

**RĪGAS TEHNISKĀ UNIVERSITĀTE**  
Datorzinātnes un informācijas tehnoloģijas fakultāte  
Informācijas tehnoloģijas institūts

**Pēteris GRABUSTS**

**NOSACĪTO LIKUMU VEIDOŠANAS METOŽU  
IZPĒTE UN IZSTRĀDĀŠANA  
UZ DAUDZDIMENSIJU DATU IZLASES PAMATA**

**Promocijas darba kopsavilkums**

Zinātniskais vadītājs  
Dr.habil.sc.comp., profesors  
A. BORISOVS

Rīga 2006

## VISPĀRĒJS DARBA RAKSTUROJUMS

Intelektuālās analīzes datortehnoloģijas patlaban piedzīvo uzplaukuma periodu. Tas galvenokārt saistīts ar jaunu ideju realizēšanos vairāku zinātņu nozaru saskarsmes punktos tādos kā mākslīgais intelekts, statistika, datu bāzu metodes u.c. Datu analīzes un automatiskās apstrādes metodes tiek traktētas kā zināšanu ieguve. Daudzos gadījumos ir nepieciešams kaut kādā veidā klasificēt datus vai atrast likumsakarības tajos, tāpēc jēdziens "likumsakarība" iegūst arvien lielāku nozīmi intelektuālās datu apstrādes sistēmu kontekstā. Bieži ir nepieciešams noskaidrot - kādā veidā dati ir saistīti savā starpā, kāda ir dažādu datu līdzība vai atšķirība, kāds ir šo datu salīdzināšanas mērs. Intelektuālo datu analīzi var definēt arī šādi: tas ir lēmumu pieņemšanas process, kas balstās uz slēpto likumsakarību meklēšanu datos. Lielākā daļa šāda veida metožu sākotnēji tika izstrādāta pagājušā gadsimta 70.-80. gados mākslīgā intelekta metožu pētīšanas rezultātā, taču plašu ievēriību ieguva tikai mūsdienās sakarā ar strauji pieaugošu datu apstrādes nepieciešamību.

Tēmas aktualitāte ir saistīta ar to, ka likumsakarību meklēšanas rezultāti intelektuālās datu analīzes pielietojumos pārsvarā tiek izteikti ar nosacīto „JA-TAD” (*IF-THEN*) likumu palīdzību. Ar šādiem likumiem tiek risināti prognozēšanas, klasifikācijas, tēlu atpazīšanas un citi uzdevumi. Izmantojot dažādas pieejas - klasterizācijas algoritmus, neironu tīklu metodes, izplūdušo likumu apstrādes metodes - var iegūt likumus, kas cilvēkam saprotamā valodā raksturo datus. Tas ļauj interpretēt datus, atrast likumsakarības tajos un iegūt no datiem jaunus tos raksturojošos likumus.

Promocijas darba galvenais mērķis - izpētīt uzdevumu klases, kas realizē likumsakarību iegūšanas un nosacīto likumu veidošanas metodes, un paaugstināt intelektuālo datu apstrādes sistēmu efektivitāti, lai iegūtu likumsakarības vai tās raksturojošos likumus no daudzdimensiju datu izlasēm.

Saskaņā ar mērķi tika noteikti šādi uzdevumi:

- 1) novērtēt pašreizējo stāvokli mākslīgo neironu tīklu pielietošanas jomā likumsakarību iegūšanai;
- 2) izpētīt un salīdzināt klasiskās klasterizācijas algoritmu pielietošanas iespējas likumsakarību modelēšanā;
- 3) izstrādāt izplūdušo likumu realizācijas modeli likumsakarību meklēšanai ar izplūdušās klasterizācijas metožu palīdzību;
- 4) adaptēt arhitektūras, metodes un algoritmus, kas ļautu izmantot atsevišķas neironu tīklu klases likumu iegūšanai un apstrādei;
- 5) veikt datu analīzi ar asociāciju likumu iegūšanas metodēm.

Pētījuma objekts ir dažāda veida likumsakarības datos, kuras var izteikt ar nosacīto likumu palīdzību.

**Pētījuma priekšmets** - datu analīzes metodoloģijas, ar kuru palīdzību tiek iegūtas šādas likumsakarības un/vai likumi.

Darba mērķa sasniegšanai veiktie pētījumi balstās uz vairākām metodēm:

- 1) mākslīgo neironu tīklu izmantošanu datu analīzes uzdevumos;
- 2) klasterizācijas metožu izmantošanu klasifikācijas uzdevumos un to pielietojumu neironu tīklos;
- 3) izplūdušās klasterizācijas pielietojumu izplūdušo likumu apstrādē;
- 4) asociāciju likumu iegūšanas algoritmu izmantošanu

Pētāmo uzdevumu risināšanai tiek izmantota mākslīgo neironu tīklu teorija, klasterizācijas metožu algoritmi, izplūdušo kopu teorijas un asociāciju likumu teorijas elementi. Metožu pielietojumi ir ilustrēti ar demonstrācijas un praktiskiem piemēriem, kas raksturo pētāmo uzdevumu klasi.

Pieaugot zinātniskajai un praktiskajai interesei intelektuālās datu analīzes jomā, ievēriību ieguvuši pētījumi par iespēju apstrādāt „IF-THEN” likumus ar dažādu metožu palīdzību. Tādējādi tiek iegūts zināšanu bāzes modelis, kas var kalpot par pamatu turpmākajai datu analīzei. Darba

izstrādāšanas laikā tika veikti pētījumi un analizētas vairākas metodes un paņēmieni, ar kuru palīdzību var iegūt datus raksturojošās likumsakarības.

Darba rezultāta jaunieguvums ir apkopotā informācija ar optimālākās datu analīzes metodes izvēli, kas pamatota ar eksperimentāliem pētījumiem. Tās ļauj izvēlēties vienu no trim apskatāmajām likumu iegūšanas metodēm.

Darbā aprakstīta ideja par potenciālu funkciju metodes izmantošanu atsevišķas uzdevumu klases mērķiem.

Darba ietvaros ir izstrādāta izplūdušo likumu bāzes iegūšanas metode no skaitliskajiem datiem ar izplūdušās klasterizācijas algoritma pielietošanu.

**Praktiskā vērtība** ir metodikas un programmatūras izstrādāšana, kas ļauj pielietot datu analīzes metodes likumsakarību apstrādē. Darba ietvaros tika izstrādāts programmnodrošinājums *Matlab* vidē, ar kura palīdzību tika veikti eksperimentālās daļas pētījumi dažādu datu analīzes metožu darbības ilustrēšanai (programmu moduļi un izmantojamie dati ir pieejami autora mājas lapā [www.ru.lv/~peter](http://www.ru.lv/~peter) sadaļā *Zinātniskais darbs*).

Darba gaitā tika veikti pētījumi un izstrādātas metodes:

- 1) izpētīta mākslīgo neironu tīklu pielietošanas lietderība likumsakarību meklēšanā uz neironu tīklu pamata,
- 2) izanalizēta likumu iegūšanas procedūra no radiālās bāzes funkciju neironu tīkla;
- 3) pamatota klasterizācijas metožu izvēle likumsakarību izpētē, salīdzinot un analizējot vairākas klasterizācijas metodes to efektīvākai darbībai;
- 4) izstrādāts un realizēts izplūdušās klasterizācijas modelis likumu iegūšanai un apstrādei;
- 5) izstrādāti principi un metodoloģija likumu iegūšanai ar asociāciju likumu metodes palīdzību.

Par darba galvenajiem rezultātiem tika ziņots šādās nozīmīgākajās starptautiskajās zinātniskajās konferencēs:

1. *Grabusts P. Extracting rules from trained RBF neural networks// Proc. of the 5th International Conference. - Rezekne: RA izdevniecība, 2005. -p. 13-39.*
2. *Grabusts P. Analysing Bankruptcy Data with Neural Networks//: Proc. 10th International Conference on Soft Computing. - Brno, Czech Republic, 2004. - p: 111-117.*
3. *Grabusts P. Using Association Rules to Extract Regularities from Data// Proc. 6" International Baltic Conference on Data Bases and Information-Systems. -Rīga, 2004. - p. 117-126.*
4. *Grabust P., Borisov A. Using grid-clustering methods in data classification// Proceedings of the International Conference on Parallel Computing & iElectrical Engineering-PARELEC'2002. - Warsaw, Poland, 2002. -p. 425-426.*
5. *Grabusts P., Borisovs A. RBF neironu tīklu pielietošanas perspektīvas// II. Pasaules latviešu zinātnieku kongresa tēžu krājums. - Rīga, 2001. - 570. lpp.*
6. *Grabusts P. Klasterizācijas metodes izmantošana RBF neironu tīklos// III Starptautiskās zinātniskās konferences „ Vide. Tehnoloģija. Resursi" materiāli. -Rēzekne: RA izdevniecība, 2001. - 257.-262. lpp.*
7. *Grabust P. Using a thermal equilibrium method in the neural networks// International Conference on Parallel Computing & Electrical Engineering-PARELEC '98, - Bialystok, Poland, 1998. -p. 261-263.*

Darbu rezultāti ir iekļauti Latvijas Zinātņu padomes granta 2002.gada atskaitē (LZP doktorantūras komisijas 13.05.2002. lēmums Nr. 3-2-1 par doktorantūras granta piešķiršanu).

Par promocijas darba tēmu ir 4 publikācijas, 9 raksti zinātnisko konferenču rakstu krājumos.

Promocijas darba struktūra - darbs sastāv no ievada, piecām nodaļām, secinājumiem, izmantotās literatūras saraksta un pielikuma. Darba pamatteksts izklāstīts 120 lappusēs un paskaidrots ar 91 attēlu un 41 tabulu. Literatūras sarakstā iekļauti 68 nosaukumi.

Ievadā ir pamatota veikto pētījumu aktualitāte, noformulēts galvenais pētījumu mērķis un uzdevumi, kā arī sniegts īss pētījumu pamatvirzienu raksturojums.

Pirmajā nodaļā ir dots datu analīzes veidu un mērķu apskats likumsakarību meklēšanas kontekstā.

Otrajā nodaļā ir aprakstīta asociāciju analīzes izmantošana likumsakarību iegūšanas procesā un eksperimentu sērija statistisko datu apstrādei.

Trešajā nodaļā ir izpētītas mākslīgo neironu tīklu iespējas datu analīzes veikšanā; aprakstīti bankrotu datu analīzes eksperimentu rezultāti un piedāvāta potenciālu funkciju metodes izmantošana datu analīzē.

Ceturtajā nodaļā ir izvērtētas vairāku klasterizācijas metožu iespējas, izstrādāta metodoloģija likumu bāzes iegūšanai ar izplūdušās klasterizācijas palīdzību un aprakstīti eksperimenti likumu iegūšanai no daudzdimensiju datu izlasēm.

Piektajā nodaļā ir apskatītas neironu tīklu metodes nosacīto likumu iegūšanā un atspoguļoti eksperimentu rezultāti - iegūtie likumi no bankrotu datu izlases.

Secinājumos dots kopsavilkums par paveikto darbu un formulēti tālāko pētījumu virzieni.

Pielikumā ir demonstrēts autora izstrādātais programmnodrošinājums asociāciju likumu iegūšanai no statistiskajiem datiem.

## DARBA SATURA IZKLĀSTS

Ievada ir pamatota paveikto pētījumu aktualitāte, noformulēts galvenais pētījumu mērķis un sniegts īss pētījumu pamatvirzienu raksturojums, kā arī promocijas darba nodaļu anotācija.

**Pirmajā nodaļā** ir dots datu analīzes veidu un mērķu apskats likumsakarību meklēšanas kontekstā. Jēdziens „likumsakarība” iegūst arvien lielāku nozīmi intelektuālās datu apstrādes sistēmu darbības atspoguļošanā. Daudzos gadījumos ir nepieciešams noskaidrot - kādā veidā dati ir saistīti savā starpā un kāda ir dažādu datu līdzība vai atšķirība, kāds ir šo datu salīdzināšanas mērs. Dažādas vārdnīcas terminu likumsakarība” skaidro kā līdzību, likumību vai secinājumus pēc analogijas. Var teikt, ka likumsakarība ir pareizi noteikta, ja tā izskaidro visu eksperimentu rezultātus, kas attiecas uz doto darbības jomu (jo vairāk eksperimentālo faktu apstiprinās, jo drošāks pamatojums). Tā kā likumsakarību meklēšanas mērķis datos ir intelektuālās datu analīzes uzdevums, tad nepieciešams aplūkot mūsdienu nostādnes šajā jomā. Patlaban ir izveidojušās vairākas datu uzglabāšanas un analīzes koncepcijas:

- datu glabātuves;
- operatīvi analītiskā datu apstrāde;
- intelektuālā datu analīze.

Intelektuālās datu analīzes (IDA) joma tiek uzskatīta par ļoti būtisku sakarā ar to, ka uzkrājamie dati tiek definēti kā *zināšanas*. Angļu valodā ir termini, kas tiek traktēti kā IDA - "Knowledge Discovery in Databases" (KDD) un "Data Mining" (DM). Daudzi pētnieki tos lieto kā sinonīmus, taču vairākums KDD traktē plašākā nozīmē - kā zinātnes virzienu, kas satur sevī mākslīgā intelekta metodes, statistikas un datu bāzu metožu apvienojumu, kas ļauj iegūt informāciju no datiem; savukārt DM tiek traktēts kā šīs informācijas ieguves process. IDA var definēt tādējādi - *tas ir lēmumu pieņemšanas process, kas balstās uz slēpto likumsakarību meklēšanu datos*. Jāatzīmē, ka lielākā daļa IDA metožu sākotnēji tika izstrādātas pagājušā gadsimta 70-80. gados mākslīgā intelekta metožu pētīšanas rezultātā, taču plašu ievēriību ieguva patlaban sakarā ar strauji pieaugušo datu apstrādes nepieciešamību.

Lielākā daļa pētnieku veic IDA uzdevumu klasifikāciju pēc iegūstamās informācijas tipa. Tiek izdalīti 5 vispārīgu uzdevumu veidi.

1. Klasifikācija Visvairāk izplatītākais IDA uzdevums. Tā ļauj izdalīt pazīmes, kas raksturo viena tipa objektu grupas (klases), lai pēc zināmiem raksturlielumiem varētu jaunu objektu iekļaut tajā vai citā klasē.

2. Klasterizācija. Loģiski turpina klasifikācijas ideju sarežģītākā gadījumā - kad klases nav noteiktas. Klasterizācijas rezultātā tiek arī noteikts objektu sadalījums klasēs. Vispārīgā gadījumā klasterizācijas rezultāti ir ļoti subjektīvi. Tas ir atkarīgs no izvēlētās apmācošās kopas "tuvības" mēra. Klasterizācijas algoritmi tiek pielietoti vairākos mākslīgo neironu tīklu modeļos un izplūdušu tīklu modeļos datu sagatavošanai apmācībā „bez skolotāja”.
3. Asociāciju iegūšana. Asociāciju likumi ļauj atrast likumsakarības starp vairākiem saistītiem notikumiem. Šādu likumu pamatā ir apgalvojums - ja ir izpildījies notikums A, tad ar varbūtību X% būs spēkā arī notikums B. Sākotnēji šis uzdevums tika izmantots lielveikalos tipisku iepirkumu šablonu atrašanai, tāpēc to dažkārt sauc par "iepirkumu groza analīzi”.
4. Secību (virknes) iegūšana. Līdzīga asociācijām, taču secība ir spēkā starp notikumiem, kas nesākas vienlaicīgi, bet ar noteiktu laika nobīdi. Ja ir spēkā notikums A, tad ar varbūtību X% pēc laika T izpildīsies notikums B.
5. Prognozēšana. Esošo faktisko vērtību izmantošana nākamo vērtību prognozēšanai

Pētījumi, kas tiek veikti intelektuālās datu analīzes jomā, satur ļoti daudz dažādu metožu no vairākām zinātņu jomām. Tiek uzsvērts, ka IDA sevī ietver daudzu zinātnes nozaru saskarsmes punktus. Biežāk minētās IDA metodes ir šādas:

- lēmumu analīze un regresijas koki;
- Beijesa klasifikators;
- varbūtiskie tīkli;
- mākslīgie neironu tīkli;
- izplūdušā secināšana;
- asociāciju likumi;
- klasterizācija;
- izplūdušās klasterizācijas analīze u.c.

Viens no mītiem par intelektuālās datu analīzes metožu visspēcību ir tas, ka IDA automātiski atrod likumsakarības dotos. Jā, tiešām, daudzas IDA metodes ļauj atrast likumsakarības pētāmajos datos, taču pētīšanas procesā jābūt skaidri noteiktam mērķim- kas ir jāiegūst galarezultātā. Var izdalīt sešus etapus veiksmīgas datu analīzes veikšanai:

- 1.konkrēta mērķa noteikšana;
- 2.datu vākšana;
- 3.analīzes metožu izvēle;
- 4.programmlīdzekļu izvēle;
- 5.analīzes veikšana;
- 6.lēmuma pieņemšana par rezultātu izmantošanu.

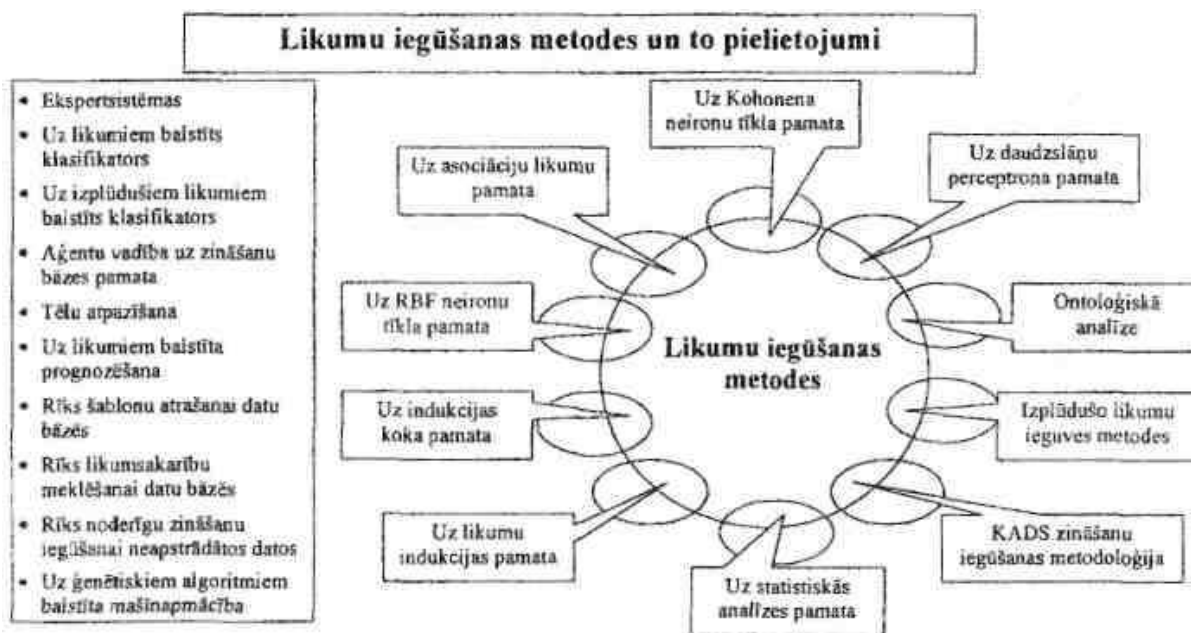
Galvenā prasība, kas tiek izvirzīta datu analīzes rezultātiem, ir tāda, ka vienmēr nepieciešams pēc iespējas korektāk interpretēt rezultātus. Likumi, kas izsaka atrastās likumsakarības, jāformulē vienkāršos un saprotamos loģisko izteikumu veidos, t.i., tiem jābūt šāda veida loģiskiem likumiem:

JA {(1. notikums) un (2. notikums) un ...(N. notikums)} TAD ...

Turpmāk darbā tiks izmantoti šāda veida loģiskie nosacītie likumi (produkciju likumi):

$$IF \underbrace{(1. \text{nosacījums}) \text{ un } (2. \text{nosacījums}) \text{ un } \dots (N. \text{nosacījums})}_{A} THEN \underbrace{(\text{slēdziens})}_{B}$$

Darba izstrādāšanas laikā radās nepieciešamība izpētīt metodes, ar kuru palīdzību iespējams iegūt nosacītos likumus no daudzdimensiju datu izlasēm. Shēma, kas parādīta 1 attēlā, tapa ilgstošas literatūras avotu izpētes procesā un atspoguļo patlaban dominējošās tendences IDA jomā.



1. att. Likumu iegūšanas metodes

Sakarā ar to, ka zinātniskās intereses bija saistītas ar mākslīgajiem neironu tīkliem un klasterizāciju, ir pamatota likumu ieguves metodes uz RBF neironu tīkla pamata izvēle, jo šis neironu tīkls apmācības etapā izmanto klasterizāciju.

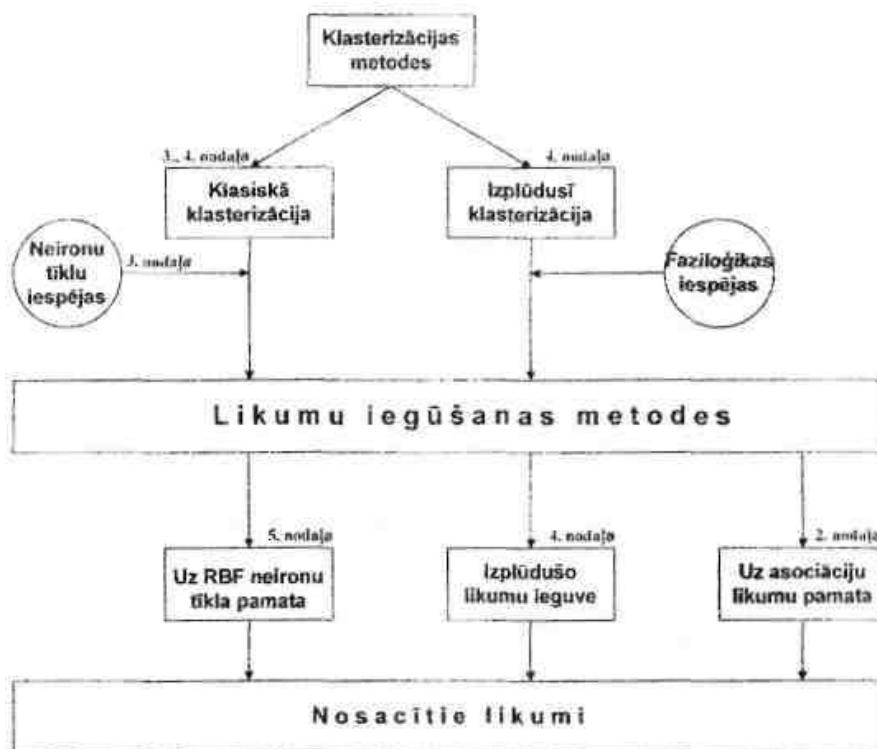
Izplūdušī klasterizācija savukārt var tikt izmantota izplūdušo likumu formēšanas sākuma posmā, kas arī turpmāk tiks parādīts.

Asociāciju likumu iegūšanas metode ir ļoti populāra intelektuālajā datu analīzē, tāpēc arī radās vēlme izpētīt šīs metodes pielietojuma jomu

Darbā apskatāmās metodes un uzdevumi, to struktūra un nodaļu numerācija, kurās tās tika pētītas, ir dotas 2. attēla shēmā.

Motivācija likumu ieguves metožu izpētei ir šāda:

- analizējamo daudzdimensiju datu apjoms kļūst pārāk liels klasiskās statistiskās analīzes iespējām;
- populārās neironu tīklu metodes darbojas pēc „melnās kastes” principa, kas lietotāju apgrūtina interpretēt iegūtos rezultātus;
- iepriekš nezināmu likumsakarību atrašana datos;
- iespēja izteikt atrastās likumsakarības lietotājam uztveramā un saprotamā veidā.



2. att. Darbā apskatāmās likumu iegūšanas metodes

**Otrajā nodaļā** apskatīta asociāciju analīzes izmantošana likumsakarību iegūšanas procesā.

Pirmajā apakšnodaļā tiek formulēts asociāciju analīzes uzdevums. Mūsdienās ir uzkrājies liels daudzums datu dažādās zinātnes, biznesa, veselības aizsardzības u.c.sfērās un rodas nepieciešamība analizēt datus labākai konkrētās nozares vadīšanai. Bieži biznesa vajadzības stimulē izstrādāt jaunas IDA metodes, kas ir orientētas uz praktisku biznesa pielietojumu. Kā piemēru var minēt vienu no problēmām, ar kuru saskaras veikalu menedžeri: ja pircējs nopērk konkrētu preci, tad X% gadījumos viņš nopērk arī citu preci, kas ir pastarpināta pirmajai precei. Piemēram, ja tiek pirktā maize un sviests, tad 90% gadījumu tiek nopirkts arī piens. Sākotnēji šis uzdevums tika izmantots lielveikalos tipisku iepirkumu šablonu atrašanai, tāpēc to dažkārt sauc par iepirkumu groza analīzi. Vispārīgā gadījumā šādas operācijas sauc par transakcijām. Likumsakarības, pēc kurām varētu spriest par šādu notikumu saistību, nosauca par asociācijām. Asociāciju likumi ļauj atrast likumsakarības starp vairākiem saistītiem notikumiem. Šādu likumu pamatā ir apgalvojums - ja ir izpildījies notikums A tad ar varbūtību X% būs spēkā arī notikums B.

Asociāciju likumu iegūšanas pamatā ir zinātnieku grupas 1993.gadā izstrādātie teorētiskie pieņēmumi par šādu likumu esamību. 1994.gadā tika publicēts efektīvs algoritms asociāciju likumu iegūšanai. Šie pētījumi stimulēja daudzu līdzīgu algoritmu izstrādāšanu, kas ļāva analizēt, piemēram, liela apjoma pirkumu operācijas un vispārināt šo uzdevumu par vienu no IDA pamatmetodēm. Asociāciju likumus var izmantot ne tikai pircēju groza analīzei, bet var pielietot dažādu datu izpētei - rūpīgi analizējot iegūtās likumsakarības.

Otrajā apakšnodaļā tiek raksturoti asociāciju likumus noteicošie raksturlielumi - ticamība un atbalsts. Visiem likumiem, kas uzdoti veidā  $1F(X) \text{ THEN } (Y)$ , ir divi raksturlielumi:

- 1) ticamība - gadījumu daļa, kad likums izpildās, starp visiem tā pielietojšanas gadījumiem (gadījumu Y daļa starp X gadījumiem);
- 2) atbalsts - gadījumu daļa, kad likums izpildās, starp visiem gadījumiem, kad izpildās Y (gadījumu X daļa starp gadījumiem Y).

Apzīmējot ar  $I = \{i_1, i_2, \dots, i_m\}$  literālu kopu, apakškopu  $X \subseteq I$  sauc par datu vienumu kopu jeb itemkopu (*itemset*).  $D = \{T_1, T_2, \dots, T_n\}$  ir transakciju kopa, kur katra transakcija  $T$ , ir itemkopa. Katra transakcija saistās jeb asociējas ar unikālu identifikatoru, sauktu par TID. Transakcija  $T$  satur itemkopu  $X$ , ja ir spēkā  $X \subseteq T$ .

Asociāciju likums formāli ir implikācija formā  $X \Rightarrow Y$ , kur  $X \subseteq I$ ,  $Y \subseteq I$  un  $X \cap Y = \emptyset$ .  $X$  tiek dēvēts par likuma nosacījumu daļu un  $Y$  - par secinājumu daļu. Katrai itemkopai ir zināms statistisks nozīmības mērs, ko sauc par atbalstu. Likumam  $X \Rightarrow Y$  ir atbalsts  $s$  transakciju kopā  $D$ , ja  $s\%$  no transakcijām  $D$  kopā satur  $X \cup Y$  t.i.,

$$s(X) = \frac{| \{T \in D \mid X \subseteq T\} |}{|D|}.$$

Likums  $X \Rightarrow Y$  ir spēkā transakciju kopā  $D$  ar ticamību  $c$ , ja  $c\%$  no  $D$  transakcijām, kas satur  $X$ , satur arī  $Y$ , t.i.,

$$c(XY) = \frac{s(X \cup Y)}{s(X)} \quad (1)$$

Ticamība nosaka likuma „stiprumu”. Ticamības robežvērtība  $c_{\min}$  tiek izmantota, lai izslēgtu likumus, kas nav pietiekoši stipri. Attiecīgi atbalsta robežvērtība  $s_{\min}$  izslēdz visus likumus, kuriem transakcijas satur nosacījumu un secinājumu daļas ar nepietiekošu apjomu.

Atbalsta robežvērtība ir definēta caur visām itemkopām. Attiecībā uz asociāciju likumiem tā procentuāli raksturo transakciju skaitu, kas satur visas itemkopas, kas parādās likumā. Ticamības robežvērtība raksturo minimālo varbūtību, ka secinājumu daļa ir patiesa tad, ja nosacījumu daļa ir patiesa. Ticamības vērtība tuvu pie 100% raksturo ļoti "spēcīgus" likumus. Formāli asociāciju likumu atrašanas problēmu var raksturot šādi: ja dota itemkopu datu bāze  $D$ , atbalsta robežvērtība  $s_{\min}$  un ticamības robežvērtība  $c_{\min}$ , tad ir jāatrod visi asociāciju likumi  $X \Rightarrow Y$  veidā. Tādējādi ir jāatrod visi asociāciju likumi  $X \Rightarrow Y$  kopā  $D$  ar atbalsta vērtību  $s(X \cup Y) \geq s_{\min}$  un ticamību  $c(X, Y) \geq c_{\min}$ .

Ja transakciju datu bāze sastāv no pirkumu groza ar produktiem  $A, B, C$  un  $D$  (sk 1.tabulu), kur transakcijas identifikators TID raksturo pircēja veikto operācijas numuru, tad var iegūt šos datus raksturojošos likumus.

1.tabula

Transakciju datu bāzes piemērs

TID	Pirkumu grozs	TID	Pirkumu grozs
1	{A,C}	6	{A,B}
2	{B}	7	{A,D}
3	{A,B,C,D}	8	{B,C,D}
4	{B,D}	9	{C,D}
5	{A,B,D}	10	{A,B,D}

Itemkopas  $\{A,B\}$  atbalsts ir 0.4.  $\{A,B,D\}$  atbalsta vērtība ir 0.3. Tādējādi likuma  $\{A,D\} \Rightarrow \{B\}$  ticamība ir 0.75, jo pēc formulas (1) iegūst:

Ja atbalsta robežvērtība  $s_{\min}$  ir mazāka vai vienāda ar 0.3 un ticamības robežvērtība  $c_{\min}$  ir mazāka vai vienāda ar 0.75, tad šis likums tiek uzlūkots kā pieņemams asociāciju likums. Var teikt, ka ir iegūts asociāciju likums „Ja pircējs nopērk produktus  $A$  un  $D$ , tad ir iespējams, ka 75% gadījumos viņš nopirks arī produktu  $B$ ”.

$$c(A, D \Rightarrow B) = \frac{s(A \cup B \cup D)}{s(B)} = \frac{0.3}{0.4} = 0.75.$$

Trešajā un ceturtajā apakšnodaļā tiek analizēti itemkopu atrašanas algoritmi *Apriori* un *AprioriTID*, kuru darbības rezultātā arī tiek iegūti asociāciju likumi. Vispārīgā gadījumā visu asociāciju likumu iegūšanas process tiek reducēts uz diviem apakš uzdevumiem:



- 1) atrast visas itemkopas, kurām transakciju atbalsta vērtības pārsniedz atbalsta robežvērtību. Itemkopa ar minimālo atbalstu tiek dēvēta par lielo itemkopu, pārējās par mazajām itemkopām;
- 2) izmantojot lielo itemkopu, jāatrod attiecīgie likumi.

Transakciju datu bāzei  $D = \{\{A,C,D\}, \{B,C,E\}, \{A,B,C,E\}, \{B,E\}\}$  ar minimālo atbalsta robežvērtību 0.5 *Apriori* algoritma izpildes gaitā iegūtās itemkopas  $L_k$  un kandidātkopas  $C_k$  pēc trijām iterācijām ir parādītas 3. attēlā.

$L_1$		$C_2$	$L_2$		$C_3$	$L_3$	
Itemkopa	Skaitis	Itemkopa	Itemkopa	Skaitis	Itemkopa	Itemkopa	Skaitis
{A}	2	{A,B}	{A,C}	2	{B,C,E}	{B,C,E}	2
{B}	3	{A,C}	{B,C}	2			
{C}	3	{A,E}	{B,E}	2			
{E}	3	{B,C}	{C,E}	2			
		{B,E}					
		{C,E}					

3. att. *Apriori* algoritma izpildes piemērs

*AprioriTID* algoritms ir līdzīgs *Apriori* algoritmam, taču atšķiras ar papildus kopu izmantošanu potenciālo kandidātkopu apstrādei. Kandidātkopu iegūšanas process ir ilustrēts ar 4. attēlā doto transakciju datubāzi pie minimālās atbalsta vērtības 2.

Datu bāze		$C_1$		$L_1$	
TID	Elementi	TID	Elementu kopas	Itemkopa	Atbalsts
100	1 3 4	100	{{1},{3},{4}}	{1}	2
200	2 3 5	200	{{2},{3},{5}}	{2}	3
300	1 2 3 5	300	{{1},{2},{3},{5}}	{3}	3
400	2 5	400	{{2},{5}}	{5}	3

$C_2$		$C_2$		$L_2$	
Itemkopa	Atbalsts	TID	Elementu kopas	Itemkopa	Atbalsts
{1 2}	1	100	{{1 3}}	{1 3}	2
{1 3}	2	200	{{2 3},{2 5},{3 5}}	{2 3}	2
{1 5}	1	300	{{1 2},{1 3},{1 5}}	{2 5}	3
{2 3}	2		{{2 3},{2 5},{3 5}}	{3 5}	2
{2 5}	3	400	{{2 5}}		
{3 5}	2				

$C_3$		$C_3$		$L_3$	
Itemkopa	Atbalsts	TID	Elementu kopas	Itemkopa	Atbalsts
{2 3 5}	2	200	{{2 3 5}}	{2 3 5}	2
		300	{{2 3 5}}		

4. att. *AprioriTID* algoritma izpildes piemērs

Piektajā apakšnodalā aprakstīta eksperimentu sērija, lai noskaidrotu iegūto līkumu skaita atkarību no atbalsta sākumvērtībām. Par eksperimentālajiem datiem kalpoja Centrālās statistikas pārvaldes informācija par 3044 respondentu atbilžu variantiem (datu struktūra un anketa dota darba pielikumā) Eksperimentam izvēlētie dati skāra iedzīvotāju māsaimniecību ekonomisko pašnovērtējumu. Respondentiem bija uzdoti 4 jautājumi.

- 1) Ņemot vērā Jūsu māsaimniecības kopējo ekonomisko situāciju, lūdzu, pasakiet, kurš no izteikumiem vislabāk raksturo Jūsu situāciju (doti 7 atbilžu varianti).
- 2) Ja Jūsu māsaimniecībai pēkšņi ievajadzētos 120 latu, vai Jūs varētu sagādāt šādu naudas summu nedējas laikā? (Doti 6 atbilžu varianti).

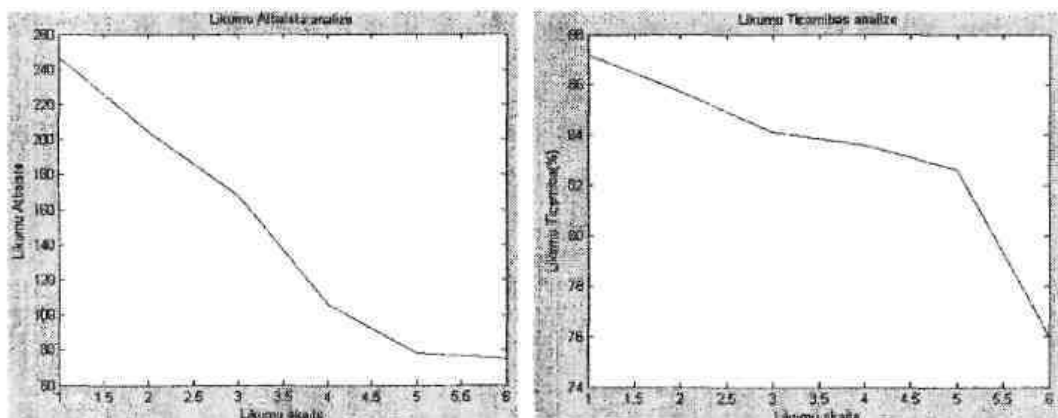
3) Vai Jūsu mājsaimniecības ekonomiskā situācija patlaban ir labāka, tāda pati vai sliktāka nekā pirms 5 gadiem? (Doti 5 atbilžu varianti).

4) Vai Jūsu mājsaimniecības ekonomiskā situācija pēc 5 gadiem būs labāka, paliks tāda pati vai būs sliktāka, salīdzinot ar pašreizējo situāciju? (Doti 5 atbilžu varianti).

Eksperimenta mērķis bija noteikt iespējamās likumsakarības šajos datos un noteikt iegūto likumsakarību skaita atkarību no sākumā uzdotajām atbalsta un ticamības robežvērtībām. Pirmajā eksperimenta daļā tika pieņemts, ka ticamības robežvērtība  $c_{\min}=50$  un atbalsta robežvērtība  $s_{\min}=10$ . Tika iegūts 201 likums (visu iegūto likumu saraksts dots darba pielikumā). Otrajā eksperimenta daļā tika pieņemts, ka ticamības robežvērtība  $c_{\min}=75$  un atbalsta robežvērtība  $s_{\min}=75$ . Tika iegūti 6 likumi. Katram likumam tika izskaitļota atbalsta vērtība un ticamības vērtība:

- 1) 25 ->34 Atbalsts=247 Ticamība=82
- 2) 32 42 ->23 Atbalsts=205 Ticamība=75
- 3) 25 43 ->34 Atbalsts=168 Ticamība=83
- 4) 25 53 ->34 Atbalsts=106 Ticamība=84
- 5) 25 59 ->34 Atbalsts=75 Ticamība=87
- 6) 25 43 53 ->34 Atbalsts=78 Ticamība=85

Iegūto likumu atbalsta analīzes grafiku un likumu ticamības analīzes grafiku var apskatīt 5. attēlā.



5. att. Likumu skaita atkarības no atbalsta un ticamības vērtībām  $c_{\min}=75$  un  $s_{\min}=75$

Pirmais likums nosaka:

**IF** „Mūsu saimniecība ir nabadzīga” **THEN** „Mēs nevarētu pēkšņi sagādāt 120 latu nedēļas laikā”.

Otrais likums nosaka:

**IF** „Mums būtu nepieciešama citu palīdzība sagādāt 120 latu nedēļas laikā” **AND** „Mūsu mājsaimniecības ekonomiskā situācija ir tādi pati, kā 5 gadus atpakaļ” **THEN** „Mēs neesam ne bagāti, ne nabadzīgi”.

Trešais likums nosaka:

**IF** „Mēs esam nabadzīgi” **AND** „Mūsu mājsaimniecības ekonomiskā situācija ir sliktāka, kā 5 gadus atpakaļ” **THEN** „Nebūtu iespējams sagādāt 120 latu nedēļas laikā”.

Ceturtais likums nosaka:

**IF** „Mēs esam nabadzīgi” **AND** „Mūsu mājsaimniecības ekonomiskā situācija pēc 5 gadiem būs sliktāka nekā tagad” **THEN** „Nebūtu iespējams sagādāt 120 latus nedēļas laikā”.

Piektais likums: kļūdainis, jo anketā jautājumā D05 nav tāda koda. Visticamāk, tā ir anketas aizpildītāja kļūda.

Sestais likums nosaka:

IF „Mēs esam nabadzīgi” AND “Mūsu mājsaimniecības ekonomiskā situācija ir sliktāka, kā 5 gadus atpakaļ” AND „Mūsu mājsaimniecības ekonomiskā situācija pēc 5 gadiem būs sliktāka nekā tagad” THEN „Nebūtu iespējams sagādāt 120 latu nedējas laikā”.

Var secināt, ka iegūtie likumi ir loģiski saprotami un atspoguļo reālo situāciju.

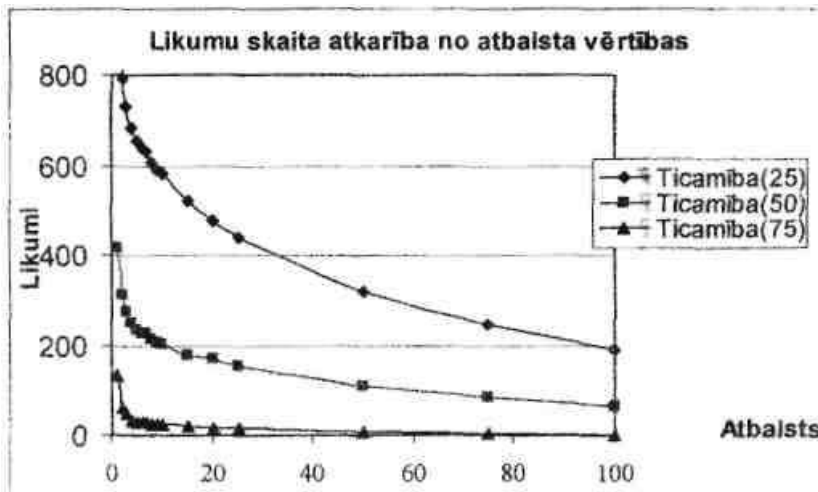
Trešajā eksperimenta daļā tika iegūtas likumu skaita vērtības pie dažādām atbalsta vērtībām un ticamības robežvērtībām (rezultāti parādīti 2.tabulā)

2 tabula

Likumu skaita atkarība no atbalsta vērtības

Atbalsts	1	2	3	4	5	6	7	8	9	10	15	20	25	50	75	100
Ticamība (25)	957	792	731	684	656	639	629	608	59t	582	522	478	439	319	245	189
Ticamība (50)	417	310	273	249	235	226	225	215	205	201	177	168	153	108	83	64
Ticamība (75)	133	62	48	32	28	28	28	26	24	23	19	17	15	10	6	n/a

Grafiskā veidā atbilstība parādīta 6. attēlā. No tabulas datiem un grafiskās atbilstības var secināt - jo lielāks uzdotais ticamības līmenis un atbalsta robežvērtība; jo mazāks iegūto asociāciju likumu skaits un līdz ar to likumi ir „stingrāki”.



6. att. Likumu skaita atkarība no atbalsta vērtībām pie dažādām ticamības robežvērtībām

Sestajā apakšnodaļā izdarīti secinājumi, ka atsevišķu uzdevumu klasēm asociāciju likumu ieguves mehānisms ir ļoti lietderīgs. Svarīgi ir apzināties, ka iegūtie asociāciju likumi prasa rūpīgu analīzi, lai to izmantošana būtu lietderīga. Asociāciju likumu ieguves mehānisms, kas sākotnēji bija izstrādāts patērētāju pirkuma groza analīzei, izrādījās labs instrumentārijs plašākam uzdevumu lokam. Ar tā palīdzību var meklēt un atrast likumsakarības  $X \Rightarrow Y$  veidā dažāda tipa datus. Tagad tas ir ieguvis plašu pielietojumu lielu datubāzu apstrādes un analīzes uzdevumos. Pilnīgi pamatoti asociāciju likumu ieguves metode ir starp galvenajām intelektuālās datu apstrādes pamatmetodēm. Tomēr jāsecina, ka asociāciju likumu analīzē ir arī savas vājās vietas, kuru izpēte varētu būt vērtīgs pielietojumu lauks.

- asociāciju likumu programmrealizācijas izpilde prasa ievērojamu laika patēriņu;
- analizējamajiem datiem jābūt pēc iespējas viendabīgiem;
- kļūdaini dati arī piedalās likumu veidošanā.

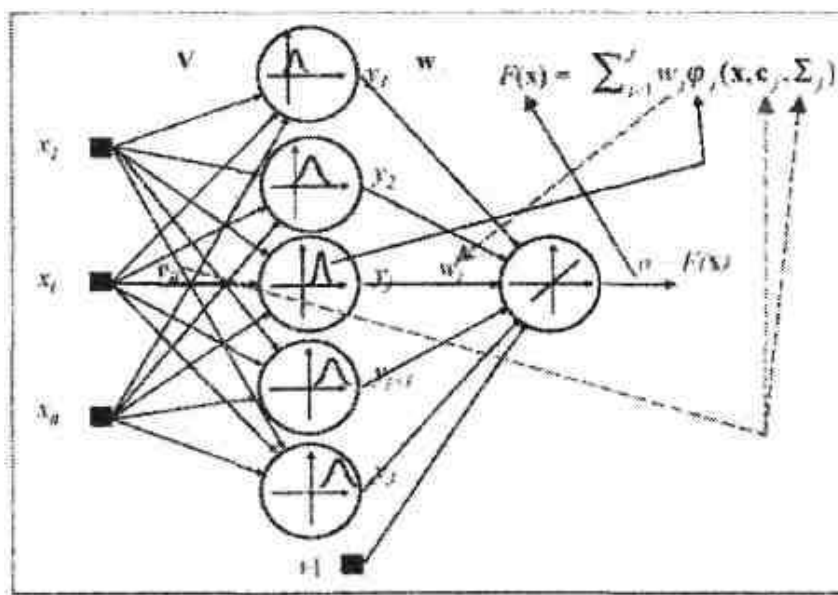
**Trešajā nodaļā** tiek pētītas mākslīgo neironu tīklu iespējas datu analīzes veikšanā.

**Pirmajā apakšnodaļā** dota uzdevuma nostādne - izpētīt neironu tīklu iespējas likumsakarību meklēšanā. Neironu tīkli (tas ir vispārināts apzīmējums noteiktai algoritmu klasei) ir spējīgi

apmācīties no piemēriem, iegūstot slēptās likumsakarības no datiem. Ja starp ieejas un izejas datiem pastāv kaut kāda sakarība (pat ja tradicionālās metodes to neuzrāda), neironu tīkls ir spējīgs automātiski noskaņoties uz to ar noteiktu precizitāti. Turklāt daudzi neironu tīkli ļauj novērtēt ieejas datu atsevišķu pazīmju nozīmīgumu, ļauj samazināt datu apjomu bez būtiskāko datu zaudēšanas, ļauj atpazīt kritisku situāciju tuvošanos. Daudzos gadījumos neironu tīkli ļauj atrast likumsakarības, kuras praktiski nav iespējams atklāt, analizējot datus manuāli.

Otrajā apakšnodaļā aprakstīti neironu tīklu darbības vispārējie pamatprincipi - dota neirona un neironu tīkla definīcija, uzdotas vairākas aktivizācijas funkciju formulas un parādīta neironu tīklu klasifikācijas shēma. Atsevišķās darba apakšnodaļās tiks izmantoti trīs neironu tīklu modeļi; radiālās bāzes funkciju (RBF) neironu tīkls, daudzslāņu tīkls un Bolcmana mašīna.

Trešajā apakšnodaļā aprakstīta neironu tīklu metožu iespēju pētīšana likumsakarību meklēšanā. Par pamatu tiek ņemts RBF neironu tīkls, kura arhitektūra attēlota 7. attēlā.



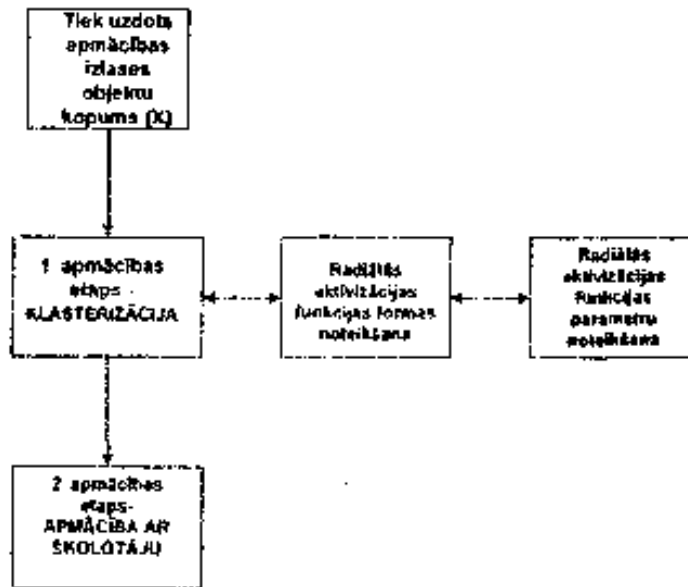
7.att. Neironu tīkla ar radiālo aktivizācijas funkciju arhitektūra

Vispārīgā gadījumā RBF neironu tīkls satur ieejas slāni, slēpto slāni ar radiālajām aktivizācijas funkcijām un izejas slāni. Par radiālajām funkcijām izmanto Gausa jeb potenciālo funkciju veidā:

$$\Phi(x) = e^{-\frac{\|x_i - c_i\|^2}{\sigma^2}},$$

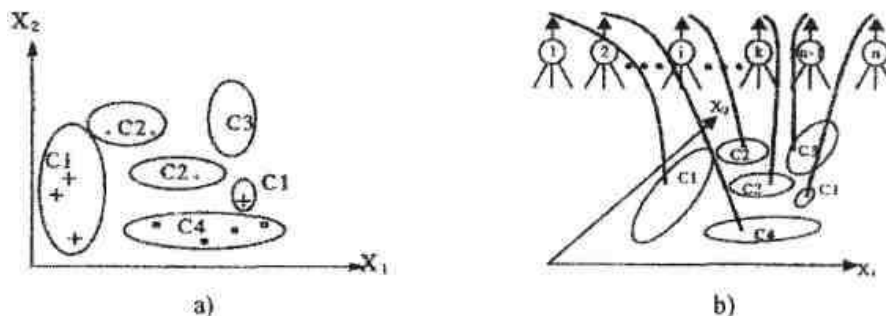
- kur  $x_i$  - n-dimensiju ieejas vektora komponentes;
- $c_i$  - radiālās funkcijas centrs (RBF tīklos bieži uzdod kā  $w_i$ );
- $\sigma$  - standarta novirze.

Slēptā slāņa neironu skaits tiek noteikts apmācības gaitā. Parasti katrs slēptā slāņa neironi atbilst konkrētai objektu klasei. Uzskatāmības labad var teikt, ka slēptā slāņa neironi izskaitļo Eiklīda attālumu starp ieejas objektiem un radiālās funkcijas centru. Slēptā slāņa neironu izejas vērtības ir ieejas signāli izejas slāņa neironam. Tīkla izeja ir svērta izejas neirona ieejas signālu summa Neironu tīkla ar radiālo aktivizācijas funkciju apmācība notiek divos etapos (sk. 8. att.) Šis apmācības algoritms ir izvēlēts tāpēc, ka tā ir galvenā metode šāda tipa neironu tīklu apmācībā.



8.att Neironu tīkla ar radiālo aktivizācijas funkciju apmācības shēma

Pirmajā apmācības etapā apmācība slēptajā slānī tiek veikta ar nekontrolējamu apmācības algoritmu palīdzību, kuri tiek saukti par klasterizācijas algoritmiem. Klasterizācijas mērķis ir sadalīt ieejas datus objektu grupās (klasteros, klasēs, taksonos) un noteikt šo objektu centrus. Klasterizācijas būtība ir attēlota 9.a un b attēlā.



9.att. a) divdimensiju objektu sadalījums klasteros; b) klasteru attēlošana RBF neironos

Klasteranalīzes process formāli sastāv no šādiem posmiem:

- analīzei nepieciešamo datu savākšana;
- klašu datu (klasteru) raksturojošo lielumu un robežu noteikšana;
- datu grupēšana klasteros;
- klašu hierarhijas noteikšana un rezultātu analīze.

RBF tīkla apmācībai tiek pielietots klasterizācijas algoritms ar nosaukumu *C-vidējā*. Tas minimizē kvalitātes rādītāju, kurš noteikts kā visu punktu, kas pieder klastera apgabalam, attālumu līdz klastera centra kvadrātu summa. Šī procedūra ieguva tādu nosaukumu, pamatojoties uz klasteru grupas iekšienē vidējo attālumu aprēķināšanu līdz klastera centram.

Algoritma izpildes soļi ir sekojoši:

- (1) inicializē klasteru centrus  $w_j$  ( $j$  - nepieciešamo klasteru skaits uzdevuma risināšanai).
- (2) grupē visus apmācības izlases punktus ap tuvākā klastera centru, t.i., katru punktu  $x_i$  saista ar klasteru  $j^*$ , kuram 
$$\|x_i - w_{j^*}\| = \min_j \|x_i - w_j\|$$
- (3) izskaitļo jaunus klasteru centrus, t.i., visiem  $w_j$  izskaitļo:

$$\sigma_j^2 = \frac{1}{M_j} \sum_{x_i \in \text{klasteram}_j} x_i, \text{ kur } m_j - \text{klasterim } j \text{ piederošo punktu skaits,}$$

(4) atkārtoti (2) soli tik ilgi, kamēr iterāciju laikā nemainās klasteru centru vērtības.

Algoritma darbības rezultātā tiek noteikti galīgie klasteru centri  $w_i$ , ievērojot nosacījumu, ka attālumu kvadrātu summai starp visiem punktiem, kas pieder grupai  $j$ , un klastera centru ir jābūt minimālai.

Pēc slēptā slāņa apmācības pabeigšanas tiek iegūti aktivizācijas funkciju parametri. Tie ir klasteru centri  $W_j^h$  un klasteru standarta novirze  $\sigma_j^2$  ( $j$  ir klasteru skaits). Lielums  $\sigma_j^2$  tiek noteikts pēc formulas:

$$\sigma_j^2 = \frac{1}{M_j} \sum_{x \in \Theta_j} x_i (x - w_j^h),$$

kur  $\Theta_j$  - izvēlētais klasteris  $j$ , kura punkti grupējas ap klastera centru  $w_j^h$ ;

$M_j$  - klasterim  $j$  piederošo punktu skaits;

$T$  - transponētās matricas vai vektora apzīmējums.

Par algoritma *C-vidējā* priekšrocībām var uzskatīt popularitāti, lielo efektivitāti un procedūras vienkāršību. Bet gadījumā, kad objektu izvietojums ir neviendabīgs, algoritms var nesasniegt labus rezultātus. Tad ir jāmaina parametri (klasteru centru skaits) un atkal jāmēģina atkārtot algoritma darbību. Par trūkumu tiek uzskatīts tas, ka algoritms nav universāls.

Pēc apmācības slēptajā slānī un radiālo funkciju parametru noteikšanas notiek apmācība neironu tīkla izejas slānī, pielietojot kontrolējamās apmācības metodes vai tā saucamo "apmācību ar skolotāju". Šim mērķim tiek izmantots mazākās vidējās kvadrātiskās kļūdas algoritms (LMS).

Apmācošie dati sastāv no pāriem  $\{u_k/d_k\}$ , kur  $u_k$  ir ieejas objekts izejas slānim, bet  $d_k$  ir tīkla mērķa izejas,  $k$  - objektu skaits. Ieejas signāls izejas slānim ir apstrādāts slēptajā slānī pirms tam, kad tiks padots uz izejas slāni. Citiem vārdiem sakot, izejas neirona ieejas lielumi ir slēptā slāņa izejas vērtības  $u_{ki}$  (radiālo funkciju vērtības), kur  $i = 1, \dots, M$ . Tie ir jāaprēķina pēc formulas:

$$u_i = \exp \left[ - \frac{(x - w_{ji}^h)^T (x - w_{ji}^h)}{2\sigma^2} \right],$$

kur  $x$  - ieejas vektors;

$W_j^k$  -  $j$  neirona svaru vektors (faktiski tas ir klastera centrs, kam pieder dotais ieejas vektors);

$\sigma_j^2$  -  $i$  neirona standarta novirze;

$M$  - slēptā slāņa neironi;

$N$  - izejas slāņa neironu skaits.

Izejas slāņa  $y_k$  izeju  $k$  ieejas vektoram aprēķina pēc formulas:

$$y_k = w_i^0 u_i,$$

kur  $w_i^0$  - svaru vektors, kas saista  $i$  slēptā slāņa neironu ar izejas neironu;

$u_i$  - ieejas signāls izejas neironam.

Apmācības procesā, ievērojot reālo un mērķa izejas vērtības (attiecīgi  $y_k$  un  $d_k$ ), tiek izskaitļota neironu tīkla vidējā kvadrātiskā kļūda pēc formulas:

$$E_k = \frac{1}{2} \sum_{k=1}^K (d_k - y_k)^2 = \frac{1}{2} \|d_k - y_k\|^2$$

Par algoritma apstāšanās nosacījumu pieņem kādu noteiktu minimālās kļūdas vērtību  $E_{\min}$ . Ja apmācības cikla (iterācijas) vidējā kvadrātiskā kļūda ir vienāda vai mazāka nekā  $E_{\min}$ , tad apmācības process tiek uzskatīts par pabeigtu un līdz ar to pēc otrā apmācības etapa pabeigšanas RBF tīklu var uzskatīt par apmācītu un sagatavotu tālākajiem pētīšanas eksperimentiem.

Tālāk trešajā apakšnodaļā tiek aplūkots vēl viens darbā apskatāmais neironu tīklu tips -

Bolcmana mašīna, kurš pārsvarā tiek izmantots optimizācijas uzdevumos, analizēts Bolcmana mašīnas apmācības algoritms un imitētās atkvēlināšanas (SA) metodes būtība.

Ceturtajā apakšnodalā ir parādītas neironu tīklu pielietojumu iespējas. Pirmajā piemērā uz nelielas datu izlases pamata tiek detalizēti aprakstīta RBF neironu tīkla apmācības procedūra tēlu atpazīšanas uzdevumā Otrajā piemērā SA metode tiek piemērota klasiskajai komivojažiera problēmas atrisināšanai - tika izskaitļots īsākais attālums starp Latvijas pilsētām (TSP-26).

Piektā apakšnodalā ir veltīta praktiska uzdevuma risināšanai - bankrotu datu analīzei. Realizācijas uzdevuma mērķis ir noteikt firmu finansiālo stāvokli (bankrots vai ne-bankrots). Bankrota diagnostika ir finanšu analīzes sistēma, kuras mērķis ir krīzes situācijas kontrolēšana uzņēmumos. Šim nolūkam kopējā uzņēmuma finansiālā stāvokļa analīzē tiek izveidota atsevišķa finanšu rādītāju grupa, pēc kuras var spriest par bankrota draudiem.

Pastāv divas pieejas bankrotu prognozēšanā. Pirmā pieeja pamatojas uz finansiālajiem datiem un ietver darbības ar dažādiem rādītājiem: ļoti lielu popularitāti ieguvušais Altmana Z-skaitlis, Taflera koeficienti u.c. Otrā pieeja pamatojas uz datiem par bankrotējušām kompānijām, kurus salīdzina ar analizējamo kompāniju datiem.

Eksperimenta mērķis bija salīdzināt vairākas metodes bankrotu datu analīzē un izvērtēt to rezultātus. Par pētījuma datiem tika izmantoti 63 firmu bilances dati (46 - bankrotējušas un 17 - nebankrotējušas). No datiem tika izskaitļoti šādi finanšu rādītāji:

- R2: Naudas plūsma / Īstermiņa saistības;
- R3: Naudas plūsma / Aktīvu kopsumma;
- R7: Apgrozāmie līdzekļi / Īstermiņa saistības;
- R9: Apgrozāmie līdzekļi / Aktīvu kopsumma;
- R31: Tīrais apgrozāmais kapitāls / Aktīvu kopsumma.

Pirmā metode - diskriminantanalīze (MDA), kas tika veikta ar statistiskās paketes SPSS palīdzību. Pielietojuma rezultāti doti 3.tabulā.

3.tabula

Diskriminantanalīzes klasifikācijas rezultāti

		Prognozējamās grupas elementi		Kopā
		0	1	
Oriģināls	Skaitis	0	39	46
		1	4	17
	%	0	84.8	100.0
		1	23.5	100.0

Rezultātā 82.5% no visiem elementiem ir korekti klasificēti. Neklasificēto elementu numuri: 14, 26, 28, 35, 36, 37, 41, 58, 59, 60, 62.

Kā otro metodi autors piedāvā izmantot vienu no tēlu atpazīšanas metodēm - potenciālu funkciju metodi ar nolūku pārbaudīt pieņēmumu par šīs metodes izmantojamību bankrotu diagnostikas uzdevumos. Potenciālu funkciju metodes algoritms pamatojas uz hipotēzi par funkciju raksturu, kas atdala kopas atbilstoši dažādām tēlu klasēm. Par potenciālo funkciju tika ņemta funkcija:

$$\varphi(R) = -\frac{\lambda}{1 + \alpha R^2}$$

kur  $\alpha$  - apmācības koeficients;

R- attālums starp punktu, kurā tiek rēķināts potenciāls, un punktu no apmācošās kopas;

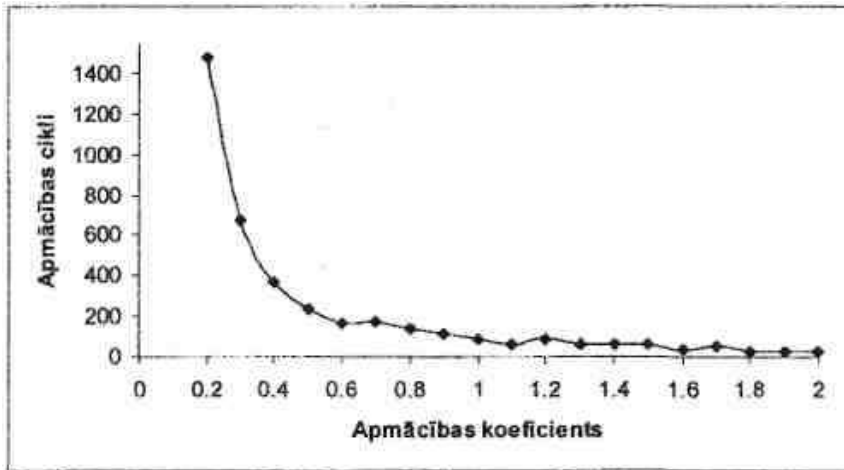
X - potenciāla lielums, kas tiek uzdots punktam apmācības procesā.

Pirmajā eksperimenta daļā tika pētīta apmācības koeficienta ietekme uz apmācības ilgumu. Par apmācošo izlasi tika ņemti bankrotu dati, par testu datiem kalpoja izlase no tiem pašiem firmu bankrotu datiem. Apmācības koeficients a mainījās robežās [0.1, 2.0]. Apmācības rezultāti parādīti 4. tabulā un 10. attēlā.

4.tabula

Iegūtās apmācības koeficienta a un apmācības ciklu skaita vērtības

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
cikli	5203	1477	674	371	238	166	173	142	117	91	64	84	60	59	62	36	45	30	25	25



10.att. Apmācības koeficienta atkarība no apmācības ciklu skaita

Analizējot apmācības rezultātus, var secināt, ka algoritms konverģē pie visiem uzdotajiem a un apmācības ciklu skaits samazinās, palielinoties  $\alpha$  vērtībai.

Otrajā eksperimenta daļā tika veikta testēšana un analizēta apmācības algoritma darbība, ņemot vērā potenciāla vērtību  $\lambda$ . Tika konstatēts, ka pēc apmācības dotās testu kopas elementi tiek identificēti korekti, t.i., apmācības procesā izskaitļotās  $\lambda$  vērtības dod iespēju pareizi noteikt ieejas datu klasi - bankrots vai ne-bankrots. Taču, ņemot vērā to, ka MDA metode tikai 82.5% gadījumu korekti klasificē ieejas datus, radās jautājums: ar kādiem apmācāmo datu elementiem apmācības procesā bija grūtības ar to identificēšanu. Pēc potenciālo funkciju izmantošanas algoritma sākotnējās  $\lambda$  vērtības ir vienādas ar 0. Ja apmācības procesā elements tiek nepareizi atpazīts, parametra vērtība  $\lambda$  tiek palielināta par 1.

Eksperimenta rezultātā tika konstatēts, ka bankrota datu kopā ir 7 elementi, kuriem apmācības laikā visbiežāk tiek veikta parametra  $\lambda$  korekcija. 5. tabulā parādīti šie elementi un veikto korekciju skaits pie dažādām  $\alpha$  vērtībām (pie pārējām a vērtībām aina būtiski nemainās). Grafiskā veidā atbilstība ilustrēta 11. attēlā.

5.tabula

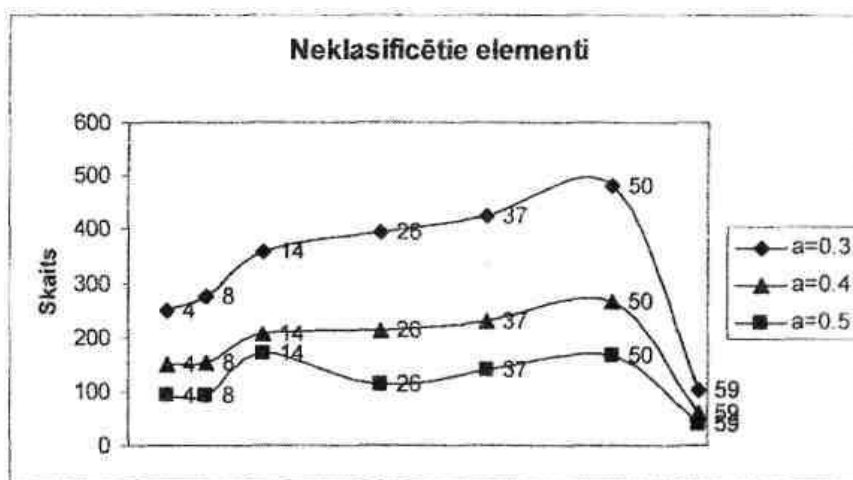
Neklasificētie elementi un to parametri  $\lambda$

$\alpha=0.3$	Elements	$\lambda$
	4	249
	8	274
	14	357
	26	393
	37	424
	50	482
	59	104

$\alpha=0.4$	Elements	$\lambda$
	4	148
	8	153
	14	206
	26	212
	37	228
	50	265
	59	60

$\alpha=0.5$	Elements	$\lambda$
	4	92
	8	94
	14	168
	26	112
	37	140
	50	165
	59	39





11 .att. Neklasificēto elementu skaita atkarība no parametra  $\lambda$

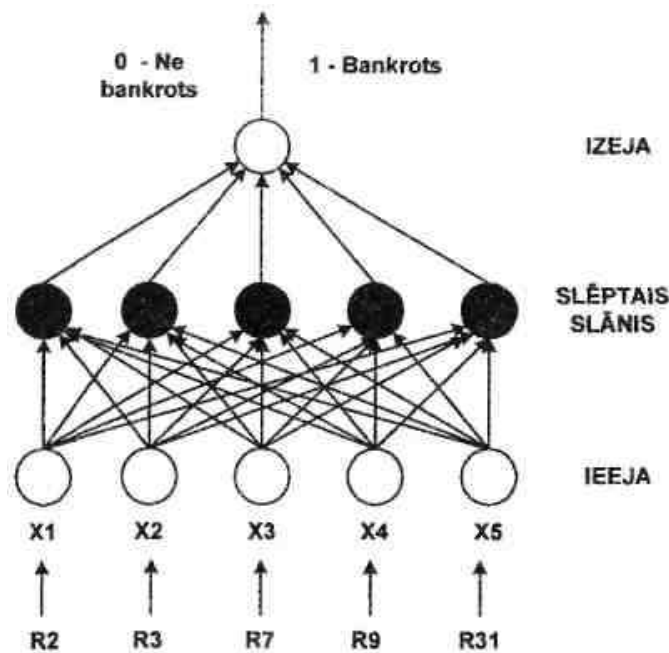
Ar šiem bankrotu datiem potenciālu funkciju gadījumā neklasificētie elementi ir: 4, 8, 14, 26, 37, 50, 59. Redzams, ka potenciālu funkciju metode, līdzīgi kā MDA, nespēj korekti klasificēt elementus 14, 25, 37 un 59, bet parādās arī citi elementi, kas ir atšķirīgi no MDA. Var secināt, ka potenciālu funkciju metodi var pielietot bankrotu datu analīzē, taču jābūt piesardzīgiem rezultātu interpretācijā.

Trešās metodes demonstrēšanai bankrotu datu analīzē tika izmantota neironu tīklu pieeja. Neironu tīklu pieeja kļuva ārkārtīga populāra pagājušā gadsimta 90. gados. Sākot ar 1990. gadu, periodiski tika veikti pētījumi par neironu tīklu izmantošanu bankrotu analīzē. Vieni no pirmajiem, kas izmantoja neironu tīklu metodes bankrotu analīzes pētījumos, bija zinātnieki Odoms un Šarda. Tīkla ieejā tika izmantoti Altmana Z-skaitļi par 128 kompānijām un tika parādīts, ka neironu tīklu pieeja dod labākus rezultātus par MDA. Zinātnieki Tams un Kiangs salīdzināja vairākas bankrotu diagnostikā izmantojamās metodes (MDA, LA, ID3, vienslāņa tīklu un daudzslāņu tīklu) un parādīja, ka atsevišķos gadījumos daudzslāņu tīkls ir visefektīvākais.

Analizējot neironu tīklu izmantošanu bankrotu analīzē, tika izdarīti šādi secinājumi.

- neironu tīklu izmantošana dod apmēram 90% precizitāti, salīdzinot ar 80-85% citām metodēm (MDA, LA, ID3);
- bankrotus var paredzēt vairākus gadus līdz to notikšanai, turklāt paredzēšanas precizitāte datiem „divus gadus uz priekšu” praktiski neatšķiras no datiem „vienu gadu uz priekšu”.

Bankrotu analīzē priekšroka pārsvarā tiek dota daudzslāņu tīklam ar kļūdas atpakaļ izplatības apmācību. Bankrotu analīzes uzdevums, formāli ņemot, neatšķiras no klasiskā tīklu atpazīšanas uzdevuma. Tīkla ieejā tiek padoti finanšu rādītāji, notiek tīkla apmācība, tiek izmantoti slēptie neironi. Tīkla izejā ir tikai 2 stāvokļi - bankrots un ne-bankrots. Eksperimentos izmantojamā tīkla shēma parādīta 12. attēlā.



12.att. Daudzslāņu tīkla 5-5-1 arhitektūra

Izvēlēta tīkla modeļa svarīgākie parametri ir aktivizācijas koeficients  $\alpha$  un aktivizācijas funkcijas  $\tanh$  slīpums  $\beta$ . Pirmajā eksperimenta daļā tika pārbaudīta apmācības kvalitātes atkarība no  $\alpha$  vērtības. Dotais eksperimentālais modelis parādīja, ka pie  $\alpha$  vērtībām robežās līdz 0.8 algoritms vai nu nekonverģē, vai arī nenotiek tīkla korekta apmācība. Ticami apmācību rezultāti tika iegūti pie  $\alpha=0.8$  un  $\alpha=0.9$ . Otrajā eksperimenta daļā tika pārbaudīta apmācības kvalitātes atkarība no aktivizācijas funkcijas  $\tanh$  slīpuma  $\beta \in [0.1; 1]$ . Eksperimentu rezultātā izrādījās, ka tikai pie atsevišķām  $\beta$  vērtībām tika iegūta pieņemama apmācības kvalitāte. 6. tabulā parādīti zīmīgākie rezultāti un neklasificētie elementi pie dažādām  $\alpha$  un  $\beta$  vērtībām.

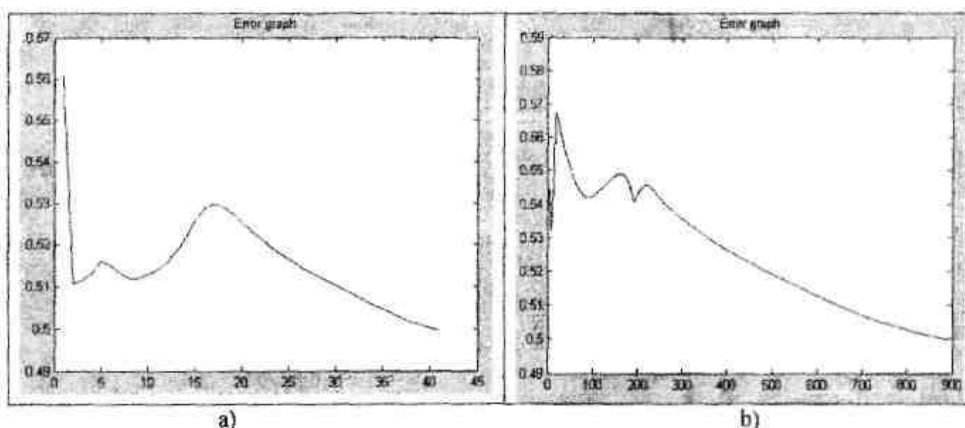
6.tabula

Eksperimenta rezultāti (parametri  $\alpha$ ,  $\beta$  un neklasificētie elementi)

Parametrs $\alpha$	Parametrs $\beta$	Cikli	Neklasificēto elementu skaits	Elementi
$\alpha = 0.8$	$\beta = 0.8$	41	9	14,26,36,37,41,58,59,60,62
	$\beta = 0.9$	889	6	37,50,58,59,60,62
$\alpha = 0.9$	$\beta = 0.8$	46	9	14,26,35,36,37,41,58,59,62
	$\beta = 0.9$	1489	7	37,50,58,59,60,62,63

13.a un b attēla parādīts apmācības kļūdu grafiks pie  $\beta = 0.8$  un  $\beta = 0.9$ .

7. tabula apkopo datus par visu pielietoto metožu neklasificētajiem elementiem.



13. att. Apmācības kļūdu grafiks pie  $\beta = 0.8$  (a) un  $\beta = 0.9$  (b)

7 tabula

Kopsavilkuma dati par izmantotajām metodēm un to neklasificētajiem elementiem

Metode	Neklasificētie elementi														
MDA			14	26	28	35	36	37	41	58	59	60	62		
Potenc	4	8	14	26				37		50	59				
NN-1	$\alpha = 0.8, \beta = 0.8$		14	26			36	37	41	58	59	60	62		
NN-2	$\alpha = 0.8, \beta = 0.9$							37		50	58	59	60	62	
NN-3	$\alpha = 0.9, \beta = 0.8$		14	26		35	36	37	41	58	59		62		
NN-4	$\alpha = 0.9, \beta = 0.9$							37		50	58	59	60	62	63

No 7. tabulas datiem redzams, ka dotajai bankrota datu izlasei visas metodes nespēj klasificēt 37 un 59 datu elementu. Rēķinot absolūtajos skaitļos, ir redzams, ka NN-2 korekti klasificē 90.5% elementu, potenciālo funkciju metode un NN-4 - 89% elementu, NN-1 un NN-3 - 85.7%, MDA - 82.5%. Var konstatēt, ka dotajai datu izlasei neironu tīklu metode efektīvāk veic bankrotu datu diagnostiku, kas faktiski atbilst iepriekš izteiktajiem: secinājumiem par Tama un Kianga rezultātiem.

Sestajā apakšnodaļā izdarīti secinājumi par neironu tīklu iespējām datu analīzes uzdevumos.

**Ceturtajā nodaļā** ir pētītas dažādu klasterizācijas metožu iespējas datu analīzē.

Pirmajā apakšnodaļā dots uzdevuma formulējums - izpētīt dažādu klasterizācijas metožu iespējas un noskaidrot to spēju iegūt IF-THEN likumus.

Otrajā apakšnodaļā tiek turpināts populārāko klasterizācijas algoritmu apskats, kas tiek izmantoti datu analīzē. Starp hierarhiskajiem klasterizācijas algoritmiem ievērtību izpelnījies taksonomijas algoritms FOREL. Tāpēc analizēta šī algoritma būtība un darbības princips, demonstrējot to uz IRIS datu bāzes piemēra.

Pēdējā laikā ievērtību izpelnījies jauns klasterizācijas algoritms - režģa klasterizācija. Visi tradicionālie klasterizācijas algoritmi skaitļo "attālumu" metriku (Eiklīda vai citu) starp tēliem un klasteru centriem. Tēli tiek klasterizēti saskaņā ar šo metriku. Režģa klasterizācijas algoritms ir atšķirīgs no citiem ar to, ka tas neorganizē tēlus, bet gan vērtību telpu, kas raksturo tēlu. Organizējot šādu vērtību telpu, tiek izmantota daudzdimensiju struktūra, ko sauc par režģa failu. Tas viss kopā tiek dēvēts par režģa struktūru. Tēli tiek apskatīti kā punkti n-dimensiju vērtību telpā un gadījuma veidā tiek ievietoti režģa struktūrā. Punkti tiek ievietoti saskaņā ar to tēlu vērtībām. Režģa struktūras sadala vērtību telpu taisnstūrveida formas blokos. Tiek dots režģa klasterizācijas algoritms un demonstrācijas piemērs.

Apakšnodaļā apskatīta vēl viena klasterizācijas metode - subtraktīvā klasterizācija, kas balstās uz datu punktu blīvuma mēra jēdzienu. Pazīmju telpā ir jāatrod reģioni ar lielāko punktu blīvuma vērtību. Punkts ar vislielāko kaimiņu skaitu tiek izvēlēts par klastera centru. Tiek specificēts rādiuss un visi punkti, kas atrodas klastera iekšienē, tiek novākti, un algoritms meklē jaunu punktu ar vislielāko kaimiņu skaitu. Tas turpinās tik ilgi, kamēr visi datu punkti tiek pārbaudīti. Tālāk tiek definēts punktu blīvuma mērs un aprakstīts algoritma darbības princips. Algoritma darbības demonstrēšanai tiek izmantota eksperimentāla datu izlase.

Trešajā apakšnodaļā uzmanība tiek pievērsta izplūdušajai klasterizācijai. Pamatoti būtu pieņemt, ka punkti starp diviem klasteru centriem nosaka piederības attiecības starp šiem klasteriem. Izplūdušais *C-vidējā* algoritms uzdod katram punktam, kas pieder klasterim, noteiktu piederības pakāpi un tādējādi nodala punktus starp dažādiem klasteriem. Algoritms sadala  $K$  datu punktus ar specificētiem piederības matricas vektoriem  $u_k$  ( $k=1,2,\dots,K$ ) izplūdušajos  $c$  klasteros un, minimizējot piederības funkciju, atrod katra klastera centru. Izplūdušais *C-vidējā* algoritms (FCM) atšķiras no klasiskā *C-vidējā* algoritma galvenokārt ar to, ka tas izmanto izplūdušu sadalījumu, kad klasteru punkti tiek nodalīti pēc piederības pakāpes. Piederības matricā  $M$  tiek izmantoti skaitļi intervālā  $[0,1]$ , Punktu kopējai piederībai visiem klasteriem jāpiemīt īpašībai, ka jebkuras kolonnas elementu summa ir vienāda ar 1 un visu elementu summa ir vienāda ar  $K$ , un mērķa funkciju uzdod šādi:

$$J(M, c_1, c_2, \dots, c_c) = \sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{k=1}^K m_{ik}^q d_{ik}^2, \quad (2)$$

Kur  $m_{ik}$  - piederības skaitļi starp 0 un 1;

$c_i$  - izplūdušā klastera  $i$  centrs;

$d_{ik} = \|u_k - c_i\|$  - Eiklīda attālums starp  $i$  klastera centru un  $k$  punktu

$q \in (1, \infty)$  - svara rādītājs.

Pastāv divi nosacījumi, lai mērķa funkcija (2) sasniegtu minimumu :

$$c_i = \frac{\sum_{k=1}^K m_{ik}^q u_k}{\sum_{k=1}^K m_{ik}^q} \quad (3)$$

un

$$m_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}} \quad (4)$$

S.Jangs piedāvāja FCM algoritmu, kas nosaka klasteru centrus  $c$ , un piederības matricu  $M$ , izdarot noteiktu soļu secību:

- (1) inicializē piederības matricu  $M$  ar gadījumvērtībām starp 0 un 1.
- (2) izskaitļo klasteru centrus  $c_i$  ( $i=1, 2, \dots, c$ ), izmantojot formulu (3).
- (3) izskaitļo mērķa funkciju saskaņā ar (2). Ja sasniegts apstāšanās nosacījums - beigt darbu, pretējā gadījumā - turpināt.
- (4) izskaitļo jauno piederības matricu  $M$ , izmantojot (4).
- (5) pāriet uz soli (2).

Algoritms ir iteratīvs, un nav garantijas, ka tas konverģēs uz optimālo risinājumu. Izpilde atkarīga no sākotnējās klasteru centru izvēles un piederības matricas  $M$  sākuma inicializācijas. Tālāk tiek dota FCM algoritma demonstrācija ar eksperimentālu datu izlasi.

Ceturtajā apakšnodaļā ir apskatīta likumu iegūšana ar izplūdušās klasterizācijas palīdzību. Šeit notiek mēģinājums paplašināt FCM algoritma pielietojuma sfēru un iegūt datus raksturojošās likumsakarības likumu veidā.

Piederības funkcijas parasti mēdz izteikt ar lingvistisko mainīgo palīdzību, kas ļauj vieglāk interpretēt izplūdušās sistēmas. Tādi jēdzieni kā "augsts", "zems" vai "vidējs" var labi raksturot viendimensijas objektu stāvokli. Daudzdimensiju objektu gadījumā tss vairs nav tik vienkārši. Tiek pieņemts, ka ir  $p$  mainīgie  $x_1, x_2, \dots, x_p$ , kas definēti apgabālā  $X_i = [a_i, b_i]$ ,  $a_i < b_i$ , ir dota galīga klašu kopa  $C$ , kam spēkā sadalījums:

$$\text{klase: } X_1 \times X_2 \times \dots \times X_p \rightarrow C.$$

Mērķis ir atrast izplūdušu klasifikatoru, kas atrisina klasifikācijas problēmu. Izplūdušais klasifikators balstās uz galīgu likumu  $R$  kopu, kam izpildās:

$$R: \text{If } x_1 \text{ ir } \mu_R^{(1)} \text{ un } \dots \text{ un } x_p \text{ ir } \mu_R^{(p)} \text{ THEN klase ir } C_R$$

kur  $C_R \in C$ .  $\mu_R^{(i)}$  ir  $X_i$  izplūdušā kopa, t.i.,  $\mu_R^{(i)}: X_i \rightarrow [0,1]$ . Izplūdušās kopas  $\mu_R^{(i)}$  tiek tiešā veidā iekļautas likumā. Reālajā situācijā tās var tikt aizvietotas ar piemērotiem lingvistiskajiem mainīgajiem.

Daudzu praktisku pielietojumu risināšanā informāciju, kas nepieciešama izplūdušās sistēmas izstrādei un realizācijai, var iedalīt divos veidos: skaitliskā (mērījumu rezultātā) un lingvistiskā (no ekspertiem). Lielākā daļa izplūdušo sistēmu tiek realizētas ar otrā veida iegūtajiem rezultātiem, kas pārsvarā tiek izteikti izplūdušu likumu bāzes veidā. Kā jauninājums tiek piedāvāta autora izstrādātā metode likumu bāzes konstruēšanā no skaitliskajiem datiem ar izplūdušās klasterizācijas palīdzību.

Uzskatāmības labad pieņem, ka tiek veidota izplūdušā sistēma ar divām ieejām ("ieejas signāliem") un vienu izeju. Tādējādi ir nepieciešami šādi apmācošie dati formā:

$$(x_1(i), x_2(i), d(i)), i=1,2,\dots$$

kur  $x_1(i)$ ,  $x_2(i)$  - sistēmā ienākošie dati un  $d(i)$  - gaidāmā izeja. Sistēmas uzdevums ir veidot izplūdušus likumus, lai izejā tiktu iegūts pēc iespējas precīzāks rezultāts. Doto uzdevumu var veikt piecos etapos.

**Pirmais etaps** - ieejas un izejas datu nodalīšana. Tā kā faktiski ir zināmas ieejas datu minimālās un maksimālās vērtības, tad tiek noteikti intervāli, kuros atrodas pieļaujamās vērtības:  $[x_1^-, x_1^+]$ ,  $[x_2^-, x_2^+]$  un  $[d^-, d^+]$ . Katrs no intervāliem tiek sadalīts  $(2N+1)$  daļās ( $N$  vērtība katrā intervālā tiek uzdota pēc nepieciešamības). Atsevišķām intervāla daļām var uzdot lingvistiskos mainīgos, piemēram,  $M_N$  (mazais  $N$ ), ...,  $M_1$  (mazais 1),  $V$  (vidējais),  $L_1$  (lielais 1), ...,  $L_N$  (lielais  $N$ ) un noteikt to piederības funkcijas. 14. attēlā parādīts tamlīdzīga sadalījuma piemērs, kur signāla  $x_1$  definīcijas apgabals sadalīts 5 apakšintervālos ( $N=2$ ), signāla  $x_2$  definīcijas apgabals sadalīts 7 apakšintervālos ( $N=3$ ) un izejas signāla definīcijas apgabals - 5 apakšintervālos ( $N=2$ ).

**Otrais etaps** - izplūdušu likumu konstruēšana no apmācošās kopas datiem. Jānosaka apmācošo datu  $(x_1(i), x_2(i))$  un  $d(i)$  piederības pakāpes katram no izvēlētajiem apgabaliem. Tas tiek izteikts ar piederības funkciju vērtībām. Piemēram, 14. attēlā  $x_1(1)$  piederības pakāpe apgabalam  $L_1$  ir 0.8, piederība apgabalam  $L_2$  - 0.2, piederība citiem apgabaliem - 0. Analogiski  $x_2(2)$  piederības pakāpe apgabalam  $V$  ir 1 un piederība citiem apgabaliem - 0. Līdzīgā veidā attiecina  $x_1(i)$ ,  $x_2(i)$  un  $d(i)$  uz apgabaliem, kur tiem ir maksimālās piederības pakāpes ( $x_1(1)$  ir lielākā piederības pakāpe apgabalam  $L_1$ , bet  $x_2(2)$  - apgabalam  $V$ ). Tādējādi katram apmācošo datu pārim var uzdot vienu likumu:

$$(x_1(1), x_2(1); d(1)) \rightarrow$$

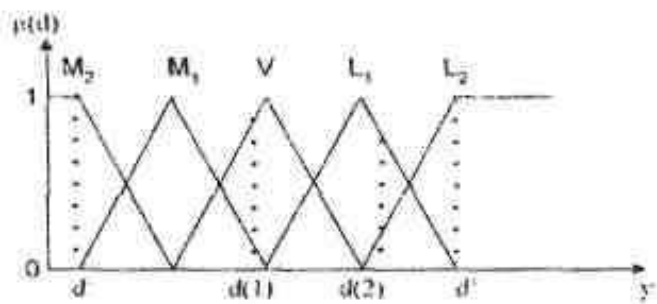
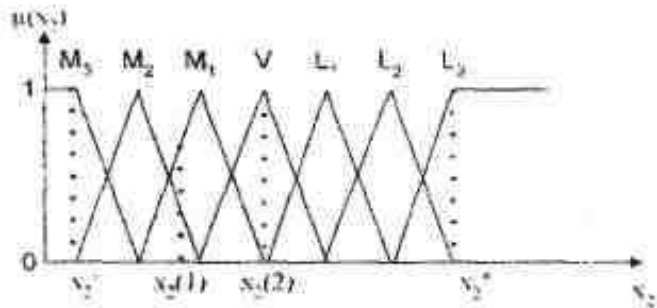
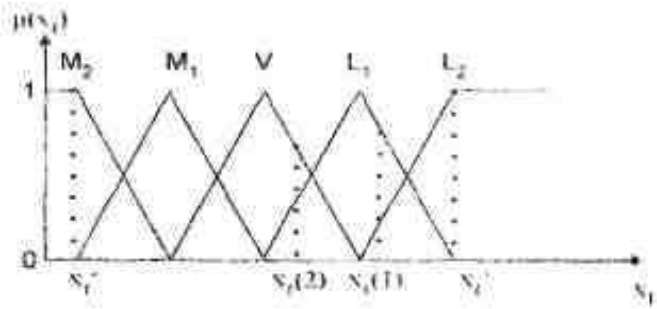
$$\{x_1(1)[\text{max:0.8 apgabālā } L_1], x_2(1)[\text{max:0.6 apgabālā } M_1]; d(1)[\text{max:0.9 apgabālā } V]\} \rightarrow$$

$$R^1: \text{IF } (x_1 \text{ ir } L_1 \text{ AND } x_2 \text{ is } M_1) \text{ THEN } y \text{ ir } M.$$

$$(x_1(2), x_2(2); d(2)) \rightarrow$$

$$\{x_1(2)[\text{max:0.7 apgabālā } V], x_2(2)[\text{max:1.6 apgabālā } V]; d(2)[\text{max:0.7 apgabālā } L_1]\} \rightarrow$$

$$R^2: \text{IF } (x_1 \text{ ir } V \text{ AND } x_2 \text{ is } V) \text{ THEN } y \text{ ir } L_1.$$



14. att. Ieejas un izejas datu sadalīšana intervālos un attiecīgas piederības funkcijas

**Trešais etaps-** ticamības pakāpes noteikšana katram likumam. Protams, ka ir liels skaits apmācošo datu pāru un katram no tiem var tikt veidots viens likums, tāpēc pastāv iespēja, ka likumi var būt pretrunīgi. Tas attiecas uz likumiem ar vienu un to pašu nosacījumu, bet dažādiem secinājumiem. Viena no iespējām šīs problēmas risināšanā ir katram likumam piešķirt ticamības pakāpi ar nolūku turpmāk izvēlēties likumu ar visaugstāko ticamības pakāpi. Tādā veidā ne tikai atrisināsies likumu pretrunības problēma, bet arī būtiski samazināsies kopējais likumu skaits.

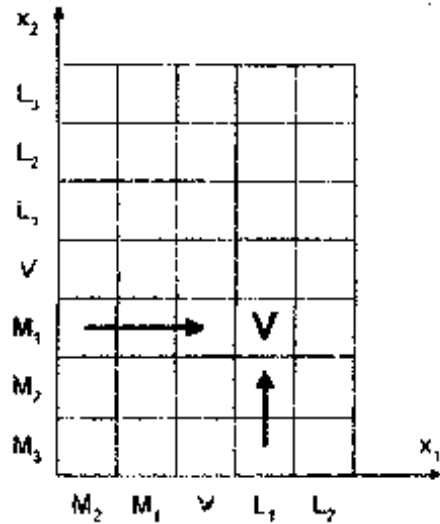
Likumam R: IF ( $x_1$  ir  $A_1$  AND  $x_2$  ir  $A_2$ ) THEN ( $y$  ir B) ticamības pakāpi SP(R) nosaka šādi:

$$SP(R) = \mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \mu_B(y).$$

**Ceturtais etaps-** izplūdušo likumu bāzes izveidošana. Izplūdušo likumu bāzes uzdošanas princips parādīts 15. attēlā. Likumu bāze tiek uzdots tabulas veidā, kas tiek aizpildīta ar likumiem šādi: ja likums ir uzdots formā

$$R^1: \text{IF } (x_1 \text{ is } L_1 \text{ AND } x_2 \text{ is } M_1) \text{ THEN } y \text{ is } V,$$

tad kolonnas  $L_1$  un rindas  $M_1$  krustpunktā ieraksta izplūdušas kopas vērtību, kas ietilpst likuma THEN daļā, t.i., šajā piemērā V vērtību. Ja eksistē vairāki likumi ar vienu un to pašu nosacījumu, tad no tiem izvēlas likumu ar lielāko ticamības pakāpi.



15. att. Izplūdušo likumu bāzes veids

**Piektais etaps** - defazifikācija. Ar iegūtas likumu bāzes palīdzību ir jāiegūst attēlojums  $f: (x_1, x_2) \rightarrow \bar{y}$ , kur  $\bar{y}$  ir izplūdušas sistēmas izejas vērtība. Defazifikācija ir veikta, ja katram lingvistiskajam mainīgajam tiek iegūta kāda konkrēta vērtība. Lai to izdarītu, vispirms tiek izskaitļota  $k$  likuma aktivitātes pakāpe pēc formulas.

$\tau^{(k)} = \mu_{A_1^{(k)}}(x_1) \cdot \mu_{A_2^{(k)}}(x_2)$  Faktiski tiek noskaidrots, kurš no iegūtajiem likumiem ir vairāk aktīvs konkrētajam ieejas datu vektoram.

Likumam  $R^1$  no iepriekš dotā piemēra aktivitātes pakāpe ir  $\tau^{(1)} = \mu_{L_1}(x_1) \cdot \mu_{M_1}(x_2)$ .

Pēc tam izejas vērtības  $\bar{y}$  noteikšanai pielieto kādu no metodēm smaguma centra noteikšanai, piemēram, defazifikāciju pēc smagumu centra metodes (COGS):

$$\bar{y} = \frac{\sum_{k=1}^N \tau^{(k)} \bar{y}^{(k)}}{\sum_{k=1}^N \tau^{(k)}}$$

Pēc visu piecu etapu sekmīgas veikšanas var uzskatīt, ka ir iegūta izplūdušo likumu bāze.

Tālāk darbā ir dots likumu bāzes iegūšanas pielietojums IRIS datiem. Eksperimenta mērķis:

- iegūt likumus no IRIS datu bāzes, izmantojot FCM algoritmu;
- noskaidrot piederības funkciju skaita ietekmi uz iegūto likumu skaitu;
- pārbaudīt iegūto likumu pareizību.

Pirmajā eksperimenta daļā tika izskaitļotas 3 piederības funkcijas 3 klasteriem un iegūti 4 likumi I klasei, 3 likumi II klasei un 11 likumi III klasei. Iegūtie likumi parādīti 8. tabulā.

Otrajā eksperimenta daļā tika izvēlētas dažādas piederības funkciju sākumvērtības un tika veikta likumu ieguve. Rezultāti parādīti 9. tabulā.

Trešajā eksperimenta daļā tika pārbaudīta iegūto likumu pareizība. Veikto eksperimentu rezultātā tika konstatēts, ka konkrētajai klasei iegūtie likumi korekti apraksta klases elementus, t.i., jebkuram datu vektoram derēs vismaz viens iegūtais likums, kas iekļauj šo datu vektoru atbilstošajā klasē. Palielinot piederības funkciju skaitu, kā tas ir redzams 9. tabulā, var iegūt pietiekami daudz likumu, kas vēl rūpīgāk apraksta datus.

## Iegūtie likumi no IRIS datu bāzes

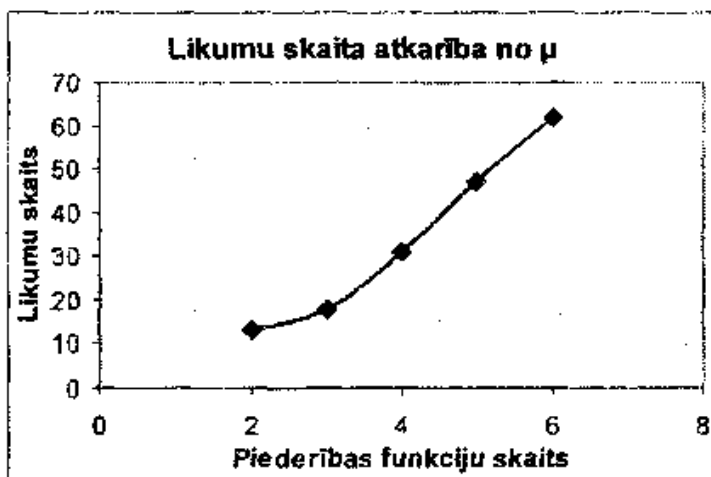
1. likums: IF X1 ir MF1 ar piederību 1 AND X2 ir MF2 ar piederību 0.83333 AND X3 ir MF1 ar piederību 0.9661 AND X4 ir MF1 ar piederību 1 THEN KLAŠE ir 1 ar piederību 0.76483
2. likums: IF X1 ir MF2 ar piederību 0.7778 AND X2 ir MF3 ar piederību 1 AND X3 ir MF1 ar piederību 0.83051 AND X4 ir MF1 ar piederību 0.75 THEN KLAŠE ir 1 ar piederību 0.46024
3. likums: IF X1 ir MF2 ar piederību 0.66667 AND X2 ir MF2 ar piederību 0.75 AND X3 ir MF1 ar piederību 0.89831 AND X4 ir MF1 ar piederību 0.91667 THEN KLAŠE ir 1 ar piederību 0.39114
4. likums: IF X1 ir MF1 ar piederību 0.88889 AND X2 ir MF1 ar piederību 0.75 AND X3 ir MF1 ar piederību 0.89831 AND X4 ir MF1 ar piederību 0.83333 THEN KLAŠE ir 1 ar piederību 0.47411
5. likums: IF X1 ir MF2 ar piederību 1 AND X2 ir MF2 ar piederību 0.66667 AND X3 ir MF2 ar piederību 0.98305 AND X4 ir MF2 ar piederību 1 THEN KLAŠE ir 2 ar piederību 0.6226
6. likums: IF X1 ir MF2 ar piederību 0.94444 AND X2 ir MF1 ar piederību 0.83333 AND X3 ir MF2 ar piederību 0.98305 AND X4 ir MF2 ar piederību 0.75 THEN KLAŠE ir 2 ar piederību 0.55126
7. likums: IF X1 ir MF1 ar piederību 0.61111 AND X2 ir MF1 ar piederību 1 AND X3 ir MF2 ar piederību 0.84746 AND X4 ir MF2 ar piederību 0.75 THEN KLAŠE ir 2 ar piederību 0.369
8. likums: IF X1 ir MF2 ar piederību 0.88889 AND X2 ir MF2 ar piederību 0.91667 AND X3 ir MF3 ar piederību 0.69492 AND X4 ir MF3 ar piederību 1 THEN KLAŠE ir 3 ar piederību 0.53792
9. likums: IF X1 ir MF2 ar piederību 1 AND X2 ir MF2 ar piederību 0.83333 AND X3 ir MF2 ar piederību 0.67797 AND X4 ir MF2 ar piederību 0.58333 THEN KLAŠE ir 3 ar piederību 0.31309
10. likums: IF X1 ir MF3 ar piederību 0.88889 AND X2 ir MF2 ar piederību 0.83333 AND X3 ir MF3 ar piederību 0.72881 AND X4 ir MF3 ar piederību 0.83333 THEN KLAŠE ir 3 ar piederību 0.42739
11. likums: IF X1 ir MF2 ar piederību 0.83333 AND X2 ir MF2 ar piederību 0.91667 AND X3 ir MF3 ar piederību 0.52542 AND X4 ir MF2 ar piederību 0.58333 THEN KLAŠE ir 3 ar piederību 0.22242
12. likums: IF X1 ir MF1 ar piederību 0.66667 AND X2 ir MF1 ar piederību 0.58333 AND X3 ir MF2 ar piederību 0.81356 AND X4 ir MF2 ar piederību 0.66667 THEN KLAŠE ir 3 ar piederību 0.20038
13. likums: IF X1 ir MF3 ar piederību 0.61111 AND X2 ir MF2 ar piederību 1 AND X3 ir MF3 ar piederību 0.69492 AND X4 ir MF2 ar piederību 0.58333 THEN KLAŠE ir 3 ar piederību 0.23534
14. likums: IF X1 ir MF2 ar piederību 1 AND X2 ir MF1 ar piederību 0.5 AND X3 ir MF3 ar piederību 0.55932 AND X4 ir MF2 ar piederību 0.91667 THEN KLAŠE ir 3 ar piederību 0.24354
15. likums: IF X1 ir MF2 ar piederību 0.83333 AND X2 ir MF2 ar piederību 1 AND X3 ir MF2 ar piederību 0.54237 AND X4 ir MF3 ar piederību 0.83333 THEN KLAŠE ir 3 ar piederību 0.35782
16. likums: IF X1 ir MF2 ar piederību 0.7778 AND X2 ir MF1 ar piederību 0.58333 AND X3 ir MF2 ar piederību 0.64407 AND X4 ir MF3 ar piederību 0.58333 THEN KLAŠE ir 3 ar piederību 0.16194
17. likums: IF X1 ir MF3 ar piederību 0.88889 AND X2 ir MF1 ar piederību 0.5 AND X3 ir MF3 ar piederību 1 AND X4 ir MF3 ar piederību 0.83333 THEN KLAŠE ir 3 ar piederību 0.35185
18. likums: IF X1 ir MF2 ar piederību 0.94444 AND X2 ir MF1 ar piederību 0.83333 AND X3 ir MF2 ar piederību 0.64407 AND X4 ir MF2 ar piederību 0.83333 THEN KLAŠE ir 3 ar piederību 0.4013

Iegūto likumu skaita atkarība no sākotnēji uzdoto piederības funkciju skaita

Piederības funkciju skaits	I klase	II Klase	III klase	Likumi kopā
2	2	8	3	13
3	4	3	11	18
4	7	11	13	31
5	13	16	18	47
6	21	19	22	62

Grafiskā veidā atkarība parādīta 16. attēla.



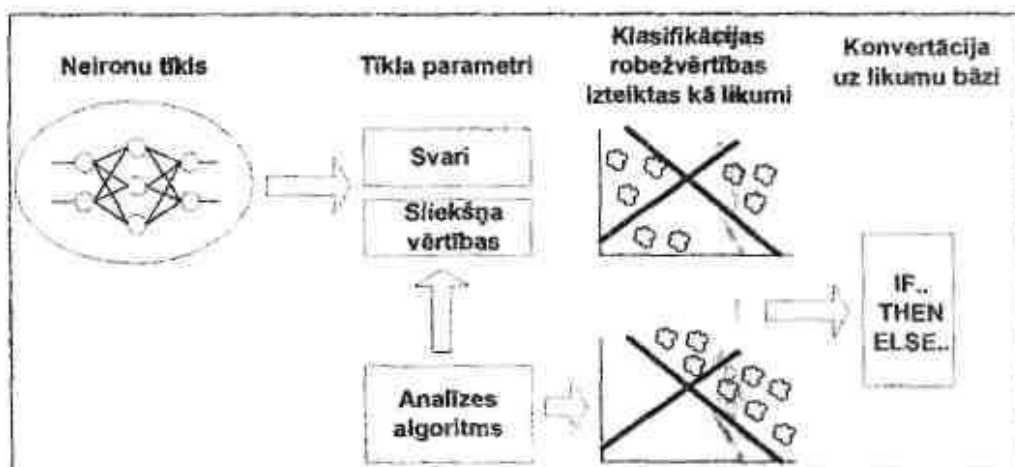


16. att. Likumu skaita atkarības no piederības funkciju skaita grafiks

Piektajā apakšnodaļā izdarīti secinājumi par klasterizācijas algoritmu pielietojumiem. Klasterizācijas algoritmi pārsvarā paredzēti daudzdimensiju statistisko datu apstrādei, kad dati uzdoti tabulas veidā "objekts-īpašība". Tie ļauj grupēt objektus noteiktās grupās, kurās objekti saistīti savā starpā pēc kāda konkrēta nosacījuma. Nav svarīgi, kā tiek dēvētas šādas grupas - taksoni, klasteri, klases, galvenais, ka tās pietiekoši precīzi atspoguļo to objektu īpašības.

**Piektajā nodaļā** apskatītas neironu tīklu metodes nosacīto likumu iegūšanas procesā.

Pirmajā apakšnodaļā dots problēmas formulējums un veikta iepriekšējo pētījumu analīze. Izmantojot neironu tīklu tehniku, var papildināt risināmo uzdevumu klasi, paplašinot neironu tīklu iespējas. Ievērtību ir ieguvuši pētījumi, kas ļauj „lasīt simbolisko jeb lingvistisko” informāciju no mākslīgajiem neironu tīkliem. Pētījumi liecina, ka iespēja tiešā veidā pārveidot neironu tīkla informāciju simboliskā zināšanu veidā notiek caur likumu iegūvi. Šis process paredz dot skaidrojumu, kādā veidā neironu tīkls klasificē dotos ieejas datus. Likumu iegūve ir process, kas ļauj atrast pieņēmumus par neironu tīkla darbu no „ieejas-uz-slēptajiem” elementiem līdz pat no „slēptajiem-uz-izejas” elementiem. Tādi pieņēmumi tiek formulēti kā IF-THEN likumi, kuros likuma nosacījumu daļas loma ir atvēlēta būtiskāko ieejas elementu aktivitātes līmenim. Likumu iegūves no neironu tīkliem vispārīgā shēma ir parādīta 17. attēlā.



17. att. Likumu iegūves procesa shēma

Tālāk īsumā ir aprakstītas dažas pieejas likumu ieguvē no neironu tīkliem: VIA, N-of-M un KBAN. Likumu ieguves no neironu tīkliem priekšrocības ir šādas:

- zināšanas, kas tiek iegūtas neironu tīklu apmācības procesā, vispārīgā veidā cilvēkam ir grūti uztveramas. Metožu izstrādāšana, kas ļautu interpretēt tīkla ieejas -izejas attiecības likumu veidā, ir ļoti lietderīga;
- var tikt identificētas nepilnības apmācošajos datos, līdz ar to tīkla darbības funkcionēšanu var uzlabot ar kļāšu modificēšanu;
- iespējama iepriekš nezināmu likumsakarību noteikšana.

Otrajā apakšnodaļā iztirzāta neironu tīklu piemērotība likumu ieguvei. Tipisks neironu tīkls satur ļoti daudz to raksturojošo vērtību parametru. Šie parametri uzdod attiecības starp datu ieejas vektoriem  $x$  un izejas vērtību  $y$ . Lai gan parametru uzdošanas būtība dažādiem apmācības algoritmiem ir saprotama, tomēr lielais skaits tipiska tīkla parametru padara tīkla būtības izprašanas uzdevumu par ļoti sarežģītu. Turklāt daudzslāņu tīklos šādi parametri raksturo daudzveidīgas attiecības starp datu ieejas vektoriem un vēlamajām izejas vērtībām. Šādos gadījumos praktiski nav iespējams noteikt konkrētā ieejas vektora ietekmi uz izejas vērtību, jo ietekme izpaužas kontekstā ar citu ieejas vektoru vērtībām.

Tādas attiecības tiek uzdotas ar tīkla slēptā slāņa elementiem, kas noteiktā veidā kombinē daudzus ieejas vektorus, iegūstot to raksturojošās vērtības. Tika izvirzīta hipotēze, ka tieši ar slēptā slāņa elementu palīdzību varētu raksturot ieejas datu savstarpējās attiecības.

Trešā apakšnodaļa vēltīta likumu ieguves metodoloģijai no RBF neironu tīkliem. RBF neironu tīkla raksturs padara to par ļoti piemērotu instrumentu likumu ieguves procesā. Trešās nodaļas trešajā apakšnodaļā tika aprakstīta RBF neironu tīkla apmācības vispārīgā shēma un parādīts, ka RBF tīkla apmācība tradicionāli sastāv no diviem soļiem.

Pirmajā solī tiek pozicionēti RBF centri (slēptā slāņa elementi) un bāzes funkcijas izmērs. Otrajā apmācības solī tiek skaitļoti svāri no slēptajiem elementiem uz izeju un izejas reakcija. Radiālās bāzes funkcijas slēptajā slānī ir realizētas ar kodola funkcijām, kuru darbības diapazons tiek noteikts ar bāzes funkcijas centra koordināšu vērtībām un bāzes funkcijas platumu jeb rādiusu.

$$Z_j(x) = \exp\left(-\frac{\|x - \mu\|^2}{\sigma_j^2}\right) \quad (5)$$

RBF tīkla izejas vērtības tiek izskaitlotas saskaņā ar vienādojumu:

$$y = \sum_{j=1}^J W_{ij} Z_j(x), \quad (6)$$

Jau tika norādīts, ka visbiežāk par bāzes funkciju izmanto Gausa funkciju, kuras izteiksmi pārraksta šādā veidā:

kur:  $W$  - svāru matrica;  
 $Z$  - slēpto elementu aktivizācijas;  
 $x$  - ieejas vektors;  
 $\mu$  - bāzes funkcijas centrs;  
 $\sigma$  - bāzes funkcijas platumas.

Turpmāk RBF tīkls tiks izmantots likumu ieguves kontekstā. Likumu ieguvē tiek izmantota tā slēpto elementu īpašība, ka katrs slēptais elements pēc apmācības faktiski reprezentē vien elementu klasi. RBF slēptā elementa lokālais raksturs ļauj veikt pārveidojumu uz vienkāršu likumu

```
IF Feature1 is TRUE AND
IF Feature2 is TRUE AND
IF Featuren is TRUE
THEN Classx
```

Kompleksais elements *Feature* sastāv no:

- izskaitļotajām RBF centru  $\mu_n$  augšējām un apakšējām robežvērtībām;
- RBF platuma jeb rādiusa  $\sigma$ ;
- Gausa funkcijas parametra  $S$ .

$S$  vērtība ir noteikta empīriski un ir atkarīga no platuma parametra. Mainīgo  $\mu$  un  $\sigma$  vērtības ir noteiktas pēc RBF apmācības algoritma. Augšējā un apakšējā robeža tiek izskaitļotas pēc formulas.

$$X_{\text{lower}} = \mu_i - \sigma_i + S \quad \text{un} \quad X_{\text{upper}} = \mu_i + \sigma_i - S. \quad (7)$$

RBF likumu ieguvei tiek izmantots algoritms RULEX (sk. 18.att.).

Ieeja:	RBF centra vērtības $\mu$ Gausa funkcijas platums $\sigma$ Parametrs $S$
Izeja:	Viens likums katram slēptajam elementam
Procedūra:	RBF tīkla apmācības procedūra Katram slēptajam elementam: Katram $\mu_i$ $X_{\text{lower}} = \mu_i - \sigma_i + S$ $X_{\text{upper}} = \mu_i + \sigma_i - S$ Konstruē likumu ar: nosacījumu daļā $[X_{\text{lower}}, X_{\text{upper}}]$ Apvieno nosacījumu daļas ar AND Pieliek klases iezīmi Ieraksta likumu.

18. att. Likumu ieguves algoritms RULEX

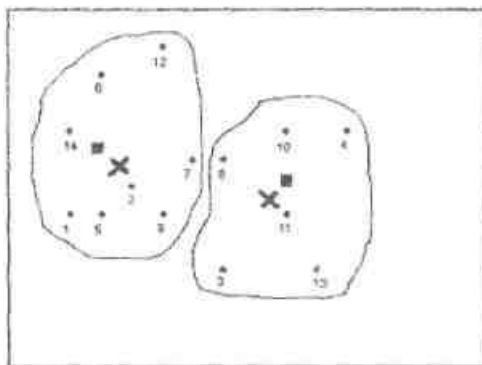
Tālāk tiek dots demonstrācijas piemērs likumu ieguves principa izprašanai. Demonstrācijas nolūkā tika izmantota divdimensiju datu izlase, kas parādīta 10. tabulā.

10. tabula

RBF tīkla ieejas vektori

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$X_1$	1	3	6	10	2	2	5	6	4	8	8	4	9	1
$X_2$	3	4	1	6	3	8	5	5	3	6	3	9	1	6

Pēc RBF tīkla apmācības algoritma pielietošanas tika iegūti 2 klasteri (sk. 19. att.)



19.att. Iegūtie klasteri ar centriem  $(-0.73; 0.26)$  un  $(0.97; -0.35)$

Tika iegūti svaru vektori  $\mu_1=(-0.73; 0.26)$ ,  $\mu_2=(0.97;-0.35)$  un izskaitļotas klasteriem atbilstošās rādiusu vērtības  $\sigma_1^2 = 1.07$  un  $\sigma_2^2 = 1.04$ . Apmācības otrajā etapā, izmantojot formulas (5) un (6), tika izskaitļotas radiālās funkcijas un tīkla izeja. RBF neironu tīkls ir apmācīts.

Turpmākajā RULEX algoritma pielietošanas gaitā, izmantojot izteiksmi (7), katram klasterim izskaitļo  $X_{lower}$  and  $X_{upper}$  vērtības:

$$\begin{aligned} \text{Class 1.} \quad X_{1\_lower} &= -0.73 - 1.03 + 0.6 = -1.16; & X_{2\_lower} &= 0.26 - 1.03 + 0.6 = -0.17; \\ X_{1\_upper} &= -0.73 + 1.03 - 0.6 = -0.3; & X_{2\_upper} &= 0.26 + 1.03 - 0.6 = 0.69. \\ \text{Class 2.} \quad X_{1\_lower} &= 0.97 - 1.01 + 0.6 = 0.56; & X_{2\_lower} &= -0.35 - 1.01 + 0.6 = -0.76; \\ X_{1\_upper} &= 0.97 + 1.01 - 0.6 = 1.38; & X_{2\_upper} &= -0.35 + 1.01 - 0.6 = 0.06. \end{aligned}$$

Tādējādi katram slēptajam neironam, kas reprezentē konkrētu klasi, ir iegūti šādi likumi:

IF ( $x_1 \geq -1.16$  AND  $\leq -0.3$ ) AND IF ( $x_2 \geq -0.17$  AND  $\leq 0.69$ ) THEN CLASS 1.

IF ( $x_1 \geq 0.56$  AND  $\leq 1.38$ ) AND IF ( $x_2 \geq -0.76$  AND  $\leq 0.06$ ) THEN CLASS 2.

Analizējot iegūtos likumus, var spriest, ka:

- pie parametra vērtības  $S=0.6$  iegūtie likumi nepareizi apraksta 12 ieejas vektorus no 14 (86% kļūdaini).

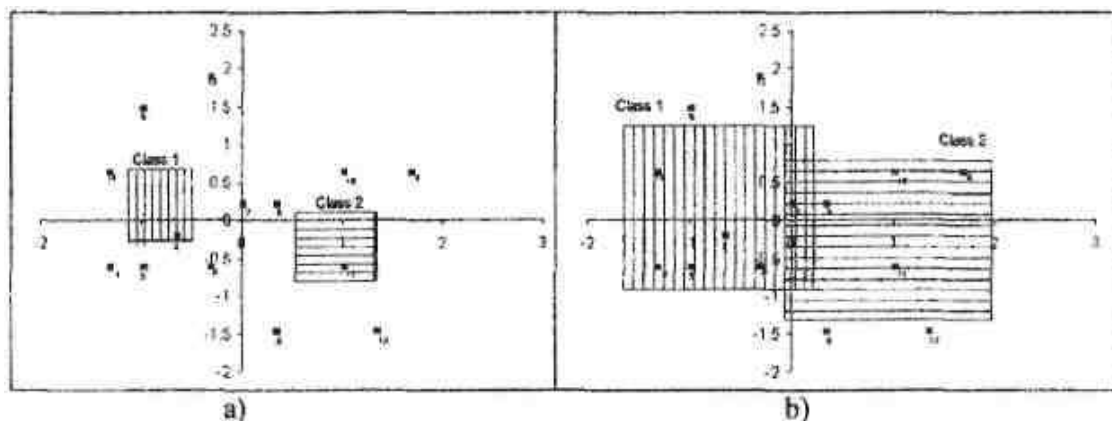
- pie parametra vērtības  $S=0.2$ , iegūtie likumi nepareizi apraksta 9 ieejas vektorus no 14 (64% kļūdaini).

- pie parametra vērtības  $S=0$ , iegūtie likumi nepareizi apraksta 4 ieejas vektorus no 14 (28% kļūdaini), (19. attēlā tie ir punkti 3,6,12 un 13). Iegūtie likumi šajā gadījumā ir:

IF ( $x_1 \geq -1.76$  AND  $\leq 0.3$ ) AND IF ( $x_2 \geq -0.77$  AND  $\leq 1.29$ ) THEN CLASS 1.

IF ( $x_1 \geq -0.04$  AND  $\leq 1.98$ ) AND IF ( $x_2 \geq -1.36$  AND  $\leq 0.66$ ) THEN CLASS 2.

Iegūto likumu pielietojuma apgabali parādīti 20. attēlā.



20.att. Iegūto likumu pielietojuma apgabali -  $S=0.6$  (a) un  $S=0$  (b)

Var izdarīt secinājumu, ka korektu likumu ieguve no apmācītiem neironu tīkliem ir atkarīga no sākotnēji izvēlēto klasteru skaita, t.i., ja būs vairāk klasteru, tad tiks iegūts vairāk likumu.

Tālāk tika veikti 2 eksperimenti praktiska uzdevuma risināšanā ar bankrotu datu izlasi un IRIS datubāzi. Pirmajā eksperimentā tika izmantoti trešās nodaļas sestajā apakšnodaļā aprakstītie firmu bankrotu dati (46 bankrotējušas un 17 nebankrotējušas firmas), kuru datu fragments parādīts 11. tabulā. Eksperimenta mērķis- iegūt likumus no bankrotu datiem.

11. tabula

Bankrotu datu fragments

Bankrotējošās firmas				Nebankrotējušās firmas			
R3	R7	R9	R31	R3	R7	R9	R31
0.29	1.75	0.67	0	0.01	1.01	0.93	0.04
0.28	1.75	0.65	0.08	0.04	1.04	0.97	0.05
0.37	2.77	0.58	0.08	0.05	1.05	0.95	0.04
0.05	1.13	0.39	0.02	-0.45	0.51	0.47	-0.13
...	...	...	...	...	...	...	...

12. tabulā parādīti eksperimentā iegūtie rezultāti un 13 tabulā - iegūtie likumi pie atsevišķām S vērtībām (I klase satur nebankrotējušo firmu datus, II klase - bankrotējušo firmu datus).

12. tabula

Likumu ieguves no bankrotu datiem rezultāti

Korekti	Parametra S vērtības								
	-0.9	...	0	0.1	0.2	0.3	0.4	0.5	0.6
I	15	...	12	10	9	6	5	1	0
II	44	...	44	44	44	44	44	44	44
%	93.7	...	88.9	85.7	84.1	79.4	77.8	71.4	69.8

13. tabula

No bankrotu datiem iegūto likumu raksturlielumi

	Parametra S vērtības	
	-0.9	0.4
Centru un rādiusu vērtības	1. Kl. centrs= 0.03 1.25 0.74 0.01 2. Kl. centrs= 0.13 1.86 0.59 0.10 Rādiusu vērtības= 0.68 3.92	1. Kl. centrs= 0.03 1.25 0.74 0.01 2. Kl. centrs= 0.13 1.86 0.59 0.10 Rādiusu vērtības= 0.68 3.92
Likumi korekti apraksta klašu elementus (%)	93.7	77.8
1. klases likums	IF (X1) >= -1.54 AND < 1.61 ) AND IF (X2) >= -0.33 AND < 2.83) AND IF (X3) >= -0.84 AND < 2.32) AND IF (X4) >= -1.57 AND < 1.59) THEN NE-BANKROTS	IF (X1) >= -0.24 AND < 0.31 ) AND IF (X2) >= 0.97 AND < 1.53) AND IF (X3) >= 0.46 AND < 1.02) AND IF (X4) >= -0.27 AND < 0.29) THEN NE-BANKROTS
2. klases likums	IF (X1) >= -4.69 AND < 4.95 ) AND IF (X2) >= -2.97 AND < 6.68) AND IF (X3) >= -4.23 AND < 5.41) AND IF (X4) >= -4.72 AND < 4.93) THEN BANKROTS	IF (X1) >= -3.39 AND < 3.65 ) AND IF (X2) >= -1.67 AND < 5.38) AND IF (X3) >= -2.93 AND < 4.11) AND IF (X4) >= -3.42 AND < 3.63) THEN BANKROTS

No 12. tabulas redzams, ka atrastie likumi korekti apraksta bankrotu datus (44 no 46) visā parametra S maiņas apgabalā, t.i., bankrotu dati atrodas pietiekami kompaktā klasē.

Otrajā eksperimentā tika izmantota IRIS datu bāze, kas satur trīs ziedu klases - pa 50 elementiem katrā: *setosa*, *versicolor* un *virginica*. Katram ziedam ir 4 parametri: SL-kauslapas garums, SW-kauslapas platums, PL-ziedlapas garums un PW-ziedlapas platums. 14 tabulā parādīts katras klases datu fragments.

## IRIS datu fragments

<i>Setosa</i>				<i>Versicolor</i>				<i>Virginica</i>			
SL	SW	PL	PW	SL	SW	PL	PW	SL	SW	PL	PW
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
...	...	...	...	...	...	...	...	...	...	...	...

Eksperimenta mērķis:

1) veikt tikla apmācību ar RULEX algoritmu pie dažādām apmācošajām kopām:

- A - katras klases pirmie 25 elementi;
- B - katras klases patvaļīgi 20 elementi;
- C - katras klases visi 50 elementi;

2) izpētīt parametra S ietekmi uz likumu pareizību.

Tālāk tiek analizēti B variantā iegūtie rezultāti, kad par apmācošo kopu tika ņemti katras klases patvaļīgi izvēlēti 20 elementi. 15. tabulā parādīti iegūtie rezultāti un 16. tabulā - iegūtie likumi pie atsevišķām S vērtībām.

15. tabula

B - apmācības kopas rezultāti (katras klases patvaļīgi 20 elementi)

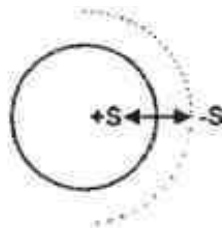
Korekti	Parametra S vērtības													
	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0	0.1	0.2	0.3	0.6
I	49	49	48	48	45	40	39	27	14	9	2	0	0	0
II	50	49	49	48	45	44	40	36	28	20	10	3	0	0
III	49	49	48	47	45	43	43	42	39	35	29	23	16	0
%	98.7	98	96.7	95.3	90	84.7	81.3	70	54	42.7	27.3	17.3	10.7	0

16. tabula

B variantā iegūto likumu raksturlielumi

	Parametra S vērtības	
	-0.9	0
Centru un rādiusu vērtības	1. Kl. centrs= 5.04 3.45 1.49 0.25 2. Kl. centrs= 5.99 2.77 4.32 1.35 3. Kl. centrs= 6.54 2.95 5.44 1.94 Rādiusu vērtības= 0.19 0.43 0.73	1. Kl. centrs= 5.04 3.45 1.49 0.25 2. Kl. centrs= 5.99 2.77 4.32 1.35 3. Kl. centrs= 6.54 2.95 5.44 1.94 Rādiusu vērtības= 0.19 0.43 0.73
Likumi korekti apraksta klašu elementus (%)	98.67	42.67
1. klases likums	IF (X1>= 3.95 AND < 6.13 ) AND IF (X2>= 2.36 AND < 4.54) AND IF (X3>= 0.40 AND < 2.59) AND IF (X4>= -0.84 AND < 1.34) THEN SETOSA	IF (X1>= 4.85 AND < 5.23 ) AND IF (X2>= 3.26 AND < 3.64) AND IF (X3>= 1.30 AND < 1.69) AND IF (X4>= 0.06 AND < 0.44) THEN SETOSA
2. klases likums	IF (X1>= 4.66 AND < 7.32 ) AND IF (X2>= 1.44 AND < 4.10) AND IF (X3>= 2.99 AND < 5.65) AND IF (X4>= 0.01 AND < 2.68) THEN VERSICOLOR	IF (X1>= 5.56 AND < 6.42 ) AND IF (X2>= 2.34 AND < 3.20) AND IF (X3>= 3.89 AND < 4.75) AND IF (X4>= 0.91 AND < 1.78) THEN VERSICOLOR
3. klases likums	IF (X1>= 4.91 AND < 8.17 ) AND IF (X2>= 1.33 AND < 4.58) AND IF (X3>= 3.81 AND < 7.06) AND IF (X4>= 0.31 AND < 3.57) THEN VIRGINICA	IF (X1>= 5.81 AND < 7.27 ) AND IF (X2>= 2.23 AND < 3.68) AND IF (X3>= 4.71 AND < 6.16) AND IF (X4>= 1.21 AND < 2.67) THEN VIRGINICA

Analizējot iegūtos rezultātus, var secināt, ka RULEX algoritma pielietošanā būtiska loma ir parametram  $S$  - jo lielāka negatīvā  $S$  vērtība, jo vairāk samazinās likuma darbības diapazons zemākā robeža  $X_{lower}$ , vienlaikus palielinot diapazona augšējo robežu  $X_{upper}$ . Tas rada klasteri aprakstošās nosacījumu daļas palielināšanos, tādējādi palielinot apgabala vērtību, kādā iegūtais likums izpildās. Shematiski šo efektu var attēlot ar 21. attēlā redzamo attēlu.



21.att. Parametra  $S$  palielināšanas/samazināšanas efekts

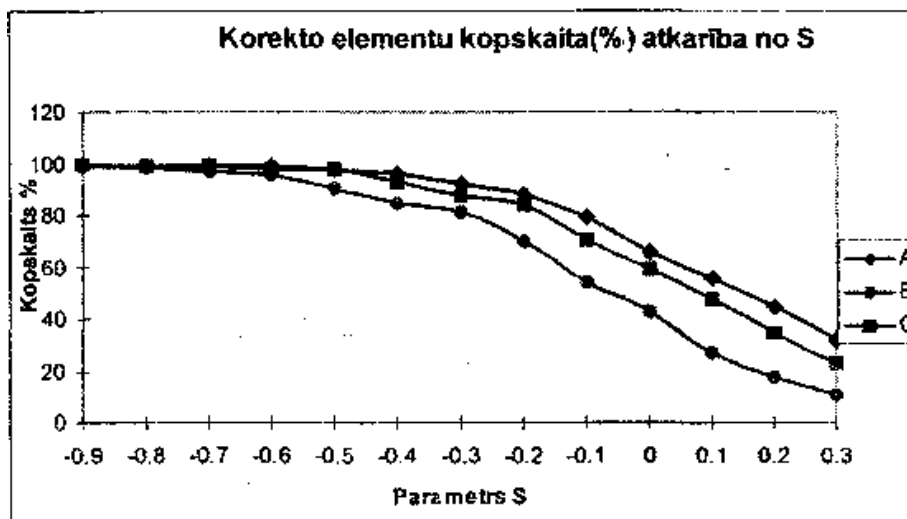
A, B un C apmācošajām izlasēm kopējā korekti likumus aprakstošo elementu skaita atkarība no parametra  $S$  parādīta kopsavilkuma 17.tabulā un 22. attēlā redzamajā grafikā.

22.att. Likumu pareizi aprakstošo elementu skaita atkarība no parametra  $S$

17. tabula

Likumiem atbilstošo elementu kopskaita (%) atkarība no  $S$

%	Parametra $S$ vērtības												
	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0	0.1	0.2	0.3
A	100	99.3	99.3	98.7	97.3	96	92	88	79.3	66	55.3	44.7	32
B	98.7	98	96.7	95.3	90	84.7	81.3	70	54	42.7	27.3	17.3	10.7
C	100	98.7	98.7	97.3	97.3	92.7	87.3	84	70.7	58.7	47.3	34.7	23.3



22.att. Likumu pareizi aprakstošo elementu skaita atkarība no parametra  $S$

Ceturtajā apakšnodaļā izdarīti secinājumi par likumu ieguves no neironu tīkliem lietderību.

**Nobeigumā** tiek aplūkoti promocijas darba rezultāti un izklāstīts tālāko pētījumu virziens.

**Pietikumā** ir demonstrēts autora izstrādātais programmnodrošinājums eksperimentu veikšanai asociāciju likumu iegūšanai no statistiskajiem datiem. Programmnodrošinājuma izstrādei ir izmantota programmu pakete *Matlab*.

## DARBA GALVENIE REZULTĀTI

Promocijas darba galvenais mērķis bija izpētīt un izstrādāt nosacīto likumu veidošanas metodes, kas balstās uz trim likumu iegūšanas paņēmieniem.

Darba izpildes gaitā tika iegūti šādi **rezultāti**:

1) mākslīgo neironu tīklu iespēju pētīšanā:

- izpētīta neironu tīklu darbība, par pamatu ņemot radiālās bāzes funkciju neironu tīklu;
- analizēts RBF tīklu apmācības algoritms, pievēršot uzmanību pirmajam apmācības etapam - ktasterizācijai;
- apskatītas neironu tīklu iespējas tēlu atpazīšanas un optimizācijas uzdevumos;
- realizēti eksperimenti ar nolūku parādīt neironu tīklu izmantošanas iespējas dažādos datu analīzes veikšanas uzdevumos;
- veikta bankrotu datu analīze ar vairāku metožu palīdzību un
  - o kā jauninājums piedāvāta potenciālu funkciju metodes izmantošana, kas dod labus rezultātus bankrotu datu analīzē;
  - o eksperimentāli iegūta neironu tīkla konfigurācija ar konkrētiem apmācības parametriem, pie kuriem tīkla darbības rezultāti adekvāti pētnieku izteiktajam viedoklim par neironu tīklu pielietošanas efektivitāti bankrotu diagnostikā.

2) klasterizācijas metožu pielietošanā:

- izpētīta klasterizācijas algoritmu pielietošanas joma;
- apskatītas un izvērtētas tipiskākās klasterizācijas metodes un to iespējas;
- pamatota klasterizācijas metožu izvēlē likumsakarību modelēšanā;
- realizēti eksperimenti klasterizācijas algoritmu darbības demonstrācijai.

3) izplūdušās klasterizācijas iespēju pētīšanā:

- izstrādāta un iztirzāta izplūdušo likumu bāzes iegūšanas metode no skaitliskajiem datiem;
- kā jauninājums piedāvāta izplūdušās klasterizācijas algoritma FCM pielietošana likumu bāzes iegūšanā;

4) nosacīto likumu iegūšanas no neironu tīkliem metožu izpētē:

- izpētīta RBF neironu tīkla piemērotība likumu iegūšanā no apmācīta neironu tīkla;
- realizēta likumu ieguves procedūra no RBF neironu tīkla;
- izpētīta parametra  $S$  ietekme uz likumu kvalitāti;
- metodes darbība pielietota praktisku uzdevumu risināšanā.

5) asociāciju analīzes izmantošanā:

- apskatīti un izpētīti asociāciju likumu ieguves pamat algoritmi;
- izpētīta asociāciju likumu ticamības un atbalsta vērtību ietekme uz likumu ieguves procesu;
- izstrādāti principi un metodoloģija likumu iegūšanai ar asociāciju likumu metodes palīdzību;
- izstrādāts programnodrošinājums asociāciju likumu iegūšanai no datiem;
- metodes darbība pielietota praktisku uzdevumu risināšanā statistisko datu apstrādē.

Darba izstrādāšanas gaitā tika secināts, ka ar nosacīto IF-THEN likumu palīdzību var veikt klasifikācijas, tēlu atpazīšanas, asociāciju iegūšanas u.c. uzdevumus. Šīs metodes var pielietot



dažādām datu izlasēm Daudzie klasterizācijas algoritmi pārsvarā tiek izmantoti vai nu patstāvīgi, vai arī neironu tīklu apmācības datu sagatavošanas etapā.

Motivācija likumu ieguves metožu izpētei bija šāda:

- analizējamo daudzdimensiju datu apjoms kļūst pārāk liels klasiskās statistiskās analīzes iespējām;
- iepriekš nezināmu likumsakarību atrašana datos;
- iespēja izteikt atrastās likumsakarības lietotājam uztveramā un saprotamā veidā.

Asociāciju likumu iegūšanas metodi ieteicams pielietot gadījumos, kad dati ir veseli skaitļi un datos var pamanīt atkārtojošos elementus.

Metode uz RBF tīkla pamata un izplūdušo likumu metode ir līdzvērtīgas. To darbības rezultātus var salīdzināt tikai pēc iegūto likumu „kvalitātes” rādītāju pārbaudes, t.i., izskaitļojot likumus pareizi aprakstošo elementu skaitu.

Darba nobeigumā tika iegūta pārliecība par to, ka dažādās likumu ieguves metodes spēj iegūt likumus no daudzdimensiju datiem un šie iegūtie likumi ir kvalitatīvi.

**Tālāko pētījumu virziens** tiks orientēts uz klasterizācijas metožu izpēti un likumu iegūšanu no neironu tīkliem.

Interesi rada izplūdušās klasterizācijas modeli, kuros klasterim ir elipsoīda forma. Tas dod iespēju precīzāk aprakstīt klasterus un samazināt informācijas zudumus.

Tiek plānots turpināt pētīt likumu iegūšanas metodes no neironu tīkliem. Galvenā uzmanība tiks veltīta RBF neironu tīklam.

Praktisku interesi var izraisīt programmnodrošinājuma izstrādāšana likumu iegūšanas metožu pielietošanai tautsaimniecības uzdevumu risināšanā.

## LITERATŪRA

1. Ackley D.H., Hinton G.E., Sejnowski T.J. A learning algorithm for Boltzmann machines// *Cognitive Science*. - 1985. -9. -p. 147-169.
2. Aggarwal C.C., Procopius C, Yu P.S. Finding localized associations in market basket data// *IEEE Transactions on Knowledge and Data Engineering*. - 2002. -vol. 14. - p. 51-62.
3. Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases//*Proc. Conf. on Management of data*. - ACM Press, 1993. -p. 207-216.
4. Agrawal R., Srikant R. Fast algorithms for mining association rules// *Proceedings of the 20th International conference on Very Large Databases*. - Chile., 1994. - p. 487-499.
5. Altman E. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy// *Journal of Finance*. - 1968. -vol. 13. -p. 589-609.
6. Andrews R., Diederich J., Tickle A. A survey and critique of techniques for extracting rules from trained artificial neural networks// *Knowledge-Based Systems*. -1995. -8(6). -p. 373-389.
7. Andrews R., Gawa S. Rule extraction from a constrained error back propagation MLP// *Proc. 5th Australian Conference on Neural Networks*. -Brisbane, 1994. -p. 9-12.
8. Andrews R., Gawa S. RULEX and CEBP networks as the basic for a rule refinement system// *Hybrid Problems, Hybrid Solutions*. - 1995. -p. 1-12.
9. Back B., Laitinen T., Sere K. Neural networks and bankruptcy prediction: funds flow, accrual ratios and accounting data// *Advances in Accounting*. -1996. -14. -p. 23-37.
10. Beaver R. Financial ratios as predictors of failure, *Empirical Research in Accounting: Selected Studies//J.Accounting Research*. - 1966. - vol. 4. -p. 71-111.
11. Bishop CM. *Neural networks for pattern recognition*. - Oxford.Oxford University Press, 1995.- 504 p.
12. Chiu S. Fuzzy Model Identification Based on Cluster Estimation// *Journal of Intelligent and Fuzzy Systems*. - 1994. -2. - p. 267-278.
13. Coughlin J.P., Baran R.H. *Neural Computation in Hopfield Networks and Boltzmann Machines*. - Newark: University of Delaware Press, 1995. - 281 p.
14. Craven M., Shavlik J. Extracting tree-structured representations of trained networks// *Advances in Neural Information Processing Systems*. -1995. - vol. 8. -p. 24-30.
15. Duch W., Setiono R., Zurada J.M. Computational Intelligence Methods for Rule-based Data Understanding// *Proceedings of IEEE*. - 2004. - 92(5). -p. 771-805.
16. Everitt B.S. *Cluster analysis*. - London:Edward Arnold, 1993. -170 p.
17. Fausett L. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. - New York: Prentice Hall International Inc., 1994. - 461 p.
18. Fayyad U., Piatetsky-Shapiro G., Smyth P. From data mining to knowledge discovery in databases//*AI Magazine*. - 1996. - 17 (3), -p. 37-54.
19. *Fuzzy Cluster Analysis/ Hoppner F., Klawonn F, Kruse R. etc.* - England: John Wiley and Sons, 1999. -289 p.
20. Giudici P. *Applied Data Mining: Statistical Methods for Business and Industry*. - John Wiley & Sons Ltd, 2003. - 359p.

21. Grabusts P. *Extracting rules from trained RBF neural networks// Proc. of the 5th International Conference. - Rēzekne: RA izdevniecība, 2005. - p.333-39.*
22. Grabusts P. *Analysing Bankruptcy Data with Neural Networks//Proc. 10<sup>th</sup> International Conference on Soft Computing. -Brno, Czech Republic, 2004. -f. 111-117.*
23. Grabusts P. *Using Association Rules to Extract Regularities from Data// Proc. 6<sup>th</sup> International Baltic Conference on Data Bases and Information. Systems. -Rīga, 2004. -p. 117-126.*
24. Grabusts P. *Asociāciju likumu analīzes izmantošana likumsakarību meklēšanā// RTU Zinātniskā Raksti: Datorzinātne. Datorvadības Tehnoloģijas. -Rīga: RTU izdevniecība, 2003. - 15.sējums. -110.-118. lpp.*
25. Grabust P., Borisov A. *Using grid-clustering methods in data classification// Proceedings of the International Conference on Parallel Computing & Electrical Engineering-PARELEC'2002. - Warsaw, Poland, 2002 -p. 425-426.*
26. Grabusts P., Borisovs A. *RBF neironu tīklu pielietojanas perspektīvas// II. Pasaules latviešu zinātnieku kongresa tēžu krājums. - Rīga, 2001. - 570. lpp.*
27. Grabusts P. *Klasterizācijas metodes izmantošana RBF neironu tīklos// III Starptautiskās zinātniskās konferences „ Vide. Tehnoloģija. Resursi” materiāli. -Rēzekne: RA izdevniecība, 2001. - 257.-262. lpp.*
28. Grabusts P. *A study of clustering algorithm application in RBF neural networks// Scientific Proceedings of Riga Technical University: Computer Science. Information Technology and Management Science. - Riga: RTU, 2001. -vol. 5. -p. 50-57.*
29. Grabusts P. *Solving TSP using classical simulated annealing method// Scientific Proceedings of Riga Technical University: Computer Science. Information Technology and Management Science. - Riga: RTU, 2000. - vol. 2. - p. 32-39.*
30. Grabust P. *Using a thermal equilibrium method in the neural networks// International Conference on Parallel Computing & Electrical Engineering-PARELEC'98. - Bialystok, Poland, 1998. -p. 261-263.*
31. Grabusts P. *A thermal equilibrium method for Boltzmann machines// Modern Aspects of Management Science. - Riga: RTU, 1998. - Nr. 2. -p. 13-19.*
32. Hopfield J. *Neural networks and physical systems with emergent collective computational abilities//Proc. Natl.Acad.Sci. - USA, 1982. -p. 2554-2558.*
33. Hush D.R., Home B.G. *Progress in Supervised Neural Networks. What's new since Lippmann?//IEEE Signal Processing Magazine. -1993. - vol. 10,-No 1. -p. 32-43.*
34. *Izplūdušā loģika, iespējamību teorija un to pielietojumi: Metodiskais līdzeklis/ A.Borisovs, L.Dubrovskis, L.Aleksejeva u.c. - Riga: RTU, 1995. -136 lpp.*
35. Jang J.S., Sun C.T., Mizutani E. *Neuro-Fuzzy and Soft Computing. - New Jersey: Prentice Hall, 1997. - 614p.*
36. Kirkpatrick S., Gelatt C.D., Vecchi M.P. *Optimization by simulated annealing//Science. -1983. -220. -p. 671-680.*
37. Laarhoven P.J., Aarts E.H. *Simulated Annealing: Theory and Applications. - Holland: D.Reidel Publishing Company, 1987. - 204p.*
38. McCarty K., Wermter S., Macintyre J. *The extraction and comparison of knowledge from local function networks// International Journal of Computational Intelligence and Applications. - 2001. - 1(3). -p. 369-382.*

39. *Mākslīgie neironu tīkli: arhitektūra, atgoritmi un pielietojumi: Mācību līdzeklis/ A.Borisovs, L.Dubrovskis, G.Kulesova u.c. - Rīga: RTV, 1998 - 110 lpp.*
40. *Newquist H.P. Data Mining: the AI metamorphosis// Database programming and design. - 1997. - Nr. 9.*
41. *Odom M., Sharda R. A neural network model for bankruptcy prediction// Proc. Int. Joint Conf. Neural Networks. - San Diego, CA, 1990. -p. 163-168.*
42. *Roiger R.J., Geatz M.W. Data mining. A tutorial-based primer. - Addison Wesley, 2003. -364 p.*
43. *Rudorfer G. Early bankruptcy detecting using neural networks// APL Quote Quad. -1995. -vol. 25, N. 4. -p. 171-178.*
44. *Rumelhart D.E., Hinton G.E., Williams R.J. Learning representations by backpropagating errors//Nature. - 1986. -323(9). -p. 533-536.*
45. *Russel S., Norvig P. Artificial Intelligence: A Modern Approach. - Englewood Cliffs, NJ: Prentice Hall, 1995. - 932 p.*
46. *Saito K., Nakano R. Medical diagnostic expert system based on PDP model// Proceedings of the IEEE International conference on Neural Networks - IEEE Press, 1988.-p. 255-262.*
47. *Schikuta E. Grid-Clustering: A fast hierarchical clustering method for very large data sets//Proc. 13th Int.Conf on Pattern Recognition. - 1996. -p. 101-105.*
48. *Software for data mining - Ghost Miner/ Internet. - <http://www.faspl.com.pl/?a=product list&id=1>*
49. *Software for statistical analysis - SPSS/ Internet. - <http://www.spss.com/>*
50. *Software for technical computing - Matlab/ Internet. - <http://www.mathworks.com/>*
51. *Srikant R., Agrawal R. Mining quantitative association rules in large relational tables// Proc. Int. Conf. Management of Data. - ACM Press, 1996. -p. 1-12.*
52. *Tarn K., Kiang M. Managerial applications of the neural networks: The case of bank failure predictions// Management Science. - 1992. - vol. 38. -p. 416-430.*
53. *Tan P.N., Steinbach M., Kumar V. Introduction to Data Mining. - Pearson Education, Inc., 2006. - 769p.*
54. *Thrun S. Extracting Rules from Artificial Neural Networks with Distributed representations// Advances in Neural Information Processing Systems. - 1995. - vol. 7. -p. 505-512.*
55. *Towell G., Shavlik J. Extracting of refined rules from Knowledge-based neural networks//Machine Learning. - 1993. - vol. 13, No 1. -p. 71-101.*
56. *Towell G., Shavlik J., Noordwier M. Refinement of approximate domain theory by Knowledge-based Neural Networks// Eight National conference on Artificial Intelligence. - 1990. -p. 861-866.*
57. *Tresp V., Hollatz J., Ahmad S. Network structuring using Rule -based knowledge// Advances in Neural Information Processing Systems. - 1993. - vol. 4. -p. 871-878.*
58. *XJCI Machine learning repository/ Internet. - <http://www.ics.iici.edu/~mlearn/MLSummary.html>*
59. *Wang L.X., Mendel J.M. Generating fuzzy rules by learning from examples// IEEE Transactions on Systems, Man, and Cybernetics. - 1992. - vol. 22, nr 6. -p. 1414-1427.*

60. Wielinga B.J., Schreiber A. Th., Breuker J.A. *KADS: a modelling approach to knowledge engineering//Knowledge Acquisition*- - 1992. - 4(1). -p. 5-53.
61. "Iaden LA. *Fuzzy Sets, Fuzzy Logic and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh//Advances in Fuzzy Systems - Applications and Theory*. - 1996. - Vol 6. - 826 p
62. Zhou Z.H. *Rule extracting: using neural networks or for neural networks?// Journal of Computer Science and Technology*. -2004. - 19(2). -p. 249-253.
63. Zhou Z.H., Jiang y., Chen S.F. *Extracting symbolic rules from trained neural network ensembles//AI Communications*. -2003. - 16(1). -p. 3-15.
  
64. Дюк В., Самойленко А. *Data Mining*. -Санкт-Петербург: Питер, 2001. - 366 с.
65. Забродин В.Ю. *Проблемы классификации (обзор)//ИТИ, 1980. - Сер. 2. - с. 35 -40.*
66. Загоруйко Н.Г., Елкина В.П., Лбов ГС. *Алгоритмы обнаружения эмпирических закономерностей*. - Новосибирск: Наука, 1985. -110 с.
67. Рутковская Д., Пилиньский М., Рутковский Л. *Нейронные сети, генетические алгоритмы и нечеткие системы*. - Москва: Горячая линия-Телеком, 2004. - 383 с.
68. Хорошевский В.Ф., Гаврилова Т.А. *Базы знаний интеллектуальных систем*. -Санкт-Петербург: Питер, 2001. - 382 с.