikšmę, kai $k=11$, o KM ir MRRE neįgyja geriausios reikšmės šiuo atveju. Ši prieštara reiškia, kad kartais išvados, naudojant skirtingus matus, gali būti skirtingos. Kyla klausimas: kuris topologijos išlaikymo matas yra geriausias? Atsakymas slypi 7.12 paveiksle. Akivaizdu, kad spiralinis cilindras geriau „išvyniojamas“, kai $k=6$, $k=7$, o ne kai $k=11$. Vadinasi, Spirmano koeficientas nėra tinkamas šios daugdaros topologijos išlaikymui įvertinti po jos transformavimo į dvimatę erdvę.



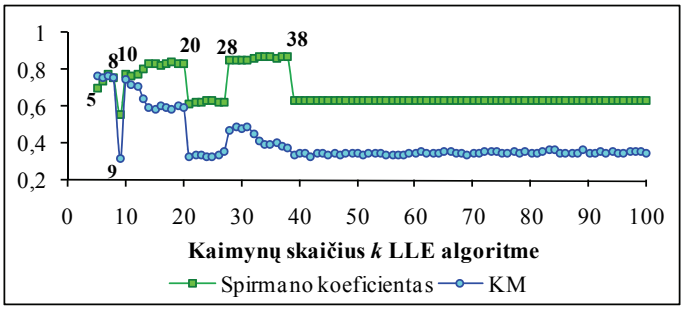

**7.11 pav.** Spirmano koeficiento, KM ir MRRE priklausomybės nuo LLE parametro *k*, gautos transformavus spiralinio cilindro taškus į plokštumą LLE metodu



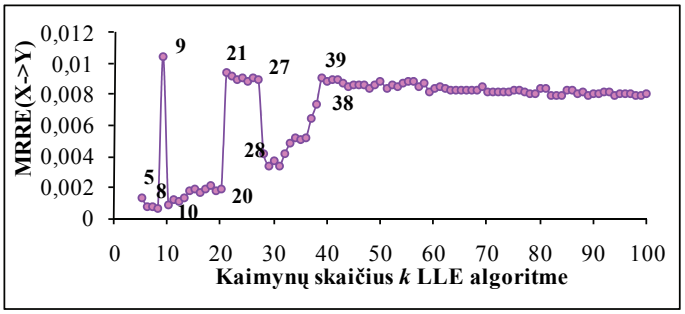
$k = 6$ $k = 7$ $k = 11$

**7.12 pav.** LLE metodu gautos spiralinio cilindro taškų projekcijos plokštumoje

Kitas tyrimas atliktas su sferos nuopjovos taškais (*„punctured sphere“*) $(m = 1000, n = 3)$ (7.16d pav.). Šią daugdarą išskleisti taip pat gana sudėtinga, nes ji artima sferai, kurios LLE metodu išskleisti nepavyksta (2.25d pav.). Iš 7.13 ir 7.14 paveikslų matyti, jog, kaip ir spiralinio cilindro atveju, nagrinėjami topologijos išlaikymo matai pateikia prieštaringus rezultatus: kai $k \in [5; 8] \cup [10; 20] \cup [28; 38]$, KM ($k_1 = 4, k_2 = 10$) ir MRRE($X \to Y$) ($K = 5$) reikšmės palaipsniui blogėja, o Spirmano koeficiento reikšmės, priešingai, palaipsniui gerėja, nors daugdaros globali struktūra atskleidžiama blogiau ir blogiau. Sferos nuopjovos taškų projekcijos pavaizduotos 7.15 paveiksle. Taigi ir šiuo atveju Spirmano koeficientas nėra tinkamas daugdaros topologijos išlaikymui įvertinti po jos transformavimo į dvimatę erdvę.

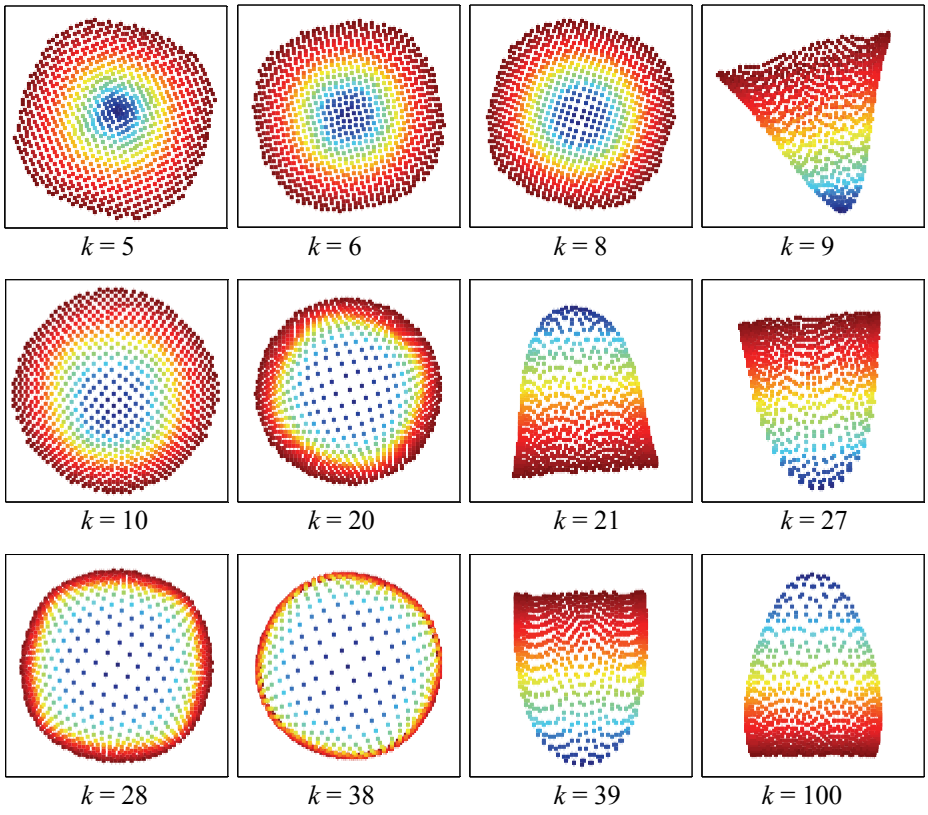



**7.13 pav.** Spirmano koeficiento ir KM priklausomybės nuo LLE parametro *k*, gautos transformavus sferos nuopjovos taškus į plokštumą LLE metodu




**7.14 pav.** MRRE priklausomybė nuo LLE parametro *k*, gauta transformavus sferos nuopjovos taškus į plokštumą LLE metodu

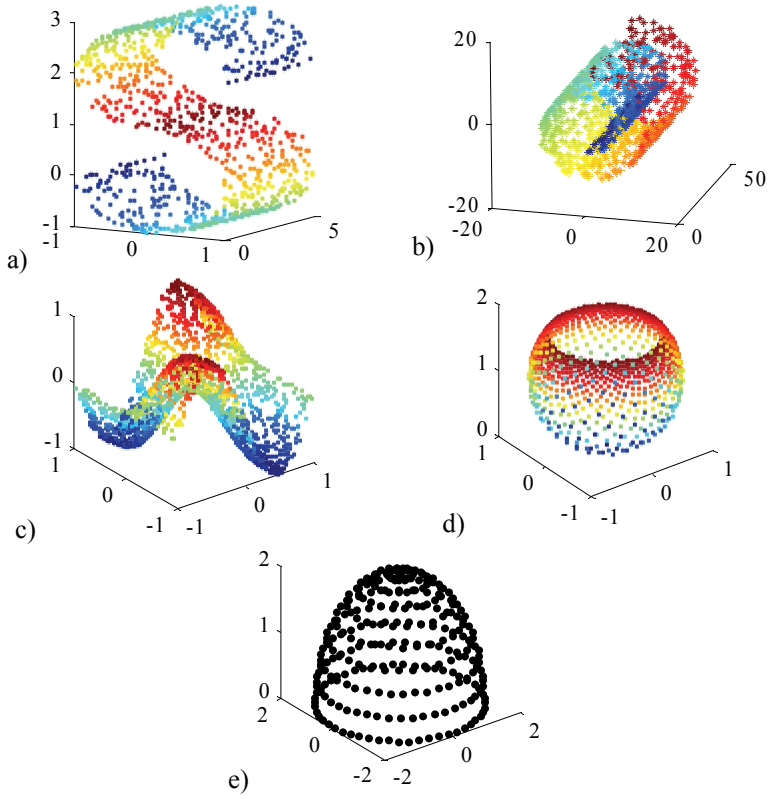
Vienas iš pagrindinių kriterijų, vertinant algoritmus, yra skaičiavimo laikas. Siekiant palyginti minėtus topologijos išlaikymo matus laiko atžvilgiu, analizuoti daugiamačiai taškai ant įvairių daugdarų, besiskiriančių savo struktūra ir taškų skaičiumi. LLE algoritmu gavus jų projekcijas plokštumoje, rastas laikas, reikalingas topologijos išlaikymo matams – Spirmano koeficientui, KM ir MRRE – apskaičiuoti. 7.1 lentelėje pateiktos šių matų apskaičiavimo laiko vidutinės reikšmės vienam artimiausių kaimynų skaičiui *k*. Akivaizdu, kad KM ir MRRE matai apskaičiuojami daug greičiau nei Spirmano koeficientas. Ieškant tinkamo artimiausių kaimynų skaičiaus *k* (ar jo intervalo) LLE metode, turėtų būti skaičiuojama topologijos išlaikymo mato priklausomybė nuo *k*. Kadangi paprastai nagrinėjamas didesnis artimiausių kaimynų skaičiaus intervalas, t. y. parametro *k* reikšmės gali būti imamos iki 100, tai labai svarbu, kad būtų kuo mažiau užtrunkama laiko, skaičiuojant mato reikšmę. Trumpesnis skaičiavimo laikas yra didelis KM ir MRRE privalumas lyginant su Spirmano koeficientu.




**7.15 pav.** Sferos nuopjovos taškų projekcijos plokštumoje, gautos LLE

**7.1 lentelė.** Vidutinis laikas matams apskaičiuoti

| Duomenys | Topologijos išlaikymo matai | | |
|---|---|---|---|
| | KM | MRRE | Spirmano koeficientas |
| Besisukančio ančiuko paveikslėlių duomenys | 0,18 s | 0,16 s | 0,20 s |
| Pussferės taškai | 0,05 s | 0,04 s | 1,36 s |
| S-formos daugdaros taškai | 0,46 s | 0,44 s | 81,61 s |
| Spiralinio cilindro taškai | 0,48 s | 0,45 s | 81,81 s |
| Sferos nuopjovos taškai | 0,47 s | 0,45 s | 81,86 s |
| „Dvynių viršukalnių“ taškai | 2,15 s | 2,12 s | 478,45 s |




**7.16 pav.** Trimačiai taškai, išsidėstę ant netiesinių dvimačių daugdarų : a) S-formos daugdaros, b) spiralinio cilindro, c) daugdaros „Dvynės viršukalnės“, d) sferos nuopjovos, e) pussferės

## 7.3. Septintojo skyriaus apibendrinimas ir išvados

Duomenų analizėje bei vizualizavime įprastas uždavinys – atvaizduoti duomenis iš labai didelės dimensijos erdvės į mažesnės dimensijos erdvę, taip, kad kiek galima tiksliau būtų išlaikyta duomenų struktūra. Daugdaros tipo daugiamačiai duomenys dažnai yra labai didelės dimensijos. Analizė, paremta daugdaros atpažinimu, leidžia daugiau sužinoti apie analizuojamą duomenų aibę. Šiame skyriuje taip pat nagrinėjome LLE metodą (lokaliai tiesinį vaizdavimą), kuris priklauso netiesiniams daugdaros atpažinimo metodams, kurie leidžia atrasti glodųjį mažesnės dimensijos paviršių, įterptą į didesnės dimensijos erdvę.

Siekiant kiekybiškai įvertinti daugdaros topologijos išlaikymą po jos transformavimo į mažesnės dimensijos erdvę, reikia naudoti kiekybinius skaitinius matus. Topologijos išlaikymui įvertinti sukurta daugybė įvairių matų. Šiame skyriuje tiriami trys matai: Spirmano koeficientas, Konigo matas (KM) ir kaimynystės klaidos (MRRE). Šių trijų topologijos išlaikymo matų lyginamajai analizei panaudoti du kriterijai – topologijos išlaikymo kokybė ir skaičiavimo sąnaudos.

Ištyrus paprastesnės struktūros daugdarų taškus (S-formos daugdaros, pussferės, daugdaros „Dvynės viršukalnės“ taškus bei realių paveikslėlių duomenis), pastebėta, jog visi trys matai – Spirmano koeficientas, KM ir MRRE – sėkmingai gali būti taikomi daugdarų topologijos išlaikymui įvertinti, vizualizavus jas plokštumoje LLE metodu. Tačiau, atlikus tyrimus su sudėtingesnės struktūros daugdarų, t. y. spiralinio cilindro, sferos nuopjovos taškais, paaiškėjo, jog tik KM ir MRRE tinka šių daugdarų topologijos išlaikymui įvertinti. KM ir MRRE matų apskaičiavimas yra greitesnis, nes šie kriterijai naudoja tiktai Euklido atstumus. Tuo tarpu, Spirmano koeficientas naudoja geodezinius atstumus, kurių apskaičiavimas reikalauja daugiau laiko sąnaudų. Be to, Spirmano koeficientas vertina visus atstumus tarp tiriamos duomenų aibės taškų, o KM ir MRRE – tarp nedidelio kaimyninių taškų skaičiaus. Taigi Spirmano koeficientas stengiasi atsižvelgti į globalią daugdaros struktūrą. Tačiau kai kuriais atvejais tai nėra optimalu, nes gali būti prarastos kai kurios daugdaros lokalios savybės. Šiame skyriuje parodyta, kad KM ir MRRE matai visada gerai nusako daugdaros topologijos išlaikymą po jos transformacijos į mažesnio matavimo erdvę.

# 8

# Bendrosios išvados

1. Trianguliacijos metodo ir Sammono algoritmo bei jų jungimo tyrimas atskleidė faktus:
   - Naudoti vien tik trianguliacijos metodą daugiamačiams duomenims vizualizuoti nėra pakankama, nes, eksperimentiškai ištyrus trianguliacijos metodo realizacijas, naudojančias antrojo arčiausiojo kaimyno ir atramos taško metodus atraminiams taškams parinkti, nustatyta, jog abiem atvejais projekcijos paklaida labai priklauso nuo taškų atvaizdavimo sekos. Be to, paklaida gana didelė.
   - Trianguliacijos metodas yra pakankamai greitas, bet projekcijos paklaida didelė. Sammono algoritmu gauta projekcijos paklaida nedidelė, tačiau jis yra gana lėtas. Sammono ir trianguliacijos metodų junginį verta naudoti, kai reikia greitai atvaizduoti naujus analizuojamos aibės taškus neprarandant didelio tikslumo.
2. Ištyrus santykinės perspektyvos metodą (RPM), padarytos šios išvados:
   - Pradinio RPM algoritmo rezultatai labai priklauso nuo parametrų $w$ (stačiakampio plotis), $h$ (stačiakampio aukštis) ir $\tilde{r}$ (pradinis

mokymo greitis). Deja, literatūroje nėra suformuluotų aiškių taisyklių, kaip šiuos parametrus parinkti.

- Darbe pasiūlytame algoritme nėra didelės priklausomybės nuo parametrų $w$ ir $h$. Tyrimai parodė, jog naudojant mūsų algoritmą, energijos santykinė reikšmė daugiausia pakinta 0,4%, tuo tarpu, naudojant RPM pradinį algoritmą, ji gali kisti net iki 27%. Tačiau, atsisakius parametro $\tilde{r}$, potencinė energija nekonverguoja į minimumą. Tyrimų metu pastebėta, kad, atlikus apie 100 iteracijų, procesas stabilizuojasi.

3. Lokaliai tiesino vaizdavimo (LLE) metodo tyrimas parodė:

   - LLE algoritmo parametrai – artimiausių kaimynų skaičius kiekvienam duomenų taškui ir lokaliosios Gramo matricos reguliarizacijos parametras – labai įtakoja duomenų vizualizavimo kokybę.

   - Pasiūlytas naujas būdas artimiausių kaimynų skaičiui parinkti leido nustatyti tinkamą artimiausių kaimynų skaičiaus intervalą, o ne tik vieną skaičių. Eksperimentai parodė, kad kiekybinis matas – Spirmano koeficientas – yra tinkamas duomenų topologijos išlaikymui įvertinti, duomenis transformavus LLE metodu į mažesnės dimensijos erdvę. Tam, kad Spirmano koeficientas tinkamai atspindėtų gautas projekcijas, skaičiuojant jo reikšmę reikia $n$-matėje erdvėje vertinti geodezinius, o ne Euklido atstumus, ir geodezinių atstumų skaičiavimo algoritme fiksuoti gana mažą artimiausių kaimynų skaičių ($\leq 10$).

   - Pasiūlytas naujas algoritmas lokaliajai Gramo matricai reguliarizuoti (R2) pateikia panašius rezultatus kaip ir algoritmas (R1), pasiūlytas Roweis ir Saul. Taigi abu algoritmai gali būti alternatyviai naudojami LLE metode reguliarizuojant Gramo matricą. Abu algoritmai pateikia panašią galimybę analizei, kiekvienas iš jų turi po vieną valdymo parametrą: $t$ R1 atveju ir $D$ R2 atveju. Tačiau parametro $D$ privalumas lyginant su $t$ yra tai, kad parametras $D$ turi realią prasmę (jis yra reguliarizuotos Gramo matricos determinantas), o $t$ yra tiktai tam tikras daugiklis.

4. Detaliai ištyrus Laplaso matricos tikrinių žemėlapių metodą (LE), padarytos šios išvados:

   - LE algoritmu gautų projekcijų kokybė labai priklauso nuo parinktų valdymo parametrų reikšmių – artimiausių kaimynų skaičiaus $k$

kiekvienam duomenų taškui ir šiluminio branduolio parametro $T$, naudojamo Gauso branduolio funkcijoje svoriams apskaičiuoti. Sunku teisingai parinkti $T$, jei parametro $k$ reikšmės didelės.

- Pasiūlyta LE algoritmo modifikacija, turinti tris valdymo parametrus ($T$, $k^*$, $w^*$). Šios modifikacijos privalumas yra tai, kad šiluminio branduolio parametro $T$ reikšmės parinkinėti nereikia, nes sudaryta galimybė ją įvertinti automatiškai. Kai pasirinktas tinkamas maksimalus artimiausių kaimynų skaičius $k^*$, parametro $w^*$ reikšme gali būti bet koks realus skaičius iš intervalo $(0;1)$. Tačiau, norint pasiekti kuo geresnę vizualizavimo kokybę, patartina pasirinkti parametro $w^*$ reikšmę kuo mažesnę, pvz., 0,1. Taigi mūsų LE modifikacijoje lieka tik vienas svarbus valdymo parametras $k^*$ – maksimalus artimiausių kaimynų skaičius. Kintamas (iš anksto nefiksuotas) artimiausių kaimynų skaičius kiekvienam taškui yra šios modifikacijos unikalumas.

5. Siekiant kiekybiškai įvertinti daugdaros topologijos išlaikymą, transformavus ją į mažesnės dimensijos erdvę, reikia naudoti kiekybinius skaitinius matus. Išanalizavus tris topologijos išlaikymo matus: Spirmano koeficientą, Konigo matą (KM) ir kaimynystės klaidas (MRRE), nustatyta:

   - Visi trys matai – Spirmano koeficientas, KM ir MRRE – sėkmingai gali būti taikomi paprastos struktūros dvimačių daugdarų topologijos išlaikymui įvertinti, transformavus jų taškus į plokštumą LLE metodu. Tačiau, atlikus tyrimus su sudėtingesnės struktūros daugdarų taškais, paaiškėjo, jog tik KM ir MRRE tinka šių daugdarų topologijos išlaikymui įvertinti.

   - KM ir MRRE matų apskaičiavimas yra žymiai greitesnis nei Spirmano koeficiento, nes šie kriterijai naudoja tiktai Euklido atstumus. Tuo tarpu, Spirmano koeficientas naudoja geodezinius atstumus, kurių apskaičiavimas reikalauja daugiau laiko sąnaudų. Be to, Spirmano koeficientas vertina visus atstumus tarp tiriamos duomenų aibės taškų, o KM ir MRRE – tarp nedidelio kaimyninių taškų skaičiaus. Taigi Spirmano koeficientas stengiasi atsižvelgti į globalią daugdaros struktūrą. Tačiau kai kuriais atvejais tai nėra optimalu, nes gali būti prarastos kai kurios daugdaros lokalios savybės.

# Literatūros sąrašas


Abusham, E. E.; Ngo, D.; Teoh, A. 2005. Fusion of locally linear embedding and principal component analysis for face recognition (FLLEPCA). *Lecture Notes in Computer Science*, 3687, 326–333.

Adler, R. J.; Taylor, J. E. 2007. *Random Fields and Geometry*. Springer. ISBN-978-0-387-48112-8.

Aggarwal, C. C.; Hinneburg, A.; Keim, D. A. 2001. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science*, 1973, 420–434. ISSN 0302-9743.

Basalaj, W. 1999. Incremental multidimensional scaling method for database visualization. In *Proceedings of Visual Data Exploration and Analysis VI*, *SPIE'99*, 3647, 149–158.

Belkin, M.; Niyogi, P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14*, 585–591. Cambridge, MA: MIT Press.

Belkin, M.; Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6), 1373–1396. ISSN 0899-7667.

Bengio, Y.; Paiement, J. F.; Vincent, P.; Delalleau, O.; Roux, N. L.; Ouimet, M. 2004. Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. In *Proceedings of Neural Information Processing Systems 16*, 177–184. Cambridge, MA: MIT Press.

Bernatavičienė, J.; Dzemyda, G.; Kurasova, O.; Marcinkevičius, V. 2006. Optimal decisions in combining the SOM with nonlinear projection methods. *European Journal of Operation Research*, 173(3), 729–745. ISSN 0377-2217.

Bernatavičienė, J.; Dzemyda, G.; Marcinkevičius, V. 2007. Diagonal majorization algorithm: properties and efficiency. *Information Technology and Control*, 36(4), 353–358. ISSN 1392 – 124X.

Bernatavičienė, J. 2008. *Vizualios žinių gavybos metodologija ir jos tyrimas.* Daktaro disertacija. Vilnius: Technika.

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

Biswas, G.; Jain, A. A.; Dubes, R. C. 1981. Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6), 701–708. ISSN 0162-8828.

Borg, I.; Groenen, P. 2005. *Modern Multidimensional Scaling: Theory and Applications* (2nd ed.). New York, USA: Springer-Verlag.

Brand, M. 2003. Charting a manifold. *Advances in Neural Information Processing Systems 15*, 961–968. Cambridge, MA: MIT Press.

Camastra, F.; Vinciarelli, A. 2002. Estimating the intrinsic dimension of data with a fractal-based approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10), 1404–1407.

Chang, Y.; Hu, C.; Turk, M. 2004. Probabilistic expression analysis on manifolds. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 520–527. Washington.

Costa, J. A.; Hero, A. O. 2004. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, 52(8), 2210–2221. ISSN 1053-587X.

Cristianini, N.; Taylor, J. S. 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press.

DeCoste, D. 2001. Visualizing Mercer kernel feature spaces via kernelized locally linear embeddings. In *Proceedings of the 8th International Conference on Neural Information Processing (ICONIP 2001).* Shanghai, China.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.

Dollar, P.; Rabaud, V.; Belongie, S. 2007. Learning to traverse image manifolds. *Advances in Neural Information Processing Systems 19*, 361–368. Cambridge, MA: MIT Press.

Donoho, D. L.; Grimes, C. 2003. Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. In *Proceedings of the National Academy of Sciences*, 100(10), 5591–5596.

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

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.; Žilinskas, J. 2008. *Daugiamačių duomenų vizualizavimo metodai.* Vilnius: Mokslo aidai.

Elgammal, A.; Lee, C. S. 2004a. Separating style and content on a nonlinear manifold. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 478–485.

Elgammal, A.; Lee, C. S. 2004b. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 681–688.

Estevez, P. A.; Figueroa, C. J.; Saito, K. 2005. Cross-entropy embedding of high-dimensional data using the neural gas model. *Neural Networks*, 18(5-6), 727–737.

Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 179–188.

Floyd, R. W. 1962. Algorithm 97: shortest path. *Communications of the ACM*, 5(6), 345.

Gantmacher, F. R. 1988. *The Theory of Matrices.* Moscow: Nauka (in Russian).

Ge, S. S.; Yang, Y.; Lee, T. H. 2008. Hand gesture recognition and tracking based on distributed locally linear embedding. *Image and Vision Computing*, 26(12), 1607–1620 .

Goodhill, G. J.; Sejnowski, T. J. 1996. Quantifying neighbourhood preservation in topographic mappings. In *Proceedings of the Third Joint Symposium on Neural Computation*, 6, 61–82. Pasadena: California Institute of Technology.

Graham, R. L.; Hell, P. 1985. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7, 43–57.

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. New York: ACM Press.

Grinstein, G. G.; Ward, M. O. 2002. Introduction to data visualization. *Information Visualization in Data Mining and Knowledge Discovery*.

Hadid, A.; Kouropteva, O.; Pietikäinen, M. 2002. Unsupervised learning using locally linear embedding: experiments with face pose analysis. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, 1, 111–114. Washington, DC, USA: IEEE Computer Society.

He, X.; Niyogi, P. 2004. Locality preserving projections. In *Proceedings of Advances in Neural Information Processing Systems (NIPS 2003) 16.* Cambridge, MA: MIT Press.

He, X. F.; Yan, S. C.; Hu, Y. X.; Niyogi, H. G.; Zhang, H. J. 2005. Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), 328–340.

Hellemaa, P. 1998. *The Development of Coastal Dunes and Their Vegetation in Finland.* Dissertation, Fenia 176: 1, Helsinki. ISSN 0015-0010.

Hoffman, P. E.; Grinstein, G. G. 2002. A survey of visualizations for high-dimensional data mining. *Information Visualization in Data Mining and Knowledge Discovery*.

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

Yang, L. 2004. Sammon's nonlinear mapping using geodesic distances. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 2, 303–306. Washington, DC: IEEE Computer Society.

Yang, M. -H. 2002. Face recognition using extended isomap. In *Proceedings of IEEE International Conference on Image Processing (ICIP 2002)*, 2, 117–120.

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.

Yin, J.; Hu, D.; Zhou, Z. 2007. Growing locally linear embedding for manifold learning. *Journal of Pattern Recognition Research*, 2(1), 1–16.

Jain, V.; Saul, L. K. 2004. Exploratory analysis and visualization of speech and music by locally linear embedding. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5, 984–987. Piscataway: IEEE .

Jenkins, O. C.; Mataric, M. J. 2004. A spatio-temporal extension to Isomap nonlinear dimension reduction. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, 69, 441–448. New York: ACM.

Jolliffe, I. T. 1989. *Principal Component Analysis.* New York: Springer-Verlag.

Kaski, S. 1997. *Data Exploration Using Self-Organizing Maps.* PhD thesis, Helsinki University of Technology, Department of Computer Science and Engineering.

Kegl, B. 2005. Intrinsic dimension estimation using packing numbers. *Advances in Neural Information Processing Systems 15*, 681–688. Cambridge, MA: MIT Press.

Keim, D. A.; Kriegel, H. P. 1996. Visualization techniques for mining large databases: a comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), 923–938.

Keim, D. A.; Ward, M. 2003. Visualization. In: M. Berthold, D. J. Hand (Eds.). *Intelligent Data Analysis: an Introduction*, 403–427. Springer-Verlag.

Klock, H.; Buhmann, J. M. 2000. Data visualization by multidimensional scaling: a deterministic annealing approach. *Pattern Recognition*, 33(4), 651–669.

Kohonen, T. 2001. Self-Organizing Maps (3rd ed.). *Springer series in information sciences*, 30. 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.

Kouropteva, O.; Okun, O.; Pietikainen, M. 2002. Selection of the optimal parameter value for the locally linear embedding algorithm. In *Proceedings of 1st International Conference on Fuzzy Systems and Knowledge Discovery: Computational Intelligence for the E-Age (FSKD'02)*, 1, 359–363.

Kouropteva, O.; Okun, O.; Pietikainen, M. 2005. Incremental locally linear embedding. *Pattern Recognition*, 38(10), 1764–1767.

Kruskal, J. B. 1956. On the shortest spanning subtree of a graph and the travelling salesman problem. In *Proceedings of the American Mathematical Society*, 7, 48–50.

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

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

Lafon, S.; Lee, A. B. 2006. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9), 1393–1403.

Law, M. H.; Jain, A. K. 2006. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3), 377–391.

Lee, J. A.; Verleysen, M. 2007. *Nonlinear Dimensionality Reduction.* New York: Springer.

Lee, R. C.; Slagle, J. R.; Blum, H. 1977. A triangulation method for the sequential mapping of points from *n*-space to two-space. *IEEE Transactions on Computers*, 26(3), 288–292.

Levina, E.; Bickel, P. J. 2005. Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems 17*, 777–784. Cambridge, MA: MIT Press.

Li, H.; Li, X. 2004. Gait analysis using LLE. In *Proceedings of the 7th International Conference on Signal Processing (ICSP'04)*, 3, 1423–1426. Piscataway: IEEE.

Li, H. -G.; Shi, C. -P.; Li, X. -G. 2005. LLE based gait recognition. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 7, 4516–4521.

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

Li, S. Z.; Xiao, R.; Li, Z. Y.; Zhang, H. J. 2001. Nonlinear mapping from multi-view face patterns to a Gaussian distribution in a low dimensional space. In *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, 47–54.

Liou, C.-Y.; Kuo, Y.-T. 2002. Economic states on neuronic maps. In *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02)*, 2, 787–791.

Liu, K.; Weissenfeld, A.; Ostermann, J. 2006. Parameterization of mouth images by LLE and PCA for image-based facial animation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal (ICASSP 2006)*, 5, 461–464.

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 tyrimai*. Daktaro disertacija. Vilnius: Technika.

Mekuz, N.; Bauckhage, C.; Tsotsos, J. K. 2005. Face recognition with weighted locally linear embedding. In *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, 290–296.

Michalewicz, Z. 1996. Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer.

Nadler, B.; Lafon, S.; Coifman, R. R.; Kevr, I. G. 2006. Diffusion maps, spectral clustering and eigenfunction of Fokker-planck operators. In: Y. Weiss; B.

Schölkopf; J. Platt (Eds.). *Advances in Neural Information Processing Systems (NIPS 2005) 18*. Cambridge, MA: MIT Press.

Naud, A.; Duch, W. 2000. Interactive data exploration using MDS mapping. In *Proceedings of the 5th conference: Neural networks and soft computing*, 255–260. Poland, Zakopane.

Nene, S. A.; Nayar, S. K.; Murase, H. 1996. *Columbia Object Image Library (COIL-20)*. Technical Report CUCS-005-96, http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

Ng, A. Y.; Jordan, M.; Weiss, Y. 2002. On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems 14*, 849–856.

Niskanen, M.; Silven, O. 2003. Comparison of dimensionality reduction methods for wood surface inspection. In *Proceedings of the 6th International Conference on Quality Control by Artificial Vision*, 5132, 178–188.

Opitz, O.; Hilbert, A. 2000. Visualization of multivariate data by scaling and property fitting. In: W. Gaul; O. Opitz; M. Schader (Eds.). *Data Analysis: Scientific Modeling and Practical Application*, 505–514.

Patwari, N.; Hero, A. O. 2004. Manifold learning algorithms for localization in wireless sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3, 857–860.

Pena, J.; González-Castaño, D. M.; Gómez, F.; Gago-Arias, A.; González-Castaño, F. J.; Rodríguez-Silva, D.; Gómez, A.; Mouriño, C.; Pombar, M.; Sánchez, M. 2009. eIMRT: a web platform for the verification and optimization of radiation treatment plans. *Journal of Applied Clinical Medical Physics*, 10(3), 205–220.

Podlipskytė, A. 2004. *Daugiadimensinių duomenų vizualizacija ir jos taikymas biomediciniņių duomenų analizei*. Daktaro disertacija. Kaunas.

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

Prim, R. C. 1957. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36, 1389–1401.

Ridder, D. D.; Kouropteva, O.; Okun, O.; Pietikainen, M.; Duin, R. P. 2003. Supervized locally linear embedding. *Computer Science*, 2714, 333–341.

Ridder, D. D.; Loog, M.; Reinders, M. J. 2004. Local Fisher embedding. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 2, 295–298. Washington, DC, USA: IEEE Computer Society.

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

Roweis, S. T.; Saul, L. K.; Hinton, G. E. 2002. Global coordination of local linear methods. In: T. G. Dietterich; S. Becker; Z. Ghahramani (Eds.). *Advances in Neural Information Processing Systems 14.* Cambridge, MA: MIT Press.

Sachinopoulou, A. 2001. *Multidimensional Visualization.* Julkaisuvuosi. ISBN 951-38-5920-7

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

Saul, L. K.; Roweis, S. T. 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4, 119–155. ISSN:1532-4435.

Sha, F.; Saul, L. K. 2005. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, 119, 784–791. New York, NY, USA: ACM Press.

Shepard, D. M.; Ferris, M. C.; Olivera, G. H.; Mackie, T. R. 1999. Optimizing the delivery of radiation therapy to cancer patients. *SIAM Review*, 41(4), 721–744.

Siegel, S.; Castellan, N. J. 1988. *Nonparametric Statistics for the Behavioral Sciences* (2nd ed.). New York: McGraw-Hill.

Silva, V. D.; Tenenbaum, J. B. 2003a. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15*, 705–712. Cambridge, MA: MIT Press.

Silva, V. D.; Tenenbaum, J. B. 2003b. Unsupervised learning of curved manifolds. *Nonlinear Estimation and Classification*, *Lecture Notes in Statistics*, 171.

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

Taylor, P. 2003. Statistical methods. In: M. Berthold; D. J. Hand (Eds.). *Intelligent Data Analysis: An Introduction*, 69–129. Springer-Verlag.

Teh, Y. W.; Roweis, S. 2003. Automatic aligment of local representations. In: S. Thrun; S. Becker; K. Obermayer (Eds.). *Advances in Neural Information Processing Systems 15.* Cambridge, MA: MIT Press.

Tenenbaum, J. B.; Silva, V. D.; Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323.

Teng, L.; Li, H.; Fu, X.; Chen, W.; Shen, I. -F. 2005. Dimension reduction of microarray data based on local tangent space alignment. In *Proceedings of the 4th IEEE International Conference on Cognitive Informatics (ICCI'05)*, 154–159. Washington, DC, USA: IEEE Computer Society.

Tikhonov, A. N.; Arsenin, V. A. 1977. *Solution of Ill-posed Problems.* Washington: Winston & Sons. ISBN 0-470-99124-0.

Varini, C.; Nattkemper, T. W.; Degenhard, A.; Wismuller, A. 2004. Breast MRI data analysis by LLE. In *Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, 3, 2449–2454.

Venna, J.; Kaski, S. 2001. Neighborhood preservation in nonlinear projection methods: an experimental study. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'01)*, *Lecture Notes In Computer Science*, 2130, 485–491. ISBN 3-540-42486-5.

Verbeek, J. J.; Roweis, S. T.; Vlassis, N. 2004. Non-linear CCA and PCA by alignment of local models. *Advances in Neural Information Processing Systems 16*, 297–304. Cambridge, MA: MIT Press.

Wegenkittl, R.; Loffelmann, H.; Groller, E. 1997. Visualizing the behavior of higher dimensional dynamical systems. In *Proceedings of IEEE Visualization 1997*, 119–125. Los Alamitos, CA, USA: IEEE Computer Society Press.

Weinberger, K. Q.; Packer, B. D.; Saul, L. K. 2005. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 381–388. Society for Artificial Intelligence and Statistics.

Weinberger, K. Q.; Saul, L. K. 2006. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1), 77–90. ISSN 0920-5691.

Weinberger, K. Q.; Sha, F.; Zhu, Q.; Saul, L. K. 2007. Graph regularization for maximum variance unfolding, with an application to sensor localization. *Advances in Neural Information Processing Systems 19*, 1489–1496. Cambridge, MA: MIT Press.

Wong, P. C.; Bergeron, R. D. 1997. 30 years of multidimensional multivariate visualization. *Scientific Visualization: Overviews, Methodologies and Techniques*, 3–33.

Xiao, J.; Zhou, Z. T.; Hu, D. W.; Yin, J.; Chen, S. 2005. Self-organized locally linear embedding for nonlinear dimensionality reduction. *Lecture Notes in Computer Science*, 3610, 101–109. ISSN 0302-9743.

Zhang, J.; Li, S. Z.; Wang, J. 2004. Nearest manifold approach for face recognition. In *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition*, 223–228.

Zhang, Z.; Zha, H. 2004. Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1), 313–338.

Zhao, Q.; Zhang, D.; Lu, H. 2005. Supervised LLE in ICA space for facial expression recognition. In *Proceedings of International Conference on Neural Networks and Brain (ICNN&B '05)*, 3, 1970–1975.

Zhu, L.; Zhu, S. A. 2006. Face recognition based on extended locally linear embedding. In *Proceedings of 2006 1ST IEEE Conference on Industrial Electronics and Applications*, 1–4.

Žilinskas, A.; Žilinskas, J. 2007. Two level minimization in multidimensional scaling. *Journal of Global Optimization*, 38(4), 581–596. ISSN 0925-5001.

Žilinskas, A.; Žilinskas, J. 2009. Branch and bound algorithm for multidimensional scaling with city-block metric. *Journal of Global Optimization*, 43(2-3), 357–372. ISSN 0925-5001.

# Autorės publikacijų sąrašas disertacijos tema

## Straipsniai recenzuojamuose periodiniuose mokslo leidiniuose

Karbauskaitė, R.; Dzemyda, G. 2009a. Topology preservation measures in the visualization of manifold-type multidimensional data. *Informatica*, 20(2), 235–254. ISSN 0868-4952 (*ISI Web of Science*).

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. ISSN 1392-124X (*ISI Web of Science*).

Karbauskaitė, R.; Dzemyda, G. 2006. Multidimensional data projection algorithms saving calculations of distances. *Information Technology and Control*, 35(1), 57–64. ISSN 1392-124X (*VINITI, INSPEC*).

Karbauskaitė, R.; Marcinkevičius, V.; Dzemyda, G. 2006. Testing the relational perspective map for visualization of multidimensional data. *Technological and Economic Development of Economy*, 12(4), 289–294. ISSN 1392-8619 (*ASCE Civil Engineering Abstracts, Business Source Complete, Business Source Premier, Current Abstracts, ICONDA, SCOPUS, TOC Premier*).

## Straipsniai konferencijų pranešimų rinkiniuose

Karbauskaitė, R.; Dzemyda, G. 2009b. Dependence of the Laplacian Eigenmaps method and its modification on the parameters. In *Proceedings of the 13th International Conference "Applied Stochastic Models and Data Analysis" (ASMDA-2009): selected papers*, 263–268. Vilnius: Technika. ISBN 978-9955-28-463-5.

Karbauskaitė, R.; Dzemyda, G.; Marcinkevičius, V. 2008. Selecting a regularization parameter in the locally linear embedding algorithm. In *Proceedings of the 20th international EURO mini conference "Continuous optimization and knowledge-based technologies" (EurOPT'2008): selected papers*, 59–64. Vilnius: Technika. ISBN 978-9955-28-283-9 (*Conference Proceedings Citation Index*).

2010 08 10. 11,5 sp. l. Tiražas 20 egz.
Parengė spaudai ir išleido Matematikos ir informatikos institutas
Akademijos g. 4, LT-08663 Vilnius.
Interneto svetainė: http://www.mii.lt

Spausdino UAB „Kauno technologijos universiteto spaustuvė"
Studentų g.54, LT-51424 Kaunas