KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

Giedrė Vaišnoraitė

POŽYMIŲ ERDVĖS MAŽINIMO METODŲ KOKYBĖS TYRIMAS

Magistro darbas

Vadovai
doc. dr. A. Lipnickas
doc. dr. R. Markauskas

KAUNAS, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU
Katedros vedėjas
prof. dr. J.Rimas
2007 06 08

POŽYMIŲ ERDVĖS MAŽINIMO METODŲ KOKYBĖS TYRIMAS

Taikomosios matematikos magistro baigiamasis darbas

Vadovai
( ) doc. dr. A. Lipnickas
( ) doc. dr. R. Markauskas
2007 06 05

Recenzentas
( ) doc. dr. V.Bagdonas
2007 06 05

Atliko
FMMM 5 gr. stud.
( ) G. Vaišnoraitė
2007 06 05

KAUNAS, 2007

KVALIFIKCINĖ KOMISIJA

Pirmininkas: Leonas Saulis, habil. dr., Vilniaus Gedimino technikos universiteto profesorius

Sekretorius: Eimutis Valakevičius, docentas (KTU)

Nariai: Algimantas Jonas Aksomaitis, profesorius (KTU)

Arūnas Barauskas, dr., UAB „Elsis“ generalinio direktoriaus pavaduotojas

Vytautas Janilionis, docentas (KTU)

Zenonas Navickas, profesorius (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Rimantas Rudzkis, habil. dr., banko „NORD/LB“ vyriausiasis analitikas

Vaisnoraite G. Comparison of methods for features space reduction: Master's work in applied mathematics / supervisors: dr. assoc. prof. R. Markauskas; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology., dr. assoc. prof. A. Lipnickas; Mechatronics Centre for Studies and Research, Kaunas University of Technology. – Kaunas, 2007. – 67 p.

SUMMARY

The process of finding features that meet the given constrains out of a large group of features is called feature reduction. The reduction concept can be divided into feature selection and feature extraction techniques. The feature selection approach selects the independent features that provide sufficient information for a satisfactory separation between the different situations we want to discriminate. The physical values of selected features remain unchanged. The redundancy of features might be identified by a feature clustering and selection algorithm or we might remove features with the highest correlation. The algorithm removes similar features. This implies a faster training of consequent classifiers on reduced feature space.

The feature extraction method works in opposite. Hereby, the features are projected onto a set of reduced feature space by some transformation function. The features in transformed space are no longer representing the same physical meaning as in original space. The transformation function is an analytical function and the challenge is to find representative and informative transformation for the given feature set. Very well known techniques are: the principal components analysis (PCA) and dimensionality reduction by *auto-associative* mapping using MLP neural.

Four methods for features space reduction were analyzed in this work. All these methods have been used with four publicly available databases and applied to very well known *k*-nearest neighbor (*k*-NN) classifier method and the best methods for features space reduction were chosen. In all the tests performed, the feature clustering method was the best, i.e. the least average classification errors was made by feature clustering method. Features space reduction method based on MLP neural achieved the worst (bad) average classification errors.

DARBO TIKSLAS IR UŽDAVINIAI

Magistro darbo tikslas yra tarpusavyje palyginti klasifikavimui skirtų požymių mažinimo metodus, kurie turimą požymių aibę transformuoja į mažesnės eilės aibę. Duomenų klasifikavimo kokybė transformuotoje požymių erdvėje turi nenukentėti. Eksperimentams naudotos keturios realių duomenų bazės.

Kiekvienai duomenų bazei tikrinama hipotezė apie vidutinių reikšmių lygybę, t.y. lyginamos dvi skirtingos vidutinės klasifikavimo klaidos ir nuspręsta ar jos yra panašios, ar skirtingos, naudojant Stjudento (t) testą. Tam, kad tai patikrinti bus skaičiuojama T statistika.

Pirmą kartą duomenų požymių atrinkimui panaudotas neraiškaus integralo metodas su pilnuoju matu.

Visi gauti eksperimentų rezultatai pateikti paveiksluose ir apibendrinti lentelėse.

Magistrinio darbo išvadose pateiktas trumpas gautų rezultatų aprašymas.

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

ĮVADAS

Daugelyje mokslo sričių susiduriama su duomenų dimensiškumo problema, t.y. kiekvienas objektas apibūdinamas dideliu požymių kiekiu, kurių skaičius gali siekti šimtus ar tūkstančius. Naudojant požymių erdvės mažinimo metodus siekiama suprojektuoti turimų duomenų požymių erdvę į erdvę turinčią mažesnę dimensiją. Taip siekiama atrinkti esminius požymius, sumažinant skaičiavimų apimtis, tuo pačiu išlaikant duomenų apdorojimo tikslumą.

Daugiamatės duomenų bazės taip pat yra problemiškos dėl jų saugojimo bei duomenų paieškos ir išrinkimo juose. Todėl pagal poreikius yra tikslinga identifikuoti svarbiausius duomenų požymius, tam, kad tolimesnis duomenų apdorojimas būtų supaprastintas, nesumažinant tolimesnių tyrimų rezultatų kokybės.

Tačiau taikant požymių erdvės mažinimo metodus egzistuoja tam tikra klaida, kuri atvaizduoja pradinių duomenų informacijos praradimą. Taigi, požymių erdvės mažinimas naudingas tik tada, kai informacijos praradimas nėra lemiamas uždavinio (problemos) sprendiniui.

Šiame darbe bus išnagrinėti keturi požymių erdvės mažinimo metodai, t.y. klasterizavimo metodas, požymių erdvės suspaudimas naudojant neuroninį tinklą, principinių komponenčių analizės (PCA) metodas, bei neraiškus integralas su pilnuoju matu. Visi šie metodai lyginami su *k*-artimiausių kaimynų (*k*-NN) klasifikatoriumi atliekant eksperimentus su keturiomis laisvai prieinamomis duomenų bazėmis. Eksperimentams atlikti naudojamas programinis paketas MATLAB.

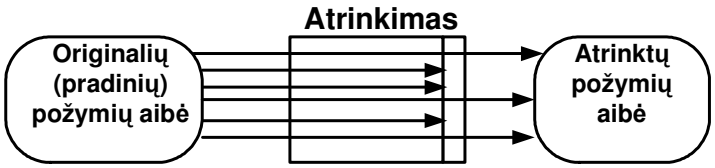
Šio darbo struktūra: pirmame skyriuje aprašomi požymių erdvės mažinimo metodai, pateikiama informacija apie neraiškiąsias aibes, neraiškų integralą. Antrame skyriuje aprašomos naudotos keturios duomenų bazės, jų struktūra bei trumpa analizė, pateikiama informacija apie kitų mokslininkų gautus rezultatus su šiomis keturiomis duomenų bazėmis, aprašoma eksperimento eiga, pateikiami eksperimento rezultatai. Gauti rezultatai parodyti grafikuose bei apibendrinti lentelėse.

Magistrinio darbo gale yra pateiktos išvados, kurios apibendrina atliktų eksperimentų rezultatus.

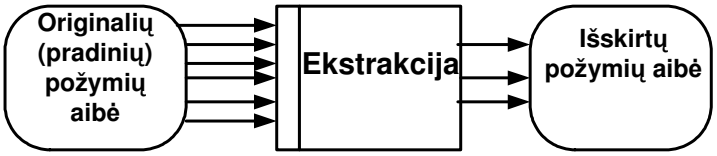
1. TEORINĖ DALIS

1.1. POŽYMIŲ SUMAŽINIMAS

Procesas, kai iš didelės požymių grupės yra surandami tik esminiai požymiai yra vadinamas požymių sumažinimu (redukcija). Požymių sumažinimo metodas gali būti skirstoma į požymių atrinkimo ir požymių ekstrakcijos metodus. Pagrindinė idėja yra pavaizduota 1.1 pav.



(a)



(b)

1.1 pav. Požymių atrinkimo (a) ir požymių išskyrimo (b) schema

Požymių ekstrakcijos iš gausybės požymių aibės pagrindinis tikslas yra surasti tokią transformaciją kuri sumažintų požymių skaičių, t.y. kuris iš gausybės požymių atrinktų tik esminius, kurie suteiktų pakankamą informaciją uždaviniui spręsti. Atvaizdavimo suradimui, pirmiausia *n*-matės erdvės duomenys projektuojami į *d*-matės erdvės duomenis, kur $d<n$. Požymių atrinkimas yra atskiras požymių ekstrakcijos atvejis. Taikant požymių ekstrakciją, visi *n* požymiai yra panaudojami *m*-matiems požymiams gauti. Todėl visi *n* požymiai turi išlikti. Tačiau požymių atrinkimas, priešingai, leidžia mums atmesti $(n - m)$ nereikšmingus požymius. Vadinasi, kaupiami tik reikšmingi požymiai.

Požymių atrinkimo metodas atrenka nepriklausomus požymius, kurie suteikia pakankamą informaciją norimam uždaviniui spręsti. Fizinė atrinktų požymių prasmė išlieka nepakitusi. Pertekliniai požymiai pašalinami naudojant požymių klasterizavimo ir atrinkimo algoritmą arba pašalinami požymiai su didžiausia koreliacija. Sumažinus perteklinę informacija, pagreitėja informacijos apdorojimas ir analizė.

Požymių ekstrakcijos metodas veikia priešingai. Šiuo atveju požymiai projektuojami į mažesnės eilės požymių aibę panaudojant įvairias transformacijos funkcijas. Po projektavimo fizinės reikšmes

pakinta ir fizikinė prasmė nebėra tokia pati kaip pradinėje aibėje. Transformacijos funkcija yra analizinė funkcija, kurios pagrindinis reikalavimas yra sukurti informatyvią transformuotą požymių aibę. Labai gerai žinomi du tradiciniai požymių išskyrimo metodai – tai principinių komponenčių analizė (PCA) [19, 20] ir požymių skaičiaus sumažinimas su *auto-asociatyviniu* neuroniniu tinklu [20, 21, 22, 23, 24, 25, 26, 27].

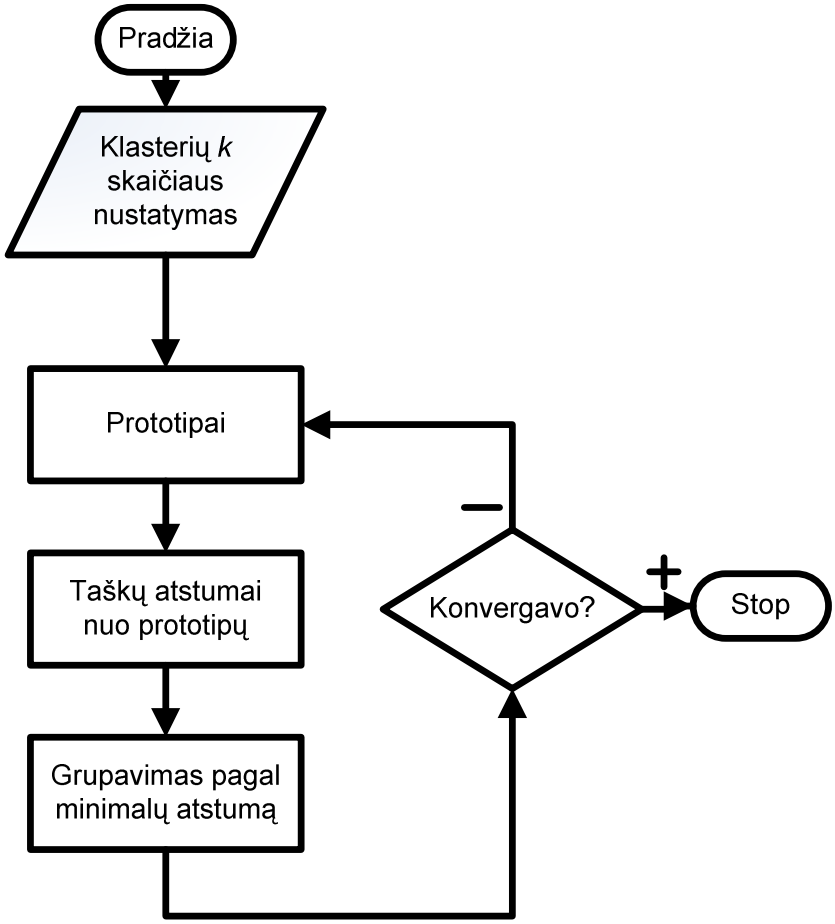
1.1.1. POŽYMIŲ ATRINKIMAS PANAUDOJANT KLASTERIZAVIMO METODĄ

Požymių atrinkimas ir mažinimas vykdomas *k*-means (iš angl. *„vidurkis“*) klasterizavimo būdu. *k*-means algoritmas yra skirtas požymių erdvėje klasifikavimui ar sugrupavimui į sveiko teigiamo skaičiaus *k* grupių (klasterių). Klasterizavimas vykdomas minimizuojant atstumų tarp duomenų taškų ir atitinkamo klasterio centro kvadratų sumas.

k-means algoritmas

k-means klasterizavimo metodas vykdomas tokia tvarka:

1. Reikšmės *k* nustatymas, kur *k* – pageidaujamų klasterių skaičius.
2. Iš duomenų erdvės parenkami *k* taškai, kurie priskiriami vieno elemento klasterių prototipais (centroidais):
   2.1. Kiekvienas iš likusių taškų $(N-k)$ priskiriamas klasteriui su jam artimiausiu prototipu;
   2.2. Priskyrus visus taškus perskaičiuojamos visų prototipų pozicijos.
3. Eilės tvarka apskaičiuojamas kiekvieno taško atstumas nuo atitinkamo prototipo. Jei taškas nepriklauso klasteriui su artimiausiu jam prototipu, jis priskiriamas prie kito klasterio (su artimiausiu taškui prototipu). Po kiekvieno naujo priskyrimo, iš naujo perskaičiuojamos naujos klasterių, į kuriuos priskirtas ir iš kurių atimtas naujas taškas, prototipų pozicijos.
4. 3 žingsnis kartojamas tol, kol prototipai daugiau nebeperskaičiuojami, t.y. kol pasiekiama konvergencija.



1.2 pav. *k*-means algoritmas

1.1.2. POŽYMIŲ KORELIACIJA

Koreliacija (arba **koreliacijos koeficientas**) tikimybių teorijoje ir statistikoje yra statistinio ryšio tarp kintamųjų (požymių) stiprumo matas. Jeigu dviejų požymių koreliacija lygi nuliui, tai tie požymiai yra statistiškai nepriklausomi.

Matematinės savybės:

Dviejų atsitiktinių dydžių *X* ir *Y*, kurių vidurkiai yra μ_X ir μ_Y, o standartiniai nuokrypiai - σ_X ir σ_Y, koreliacijos koeficientas ρ_{XY} apibrėžiamas taip:

$$\rho_{XY} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y},$$

čia $\operatorname{cov}(X,Y)$ yra dydžių *X* ir *Y* kovariacija.

Kadangi $\mu_X = E(X)$, $\sigma_X^2 = E(X^2) - E^2(X)$ (ir atitinkamai tą patį galima pasakyti apie *Y*), tai koreliacijos koeficiento formulę galima užrašyti ir taip:

$$\rho_{XY} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)} \cdot \sqrt{E(Y^2) - E^2(Y)}}.$$

Kad koreliacijos koeficientas turėtų apibrėžtą reikšmę, standartiniai nuokrypiai σ_X ir σ_Y turi būti baigtiniai ir nelygūs nuliui.

Koreliacijos koeficientas visada yra skaičius iš intervalo [-1; 1].

Jei tarp *X* ir *Y* egzistuojanti priklausomybė yra tiesinė, tai ρ_{XY} lygus 1 arba -1. Jis lygus 1, kai egzistuoja tokios konstantos $a > 0$ ir *b*, kad $Y = aX + b$. Jis lygus -1, kai egzistuoja tokios konstantos $a < 0$ ir *b*, kad $Y = aX + b$.

Koreliacija ir priežastingumas:

Iš to, kad dviejų kintamųjų koreliacijos koeficientas nelygus nuliui, galima daryti tik tokią išvadą, jog egzistuoja statistinis ryšys, o ne koks nors priežastingumas (t.y. *X* nebūtinai veikia *Y*, nors *X* ir *Y* yra statistiškai susiję). Koreliacija, kuri tiesiogiai neatspindi priežastingumo, statistikoje vadinama „klaidingąja koreliacija“ (angl. *„spurious correlation“*).

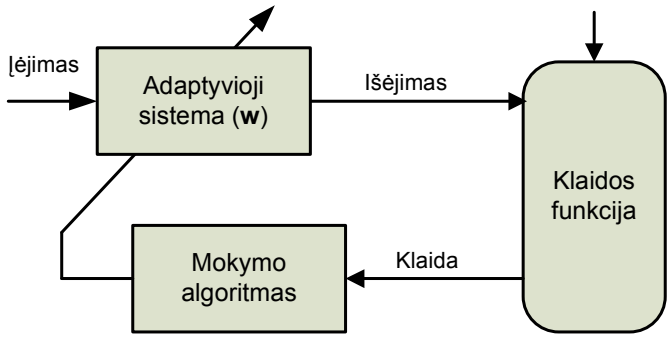
Viena geriausių tokio teiginio iliustracijų yra pavyzdys su ledų suvartojimu ir nuskendusiųjų skaičiumi: pastebėta, jog padidėjus ledų suvartojimui, padidėja ir skenduolių skaičius, tad lyg ir norėtųsi daryti išvadą, jog ledų valgymas yra labai kenksmingas plaukikams. Šitame pavyzdyje neatsižvelgiama į svarbiausią užslėptą kintamąjį - oro temperatūrą. Vasarą būna karšta, todėl padidėja tiek ledų suvartojimas, tiek skenduolių skaičius, nes daugiau žmonių maudosi. Gali būti, jog koreliacinis ryšys yra nustatomas, o iš tikrųjų priežastingumas buvo visiškai priešingas: štai praeitame amžiuje buvo manoma, jog žmonėms yra naudinga turėti blusų, nes statistiškai buvo pastebėta koreliacija tarp žmonių sveikatos ir blusų turėjimo – blusas turėjo sveikesni žmonės. Iš tikrųjų, ryšys yra visiškai priešingas: blusos dažnai sukelia ligas, o žmogui susirgus karštine, jog nebegali gyventi žmogaus plaukuose, nes ten per karšta, todėl susirgęs žmogus blusų nebeturi.

Šiais laikais prastas koreliacijos ir priežastingumo suvokimas irgi yra dažnas, o ypač „tyrimuose“, kurie yra daromi spaudoje, norint sukelti sensaciją ir pan. JAV buvo atliktas tyrimas, kuris nustatė, jog rūkymas kenkia studentų mokslams, nes rūkantys studentai gauna mažesnius pažymius. Nepagalvota, jog pagrindinė tokios koreliacijos priežastis gali būti tokia, kad rūkantys studentai yra tokie studentai, kurie ir šiaip mokslams skiria mažiau dėmesio, o gal kaip tik, jie rūko tik todėl, jog nesiseka moksluose?

1.1.3. POŽYMIŲ ERDVĖS MAŽINIMAS NAUDOJANT NEURONINIUS TINKLUS

Dirbtiniai neuroniniai tinklai – tai informacijos apdorojimo struktūros, netiksliai imituojančios kai kuriuos gyvųjų organizmų smegenyse vykstančius informacijos apdorojimo procesus. Dirbtiniai neuroniniai tinklai sudaromi iš daugelio tarpusavyje sujungtų labai paprastų skaičiavimo elementų. Šie elementai, jungiami vieni su kitais įvairaus stiprumo jungtimis, yra apytikris biologinių neuronų modelis. Dirbtiniu neuroniniu tinklu siekiama imituoti kai kurias biologinių sistemų savybes. Labiausiai viliojantis atrodo biologinių sistemų gebėjimas mokytis, prisitaikyti ir adaptuotis.

Neuroninės sistemos sudaromos ne pagal specifikaciją, formules ar aprašymą naudojant išankstines žinias. Vietoj to sistema naudoja išorinius duomenis savo parametrams nustatyti. Neuroniniai tinklai apmokomi žinant įėjimo ir atitinkamas išėjimo vertes, kurios dar vadinamos užduoties vertėmis. Mokymo metu minimizuojama tam tikros klaidos funkcija (žr. 1.3 pav.).



1.3 pav. Adaptyviosios sistemos kūrimo proceso schema

Klaidos funkcija labai dažnai yra neuroninio tinklo išėjimo ir užduoties vertės tarpusavio skirtumo funkcija. Mokymo metu svoriai keičiami taip, kad sistemos išėjimo vertės artėtų prie norimų verčių (mažėtų klaida). Neuroniniai tinklai mokomi (įvertinami parametrai-svoriai) naudojant pasirinktą duomenų rinkinį, vadinamą duomenų imtimi. Naudojant jau apmokytą neuroninį tinklą, parametrai paprastai būna fiksuoti.

1.1.3.1 DIRBTINIO NEURONO MODELIS

Dirbtinio neuroninio tinklo neuroną galima apibūdinti taip:

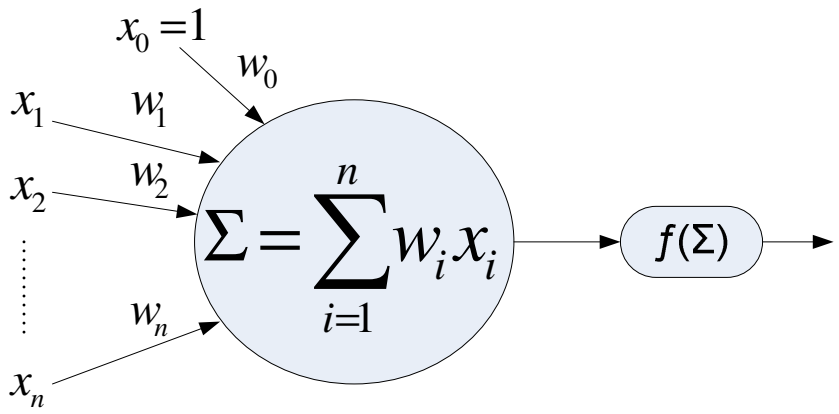
1. Neurono įėjime veikia keletas įėjimo signalų (verčių). Šie signalai gali būti viso neuroninio tinklo įėjimo signalai ar kitų tinklo neuronų išėjimo signalai. Kiekviena įėjimo jungtis turi savo perdavimo koeficientą (svorį). Kiekvienas neuronas turi jam

priskirtą slenksčio vertę. Neurono sužadinimo signalas formuojamas skaičiuojant svorinę įėjimo signalų sumą ir atimant slenksčio vertę.

2. Turint sužadinimo signalą ir naudojant neurono perdavimo funkciją, skaičiuojamas neurono išėjimo signalas (vertė).

Naudojant šuolinę neurono perdavimo funkciją neurono išėjimo signalas lygus nuliui, jei sužadinimo signalo amplitudė mažesnė už nulį ir išėjimo signalas lygus vienetui, jei sužadinimo signalo amplitudė didesnė ar lygi nuliui. Neurono svoriai gali būti ir neigiami. Neigiamas svoris reiškia, jog jungtis turi slopinamąjį, bet ne žadinamąjį efektą.

Labiausiai paplitusi dirbtinio neurono schema pavaizduota 1.4 paveiksle. Čia $x_1,...,x_n$ žymi neurono įėjimo signalus. Atitinkami svoriai pažymėti $w_1,...,w_n$.



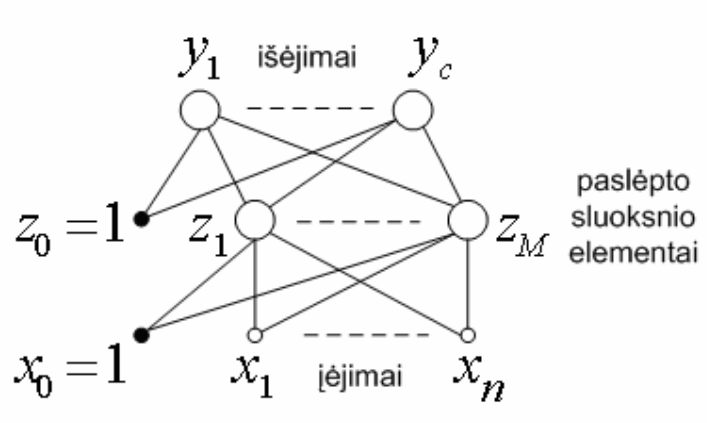
1.4 pav. Dirbtinio neuroninio tinklo neuronas

Žymėjimas w_0 reiškia slenksčio vertę, $f()$ – perdavimo funkciją ir y – neurono išėjimą.

Viensluoksniai dirbtinių neuronų tinklai netinka sudėtingiems uždaviniams spręsti Tam naudojami tinklai, turintys keletą svorių sluoksnių. Dažnai sunku nustatyti reikalingą sluoksnių skaičių ir elementų skaičių juose. Didinant sluoksnių skaičių ir neuronų skaičių juose, tinklas darosi sudėtingesnis. Reikalingas neuronų skaičius priklauso ne nuo įėjimo erdvės matavimų skaičiaus, o nuo uždavinio sudėtingumo.

Bendruoju atveju neuroninis tinklas, gali būti kokio tik norima dydžio. Skaičiavimo elemento išėjimas – tiesinė įėjimo kintamųjų kombinacija, kurią transformuoja perdavimo funkcija. Dažniausia daugiasluoksnio tinklo struktūra – vienas po kito einantys neuronų sluoksniai, kurių kiekvienas neuronas sujungtas su visais kitais kito sluoksnio neuronais, ir neturintys jokių kitų jungčių (1.5 pav.). Tokius tinklus patogiau analizuoti teoriškai ir juos lengviau modeliuoti. Įprastinė daugiasluoksnio

tiesioginio sklidimo neuroninio tinklo santrumpa – *MLP* (angl. *„MultiLayer Perceptron“*), kitaip – daugiasluoksnis perceptronas [34].

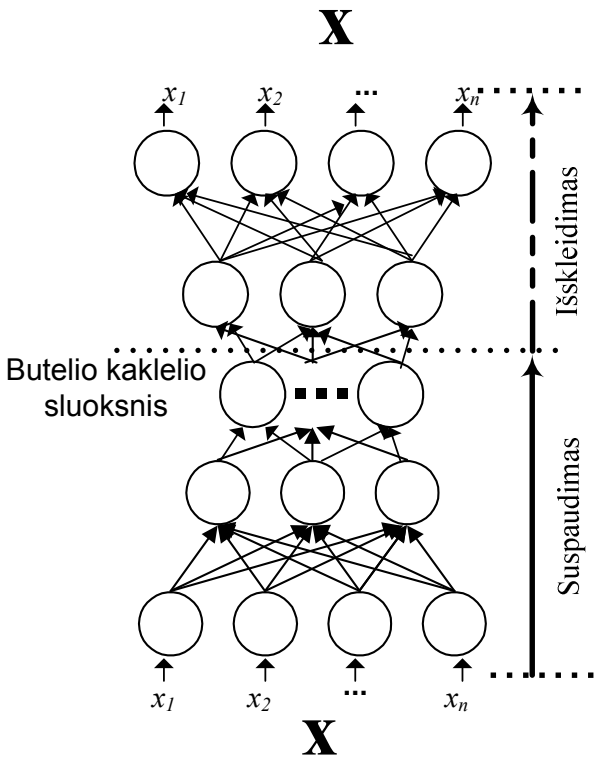


1.5 pav. Tiesioginio sklidimo daugiasluoksnis neuroninis tinklas

1.1.3.2. DUOMENŲ SUSPAUDIMAS DAUGIASLUOKSNIU NEURONINIU TINKLU. BUTELIO KAKLELIO METODAS

Daugiasluoksnio perceptrono neuroninis tinklas, turintis „butelio kaklelio“ struktūrą, naudojamas požymių erdvės suspaudimui. Daugiasluoksnis neuroninis tinklas, parodytas 1.6 pav. turi *n* įėjimų, *n* išėjimo neuronų bei tarpinėje tinklo dalyje yra suspaudžiami į *d* paslėptų neuronų, kur $d<n$. Tokios struktūros tinklo įėjimo vektorius kartu yra ir tikslinis mokymo vektorius, t.y. neuroninis tinklas mokomas atvaizduoti duomenis į tas pačias vertes informaciją suspaudžiant „butelio kaklelyje“. Siekiant padidinti duomenų suspaudimo laipsnį, neuronų skaičius paslėptame „butelio kaklelyje“ yra mažesnis nei įėjimo sluoksnio, t.y. $d<n$.

Toks tinklas gali būti apmokomas standartiniu klaidos skleidimo atgal metodu, minimizuojant vidutinę kvadratinę klaidą. Apmokius tinklą, pirmoji tinklo dalis atlieka įėjimų suspaudimą, o antroji – išskleidimą. Požymių suspaudimui naudojama tiktai pirmoji apmokyto neuroninio tinklo dalis. Suspaudimo lygis nustatomas parenkant atitinkamą neuronų skaičių paslėptajame sluoksnyje [20].



1.6 pav. Daugiasluoksnis perceptrono tinklas su „butelio kaklelio“ sluoksniu

1.1.4. PRINCIPINIŲ KOMPONENČIŲ ANALIZĖS METODAS

Principinių komponenčių analizės (PCA) metodas yra klasikinis statistinis metodas, kuris plačiai taikomas statistikoje ir duomenų suspaudime, taip pat informacijos apdorojimo uždaviniuose, požymių išskyrimui [5]. Šis tiesinės transformacijos metodas pagrįstas atsitiktinių kintamųjų statistiniu vaizdavimu. Šio metodo tikslas: vektorių $\mathbf{x} = (x_1,\ldots,x_n)$ atvaizduoti į d-matę erdvę $(z_1,\ldots,z_d)$ su $d<n$, t.y. atrinkti visoje duomenų aibėje maksimalią informaciją turinčius duomenis.

Pirmiausia pažymėkime, kad vektorius **x** gali būti pavaizduotas kaip d ortonormalių vektorių $\mathbf{u}_i$ aibės tiesinis darinys:

$$\mathbf{x} = \sum_{i=1}^{d} z_i \mathbf{u}_i \,, \tag{1.1}$$

kur vektorius $\mathbf{u}_i$ atitinka ortonormalumo sąryšį:

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \,, \tag{1.2}$$

čia δ_{ij} yra Kronekerio delta simbolis. Norint rasti koeficientą z_i, esantį formulėje (1.1), galime jį gauti pasinaudodami formule (1.2):

$$z_i = \mathbf{u}_i^T \mathbf{x}, \tag{1.3}$$

(1.3) formulę galima laikyti paprastu koordinačių sistemos pasukimu iš originalaus **x** į naują *z* koordinačių sistemą. Galime daryti prielaidą, kad išsaugojome tik tuos pogrupius, kur pagrindinio vektoriaus $\mathbf{u}_i$ *d*-matė erdvė yra mažesnė už *n*-matę erdvę, t. y. $d<n$. Todėl tolimesniems skaičiavimams naudojame tik *d*-matės erdvės koeficientus z^n. Likę koeficientai pakeičiami į konstantas b_i, todėl pirminis duomenų vektorius **x** aproksimuojamas tokia forma:

$$\bar{\mathbf{x}} = \sum_{i=1}^{d} z_i \mathbf{u}_i + \sum_{i=d+1}^{n} b_i \mathbf{u}_i \,. \tag{1.4}$$

Ši forma atitinka duomenų erdvės suspaudimo išraišką, kurioje pirminis duomenų vektorius **x,** turintis *n* laisvės laipsnių, aproksimuojamas naujuoju vektoriumi **z,** turinčiu *d* laisvės laipsnių ($d<n$). Dabar laikoma, kad visi aibės *N* duomenys yra vektoriai $\mathbf{x}_j$, kur $j = 1,\ldots,N$. Tikimasi išsirinkti pagrindinį vektorių $\mathbf{u}_i$ ir koeficientus b_i tokius, kad aproksimacija (žr. 1.4 formulėje) su z_i reikšmėmis, apskaičiuotomis (1.3) formulėje, originaliems vektoriams **x** iš visos duomenų aibės duos geriausią aproksimaciją. Vektoriaus $\mathbf{x}_j$ paklaida apskaičiuojama taip:

$$\mathbf{x}_j - \bar{\mathbf{x}}_j = \sum_{i=d+1}^{n} (z_i - b_i)\mathbf{u}_i \,. \tag{1.5}$$

Dabar galima apibrėžti geriausią aproksimaciją, kuri minimizuoja paklaidų kvadratų sumą iš visos duomenų aibės:

$$E_d = \frac{1}{2}\sum_{j=1}^{N} \left\| \mathbf{x}_j - \bar{\mathbf{x}} \right\|^2 = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=d+1}^{n} (z_{j,i} - b_i)^2 \,, \tag{1.6}$$

(1.6) formulėje panaudojamas ortonormuotas ryšys (žr. 1.2 formulę). Jeigu apskaičiuotume dydžio E_d išvestinę atžvilgiu b_i, tai gautume:

$$b_i = \frac{1}{N}\sum_{n=1}^{N} z_i^n = \mathbf{u}_i^T \bar{\mathbf{x}}, \tag{1.7}$$

kur $\bar{\mathbf{x}}$ - vektoriaus *x* vidurkis, kuris randamas iš formulės:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^n \ . \tag{1.8}$$

Pasinaudodami (1.3) ir (1.7) formulėmis, paklaidų kvadratų sumą galima parašyta taip:

$$E_d = \frac{1}{2}\sum_{i=d+1}^{n}\sum_{j=1}^{N}\left\{\mathbf{u}_i^T(\mathbf{x}_j - \bar{\mathbf{x}})\right\}^2 = \frac{1}{2}\sum_{i=d+1}^{n}\mathbf{u}_i^T\sum\mathbf{u}_i \ , \tag{1.9}$$

kur Σ – vektorių $\{\mathbf{x}^n\}$ aibės kovariacinė matrica, užrašoma tokia formule:

$$\sum = \sum_j(\mathbf{x}_j - \bar{\mathbf{x}})(\mathbf{x}_j - \bar{\mathbf{x}})^T \ . \tag{1.10}$$

Liko užduotis įrodyti E_d minimizaciją, kai pasirenkami pagrindiniai vektoriai $\mathbf{u}_i$. Minimizacija įvyksta, kai pagrindiniai vektoriai atitinka šia formulę:

$$\sum\mathbf{u}_i = \lambda_i\mathbf{u}_i \ , \tag{1.11}$$

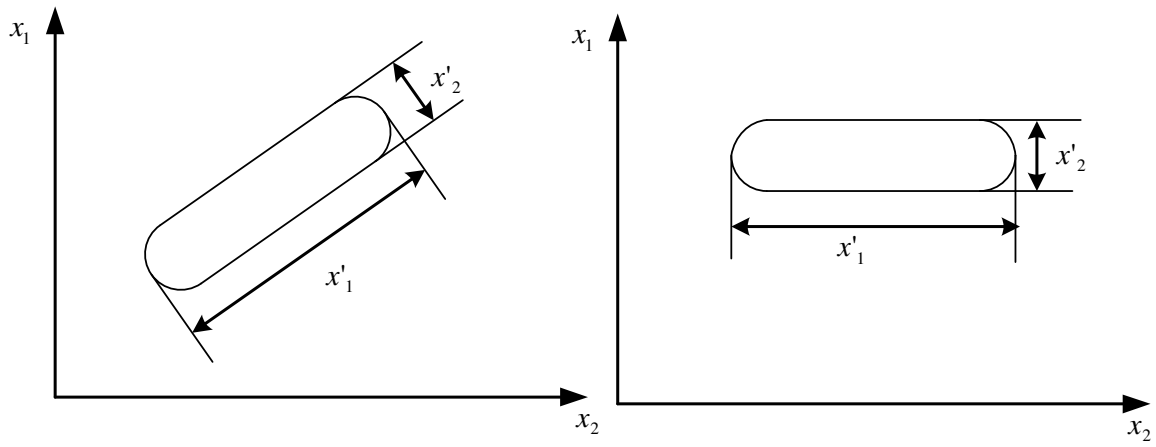
todėl šie pagrindiniai vektoriai yra kovariacinės matricos nuosavi vektoriai. Pastebime, kad jei kovariacinė matrica yra realioji ir simetrinė, jos nuosavi vektoriai gali būti pasirenkami taip, kad jie būtų ortonormalūs. Pakeičiant (1.11) formulę į (1.9) formulę ir pasinaudodami ortonormalumo sąryšiu (žr. 1.2 formulę), gauname klaidų kriterijaus reikšmę minimumo taške pagal tokią formulę:

$$E_d = \frac{1}{2}\sum_{i=d+1}^{n}\lambda_i \ . \tag{1.12}$$

Vadinasi, gaunama minimali klaida priklauso nuo pasirinkto d . Kuo mažesnis d, tuo didesnė klaida.

Taigi, atliekant PCA, pirma principinė komponentė yra koordinatė maksimalios dispersijos kryptimi, o antroji – ortogonali pirmajai vėlgi didžiausios dispersijos kryptimi (išlaikant ortogonalumo sąlygą). Komponenčių kryptys nusakomos kovariacinės matricos nuosavais vektoriais, svarba rikiuojama pagal nuosavas reikšmės.

Pavyzdys parodytas 1.7 pav., kur esant dviems matavimams x_1 ir x_2, ir prireikus rinktis tik vieną požymį, duomenys projektuojami kryptimi, turinčia didžiausią dispersiją (kryptimi x'_1).



1.7 pav. Duomenys prieš ir po pasukimo

1.2. POŽYMIŲ ATRINKIMAS NAUDOJANT NERAIŠKŲ INTEGRALĄ SU NERAIŠKIU MATU

1.2.1. NERAIŠKIOS AIBĖS IR NERAIŠKI LOGIKA

Kasdieniniame gyvenime mes dažnai susiduriame su ne griežtai apibrėžtais reiškiniais, kuriuose sunku įžvelgti ribą tarp tiesos ir netiesos, t.y. nėra taip, kad reiškinys arba vyksta arba ne, jis gali vykti ir iš dalies. Tokius netikslius, neaiškius reiškinius aprašyti klasikine aibių teorija pagrįsta matematika yra sudėtinga, o be tikslaus matematinio modelio valdyti neaiškius procesus yra sunku arba visai neįmanoma. Tokiu atveju išeitis gali buti neraiški (angl. *„fuzzy”*) logika, kuri leidžia matematiškai apibrėžti dalinį teisingumą, netikslumą, panaudojant lingvistinius kintamuosius. Pavyzdžiui, terminai „aukštas” arba „žemas” gali buti lingvistiniai terminai, apibūdinantys fizikinį dydį – slėgį. Pasitelkiant neraiškių aibių teoriją, lingvistinę kasdieninės kalbos informaciją galima perteikti kompiuteriui, o tai leidžia lengviau ir aiškiau formuluoti įvairias valdymo ar kitas užduotis.

Pagrindinės neraiškios logikos idėjos siekia Platoną, kuris pripažino ne tik TIESĄ ir MELĄ, o taip pat sritį tarp jų. Vėliau atsirado „Poli-logika“, kurios pradininkas buvo Lukasiewicz (1878-1956). 1960 metais Lotfi A. Zadeh Kalifornijos Universitete apgynė daktaro disertaciją apie neraiškias aibes ir jų panaudojimą priimant sprendimus remiantis nepilna informacija. Vėliau iki 1980~90 metų neraiškių aibių logika buvo vangiai studijuojama. Šiuo metu Japonija ir Korėja yra šios srities lyderiai tiek taikyme, tiek ir teoriniuose tyrinėjimuose. Neraiškios aibės intensyviai tyrinėjamos bei diegiamos kokybės užtikrinime, gedimų diagnostikoje, valdymo technologijoje, duomenų atpažinime,

ekspertinėse sistemose, t.y. medicinoje, ligų diagnozavimo sistemose, vertybinių popierių rinkos analizėje, orų prognozėse, ekonomikoje, politikoje.

1.2.1.1. NERAIŠKIŲ AIBIŲ APIBRĖŽIMAS

Tradicinėje aibių teorijoje elementas priklauso konkrečiai aibei arba ne. Jis negali tik iš dalies priklausyti tai aibei. Tuo tarpu neraiški logika leidžia elementui ir iš dalies priklausyti aibei, suteikiant jam atitinkamą priklausomumo koeficientą.

Neraiškiose aibėse kiekvienas elementas iš intervalo [0,1] yra aprašomas tokia funkcija:

$$\mu_A : X \rightarrow [0,1], \tag{1.13}$$

kur intervalas [0,1] susideda iš realių skaičių nuo 0 ir 1, įtraukiant 0 ir 1 [1].

Kiekviena neraiški aibė turi atskirą jos paviršių aprašančią funkciją, t.y. elementų priklausomumo aibei dydžius nustatančią funkciją $\mu(x)$, kuri parenkama priklausomai nuo norimo tikslumo. Neraiškios aibės paviršiai gali būti trikampio, trapecijos, S formos, varpo ir kitokių formų [2]. Dėl skaičiavimo paprastumo populiariausia yra trikampio formos priklausomumo funkcija. Ji yra aprašoma trimis taškais, kairys pagrindo taškas, centras ir dešinys pagrindo taškas.

Tegul x - lingvistinis kintamasis, kurio galimų reikšmių aibė yra U, o $x \in U$. Parenkam tris taškus a, b ir c iš reikšmių aibės U su sąlyga $a < b < c$. Taškai a ir c yra simetriškai nutolę taškui b ir yra lingvistinio kintamojo vienos iš neraiškių aibių reikšmių intervalo ribinės reikšmės. Tuomet elemento $x \in U$ priklausomumo neraiškiai aibei reikšmė bus apskaičiuojama pagal formulę [3]:

$$\mu(x) = \begin{cases} 0, \text{ jei } x < a, \\ (x-a)/(b-a), \text{ jei } a < x < b, \\ (c-x)/(c-b), \text{ jei } b < x < c, \\ 0, \text{ jei } x > c \end{cases} \tag{1.14}$$

1.2.1.2. PAGRINDINĖS NERAIŠKIŲ AIBIŲ OPERACIJOS

Panašiai kaip tradicinėje aibių teorijoje, neraiškioms aibėms yra taikomos tokios pat operacijos, kaip sąjunga, sankirta ir neigimas. Visi veiksmai atliekami ne su pagrindiniais aibės porų elementais, o su jų priklausomumo analizuojamai aibei reikšmėmis.

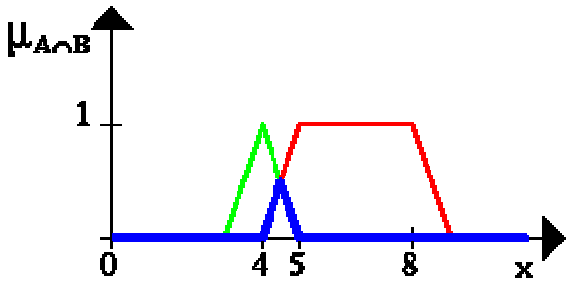
Pasirinkime dvi neraiškias aibes A ir B, kurių galimų reikšmių intervalas yra X, o $x \in X$.

- Aibių A ir B sankirta yra nauja aibė, priklausanti tam pačiam reikšmių intervalui X, kurios elementų priklausomumo aibei reikšmės nustatomos skaičiuojant minimumą [5]:

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \forall x \in X \tag{1.15}$$

arba sandaugą:

$$\mu_{A\cap B}(x) = \{\mu_A(x) \cdot \mu_B(x)\} \quad \textbf{(1.16)}$$



1.8 pav. Grafiškai pavaizduota sankirtos operacija

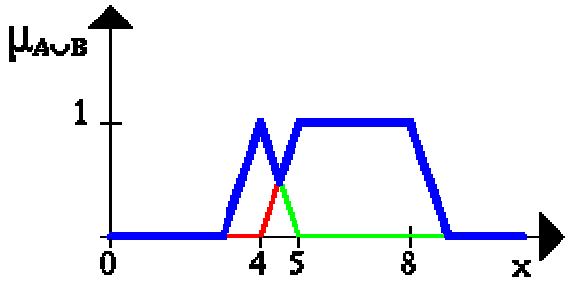
Šiame paveikslėlyje žalioji trikampio formos ir raudonoji trapecijos formos neraiškios aibės. Norint surast šių dviejų neraiškių aibių sankirtą, reikia paimti minimalias atitinkamas vertes (mėlynoji kreivė)

- Aibių *A* ir *B* sąjunga yra nauja aibė iš to pačio galimų reikšmių intervalo *X*, kurios elementų priklausomumo reikšmės randamos skaičiuojant maksimumą [5]:

$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \forall x \in X \quad \textbf{(1.17)}$$

arba algebrinę sumą:

$$\mu_{A\cup B}(x) = \max\{\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)\}, \forall x \in X \quad \textbf{(1.18)}$$

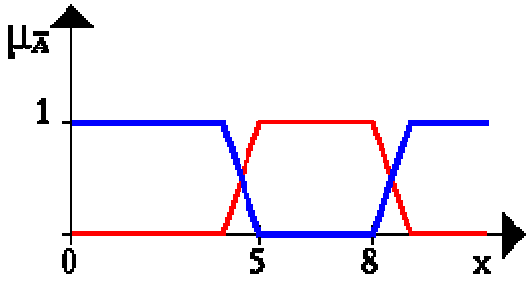


1.9 pav. Grafiškai pavaizduota sąjungos operacija

1.9 paveiksle grafiškai pavaizduotos žalioji trikampio formos ir raudonoji trapecijos formos neraiškios aibės. Ieškant sankirtos tarp šių dviejų aibių, reikia surast maksimalias atitinkamas vertes (mėlynoji kreivė).

- Neraiškios aibės *A* papildymas yra nauja aibė iš galimų reikšmių intervalo *X*, kurios elementų priklausomumas nustatomas taip [5]:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad \textbf{(1.19)}$$



1.10 pav. Grafiškai pavaizduota papildymo operacija

Papildymo operacija pavaizduota trapecinė funkcija. Raudonoji linija yra neraiški aibė (žr. 1.10 pav.). Neraiškios aibės papildymui, reikia atimti atitinkamą neraiškios aibės vertę iš vieneto.

1.2.2. NERAIŠKŪS MATAI IR NERAIŠKUS INTEGRALAS

Prieš apibrėžiant kas yra neraiškus matas ir neraiškus integralas, pirmiausia pateiksime trumpą istoriją apie šias dvi sąvokas. 1974 metais M. Sugeno įvedė neraiškaus mato ir neraiškaus integralo sąvoką, praplėsdamas (apibendrindamas) įprasto mato apibrėžimą, pakeičiant įprastą adityvumo savybę. Istoriškai, tai buvo artimai susiję erdvės (tūrio) sąvoka, kurią įvedė G. Choquet prieš 20 metų.

Nuo to laiko vis daugiau mokslininkų susidomėjo neraiškiais matais ir neraiškiais integralais ir juos nagrinėjo daugiau iš matematinės pusės. Buvo sukurta įvairių programų subjektyviam daugiakriteriniam vertinimui ir čia neraiškus integralas buvo panaudotas kaip nauja grupavimo priemonė.

Dabar galime apibrėžti neraiškaus mato ir neraiškaus integralo sąvokas.

Tegul $Z = \{z_1, z_{2,\ldots,} z_L\}$ yra baigtinė, netuščia aibė, kur kiekvienas požymis z_i, $i = 1,2,\ldots,L$ reiškia informacijos šaltinį. Ir tegul 2^Z žymi visų aibės Z poaibių aibę.

1 Apibrėžimas. Aibės funkcija $g: 2^Z \to [0,1]$ vadinama neraiškiu matu, jeigu tenkina šias sąlygas:

1) $g(\varnothing) = 0;\ \ g(Z) = 1$,

2) Jeigu $A, B \subset 2^Z$ ir $A \subset B$, tada $g(A) \le g(B)$,

3) Jeigu $A_n \subset 2^Z$ visiems $1 \le n < \infty$ ir seka $\{A_n\}$ yra monotoniška apjungimo prasme, tada $\lim_{n\to\infty} g(A_n) = g(\lim_{n\to\infty} A_n)$.

Neraiškūs matai yra reikšmingesni lyginant su klasikiniais matais. Tegul turime neraiškią išmatuojamą aibę (Z, 2^Z, g) ir aibės A ir B tenkina tokią sąlygas: $A, B \subset 2^Z$, $A \cap B = 0$, $A \cup B \subset 2^Z$. Tada neraiškus matas g gali aprašyti:

(a) teigiamą papildymą tarp aibės A ir aibės B: $g(A \cup B) > g(A) + g(B)$;

(b) neigiamą papildymą tarp aibės A ir aibės B: $g(A \cup B) < g(A) + g(B)$;

(c) nėra sąveikos tarp aibių A ir B, t. y. aibės A ir B nepersidengia informacija: $g(A \cup B) = g(A) + g(B)$.

Klasikinė tikimybių teorija apima tik atvejį (c). Visi šie trys atvejai (a), (b), ir (c) iliustruoja, kad neraiškaus mato naudojimas yra platesnis negu klasikinio mato.

2 Apibrėžimas. Tegu g yra aibėje Z apibrėžtas neraiškus matas. Tada funkcijos $h: Z \to R^+$ diskretinis Čoket (angl. *„Choquet“*) integralas atžvilgiu g skaičiuojamas pagal formulę:

$$C_g\left(h(z_1),...,h(z_L)\right) = \sum_{i=1}^{L} \left[h(z_i) - h(z_{i-1})\right] g(A_i), \qquad \textbf{(1.20)}$$

čia indeksai i yra surikiuoti taip, kad $0 \le h(z_1) \le ... \le h(z_L) \le 1$, $A_i = \{z_i,...,z_L\}$, ir $h(z_0) = 0$, o L yra elementų skaičius aibėje Z. Kita skaičiavimo formulė baigtinei aibei gali būti užrašoma taip:

$$C_g\left(h(z_1),...,h(z_L)\right) = \sum_{i=1}^{L} h(z_i)\left(g(A_i) - g(A_{i+1})\right), \qquad \textbf{(1.21)}$$

kur $g(A_{L+1}) = 0$ ir $A_i = \{z_i,...,z_L\}$.

Tokiu būdu apibrėždami diskretinį Čoket integralą pastebime, kad jis turi geras savybes grupavimui.

1.2.3. NERAIŠKAUS MATO PANAUDOJIMAS POŽYMIŲ SVARBAI IDENTIFIKUOTI

Apžvelgus literatūrą, pavyko rasti ir aprašyti 4-ių tipų neraiškiuosius matus, būtent pilnąjį neraiškų matą (angl. *„ordinary*“), λ-neraiškų matą, kiekinį (angl. *„cardinal*“) neraiškų matą ir 2-os eilės (angl. *„2-order*“) adityvųjį neraiškų matą. Tačiau eksperimentus atliksime tik su vienu iš jų, t.y. su pilnuoju neraiškiu matu.

Neraiškus integralas su pilnuoju matu priskiria svarbumo tankį kiekvienai galimai požymių kombinacijai pradedant atskirais požymiais ir einant iki grupės su visais požymiais, t.y. pilnąjį matą sudaro $2^L - 2$ realūs koeficientai kiekvienai kombinacijai. Kiekvienas tokio sudaryto mato tankis ir išreiškia požymių grupės svarbumą.

1.2.3.1. PILNASIS NERAIŠKUS MATAS

Kadangi $g(\varnothing) = 0$ ir $g(Z) = 1$ (žr. 1 apibrėžimą), tai pilnasis neraiškus matas yra apibrėžiamas pagal $2^L - 2$ realius koeficientus, kur *L* yra požymių skaičius aibėje *Z*.

1.2.3.2. λ-NERAIŠKUS MATAS

Bendru atveju, dviejų nepersidengiančių poaibių sąjungos neraiškus matas negali būti tiesiogiai išskaičiuotas iš tų poaibių neraiškaus mato tankių. M. Sugeno 1977 metais pasiūlė išskaidomą λ-neraiškų matą, tenkinantį papildomas savybes:

$$g(A \cup B) = g(A) + g(B) + \lambda \cdot g(A)g(B), \qquad \textbf{(1.22)}$$

visiems $A, B \subset Z$ ir $A \cap B = 0$, esant $\lambda > -1$.

Tegul $Z = \{z_1, z_2, ..., z_L\}$ būna baigtinė požymių aibė ir tegu $g^i = g(\{z_i\})$. Reikšmės g^i yra neraiškaus mato parametrai, vadinami tankiais. Tada reikšmė λ randama iš sąlygos $g(Z) = 1$, kas yra ekvivalentiška išspręsti lygtį:

$$\lambda + 1 = \prod_{i=1}^{L} (1 + \lambda \cdot g^i). \qquad \textbf{(1.23)}$$

Kai g yra λ-neraiškus matas, tai reikšmės $g(A_i)$ skaičiuojamos rekursijos būdu:

$$g(A_1) = g(\{z_1\}) = g^1, \qquad \textbf{(1.24)}$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda \cdot g^i g(A_{i-1}), \text{ visiems } 1 < i \le L. \qquad \textbf{(1.25)}$$

λ-neraiškus matas apima (įtraukia) tikimybinį matą, superadityvų ir subadityvų matus kaip atskirus atvejus:

$$g_\lambda(Z) = \begin{cases} \text{superadityvus matas, } \lambda > 0, \\ \text{tikimybinis matas, } \lambda = 0, \\ \text{subadityvus matas, } -1 < \lambda < 0. \end{cases}$$

Superadityvus neraiškus matas interpretuojamas kaip tikėtinumas, kad elementas *z* iš erdvės 2^Z priklauso aibei A_i, o jo papildymas priklauso aibei $\overline{A}_i$, pastebėtina, kad $g_{\lambda>0}(A_i) + g_{\lambda>0}(\overline{A}_i) \le 1$.

Tikimybiniam matui galioja taisyklė: $g_{\lambda=0}(A_i)+g_{\lambda=0}(\overline{A}_i)=1$, o subadityviam neraiškiam matui: $g_{-1<\lambda<0}(A_i)+g_{-1<\lambda<0}(\overline{A}_i)\geq 1$ [8].

1.2.3.3. KIEKINIS NERAIŠKUS MATAS

Kiekinis neraiškus matas (literatūroje dažnai dar vadinamas LOS operatoriumi) – tai toks neraiškus matas, kuris gali būti parinktas taip, kad mato tankiai priklausytų tik nuo aibės dydžio [9, 10]:

$$g(A)=\sum_{j=0}^{i-1} w_{L-j} \tag{1.26}$$

$\forall A$ toks, kad $|A| = i$, kur w_i yra koeficientas. Tada $g(A_i)$ nepriklauso nuo prijungiamos aibės vertės, t.y. nepriklauso nuo aibės sutvarkymo: $h(z_1)\leq h(z_2)\leq ...\leq h(z_L)$. Vadinasi, skirtumas $g(A_i)-g(A_{i+1})$ gali būti perrašytas taip:

$$w_i = g(A_i)-g(A_{i+1}), \tag{1.27}$$

todėl

$$C_g = \sum_{i=1}^{L} w_i h(z_i). \tag{1.28}$$

Tegul $\mathbf{z}=(z_1,z_2,...,z_L)$ yra vektorius. *i*-toji vektoriaus **z** statistika $z_{(i)}$ yra mažiausias vektoriaus z elementas, kur $z_{(1)}\leq z_{(2)}\leq ...\leq z_{(L)}$. Tegul $\mathbf{w}=(w_1,w_2,...,w_L)$ yra svorio vektorius, sukonstruotas taip, kad $\sum_{i=1}^{L} w_i = 1$ ir $0\leq w_i\leq 1,\ \forall i=1,2,...,L$.

Tada vektoriaus $\mathbf{z}=(z_1,z_2,...,z_L)$ LOS operatorius su svorio vektoriumi $\mathbf{w}=(w_1,w_2,...,w_L)$ apibrėžiama taip:

$$LOS(\mathbf{z},\mathbf{w})=\sum_{i=1}^{L} w_i z_{(i)}. \tag{1.29}$$

1.2.3.4. 2-OS EILĖS ADITYVUS NERAIŠKUS MATAS

3 Apibrėžimas [12]**.** Pseudo-Bulio funkcija yra reali reikšminga funkcija $f:\{0,1\}^L\to R$. Neraiškus matas gali būti laikomas kaip tam tikras atskiras pseudo-Bulio funkcijos atvejis, apibrėžtas bet kokiems $A\subset Z$ taip, kad *A* ekvivalentiška taškui $(z_1,...,z_L)$ aibėje $\{0,1\}^L$, kur $z_i=1$,

jeigu $i \in A$. Galima parodyti, kad bet kokia pseudo-Bulio funkcija gali būti išreikšta kaip daugiatiesinis polinomas iš *L* kintamųjų:

$$f(z) = \sum_{T \subset Z} \left[a(T) \prod_{i \in T} z_i \right] \tag{1.30}$$

su $a(T) \in R$ ir $z = (z_1, ..., z_L) \in \{0,1\}^L$.

Koeficientas $a(T)$, $T \subset Z$ yra aibės funkcija, atitinkanti Möbius transformaciją. Tegul *g* yra aibės funkcija $g : 2^Z \to R$. Tada *g* Möbius transformacija yra funkcija *a* iš aibės Z, jeigu (Mesiar, 1999)

$$a(T) = \sum_{K \subset T} (-1)^{|T \setminus K|} g(K), \ \forall T \subset Z . \tag{1.31}$$

Möbius transformacija yra atvirkštinė. Jeigu *a* yra žinoma, tai pagal Zeta-transformaciją įmanoma atkurti g:

$$g(T) = \sum_{S \subset T} a(S), \ \forall T \subset Z . \tag{1.32}$$

Jeigu matas yra adityvus ir išreikštas koeficientais g^i, $i = 1, \ldots, L$, tada atitinkama pseudo-Bulio funkcija bus tiesinė: $f(z) = \sum_{i=1}^{L} a_i z_i$. Pastaba: $g^i \equiv a_i$. Turėdami 2-os eilės neraiškų adityvųjį matą, galime apibrėžti ir *k*-os eilės adityvųjį neraiškų matą, turintį *k*-o laipsnio polinominį vaizdavimą.

4 Apibrėžimas [12]**.** Neraiškus matas *g*, apibrėžtas aibėje *Z*, yra *k*-os eilės adityvus, jeigu atitinkama pseudo-Bulio funkcija yra daugiatiesinis polinomas iš *k* laipsnių, t. y. $a(T) = 0 \ \forall T$ tokia, kad $|T| > k$ ir egzistuoja bent vienas T iš k elementų, kad $a(T) \neq 0$. Bet kokiems $K \subset Z$ ir $|K| \geq 2$ 2-adityvus neraiškus matas apibrėžiamas taip:

$$g(K) = \sum_{i=1}^{L} a_i z_i + \sum_{\{i,j\} \subset Z} a_{ij} z_i z_j \tag{1.33}$$

su $z_i = 1$, jeigu $i \in K$ ir $z_i = 0$ kitais atžvilgiais. Pagrindinė formulė yra tokia (12):

$$g(K) = \sum_{i \in K} a_i + \sum_{\{i,j\} \subseteq K} a_{ij} = \sum_{\{i,j\} \subseteq K} g^{ij} - \left(|K| - 2\right) \sum_{i \in K} g^i \tag{1.34}$$

kiekvienam $K \subset Z$ tokiam, kad $|K| \geq 2$, kur $|K|$ yra aibės *K* kiekis (dydis) ir $g^{ij} = g(\{z_i, z_j\}) = a_i + a_j + a_{ij} = g^i + g^j + a_{ij}$. Taigi, 2-adityvus neraiškus matas yra apibrėžiamas koeficientų g^i ir g^{ij}. 2-adityviojo neraiškaus mato nustatymui reikia *L*(*L*+1)/2 koeficientų.

1.2.4. NERAIŠKIŲ MATŲ APMOKYMAS

Pilnojo neraiškaus mato, 2-os eilės adityviojo neraiškaus mato ir λ-neraiškaus mato svorius galima apmokyti minimizuojant kvadratinį kriterijų [13]. Neraiškus matas g dviem klasėm randamas minimizuojant kvadratinį kriterijų:

$$J = \sum_{n=1}^{N_1} \left\{ C_g^1\left(\mathbf{x}_n^1\right) - C_g^2\left(\mathbf{x}_n^1\right) - 1 \right\}^2 + \sum_{n=1}^{N_2} \left\{ C_g^2\left(\mathbf{x}_n^2\right) - C_g^1\left(\mathbf{x}_n^2\right) - 1 \right\}^2 , \tag{1.35}$$

čia N_i mokymo imties $\mathrm{x}_1^i, \mathrm{x}_2^i, ..., \mathrm{x}_N^i$, $i = 1,2$ skaičius iš *i*-tosios klasės; $C_g^1\left(\mathbf{x}_n^i\right)$ ir $C_g^2\left(\mathbf{x}_n^i\right)$ yra Čoket integralas atitinkamai pirmai ir antrai klasei. Minimizavimo procedūra gali būti suvesta į kvadratinio programavimo uždavinį [13]. Daug klasių atveju kriterijus apibendrinamas taip [14]:

$$J = \sum_{i=1}^{Q} \sum_{n=1}^{N_i} \sum_{j=1, j\neq i}^{Q} \left| \Delta C_g^{ij}(\mathbf{x}_n^i) - d^{ij}(\mathbf{x}_n^i) \right|^2 , \tag{1.36}$$

čia

$$\Delta C_g^{ij}(\mathbf{x}_n^i) \equiv C_g^i(\mathbf{x}_n^i) - C_g^j(\mathbf{x}_n^i) \tag{1.37}$$

$d^{ij}(\mathbf{x}_n^i)$ yra reikalaujamoji reikšmė kiekvienam $\Delta C_g^{ij}(\mathbf{x}_n^i)$ ir N_i yra mokymo imties skaičius iš *i*-tosios klasės.

1.2.5. 2-OS EILĖS ADITYVAUS NERAIŠKAUS MATO SUDARYMAS

Dėl vienareikšmio atitikimo tarp Möbius transformacijos ir neraiškaus mato monotoniškumo savybės, reikšmės $a(T)$, esančios (1.30) formulėje, turi tenkinti apribojimus (žr. 1 teoremą) [15].

1 Teorema. 2^L koeficientai iš $a(T)$, $T \subset Z$ atitinka neraiškių matų Möbius transformaciją, jeigu

1. $a(0) = 0$, $\sum_{T \subset Z} a(T) = 1$;
2. $\sum_{i \in B \in T} a_B \geq 0$ visiems $T \subset Z$ ir visiems $i \subset T$.

Sudarant 2-os eilės adityvų neraiškų matą, tik $L(L+1)/2$ koeficientai g^i ir g^{ij}, $i, j \in Z$ turi būti nustatyti iš mokymo duomenų. Tam, kad gautumėme monotonišką neraiškų matą, koeficientai g^i ir g^{ij} turi tenkinti sąlygas, suformuluotas 1 Teoremoje. Monotoniškumo savybė, apribojanti 2-os eilės adityvų neraiškų matą, gali būti gauta iš antros sąlygos (žr. 1 teoremą) [16]:

$$\sum_{j\in K} g^{ij} - \sum_{j\in K} g^{j} - (L-2)g^{i} \geq 0, \quad \forall i \in Z, \quad K \subseteq Z \setminus i, \tag{1.38}$$

kur |Z| = *L*.

Tam, kad gautumėme normalizuotą į intervalą [0,1] neraiškų matą, koeficientai g^i ir g^{ij} turi tenkinti normalizavimo sąlygą, gautą (1.39) formulėje. Kai *K* = *Z*, tai:

$$\sum_{\{i,j\}\subseteq Z} g^{ij} - (L-2)\sum_{i\in Z} g^{i} = 1 \tag{1.39}$$

Jeigu *g* yra 2-os eilės adityvus neraiškus matas iš aibės *Z*, tai funkcijos $h: Z \to R^{+}$ diskretus Čoket integralas atžvilgiu *g* tampa toks:

$$C_g = \sum_{i\in L} a_i h(z_i) + \sum_{\{i,j\}\subseteq L} a_{ij}(h(z_i) \wedge h(z_j)) \tag{1.40}$$

kur ženklas „∧" atlieka minimumo operaciją, o funkcija $h(z)$ yra gaunama pagal duomenų požymių reikšmes.

Sudarant 2-os eilės adityvų neraiškų matą, M. Grabisch pasiūlė naudoti *Shapley reikšmes* ir *sąveikos indeksus*.

5 Apibrėžimas [17]**.** Tegul *g* yra neraiškus aibės *Z* matas. *i*-tojo elemento reikšmingiausias indeksas arba Sharpley reikšmė atžvilgiu *g* apibrėžiamas taip:

$$v_i = \sum_{k=0}^{L-1} \gamma_k \sum_{K\subset Z\setminus i, |K|=k} \{g(K\cup\{i\}) - g(K)\}, \tag{1.41}$$

kur

$$\gamma_k = \frac{(L-k-1)!k!}{L!}, \tag{1.42}$$

kur *L* yra elementų skaičius aibėje *Z*.

Pagrindinė Sharpley reikšmės savybė yra ta, kad $\sum_{i=1}^{L} v_i = 1$.

6 Apibrėžimas [9]**.** Sąveikos indeksas tarp dviejų elementų *i* ir *j* atžvilgiu neraiškaus mato *g* apibrėžiamas taip:

$$I_{ij} = \sum_{k=0}^{L-2} \xi_k \sum_{K\subset Z\setminus\{i,j\}, |K|=k} \{g(K\cup\{i,j\}) - g(K\cup\{i\}) - g(K\cup\{j\}) + g(K)\}, \tag{1.43}$$

čia

$$\xi_k = \frac{(L-k-2)!k!}{(L-1)!}. \tag{1.44}$$

2 Teorema [18]**.** Tegul $v_1,...,v_L$ yra Sharpley reikšmių aibė, tenkinanti $\sum_{i=1}^{L} v_i = 1$, ir tegul I_{ij}, $\{i, j\} \in Z$ yra sąveikos indeksų aibė. Čia egzistuoja vienintelis 2-os eilės adityvus neraiškus matas (galiausiai nemonotoniškas), ekvivalentus pseudo-Bulio funkcijai $\sum_{i=1}^{L} a_i + \sum_{\{i,j\}\subset Z} a_{ij}$ ir apibrėžiamas taip:

$$a_i = v_i - \frac{1}{2}\sum_{j\in Z\setminus i} I_{ij},\ i = 1,...,L \qquad \textbf{(1.45)}$$

$$a_{ij} = I_{ij},\ \{i, j\} \subset Z\ . \qquad \textbf{(1.46)}$$

Šis v_i ir I_{ij} apribojimas 2-os eilės adityviam neraiškiam matui suteikia (užtikrina) monotoniškumo savybę.

3 Teorema [18]**.** Sharpley reikšmių $v_1,...,v_L$ aibė ir sąveikos indeksai $I_{ij}, \{i, j\} \in Z$ 2-os eilės adityviam neraiškiam matui suteikia monotoniškumo savybę tada ir tik tada, jeigu jie tenkina šią apribojimų aibę:

$$-v_i \le \frac{1}{2}\left(\sum_{j\in Z\setminus K\cup\{i\}} I_{ij} - \sum_{k\in K} I_{ik}\right) \le v_i\,, \qquad \textbf{(1.47)}$$

$$K \subset Z\setminus i\,,\ i = 1,...,L$$

Grabisch (1996) rankiniu būdu nustatė aibės reikšmingumą ir sąveikos indeksus. Tačiau toks rankinis indeksų nustatymas nėra lengva užduotis.

Ieškant optimalius 2-os eilės adityvaus neraiškaus mato tankius, būtina tenkinti šiuos apribojimus:

$$\begin{cases} a(\varnothing) = 0, \\ \sum_{i\in L} a_i + \sum_{\{i,j\}\subseteq L} a_{ij} = 1, \\ a_i \ge 0, \forall i \in L, \\ a_i + \sum_{j\in T} a_{ij} \ge 0, \forall i \in L, \forall T \subseteq L\setminus i. \end{cases} \qquad \textbf{(1.48)}$$

1.3. ANALITINIS POŽYMIŲ ERDVĖS MAŽINIMO METODŲ PALYGINIMAS

Yra žinomi du požymių mažinimo būdai: požymių atrinkimo metodas ir požymių ekstrakcijos metodas. Požymių atrinkimo metodas atrenka nepriklausomus požymius, kurie suteikia pakankamą informaciją norimam uždaviniui spręsti. Fizinė atrinktų požymių prasmė išlieka nepakitusi. Požymiai su didžiausia koreliacija gali būti šalinami. Tačiau čia susiduriama su tam tikra problema – koreliacijos trūkumu: požymiai lyginami tik poromis, o ne visi kartu. Požymių atrinkimo metodams priklauso klasterizavimo metodas ir neraiškus integralas su pilnuoju matu.

Požymių ekstrakcijos metodas veikia priešingai. Šiuo atveju visi požymiai projektuojami į mažesnės eilės požymių erdvę panaudojant įvairias transformacijos funkcijas. Po projektavimo fizinės reikšmes pakinta ir fizikinė prasmė nebėra tokia pati kaip pradinėje aibėje. Transformacijos funkcija yra analizinė funkcija, kurios pagrindinis reikalavimas yra sukurti informatyvią transformuotą požymių aibę. Labai gerai žinomi du tradiciniai požymių išskyrimo metodai – tai principinių komponenčių analizės metodas ir „butelio kaklelio“ metodas su auto-asociatyviniu neuroniniu tinklu. PCA metodu požymiai projektuojami kryptimi, turinčia didžiausią dispersiją. O daugiasluoksnio perceptrono neuroninis tinklas mokomas atvaizduoti duomenis į tas pačias vertes informaciją suspaudžiant „butelio kaklelyje“. Siekiant padidinti duomenų suspaudimo laipsnį, neuronų skaičius paslėptame „butelio kaklelyje“ yra mažesnis nei įėjimo sluoksnio.

Tačiau taikant požymių erdvės mažinimo metodus egzistuoja tam tikra klaida, kuri atvaizduoja pradinių duomenų informacijos praradimą. Todėl požymių erdvės mažinimas naudingas tik tada, kai informacijos praradimas nėra lemiamas uždavinio (problemos) sprendiniui.

2. TIRIAMOJI DALIS

2.1. EKSPERIMENTE NAUDOTŲ DUOMENŲ BAZIŲ APIBŪDINIMAS

Kelios Europos mokslinės grupės, dirbančios dirbtinio intelekto srityje, realizavo projektus PROBEN1 ir ELENA. Šių projektų tikslai buvo sukurti standartines duomenų bazes bei palyginti žinomus duomenų klasifikavimo būdus. Projektų autoriai rekomendavo naudoti šias duomenų bazes tiriant žinomus bei naujus klasifikatorius.

Šiame darbe buvo paimtos keturios realių duomenų bazės: vėžio (cancer), diabeto (diabetes), stiklo (glass) ir palydovinių vaizdų (satimage). Šios duomenų bazės yra laisvai prieinamos kiekvienam vartotojui internete [29, 30]. Glausta informacija apie šios duomenis pateikta 2.1 lentelėje.

2.1 lentelė

Glausta informacija apie naudotus duomenis

Duomenų bazės	Klasių kiekis	Požymių kiekis	Pavyzdžių skaičius (duomenų taškai)
Vėžio	2	9	699
Diabeto	2	8	768
Stiklo	6	9	214
Palydovinių vaizdų	6	5	6435

2.1.1. VĖŽIO (CANCER) DUOMENŲ BAZĖ

Ši vėžio duomenų bazė buvo sukurta Viskonsino medicinos universitete, iš daktaro William H. Wolberg atliktų medicininių tyrinėjimų.

Iš duomenų bazės sudaryta matrica iš 683 eilučių (atskirų atvejų (pacientų)), ir iš 10-ies stulpelių – 9-ių požymių (tyrimų rezultatų) bei klasės. Kiekvienas atskiras atvejis gali įgyti vieną iš dviejų klasės reikšmių, reiškiantį gėrybinį arba piktybinį auglį.

Vėžio duomenų bazės požymiai:

1. Kodo numerio pavyzdys;
2. Grupė sustorėjimų;
3. Ląstelių dydžio vienodumas;
4. Ląstelių formos vienodumas;
5. Nežymus suaugimas;

6. Vieno ląstelės dengiamojo audinio dydis;
7. Labai menkas branduolys;
8. Raminamas chromuotinas;
9. Normalus branduolys;
10. Mitozė.

2.1.2. DIABETO (DIABETES) DUOMENŲ BAZĖ

„Pirma“ Indijos diabeto duomenų bazė: ji susideda iš 768 pavyzdžių, kurie aprašomi aštuoniais biologiniais požymiais. Tikslas yra nustatyti ar asmuo turi diabetą ar ne.

Pavyzdžių skaičius yra 768, 8 požymiai .

Požymiai skirstomi:

1. Nėštumo kartų skaičius;
2. Plazmos gliukozės koncentracija dviejų valandų oralinio gliukozės pakantumo teste;
3. Kraujo spaudimas (mm Hg);
4. Tricepso odos storumas (mm);
5. Dvivalandinis serumo insulinas (mu U/ml);
6. Kūno masės indeksas (svoris kg/ (ūgio m) ^ 2);
7. Diabeto genealogijos funkcija;
8. Amžius (metais)

Klasės vertė 1 yra teigiamas diabeto testas.

2.1.3. STIKLO (GLASS) DUOMENŲ BAZĖ

Stiklo duomenų bazė buvo sukurta Centrinėje Atradimų Įstaigoje, Berkšire. Duomenų bazė turi 214 pavyzdžių suskirstytų į 6 klases. Kiekvieną pavyzdį duomenų nustatyme identifikuoja identifikacijos skaičius, devyni cheminiai elementai:

ID, N -- atskiro atvejo atpažinimo numeris

RI, N -- refrakcijos indeksas

NA2O, N -- natrio oksidas

MGO, N -- magnio oksidas

AL2O3, N – aliuminio oksidas

SIO2, N -- silicio oksidas

K2O, N -- kalio oksidas

CAO, N -- kalcio oksidas

BAO, N -- bario oksidas

FE2O3, N -- geležies oksidas

TYPE, N -- nežinomasis, tačiau turi derintis prie aukščiau išvardintų

CAMG, N -- neapibrėžtas

Stiklo duomenų bazės klasės (išėjimai):

1. WF (plokščias langas)
2. WNF (ne plokščias langas)
3. C (konteineris)
4. T (stalo paviršius)
5. H (lemputė)

2.1.4. PALYDOVINIŲ VAIZDŲ (SATIMAGE) DUOMENŲ BAZĖ

Šie duomenys buvo sudaryti perdirbus daugiaspektrio skenerio vaizdus, gautus iš dirbtinio žemės palydovo. Vienas vaizdo kadras susideda iš keturių nuotraukų skirtingose spektro srityse. Dvi nuotraukos iš matomos spektro dalies (žalios ir raudonos) ir dvi iš infraraudonos dalies. Kiekvienas vaizdo pikselis aprašytas 8 bitų žodžiu, atitinkamai 0 - tai juoda ir 255 - balta spalva. Visi vaizdai sudaryti iš 2340*3380 pikselių masyvų, tačiau, sudarant šią duomenų bazę, buvo paimti tik šių vaizdų poaibiai po 82*100 pikselių.

Kiekviena eilutė duomenų bazėje sudaryta iš 36 požymių (4 spektro juostos po 9 požymius) ir klasės numerio. Devyni požymiai - tai kaimyninių pikselių reikšmės, esančios 3*3 vaizdo kvadrate. Ši duomenų bazė yra šešių klasių, klasių apibūdinimas pateiktas 2 lentelėje.

Šiame darbe buvo naudoti SATIMAGE duomenys, apdoroti Diskriminantinės faktorinės analizės metodu. Diskriminantinės faktorinės analizės metodu naujai suformuotų požymių kiekis priklauso nuo klasių skaičiaus (Q) ir įėjimo požymių kiekio (n). Kai n didesnis už Q, tai formuojamų požymių kiekis nustatomas lygiu $Q-1$. Šiai šešių klasių ($Q=6$) duomenų bazei naujų požymių kiekis lygus 5. Nauji požymiai formuojami siekiant padidinti klasių atskiriamumą ir mažinant duomenų išsibarstymą klasės viduje.

Vidutinė klasifikavimo klaida šiems 5-ų požymių duomenims gauta ELENA projekte su DSP neuroniniu tinklu lygi 12.0%.

2.2 lentelė

Palydovinių vaizdų duomenų klasės

Klasės numeris	Originalus pavadinimas	Vertimas	Pavyzdžių skaičius
1	red oil	raudona dirva	1533
2	cotton crop	medvilnės plantacija	703
3	grey soil	pilka dirva	1358
4	damp grey soil	pilka - drėgna dirva	626
5	soil with vegetation stubble	dirva su augmenija	707
6	very damp grey soil	pilka ir labai šlapia dirva	1508

2.1.5. KITŲ MOKSLININKŲ GAUTI REZULTATAI SU VĖŽIO, DIABETO, STIKLO IR PALYDOVINIŲ VAIZDŲ DUOMENŲ BAZĖMIS

Šiame skyriuje pateiksime kitų mokslininkų gautus rezultatus panaudojant vėžio, diabeto, stiklo ir palydovinių vaizdų duomenų bazes (žr. 2.3 lentelę).Šiame darbe gautus rezultatus palyginsime su 2.3 lentelėje pateiktais rezultatais [32, 33].

2.3 lentelė

Kitų mokslininkų gautos vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai vėžio, diabeto, stiklo ir palydovinių vaizdų duomenų bazėms

Duomenų bazės	Vidutinė klasifikavimo klaida %	Standartinis nuokrypis
Vėžio	3.0	1.20
Diabeto	29.6	2.20
Stiklo	49.8	6.61
Palydovinių vaizdų	11.9	1.0

2.2. *k*-ARTIMIAUSIŲ KAIMYNŲ KLASIFIKATORIUS

Klasifikavimo uždavinius galima spręsti ne tik grupuojant elementus pagal nustatytas ar analizės metu rastas taisykles, bet ir pagal elemento panašumą į savo kaimynus. Šie algoritmai remiasi daikto,

veiksmo ar kitaip apibrėžiamo nagrinėjamo vieneto lyginimu su prieš tai buvusiais. Analizė remiasi istorine informacija, todėl algoritmo trūkumas — sudaryto modelio dydis (sukauptos istorinės informacijos dydis).

Vienas iš labiausiai žinomų ir nagrinėjamų klasifikatorių yra *k*-artimiausių kaimynų (*k*-NN) klasifikatorius, kuris naują nežinomą įėjimo požymį (duomenį) klasifikuoja pagal jo panašumą į *k* žinomų artimiausių kaimynų. Pagal užsiduotas ribas, jis priskiria elementą tai klasei, kurios elementų yra daugiausia iš artimiausių *k* kaimynų.

Vienas iš pagrindinių trūkumų naudojant *k*-NN klasifikatorių yra ganėtinai ilgas kompiuterinio skaičiavimo laikas apdorojant didelius duomenų rinkinius, ypatingai kai pasirinktas artimiausių kaimynų skaičius $k>1$ [31].

2.3. EKSPERIMENTO EIGOS APIBŪDINIMAS

Visi šiame darbe aprašyti eksperimentai buvo atlikti naudojant 2.1 skyriuje aprašytus duomenis.

Šiame darbe eksperimentai atlikti su MATLAB programų paketu. MATLAB paketas remiasi nesudėtinga ir lanksčia programavimo/valdymo kalba, kuria patogu aprašyti sprendžiamus matematinius uždavinius.

Duomenų imtis atsitiktinai sudalinta į dvi lygias dalis: *mokymo* ir *testavimo*. Suspaudimo kokybė vertinama *k*-artimiausių kaimynų (*k*-NN) klasifikatoriumi, klasifikavimo uždavinį kartojant 20 kartų.

Visuose bandymuose buvo naudoti 4 metodai: klasterizavimo, neraiškaus neuroninio tinklo, principinių komponenčių analizės (PCA) metodai ir požymių atrinkimo neraiškiu integralu metodas. Gautos vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai buvo paskaičiuoti iš tų 20 realizacijų. Geriausias suspaudimo laipsnis atrenkamas eksperimentiškai stebint rezultatus. Suspaudimo laipsnis su mažiausia vidutine klasifikavimo klaida bei mažiausiu išsibarstimu (standartiniu nuokrypiu) laikomas geriausiu ir lyginamas su ta vidutine klasifikavimo klaida ir standartiniu nuokrypiu, kurie buvo gauti *k*-NN artimiausių kaimynų klasifikatoriumi.

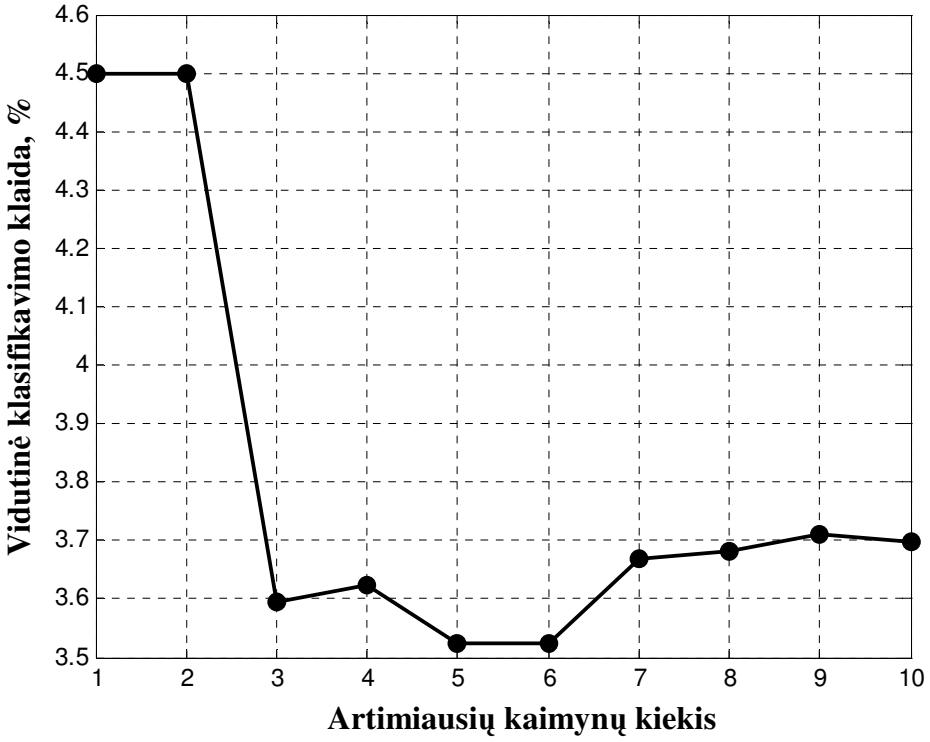
Atliekant eksperimentus su neuroniniu tinklu, mokymas vyko minimizuojant suminės kvadratinės klaidos funkciją. Neuroninio tinklo struktūrai parinkti neuronai su tiesine perdavimo funkcija.

Pateiktose rezultatų lentelėse pateiktos vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai skaičiuoti iš tų 20 eksperimentų.

2.4. EKSPERIMENTAI SU VĖŽIO DUOMENŲ BAZE

2.4.1. *k*-NN KLASIFIKATORIAUS ANALIZĖ

Konstruojant *k*-NN klasifikatorių vėžio duomenims klasifikuoti, reikėjo nustatyti šiai duomenų bazei tinkamiausią artimiausių kaimynų kiekį *k.* Atlikdami eksperimentus su *k*-NN artimiausiųjų kaimynų klasifikatoriumi, koeficientą *k* buvo keičiamas nuo 1 iki 10 ir stebima, prie kokių optimalių *k* reikšmių vidutinė klasifikavimo klaida yra mažiausia. Iš atliktų eksperimentų pastebėjau, jog klasifikuojant vėžio duomenis, vidutinė klaida yra 3.5%, kai artimiausių kaimynų skaičius $k = 6$ (žr. 2.1 pav.).



2.1 pav. *k* artimiausiųjų kaimynų vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su vėžio duomenų baze

2.4.2. KLASTERIZAVIMO METODO ANALIZĖ

Iš analizuotų vėžio požymių erdvės buvo išrenkami 2 ir daugiau požymių (maksimumas atitiko originalų požymių skaičių). Geriausias suspaudimo laipsnis atrenkamas eksperimentiškai stebint rezultatus. Suspaudimo laipsnis su mažiausia vidutine klasifikavimo klaida bei mažiausiu išsibarstymu (pagal *k*-NN klasifikatoriaus vidutinę klasifikavimo klaidą ir išsibarstymą) laikomas geriausiu.

2.4.3. MLP TINKLO ANALIZĖ

Kadangi suspaudimui buvo naudojamas MLP neuroninis tinklas, buvo būtina išlaikyti „butelio kaklelio“ struktūrą tinklo neuronų sluoksniuose. Todėl įvertinant rezultatus buvo nagrinėjami tik tie rezultatai, kuriuos atliekant paslėptųjų neuronų skaičius yra didesnis už suspaudimo laipsnį („butelio kaklelio“ sluoksnio elementų skaičių). Analizuotos vėžio duomenų bazės požymių aibė buvo spaudžiama į 2 ir daugiau požymių (maksimumas atitiko originalų požymių skaičių). Geriausias suspaudimo laipsnis atrenkamas eksperimentiškai stebint rezultatus. Suspaudimo laipsnis su mažiausia vidutine klasifikavimo klaida bei mažiausiu išsibarstymu laikomas geriausiu.

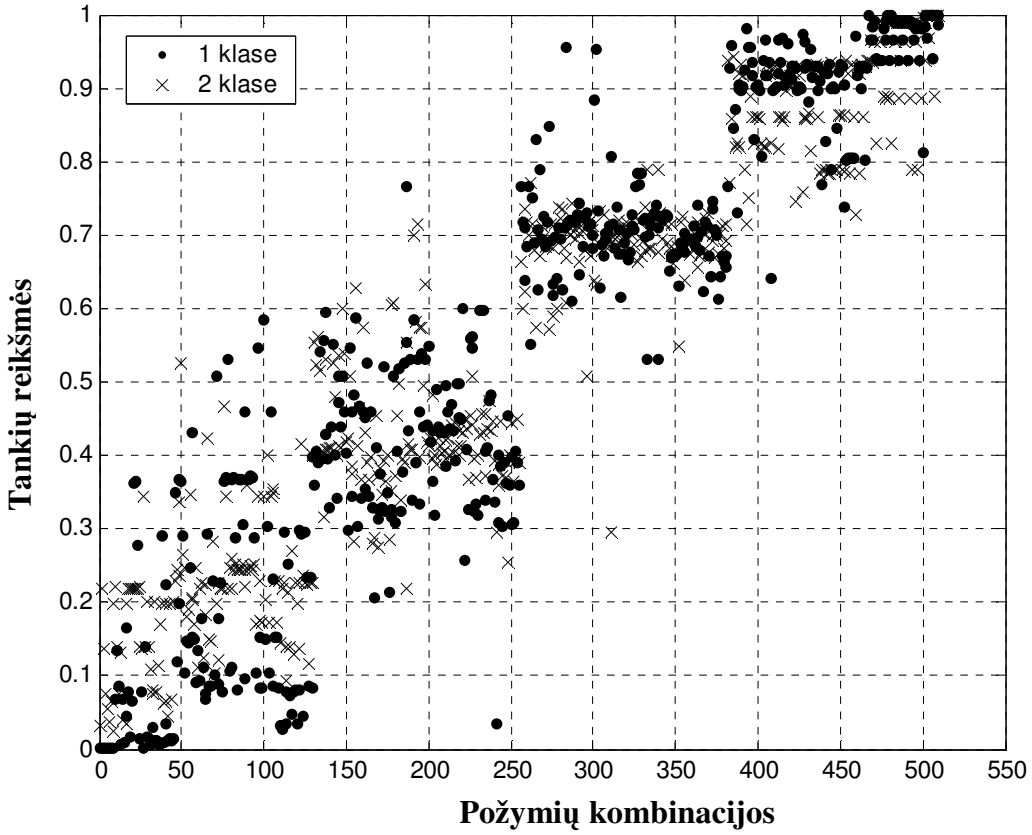
Konstruojant MLP klasifikatorių vėžio duomenims klasifikuoti buvo parinkta neuroninio tinklo struktūra su tiesinėmis neuronų perdavimo funkcijomis. Šiam eksperimentui atlikti buvo surašyta programa veikianti MATLAB programų paketo aplinkoje.

2.4.4. PCA METODO ANALIZĖ

Analizuotos vėžio, diabeto ir stiklo duomenų bazių požymių aibės buvo spaudžiamos nuo 1 iki $N-1$ principinių komponenčių (N - originalių požymių skaičius). Geriausias suspaudimo laipsnis atrenkamas eksperimentiškai stebint vidutines klasifikavimo klaidas. Suspaudimo laipsnis su mažiausia vidutine klasifikavimo klaida bei mažiausiu išsibarstymu (pagal minimalias ir maksimalias klaidas) laikomas geriausiu.

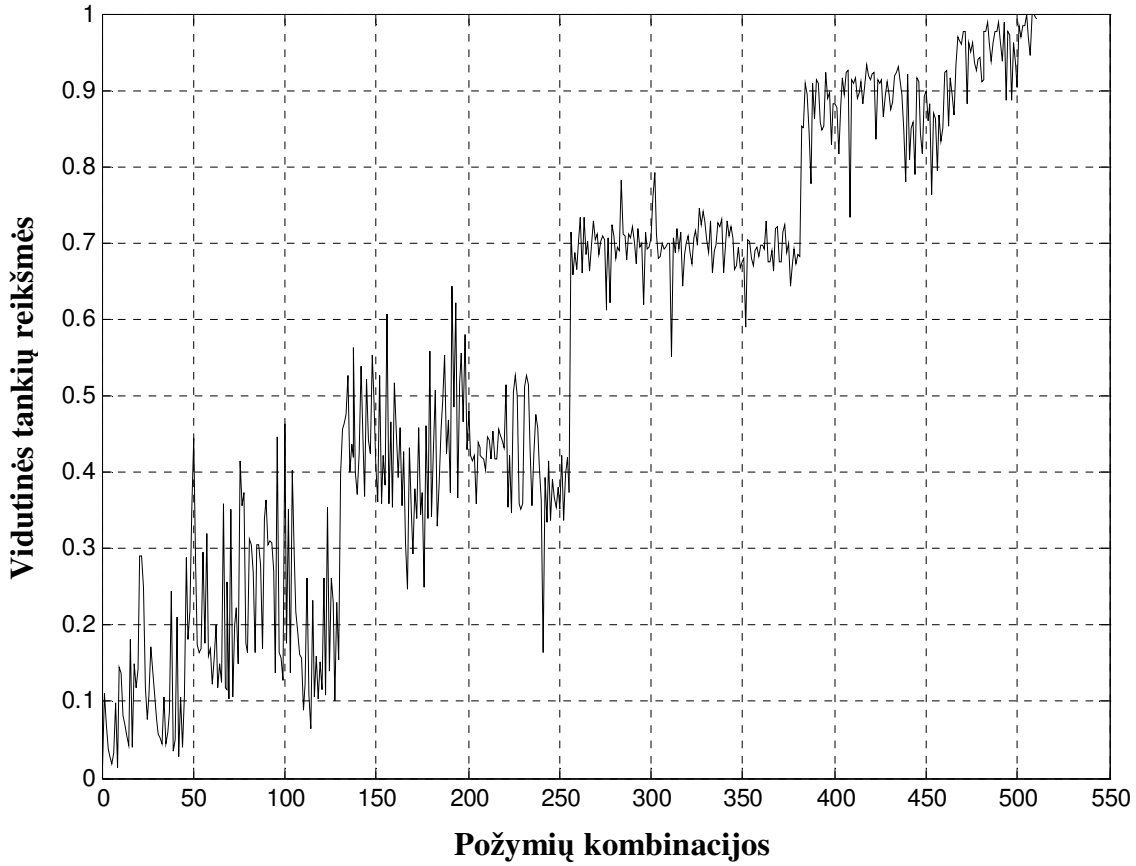
2.4.5. NERAIŠKAUS INTEGRALO SU PILNUOJU MATU ANALIZĖ

Analizuotos vėžio duomenų bazės požymių tankiai yra sugrupuoti poromis po viena, po du, po tris ir taip toliau, priklausomai nuo to, koks požymių kiekis yra šioje duomenų bazėje ir sudaromos požymių kombinacijos. 2.2 paveiksle pavaizduotos dvi požymių klasės, t.y. kaip tų klasių sudarytos požymių kombinacijos priklauso nuo tankių reikšmių. Kombinacija su didžiausia tankio reikšme laikoma geriausia (svarbiausia).



2.2 pav. Sudarytų požymių kombinacijų priklausomybė nuo tankių reikšmių

2.3 paveiksle pavaizduota požymių kombinacijų priklausomybė nuo vidutinių tankių reikšmių. Iš jo matyti, kad kombinacija su didžiausia vidutine tankio reikšme laikoma geriausia.



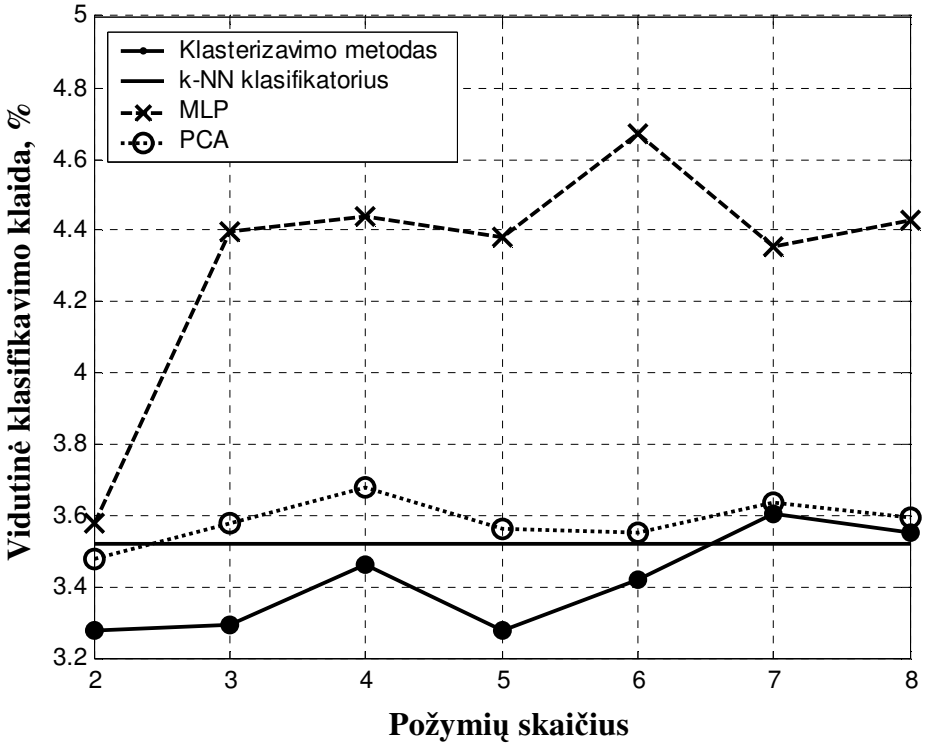
2.3 pav. Sudarytų požymių kombinacijų priklausomybė nuo vidutinių tankių reikšmių

Tyrimo rezultatai su vėžio duomenų baze apibendrinti 2.4 lentelėje. Atlikus eksperimentą 20 kartų ir palyginus gautus klasifikavimo rezultatus vėžio duomenų bazei (požymius mažinant iš 9 į 2,...,8) su *k*-NN klasifikatoriumi ir visais požymiais, matome, kad šiai duomenų bazei tinkamiausias yra požymių atrinkimo metodas, paremtas klasterizavimu. Šiuo metodu pakanka atrinkti tik du esminius požymius klasifikavimo uždaviniui spręsti. PCA metodas pasirodė pakankamai gerai ir jo pasiekta vidutinė klasifikavimo klaida buvo artima *k*-NN klasifikatoriaus vidutinei klaidai. Klasifikuojant požymius neraiškiu integralu su pilnuoju matu, gauta vidutinė klasifikavimo klaida taip pat buvo artima *k*-NN klasifikatoriaus vidutinei klaidai, tačiau šis metodas vėžio duomenų bazei netinka, nes nerekomenduoja mažinti požymių skaičiaus. Prasčiausiai šią duomenų bazę klasifikavo MLP tinklas, tai greičiausia dėl to, kad šis tinklas ieško skiriamųjų paviršių tarp duomenų taškų. Tačiau šio klasifikatoriaus privalumas – mažiausia struktūra, o tai garantuoja didelę klasifikavimo greitaveiką.

2.4 lentelė

Atrinktų požymių skaičius, vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai vėžio duomenų bazei, kai atlikta 20 eksperimentų

Duomenų bazė	Metodai, panaudoti požymių atrinkimui	Atrinktų požymių skaičius	Vidutinė klasifikavimo klaida %	Standartinis nuokrypis
Vėžio	*k*-NN klasifikatorius	9	3.5	0.67
	Klasterizavimo	2	3.3	0.62
	MLP	2	3.6	1.01
	PCA	2	3.5	0.70
	Neraiškus integralas su pilnuoju matu	8	3.4	0.80



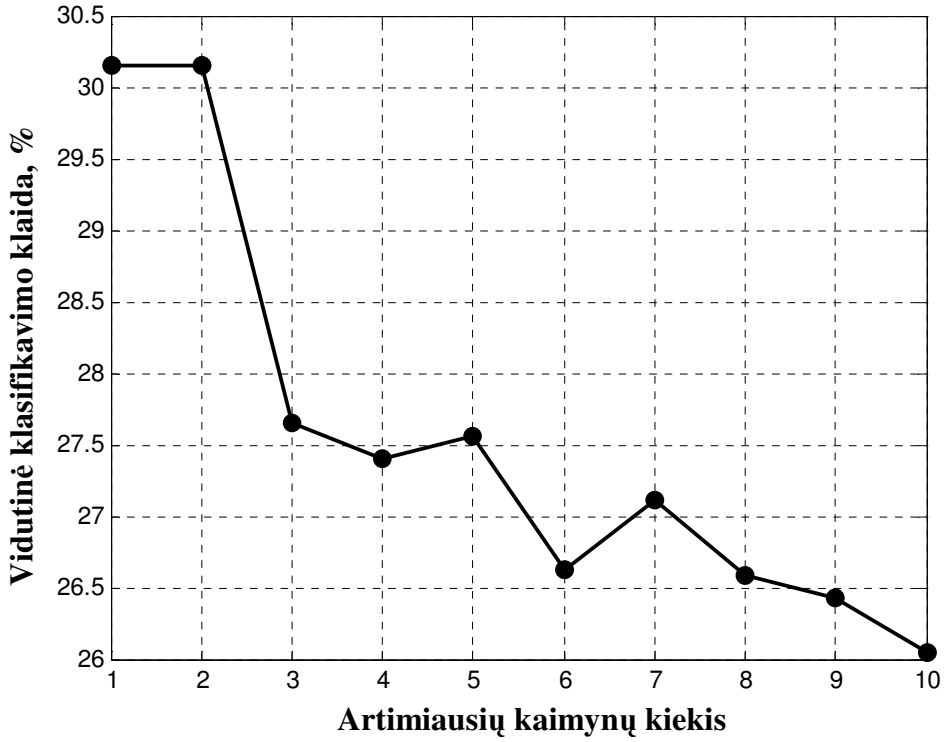
2.4 pav. Vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su vėžio duomenų baze

Kadangi vėžio duomenų bazę geriausiai klasifikavo klasterizavimo metodas su 3.3% vidutine klasifikavimo klaida, tai pasinaudoję Stjudento (t) testu ir pritaikę (1) formulę (žr. 1 priede), gauname, kad statistika T yra mažesnė už t_α ($t_\alpha = 1.684$) reikšmę, t. y. $0.464 < 1.684$.

Hipotezė H_0 su reikšmingumo lygmeniu $\alpha = 0.05$ yra priimama. Tai yra su 5% klaida teigiame, kad klasifikavimo klaidos yra panašios, o klaidos sumažėjimas nežymus.

2.5. EKSPERIMENTAI SU DIABETO DUOMENŲ BAZE

Konstruojant *k*-NN klasifikatorių diabeto duomenims klasifikuoti, reikėjo nustatyti tinkamiausią artimiausių kaimynų kiekį *k* šiai duomenų bazei. Atlikdami eksperimentus su *k*-NN klasifikatoriumi, koeficientą *k* keitėme nuo 1 iki 10. Eksperimentų rezultatai parodė, jog klasifikuojant diabeto duomenis, mažiausia vidutinė klaida buvo 26%, kai artimiausių kaimynų skaičius $k = 10$ (žr. 2.5 pav.)



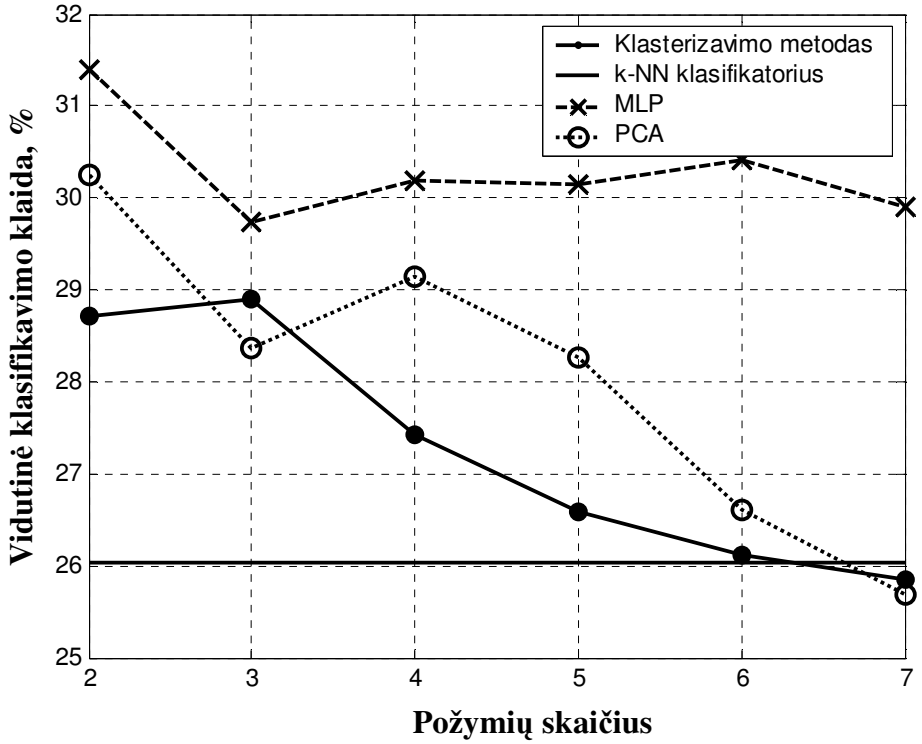
2.5 pav. *k* artimiausiųjų kaimynų vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su diabeto duomenų baze

Atlikę tyrimus su šia duomenų baze, gautus rezultatus apibendrinome 2.5 lentelėje. Iš jos matome, jog šiai diabeto duomenų bazei klasifikuoti geriausias yra klasterizavimo ir PCA metodai. Turėdami aštuonis klasifikavimo požymius ir panaudoję klasterizavimo arba PCA metodus, požymius galime sumažinti iki septynių. Prasčiausiai šią duomenų bazę klasifikavo MLP tinklas ir neraiškus integralas su pilnuoju matu.

2.5 lentelė

Atrinktų požymių skaičius, vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai diabeto duomenų bazei, kai atlikta 20 eksperimentų

Duomenų bazė	Metodai, panaudoti požymių atrinkimui	Atrinktų požymių skaičius	Vidutinė klasifikavimo klaida %	Standartinis nuokrypis
Diabeto	*k*-NN klasifikatorius	8	26.0	1.89
	Klasterizavimo	7	25.9	1.92
	MLP	3	29.7	2.71
	PCA	7	25.7	1.98
	Neraiškus integralas su pilnuoju matu	7	29.8	3.05



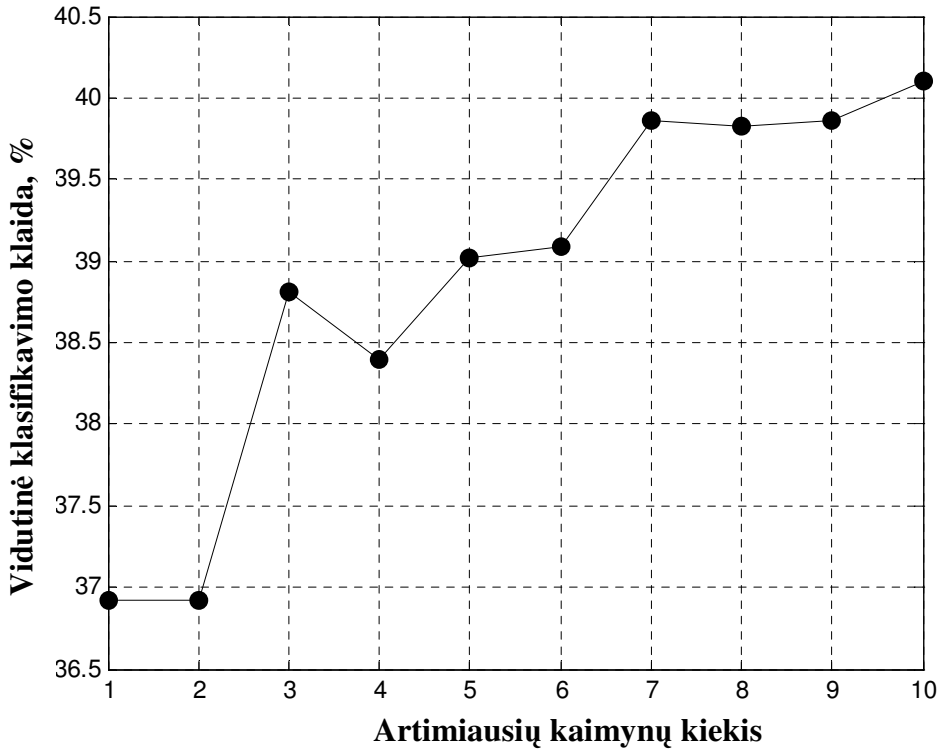
2.6 pav. Vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su diabeto duomenų baze

Pasinaudoję Stjudento (t) testu ir pritaikę (1) formulę (žr. 1 priede) *k*-NN ir klasterizavimo metodams, gauname, kad statistika T yra mažesnė už t_α reikšmę, t. y. $0.083 < 1.684$. Hipotezė H_0 su reikšmingumo lygmeniu $\alpha = 0.05$ yra priimama. Todėl su 5% klaida galime teigti, kad

klasifikavimo klaidos yra panašios, o klaidos sumažėjimas nežymus. Pritaikę (1) formulę *k*-NN ir PCA metodams, gauname, kad $|T| > t_\alpha \Rightarrow 0.045 < 1.684$. Hipotezė H_0 su reikšmingumo lygmeniu $\alpha = 0.05$ taip pat yra priimama. Tai yra su 5% klaida teigiame, kad klasifikavimo klaidos yra panašios, o klaidos sumažėjimas nežymus.

2.6. EKSPERIMENTAI SU STIKLO DUOMENŲ BAZE

Konstruojant *k*-NN klasifikatorių diabeto duomenims klasifikuoti, reikėjo nustatyti tinkamiausią artimiausių kaimynų kiekį *k* šiai duomenų bazei. Atlikdami eksperimentus su *k*-NN klasifikatoriumi, koeficientą *k* keitėme nuo 1 iki 10. Eksperimentų rezultatai parodė, jog klasifikuojant diabeto duomenis, mažiausia vidutinė klaida buvo 36.9%, kai artimiausių kaimynų skaičius $k = 2$ (žr. 2.7 pav.)



2.7 pav. *k* artimiausiųjų kaimynų vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su stiklo duomenų baze

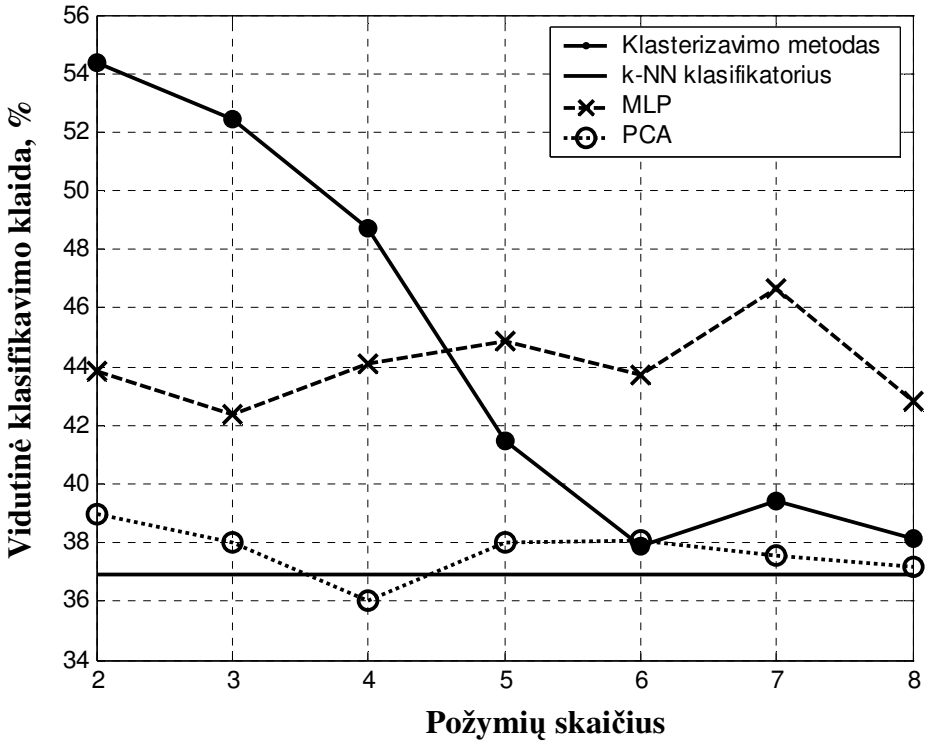
Atlikto tyrimo rezultatai su stiklo duomenų baze apibendrinti 2.6 lentelėje. Iš jos matome, kad šią duomenų bazę geriausiai klasifikavo neraiškus integralas su pilnuoju matu. Pagal rezultatus truputi išsiskyrė PCA metodas. Prasčiausiai šią duomenų bazę klasifikavo MLP tinklas.

Turėdami 9 požymius ir juos apjungę neraiškiu integralu su pilnuoju matu, nerekomenduojama mažinti požymių. Tačiau matome, kad PCA metodu gautas rezultatas yra daug stabilesnis ir turėtus 9 požymius galima sumažinti iki 4 požymių.

2.6 lentelė

Atrinktų požymių skaičius, vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai stiklo duomenų bazei, kai atlikta 20 eksperimentų

Duomenų bazė	Metodai, panaudoti požymių atrinkimui	Atrinktų požymių skaičius	Vidutinė klasifikavimo klaida %	Standartinis nuokrypis
Stiklo	*k*-NN klasifikatorius	9	36.9	3.97
	Klasterizavimo	6	37.9	4.06
	MLP	3	42.3	5.68
	PCA	4	36.0	3.66
	Neraiškus integralas su pilnuoju matu	8	34.9	4.55

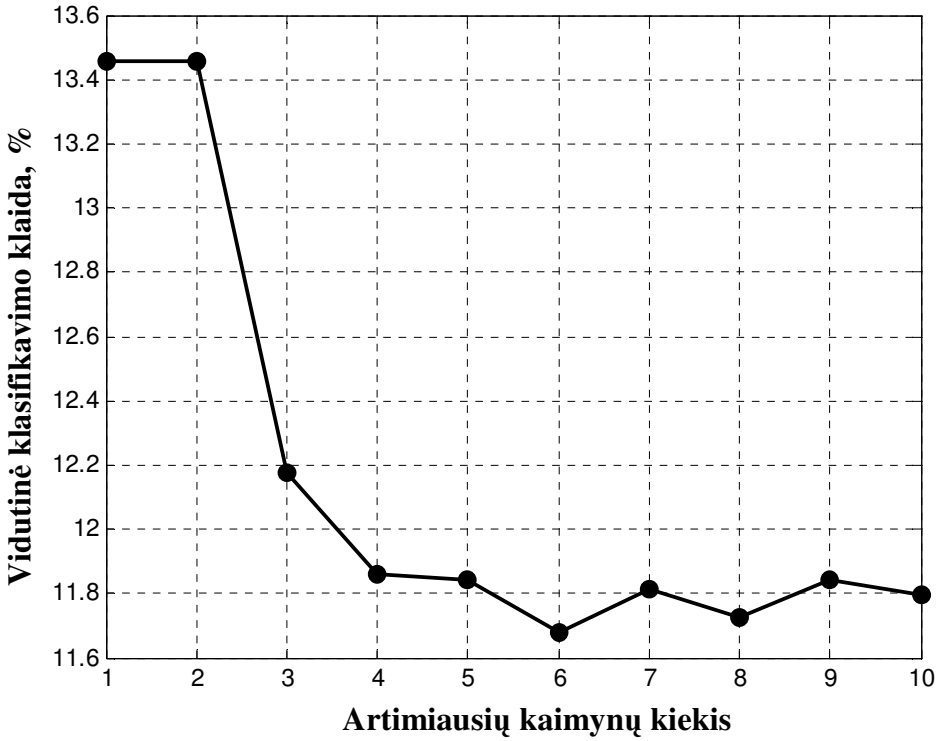


2.8 pav. Vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su stiklo duomenų baze

Pasinaudoję Stjudento (t) testu ir pritaikę (1) formulę (žr. 1 priede), gauname, kad statistika T yra mažesnė už t_α reikšmę, t. y. $0.097 < 1.684$. Hipotezė H_0 su reikšmingumo lygmeniu $\alpha = 0.05$ yra priimama. Tai yra su 5% klaida teigiame, kad klasifikavimo klaidos yra panašios, o klaidos sumažėjimas nežymus.

2.7. EKSPERIMENTAI SU PALYDOVINIŲ VAIZDŲ DUOMENŲ BAZE

Konstruojant *k*-NN klasifikatorių palydovinių vaizdų duomenims klasifikuoti, reikėjo nustatyti tinkamiausią artimiausių kaimynų kiekį *k* šiai duomenų bazei. Atlikdami eksperimentus su *k*-NN klasifikatoriumi, koeficientą *k* keitėme nuo 1 iki 10. Eksperimentų rezultatai parodė, jog klasifikuojant palydovinių vaizdų duomenis, mažiausia vidutinė klaida buvo 11.7%, kai artimiausių kaimynų skaičius $k = 6$ (žr. 2.9 pav.)



2.9 pav. *k* artimiausiųjų kaimynų vidutinės klasifikavimo klaidos priklausomybė nuo klasifikatoriaus struktūros su palydovinių vaizdų duomenų baze

Gauti tyrimo rezultatai su palydovinių vaizdų duomenų baze apibendrinti 2.7 lentelėje. Atlikus eksperimentą 20 kartų ir palyginus gautus klasifikavimo rezultatus palydovinių vaizdų duomenų bazei (požymius mažinant iš 5 į 3,...,4) su *k*-NN klasifikatoriumi ir visais požymiais, matome, kad šiai duomenų bazei tinkamiausias yra požymių atrinkimo metodas, paremtas klasterizavimu.

Prasčiausiai šią duomenų bazę klasifikavo daugiasluoksnio perceptrono (MLP) tinklas, tai greičiausia dėl to, kad šis tinklas ieško skiriamųjų paviršių tarp duomenų taškų. Tačiau šio klasifikatoriaus privalumas – mažiausia struktūra, o tai garantuoja didelį klasifikavimo greitį. Tuo tarpu principinių komponenčių metodas šiai duomenų bazei netinka, nes vidutinė klasifikavimo klaida stipriai skyrėsi nuo *k*-NN klasifikatoriaus gautos vidutinės klasifikavimo klaidos, o tai duoda blogą rezultatą.

Iš 2.6 lentelės matome, kad tiksliausiai iš jų klasifikavo klasterizavimo metodas.

2.7 lentelė

Atrinktų požymių skaičius, vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai palydovinių vaizdų duomenų bazei, kai atlikta 20 eksperimentų

Duomenų bazė	Metodai, panaudoti požymių atrinkimui	Atrinktų požymių skaičius	Vidutinė klasifikavimo klaida %	Standartinis nuokrypis
Palydovinių vaizdų	*k*-NN klasifikatorius	5	11.7	0.43
	Klasterizavimo	4	12.2	0.46
	MLP	4	14.1	1.06
	PCA	4	19.8	0.43
	Neraiškus integralas su pilnuoju matu	4	12.6	0.46

Pasinaudoję Stjudento (t) testu ir pritaikę (1) formulę (žr. 1 priede) *k*-NN ir klasterizavimo metodams, gauname, kad statistika T yra didesnė už t_α reikšmę, t. y. $1.834 > 1.684$. Hipotezė H_0 su reikšmingumo lygmeniu $\alpha = 0.05$ yra atmetama. Tai yra su 5% klaida teigiame, kad klasifikavimo klaidos yra nepanašios. Todėl nerekomenduojama mažinti požymių skaičių šiai duomenų bazei.

2.8. EKSPERIMENTŲ APIBENDRINIMAS

Eksperimentų metu su šiame darbe tirtais požymių erdvės mažinimo metodais gauti apibendrinti rezultatai, t.y. apibendrintos vidutinės klasifikavimo klaidos visoms naudotoms keturioms duomenų bazėms surašytos 2.8 lentelėje. Rezultatai pateikiami suranguoti balais, priskiriant balą = 4 geriausiam, t.y. mažiausią klasifikavimo klaidą pasiekusiam metodui, o didžiausią klaidą dariusiam metodui priskiriant mažiausią balą = 1.

2.8 lentelė

Gautų eksperimentų apibendrinimas

Metodai, panaudoti požymių atrinkimui	Balai vėžio duomenų bazei	Balai diabeto duomenų bazei	Balai stiklo duomenų bazei	Balai palydovinių vaizdų duomenų bazei	Vidutinis balas
Klasterizavimo	4	3	2	4	**3.25**
MLP	1	2	1	2	1.5
PCA	2	4	3	1	2.5
Neraiškus integralą su pilnuoju matu	3	1	4	3	2.75

Paskaičiavus vidutinį balą iš gautų rezultatų su keturiomis duomenų bazėmis, galime teigti, kad vidutiniškai mažiausią klasifikavimo klaidą pasiekia klasterizavimo metodas, nes jo vidutinis balas didžiausias. Tuo tarpu „blogiausią“ vidutinį balą surinko daugiasluoksnis neuroninis tinklas, nes jis dažniausiai pasiekdavo didžiausias klasifikavimo klaidas. Vidutiniškai didelės klasifikavimo klaidos su MLP tinklu pasiektos dėl to, kad „butelio kaklelio“ sluoksnyje duomenų požymiai transformuojami į duomenis turinčius mažiausiai požymių. Duomenys su tokia maža požymių aibe neužtikrina mažos klasifikavimo klaidos (per didelis informacijos praradimas).

PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI

Šiame darbe naudojausi MATLAB programų paketu, kurio aplinkoje buvo sukurtos požymių erdvės mažinimo metodų programos kiekvienai duomenų bazei ir su jomis atlikti eksperimentai. Duomenų imtys buvo atsitiktinai dalinamos į dvi lygias aibes: mokymo ir tikrinimo. Kiekvienas eksperimentas kartojamas po 20 kartų. Gautos vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai buvo paskaičiuoti iš tų 20 realizacijų.

Kiekvienai duomenų bazei (vėžio, diabeto, stiklo ir palydovinių vaizdų) sukurta keturios požymių erdvės mažinimo metodų programos (žr. 2.9 lentelę):

- Klasterizavimo metodui,
- Principinių komponenčių analizės metodui,
- „Butelio kaklelio“ metodui su neuroniniu tinklu,
- Neraiškaus integralo metodui su pilnuoju matu.

Kiekvienoje programoje užduodamas norimas eksperimentų skaičių (mano atveju, 20). Priklausomai nuo tiriamos duomenų bazės, keičiami klasių skaičiaus ir požymių kiekio parametrai.

Vykdant skaičiavimams programoje su neraiškiu integralu, vartotojui rodoma įvykių eigos juostelė (angl. „*wait bar*“) informuojanti kiek dar liko iki skaičiavimų pabaigos.

Kiekvienoje programų iteracijoje MATLAB lange „Command Windows“ išvedinėjamos klasifikavimo klaidos ir po visų realizacijų paskaičiuojamos vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai.

2.9 lentelė

Požymių erdvės mažinimo metodų programų failai

Vėžio (cancer) duomenų bazė	
Programų failai	Paskirtis
cancer_atr.m	Vykdomas PCA metodas ir daugiasluoksnis neuroninis tinklas
cancer_clust.m	Vykdomas klasterizavimo metodas
cancer_CI.m	Vykdoma programa su neraiškiu integralu, kai pasirinktas pilnasis neraiškus matas
Diabeto (diabetes) duomenų bazė	
diabetes_atr.m	Vykdomas PCA metodas ir daugiasluoksnis neuroninis tinklas
diabetes _clust.m	Vykdomas klasterizavimo metodas
diabetes _CI.m	Vykdoma programa su neraiškiu integralu, kai pasirinktas pilnasis neraiškus matas

2.9 lentelės tęsinys

Stiklo (glass) duomenų bazė	
glass_atr.m	Vykdomas PCA metodas ir daugiasluoksnis neuroninis tinklas
glass_clust.m	Vykdomas klasterizavimo metodas
glass_CI.m	Vykdoma programa su neraiškiu integralu, kai pasirinktas pilnasis neraiškus matas
Palydovinių vaizdų (satimage) duomenų bazė	
satimage_atr.m	Vykdomas PCA metodas ir daugiasluoksnis neuroninis tinklas
satimage _clust.m	Vykdomas klasterizavimo metodas
satimage _CI.m	Vykdoma programa su neraiškiu integralu, kai pasirinktas pilnasis neraiškus matas

DISKUSIJA

Šiame darbe buvo išnagrinėti keturi požymių erdvės mažinimo metodai, t.y. klasterizavimo metodas, požymių erdvės suspaudimas naudojant neuroninį tinklą, principinių komponenčių analizės metodas, bei neraiškus integralas su pilnuoju matu. Visų šių metodų kokybė lyginama naudojantis *k*-artimiausių kaimynų (*k*-NN) klasifikatoriumi. Metodų kokybė tirta atliekant eksperimentus su keturiomis laisvai prieinamomis duomenų bazėmis. Eksperimentams atlikti naudotas programinis paketas MATLAB.

Buvo skaičiuojamos tokios charakteristikos:

- vidutinės klasifikavimo klaidos,
- standartiniai nuokrypiai,.

Gautos vidutinės klasifikavimo klaidos keturioms duomenų bazėms (žr. 2.4, 2.5, 2.6 ir 2.7 lenteles) palygintos su kitų mokslininkų gautomis vidutinėmis klasifikavimo klaidomis ir standartiniais nuokrypiais (žr. 2.3 lentelę). Pastebėta, kad vidutinės klasifikavimo klaidos ir standartiniai nuokrypiai yra panašūs.

Taip pat kiekvienai duomenų bazei buvo tikrinama hipotezė apie vidutinių reikšmių lygybę, t.y. buvo lyginamos dvi skirtingos vidutinės klasifikavimo klaidos ir nuspręsta ar jos yra panašios, ar skirtingos, naudojant Stjudento (t) testą. Tam, kad tai patikrinti buvo skaičiuojama T statistika. Buvo pastebėta, kad:

- Vėžio, diabeto ir stiklo duomenų bazėms hipotezės (žr. 1 priedą) su reikšmingumo lygmeniu $\alpha = 0.05$ buvo priimamos. Tai yra su 5% klaida teigiame, kad klasifikavimo klaidos yra panašios, o klaidos sumažėjimas nežymus.
- Palydovinių vaizdų duomenų bazei hipotezė apie vidutinių reikšmių lygybę buvo atmetama. Tai yra su 5% klaida teigiame, kad klasifikavimo klaidos yra nepanašios. Todėl nerekomenduojama mažinti požymių skaičių šiai duomenų bazei.

Atlikus eksperimentus su kiekvienu iš keturių požymių erdvės mažinimo metodų, pastebėjome, kad vidutiniškai mažiausią klasifikavimo klaidą pasiekia klasterizavimo metodas, nes jo vidutinė klasifikavimo klaida mažiausia. Tuo tarpu didžiausias vidutines klasifikavimo klaidas pasiekdavo daugiasluoksnis neuroninis tinklas.

IŠVADOS

1. Šio darbo analitinėje dalyje išanalizuoti keturi požymių erdvės mažinimo metodai: klasterizavimo, principinių komponenčių analizės (PCA) metodai, daugiasluoksnis neuroninis tinklas (MLP) ir neraiškus integralas su neraiškiu pilnuoju matu.
2. Eksperimentinėje dalyje šie metodai lyginti pagal vidutines klasifikavimo klaidas, gautas su *k*-NN klasifikatoriumi suspaustoje erdvėje, naudojant keturias duomenų bazes. Suspaudimo laipsnis buvo laikomas geru, jei vidutine klasifikavimo klaida gauta su *k*-NN klasifikatoriumi yra mažesnė nei klaida gauta nesuspaustiems duomenims.
3. Atlikus eksperimentus su kiekvienu iš keturių požymių erdvės mažinimo metodų, gavome, kad vidutiniškai mažiausią klasifikavimo klaidą pasiekia klasterizavimo metodas, nes jo vidutinė klasifikavimo klaida mažiausia. Tuo tarpu didžiausias vidutines klasifikavimo klaidas pasiekdavo daugiasluoksnis neuroninis tinklas.
4. Eksperimentų metu neraiškaus integralo požymių atrinkimo metodas su pilnuoju matu pasiekdavo vidutiniškai geras klasifikavimo klaidas, tačiau turėjo mažiausią požymių suspaudimo laipsnį.
5. Pasinaudoję Stjudento (t) testu palydovinių vaizdų (satimage) duomenų bazei, gavome, kad hipotezė su reikšmingumo lygmeniu $\alpha = 0.05$ yra atmetama. Tai yra klasifikavimo klaidos yra nepanašios. Todėl šiai duomenų bazei nerekomenduojama mažinti požymių skaičių.

.

PADĖKOS

Magistrinio darbo vadovams: KTU Fundamentaliųjų mokslų fakulteto Taikomosios matematikos katedros doc. dr. R. Markauskui, KTU Mechatronikos mokslo, studijų ir informacijos centro doc. dr. A. Lipnickui ir KTU informatikos inžinerijos doktorantui V. Raudoniui už techninę ir intelektualią pagalbą.

LITERATŪRA

1. Driankov D., Palm Rainer**.** Advances in Fuzzy Control. 1998, pp. 263-283.
2. Cox, Earl. The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems. Academic Press, New York, 1994.
3. Passino K.M., Yurkovich S**.** Fuzzy Control. Addison Wesley, 1998.
4. Lee K-H. Fuzzy Theory. Textbook. Prieiga per internetą: <http://if.kaist.ac.kr/lecture/cs670/2001/index.html>
5. Bishop C.M. Neural Networks for Pattern Recognition. Clarendon Press, Oxford, 1996, pp. 310-313.
6. Pham T.D., and Yan H. Color image segmentation using fuzzy integral and mountain clustering, Fuzzy Sets and Systems 107. 1999, pp. 121-130.
7. Tahani H., Keller J.M. Information fusion in computer vision using the fuzzy integral. IEEE Trans Systems, Man and Cybernetics 20(3). 1990, pp. 733-741.
8. Dubois D., and Prade H. Fuzzy sets and Systems: Theory and Applications, Academic press. Inc., 1980.
9. Murofushi T. and Soneda S. Techniques for reading fuzzy measures: interaction index. In: Proceedings of the 9th Fuzzy Systems Symposium, Sapporo, Japan, 1993, pp. 693-696 (in Japanese).
10. Hocaoglu A. K., Gader P.D. Choquet integral representations of nonlinear filters with applications to LADAR image processing. In: Proceedings of the SPIE conference "Nonlinear image processing IX", San Jose CA, 1998, pp. 66-72.
11. Yager R.R. On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Systems, Man and Cybernetics 18, 1988, pp.183-190.
12. Grabisch M. *k*-order additive discrete fuzzy measures and their representation. Fuzzy Sets and Systems 92, 1997, pp. 167-189.
13. Grabisch M. k-order additive discrete fuzzy measures and their representation. Fuzzy Sets and Systems 92, 1997, pp. 167-189.
14. Grabisch M. and Sugeno M. Multi-attribute classification using fuzzy integral, In: Proceedings of the First IEEE International Conference on Fuzzy Systems, San Diego, 1992, pp. 47-54.
15. Chateauneuf A., Jaffray J.Y. Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion. Math. Social Sci. 17, 1989, pp. 263-283.

16. Mikenina L., Zimmermann H.-J., Improved feature selection and classification by the 2-additive fuzzy measure. Fuzzy Sets and Systems 107, 1999, pp. 197-218.

17. Shapley L.S. A value for n-person games, In: Kuhn, H.W., Tucker, A.W. Contributions to the Theory of Games, Vol. 2(28), In: Annals of Mathematics Studies, Princeton University Press, Princeton, 1953, pp. 307-317.

18. Grabisch M. The representation of importance and interaction of features by fuzzy measures. Pattern Recognition Letters 17. 1996, pp. 567-575.

19. Jollife I.T. Principal Component Analysis (Springer series in statistics). Springer-Verlag New York Inc., 1986.

20. Bishop C.M. Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1996.

21. Bouldand H. and Kamp Y. 1988, Auto-association by multi-layer perceptrons and singular value decomposition. Biological Cybernetics, 1988, pp. 291-294.

22. Kramer M. Nonlinear principal component analysis using autoassociative neural networks, Am. Ins. Chem. Eng. J. 37 (2), 1991, pp. 223-243.

23. Oja E. Data compression, feature extraction, and autoassociation in feed-forward neural networks. In O. Simula, T. Kohonen, K. Makisara, and J. Kangas, editors, International Conference on Artificial Neural Networks, Helsinki, Finland. Elsevier, Amsterdam, 1991, pp. 737-745.

24. DeMers D., Cottrell G.W. Non-linear dimensionality reduction. In C.L. Giles, S.J. Hanson, and J.D. Cowan, editors, Advances in Neural Information Processing Systems 5, pages 580-587. Morgan Kaufmann, San Mateo, CA. 1993. Prieiga per internetą: <ftp://ftp.cs.ucsd.edu/pub/guest/demers/demers.nips92-nldr.ps.Z>.

25. Basso A. Neural network applications. Data compression. Handbook of Neural Computation, IOP publishing Ltd and Oxford University Press, 1997.

26. Lampinen J., Laaksonen J., Oja E. Pattern Recognition. In Image processing and pattern recognition Vol. 5, edited by C.T. Leondes, Academic press, 1998.

27. Teeuwsen S.P., Erlich I., and El-Sharkawi M.A. Feature reduction for neural network, 2002.

28. Duda R.O., Hart P.E. Classification and Scene Analysis. Wiley, New York, 1973.

29. ESPRIT, ELENA Basic Research Project Number 6891. – [žiūrėta 2007-03-07]. Prieiga per internetą: <http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm>.

30. Index of /~pbialas/proben1. – [žiūrėta 2007-03-07]. Prieiga per internetą: <http://kiwi.if.uj.edu.pl/~pbialas/proben1>.

31. A. Lipnickas, C.Bocãnialã, J. Costa. Performance analysis for a new Fuzzy β-NN classifier. „Informacinės technologijos ‘2004“, Kauno Technologijos Universitetas, Kaunas 2004, sausio 28-29d.

32. Sexton R. S., Dorsey R. E., Reliable Classification Using Neural Networks: A Genetic Algorithm and Backpropagation Comparison, pp. 1-26. Prieiga per internetą: <http://www.faculty.missouristate.edu/R/RandallSexton/clasification.pdf>.

33. Loo C.K., Rao M.V.C., Accurate and Reliable Diagnosis and Classification Using Probabilistic Ensemble Simplified Fuzzy ARTMAP, IEEE Transactions on knowledge and data engineering, Vol. 17, No 11, November 2005, pp. 1589-1593. Prieiga per internetą: <http://ieeexplore.ieee.org/iel5/69/32375/01512043.pdf>.

34. Zhang G.P. Neural Networks in Business Forecasting. Georgia State University, USA, 2003.

1 PRIEDAS. DVIEJŲ SKIRTINGŲ VIDUTINIŲ KLASIFIKAVIMO KLAIDŲ PALYGINIMAS

Šiame darbe norėdami palyginti dvi skirtingas vidutinių klasifikavimo klaidas ir nuspręsti ar jos yra panašios, ar skirtingos, naudojome Stjudento (t) testą (angl. *„t-test“*). Tikrinome nulinė hipotezę $H_0: \mu_1 = \mu_2$. Nustatant ar dvi vidutinės klasifikavimo klaidos μ_1 ir μ_2 labai skiriasi, skaičiavome *T* statistiką:

$$T = \frac{\mu_1 - \mu_2}{\sqrt{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}} \cdot \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}}, \qquad \textbf{(1)}$$

kur n_1 ir n_2 yra eksperimentų skaičius (mūsų atveju, $n_1 = n_2 = 20$), μ_1 ir σ_1 yra *k*-NN metodu gauta vidutinė klasifikavimo klaida ir standartinis nuokrypis, μ_2 ir σ_2 - kitu metodu gauta vidutinė klasifikavimo klaida ir standartinis nuokrypis.

Reikšmės t_α ir $r = n_1 + n_2 - 2$ apibrėžtos 1 lentelėje. Jeigu $|T| > t_\alpha$, tai nulinė hipotezė H_0 atmetama ir tai reiškia, kad dvi vidutinės klasifikavimo klaidos yra nepanašios. Tikimybė, kad Stjudento (t) testo rezultatas yra blogas, priklauso nuo t_α reikšmės.

1 lentelė

r	**75%**	**80%**	**85%**	**90%**	**95%**	**97.5%**	**99%**	**99.5%**	**99.75%**	**99.9%**	**99.95%**
1	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	127.3	318.3	636.6
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	3.581	4.144	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	3.497	4.025	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.428	3.930	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.372	3.852	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.326	3.787	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.286	3.733	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.252	3.686	4.015

17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.222	3.646	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.197	3.610	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.174	3.579	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.153	3.552	3.850
21	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.135	3.527	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
23	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.104	3.485	3.767
24	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.091	3.467	3.745
25	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.078	3.450	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.067	3.435	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.057	3.421	3.690
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.047	3.408	3.674
29	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.038	3.396	3.659
30	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.030	3.385	3.646
40	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	2.971	3.307	3.551
50	0.679	0.849	1.047	1.299	1.676	2.009	2.403	2.678	2.937	3.261	3.496
60	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	2.915	3.232	3.460
80	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	2.887	3.195	3.416
100	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	2.871	3.174	3.390
120	0.677	0.845	1.041	1.289	1.658	1.980	2.358	2.617	2.860	3.160	3.373
∞	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291

2 PRIEDAS. NERAIŠKAUS INTEGRALO SU PILNUOJU MATU METODO PROGRAMA

```
clear % Viskas išvaloma
close all
warning off
global REZULTATAS exnum
exnum=1;
rand('state',sum(100*clock));

out=2; % Klasių skaičius

EXPNUM=20; % Eksperimentų skaičius

%---------------Pagrindinio ciklo pradžia---------------
for eksp=1:EXPNUM

load cancer.mat; % Užkraunama duomenų bazė - cancer (vėžio) duomenų bazė

[m,n]=size(data);
nin=n-1;
Ndata=m;
Ntest=Ndata;
lpart=2;


%---------------Duomenų apmokymas ir testavimas---------------
dr=ones(Ndata,out)*.1;

tem=zeros(1,Ndata); %temporary array to mark used data
k=0;
learn_ind=[];
while k~=round(Ndata/lpart),
tk=ceil(rand*Ndata); if(tk==0) tk=1; end;
if(tem(tk)==0) k=k+1; learn_ind=[learn_ind tk]; tem(tk)=1; end;
end;

test_ind=find(tem==0);


%---------------Duomenų ir klasių matricų formavimas---------------
for i=1:Ndata  dr(i,1+data(i,n))=0.9; end;
```

```
T=1+data(:,n)';
data=data(:,1:n-1);
ldata=[];
ldr=[];
iT=[];


%---------------Normalizuojami duomenys---------------
midata=abs(min(data));
madata=midata+max(data);
for i=1:nin CIdata(:,i)=(data(:,i)+midata(i))/madata(i); end;
CIldata=CIdata(learn_ind,:)';

ldata=data(learn_ind,:)';
ldr=dr(learn_ind,:)';
iT=T(learn_ind);

for kk=1:10, c = knn(data(test_ind,:),ldata',iT',kk);
   error(eksp,kk)=mist(c,T(test_ind));
end
[knn_vid,K_num]=min(mean(-error(:,1:end)));
K_num=K_num;
knn_std=std(-error(:,K_num));



%---------------Neraiškus integralas---------------
A=zeros(9,out,round(Ndata/lpart));
for netj=1:out
A(:,netj,:)=CIldata;
end;


[Ds,Gf,Af,Bf,ER_qf,Visi_f,SSz]=ch_feat_aint(A,iT);

load SSz9

[u_full]=loqo(Ds,0.0175*Gf,-Af,Bf,zeros(length(Ds),1),ones(length(Ds),1),[],[],2);
u_full';
clf
 plot(u_full(1:510),'.')
 hold on
```

```
 plot(u_full(511:1020),'r.')

poz=u_full(1:510)+u_full(511:1020); % Sudaromos visos galimos tankių kombinacijos

clf
hold on

bar(poz) % Nubraižoma sudarytų tankių kombinacijų priklausomybė nuo vidutinių tankių reikšmių


[qwqw,gera_komb]=max(poz);
geri_poz=SSz(gera_komb,:)

geri_poz1(eksp,:)=geri_poz;



for kk=1:10,
 c = knn(data(test_ind,find(geri_poz>0)),data(learn_ind,find(geri_poz>0)),iT',kk);
       error_CI(eksp,kk)=mist(c,T(test_ind))
end

end % Pagrindinio ciklo pabaiga


[knn_CI_vid,K_num]=min(mean(-error_CI(:,1:end)));
K_num=K_num;
knn_CI_std=std(-error_CI(:,K_num));


%----------Išvedami rezultatai----------
geri_poz1
knn_CI_vid % Išvedama vidutinė klasifikavimo klaida, gauta eksperimentus atlikus su neraiškiu integralu
knn_CI_std % Išvedamas standartinis nuokrypis, gautas eksperimentus atlikus su neraiškiu integralu

% k-NN metodu gauti rezultatai
knn_vid
knn_std
```

3 PRIEDAS. POŽYMIŲ ERDVĖS MAŽINIMO METODŲ PROGRAMA

```
clear % Viskas išvaloma
close all
warning off
global REZULTATAS exnum
exnum=1;
rand('state',sum(100*clock));

out=2; % Klasių skaičius

EXPNUM=20; % Eksperimentų skaičius

%----------Pagrindinio ciklo pradzia----------
for eksp=1:EXPNUM

load cancer.mat; % Užkraunama duomenų bazė - cancer (vėžio) duomenų bazė


[m,n]=size(data);
nin=n-1;
Ndata=m;
Ntest=Ndata;
lpart=2;

%----------Duomenų apmokymas ir testavimas----------
dr=ones(Ndata,out)*.1;

tem=zeros(1,Ndata); % temporary array to mark used data
k=0;
learn_ind=[];
while k~=round(Ndata/lpart),
tk=ceil(rand*Ndata); if(tk==0) tk=1; end;
if(tem(tk)==0) k=k+1; learn_ind=[learn_ind tk]; tem(tk)=1; end;
end;

test_ind=find(tem==0);

%----------Duomenų ir klasių matricų formavimas----------
for i=1:Ndata  dr(i,1+data(i,n))=0.9; end;
T=data(:,n)';
```

```
data=data(:,1:n-1);
ldata=[];
ldr=[];
iT=[];

%----------Normalizuojami duomenys----------
ldata=data(learn_ind,:)';
ldr=dr(learn_ind,:)';
iT=T(learn_ind);

[data_n,meanp,stdp] = prestd(data');
[ptrans,transMat] = prepca(data_n,0);


for kk=1:10, c = knn(data(test_ind,:),ldata',iT',kk);
   error(eksp,kk)=mist(c,T(test_ind));
end
[knn_vid,K_num]=min(mean(-error(:,1:end)));
K_num=K_num;
knn_std=std(-error(:,K_num));

%----------k-means metodas----------
for poz=2:n-2
   [IDX, C] = KMEANS(data', poz);
   for kk=1:10, c = knn(C(:,test_ind)',C(:,learn_ind)',iT',kk);
      error_CLUST(poz-1,kk,eksp)=mist(c,T(test_ind))
   end
end



%----------MLP neuroninio tinklo metodas----------
for poz=2:n-2
   hid=1;
net=[];
w1=[];
w2=[];
b1=[];
b2=[];
netb=[];
rezu=[];
net2=[];
```

```
%----------Tinklo struktūros kūrimas----------
   netb=newff(0.3*minmax(ldata),[hid+poz,poz,hid+poz,n-1],{'purelin','purelin', 'purelin','purelin'},'trainlm');
   net=netb;
   rezu    =sim(netb,ldata);
   kl=sse(rezu-ldata);
   for kari=1:20
      netb=newff(0.3*minmax(ldata),[hid+poz,poz,hid+poz,n-1],{'purelin','purelin', 'purelin','purelin'},'trainlm');
      rezu    =sim(netb,ldata);
      if kl>sse(rezu-ldata)
         kl=sse(rezu-ldata);
         net=netb;
      end
   end

   net.trainParam.show=10;
   net.trainParam.epochs=40;
   net.trainParam.goal=0.000001;
   net=train(net,ldata,ldata); % Tinklo mokymas
   eval('w1= net.IW{1,1};');
   eval('w2= net.LW{2,1};');
   eval('b1= net.b{1};');
   eval('b2= net.b{2};');
   net2=newff(0.3*minmax(ldata),[hid+poz,poz],{'purelin','purelin'},'trainbr');
   net2.IW{1,1}=(eval('w1' ));
   net2.LW{2,1}=(eval('w2' ));
   net2.b{1}   =(eval('b1' ));
   net2.b{2}   =(eval('b2' ));

   NNT_C=sim(net2,data');

   for kk=1:10, c = knn(NNT_C(:,test_ind)',NNT_C(:,learn_ind)',iT',kk);
      error_NNT(poz-1,kk,eksp)=mist(c,T(test_ind))
   end
end

%----------MLP neuroninio tinklo metodas----------
for poz=2:n-2
   [C] = ptrans(1:poz,:);
   for kk=1:10, c = knn(C(:,test_ind)',C(:,learn_ind)',iT',kk);
      error_PCA(poz-1,kk,eksp)=mist(c,T(test_ind))
   end
end
```

```
end % Pagrindinio ciklo pabaiga


for poz=2:n-2
   for kk=1:10
      e_clust(poz-1,kk)= mean(error_CLUST(poz-1,kk,:));
      std_clust(poz-1,kk)= std(error_CLUST(poz-1,kk,:));
   end
end

for poz=2:n-2
[knn_clust_vid(poz-1),K_num]=min(-e_clust(poz-1,1:end));

K_num=K_num+0;
knn_clust_std(poz-1)=std_clust(poz-1,K_num);
end

%---------------Išvedami rezultatai---------------
knn_clust_vid
knn_clust_std



for poz=2:n-2
   for kk=1:10
      e_nnt(poz-1,kk)= mean(error_NNT(poz-1,kk,:));
      std_nnt(poz-1,kk)= std(error_NNT(poz-1,kk,:));
   end
end

for poz=2:n-2
[knn_nnt_vid(poz-1),K_num]=min(-e_nnt(poz-1,1:end));

K_num=K_num+0;
knn_nnt_std(poz-1)=std_nnt(poz-1,K_num);
end

%---------------Išvedami rezultatai---------------
knn_nnt_vid
knn_nnt_std
```

```
for poz=2:n-2
   for kk=1:10
      e_pca(poz-1,kk)= mean(error_PCA(poz-1,kk,:));
      std_pca(poz-1,kk)= std(error_PCA(poz-1,kk,:));
   end
end

for poz=2:n-2
[knn_pca_vid(poz-1),K_num]=min(-e_pca(poz-1,1:end));

K_num=K_num+0;
knn_pca_std(poz-1)=std_pca(poz-1,K_num);
end

%----------Išvedami rezultatai----------
knn_pca_vid
knn_pca_std

%----------Išvedami rezultatai----------
knn_vid
knn_std

clf

plot(mean(-error(:,1:end)))
```